Automatic Camera Control Using Unobtrusive Vision and Audio Tracking

Abhishek Ranjan[1], Jeremy Birnholtz[1,2], Rorik Henrikson[1], Ravin Balakrishnan[1], Dana Lee[3]

[1] Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4

[2] Department of Communication
Faculty of Computing &
Information Science
Cornell University - Ithaca, NY 14853

[3] School of Radio & Television Arts
Ryerson University
Toronto, Ontario M5B 2K3

aranjan@dgp.toronto.edu, jpb277@cornell.edu, rorik@dgp.toronto.edu ravin@dgp.toronto.edu, danalee@ryerson.ca

ABSTRACT

While video can be useful for remotely attending and archiving meetings, the video itself is often dull and difficult to watch. One key reason for this is that, except in very high-end systems, little attention has been paid to the production quality of the video being captured. The video stream from a meeting often lacks detail and camera shots rarely change unless a person is tasked with operating the camera. This stands in stark contrast to live television, where a professional director creates engaging video by juggling multiple cameras to provide a variety of interesting views. In this paper, we applied lessons from television production to the problem of using automated camera control and selection to improve the production quality of meeting video. In an extensible and robust approach, our system uses off-the-shelf cameras and microphones to unobtrusively track the location and activity of meeting participants, control three cameras, and cut between these to create video with a variety of shots and views, in real-time. Evaluation by users and independent coders suggests promising initial results and directions for future work.

KEYWORDS: Meeting capture, computer vision, automated camera control, video.

Index Terms: H.5.3 Group and Organization Interfaces.

1 INTRODUCTION

Geographically distributed work teams are an increasingly common facet of the modern workplace [8, 15]. Such teams enable organizations to more easily bring individuals with necessary skills and expertise to bear on difficult problems [8]. Despite these advantages, however, distributed teams often perform worse than collocated groups charged with similar tasks [26]. One key reason for this is difficulty in coordination and communication [7]. Given the amount of time spent in meetings [1] and the importance of meetings in coordination, there is a clear need for effective and improved technologies to support distributed participation in, and archival access to meetings.

While technologies such as videoconferencing have supported the transmission and recording of meetings (processes we refer to as “meeting capture”) for many years (e.g., [11]), many have not been regarded as successful. In particular, video from meetings has been described as boring or unengaging, as compared with face-to-face participation [27]. One reason for this is that many videoconferencing systems (e.g., a basic Polycom system [29]) use only one camera, and people rarely take the time to pan, zoom and otherwise control this camera to provide for visual variety and perhaps show a detailed view of what is taking place [28].

Even though people generally opt not to change the camera position during conferencing, combining a variety of shots is a technique that television and film directors often use to make their programs more compelling [21]. Adding a similar level of engagement to conferencing and capture technologies is challenging, however, because it is usually not cost effective to pay professional human camera operators. There has been some interest, though, in reducing these costs by automatically controlling and switching between cameras. At its root, this is a problem of understanding how to capture what is taking place, and present this dynamically to improve viewer experience.

In this paper we present a novel camera control technique aimed at improving videoconferencing. Our approach, motivated by principles from television production, is novel in two respects. First, it uses a robust, extensible and decentralized tracking and detection scheme consisting of multiple cameras and microphones. Second, it uses off-the-shelf technology and unobtrusive tracking. The system was evaluated using data from logs, users, and human coders, and found to perform well.

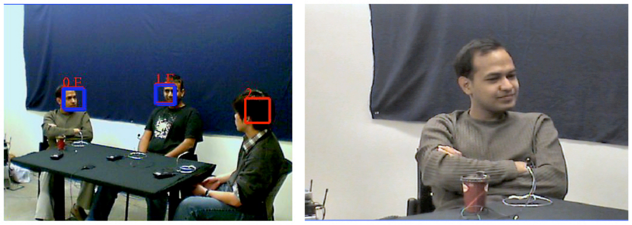

Figure 1. An example of locating a speaker’s face using vision based detection (left) and showing a close-up shot of the speaker (right) in our system

2 BACKGROUND

Videoconferencing is a common mode of interaction [28], but has been the subject of much criticism (e.g., [11]). Critics have called video boring and unengaging, and suggested that nonverbal exchanges and side conversations are not well supported [26]. As such, there have been attempts to improve the videoconferencing experience. Systems have been developed to more realistically replicate eye contact and gaze [25, 35], to support gesturing [16], as well as to improve the quality and variety of camera shots. In this paper, we focus on the latter of these.

We have chosen to focus on increasing shot variety and quality for two primary reasons. First, there is a rich set of principles and heuristics used by film and television directors for increasing viewer engagement. Second, we believe this is an accessible way to improve videoconferencing experience.

2.1 Camera Control: The Challenges

Camera control can involve either using one camera to provide a variety of shots, or cutting between several cameras for multiple views. A single camera is appropriate when activity is taking place within the camera’s range of possible views, and where very frequent and drastic shot changes (i.e., needing close-ups of people at opposite ends of the room) are unlikely. One such scenario is a lecture room in which a single speaker dominates the audience's attention. Some systems [3, 23, 34] track the location of the speaker at the front of a room, and use this to control a camera that maintains a "waist-up" shot of the speaker.

When activity is taking place in a larger area or frequent shot changes are desirable, additional cameras can be useful. In Gaver, et al.'s study [13], for example, participants could select between multiple views of a remote location. Similarly, Fussell, et al. [12] allowed participants to choose between a wide-shot of the workspace, and a camera mounted on the head that provided detailed views of whatever they were looking at.

Others have used an omnidirectional camera to provide many views via a single camera [32]. This system uses microphone array based tracking technology to identify the current speaker, and then extracts only the relevant portion of the 360-degree view.

An alternative approach is to use a hybrid of manual and automatic control. The FLYSPEC system [22], for example, combines both a panoramic and pan-tilt-zoom cameras to allow for both automated and manual control. Others have experimented with allowing meeting participants to attract camera focus via pre-defined gestures [16]. Here the framing of shots is automatic, but their selection is not. This has the advantage of not requiring the system to determine appropriate shots, but depends on active participant control, which they may not be willing or able to do.

These hybrid solutions highlight a key challenge in camera control: determining what should be shown. Most systems accomplish this via some combination of object or motion tracking, and algorithms to allow for framing and shot selection.

2.2 Tracking Technologies

Active camera control depends critically on information about what is going on in the scene. Such information is typically obtained via audio and/or visual tracking.

Visual tracking uses sensing technologies to maintain a dynamic record of the location of specific objects. Sensing systems may be active (i.e., information is transmitted from objects to a receiver) or passive (i.e., objects are "noticed" by a camera or other sensor using vision systems) in nature [14, 24, 36]. In the systems developed by Ranjan et al. [30, 31] detailed tracking was achieved via high-resolution motion capture using infrared cameras and passive reflective markers. This provides very detailed tracking, but reflective markers had to be attached to all objects (including meeting participants).

Others have used sound-based tracking, in which microphone arrays [5] are used to isolate the location of sounds in the physical environment, and a camera can then be aimed at that region [23].

Regardless of the type of tracking, however, tracking involves inherently imperfect techniques [6, 24]. Basing camera control exclusively on tracking technologies (i.e., moving a camera every time tracking information changes) can result in erroneous camera movements that are distracting and potentially misleading [4].

One way to avoid this problem is to use tracking information in combination with heuristics to determine when a camera shot change should take place [30]. In this way, tracking information can be used more judiciously – it is assumed to be imperfect and some "intelligence" goes into determining when a shot change should take place. The key question then becomes one of isolating a set of heuristics that work in different scenarios.

2.3 TV Production Principles

One potential source of heuristics to guide camera control systems is television production. Others have looked at the frequency and rate of shot changes in professionally produced programs to improve the timing and rhythm of conferencing video [23]. Heuristics regarding shot framing, such as allowing for "head" and "nose" room have driven camera control systems [23, 31], and the layout of television studios have inspired the structure of some meeting capture systems [18, 31, 33]. Pinhanez et al., also also developed a theoretical framework for incorporating program scripts with these heuristics to automatically capture videos and applied it to capture a cooking show [27].

We focus here on heuristics used by directors to instruct camera operators and cut between shots during live broadcasts. Like an effective automated videoconferencing system, directors of live television work in a constantly-changing environment, deal with inherently imperfect camera shots, and do not have the luxury of post-production/editing to fix mistakes [21]. They constantly make do with what they have, do their best to anticipate the next needed, and avoid the appearance of errors in the live feed [9].

We suggest that these are also useful heuristics for a videoconferencing system. Such systems must rely on imperfect tracking technologies, control and select between cameras to deliver the best possible video images, and also avoid the appearance of errors. In particular, our system applies the following ideas from live TV directing, which we will describe in greater detail below: 1) focusing on cutting between cameras and relying on camera operators to frame shots; 2) maintaining consistent left/right orientation via the 180 degree axis, 3) anticipating and preparing the likely next shot, and 4) always having a backup shot ready in case the "right" shot is not ready.

3 The Present System

In this section we present our design goals and a description of the system we developed

3.1 Design Goals

We identified the following design goals for our system:

1. *Unobtrusive*. The system should not require meeting participants to wear sensors or be tethered. This will make the system more readily usable for informal meetings that might often benefit the most from effective archiving.

2. *Robust*. Most current unobtrusive tracking sensors provide noisy tracking data. The system should be able to handle this by making provisions for graceful degradation and recovery. A robust capture system should not fail when tracking provides erroneous data.

3. *Low overhead*. The setup cost of the capture system should be low, both in terms of time and money. It should not require substantial human effort to set up and operate. Furthermore, the components should be cost effective.

4. *Reconfigurable*. Although we consider only small group meetings, multiple variations could be found even in small meetings. The architecture should allow for small variations in setup without substantially influencing the performance.

3.2 Cameras: Video and Visual Tracking

Cameras are at the heart of any video system. If the system is to be reconfigurable and have low setup overhead, camera selection and placement are nontrivial problems.

We were intrigued by the versatility of TV crews who use a relatively small number of cameras (3-4 in a typical studio setting) to provide a wide range of shots. We therefore turned to TV production professionals for ideas. We learned that each studio camera operator is assigned a camera and that several camera-operator units essentially operate independently of one another.

While they are under the supervision of a director, their framing decisions are made individually [9, 21]. We aimed to design our cameras in a similar fashion so that they can operate independently, thus enhancing system flexibility.

3.2.1 Camera Sets

We used cameras for two purposes: capturing video and tracking the location of participants' faces. While both of these require a view of the scene, the nature of these functions are importantly distinct. Tracking participant location requires a wide shot in which all participants are visible, and therefore trackable. In capturing video, on the other hand, a close-up shot of one person is often desirable. Because this close-up necessarily restricts the video view to a single participant, that shot could then not be used for face tracking of all participants. We therefore used separate cameras for these two purposes.

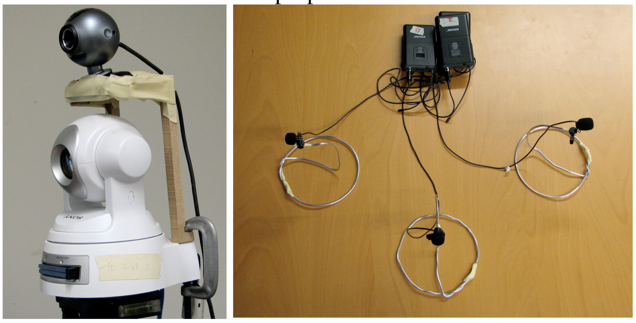

Figure 2. *Left*: Camera set with a webcam on top of a PTZ camera, *Right*: Microphone fan with three microphones

The cameras were physically attached to each other in pairs that we refer to as "camera sets." We used three camera sets in our system. Each consisted of a relatively inexpensive pan-tilt-zoom (PTZ) camera often used in off-the-shelf conferencing systems (SONY SNC-RZ30) and a basic webcam (Logitech Quickcam pro5000). The webcam was attached on top of the PTZ camera, and their views were calibrated with respect to one another (see Figure 2). Each PTZ camera was connected to the controlling computer through an Ethernet connection. Webcams were connected to the USB ports of the same computer.

Each camera set was controlled by a software module running on the controlling computer. This software module had two functions: (1) processing the webcam frames to detect faces and motion using vision techniques (see below), and (2) controlling the PTZ camera to frame shots of participants. In this way, the webcam and the vision based processing module acts like a camera operator that controls the PTZ camera assigned to it.

This setup allows any camera set to be placed anywhere in the room. Multiple camera sets can be placed in the room appropriately to cover the entire scene. In our setup, we placed the three camera sets so that there was a camera set facing each portion of the scene (see Figure 3), as in a TV studio. As long as a camera set is not broken apart after a one-time calibration, no additional calibration is required to set up the system for use or to move camera sets around the room.

3.2.2 Face Tracking

Tracking of faces was achieved via face detection using a modified version of the popular Viola-Jones algorithm [37]. To improve detection speed, we modified the OpenCV implementation of the algorithm [19] so that it searched for faces within a constrained visual range. This way the algorithm could detect multiple faces and be usable for real-time camera control.

To aid in participant identification, the system required a very brief initialization procedure. This involved looking at each camera for one second. This allowed the system to detect the initial position of their faces and assign an ID to each one. Once the system starts, face positions are updated every 2 seconds via a spatial proximity based approach that finds correspondence between faces in consecutive frames.

3.3 Audio: Who's Talking?

In addition to locating participants within the scene, we also needed to identify the current speaker.

3.3.1 Microphone Fan

We accomplished this using three Shure SLX wireless hyper-cardioid microphones arranged in a fan layout (see Figure 2). The number of microphones must equal the number of participants.

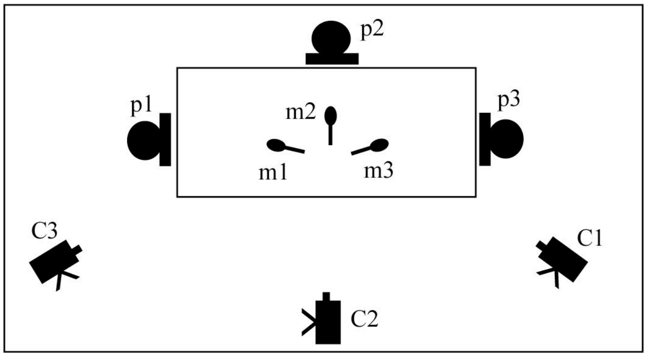

Figure 3. Room layout for the prototype system. There are three participants (p1, p2, p3), three camera sets (C1, C2, C3), and a fan with three microphones (m1, m2, m3)

Although microphone fans sometimes exhibit low directionality resolution [34], we used this design because: (1) it is reconfigurable and simple to set up; (2) audio tracking was used only for coarse-grained speaker detection (i.e., high resolution was not required); (3) it allowed us to easily detect multiple simultaneous speakers. Detection of speakers via microphones was based on signal intensity: first we detected single or multiple *active microphones* and then determined the person-IDs corresponding to those microphones.

Identifying the current speaker. To determine the current speaker, several steps were followed. As the nature of conversations does not always involve clear turn-taking with a single speaker [20], we did not want the speaker detection system to consider every utterance (including minor sounds and acknowledgements like "Um" and "Uh huh") as an occasion for a change of camera shot. We therefore used a temporal signal averaging filter to smooth out intensity generated by these short bursts. The microphone with the highest average intensity level was selected as the *active microphone* (i.e., a microphone with a corresponding active speaker).

Estimating multiple speakers. In an informal meeting scenario, participants often talk over each other or quickly take turns. When this happens, the output of the algorithm described above will keep switching from one speaker to another. However, we were interested in detecting all the speakers so that, for example, if several people were talking at the same time, the system would switch to a wide shot of everybody.

Our approach for detecting multiple speakers is to detect all microphones which have similar intensity levels and which are all above a certain level. In order to detect multiple speakers, we first detect the intensity of the primarily active microphone. Next, all microphones with intensity levels above a specified noise threshold are detected as active and corresponded to speakers.

3.4 Merging audio and video

Having identified microphones associated with active speakers, the next step was to reconcile the microphones with the face tracking system. To do so, the system assigns each microphone in the fan a unique microphone ID (e.g., m1, m2, and m3 in Figure 3). Since the number of microphones is the same as the number of speakers, there is necessarily a unique mapping from microphone ID to participant.

We are aided in this determination by the well known TV production principle referred to as the "180 degree axis" [2, 9, 38]. This principle is intended to ensure that spatial notions of "left" and "right" are consistent between multiple video images of the same space, so as not to confuse viewers. This is achieved by placing all cameras on the same side of an imaginary 180 degree line that can be drawn across the set. Interestingly, the goal of not confusing TV viewers also simplifies our tracking problem.

If the camera sets are placed according to this principle, each one will "see" participants in the same left-to-right order (i.e., C1, C2 and C3 see the participants in the order p1, p2, p3). The system assigns a unique number to each participant corresponding to his/her left-to-right position. Since the microphones in the fan are also ordered, the system can map each microphone to a participant (e.g., m1 to p1, m2 to p2, etc. in Figure 3).

This framework can be extended to other room and camera configurations, as long as they have an 'open side' – that is, the cameras are on the same side of the space relative to participants.

3.5 Robustness via Error Detection

Any system aiming to recover from errors must be able to detect them. While face tracking is good for unobtrusively identifying participant location, it is inherently imperfect. Facial positions changed, and people were sometimes difficult to spot due to variations in lighting, occlusion, and facial expressions. In Figure 4 we show two views of the same scene as captured by two camera sets (C1, C3 in Figure 3). One view has two faces detected (shown as blue rectangles), and the other has only one. The red rectangles show the last position where the face was detected. Below we describe how we detect the two most common types of vision tracking errors [24].

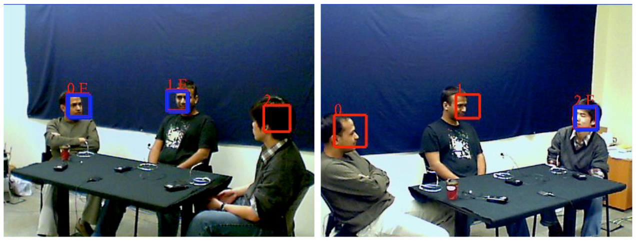

Figure 4. Same scene viewed by two cameras. Blue rectangle: face detected, Red rectangle: face not detected. Position of Red rectangle is the position where a face was last detected

False positive errors. These errors occur when the tracker detects a face but no actual face is present. A system directly following the tracking results without handling these errors would "think" it was showing participant faces, but actually show irrelevant objects in the meeting room, which could make the video confusing or disjoint. These errors can be identified by considering the confidence of the face tracking algorithm [17] in that low confidence scores indicate potential false positives. In addition, knowledge of the scene from other sources could be applied to flag these errors. Based on our knowledge of the scene, we derived the following heuristics to identify false positives.

1. *Overlap*. Since the cameras followed the 180 degree rule from TV production (see Figure 3), faces could not overlap when participants were sitting on their chairs. In cases where two faces were found to overlap, an error was assumed.
2. *Face size*. Plausible face sizes were determined based on the distance of the camera sets from the participants. Faces that were implausibly large or small were considered errors.
3. *Face location*. If the camera view is centered on a participant's face when she is seated, it was unlikely that the face could later be at the very bottom or top of the webcam (tracking camera) frame at any point. Faces in these regions were assumed to be errors.
4. *Face movement*. Face location information is not permitted to vary by a distance more than a predefined threshold in two consecutive frames. This threshold was defined assuming smooth, plausible participant movements. When participants did make a sudden movement, e.g. standing up from a sitting down position, this was treated separately as discussed later.

False negative errors. These errors occur when the tracker fails to detect a face where an actual face is present. These errors are, in some ways, more severe for our system because they can lead to a loss of valuable information. For example if a speaker's face is present and the detector fails to detect it, the system might not capture the speaker at all.

To identify false negatives, we first detected significant motion (e.g., person standing after sitting) via background subtraction on the video frames. Since large motions could potentially result in occlusion and face posture change, this step can provide preemptive warning to the camera control system that an error is likely, and allow for appropriate response (see Figure 5).

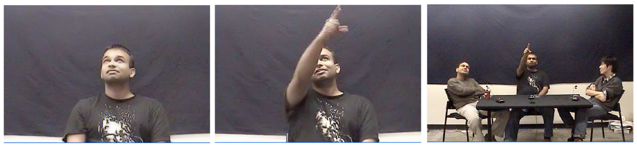

Figure 5. Shot transition sequence due to the detection of large movements in the scene: Close-up on the left, close-up with movement in the center, overview shot on the right

Possible false negatives were also identified when the number of faces detected was lower than the number of participants in the meeting (see Figure 4). In addition to knowing *that* a participant is missing, however, it is also helpful to know *who*. To accomplish this, the person-IDs for the faces detected in the current frame were determined by finding the person-IDs of the faces in the previous frame that are closest to those in the current frame. If a person-ID that could not be assigned in the current frame, that face was reported to be missing —and a potential false negative.

This strategy can be described formally. Let $p_{t,i}$ represent the face position vector of the ith person at time t. Let there be three participants in a meeting, and in a frame at time t the three face positions with the person-IDs assigned are $p_{t,1}, p_{t,2}, p_{t,3}$. Suppose at time $t+1$, the detector detects only two faces: f_1 and f_2.

The system assigns person-ID k to f_i if it satisfies the following condition: $distance\ (f_i, p_{t,k}) = min_{j\ in\ \{1,2,3\}}\{\ distance\ (f_i, p_{t,j})\ \}$

This procedure assumes that f_1 and f_2 are not false positives. Thus, if a person-ID p could not be assigned to any f_i then that person-ID face is declared to be missing. When the number of faces in the current frame and the last frame was equal to the number of participants, the tracking was assumed correct, and every f_i gets a person-ID assigned to it.

How our algorithm used information about a missing face and the person-ID of the face is described in detail in the next section.

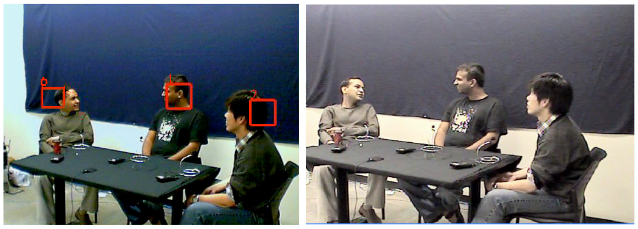

Figure 6. Left: Speaker's face (leftmost person) not detected in webcam frame. Right: Overview shot framed by PTZ camera opposite speaker

3.6 Controlling Cameras and Selecting Shots

Just as the director in a television control room relies on camera views and microphones to know what is taking place in the studio and decide how to best capture it, our system relies on information from the visual and audio tracking systems. From the audio-based speaker identification component, it gets the number of people talking and the person IDs of those who are talking.

From the visual face-tracking system, the software-based controller module for each camera set independently provides the person IDs of the people whose faces were detected accurately by that camera set, as well as a binary indicator of the presence of significant motion (large body movement, standing/walking). Based on these inputs, the algorithm determines what the next shot will be, selects a camera for framing that shot, and cuts to that shot. We describe each of these steps below.

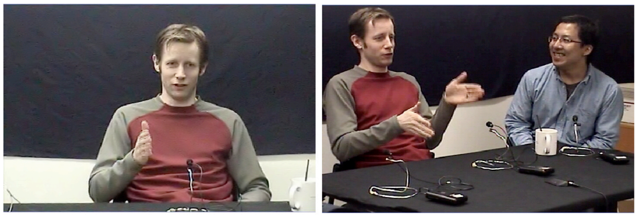

Figure 7. *Left*: A sample close-up shot, *Right:* A sample two-person shot

3.6.1 Determining Shot Type and Camera

There were three types of shots used in the system:

1. *Close-up shot*: This shot is used to show a close-up of the speaker or reaction of one of the participants (see Figure 7).
2. *Two person shot* (multiple person shot): This shot is used when multiple people are talking at the same time or quickly taking turns. In our prototype, there were three meeting participants, so this shot is a two person shot (see Figure 7). However, this can be extended to include more people.
3. *Overview shot*: This shot captures an overview of the entire setting, including the orientation and position of the participants, and other artifacts in the scene.

When the audio and video trackers do not report errors, then the system determines which shot to use based on simple principles:

- When a single speaker is detected, the next shot should be a close-up shot of the speaker
- When two speakers are detected, the next shot should be a two-person shot
- When more than two participants are talking, the next shot should show the overview.

When a possible error is reported by either tracking system, the system attempts to use a "safety net" shot that will not appear to be an error to viewers. Here we describe two possible scenarios involving erroneous tracking:

1. Occurs when the speaker detector correctly detects a single microphone as active and returns the corresponding person-ID, but the vision based detector fails to detect the face of the person. The system reacts to this problem by showing an overview shot using the camera covering the portion of the scene where the microphone is located. By using this shot, the system does capture the speaker, though the shot is not a close-up and therefore lacks detail (see Figure 6).
2. Occurs when there is a single speaker, but the speaker detector detects multiple active microphones, and the vision detector can track all faces. In this scenario, the system shows a multiple person shot including all the potential speakers identified by the tracker.

This provision ensures that the speaker is still captured in case of tracking errors. In Table 1 we summarize the different possible tracking result combinations and the corresponding shot selected.

Table 1. Possible detector outputs and resulting behavior

		Audio detector output	
		One source (without error)	*Multiple sources(with or without error)*
Vision detector output	*Face detected*	Close-up	Multiple person shot
	Face not detected	Overview from the opposite direction of the source	Overview

3.6.2 Managing Camera Sets for Shot Framing and Cuts

As in a TV studio, our system uses three camera sets to capture far more than three possible shots. As such managing camera sets for framing a new shot becomes a non-trivial task. Once the control algorithm determines the person or persons who need to be in the shot, it determines which is the appropriate camera set for the shot, based on three criteria:

1. The camera set should have already detected the face of the person to be framed.
2. The camera set should have the best possible view of the person to be framed.
3. The camera set should not be currently on-air.

The first requirement ensures that vision tracking errors are appropriately handled. If a camera is found that satisfies only the first two requirements, then the algorithm briefly cuts to another camera while the required camera frames the new shot. Only when the new shot is ready does the algorithm cut to that camera. If no camera set meets the first condition, the situation is handled as a vision tracker error (see previous sub-section).

An important aspect of the algorithm is to make sure that none of the camera sets is framing something irrelevant (e.g. empty space, or an empty chair). This occurs when a camera set frames a person and that person moves out of the frame, but vision tracking fails to track the person going out of the frame. In order to address this issue, our camera control algorithm examined all of the offline (i.e., not displayed) camera set views at regular intervals (once every 2 seconds). If a camera must frame a person who cannot be tracked by the vision tracker, that camera set is changed to a wide shot which can always be used as a "safety" shot.

4 System Evaluation

To evaluate the system we had several groups come to the lab to conduct mock meetings in which one participant was "remote." We then used log data, manual coding of videos, and questionnaires to assess system performance. We note at the outset that our goal in this evaluation is primarily to explore and validate the potential of the principal techniques we introduce, and not to demonstrate the superiority of our approach, per se.

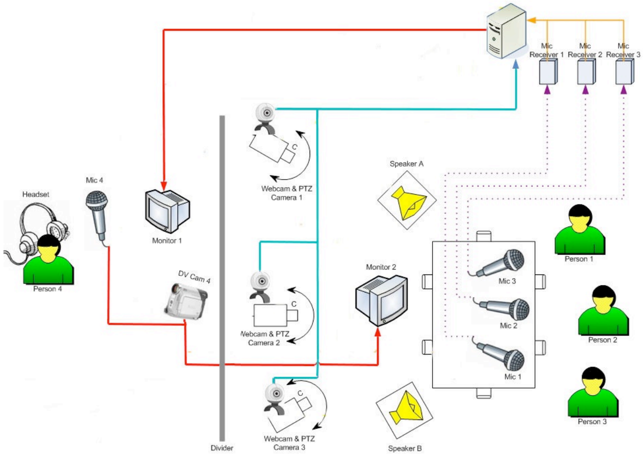

Figure 8. System setup diagram for the evaluation

4.1 Participants

Participants were recruited via flyers placed around the campus of a large university in North America. Six groups of four people used our system, for a total of 24 (8 male, 16 female). Their age ranged from 19 to 26 (*M* = 21.8, *SD*= 2.8), and 18 were currently enrolled students. Each received $10 for their participation.

4.2 Procedure

Participants were randomly assigned to be "local" or "remote" participants. Three participants were local, and one was remote. As shown in Figure 8, local participants sat at the conference table using the setup described earlier, with the addition of a 26" LCD monitor on which the remote participant was displayed. The monitor was placed between cameras 1 and 2 for easy visibility and relatively natural gaze patterns with the local participants. Remote audio was conveyed via a speaker near the monitor.

The remote participant used a simulated desktop conferencing system. By "simulated" we mean that the audio/video were not transferred over a network, but rather via local cables to an adjacent room. This was done to ensure that network delays and resulting deterioration in video quality were not confounds. The remote participant sat in front of a 26" LCD display that showed video from our camera control system. Behind the screen was a video camera and a microphone, which were used to capture video and audio for the local participants.

Once in place, they completed a pre-experiment questionnaire and then carried out two "meetings" in which they had to reach consensus on the rank ordering of a set of items. In the first, a practice task, they were instructed to rank order a series of five fruits (i.e., pineapple, mango, apple, banana, etc.). Once this task, intended to familiarize participants with the conferencing system, was completed, they moved on to complete either the Arctic Survival task [10] or the NASA Lost on the Moon task, both of which are standard tools designed to elicit conversation.

In these tasks, participants are given a written scenario indicating that they are stranded either in the Arctic or on the Moon, and have a limited number of items that they can carry with them. They are told to decide which are the most important items to take, by rank ordering them. Each person ranks the items individually, and then the group meets to determine the collective rankings. They had 20 minutes to carry out this ranking task, and then completed a post-experiment questionnaire. We used these scenarios for consistency across the several mock meetings and to make it likely that all group members would participate.

4.3 Results

We present results from analysis of system logs, human coding of videos, and participant questionnaire responses.

4.3.1 Log Analyses

To analyze the performance of the system, we looked at the duration and frequency of the camera shots. On average, each video clip was 16 minutes long (SD=4), with a mean of 9 shot changes per minute (SD=1). Of these shots, 38.6% were close-ups (SD=8.1%), 7.5% were two-person shots (SD=3.8%) and 53.9% were wide-shots (SD=10.3%).

4.3.2 Human Coding of Videos

While analyses of log data can tell us whether the system was internally consistent, this does not tell us if the system actually resulted in videos that provide appropriate information at appropriate times, as judged by human viewers. We therefore had two independent coders view each of the videos to assess the quality of shot framing and shot cuts.

Coders were instructed to assess each shot in terms of whether it was appropriate or not (i.e., whether it showed something relevant), and whether it seemed correctly framed or not. The basic heuristic used in assessing both of these criteria was whether or not the coder could reasonably wonder "Why am I seeing this?" or "Why is that framed that way?".

On average, each coder rated 88% of the shots in each video as appropriate. They both agreed that 82.3% (SD = 8.3%) of the shots were appropriate, and that 6.2% (SD=4.2) were inappropriate. When considering only overview shots, 99% (SD = 1.9%) were considered appropriate by both coders. And when considering only close-up shots, the number drops to 63.5% (SD = 18.4%). This suggests that the overview shot was a good "safety" shot, but that the logic of selecting close-up shots could be improved. As for framing, both coders agreed that shots were framed correctly 73.1% of the time, on average (SD = 16.5%).

We analyzed the instances (M=9, SD=7) when both coders agreed that the shot was not appropriate. To better understand why these shots were rated this way, we checked to see if the system had correctly detected the speaker for those shots. Upon comparing the system log (which reflected the system-identified speaker) with the manual video coding (which identified the actual speaker), we observed that on average 61.6% (SD=35.6) of not appropriate shots were when the system had misidentified the current speaker. Of those, 91.2% were close-ups and the rest were two person shots. This can be attributed to inaccuracies of the microphone fan in detecting the right speaker.

We also looked at the shots when the system did identify the correct speaker, but still failed to provide what coders would agree was an appropriate shot (17 shots in total across all the sessions). Of these, 93% were close-up shots of a person not speaking and the rest were two-person shots. Here, the system did not capture the correct person because the camera needed was already in use. The system therefore switched to another available close-up shot to show the reaction of another participant, and free up the camera. While this is a common technique in TV production, it is reasonable that human coders might disagree on whether or not the shot choice was appropriate, given that it was not of the person speaking.

4.3.3 Questionnaires: User Impressions

The post-experiment questionnaire was based on the one used by Ranjan et al. [30], and involved 20 items from 5 constructs: group efficacy (4 items, Cronbach's α = .80), individual efficacy (5 items; α = .89), video utility (6 items; α = .88), video predictability (3 items; α = .83) and frustration (2 items; α = .65). Note that α is a measure of scale reliability, for which values of .79 or higher are considered adequate for social science research. All values presented below reflect the mean of the individual scale items. Items all used 7-point Likert scales anchored by Strongly Agree (7) and Strongly Disagree (1), with a neutral midpoint (4).

Because only the remote participant in each group saw the system output, we present only their responses here. Figure 9 summarizes the questionnaire results. Responses between 1-3 were aggregated into a 'Disagree' category, 4 was 'Neutral,' and 5-7 were aggregated into an 'Agree' category.

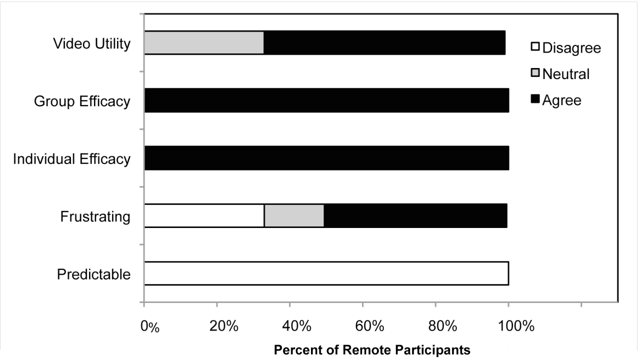

Figure 9. Questionnaire responses for remote participants (n=6).

As can be seen in the figure, the results are mixed. All participants felt they were able to perform their tasks effectively both individually and as a group, as indicated by overall agreement with individual and group efficacy items. Participants also tended to agree with items suggesting that the video views provided were generally useful. Four participants agreed with these items, and two were neutral. None disagreed.

Participants were also asked whether they sometimes found the system frustrating. Two disagreed, and one was neutral, meaning that half either did not find it frustrating or were neutral. The other three did find the system frustrating. Participants were asked how predictable the system was in terms of whether or not they could anticipate what the next shot would be. No participants found the system to be predictable. This is understandable, however, as participants only used the system for a short time. It is also possible that unpredictability referred to the variety of shots, rather than a drawback of the system.

5 Discussion

We began with the goal of a system that could robustly and automatically create more dynamic videos of meetings using low-cost technologies that are minimally obtrusive. We developed a system that satisfied these constraints through the use of automatically controlled pan-tilt-zoom cameras in combination with face-tracking via webcam and audio sensing via a microphone fan. Principles from television studio directing are applied to improve the robustness of the system. Our evaluation suggests significant progress toward these goals, but that challenges remain. We highlight here our key contributions.

First, we introduced "camera sets" as a novel and extensible method of automating camera control. By pairing two cameras, a PTZ camera for video and a webcam for face tracking, our system decentralized and separated the functions of shot framing and shot selection, as is done in live television production with camera operators and a director. By allowing each camera set to frame shots and report tracking data independently, cameras can be easily moved, added, or subtracted as situations demand. The shot selection system then takes input from whatever camera sets are available and selects the best shot from among these.

Our preliminary evaluation results suggest that this is a promising approach. The system did automatically change shots several times per minute, and framing of shots was agreed by both human coders to be effective a substantial fraction of the time.

Camera sets can be added as long as their placement satisfies the TV-production principle of a 180-degree line, i.e., all the cameras should see all the participants in the same order from left to right. Since each camera set operates and reports tracking and error information independently to the camera control algorithm, inclusion of a new camera in the existing setup does not require any change in the previous camera sets.

Second, we developed a system that is robust in the face of imperfect tracking technology. The system identifies likely errors and uses a "safety" shot to cover these potential errors. This aspect of the system was successful in that it did cut to "safety" shots when the tracking system reported errors. For example, in the event of incorrect speaker detection (M=62, SD=29), it managed to provide an appropriate shot most of the times (M=74.9%, SD=10.9%). It was less successful, however, in that the choice of shots was sometimes confusing or frustrating to participants and our coders, particularly when the safety shot was a close-up of a person not talking. Additional work is needed to improve the system's robustness not only in identifying errors, but recovering with an appropriate shot more quickly.

5.1 Limitations and Future work

While our results present promising preliminary evidence in support of our novel approach, we acknowledge several limitations and suggest areas for future research in this area.

First, the system used the wide shot a significant fraction of the time. This could be the result of several factors. One is that there may have been multiple speakers or frequent, rapid changes in who was speaking. These would be appropriate times to use the wide-shot, as close-ups would be confusing. Some of the time, however, it is likely that frequent wide shot usage results from errors in tracking or speaker identification, such that the system did not "know" who to get a shot of, and used the wide shot as a "safety shot." While this is a good response given an error, future research could improve speaker identification or tracking algorithms to minimize these problems.

Second, close-up shots were rated appropriate by our coders the majority of the time, but not all of the time. This suggests that we should also attempt to improve camera management strategy.

5.1.1 Applying the Framework to Other Scenarios

Although our prototype system consists of three camera sets and can capture three participants or less, the algorithms and the system framework can be extended.

More people. Our framework requires as many microphones as the number of meeting participants. The framing strategy and the camera control algorithm will automatically include multiple person shots (e.g., two-person and three-person shots if there are four participants) based on inputs from the tracking components.

Different room layouts. Our framework makes one important assumption about the way participants are located in the room: they are all sitting around a desk with one edge of the desk open. Various common meeting room layouts follow this constraint [18]. While Rui, et al. [33] asked videographers how they would arrange cameras for different types of lecture room scenarios, we aim to incorporate part of the knowledge of professionals in our framework itself. This general framework can then readily be applied to different meeting room layouts.

5.1.2 Technical Limitations: Computational Cost

When we ran our system on an Intel Pentium 4 processor (3.00 GHz) computer with 2GB of RAM, CPU usage was approximately 90%. Vision processing was the most expensive part of the computation. Most vision-based tracking algorithms are computationally expensive for real-time applications [24], and this is a bottleneck for our system. We use a modified version of the Viola-Jones face tracker and dynamic background subtraction to detect faces and large motion. Despite our modifications to improve speed, extending the system to include more cameras and participants could increase system response time.

REFERENCES

[1] Meetings in America: Meeting of the Minds. http://e-meetings.verizonbusiness.com/meetingsinamerica/pdf/MIA5.pdf, 2003.

[2] Arijon, D. *Grammar of the Film Language*. Hastings House, New York, 1976.

[3] Bianchi, M. H. AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations. In *Proc. Joint DARPA/NIST Smart Spaces Technology Workshop*. 1998.

[4] Birnholtz, J. P., Ranjan, A. and Balakrishnan, R. Error and Coupling: Extending Common Ground to Improve the Provision of Visual Information for Collaborative Tasks. Paper presented at the Conference of the International Communication Association. Montreal, Canada, 2008.

[5] Brandstein, M. and Ward, D. Microphone Arrays: *Signal Processing Techniques and Applications*. Springer Verlag, 2001.

[6] Compernolle, D. V. Future Direction in Microphone Array Processing. In M. S. Brandstein and D. Ward, ed. *Microphone Arrays*. 389--394. Springer, 2001.

[7] Cummings, J. and Kiesler, S. Coordination and success in multidisciplinary scientific collaborations. Paper presented at the Int'l Conf. on Info. Systems. 2003.

[8] Desanctis, G. and Monge, P. Communication processes for virtual organizations. *Journal of Comp.-Mediated Communication*, 3, 4 (1998).

[9] Donald, R. and Spann, T. *Fundamentals of TV Production*. Blackwell Publication, Ames, IA, 2000.

[10] Eady, P. M. and Lafferty, J. C. *The subarctic survival situation*. Synergistics, Plymouth, MI, 1975.

[11] Egido, C. Videoconferencing as a Technology to Support Group Work: A Review of its Failure. In *Proc. ACM CSCW*. 13-24,1988.

[12] Fussell, S. R., Setlock, L. D. and Kraut, R. E. Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In *Proc. ACM CHI*. 513-520, 2003.

[13] Gaver, W. W. The affordances of media spaces for collaboration. In *Proc. ACM CSCW*. 17-24,1992.

[14] Gross, R., Yang, J. and Waibel, A. Face Recognition in a Meeting Room. In *Proc. IEEE Conf. on Automatic Face and Gesture Recognition*. 294, 2000.

[15] Hinds, P. and McGrath, C. Structures that work: social structure, work structure and coordination ease in geographically distributed teams. In *Proc. ACM CSCW*. 343-352, 2006.

[16] Howell, A. J. and Buxton, H. Visually Mediated Interaction Using Learnt Gestures and Camera Control. In *Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in HCI*. 272--284. Springer-Verlag, London, UK, 2002.

[17] Ilonen, J., Paalanen, P., Kamarainen, J.-K. and Kalviainen, H. Gaussian mixture pdf in one-class classification: computing and utilizing confidence value. In *Proc. Conf. on Pattern Recognition*. 577-580, 2006.

[18] Inoue, T., Okada, K. and Matsushita, Y. Learning from TV programs: Application of TV presentation to a videoconferencing system. In *Proc. ACM UIST*. 147-154,1995.

[19] Intel Learning-Based Computer Vision with Intel's Open Source Computer Vision Library. Compute-Intensive, *Highly Parallel Applications and Uses*, 9, 1 (2005).

[20] Isaacs, E. and Tang, J. What video can and cannot do for collaboration. *Multimedia Systems*, 2(1994), 63-73.

[21] Kuney, J. *Take One: Television Directors on Directing*. Praeger Publishers, New York, 1990.

[22] Liu, Q., Kimber, D., Foote, J., Wilcox, L. and Boreczky, J. FLYSPEC: A Multi-User Video Camera System with Hybrid Human and Automatic Control. In *Proc. ACM Multimedia*. 484-492,2002.

[23] Liu, Q., Rui, Y., Gupta, A. and Cadiz, J. J. Automating camera management for lecture room environments. In *Proc. CHI*. 442-449, 2001.

[24] Ming-Hsuan Yang, David J. Kriegman and Ahuja, N. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, 24, 2 (2002), 34-58.

[25] Nguyen, D. and Canny, J. Multiview: improving trust in group video conferencing through spatial faithfulness. In *Proc. CHI*. 1465-1474, 2007.

[26] Olson, G. M. and Olson, J. S. Distance matters. *Human-Computer Interaction*, 15(2001), 139-179.

[27] Pinhanez, C. S. and Bobick, A. F. Using computer vision to control cameras. In *Proc. AJCAI Workshop on Entertainment and AI/ALife*. 69-76,1995.

[28] Poltrock, S. E. and Grudin, J. Videoconferencing: Recent Experiments and Reassessment. In *Proc. HICSS*. 104a, 2005.

[29] Polycom http://www.polycom.com/.

[30] Ranjan, A., Birnholtz, J. and Balakrishnan, R. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. In *Proc. ACM CHI*. 1177-1186,2007.

[31] Ranjan, A., Birnholtz, J. and Balakrishnan, R. Improving Meeting Capture by Applying Television Production Principles with Audio and Motion Detection. In *Proc. ACM CHI*, 227-236, 2008.

[32] Rui, Y., Gupta, A. and Cadiz, J. J. Viewing meeting captured by an omni-directional camera. In *Proc. ACM CHI*. 450-457,2001.

[33] Rui, Y., Gupta, A. and Grudin, J. Videography for telepresentations. In *Proc. ACM CHI*. 457-464, 2003.

[34] Rui, Y., He, L., Gupta, A. and Liu, Q. Building an intelligent camera management system. In *Proc. ACM Multimedia*. 2-11,2001.

[35] Vertegaal, R. The GAZE groupware system: mediating joint attention in multiparty communication and collaboration. In *Proc. ACM CHI*. 294-301,1999.

[36] Vicon http://www.vicon.com/.

[37] Viola, P. and Jones, M. J. Robust Real-Time Face Detection. *Int. J. Comput. Vision*, 57, 2 (2004), 137--154.

[38] Zettl, H. *Television Production Handbook*. Wadsworth Publishing, Belmont, CA, 2005.