

論文95-32B-8-5

고속 DCT 병렬처리기의 개발

(A Development of a high speed DCT parallel processor)

朴宗元*, 柳基鉉**

(Jong Won Park, and Ki Hyun You)

요약

DCT 변환은 영상압축을 위한 효율적인 기법으로, 디지털 신호처리분야에서 폭넓게 사용되어져 왔다. 본 논문에는 새로운 알고리즘을 사용한 효율적인 DCT 처리기를 제안하고, Verilog HDL을 사용하여 시뮬레이션하였다. 이 알고리즘은 Y. Arai의 알고리즘에 비하여 다소 복잡하지만, 처리속도에서 60% 정도 향상되었다. 이 알고리즘은 실시간 영상처리에 효율적으로 사용될 것이다.

Abstract

The Discrete Cosine Transform(DCT) is effective technique for image compression, which is widely used in the area of digital signal processing. In this paper, an efficient DCT processor is proposed and simulated by using Verilog HDL. This algorithm is improved 60% in processing speed, but it's somewhat complicate compared with Y. Arai's algorithm. This algorithm will be used efficiently for real time image processing.

I. 서론

영상정보의 중요성이 점차로 더해감에 따라 영상의 효율적인 변환, 압축, 처리, 저장 및 송,수신방법에 대한 연구가 지속적으로 이루어 지고 있다.

영상을 압축하는 목적은 영상의 전송과 저장에 있는데, 디지털 영상을 효율적으로 운용하기 위해서는 영상을 표현하는데 필요한 비트수를 줄이는 기술이 필요하다. SDTV(Standard Definition Television) 영상은 프레임(frame)당 720 x 480 화소(pixel)의 해상도를 가지는데, 세가지 색상을 사용하고 각 색상에 대

해 한 화소를 표시하기 위해 8비트가 필요하고 초당 30 프레임을 가지게 될 경우, 82,944,000 bit/sec의 대역폭(bandwidth)을 요구하게 된다. 기존에 사용하는 TV전송방식보다 훨씬 많은 정보량을 갖는 디지털 전송방식의 HDTV(High Definition Television)의 경우에는 이 보다 더욱 큰 대역폭을 요구하게 되므로 더욱 고도의 영상 압축을 필요로 하게 된다.

현재 동영상표준으로서 자리잡고 있는 MPEG (Moving Picture Experts Group)의 경우에 빠른 시간에 동영상을 전송하기 위해서 영상을 압축하는 기법으로 공간적 중복성을 제거하기 위하여 DCT (Discrete Cosine Transform)방법을 채택하고 있다^{1), 2)}.

DCT는 처음에는 FFT(Fast Fourier Transform)에 기반을 두고 있었으며, 그 후에 DCT를 효율적으로 계산할 수 있는 여러 알고리즘들과 하드웨어 구현에 용이한 방식들이 발표되었다.

*正會員, **準會員, 忠南大學校 情報通信工學科

(Dept. of Inf. Comm. Eng., Chungnam Univ.)

接受日字: 1995年 2月 11日, 수정완료일: 1995年 7月 21日

실시간 영상처리를 위하여 다량의 영상자료를 고속으로 DCT 처리하기 위해서는, 여러 영상 자료들을 동시에 DCT 처리할 수 있는 DCT용 병렬처리의 개발이 필요하다. DCT 연산을 소프트웨어적인 방법으로 처리하면 많은 처리시간을 필요로 하기 때문에 이를 하드웨어로 구현해야 할 필요성이 있으며 이에 관한 연구들이 진행되고 있다. 지금까지는 DCT 연산을 위해 필요한 곱셈기와 덧셈기의 수를 줄이는 방향으로 연구가 진행되었다^{13, 41}.

본 논문은 곱셈기와 덧셈기의 수를 다소 증가시키더라도 전체적인 DCT 처리속도를 향상시킬 수 있는 효율적인 DCT용 병렬처리를 개발하였다. 이러한 병렬 처리기는 다량의 영상자료들을 실시간에 DCT 처리할 수 있도록함으로써 디지털 영상통신을 이용한 다양한 영상 정보 서비스를 실시간에 처리할 수 있도록 하는데 목적이 있다.

이를 위하여 본 논문에서는 기존의 방법에 비하여 속도면에서 개선된 고속 DCT 병렬알고리즘을 제안하며, 하드웨어 디자인 툴인 Cadence사의 Verilog-XL을 사용하여 시뮬레이션하였다. 제안된 알고리즘 및 Verilog HDL로 작성된 DCT 처리기는 Chip 으로 제작될 경우 DCT처리를 하는 모든 분야에 사용될 수 있으며, 하드웨어가 병렬처리를 목적으로 구현되었으므로 실시간 영상처리를 위한 JPEG¹⁵¹이나 MPEG의 응용에 적합하다.

본 논문의 순서는 다음과 같다. 먼저 II장에서는 DCT알고리즘에 대하여 간단히 설명하고, III장에서는 현재까지 연구된 Y. Arai¹⁴¹의 DCT 방법과 본 논문에서 제안하고자 하는 DCT병렬처리의 알고리즘의 방법과 연산방식을 설명하고, IV장에서는 제안된 DCT 병렬처리의 알고리즘을 verilog HDL 로 구현하여 시뮬레이션한 방법 및 과정을 설명하고 제안된 고속 DCT 병렬알고리즘과 Y. Arai¹⁴¹의 알고리즘의 시뮬레이션 결과를 비교하여 분석한다. 마지막 V장에서는 시뮬레이션 된 결과를 토대로 결론을 맺었다.

II. DCT 알고리즘

DCT는 직교변환(orthogonal transform)이며, 고속알고리즘을 갖는 변환이고, DCT 기본 함수는 시각특성기준(psycho-visual criteria)의 효과적인 이용을 매우 용이하게 한다는 장점과 구현이 비교적 쉽다

는 이점이 있다¹⁶¹.

DCT는 공간상의 영상데이터를 주파수상의 에너지 분포로 변환시키는 것으로, 일반적인 영상데이터는 저주파수 영역으로 집중되는 경향이 있다. 이러한 특성을 가진 DCT는 고능률 부호화/복호화방식으로서 JPEG (Joint Photographic Experts Group)과 MPEG (Moving Picture Experts Group)의 표준 부호화방식으로 채택되었다^{11, 2, 51}. DCT가 개발되기 이전에는 DFT(Discrete Fourier Transform)이 DSP (Digital Signal Processing)의 여러 분야에 사용되었으나, 1974년에 Ahmed¹⁷¹에 의해 DCT가 개발된 이후 DSP 및 여러 영상응용분야에 이용되어져 왔다. 처음 발표된 DCT는 N-point DCT 계수를 2N-point FFT를 이용하여 계산하는 방식을 취하였으나, $2N\log 2N$ 번의 복소수덧셈과 $N(\log 2N + 1)$ 번의 복소수 곱셈연산을 해야 하는 많은 계산량으로 인하여 속도가 느려지는 단점을 보완하기 위하여 구현이 용이한 고속의 DCT알고리즘들^{13, 4, 8, 9, 10, 111}이 지속적으로 연구되어져 왔으며, Fast DCT알고리즘과 하드웨어들이 발표되어져 왔다¹²¹.

DCT는 다음 식과 같이 정의된다.

$$m = 0, 1, \dots, (N-1)$$

$$k = 0, 1, \dots, (N-1)$$

$$k = 0, 1, \dots, (N-1)$$

$$X(k) = \sqrt{(2/N)} C(k) \sum_{m=0}^{N-1} x(m) \cos \frac{(2m+1)k\pi}{2N}$$

$$x(m) = \sqrt{(2/N)} \sum_{k=0}^{N-1} X(k) \cos \frac{(2m+1)k\pi}{2N}$$

$$c(k) = 1/\sqrt{(2)} \text{ if } k = 0$$

$$c(k) = 1 \text{ otherwise}$$

DCT를 하드웨어적으로 구현하기 위한 연구는 국내 및 국외에서 활발히 연구되어져 왔으며^{12, 13, 14, 151}, 발표된 DCT 처리기의 구조에는 곱셈기(multiplier)를 이용하여 벡터행렬연산(Vector matrix multiplication)을 하는 방식과 분산연산을 이용하는 방식 그리고 ROM(Read Only Memory)와 누산기(Accumulator)을 이용하는 방식들이 있다. 곱셈을 사용하는 방법은 속도는 빠르나 집적도가 낮아서 IC화 하는데 문제가 있고, 덧셈을 사용하는 방법은 속도는 느리나 IC화에 적합하다고 알려져 있다.

III. 제안된 병렬 DCT 알고리즘

1. Arai 의 DCT 알고리즘

현재까지 DCT 처리기에 대한 연구방향은 곱셈과 덧셈의 수를 최대한 줄이는 연구가 계속되어 왔다. 그 중 Y. Arai [4] 등은 곱셈 5개와 덧셈 29개로 줄였으나(표 1, 그림 1, 그림 2), Y. Arai [4] 등의 논문에서 실제로 DCT를 처리하는 데는 8-point DCT의 경우, Coefficient들을 곱하여 주어야 하므로 8 개의 곱셈이 더 소요되어 실제로 곱셈수는 13개가 된다.

Y. Arai [4] 의 알고리즘은 전체 곱셈수와 덧셈수를 줄이는 데만 주력하였기 때문에 입력에서 출력까지의 시간은 크게 고려하지 않았다.

표 1. Arai 알고리즘의 계수
Table 1. Coefficient of Arai's algorithm.

n	constant(n)
0	0.176776695
1	0.127448894
2	0.135299025
3	0.150336221
4	0.176776695
5	0.224994055
6	0.326640741
7	0.640728861

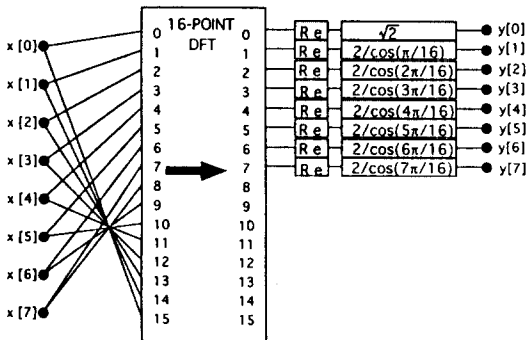


그림 1. 16-point DFT를 이용한 8-point DCT의 계산(Arai 알고리즘)
Fig. 1. Calculation of 8-point DCT using 16-point DFT(Arai's algorithm).

즉, Y. Arai [4] 의 알고리즘은 16-point DFT를 이용하여 Real part를 계산하고 DCT 계수를 계산하여 DCT를 완성하는데 네 번의 덧셈, 한 번의 곱셈, 세 번의 덧셈, 한 번의 곱셈과정이 연속적으로 수행되어야 하므로 결국 7번의 덧셈과 2번의 곱셈과정이 DCT에 소요된다.

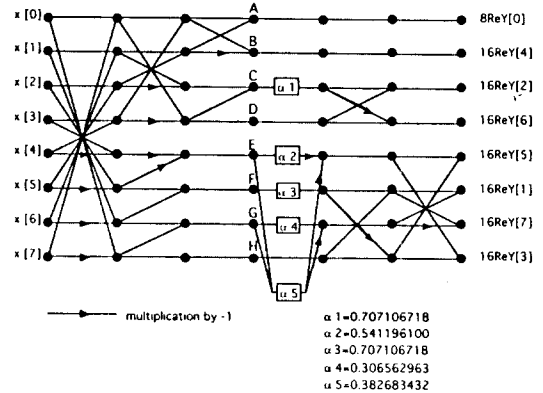


그림 2. 16-point DFT의 실수부분을 계산하는 Flow chart(Arai 알고리즘)
Fig. 2. Flow chart to find the real parts of 16-point DFT(Arai's algorithm).

2. 고속 DCT 알고리즘

MPEG에서 고속 고품격의 영상자료들을 실시간으로 처리하기 위해서는 고속의 병렬 DCT처리가 필요하다. 지금까지 DCT를 처리하는데 소요되는 곱셈기와 덧셈기의 갯수를 줄이는 방법은 고속의 DCT 처리를 위한 방법과 약간의 거리가 있다. 즉, 아무리 곱셈과 덧셈의 수효를 줄인다해도 이들의 연산이 순차적으로 이루어져야 한다면 좀더 많은 곱셈과 덧셈이 병렬로 이루어지는 방법에 비해 그 속도가 늦기 때문이다. 현재는 곱셈기와 덧셈기 등의 하드웨어부품이 저렴해지고 있으며 ASIC 기술이 발전함에 따라 집적도가 지속적으로 향상되고 있으므로 부품을 몇개 더 두더라도 그 전체 속도가 향상되는 방법이 더 좋은 방법일 수 있다. 즉, ASIC 제작기술이 상당히 발달되었기 때문에 DCT처리를 외부 pin의 수가 동일한 하나의 IC로 제작이 가능하다면 그 IC내의 복잡도가 다소 증가되더라도 속도가 빠른 쪽을 선호하게 될 것이다.

이러한 측면에서 본 논문은 8-point 1-D DCT를 처리하는데 소요되는 곱셈기와 덧셈기의 수를 다소 늘리더라도 병렬로 연산이 이루어지게 함으로써 전체 속

도를 향상시키고자 한다. 현재 연구된 바로는 8-point DCT를 처리하는데 7번의 덧셈과정과 1개의 곱셈과정이 연속적으로 소요되는 방법이 고안되었다.

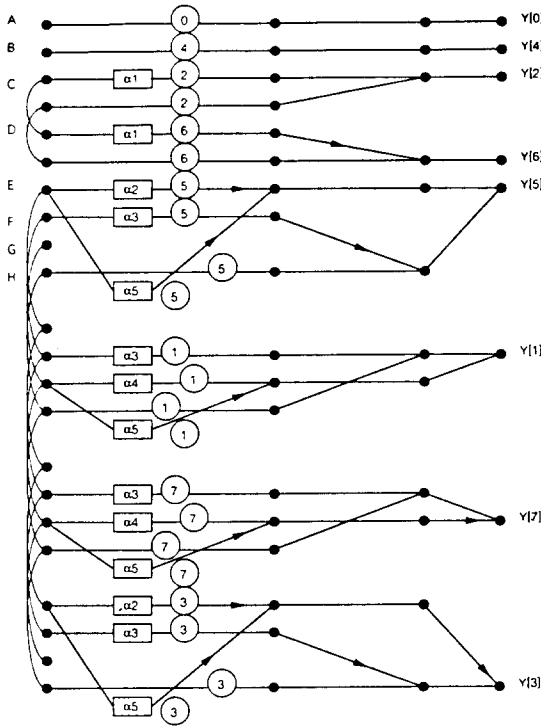


그림 3. 제안된 병렬 DCT의 Flow Graph
Fig. 3. Flow Graph of the proposed parallel DCT.

그림 3의 제안된 DCT알고리즘은 그림 2의 Y. Arai [4]의 알고리즘을 개선한 것으로 여섯 개의 과정 중 뒤의 3개의 단계를 바꾼 것이다. 즉 Coefficient의 곱셈을 그림 3의 3개의 단계에서 계산하도록 알고리즘을 개선하여 Y. Arai [4]의 알고리즘에서 전체적인 곱셈과정을 한 과정 줄인 것이다. 그림 3에 나오는 a1, a2, a3, a4, a5는 그림 2에 나오는 실수값을 그대로 사용한 것이며, 원의 숫자는 표 1의 값이다. 원과 사각형은 실수값이므로 두 실수값를 미리 곱한 값을 메모리에 저장한 후 사용하게 되므로 실제의 부동소수점 곱셈연산은 하나의 직선상에 이어진 두 개의 실수라하더라도 한 번에 이루어 질 수 있다. 그림 3에서 좌측의 호선으로 연결된 자료들은 동일한 자료임을 의미하며 가장 많이 연결된 점의 수는 4개이므로 실제로 칩으로 구현될 경우 별 문제가 없다.

최종적으로 나오는 값은 Y [0]에서 Y [7]까지의 DCT처리값이 나오게 되며 이것은 실수값이 나오게 되므로 양자화과정을 거치게 되면 12 비트의 값을 가지게 된다.

그림 3에서 보듯이 제안된 알고리즘은 Y. Arai [4]의 알고리즘보다 다소 복잡하고, 곱셈기와 덧셈기가 22개, 33개로 증가하지만 전체적인 곱셈과정을 줄임으로써 DCT 처리속도는 향상되었음을 알 수 있다. 이 알고리즘은 Y. Arai [4]의 알고리즘에서 Coefficient 값을 곱하는 과정을 없애고, 전 단계에 이 연산을 집어넣음으로써 한 단계를 감소시킨 것이다. 결국 DCT를 계산하는 과정은 복잡하게 되었지만, 8개의 연산흐름이 병렬로 이루어지게 되므로 가장 느린 픽셀의 계산에 전체의 연산시간이 의존하게 됨으로써 전체적인 연산속도는 Y. Arai [4]의 알고리즘에 비하여 향상되었음을 알 수 있다.

IV. 시뮬레이션 및 성능 분석

시뮬레이션 환경은 SUN sparc 10에서 Cadence사의 verilog-XL을 사용하였으며, 제안된 DCT처리알고리즘과 Y. Arai [4]의 알고리즘을 verilog-HDL을 사용하여 설계한 후, 8 x 1의 영상블럭을 입력하여 DCT 처리기를 통하여 변환을 시킨 후 결과를 출력하도록 하였다. 알고리즘의 속도비교는 가장 느린 픽셀에 의해 이루어졌다.

곱셈기의 경우 덧셈기에 비하여 지연속도를 3배로 하여, 덧셈기는 100ns의 연산시간을 가정하고 곱셈기는 300ns의 연산시간을 가정하였다. 이 속도는 값이 전달되어 계산이 이루어지고 난 후, 비트수에 맞추어 절삭과정을 거친 후, 결과값이 출력되어 전달되어지는 시간까지 포함된 것이며, 실제 하나의 칩으로 구현되어질 경우 이보다 빨라질 것으로 예상된다.

표 2. 시뮬레이션 된 결과의 비교
Table 2. Comparison of the simulated results.

	Adder	Multiplier	Delay(ns)
Arai의 알고리즘	29	13	130000
제안된 알고리즘	33	22	81000

각 알고리즘에서 소요되는 덧셈기와 곱셈기의 숫자와 지연시간은 다음 표 2와 같다.

표 2의 두 알고리즘의 지연시간을 보면, 제안된 알고리즘은 Y. Arai [4]의 알고리즘보다 60%정도 빠름을 예상할 수 있다.

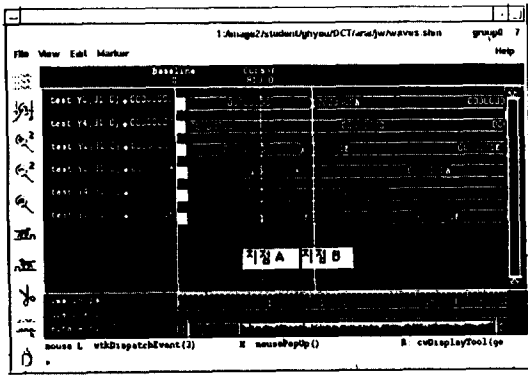


그림 4. Arai의 알고리즘과 제안된 알고리즘의 파형도

Fig. 4. Waveforms of the Arai's algorithm and the proposed algorithm.

그림 4에 시뮬레이션의 결과파형이 나타나 있는데, 윗 부분의 3개의 파형은 Y. Arai [4]의 알고리즘을 수행한 결과, 8개의 출력 $Y[0]$, $Y[1]$, $Y[2]$, $Y[3]$, $Y[4]$, $Y[5]$, $Y[6]$, $Y[7]$ 중 지연시간이 가장 큰 $Y[0]$, $Y[2]$, $Y[4]$ 를 나타낸 것이며, 아래의 3개의 파형은 본 논문에서 제안된 알고리즘의 수행결과, 지연시간이 가장 큰 $Y[0]$, $Y[2]$, $Y[4]$ 를 나타낸 것이다. 제안된 알고리즘의 경우 $Y[2]$ 가 가장 많은 지연시간(지점 A)을 요구하며, Y. Arai [4] 알고리즘의 경우에는 $Y[0]$ 가 가장 많은 지연시간(지점 B)이 요구됨을 알 수 있다.

그림 4에서 두 와이어의 값이 최종적인 결과값을 내는 지연시간을 비교하면 제안된 알고리즘이 Y. Arai [4]의 알고리즘보다 속도면에서 향상되었음을 보인다.

이 결과는 Y. Arai [4]의 알고리즘에서 DFT를 거친 후, 부동소수점연산을 한번 더 해주어야 하는 것을 미리 계산하여 주므로 약 한 부동소수점곱셈의 연산속도만큼 제안된 논문이 빨라짐으로써 전체적인 속도가 향상되었음을 보여준다.

V. 결 론

본 논문에서는 기존의 Y. Arai [4]에 의하여 제안되어진 DCT알고리즘을 개선하여 고속 DCT 처리를 위한 병렬알고리즘을 제안하였다. 제안된 알고리즘은 CADENCE 사의 verilog HDL 로 시뮬레이션한 결과 속도면에서 60% 정도 향상되었음을 알 수 있었다. Y. Arai [4]의 알고리즘에서는 덧셈기가 29개, 곱셈기가 13개필요하지만, 제안된 알고리즘은 덧셈기가 33개, 곱셈기가 22개가 사용되므로 복잡도에서 다소 불리함을 알 수가 있다.

그러나, 현재의 ASIC기술은 집적도면에서 상당한 발전을 하였으므로 덧셈기와 곱셈기의 수가 증가되었으나 제안된 고속 DCT처리를 하나의 Chip으로 DCT처리를 제작하는 데에는 큰 문제가 없다고 본다.

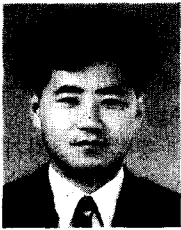
※ 본 논문은 1993년도 한국학술진흥재단 학술연구 조성비에 의하여 연구되었음.

참 고 문 헌

- [1] Didier Le Gall, "MPEG: A video compression Standard for multimedia applications", *Comm. of the ACM*, vol.34, No. 4, pp.47~58, April 1991.
- [2] 한국산업연구표준원, "MPEG 동영상상호화기술 및 응용 Workshop", Nov. 1994.
- [3] B. G. Lee, "A new algorithm to compute the Discrete Cosine Transform," *IEEE Trans. ASSP*, Vol. Assp-32, No 6, pp.1243-1245, Dec. 1984.
- [4] Y. Arai, et al., "A fast DCT-SQ scheme for images," *The Trans. of the IEICE*, Vol. E71, No 11, pp.1095-1097, Nov. 1988.
- [5] Gregory K. Wallace, "The JPEG Still Picture Compression Standards," *Communications of ACM*, 34(4), April, 1991.
- [6] K.R.Rao and P.Yip, *Discrete Cosine Transform, Algorithms, Advantages, Applications*, Academic Press, 1990.
- [7] N. Ahmed, T. Natarajan, and K. R.

- Rao, "Discrete Cosine Transform", *IEEE Trans. Comm.*, vol. Com-23, pp. 90~93, Jan. 1974.
- [8] Hsieh S. Hou, "A Fast Recursive Algorithm For Computig the Discrete Cosine Transform", *IEEE Trans. On Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 10, pp1455~1461, Oct. 1987.
- [9] Meghanad D. Wagh and H. Ganesh, "A new algorithm for the Discrete Cosine Transform of arbitrary number of points", *IEEE Trans. on Computer*, Vol. c-29, No. 4, pp.269~277, April. 1980.
- [10] S. S. Kang and M.H. Lee, "An expanded 2-D DCT algorithm based on convolution", *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 3, pp.159~165, Aug. 1993.
- [11] Weiping Li, "A new algorithm to Compute the DCT and its inverse", *IEEE Trans. on signal processing*, Vol. 39, No. 6, pp.1305~1313, June 1991.
- [12] N. I. Cho, et al., "DCT Algorithms for VLSI Parallel Implementations", *IEEE Trans. On Acoustics, Speech, and Signal Processing*, Vol. 38, No. 1, pp.121~127, Jan. 1990.
- [13] Ming-Ting Sun, et al., "VLSI Implementation of a 16 x 16 Discrete Cosine Transform", *IEEE Trans. On Circuits and Systems*, Vol. 36, No. 4, pp.610~617, April, 1989.
- [14] M. H. Lee, "On computing 2-D Systolic Algorithm for DCT", *IEEE Trans. On Circuits and Systems*, Vol. 37, No. 10, 1990.
- [15] 이승욱, 임강빈, 정화자, 정기현, 김용득, "고속 DCT 알고리즘을 이용한 DCT 및 IDCT 구조", *전자공학회논문집 제 31권 제 3호*, pp. 103~114, 1994년 3월.

저 자 소 개



朴宗元(正會員)

1979년 2월 충남대학교 공과대학 전자공학과 졸업(공학사), 1981년 2월 한국과학원 전산학과 졸업(이학석사), 1991년 8월 한국과학기술원 전산학과 졸업(공학박사), 1990년 10월 충남대

학교 전산학과 부교수, 1992년 미국 텍사스 대학교 전기 및 전자공학과 교환교수, 현재 충남대학교 정보통신공학과 부교수 및 전자계산소장, 주관심분야는 컴퓨터 구조 및 영상처리 등임.



柳基鉉(準會員)

1993년 2월 충남대학교 생화학과 졸업(이학사), 1995년 2월 충남대학교 전산학과 졸업(이학석사), 주관심분야는 컴퓨터 구조 및 영상처리 등임.