Design and Implementation of Real-time Multi-user Multi-screen Wireless Projection System Based on ARM Embedded System

Zhaohua Long and Jiangnan Chen

ABSTRACT

Aiming at the shortage of existing real-time screen sharing system, a real-time multi-user multi-screen wireless projection system is proposed in this paper. On the basis of the proposed system, a weight-based rendering method combined with response time and the amount of data change is proposed. By implementing the system on hardware platform, the results show that the system can well adapt to multiple users, and the weight-based rendering method can balance the response time of the server to the client and optimize the real-time customer service system.

INTRODUCTION

Projection is one of the most important means of screen sharing, play a essential role in the meeting, entertainment, education, etc. The existing projection methods are mainly divided into three types: First, screen sharing through video data lines such as VGA, HDMI[1]. Second, screen sharing based on WIFI-DISPLAY standard. Third, Screen sharing based on wireless projection card[2]. However, There are some disadvantages: 1, limited by space, such as wiring[3]; 2, only be one to one projection.

In view of the above problems, a real-time multi-user multi-screen wireless projection system is designed. The system can decode and render the image compressed data stream sent by the client, and adaptively divide the screen according to the number of access users, and respond to the data of multiple clients at the same time. Taking into account the limitations of embedded CPU computing ability[4], a weight-based data rendering method is proposed. Experimental results show that the system can not only avoid the shortcomings of the traditional projection, but also respond to a number of projection users.

Zhaohua LONG¹, Jiangnan CHEN²
¹College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China
Corresponding author:asd44555779@163.com
SYSTEM ARCHITECTURE

The system is divided into two parts: client and server. After client and server interacts via the control message, the client is responsible for collecting image data, encoding and then using the wireless network for transmission[5], the server is responsible for decoding and rendering after receives data, the system architecture is shown in figure 1:

![System Architecture Diagram](image)

Figure 1. System architecture.

Server renders the data of clients according to the following steps:
1) Client send projection-request message;
2) Divide memory address, calculate the first address of client in the display memory address;
3) Receive the data of client, insert data into the rendering buffer queue according to its weight;
4) Extract the maximum value of the data from the render queue;
5) Receive the projection-quit message of client, re-divide the memory address, calculate the first address in the display memory space of other client.

ADAPTIVE MULTI-SCREEN ALGORITHM

Server would assign a id for each access client. According to the id of each client, the server calculates the corresponding region in the display space. The algorithm is as follows:
1 Calculation of order: In this step, each user monopolizes a regional system display space, server scale the customer area equally. So take the concept of order. Order represents the number in the horizontal and vertical direction of the system display area, as shown in figure 2:

![Multi-screen Sketch](image)

Figure 2. Sketch of muti-screen.

So the total number of the divided display areas is $1, 4, 9, 16, 25, 36, 47, \ldots, \text{order}^2$;
2 Calculating the customer's video memory address;
3 Rendering of client area.
Finally, the correspondent relationship between the screen data and memory pixels is shown in figure 3:

**Figure 3.** The correspondence between the data of the screen and the data of frame buffer.

In figure 3, `paddr_(i){1,2,3,4}` represents the first memory address[6] of each calculated customer.

**WEIGHT-BASED DATA RENDERING METHOD**

In the initial design, the server stored data sent by user in the buffer queue[7], and then the decoder take out the data from the buffer queue to decode, after that the data would be rendered by the renderer[8], as shown in figure 4:

**Figure 4.** schematic diagram of buffer queue.

The buffer queue is a queue of FIFO. Color space transform after decoding needs to spend a lot of CPU resources. However, for a real-time system, it is intolerable that the client's data cannot be timely responded because the buffer queue data cannot be processed in time as the decoding time is too long. In view of this situation, this paper proposed a data rendering method based on weight `w` (Weight-based data rendering method), the method makes the renderer render according to the weight of the data, so as to achieve the real-time response to the user's data in real-time system.

In this algorithm, the weight coefficient `w` is mainly affected by two factors, the time factor `T` and the mean square error[9] (MSE). The former represents the time difference that the client has not received a response, the later represents the size of the client's data change. The calculation method of time factor `T` is shown in formula 1:

\[ T = (a \times (t_2 - t_1) + b) \times (t_2 - t_1) \]  

(1)

In Formula 1, `tl` represents the time at which the client has recently received a response, and `t2` represents the current time. `a` and `b` are time constants. The formula 1 derivation can be:
\[
\frac{\partial T}{\partial (t_2 - t_1)} = 2a(t_2 - t_1) + b
\]  
(2)

According to formula 2, as the client cannot get the response in time, the time factor is greater. In order to investigate the difference of the two images, we introduce the concept of mean square error (MSE)[10]. The calculation method is shown in formula 3:

\[
MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (X(i, j) - Y(i, j))^2
\]  
(3)

In formula 3, \(H\) and \(W\) represent the height and width of the image. \(X(i, j), Y(i, j)\) represents pixel value of two images at the point \((i, j)\). Calculate the mean square error of the image at the client's response time and the image at current time. The Weight-based data rendering method \(w\) is shown in formula 4:

\[
w = (MSE + 1)^T
\]  
(4)

In formula 4, calculating the weight coefficient \(w\) in \(T\) to 0 and \(MSE\) to 0 respectively, The results are shown as below:

\[
\lim_{T \to 0} w = 1
\]  
(5)

\[
\lim_{MSE \to 0} w = 1
\]  
(6)

The results correspond to two cases:

Case one: If the time factor \(T\) tends to 0, it can be considered that the client received a response in a very short period of time, influenced by the human visual system, visual changes of eyes on a very short period of time is not sensitive. Thus the weight coefficient \(w\) is the minimum value of 1, meaning that it is not necessary to response to a client in which image changes on a very short period of time.

Case two: If the \(MSE\) tends to 0, it can be considered that the change of the client in a certain time is small[11]. In this case the weight coefficient \(w\) is the minimum value of 1, the server does not have to respond to the client.

**EXPERIMENTAL RESULT**

In order to verify the effect of real-time multi-user multi-screen projection system, the system is implemented on the embedded platform. Multi-screen results shown in Figure 5:

![Figure 5. muti-screen.](image-url)
The result of figure 5 shows that the system is well suited for multiple screen access to multiple access clients. Despite the limitations of the hardware such as CPU computing power, memory size, network bandwidth and screen pixel value, the system can support unlimited access to all clients.

In order to measure the real-time performance of the system with weight-based rendering method, the data of the client and server are collected at the same time, and the $MSE$ is calculated. The results are shown in figure 6:

![Figure 6. Client-server mean square error.](image)

Figure 6 shows that, in the case of a single screen, the image mean square error between the client and the server before and after using the weight-based rendering method is small. This is because in the case of a single screen, the server load is light, the server can respond to the client's data in a timely manner.

In the case of four sub screens, with the increasing of the number of sub screens and the amount of client data, the mean square error of the image before and after using the weight-based rendering method is increased. This is because when the server load increases, CPU computing power cannot be fully real-time response to multiple clients. However, the mean square error of the system is lower than before using weight-based rendering method. This is because the system selectively renders the image data in the case of insufficient CPU computing resources to ensure the real-time performance of each client.

In the case of nine sub screens, the real-time performance of the system decreased sharply, when the server load increases, the mean square error increases significantly. But in the application of the weight-based rendering method, the real-time performance of the system is better than before using weight-based rendering method.

**CONCLUSIONS**

The experimental results show that the proposed system can respond well to multiple access clients at the same time, and adaptive divide screen according to the number of clients. This is a great potential for applications in meetings, monitoring, and entertainments. The weight-based rendering method is used to combine the response time of the client with the magnitude of the change of the data, it has a better optimization for the system which requires both high real-time performance and low data response integrity.

**REFERENCES**

[1] Y. Zhang, J. Lv. ARM9 and embedded Linux based Internet projector system design [J]. Microcomputer & Its Applications, 2011,10:15-17.
[2] P. Chi, C. Su, G. Xie, D. Liu. A design of a multi-user wireless projection system on Android [J]. Computer Engineering and Science, 2015, 11: 2148-2153.

[3] X. Geng, W. Du, N. Ge. Research on Embedded Video Frequency Processing and Wireless Transmission System [J]. Fire Control and Command Control, 2016, 01: 136-138+142.

[4] J. Wei, Y. Deng, Y. Gao, J. Yang, J. Song. Design and implementation of embedded video acquisition system based on ARM [J]. Modern Electronics Technique, 2016, 03: 30-34.

[5] P. Wu, Z. Long, L. Zhang. Research and design of wireless projector gateway on Wi-Fi Direct [J]. Computer Engineering and Design, 2013, 06: 2223-2227.

[6] C. Zhang, X. Yang, J. Wang. Design of LCD driver based on Frame Buffer [J]. Computer Engineering and Design, 2009, 23: 5372-5375.

[7] Y. Zhao, L. Zeng, S. Wen. System Design for LCD Splitter in Subway Train Based on Ethernet [J]. Video Engineering, 2009, 06: 102-104+113.

[8] Y. Hu, Z. Hou. Design and Implementation of Multi-display System and Its Simulation [J]. Computer Engineering, 2006, 22: 252-254.

[9] L. Sha, G. Wang, A. Zhao, X. Ling, K. Tang. H. 264 coding PSNR estimation method based on quantized DCT coefficients [J]. Journal of Tsinghua University (Science and Technology), 2008, 04: 510-513.

[10] F. Chen, L. Ma, S. Zhou, X. Zhu. The Research & Development of Video Image Capturing Based on Embedded System [J]. Microcomputer Information, 2005, 25:12-13+70.

[11] J. Wu, J. Song, H. Liu. Design and Implementation of Embedded Video Capture System Based on ADV7180 and S3C6410 [J]. Computer Engineering and Science, 2012, 12: 115-119.