J Med Syst (2014) 38:44
DOI 10.1007/s10916-014-0044-y

A Medical Application Integrating Remote 3D Visualization Tools to Access Picture Archiving and Communication System on Mobile Devices

Longjun He · Xing Ming · Qian Liu

Received: 15 December 2013 / Accepted: 18 March 2014 / Published online: 5 April 2014

Abstract With computing capability and display size growing, the mobile device has been used as a tool to help clinicians view patient information and medical images anywhere and anytime. However, for direct interactive 3D visualization, which plays an important role in radiological diagnosis, the mobile device cannot provide a satisfactory quality of experience for radiologists. This paper developed a medical system that can get medical images from the picture archiving and communication system on the mobile device over the wireless network. In the proposed application, the mobile device got patient information and medical images through a proxy server connecting to the PACS server. Meanwhile, the proxy server integrated a range of 3D visualization techniques, including maximum intensity projection, multi-planar reconstruction and direct volume rendering, to providing shape, brightness, depth and location information generated from the original sectional images for radiologists. Furthermore, an algorithm that changes remote render parameters automatically to adapt to the network status was employed to improve the quality of experience. Finally, performance issues regarding the remote 3D visualization of the medical images over the wireless network of the proposed application were also discussed. The results demonstrated that this proposed medical application could provide a smooth interactive experience in the WLAN and 3G networks.

This article is part of the Topical Collection on *Mobile Systems*

L. He · X. Ming · Q. Liu
Britton Chance Center for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, Hubei, China

L. He · X. Ming · Q. Liu (✉)
Key Laboratory of Biomedical Photonics of Ministry of Education, Huazhong University of Science and Technology, Wuhan 430074, China
e-mail: qianliu@mail.hust.edu.cn

Keywords Medical images · PACS · Remote visualization · Mobile health

Introduction

Picture archiving and communication system (PACS) is an integrated system to distribute and archive medical images acquired by different imaging modalities, such as computed tomography (CT), magnetic resonance (MR), computed radiography (CR) and ultrasonography (US), and stored in the digital imaging and communications in medicine (DICOM) format. During the past decades, PACS strongly changed the radiographers' work practice, skills and technology and improved the diagnostic efficiency [1]. And the PACS workstation integrated with advanced 3D visualization techniques can provide the shape, location and depth information of the tissue to the radiographers, which has significance for vascular reconstruction and a wide range of other clinical settings, including traumatic injury, tumor mapping bronchial visualization and urinary tract imaging [2]. The 3D visualization techniques refer to a range of specific graphics processing techniques, so different 3D visualization techniques will show different information. With maximum intensity projection (MIP), only one layer of the brightest or darkest voxels parallel to the line of sight is used for display. Thus, from a given viewing direction, it is possible to get some information about the brightness of an object, which is a distinctive feature of differentiation of calcification or thrombosis within a blood vessel. Multi-planar reconstruction (MPR) has the advantage that no information is lost. When an isotropic volume is used, axial, coronal, sagittal and all kinds of curved planes can be reconstructed with the same quality as the original images. Direct volume rendering (VR) make use of all information contained in a volume. Theoretically, direct volume rendering is a lossless approach, and it can provide shape, brightness, depth and location information of the tissues.

At present, radiologists mainly view medical images and the 3D visualization results on the PACS workstations located at fixed locations and restricted by wired local networks in the hospital. A more flexible access to PACS which is not bounded by the time and space—"mobile access"—would be attractive and practically very useful. In recent years, computing capability of mobile devices increases quickly and advanced compression algorithms are introduced to reduce the size of medical images [3, 4], so that it is possible to show DICOM images on mobile devices now. In general, there are two ways to view images on mobile devices, on the web site [5, 6] and in the mobile application [7–9]. The web access mode is easy to deploy and can be accessed on different platforms, such as iPhone OS, android, or Windows phone, but it is not friendly to interact, especially for complex 3D interactive operation. Therefore, it is more popular to show medical images in mobile application [10], which can be customized to user interaction.

At now the 3D visualization on mobile devices falls in two categories: local rendering and remote rendering. According to [11–13], there are almost no mobile devices on the marker that support 3D, textures, so they provided local volume rendering systems take the 2D texture-based volume rendering algorithm to realize the function of 3D visualization. It is said that it need to use a large number of 2D textures to store all the voxel data instead of a 3D texture. Another drawback is that the visualization performance of 2D texture-based volume rendering algorithm is related to the line of sight, and the quality of visualization image will obviously reduce when the angle between the line of sight and the axis is close to 45°. The most critical issue is that advancements in multi-detector CT and the advent of high-resolution isotropic MRI have increased the size of the volumetric data, so more powerful computing capability and bigger memory size of the mobile device is required to meet the demand of the local rendering. Instead, remote rendering uses a remote server to perform the computation, and transmits rendered views instead of row data to the client. In [14, 15], a non-real-time remote rendering system was introduced. The system generated 3D visualization images of medical images on the workstation in advance and then stored them in DICOM format on PACS. Mobile client can get the 3D visualization images as well as the ordinary DICOM images. As the 3D visualization images are not generated in real-time, the interaction of the client is very limited. Mitchell et al. [9] introduced a real-time teleradiology system that enables user to access to interactive advanced 2D and 3D visualization on a current smartphones such as Apple iPhone or iPod Touch, or an Android phone without storing patient image data on the device. According to their experience, a single remote reordering server can simultaneously support more than ten iPhone OS devices and is capable of delivering and displaying up to 14 frames per second on an iPhone OS device connected with the 802.11g Wi-Fi network. Therefore, the interactivity and stability need to be optimized for better experience of 3D visualization in real-time [16].

This paper presents a medical application, which allows access to PACS on mobile devices via wireless network connection. The MIP, MPR and direct volume rendering techniques for 3D Visualization of medical images are performed on a remote server to enhance this application. In order to improve the quality of experience, an algorithm for automatic adaption of remote rendering parameters based on the network status is proposed.

Materials and methods

Architecture

As depicted in Fig. 1, the proposed medical application follows three-tier architecture, consisting of the mobile client, the proxy server and the PACS server. As the core of this application, the proxy server establishes the contraction between the mobile client and the PACS server like a bridge, and completes the complex data-processing, such as formatting patient information, preprocessing the DICOM images, 2D and 3D visualization of medical images. The mobile client communicates with the proxy server using a custom protocol. The transmission of data between the proxy server and the PACS server is performed though the DICOM message service element-composite service (DIMSE-C), which supports the store, query/retrieve and echo services. Generally, the mobile client's operations may be divided into two categories according to whether they are performed locally or through the proxy server. Image manipulations, such as zooming, panning, rotation, calibrated distance measurement, Region of Interest (ROI) mark and measurement of area, are handled by applets on the client's side. User authentication, patient information, image retrieval, window leveling and 3D visualization are performed through the proxy server. In this paper, we take the open source project GDCM [17] to parse the DICOM files.

The medical personnel can query/retrieve patient information records with patient's ID, name and study date through the proposed medical application. The query/retrieve results are packaged into extensible markup language (XML) format on the proxy server and send to the mobile client to display as numbers and characters. The 2D and 3D views of the medical images selected by user are compressed with joint photographic experts group (JPEG) format after processing according to user requirements and sent to the mobile client to display. Additionally, the 3D Visualization functions, such as MIP, MPR, direct volume rendering, are speed up with an

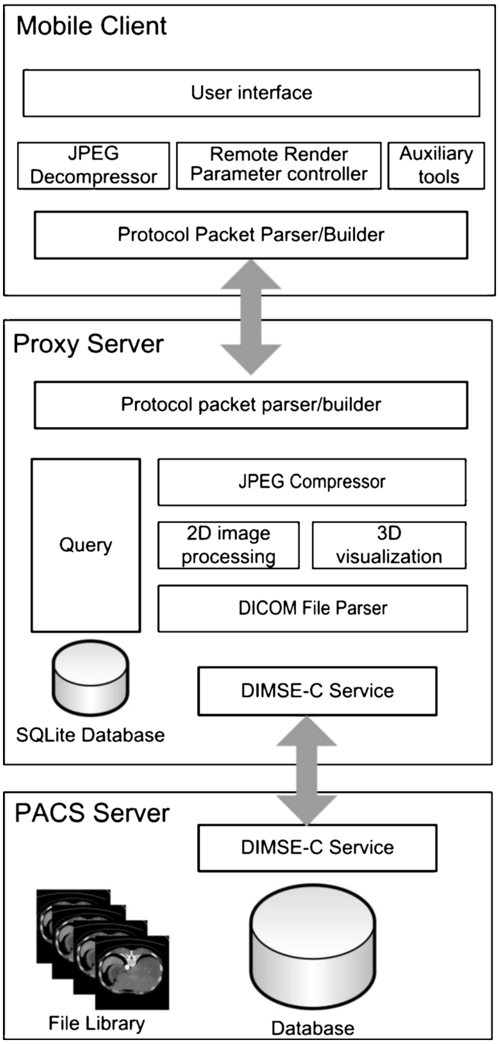

Fig. 1 Three-tier architecture of the proposed medical application

algorithm that automatically configures remote render parameters with the change of the wireless network status.

Communication protocol

The custom communication protocol between mobile client and proxy server is an application layer protocol based on transmission control protocol/internet protocol (TCP/IP) as shown in Fig. 2(a). The custom protocol is responsible for transmitting the user interactions from the mobile client to the proxy and the processing results from proxy server to mobile client, and all data will be added a packet header as shown in Fig. 2(b). Follow is a brief description of each field: The timestamp (TS) flag is the time when the packet is sent. The protocol version (V) flag is set to ensure that the proxy and mobile client is followed the unified protocol. The data transmit direction (D) flag is used to differentiate the upload and download data. The command type (T) flag indicates that the client will get data from the proxy or only change the status of the proxy server. The field (C) responds the commands as shown in Table 1, including user authentication, query/retrieval, 2D post-processing and 3D post-processing. The data type (DT) is the type of the data, and it may be bit stream (e.g. JPEG image) or string (e.g. study list). The data length (DL) is the length of the data value (DV) followed. The DV filed is used to store the command parameters or processing results.

Remote rendering

Figure 3 presents the progress of 3D visualization of medical images by with remote rendering. The progress is divided into two stages: load volumetric data and direct volume render. In the first stage, when the mobile client sends the select series UID to the proxy server, the proxy server first find the medical images with the series UID on local. If the target medical images is on local, the proxy will load them to build volumetric data, else the proxy server build C-MOVE connection to the PACS server and generates the medical images from the PACS server with the same series UID and archives them to the local database. When the user selects the medical images with the same series UID again, the proxy server loads the medical images directly from the local database instead of calling the PACS server. Then the proxy server loads volumetric data generated from the medical images into memory and send the status to the mobile client to announce that the proxy server is ready for volume rendering. In the second stage, the mobile client sends rendering parameters to the proxy server and requests for the 3D visualization images. The proxy server responds to the request, performs render operations on the current volumetric data with the parameters, compresses the 3D visualization images with libjpeg [18] and then transmits them to the mobile client. When the user changes the render parameters, the second stage needs to repeat, and when the user select other sequence of medical images, both the first and second stages need to repeat. All the operations are followed the custom protocol described in the previous section.

Adaptive remote rendering algorithm

In the process of remote rendering, interactive speed and image quality are important to the quality of the user interaction experience. In the proposed medical application, the mobile client accesses the proxy server through the 802.11g wireless area network (WLAN) or 3G cellular network with a limited bandwidth. In order to achieve a satisfying frame rate, we need to reduce the quality of the image. On the other hand, the image quality will directly influence the accuracy of the radiology diagnosis, so we can't reduce the image quality without limit. Therefore, we employ an adaptive algorithm that takes both the interactive speed and image quality into

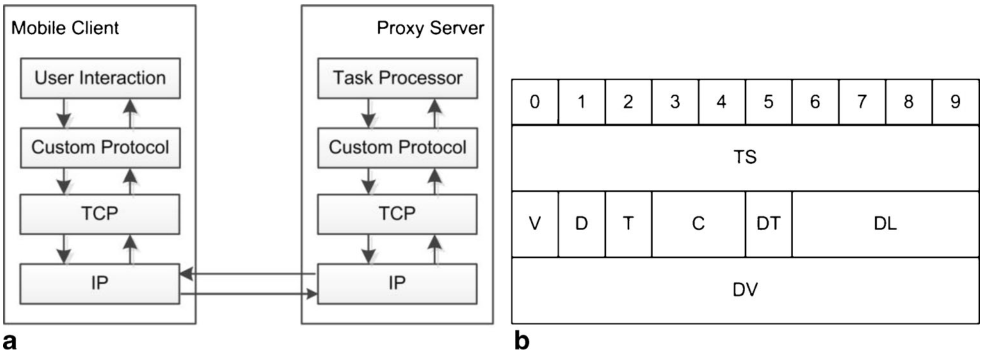

Fig. 2 The overview of the custom protocol stack (**a**) and the header of the custom protocol (**b**)

account to automatically change the remote render parameters with the network status.

There are three processes: rendering, compression and transmission need to be completed to achieve the remote rendering. The parameters which have influences on the quality of the interaction experience under the same computer hardware and wireless network status are list in the Table 2. According to Table 2, Both render resolution (R) and JPEG (Q) quality factor play an important role in improving the speed of the remote rendering. The R does not impact on the rendering time, but also determine the original image size which will have an effect on the memory size of the final image, and the Q determines both compression and impact on both of the memory size of the image and the image quality. So this paper take this two parameters as the mainly parameters to control the remote rendering.

The proposed system is depicted in Fig. 4. It is composed by a controller on the mobile client and a renderer, a compressor, a streamer server on the proxy server. In this system, the user will first set a target frame rate (F_t) and every time the mobile client receives the 3D visualization image, the mobile client calculate the actual frame rate (F_a) and compares it with F_t to estimate the parameters R and Q, which will be described in detail later. Then the mobile client feed the R and Q back to the proxy server to guide the renderer, compressor and stream server to generate target image. Finally, the proxy server transmits the image to mobile client and launches the next loop.

In this algorithm, the R and Q vary in a predefined range during interaction to give consideration to the image quality and Interactive fluency. Previous research [19] indicates that for neurological CT scans, a quality factor of 85 produces a compression ratio of 10:1, while a quality factor of 92 produces a compression ratio of 8:1. In our study, the range of JPEG quality factor is from 80 to 100, and that of render resolution is from 256×256 to 640×640. The detail process of the controller can be summarized in nine steps, as show in Fig. 5.

1) Set F_t, the time window (T_w) beyond which the records of receiving data are removed and the time between calculating the render parameters (ΔT).
2) Record the length of data (L_d), the time between request is sent and the whole image is received (L_t), and the moment when the client receives the image (t_m), when the mobile client receives every 3D visualization image.
3) Remove the records of data received beyond T_w.

Table 1 command supported by the custom protocol

Command name	C	T	Command name	C	T
Login	0x00	0	LoadDICOMSeries	0x50	0
CloseConnect	0x01	0	ReleaseDICOMSeries	0x51	0
GetStudylist	0x10	0	Set3Dmode	0x52	0
GetSerieslist	0x11	0	SetViewParameter	0x53	0
GetImagelist	0x12	0	SetSample	0x54	0
LoadDICOMData	0x20	0	SetVRLut	0x55	0
LoadDICOMDataByLocation	0x21	0	ObjectScale	0x56	0
ReleaseDICOMData	0x22	0	ObjectMove	0x57	0
SetImageProcessParameter	0x23	0	ObjectRotate	0x58	0
SetImageQuality	0x24	0	GetSeriesHistogram	0x59	1
GetDisplayImage	0x25	1	GetRayCasterImage	0x5A	1
GetDICOMInfo	0x26	1	GetMPRImage	0x5B	1
GetPixelValue	0x27	1	GetCPRImage	0x5C	1
			GetMIPImage	0x5D	1

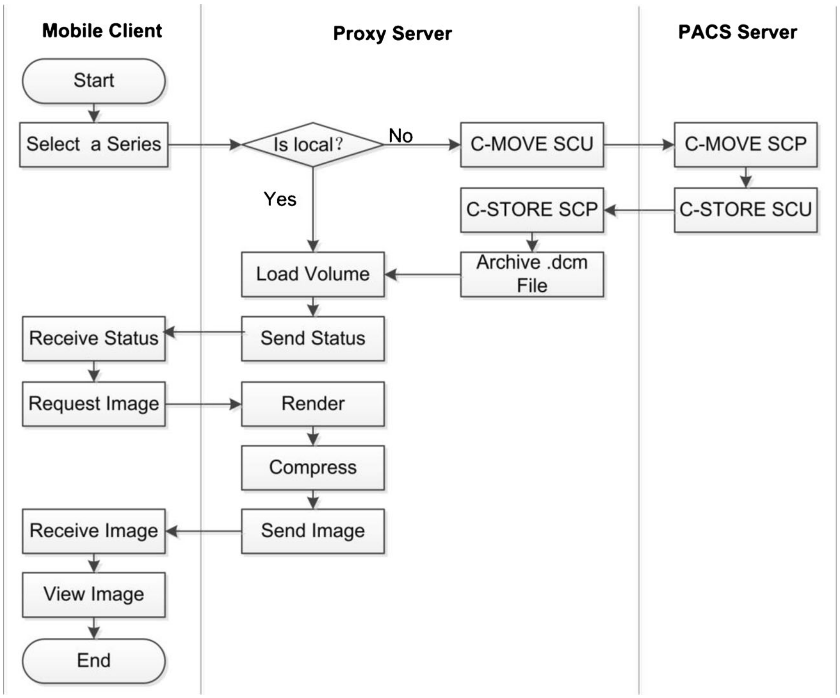

Fig. 3 The progress of 3D visualization of medical images with remote rendering

4) Calculate the sum of L_t (S_t) in the records and the number (N) of records within T_w.
5) Calculate F_a with S_t and N.
6) Compare the F_a with F_t to determine how to change R and Q.
7) Repeat steps 3), 4), 5) and 6) for every other ΔT.
8) Request 3D visualization image with R and Q;
9) Repeats the 2), 3), 4), 5), 6), 7), 8) steps.

For step 6), the compression ratio (C_r) of the last image is taken as a reference to calculate R of the next image. Assuming two adjacent images have the same C_r, the R and size of the compressed image (S) for of the adjacent two images have relation:

$$C_r = \frac{R_0}{S_0} = \frac{R_1}{S_1} \tag{1}$$

Where R_0 is the resolution of the last image, S_0 is the compressed size of last image, R_1 is the resolution of current image, and S_1 is the compressed size of current image. If the current network rate is N_s KB/s and the target frame rate is F_n Fps, S_1 should be described as (2) to achieve F_n:

$$S_1 = \frac{N_s}{F_n} \tag{2}$$

According to (1) and (2), the resolution of the next image can be computed as:

$$\mathrm{R}_1 = \frac{\mathrm{R}_0 \times \mathrm{N}_s}{\mathrm{S}_0 \times \mathrm{N}_n} \tag{3}$$

Therefore, the proposed medical application will generate a suitable render resolution immediately when the N_s changes.

With this algorithm, when the network status changes to a poor level that the real frame rate is lower than the targeted frame rate, the application first reduces R and Q to assure the F_a is close to the F_t, and if the R and Q drops to the lower limit, the F_a decreases accordingly. When the network status

Table 2 the parameters determine the quality of the interaction experience for remote rendering

Process	Parameter	Effect
Rendering	Resolution	Impact on the rendering time and the image size (width and height)
	Sample	Impact on the rendering time and the image quality
	Illumination	Impact on the rendering time and the image quality
Compression	Image size	Equal to rendering resolution, and impact on the memory size of image
	JPEG quality factor	Determine the compression ratio, and impact on the memory size of image and image quality
Transmission	Memory size of image	Impact on the rendering time, up to the compression ratio and image size

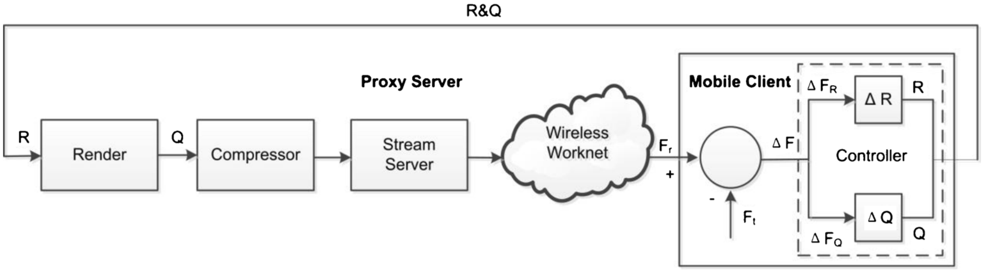

Fig. 4 the controller system for the adjustment of the render parameters

changes to a good level, where the F_a is higher than the $F_{t,}$ the application first improves the R and Q, and if the R and Q reach the upper limit, the F_a increases. After interaction, a full quality JPEG image with a resolution of 640×640 pixels is automatically sent to the mobile client to display.

Interaction with multi-touch screen

In the proposed medical application, the mobile client runs on iPad 2, which has a 9.7-in. diagonal multi-touch screen with 1,024×768 pixels, a pixel pitch of 0.19 mm, a contrast ratio=960:1 and a luminance=500 cd/m^2 [20]. The multi-touch screen of iPad enables faster and more convenient image zooming, panning, and rotation than using the mouse/keyboard. However, the interaction on the multi-touch screen can't accurately determine the coordinates the finger touch at, so we optimized the window leveling, image transformation, measurement, ROI marking analysis tool. For window leveling, the proposed application allowed user to input an accurate number in the textbox, select a default look-up-table (LUT), or swipe on the screen. For measurement and ROI marking, when there is a deviation between the marking lines with the target object, the user can adjust the whole line or the control points on the line to correct.

Result and discussion

This section discusses the performance of the proposed medical application. In this study, the open source PACS soft conquest [21] is took as the PACS server.

Figure 6 shows a session that the user examines the 3D visualization image of a magnetic resonance angiography (MRA) image sequence on the mobile client as well as on the proxy server. The screen captures of operations on an iPad are shown in Fig. 7. Once a study in the list is selected, the imaging sequences are listed (Fig. 7a). After an image sequence is selected, the image is presented in the image viewer. Tools can be selected for distance measurement, or ROI area measurement and marking (Fig. 7b), for selecting preset window levels, or manual adjustment of the levels (Fig. 7c), for 3D visualization with MPR (Fig. 7d), volume render (Fig. 7e) and edit the transfer function (Fig. 7f).

The performance of the proposed medical application was estimated from the followed three aspects: the performance of MIP, MPR, and direct VR techniques in WLAN network with various data transfer rate, the performance of direct VR in WLAN and 3G network, and the robustness of the adaptive remote render algorithm. All the set of measurements for performance evaluation were repeated in the networks as shown in the Fig. 8. The proxy server ran on a HP Z800 workstation with two 2.27 GHz Intel Xeon E5520 central processing unit (CPU), 8 GB RAM and a NVIDIA Quadro FX4800 (1.5GB) graphics cards in the WLAN path and ran on a command workstation with eight 3.4 GHz Intel Core i7-3770 central processing unit (CPU), 16 GB RAM and a NVIDIA GeForce GTX680 (2GB) graphics cards in the 3G path. As mentioned above, the mobile client ran on an iPad 2.

In all tests, $\mathbf{BW_{max}}$ is the maximum bandwidth limit on the proxy server, which limits the max data transfer rate of the wireless network. **R** is render resolution. **Q** is JPEG quality factor. **S** is the size of the compressed 3D visualization image. $\mathbf{F_r}$ is the average of received frame rate on the mobile client

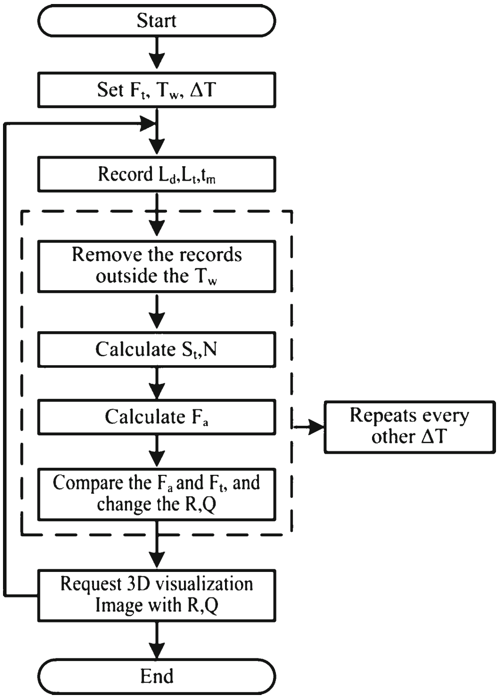
Start
Set F_t, T_w, ΔT
Record L_d,L_t,t_m
Remove the records outside the T_w
Calculate S_t,N
Calculate F_a
Compare the F_a and F_t, and change the R,Q
Repeats every other ΔT
Request 3D visualization Image with R,Q
End

Fig. 5 The detail process of the controller to change the remote render parameters automatically with the network status

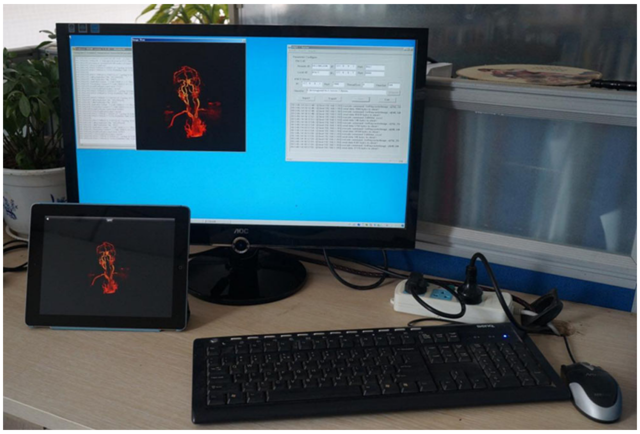

Fig. 6 A remote 3D visualization scenarios with the proposed medical application

over multiple measurements. When start a test, the mobile client first sent image requests to the proxy server continuously, and recorded the number of received images every 4 s. So it could calculate the temporary frame rate and mark it as F_t. When we got ten values of $\mathbf{F_t}$, we took the average as the $\mathbf{F_r}$ value. Every time the test environment changed, this process was iterated. Finally, **C.V.** is the coefficient of variation of $\mathbf{F_r}$, which is a parameter to estimate stability of received frame rate.

For the first aspect of performance evaluation, two series of medical images as shown in Table 3 were took to estimate the performance of MIP, MPR, and direct VR techniques in WLAN network with various data transfer rate.

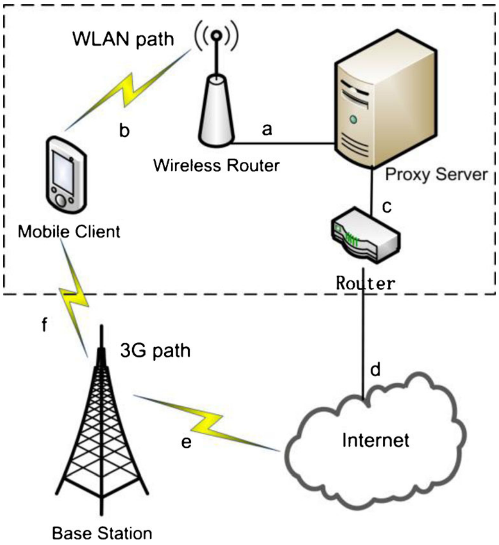

Fig. 8 the network architecture for performance evaluation

In this test, we set **Q** as 100 and **R** as the image's resolution. While the data was transmitted over WLAN network, we limited the maximum data output speed of the proxy server on the HP workstation with the software NetLimiter [22] to control the $\mathbf{BW_{max}}$ from 50 kB/s to 1,200 kB/s. As shown in the Fig. 9, we could get three results. First, the frame rates of the MIP, MPR and direct VR increased quickly as well as the maximum bandwidth of the proxy server until the frame rates reached to a plateau. The reason is that the bandwidth of the network was the bottleneck at the start, and when it increased to be large enough, it met the data transmission demand. Second, the frame rate of MPR was higher than the MIP and

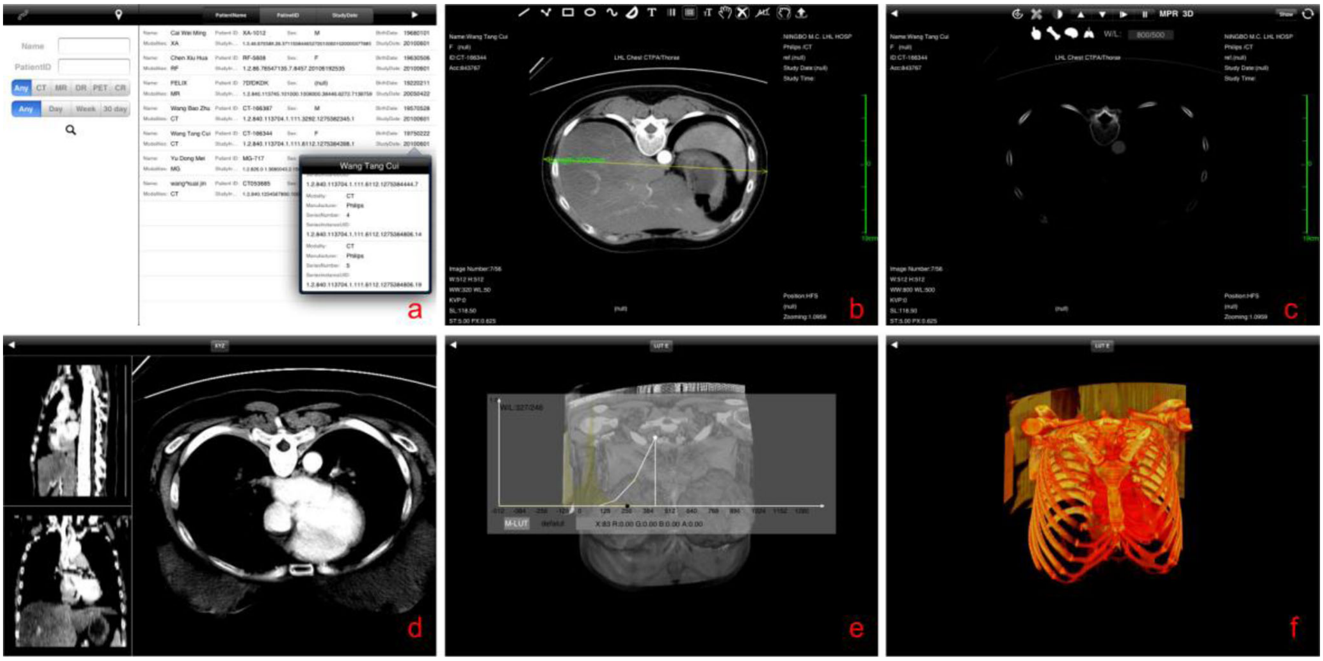

Fig. 7 Screen captures of selecting a study in the list (**a**). When using the viewing software, tools can be selected for distance measurements or ROI marking and measurements (**b**), window leveling (**c**), MPR (**d**), volume render (**e**), edit transfer function

Table 3 Medical images for performance evaluation

Serial number	Image type	Resolution	Image number	Size of volumetric data
1	CT	512×512	95	23.75 MB
2	MR	352×352	400	47.26 MB

direct VR in the same $\mathbf{BW_{max}}$. For example, when the $\mathbf{BW_{max}}$=600 kB/s, the max frame rate of MPR was 41.36 fps, while this of MIP was 17.39 fps and VR was 18.30 fps. This is because that the MIP and direct VR images were calculated with all of the original images, while the MPR images was only related to one slice of the volumetric data. Third, the frame rate of MIP and direct VR measured with CT images was faster than that of MR images. Otherwise, the frame rate of MPR measured with CT images was slower than that of MR images. This is because that the **R** of MIP and direct VR is equal to the original image's resolution and this of the MPR is equal to the original image's width or height multiplied by number of original images. The resolution of CT images was higher than the resolution of MR images, but the number of CT images was less than the number of MR images.

For the second aspect of performance evaluation, 120 MRA images with a resolution of 576×448 pixels were took as the volumetric data to investigate the performance of direct VR over WLAN and 3G network.

First, we measured the frame rate with various render resolution (test 1), JPEG quality factor (test 2) and data transfer rate of the wireless network (test 3) over WLAN network. The test environment and results are shown in Tables 4, 5, and 6. For instance, the first row of Table 4 shows the received frame rate was 33.6 fps and the size of the received image was 8.42 KB, while the render resolution was 256×256, the JPEG quality factor was 90 and the maximum bandwidth limit of wireless network was 500 kB/s.

According to the test1 and test 2 the memory size of received image increased with higher render resolution and JPEG quality factor, which formed an incremental burden to the wireless network. However, the maximum bandwidth limit was fixed, so the received frame rate went down. Specially, the render resolution had a relatively larger influence on the received frame rate than the JPEG quality factor, since the render resolution did not only directly decide the memory size of 3D visualization image, but also had a significant influence on the volume render time. The higher render resolution took more time to render and results in a larger image size, while the JPEG quality factor decided the quality of the image sent to the mobile client, and only affected the size of the image. We could change render resolution or quality factor to achieve targeted frame rate in a stable network environment, which depended on the scenario. For example, in order to reach a frame rate of 13.5 fps, we should configure the render parameters as indicated in row 7 of Table 4 or row 4 of Table 5.

In the test 3, we changed the maximum bandwidth limit on the proxy server to estimate the performance of the remote render in various wireless networks. According to Table 6, the received frame rate went from 1.9 fps up to 19.1 fps while the maximum bandwidth limit increases from 50 kB/s to 600 kB/s.

It was noteworthy that the actual device throughput ($S\times F_a$) was lower than the $\mathbf{BW_{max}}$, as shown like row 1 of Table 4, row 1 of Table 5, row 7 of Table 6. Because of the limit of the device capabilities, the bandwidth didn't be fully utilized with a high actual frame rate.

Second, China Unicom's WCDMA network was took as the 3G network to measure the frame rate with various render resolution, JPEG quality factor with 3G path. In these two assessments, the actual data transmission speed was about 200 kB/s. The test environment and results were shown in Tables 7 and 8.

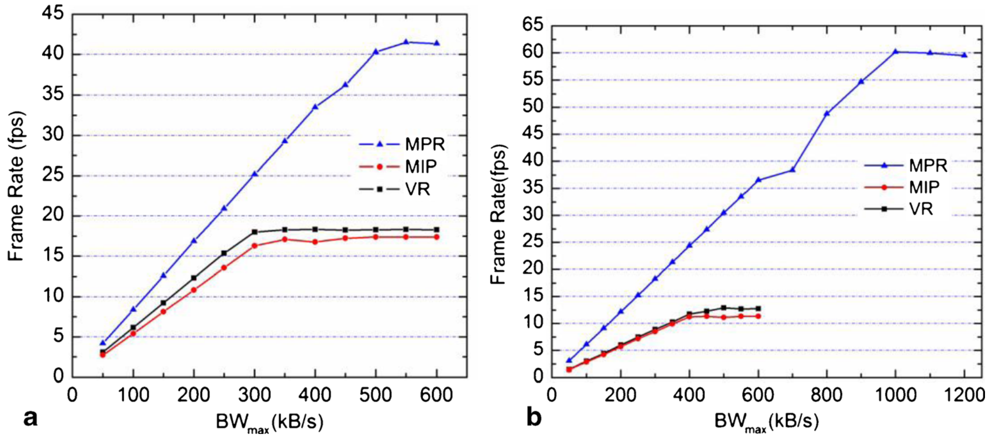

Fig. 9 Performance of MIP, MPR, and direct VR techniques in WLAN with various data transfer rate

Table 4 The performance of direct VR with various render resolution over WLAN network

BW_{max} (kB/s)	R (pixels)	Q	S (KB)	F_r (fps)	C.V.
500	256×256	90	8.42	33.6	2.1 %
500	320×320	90	11.99	27.3	3.7 %
500	384×384	90	15.51	24.4	3.8 %
500	448×448	90	20.86	22.6	2.4 %
500	512×512	90	25.76	17.4	2.9 %
500	576×576	90	30.39	16.6	0.8 %
500	640×640	90	36.78	13.6	0.5 %

According to the test results, we can see that the frame rate was less and **C.V.** was larger in 3G network than in the WLAN network with the same volumetric data, resolution, JPEG quality factor and similar bandwidth. In the 3G network, there was a wide area network (WAN), the topology of which was unknown, between the mobile client and proxy server. Before the images generated on the proxy server reached to the mobile client, they were transmitted several times on the network node like **c, d, e, f** in the Fig. 8. Though the data transmission speed at the node **f** was about 200 kB/s, the other node may be lower than 200 kB/s. As a result, the frame rate in the 3G network was less than that in the WLAN network. On the other hand, because any change of each node had effect on the current network transmission rate, the 3G network was more unstable than the WLAN network. It lead to that the C.V. was larger in 3G network.

Finally, the performance of the proposed medical application concerns the robustness of the adaptive remote render algorithm was shown in the Fig. 10. We started the remote render with an initial resolution of 512×512 pixels, an initial JPEG quality factor of 100 and a target frame rate of 15 fps. In fact, the target fame was not an exact value, and the actual frame rate that had a difference value less than 2 fps with the target frame rate was considered to meet the demand. The users can also customize the target frame rate in the mobile client. During the interaction, we changed the different $\mathbf{BW_{max}}$ on the proxy server to simulate various networks status. As shown in the Fig. 10, the actual frame rate adapted to the network bandwidth automatically.

Table 5 The performance of direct VR with various jpeg quality factor over WLAN network

BW_{max} (kB/s)	R (pixels)	Q	S (KB)	F_r (fps)	C.V.
500	512×512	80	17.56	21.1	1.5 %
500	512×512	85	20.58	20.2	2.7 %
500	512×512	90	25.76	17.4	2.9 %
500	512×512	95	36.31	13.5	3.9 %
500	512×512	100	49.2	10.2	0.2 %

Table 6 The performance of direct VR with various single strength of the wireless network over WLAN network

BW_{max} (kB/s)	R (pixels)	Q	S (KB)	F_r (fps)	C.V.
50	512×512	90	25.67	1.9	0.6 %
100	512×512	90	25.76	3.9	0.9 %
200	512×512	90	25.53	7.8	0.3 %
300	512×512	90	25.67	11.7	0.7 %
400	512×512	90	25.67	15.6	1.0 %
500	512×512	90	25.76	17.4	2.9 %
600	512×512	90	25.67	19.1	1.6 %

During the first part of the test (from $t=t_0$ to $t=t_1$), the proposed medical application ran in a stable network (BW_{max}=500 kB/s). In this case, the frame rate received at the mobile client didn't reach the target frame rate with a high resolution (R=512×512) and quality (Q=100). So the medical application continuously reduced the resolution to adapt to the bandwidth until it reached a suitable value (R=400×400), with which the mobile client received images at the target frame rate ($F_a \approx 15$ fps). At time $t=t_1$, the maximum bandwidth limit decreased to a medium level (BW_{max}=250 kB/s), and the actual frame rate received on the mobile client dropt fast from F_a=16.22 fps to F_a=9.35 fps. In order to minimize the change of actual frame rate caused by the drop-off of the network, the medical application reduced the resolution again to a suitable value (R=256×256), and the received frame rate on the mobile client reached the target frame rate ($F_a \approx 16$ fps). At time $t=t_2$, the maximum bandwidth limit was set to a low level (BW_{max}=100 kB/s), which cause another dropt of the actual frame rate from F_a=16.71 fps to F_a=7.49 fps. In this time, the resolution had been the lower limit, so the medical application reduced the quality factor to respond to the drop of actual frame rate caused by the bandwidth limit of the network. At the end of this stage, the proxy server compressed the image with a quality factor of 84, and the received frame rate on the mobile client kept at a steady level ($F_a \approx 15$ fps).

In contrast, at time $t=t_3$, the maximum bandwidth limit went back to the medium level (BW_{max}=250 kB/s), which

Table 7 The performance of direct VR with various render resolution over 3G network

R (pixels)	Q	S (KB)	F_r (fps)	C.V.
256×256	90	7.89	5.83	4.9 %
320×320	90	11.25	4.98	6.7 %
384×384	90	14.73	4.41	6.0 %
448×448	90	19.65	4.38	4.8 %
512×512	90	24.46	4.08	6.2 %
576×576	90	28.77	3.74	3.7 %
640×640	90	35.02	3.62	6.6 %

Table 8 The performance of direct VR with various jpeg quality factor over 3G network

R (pixels)	Q	S (KB)	F_r (fps)	C.V.
512×512	80	16.54	4.98	5.7 %
512×512	85	19.36	4.70	2.8 %
512×512	90	24.46	4.08	6.2 %
512×512	95	34.76	3.23	2.9 %
512×512	100	59.42	2.82	5.0 %

caused a dramatic increase in the actual frame rate received on the mobile client from F_a=14.94 fps to F_a=30.58 fps. In this case, we paid more attention to the image quality than the interaction speed, so the proposed medical application improved the render resolution and quality factor. First, the quality factor increased to 92 (from $t=t_3$ to $t=t_4$), and then the render resolution increased as far as possible (R=384×384) until the actual frame rate approached the target frame rate (F_a≈15 fps). At $t=t_4$, the maximum bandwidth limit resumed to the initial level (BW_{max}=500 kB/s), which cause another increased in the actual frame rate received on the mobile client from F_a=15.57 fps to F_a=21.98 fps. Finally, the proposed medical application ran at a frame rate of 15 fps with a resolution of 560×560 and a JPEG quality factor of 92. This suggested every time a significant change occurred in the network environment, the actual frame rate changed immediately. The proposed medical application automatically adjusted the render resolution and JPEG quality factor to keep the actual frame rate stable as excepted value, which gave the user good quality of experience.

Conclusion

This paper describes a medical application that can deliver medical images in PACS system to the mobile device with a real-time interactive remote render technique. Through this application, the radiographers can view real-time 2D and 3D views of the medical images wherever a wireless network is available. This application may be useful to the radiographers and clinicians. On one hand, examining the patient information and medical images on the mobile device is adequately useful for early diagnosis and treatment in emergent cases according to [8, 23]. On the other hand, the clinician can describe the patient's condition more detail with the medical images shown on this medical application to the patients. Of course, further researches are needed. At first, it needs to evaluate whether there is a difference between the sensitivity, specificity and accuracy of detecting on the mobile devices and on the PACS workstations just like [20, 24]. Second, the secure cryptographic [25] and digital signatures [26] techniques should be employed to protect the validity and security of medical images.

At now the mobile device can handle many computing works which can only be implemented on the personal computer before. The mobile device can not only complete the processing of the medical images in two-dimensional, but also hold volume rendering of the small data on local, as previously described [11, 27]. But there are still a number of reasons to take remote rendering as a good way to show medical imaging on the mobile device. First, a study of the patient will generate several series medial images, which maybe reach hundreds of megabytes. Before processing, it needs to transfer the medical images to the mobile device, which is time-consuming work, and this couldn't meet the demand of real-time examining of medical images in patient rounds. Second, the storage of the mobile device is limited, which can only store a dozen patients' image data. Third, when examining image by local processing, the medical images must be stored on the mobile device, and everyone uses the mobile device can get them. However, the mobile device will not store any images or patient information on local by the remote rendering, once the mobile device disconnect to the proxy server. Fourth, data transmission and complex three-dimensional calculations are seriously energy-consuming operation, which will accelerate energy depletion of the mobile device.

In this paper, the performance of the proposed medical application was estimated from the three aspects. In first test, when the $\mathbf{BW_{max}}$=200 kB/s, **R**=512×512 and **Q**=100, the frame rate of MPR was 12.18 fps, this of MIP was 5.69 fps and VR was 5.98 fps. In second test, when rendering the volume data with a resolution of 512×512 and compressing the render result with a JPEG quality factor of 90 by a 200 KB/s transmission speed, the frame rate kept at 7.8 fps in the WLAN network and 4.08 fps in the 3G network. According to the literature [28], subjective reactions to the quality and watch ability of videos seem to support rates of 5 fps. In third test, balance, sensitivity and stability considered were taken as the main factors to improve the interactive experience. First, when the wireless network changed to be pool in activation, the application will automatically modify the **R** and **Q** to keep the actual frame rate stable and be closed to the target frame rate. At the same time, the **R** and **Q** only varied in a set range, which can ensure the image viewable in inaction. Once the interaction was end, a clear image with full resolution was sent to the client. This mechanism ensured the speed of interaction and quality of image for view both were best. Second, the compression ratio of last image was taken as a guide to evaluate the R of current image, which improve the sensitivity significantly. As shown in Fig. 10, the maximum bandwidth reduced to 250 kB/s at the time $t=t_1$, and it lead to that the actual frame rate dropt into 9.35 fps. However, the application recovered the actual frame rate to the target frame rate 15 fps just in 4 s. Third, stability was one of the most important

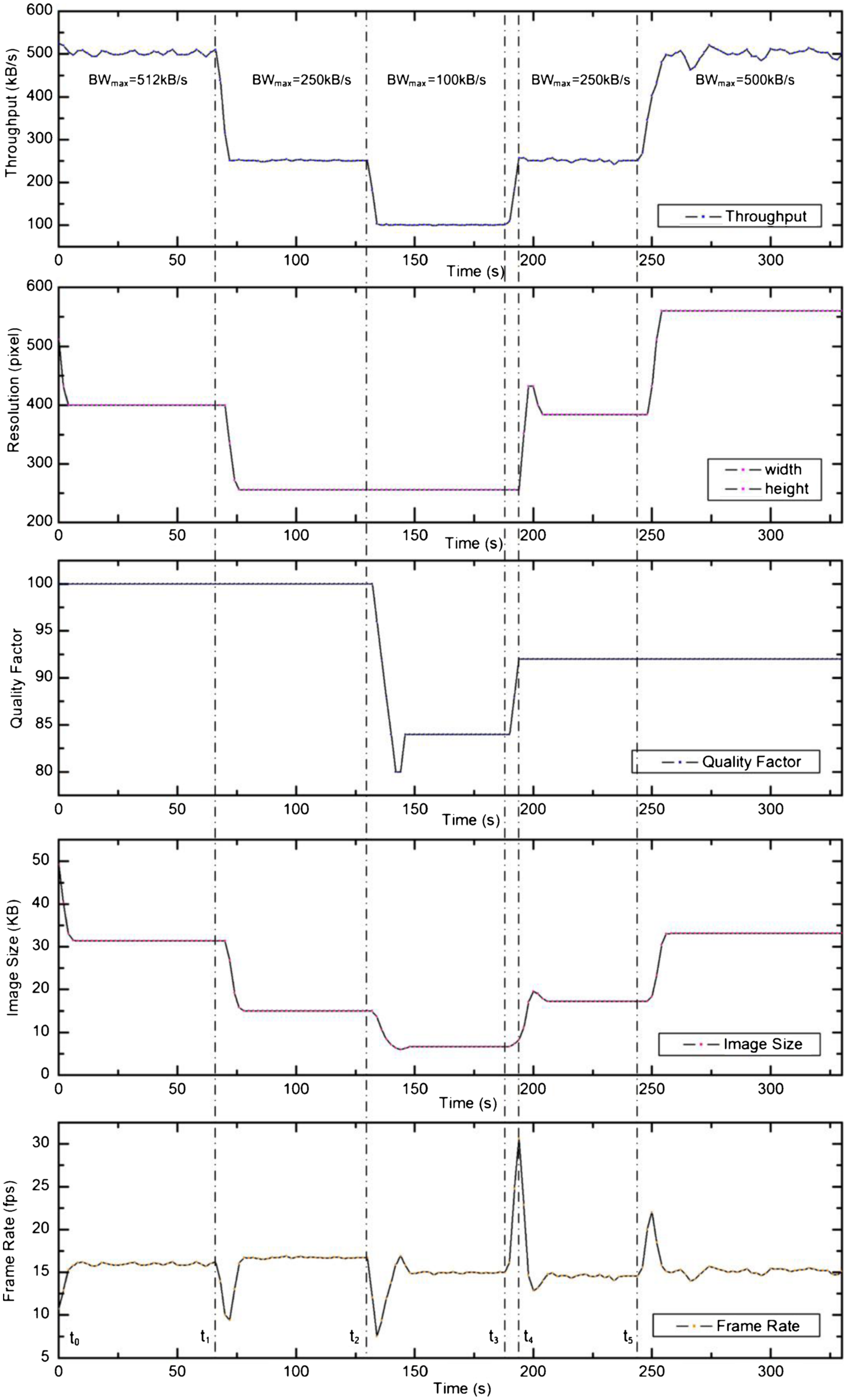

Fig. 10 Performance of the proposed medical application concerns robustness of the remote render

factors that affected experience. In order to avoid the render parameters varyed frequently, the application set the target frame rate within a range to keep from the render parameters feeding back to the normal fluctuation of wireless network.

What's more, the application calculated the rendering and compression parameters on mobile client instead of on the proxy server, which reduced the load on the server and make full use of mobile client's computing performance. In summary, the benefits of the proposed application can be summarized into the following:

1) This application may help doctors to view medical images at any time and at everywhere with a wireless network. With this software, the images of emergent patients can be delivered to the radiologists out of hospital, which will shortens the time from receiving emergency rescue to performing treatment.
2) This application can present a high quality 3D visualization image to the radiologists on a mobile device despite its limited computational capabilities. Therefore, we design an adaptive algorithm that takes both the interactive speed and image quality into account to improve the quality of experience in different network environment.
3) This application is based on three-tier architecture. The original medical images of patient in PACS will only be delivered to the proxy server and there is no file on the mobile device, which can ensure the security and protect patient privacy.

In additional, the radiologists can configure the remote render parameters to obtain 3D visualization images with different qualities by requirement of diagnosis. The result of performance test shows that the proposed medical application can provide an acceptable quality of interactive experience.

In the future, image data on mobile devices like PDAs or smart-phones will be widely used. We will pay more attend to design more efficient interactive way. For instance, the statistical result of users' operations can be taken into account to design the template parameters, which will help the radiologist to spend less time to get satisfactory results. In addition, there will be multiple clients simultaneously access the proxy server in a real environment, thus optimization of multiple clients concurrent operations need to increase.

Acknowledgments This work was supported by the National High-Tech Research and Development Program of China (863 Program: 2012AA02A606), the Program for New Century Excellent Talents in University (Grant No. NCET-10-0386) and the graduate innovation fund of Graduate Base of Innovation and Enterprise (No.HF-11-39-2013).

References

1. Fridell, K., Aspelin, P., Lars Edgren, L., and Lindsköld, L. N., PACS influence the radiographer's work. *Radiography* 15:12, 2009.
2. Wang, K. C., Filice, R. W., Philbin, J. F., Siegel, E. L., and Nagy, P. G., Five levels of PACS modularity: Integrating 3D and other advanced visualization tools. *J. Digit. Imaging* 24(6):1096–1102, 2011. doi:10.1007/s10278-011-9366-1.
3. Ivetic, D., and Dragan, D., Medical image on the go! *J. Med. Syst.* 35(4):499–516, 2011. doi:10.1007/s10916-009-9386-2.
4. Spelic, D., and Zalik, B., Lossless compression of threshold-segmented medical images. *J. Med. Syst.* 36(4):2349–2357, 2012. doi:10.1007/s10916-011-9702-5.
5. Valente, F., Viana-Ferreira, C., Costa, C., and Oliveira, J. L., A RESTful image gateway for multiple medical image repositories. *IEEE Trans. Inf. Technol. Biomed.* 16(3):356–364, 2012. doi:10.1109/titb.2011.2176497.
6. Lipton, P., Nagy, P., and Sevinc, G., Leveraging internet technologies with DICOM WADO. *J. Digit. Imaging* 25(5):646–652, 2012. doi:10.1007/s10278-012-9469-3.
7. Lee, S., Lee, T., Jin, G., and Hong, J., An implementation of wireless medical image transmission system on mobile devices. *J. Med. Syst.* 32(6):471–480, 2008.
8. Hsieh, J. C., and Lo, H. C., The clinical application of a PACS-dependent 12-lead ECG and image information system in E-medicine and telemedicine. *J. Digit. Imaging* 23(4):501–513, 2010.
9. Mitchell, J. R., Sharma, P., Modi, J., Simpson, M., Thomas, M., Hill, M. D., and Goyal, M., A smartphone clientâ€"server teleradiology system for primary diagnosis of acute stroke. J. Med. Internet Res. 13(2), 2011. doi:10.2196/jmir.1732.
10. Székely, A., Talanow, R., and Bágyi, P., Smartphones, tablets and mobile applications for radiology. *Eur. J. Radiol.* 82:829–836, 2013.
11. Noon, C., Holub, J., and Winer, E., Real-time volume rendering of digital medical images on an iOS device. In: Akopian, D., Creutzburg, R., Georgiev, T.G., et al. (Eds.), *Multimedia Content and Mobile Devices*. Vol. 8667. Proceedings of SPIE. Spie-Int Soc Optical Engineering, Bellingham, 2013. doi:10.1117/12.2005335.
12. Butson, C. R., Tamm, G., Jain, S., Fogal, T., and Kruger, J., Evaluation of interactive visualization on mobile computing platforms for selection of deep brain stimulation parameters. *IEEE Trans. Vis. Comput. Graph* 19(1):108–117, 2012. doi:10.1109/tvcg.2012.92.
13. Johnson, C. R., Biomedical visual computing: Case studies and challenges. *Comput. Sci. Eng.* 14(1):12–20, 2012.
14. Aycan mobile. http://www.aycan.com/products/aycan-mobile.html. Accessed 11 December 2013.
15. Choudhri, A. F., and Radvany, M. G., Initial experience with a handheld device digital imaging and communications in medicine viewer: OsiriX mobile on the iPhone. *J. Digit. Imaging* 24(2):184–189, 2011.
16. Paravati, G., Sanna, A., Lamberti, F., and Ciminiera, L., A novel approach to support quality of experience in remote visualization on mobile devices. In: *Eurographics*, pp 223–226, 2008.
17. GDCM. http://sourceforge.net/projects/gdcm/. Accessed 11 December 2013.
18. jpegsr8d. http://www.ijg.org/. Accessed 11 December 2013.
19. Savcenko, V., Erickson, B. J., Persons, K. R., Campeau, N. G., Huston, J., Wood, C. P., and Schreiner, S. A., An evaluation of JPEG and JPEG 2000 irreversible compression algorithms applied to neurologic computed tomography and magnetic resonance images. *J. Digit. Imaging* 13(2):183–185, 2000.
20. Johnson, P. T., Zimmerman, S. L., Heath, D., Eng, J., Horton, K. M., Scott, W. W., and Fishman, E. K., The iPad as a mobile device for CT display and interpretation: Diagnostic accuracy for identification of pulmonary embolism. *Emerg. Radiol.* 19(4):323–327, 2012.
21. Herk Mv Conquest. http://ingenium.home.xs4all.nl/dicom.html Accessed 11 December 2013.
22. NetLimiter. http://www.netlimiter.com/. Accessed 24 January 2014, 2003.
23. Toomey, R. J., Ryan, J. T., McEntee, M. F., Evanoff, M. G., Chakraborty, D. P., McNulty, J. P., Manning, D. J., Thomas, E. M., and Brennan, P. C., The diagnostic efficacy of hand-held devices for

emergency radiological consultation. *AJR Am. J. Roentgenol.* 194(2): 467–474, 2010. doi:10.2214/AJR.09.3418.

24. Park, J. B., Choi, H. J., Lee, J. H., and Kang, B. S., An assessment of the iPad 2 as a CT teleradiology tool using brain CT with subtle intracranial hemorrhage under conventional illumination. *J. Digit. Imaging* 26(4):1–8, 2013. doi:10.1007/s10278-013-9580-0.
25. Chen, W. K., and Tso, H. K., Visual sharing protection method for medical images. J. Med. Syst. 37(1), 2013. doi:10.1007/s10916-012-9900-9.
26. Lien, C. Y., Yang, T. L., Hsiao, C. H., and Kao, T., Realizing digital signatures for medical imaging and reporting in a PACS environment. J. Med. Syst. 37(1), 2013. doi:10.1007/s10916-012-9924-1
27. Mobeen, M. M., and Feng, L., Mobile visualization of biomedical volume datasets. *J. Internet. Technol. Secured Trans.* 1(2):52–60, 2012.
28. Chen, J. Y., and Thropp, J. E., Review of low frame rate effects on human performance. *IEEE Trans. Syst. Man Cybern. A* 37(6):1063–1076, 2007. doi:10.1109/tsmca.2007.904779.