Collaboration Support Using Environment Images and Videos

(invited paper)

Qiong Liu, Don Kimber, Patrick Chiu
FX Palo Alto Laboratory
3400 Hillview Avenue, Building 4
Palo Alto, CA 94304, U.S.A.
liu@fxpal.com

***Abstract*—This paper summarizes our environment-image/video-supported collaboration technologies developed in the past several years. These technologies use environment images and videos as active interfaces and use visual cues in these images and videos to orient device controls, annotations and other information access. By using visual cues in various interfaces, we expect to make the control interface more intuitive than button-based control interfaces and command-based interfaces. These technologies can be used to facilitate high-quality audio/video capture with limited cameras and microphones. They can also facilitate multi-screen presentation authoring and playback, tele-interaction, environment manipulation with cell phones, and environment manipulation with digital pens.**

Collaboration support; control through image; control through video; control with a cell phone; control with a digital pen; remote control; collaborative and automatic camera control; tele-interaction; presentation authoring; device control; gesture based camera control; video production; video communication; video conferencing; webcams; collaborative device control; distance learning; interactive image/video.

I. INTRODUCTION

To achieve the goal of a worldwide collaboration with no barriers, we have to overcome the obstacles of distance and complexity. In other words, we must develop systems for sharing information, facilitating tele-communication, and simplifying management of complex technology. With this goal in mind, we invented various novel technologies for facilitating local and remote communications and collaborations in the past several years. Our vision in this direction is to interact with a remote location through environment images and videos. More specifically, we want users of our system to organize information and devices using images/videos of an environment, and share information and devices through environment images/videos. For example, if someone asks me to give a presentation in a conference room a thousand miles away from my office, what should I do? The simplest way to do this is to open a video link to that conference room, and give the presentation by dragging the presentation file from my desktop to the image of the screen. Similarly, we should be able to perform many other actions such as adjusting a microphone on the other side when the sound quality is not good.

In this paper, we summarize novel technologies we developed at FX Palo Alto Laboratory in the past several years. More details of these technologies can be found in [1-8].

A. Video/audio Capture

Our investigation started from how to capture video according to users' needs [4]. In a relatively large conference environment, if we only use a fixed overview camera, we may miss details of a monitored meeting. On the other hand, if we only use a Pan/Tilt/Zoom (PTZ) camera, we may miss chances of finding interesting events in a meeting. To overcome this problem, we built a hybrid camera, called FlySPEC, by installing a PTZ camera on top of a panoramic camera (Figure 1). We also developed a camera control interface shown in Figure 1. With this interface, a user can select the close-up view with gestures in the overview window (video). By using the hybrid camera and this interface, our system can provide overview and necessary details at the same time to a user. Moreover, users' close-up view selections can also help the system to select proper microphones for high-quality audio pickup.

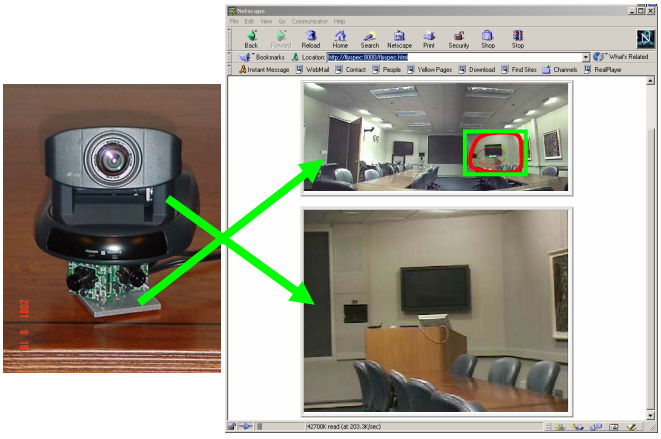

Figure 1. Our hybrid camera (FlySPEC) and its control

The basic FlySPEC system construction described in the previous paragraph can be used to serve point-to-point meetings. If users at different locations want to access the same conference room, they may compete for the camera control. To deal with multiple video requests, we compose an image canvas using signals from the panoramic camera, the PTZ camera, and an image cache, and use the canvas to serve users according to their requests. Based on users' requests and image content analysis, we developed a PTZ camera management algorithm for the best overall video quality with limited cameras [5]. Later, we also extended this management algorithm to control multiple FlySPECs (MSPEC) [7].

A system with one or more FlySPECs can allow meeting participants to control their own views. However, continuous camera operation is still a tedious task for humans. To deal with this problem, we designed a system that can automatically produce meeting video when participants don't, won't, or can't operate the camera [4,5,7]. In other words, this system will take over the camera control task if no participants want to control the camera system. In this system, we have a software cinematographer to control the video capture of each PTZ camera. We also have a software director to select the best video output from multiple video streams. Because the director can help cinematographers to hide their mistakes, this system design allows the system to compose a better video stream than a single hybrid camera system. To make the system more adaptable to various video capture environments, the cinematographer module and director module are designed to learn video composition from human requests made with the FlySPEC interface.

B. *Teleinteraction and Control Event Authoring*

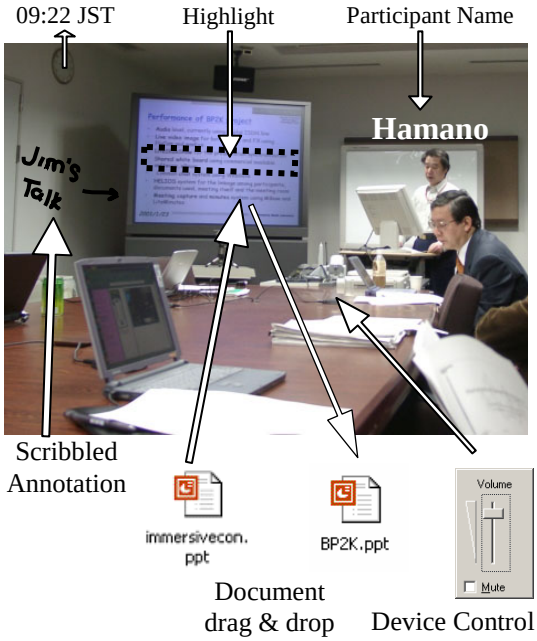

Figure 2. Scenarios for using shared interactive video in a teleconference.

Beyond camera control and meeting capture, a natural extension of our FlySPEC system is to facilitate more general control of other devices visible in environment images and videos [2,3,6]. With this extended idea, environment images and videos become active interfaces. Figure 1 shows some of the interaction possibilities available with this idea (though we have not fully implemented all of them). Through this type of interfaces, an authorized meeting participant can show a presentation on a screen by dragging the file on a personal laptop and dropping it on the video image of the presentation screen. Similarly, dragging a file onto a printer image will cause the printer to print that file. We may also use this interface to add highlights and annotations, check the local time of a remote site, or adjust a remote microphone. We built a system, called EPIC [6], to enable this kind of general control. An extra function of the real system enables users to preprogram control events in a conference room. In a conference room with multiple displays and many other supporting devices, this extra function gives users more choices for conveying information to others.

C. *Interaction Support with Cell Phones and Digital Pens*

Figure 3. Showing a document hardcopy on a display

Figure 4. The deployment environment and interface of POEMS.

The EPIC system can help people to interact with remote devices. It is reasonable to present this type of control interface on a large display or desktop. However, it is not very easy to deploy it for a large amount of audience members. For example, if 20 people have a discussion in a conference room, and each of them have spontaneous ideas or documents to show, it will be hard to give each person a laptop with the EPIC interface. Passing laptops among the discussion group is also awkward in that scenario. To overcome this problem, we developed a special interaction approach for cell phone users [8]. With our approach, a user is able to move documents among electronic devices, post a paper document to a selected public display, or make a printout of a white board with simple point-and-capture operations. More specifically, the user can move a document from its source to a destination by capturing a source image and a destination image in a consecutive order.

Figure 3 shows an example of posting a document hardcopy on a large public display.

To facilitate user-environment interaction and system deployment, we also developed a system (Figure 4), called POEMS (Paper Offered [Oriented?] Environment Management Service) [2], which allows meeting participants to control services in a meeting environment through a digital pen and an environment photo on digital paper.

The rest of the paper reports on our exploration in these directions in more details.

II. VIDEO CAPTURE

How to capture video and audio data according to users' needs is a basic problem we have to face for tele-communication and tele-collaboration. To overcome this problem, we have to design hardware that can capture signals according to various requirements. On the other hand, we also need to design interfaces to collect users' requirements for good service. Additionally, the system should minimize the burden on each user. In this section, we will report our approaches for tackling the video/audio capture problem in these three directions.

A. The FlySPEC Camera

Figure 1 shows a FlySPEC camera. This is a hybrid camera constructed by combining a PTZ camera and a panoramic camera. The panoramic camera covers a very wide field of view, and can serve different video requests through electronic pan/tilt/zoom. The PTZ camera can be used to capture details of small objects. The close proximity of the panoramic and PTZ cameras makes it easy to find the correspondence between the PTZ view and a region in the panoramic view. This correspondence is useful for controlling the PTZ camera based on low-resolution panoramic video. It also allows seamless switching of the video source between the panoramic and PTZ cameras.

B. The Camera Control Interface

The camera control interface for collecting a user's request at one view point is also shown in Figure 1. In the web browser window, the upper window shows a resolution-reduced video from the panoramic camera, and the lower window shows the close-up video produced by the FlySPEC system. In other words, the panoramic camera view provides sensory information about the environment to the human operator, and the lower window provides output video to the operator for feedback. Using this interface, the human operator adjusts the video output by selecting an interesting region in the panoramic view with a simple mouse-based gesture. After the interesting region is marked with a line or a circle, the region inside the bounding box of the mark will be shown in the close-up view window.

We also designed an interface for collecting a user's video preference of multiple view points. This interface is shown in Figure 5. In this interface, the three panoramic views come from panoramic cameras of three FlySPECs, and the close-up view comes from one selected FlySPEC camera. With this interface, a user can select a close-up view based on a gesture performed in one of the three overview windows. This design gives a user more chances to select a better stream when one FlySPEC is not enough to handle the capturing task well. The interfaces for controlling one or more FlySPECs can also be delivered to each end user through web page to maximize the collection of user requests.

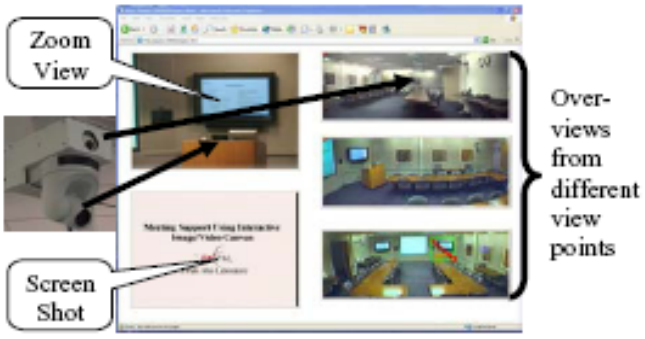

Figure 5. An interface for controlling multiple FlySPECs

C. The Camera Control Architecture and Algorithms

1) Architecture and general control strategy: After we have the hardware and interface, the next thing we need to think about is the control architecture and control algorithms. With a proper architecture and algorithm, we can gracefully distribute the camera control burden to multiple users and use machine intelligence to further minimize every user's control efforts. To achieve this goal, we designed following architecture and algorithm (Figure 6) for one FlySPEC control. With this architecture, the system captures the real world image with a panoramic camera and a Pan/Tilt/Zoom camera. Based on captured images, the system can make an estimation of the real world image. It can also estimate the spectra of the difference between the captured image and the real image based on a 1 over f squared model. Since various users have interests to various portions of the image, the image difference is weighted according to users' interests. The control strategy of this system is to move the PTZ camera to minimize the overall distortion of users' views.

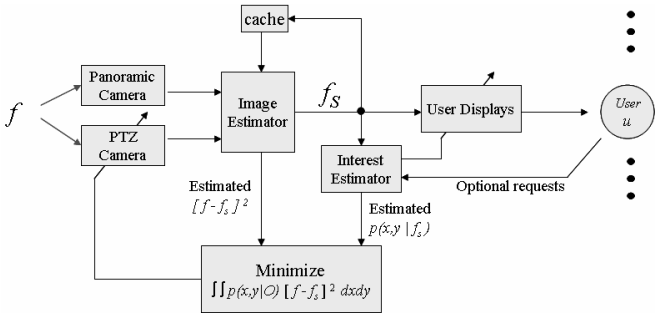

Figure 6. Minimum distortion control architecture

In our formulation, we use the video reaching each FlySPEC camera as the best quality video which has infinite spatial and temporal resolution. Let $f(x,y,t)$ be the ideal video, where x and y are panoramic image canvas coordinates

and t denotes time. Due to limited resolution of imaging sensors, a FlySPEC camera may only obtain an approximation $\hat{f}(x,y,t)$ of the ideal signal $f(x,y,t)$. Various regions of $\hat{f}(x,y,t)$ are transmitted to viewers according to their requests. To improve the video quality for each viewer, we have to improve $\hat{f}(x,y,t)$ estimation to reduce the difference between the displayed videos and the ideal video. To simplify our camera control problem, we tentatively assume all users' displays have enough resolutions to show $\hat{f}(x,y,t)$ without information loss.

With FlySPEC hardware and the control architecture proposed in Figure 6, there are two ways to improve video quality for viewers. First, the system can change the PTZ camera pose to improve $\hat{f}(x,y,t)$ estimation. Second, the system can use a buffered high quality image, $\hat{f}_{t-T}$, to substitute for $\hat{f}(x,y,t)$ when some image regions do not change much over a short time period T between consecutive video frames.

Denote $\{R_i\}$ as a set of non-overlapping small regions, N as the total number of requests, and $p(R_i,t \mid O)$ as the probability of viewing region-R_i details conditioned on environmental observation O at time t (e.g. the probability of viewing region-R_i when skin-color, body shape etc. appear in that region.) The total weighted distortion $D[\hat{f}_{t-T}, f_t]$ between users' requested images and the real image can be estimated with:

$$D[\hat{f}_{t-T}, f_t] \approx \sum_i p(R_i,t \mid O) \cdot \int_{R_i} |\hat{F}(\omega_{xy}, t-T) - F(\omega_{xy}, t)|^2 \, d\omega_{xy}, \quad (1)$$

Since all cameras have limited resolutions, $\hat{f}(x,y,t)$ is typically modeled as a band limited representation of $f(x,y,t)$ with cutoff frequency determined by the resolution of a camera. Let $F_{R_i}(\omega_{xy},t)$ and $\hat{F}_{R_i}(\omega_{xy},t)$ be the spectrum representation of $f(x,y,t)$ and $\hat{f}(x,y,t)$ respectively, where ω_{xy} is the rotational spatial-frequency. The band limited model assumes $\hat{F}(\omega_{xy},t) = F(\omega_{xy},t)$ below certain spatial-frequency $a(t)$ and $\hat{F}(\omega_{xy},t) = 0$ above the frequency. Let $F_{M,t}$ be $F(\omega_{xy},t) - F(\omega_{xy},t-T)$ and $F_{S,t}$ be $F(\omega_{xy},t) - \hat{F}(\omega_{xy},t)$, the above integration may be estimated with:

$$\begin{aligned} &\int_{R_i} \left| \hat{f}_{t-T} - f_t \right| dxdy = \int_{R_i} |\hat{F}(\omega_{xy}, t-T) - F(\omega_{xy}, t)|^2 \, d\omega_{xy} \\ &= \int_{R_i, \omega_{xy} \le a_i(t-T)} |F_{M,R_i,t}|^2 \, d\omega_{xy} + \int_{R_i, \omega_{xy} > a_i(t-T)} |F_{S,R_i,t}|^2 \, d\omega_{xy} \end{aligned} . \quad (2)$$

This integration reflects the distortion between the real image and the cached image, where the first term on the right side reflects the distortion caused by environmental changes, and the second term reflects the distortion caused by environmental details missed because of the limited resolution of the cached image. By sampling region R_i at frequency $a_i(t)$ and updating the cached image, the expected distortion reduction is:

$$\Delta D_{R_i} = \begin{cases} \int_{R_i, \omega_{xy} \le a_i(t-T)} |F_{M,R_i,t}|^2 \, d\omega_{xy} + \int_{R_i, a_i(t) \ge \omega_{xy} > a_i(t-T)} |F_{S,R_i,t}|^2 \, d\omega_{xy} & * \\ \int_{R_i, \omega_{xy} \le a_i(t-T)} |F_{M,R_i,t}|^2 \, d\omega_{xy} - \int_{R_i, a_i(t) < \omega_{xy} \le a_i(t-T)} |F_{S,R_i,t}|^2 \, d\omega_{xy} & ** \end{cases}$$

$$where \quad \begin{cases} * & a_i(t-T) \le a_i(t) \\ ** & a_i(t-T) > a_i(t) \end{cases}. \quad (3)$$

In our system, the sampling frequency of a region is directly related to the camera zoom level at that region. Therefore, the above distortion can be adjusted by changing the camera zoom level associated with region R_i. With equation 1-3, the total distortion reduction (information gain) over all requested images is proportional to:

$$\Delta D \approx \sum_i p(R_i, t \mid O) \cdot \Delta D_{R_i}. \quad (4)$$

To improve video quality, the control strategy of our system is to maximize the distortion reduction ΔD by using proper cameras (i.e. the PTZ camera, the panoramic camera, or no-updating) to update the cached image. Denote *(P,T,Z)*, corresponding to pan/tilt/zoom, as the best pose for the PTZ camera. *(P,T,Z)* can be obtained with

$$(P,T,Z) = \arg\max_{(p,t,z)} (\Delta D), \quad (5)$$

where *(P,T)* decides the location of the updated regions and *Z* decides the sampling frequency of those updated regions.

With the above control equations, the system can move each PTZ camera to form a very high-resolution image for future requests when the environment is static. In a dynamic environment, the algorithm will guide the PTZ camera to follow moving objects that interest most viewers.

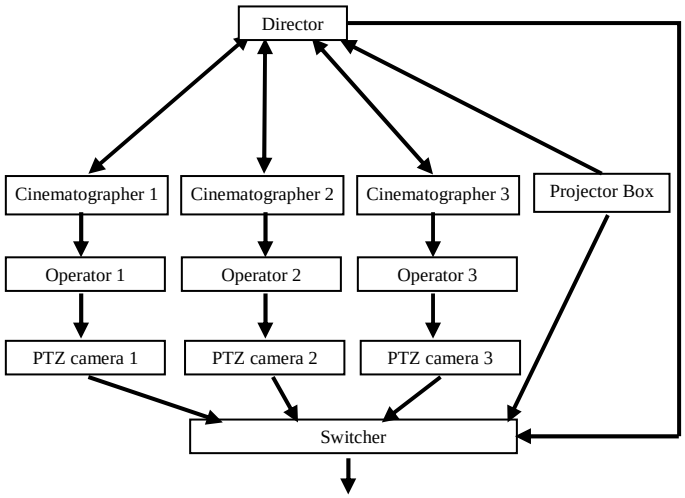

Figure 7. System architecture for video production multiple FlySPEC cameras

The control architecture in Figure 6 and equations 1-5 make the control of one FlySPEC camera possible. To control multiple FlySPEC cameras, we built a system, called MSPEC,

according to the architecture illustrated in Figure 7. In this system, we assigned a software cinematographer to control the video capture of each PTZ camera. We also assigned a software director to select the best video output from multiple video streams. To use equations 1-5 within this architecture, we have to generalize the distortion estimation from one FlySPEC camera to multiple FlySPEC cameras. More specifically, we have to estimate the probability of selecting a region based on available regions from all FlySPEC cameras. Additionally, by putting images in non-overlapping regions in one coordinate system, $f(x,y,t)$ can be used to represent the real signal at all camera locations, and $\hat{f}(x,y,t)$ can be used to represent the estimated signal at all camera locations. Denote *q* as a PTZ camera id number, *Q* as the best PTZ camera for the zoom view in the above interface, cinematographer *q* can estimate $\Delta DMAX_q$ as the maximum distortion reduction of camera *q* and adjust camera *q* accordingly. On the other hand, each cinematographer will send its estimated $\Delta DMAX_q$ to the video composition director and the director can select the best camera *Q* with

$$Q = \arg\max_q \left(\Delta DMAX_q\right). \quad (6)$$

After *Q* is computed the director will sends it to the video switcher for selecting the best output video stream.

D. Estimating the Distortion Reduction from an Image Cache Update

Since the system cannot try all PTZ camera poses in practice, it has to seek the optimal camera pose via simulation before moving each PTZ camera. More specifically, the system has to try the distortion reduction equations (3) and (4) with sampling regions and cutoff frequencies corresponding to various camera poses, and select the optimal camera pose based on equation (5).

During computer simulation, accurate estimation of equation (3) is difficult without sufficient camera resolution. To compensate for this problem, we use Dong and Atick's image/video power spectrum models [3] to assist the evaluation of distortion reduction corresponding to various poses. According to these models, if a system captures object movements from distance zero to infinity, $|F_{S,R_i}|^2$ and $|F_{M,R_i}|^2$ statistically fall with spatial frequency, ω_{xy}, according to $1/\omega_{xy}^m$ and $1/\omega_{xy}^{m-1}$ respectively, where *m* is around *2.3*.

Based on these simple models and available images, the estimation of each distortion term may vary. Due to space limitations, we only give the estimation procedure of a typical case. More specifically, we assume that only the panoramic videos are available for the estimation. Let *b* be the spatial cutoff frequency of a panoramic video. Since the panoramic video is available for cache update at any time, *b* cannot be larger than the spatial cutoff frequencies of cached images. In other words, we have $b \le a_i(t)$, and $b \le a_i(t-T)$. Let $E_{s,i,t}$ be the R_i-region AC-power between spatial frequency *1* and *b*, $E_{m,i,t}$ be the R_i-region frame-difference AC-power between spatial frequency *1* and *b*, $J_{m,i,t}$ be the R_i-region frame-difference power up to spatial frequency *b*, and $\hat{f}_b(x,y,t)$ acquired by the panoramic camera be a band-limited representation of $f(x,y,t)$. $J_{m,i,t}$ can be estimated with:

$$J_{m,i,t} = \int_{R_i} \left| \hat{f}_b(x,y,t) - \hat{f}_b(x,y,t-T) \right|^2 dxdy. \quad (7)$$

$E_{s,i,t}$, $E_{m,i,t}$ can be estimated in a similar way. With these values, terms for $\Delta D_{c,R_i}$ may be obtained with:

$$\begin{aligned}
\int_{R_i, a_i(t) \ge \omega_{xy} > a_i(t-T)} |F_{S,R_i,t}|^2 \, d\omega_{xy} &= \frac{[a_i(t) - a_i(t-T)] \cdot b}{a_i(t) \cdot a_i(t-T) \cdot (b-1)} \cdot E_{s,i,t} \\
\int_{R_i, a_i(t-T) \ge \omega_{xy} > a_i(t)} |F_{S,R_i,t}|^2 \, d\omega_{xy} &= \frac{[a_i(t-T) - a_i(t)] \cdot b}{a_i(t) \cdot a_i(t-T) \cdot (b-1)} \cdot E_{s,i,t} \quad (8) \\
\int_{R_i, \omega_{xy} \le a_i(t-T)} |F_{M,R_i,t}|^2 \, d\omega_{xy} &= J_{m,i,t} + \frac{1 - [b/a_i(t-T)]^{0.3}}{b^{0.3} - 1} \cdot E_{m,i,t}
\end{aligned}$$

By using the estimation approach described in equations 6-7, the system can estimate the high-frequency distortion based on low-frequency captures and move the PTZ camera according to this estimation for minimizing overall video distortions. Moreover, this approach gives more emphasis to regions with motion and fine texture. These emphases align well with many cinematographer rules in [9].

E. Weighting Distortions According to Users' Requests

To compute the distortion of all requests, users' requests to different portions of an image are modeled with a probability function $p_t(R_i \mid O)$. This gives rise to the form of a Bayes estimator. $p_t(R_i \mid O)$ may be estimated directly based on users' requests. Assume *N* is the total number of requests and n_i users request the view of region R_i during the time period from *t* to *t+T* when the observation *O* is presented, and *p* and *O* do not change much during this short period, $p_t(R_i \mid O)$ may be estimated with:

$$p_t(R_i \mid O) = \frac{n_i}{N}. \quad (9)$$

With this weighting function estimation approach, the system gives human requests the highest priority for controlling the system, and gracefully distributes the camera control task. More specifically, when a single user marked region is requested, the control system will immediately move the PTZ camera to focus on the user marked region. On the other hand, when many users mark different regions, the control system will give more weights to common interests while it takes care of each individual.

F. Video Distribution according to Users' Requests

In previous sessions, we described ways to control the PTZ camera according to users' requests. With a controlled PTZ camera, a panoramic camera, and some cached images, we can have a multi-resolution estimation (canvas) of the real world. This canvas can then be used to serve each individual through canvas cropping and digital zoom. In this way, each user can

have an independent virtual camera for the monitored event. This independent virtual camera is useful for predictable video output in an operator's control interface.

G. *Automate Video Composition without Users' Requests*

When users' requests are not available, the estimation of $p_t(R_i \mid O)$ may become a problem. This problem may be tackled by using the system's history of users' requests. More specifically, if we assume that the probability of selecting a region does not depend on time *t*, the probability may be estimated with

$$p_t(R_i \mid O) = p(R_i \mid O) = \frac{p(O \mid R_i) \cdot p(R_i)}{p(O)}. \quad (10)$$

Figure 8. The learning process of $P(R_i)$. The left image shows the arrangement in a conference room. The right image shows the 'learned' $P(R_i)$ after requests collection from users.

In a tele-conferencing environment, it is reasonable to assume that signals from different sources (i.e. objects), such as a presenter or an audience member, are independent. It is also reasonable to assume that a human's view selection separates various sources well into two categories (i.e. proper segmentation). Based on these assumptions, the feature vector O may be separated into independent feature vectors O_i and O_{other}, where O_i is the feature vector based on the data in R_i and O_{other} is the feature vector based on the data outside of R_i. Moreover, we can further assume that R_i and O_{other} are independent. With these assumptions, $p(R_i \mid O)$ may be estimated with

$$p(R_i \mid O) = p(R_i \mid O_i, O_{other})$$
$$= \frac{p(O_i \mid R_i, O_{other}) \cdot p(R_i, O_{other})}{p(O_i, O_{other})} = \frac{p(O_i \mid R_i) \cdot p(R_i)}{p(O_i)}. \quad (11)$$

The observation O_i may be further separated into 'independent' features $O_i = \{o_1, o_2, \ldots, o_n\}$. With these independent features, $p(R_i \mid O)$ may be estimated with

$$p(R_i \mid O) = \frac{p(o_1 \mid R_i) \cdot p(o_2 \mid R_i) \cdots p(o_n \mid R_i) \cdot p(R_i)}{p(o_1) \cdot p(o_2) \cdots p(o_n)}, \quad (12)$$

where $p(R_i)$ is the prior probability of selecting region R_i, and $p(o_j \mid R_i)$ is the probability of observing o_j in R_i when R_i is selected. Probabilities on the right side of this equation may be 'learned' online. Figure 8 shows an example of this learning process. In this example, the system has no knowledge of users' preferences of different regions (i.e. a uniform $p(R_i)$). By presenting overviews to remote meeting participants and collect users' requests, it can 'learn' the importance of various regions ($p(R_i)$ shown in the right image). Similarly, the system can learn $p(o_j \mid R_i)$. With the $p(R_i \mid O)$ estimate available, it is straightforward to compute equation (5) for the optimal PTZ camera pose. This enables the system to automate video composition based on users' past selection patterns.

III. Teleinteraction and Control Event Authoring

As pixels and bandwidth cost less and less, we expect people to see a remote place through wall or other cheap surfaces before long. If people can see a remote place through wall size displays, it is nearly unavoidable that many of them will have the desire to control remote things they see on displays. Even with common existing displays, interacting with devices through live video of an environment is an intuitive and attractive technology.

Beyond live interaction, we may also use visual cues in an environment picture to assist control event authoring in a media rich environment. This technology may give people more choices for conveying information to others by using devices beyond a single display and a stereo audio channel.

A. *Shared Interactive Video*

This session focuses on our interactive video techniques to support meetings and teleconferences. For the purposes of discussion, we will define "local" as the physical meeting room, and "remote" as anywhere else.

In the literature, "interactive video" appears with several different meanings. Zollman considers it to be any video where the user has more than minimal on-off control over what appears on the screen, including random access, still frame, step frame, and slow play [13]. Tantaoui et al. use the term similarly in their paper [11]. VideoClix™ sees interactive video as hotspot enhanced recorded video [12]. Darrell's "interactive video environment" refers to a virtual reality system where people can interact with virtual creatures [14]. Aside from these, a relatively common usage in the distance-learning field is two-way video/audio/data communication shared among participants [15]. A more formally specific definition of interactive video, from [16] is: "A video application is interactive if the user affects the flow of the video and that influence, in turn, affects the user's future choices." The meaning we adopt in this paper is very close to the union of the last two definitions. More specifically, we view interactive video as a live video interface that allows users to affect and control real-world devices seen in the video.

We presented a live video supported camera control system in an earlier section. There, we described a hybrid camera system, called FlySPEC, which was constructed by installing a Pan/Tilt/Zoom (PTZ) camera on top of a panoramic camera. Users are presented with a panoramic video overview, and may select regions of that window for closer inspection in a close-up

video window. In other words, the panoramic video overview serves as the control interface for mouse actions that steer a PTZ camera (which provides a high-resolution close-up view).

A natural extension is to let users control other devices using related mouse or pen based gestures performed in video windows. For example, an authorized participant may drag a presentation file from a desktop to a remote presentation screen seen in the video window, causing that presentation to be remotely displayed. To facilitate this, the video may be augmented with metadata indicating the controllable devices, or annotations such as the names of participants.

Our work differs from the device control system proposed in [17], in that it does not require specific devices for control, nor does it require people to be in the controlled environment. Unlike the "graspable interface" that allows people to manipulate virtual objects with real "bricks" [18], our system operates real world equipment through the live video. Moreover, in contrast to the VideoClix™ or iVast™ products [12][19] that support metadata editing of recorded video, our system works on live video and controls real-world devices.

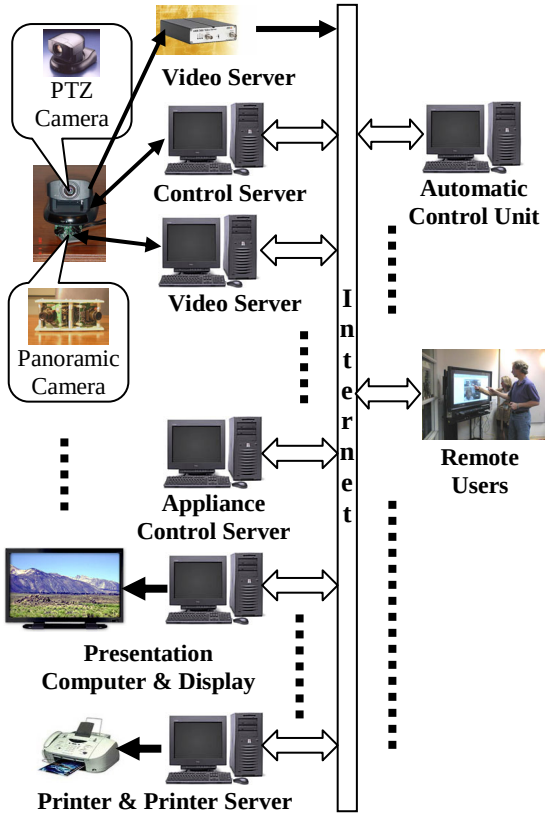

Figure 9. Connecting multiple control components through the Internet

Live video as a reference cue for power-plant control has been explored in [10]. Goldberg et al. use video as a reference cue to convey "Tele-Actor," a skilled human who collects the video and performs actions during navigation [20]. To our knowledge, more general video-enabled device control for meetings has not been presented in recent literature. By using a video interface to control devices in a meeting environment, meeting participants may easily overcome the limitations of their physical locations. For example, a remote meeting participant may use a mouse to draw figures on a local presentation screen from 2000 miles away. As we illustrate in Figure 1, a remote participant may also drag a copy of a given presentation from the video image of the (local) screen to a remote desktop, check the local time, or choose among microphones for better sound quality. If the remote person wishes to give a presentation, they may drag a presentation file onto the video image of the (local) presentation screen, and adjust the screen brightness for proper visual effect.

Our current implementation of this idea supports "interactive video" by augmenting the live video captured by our FlySPEC subsystem with device-control messages. Currently, it supports useful file operations like printing and local/remote file transfer.

A basic idea in our system is the "Video Canvas," which associates and orients all images, annotations and other metadata. A Video Canvas can be thought of as a shared blackboard onto which the cameras paste images and from which users request views, control devices seen in the views, and draw annotations to be shared with other users. Each interface provides an overview of the canvas, and a close-up view of some region. A meeting participant may control their close-up view by circling a region of interest in the overview. Furthermore, they may add annotations such as "name-tags" and digital ink marks, or transfer files using drag and drop operations.

Figure 9 shows the system architecture of our tele-interaction system. It is just a simple extension of our FlySPEC control system. Video from a controlled scene is captured using a hybrid camera that includes both a PTZ and a panoramic camera. Videos from both cameras are sent to users as sequences of JPEG images in real time. A camera control server running on a networked workstation controls the PTZ camera through a RS-232 serial link. The camera control server accepts HTTP control requests through the Internet/Intranet, and sends pan/tilt/zoom commands to the PTZ camera. Given the camera constraints, the camera control server optimally satisfies users with a combination of optically and digitally zoomed video.

Beyond the camera control, an appliance-control server manages appliances other than the cameras. This server translates control requests into specific device control commands. When this control server receives requests from remote users, it sends appropriate commands and data to local devices. With the architecture of Figure 9, we can add arbitrarily many cameras, screens, printers, and other devices in the system.

As seen on the right side of Figure 9, remote clients connect to the system through networks. Remote users can control the FlySPEC camera and other devices from these remote clients. They can also immediately see the effects of control actions through the video. We have implemented several different client applications specialized for different uses. With these applications, users can define hotspots and metadata in a video window; add annotations in the video; share drawings on a common area in a video; attach a name tag to a sitting

participant; drag presentation slides among screens shown in the same video window; drag a slide from a screen to a visible printer for printout; drag desktop files to remote screens, loudspeakers, and printers; or drag a slide from a remote screen to a tablet for storage or slide annotation etc.

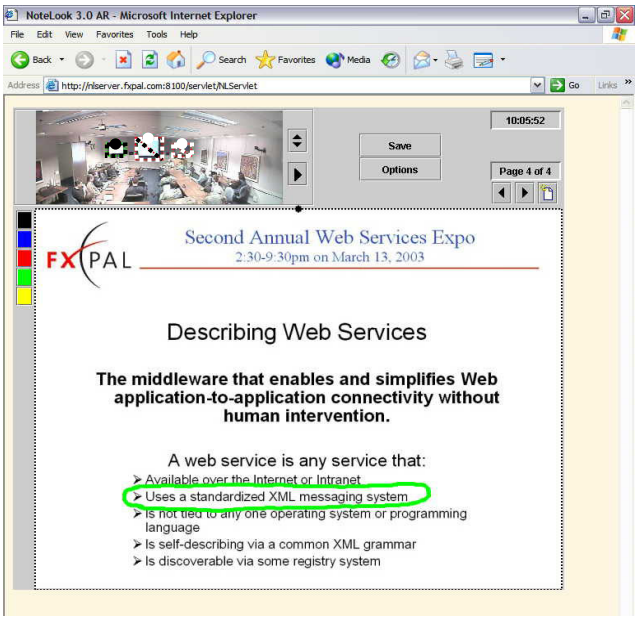

Figure 10. NoteLook 3.0 Screen Shot.

Figure 10 shows our NoteLook 3.0 interface [1]. In this interface, a live panoramic video of the meeting room is shown on top of the screen. On the right side of the video window, a spin button is used to cycle through multiple panoramic video sources. The 4 dashed boxes (three appear in the video window and one surrounds the note-taking page) are hot spots that can support gesture-based screen operations. The big dot on top of each dashed box is the handle for moving slides from that hot spot, while the slash in a hot spot bans a user from dragging slides from other hot spots to it. With this interface, a user can move a slide from the first hot spot to the second by dragging the handle of the first hot spot and put it in the second hot spot that authorizes this operation. For example, a user can drag the slide shown on the center public screen to the note-taking area. After making annotations on the slide (as shown in the figure), the user can drag this annotated slide to the left side public screen for discussion.

B. Control Event Authoring

US famous architects Charles and Ray Eames designed an astonishing presentation for American Exhibition in Moscow in 1959. That presentation, Glimpses of the USA, was a seven screen presentation of extraordinary rich image saturation. It showed 2200 images in a little over 12 minutes. Even though their presentation techniques are successful, setting up seven screens and properly arranging presentation materials on multiple screens are not easy tasks. As presentation devices get cheaper every day, presenters have more choices for conveying information to others with more media devices available in a meeting environment. However, existing presentation authoring and playback tools still lack handy functions for supporting devices beyond a single display and a stereo audio channel. This situation hinders presenters from using additional devices for presentation enhancement or tele-presentation.

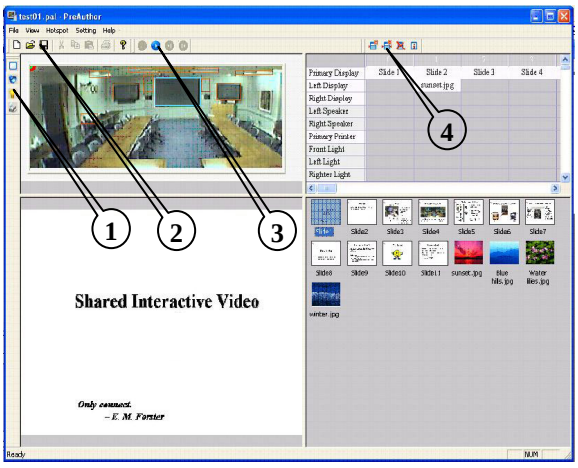

Figure 11. Main User Interface of EPIC (a)Top-left: A environmental image/video canvas depicts the device configuration of a venue. (b)Top-right: A device-state table reveals hyper-slides' relations for a unified presentation. (c)Bottom-left: A zoom pane for checking details of a user's selection. (d)Bottom-right: A hyper-slide pool shows thumbnails of all hyper-slides.

To encourage fuller utilization of media rich environments, we designed a presentation authoring and replaying tool, named EPIC, to facilitate presentation preparation and playback for arbitrary device configurations. The main idea of EPIC is to author a unified multi-device presentation using an image canvas (i.e. an environment picture). This idea can be achieved by associating presentation devices with various image regions. This tool is complementary to tools used for authoring specific media. It can organize media units prepared for simple devices and synchronously present coherent media units in one or more multimedia venues. In our prototype, various configurations of displays, printers, speakers and lights are supported.

Figure 11 illustrates the EPIC GUI, which includes an environmental image/video canvas for device references, a hyper-slide pool, a zoom pane for checking details of a user's selection, and a playlist for revealing which presentation unit is rendered on each device as the presentation progresses. The GUI also has four embedded tool bars, marked with callouts 1-4, for device definition, file manipulation, presentation control, and playlist manipulation.

During a preview or actual presentation, EPIC controls the mapping of media to devices according to the playlist. When a presenter triggers a slide change, EPIC synchronously changes the media rendered for each device, and records the event. The presenter may also make ad-hoc changes by dragging hyper-slides to various media devices using the GUI. The resulting history consists of a new playlist with timing information. This history may be used to index a multimedia recording of the presentation, but also the new playlist can be saved for future

presentations. Moreover, when a presenter gives a presentation at a remote site, the presenter may also use a mouse to mark a region in the video canvas and see the detailed video in the zoom pane.

There are 6 steps for authoring a presentation using EPIC:

1. Define hotspots based on a conference room picture or conference room pictures
2. Define web camera/microphone connections
3. Import media files
4. Author the play list and media files
5. Preview presentation
6. Show presentation.

The first two steps only need to be performed once for every meeting environment. The 3~5 steps are useful for a user to customize a presentation for a specific environment. If the user does not want to customize the presentation but still want to use various devices in the environment, s/he can set rules for authoring automation and let the system prepare the presentation.

Hotspots are canvas regions showing devices in a venue. They are useful for referring various devices seen in the canvas. In the video, hotspots are bounded with colored boxes. When a hotspot definition does not exist, a user may import an image file and use the mouse to draw bounding boxes on the image canvas. Whenever a bounding box is done, the EPIC software will pop out a dialog box for defining name, computer host, and port for the hotspot. A Right mouse-click on a hotspot may activate a menu for redefining or removing the hotspot. After all hotspots are defined on an image canvas, the definition can be saved in a file for future imports. During authoring or an actual presentation, a user may define the playback device of a presentation unit by dragging the thumbnail of the presentation unit to a hotspot for the device.

There are two types of settings for EPIC. The first type of settings allows people to define, save, and import web connections of cameras used in a teleconference. The second type of settings allows people to define a presentation mode (i.e. remote presentation, local presentation, or preview.)

EPIC can accept various media files, and manage presentation units from all imported files. After all related media files are imported, the user may customize playback device and event time for every presentation unit.

A playlist is used to organize various presentation units into a unified presentation. It is also useful for revealing presentation units' relations on a display. Each row of playlist corresponds to an available device, while each column of playlist corresponds to an indexed state, which is used to synchronize presentation units' playbacks on various devices. The authoring process of EPIC is to find proper playlist slots for all presentation units. Users may define a slot for a presentation unit by dragging the thumbnail of the presentation unit to the slot.

After a user defines a playback device for each presentation unit, the user may set EPIC to preview mode, and use the presentation control tool bar to navigate the authored presentation. During the preview process, slide images will be inserted in hotspots to mimic the playback effect in the real environment. When EPIC is set to 'Remote Presentation Mode', the image/video canvas and zoom pane will show videos captured by remote cameras. When the user navigates the presentation, remote displays will be changed according to the playlist. When EPIC is set to 'Local Presentation Mode', the computer running EPIC UI will also be used as a display. A user can only use the keyboard to navigate the presentation when the UI display is covered by a slide. This mode is useful for a presentation in a local meeting room where the presenter does not need the meeting room video.

IV. INTERACTION SUPPORT WITH PORTABLE DEVICES

Even though systems described in previous sessions give us the chance to control cameras and devices through an image/video window on a large display, PC, or tablet. Those interfaces are not very convenient for wide deployment. For example, if a conference room can hold 20 or more participants, it will be too difficult for us to deploy this many PCs or tablets in the conference room for user-environment interaction or user-user collaboration. To deal with this problem, we designed control interfaces specialized for cell phones and digital pens.

A. Reach Through Capture

Since cell phones' displays are limited in size, directly using the MSPEC or EPIC control interfaces on a cell phone is not a good idea when users have better environment views through other channels (e.g. a large display). On the other hand, most cell phones have cameras on them and moving a cell phone around is as easy as moving a mouse. Based on these considerations and observations, we proposed a Reach Through Capture (RTC) approach for user-environment interaction. The main idea of this approach is to use mobile cameras to identify devices and enable functions associated with the identified devices. By using this method, a user can easily activate a device control interface on a cell phone. Moreover, the user can easily layout documents in a meeting scenario for better content visibility or usability.

Unlike a laser pointer or IR/RF based remote control [21], this approach uses images captured by a mobile camera to guide the device control. With this technology, existing camera enhanced mobile devices, such as cell phones and PDAs, will not need extra laser pointers, IR transmitters, mini-projectors, etc. for our control tasks. Powered by this technology, it will be easy for a person to control document placement with a cell phone. That can save people from the burden of moving back and forth, or tracking various remote controls in a meeting environment. Moreover, the control task does not demand fixed cameras and IR receivers in control environments. Different from controls based on LED (light emitting diode) or visual identity tags, this method does not need users to install LEDs or paste bar code labels on various objects for device control tasks [22, 23].

Based on the idea described above, we developed a system that can support users to move documents among electronic devices, post a paper document to a selected public display, or make a printout of a white board with simple point-and-capture operations. More specifically, the user can move a document from its source to a destination by capturing a source image and a destination image in a consecutive order. The system uses SIFT (Scale Invariant Feature Transform) features of captured images to identify the devices a user is pointing to, and issues corresponding commands associated with the identified devices. Figure 3 shows a usage scenario of this system.

B. A Paper Based Meeting Service Management Tool

To address the deployment problem in large conference rooms, we propose POEMS (Paper Oriented Environment Management Service) that allows meeting participants to control services in a meeting environment through a digital pen and an environment photo on digital paper. Unlike state-of-the-art device control interfaces that require interaction with text commands, buttons, or other artificial symbols, our photo enabled service access is more intuitive. Compared with PC and PDA supported control, this new approach is more flexible and cheap. It is also much easier to deploy. With this system, a meeting participant can initiate a whiteboard on a selected public display by tapping the display image in the photo, or print out a display by drawing a line from the display image to a printer image in the photo. The user can also control video or other active applications on a display by drawing a link between a printed controller and the image of the display. Figure 4 shows a deployment environment and interface of POEMS.

POEMS is a modularized meeting service management system which uses a digital pen and paper as its interface, which supports paper buttons, controls based on time-stamped drawings, scribbling, and a new type of interaction, Dynamic Buttons that a user can draw and define on the paper. Paper has been the most popular media in meetings for years due to its low cost and flexibility, and the pen is the major tool for writing notes on papers. Since many people use this media and tool set regularly, this set of interface requires minimum learning effort on the user side. Using pen and paper also guarantees low cost, and light weight. Moreover, as the meeting context and supporting hardware are rapidly changing, the proposed meeting support system is modularized to enable frequent reconfigurations.

POEMS is deployed in a corporate meeting room with three large public displays. Figure 4 shows the deployment environment and a meeting service management interface. With this tool, the screens are automatically started when a pen is pressed on a piece of paper. A presenter can control the presentation progress with time-stamped paper taps or drawings. A meeting participant can use the paper interface to initiate a white board on any public display for scribbles. Users can use a digital pen to point or write at any place on a screen for group discussions. They can also initiate an onsite screen printout by a time-stamped paper drawing from a screen image to a printer image.

REFERENCES

[1] P. Chiu, Q. Liu, J. Boreczky, J. Foote, T. Fuse, D. Kimber, S. Lertsithichai, and C.Y. Liao, "Manipulating and annotating slides in a multi-display environment." In *Proc. of INTERACT '03*, pp. 583-590.

[2] C. Hu, Q. Liu, X.M. Liu, C.Y. Liao, P. McEvoy, "POEMS: A PAPER BASED MEETING SERVICE MANAGEMENT TOOL", in Proceedings of ICME 2007.

[3] C.Y. Liao, Q. Liu, D. Kimber, P. Chiu, J. Foote, and L. Wilcox, "Shared Interactive Video for Teleconferencing. In *Proc. ACM Multimedia* 2003, pp. 546-554 (11/2/2003).

[4] 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 *Proc.ACM Multimedia 2002,* pp. 484-492.

[5] Q. Liu, and D. Kimber, Learning Automatic Video Capture from Human Camera Operations. In *Proc. IEEE Intl.Conf. on Image Processing 2003.*

[6] Q. Liu, F. Zhao, and D. kimber, "Computer Assisted Presentation Authoring for Enhanced Multimedia Venues", in Proc. of ACM Multimedia 2004.

[7] Q. Liu, X. J. Shi, D. Kimber, F. Zhao, F. Raab, "An Online Video Composition System", In Proceedings of ICME 2005.

[8] Q. Liu, D. Kimber, H. Zhou, P. Chiu, "On Redirecting Documents with a Mobile Camera", In Proc. of IEEE Multimedia Signal Processing 2006.

[9] Q. Liu, Y. Rui, A. Gupta, and JJ Cadiz, "Automating Camera Management for Lecture Room Environments", in Proc. of CHI 2001, pp. 442-449.

[10] M. Tani, K. Yamaashi, K. Tanikoshi, M. Futakawa, and S. Tanifuji, "Object-oriented video: Interaction with real-world objects through live video." In *Proc of CHI* '92, ACM Press, pp.593-598, 711-712.

[11] Tantaoui, M.A., Hua, K.A., and Sheu, S. Interaction with Broadcast Video, *Proceedings of ACM Multimedia 2002*, pp.29-38.

[12] VideoClix™, VideoClix Authoring Software, *http://www.videoclix.com/videoclix_main.html*

[13] Zollman, D.A., and Fuller, R.G. Teaching and Learning Physics with Interactive Video, *http://www.phys.ksu.edu/perg/dvi/pt/intvideo.html*

[14] Darrell, T., Maes, P., Blumberg, B., Pentland, A.P. A Novel Environment for Situated Vision and Behavior, *MIT Media Lab Perceptual Computing Technical report*, No. 261, (1994).

[15] Mississippi ETV Interactive Video Network. *http://www.etv.state.ms.us/inter-net/home.html*

[16] *Stenzler, M.K., and Eckert, R.R. Interactive Video, SIGCHI bulletin, vol. 28, no.2, April 1996.*

[17] *Khotake, N., Rekimoto, J., and Anzai, Y. InfoPoint: A Direct-Manipulation Device for Inter-Appliance Computing, http://www.csl.sony.co.jp/person/rekimoto/iac/*

[18] Fjeld, M., Ironmonger, N., Voorhorst, F., Bichsel, M., & Rauterberg, M. Camera control in a planar, graspable interface, *Proceedings of the 17th IASTED International Conference on Applied Informatics (AI'99),* pp.242-245.

[19] *iVast™, iVast Studio SDK™ -- Author and Encode, http://www.ivast.com/products/studiosdk.html*

[20] *Goldberg, K., Song, D.Z., and Levandowski, A. Collaborative Teleoperation Using Networked Spatial Dynamic Voting, Proceedings of IEEE, Special issue on Networked Robots, 91(3), pp. 430-439, March 2003.*

[21] *C. Kirstein, and H. Müller, "Interaction with a Projection Screen Using a Camera-Tracked Laser Pointer." Proc. of The International Conference on Multimedia Modeling. IEEE Computer Society Press, 1998.*

[22] *N. Kohtake, T. Iwamoto, G. Suzuki , S. Aoki, D. Maruyama, T. Kouda, K. Takashio, H. Tokuda, "u-Photo: A Snapshot-based Interaction Technique for Ubiquitous Embedded Information" in Proc. of PERVASIVE200), pp.389 - pp.392, Linz/Wienna Austria, 2004.*

[23] *R. Sharp, "Overview: New Uses for Camera Phones." (Jul. 1, 2004), http://www.deviceforge.com/articles/AT5785815397.html*