

# SMART VIDEOCONFERENCING

*Dmitry Zotkin, Ramani Duraiswami, Vasanth Philomin, Larry S. Davis*

Institute for Advanced Computer Studies  
University of Maryland, College Park  
{dz,ramani,vasi,lsd}@umiacs.umd.edu

## Abstract

The combination of acoustical and video processing to achieve a smart audio and video feed from a set of  $N$  microphones and  $M$  cameras is a task that might conventionally be accomplished by camerapersons and control room staff. However, in the context of videoconferencing this process needs to be performed by control software. We discuss the use of a multi-camera multi-microphone set up for unattended videoconferencing, and present details of a prototype implementation being developed.

## 1. INTRODUCTION

As videoconferencing based meetings become more commonly used as replacements for in-person interactions, and for multi-user remote collaborative projects, the quality of interaction provided by conventional systems becomes a barrier to use. One problem is due to the lack of bandwidth or protocols that do not guarantee quality of service, leading to poor quality audio and video [6]. However, a more fundamental problem is that the basic system (a camera and one or two microphones) will not capture all the information being presented by a conferee and transfer it to the other parties, thereby making the videoconferencing experience a poor substitute for in-person interaction. Information that is missed could be due to the audio output not encoding sufficient information about the spatial distribution of sound, or not suppressing noise, to more subtle issues like the images not focussing on the speaker's gestures, or the speaker not being in the frame of the picture. A smart videoconferencing system could alleviate these drawbacks.

A first extension to the basic videoconferencing set-up is one that provides better audio via beamforming, and in addition locates the speaker and uses it to point the camera. A simple system of this nature is already available as a commercial product [7]. A next version of the system might employ both audio and video based location of people; work in this area include [11],[12] and [13]; we developed our own prototype multi-sensor tracking system, which was described in [3]. Several papers in the HCI community consider the dynamics of the interaction in video (e.g., [1, 5]). These papers all point to the fact that for enriching the conferencing experience different kinds of information need to be transmitted.

In designing a multiple sensor intelligent videoconferencing system, we face the problem that the tasks of conveying information to conferees and to the operator (or, in automated conferencing case, the controlling computer) can be conflicting. The sensors

should be used in a way that the controlling software is provided enough information to allow it to completely replace the operator in the video and audio control room; in addition, the remote site should receive the picture that is judged to be the best available for the current scene circumstances, i.e. simulating the switch and zoom operations a skilled operator would provide. These tasks are often contradictory and thus can't be performed using just a single sensor. For example, there might be a situation when the software wants to keep track of all persons in the room, but the camera is zooming on the speaker to clearly identify her for the remote site. In addition, there is a need for multiple sensors both for scene understanding (e.g. depth recovery) and for the enhancement of remote site viewing experience (e.g. switching between views without associated motion blur and time delays).

To address these problems, we propose a multi-sensor videoconferencing system which we call the "smart videoconferencing system". It involves the unobtrusive replacement of operator(s) by unattended computer controlled systems that take inputs from  $M$  cameras and  $N$  microphones and distil them in an information maximizing way (with respect to remote viewers) into output in one of the standard protocols (e.g., H.323). We describe general considerations in the development of such a system, describe component algorithms for audio and video processing, and discuss a specific prototype system implemented in our laboratory.

## 2. SMART VIDEOCONFERENCING

Ideally, the system should possess sufficient intelligence to perform as well as a human operator, who has knowledge of the current conference situation and uses it to provide "best" output. The controlling software should attempt to utilize the available sensors similarly (e.g. to get the picture for transmission from one camera with suitable zoom factor, and to use other available cameras to maintain awareness of the changing environment). The task of being aware of or learning about, the environment could include keeping track of all the participants, guessing who might be the next possible speaker(s), learning the 3D positions of the speakers from audio and video information, and so on. To maintain awareness, the control software could have a model of the environment that is periodically updated. The update is scheduled with the planner; some parameters need be updated only infrequently (e.g., number of participants and their video and audio identities) and some require faster updates (e.g. speaker positions and identity of the current speaker). Thus, the general update cycle is posed as a set of tasks, each of them having a specific priority and a deadline. The planner schedules these tasks on the physically available resources according to priorities and deadlines. The goal is to

---

Support from the W.M. Keck Foundation, ATR MIC Labs, and ONR via Contract N00014951021 is gratefully acknowledged.

have all tasks completed by their appropriate deadlines. Since the available resources are usually scarce, some resources might be assigned to different tasks at different times. These assignments are done by a resource manager, which is aware of physical equipment constraints (e.g., cameras have a finite pan-tilt-zoom mechanism response time) and in addition have knowledge of the suitability of particular sensors to particular tasks. Examples of such knowledge include: i) the best camera to get the view of the room as a whole is probably the one that is placed as far from speakers as possible or has the widest zoom; ii) 3D-position might be estimated by microphone array data, iii) by one camera using approximate knowledge of the proportions of a human body, iv) or by two or more cameras using stereo information to extract the depth information from the scene.

A sample videoconferencing scenario might be as follows: Somebody walks into the room. The camera that has the widest field of view detects a change in the number of participants, and the room model module makes a request to build the identity of the person to recognize him later. As soon as a camera is available, it focuses on the person and provides the necessary information. Alternately, the speaker displays an exhibit and this event can be detected and a camera pointed and zoomed at the item to show it to the remote participants.

### 3. PROTOTYPE SYSTEM SETUP

To explore these issues we implemented a prototype system with two seven-element microphone arrays and two pan-tilt-zoom video cameras with video output to both a television and computer capture cards. The television output is controlled via a computer-controlled video multiplexor. Even with this limited configuration, it is possible to enhance the videoconferencing environment significantly.

In a two-camera system, the following sample camera management schemes can be used. One camera can focus and zoom on the speaker and the other can get an overview picture of the room and track the movement of people. If a speaker moves and gets out of frame of the zoomed camera, the picture transmitted is switched to the zoomed-out camera for a few seconds (avoiding unsettling motion blur that is transmitted poorly by current compression schemes) and then back to a close-up view when the first camera acquires the subject again. Another possibility is to get the transmitted picture instantly switched when different people speak. In this scenario, the second camera is focused and zoomed on the new speaker and only then is the output view switched to it. When no speech signal is detected for a long time, the view may be switched to a zoomed-out one. Yet another possibility is to use two cameras as a stereo pair to obtain the 3D positions of participants when sound is noisy or unavailable.

The two microphone arrays can also serve in different ways; one can localize the primary speaker while the other scans the space for other sound sources. Alternatively, the microphones can perform beamforming for multiple sources using positions supplied by video or audio for targeting, or reduce the room noise and reverberation, or attenuate the received remote site audio signal played through a local loudspeaker.

The main software module maintains limited awareness of the environment by communicating with the audio and video processing subsystems. The planner module accepts requests for the resources from the main module, dispatches them and reports the fulfillment of the request to the main module. The “control room” module decides what picture should be transmitted to the remote

site, sends requests to the main module to get that picture on one of the cameras and controls the hardware video multiplexor. This module is controlled by simple heuristic algorithms (e.g. when a different speaker begins to talk, the second camera zooms at him and then the view is switched to this camera; when no speech is detected for a long time, the view switches to the zoomed-out camera; and if an apparent conversation between two people takes place, each camera keeps one speaker in view).

There are also lower-level modules which communicate directly with the available hardware sensors; they are described in detail below. The audio detection and localization module uses algorithms based on the signal power and time difference of sound arrivals (TDOAs) to perform signal detection and localization and can provide the remote site with the beamformed signal for the desired location. The video localization and tracking module uses algorithms based on subtraction of successive frames to detect, label and track moving objects within the camera field of regard.

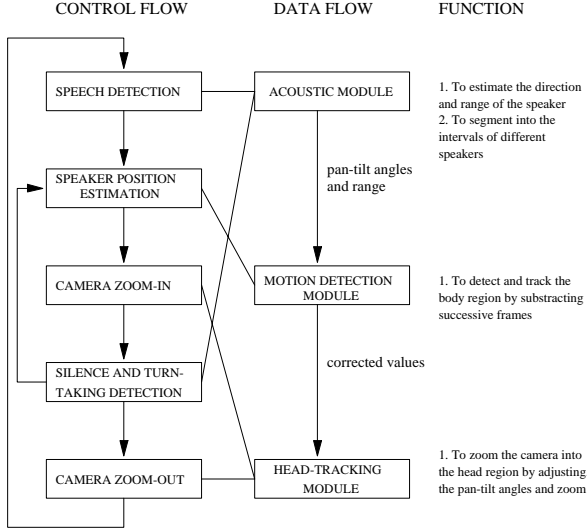
The audio detection and localization module is built around the PowerDAQ data acquisition board, capable of acquiring up to 16 channels with total frequency of up to 1.25 MHz and 12 bits resolution. In our system, 14 channels are used at a sampling rate of 22.05 kHz each. Each channel is formed by a microphone connected to a custom-built low-noise, low-distortion preamplifier on an AD797 chip. The preamplifier output is fed to the data acquisition board. The microphones are divided into two blocks of 7 microphones, forming two “ears” of the system. Each ear is an approximately 30 cm square piece of cardboard. The microphones are arranged in a circular pattern with 6 microphones at the circumference and one at the center. These two “ears” are attached to the wall of the videoconferencing room with a separation of about 1 m. The audio channel transmitted to the remote site is formed from the signals received by all microphones by noise filtering and adaptive beamforming. The PC sound card is used to output the beamformed signal in real-time.

The video localization and tracking module uses two Sony EVI-D30 video cameras connected to Matrox Meteor II digital video capturing board. The camera has a motorized pan-tilt head and variable zoom lens and can be controlled via a serial interface by software. Camera output is split with one channel being captured by the software to perform video localization and tracking algorithms and another channel connected to the multiplexor. Our system is set up so that video based processing is secondary to the audio processing, and the video analysis frame rate is variable governed by available resources. However, as far as the output image is concerned, the use of direct multiplexed output from the cameras ensures that no frame loss occurs. The multiplexed output is controlled by a computer controlled switch so that the picture from any of them can be transmitted to the remote site. The two cameras could also be used to get video-based depth information by exploiting stereo cues, but we have not implemented this yet.

### 4. ALGORITHMS

**Acoustic Algorithms.** Time differences of arrivals (TDOAs) between microphones are determined by computing generalized cross-correlations (GCC) between signals arriving at the  $N$  microphones and obtaining the peaks. The signal at microphone  $i$  is given by  $S_i(t)$  and the Fourier transform of this signal is  $X_i(\omega)$ . The GCC is computed as

$$C_{ij}(\tau) = IFFT[W(\omega)X_i(\omega)X_j^*(\omega)](\tau) \quad (1)$$



where  $W(\omega)$  is a frequency weighting function. Following [2] and assuming that the noise is mostly acoustic and thus the same for all channels, we use the inverse of the noise power spectrum as a weighting function ( $W(\omega) = |N(\omega)|^{-2}$ ) and take the TDOA to be  $t_{ij} = \arg \max_{\tau} C_{ij}(\tau)$ . The noise power spectrum is initially obtained with no sound sources.

Computed TDOAs are used to determine the azimuth and elevation of the source. (It is possible to compute the distance to the source also, but the solution with a single array is less robust). Define  $\chi_i$  to be the distance from  $i^{th}$  microphone to the source; the TDOAs  $t_{ij}$  are related to distances as  $ct_{ij} = \chi_i - \chi_j$ , where  $c$  is the sound speed. From the measurements of TDOAs one can obtain  $C_{ij}^N$  such equations; only  $N - 1$  of them are independent. To solve this overdetermined system, different methods can be employed. Our algorithm separates the problem into two parts. First, we find the distances between the source and the receivers up to an additive constant by solving the linear system of equations  $\chi_i - \chi_j = ct_{ij}$  using constrained  $L_1$  (CL1) optimization algorithm [4] that has the advantage of being able to take into considerations different environmental, computational and physical constraints (for example, all distances must be non-negative, and all time delays should obey the rule  $t_{ik} = t_{ij} + t_{jk}$  within the precision of the measurements). This has the advantage that TDOA outliers are excluded from computations automatically.

The CL1 solution has one  $\chi_i$  set to be zero, which corresponds to the receiver closest to the source. This is sufficient to estimate the azimuth  $\phi$  and elevation  $\theta$  of the source. Using the plane wave approximation and simple geometric derivations, we can write

$$A(x_i - x_1) + B(y_i - y_1) + C(z_i - z_1) = -(\chi_i - \chi_1), \quad (2)$$

where  $A = \sin \phi \cos \theta$ ,  $B = \sin \theta$ ,  $C = \cos \phi \cos \theta$ . This can be again solved using CL1 for 3 unknowns and  $N - 1$  equations with appropriate constraints on  $A$ ,  $B$ , and  $C$ .

These solution is obtained separately for the two microphone arrays, and the bearings obtained intersected to find the source location in 3D. If the arrays are at the same height, the distance between their centers is given by  $D$  and the azimuths and elevations are  $(\phi_1, \theta_1)$  and  $(\phi_2, \theta_2)$  respectively, with the source coordinates

with respect to the array center given as

$$Z = \frac{D}{(\tan \phi_2 - \tan \phi_1)}, \quad X = Z \tan \phi_1, \quad Y = Z \frac{\tan \theta_1}{\cos \phi_1}. \quad (3)$$

The coordinates of the source are transformed into the camera coordinate frame and camera look angles are computed for each camera. To prevent camera motion due to false source detection, several methods are employed. First, the acoustic signal is checked to ensure there is sufficient power in the speech frequency bands; frames with low signal power are not processed. After that, additional filtering is performed based on acceptance regions. The algorithm keeps track of the last  $M$  pairs of source bearing angles  $(\phi, \theta)$  in the camera coordinate frame in the angle buffer. The last obtained bearing angles are checked if they belong to a cluster of similar values stored in the buffer; that is, if the buffer contains more than  $\beta M$  values that are within the  $\epsilon$  (using  $L_2$ -norm) to the values in question, the value is accepted; otherwise, the algorithm decides that there is no reliable source position data for the current time frame. This has the additional advantage that short sounds that are usually not informative do not cause jerky camera motion.

When the sound source coordinates are available from video or audio localization module, the acquired data from the microphone array can be processed by the beamforming algorithm to enhance the SNR. The beamformed signal  $\hat{S}(t)$  can be expressed as

$$\hat{S}(t) = \sum_i S_i(t + \tau_i - \tau_{min}), \quad (4)$$

where  $\tau_i$  is a delay of sound propagation from the source located at  $(x_s, y_s, z_s)$  to the  $i^{th}$  receiver located at  $(x_i, y_i, z_i)$  and can be expressed as

$$\tau_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2} / c. \quad (5)$$

and  $\tau_{min} = \min_i \tau_i$ .

**Video Algorithms.** Once the audio component of our system locates a source and moves the camera to get the source in the field of view, the camera is then controlled by a vision tracking system which tracks the head of the person in the view. The head is modelled by an ellipse with a fixed vertical orientation and a fixed aspect ratio of 1.2 similar to [8]. We use a condensation tracker [9] to track the ellipse, whose state is given by  $\mathbf{s} = (x, y, \sigma)$ , where  $(x, y)$  is the center of the ellipse and  $\sigma$  is the minor axis length of the ellipse. The tracking algorithm is based on a Monte Carlo technique, where the probability density function (pdf) is represented by a set of random samples. As new information becomes available, the posterior distribution of the state variables is updated by recursively propagating these samples (using a second order constant velocity model) and resampling. We use quasi-random points for sampling instead of the standard pseudo-random points since these points improve the asymptotic complexity of the search (number of points required to achieve a certain sampling error), can be efficiently generated and are well spread in multiple dimensions (see [14] for details). For a given tolerance to tracking error, the quasi-random sampling needs significantly lower number of sampling points (about 1/2) as compared to pseudo-random sampling, thereby speeding up the execution of the algorithm significantly. Our measurement model is a combination of two complementary modules (see [8] for why this is good), one that makes measurements based on the object's boundary and the other that focuses on the object's interior (color histograms [10]).

## 5. SOFTWARE IMPLEMENTATION

The system is implemented on a dual-processor Pentium III 500 MHz personal computer with 512 MB of RAM which performs audio and video data acquisition, acoustic source localization and tracking, video object localization and tracking, camera control and beamforming in real-time. In this section we describe the software details of the implementation of algorithms described above. The software is written in C/C++ under Windows NT 4.0. There are separate threads for the acoustic localization algorithms (actually, two threads are used here to parallelize FFT computations), video localization, video tracking, camera control and beamforming. The threads are synchronized using Windows NT non-busy wait. The acquisition hardware allows for direct transfers of the digitized sound to the host memory using DMA and PCI bus-master access mode; thus, once the data acquisition is started, it fills the buffer pre-allocated in the main memory transparently in the background without loading the host CPU by data transfers. Once the buffer is exhausted, it is recycled and acquisition restarted automatically.

To perform a location estimate of a sound source, the current data position in the buffer is retrieved via an interface call to the board driver, corresponding to the most recent window of data. The data for the window are retrieved, de-multiplexed and stored in a format suitable for doing FFTs for computing TDOAs. Two threads are started to perform parallel FFT of input channels, complex multiplications and inverse FFT of pairwise products to extract GCC peaks. These threads need to be resynchronized after doing all forward FFTs. The GCC peaks are used to obtain the bearing angles as described above. The strength of the peaks and frequency content of the signal is used in combination with heuristic strategies to determine the presence and origin of the sound. Using information from both subarrays, the 3D-position of the source is obtained. Finally, if the source is determined to be valid, its coordinates are passed to the camera control thread and the camera rotated and zoomed to get the source in the field of view.

The sound transmitted to the remote site is formed in real-time by the beamformer thread. This thread can be turned off if there is no speech signal for a long time, if the video tracker loses the target, or if multiple sound sources are detected; in this case, the sound received at the remote site is a copy of the sound from one microphone.

Since most of the CPU time is spent in the FFT routines, the acoustic localization thread is given priority in the sense that video tracking is allowed to run only after the audio data is processed. (Note, however, the smart switching strategy which ensures that video at the required frame rate is output). The video processing algorithms grab and process one video frame between position estimates for the total rate of about 10 fps for data analysis. (It is possible to process the data at full camera rate of 30 fps if the system is set up using two separate computers – one for audio and one for video processing. In fact, this is the setup used initially in the system. The communication in this configuration was done with UDP packets over local Ethernet). The beamforming takes up the remaining time. The output audio signal is delayed by the beamformer by the time of about half of the main program loop length (185 ms), and a loss of synchronicity between audio and video data is noticeable; we are working on eliminating it.

## 6. SUMMARY AND FUTURE WORK

We have described a prototype next generation multi-camera multi-microphone array system for use in videoconferencing. This system can be implemented on a regular personal computer and can provide functions such as tracking multiple speakers using video and audio information, beamforming, person tracking, and intelligent frame switching. We anticipate continued development of the system will lead to enhanced user interfaces for remote communications and collaboration.

## 7. REFERENCES

- [1] Buxton, W., Sellen, A. & Sheasby, M. (1997). Interfaces for multiparty videoconferencing. In K. Finn, A. Sellen & S. Wilber (Eds.). Video Mediated Communication. Hillsdale, N.J.: Erlbaum, 385-400.
- [2] D.Feitelson, A.Weil. "A robust method for speech signal time-delay estimation in reverberant rooms", *ICASSP-96, Atlanta*.
- [3] Zotkin, D., Duraiswami, R., Hariatoglu, I., Otsuka, T., and Davis, L.S. (1999). "An audio-video front end for multimedia applications," Submitted to IEEE SMC2000.
- [4] I. Barrodale and F.D.K. Roberts, (1973) "An improved algorithm for discrete  $l_1$  linear approximation," *SIAM J. Numer. Anal.*, **10**, 839-848
- [5] Anderson, A. H., O'Malley, C., Doherty-Sneddon, G., Langton, S., Newlands, A., Mullin J., Fleming, A. M. & Van der Velden, J. (1997). "The impact of VMC on collaborative problem solving". In Finn, K. E. et al. (Eds.) Video-Mediated Communication. L. Erlbaum Assoc.
- [6] D. Brown (1999), Videoconferencing and Multimedia Delivery, Network Design Manual, <http://www.networkcomputing.com>.
- [7] <http://www.picturetel.com/products/80p.htm>
- [8] S. Birchfield (1998) Elliptical head tracking using intensity gradients and color histograms. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Santa Barbara, California, June 1998.
- [9] M. Isard and A. Blake.(1996) Contour tracking by stochastic propagation of conditional density, in Proc. European Conf. Computer Vision, pages 343-356, Cambridge UK, 1996.
- [10] M. Swain and D. Ballard.(1991) Color indexing. International Journal of Computer Vision, 7(1), pp. 11-32.
- [11] C. Wang and M. Brandstein.(1998) A hybrid real-time face tracking system, in Proc. ICASSP98, Seattle, WA, May 12-15 1998.
- [12] C. Wang and M.Brandstein.(1999) Multi-source face tracking with audio and visual data, in IEEE Intl. Workshop on Multimedia Signal Processing, Copenhagen, Denmark, September 13-15 1999.
- [13] M. Bett, R. Gross, H. Yu, X. Zhu, Y. Pan, J. Yang, A. Waibel.(2000) Multimodal meeting tracker, in Proc. of RIAO2000, Paris, France, April 2000.
- [14] V. Philomin, R. Duraiswami, L.S. Davis (2000). Quasi-random sampling for Condensation. Proceedings of the European Conference of Computer Vision, Dublin, Ireland 2000.