TCAM:

TOUCH DRIVEN COMMUNICATION FOR VIRTUAL MOVIEMAKING

by

Emily Cairns Duff

A Thesis Presented to the
FACULTY OF THE USC SCHOOL OF CINEMATIC ARTS
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF FINE ARTS
(INTERACTIVE MEDIA)

May 2011

Acknowledgements

A simple thank you is not enough to convey my gratitude to all of the individuals who have helped me bring this project to life.

Thank you to my advisors Mark Bolas, Jeremy Gibson, Perry Hoberman, Andreas Kratky, and Bei Yang for your guidance, insight and genuine interest in my work. I would also like to thank the IMD faculty and my classmates, now colleagues, for your support, feedback, and inspiration over the past three years on the myriad of projects we have taken on together. Our combined wisdom has been put to good use.

Thank you to everyone that has been involved in the conceptualization, development, and debugging of TCam: John Brennan, Kevin Lipnos, Bradley Newman, Logan Olson, Sean Plott, Evan Suma and Origami Digital: Oliver Hotz, Christian Aubert, Matt Rubin, and Koichi Minamizuka. I finally understand gimbal lock. Thank you to USC Institute for Creative Technology, PhaseSpace and Microsoft Research for providing the resources to make this research possible.

Thank you to my parents Alfred and Barbara Duff for your unwavering support from day one, and to Nick Bartel for always believing in me.

And finally thank you to Eric Furie for introducing me to motion capture and James Knight for the opportunity to experience it first hand.

Table of Contents

Title	Page
Acknowledgements	ii
List of Figures	iv
Abstract	vi
Keywords	vi
Introduction	1
Rational for this Effort	4
Project Description	6
Motivation and Goals	11
Configuration	12
User Experience	14
Modification Toolset	17
Camera Toolset	24
Form Factor Adaptability	26
Current Status and Next Steps	27
Conclusion	28
Glossary	31
Bibliography	32
Appendix A: Parallel Efforts and Sources of Inspiration	34
Appendix B: Prior Work	47

List of Figures

Title	Page
Figure 1: InterSense VCam™	2
Figure 2: NaturalPoint VCS:Pro	2
Figure 3: NaturalPoint VCS:Mini	2
Figure 4: PhaseSpace Virtual Camera	2
Figure 5: TCam Networking and Communication Diagram	12
Figure 6: TCam Prototype	14
Figure 7: Connecting to the System	15
Figure 8: Camera Lock Gesture	15
Figure 9: Animation Lock Gesture	16
Figure 10: Three-finger Menu Access Gesture	17
Figure 11: Disconnecting from the System	17
Figure 12: Light Rotation Gesture	18
Figure 13: Light Intensity Gesture	19
Figure 14: Light Positioning Gesture –X &Y-axis	20
Figure 15: Light Positioning Gesture – Z-axis	20
Figure 16: Object Positioning Gesture – X & Y-axis	22
Figure 17: Object Positioning Gesture – Z-axis	22
Figure 18: Object Rotation Gesture	23
Figure 19: Object Scaling Gesture	23
Figure 20: Focal Length Slider	24

Figure 21: Tracking Volume Diagram 25

Figure 22: Motion Recording Toggle 26

Figure 23: *Avatar* Virtual Camera 34

Figure 24: PhaseSpace Rigid Body 35

Figure 25: InterSense Virtual Camera 35

Figure 26: NaturalPoint Virtual Cameras 36

Figure 27: PhaseSpace Virtual Camera 37

Abstract

As the use of motion capture expands rapidly across the cinematic industry, an increasing amount of focus is being placed on the tools and techniques of virtual moviemaking. Of these tools, one of the more popular is the virtual camera and its ability to compose shots within a virtual environment. TCam, a combination of tablet and camera, extends the existing virtual camera concept to a wireless multi-touch tablet device, utilizing a game engine to render the virtual environment and animation in real-time. Within this environment, a user can create, modify and remove elements directly composing the shot as he or she envisions. Artists, directors and cinematographers can quickly manipulate environmental setups and explore a variety of camera moves with the help of TCam. The beauty behind TCam lies in its ability to allow a variety of artists to communicate complicated ideas through a series of multi-touch gestures. While TCam can provide many creative tools, it should be thought of more as a customizable communication platform rather than a specific tool set. This paper details the design and construction of a new type of virtual camera for motion capture which allows a user to make real-time in-camera modifications to a virtual environment for both previsualization and production purposes.

Keywords

Communication, Collaboration, Game Engine, Interactive, Motion Capture, Multi-Touch, Real-Time, Tool, Previsualization, Virtual Camera, Virtual Moviemaking

Introduction

Cameras provide a window into another world. Whether it is a candid snapshot of a sunset or a large budget action scene, cameras allow us to capture a moment in time as it is happening. A virtual camera provides users the same capabilities a film camera provides in the real world, but inside of a virtual environment. In 3D modeling and animation programs such as Autodesk® Maya®, Autodesk® 3ds Max® or NewTek LightWave3D®, a user can position a camera to cover a specific shot angle or perspective in a virtual space. Artists can also animate a camera along a motion path to mimic live-action camera moves. Additionally, multiple virtual cameras can be configured and placed within a scene, working simultaneously or independently from one another. Since the cameras are hidden by default, a user does not have to be concerned with additional cameras accidentally appearing within a shot, as her or she would on a live action set.

In motion capture, a virtual camera behaves in a similar manner as it does in animation software, however instead of being driven physically via mouse and keyboard input, the camera is driven virtually by tracking a tangible object within a motion capture *volume*. The tracked object can range from an abstract Tinkertoy™ creation to a highly functional production grade video camera. To track the object within the *volume*, markers are affixed directly to the object or attached to a *rigid body*, which is then mounted on the object. Essentially, as long as an object is able to support and present a trackable *rigid body*, it can serve as a virtual camera.

   

A variety of consumer virtual cameras, from left: Figure 1: Intersense VCam™. Figure 2: NaturalPoint VCS: Pro. Figure 3: NaturalPoint VCS:Mini. Figure 4: PhaseSpace customized HP® TouchSmart.

As it currently exists in many production environments today, the virtual camera has taken two representative forms. The first is a production video camera, either hand-held or mounted, with a raised *rigid body* attached to it. The raised *rigid body* helps to better present the markers to the tracking system, while the camera itself provides a familiar form factor and weight for a user to work with. Customizable cameras can also be configured to map camera controls to external program controls in Autodesk® MotionBuilder®, giving the user the direct control of recording and playback functionality. MotionBuilder® is currently considered an industry standard 3D software application for motion capture that allows users to work with both real-time and pre-recorded motion capture data. Additionally, a real-time video feed can be configured to play back through the camera's viewfinder or attached external monitor, providing the user a direct view into the virtual world from the perspective of the camera.

The second approach to a virtual camera takes a less traditional form by using just a LCD monitor in conjunction with joysticks for precision movement controls. The monitor can be attached to a shoulder or chest rig like a Steadicam, or can be held by the user without additional assistance. The larger monitor allows for better viewing of real-time playback, and the input controls allow an operator to duplicate almost any type of camera move, such as a crane shot, without having to physically move. This lightweight and smaller design allows the user to "block out" shots from angles where a large film camera rig may not otherwise fit. This design is also useful when working within a motion capture *volume*, as a film camera may not be necessary or readily available.

Regardless of design, the objective of the virtual camera still remains the same, to replicate and extend the functionality of a real camera, providing the director creative options far surpassing the limitations of a real camera (Autodesk 8). While these two forms of virtual cameras are the most commonly seen in a production environment, one of the significant advantages of a virtual camera is that it can take any customizable form factor.

As the interest in virtual camera technology builds, the field of motion capture continues to evolve technologically. Motion capture is becoming a more viable component in the virtual production pipeline as it continues to increase in both accuracy and flexibility, and decrease in price (Autodesk 8). One of the areas where motion capture is appearing more rapidly is the previsualization (previs) process. The main purposes of the previs process are to visually explore creative ideas, plan technical solutions and communicate a shared vision to the production team. Currently, previs is done predominately with the use of 3D animations and virtual environments (The Previsualization Society). As the results of motion capture become easier and faster to use, many studios, such as the Los Angeles based previs and visual effects house The Pixel Liberation Front (PLF), are turning to the technology for use in previs. PLF's Executive Producer Sean Cushing describes their use of motion capture and how it relates to the previs process:

> Previs sequences are more and more looked to as tools to work out emotional content and overall storytelling. Therefore, the performance of the previs characters has had to improve dramatically over the years. However, better character animation takes time. Previs has to be fast and interactive. The quality and emotional impact are at odds with the process and the needs of the production schedule. (Peszko 3)

Real-time previs provides remarkable advantages for shot set-up and composition that cannot be matched on a live-action set or created in reality, including the ability to try multiple lighting configurations with a variety of camera angles and lenses with little prep or down time. Previs provides a solution for the need to communicate complex virtual camera moves that cannot be created in the real world.

Rational for this Effort

This thesis explores the concept of communication and how it can be better augmented with multi-touch input. If broken down into its smallest pieces, communication can be considered a design practicum, a tool set, or even a networked system. More importantly, communication is a tool for sharing an idea across a network of individuals in a universally understood language.

TCam is a virtual camera with additional capabilities that allow a user to manipulate a virtual environment in real-time. It uses a multi-touch interface to quickly communicate an idea visually, allowing users to more quickly and efficiently share their ideas. Now complex ideas that are difficult to explain verbally can be done so visually, in a manner similar to loose sketches or jotting ideas on a napkin. The intended user ranges from a director or cinematographer to an effects supervisor or previs artist; essentially any member of the production pipeline familiar with the process of composing for and working within a virtual environment.

With TCam, a user can move around a motion capture *volume* using the device as a virtual camera or an environment-sketching tool. He or she has the capability to directly control various camera functions including recording capabilities, lens modifications, and single axis constraints for pan, tilt, and roll. While using TCam as a virtual camera, should the user want to modify an element in the environment, he or she can lock the camera, freezing the current shot framing, and use the modification tools to begin manipulating the element directly. Modification capabilities include moving, scaling, and rotating elements within the environment as well as adjusting lighting setups. After making modifications, the user can unlock the camera and continue to compose shots within the newly designed space. The modifications made are saved in alternative scene files for future reference purposes.

As technology continues to evolve, the multi-touch gestural language becomes more fluent and recognizable on a broad spectrum. Anyone familiar with a multi-touch device can quickly begin using TCam to modify a virtual environment. Complicated lighting and environment setups no longer have to only be explained verbally, but instead can be quickly mocked-up and communicated visually.

Project Description

With a combination of virtual camera features and previsualization theories, TCam uses the power of a game engine to create a modifiable virtual environment. The goal of TCam is to explore the potential of real-time cinematic design, and how this new modality of communication can contribute to the production pipeline.

While a variety of virtual cameras exist on the consumer market, the concept of a virtual camera is still relatively new. Because of this, companies and studios alike are still experimenting with the design, technology and general philosophies of how a virtual camera should look, feel and work. As previously discussed, any object can ultimately be turned into a virtual camera, and with this type of freedom, ergonomic design becomes more of a primary focus, with technological functionality as a close second.

When first thinking about the design of TCam, device weight was one of the most important physical design factors taken into consideration for the camera. When designing tangible interfaces and physical input devices, it is import to not unnecessarily strain or tax the user (Hinckley, Pausch, Goble and Kassell 219). Since the main function of a virtual camera is to be moved through free space for extended periods of time, a relatively simple task quickly becomes tiresome if the camera is too heavy or too awkwardly shaped. Along this same line, the product casing is also important. If the camera is too sleek, too thick, or does not rest nicely on the user's arm, the casing becomes another usability hurdle. The third physical element considered was screen size. Since TCam is meant to function both as a virtual camera and a touch-based editing tool, it is important to work within a larger screen space than that of a touch-enabled smart phone.

Two additional areas were also evaluated in regards to technical capabilities and performance: multi-touch input and development platform flexibility. With the physical design elements taken into consideration, a touch-enabled tablet became the frontrunner in device choices. Several tablet notebook options were ruled out because of the size and placement of the battery packs, which made it difficult to comfortably hold the device with one hand for an extended amount of time. Lightweight LCD touch-enabled monitors were also eliminated because of their unavoidable cording requirements. As of August 2010, the Apple iPad provided the best out-of-the-box solution for the initial design and development requirements: lightweight, large multi-touch screen and customizable casing options. At the time, the iPad also excelled over competing tablets because of its added benefit of supporting multi-touch input for the Unity 3 game engine.

It is important to note that when initial prototyping began, the iPad was thought of more as a toy than an actual business device. However, over the past year, the iPad has quickly become an industry standard as companies move towards a more cloud-based content management solution in areas such as corporate business, construction and even filmmaking. In a recent Apple store interview, *I Am Number Four* director DJ Caruso admitted to first buying his iPad as a toy, but after realizing the power the iPad has as a production tool, he began using it to store his storyboards, previs, call sheets and even scripts (Hinojosa). This same trend of using the iPad as a production tool is also being seen on the set of *The Amazing Spider-Man*. With this taken into account, continuing to develop production tools for a platform that is integrating itself seamlessly into the production pipeline seems even more advantageous than before.

As more and more lightweight multi-touch tablet devices continue to enter the consumer market, TCam is expected to be able to extend across multiple platforms. Much like how the design of current virtual cameras is driven first by comfort and second by technology, TCam is driven first by theory and second by technology. While technology is expected to change, the underlying ideas of TCam will be able to adapt to changing technology.

With the incorporation of multi-touch input, TCam re-conceptualizes the motion capture virtual camera. Existing virtual cameras, if they allow for user input other than movement tracking, often restrain input to button and slider controls. The most dynamic user input comes from cameras that incorporate the use of joysticks to allow for 4 or 6-axis camera controls. With the addition of the joystick, an operator can perform more specialized camera moves that might be otherwise unfeasible. When working with real-time data, it seems only natural to continue to draw from the game industry by pairing the concept of a virtual camera with a more dynamic game engine. Gaming technology continues to push the boundaries of real-time rendering capabilities, providing not only photorealistic results, but also greater interactivity (Autodesk 11). In the early 2000s, Los Angeles based studios such as House of Moves and Halon were already considering the potential of powering real-time motion capture via a game engine, but at that time desired engine customization and flexibility were not yet available (Desowitz; Hetherington). Now, commercially available game engines are more customizable in terms of networking and input capabilities, and some developers, such as Crytek, are even creating game engine tools specifically for moviemaking purposes.

TCam takes this idea one step further and combines it with the familiar gestural interface of a multi-touch device. TCam's multi-touch interface uses a common dictionary of touch gestures through which users can manipulate an environment. By using simple, familiar gestures, the learning curve for TCam is immediately lowered, allowing anyone to quickly pick up the device and begin experimenting with either the environment or the shot. Similar to other expert user applications that have been scaled for mobile use, such as Autodesk® SketchBook® Mobile or Adobe® Photoshop® Express, TCam provides a robust toolset for use in the field. Like these applications, TCam can also integrate smoothly with desktop software for more complicated or time-intensive tasks, such as initial environment design and development.

With the use of a lightweight tablet and multi-touch interface, TCam diverges from other virtual cameras in more than just physical appearance. The included ability to modify a virtual environment in real-time directly within the camera viewer has not yet been fully conceptualized in a production environment by the cinematic industry. On the set of *Avatar*, Rob Powers developed a process called Character Emanation, which consisted of an algorithm auto-generated a virtual environment in real-time based on a motion capture actor's movements (Powers). While it is extremely helpful to generate new environments based off a character's interaction, the ability to edit an environment in real-time allows a director to see how modifications add or detract from a given shot. For example, when developing a video game, changing the height and distance that a character can jump affects the relationship of the character to its surrounding environment; the height of the ceilings and platforms, gap widths and the amount of ground traversed during a jump all affect the dynamics of a game. Similar to game development, small environmental changes can drastically affect a shot. When discussing future virtual camera features, James Cameron stated that he would like to see more shadows: "…the characters don't cast shadows on the ground, and sometimes shadows are a very important part of a scene; you compose the shot to the shadow" (Warner). Being able to see these types of variations in real-time can help to better guide the virtual moviemaking process. More importantly, being able to share detailed images of desired settings or environment modifications directly with anyone involved in the production helps prevent miscommunications.

Motivation and Goals

Every day we communicate hundreds of ideas both verbally and non-verbally. Depending on the context in which these transactions occur, the message can be lost in the delivery. Miscommunications can range from small and inconsequential to largely detrimental in any given situation. As an example, at a restaurant, the difference between the waiter misunderstanding or ignoring “Dressing on the side, please” and “No avocados, I'm allergic” is a few extra calories and some soggy lettuce versus anaphylactic shock and an Emergency Room visit. While miscommunications in the cinematic industry are often not as dire, they can be just as significant in terms of time and monetary losses during a production.

The question then becomes: how do people better communicate with one another? For the purposes of this paper and research, the question has been significantly narrowed down to: how does a person communicate an idea more clearly when working with virtual environments designed for cinematic purposes.

Walt Disney summarized it well when he said that, “Of all of our inventions for mass communication, pictures still speak the most universally understood language.” With the ability to track the camera in the *volume* and project its perspective on a screen, the virtual camera exemplifies Disney’s proclamation. Now the director and actors can see directly into the virtual environment they inhabit. The virtual camera acts as the translation device between the virtual and the real, allowing the human eye to see what exists digitally. This visual articulation allows the director or cinematographer to better frame a shot as if he or she were standing in the real environment or on a live set instead of in a motion capture *volume*.

While the technology behind the virtual camera is essentially limitless, the functionality of the camera itself is restrained to, at best, what can be duplicated by a high-end production camera or a virtual camera in a 3D software package, such as MotionBuilder®. What happens when you explore other ways to build a virtual camera, diverging from the current industry standard? How does a game engine augment the "picture"? Does the incorporation of multi-touch make it easier to operate? TCam is exploring ways to easily make this "picture" more vivid, dynamic and communicative.

Configuration

Prior to using TCam, configuration processes need to take place on both the iPad and the computer running MotionBuilder®. These configurations are performed "behind the scenes" and, while invisible to the overall user experience, are essential to the process.

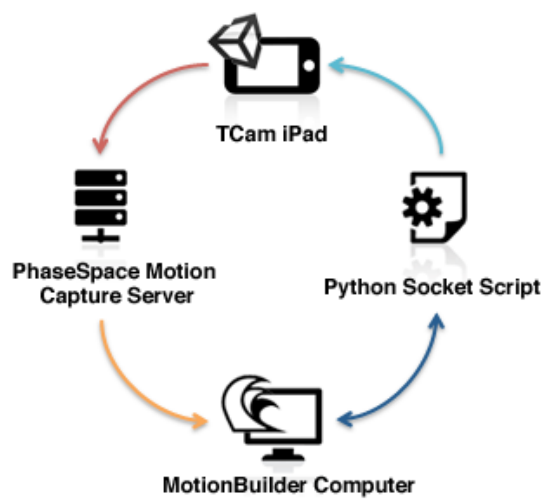

Figure 5: TCam Networking and Communication Diagram

First, the existing virtual environment, or set piece, is loaded into the TCam application along with any desired character animation sequences. These items are then scaled to the desired proportions and oriented to match the *volume*. This process only needs to happen when a new environment is introduced, when the current environment model or character animation sequences are modified external to the application, or when the motion capture *volume* itself is modified significantly.

Next, the virtual camera needs to be configured in MotionBuilder®. This configuration needs to be performed at the start of each day to ensure the iPad is properly tracked by the system and transmitting the correct camera coordinates. From within MotionBuilder®, the tracking markers on the iPad are configured to create a *rigid body* and then constrained to a camera object.

After the *rigid body* is configured in MotionBuilder®, an external Python script is started that creates a Telnet connection with MotionBuilder® and continually requests translation and rotation data for the *rigid body*. Upon receiving this information, the script then passes these coordinates to the iPad via a UDP connection. TCam uses this data to replicate the real world camera rotation and translation with the game engine's virtual camera.

With future advances in technology, it could be possible to receive this tracking data internally from gyroscopes, accelerometers, or inertial measurement units embedded within the device, ultimately eliminating the need for a tracking system. However, the accuracy of such data is not yet reliable enough to depend on, necessitating the use of an external tracking system. Additionally, by using MotionBuilder® as the communicator for receiving the camera position and rotation, TCam does not rely on a specific tracking system. Since MotionBuilder® is an industry standard for motion capture software, tracking systems already provide the plug-ins allowing direct communication between MotionBuilder® and the tracking system. TCam bypasses the need for an additional plug-in by connecting to MotionBuilder® rather than directly to the tracking system. This makes it easy to use TCam between various tracking systems, as only the markers need to be altered. Once the camera and environment are properly configured, TCam is ready to run.

User Experience

TCam's current prototyped feature set was chosen because it was thought to be the most evocative of the new type of technology TCam provides for virtual moviemaking. The current features are by no means an

Figure 6: TCam Prototype

exhaustive list of the capabilities of TCam but rather features that showcase the potential of the device.

Connecting to the System

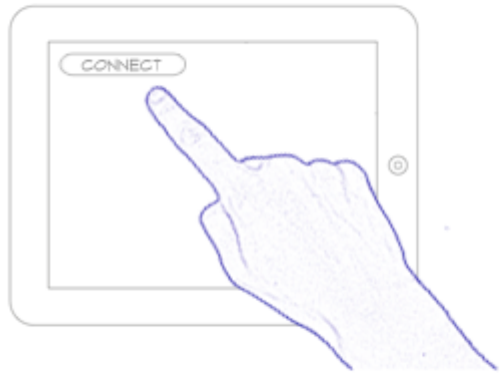

Figure 7: Connecting to the System

When ready to begin, the user connects to the tracking system data stream via the "Connect" button on the TCam interface. Once connected, the "Connect" button becomes the "Disconnect" button to save vital screen real estate. By moving the tracked iPad through the *volume*, the user is able to move the camera around the virtual environment in the TCam application.

Locking the Camera

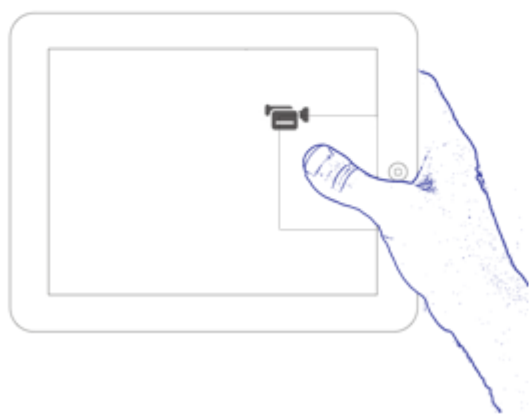
Figure 8: Camera Lock Gesture

At any time, the user can lock the camera tracking and modify the environment. When locking the camera, TCam is still connected to the tracking data stream, but is not utilizing the information. The camera remains fixed at the same position and rotation as when the camera lock was enabled. This allows a user to frame a shot and then make scene modifications within the desired framing. To active the camera lock, a user taps the middle section of the right side of the screen. This area acts as a toggle, locking or unlocking the camera with a quick tap.

Locking the Animation

Similar to locking the camera, a user can start or stop any pre-recorded animations in the scene with the animation lock. The animation lock allows a user to pause the animation at a specific moment and compose the shot around the animation. To activate the animation lock, a user taps the middle section of the left side of the screen. This area also acts as a toggle, starting or stopping the animation within the scene. Both the camera and animation lock areas are designed to be clutching mechanisms, quickly accessible with a user's thumbs. These two features allow for the device to be adjusted in physical space and deliberately affect or not affect its position in virtual space (Hinckley, Pausch, Goble and Kasell 219).

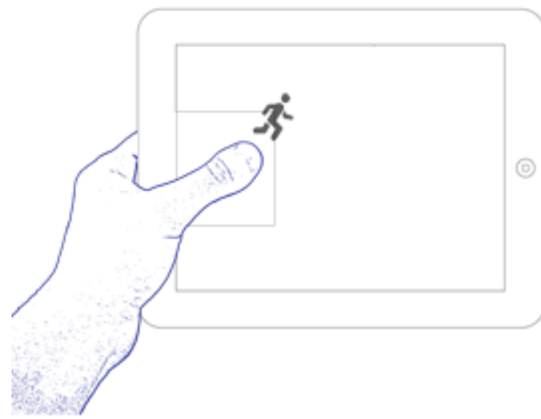

Figure 9: Animation Lock Gesture

Accessing the Tools Menu

A three-finger tap is used to access the tools menu. The menu will open where the user taps, allowing the user to position the menu in a convenient location. A single one-finger tap on screen will close the menu, provided no sub menus are currently open. The menu can also be opened or closed by tapping on a small circular icon located near the bottom of the screen. Both the modification and camera tools are located within the same menu.

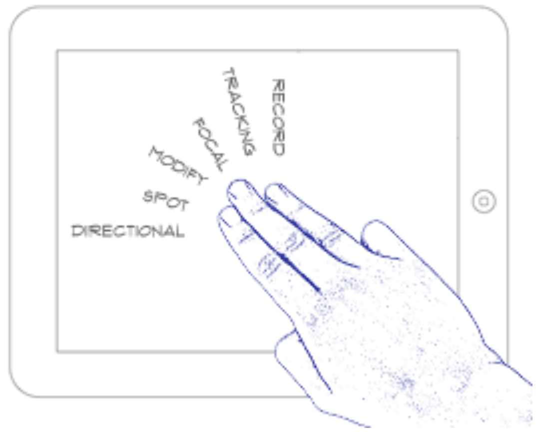

Figure 10: Three-finger Menu Access Gesture

Disconnecting from the System

To disconnect from the system entirely, the user taps on the “Disconnect” button. Disconnecting from the system means TCam is no longer connected to the data stream providing the camera's position and rotation, and the camera is reset back to its default location within the scene. Environmental modifications can still take place while disconnected from the system, but the user no longer has control over the placement of the camera.

Figure 11: Disconnecting from the System

Modification Toolset

Using TCam as a tool for previsualization allows the user to design a virtual environment in real-time, experimenting with the look and feel of each shot. The modification tools allow the user to alter the scene’s lighting as well as the placement and scale of any physical objects within the scene. Both locking functionalities are extremely useful with the modification tools, allowing the user to frame and lock a shot and then continue to alter the environment within that specific framing.

Lighting

TCam currently lets a user manipulate two types of lights: Directional Lights and Spotlights. Additional functionality could also be built to include Point lights. Directional lights affect the global lighting of the scene similar to the sun, while Spotlights affect a much smaller area, highlighting individual objects or specific areas. Individual lights are selected by tapping on the corresponding menu buttons. If multiple lights of the same type exist within the environment, their buttons are named accordingly to reflect the same naming conventions used in the original source file.

Directional Lights

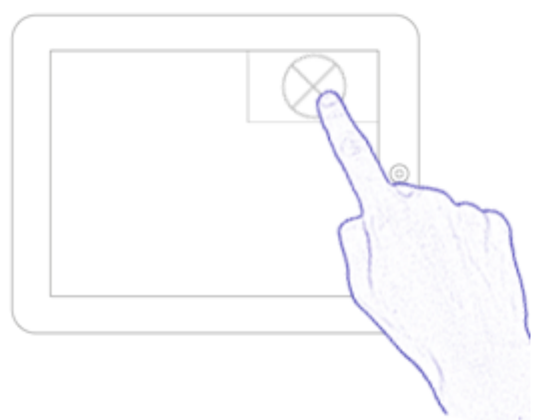

Figure 12: Light Rotation Gesture

In Unity, a directional light affects an environment globally. Only its rotation is important; its position within the environment is neglected. This is similar to film lighting, where a directional light is any light, hard or soft, that affects an area equally and does not spill (Lowel). By tapping on the "Directional Light" button, the user activates the menu for the directional light. This menu provides information on the intensity and rotation of the light, along with the ability to reset the light to its default rotation or remove the light from the scene entirely. To modify the rotation of a directional light, a rotational globe appears in a smaller window on the screen. This globe represents the orientation of the directional light, with an arrow representing the exact direction of the light. The user can control the orientation of the light by rotating the globe with one finger. A two-finger pinch and spread gesture will modify the intensity of the directional light: pinching to lessen, spreading to increase. When satisfied with the modifications, the user taps again on the "Directional Light" button to close the editing menu.

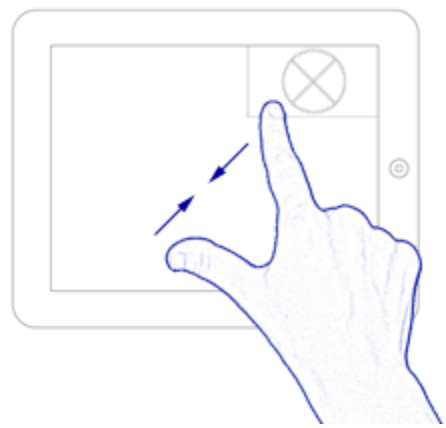

Figure 13: Light Intensity Gesture, pinch and spread to modify light intensity

Spotlights

A spotlight affects a much smaller area of a scene than a directional light and is dependent on both position and rotation of the light. To modify a spotlight, the user taps on the "Spotlight" button, activating the menu and controls. Similar to the directional light, the spotlight menu details the position, rotation and intensity of the light, as well as the ability to reset or remove the light from the scene. The rotation controls for the spotlight function the same as the directional light, with the rotational globe appearing in the corner. Since a spotlight also has a position in space, the main screen controls differ slightly from the directional light. A one-finger touch functions as a broad aiming tool for the spotlight, pointing the light in a specific direction. From here, single finger movement along the X or Y-axis will make fine adjustments to the position of the light along the desired axis. This snapping technique often referred to as "snap-dragging" increases the precision of direct manipulation tasks, such as object placement (Bier and Stone 233-240). A second finger is added to move along the Z-axis; the second finger can remain stationary or move with the first. Research has shown when considering a 3D world in 2D space, moving the finger towards or away from the user is an acceptable, comfortable and preferred touch input for depth control, especially when working on a limited display surface (Martinet, Casiez, Grisoni 5.4). Additionally, like the directional light, the pinch/spread gesture adjusts the spotlight intensity.

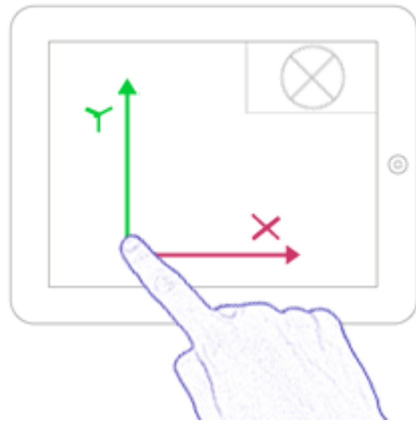

Figure 14: Light Positioning Gesture – X&Y-axis

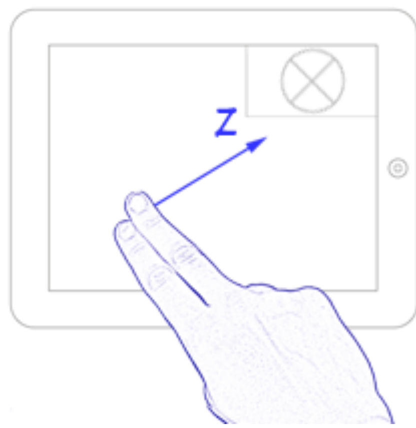

Figure 15: Light Positioning Gesture – Z-axis

Manipulating the Environment

To begin modifying the environment, the user can tap on the "Modify" button. The button allows the elements within the scene to be translated, rotated and scaled. A one-finger touch is used to select elements within the scene. To properly select the correct element, a ray cast is performed in Unity that extends from the camera to a converted point in space where the touch initially occurred. The ray casting technique, by default, will select the foremost object to the camera, which in many circumstances is the desired object (Hinckley, Pausch, Goble and Kassell 217). When selecting an object, the object name is displayed on the screen for additional selection confirmation. Should the wrong object be selected, the user simply has to remove his or her finger and try again.

Scene element manipulation depends in part on how the scene is originally modeled. To be modifiable, elements must exist independently within the scene. Multiple elements built from the same mesh will be selectable, but only as a whole mesh. For example, if the virtual environment is a city, and all of the buildings are part of the same, singular mesh, the city will translate and scale as a whole city. If the buildings are created out of individual meshes, each building can translate and scale independently of the others in the scene. Additionally, not all objects in a virtual environment need to be selectable. Colliders can be quickly removed from stationary elements, such as the ground plane, during the initial configuration phase, to avoid accidental selection.

Positioning, rotating, and scaling features are activated independently via menu button selection. The decision to use a menu style selection for these features helps to separate the functions from one another, which increases the ease of performance and accuracy of each task (Hinckley, Pausch, Goble and Kassell 216).

Repositioning Elements

To begin repositioning elements in the environment, the user taps on the "Position" button to activate positioning mode. A user can move an element around the environment by selecting the object with a one-finger touch. Similar to the fine positioning of the spotlight, a one-finger touch will move the object along the X or Y-axis. A second finger will activate Z-axis movement. When repositioning lights or scene elements, motion towards the right or top of the screen will move the object further from the camera, while motion towards the left or lower portion of the screen will move the object closer to the camera. As Anthony Martinet, Gery Casiez, and Laurent Grisoni's experiments showed, not all users think in these spatial confines, thus these parameters can easily be modified in the initial TCam configuration phase to suit a user's preference (5.4).

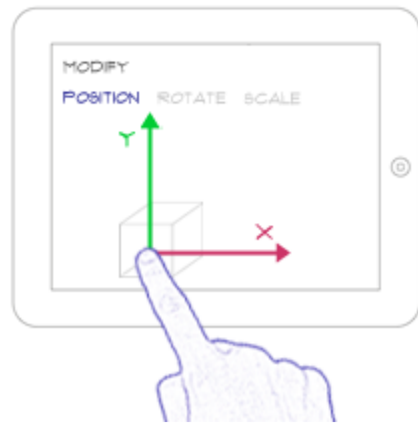

Figure 16: Object Positioning Gesture – X&Y-axis

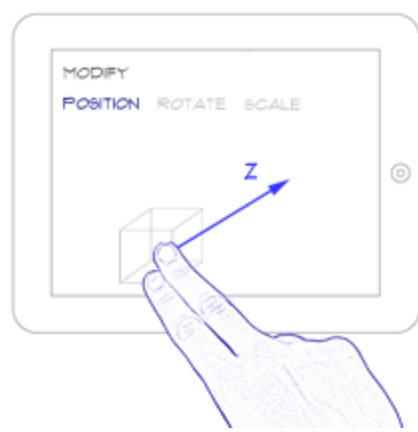

Figure 17: Object Positioning Gesture – Z-axis

Rotating Elements

A user can also rotate an object with one finger by tapping on the “Rotate” button. Similar to how lights are rotated with the rotation globe, a user can rotate an object by dragging one finger across the object to rotate it. A two-finger twist gesture can be used to rotate the object along the axis most parallel to the camera.

Figure 18: Object Rotation Gesture

Scaling Elements

TCam recognizes the well-known pinch and spread convention for scaling purposes. Similar to how the pinch and spread scales the intensity of the lights in the scene, it can also scale the size of the objects in the scene. To select an object to scale, place two fingers on the object. Expanding the two fingers will scale the object larger; pinching the fingers will scale the object smaller.

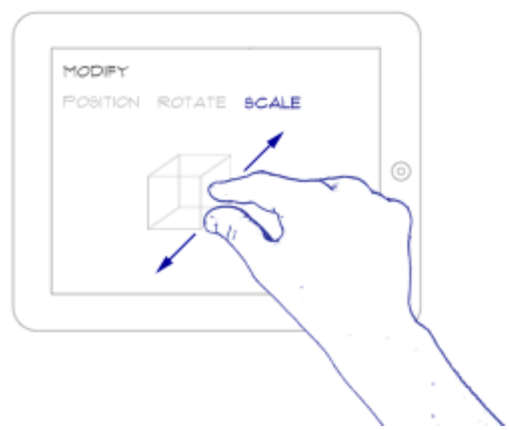

Figure 19: Object Scaling Gesture, pinch and spread to scale object

Exporting Scene Configuration

Once the user is satisfied with the scene configuration, he or she can use the iPad's built-in Screen Grab feature to save a reference image of the screen. Simultaneously holding the power button and the home button will activate the feature and the screen will flash as an image of the screen is stored in the iPad's photo gallery for future reference. It is also possible to save out XML data of the environmental settings to give 3D artists as additional reference materials. This allows for easier recreation of the modified environment in other 3D applications.

Camera Toolset

TCam's camera functionality is also part of the tools menu. Camera tools can be modified in real-time, allowing the user to see his or her changes reflected immediately.

Focal Length

TCam allows a user to change the camera lens on the fly via a slider-like input. By tapping on the "Focal Length" button, the main screen acts as an invisible slider. The user can swipe his or her finger to the left or right to adjust the focal length. Numerical feedback is provided for fine-tuning a selection. This gesture allows a

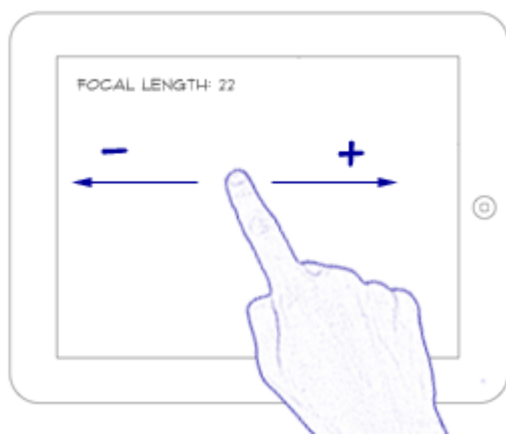

Figure 20: Focal Length Slider

user to quickly swap lenses, focusing on the real-time results and not the interaction with the slider.

Tracking Volume Scaling

With the ability to scale individual objects within the environment, the camera's relationship to the environment must also be capable of scaling to account for any changes. Sliders allow a user to scale the relationship between the camera and the virtual *tracking volume*. The *tracking volume* is the area in the virtual environment relative to the where the camera is moving in physical space. Perhaps the shot needed is a point of view shot from the ground up, such as that of a child or animal. In this case, the *tracking volume* would be positioned low to the ground. A larger *tracking volume* would be well suited for a crane or aerial shots. Figure 21 compares two tracking volumes, denoted in red on the left images.

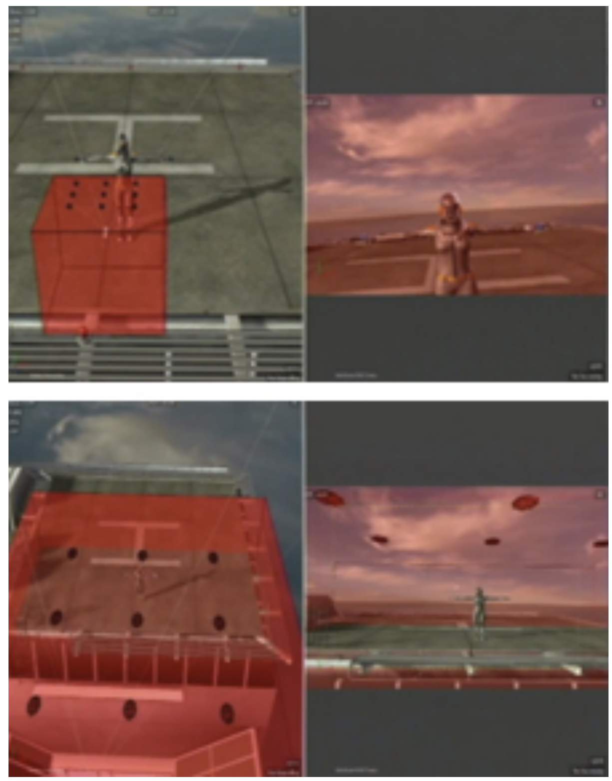

Figure 21: Two different sized tracking volumes and their respective virtual camera perspectives.
Source: InterSense

Local Recording Capabilities

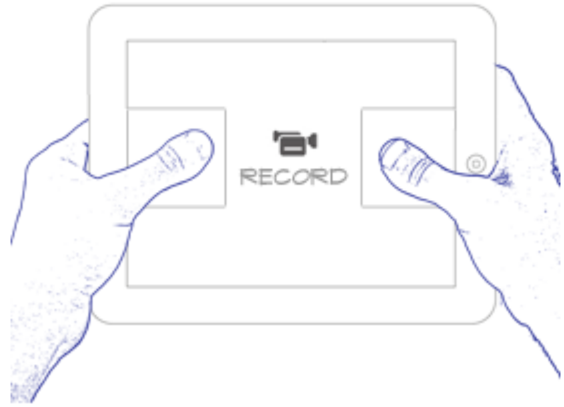

Figure 22: Motion Recording Toggle Gesture

Not all virtual cameras provide the user with the direct ability to record and stop playback locally, leaving the responsibility of starting and stopping data recording to another operator. TCam incorporates the recording start and stop controls into the device interface. Selecting "Record" from the tools menu activates the Record mode. When in Record mode, simultaneously tap the right and left lock buttons, with both thumbs, will start or stop data recording. Similar to the InterSense VCam™, the buttons are mapped directly to MotionBuilder® controls, allowing the user to independently operate the virtual camera.

Form Factor Adaptability

In its current form, TCam can exist alone as a fully functioning virtual camera, or if a user prefers to work with the weight of a hand-held camera or with the support of a shoulder mount or Steadicam vest, TCam can be easily adapted to meet these requirements. For example, additional mounts or attachments can be built to add the iPad to an existing camera rig. By attaching the iPad to a film camera, in place of an additional LCD monitor, the user now has a desirably weighted object, along with the TCam functionality. Since TCam is a wireless device, it can be attached to any type of rig without worry about where to wrap wires or run cables.

Current Status and Next Steps

Of the many individual features TCam is capable of providing, the most important is the overarching technological proof of concept. By developing a functional and varied toolset, one can begin to explore the potential of the device. As previously mentioned, TCam's current feature set is not only built from elements that showcase the potential of the device, but is also built from features that are currently used and sought after in existing virtual cameras. By providing features that encompass the broad capabilities of TCam, discussions of an even more dynamic tool set are encouraged: what is useful to have, what is not, and what is absolutely essential in the design of a virtual camera.

Perhaps, in the future, TCam could be useful in the process of better defining what a virtual camera is and how it can be better used within the cinematic industry. As the industry itself moves towards incorporating more elements of virtual moviemaking into the production processes, the need for such hybrid devices becomes more apparent. One of the next steps for TCam would be to develop a more accurate, marker-less tracking method to generate the camera data. The inclusion of a device such as an inertial measurement unit (IMU) would allow the camera tracking functionality of TCam to exist outside of the space of a motion capture *volume*. Camera moves and shot framing could then be done in a cubicle, living room or even outside, making the virtual camera an even more ubiquitous piece of technology. Additionally, re-evaluating the development platform could potentially lead to the creation of a more dynamic feature set. As game engines continually increase performance on mobile devices, the possibility of higher quality renders and faster processing performance means TCam could handle more complex models, graphics and virtual environments. TCam could render effects such as particles, fog, smoke, wind or even soft shadows more realistically. Lastly, as 3D multi-touch interaction techniques become more standardized and practiced, more and more people will become familiarized with the gestural language, helping to further the production benefits of TCam. The combination of these advancements leads to the possibility of creating a more powerful and realistic real-time virtual camera that fits into the ever-growing mobile production toolkit.

Conclusion

How does one communicate a complicated concept? A lecture. A diagram. An experiment. In the Sciences, complex theories and principles are often accepted as definitive truths because they have been proven. In the Arts, there is more room for personal opinions, visions and open-ended interpretations. Definitive truths are not as easy to find, leaving it difficult to verbally describe detailed scenes and specific style decisions. When working within a virtual environment, theses hurdles are combined not only with the limited ability to physically replicate the world, but also with a difficulty in experiencing and viewing that world.

Scaled models are often made to replicate the virtual film world. Tiny miniaturizations of large cities allow directors, cinematographers and actors alike to better envision the world they inhabit. For example, director Robert Zemeckis used miniature models and detailed reference imagery to help his actors better immerse themselves in the world of the film (Beowulf). In addition to miniatures, detailed set pieces and costume extensions assist in better communicating the characters and virtual environment to the actors. On *The Polar Express*, Zemeckis had large-scale set pieces built to replicate train seats in order to help the adult actors perform better as small children (The Polar Express). A similar approach was taken on *Avatar*, as the motion capture actors' headsets were equipped with Na'vi style ears and hair extensions (Avatar). While set and costume extensions help the actors better embody their virtual characters, the virtual camera becomes the true window into the virtual environment, allowing viewers to see the exact relationship of the characters to the environment, their interactions with the virtual environment and how the shots are framed. However, with all of these solutions, the virtual environment can only be seen in one instance, one specific replication. Unlike a live action set, the ability to make real-time adjustments to the set dressing and lighting is removed. With new technological capabilities, virtual cameras adversely create a new communication challenge in the production pipeline. Since a user can see and interact better with the virtual environment, he or she soon wants to begin making staging changes to that virtual environment. Making quick set dressing or lighting changes now requires a chain of communication down to the artist who created the original environment. With everything perfectly in place and change requests remaining intact as they are transferred from person to person, there is still a delay in turnaround time and a potential gap in productivity.

By driving the virtual camera with a game engine, as opposed to a software suite, the virtual environment comes to life. It now has the added benefit of being directly manipulated in real-time, much like simulation games, and the effects of these modifications are seen immediately. No longer does a request need to pass through multiple people to change the layout of an environment, or extend over several days to see the results of new camera work (Duncan and Fitzpatrick 43). The general functionality and features behind a game engine afford the user the ability to modify the scene as they desire and see a relatively realistic real-time render of how their changes affect a shot. These changes can then be saved as reference materials to assist in better communicating the changes once the final request is passed through the production pipeline. The artists' pipeline is also streamlined, as they are now receiving requests with a specific direction and end goal, versus requests that are more open-ended and experimental. By placing the tools to communicate directly into the hands of the author, their vision comes across more clearly, more concisely and ultimately more effectively, affording him or her the ability to finally reach through the window and touch the virtual environment.

Glossary

Inertial Measurement Unit (IMU) – An electronic device that measures and reports an object's velocity, orientation and gravitational force using a combination of accelerometers and gyroscopes.

Motion Capture – The process of recording movement and translating that movement onto a digital model.

Previsualization (Previs) – A collaborative process that generates preliminary versions of shots or sequences, predominately using 3D animation tools and a virtual environment. It enables filmmakers to visually explore creative ideas, plan technical solutions and communicate a shared vision for efficient production.

Real-Time – A form of data processing in which a computer updates information at the same time it is receiving the information.

Rigid Body – An idealized extended solid whose size and shape are definitively fixed and remain unaltered when forces are applied. Tracking markers on a rigid body should be placed in a way that the orientation and position is distinguishable at any location within the volume.

Tracking Volume – The area in the virtual environment relative to where the camera is moving in physical space.

Volume – The designated area for capturing motion; the volume is the area inside of the surrounding motion capture cameras.

Bibliography

Autodesk, Inc. "The New Art of Virtual Moviemaking." *Autodesk – Film – Virtual Moviemaking.* Autodesk, Inc., 2009. Web. 5 Mar 2011. <http://images.autodesk.com/adsk/files/the_new_art_of_virtual_moviemaking_-_autodesk_whitepaper2_1_.pdf>.

Avatar. Dir. James Cameron. 20th Century Fox: 2009, Film.

Beowulf. Dir. Robert Zemeckis. Paramount Pictures: 2007, Film.

Bier, Eric, and Maureen Stone. "Snap-Dragging." *Computer Graphics*. 20.4 (1986): 233-240. Print.

Desowitz, Bill. "The Digital Eye: Dissecting a Previs Roundtable – Part 1." *Animation World Network.* AWN, 25 Jan 2008. Web. 17 Feb 2011. <http://www.awn.com/articles/digital-eye-dissecting-previs-roundtable-part-1>.

Duncan, Jody and Lisa Fitzpatrick. *The Making of Avatar*. New York, NY: Abrams, 2010. Print.

Hetherington, Janet. "Videogame Technology: Driving the Show." *Animation World Network.* AWN, 28 Nov 2006. Web. 17 Feb 2011. <http://www.awn.com/articles/gaming/videogame-technology-driving-show>.

Hinckley, Ken, Randy Pausch, John C. Goble, and Neal F. Kassell. "A Survey of Design Issues in Spatial Input." *Proceedings of the 7th annual ACM Symposium on User Interface Software and Technology (UIST '94)*. ACM: New York, NY, 1994. 213-222. Print.

Hinojosa, Stacy. "Exclusive: 'I Am Number Four' Director DJ Caruso Talks About Movie Making With iPads." *Just So You Know*. AOL, 18 Feb 2011. Web. 18 Feb 2011. <http://www.jsyk.com/2011/02/18/exclusive-i-am-number-four-director-dj-caruso-talks-about-mov/>.

Lowel-Light Manufacturing, Inc. "Lowel EDU." *Glossary of Lighting Terms*. Lowel-Light Manufacturing, 2008. Web. 22 Mar 2011. <http://www.lowel.com/edu/glossary/>.

Martinet, Anthony, Géry Casiez, Laurent Grisoni. "3D positioning techniques for multi-touch displays." *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST '09)*. ACM: New York, NY, 2009. 227-228. Print.

Peszko, J. Paul. “MoCap Makes Realtime Previs A Reality.” *Animation World Network*. AWN, 19 Sep 2007. Web. 18 Feb 2011. <http://www.awn.com/articles/review/mocap-makes-realtime-previs-reality>.

Plumer, Cliff. *Animation World Network*, 28 Aug 2005. Interview by Barbara Robertson. Web. 17 Feb 2011. <http://www.awn.com/articles/people/lucasfilm-cto-cliff-plumer-talks-technology-0>/

The Polar Express. Dir. Robert Zemeckis. Warner Bros. Pictures: 2004, Film.

Powers, Rob. *Animation World Network*, 21 Apr 2010. Interview by Bill Desowitz. Web. 17 Feb 2011. < http://www.awn.com/articles/technology/rob-powers-talks-vad-and-more-avatar>.

The Previsualization Society. “What Is Previs?.” *Previsualization Society*. Web. 5 Mar 2011. <http://www.previssociety.com/what-is-previs/>.

Rose, Frank. “5 Steps to *Avatar*: Reinventing Moviemaking.” *Wired Magazine*. Condé Nast Digital, 17 Nov 2009. Web. 11 Mar 2011. <http://www.wired.com/magazine/2009/11/ff_avatar_5steps>.

Warner, Karen. “James Cameron Dreaming Up Technology ‘Wish List’ for ‘Avatar’ Sequels.” *MTV*. MTV, 23 Aug 2010. Web. 24 Aug 2010. <http://www.mtv.com/news/articles/1646340/james-cameron-dreaming-up-technology-wish-list-avatar-sequels.jhtml>.

Weaver, Michael. PhaseSpace Camera Tracking Rigid Body. 13 Mar 2011. JPEG file.

Wilde, Tyler. “How Halo: Reach’s ‘Deliver Hope’ trailer was virtually filmed, and how motion controls ought to work.” *GamesRadar*. Future US, Inc., 2 Mar 2011. Web. 22 Mar 2011. <http://tinyurl.com/48vyy69>.

Wolff, Ellen. “Tool Time at Pixar.” *Digital Content Producer*. Penton Media, Inc., 1 Nov 2004. Web. 24 Aug 2010.

Appendix A: Parallel Efforts and Sources of Inspiration

The idea behind TCam grew from a series of industry talks and classroom discussions grounded in animation and motion capture processes. TCam draws on a variety of virtual camera features, previsualization and production processes, and new theories of cinematic production and virtual moviemaking. Hands-on work with several virtual cameras helped to experience the inner workings of each system, to learn how each could benefit the design of TCam.

External Cross-Section Rigid Body

A virtual camera can be created from any type of object by attaching markers to create a *rigid body*. When using a hand-held or production camera, the markers are often attached to an additional *rigid body* in a distinguishable pattern. In general, the *rigid body* consists of a center pole that has at least three arms branching from it, with markers placed at each end. Marker configuration patterns are dependent on the tracking system that is being used, as tracking systems recognize the markers differently. When using a passive optical tracking system, such as Vicon or the Giant Studios system, each reflective marker is seen independently but not uniquely by the system. The *rigid body* must be configured in such a way that the marker relationship is unique and distinguishable at any

Figure 23: The virtual camera used on *Avatar* utilized a passive optical tracking system marker configuration. Source: Frank Rose

orientation. A combination of horizontal and vertical extensions of varying length, similar to that in Figure 23, creates a pattern that is unique to the system at any rotation.

In the case of active optical tracking systems, such as the PhaseSpace system, each individual marker receives a unique ID from the system, thus orientation can be determined with a less constricted marker pattern. With any system, it is more important to create a *rigid body* that can be presented prominently to the cameras. The addition of a *rigid body* to track a camera is a concept that is utilized by the NaturalPoint Insight VCS system and many custom-built virtual cameras.

Figure 24: A rigid body designed for use with a PhaseSpace system. Source: Michael Weaver

InterSense VCam™
http://www.intersense.com/pages/19/28

The InterSense VCam™ system uses motion-tracking sensors integrated directly into a production camera body. The camera controls are then mapped to the camera functionality in MotionBuilder®, giving the user complete local control over virtual camera settings, recording, and playback options. By using

Figure 25: InterSense VCam™ LCD viewfinder with real-time MotionBuilder® feed. Source: InterSense

internal sensors, the VCam™ does not need to be used inside of a motion capture *volume*, but instead relies on the company's own IS-900 tracking system. Extended models also have wireless capabilities, as well as the ability to receive a real-time video feed of the virtual environment in the LCD viewfinder window. The VCam™ has a built in joystick allowing the user to translate, scale, and rotate the recording *volume* in MotionBuilder®, as well as additional controls for zoom, jog, scrub, and key frame capabilities.

NaturalPoint Insight VCS

http://www.naturalpoint.com/optitrack/products/insight-vcs/

NaturalPoint offers two consumer options for virtual cameras. The first, VCS:Pro is a more heavy-duty virtual camera suited for production environments. The VCS:Pro provides real-time feedback via an attached HD LCD monitor and precision camera controls with both button and joystick input. The monitor and controls are attached to a shoulder mount that allows a user to comfortably wear the virtual camera balanced by his or her body. The second camera solution is the VCS:Mini, a lower-cost, lightweight

Figures 26: NaturalPoint VCS:Pro (left) provides vital support and camera controls with the use of a shoulder rig, while the VCS:Mini provides a lightweight and more flexible alternative.
Source: Tyler Wilde

virtual camera that utilizes a Xbox 360 controller with an attached cross-sectional *rigid body* for tracking purposes. Similar to the VCS:Pro, the controller provides the ability to map camera controls to MotionBuilder® or Maya. Both solutions provide customizable access to the virtual camera pipeline.

PhaseSpace Multi-Touch Tablet

http://www.phasespace.com/virtualcamera.html

PhaseSpace customized a HP® TouchSmart tm2 tablet notebook by embedding tracking markers in the casing, to create a touch driven virtual camera. Since the markers are embedded directly into the camera casing, it is not necessary to utilize an additional *rigid body*, leaving the casing free of an external tracking ornament. By running MotionBuilder® natively on the tablet, the screen is transformed into the virtual camera viewfinder. Unlike other virtual cameras that receive a video feed from MotionBuilder® workstation, the PhaseSpace virtual camera runs an instance of MotionBuilder® directly on the tablet. A second workstation is used to record the motion capture data. While the data cannot be recorded locally from the tablet, the .fbx file can be saved for later use. In addition to acting as a camera viewfinder, a custom script generates sliders allowing the user to control the lens focal length and camera height offset. While TCam draws on

Figure 27: TCam draws from the PhaseSpace virtual camera (above) by embedding markers on the casing. Source: PhaseSpace

several of these same ideas, it removes the need of the user to be proficient with MotionBuilder®.

During discussions surrounding these cameras, the ability to locally modify camera settings directly on the device continually arose. TCam provides this capability with its camera toolset features. Additionally, like the PhaseSpace virtual camera, TCam's tracking markers are embedded in the sides of the iPad, eliminating the need for an additional *rigid body*, which would detract from the device's already portable design.

TCam's modification toolset draws on the concepts and motivations behind several custom-built and proprietary moviemaking applications. While not all of these applications are directly related to motion capture, they provide insight on the power of custom designed cinematic software packages.

Pixar's Review Sketch

http://www.pixar.com/

Since process is absolutely paramount to Pixar's production pipeline, Dr. Michael B. Johnson and the Studio Tools group developed the Review Sketch application; a custom built application that runs on the Wacom Cintiq tablets. Review Sketch allows users to directly draw over each frame of an animation sequence. "It's a classic animation thing to want to put a clear sheet over an image and draw on top of it," stated Johnson (Wolff). The edits are then automatically saved off as separate files on a network server, accessible from anywhere within Pixar. This networking feature allows artists to collaborate on a sequence in a review session or during dailies and immediately access any modifications at their workstations. Review Sketch also allows users to see the past history of any given frame and to compare changes across sequences or shots.

FrameForge Previz Studio

http://www.frameforge3d.com/

FrameForge is a leading previsualization and storyboarding software suite, with both desktop and iPhone applications. Originally a 2D storyboarding application, FrameForge now supports 3D environments, as well as stereoscopic 3D. Within a set, a user can create and place any number of assets including set pieces, characters, lighting fixtures, and cameras. FrameForge also assists in improving physical space planning by accounting for dolly tracks, camera rigs and tripod placement. Blueprint views allow a user to see how he or she would physically complete the planned shots. FrameForge is focused on providing users the ability to visualize the end product and allows them to see what their equipment will eventually see.

Xtranormal

http://www.xtranormal.com/

With over 10 million projects created, Xtranormal boasts, "If you can type, you can make movies." Xtranormal provides two outlets for creativity: State, a free desktop application and a web platform called Movie Maker. Both applications allow users to quickly create

3D animated shorts using text-to-speech capabilities and buttons to generate animations. After creating a script, a user can drag in action icons, facial expressions, camera movements and sound effects directly into the script. Users can also choose from a growing variety of characters and scene packages to use in their animations. Xtranormal also ties into the social networking spectrum with direct connections to share creations on Twitter and Facebook, as well as recently launching their first video creation contest with a cash prize and YouTube publicity. Xtranormal is a unique adaptation of an animation style WYSIWYG editor, scaled for mass use and distribution.

Oblong Industries' Tamper

http://oblong.com/ | http://vimeo.com/2821182

Tamper is a collage style moviemaking application that runs on the company's own g-speak system. The g-speak spatial operating environment utilizes a user's gestural input to drive the system. Tamper consists of a collection of full-length feature films with various elements rotoscoped throughout the film. Through a variety of gestural commands, a user can access these rotoscoped elements within a scene and duplicate them on a secondary surface by clicking and dragging between the main screen and secondary surface. Since each element has been rotoscoped independently from the scene, they continue to animate in a loop on the secondary surface. Ultimately, a user can selected several elements from various films and compile them together, much like a magazine collage, creating a new type of recombinant short film.

Industrial Light & Magic's Zeno
http://www.ilm.com/

Zeno is considered to be the backbone of ILM's new in-house production pipeline. Zeno, a suite of desktop applications, combines the ILM pipeline with that of LucasArts, allowing ILM artists to take advantage of the LucasArts game engine. Lucasfilm CTO Cliff Plumer describes the new benefits of Zeno, "A visual effects supervisor can sit with the director and have a synthetic scene move around in realtime. The director can block in a scene and do a camera move with a virtual camera. It feeds the whole post process" (Plumer). Zeno also makes the production pipeline bi-directional; artists can copy out and drop in assets in real-time between the engine and other Zeno-based tools.

Crystal

http://vimeo.com/8083372

Crystal is a previsualization application that was designed by a team of USC graduate students as part of the Immersive Moviemaking: Gestural Interfaces for Cinematic Design course. Crystal brings the paper-prototyping aspect of game development into the previsualization world by creating a sandbox application that allows users to create camera paths based on gestures and configure shot layouts with tangible objects such as blocks, cars and plastic Army men. Crystal is built using Oblong Industries' g-speak system.

These applications all provide the user the functionality to quickly modify or annotate whatever he or she is working on. Whether it is a set layout in FrameForge or a walk cycle in Review Sketch, these applications afford the user the ability to try multiple configurations quickly, with little effort or downtime. The ability and direct control to rapidly iterate and experiment with the virtual environment is one of the capabilities that TCam provides that sets it apart from other virtual cameras.

When considering the general interaction paradigm from TCam, it is important to look at current mobile applications, more specifically those used for sketching and moviemaking, and how they are used for both creative and communicative purposes.

Autodesk® SketchBook® Mobile

http://usa.autodesk.com/adsk/servlet/pc/item?siteID=123112&id=13872203

Available for the iPhone and iPad, as well as recently released Android version, SketchBook® quite literally turns a mobile device into a digital sketchbook. Like many desktop sketching applications, a user can chose between different drawing utensils, mediums, and colors. Imported photos can be sketched over top of by using layers and then exported to a .psd file for later use in Adobe® Photoshop®. Similar to most high-end drawing applications, SketchBook® has a complex menu system, but tries to make the menus less obtrusive by putting their control in the user's hands. By tapping on a small circular icon or using a three-finger tap, the menu appears over the workspace. Once the user is done making selections, tapping once on the screen quickly hides the menu again. TCam utilizes this same menu interaction philosophy, as it is a simple and efficient solution for maximizing screen space while not reducing menu size or functionality.

Adobe® Photoshop® Express

http://mobile.photoshop.com/

Photoshop® Express provides several of the more common Photoshop® features in a mobile application. While a user can modify photos stored locally on his or her mobile device, Photoshop® Express also provides a sharing feature that allows a user to grab photos he or she has previously stored in an online account. Along with being able to instantly modify shared photos, a user can auto-upload photos stored on the device once they have been modified, streamlining the sharing process. One of the more unique and efficient interaction features of the application is its invisible slider control. By swiping a finger left or right across the screen, mimicking a slider, the user can adjust various settings such as contrast or exposure. As with the TCam sliders, this allows the user to focus on the modifications being made and not the accuracy of interacting with the slider.

A plethora of mobile applications currently exist that mobilize moviemaking tools, making several extremely useful tools available on one device, at the tap of a single button. These applications give an overarching idea to the larger impact that mobile devices are having on the cinematic industry, for both professionals and novices alike.

Action Log Pro

http://www.andris.co.uk/actionLogPro.html

Action Log Pro keeps track of reel names, time codes, and additional comments for up to

25 devices on set. Data files can be emailed to digitizers directly from the app in formats optimized for Avid or Final Cut Pro.

Artemis Director's Viewfinder

http://chemicalwedding.tv/artemis.html

Artemis allows a user to experiment with multiple camera formats, lens settings and aspect ratios by simulating the camera settings on top of the iPhone camera viewfinder. Artemis also has a remote iPad connection, allowing a user to share images directly from the iPhone with an iPad.

FiRe Field Recorder

http://www.audiofile-engineering.com/fire/

FiRe is the first professional audio field recorder for the iPhone. With several metering tools, file formats and even audio markers, FiRe makes recording mono or stereo sound quick and easy.

MovieSlate

http://www.pureblendsoftware.com/

MovieSlate is a digital slate and clapboard complete with shot log and notepad capabilities. It also provides the ability to store time code information, as part of a complete history log that allows a user to rate and sort take information.

Pocket Call Sheet

http://www.snakebytestudio.com/pocketcallsheet/

Currently the only app that allows a user to create and share call sheets directly from an iPad or iPhone.

Reel Director

http://www.nexvio.com/product/ReelDirector.aspx

Reel Director is a feature-rich iPhone and iPad editing app that lets a user drag and drop clips together in a timeline, as well as add in transitions, text overlays, watermarks and voiceovers or other audio stored on the device. Completed videos can be saved to the device, emailed, or directly uploaded to YouTube.

Scripts Pro

http://www.scriptsapp.com/

Scripts Pro is a screenwriting app for both the iPhone and iPad, that allows a user to compose directly in the app, as well as import additional pages and images into a script. A future update of Scripts Pro promises Final Draft® connectively to better share scripts between the two applications.

Storyboard Composer

http://www.cinemek.com/storyboard/

Storyboard Composer, previously released at Hitchcock, helps users create and share storyboards without ever having to draw a single stick figure. A user can quickly create storyboards from images saved on the device and insert camera moves with the tap of a button. Additional actors, motion commands and notes can be placed over already storyboarded images; finished boards can be exported as PDF files directly from the application.

Appendix B: Previous Work

Much inspiration and direction for TCam grew from and drew on past work, which expanded across several mediums and genres, reinforcing the importance and necessity in considering new modalities for communication and exploring the benefits that these paradigms could bring to their respective industries.

Flatbed

http://vimeo.com/7777765 | http://vimeo.com/8225409

A gesture-based moviemaking suite, Flatbed is a prototype concept for a new methodology of moviemaking. Built using the g-speak system, Flatbed was conceived by a team of three other USC graduate and undergraduate students during the Immersive Moviemaking: Gestural Interfaces for the Cinematic Industry class. Flatbed explored the potential of pairing the combination of hand gestures and multi-touch input with the standard features of a digital editing suite. After conceptualizing a full prototype of the system, a working prototype was built demonstrating the most innovative sequence-editing element. In the application, the user has a large main screen for reviewing purposes and a secondary projected tabletop display hosting a variety of clips like an editing bin. The user can select any number of clips with a one-finger point, mimicking a mime gun, towards the table. By dropping the thumb, or clicking, the user can begin to draw a line through the clips to select multiple clips at once. When finished, the user releases his or her thumb and can play back the sequence on the main screen via a series of playback gestures. This style of interaction takes on a more montage or building block approach to editing, allowing a user to quickly shuffle through a variety of clips and sequence configurations.

BMW R.I.D.E.

http://mobilemedia.usc.edu/?p=47

Developed in a collaborative effort between USC's Mobile and Environmental Media Lab and BMW's Palo Alto research department, BMW R.I.D.E. is a new approach to understanding and utilizing the interactions between the vehicle and driver. Through the use of three interactive driving modes, Soundscape, Play and Adventure, the car can begin to learn about its driver, his or her likes, dislikes, driving styles and general daily driving behaviors. In addition to the in-car experience, a driver is able to continue the relationship with the car away from the vehicle via the use of a roundel and the Lifelog application. The roundel serves as both a communication device between the driver and the vehicle, and also more importantly as the physical key to the vehicle, strengthening the relationship the driver has with the tangible object. Once at home, the roundel activates the Lifelog application on our prototyped interface, a Microsoft Surface. The Lifelog provides an interactive interface of aggregate data for a driver to explore and better connect with their vehicle. The Lifelog provides access to driving mementos such as photos and videos taken by the car, previously generated ambient soundscapes, and newly earned badges and rewards, as well as more practical statistics such as mileage, gas consumption, and daily drive maps. The Lifelog also provides trip-planning capabilities as well as incorporates information from the driver's social networks, such as Yelp! recommendations or Foursquare check-ins, to better engage and inform the driver.

Microsoft Surface Explorative Research

As part of the ongoing gestural research at the Mixed Reality Lab at USC's Institute for Creative Technologies (ICT), the Microsoft Surface was used to explore and define several types of multi-touch interactions. Investigation first began by experimenting with new ways of incorporating multi-touch gesture controls into the music-editing pipeline. Explorations first started with multi-touch digitization of various editing tools, such as a low pass filter. After industry discussions, it was more apparent that current pipeline limitations in music editing did not lie in the interface design, but more so in the ability to quickly locate and retrieve files. Assuming the Surface was transformed into a digital audio mixing board, experimentation continued with various contextual saving algorithms. Of these, the one with the most resonances was a system that allowed a user to quickly touch the audio timeline to create a visual "mental note". The system would then record and store the interactions with the Surface and ambient room audio 10 seconds prior to the touch and 10 seconds following the touch at the given point on the timeline. Adding a vertical line to the timeline marked this "mental note" point. By touching the line, a user could replay the recorded 20 seconds of Surface interactions and ambient audio. This experiment demonstrated the usefulness of contextual audio to a jog a user's memory.

Gestural Interfaces for Statistical Exploration

In a partnership with USC's Information Sciences Institute, a prototype application was developed to explore the use of gesture-based interactions for editing population stratification plots. Using the g-speak system, a user could rotate or translate a tetrahedral prism plot with a single-handed gesture, or scale the plot size with a two handed gesture. The application also handles multiple plots, allowing a user to compare plots side by side. In addition to the manipulation gestures, a user could select individual data points with a one-finger point, similar to a mime gun, to receive more detailed information about the data point. By pointing and clicking, dropping the thumb, on a data point, a user can change its visibility settings, allowing for easy removal or inclusion of outliers.