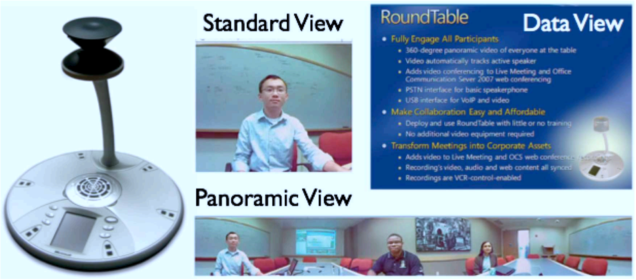
ViewMark: An Interactive Videoconferencing System for Mobile Devices

Shu Shi [#] and Zhengyou Zhang [*]

[#] *Department of Computer Science*
University of Illinois at Urbana-Champaign
201 N Goodwin Ave, Urbana, IL 61801, USA
shushi2@illinois.edu

[*] *Microsoft Research*
One Microsoft Way, Redmond, WA 98052, USA
zhang@microsoft.com

***Abstract*—ViewMark, a server-client based interactive mobile videoconferencing system is proposed in this paper to enhance the remote meeting experience for mobile users. Compared with the state-of-the-art mobile videoconferencing technology, ViewMark is novel in allowing a mobile user to interactively change the viewpoint of the remote video, create viewmarks, and hear with spatial audio. In addition, ViewMark also streams the screen of the presentation slides to mobile devices. In this paper, we introduce the system design of ViewMark in details, compare the devices that can be used to implement interactive videoconferencing, and demonstrate the prototype system we have built on Windows Mobile platform.**



- Fully Engage All Participants
  - 360-degree panoramic video of everyone at the table
  - Video automatically tracks active speaker
  - Adds video conferencing to Live Meeting and Office Communication Sever 2007 web conferencing
  - PSTN interface for basic speakerphone
  - USB interface for VoIP and video
- Make Collaboration Easy and Affordable
  - Deploy and use RoundTable with little or no training
  - No additional video equipment required
- Transform Meetings into Corporate Assets
  - Adds video to Live Meeting and OCS web conference
  - Recording's video, audio and web content all synced
  - Recordings are VCR-control-enabled

Fig. 1. Microsoft RoundTable System [2]

I. Introduction

In this paper, we present a mobile-to-conference-room videoconferencing system that can significantly improve the remote meeting experience for mobile users. Traditionally, the mobile user can only join a remote conference through teleconferencing systems, which only deliver voice signals. The explosive development of smart phones and mobile networks makes videoconferencing possible and popular on mobile devices. The current mobile videoconferencing technologies allow the mobile user to see and talk to other mobile or desktop users wherever Wi-Fi or 3G data network is available. However, even the state-of-the-art mobile videoconferencing technology (e.g., Apple FaceTime [1]) does not provide good remote meeting experiences. For example, in the scenario of a conference room meeting where people are sitting around a table, it is difficult for the remote mobile user to see everyone in the conference room clearly because he has no control over the video camera in the conference room.

RoundTable [2] is a videoconferencing device designed by Microsoft (Fig. 1). It features a 360-degree camera and provides remote conferencing participants with panoramic video of everyone sitting around the conference table. In addition, RoundTable can automatically detect active speakers in real time, generate high-resolution video of the active speaker in a meeting, and switch between different meeting participants as they speak. RoundTable system is integrated with Windows Live Meeting, which can also stream the PowerPoint presentation screen. From our experience with RoundTable and Windows Live Meeting, we have learnt three important features necessary for good remote meeting experiences:

- *Environment*: The remote user should be able to see the whole conference room and everyone in the room;
- *Active Speaker*: The remote user should have a clear view of the person who is currently talking;
- *Presentation Data*: The remote user should have access to the presentation data (e.g., presentation slides).

We propose ViewMark: a server-client based mobile videoconferencing system that has all three features listed above. ViewMark targets only the scenario of mobile to conference room videoconferencing. The mobile device runs as the client and connects to the video server deployed in the conference room. ViewMark enables the mobile client to interactively change the viewpoint of the camera connected to video server, so that the mobile user is able to see everyone in the conference room. With the support of appropriate videoconferencing devices (e.g, RoundTable), the spatial audio corresponding to the camera viewpoint is synthesized in real-time to provided the most immersive experience. The camera viewpoint can be saved as a viewmark. The mobile user can create the viewmark for any meeting participant of interest (say the team leader). Clicking on the pre-saved viewmark will fast switch the camera to that individual. Furthermore, ViewMark allows the mobile user to have a better view of presentation slides. The presentation screen is encoded as a data stream and sent together with A/V streams to the mobile client.

978-1-4577-1434-4/11/$26.00 ©2011 IEEE

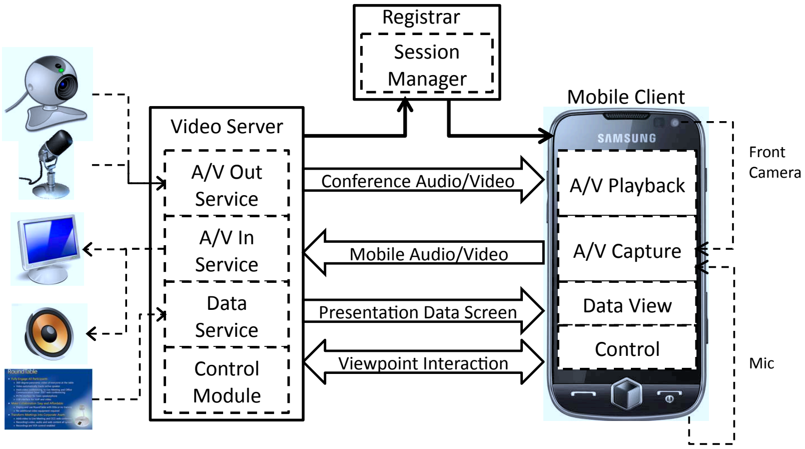


Fig. 2. ViewMark System Framework

Of all the typical videoconferencing systems [1]-[6] on both mobile and desktop we are aware of, ViewMark is the first system that combines all of the following elements: videoconferencing, viewpoint interaction, and data streaming, into one mobile application. ViewMark uses a mobile-to-server framework. Compared with other server-to-server or mobile-to-mobile systems, the mobile-to-server system can take advantage of both the flexibility of mobile devices and the computing resources on server. For example, the touch screen and accelerometers of the mobile device can be used to develop new user interaction methods. Meanwhile, the computation-intensive algorithms (e.g., face detection, spatial audio synthesis) can easily run on the server.

For the rest of this paper, we introduce the design of ViewMark in details. In Section II, we discuss several design issues of ViewMark, including the overall framework, the user interface for mobile client, and the server design when using different videoconferencing devices. In Section III, we present the ViewMark prototype system we have implemented, and share some design and implementation experiences. We conclude the paper in Section IV.

II. VIEWMARK SYSTEM DESIGN

A. *System Overview*

ViewMark has a simple server-client architecture. Fig. 2 gives an illustration of system framework in diagrams. The video server is deployed in the conference room and the mobile clients are connected to the video server through wireless networks. In order to start videoconferencing, the video server should start first and register a meeting session with *Registrar*. The mobile client can request the whole meeting session list from *Registrar* and select to join one meeting. *Registrar* returns the detail connection information of the specified meeting session and the mobile client can directly connect to the video server. After the connection is established, the video server sends audio and video streams (recorded by the videoconferencing devices in the conference room) along with the data stream (the screen copy of the presentation computer) to mobile client while the mobile client sends back its own audio and video streams (recorded by the mobile microphone and front camera). In addition to the data channels, there is a separate control channel reserved for video server and mobile client to exchange interaction messages.

The A/V streaming components of ViewMark is similar to other conventional videoconferencing systems. Therefore, we mainly introduce viewpoint interaction and data streaming.

B. *Viewpoint Interaction*

Obviously, special videoconferencing devices are needed to support viewpoint interaction in the proposed ViewMark system. We have studied two different approaches:

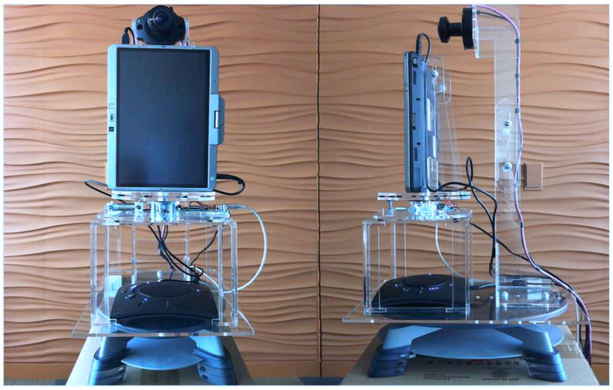


Fig. 3. SpinTop with webcam, micphone, display, and speaker

1) **SpinTop**: SpinTop is an experimenting prototype device designed in Microsoft. It has a turntable that can spin and the driven motor is programmatically controlled through USB. We can mount a webcam on the SpinTop (Fig. 3) and this combination can be effectively used for viewpoint interaction. The video server should control the motor to spin when receiving the command to change viewpoint, and update the angle position of SpinTop to mobile client.

Interaction latency [7] is a major disadvantage of using SpinTop for viewpoint interaction. Interaction latency is defined as the time from the generation of user interaction request till the video captured at the target viewpoint appears on the mobile screen. Since SpinTop is connected to the video server and spinning is a mechanical process, when the mobile client detects any user interaction, it takes at least a network round trip time and the mechanical spinning time before the video at the requested viewpoint shows on the mobile device. Due to the high-latency nature of wireless networks used by mobile devices, the long interaction latency can impair the user experience in the scenario when the mobile user needs to frequently switch from one speaker to another.

2) **RoundTable**: As we have introduced at the beginning, RoundTable can generate 360-degree panorama video in real-time. The video server can simply send the whole panorama video to mobile client, which maintains the camera viewpoint and displays only the part of the whole panorama that corresponds to the current camera viewpoint.

Using RoundTable and panorama video has two advantages over SpinTop. First, it supports individual viewpoint interaction of multiple mobile clients. Second, the problem of long interaction latency is solved. However, sending the whole panorama video significantly increases the network bandwidth. A favorable optimization is to crop the panorama video to different sub-views on the video server and send only the useful views to mobile client. We consider only the views that are viewmarked or close to the current camera viewpoint are useful. This optimization can effectively save the network bandwidth for panorama streaming and maintain the low interaction latency in most cases.

C. Data Streaming

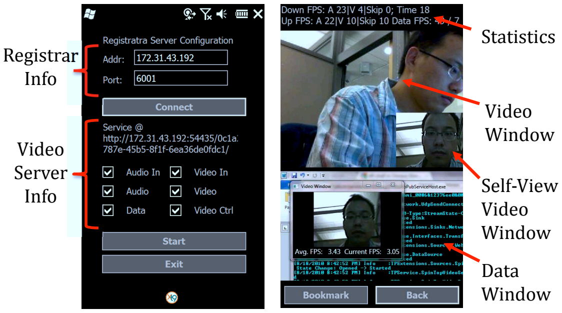
The data stream has very different characteristics from the video stream in our system. First, it does not update frequently. For most conference presentations, the speaker may spend tens of seconds or up to minutes on a single slide. Second, the presentation data needs to maintain high quality for the mobile user to actually read the texts in the slides. Therefore, we integrate the tool introduced in [8], a different approach from normal video coding, to encode the presentation data screen. Several techniques, such as adaptive screen compression and interactive ROI control are applied to improve the system performance and user experience.

III. Implementation

In this section, we present the ViewMark prototype system we have implemented[1]. The video server runs on a Windows 7 laptop and the mobile client is a Samsung Omnia II smart phone based on Windows Mobile 6.5. Both server and client connect to Microsoft Corporation network through Wi-Fi 802.11g. The system supports both RoundTable and SpinTop as the videoconferencing device. UDP is used for A/V streaming between video server and mobile client, and TCP is used for the communication in control channel.

A. Mobile Interface

When the mobile app for ViewMark starts, the initial setting interface shows up. The mobile app tries to connect to *Registrar* first and obtain the currently available meeting session list. After the user joins the meeting, the mobile app switches to the videoconferencing view (Fig. 4). The user interface comprises three major views: remote video (from video server), self-view video (from mobile front camera), and presentation data. The statistics of audio and video streaming is printed on top of the screen. The viewpoint of remote video can be changed by moving a finger in the remote video window (Fig. 5). Double clicking the remote video window creates a viewmark and double clicking the previously saved viewmark will fast switch the remote video back to the corresponding viewpoint.



Registratra Server Configuration

Addr: 172.31.43.192

Port: 6001

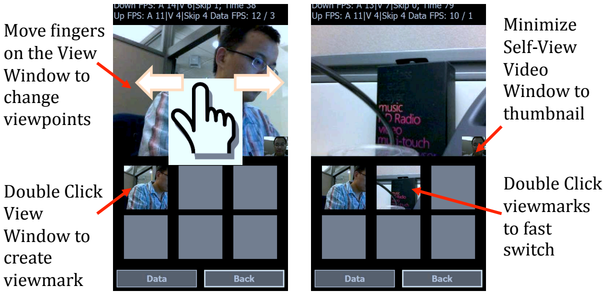
Service @
http://172.31.43.192:54435/0c1a:
787e-45b5-8f1f-6ea36de0fdc1/

☑ Audio In ☑ Video In
☑ Audio ☑ Video
☑ Data ☑ Video Ctrl

Down FPS: A 23|V 4|Skip 0; Time 18
Up FPS: A 22|V 10|Skip 10 Data FPS: 4? / 7

Avg. FPS: 3.43 Current FPS: 3.05

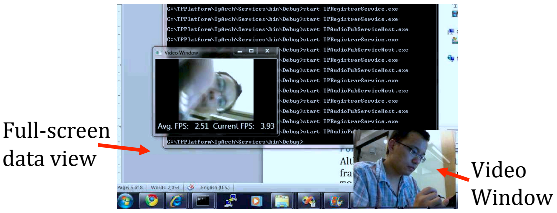
Fig. 4. ViewMark Mobile screen copy: initial setting interface (left) and videoconferencing view interface (right)


Move fingers on the View Window to change viewpoints
Double Click View Window to create viewmark
Minimize Self-View Video Window to thumbnail
Double Click viewmarks to fast switch
Data
Back
Data
Back

Down FPS: A 14|V 6|Skip 1; Time 38
Up FPS: A 11|V 4|Skip 4 Data FPS: 12 / 3

Down FPS: A 13|V 7|Skip 0; Time 79
Up FPS: A 11|V 4|Skip 4 Data FPS: 10 / 1

Fig. 5. ViewMark Mobile screen copy: viewpoint interaction and viewmarks



```
C:\TPPlatform\TpArch\Services\bin\Debug>start TPRegistrarService.exe
C:\TPPlatform\TpArch\Services\bin\Debug>start TPRegistrarService.exe
C:\TPPlatform\TpArch\Services\bin\Debug>start TPAudioPubServiceHost.exe
C:\TPPlatform\TpArch\Services\bin\Debug>start TPRegistrarService.exe
\Debug>start TPAudioPubServiceHost.exe
\Debug>start TPRegistrarService.exe
\Debug>start TPAudioPubServiceHost.exe
\Debug>start TPRegistrarService.exe
\Debug>start TPAudioPubServiceHost.exe
\Debug>start TPRegistrarService.exe
\Debug>start TPAudioPubServiceHost.exe
\Debug>start TPRegistrarService.exe
\Debug>start TPAudioPu
C:\TPPlatform\TpArch\Services\bin\Debug>
```

Avg. FPS: 2.51 Current FPS: 3.93

Fig. 6. ViewMark Mobile screen copy: full-screen data view

[1] A live demo recording is at http://research.microsoft.com/~zhang/Videos/ViewMarkDemo.wmv

TABLE I
VIEWMARK MOBILE PERFORMANCE

Media	Direction	Description	FPS	Bandwidth
Audio	Server-to-Mobile	PCM, Stereo, 16-bit, 16k sample rate	12.5	256Kbps
	Mobile-to-Server	PCM, Mono, 16-bit, 16k sample rate	10	128Kbps
Video	Server-to-Mobile	MJPEG, 320x240	8	320Kbps
	Mobile-to-Server	Raw YUV420, 176x144	4	1.2Mbps
Data	Server-to-Mobile	Support the screen resolution up to 1920x1080	1-4	200Kbps

Due to the size limitation of the mobile screen (800×480), the data view in Fig. 4 cannot clearly present the details in the presentation slides (e.g., the text of small font or images with a lot of details). Therefore, we add a feature of showing the presentation data in a full-screen view when the mobile device is in landscape orientation (Fig 6). A downscaled remote video is displayed in the corner so that the mobile user can watch the presentation slides and the speaker at the same time.

B. Performance

Since the main purpose of our prototype is to demonstrate the concept of interactive mobile videoconferencing, we did not focus on the optimization of A/V streaming. We present the videoconferencing related performance of ViewMark in Table 1. Due to the lack of hardware accelerated codec APIs when we implemented the prototype system, we chose to use uncompressed PCM for audio streaming, motion JPEG for the video streaming from video server to mobile client, and raw YUV420 for the video streaming from mobile client to video server. We believe the performance of videoconferencing can be easily improved as mobile devices become more powerful and standard APIs for videoconferencing are available.

The video server manages a control module that translates the viewpoint change commands to different operations for different videoconferencing devices, and therefore hides the information of videoconferencing devices from mobile clients. We have tested both RoundTable and SpinTop to connect into ViewMark system.

In order to leverage our auditory systems cocktail party effect, the audio in the remote meeting room is captured with multiple microphones and is presented to the mobile user as spatial audio so she can hear clearly even when multiple remote participants speak simultaneously.

C. Discussion

One important lesson we have learnt from this project is that the performance is a critical issue in designing a mobile-to-server system. The workload that runs smoothly on desktop may not perform the same on mobile due to insufficient computation resources. For example, we initially set the default audio chunk size to 320 bytes (equivalent to 10ms audio). This setting never caused any problem for our desktop program but led to serious performance issues on mobile, because the overhead of receiving an audio packet and passing to the audio play function can be larger than 10ms. We found that the audio chunk size should be set no smaller than 1280 bytes (equivalent to 40 ms) to allow normal audio playback on mobile devices. Therefore, we believe a good design of mobile-to-server system should always consider the limitation of mobile performance first.

IV. CONCLUSION AND FUTURE WORK

We have discussed a mobile videoconferencing system in this paper to present how server and mobile together can provide a better remote meeting experience. For the future work, we will exploit this integrated mobile-server solution for new features. For example, ViewMark system currently depends on the mobile user to manually switch between different viewmarks to find the active speaker. The video server can analyze the stereo audio streams, determine who is currently speaking in the meeting, and show hints on mobile app to suggest which viewmark to click. Another example is to use ViewMark for meeting recording. The viewpoint interaction provides extra semantic information. The video streamed to the mobile client is considered a good video record that includes all the important information in the meeting (e.g., the speech by important person) if the mobile user always switches the camera to the right person in time.

ACKNOWLEDGMENT

Anoop Gupta, Gavin Jancke, and Rajesh Hegde from Microsoft Research have offered great ideas and numerous supports during the whole project. Haicheng Sun and Yan Lu from Microsoft Research Asia have provided excellent tools for the development of ViewMark mobile applications.

REFERENCES

[1] Apple, "Facetime," http://www.apple.com/iphone/features/facetime.html.

[2] Microsoft, "Roundtable," http://en.wikipedia.org/wiki/Microsoft_Round Table.

[3] Cisco, "Tele-presence," http://www.cisco.com/en/US/products/ps7060/ index.html.

[4] Halo, "Hp," http://h20338.www2.hp.com/enterprise/us/en/halo/products.html.

[5] S. Fabri, S. Worrall, A. Sadka, and A. Kondoz, "Real-time video communications over gprs," in *3G Mobile Communication Technologies, 2000. First International Conference on (Conf. Publ. No. 471)*, 2000, pp. 426 –430.

[6] P. Kauff and O. Schreer, "An immersive 3d video-conferencing system using shared virtual team user environments," in *Proceedings of the 4th international conference on Collaborative virtual environments*, ser. CVE '02, 2002, pp. 105–112.

[7] S. Shi, M. Kamali, K. Nahrstedt, J. C. Hart, and R. H. Campbell, "A high-quality low-delay remote rendering system for 3D video," in *Proc. of the ACM International Conference on Multimedia (MM'10)*, Firenze, Italy, October 2010, pp. 601–610.

[8] W. Sun, Y. Lu, and S. Li, "Redi: an interactive virtual display system for ubiquitous devices," in *Proceedings of the international conference on Multimedia*, ser. MM '10, 2010, pp. 759–762.