Smart Cameras for Smart Conference Rooms

Zhigang Chen   Xin Hu   Scott Gifford

July 29, 2006

1 Introduction

Today's Internet age is a global, fast-moving, information-rich time. Your colleagues are as likely to be three timezones away as three offices. If something exciting is happening in the world, you don't have time to schedule a meeting to discuss it; by the time the meeting happens, the world will be on to the next exciting thing. With things moving so quickly, it's impossible to keep track of everything, so tools are necessary to make this information accessible.

Many parts of a typical office or University have adapted to these rapid changes. One area that's still behind is the conference room. Rooms are typically scheduled in advance, and unscheduled meetings require searching for an empty room and hoping nobody's meeting starts soon. Conference rooms can't easily link up with rooms in other buildings, let alone other countries. After the meeting, the only record is the notes that somebody may have taken, and they generally aren't electronically searchable.

To address these issues, we are working on a "smart conference room appliance," an easy-to-use networked device which transforms an everyday conference room into a self-aware collaboration space. This space should have access to the video and audio of what's going on in the room, and be able to share or record it according to the wishes of the meeting participants. It should be able to index the content of the meeting, for easy searching and filtering, and should know who was at a meeting, to allow filtering based on that information. It should be able to coordinate with other conference rooms to figure out optimal room schedules and dynamically detect which rooms are in use, so an empty room can be quickly found for an impromptu meeting. All of this should happen with minimal configuration required by the meeting participants, while respecting their privacy wishes.

As a first step towards this vision, we have implemented a "smart camera" which uses sensors placed in a conference room to detect the person speaking, find the best camera to get video of the speaker, automatically points the camera at that person, then streams the video over the Internet. This allows any sort of meeting to proceed naturally, with the system automatically adjusting to the participants use of the space. For example, in a typical presentation, the camera would be focused on the presenter through most of the talk, then would start switching between audience members' questions and the speaker's responses during the question and answer part of the presentation; for a brainstorming meeting, the camera would move between the various participants as they talked about their ideas.

2 System

Windows CE is an ideal platform for this system. It is designed for building computerized "appliances" which require straightforward configuration and little or no maintenance. It has built-in support for a wide variety of hardware and applications, and can run on inexpensive and power-efficient hardware, which be deployed throughout a building at a minimal cost.

We have integrated our Windows CE system with Berkeley Motes, tiny devices which can sense information about their environment and wirelessly relay the data to a PC.

2.1 Hardware

To prototype our system, we built a device from an old PC we had lying around. The PC already had USB support, and we installed an Ethernet card to allow us to connect to our network. We connected two LogiTech QuickCam Orbit MP USB cameras to the device, to provide video input. These devices are 1.3 megapixel webcams with a robotic camera head, which can swivel 189 degrees horizontally and 102 degrees vertically under electronic control.

Because of the ease with which Windows CE can be ported to new platforms, we could easily replace the PC with a smaller, quieter, and less expensive system with minimal effort. Our only requirement is USB and network support.

2.2 Motes

"Motes" are a specific type of environmental sensor developed at the University of California at Berkeley. Different modules can be connected to the motes to sense a wide variety of things about their environment, then act collectively to process that information while moving it over large areas. They are powered by simple AA batteries, and can run for weeks or months without maintenance.

We use these motes to detect sound information inside a conference room. By placing several motes throughout a conference room, we can detect which one senses the loudest sound source, and from that infer the location of the person speaking.

Currently the motes relay their information to a standard Windows PC, which sends the information across the Ethernet to the CEPC. Long-term, it would be useful to port the mote management code to the Windows CE platform, so the CEPC could interact directly with the motes.

2.3 Software

We built the basic system with the Windows CE Platform Builder. We used a basic platform setup, and added in the Web camera drivers. To allow network access to the video, we also installed the shared-source CamServ service. We made numerous changes to both the application and the driver, which we will describe below.

2.4 Client

To display video streamed over the network, we used the ViewRCam client included in the shared-source CamServ application.

3 Development

To implement our system, we made numerous changes to both the USB WebCam driver, the CamServ server, and the ViewRCam client.

4 Extension to Drivers

1. Add camera movement control to the driver allowing the application to rest and move the camera to a certain angles. We allowing the input to be absolute or relative angles.

2. Extend the Cameracode program and make it multi-thread which significantly

5 Setup

The CEPC machine is put in the middle of the room and two cameras are connected to the CEPC back to back, each of which captures half of the conference room and stream the video to the CEPC. Multiple sensors are distributed in the room to collect the sound information (as the microphones shown in the figure) and send that information back to the mote gateway which is also connected to CEPC. Mote gate way integrate all the data, calibrate the readings and compare them to find the position of sound source. That position information which is translated to the angle of Pan and tilt of the corresponding camera, is sent back to CEPC and move the camera to point to the target providing our system the ability of tracking the active sound source or speakers in the conference room.

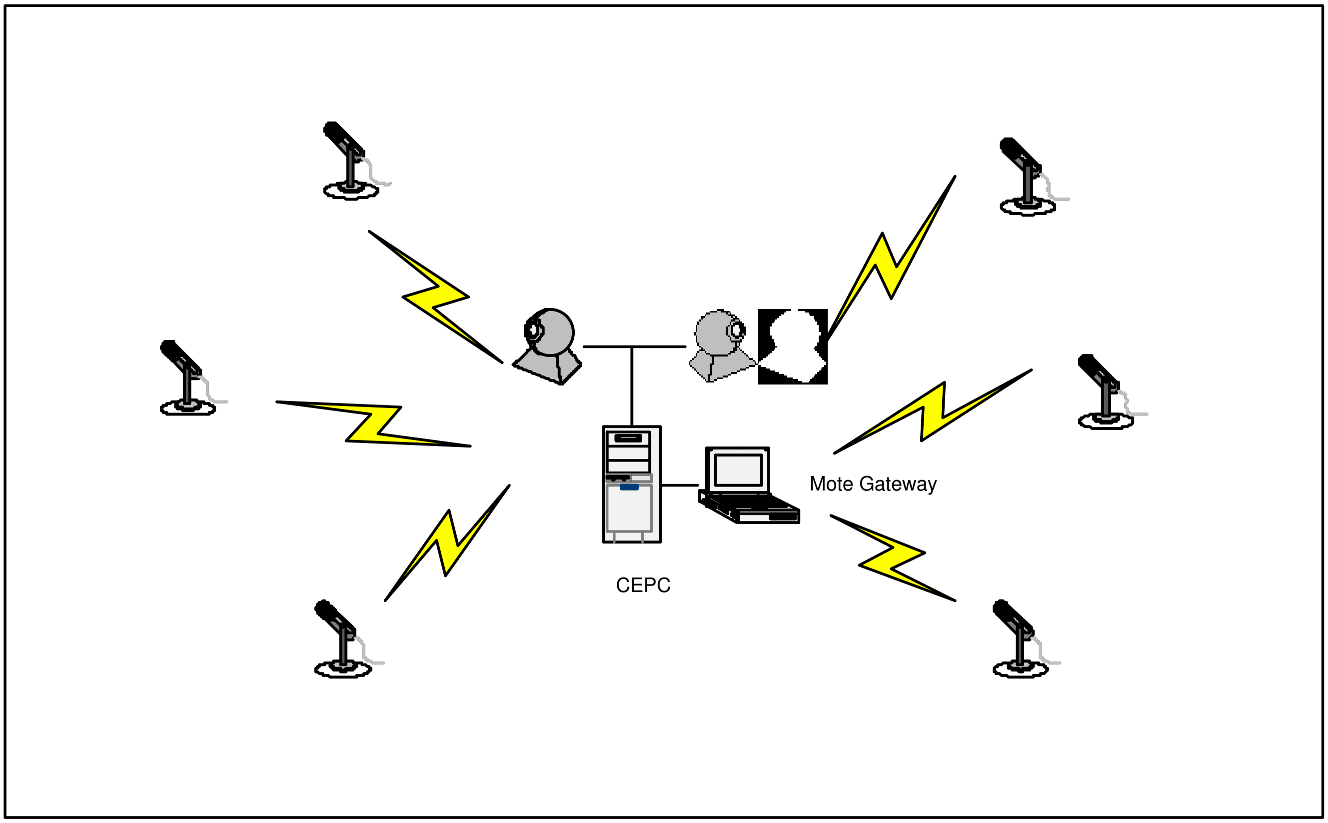

Figure 1: *Basic Setup*

5.1 Client

We extended the ViewRCam client by making it multi-threaded, which we found made the video run more smoothly.

6 Setup

The setup for the system is straightforward. Currently it requires some technical knowledge; we would like to extend the setup to make it easier.

6.1 Hardware

We placed the CEPC, the mote gateway, and the cameras in the middle of the room. We connected two cameras to the CEPC, and set them back-to-back. With this setup, the cameras together have a view of the entire room, and we can pan to a speaker no matter where they are.

We then set up the motes at locations throughout the room. We chose 6 locations around the table, each encompassing about 1/6 of the room. This gave rough coverage of the whole room. We could improve the system's accuracy by place additional motes throughout the room, but we found that this number of motes was adequate for our demo.

To accurately sense sound with the motes, they require some calibration to avoid small measuring variations between the sensors. We measure a sound with a fixed frequency from each mote, record the amplitude of the soundwave detected by each one, and then configure the gateway to normalize the output of all the motes.

We also use a standard PC as a networked client to CamServ.

We link all of the computers together over standard Ethernet.

6.2 Software

With the hardware in place, we have a view of the whole room and can sense the location of a person speaking in the room. What's left is to correlate positions in the room with camera angles.

We have extended the CamTest application to allow the camera to be panned, tilted, and zoomed while displaying the video. We use this feature to find the best camera angle for each of the locations.

For each mote, we have a person speak near the mote, then use CamTest to select the appropriate camera and adjust it camera so that it captures that person effectively. We then ensure that other positions within sensing range of that mote are also accurately captured by the cameras. Once we have a camera angle we are satisfied with, we record the pan and tilt as an absolute angle, and configure the software to link the appropriate mote to these angles.

When all motes have been configured, we set up the PC running ViewRCam to connect to the CEPC, and ensure that the video is streamed correctly. We then test the system by walking around the room talking in a normal voice. If the system is working correctly, ViewRCam will always display the speaker, switching between cameras and changing angles appropriately.

7 Conclusion

In this project, we explore the capability of a few embedded devices, and more importantly, how they can collaborate to make useful and cool applications. Nowadays, we are surrounded by many embedded devices, which have become crucial part of our life. For example, cell phones, PDAs, game boxes, and various sensors, etc. And more are coming out in the next decade. Such versatile embedded devices have very broad range of capability and may run different platforms. It is very challenging how to make them work together and accomplish some task as a whole. Our project is an attempt to look into this challenge and illustrate the potential how various devices can be put together and what such combinations can achieve. By putting together one CEPC, two web cameras, six sensor nodes, and a gateway PC for the network of sensors, we make a smart conference room system which automatically track the location of speakers and point the web cameras accordingly to the speaker. We extend the web camera driver code ,which allows the LogiTech Orbit web camera to move elegantly, and it also supports multiple cameras simultaneously. Additionally, we modify the server code to separate grabbing video out of the driver from rendering it via networks. This significantly improves the performance and robustness of the server. Finally, we integrate a sensor network with the CEPC and web cameras, so the sensors can compute the location of sound sources and feed the information to the CEPC, which selects the right camera, move it accordingly to point to the speaker, and render the video from the camera to the remote client.