

論 文

고속DCT 변환 방식의 정수형 연산에 관한 연구

正會員 全 俊 鉉* 正會員 高 鍾 錫**
正會員 金 聖 大*** 正會員 金 在 均****

On the Finite-word-length Effects in fast DCT Algorithms

Joon Hyeon JEON*, Jong Seog KOH**,
Seong Dae KIM***, Jae Kyoon KIM**** *Regular Members*

要 約 오늘날 데이터 감축을 위한 디지털 영상처리에 KLT와 매우 유사한 성능을 갖는 DCT에 관한 관심이 점점 늘고 있다. 지금까지 계산량을 줄이기 위해 발표된 많은 알고리즘 중에서 Chen의 알고리즘이 가장 많이 알려져 있다. 최근에 발표된 Lee 알고리즘은 Chen의 알고리즘에 비해 복잡한 계산량을 줄였지만 고정 소수점 연산시 FWL에 의한 성능감소가 발생한다. 본 논문에서는 FWL 영향에 따른 오차 분석을 행하여 두 알고리즘들의 성능비교를 하였다. 또한 고정 소수점 연산에 따른 성능감소를 줄이기 위해 up & down-scaling 기법을 제안하였다. 16비트 고정 소수점 연산으로 16 × 16 2차원 DCT를 구현하여 영상 데이터에 적용할 경우 이론적인 고찰과 시뮬레이션 결과 Lee 알고리즘이 Chen 알고리즘과 유사한 성능을 갖는다는 것을 입증하였다.

ABSTRACT In recent years there has been an increasing interest with respect to using the discrete cosine transform(DCT) of which performance is found close to that of the Karhunen-Loeve transform, known to be optimal in the area of digital image processing for the purpose of the image data compression. Among most of reported algorithms aimed at lowering the computation complexity, Chen's algorithm is found to be most popular. Recently, Lee proposed a new algorithm of which the computational complexity is lower than that of Chen's, but its performance is significantly degraded by FWL(Finite-Word-Length) effects as a result of employing a fixed-point arithmetic. In this paper performance evaluation of these two algorithms and error analysis of FWL effect are described. Also a scaling technique which we call Up & Down-scaling is proposed to alleviate a performance degradation due to a fixed-point arithmetic. When the 16×16 point 2-D DCT is applied on image data and a 16-bit fixed-point arithmetic is employed, both the analysis and simulation show that is close to that of Chen's.

I. 서 론

디지털 영상 데이터의 많은 중복성(redundance)을 크게 줄이기 위해서는 변환 방식(transformation)

이 많이 사용되고 있다⁽¹⁾. 이것은 변환 영역(transform)에서의 에너지 집중(energy compaction) 현상을 이용하는 것으로서, 1 bpp(bit per pel)이하의 낮은 전송 속도에서 좋은 성능을 갖으며 원영상(original image)의 통계적 변화 및 전송 잡음에 의한 성능 저하가 적다는 장점이 있기 때문이다^(2,3).

여러가지 변환 방식 중에서도 최적(optimum)

*, **, ***, **** 韓國科學技術院 電氣및 電子工學科
Korea Advanced Institute of Science and
Technology Electrical Engineering
論文番號 : 87-32(接受 1986. 8. 25)

변환 감축 방식인 KLT(Karhunen-Loeve Transform)⁽⁴⁾와 가장 유사한 성능을 갖는 이산 여현 변환(Discrete Cosine Transform=DCT)⁽⁵⁾은 그동안 발표된 여러 알고리즘들에 의해서 실시간 처리(real-time processing)가 가능해졌다(6, 7, 8, 9, 10, 11). 특히 Chen 방식과 Lee 방식의 고속 알고리즘(fast algorithm)들이 우수한 것으로 알려져 있다.

본 논문에서는 Chen 방식과 Lee 방식의 알고리즘들을 유한 길이연산(finite-length arithmetic) 방법으로 구현한 경우에 발생하는 성능감소와 그 원인들이 정량적으로 고찰되었다. 그리고 이 두방식에 대해서 실시간 구현이 쉬운 정수형 연산을 사용한 경우에 반올림(rounding) 및 up & down-scaling 방법에 따른 성능 향상이 비교 검토되었다.

그 결과 두 방식이 모두 이상적인 DCT(real DCT=floating-point DCT)에 가까운 성능 향상이 가능함이 확인되었으나 하드웨어가 간단한 후자의 방식이 실시간 처리용 DCT의 구현에 편리하다고 할 수 있다.

성능 평가를 위해서 사용된 입력 영상은 한 화소(picture element) 당 8 비트의 해상도를 갖고 256×256 개의 화소로 구성된 "GIRL"과 "PSEUDO"이며 그 통계적 특성은 표 1 과 같다. 특히 "PSEUDO" 영상은-128에서 127 범위에서 균일한 분포를 갖는 영상데이터이다. 그리고 적용된 영상변환은 크기가 16×16인 영상 블록에 대한 2 차원 DCT이지만 실시간 구현을 고려해서 그림 1 과 같이 수평 및 수직 방향에서의 단계적인 1 차원 변환으로 하였다. 본 논문에서 사용된 성능 비교의 척도는 식 (1)과 같이 원영상 및 재생 영상(reconstructed image)의 화소 밝기(intensity)인 $f(i, j)$ 와 $f^*(i, j)$ 의 상대적인 평균 자승 오차 NMSE(Normalized Mean Square Error)

표 1 영상데이터의 통계특성
Statistics of image data

	Mean	Variance	Correlation Coefficient		
			Horizontal	Vertical	Diagonal
Girl	181.4'	1,816.6	0.973	0.962	0.948
Pseudo	0	5,446.44	0	0	0

이며, 별도로 실제 영상을 비교하는 subjective test도 행하였다.

$$NMSE = 10 \log_{10} \frac{E\{|f(i, j) - f^*(i, j)|^2\}}{E\{f^2(i, j) - f^2\}} \quad (1)$$

II. 이산 여현 변환

주어진 한 일차원 이산 신호 $f(j)$, $j = 0, 1, \dots, N-1$ 에 대한 순변환(Forward DCT) 및 그 역변환(Inverse DCT)은 각각 다음과 같이 정의된다⁽¹⁾.

$$F(k) = \frac{2}{N} D(k) \sum_{j=0}^{N-1} f(j) C_{2N}^{(2j+1)k} \quad (2)$$

$$f(j) = \sum_{k=0}^{N-1} D(k) F(k) C_{2N}^{(2j+1)k} \quad (3)$$

여기서

$$D(k) = \begin{cases} \frac{1}{\sqrt{2}} & (k=0) \\ 1 & \text{else} \end{cases}$$

$$C_{2N}^{(2j+1)k} = \cos[\pi(2j+1)K/2N]$$

이며, $F(k)$ 는 변환 계수(transform coefficient)를 나타낸다.

Chen의 변환 방식⁽⁸⁾에서는 기본 행렬(basic

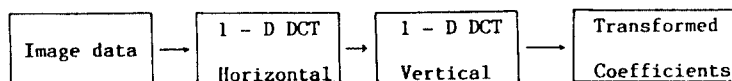


그림 1 2 차원 DCT의 블록도
Block diagram of 2-D DCT.

matrix)을 순환형(recursive form)으로 구하는 행렬 분리(matrix decomposition) 방식이 제시되었으며, 2 신호 흐름도는 그림 2와 같다. 반면에 Lee^[10]는 FFT와 유사한 전개 방식을 사용하여 그림 3과 같은 신호 흐름도를 제안하였다.

Chen과 Lee 방식 들에 대한 그림 2, 그림 3의 신호 흐름도를 보면 모두 하드웨어 구현시에 pipeline 방식을 적용할 수가 있어서 실시간 처리가 용이함을 알 수가 있다. 표 2에서 처럼 직접적인 DCT 계산 방식에 비하여 덧셈과 곱셈의 연산 수가 크게 감소되었음을 볼 수 있으며, 특히 Lee 방식은 하드웨어 구현에 상대적으로 어려운 곱셈 수가 더욱 적어진 것을 알 수가 있다. 그러나 Chen 방식에 비해 하드웨어가 간단하나, 신호 흐름도 상에서 연산 단계(stage)가 한 번 더 있기 때문에 정수형 연산시에 연산 오차(error)가 더 커지는, 즉 성능이 저하되는 단점이 있다.

표 2 DCT 구현에 따른 연산 Computational requirements for DCT.

Transform	N = 8		N = 16		N = 32	
	ADD	Mult	ADD	Mult	ADD	Mult
Direct	56	64	240	256	996	1,024
Chen's	26	16	74	44	194	116
Lee's	29	12	82	32	209	88

Ⅲ. 정수형 연산의 성능 고찰 및 분석

실제 DCT 고속 알고리즘을 하드웨어로 구현 할 때 정확한 변환 계수를 얻기 위해서는 유동소수점 연산을 이용할 수가 있으나 가수부와 지수부를 별도로 계산해야 한다는 어려움이 있다. 따

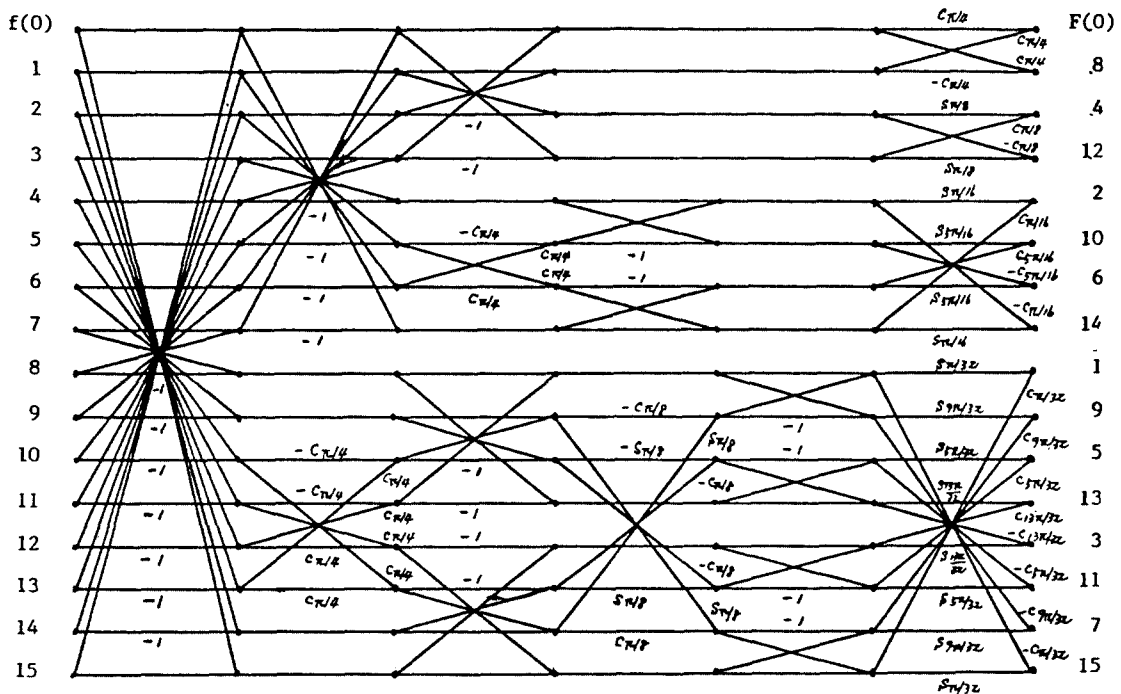


그림 2 N=16일때 chen 방식의 FDCT 신호흐름도, FDCT signal flow graph for N=16 of Chen's algorithm.

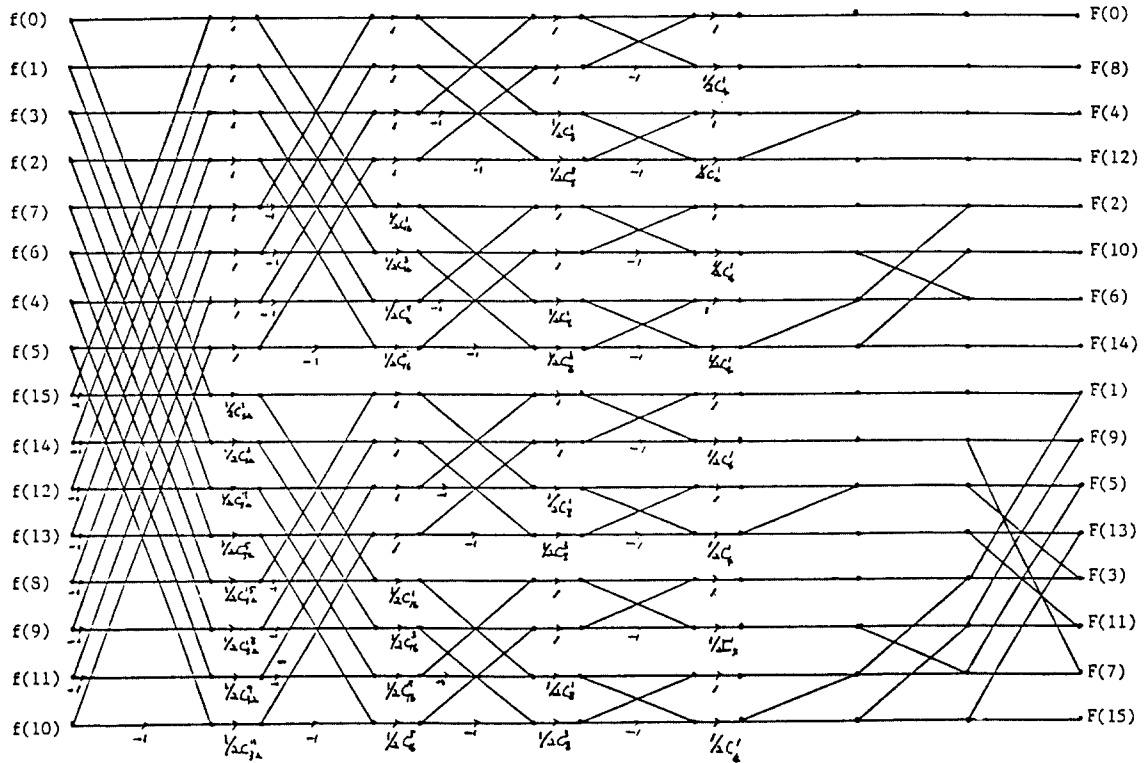


그림 3 N=16일때 Lee방식의 FDCT 신호흐름도
FDCT signal flow graph for N=16 of Lee's algorithm.

라서 본 논문에서는 하드웨어 구현을 간단히 하기 위해 가장 낮은 비트(Least Significant Bit) 아래에 소수점을 갖는 고정 소수점의 정수형 연산을 하였다. 그리고 삼각 함수 계수의 정수형(integer with fixed point) 연산은 소수점 이하의 값을 적절한 비트 만큼 scale한 후에 연산(multiply)이 끝난 다음 다시 scale한 만큼 나누어 주었으며, scale값은 overflow가 일어나지 않는 범위에서 최대 비트로 하였다. 모든곱셈은 이에 해당되며 이것은 신호 흐름도 내에서 정수형 연산에 따른 삼각 함수 계수들의 고정 소수점 승산(fixed-point multiplication)에 의한 오차를 발생시킨다. 이 오차는 신호 흐름도 상의 단계(stage)가 증가함에 따라 계속 누적되어 결국 성능 감소의 큰 요인이 된다. 본 장에서는 정수형 연산에서의 이러한 문제점들을 극복하기 위한 반

올림(rounding) 및 up & down-scaling 방법들을 정량적으로 고찰하고, 이로서 실시간 구현이 편리한 Lee 방식으로도 Chen 방식의 성능에 접근할 수 있는 성능개선의 정도를 분석하였다.

가. 반올림(rounding)

정수형 연산에서 고정 소수점 승산 후에 생기는 오차는 절삭 오차(truncation error)와 반올림 오차(rounding error)의 경우를 비교하면 다음과 같다⁽¹²⁾.

정수와 과정에서 절삭과 반올림에 의해서 발생하는 오차값 e_T 와 e_R 은 식 (4), (5)의 범위 내에서 균일한 확률 분포를 갖는다고 가정하면

$$-1 < e_r \leq 0 \tag{4}$$

$$-\frac{1}{2} < e_R \leq \frac{1}{2} \quad (5)$$

평균 가능 오차 값 (Mean Square error) 은 각각 다음과 같다.

$$\sigma_{e_T}^2 = E[e_T^2] = \frac{1}{3} \quad (6)$$

$$\sigma_{e_R}^2 = E[e_R^2] = \frac{1}{12} \quad (7)$$

따라서 정수형 연산에서 반올림 방식을 택하면 절삭 방식에 비해서 6dB의 성능 개선을 얻게 됨을 알 수 있다.

이제 Chen과 Lee 방식에서 매번의 곱셈마다 생기는 반올림 오차(roundoff error)에 대한 전체 NMSE를 구하여 두 방식의 성능을 고찰해

보기로 하자.

각각의 승산 후에 발생하는 반올림으로 인한 오차들을 같은 크기의 독립적인 잡음원으로 가정하면 최종 출력 node에서의 잡음은 이들 잡음의 총합이 될 것이다. $N=16$ 에 대한 그림 2와 그림 3의 신호 흐름도를 살펴 보면 $N=4$ 와 $N=8$ 의 신호 흐름도들과 서로 일정한 연관성을 가지고 구성되어 있음을 알 수가 있다. 따라서 이를 바탕으로 해서 임의의 N -point에 대한 신호 흐름도로 부터 최종 출력 node들에 대한 잡음원의 갯수를 알 수가 있다. 뿐만아니라 데이터의 갯수 N 이 두 배로 늘어날 때에는, 증가된 부분에 대해서만 잡음원의 영향을 고려해 주면 된다. 또한 그림 2, 3에서 Chen과 Lee 방식의 기본 Butterfly 구조로부터 일반적인 승산 오차 잡음 모델은 그림 5와 같이 표현할 수 있다.

Chen 방식은 N -point DCT를 위한 변환 행렬 $[A_N]$ 을 matrix decomposition에 근거하여 순환



그림 4 반올림 및 절삭오차에 대한 확률밀도함수
Probability density functions for
(a) rounding; (b) truncation.

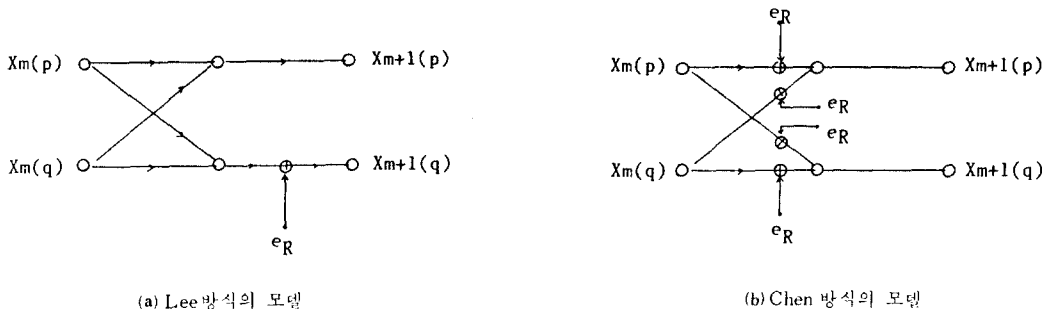


그림 5 Butterfly 내에서 고정소수점 반올림잡음에 대한 통계적 모델
Statistical model for fixed-point roundoff noise in a butterfly.
(a) Lee's algorithm (b) Chen's algorithm

형 (recursive form) 으로 구할 수 있는다는데 그 기본을 두고 있으며 그 방법은 다음식과 같다⁽⁸⁾.

$$[A_N] = [P_N] \begin{bmatrix} A_{N/2} & 0 \\ 0 & R_{N/2} \end{bmatrix} [B_N] \quad (8)$$

$$[R_{N/2}] = [D(k) \cos \frac{(2j+1)(2k+1)\pi}{2N}]$$

$$j, k = 0, 1, \dots, \frac{N}{2} - 1 \quad (9)$$

여기서 단위 행렬로 구성된 $[B_N]$ 과 순환 행렬 $[P_N]$ 은 기지의 행렬이다. 위 식은 $N \times N$ 변환행렬 $[A_N]$ 이 $[A_{N/2}]$ 행렬과 $[R_{N/2}]$ 로부터 구해질 수 있다는 것을 의미한다. 그림 2의 $N=16$ 일 때 흐름도를 보면 $[A_8]$ 의 흐름도에 $[R_8]$ 의 새로운 흐름도가 확장된 것을 보여주고 있다. N 이 증가할 때 새로 형성되는 $[R_{N/2}]$ 은 다시 $(2 \log_2 N - 3)$ 개의 sparse matrix 의 곱으로 나타낼 수가 있는데, 이때 $[R_{N/2}]$ 의 sparse matrix 들은 N 이 2 배로 증가할 때마다 식(10)과 같이 2 개씩 (Even, Modd) 늘어나게 된다. 이때 늘어나는 각 mode 의 잡음원의 개수는 같으면 N 이 2 배 증가함으로 따라 3 배로 증가함을 알 수가 있다.

$$[R_{N/2}] = [M_1] [N_2] [M_3] \dots$$

$$[M(2 \log_2 N - 3)] \quad (10)$$

따라서 임의의 출력 node K 에서의 오차 분산, $\text{chen} \sigma^2(k)$ 는

$$\text{chen} \sigma^2(2k) = \text{chen} \sigma_{N/2}^2(k) \quad (11. a)$$

$$\text{chen} \sigma^2(2k+1) = 2 \cdot 3^{m-2} \sigma_{eR}^2 \quad (11. b)$$

$$k = 0, 1, 2, \dots, N/2 - 1$$

$$m = \log_2 N \quad (N \geq 4)$$

$$\sigma_2^2(k) = 2 \sigma_{eR}^2$$

Lee 방식의 경우 그림 3의 흐름도에서 보듯이 $N = 4, 8, 16, \dots$ 으로 증가할 때 $\log_2 N$ 개의 입력단(input stage)과 $(\log_2 N - 1)$ 개의 출력단을 갖는다. 우선 입력단으로 부터 $\log_2 N$ 번째 단계 stage의 node까지 오차 분산은 BRO (Bit Reversed Order) 된 node 순서에 따라 일정하게 증가된다. 이것은 입력 신호의 수 N 이 두 배로 증가할 때마다 입력단에서는 새로운 단계 (stage)가 하나씩 추가되기 때문에, 입력단으로 부터 $\log_2 N$ 번째 단계 (stage)까지 BRO된 node들의 오차 분산의 갯수는 node가 4-point씩 증가할 때마다 처음 4-point DCT의 2번째 단계까지 BRO된 node에서 발생하는 오차 분산갯수의 4의 배수로 증가하기 때문이다⁽¹⁰⁾. 여기서 4-point DCT의 2번째 단계까지 BRO된 node에서 오차 분산의 갯수는 node의 순서에 따라 각각 0, 1, 2, 그리고 3개이다. Lee 방식의 $(\log_2 N - N - 1)$ 개의 출력단의 BRO (Bit Reverse Order) 된 임의의 출력 node K , ($k = 0, 1, \dots, N - 1$)에서 출력 오차 분산은 각각

$$\text{Lee} \sigma_N^2(2k) = \text{Lee} \sigma_{\frac{N}{2}}^2(k) \quad (12. a)$$

$$\text{Lee} \sigma_N^2(2k+1) = \sigma_{\frac{N}{2}}^2(k) + \hat{\sigma}_{\frac{N}{2}}^2(k+1) \quad (12. b)$$

i) $k = \text{even}$

$$\hat{\sigma}_N^2(2k) = \hat{\sigma}_{\frac{N}{2}}^2(k) + 2^{m+1} \cdot \sigma_{eR}^2$$

(for $N \geq 8$ and $k = 8n - 6$)

$$\hat{\sigma}_N^2(2k) = \hat{\sigma}_{\frac{N}{2}}^2(k) + 2^m \cdot \sigma_{eR}^2$$

(for $k \neq 8n - 6$)

ii) $k = \text{odd}$

$$\hat{\sigma}_N^2(2k+1) = \text{Lee} \sigma_{\frac{N}{2}}^2(k) + 3 \cdot 2^m \cdot \sigma_{eR}^2$$

$$+ \text{Lee} \sigma_{\frac{N}{2}}^2(k+1) + 3 \cdot 2^m \cdot \sigma_{eR}^2 \quad (\text{for } k = \frac{N}{4})$$

$$\hat{\sigma}_N^2(2k+1) = \text{Lee} \sigma_{\frac{N}{2}}^2(k) + 3 \cdot 2^m \cdot \sigma_{eR}^2$$

(for $k = N - 1$)

$$\hat{\sigma}_N^2(2k+1) = \text{Lee}\sigma_N^2(k) + 3 \cdot 2^m \cdot \sigma_{e_R}^2 + \text{Lee}\sigma_N^2(k+1) + 3 \cdot 2^{2m-1} \cdot \sigma_{e_R}^2$$

$$\text{(for } k \neq \frac{N}{4} \neq N-1 \text{)}$$

$$k = 0, 1, \dots, \frac{N}{2} - 1$$

$$n = 1, 2, \dots, 2^{m-2}$$

$$m = \log_2 N - 1$$

$$\text{Lee}\sigma_N^2(0) = \text{Lee}\sigma_N^2(N) = \hat{\sigma}_N(N) = 0$$

$$\text{Lee}\sigma_N^2(1) = \sigma_{e_R}^2, \quad \hat{\sigma}_N^2(0) = 2 \cdot \sigma_{e_R}^2, \quad \hat{\sigma}_N^2(1) = 3\sigma_{e_R}^2$$

이며, 서로 순환형(recursive form)으로 구성되어 있다.

위의 식 (11)과 식 (12)에서 Chen과 Lee 방식은 N이 증가함에 따라 N-point의 출력단이 N/2-point의 출력단과 서로 맞물려 있는 관계로 인해서 전체 잡음원의 갯수를 쉽게 알 수가 있다. 따라서 Chen과 Lee방식의 전체 신호 흐름도의 반올림에 의한 출력 오차 분산은 각 node의 출력 오차분산들의 합이 되며 식 (11)과 식 (12)로부터 Chen과 Lee방식의 흐름도 전체에서의 반올림에 의한 오차 분산은 각각

$$\sum_{k=0}^{N-1} \text{chen}\sigma_{e_R}^2(k) = 2^3 \cdot \sigma_{e_R}^2 \left[1 + 3 \sum_{k=1}^{m-1} 6^{k-1} \right] \quad (13. a)$$

$$\sum_{k=0}^{N-1} \text{Lee}\sigma_{e_R}^2(k) = 3^2 \cdot \sigma_{e_R}^2 \left[1 + \sum_{k=1}^{m-1} (2^{3k-1} + 2) \right] \quad (13. b)$$

로 주어진다. 여기서 N=32point 이상에서는 좀더 자세한 고찰이 필요하다¹³.

식(13)은 N > 1에 대하여 각각

$$\sum_{k=0}^{N-1} \text{chen}\sigma_N^2(k) \approx \frac{2}{15} \cdot \sigma_{e_R}^2 \cdot N \cdot 3^{10\log_2 N} \quad (14. a)$$

$$\sum_{k=0}^{N-1} \text{Lee}\sigma_N^2(k) \approx \frac{9}{112} \cdot \sigma_{e_R}^2 \cdot N^3 \quad (14. b)$$

로 근사화된다. 각 node K에서 평균 잡음 분산(average noise variance)를 σ_N^2 라 하면 식 (14)에서

$$\text{chen}\sigma_N^2 \approx \frac{2}{15} \cdot 3^{10\log_2 N} \cdot \sigma_{e_R}^2 \quad (15. a)$$

$$\text{Lee}\sigma_N^2 \approx \frac{9}{112} \cdot N^2 \cdot \sigma_{e_R}^2 \quad (15. b)$$

가 된다.

또 출력 F (=transformed coefficient)의 분산은 입력 f가 서로 독립이라고 가정되었으므로 식(2)로부터

$$E[|F(k)|^2] = \frac{2}{N} \cdot E[|\hat{F}(k)|^2] \triangleq \sigma_f^2 \quad (16)$$

여기서

$$\hat{F}(k) = D(k) \sum_{j=0}^{N-1} f(j) C_{2N}^{(2j+1)k}$$

이다. 식 (15)와 (16)로부터 출력 F에 대한 오차의 평균 NMSE는,

$$\text{NMSE} = \frac{\text{Average Noise Power}}{\text{Average Signal Power}} \quad (17)$$

로부터 평균 잡음 전력에 정규화 계수 연산에 따른 잡음 전력이 포함되어

$$\begin{aligned} \text{NMSE}_{\text{Chen}} &= \frac{\text{chen}\sigma_N^2 + \sigma_{e_R}^2}{\sigma_f^2} \\ &= \left(\frac{2}{15} \cdot 3^{10\log_2 N} + 1 \right) \cdot \frac{\sigma_{e_R}^2}{\sigma_f^2} \end{aligned} \quad (18. a)$$

$$\begin{aligned} \text{NMSE}_{\text{Lee}} &= \frac{\text{Lee}\sigma_N^2 + \sigma_{e_R}^2}{\sigma_f^2} \\ &= \left(\frac{9}{112} \cdot N^2 + 1 \right) \cdot \frac{\sigma_{e_R}^2}{\sigma_f^2} \end{aligned} \quad (18. b)$$

이 되며, 식 (18)은 다시 N > 1에 대하여

$$\text{NMSE}_{\text{Chen}} \approx \frac{2}{15} \cdot 3^{10\log_2 N} \cdot \frac{\sigma_{e_R}^2}{\sigma_f^2} \quad (19. a)$$

$$NMSE_{Lee} \approx \frac{9}{112} \cdot N^2 \cdot \frac{\sigma_{eR}^2}{\sigma_f^2} \quad (19. b)$$

로 근사화 된다. 만약 이상적인 (real) DCT 의 경우 흐름도 내에서 아무런 오차도 발생하지도 않고, 단지 출력 node에서 정규화 계수에 의한, 단 한 번씩의 반올림 오차만을 갖는다고 가정하면 이때 각 출력 node의 평균 NMSE(real)는

$$NMSE_{real} = \frac{\sigma_{eR}^2}{\sigma_f^2} \quad (20)$$

가 되어 식 (19)와 상당히 많은 성능 차이가 있음을 알 수가 있다. 또한 식 (19)로부터 Chen과 Lee의 성능 차이는

$$NMSE_{Chen} - NMSE_{Lee} = -(4.15 \log_{10} N - 2.2) [dB] \quad (21)$$

이며, 여기서 N이 증가할수록 성능 차이는 더욱 커짐을 알 수가 있다. 그림 6은 N=16일 때 각 변환 계수에 따른 오차들의 분산을 구한 것으로서 역시 Lee 방식의 오차가 매우 큰 것을 알 수가 있다.

만약 이것을 2 차원 DCT (FDCT & IDCT)로 확장할 경우 Chen과 Lee 방식들의 성능 차이는 식 (21)에서 보다 더 커질 것이다. 따라서 다음에 논의될 up & mdown-scaling 방법은 이 오차를

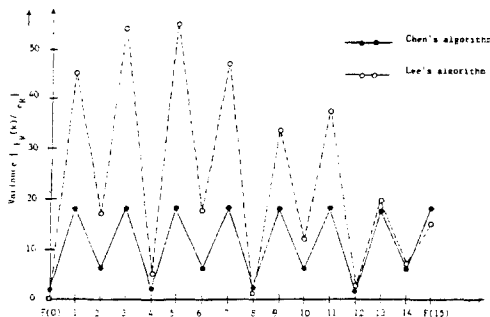


그림 6 N=16일때 각 변환 계수에 따른 잡음원의 분산
The variances of noise source in each transform coefficients for N=16 points.

최소로 줄이기 위함이다.

나. Up & Down-scaling

8 비트 입력 데이터에 대해 DCT된 변환 계수 최대치 (DC 값)는 N=16일때 12비트가 되므로 16비트 하드웨어를 사용할 경우 상위 4비트는 항상 비어 있게 된다. Up-scaling은 이 비어 있는 상위 비트를 사용하는 방법으로 overflow가 일어나지 않는 범위에서 입력데이터를 일정하게 키워서 계산하는 방법이다. 이 up-scaling 효과는 입력 신호 전력을 증가시키므로써 오차를 잡음으로 볼 때, 상대적으로는 잡음 전력을 감소시키는데 있다. 즉 up-scaling을 하므로써 연산도 중 생기는 반올림에 의한 오차의 범위를 줄여 성능 개선을 가져 오는 것이다. 그리고 up-scaling 비트를 Lu라 하면, up-scaling을 했을 때는 up-scaling비트(=Lu)만큼 down-scaling을 해야하며, 이것은 정규화 계수 연산 비트(=log₂N-1)와 더불어 그 차이(=|Lu-(log₂N-1)|)를 down-scaling하면 되므로 더 이상의 오차를 발생시키지 않는다. 이 down-scaling 방법은 뒤에서 다시 자세히 논의하겠다.

Lu(<=4) 비트를 Up-scaling 했을 때 반올림에 의한 오차의 범위는

$$-2^{-(Lu+1)} < e_R \leq 2^{-Lu+1} \quad (22)$$

의 uniform한 분포를 갖으며, 이것은 up-scaling을 안 한 식 (5)의 오차 범위보다 작아졌음을 알 수가 있다. 식 (22)로부터 오차의 평균 분산(average variance), $E[e_R^2] \approx \sigma_{eR}^2$ 은

$$\sigma_{eR}^2 \approx \int_{-2^{-(Lu+1)}}^{2^{-Lu+1}} 2^{Lu} \cdot e_R^2 de_R = \frac{2^{-2Lu}}{12} \quad (23)$$

이며, 이때 각 출력 node에서의 평균 잡음 분산은 식 (15)로부터

$$Chen \sigma_{eR}^2 \approx \frac{2}{15} \cdot 3^{\log_2 N} \cdot \sigma_{eR}^2 \quad (24. a)$$

$$\sigma_{eR}^2 \approx \frac{9}{112} \cdot N^2 \cdot \sigma_{eR}'^2 \quad (24. b)$$

이 되며, 이것은 up-scaling 비트에 따라 각각 2 제곱 (= 6 dB) 의 지수승(exponential)으로 감소됨을 보여 주고 있다. 따라서 식 (16)으로 부터 출력 F에 대한 오차의 평균 NMSE는

$$NMSE_{chen} = \left(\frac{2}{15} \cdot 3^{\log_2 N} \cdot \sigma_{eR}^2 + \sigma_{eR}^2 \right) \cdot \frac{1}{\sigma_f^2} \quad (25. a)$$

$$NMSE_{Lee} = \left(\frac{9}{119} \cdot N^2 \cdot \sigma_{eR}^2 + \sigma_{eR}^2 \right) \cdot \frac{1}{\sigma_f^2} \quad (25. b)$$

이다. 만약 N (: fixed)에 대하여 충분한 up-scaling ($\sigma_{eR}^2 \approx 0$)을 한다면, 식 (25)는 각각

$$NMSE_{chen} \approx \frac{\sigma_{eR}^2}{\sigma_f^2} \quad (26. a)$$

$$NMSE_{Lee} \approx \frac{\sigma_{eR}^2}{\sigma_f^2} \quad (26. b)$$

로 Chen과 Lee 방식의 성능이 식 (20)의 이상적인 DCT 성능에 근사화됨을 볼 수가 있다. 실제 $N=16$ 이고 $L_u=4$ 일 때 식 (18)과 식(25)로 부터 Chen과 Lee의 성능 차이는 각각

$$\text{식(18)} \quad NMSE_{chen} - NMSE_{Lee} = -2.8 \text{ dB} \quad (27. a)$$

$$\text{식(25)} \quad NMSE_{chen} - NMSE_{Lee} = -0.16 \text{ dB} \quad (27. b)$$

로 된다. 더우기 2차원 DCT processor로 확장할 경우, 모든 순변환(FDCT)과 역변환(IDC T)이 각각 independent 하다고 가정하면, 그림 7의 up-scaling의 경우 1번의 down-scaling오차를 한 번 더 포함하게 된다. $N=16$ 에 대한 Chen과 Lee의 성능 차이는 식 (27)에서

$$\text{식(18)} \quad 2\text{-D}NMSE - 2\text{-D}NMSE_{Lee} \approx -11.2 \text{ dB} \quad (28. a)$$

$$\text{식(25)} \quad 2\text{-D}NMSE - 2\text{-D}NMSE_{Lee} \approx -0.64 \text{ dB} \quad (28. b)$$

로 된다. 이때 up-scaling으로 많은 성능 개선 (: $11.2 - 0.64 = 10.56 \text{ dB}$)을 얻을 수가 있었다. 또한 식 (28. b)의 성능 차이는 매우 높은 SNR에서의 0.64dB 차이므로 실제 재생 영상에 있어서는 거의 영향을 미치지 못한다. 충분한 2-Dim. up-scaling ($\sigma_{eR}^2 \approx 0$)에 의한 성능 향상 정도는 식 (19)와 (26)에서

$$\Delta NMSE_{chen} \approx - (15.85 \cdot \log_{10} N - 8.75) \cdot 2 \quad (29. a)$$

$$\Delta NMSE_{Lee} \approx - (20 \log_{10} N - 10.95) \cdot 2 \quad (29. b)$$

이 되어 $N \geq 16$ 일 때 20dB 이상씩의 성능 개선을 가져왔음을 알 수가 있다.

Up-scaling을 했을 때는 up-scaling 비트 만큼 반드시 down-scaling을 해야 하는데 그림 7과 같이 2가지 방법이 있다. 기존의 방법인 그림 7.(a)는, 상위 4비트를 모두 사용 (= full up-scaling)하여 down-scaling 비트를 조절하는 그림 7.(b)의 방법에 비해, 낮은 up-scaling 비트에서 좋은 성능을 얻을 수가 없다. 즉 up-scaling 비트를 4비트 (= 2^4)까지 충분히 하였을 경우 down-scaling을 하더라도 낮은 에너지를 갖는 씨퀀스(sequence)의 계수들을 역변환(IDCT) 전까지도 유지할 수가 있기 때문이다. 이들 2차원 up & down-scaling에 따른 2차원 DCT processor의 최대 성능을 구해보면 다음과 같다. 그림 7에서 모든 순변환(FDCT)과 역변환(IDCT)가 각각 independent 하다고 가정하였으므로 2차원 DCT 후에 생기는 각출력 node에서의 평균 잡음 분산은 2차원 역변환 후 생기는 한 번의 down-scaling에 따른 오차를 포함하여

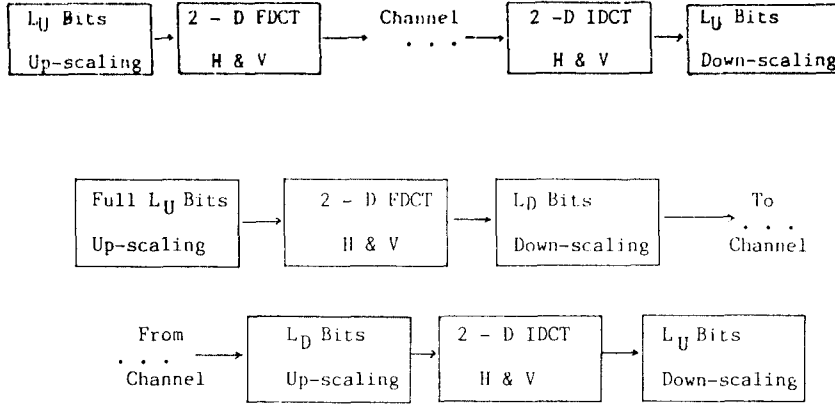


그림 7 두가지 up & down-scaling 방법에 따른 블록도
Block diagram of two up & down-scaling method

$$\sigma_{\text{chen}}^2 \sigma_N^2 = \frac{2}{15} \cdot 3^{\log_2 N} \cdot \sigma_{eR}^2 \cdot 4 + 3 \sigma_{eR}^2 \quad (30. a)$$

$$\sigma_{\text{Lee}}^2 \sigma_N^2 = \frac{9}{112} \cdot N^2 \cdot \sigma_{eR}^2 \cdot 4 + 3 \sigma_{eR}^2 \quad (30. b)$$

이 된다. 식(30)은 충분한 up-scaling을 하면 $\sigma_{eR}^2 \approx 0$ 이 되므로

$$\sigma_{\text{chen}}^2 \sigma_N^2 \approx 3 \sigma_{eR}^2 \quad (31. a)$$

$$\sigma_{\text{Lee}}^2 \sigma_N^2 \approx 3 \sigma_{eR}^2 \quad (31. b)$$

으로 근사화 된다. 2차원 역변환 후 생기는 출력 f' 의 평균 분산은 입력 신호 f 가 uniform한 분포를 갖고 DCT가 준최적 직교 함수(suboptimum orthogonal function)이므로

$$E[|f'(j)|^2] = \sigma_f^2 \quad (32)$$

으로 입력 배열 분산과 같다. 따라서 출력의 평균 NMSE는

$$NMSE_{\text{chen}} \approx NMSE_{\text{Lee}} \approx 10 \log_{10} \frac{3 \sigma_{eR}^2}{\sigma_f^2} \quad (33)$$

이 된다. 만약 입력 신호 f 가 $(-128, 127)$ 에서 균일한 분포를 하는 N 개의 임의의 정수들로 이루어졌다고 가정하면 입력 배열의 분산은

$$\sigma_f^2 \approx E[|f(j)|^2] \approx \frac{128^2}{3} \quad (34)$$

가 된다. 이때 식 (33)에서 $N=16$ 에 대한 4 비트 up & down-scaling에 따른 평균 NHSE는 식(7)의 $\sigma_{eR}^2 = \frac{1}{12}$ 을 사용하면,

$$NMSE_{2-Dim} = 10 \log_{10} \frac{(3/12)}{(128^2/3)} \text{ dB} = -43.39 \text{ dB} \quad (35)$$

로 근사화 된다. 실제로 식 (35)에서 Chen과 Lee의 NMSE 차이는 2dB가 되나 앞에서 언급한 바와 같이 무시할 수가 있다. 이 2차원 up & down-scaling 방법들의 컴퓨터 시뮬레이션에 의한 성능 평가가 그림 8~그림 11에 잘 나타나 있다. Up-scaling에 의해 Chen과 Lee 방식의 성능은 이상적인 DCT를 사용했을 때의 성능에 가까워졌다. 또한 오차에 의한 성능 감소가 별로 없는 Chen 방식은 up-scaling 방법에 따라 큰 성

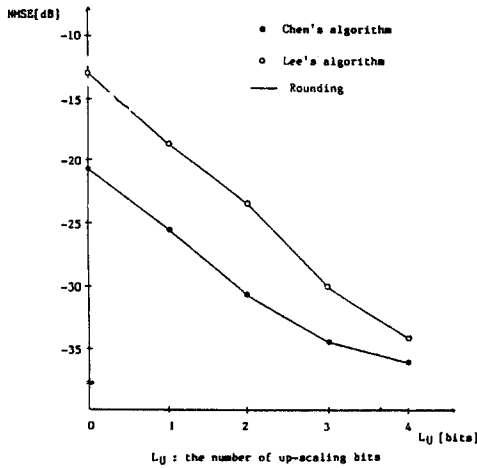


그림 8 그림7.(a)에 근거한 성능평가
Comparison of performance based on the method of Fig. 7(a) (Data base: Girl).

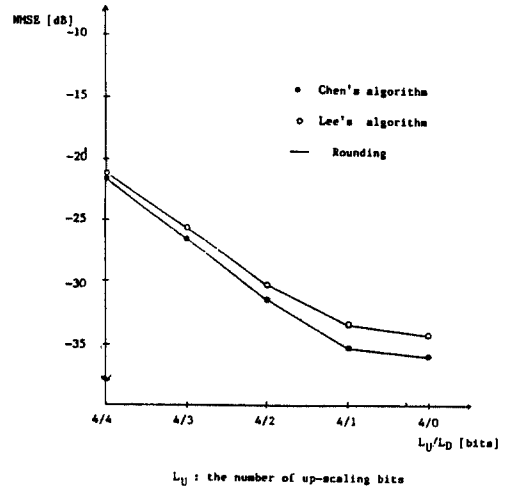


그림 9 그림7.(b)에 근거한 성능평가
Comparison of performance based on the method of Fig. 7(b) (Data base: Girl).

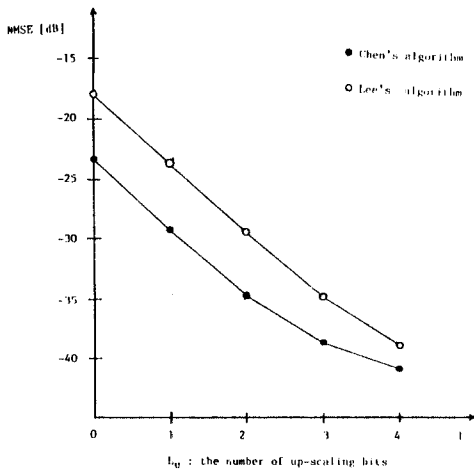


그림 10 그림7.(a)에 근거한 성능평가
Comparison of performance based on the method of Fig. 7(a) (Data base: Pseudo).

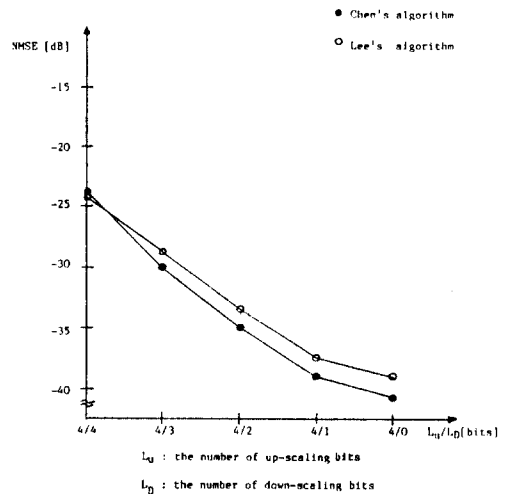


그림 11 그림7.(b)에 근거한 성능 평가
Comparison of performance based on the method of Fig. 7(b) (Data base: Pseudo).

능 향상은 없었으나 성능 감소가 큰 Lee 방식은 그림 7(b)의 방법을 사용하므로써 Chen 방식에 가까운 성능을 얻을 수가 있었다. 여기서 컴퓨터 시뮬레이션은 그림 12와 같이 행하였으며, Chen

과 Lee 방식의 모든 연산은 정수형으로 하였다. 이제 시뮬레이션에 의한 2차원 DCT의 성능을 고찰하고 실제 원영상과 재생 영상을 비교해 보기로 하자.

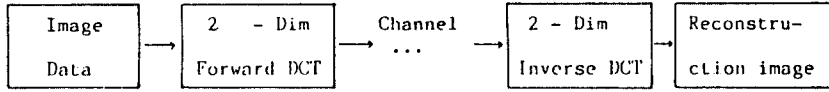


그림12 DCT processor의 컴퓨터 시뮬레이션 방법
Computer simulation of DCT processor.

IV. 시뮬레이션 결과 및 검토

Chen과 Lee방식들을 하드웨어 구현에 맞게 정수형 연산 만으로 컴퓨터 시뮬레이션 결과 입력 데이터 "GIRL"과 "PSEUDO"에 대하여 NMSE가 최고 -20dB이상의 성능 향상 효과를 얻을 수가 있었다. 이것은 이상적인 DCT를 사용했을 때와 근사화된 성능이다. 특히 오차가 많은 Lee 방식의 성능 개선 효과는 매우 놀라운 것으로 그림 7(b)의 방법을 사용한 결과 평균 약 0.9 dB의 성능 차이 만을 나타내었으며 이 값은 앞에서 구한 이론적인 값인 식(28. b)의 값과 유사하다. 실제 시뮬레이션 결과 앞에서 고찰한 이론 값(N=16)과 약간의 차이가 있는데, 이것은 입

력 영상의 실제 통계적 특성("GIRL" data) 및 실제 오차의 모델링 등의 차이로 생각되며, 이것에 좀 더 접근하기 위해서 uniform한 통계적 특성(modeling for "PSEUDO" data)을 갖은 입력 영상을 만들어 시뮬레이션한 결과 실험치와 이론치가 어느 정도 일치함을 알 수가 있었다. 그림 13은 입력 데이터 "GIRL"에 대한 원영상 및 성능 개선을 하지 않았을 때의 재생 영상으로써 (b)는 경계 현상(block effect)이 심한 것을 알 수 있다. 그러나 성능 개선을 한 그림14와 그림 15의 재생 영상에서는 Chen과 Lee 방식들 모두 원영상에 뒤지지 않는 재생 영상을 보여 주고 있으며, 또한 식(28. b)의 성능 차이를 전혀 느낄 수가 없다. 그림16은 Lee 방식을 사용하여 down-scal-



그림 13 (a) 원 영상 "Girl"
(b) 성능개선방법을 사용하지 않은 재생 영상
(a) Original image "Girl"
(b) Reconstructed image without improving performance.



그림14 그림7(b)를 근거로 한 Lee 방식에 따른 재생 영상
Reconstructed images based on the method of Fig.
7(b) by Lee's algorithm (Data base: Girl).

ing 비트에 따른 엔트로피(entropy)를 구한 것이며 계수 당 비트(bit per coefficient)수를 나타낸 것이다. 이것과 그림10과 비교해 보면 적은 bit rates(bpp) 증가로 많은 성능 개선을 얻을

수가 있음을 알 수가 있다. 이것은 전체 코덱 시스템에서 채널 전송량(channel bit rates)과 버퍼량의 조절(buffer control)에 있어서 많은 도움을 준다.



그림 15 그림 7(b)를 근거로한 Chen 방식에 따른 재생 영상.
Reconstructed images based on the method of Fig. 7(b) by Chen's algorithm (Data base: Girl).

V. 결 론

디지털 영상 데이터의 많은 중복성을 제거하기 위한 데이터 감축 방법으로 본 논문에서는 최적

변환인 KLT에 가까운, 하드웨어 구현이 가능한 DCT를 사용하였다. 여러 DCT 알고리즘들 중에서 Chen과 Lee 방식들은 계산 수를 많이 줄였을 뿐만 아니라 신호 흐름도가 pipeline 구조를 하

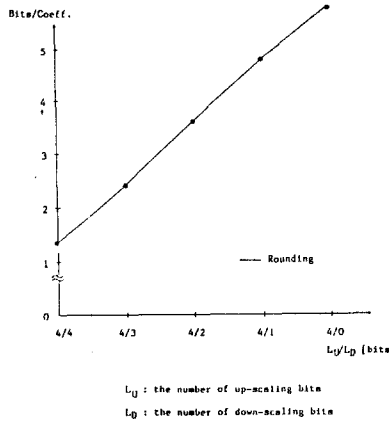


그림 16 그림 7(b)를 근거로한 down-scaling 비트의 증가에 따른 엔트로피증가율
Entropy with the increase of down-scaling bits based on the method of Fig. 7(b).

고 있어 하드웨어 구현이 매우 용이하다. 그러나 실제 하드웨어 구현에 맞는 정수형 연산만으로 컴퓨터 시뮬레이션을 한 결과 삼각 함수 계수들의 고정 소수점 승산(fixed-point multiplication)에 따른 성능 감소로 경계 현상이 심각하게 나타났다. 특히 하드웨어가 간단한 Lee 방식은 신호 흐름도 상에서 연산 단계가 한 번 더 있는 등으로 더 많은 성능 감소가 있었다. 본 논문에서 사용한 반올림과 up & down-scaling 방법 등으로 Chen 방식에 못지 않는 성능을 얻었으며 화질(image quality) 면에서도 원 영상에 뒤지지 않았다. 따라서 하드웨어가 간단하면서도 성능이 뒤지지 않는 Lee 방식을 구현하는 것이 매우 유리하게 되었다.

마지막으로 하드웨어 구현에 있어서, 사용하는 동작 주파수가 높기 때문에 기존의 microprocessor를 사용하지 못한다. 따라서 모든 제어(control)를 ROM에 저장하여 사용하는 ROM Control 방식을 채택하는 것이 바람직하다¹⁴⁾

參 考 文 獻

- (1) A. K. Jain, "Image Data Compression: A Review," *Proc. IEEE*, vol. 69, No. 3, pp. 349-389, March 1981.
- (2) A. G. Tescher, R. V. Cox, "An Adaptive Transform Algorithm," *Picture Coding Symposium, Asilomar Co.*, pp. 47-20-47-23, 1976.
- (3) A. N. Netravali, C. B. Rubinstein, "Luminance Adaptive Coding of Chrominance Signals," *IEEE Trans. Commun.*, vol. COM-27, No. 4, pp. 703-710, April 1979.
- (4) A. K. Jain, "A Fast Karhunen-Loeve Transform for A Class of Random Process," *IEEE Trans. Commun.*, vol. COM-24, pp. 1023-1029, 1976.
- (5) N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Comput.*, pp. 90-93, Jan. 1974.
- (6) M. R. Harlick, "A Storage Efficient Way to Implement the Discrete Cosine Transform," *IEEE Trans. Comput.*, C-25, pp. 764-765, July 1976.
- (7) B. D. Tseng and W. C. Miller, "On Computing the Discrete Cosine Transform," *IEEE Trans. Comput.*, vol. C-27, pp. 966-968, Oct 1978.
- (8) W. H. Chen, C. H. Smith, and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Commun.*, vol. COM-25, No. 9, pp. 1004-1009, Sept. 1977.
- (9) M. J. Narasimha and A. M. Peterson, "On the Computation of Discrete Cosine Transform," *IEEE Trans. Commun.*, vol. COM-26, pp. 934-936, June 1978.
- (10) Byeong Gi Lee, "A New Algorithm to Compute the Discrete Cosine Transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, No. 6, pp. 1243-1245, DEC. 1984.
- (11) J. Markhoul, "A Fast Cosine Transform in One and Two Dimensions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. Assp-28, pp. 27-34, Feb. 1980.
- (12) A. V. Oppenheim and R. W. Schaffer, *Digital Signal processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
- (13) Z. D. Wang, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Commun.*, vol. COM-31, No. 1, pp. 121-123, January 1983.
- (14) 전준현, "DCT 고속 알고리즘의 하드웨어 구현을 위한 성능 개선 및 설계에 관한 연구," 한국과학기술원 석사학위논문, 1986.



全 俊 鉉(Joon Hyeon JEON) 正會員
1961年 8月19日生
1984年 2月: 東國大學校 電子工學科 卒業(工學學士)
1986年 2月: 韓國科學技術院 電氣·電子工學科 卒業(工學碩士)
1986年 3月~現在: 韓國科學技術院 電氣 및 電子工學科 博士過程



金 聖 大(Seong Dae KIM) 正會員
1953年12月26日生
1977年 2月: 서울大學校 電子工學科 卒業(工學 學士)
1979年 2月: 韓國科學技術院 電氣·電子工學科 卒業(工學碩士)
1983年12月: 프랑스 E. N. S. E. E. I. H. T 卒業(工學博士)
1984年 9月~現在: 韓國科學技術院 電氣 및 電子工學科 教授



高 鍾 錫(Jong Seog KOH) 正會員
1959年 8月 7日生
1982年 2月: 高麗大學校 電子工學科 卒業(工學學士)
1984年 2月: 韓國科學技術院 電氣 및 電子工學科 卒業(工學碩士)
1984年 3月~現在: 韓國科學技術院 電氣 및 電子工學科 博士過程



金 在 均(Jae Kyoon KIM) 正會員
1938年 9月17日生
1958. 4 ~1962. 2: 韓國航空大學 應用電子科(工學士)
1962. 3 ~1967. 2: 서울大學校 大學院 電子工學科(工學碩士)
1967. 9 ~1971. 8: 美國 남가주大學校大學院 電氣工學科(工學博士)

1962. 4 ~1966. 7: 空軍服務 中尉
1967. 9 ~1972. 3: 美國 남가주 大學校(研究院)
1972. 4 ~1973. 3: 美國 NASA GSFC(研究院)
1973. 4 ~現在: 韓國科學技術院 教授
1987. ~現在: 本 學會 副會長
1984. 1 ~1985. 6: 科學技術處 電氣·電子조정관