JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 8, NO. 12, DECEMBER 2005(pp. 1649-1657)

A Fast Implementation of JPEG and Its Application to Multimedia Service in Mobile Handset

Gu-Min Jeong†, Doo-Hee Jung††, Seung-Won Na†††, Yang-Sun Lee††††

ABSTRACT

In this paper, a fast implementation of JPEG is discussed and its application to multimedia service is presented for mobile wireless internet. A fast JPEG player is developed based on several fast algorithms for mobile handset. In the color transformation, RCT is adopted instead of ICT for JPEG source. For the most time-consuming DCT part, the binDCT can reduce the decoding time. In upsampling and RGB conversion, the transformation from YCbCr to RGB 16 bit is made at one time. In some parts, assembly language is applied for high-speed. Also, an implementation of multimedia in mobile handset is described using MJPEG (Motion JPEG) and QCELP(Qualcomm Code Excited Linear Prediction Coding). MJPEG and QCELP are used for video and sound, which are synchronized in handset. For the play of MJPEG, the decoder is implemented as a S/W upon the MSM 5500 baseband chip using the fast JPEG decoder. For the play of QCELP, the embedded QCELP player in handset is used. The implemented multimedia player has a fast speed preserving the image quality.

Keywords: JPEG, Fast implementation, Multimedia, Mobile handset

1. INTRODUCTION

The development of wireless network and mobile handset makes it possible to implement various multimedia solutions in handset. 3GPP and 3GPP2 make a standard of multimedia codecs and recommend the service providers to apply them. However, several limitations of mobile handset make it difficult to utilize the multimedia codecs. In addition, the characteristics of multimedia in mobile handsets are different from those in PC. Main characteristics of handset are low processing power, limited memory, and narrow network bandwidth, which should be considered in the mobile handset services.

Especially, in mobile handset, the speed of codec must be fast and the compressed size must be small. Nowadays, one of the most popular multimedia codec is MPEG. However, DSP chip is essential since it is difficult to implement MPEG as a S/W in mobile handset. In this paper, an implementation of a multimedia using QCELP [1] and MJPEG is presented. MJPEG extends JPEG [2] by supporting videos and each frame in the video is stored with the JPEG format. For the play of MJPEG, it is required to implement fast JPEG decoder. QCELP is a vocoder which is widely used for a voice call in CDMA network. It is embedded in CDMA handset where playing sound is possible using QCELP if API's are provided to application. One can implement a multimedia using QCELP and MJPEG without any HW. In this paper, the multi-

※ Corresponding Author : Gu-Min Jeong, Address : (136-702) 30 861-1, JeongNung-Dong, SeongBuk-Ku, Seoul, TEL : +82-2-910-4408, FAX : +82-2-910-4449, E-mail : gm1004@kookmin.ac.kr
Receipt date : Mar. 25, 2005, Approval date : Sep. 9, 2005
† School of Electrical Engineering., Kookmin University.
†† Dept. of Electrical and Electronics Engineering, Korea Polytechnic University.
(E-mail : doohee@kpu.ac.kr)
††† Terminal Development Team, SK Telecom.
(E-mail : nasw@dgu.ac.kr)
†††† Department of Computer Engineering, SeoKyeong University.
(E-mail : yslee@skuniv.ac.kr)

media player is ported on BREW environment [3] in a CDMA handset with MSM 5500 baseband chipset. The fast JPEG decoder is developed using IJG open source [4]. QCELP player is made out using KTF extension in BREW. Comparing with the fast QCELP API which is processed on DSP, JPEG is implemented as a S/W and can be slow.

In this paper, fast JPEG player is developed based on several fast algorithms considering the characteristics of mobile handset. In some time-consuming parts, assembly language is used for high speed. The implemented codec has a high speed preserving the image quality and it makes wireless multimedia service possible in mobile handset. It has a performance of 4-5 frames per second for 128×96 images and similar quality comparing to conventional JPEG player. Also, if the well-known R-D optimization scheme [5] or its variation for high speed [6] are used, the proposed codec can have more improved performance. An implementation of a multimedia using QCELP and MJPEG is presented using the proposed codec. The remainder of this paper is organized as follows. In Section 2, the MJPEG and QCELP are briefly summarized. Also the test environment is described. In Section 3, the fast implementation methods of JPEG decoder are presented. In Section 4, implementation results are presented and the conclusion follows in Section 5.

2. MJPEG, QCELP AND SYSTEM SETTINGS

The methods for porting multimedia player in handset can be divided into 3 categories. The first one is using S/W player that runs over baseband chipset. It is used for low end handset or usual handset. The next one is using ASIC chip. Some JPEG players in camera module or some midi players are applied as a HW ASIC chip. The last one is using application CPU and DSP. The multimedia solutions like MPEG or 3D are developed on those environments. It should be noted that if there is another HW chipset, the handset becomes expensive. In this paper, the player is ported on the baseband chipset for all of the handsets. Baseband chipset mainly performs the wireless communications. Full use of the baseband chipset cannot be achieved for the application such as multimedia player. If the player needs to be implemented on the baseband chipset, fast player is essential. The fast player proposed in this paper also can be used in ASIC or DSP. In mobile multimedia services, the handset must have functions of decoder or player. The encoder may be in PC, server or handset. In some services such as MMS, the encoder is in handset. However in most of services, the contents are encoded in PC or server and played in handset. In this paper, we propose a design for encoder and decoder. But the implementation in handset is done only for the decoder part.

2.1 MJPEG

MJPEG is a video format extending JPEG [2]. It has only intraframe coding JPEG without interframe coding. Thus the code structure is very simple, and it is easier to port the decoder than MPEG decoder. Fast implementation of JPEG is needed for fast MJPEG player. As seen in Fig. 1, JPEG decoder can be divided into several parts

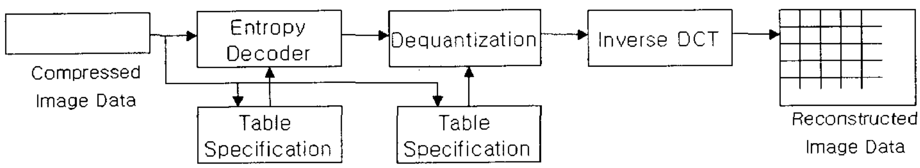

Fig. 1. The process of JPEG baseline decoding.

such as entropy coding, dequantization, IDCT and color space transformation. It is important to optimize each part for fast decoder. Especially, the IDCT part is most time-consuming and the display part for handset LCD is also time-consuming.

In this paper, IJG open source is used for the JPEG decoder. Fast JPEG decoder is developed on BREW and MJPEG decoder is implemented with QCELP player.

2.2 QCELP

For the voice communication, vocoder is developed and used in handset. In CDMA, there are two vocoders, EVRC and QCELP [1]. QCELP is one of the CELP vocoder developed by Qualcomm. QCELP 8 kbps and QCELP 13 kbps are used respectively. QCELP 13 kbps is generally used for the voice quality. QCELP uses 20 ms as the data frame size. Each frame may be encoded at one of four different data rates: full rate, 1/2 rate, 1/4 rate, and 1/8 rate and variable rate is also supported. Vocoder is essential for the voice communication. For the high compression, complex algorithms are used in vocoder and it is implemented on DSP. If the vocoder API's are provided for applications, one can play the vocoder data. In this paper, we implement the QCELP player using QCELP extension API's over the BREW environment. QCELP is played on DSP and the decoder speed is very fast. So, for the implementation of MJPEG and QCELP, it is important to implement fast JPEG decoder. Since QCELP uses 20ms as the data frame size, it is good to streaming or synchronization.

2.3 System settings

CDMA handset with MSM 5500 is used in this paper. MSM 5500 has ARM 7 core and the CPU clock is up to about 40 MHz. The multimedia player is developed and runs over BREW platform. LCD size is 128×160 and the bit depth of LCD is 16 bit. Target speed of JPEG is about 3-5 fps. Table 1 shows the system settings in this paper.

3. FAST IMPLEMENTATION OF JPEG

3.1 JPEG porting in handset using IJG source

The JPEG can be easily ported using IJG open source. Actually, the decoding time depends on CPU speed, the memory access time or other environment. In our porting of IJG JPEG decoder

Table 1. Test environment.

Settings	
Network	CDMA 1x EVDO
Baseband Chipset	MSM 5500 (ARM 7 core)
Image	MJPEG
Sound	QCELP
Image size	128×96
LCD size	128×160
LCD bit depth	16 bit
Memory size	1M bytes
Heap memory size	200 k bytes
Porting environment	Qualcomm BREW
Target FPS	3-5 fps

Table 2. Ratio of required time in JPEG decoding.

Part	Decoding Time (ms)	Ratio(%)
IDCT	84	29
Decoding MCU	58	20
Upsampling and RGB conversion	84	29
The rest	64	22
Total	290	100

based on the system setting of Table 1, it takes about 300 ms to decode 1 frame whose size is 128×96. To implement fast decoder, the analysis of decoding time is shown in Table 2. Table 2 shows the time ratio required for IJG JPEG decoding in handset with MSM 5500. In this test, quality factor is set to 75 and Y:Cb:Cr is set to 4:2:0. As shown in Table 2, the portion of IDCT is about 30 %.

3.2 Main direction of codec design

As discussed before, the fast decoder with preserving image quality is required for mobile handset. Several parts in Table 2 are implemented by use of fast algorithms. In the color transformation, RCT is adopted instead of ICT. For the most time-consuming DCT part, binDCT can reduce the decoding time. In upsampling and RGB conversion, the transformation from YcbCr to RGB 16 bit is made at one time. In some parts, assembly language is applied. Fig. 2 shows the total structure of the decoder.

3.3 IDCT part

DCT has been adopted in many standards such as JPEG, MPEG and H.26x. Though DCT is good for compression, it requires a great amount of computation. Thus various fast algorithms have been proposed. Especially, Arai et. al. [7] have proposed the fast algorithm for DCT, which is adopted in IJG JPEG open source. In ARM [8], multiplication processing time is 4 times larger than that of addition (multiplication 8 clock,

Table 3. Forward transform matrix of binDCT C7.

1	1	1	1	1	1	1	1
15/16	101/128	35/64	1/4	-1/4	-35/64	-101/128	-15/16
3/4	1/2	-1/2	-3/4	-3/4	-1/2	1/2	3/4
1/2	3/32	-11/16	-1/2	1/2	11/16	-3/32	-1/2
1/2	-1/2	-1/2	1/2	1/2	-1/2	-1/2	1/2
1	-23/16	-1/8	1	-1	1/8	23/16	-1
1/2	-1	1	-1/2	-1/2	1	-1	1/2
1/4	-21/32	13/16	-1	1	-13/16	21/32	-1/4

Table 4. Inverse transform matrix of binDCT C7.

1/2	1	1	1	1	1/2	1/2	1/4
1/2	13/16	1/2	1/8	-1	-11/16	-3/4	-35/64
1/2	21/32	-1/2	-23/16	-1	-3/32	3/4	101/128
1/2	1/4	-1	-1	1	1/2	-1/2	-15/16
1/2	-1/4	-1	1	1	-1/2	-1/2	15/16
1/2	-21/32	-1/2	23/16	-1	3/32	3/4	-101/128
1/2	-13/16	1/2	-1/8	-1	11/16	-3/4	35/64
1/2	-1	1	-1	1	-1/2	1/2	-1/4

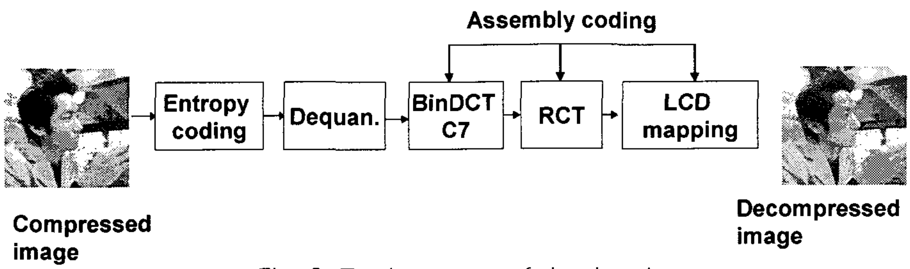

Fig. 2. Total structure of the decoder.

addition 2 clock, shift 2 clock). If we use multiplierless fast algorithms, the decoding time can be reduced. Recently, Liang and Tran [9] have proposed fast multiplierless DCT algorithms. The proposed DCT algorithm, binDCT, approximates the DCT with shift and addition using lifting structure. There is a tradeoff between image quality and speed for the approximation. Through several tests from binDCT C1 to binDCT C9, we select binDCT C7 for this implementation. binDCT C7 has 9 shifts and 28 additions preserving coding gain. Table 3 and Table 4 show the transform matrices of binDCT C7.

3.4 Color Transformation

In JPEG, color space conversion transforms RGB pixels into YCbCr. In IJG source [4], ICT (Irreversible Component Transformation) has been implemented. The forward ICT is defined as

$$\begin{aligned} Y &= 0.299R+0.587G+0.114B, \\ Cb &= -0.16875R-0.33126G+0.5B, \\ Cr &= 0.5R-0.41869G-0.08131B. \end{aligned} \quad (1)$$

There is another transformation called RCT (Reversible Component Transformation) [10]. RCT approximates the ICT and the implementation can be done using only shift and addition. The forward and inverse RCT is defined as

$$\begin{aligned} &Y' = 0.25R+0.5G+0.5B,\ C'r = R-G,\ C'b = B-G \\ &G = Y'-0.25C'b-0.25C'r,\ R = C'r+G,\ B = C'b+G \end{aligned} \quad (2)$$

Using RCT, the color transformation can be made faster than ICT.

3.5 LCD mapping

The test handset has 16 bit LCD. Many handsets have 16 bit LCD since it is easier to handle the memory structure with 2 bytes. In JPEG, 24 bit display is supported. When using 24 bit in handset, the decoder consumes 2 times bigger memory than 16 bit LCD. As image quality degradation in 16 bit is not so big, 16 bit LCD is widely applied. The LCD in test handset has 5, 6 and 5 bits for RGB respectively. In JPEG decoding and display, color transform should be made from YCbCr 24 bit to RGB 24 bit and next RGB 24 bit should be converted to 16 bit as in Fig. 3. These all processes are also time-consuming. In this paper, the whole transformation is made at once by one function. As

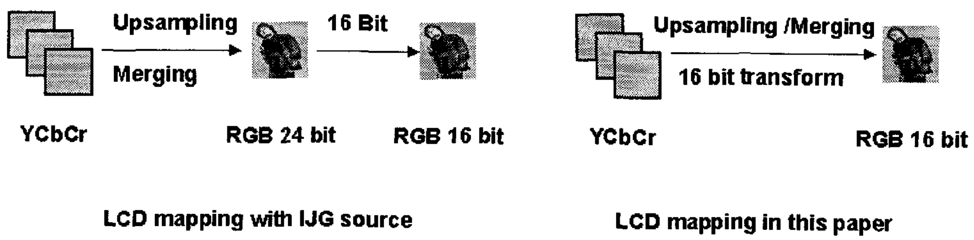

Fig. 3. LCD mapping.

Table 5. Decoding time with the proposed implementation with C.

Part	Decoding Time (ms)	Ratio(%)
IDCT	66	30
Decoding MCU	77	36
Upsampling and RGB conversion	38	18
The rest	34	16
Total	215	100

a result, the decoding time can be reduced.

4. IMPLEMENTATION RESULTS

4.1 JPEG decoder

Table 5 shows the decoding time of the proposed implementation in each part. Comparing to Table 2, the decoding time is 25% faster than that of Table 2. The run-time memory size required for the codec is smaller than 80K bytes (25K bytes for the coder and 42K bytes for input and output buffer). The coder size is about 42K bytes in case of C implementation and is about 41K bytes if some parts are implemented using assembly.

If assembly language is applied for certain parts, the decoding time can be faster. Table 6 implies the decoding time with assembly programming for IDCT, upsampling and RGB transformation. As in Table 6, the total decoding time is 40 % faster than the original IJG source porting. Using this implementation JPEG can be decoded with the speed 4~5 fps. In this paper, Decoding MCU part has not been changed. If some algorithms such as [11] are applied, better results may be achieved.

To check the image quality and size, we use the images in Fig. 4 with size 128×128. Table 7 shows the image quality and bpp for IJG source and the proposed method. The quality is measured by

Fig. 4. Test images (source images).

PSNR such as $PSNR(dB) = 10\log_{10}\frac{255^2}{\sigma^2}$ where

$$\sigma^2 = \frac{1}{3N}\sum_{1}^{N}\{(R_a - R_b)^2 + (G_a - G_b)^2 + (B_a - B_b)^2\}$$

Image size in the proposed method is slightly smaller than that of IJG. In image quality, there are no significant differences. It should be noted that the decoder has a fast speed preserving the image quality and size.

4.2 Performance improvement of the designed codec

R-D optimization [5] is an algorithm to optimize rate-quality tradeoffs in an image-specific way. If

Table 6. Decoding time with the proposed implementation with C and assembly.

Part	Decoding Time (ms)	Ratio(%)
IDCT	41	23
Decoding MCU	77	42
Upsampling and RGB conversion	28	16
The rest	34	19
Total	181	100

Table 7. Image quality and size (for 128×128).

Test image	Quality (IJG)	bpp (IJG)	Quality (Proposed)	bpp (Proposed)
Image 1	21.40 dB	0.0617	21.77 dB	0.0607
Image 2	25.59 dB	0.0303	25.52 dB	0.0294
Image 3	23.07 dB	0.0510	23.20 dB	0.0502
Image 4	25.73 dB	0.0553	25.73 dB	0.0548

this scheme is adopted for the proposed codec, more improvement in image quality and rate can be obtained. Generally, it takes too much time to calculate optimized quantization table. In some applications of mobile handset, the contents is made beforehand and downloaded afterwards. In this case, R-D optimization can be utilized as it is acceptable to take more time for encoding.

If the fast encoding is essential and the images have similar characteristics, the quantization table design technique in [6] can be adopted. In [6], only photos with face are considered. For these images, the file size of the images is 10%-20% smaller than that of JPEG, preserving similar image quality.

4.3 Wireless multimedia service using MJPEG and QCELP

Fig. 5 and Fig. 6 show the encoder and the multimedia player, respectively. MJPEG player works on BREW with C code or assembly code. QCELP player use the QCELP API. QCELP is played over DSP and it does not affect JPEG decoding too much. Syncronization of sound and image is done by dividing the QCELP data based on the data rate. In this implementation, JPEG decoding and display time is smaller than 200 ms. It is possible to display the video with 4-5 fps. For 3 fps, total file size is about 300-400K bytes with

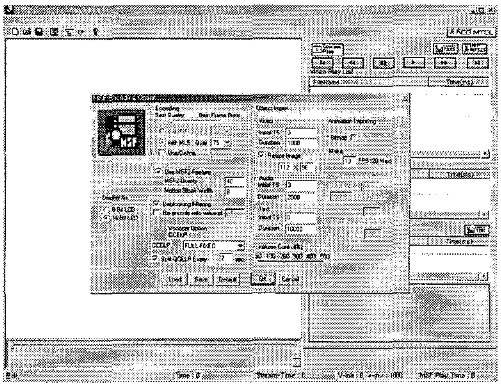

Fig. 5. Multimedia encoder.

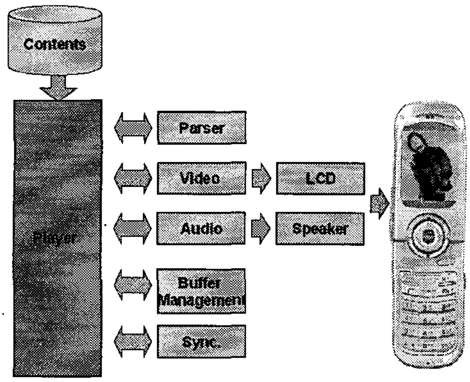

Fig. 6. The total structure of the multimedia player.

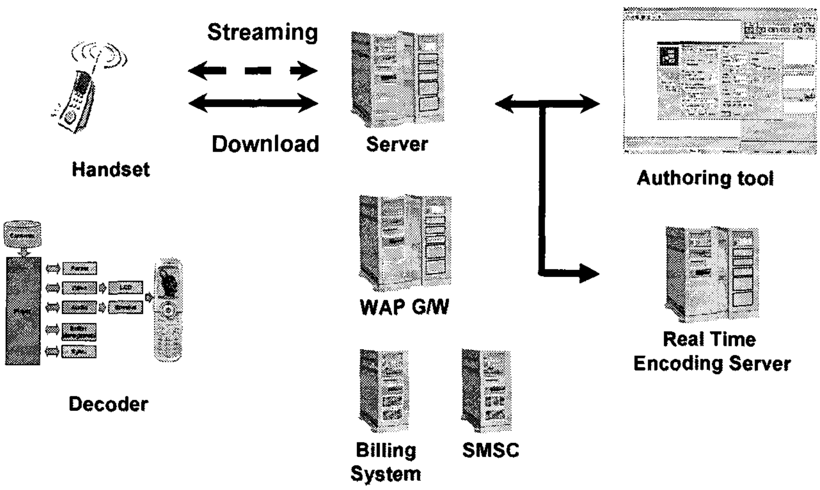

Fig. 7. Service flow.

JPEG (quality 75) and QCELP (full rate) for 1 minute.

Fig. 7 shows the total service flow. Multimedia contents can be made using authoring tool or real time encoding server. It is uploaded to the server of service provider and the user can enjoy the contents using download or streaming.

5. CONCLUSION

In this paper, a multimedia implementation has been discussed in mobile handset using MJPEG and QCELP. MJPEG has been used for an image with QCELP for a sound. Considering slow CPU in mobile handset, fast player has been developed. For each part of JPEG decoder, fast algorithms have been adopted. Some parts have been implemented using assembly code. In color transformation, RCT has been adopted instead of ICT in JPEG source. In most time-consuming DCT part, the decoding time has been reduced using binDCT. In upsampling and RGB conversion, the transformation from YCbCr to RGB 16 bit has been made at once. This implementation has been made using CDMA handset with the CPU MSM5500 and 16 bit LCD size. The implemented multimedia has a performance of 4-5 frames per second for 128×96 images. The designed decoder for JPEG has fast speed while preserving good image quality similar to JPEG. The performance of proposed design can be improved using R-D optimization or similar quantization design technique.

6. REFERENCES

[1] "Rate Speech Service Option 17 for Wideband Spread Spectrum Communication Systems," *TIA/EIA/IS-733*, March 1998.

[2] G. K. Wallace, "The JPEG still-picture compression standard," *Commun. ACM*, Vol. 34, pp. 30-44, Apr. 1991.

[3] BREW, http://brew.qualcomm.com/brew/en/

[4] Independent JPEG Group, http://www.ijg.org.

[5] M. Crouse and K. Ramchandran, "Joint thresholding and Quantizer Selection for Transform Image Coding : Entropy-Constrained Analysis and Application to Baseline JPEG," *IEEE Trans. on Image Processing*, Vol. 6, No. 2, Feb. 1997.

[6] Gu-Min Jeong, Jun-Ho Kang, Yong-Su Mun, and Doo-Hee Jung, "JPEG Quantization Table Design for Photos with Face in Wireless Handset," *Lecture Notes in Computer Science*, vol. 3333, pp.681-688, Dec. 2004.

[7] Y. Arai, T. Agui, and N. Nakajima, "A few DCT-SQ scheme for images," *Trans. IEICE*, Vol. E71, pp. 1095-1097, 1988.

[8] ARM, http://www.arm.com.

[9] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. on Signal Processing*, Vol. 49, No. 12, pp 3032-3044, DEC. 2001.

[10] M. D. Adams, "The JPEG-2000 Still Image Compression Standard," *ISO/IEC JTC 1/SC 29/WG 1 N 2412*, Dec. 2002.

[11] Gopal Lakhani, "Modified JPEG Huffman Coding," *IEEE Trans. on Image Processing*, Vol. 12, No. 2, Feb. 2003.

Gu-Min Jeong

1995 Dept. of Control and Instrumentation, Seoul National University (B.S.)
1997 Dept. of Control and Instrumentation, Seoul National University (M.S.)
2001.8 School of Electrical Engineering, Seoul National University (Ph. D.)
2001.8~2004.9 Senior engineer, NeoMtel
2004.9~2005.2 Manager, SK Telecom
2005.3~Present Assistant Professor, School of Electrical Engineering, Kookmin University
Research Areas: Mobile Multimedia, Embedded system, WPAN

Doo-Hee Jung

1991 Dept. of Control and Instrumentation, Seoul National University (B.S.)
1993 Dept. of Control and Instrumentation, Seoul National University (M.S.)
1998 School of Electrical Engineering, Seoul National University (Ph. D.)
1998~2001 Senior engineer, Samsung Electronics
2001~2004 Manager, NeoMtel
2004~Present Assistant Professor, Dept. of Elctronics engineering, Korea Polytechnic University
Research Areas: Embedded system programming, mobile multimedia and home networking

Seung-Won Na

1993 Agricultural Economics, Dankook University(B.S.)
1996 Electronic Information Management, Dankook University (M.S.)
2004 Computer Engineering, Dongguk University (Ph.D.)
1997~Present SK Telecom Platform Researcher
Research Areas: Mobile Computing, Mobile agent, Ubiquitous Computing, Programming Languages

Yang-Sun Lee

1985 Computer Science, DonggukUniversity (B.S.)
1987 Computer Engineering, Dongguk University (M.S.)
1993 Computer Engineering, Dongguk University (Ph.D.)
1994~Present Associate Professor, Dept. of Computer Engineering, Seokyeong University
2000~Present Director of Korea Multimedia Society
Research Areas : Programming Languages, Embedded Systems, Mobile Computings