One-Man-Band: A Touch Screen Interface for Producing Live Multi-Camera Sports Broadcasts

Eric Foote
Disney Research
Pittsburgh, USA
efoote@nrec.ri.cmu.edu

Peter Carr
Disney Research
Pittsburgh, USA
carr@disneyresearch.com

Patrick Lucey
Disney Research
Pittsburgh, USA
pjlucey@disneyresearch.com

Yaser Sheikh
Carnegie Mellon University
Pittsburgh, USA
yaser@cs.cmu.edu

Iain Matthews
Disney Research
Pittsburgh, USA
iainm@disneyresearch.com

Generating live broadcasts of sporting events requires a coordinated crew of camera operators, directors, and technical personnel to control and switch between multiple cameras to tell the evolving story of a game. In this paper, we present an unimodal interface concept that allows one person to cover live sporting action by controlling multiple cameras and and determining which view to broadcast. The interface exploits the structure of sports broadcasts which typically switch between a zoomed out *game*-camera view (which records the strategic team-level play), and a zoomed in *iso*-camera view (which captures the animated adversarial relations between opposing players). The operator simultaneously controls multiple pan-tilt-zoom cameras by pointing at a location on the touch screen, and selects which camera to broadcast using one or two points of contact. The image from the selected camera is superimposed on top of a wide-angle view captured from a *context*-camera which provides the operator with periphery information (which is useful for ensuring good framing while controlling the camera). We show that by unifying directorial and camera operation functions, we can achieve comparable broadcast quality to a multi-person crew, while reducing cost, logistical, and communication complexities.

Keywords

Camera Control, Touch screen, Multiple Cameras

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*

General Terms

Human Factors

1. INTRODUCTION

Creating production-quality broadcasts of live sporting events requires a coordinated crew of skilled individuals constantly aware of what is happening in the game and how the rest of the crew is cooperating to cover the action [11]. For popular team sports (e.g. professional football, basketball or hockey), a large broadcast crew is required to coordinate multiple cameras so that the broadcast can follow the fast-paced action on the field. There is also a much larger number of niche sporting events with a smaller, but devoted following, such as little-league, high school, and lower-divisional collegiate sports. Cost and logistical complexity preclude professional multi-camera setups, which is why these smaller events are typically webcast by a novice operator controlling a single camera. In this paper, we present a single operator interface that unifies the roles of director and camera operators, and allows a single user to create compelling broadcasts of live sporting events by simultaneously controlling

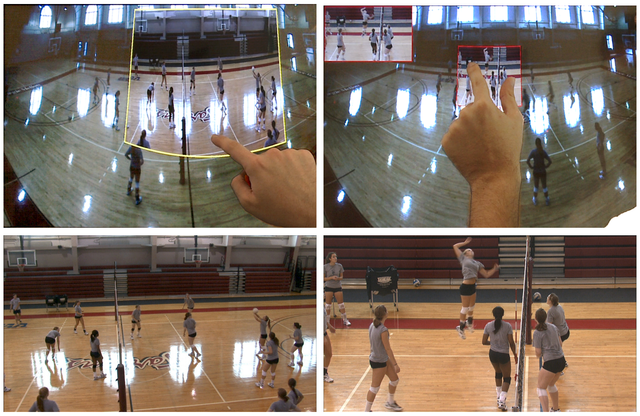


Figure 1: (Top) To control the *game*-camera, the operator uses one finger to drag a box highlighting the portion of the court that the user wishes to frame, and two fingers to switch to the *iso*-camera. (Bottom) The output of our interface is a broadcast video produced by a single user (see accompanying video for full effect).

MM'13, October 21–25, 2013, Barcelona, Spain.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502092.

and switching between multiple pan-tilt-zoom (PTZ) cameras.

The interface exploits the particular structure of focus and context between the different roles on a sports broadcast production crew. As live sport is fast moving and highly variable, there is a necessity for the interface to include both spatial and temporal contexts. In a typical setup, one camera operator controls the *game*-camera, which maintains a wide, zoomed-out shot to capture the team-level action. A second camera operator controls the *isolated*-camera (or iso-camera), which captures the emotion and drama of the play using a zoomed-in view for close-ups of players, coaches, or fans. Both camera operators focus on their viewfinder to ensure they have well composed shots. Simultaneously, their periphery vision gives them context of how the play is unfolding, which allows them to follow the action. The director relies on a bank of live monitors in a control room to select a series of camera angles to best focus on what is happening in the game. The director is in continual dialogue with the camera operators to get particular shots which will help place the action in context of the story the director is trying to tell. Thus, for a high-quality live broadcast to be obtained, the communication and understanding of instruction between the director and camera operators has to be very clear, which is hard to get achieve with an inexperienced team.

Our *One-Man-Band* (OMB) interface unifies the roles of director, game-camera operator, and iso-camera operator into an unimodal interface that allows one operator to produce a compelling broadcast of a sporting event. Our interface combines focus and context by superimposing video feeds from multiple robotic pan-tilt-zoom cameras onto video from a stationary fish-eye *context*-camera that captures the entire playing area (see Figure 1). The cameras are precisely calibrated and share a common vantage point. The overlay of the broadcast camera resembles a viewfinder, and the static context-camera mimics periphery vision. The user controls the game-camera by dragging the highlighted overlay to a new area of the touch screen using a single finger. To cut to the iso-camera, the user places a second finger on the screen. The view from the iso-camera is then displayed, and the iso-camera can be controlled in a similar fashion by dragging both fingers. To cut back to the game-camera, the user lifts one finger (so that only one finger is in contact with the touch screen) and the superimposed image returns to the game-camera. Thus, the interface focuses the user's attention on one of the camera roles at a time, while simultaneously providing live spatiotemporal context of the game, allowing the user to make timely cuts from one camera to the other, so that the evolving storyline of the game is captured in the broadcast[1].

We compare our touch interface concept against a traditional three-person crew, with two joystick operators for the game- and iso-cameras, and one director responsible for switching between views. To determine if there is any significant difference between the two approaches, we conducted two user studies and a perception study comparing the two methods. For the perception study, the users were asked to evaluate the two methods against three criteria: (1) action following: the quality of individual camera control, (2) appropriateness of cuts: the quality of cuts between cameras, and (3) overall quality: the quality of the final broadcast footage. In addition to the perception studies, we also carried out quantitative analyses which numerically shows how similar the approaches are in terms of the broadcast generated.

2. RELATED WORK

Interactive applications often require users to interact with more information than the screen can display. The *Focus + Context* interface scheme, introduced in the seminal work of Sarkar et al. [15] on fisheye lenses, enables users to interact with different level of details. More recently, Pietriga et al. [12] investigated a general framework for lenses that helped users in the navigation of large scenes, while Cockburn et al. [4] presented a general survey on Focus + Context techniques. We can apply a Focus + Context analogy to sport broadcasting with camera operators providing high resolution content through multiple view points. The director then requires an overview of all the video content in realtime and uses available visual information to decide the focus (i.e., select the appropriate camera viewpoint). Our system builds upon the Focus + Context interface scheme to provide both an overview of the area covered by the cameras and to enable the user to dynamically set the current area of interest.

Another approach is to use computer vision to automate the selection of current viewpoint. Some applications propose to remove the user from the decision process in the context of sport broadcasting [2, 13]. Compared to our system, which is generalizable to many sports, these approaches are often tailored specifically to a single sport. For example, Ariki et al. [2] proposed a system for producing a fully automated broadcast of a soccer game using computer vision and action recognition. Their application did not run in realtime, meaning the broadcast could only be generated as a post-processed recording rather than a live event.

Some methods have placed control of the broadcast in the hands of the viewer, instead of automating the broadcast. For example in [10], Matthews proposed a system allowing the viewer of a sporting event to select between multiple camera views, eliminating the need for a director, though still requiring the use of human operators to control each of the cameras. Free-Viewpoint Television [18] eliminated the need for camera operators with a system that uses an array of cameras to synthesize a view from any angle. Although this approach requires neither a director nor the use of any camera operators, the overhead involved in setting up the camera array is not suited for the low-budget scenario that we target with our system.

In the more general category of remote camera control, Liu et al. [9] proposed a system which, similar to the method proposed here, made use of a controllable pan-tilt-zoom camera plus a wider-angle view to provide the context. In this system, the views of both the context camera and the pan-tilt-zoom camera were displayed to the user on separate screens. The user would select a region to view by drawing a box on the context camera view, and the pan-tilt-zoom camera would position itself to view that region. Although our method makes similar use of a wider context camera for helping the user select a view, the "region-selecting" method

[1] In terms of a high-definition (HD) live broadcast, cropping the iso-camera view from the game-camera is not a reasonable option as the resolution will be substantially lower, and the iso-camera would be constrained to be taken from only within the confines of the game-camera view.

in [9] requires the user to draw a new box each time they want to move the camera, and to continuously switch between the context and pan-tilt-zoom views. By contrast, our system displays both views on a single screen, and allows the user to drag the pan-tilt-zoom display region to respond more easily to a more fast-paced and dynamic scenario such as a sporting event. Our system also allows for the control of multiple cameras, while [9] uses only a single camera.

Other work on camera control has focused on the control of virtual cameras in a graphics or animation environment [3, 5, 16, 19, 20]. A common premise in each of these methods is that it is easier to control a camera by specifying the desired framing than it is to manipulate the degrees of freedom of the camera directly. For example, Singh et al. [16] presented a controller for manipulating a virtual camera by specifying the vanishing points and horizon line of the camera view. Manipulating the camera perspective would then affect the focal length, center of projection, or pose of the camera. Drucker and Zeltzer [5] proposed an interface for specifying constraints on the camera view, such as the size and/or position of one or more objects. Both [8] and [17] present a method for camera control by directly selecting an object for the camera to point at, and [17] also allowed users to control a virtual camera in a 3D graphics environment by specifying the framing of two points in the virtual world. In [6], Gleicher and Witkin proposed a virtual camera controller in which the user could "pin" one or more objects to sub-regions of the camera view, and using these constraints to solve for the time derivatives of the camera parameters.

A touch interface for controlling a remote camera was proposed by Kuzuoka et al. [8] as part of a system for remote collaboration. In this system, the camera view appeared on a touch screen display, and touching a displayed object would cause the camera to automatically center on that object. The interface also supported velocity control by holding a finger on the display. The velocity and direction of the resulting camera motion directly corresponded to the direction and distance from the touch position to the center of the display. However, the proposed controller did not allow for the use of multiple cameras, and did not provide a wider context view as our system does.

Other methods for making robotic camera control more intuitive have been proposed as well. Aiono et al. proposed a robotic camera that could be controlled by head movements, to be used by a doctor during surgical procedures [1]. Rui et al [14] proposed a system for remote viewing of meetings that used a 360 degree panorama view of the meeting room. Users would control a virtual pan/tilt camera that cropped the panorama to view only the portion that the user wanted to look at.

3. INTERFACE DESCRIPTION

Many elements for achieving simultaneous focus and context are common to both camera operation task (game- and iso-camera), so we describe these shared aspects first. A stationary *context*-camera with a fish-eye lens is used to display the entire area of play. The video from the active camera (either the game-camera or iso-camera) is superimposed on top of the context image in its correct location so that it appears as an inset on the context-camera view. To focus the user's attention on the active camera, the image of the context camera is desaturated, and the superimposed image of the broadcast camera image is outlined with a solid color border (yellow for the game-camera and red for the iso-camera). All three cameras are calibrated and share the same vantage point. The views from all cameras are displayed and updated live.

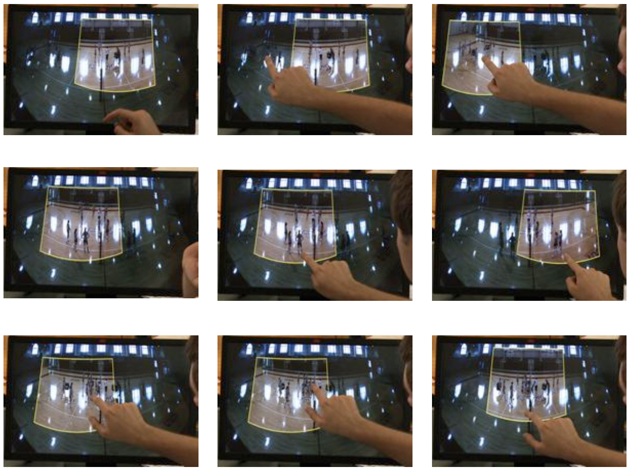


Figure 2: There are three ways to control the game camera from our interface. (Top Row) Touching a location at any point outside of the game-camera highlight box causes the camera to center on that location. (Middle Row) Touching and dragging any point inside of the game-camera box will move the camera relative to its current position. (Bottom Row) Touching the screen and then immediately releasing will also center the game camera on the point of contact, whether or not the user touched the screen inside the game-camera box.

Game-Camera Control.

The game-camera operator is responsible for covering the general action of the sporting match. For ball sports, this means that the ball should always be in shot as well as most of the players. If the game has a goal (or basket) then this should also be included, if possible. To accomplish these objectives, the game-camera operator maintains a wide-angle shot. As this view contains most of the information of interest, it will be used the majority of the time.

The three methods for controlling the game-camera are illustrated in Figure 2. In the first method (top row), the user touches a location on the screen outside the current view and the highlighted box containing the game-camera view will move to the center of that location. We call this the "tap-to-center" approach. The second method (middle row) or "click-and-drag" method is activated by touching the screen at any location inside the game-camera highlighted box and maintaining contact. With this form of control, the user drags his/her finger across the screen, and the game camera moves as needed to to maintain the same position relative to the user's finger. The third method of control (bottom row) is to touch and immediately release a location on the screen. The game-camera will center to that location,

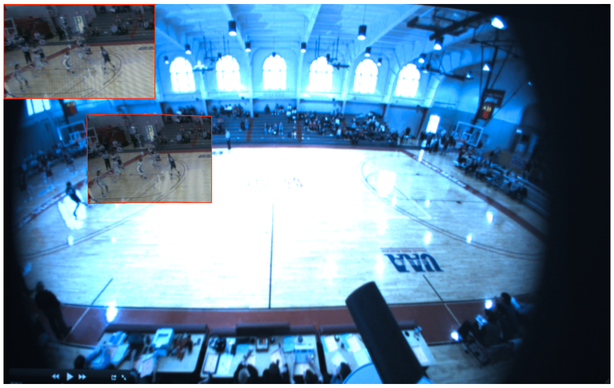


Figure 3: When the user switches to the iso-camera, the view is displayed in a highlighted red box and is also shown in the upper left corner of the interface to avoid occlusion from the user's hand.

regardless of whether it is currently in the highlighted box or not.

Iso-Camera Control.

The role of the iso-camera operator is to capture close-up shots when something interesting happens. This normally results in a zoomed-in shot of a player, coach or fan which illustrates the emotion and drama of the play that has just occurred. In the OMB interface, the user chooses and operates the iso-camera by placing a second finger on the screen. Like the game-camera view, the iso-camera view is displayed in a highlighted box on top of the context view, with its position corresponding directly to the position of the actual camera so that the views line up. The iso-camera controller also supports both the tap-to-center and click-and-drag control methods, using the midpoint of the user's fingers as reference.

The iso-camera is operated at a high zoom, since its purpose is to get close-up shots. As a result, when superimposed on the context image, the iso-camera image may appear quite small, making it difficult for the user to judge whether the shot is well composed. In addition, the user's hand may occlude the superimposed image. As a result, a virtual viewfinder image is superimposed in the top-left corner of the screen. Unlike the regular superimposed view, the virtual viewfinder image is not registered to the context view but simply inserted over top (See Figure 3).

Usage.

In a live sporting event, the user must constantly decide whether to broadcast the game-camera or the iso-camera. In this role, they are essentially the director, as they choose which view best conveys the story unfolding in the match. To make this easier, a series of simple rules can be applied. For instance, the main rule is to use the game-camera when the game is active, which occurs the majority of time. This view should include the ball and the majority of players in this view. When there is a break in play, it is then appropriate to cut to the iso-camera centered on a player/coach/fan who is reacting to recent events which have just transpired in the game. Examples of the system in use for basketball are given in the Experiments section.

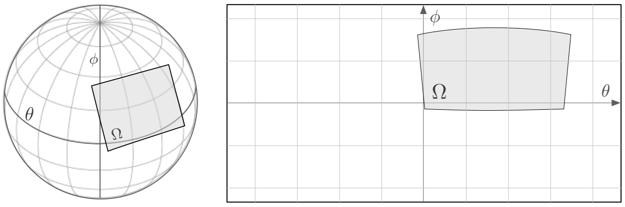


Figure 4: The image plane of a PTZ camera can be considered as a tangent plane on a unit sphere, where the position of the tangent plane is determined by the pan and tilt angles of the camera, and the size is determined by the focal length (zoom) of the camera. In our interface, the surface area of the sphere is warped onto the flat plane of the display by sampling the spherical angles ϕ and θ uniformly (right). The viewable region in spherical coordinates remains mostly rectangular for small tilt angles and reasonably large focal lengths.

4. SYSTEM DESCRIPTION

Our system uses a pair of Sony BRC-Z700 cameras for the game- and iso-cameras, and an Allied Vision GE 1910C for the context-camera with a fisheye lens. A 3M M2256PW touch screen monitor is connected to a Mac Pro, and a Gefen 4x1 video switcher toggles which broadcast signal is sent to the computer. Both Sony BRC-Z700 cameras and the Gefen video switcher are controlled by the Mac Pro via RS-232 serial links.

4.1 Graphical Display

The touch screen display always shows a composite of two live video feeds. The image of the currently broadcasting camera is aligned to and superimposed over the image of the context-camera by mapping both images onto the surface of a sphere. The necessary portion of the spherical surface is then displayed on the touch screen by plotting the composited image as a function of spherical angles (see Figure 4). This unwarping operation is computed by interpolating a spherical area of interest $\Omega = [\theta_{\min}, \theta_{\max}] \times [\phi_{\min}, \phi_{\max}]$ over the number of pixels in the display, and associating to every pixel location (u, v) a 3D point $\mathbf{M} = [M_x, M_y, M_z]$ on the surface of the unit sphere

$$M_x = \sin\theta\cos\phi \tag{1}$$
$$M_y = \sin\phi \tag{2}$$
$$M_z = \cos\theta\cos\phi. \tag{3}$$

The composited spherical image is computed by projecting each 3D point $\mathbf{M}$ into the context-, game- and iso-cameras to determine appropriate sampling locations $\mathbf{m}_c$, $\mathbf{m}_g$ and $\mathbf{m}_i$; in each camera for image interpolation (where the subscripts stand for **c**ontext, **g**ame and **i**so). Each camera has its own coordinate system, and the reference frame of the unit sphere is arbitrary. However, we assume the centers of projection (lenses) of all three cameras are located in approximately the same 3D position $\mathbf{C}$, and define this as the centre of the unit sphere. By convention, a camera coordinate system is defined such that the $\hat{\mathbf{z}}$ axis points in the direction the camera is looking, and the $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ directions correspond to the right and down directions on the image plane.

A camera's image plane is located one focal length f away from its center of projection $\mathbf{C}$ along its optical axis $\hat{\mathbf{z}}$. The principle point $\mathbf{p} = [p_u, p_v]$ corresponds to the intersection of the optical axis and the image plane, and is often assumed to coincide with the center of the image. The relationship between a 3D point $\mathbf{M}$ expressed in the camera coordinate system and its 2D projected location $\mathbf{m} = [m_u, m_v]$ on the image plane is governed through similar triangles (see Figure 5). Using homogeneous coordinates [7], the relationship is compactly expressed via matrix multiplication

$$\mathbf{m} = \mathtt{K}\mathbf{M}, \tag{4}$$

where

$$\mathtt{K} = \begin{bmatrix} f & 0 & p_u \\ 0 & f & p_v \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The remainder of this discussion addresses how each 3D coordinate $\mathbf{M}$ is transformed from the coordinate system of the unit sphere to the coordinate system of each camera. Once $\mathbf{M}_c, \mathbf{M}_g$ and $\mathbf{M}_i$ are known, the projected image locations $\mathbf{m}_c, \mathbf{m}_g$ and $\mathbf{m}_i$ can be recovered from (4), where a particular $\mathtt{K}$ is used for each camera (since the cameras will have different focal lengths and principle points).

A PTZ camera has a camera coordinate system (as defined earlier) and a motor coordinate system. The camera coordinate system is determined by the camera's current pan and tilt angles, whereas the motor coordinate system remains fixed. For a given pair of pan/tilt angles, the mapping from motor to camera coordinate systems is governed by a 3D rotation

$$\mathtt{R}(\theta, \phi) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ \sin\theta\sin\phi & \cos\phi & \cos\theta\sin\phi \\ \sin\theta\cos\phi & -\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \tag{6}$$

The orientations of the motor coordinate systems and focal lengths of the context-, game- and iso-cameras are determined from homographies [7] between the image planes and the world ground plane. Since each camera may not be in its home position ($\theta = 0, \phi = 0$) when the homography is estimated, the extracted orientation $\mathtt{S}$ describes the changeable camera coordinate system. As a result, the orientation $\mathtt{Q}$ of the camera's stationary motor coordinate system must take into account any non-zero pan θ_0 and tilt ϕ_0 angles at which the homography was estimated

$$\mathtt{Q} = \mathtt{R}^{-1}(\theta_0, \phi_0)\mathtt{S}. \tag{7}$$

Each 3D point $\mathbf{M}$ is transferred into the three camera coordinate systems by first rotating into each camera's motor coordinate system, and then rotating to the camera coordinate system for the camera's current pan/tilt angles

$$\mathbf{M}_c = \mathtt{Q}_c\mathbf{M} \tag{8}$$

$$\mathbf{M}_g = \mathtt{R}(\theta_g, \phi_g)\mathtt{Q}_g\mathbf{M} \tag{9}$$

$$\mathbf{M}_i = \mathtt{R}(\theta_i, \phi_i)\mathtt{Q}_i\mathbf{M}. \tag{10}$$

The context-camera is stationary, so it has no pan/tilt rotation matrix $\mathtt{R}$. As a result, the corresponding projected image locations are

$$\mathbf{m}_c = \mathtt{K}_c\mathtt{Q}_c\mathbf{M} \tag{11}$$

$$\mathbf{m}_g = \mathtt{K}_g\mathtt{R}(\theta_g, \phi_g)\mathtt{Q}_g\mathbf{M} \tag{12}$$

$$\mathbf{m}_i = \mathtt{K}_i\mathtt{R}(\theta_i, \phi_i)\mathtt{Q}_i\mathbf{M}. \tag{13}$$

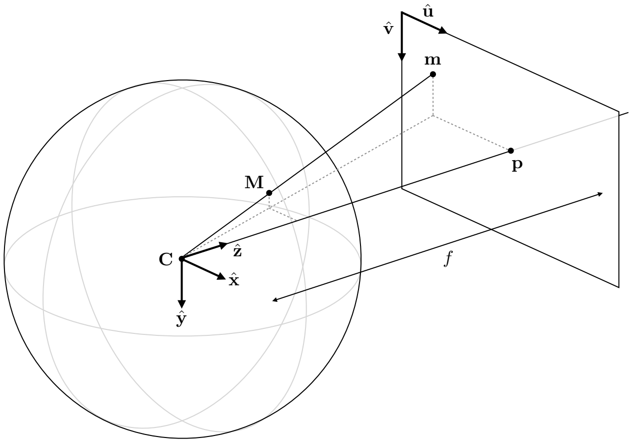


Figure 5: The composited spherical image is computed by transforming each pixel's corresponding 3D point M into the camera's coordinate system (shown here) and projecting onto the image plane. The resulting location m is determined by similar triangles and depends on the camera's focal length f and principal point p.

4.2 Interpreting Gestures

When the user touches the screen, we first determine the number of contact points. If there is only one point of contact, we display the view from the game-camera, and select this camera to control. If there are multiple points of contact, we display and control the iso-camera.

In the first case, we use the mapping described in Figure 5 to map the point of contact to a pan and tilt angle. As mentioned in the previous section, if the point of contact is outside of the camera view, or if the user lifts his/her finger within one second of making contact, then the game-camera will move to center on the point of contact. Otherwise, we measure the displacement from the initial point of contact as the user drags his/her finger across the screen. The necessary game-camera pan/tilt angles to maintain the relative displacement to the point of contact is determined by adding the relative displacement vector to the camera's position at initial contact.

In the second case involving multiple points of contact, we calculate the centroid of the contact points on the screen to determine target pan and tilt angles on the unit sphere. We then compute target pan/tilt angles in the iso-camera's motor coordinate system using $\mathtt{Q}_i$.

5. EXPERIMENT

We conducted two user studies and a perception study to determine whether a broadcast using the One-Man-Band (OMB) interface is comparable to a broadcast generated by a Three-Person-Crew (TPC). Our hypothesis was that: *"there is no significant difference in perceivable quality between broadcasts generated using the OMB interface and a conventional TPC."*

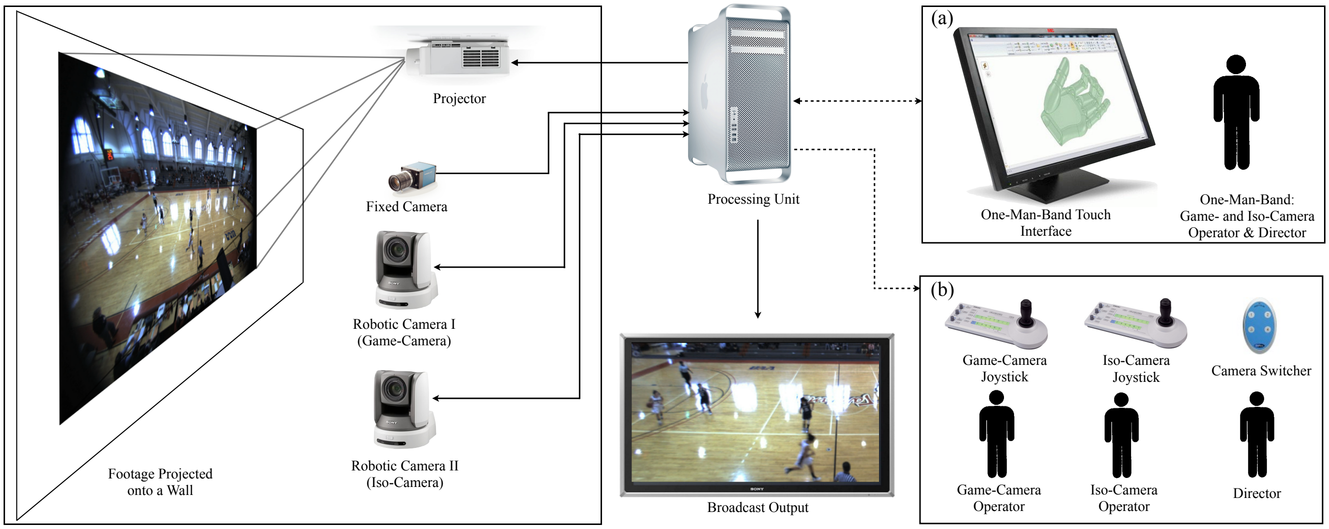


Figure 6: The setup used for both the OMB and the TPC experiments. A projector projects a full view of a basketball game onto a screen (left). Three cameras (two robotic and a fixed camera) are set up facing the wall with the following roles: i) one robotic camera captures the game view, ii) the other robotic camera captures the iso-view, and iii) the fixed camera records the whole field of view. Using this setup, we conducted two user studies: (a) For the OMB user study, all three cameras were connected to our touch screen interface. (b) For the TPC user study, each camera was connected to a joystick (i.e. game camera and iso camera) and the view was switched via a directors switching unit. Both studies yielded a broadcast video of the user studies which were stored, logged and analyzed to compare both setups.

5.1 User Study I: One-Man-Band

5.1.1 Participants and Apparatus

Thirty-three participants (15 females) aged 19-59 (mean 30.8) were recruited from a local university.

A schematic of the OMB system used in user study is shown in Figure 6(a). In order to create an experiment that was repeatable while being interesting for the participants, we first recorded a NCAA Division III college basketball game using a wide-angled camera. This view was wide enough to capture the entire court. Using a projector screen, we displayed the video from the camera in order to produce a repeatable live event. The two robotic pan-tilt-zoom cameras were pointed at the projector screen, with the field of view of the two broadcast cameras set such that they could only see a portion of the screen, which coincided with the zoom settings of the game- and iso-view camera. Using this setup, we allowed users to participate in a mock broadcast, using the cameras to capture the action on the screen just as they would with a live sporting event. By simulating the study in this way we could repeatedly generate identical conditions for the user studies.

5.1.2 Procedure

Participants were brought in individually and were tasked with simulating a broadcast using the OMB interface. To facilitate all studies, we edited out thirty-three (33) short clips from the wide angle video, ranging from 10 to 30 seconds long. Each clip contained a single "event" which would require the user to switch to the iso-view camera for a brief close-up. An "event" was defined as one of the following: (1) a player successfully scores a basket, (2) the ball goes out of bounds, or (3) a foul occurs.

Each participant first viewed the same short introductory video. This video explained: 1) the role of each person in a broadcast (i.e. director, game camera operator, iso-view camera operator), 2) how to use the interface, and 3) specific instructions for when to use the game camera and when to cut to the iso camera. Table 1 lists the specific instructions given to the users.

After receiving these instructions, each participant was shown a random assortment of twenty clips on the projector screen and used our OMB interface to record the action. The first ten clips were used as a practice run, so that the participants could get familiar to the interface. The second ten clips were recorded and used for analysis. Each study lasted approximately thirty minutes, and participants were compensated for their time.

5.2 User Study II: Three-Person-Crew Study

5.2.1 Participants

A different set of thirty-two participants (11 female) aged 20-55 (mean 24.9), were recruited to take part in the TPC study. We attempted to break these participants into groups of three, however due to four not showing up for the study, we ended up with eight groups of three (8 × 3) and four groups of two (4 × 2). To provide a fair comparison, we did not use the four groups of two. None participated in the One-Man-Band study.

5.2.2 Apparatus

The experiment used the same projector screen displaying a wide angle view of a basketball game, with cameras pointed at the screen. In this case, however, each of the robotic cameras was controlled by a separate joystick, and

1. The game-camera view (or wide-angled view) should capture a third of the court.

2. When the ball is moving from one side of the court to the other, frame the player with the ball on the last third of the frame (i.e., lead to where the play is going).

3. When the player with the ball arrives into the attacking third of the court, make sure that the hoop, backboard, and the player with the ball are all in view.

4. Cut to the iso-camera to get a close up of the player who either: (a) scored a basket, (b) touched the ball before it went out, (c) committed a foul or got fouled. If you are not sure who it was, give your best guess.

Table 1: The rules given to the participants for using the game-camera, iso-camera and when to cut between these two views.

the views from the two broadcast cameras were each displayed on their own monitor. A video switcher was used to change the view, which was used by the director. A diagram of the setup is given in Figure 6(b).

5.2.3 *Procedure*

Each group was shown the same introductory video as the previous study which explained the roles of production crew member (game-camera operator, iso-camera operator, and director), in addition to what each person should do when an "event" occurs. Using the setup depicted in Figure 6(b), one participant of the group was assigned to operate the game-camera, another to the iso-camera and the third to the role of the director. The director decided which view to broadcast and would give instructions to the other camera operators if desired.

For this study, a random assortment of forty clips were played on the projector screen for each group. As with the OMB study, the first ten were used as a practice run and data was not recorded. For the eight groups of three, during the practice run, the three participants rotated positions every three clips (leaving four clips for the last rotation), so that everyone was able to practice each role. Data was recorded for the last thirty (30) clips shown.

5.3 Perception Study: Method and Results

5.3.1 *Participants*

To gauge what naive observers thought of the resulting broadcasts that came from either the OMB and TPC user studies, thirty-one participants (12 females) aged 18-54 (mean 24.8) were recruited from a local university. None of these participants were used in either of the previous user studies.

5.3.2 *Apparatus*

Each participant viewed the broadcast videos from the OMB and TPC user studies. Two perception studies were conducted. The first used a computer program to display the videos and had three sliders underneath the viewing window which were used to indicate the viewer's ratings (i.e., 1-5 rating scale). The second study was to compare OMB and TPC videos generated for the same basketball clip, which were displayed side-by-side. A single slider underneath the videos was used to indicate preference, and a text box could be used to record comments.

5.3.3 *Procedure*

Each participant was shown an assortment of twenty recorded clips which were randomized each time. Ten were recorded by users in the OMB study, and ten from the TPC study. The clips were shown in random order, and participants rated each clip on a scale from 1 to 5 in three categories:

1. Overall quality,

2. Appropriateness of cuts, and

3. Game-camera smoothness.

Participants were shown examples of "good" and "bad" clips for each category before beginning the study. These examples were based on the opinion of an independent expert who did not know which interface was used.

For the second part of the study, participants viewed pairs of clips side by side. Both recordings were of the same game event, with one clip being taken from the OMB interface and the other from the TPC setup. Participants then selected which recording they thought was better at capturing the event.

5.3.4 *Results*

The results of the perception study are summarized in Table 2. These results were calculated from each participant's mean rating. T-tests were used to compare mean ratings for each group on the three questions.

	Overall	Cut	Game-Camera
	Quality	Quality	Smoothness
	[1-5]	[1-5]	[1-5]
OMB	2.87 (0.40)	2.70 (0.39)	3.28 (0.38)
TPC	3.08 (0.48)	2.65 (0.44)	3.52 (0.40)
p-value	0.061	0.599	0.016

Table 2: Mean (and stdev) and the p-value associated with the t-tests comparing the One-Man-Band to the three-person crew broadcast based on the perception ratings using a 5 point scale (1-5).

We see that for the categories of overall quality and cut appropriateness, the average ratings were not significantly different, and therefore the evidence indicates that a single user with our OMB interface can perform as well as a three-person crew with an existing interface. Only the category of game-camera smoothness showed a significant but slight preference for the three-person-crew results. For the side-by-side analysis, we compared the OMB to TPC clips and we found that participants selected the OMB produced clip 46% of the time, which is not significant ($p > 0.05$).

5.4 Quantitative Analysis: Results and Discussion

Prior to any user or perception studies, we first established "ground-truth" positions for each of the video clips.

The ground-truth positions were determined on a frame-by-frame basis for each clip using the rules given in Table 1. These rules define a "correct" way to film each clip [11]. The ground truth camera positions were generated according to these same rules; however, to eliminate the possibility of human error, we did not record the ground truth positions in realtime from a human camera operator. Instead, we stepped through each of the 33 sequences frame-by-frame, and determined the "text-book" camera position and selection for each frame. Although there is subjectivity in determining how "good" a broadcast is, we can use these results to get a relative indication on how close both interfaces are to a baseline broadcast.

After both user studies were conducted, we compared the position logs from both the OMB and TPC to our ground truth data. By comparing ground truth camera positions to those recorded by the user studies, we obtained some quantitative sense of how well users performed against an ideal broadcast.

	Correct Camera Selection [%]	Average Pan Error [°]	Average Tilt Error [°]
OMB	**79.01** (4.65)	5.17 (1.03)	**1.17** (0.24)
TPC	73.69 (10.87)	**4.51** (0.99)	1.48 (0.44)
p-value	0.015	0.011	<0.001

Table 3: Mean (and stdev) and the *p*-value associated with the *t*-tests comparing the OMB and the TPC to the ground-truth broadcast for: 1) the average number of frames the user selected the correct camera, 2) the average pan error per user, and 3) the average tilt error per user. Statistically significant results are highlighted in bold typeface.

The results of the quantitative analysis are summarized in Table 3 and visualized in Figure 7. Similar to the results reported in the perception studies, these results refer to the mean performance for each user. In the second column, the results describe the percentage of frames which coincided with the camera that was selected in the ground-truth data (i.e. the higher the percentage the better). From these results it can be seen that the OMB was better ($p < 0.05$) than the TPC for selecting the correct camera. This result suggests that the director in the TPC studies was not as good as the single operator using OMB; possibly due to a lack of engagement as the director was seldomly involved. In terms of average pan error, the TPC was better than the OMB ($p < 0.05$), which is understandable as there was a noticeable lag in the camera-control of the robotic cameras. However, in terms of tilt error, the OMB was better than the TPC ($p < 0.001$).

An example of the ground-truth broadcast, compared to the broadcasts generated via the OMB and TPC setups are given in Figure 8(a). In this example, key-frames showing the camera selection and position are shown in columns A through D. In Figure 8(b), the pan and tilt positions, as well as the camera selection are given as functions of time. Based on these measurements we were able to gain some quantitative result comparing the OMB to the TPC setup.

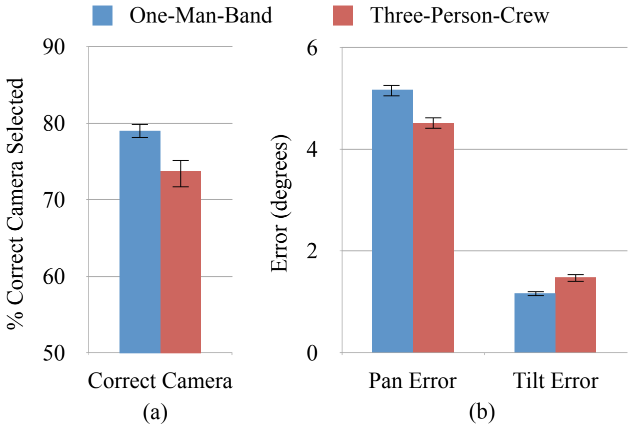


Figure 7: (a) The mean percentage of frames that users selected the correct camera. (b) The mean error in terms of pan and tilt across all the users for the OMB and TPC studies. The error bars on refer to the standard error.

6. SUMMARY AND FUTURE WORK

We present an interface which allows a single user to create a compelling broadcast of a sporting event by controlling multiple cameras simultaneously. Our experiments have shown that most novice users are able to learn the interface with very little practice, and can generate broadcasts that are approximately as good as those produced by a three-person-crew. Although the three-person-crew did slightly better in terms of camera smoothness, we showed that participants using our interface can do equally in terms of camera selection and pan-tilt positioning.

In the future, we plan to increase the number of cameras, as well as additional functionality such as replays and statistics. We will explore how these additional choices can be integrated into the touchscreen interface. For instance, while the user could still maintain direct control over the cameras (including when to switch between the game- and iso-views), the system could automatically suggest particulars cameras to cut to at opportune moments, or offer insightful statistical graphics during breaks in play.

7. REFERENCES

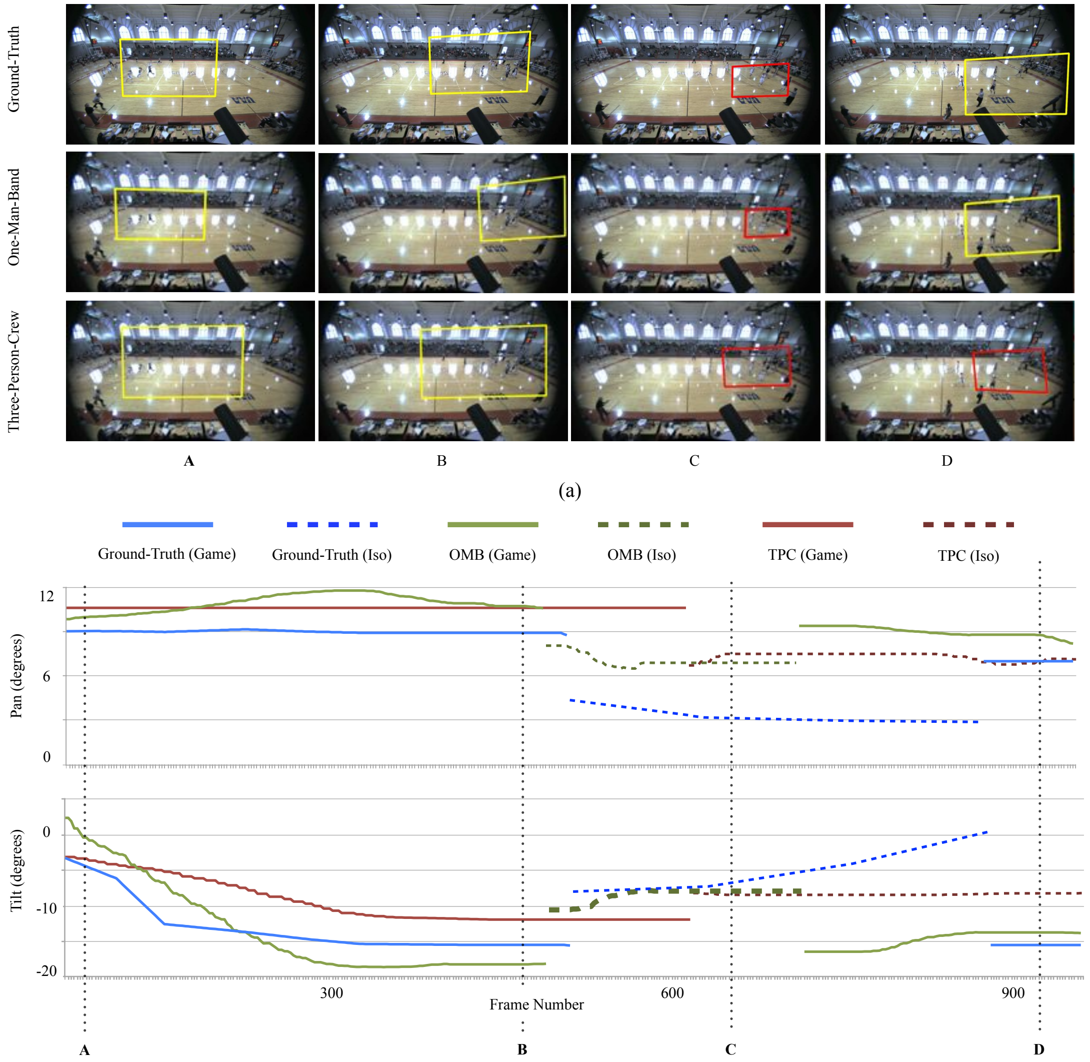
[1] S. Aiono, J. Gilbert, B. Soin, P. Finlay, and A. Gordon. Controlled Trial of the Introduction of a Robotic Camera Assistant for Laparoscopic Cholecystectomy. *Surgical Endoscopy*, 16(9):1267–1270, 2002.

[2] Y. Ariki, S. Kubota, and M. Kumano. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pages 851–860, 2006.

[3] M. Barange. Tabletop Interactive Camera Control. Master's thesis, INRIA, 2010.

[4] A. Cockburn, A. Karlson, and B. B. Bederson. A Review of Overview + Detail, Zooming, and Focus + Context Interfaces. *ACM Comput. Surv.*, 41:2:1–2:31, 2009.

[5] S. Drucker and D. Zeltzer. CamDroid: A System for Implementing Intelligent Camera Control. In



(a)

(b)

Figure 8: (a) Examples illustrating which camera and positions were selected for the various frames: (Top) ground-truth, (Middle) OMB, and (Bottom) TPC. (b) Curves comparing the camera pan (Top) and tilt (Bottom) angles for the ground-truth (blue), OMB (green), and TPC (red) results. The bold lines refer to the game-camera and the dotted refers to the iso-camera. The line going through the curves correspond to the example frames (A-D).

Symposium on Interactive 3D Graphics, pages 139–144. ACM, 1995.

[6] M. Gleicher and A. Witkin. Through-the-Lens Camera Control. In *SIGGRAPH*, volume 26, pages 331–340, 1992.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, United Kingdom, 2002.

[8] H. Kuzuoka, T. Kosuge, and M. Tanaka. Gesturecam: A video communication system for sympathetic remote collaboration. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 35–43. ACM, 1994.

[9] Q. Liu, D. Kimber, J. Foote, L. Wilcox, and J. Boreczky. Flyspec: a multi-user video camera system with hybrid human and automatic control. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 484–492, 2002.

[10] J. H. Matthews. Interactive Television System and Method for Viewer Control of Multiple Camera Viewpoints in Broadcast Programming, 02 1997.

[11] J. Owens. *Television Sports Production*. Focal Press, 2007.

[12] E. Pietriga, O. Bau, and C. Appert. Representation-independent in-place magnification with sigma lenses. *IEEE Transactions on Visualization and Computer Graphics*, 16(03):455–467, 2009.

[13] G. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 260–265, 1998.

[14] Y. Rui, A. Gupta, and J. J. Cadiz. Viewing Meetings Captured by an Omni-directional Camera. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 450–457, 2001.

[15] M. Sarkar and M. H. Brown. Graphical Fisheye Views. *Commun. ACM*, 37:73–83, 1994.

[16] K. Singh, C. Grimm, and N. Sudarsanam. The IBar: A Perspective Based Camera Widget. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 95–98. ACM, 2004.

[17] P. S. Strauss, R. C. Zeleznik, and A. S. Forsberg. Two Pointer Input For 3D Interaction. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 115–120, 1997.

[18] M. Tanimoto. Free-viewpoint television. In *Image and Geometry Processing for 3-D Cinematography*, volume 5 of *Geometry and Computing*, pages 53–76. Springer Berlin Heidelberg, 2010.

[19] R. Turner, F. Balaguer, E. Gobbetti, and D. Thalmann. Physically-Based Interactive Camera Motion Control Using 3D Input Devices. In *Scientific Visualization of Physical Phenomena*, pages 135–145, 1991.

[20] R. C. Zeleznik and A. S. Forsberg. UniCam - 2D Gestural Camera Controls for 3D Environments, 1999.