GPU Based High Speed Error Protection for Watermarked Medical Image Transmission

Md Shohidul Islam, Jongmyon Kim, Ui-pil Chong

***Abstract*—**Medical image is an integral part of e-health care and e-diagnosis system. Medical image watermarking is widely used to protect patients' information from malicious alteration and manipulation. The watermarked medical images are transmitted over the internet among patients, primary and referred physicians. The images are highly prone to corruption in the wireless transmission medium due to various noises, deflection, and refractions. Distortion in the received images leads to faulty watermark detection and inappropriate disease diagnosis. To address the issue, this paper utilizes error correction code (ECC) with (8, 4) Hamming code in an existing watermarking system. In addition, we implement the high complex ECC on a graphics processing units (GPU) to accelerate and support real-time requirement. Experimental results show that GPU achieves considerable speedup over the sequential CPU implementation, while maintaining 100% ECC efficiency.

***Keywords*—**Medical Image Watermarking (MIW), e-health system, error correction, Hamming code, GPU.

I. Introduction

MEDICAL image watermarking is a means of security service, widely used in the e-diagnosis system [1], [2]. Security of medical information derives from strict ethics and enacting rules that dictate three major characteristics: confidentiality, reliability, and availability. Confidentiality means that only the authorized person can access to the medical information. Reliability has two aspects: integrity and authenticity. Integrity means that the information should not be modified by any non-authorized people. Authenticity means that the information should be originated from the correct source [3], [4]. Availability means that entitled users have the ability to use the information in the normal condition of access. There are three main applications for inserting the watermark in the medical domain: 1) data hiding to facilitate the data management, 2) security control to verify that the image is intact, and 3) authenticity to prove that the image is really what the user supposes it is [5], [6].

In such health care information systems, medical images are sent by a patient over the internet to a primary physician and the primary physician can also transfer the images to anther physician [7]-[10]. These images are sometimes become corrupted and erroneous due to the noise in wireless medium and lead to faulty watermark detection and faulty diagnosis of disease. Thus, guaranteeing reliable and consistent transmission of watermarked medical images is an important issue, which can be achieved by applying error correction code (ECC).This paper implements (8, 4) Hamming code as ECC on the watermarked medical image for encoding and decoding.

Furthermore, to achieve real time service, the ECC systems is implemented on graphics processing unit (GPU). The proposed approach is validated using a Compute Unified Device Architecture (CUDA) [11], [12] enabled NVIDIA GeForce GTX 560 graphics card.

The remainder of this paper is organized as follows. Section II explains the proposed approach. Section III presents the (8, 4) Hamming code, Section IV discusses about the implementation and experimental results and finally Section V presents our conclusions.

Md Shohidul Islam, Jongmyon Kim (corresponding author), and Ui-pil Chong are with the department of Computer Engineering, University of Ulsan, South Korea (e-mail: shohid@mail.ulsan.ac.kr, {jmkim07, upchong}@ulsan.ac.kr).

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. NRF-2013R1A2A2A05004566).

II. Proposed Approach

Medical image watermarking serves the vital purpose information security in e-diagnosis system. The typical e-health model shown in Fig. 1, medical images can be sent by a patient over the internet to a primary physician and the primary physician can also transfer the images to anther physician. The medical images are then stored in the patient historical database for future diagnosis. Fig. 2 shows the watermarked magnetic resonance imaging (MRI) brain images at the sender end and Fig. 3 shows that at the receiver end. Received images are distorted due the pixel corruption in the transmission medium by noise factors.

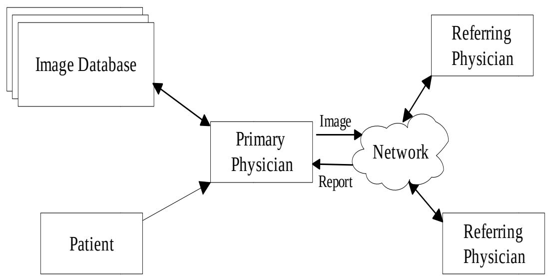

Fig. 1 Typical e-health model

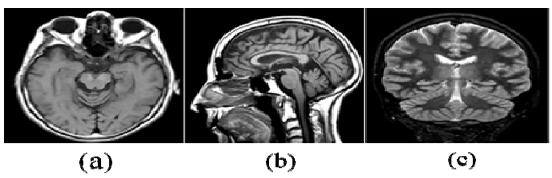

Fig. 2 Watermarked MRI brain images at the sender

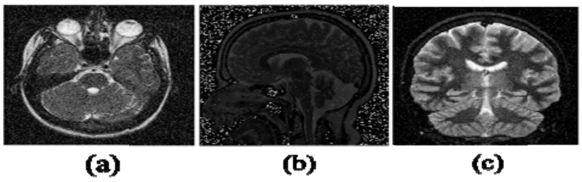

Fig. 3 Watermarked MRI brain images at the receiver

Image pixel corruption is protected by applying error correction code (ECC) that uses (8, 4) Hamming code. The encoding and decoding of (8, 4) Hamming code [13], [14] is briefly explained in Section III.

III. (8, 4) Hamming Code

A. ECC Encoding

Encoding procedure extends the original data by adding additional information to it before transmission. Encoding a gray image with (8, 4) hamming code means that each 8 bit pixel is extended to 12 bits by adding 4 redundant bits. Fig. 4 shows such encoding where **P** is the original pixel with 8 bits (*b*8 *b*7 *b*6 *b*5 *b*4 *b*3 *b*2 *b*1) and **H** is the encoded result.

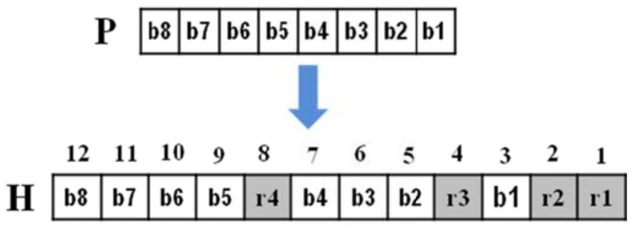

Fig. 4 Encoding of a pixel by (8, 4) Hamming Code

The redundant bits, *r*1, *r*2, *r*3, and *r*4 are calculated by (1) through (4).

$$r_1 = H_3 \oplus H_5 \oplus H_7 \oplus H_9 \oplus H_{11} \quad (1)$$

$$r_2 = H_3 \oplus H_6 \oplus H_7 \oplus H_{10} \oplus H_{11} \quad (2)$$

$$r_3 = H_5 \oplus H_6 \oplus H_7 \oplus H_{12} \quad (3)$$

$$r_4 = H_9 \oplus H_{10} \oplus H_{11} \oplus H_{12} \quad (4)$$

where, $\mathbf{H}_i$ represents *i*-th bit of **H** vector.

B. ECC Decoding

Decoding is the exact reverse process of encoding. For the error detection, same redundant bits are calculated and they are called checksum bits, *c*1, *c*2, *c*3, and *c*4. Calculations of checksum bits are through (1) to (4) Received checksums are compared with the sending checksum and error is detected by any mismatch.

IV. Implementation and Results

Error correction explained in Section III is performed on tradition CPU that requires much time for processing large size image due to sequential processing. It cannot meet real time requirement because of the long time to accomplish. The ECC can be further speedup on graphics processing unit (GPU). GPU is a parallel processing platform that employs hundreds of processors concurrently. GPU can accelerate the execution of regular image processing to a greater extent compared to CPU exploiting the architectural advantages as shown in Fig. 5.

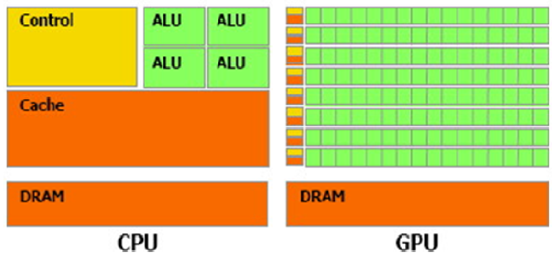

Fig. 5 CPU and GPU architecture [15]

A. Experimental Results

Experiment is conducted from two different points of views (i) implementing the ECC on CPU (ii) further speed up by using GPU to achieve real time service. The experiment is accomplished on CPU and GPU whose system specifications are given in Table I.

TABLE I
Experimental Environment

Property	CPU	Property	GPU
Processor	Intel(R) Core(TM) i5-3570K	Brand name	NVIDIA GeForce GTX 560
Clock speed	3.40GHz	Processor clock	1620MHz
No. of Cores	4	CUDA core	336
No. of threads	4	Total MP	7
RAM	8.00GB	Max thread per block	1024
Bus/Core ratio	34	Shared memory per MP	49152 Byte
Operating system	Windows 7, 32 bit	Total global memory	1 GB

Table II shows the execution time on CPU and GPU for error correction on different images, such as 32×32, 64×64, 128×128, 256×256, 512×512, and 1024×1024. As can be seen, GPU takes less than 33.33 ms for any image, which supports that GPU can achieve real time service. Also, GPU gains tremendous speed up over the CPU.

TABLE II
Performance on CPU and GPU

Image size	CPU time (ms)	GPU time (ms)	Speedup
32×32	26.670958	0.2102	127×
64×64	106.688362	0.3404	313×
128×128	424.357909	0.3617	1173×
256×256	1728.395479	1.0996	1572×
512×512	6831.579907	3.8686	1766×
1024×1204	27224.57633	14.4927	1878×

V. Conclusions

This paper proposed (8, 4) Hamming coding to securely transmit the watermarked medical image in the e-health system. Moreover, GPU is used for faster error correction that can guarantee real time service. GPU exceedingly outperforms the CPU based error correction yielding tremendous speedup.

References

[1] N. A. Memon, S. A. M. Gilani, S. Qayoom, "Multiple Watermarking of Medical Images for Content Authentication and Recovery," IEEE 13th International Multitopic Conference, 2009, pp.1-6, Dec. 14-15, 2009.

[2] A. Al-Gindy, "A Fragile Invertible Watermarking Technique for the Authentication of Medical Images," 2010 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp.191-195, Dec. 15-18, 2010.

[3] G. Coatrieux, Hui Huang, Huazhong Shu, Limin Luo, C. Roux, "A Watermarking-Based Medical Image Integrity Control System and an Image Moment Signature for Tampering Characterization," IEEE Journal of Biomedical and Health Informatics, vol.17, no.6, pp.1057-1067, Nov. 2013.

[4] K. Pushpala, R. Nigudkar, "A Novel Watermarking Technique for Medical Image Authentication," IEEE International Conference on Computers in Cardiology, pp.683-686, Sep. 25-28, 2005.

[5] B. W. R. Agung, Adiwijaya, F. P. Permana, "Medical Image Watermarking with Tamper Detection and Recovery Using Reversible Watermarking with LSB Modification and Run Length Encoding (RLE) Compression," 2012 IEEE International Conference on Communication, Networks and Satellite (ComNetSat), pp.167-171, Jul. 12-14, 2012.

[6] Adiwijaya, P. N. Faoziyah, F. P. Permana, T. A. B. Wirayuda, U. N. Wisesty, "Tamper Detection and Recovery of Medical Image Watermarking Using Modified LSB and Huffman Compression," 2013 Second International Conference onInformatics and Applications (ICIA), pp.129-132, Sep. 23-25 , 2013.

[7] Jingbing Li, Yaoli Liu, Wencai Du, Yen-Wei Chen, "The Medical Image Watermarking Algorithm Based On DFT and Logistic Map," 2012 7th International Conference on Computing and Convergence Technology (ICCCT), pp.1-6, Dec.3-5, 2012.

[8] G. Coatrieux, Hui Huang, Huazhong Shu, Limin Luo, C. Roux, "A Watermarking-Based Medical Image Integrity Control System and an Image Moment Signature for Tampering Characterization," IEEE Journal of Biomedical and Health Informatics, vol.17, no.6, pp.1057-1067, Nov. 2013.

[9] J. Sanders, and E. Kandrot. (2010, Jul. 29). CUDA by Example: An Introduction to General-Purpose GPU Programming. (1st edition). *[On-line]. Available: http://www.amazon.com/CUDA-Example-Introduction-General-Purpose-Programming/dp/0131387685*

[10] D. B. Kirk and W. W. Hwu. (2012, Dec. 28). Programming Massively Parallel Processors: A Hands-on Approach. (2ndedition). *[On-line]. Available: http://www.amazon.com/Programming-Massively-Parallel-Processors-Edition/dp/0124159923/ref=dp_ob_title_bk*

[11] R. W. Hamming, “Error Detecting and Error Correcting Codes,” *The Bell System Technical Journal*, vol. 26, no. 2, pp. 147–160, 1950.

[12] R.Ma, S. Cheng, "The Universality of Generalized Hamming Code for Multiple Sources," *IEEE Transactions on Communications*, vol.59, no.10, pp.2641-2647, Oct. 2011 .

[13] http://docs.nvidia.com/cuda/cuda-c-programming-guide/

[14] Y. Liu, L. Guo, J. Li, M. Ren, and K. Li, "Parallel Algorithms for Approximate String Matching with k Mismatches on CUDA," in Proc. 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), pp. 2414-2422, May 2012.

[15] http://docs.nvidia.com/cuda/cuda-c-programming-guide/