
MobileVideoTiles: Video Display on Multiple Mobile Devices

Ming Li

Computer Graphics & Multimedia Group
RWTH Aachen, AhornStr. 55
52074 Aachen, Germany
mingli@cs.rwth-aachen.de

Kaspar Maximilian Scharf

Computer Graphics & Multimedia Group
RWTH Aachen, AhornStr. 55
52074 Aachen, Germany
kaspar.scharf@rwth-aachen.de

Leif Kobbelt

Computer Graphics & Multimedia Group
RWTH Aachen, AhornStr. 55
52074 Aachen, Germany
kobbelt@cs.rwth-aachen.de

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
MobileHCI '16 Adjunct, September 06-09, 2016, Florence, Italy
ACM 978-1-4503-4413-5/16/09
<http://dx.doi.org/10.1145/2957265.2961826>

Abstract

Modern mobile phones can capture and process high quality videos, which makes them a very popular tool to create and watch video content. However when watching a video together with a group, it is not convenient to watch on one mobile display due to its small form factor. One idea is to combine multiple mobile displays together to create a larger interactive surface for sharing visual content. However so far a practical framework supporting synchronous video playback on multiple mobile displays is still missing. We present the design of “MobileVideoTiles”, a mobile application that enables users to watch local or online videos on a big virtual screen composed of multiple mobile displays. We focus on improving video quality and usability of the tiled virtual screen. The major technical contributions include: mobile peer-to-peer video streaming, playback synchronization, and accessibility of video resources. The prototype application has got several thousand downloads since release and received very positive feedback from users.

Author Keywords

multiple mobile; tiled; displays; video streaming; synchronization; user interface

ACM Classification Keywords

H.5.1 [Multimedia Information Systems]: Video; H.5.2 [Information interface and presentation]: User Interfaces



Figure 1: “MobileVideoTiles” can combine up to 3 mobile displays to play local/online videos.

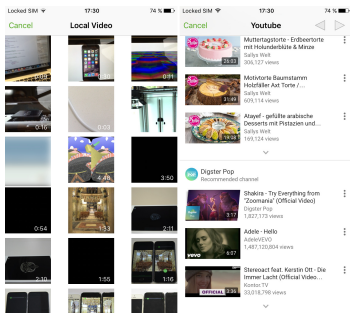


Figure 2: Local and online video browser.

Introduction

In recent years mobile phones or tablets have been equipped with high quality camera and fast processors, which makes them capable to capture and edit high definition videos. People like to use mobile devices to record their daily life and share the video content online or enjoy it together with their friends. When no large display (monitor or projector) is available, a group of people have to crowd together and watch a video on a small display of a mobile device, which disturbs the viewing experience. Considering that more than 25% mobile device owners possess more than one mobile devices [10], and additionally in social meetings usually everyone carries at least one mobile device, we have a good opportunity to leverage the combination of multiple co-located mobile devices for video playing. To achieve this, many algorithms have been proposed [9, 3, 5, 8]. However they mainly focused on the creation and interaction of the display surface. Technical challenges of video playback in the context of multiple mobile displays have not yet been fully investigated. Firstly, if a video is stored locally on one device, we must transmit it to other devices efficiently, i.e., short waiting time and acceptable visual quality. Secondly, video play should be synchronous across all participants, i.e., the same video frame should show on all screens at the same time, as if the video is playing on one large display. Since a wireless network can introduce unpredictable communication latency, which could cause asynchronous play control, we need an approach to handle the network latency and synchronize video playback. Thirdly, online video resources have over billions of views per day and more than half views come from mobile devices [11]. If we can support playing online videos on multiple mobile displays, it will make the tiled displays more useful.

We present “MobileVideoTiles”, a mobile application that enables video playing on multiple mobile displays. Our goal is

to provide users an easy-to-use mobile virtual “big screen” to watch local and online videos. Compared to previous methods of multiple mobile displays, we concentrate on solving technical challenges of video playback, such as mobile peer-to-peer video streaming, play synchronization and video accessibility.

Related Work

Previous researchers envisioned co-located mobile devices as a social and spatial interaction platform for various configurations, such as shared information visualization and distributed control [4, 7]. In order to combine multiple mobile screens into one large display surface, a precondition is to detect the spatial relation between mobile devices. Different technical solutions have been proposed. Schmitz et al. introduced a photo calibration method based on visual marker detection [9]. Ohta and Tanaka proposed to use pinch gestures to link neighboring screens [5]. Li and Kobbelt applied front cameras of mobile devices to track common visual features for localization [3]. Radle et al. designed a desk lamp with an integrated RGB-D camera to track device position and hand movement [8]. These approaches mainly focused on auto/manual construction of mobile tiled displays. Their exemplary applications demonstrated the visualization of static images, maps and games. While video playing in the same context was not investigated. To support video playing on multiple mobile displays, Ohta and Tanaka developed “MovieTile” [6], where several issues were discovered. Firstly, play commands were triggered asynchronously on each device due to network latency, which brake the big screen experience. Secondly, a video can be played only when it was fully delivered, i.e., a large file needs a long waiting time. Thirdly, only local videos were supported, which limits the accessibility of video resources. In this project we concentrate on these technical issues and propose a synchronization protocol, a mobile peer-to-peer video streaming

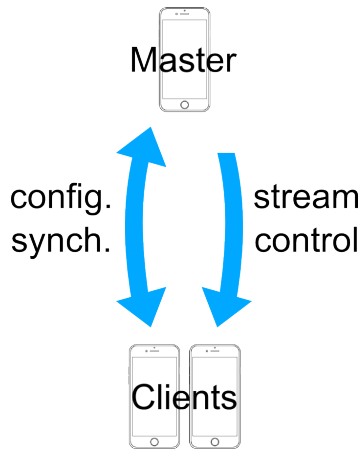


Figure 3: System overview.

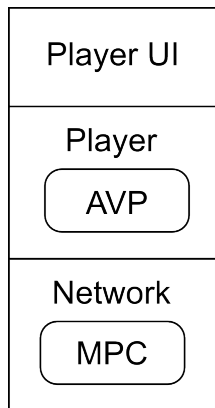


Figure 4: System structure. The network communication is based on MultiPeerConnectivity (MPC) framework. The player is based on AVPlayer (AVP) class of iOS.

method, and a flexible video resource selection.

System Description

We prototype “MobileVideoTiles” on iOS platform. A Master-Client structure is implemented, see Figure 3,4. The master device controls the virtual video player, including display configuration, resource selection, streaming, play commands, interaction and synchronization. While the client responds to corresponding instructions. They communicate via WiFi/BT network. In the following sections we will introduce these parts separately.

Network

We use Multipeer Connectivity (MPC) Framework provided by iOS, which supports reliable/unreliable data transmission in infrastructure Wi-Fi networks, peer-to-peer Wi-Fi, and Bluetooth networks. One advantage of using this framework is that it enables mobile peer-to-peer WiFi communication for nearby iOS devices with a lightning connector (e.g. since iPhone 5). It is unnecessary to connect to a common access point (AP) in order to communicate between devices, which simplifies the network interface for developers and makes the connection much easier, when no AP available. Devices prior to iPhone 5 still need to use an AP to work with the framework. Note that the network communication is not limited to MPC framework. Other frameworks supporting mobile peer-to-peer communication can also be used.

Display Configuration

We choose a regular side-by-side layout which reduces the visual gap between screens. Mobile tiled displays in an arbitrary layout are eye-catching when designing video signage [6], but in the scenario of watching videos, bezels of neighboring displays cause visual separation and confuse users [1, 2]. Therefore in “MobileVideoTiles” we use a regular side-by-side layout to reduce visual discontinuity and bezel oc-

clusion, see Figure 1. Additionally in order not to distort a video frame, we treat the area covered by bezels as a part of the whole virtual screen. From user point of view, it is like looking through a window. Since most videos are in 4:3 or 16:9 ratio, in the current version “MobileVideoTiles” supports up to 3 mobile displays which can better fit these ratios.

To construct the big virtual screen, 3 buttons are provided to indicate 3 available slots in the side-by-side layout: left (for client), middle (for master) and right (for client). To start configure, the user must indicate the master by pressing the master button on one device. Then invite a left/right client by pressing the corresponding button and select a candidate from a waiting list. Once the invitation is accepted by the client, the combination is complete. Note that alternative configuration approaches of mobile tiled displays [9, 3, 5, 8] can also be applied here to achieve arbitrary layouts if necessary.

Video Resources

“MobileVideoTiles” allows playing local and online video resources. On the master device the user first chooses a category of resources (e.g. My Video, Youtube, or Vimeo), and then he/she can browse videos in that category and select a target video to play, see Figure 2.

Local videos stored in the photo library of the master device are presented in a grid view. After selection, the video is compressed for the upcoming streaming process. The compression bitrate depends on the transmission rate of the network, so that for a faster network we can lower compression to have better visual quality. The compressed video will be stored locally for future access. Besides compression we may also crop a source video (according to the receiver’s viewport) to further reduce the transferred data size. However since the viewport could be changed (by zooming and panning) during playing, it would be more flexible to trans-

mit the whole video frame. In addition, the cropping operation consumes extra time for every client viewport, which increases the overall waiting time. Therefore the cropping process is skipped.

Online videos are presented to users via mobile web interfaces of video sharing companies, like Youtube and Vimeo (Figure 2). A user can browse/search online videos using the provided functions. Once the user clicks on a target video, the link to that video will be retrieved and sent to clients for buffering.

Streaming

Since the selected video file is stored on the master device, we have to share it to the clients for playing. In a pre-study we found that between mobile devices the data transmission rate (TCP) is about 1 - 2 MB/sec, which means it would take 50 - 100 seconds to deliver a 100 MBytes video from the master to a single client. For a configuration of three displays (1 master, 2 clients), the overall transmission duration would be doubled. If a video can only be played when it is fully transmitted, that would make users wait too long and feel bored. To reduce waiting time, we develop a mobile video streaming method that allows a video to start to play before the entire file has been transmitted. On the master, a video file (compressed in the previous step) is first split into multiple segments, where the number of segmentation depends on the file size. Then these segments will be transmitted to clients in sequence. When a client has received enough buffering, it will load the available video segments and inform the master it is ready to start. While the video is playing, the client keeps receiving further segments until the video file is fully delivered. In the meanwhile on the client we track the play progress and load new buffered segments when the previous load is running low.

Since online videos are stored remotely on video sharing

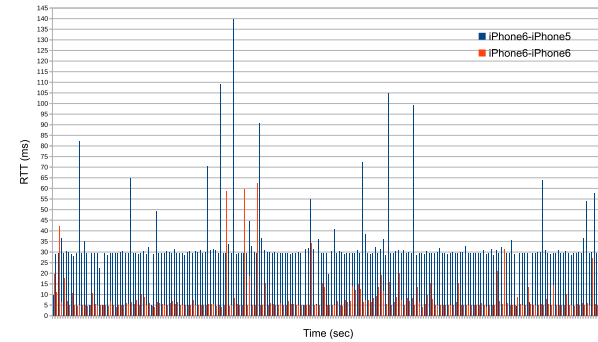


Figure 5: Round trip time profile between 2 mobile devices.

servers, given the link to a target video, every participating device requests the resource individually through the internet. The master will gather buffering status of all participants and coordinate video play. For example if one client is running low on buffer, the master will inform others to pause and wait.

Play and Synchronization

We implement a video player based on iOS AVPlayer class, which provides general play controls to a video. The player UI is reworked to suit the big virtual display scenario (Figure 1). For example, pinching and panning gestures are enabled to scale and move the video, so that a user can easily adjust the area of interest in a video.

To control video playback on the big virtual screen, a play command is broadcast to all clients via wireless network. Ideally the command should be triggered simultaneously on the master and clients, so that all displays play as one big screen. However due to unpredictable network latency, the command may reach the clients at different times. Therefore

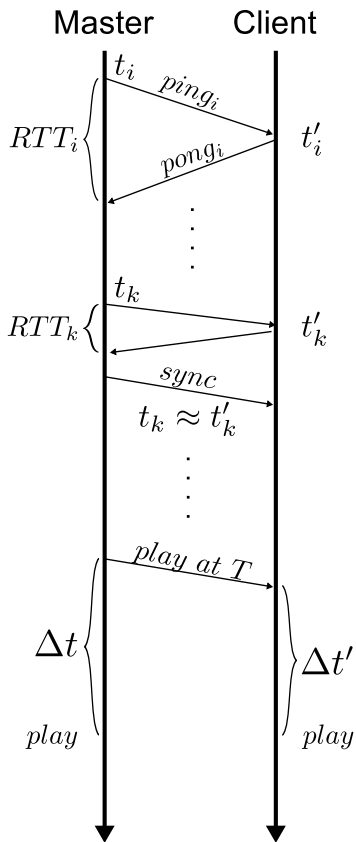


Figure 6: Synchronization.

there is a gap between the times at which the command is triggered on the master and the clients. If this gap is small ($\leq 20ms$), users may still perceive a big virtual screen. But when this gap increases, different video frames will show on neighboring screens, so the impression of having a big screen collapses.

To handle the issue of network latency we propose a synchronization protocol. Based on our observation (Figure 5), we found that despite the unpredictable occurrence of high latency there are always some periods in which the latency is low and stable. We can synchronize the “clock” of two devices in such periods. Figure 6 shows the procedure of the synchronization. The master is observing the round trip time RTT_i between itself and a client. Every ping (or pong) message local sending time is recorded on the master (or client) as t_i (or t'_i). If a very short round trip time is found ($RTT_k < 20ms$), it implies the difference between t_k and t'_k is likely less than $10ms$. We let the master (or client) take t_k (or t'_k) as a reference time point for synchronization. When a play command is triggered, the master will make an “appointment” with the client that the play should start at T after the reference time point. Then the master computes the remaining time till the appointment and schedule the play event. The client will also schedule the play once the appointment is received. Synchronization of different clients is independent from each other. Therefore we can synchronize additional clients at any time. Besides the play command, we can add extra synchronization check based on the reference time point during video playing. For example, at a certain time the video play should reach a certain progress.

System Performance and Discussion

To evaluate the performance of “MobileVideoTiles”, 3 iPhone6 were used. The user interface of the display configuration is simple and screens can be combined within a few sec-

onds. Video compressing performance of iPhone6 was also tested. For example it took 16 seconds to compress a sample video (126 sec, 1920x1080, 241 MB down to 1280x720, 59 MB). To transmit and play a local video file of 200 MB, without streaming it took 3 to 4 minutes until the video was playable. While using streaming the video was ready to play after about 10 seconds. To play online videos all devices must have access to internet, either via an access point or mobile cellular network. So far we have tested resources of Youtube, Vimeo, Trailer and several online TVs. Usually a video can be loaded on all devices within a few seconds. Occasionally there was a connection issue between one device and the video sharing server, which would need reloading the video link.

“MobileVideoTiles” has received several thousand downloads and very positive feedback from users (5 out of 5 stars). Many commented: “handy App”, “nice for home vids”, “cool effect to watch videos on a big screen”, “particularly cool is the zoom tool during the playback”, “I have watched 1 hour video from our last holiday with my parents.”. However, there are several aspects that could be improved. One concern is about wireless interference. When we accessed online videos via WiFi AP while enabling bluetooth, we had very slow download speed. But when we switched to the mobile cellular network, the download speed is back to normal. It is likely due to the interference between bluetooth and WiFi when using MPC framework, since on iPhones bluetooth and WiFi share one common antenna. In the current version we instruct users to turn off their bluetooth temporarily to achieve a better performance. Another concern is about local video streaming. During a video play when a new buffer is loaded, there is a video flickering. It is because we use existing methods of AVPlayer which are not designed for streaming purposes. A light-weight streaming server on mobile devices may solve that.

Acknowledgments

This research was a collaboration between RWTH Aachen University and appfreu.de GmbH. We would like to thank appfreu.de for their support during the development.

REFERENCES

1. Xiaojun Bi, Seok-Hyung Bae, and Ravin Balakrishnan. 2010. Effects of Interior Bezels of Tiled-monitor Large Displays on Visual Search, Tunnel Steering, and Target Selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. 65–74. DOI : <http://dx.doi.org/10.1145/1753326.1753337>
2. Jessica R. Cauchard, Markus Löchtfeld, Pourang Irani, Johannes Schoening, Antonio Krüger, Mike Fraser, and Sriram Subramanian. 2011. Visual Separation in Mobile Multi-display Environments. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. 451–460. DOI : <http://dx.doi.org/10.1145/2047196.2047256>
3. Ming Li and Leif Kobbelt. 2012. Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia (MUM '12)*. Article 24, 4 pages. DOI : <http://dx.doi.org/10.1145/2406367.2406397>
4. Andrés Lucero, Jaakko Keränen, and Tero Jokela. 2010. Social and Spatial Interactions: Shared Co-located Mobile Phone Use. In *CHI EA '10*. 3223–3228. DOI : <http://dx.doi.org/10.1145/1753846.1753962>
5. Takashi Ohta and Jun Tanaka. 2013. Using Pinching Gesture to Relate Applications Running on Discrete Touch-Screen Devices. *Int. J. Creat. Interaces Comput. Graph.* 4, 1 (Jan. 2013), 1–20. DOI : <http://dx.doi.org/10.4018/ijcicg.2013010101>
6. Takashi Ohta and Jun Tanaka. 2015. MovieTile: Interactively Adjustable Free Shape Multi-display of Mobile Devices. In *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications (SA '15)*. Article 18, 7 pages. DOI : <http://dx.doi.org/10.1145/2818427.2818436>
7. Tommaso Piazza, Morten Fjeld, Gonzalo Ramos, AsimEvren Yantac, and Shengdong Zhao. 2013. Holy Smartphones and Tablets, Batman!: Mobile Interaction's Dynamic Duo. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction (APCHI '13)*. 63–72. DOI : <http://dx.doi.org/10.1145/2525194.2525205>
8. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. 45–54. DOI : <http://dx.doi.org/10.1145/2669485.2669500>
9. Arne Schmitz, Ming Li, Volker Schönefeld, and Leif Kobbelt. 2010. Ad-Hoc Multi-Displays for Mobile Interactive Applications. In *EG 2010 - Areas Papers*. Norrköping, Sweden, 45–52. <http://diglib.eg.org/EG/DL/conf/EG2010/areas/045-052.pdf>
10. Statista. 2016. Share of mobile device owners worldwide from 2011 to 2016. (2016). <http://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/>.
11. Youtube. 2016. Statistics. (2016). <https://www.youtube.com/yt/press/statistics.html>.