Video Conferencing for a Virtual Seminar Room

S. Forchhammer[§], A. Fosgerau[§], P.S.K. Hansen[+], R. Sharp[+], E. Todirica[+], A. Zsigri[§]
[§]Research Center COM, [+]Informatics and Mathematical Modelling
Technical University of Denmark

Abstract - A PC-based video conferencing system for a virtual seminar room is presented. The platform is enhanced with DSPs for audio and video coding and processing. A microphone array is used to facilitate audio based speaker tracking, which is used for adaptive beam-forming and automatic camera-control. Recently the system was demonstrated between two geographically separated universities in Denmark. The communication was based on the use of UDP/IP. Results are reported and an overview of the system is given.

INTRODUCTION

In an interactive distributed multimedia (DMM) system, users at physically separated sites communicate with one another using a variety of media: audio, video, graphics, text, etc. Each user experiences what happens at all the sites in real time, and thus gets the impression of being in the same location as the other users, while in reality they may be many (hundreds of) kilometers apart. As a practical example of a DMM system, we have in the RTMM project at the Technical University of Denmark (DTU) aimed at the development of a *Virtual Seminar Room*: Users have the impression of being in a common teaching room, where they can see and hear one another, and have shared access to various types of *virtual equipment*, such as a virtual blackboard and a virtual slide projector.

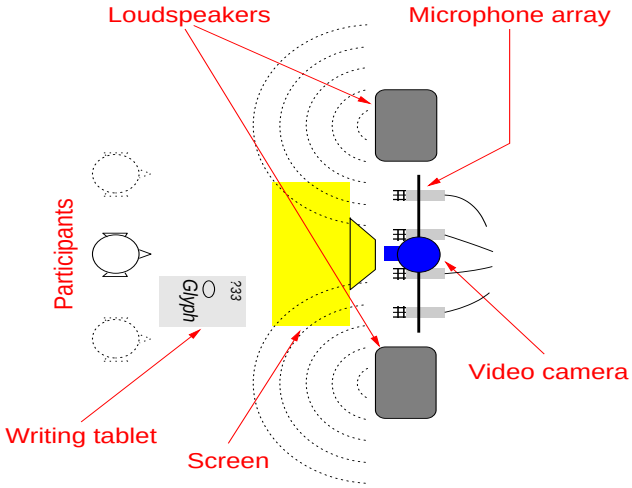

Figure 1. Overview of virtual seminar room.

The schematic layout of each site is shown in Figure 1. A number of technical challenges are associated with systems of this type, in particular to provide:

- o Real-time capture/replay of high-quality audio, video etc.
- o Real-time low-latency encoding and decoding of audio and video for transmission.
- o Low-latency transmission of data between multiple sites.

This requires coordinated research into signal processing, coding, networking and operating system design. A general PC was chosen as the platform and enhanced with digital signal processors for the audio and video. Recently a demo was carried out between DTU and the University of Aarhus. This paper will present the specific system for that demo and measurements for this system as well as our more general ideas.

AUDIO PROCESSING - Microphone Arrays and Video Camera Steering

The requirements to the audio system are high quality sound with clear intelligibility and speaker identification. We have chosen a microphone array and aim at real time on a digital signal processing core. People should be able to move freely around without wearing or holding a microphone. Thus we deal with multiple source signals, multiple microphones and multiple loudspeakers running in real time. A key problem is to estimate the acoustic source location in a real reverberant environment, which can then be used for video camera steering or to control a robust adaptive beamforming algorithm [1].

The source location algorithm considered tries directly to determine the relative delay between the direct path of two estimated channel impulse responses. This approach, known as the adaptive eigenvalue decomposition algorithm (AEDA) [2] performs well in a reverberant environment, and the focus here will be on a low-complexity, time-domain implementation. The AEDA is based on the fact that $x_1(n)*g_2(n) = x_2(n)*g_1(n)$, where g_1 and g_2 are the impulse responses from the source to the microphones, x_1 and x_2 are the microphone signals, and $*$ denotes convolution. Let h_1 and h_2 be adaptive filters applied to x_2 and x_1, respectively. The problem of minimizing the error signal $e(n) = s*(g_1*h_2 - g_2*h_1)$ implies the unique solution $h_1 = g_1$ and $h_2 = g_2$, iif the correlation matrix of the source signal $s(n)$ has full rank, and the two impulse responses do not share any common zeros [2]. In this case the source location can be obtained from this solution since the time difference between the largest coefficient in h_1 and h_2 (the direct path) is the time delay of arrivals (TDOA).

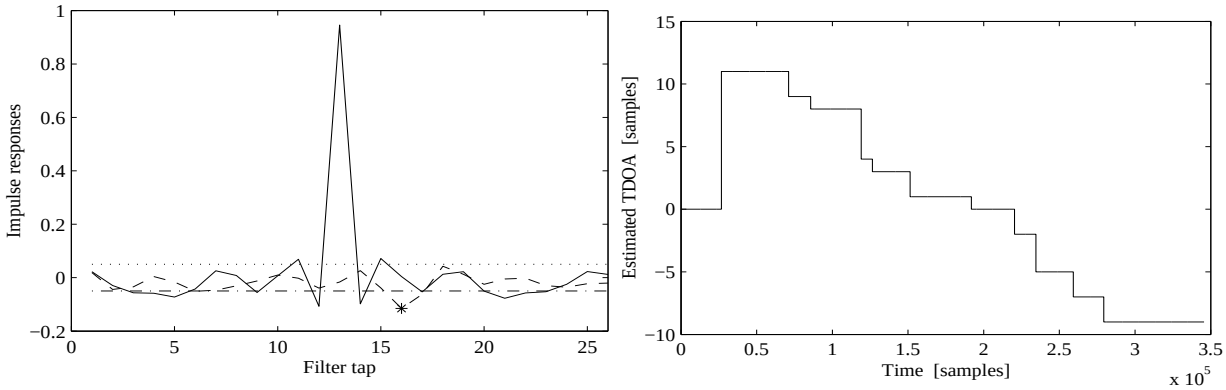

Figure 2. a) Estimated impulse repsonses h_1 and $-h_2$. b) Estimated location for a person walking from side to side.

In the video conferencing application the microphone array has been setup for beamforming, which is not optimal for the localization problem due to the small spacing between the microphones. The result is an ill-conditioned acoustic system, where g_1 and g_2 are very similar. The challenge is to devise a scheme which will make the AEDA converge to the desired solution under these conditions, or more precisely, to a solution where the two largest coefficients are revealed.

As we want to cover source locations in the range -90 to 90 degrees, the length of the adaptive filters must be $M = 2\ F_s\ d/c + 1$, where F_s is the sampling rate, d the distance between the microphones, and c the speed of sound. In order to take into account negative and positive relative delays, the middle coefficient in h_1 is initialized to one which will be considered as an estimate of the direct path of g_1. During adaption, a mirror effect will appear in the filter h_2, since the direct path of g_2 will control the adaption in order to minimize $e(n)$. Thus the time delay of arrivals will simply be the difference between the indices corresponding to these two peaks (Fig. 2a). The problem is that h_1 and h_2 will diverge from this initial solution due to the ill-conditioned acoustic system. The following scheme which maintains dominant direct path coefficients is proposed:

```
For each sample
    Update h1 and h2 using AEDA.
    If peak value of h2 > threshold then update TDOA.
    Reset h1 and h2 every N samples:
      Keep only peak value of h2 multiplied by a forgetting factor.
      Reset h1 to have only a dominant middle coefficient.
```

The threshold determines the noise sensitivity, the forgetting factor determines the tracking properties, and the reset interval controls the chosen solution h_1 and h_2. The computational complexity is mainly due to the AEDA part, i.e., $O(18M)$ flops per sample using a constrained NLMS algorithm. The described source localization method has been implemented on a system consisting of a front-end 4-element linear microphone array with preamps, a DSP board (Bittware Spinner), and a steerable camera (Sony EWI-D31, Fig. 1). Only a single TDOA was estimated using the two outer microphones, and parameters d = 13.5 cm, F_s = 32 kHz, and M = 26. The first test was in a small office, where the estimated location (TDOA) for a person walking from one side

of the array to the other is shown in Fig. 2b. The system was also tested in real-life in a larger room during the demo video conference session with all the authors present. The locator demonstrated good speaker tracking performance and robustness to noise.

VIDEO CODING

In the current version, the video is in SIF/CIF format (352x288 pixels, non-interlaced, 25 frames per second). The video is coded using JPEG coding [3] of each frame (M-JPEG) on a Matrox video board (Matrox Marvel G400) . The rate of the video delivered by the board may be reduced by the PC by dropping frames at the encoder side. A unique synchronization word is appended to each frame, to provide fast (re-)synchronization at the receiving side. M-JPEG was chosen to achieve low-delay at the expense the bandwidth-quality (rate-distortion) performance. On the receiving side the M-JPEG is decoded in software based on the IJG JPEG library [3]. The decoded video is displayed using SDL rutines. This video processing for decoding and display is the most time consuming part for the host PC. The quantization of the JPEG DCT (Discrete Cosine Transform) coefficients is controlled by the quality parameter. (In IJG the quality parameter q controls the JPEG *scale* parameter by $5000/q$ for $q<50$ and $200 - 2*q$ for $q> 50$). Experimentally a quality factor of $q= 30$-40 was evaluated to provide sufficient quality. The value $q= 30$ was used in the tests.

We are working on implementing the video processing on the Equator MAP-CA platform [4], which is a state-of-the-art video DSP. We are using the Equator Shark development board. This will off-load the host PC and boost the video capability of the system. (The MAP-CA promises up to 11-30 GOPS for 8 and 16 bit operations.) Currently we are working on implementig the M-JPEG decoding and display in full PAL resolution on the platform as well as MPEG encoding.

NETWORK INTERFACE

Transmission through the network is based on the User Datagram Protocol (UDP)/IP protocol suite. Socket programming on UNIX [5] is used to establish connection between the IP protocol and the RTMM stream layer. The type of connection is attributed to the created sockets.

Networking activities in the RTMM project are performed by the network system components, responsible for two main functions: One is the data exchange between the network system components and the other is to create the required network connection type with Quality of Service (QoS) support. Three steps are required in the creation of a network connection: 1) create a network system component, waiting for the incoming calls on the receiver site, 2) create a network system component on the transmitter site, creating and sending the data and 3) create a logical network connection between them, responsible for the traffic transmission on the physical connection.

Separate connections are created for the different system system components (video, audio). From the QoS class the bandwidth of the stream and the maximum jitter parameters are known, these parameters cannot be changed after the network socket has been created, so the connection must be taken down and created again with a new QoS value if the parameters are unsuitable.

Use of UDP is important for the multicast support that will be tested in the near future. Asynchronous Transfer Mode (ATM) on Synchronous Digital Hierarchy (SDH) support is also planned with ATM Adaptation Layer (AAL5) protocol and Permanent Virtual Connection (PVC) connections.

SYSTEM CONCEPTS

Conceptually, the system is composed of a set of *system components*, joined by *logical connections* through which various types of *streams* of data (audio, video, text, images) of various *qualities* can be passed. A system component typically consists of hardware and software for generating, consuming or transforming a particular type of stream. This architecture offers applicaton implementers a simple but efficient *toolbox* from which to construct systems, based on a standard operating system -- currently Linux. In fact, implementers are offered functions for defining and handling system components at an API which we denote the Stream Layer interface.

This has been designed to give application designers a wide range of suitable operations with as small an overhead as possible. Some timing measurements are given in the following section of the paper.

RESULTS ON SYSTEM PERFORMANCE

To illustrate the operation of the system in practice, a series of timing measurements have been made on the system. These are summarised in Table 1. The times referred to in the table are measured as follows:

o Stream layer delays: Measured on source site as the time from the start of transmission of a frame from the source video system component until the end of receipt by the source network system component. These delays illustrate the overhead introduced by the Stream Layer software.

o Interframe delays: Measured on destination site as the difference between successive timestamps on frames which actually reach the destination video system component. These timestamps are added to the frames in the source video system component at the instant when the frame is completely captured.

o Video processing delays: Measured in the destination video system component as the time from the instant when the frame is completely received by the component until the instant when the presentation function (which puts the frame into the frame buffer) returns. This includes the time required for both software decoding and actual presentation of a video frame.

o Network delays: Measured on destination site as the time from the start of transmission of a frame from the source network system component until the instant when the entire frame has been received by the destination network component. The two sites in Lyngby and Aarhus communicate over a distance of about 400 km. by using UDP/IP over the Danish Research Network. This network has enough spare bandwidth to satisfy our needs. The delay is measured as half the total delay from DTU to Aarhus and back again, where the system in Aarhus functions as a reflector.

o End-to-end video delays: Measured as the time from the instant when the frame is completely captured until the instant when the presentation function returns, when the video frames are sent to Aarhus and back again, with the system in Aarhus functioning as a reflector. The network transmission time is again halved, so that the value given in the table is comparable to the genuine end-to-end delay for video between two sites.

	Minimum	Maximum	Average	Std. dev.
Stream layer	0.03	0.52	0.08	0.03
Interframe	38.7	41.4	40.0	0.63
Video processing	26.4	33.2	27.8	0.52
Network	10.2	15.0	11.2	0.48
End-to-end video	37.3	44.8	39.2	0.72

Table 1. Processing times within the system (milliseconds).

With the set-up used in the demonstration, the rate of video frame loss was less than 1 frame per 1500 frames. The minimum, maximum and average bandwidth requirements for video were 3.15, 3.50 and 3.30 Mbit/s respectively, with a standard deviation of 0.04 Mbit/s. For audio the bandwidth was a constant 670 kbit/s for uncompressed mono CD quality. All the measurements were made over a period of about 1 minute (1500 frames at 25 frames/s). For transmission through the network, one IP packet was used to contain an encoded video frame (about 16.5 kbytes) or a mono audio packet of about 1 kbytes.

References

[1] T. Duer, P. S.K. Hansen and J. Aa. Sørensen: Robust Adaptive Beamformer with Reflection Suppression for Teleconferencing Applications, *Proc. Norsig* 2000, pp. 405-408, June 2000.

[2] Y. A. Huang, J. Benesty, and G. W. Elko: Microphone Arrays for Video Camera Steering, in S. L. Gay, J. Benesty (eds.), *Theory and Appl. of Acous. Sign. Proc. for Telecom.*, S. L. Gay, J. Benesty (eds.), Kluwer, Boston, 2000.

[3] *http://www.ijg.org*

[4] *http://www.equator.com*

[5] Douglas E. Comer, David L. Stevens – Internetworking with TCP/IP volume III