Multimed Tools Appl (2015) 74:5209–5223
DOI 10.1007/s11042-014-2039-6

# An efficient block classification for media healthcare service in mobile cloud computing

**Nguyen Thuy An · Cong-Thinh Huynh · ByungKwan Lee · Choong Seon Hong · Eui-Nam Huh**

Received: 12 September 2013 / Revised: 11 April 2014 / Accepted: 17 April 2014 /
Published online: 4 May 2014


**Abstract** Recently, the strict investment in healthcare domain achieves the professional quality and convenient demands of customers. Especially, with the rapid growth of modern high technology, healthcare is providing many kinds of services for patients. One of those is mobile healthcare that is the integration of mobile computing and health monitoring. By using remote protocol, the service via mobile can directly send the patient heart failure sign to the doctor. Among the existing client-server protocols, Remote Desktop Protocol (RDP) is the typical method but it needs enhancement to adapt for more rigorous requirements: real time, quality of service (QoS), etc. In this paper, we present the architecture with flexibly coding screen image to improve quality of experience (QoE) of users and ensure low bandwidth services. Besides, in order to guarantee the best image quality and reduce redundant communication by using different compression encoding techniques for movies, images and other formats. Based on the edge detection Sobel operator that strongly focuses through horizontal, vertical direction and low complexity computation, our proposed method introduces the efficient block classification for the media healthcare services.

**Keywords** Healthcare · Mobile thin client · Image processing · Remote protocol · Visualization · Block classification

N. Thuy An · C.-T. Huynh · C. S. Hong · E.-N. Huh
Department of Computer Engineering, Kyung Hee University, 1 Seocheon, Giheung, Yongin, Gyeonggi, South Korea
e-mail: thuyannguyen306@gmail.com

C.-T. Huynh
e-mail: thinhhc@gmail.com

C. S. Hong
e-mail: cshong@khu.ac.kr

E.-N. Huh
e-mail: johnhuh@khu.ac.kr

B. Lee
Department of Computer, Kwandong University, Kangneung, Gangwo, 210-701, South Korea
e-mail: bklee@kd.ac.kr

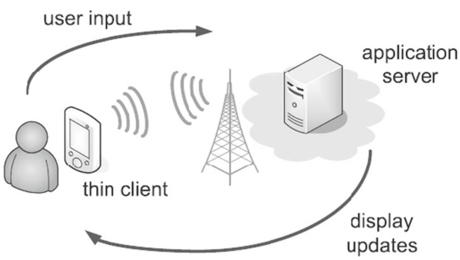
## 1 Introduction

Nowadays, by using mobile technology and cloud internet, healthcare service is being upgraded, achieve many good results and provide many services for human life. Mobile healthcare is transformed from traditional to advance method that replaces manual care into modern supported equipment. This has great meaning in daily life, brings patients and their family with the efficient results: helps them save valuable time and satisfies their wish for personal control and lower cost of long term medical care because it avoids the waste cost of spending human resources for manual processing. Heart failure is a chronic disease that is leading the causes of death, that's why the new developments in heart failure care are seriously concerned. It requires both doctor and patient make the lifestyle changes before conditions worsen with complex and costly treatments. For example, it is inconvenient for the patient to go to doctor frequently and the doctor need to follow patient's disease. The new hospital device is put inside patient and sends their signs to server then the doctor can easily diagnose, adjust the medication or recommend his advice. The screen of patient heart beating or medical sign will be transmitted to doctors who is observing the patient signs using their mobile devices which is connecting server by using remote protocol. Therefore, the real time and quality of delivery transmission contribute the useful and successful of mobile healthcare system that it is the exact diagnoses of doctor.

For that reason, there are not only healthcare but also other fields raise a big challenge in mobile service about the good response time, the high quality of services, numerous types of services as well as multiple of service providers. Together with the growth of mobile network, mobile application and cloud computing mobile service is growing rapidly. The key benefit of such mobile services is that they allow users to access instantly their applications everywhere at any time. Specifically, a client sends an input event request to a server, the sever receives and deploys several offloading computing tasks, then renders screen image, encodes, and finally transfers the results to the client (as shown in Fig. 1). These graphical images will be decoded and displayed on the client side. We can summarize several important factors to measure the performance of the system:

- Network condition: latency and bandwidth utilization. Latency should be as small as possible. And with many different bandwidth quotas, users always want to get the highest acceptable performance.
- Hardware resource consumption: CPU has a critical meaning in the processing because the limited hardware resource of the client device and overhead consumption on the server.
- User demands: Cross-platform adaptation because it is a great convenience for users who use many kinds of devices such as laptop, smart phone, tablet, etc. with different operating systems.

**Fig. 1** General thin client system



These factors directly affect QoE (Quality of Experience) that user perceives in each request which is made from local desktop, where claims for more high fidelity of display. This topic has risen as the big challenge for providers, developers and researchers. Many solutions have been developed and presented that include or focus on hardware and software improvement in order to satisfy demands of user. RDP [21], Virtual network computing (VNC) [12] and Thin Client Internet Computing (THINC) [6] were the first remote protocols but still have drawbacks. Teradici company gave hardware solution with PCoIP technology [16]. Many screen compression technologies brought big advantage to thin client computing. To overcome the limitation of mobile device (hardware resource, storage power, battery life time, etc.), the application logic will be executed on the server and rendered screen more closely. Especially for graphics, this progress helps client to save GPU, CPU requirements so that it can be overloaded due to more important processes.

Therefore, compression technology is one of the most important contributions to improve real-time video streaming. Some encoding and decoding formats such as motion MPEG, H.264 AVC were developed in hybrid remote display protocol.

Each proposed solution has overcome several weaknesses of the protocol system. The satisfaction of customer is the motivation for researcher. Especially, media healthcare is expected as the best qualitative vision and smooth transmission frame by frame. In this paper, we focus on reducing redundant data for multimedia application services as well as improving visual quality of end users for high fidelity display applications. The compound screen consists of many applications such as text, web, image, video and so on. After determining the different two regions, we can apply the best suitable compression formats depend on the quality requirement of each block, text block needs more qualitative vision and often is updated slowly. This paper contributes not only to complete block-base architecture, but also to comprehend a classification algorithm that helps us to distinguish screen into text or non-text blocks fully accurately with low complexity computation. At the end, this will let each region be coded properly before transmission.

This paper is organized as follows, in Section 2, describes some recent works on thin client architecture, Section 3 introduces our architecture, general idea for the system and the block classification algorithm and Section 4 presents experimental results. The final section includes conclusion and future work in Section 5.

## 2 Related work

There are many thin-client protocols have been developed, we can summarize collaboration into two categories: high level API display function and low level image frame buffers. High level RDP which was developed commercially by Microsoft that uses specific GDI display API calls or ICA (Independent Computing Architecture) from Citrix based on the same GDI API calls but optimized for low-speed WAN links. The disadvantages of first category protocols are operating system dependence and degradation performance in case playing video although it takes advantage of graphic interface. The low level category is RFB which is used in VNC, THINC, etc. We easily recognize the system will lack of convenience with real time application such as video playback. Hybrid remote desktop protocol was approached and implemented to overcome this problem. It flexibly transferred rendered screen to client depend on still or motion image with the support of GPU's server. Compression is considered as the critical role in that system. Lossless compression format is being used to transmit motion image that helps saving a lot of data. For example, H.264 format is approached by Simoens in his hybrid protocol [14] or MPEG4 by Tizon [20]. But it is the big problem in CPU consumption overhead on server because it handled end-to-end execution of program. This weak point affected badly to the performance of system when many users connected and wanted to work at the same time. Sharing processing between client and server seem to be considered recently. Which tasks will be handled in the client and which in the server? How can we flexibly separate and combine all tasks through the process to get high-est performance in many cases of applications. Those are difficult and important points to develop this system as was illustrated generally by Lu [11].

Another part of the problem is the high demand of finite vision of application, Shen [16] proposed a high performance remote computing platform which is compression-friendly-model where all data redundancies are exploited in the compression view port [16]. This design used the Skip and Intra blocks to denote the kind of blocks coding mechanisms that make the wise elimination of redundant data, only just forwarding the application level update regions into compressors and refusing still regions. This solution provided high quality vision to user by using prior definition of blocks and applies the corresponding compression format to each block.

On the other hand, some methods were proposed to classify block. Keslassy [8] used the method transform coefficient likelihood (TCL) in order to separate compound screen image into text and graphic blocks by computing DCT coefficient for 8*8 blocks. Li and Lei based on wavelet-based coding that recognize smoothly, text, graphics and image blocks [4].

Said [13] proposed block classification based on the number of color. While Lin and Hao [10] chose compound image compression algorithm for real time computer screen image transmission by applying a first pass of 2-pass segmentation procedure that classifies image blocks into picture and text based on threshold number of color of each block. The shape-based and palate-based coding algorithm is used for lossless format and lossy JPEG encode pictorial pixels. All those designs aim to improve performance for real time application. This is the reason why we are including in the paper an efficient algorithm to images into two types of blocks: text/graphic and picture/background.

## 3 Architecture

### 3.1 System overview

General mechanism of remote computing is communication between client and server. It has three components: the server does offloading computing task (execution application), then captures the screen, encodes and transmits to the client. The client receives, decodes and dis-plays and also sends input events to the server. This transmission is handled by communication protocol between the client and the server. But the more the graphic application captured such as complex and huge business and multimedia applications, the more challenges for high quality vision requirement and efficient transmission is affected. The screen image has many kinds of applications such as: slide, video, document and so on. Besides video is quickly updated while background or text almost is still. For that proof, selecting only one best compression format does not satisfy QoE and QoS demands of system. Classification image into text and non-text blocks is necessary and important, then apply suitable code for each kind of block.

Figure 2 shows general architecture of the system. First we compare differences between current and previous frames for determining skip or transmit and transmit full frame or a

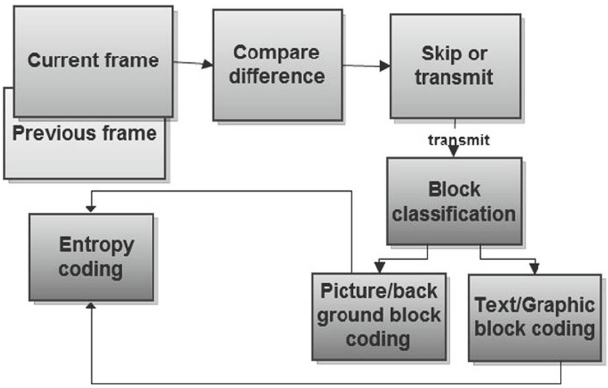



**Fig. 2** Architecture of the remote system

part of frame. Skip processing is entirely suitable for unchanged region. This work reduced a lot of amount data sending instead of transmitting redundant frames. Hence, the bandwidth network can be reduced which improves the QoS. The system only considers the different regions corresponding to the updated part that need to be transmitted to the client. It is wise to separate screen image into text and non-text blocks. The non-text block then continues to be divided into still and motion part. Video often changes every frame while background, text rarely changes in an application. Our goal is defining two kinds of blocks accurately and simply, and then finds out the moving block. "Text" block includes pure text or text on the image. The text has almost horizontal and vertical edges as its attribute. That is the reason why we choose edge detection is a good method to recognize the text. And another important problem is applying which lossy coding scheme for this block. The more encoder is effect, the stronger system becomes. This block is handled easily that may be applicable many loss-less compression standards to this kind of the block.

## 3.2 Block classification based on Sobel operator edge detection

Edge feature is a good parameter to divide image into picture and text blocks. Text is commonly characterized by having high-contrast mostly in horizontal and vertical. The idea of edge detection is calculated in opposite of gradient of the image intensity function of each point. First we detect the short linear edge and then aggregate it into the extended edge. Sobel [17] is a usual edge detection operator in image processing because it is very simple. Technically, the operator computes the gradient of the image intensity at each point, which gives the direction of the largest increase from light to dark pixel area and the rate of change in that direction. In other words, gradient shows possibility to become an edge of the image. In mathematically, the gradient at each image point is computed separately for two-variable functions (for two directions), a 2D vector with the components given by the derivatives in the horizontal and vertical directions. Each direction uses one kernel which is convolved with original image to compute approximation of derivatives. Gx and Gy were represented those computations that is shown in Fig. 3. Clearly, Sobel is designed for detect edge both directions horizontal and vertical.

$$Gx = Gx\_kernel \times A \tag{1}$$

$$Gy = Gy\_kernel \times A \tag{2}$$

**Fig. 3** Kernel 3*3 Gx, Gy of Sobel operator

| | | |
|---|---|---|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

Gx

| | | |
|---|---|---|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Gy

Here, symbol * denotes 2-dimensional convolution operation. A is defined as the source image. The gradient magnitude is given by:

$$\sqrt{Gx = Gx^2 + Gy^2} \tag{3}$$

Although typically, an approximate magnitude is computed using: $|G| = |Gx| + |Gy|$, that is easy to compute.

Another edge detection methods based on gradient are suggested by many researcher [2]. While Robert operator in is lower computation and compute quickly which uses two kernels 2*2 but it responds maximally to edges running at 45 degree that is not suitable for detecting text. Canny operator [2] is known as optimal edge detection with higher smooth and eliminating the noise but more complexity computation and difficult to implement. Besides, there are methods to detect text and non-text blocks such as color-based, DjVu, TCL-based that we compare performance in the experimental results in the next section. For that analysis, Sobel operator looks best choice to solve our problem.

The result of Sobel operator is shown in Fig. 4. The original image in the top and output scanned image by operator in the bottom. Generally, all pixel of picture/background changed into black pixels while other areas become white pixels.

**Fig. 4** Edge detection which applied Sobel operator

However, with the real application, each region of application consists of many blocks. Almost neighbor blocks have same type. For that reason, we can reduce the computation to determine if the block is text or non-text. The simple way, we can compute gradient of Sobel for 1/4 next block, then compute intensity and compare with new threshold, to confirm this block is similar with previous neighbor blocks.

**Algorithm 1** Classification based on adaptive Sobel

1: Divide image into non-overlapping 16*16 blocks
2: Set threshold T for text block
3: Apply Sobel operator for block
4: Compute intensity in each block

$$I(i, j) = \frac{number\_white\_pixel}{sizeofblock} \tag{4}$$

5: Compare I(i, j) with threshold T in order to determine which kind of block

$$Type\_Block = \begin{cases} 1 & \text{if } I(i, j) > T; /textblock \\ 0 & \text{if } I(i, j) < T; /non-textblock \end{cases} \tag{5}$$

6: Reduce Sobel's computation for neighbor block if it has the same type
7: Repeat step 3 through step 6 for all blocks of input image
8: Bind same types block into the group.

I(i, j) denotes the intensity of block(i, j) which is computed by the number of white pixels divide total number pixels that equals size of block(16*16). If I(i, j) is greater than threshold T, it is determined text block, otherwise it belongs to non-text block. We can use English documents for experience to define threshold T for text block. At the end, we will bind the same types of block into groups: text and non-text regions. These regions will be sent to the client, not block by block. The results are illustrated in the Fig. 5. The text blocks are grouped in the solid line and non-text blocks are grouped in the dotted one.

**Fig. 5** Results of our classification

## 3.3 Detection video and background

As we mentioned above, non-text blocks contain picture and background that are often video and background image. Our goal is detecting high motion and low motion part from these blocks. High motion is important region which needs to be encoded and transmitted frequently but low motion is skipped in many frames that help us save the bandwidth. We can easily retrieve these areas by some simple ways.

(a) Simple Background Subtraction

$$D(x, y) = \begin{cases} 1 & \text{if} |I_t(x, y) - B(x, y)| > \mathcal{T} \\ 0 & \text{if} |I_t(x, y) - B(x, y)| \leq \mathcal{T} \end{cases} \tag{6}$$

I(x, y) is the incoming frame and B(x, y) represents reference image which is selected from the test frames. We define threshold $\mathcal{T}$ which determines block is background or moving. D(x, y) is the result of this distinguishing. If subtraction between I(x, y) with B(x, y) is greater than threshold $\mathcal{T}$, D(x, y) is labeled "1" which means this block is movie part, otherwise, D(x, y) is labeled which means this block is background.

(b) Simple Statistical Difference
In this method, it uses the mean and standard deviation. Mean value $\mu_{xy}$ of the background image is computed based on continuous previous frames in a particular time. And threshold is the standard deviation $\sigma_{xy}$ in the same interval.

$$\mu_{xy} = \frac{1}{K} \sum_{k=0}^{K-1} I_k(x, y) \tag{7}$$

$$\sigma_{xy} = (\frac{1}{K} \sum_{k=0}^{K-1} (I_k(x, y) - \mu_{xy})^2)^{1/2} \tag{8}$$

$I_k(x, y)$ presents the $k^{th}$ frame. Then, we compare the absolute difference between the moving and background to separate them. The subtraction between incoming frame and mean value is compared with threshold $\lambda\sigma_{xy}$, $\lambda$ is predefined. If that subtraction is greater, this block is determined as moving, otherwise it is background. It is described as D(x, y) labeled as "0" or "1" in that formula as follows:

$$D(x, y) = \begin{cases} 1 & \text{if} |I_t(x, y) - \mu_{xy}| > \lambda\sigma_{xy} \\ 0 & \text{if} |I_t(x, y) - \mu_{xy}| \leq \lambda\delta_{xy} \end{cases} \tag{9}$$

Two aforementioned methods can be applied for our problem. The first method is more suitable. Because it just consider about one previous frame and current frame. In application, we discard when comparison is finished. But its problem is lack of preciseness when noise occurs in the incoming frame. If we choose the second method, we need to store many continuous frames.

## 3.4 Block for transmitting

Our goal is reduce the data transmitted as much as possible. Firstly, we bound the text part then continue to parse the non-text area for detecting the moving parts. Apply the comparison method to determine the video which often change frame by frame on that part. Focus on detecting in small part that help us reduce the computation complexity. In the interval time, we spend the computational power on that small area.

### 3.5 Coding for each block

Choosing best appropriate coding for each block is one of the most crucial problems of the system. With the support of image technology, moving block can be easily compressed with lossless format so that it can save the amount data transmission. We use the classic thin-client remote protocol RFB since it consumes less computing resource. In high-motion scenario, the MJPEG module is responsible for real-time encoding. Comparing with MPEG encoding, MJPEG compression consumes less computing resource.

## 4 Experimental results

### 4.1 Sobel edge detection with other methods

The proposed classification is implemented by MATLAB version 2010. We compare our method with some existing algorithms such as: color-based, TCL-base, DjVu, and Sobel method. Our input is the website (apple.com) which consists of text, image. Based on the output of each method, we can see the accurateness between of scanned image of detection algorithms that are shown in Fig. 6. While some text regions are seem lost with Color-base, DjVu and TCL, but Sobel presents more efficient with almost full text area than others.

We continue to test other websites such as: Yahoo, YouTube and then calculate the average of accuracy based on the following formula:

$$\sum_{i=1}^{sumblocks} \frac{accuracy\_block(i)}{sumblocks} \tag{10}$$

Where $accuracy\_block(i) = 1$ when $block(i)$ is text or non-text and exactly detected text/non-text. Otherwise, $accuracy\_block(i) = 0$ when $block(i)$ is text/non-text but detected non-text/text (not correct). Color-based method is the worst case because text block becomes blurring after classification. DjVu shows clearer than color-based on text region. Although TCL-based scheme seems to be same with Sobel operator, Sobel and our method are more precise and stable than any others. Figure 7 describes the average the accuracy between our proposed method and others.

### 4.2 Evaluation accurateness of method in different scenarios

For estimating exactly the accurateness of our method, we test some kinds of scenarios such as: text (text editing, document, news, etc.), video, image background and graphic (graph, diagram, etc.). We confirm definitely that Sobel filter is very strong on detecting text. The efficiency of method is classified level of accurateness as follow: text part with 90–95 %, graphic with 80–85 %, but image background has least accuracy about 70-75% because having some noises. Figure shows clearly this weakness of our method.

### 4.3 Time consumption with different methods

Time consumption is important factor of the performance of the system. Figure 8 illustrates the time evaluation between three methods. In this experiment, we conduct original image

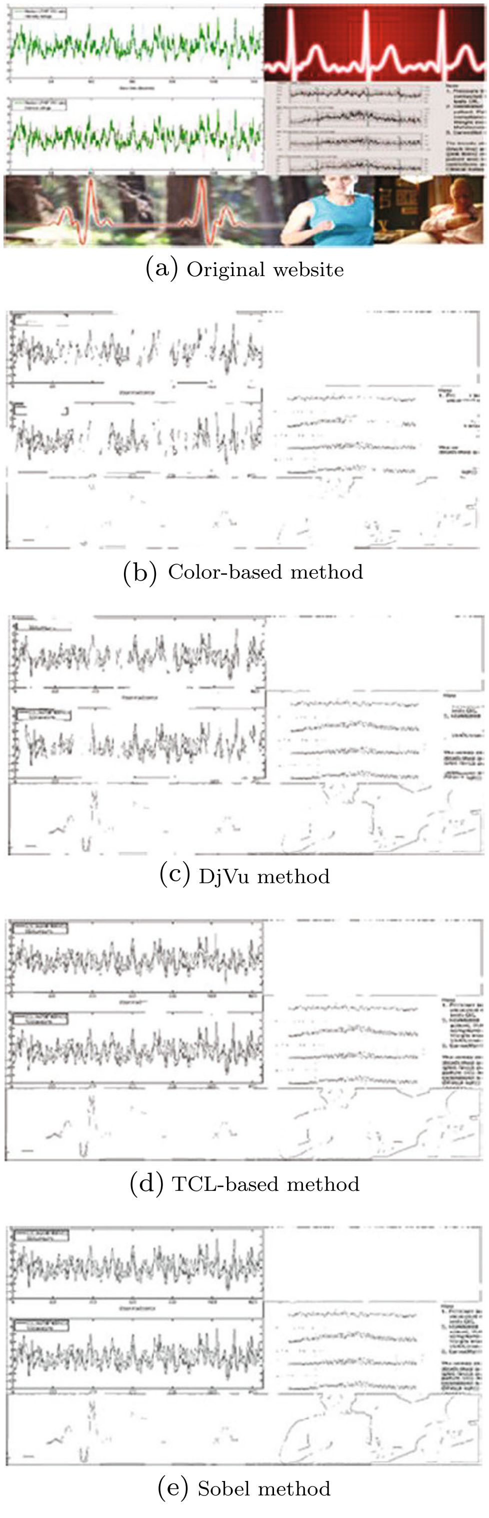


**Fig. 6** Foreground results of classification

(a) Original website

(b) Color-based method

(c) DjVu method

(d) TCL-based method

(e) Sobel method

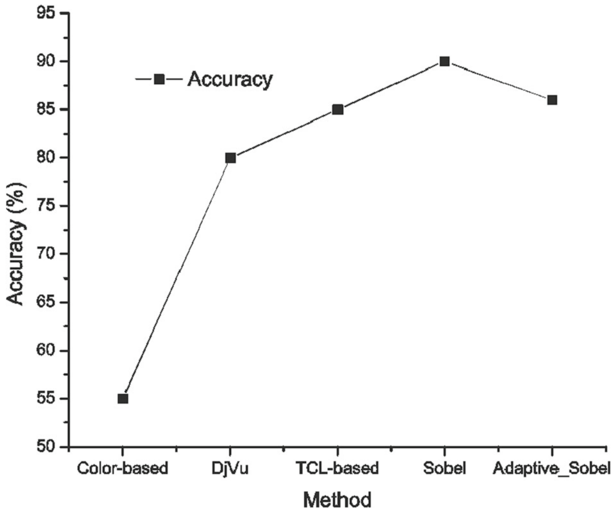



**Fig. 7** Results of our classification

including document and video, with some different size of original image by scaling them. Based on these different parameter, we compare the time consumption between color-based, Sobel and our proposed algorithm (adaptive-Sobel). Color-based and Sobel take the approximate time. But the performance of adaptive-Sobel presents about 30 % more efficient if the size of screen image is increased.

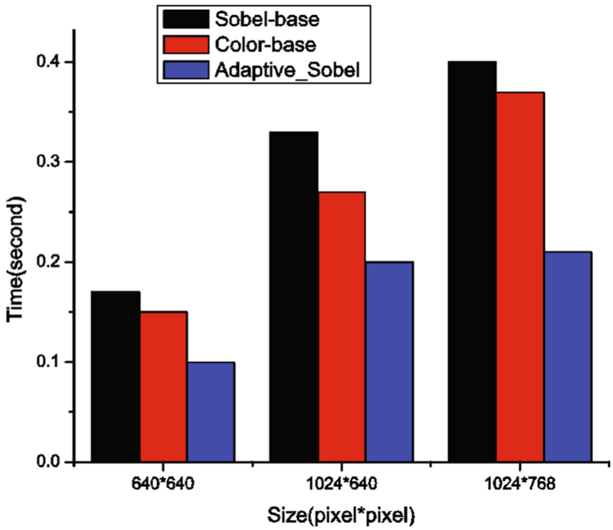



**Fig. 8** Results of comparison between Sobel and adaptive Sobel

## 5 Conclusion and future work

With the mobile healthcare service, the accurateness of diagnoses and treatment of doctor are almost dependent on qualitative vision and QoS transmission frame by frame. In this paper, we described a fixed-block adaptive compress mechanism for the different categories of contents on a specific original image. The key benefit of our proposed algorithm is the high accuracy and efficiency on classifying the screen frame into segments of text and non-text regions. From the results of the block classification, we can determine which regions of a full screen frame should be transferred to the client side, which guarantees that the amount of transmitted data is reduced along with the reasonable quality of vision at the client. Based on the experimental results, the proposed algorithm, adaptive_Sobel, is purely suitable remote computing mechanism for real time media healthcare services with low complexity computation and the high accuracy on text or non-text blocks. In the future works, we need to investigate about optimization of choosing the most proper coders for each classified region as well as deploy the adaptive_Sobel to the real remote computing environment to state its efficiency.

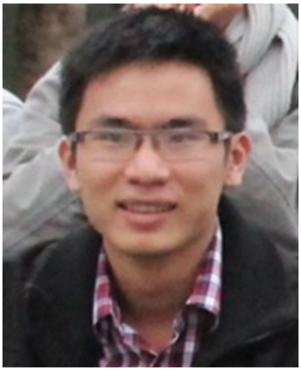
**Acknowledgements** This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-4006) supervised by the NIPA (National IT Industry Promotion Agency) and was partially supported by a grant from the Kyung Hee University in 2011 (KHU-20111209). Professor Eui-Nam Huh is corresponding author.

## References

1. Bratto RA, Nieh J, Kim L (2004) THINC: A Remote Display Architecture for Thin-Client Computing. Technical Report CUCS-02704, 2004
2. Edge detection. http://en.wikipedia.org/wiki/Edge_detection
3. Haffner P, Bottou L, Howard PG, Simard P, Bengio Y, LeCun Y (1998) High quality document image compression with DjVu. J. Electron Imaging 7(3):410–425
4. Huihui Li, Lei Guo, Hang Liu Current research on wavelet-based image fusion algorithms. In: Multisensor, multisource information fusion: architectures, algorithms, and application
5. Imai T, Horio K, Ohno T, Matsui K (2010) An adaptive desktop transfer protocol for mobile Thin Client. In: IEEE Globecom workshops, pp 1136–1140
6. Juliet SE, Florinabel DJ (2011) Efficient block prediction-based coding of computer screen images with precise block classification. Image Processing IET, pp 306–314
7. Kaplinsk KV (2001) VNC Tight Encoder Data Compression for VNC. In: Proceedings of the 7th international scientific and practical conference of students, post-graduates and young scientists on modern techniques and technologies 2001, pp 155–157
8. Keslassy I, Kalman M, Wang D, Girod B (2001) Classification of compound images based on transform coefficient likelihood. In: Proceedings of international conference on image processing, vol 1. pp 750–753
9. Kumar K, Lu Y-H (2010) Cloud computing for mobile users: can offloading computation save energy? Computer 2010, pp 51–56

10. Lin T, Hao P (2005) Compound image compression for real-time computer screen image transmission. In: IEEE transactions on image processing, pp 993–1055
11. Lu Y, Li S, Shen H (2011) Virtualized Screen: A Third Element for Cloud-Mobile Convergence. Multimedia, IEEE
12. Richardson T, Stafford-Fraser Q, Wood KR, Hopper A (1998) Virtual network computing. Internet Comput. IEEE 2(1):33, 38
13. Said A (2008) Compression of compound images and video for enabling rich media in embedded systems. In: SPIE proceeding of visual communication on image processing Conference, 2004, vol 5308. pp 69–82
14. Simoens P, Praet P, Vankeirsbilck B, De Wachter J, Deboosere L, De Turck F, Dhoedt B, Demeester P (2008) Design and implementation of a hybrid remote display protocol to optimize multimedia experience on thin client devices. In: Telecommunication networks and applications conference, Australasian, 2008, pp 391–396
15. Simoens P, Turck FD, Dhoedt B, Demeester P (2011) Remote Display Solutions for Mobile Cloud Computing. In: IEEE computer society 2011
16. Shen H, Lu Y, Wu F, Li S (2003) A high-performance remote computing platform. In: Pervasive computing and communications, 9–13 March 2009, pp1–6
17. Sobel operator edge detection. http://en.wikipedia.org/wiki/Sobel_operator
18. Tan KJ, Gong JW, Wu BT, Chang DC, Li HY, Hsiao YM, Chen YC, Lo S-W, Chu YS, Guo JI (2010) A remote thin client system for real time multimedia streaming over VNC. In: Multimedia and expo (ICME), 19–23 July 2010, pp 992–997
19. Teradici PCoIP technology. http://www.teradici.com
20. Tizon N, Moreno C, Cernea M, Preda M (2011) MPEG-4-based adaptive remote rendering for video games. In: Proceedings of the 16th international conference on 3D web technology, NY
21. Windows Remote Desktop Protocol (RDP). http://msdn.microsoft.com/en-us/library/windows/desktop/aa.383015.aspx
22. Yang C, Niu Y, Xia Y, Xu CA (2007) Fast and Efficient for Multimedia Applications in Wireless Thin-Client Computing. In: IEEE international symposium on a world of wireless. Mobile and multimedia networks, pp 1–12


**An Thuy Nguyen** received her BS from HCM University of Science, and MS from Department of Computer Engineering, Kyung Hee University in 2013. Her research interests include Cloud computing, Mobile Cloud, Healthcare services, etc.



**Cong-Thinh Huynh** received his BS from HCM University of Telecommunication and is studying MS and PhD integrated course at Department of Computer Engineering, Kyung Hee University since 2013. His research interests include Cloud computing, Media Cloud, Thin-Client Computing, etc.

**ByungKwan Lee** received his B.S. degree from Pusan National University in 1979, his M.S. degree in Computer Science from Chung University in 1986, and his Ph.D. degree in Computer Science from Chungang University in 1990. He has been on the faculty of the department of Computer Engineering, Kwandong University, Korea since 1988. He had been a Visiting Professor, in Saginaw Valley state University, Michigan, USA for two years since 2000. He is a permanent member of the KISS, KIPS, and KIIS. He is currently interested in Network Security, IoT, and Big Data.

**Choong Seon Hong** is a professor of computer engineering at Kyung Hee University. His research interests include ad hoc networks, Internet services and management, cognitive radio networks, network security, and the future Internet. Hong has a PhD in information and computer science from Keio University. Hes a senior member of IEEE, and a member of ACM, the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Korean Institute of Information Scientists and Engineers (KIISE), the Korean Institute of Communications and Information Sciences (KICS), and the Korea Information Processing Society (KIPS).

**Eui-Nam Huh** is a professor in Deparment of Computer Engineering. His interesting research areas are: Cloud Computing, Ubiquitous Computing, High Performance Network, Sensor Network, Distributed Real Time System, Grid, Network Security. He earced Masters degree in Computer Science from University of Texas, USA in 1995 and Ph. D degree from the Ohio University, USA in 2002. He has also served for the WPDRTS/IPDPS community as program chair in 2003. He has been an editor of Journal of Korean Society for Internet Information and Korea Grid Standard group chair since 2002. He was also an Assistant Professor in Seoul Womens University, South Korea.