2011 International Conference on Information Management and Engineering (ICIME 2011)
IPCSIT vol. 52 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V52.76

Study on the Key Technology of the Mobile Video Display System in the Client

Xiaoyu Ma [+] and Jixun Gao

Department of Computer Science and Engineering, Henan Institute Of Engineering, Zhengzhou, China

Abstract. This paper first describes the characteristics of mobile devices and the popular operating system of Windows CE, Then according to the problems about the resources of mobile devices, improve the key technology of mobile video display system in the client which include the improvement of multi-thread technology, display efficiency of the video image and the transform algorithm of the image resolution. The application of improved technology, not only can make the modules of receiving, decoding and displaying run in parallel, improve the utilization of CPU resources, but also can expedite the processing speed of video, improve the display efficiency of the video image.

Keywords: mobile video, transform, display efficiency, algorithm.

1. Introduction

The common mobile devices include common mobile, Smart phone, PDA, etc. The greatest features are their movability and convenience, can communicate with each other by wireless way anytime and anywhere, with the development of the science and technology, the hardware configuration and function of mobile devices are more and more powerful, has provided a guarantee of the multimedia video communication, for example, the decoding and playing of the sounds and images. Although the hardware configuration of the mobile devices is more and more higher, its network resources and system resources are still relatively limited, and this greatly affects the quality of the mobile video images, so the study of this paper is mainly on the key technology of the mobile video display system in the client.

2. Analysis of the Operating System of the Mobile Devices

The display system of mobile video in the client includes three parts: receiving, decoding and displaying, and these function blocks are all running on the operating system of mobile devices. The operating system of the mobile devices, as the key platform for connecting hardware, hosting applications, plays an important role, especially for the smart phones which are more and more popular and powerful. At present, the operating system of the mobile devices are mainly Symbian, Windows CE, Palm, and Linux. In the system tested in this paper, considering the efficiency of development, the client platform chooses PDA which uses the operating system of Windows CE, because many API functions of Windows CE are the subset of Windows API functions [1], the following is the further introduction to the operating system of Windows CE.

Windows CE is composed of many discrete modules, each module provides specific functionality. Part of these modules are divided into components. The components make the Windows CE become very compact (only requires less than 200 KB RAM), it only takes up the minimum resources of ROM, RAM and other hardware which are required to run the devices. Windows CE contains the four modules which provide the

[+] Corresponding author. Tel.: + 0371-68723925.
E-mail address: hnzzmxy@yahoo.com.cn.

most critical function of the operating system: kernel module, object storage module, graphics, windows and Events Subsystem (GWES) module and communication module, according to the difference of application to increase or decrease other modules[2].

3. The Problems About the Resources of Mobile Devices

Although the hardware configuration of the mobile devices is more and more higher, its network resources and system resources are still relatively limited, and the network resources mainly refer to the network bandwidth, and the system resources mainly refer to the power resources of the devices, CPU resources, memory resources and display screen resources.

In the case of the limited resources of mobile devices, in order to make the streaming data media transfer fluent, we must resolve the system resource allocation, because the network receiving, MPEG-4 decoding and the image displaying all need a lot of CPU and memory resources, in order to display the better image, need to transform the resolution, which also requires a lot of CPU resources.

4. The Improvement of Multi-Thread Technology

The multi-thread technology can make the modules of receiving, decoding and displaying run in parallel, improve the utilization of CPU resources

4.1. The definition and priority of the multiple threads

In Windows system, the thread is a WIN32 object used to run code, it has a different meanings from the process which is an application loaded into memory by WIN32. When WIN32 creates a new process, it only creates an image file, and copy the content of the executable files or DLL files, ActiveX files into memory [3]. The process is not active, the actual operation is carried out by the thread, the thread is the worker, the operation code belongs to the thread, and the process provides the address space and the required data for running.

WIN32 is a preemptive and multitask operating system, which in turn assigns time segment to each thread, make the thread run in the assigned time segment. Each thread of system has a Context record which contains the data structure of the CPU register status information while the thread runs. When the thread is created, system will assign a priority to it. System will schedule all the threads in the process in order of priority, only when there is no high-priority thread to perform, the low-priority thread is possible to be scheduled and performed. Windows CE supports eight different priorities, from 0 to 7, 0 represents the most highest priority, which is defined in the header file Winbasw.h.

4.2. Multi-thread operation

Multi-thread operation includes: thread creating, thread terminating, thread priority setting , thread suspending and thread resuming. When the system begins to perform a thread, another thread must call ResumeThread and pass it to the thread handle which is returned when it calls CreateThread.

4.3. The improvement of multi-thread technology

After the client program starts, system first creates a network connecting thread, requests the server to send data, after the client received the initial data, will create three separate threads: NetReceiveThread, DecodeThread, DisplayThread, That is, the data receiving thread, decoding thread and displaying thread. The creating process of the thread is shown as Fig. 1:

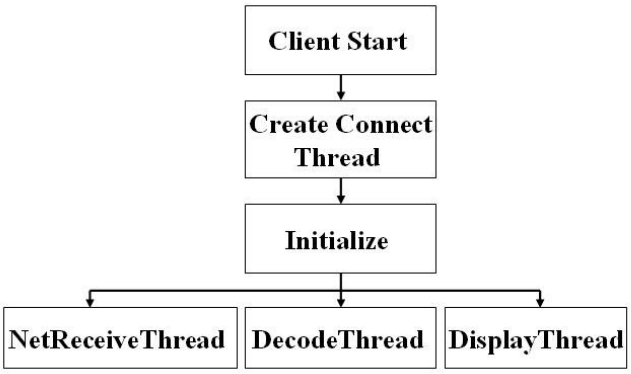

Fig. 1: The creating process of the thread.

In multi-thread system, the synchronization between threads is very important, when multiple threads access shared resources, if a conflict occurses, usually will produce incorrect results, for example, when the network receiving thread is writing data to the buffer , if the decoding thread reads the buffer data at this time, then error will occur, the same problem also exists between the decoding thread and displaying thread.

To avoid such problems, we can perform the operation of locking and unlocking, when the data receiving thread and decoding thread are writing data to the buffer, we can use the function Long interlockedExchange() and PVOID interlockedExchangePointer() provided by Win32 to perform the locking operation, and perform the unlocking operation after the writing operation is completed, allows other threads to perform the reading operation.

5. The Improvement of the Video Image's Display Efficiency

After the video data which the network received is decoded by MPEG-4 decoder, The output images' format is YUV, in the desktop Windows system, there are many ways to directly display YUV images, like the DirectDraw whose performance is best [4], it can directly operate the graphics hardware, but the terminal of mobile devices does not support the technology of DirectDraw. In order to display in the terminal of the mobile devices, we can select the Graphics Device Interface (GDI), GDI only supports the display of RGB images, so, first of all , we need to transform the images' format.

5.1. Color space conversion from YUV to RGB

In video processing technology, YUV color space is widely used from the video compression to the video transmission, because it has many advantages. Y is the brightness value of the YUV color space, U (Cb) and V (Cr) are the chroma values of the YUV color space. RGB is a representation of the three primary colors, R for red, G for green, B for blue, The conversion formula is as the following:

$$\begin{gathered} Y=0.299\times R+0.587\times G+0.114\times B \\ U = Cb =(B-Y)\times 127/226 \\ V = Cr =(R-Y)\times 127/179 \end{gathered} \tag{1}$$

The color space conversion algorithm of this formula uses the floating-point operation, but the floating-point operation consumes much more CPU cycles, especially for the mobile devices, consumes further more CPU resources. To speed up the video processing, we use the integer multiplication and right dislocation instead of floating-point multiplication and division, which effectively shortens the conversion time, the optimizing formula is as the following:

$$\begin{gathered} Y=((R\times 313524)>>20)+((G\times 615514)>>20) \\ \times((B\times 119538)>>20) \\ V=Cr=((R-Y)\times 743962))>>20 \\ U=Cb=((B-Y)\times 589244))>>20 \end{gathered} \tag{2}$$

5.2. GDI display images

The main task of GDI is responsible for information exchange between system and the drawing program, deals with all the graphics output of Windows programs. GDI provides us with a lot of functions which can be easily used to draw graphics, bitmap and text.

In Windows operating system, most of applications with graphical interfaces are inseparable from GDI, we can easily use the functions provided by GDI to output the graphics and text on the screen, printer, and other output devices The emergence of GDI makes programmers do not need to worry about hardware and device drivers, can easily convert the program's output to the hardware devices' output, and achieve the application developers' isolation from hardware, greatly facilitate the development.

However, the display efficiency of using GDI to display RGB images directly is not high, especially in the case of much higher frame rate, will generate delay [5]. In order to improve display efficiency, we use double buffering technology.

The basic principle of the Double buffering technology [6] is: first, paves a virtual canvas in memory, and then first displays the graphics which is need to be displayed in this "virtual canvas", last, paints the entire canvas to the real form. Because all of the individual graphics rendering is not really calling the display system to "paint", so do not take the overhead of display system, which greatly improves the efficiency of the drawing.

The specific steps to achieve double buffering:

Step1 Create the compatible memory DC for the screen: CreateCompatibleDC( );

Step2 Create the bitmap: CreateCompatibleBitmap();

Step3 Select the bitmap into the device context: SelectObject();

Step4 Copy the graphic which is well drawn to the screen: BitBlt().

The implementing process of the double-buffering technology is shown in Fig. 2

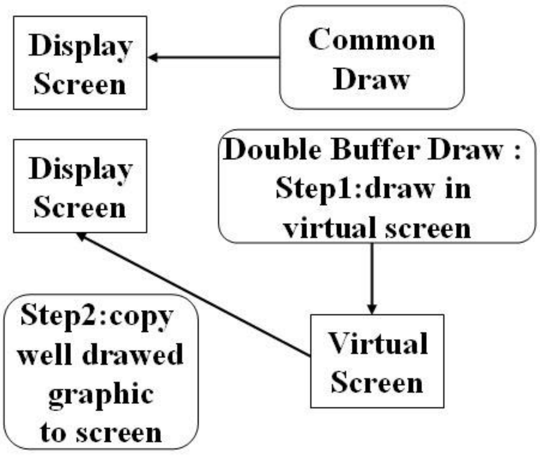
Display Screen
Common Draw
Display Screen
Double Buffer Draw : Step1:draw in virtual screen
Step2:copy well drawed graphic to screen
Virtual Screen

Fig. 2: The implementing process of the double-buffering technology.

6. The Improvement of the Image Resolution's Transform Algorithm

The transform of image resolution refers to change the image resolution by up sampling or down sampling.

In the client of the mobile video devices, the general image format is QCIF, the resolution size is 176 × 144, mainly to save the bandwidth, transfers fluent video images over the wireless network as much as possible. However, in order to be fit for the transform of the client devices' display and the image's size, this requires the transform of image resolution, which requires a lot of floating-point operation.

The transform of the image resolution must takeup the CPU time as little as possible, otherwise it will affect the local preview and real-time decoding, so the transform algorithm of the image resolution should be simple and efficient. The transform of image resolution discussed in this paper mainly refers to the transform from high resolution to low resolution, what's more, the multiple of transform is integral multiple.

According to the characteristics of the image transform, the transform algorithm can use the simple linear interpolation, the target image obtains the new resolution by taking the the source image's values of even lines' even points (or odd points) or the odd lines' even points (or odd points) ,as is shown in Fig. 3.

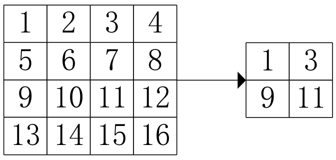

Fig. 3: Diagram of the image resolution transform.

In order to improve the speed of transform, we need to optimize the algorithm, The Intel's MMX instructions can be used to achieve optimization on the PDA. The MMX technology is essentially a "single instructions stream, multiple data stream" data processing (SIMD), allows the CPU deals with the data of 2, 4 or even 8 bytes in parallel at the same time, which effectively improves the CPU’s processing speed for video, audio, and other multimedia[7]. MMX has the conversion instruction, which can complete the processing of the data into groups and packets (such as PACKUSWB instruction), using these instructions can complete the transform of the image resolution.

The function of the PACKUSW instruction shows that it is more suitable for the transform algorithm of image resolution[8]. the transform algorithm is divided into three steps:

Step1 Reading data

Read the 16 bytes data of the source image, and store them separately in two 64-bit registers mm0, mm1.

Step2 Shielding the invalid data

For each register, use {0xff, 0, 0xff, 0,0 xff, 0,0 xff, 0} as a mask to perform "and" operation, make the high 8 bits in 4 words of the register mm0 and mm1 be zero.

Step3 Synthesizing data

Using the group instruction PACKUSWB on register mm0, mm1 to perform the group operation, the result is stored in mm0 register.

Through the above operations, we can obtain the 8 bytes data of the target image each time, repeat the steps 1-3, can complete the transform of the image resolution. The Schematic diagram of the algorithm is shown as Fig. 4.

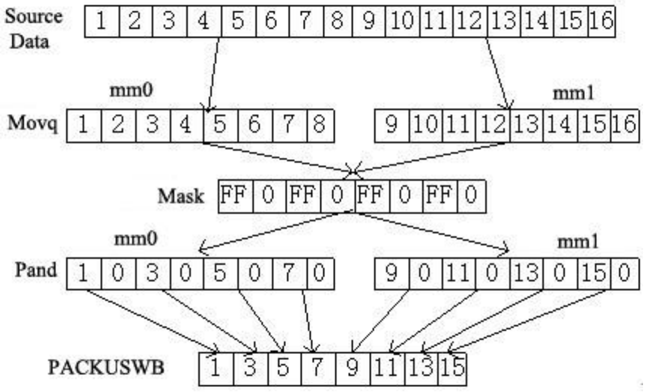

Fig. 4: The transform Diagram of the image resolution by MMX instruction.

The speed of transform algorithm is greatly improved by optimizing the transform algorithm through using MMX instructions according to the characteristics of transform algorithm (processing speed is several times than the normal assembly instructions).

7. Concluding Remarks

This paper is based on the the Windows CE platform, proposes some technical improvements according to the the specific problems encountered in development, provides a guarantee for video streaming to run fluent on mobile devices. The main technical point in this paper is the flexible application of multi-threading technology, better improves the scheduling efficiency of the CPU resources and memory resources, and then puts forward some technical improvements to the display of the specific video image, expedites the processing speed of video, and improves the display efficiency of the video image.

8. References

[1] Zhang ZhanJun, Han ChengDe, Yang XueLiang. The guaranteeof QoS for wireless multimedia streams based on adaptive session. In : Proceedings of IEEE International Conference on Personal Wireless Communications , India , pp.283-287, Feb, 2010.

[2] Kimata H, Tomita Y, Ibaraki H, et al. Concealment of damaged area of mobile video communications. Proc. workshop Audio Visual Services over Packet Networks. Scotland, U. K, pp. 17-21, Sep, 2010.

[3] Rhee I, Joshi S R. Error recovenry for interactive video transmission over the Internet,IEEE Journal on selected Areas in Communications, pp.1033-1048, Oct,2010.

[4] TALLURI, R..'Error-resilient video coding in the ISO MPEG-4 standard ', IEEE Commun. Mag., pp. 112–119, Jun, 2010.

[5] Chen, M. G.,and Weng, R. M.:Error concealment of lost motion vectors with overlapped motion compensation', IEEE Yrans. Circuits Syst. Video Technol., pp. 560-563, Jul, 2010.

[6] Salama, P., Shroff, N. B., Coyle, E. J., and Delp. E. J., :'Error concealment technology for encoded streams'. Proc. IEEE. Int. Conf. Image Process. (ICIP), pp. 9-22, Jul, 2010.

[7] A. K. Katsaggelos, F. Ishtiag, L. P. Kondi, M. C. Hong, M. Banham, and J.Brailean, Error resilience and concealment in video coding, in Proc. 1998 European Signal Processing Conf, pp.221-228, Aug, 2010.

[8] Sofia Tsekeridou and Ioannis Pitas, MPEG-2 error concealment based on block matching prinples, IEEE Trans. Circuits system Video technology, pp. 647-658, Dec, 2010.