Inter-Screen Interaction for Session Recognition and Transfer Based on Cloud Centric Media Network

Yichao Jin, Xiao Liu, Yonggang Wen and Jianfei Cai
School of Computer Engineering
Nanyang Technological University
{yjin3, n1107834k, ygwen, asjfcai}@ntu.edu.sg

***Abstract*—Recently, there is a growing trend that people tend to consume media over multi-screens simultaneously. This paper proposes an efficient and convenient inter-screen interaction approach based on our cloud centric media network, where all the ongoing sessions on the client side are always synchronized with the media cloud. This approach realizes the session recognition and transfer over different devices in a three-step manner. First, the users are required to use the mobile device camera to scan the main screen. Second, after the screen edge detection and image correction, users are allowed to choose one or more ongoing session on the corrected image via touch screen to request the transfer. Finally, the selected sessions are identified by the cloud, and those sessions are delivered to the mobile devices to complete the session transfer. The algorithms and strategies involved are discussed in detail. We also implement a testbed on top of a private cloud at NTU. The results prove our proposed method is robust and easy to use.**

I. Introduction

Nowadays, the ubiquitous penetration of high-speed Internet and mobile devices, is transforming the way people interact with the media in the daily life. People tend to consume media in front of screens by using more than one devices simultaneously. According to a study from Google [1], 90% media interactions are screen based. Moreover, 81% consumers use smartphone simultaneously with TV, 66% consumers use smartphone and laptop/PC at the same time, and another 66% consumers use laptop/PC when watching TV.

To share the sessions among different devices, we have built a multi-screen system based on our previous proposed Cloud-Centric Media Platform [2]–[4]. This platform leverages cloud computing technology to elastically and efficiently distribute media contents to consumers. We keep all the ongoing sessions of each user synchronized with a cloud clone on an event-driven scheme. By accessing to the same cloud, heterogeneous devices are able to obtain the same sessions.

This work focuses on an efficient and convenient inter-screen interaction approach for multimedia session recognition and transfer based on the multi-screen system. We use the mobile device camera as the interface to interact with other screens. Specifically, to recognize the sessions, users are required to use the camera to scan the main screen. On the mobile device, the offset, rotation and distortion on the captured image will be first corrected. An easy and fast edge detection approach will be then applied to get the main screen image. After that, users are allowed to choose one or more ongoing sessions on the image (i.e., the regions of interest) via the touch screen. The coordinates of the selected points on the touch screen are then mapped into those specific sessions by the cloud. Finally, the cloud delivers the selected sessions to the mobile devices to complete the session transfer.

We implement this inter-screen interaction scheme on top of a private cloud at NTU, to demonstrate the concept and evaluate the performance. In our deployed testbed, we find session recognition can be fulfilled successfully regardless of the different distances and rotations from the mobile device to the main screen, and the processing times under those different conditions are almost the same. It proves our proposed method is robust and easy to use.

Our proposed mechanism, when deployed, will benefit both content providers and consumers in media industry value chain. Specifically, content providers can leverage this technology to launch value-added services fast and easily. While content consumers can enjoy the media experience enriched with interactive, mobile and ubiquitous features.

The rest of the paper is organized as follows. In section II, we review some related works. In section III we introduce the system framework and the session transfer model. In section IV we discuss our session recognition and transfer method. In section V we present our implementation and the performance evaluation. In section VI we summarize the paper.

II. Related Work

There are a number of works on session transfer and mobility recently. In [5], the authors discussed a session transfer scheme for IPTV programs in the context of IP Multimedia Subsystem (IMS). S. Mate *et al.* [6] investigated device centric, network centric and hybrid approaches for session transfer. In [7], a lightweight framework for session mobility through the synchronized profits in the cloud and peer-to-peer nodes, was discussed. In [8], they set a separate session information server to enable the seamless session transfer among different terminals.

However, those approaches all focused on the back-end architecture and failed to come up with an efficient and convenient interaction to trigger the session mobility. While this article focuses on proposing a novel inter screen interaction for session recognition and transfer.

III. System Overview

In this section, we present the multi-screen system framework, and our session transfer model.

978-1-4673-5762-3/13/$31.00 ©2013 IEEE

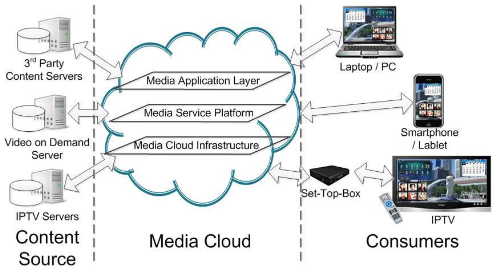


Fig. 1: Framework of cloud based multi-screen system

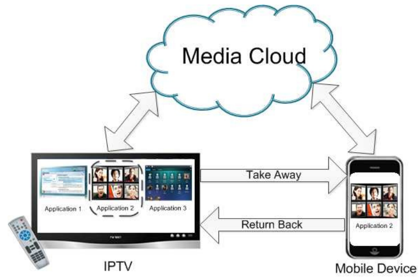


Fig. 2: The session migration model

A. System Framework

Figure 1 illustrates the system framework of multi-screen system based on the cloud centric media network. The system can be separated into three components, content sources, media cloud and content consumers.

The content sources can be from IPTV servers, online video on demand (VoD) server, 3rd party content servers and other content providers. They provide numerous contents ranged from TV programs, web applications to personal social cycles.

The media cloud contains three layers, including media cloud infrastructure layer, media service platform layer and the media application layer, from the bottom up. A resource pool powered by virtualization technology in geographically distributed data centers, constructs the media cloud infrastructure. Those resources can be utilized elastically to meet the changing demands. The media service platform offers various basic functions and interfaces on resource allocation, media service discovery and etc. The media application layer provides specific applications based on the underlying layers.

The content consumers can use different devices (e.g., TV, smart phones, and laptops) to access to the media cloud. Based on this framework, the media cloud is able to synchronize with all the active clients, and keep the information on all the ongoing sessions. As a result, it is possible to obtain the contents and services via any device.

B. Session Transfer Model

Figure 2 demonstrates the concept as well as the objective of session transfer. Specifically, when users are consuming media content on TV, where a number of sessions are ongoing, and their status are synchronized with the media cloud. The session transfer aims to transfer one or more sessions on TV screen (i.e., the main screen) to the mobile devices as required. After the transfer, the transferred sessions still continue their executions on the mobile devices by retrieving both the applications and their current status from the same media cloud. Similarly, the sessions can also be returned back to the previous main screen as requested.

From the user perspective, all they have to do to transfer the session from the main screen to their mobile devices is a convenient three-steps behavior. First, holding their mobile device in front of the main screen. The mobile application can automatically scan the main screen to capture a suitable screen-shot which is able to represent all the ongoing sessions. Second, selecting their interested sessions on the captured image via the touch screen. Finally, triggering the transfer by some specific gestures.

Our session transfer model differs from existing ones from mainly two aspects. First, most of the existing systems ”push” the sessions from one screen to another, while our approach ”pulls” the interested sessions from the main screen to your hand. Second, we enable a direct interaction between different screens by using the cameras, while the existing ones achieve this in a less user-friendly fashion (e.g., popping up a menu).

IV. Session Recognition and Transfer

In this section, we discuss our detailed approach to recognize sessions and complete the session transfer. Specifically, there are four steps involved in the session recognition and transfer, including screen edge detection, image correction, region of interest (ROI) extraction, and session transfer trigger.

A. Screen Edge Detection

Once the users use the mobile device camera to scan the main screen, the captured images inevitably contain some surrounding information. The target of screen edge detection is to extract the real display zone.

Our approach on screen edge detection relies on a set of assisted position markers. Inspired by the implementation of QR code [9], we place three position detection patterns on the left top, right top, and left bottom corner respectively, and one alignment marker on the right corner. Prior to applying the edge detection algorithm, we first use the thresholding method to turn the captured color image to a binary one, which is much easier to be processed, and the processing time can be shortened as well. Then we adopt a scan line approach to find the position detection patterns and the alignment marker on the binary image. Specifically, according to the feature of those markers, our algorithm detects them by three steps. First we try to find 7 horizontal concessive pixels following 1:1:3:1:1 for position detection patterns (1:1:1:1:1 for alignment marker) in alternative black and white style in sequence, to determine a list of potential candidates. Second, we verify those candidates by check their vertical patterns. If some of them do not follow the same style, we filter them out of the candidate list. Finally, if the candidate list contains less than three position detection patterns, then the screen edge detection has to be re-conducted by automatically capturing a new image. Otherwise, if there are more than three candidates, we will selected the most possible three ones by considering their expected coordinates.

B. Image Correction

Since the captured images usually suffer from distortion, rotation and offset, having the edge information is not enough to retrieve the image as shown in the main screen. An image correction algorithm is required to re-sharp the captured image.

Our algorithm is based on the pinhole camera model [10] and the camera matrix to correct the image by normalizing every coordinates. Specifically, in the pinhole camera model, we consider the mapping from the coordinates on the corrected image to the captured one, which can be given by [11],

$$\mathbf{K} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \mathbf{TL}, \quad (1)$$

where $\mathbf{K}$ refers the coordinate (x,y,1) on the desired image after correction, $\mathbf{L}$ is the coordinate (i,j,1) on the captured image, $\mathbf{T}$ is the camera matrix, standing for the projection from $\mathbf{L}$ to $\mathbf{K}$. In $\mathbf{T}$, a, b, c, d, e, f, g, h are all constant parameters, which define each specific projection. By substituting $\mathbf{L}$ with the captured coordinates of the position detection patterns and alignment marker, and $\mathbf{K}$ with the corresponding coordinates of those marker after correction, we can determine all the constant parameters.

By multiplying $\mathbf{T}^{-1}$ on both side, we have,

$$\mathbf{L} = \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \mathbf{T^{-1}K} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (2)$$

In this way, we can fill each single pixel (x,y,1) on the corrected image by finding the corresponding one (i,j,1) on the captured image. Notice that if the mapped coordinates i and j are non-integer, we adopt linear interpolation to calculate a approximate pixel based on the neighboring ones.

C. Extract Region of Interest

The extraction of region of interest (ROI) is a process to map the selected coordinate on the mobile device into the particular ongoing sessions. Specifically, the users are required to touch one or more interested sessions to be transferred on the corrected image via the touch screen. Then the selected coordinates of touch points will be sent to the media cloud to determine which sessions are to be transferred.

Due to the different screen size and resolution of different devices, the corrected image, which is displayed on the mobile device, may zoom in/out the origin image on the main screen with some offsets. As a result, we still have to re-size the captured image. Assuming the corrected image has been zoomed in for s times with an offset $\mathbf{t}(t_x, t_y)$, we could map any of its point $\mathbf{p}(x, y)$ to the according one $\mathbf{P_0}(x_0, y_0)$ on the main screen as,

$$\mathbf{P_0} = (\mathbf{p} - \mathbf{t})/s. \quad (3)$$

Fig. 3: Using tablet to conduct inter-screen interaction

As a result, the media cloud calculate the real coordinates P_0 according to the touch points p sent from the mobile devices, then find the sessions can be determined.

D. Session Transfer Trigger

Once the session transfer is ready, a trigger mechanism is needed to inform the media cloud to fulfill it.

The most intuitive and comfortable way is the gesture trigger. Our design takes advantage of the accelerometer in the mobile devices. Specifically, it can capture the gestures such as flipping in and out. Once such gestures are completed by the users, the session transfer is triggered. As a result, the media cloud will terminate the ongoing sessions on the main screen, and transfer those sessions to the mobile device.

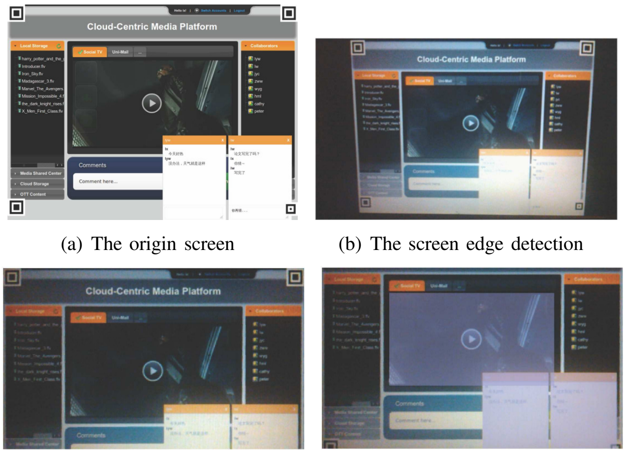
V. Implementations and Results

This section provides the details on system implementation methodology, and the performance evaluation.

A. Testbed Settings

We implement our multi-screen prototype as a web-based media service platform with cloud support. Specifically, we build the system on top of a private cloud at NTU, which consists of 10 physical servers. Each of those servers is equipped with a quad core Intel processor running at 1.80GHz, 2GB RAM, and 500GB storage space. Every server runs 2 virtual machines based on Xen hypervisor. For data storage, Hadoop Distributed File System (HDFS) is utilized as the content storage engine in the cloud, and Apache Cassandra is integrated as the distributed database management system. For the content presentation, the HTML pages are powered by Javascript and Python APIs to synchronize the real time session and fetch the information. For the video clip content, we implement it via flowplayer. For the session state presentation, we use JSON (Javascript Object Notation). We use FFmpeg in the backend to support different video formats.

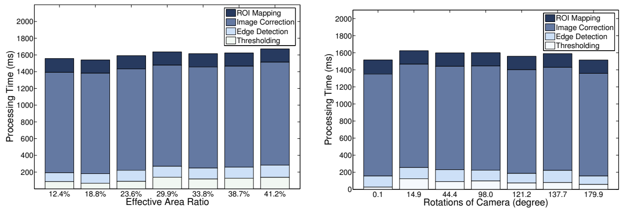
On the client side, we implement an Android based application for phones and tablets, and use a Linux-based mini PC as the set-top box for TV device. Specifically, we use Samsung P7500 Galaxy Tab as the mobile device in our prototype. The camera embedded has 2048x1536 pixels, and the CPU is dual-core 1GHz Cortex-A9. And the mini PC has dual-core 1.8GHz CPU and 2GB RAM. It connects to a Sony BRAVIA TV with 40” LCD display by using HDMI (High-Definition



(a) The origin screen (b) The screen edge detection

(c) The image after correction (d) The ROI extraction

Fig. 4: Different status for inter-screen interaction



(a) in different distances (b) in different rotations

Fig. 5: Processing time in each steps under different conditions

Multimedia Interface). Both mobile device and the set-top box communicate with the media cloud based on a Wifi network where the bandwidth capacity is 54Mbps. The experimental scenario is demonstrated by figure 3.

B. Experimental Results

We evaluate our system mainly from two aspects, including functional test and performance test.

1) Functional Test: Figure 4 illustrates the key status when recognizing and transferring sessions based on our proposed inter-screen interaction scheme. In this experiment, we assume both TV and tablet has logged in by using one same account.

Figure 4-(a) is the origin image as shown in the TV screen (i.e., the main screen). The user may hold the tablet in front of the main screen to do the screen edge detection. During this process, what the user see on the tablet is shown as figure 4-(b). After the edge has been detected, the image is fixed from rotations, offsets and distortions. Figure 4-(c) presents the image obtained by the mobile device after correction. Finally, figure 4-(d) shows the image when user finishes selecting the ROI. It proves our approach works well in each step.

2) Performance Test: We evaluate the performance of our approach in terms of the processing time under different conditions as shown in figure 5. Specifically, we test the time consumed by each step with different distances and rotations.

Figure 5-(a) presents the results in different distances. We use the effective area ratio, which represents the ratio between the real display zone and the size of the whole image, as the metric. The more the effective area ratio is, the closer from the mobile device to the main screen. It can be seen that with different distances, the processing time of each step as well as the overall process keeps roughly the same.

Figure 5-(b) presents the results in different rotations. We define the rotation as the angle between the screen of the mobile device and the horizon. A similar trend can be observed that with different rotations, our system costs almost the same time to achieve session transfer.

VI. Conclusion

This paper proposed a novel inter-screen interaction for multi-screen system based on cloud centric media network. This approach uses the mobile device camera as the interface to communicate with other screens. We discussed each sub-process during the session recognition and transfer, including screen edge detection, image correction, ROI extraction, and session transfer trigger. A proof-of-concept prototype has also been implemented on top of a private cloud at NTU. Both the functional test and performance evaluation were given. The results proved our proposed method is robust and easy to use.

Our future work will aim to further optimize the image correction process to shorten the processing time. In addition, we will also open our system to a large amount of users to gather more responses from them. Corresponding improvements will be also made according to those responses.

References

[1] Google, Ipsos, and Sterling, "The new multi-screen world: Understanding cross-platform consumer behavior," [Online], http://services.google.com/fh/files/misc/multiscreenworld_final.pdf, August 2012.

[2] Y. Wen, G. Shi, and G. Wang, "Designing an inter-cloud messaging protocol for content distribution as a service (codaas) over future internet," in *Proceedings of the 6th International Conference on Future Internet Technologies*, 2011, pp. 91–93.

[3] Y. Jin, Y. Wen, G. Shi, G. Wang, and A. Vasilakos, "Codaas:an experimental cloud-centric content delivery platform for user-generated contents," in *2012 IEEE International Conference on Computing, Networking and Communication (ICNC)*, January 2012, pp. 934–938.

[4] Y. Jin, Y. Li, X. Liu, G. Zhang, and Y. Wen, "Content delivery as a service for cloud-centric multimedia social networks," *IEEE COMSOC MMTC E-Letter*, vol. 7, no. 5, pp. 4–6, May 2012.

[5] I. Más, V. Berggren, R. Jana, J. Murray, and C. Rice, "Iptv session mobility," in *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, 2008, pp. 903–909.

[6] S. Mate, U. Chandra, and I. Curcio, "Movable-multimedia: session mobility in ubiquitous computing ecosystem," in *Proceedings of the 5th international conference on Mobile and ubiquitous multimedia*, 2006, pp. 8–12.

[7] D. Johansson, "Session mobility in multimedia services enabled by the cloud and peer-to-peer paradigms," in *2011 IEEE 36th Conference on Local Computer Networks (LCN)*, 2011, pp. 770–776.

[8] D. Pichon, J. Bonnin, P. Seite, and G. Rieublandou, "Inter-terminal multimedia session mobility in next generation networks to enhance iptv," in *2010 7th IEEE Consumer Communications and Networking Conference (CCNC)*, 2010, pp. 1–5.

[9] Wikipedia, "Qr code," [Online], http://en.wikipedia.org/wiki/QR_code.

[10] Wikipedia, "Pinhole camera model," [Online], http://en.wikipedia.org/wiki/Pinhole_camera_model.

[11] C. Chu, D. Yang, Y. Pan, and M. Chen, "Stabilization and extraction of 2d barcodes for camera phones," *Multimedia systems*, vol. 17, no. 2, pp. 113–133, 2011.