

Fermat Number 변환에 기반한 모션벡터 예측

학생회원 김 남 호*, 성 주 승**, 정회원 송 문 호***

Motion Estimation by Fermat Number Transform

Nam-Ho Kim*, Ju-Seung Sung** *Student Members*, S. Moon-Ho Song*** *Regular Members*

요 약

본 논문에서는 비디오 데이터 압축에 있어서 모션 벡터를 주파수 도메인에서 찾는 방법에 대하여 제안한다. 제안하는 알고리즘은 Fermat Number 변환에 의해 주파수 도메인에서 현재의 프레임의 매크로 블록과 참조 프레임의 매크로 블록 사이의 상관관계(correlation)가 최대인 것을 찾아 이것을 모션벡터로 인식하는 방식이다. 제안하는 방법은 최대 상관값을 찾는 데, 이 방법은 기존 최소 L2-norm을 찾는 방법과 동등하다는 것을 수학적으로 증명하였다. 제안하는 방법의 특징은 데이터의 특성에 관계없이 모든 영상에 대해서 동일한 속도로 최적의 해를 구할 수 있다는 것을 입증하였다.

ABSTRACT

In this paper, we present a method to find motion vectors in frequency domain for video data compression. The proposed algorithm is based on the Fermat Number Transform (FNT), and it declares the most correlated-block as the best matching block, as opposed to declaring the block with least sum of differences between blocks. We show that the proposed method is equivalent to declaring the block with the minimum L2-norm as the best matching block. Unlike other previous fast algorithms, the time requirement for the proposed algorithm does not depend on the image type for finding the optimum solution.

I. 서 론

멀티미디어 관련 기술의 발전은 여러 분야에 많은 영향을 미치고 있다. 그 영향 중 하나로 디지털 비디오 압축 방식인 MPEG 기술이 상용화됨에 따라 우리의 생활이 윤택해지고 있다는 것을 느낄 수 있다. 특히 MPEG-2는 고화질 TV(HDTV)나 대화형 TV, 디지털 방송이나, 디지털 비디오 디스크(DVD)등 통신, 방송, 컴퓨터 분야에 광범위하게 응용이 되고 있는 실정이다.

그러나 MPEG-2는 CD-ROM에 저장되어 단순히 복호화 과정만을 필요로 하는 MPEG-1과는 달리 양방향 통신의 기능이 필수적이다. 다시 말해서, 부호화 및 복호화가 실시간적으로 일어나야 한다는

것이다. 영상 신호는 그 데이터량이 매우 크다는 특징이 있기도 하지만, 연속되는 두 영상간에 중복되는 데이터가 많다는 것 또한 특징이라고 할 수 있다. 따라서 모든 데이터를 보내는 것보다, 연속된 두 영상 사이에 존재하는 시간적인 중복성을 제거한 차 영상만을 전송하는 것이 효율적인 방법이라는 것을 알 수 있다. 이러한 시간적인 중복성을 제거하는 방법으로 가장 많이 사용하는 것이 모션 벡터 예측이다. 따라서, 모션 벡터 예측에 근거한 예측 비디오 코딩은 비디오 데이터의 압축에 있어서 매우 중요한 역할을 하고 있다고 할 수 있다. 현재 표준으로 자리잡고 있는 MPEG과 H.26x의 경우에도 모션 예측에 기반한 비디오 압축을 사용한다. 그러나, 모션 벡터를 찾는 과정이 개념적으로는 간단

* LG산전 중앙연구소(lumina75@hanmail.net),

** 고려대학교 통신신호처리연구소(jssung@openvisual.com),

*** 교신저자, 서울대학교 기계항공공학부(smsong@snu.ac.kr)

논문번호 : 010287-1016, 접수일자 : 2001년 10월 16일

※ 본 연구의 일 부분은 서울대학교 발전기금 일반학술연구과제의 지원으로 진행되었습니다.

하지만, 이를 수행하기 위해서 필요로 하는 계산의 복잡도가 너무 높기 때문에, 모션 벡터를 찾는 과정이 전체 비디오 압축에서 사용하는 시간의 대부분을 차지하게 된다.

프레임 영상(Frame Picture)만이 존재하기 때문에 한 번에 움직임을 예측할 수 있었던 MPEG-1 표준과는 달리, MPEG-2 표준에서는 프레임 영상뿐만 아니라 필드 영상(Field Picture)도 존재하므로, 가장 정합이 잘 되는 부분을 찾기 위해서는 현재의 매크로 블록(Current Macro block)과 참조 프레임을 프레임과 필드 영상 각각에 대하여 수행해야 한다. 이는 MPEG-1의 움직임 예측과는 비교가 되지 않을 정도의 계산능력과 속도가 필요함을 의미한다. 실제 비디오 영상을 부호화할 때 움직임 예측은 전체 부호화 과정에 소요되는 연산량의 50% 이상을 차지하며, 이것은 부호화 과정에서 실시간성을 만족시키는데 있어서 최대의 걸림돌이라고 할 수 있다.

모션벡터(Motion vector)를 찾기 위한 가장 직관적인 블록정합 알고리즘은 Full-Search 알고리즘(FBMA)이다. 이 방법에서는, 탐색 범위 안에 있는 모든 매크로 블록들에 대해서 distortion measure를 구한 다음, 최소의 distortion을 나타내는 블록을 찾게 된다. 비록 FBMA방법이 최적의 모션 벡터를 찾는 방법이기도 하지만, 과도한 distortion 계산은 비디오 인코더에 상당한 부담을 주기 때문에 실제적인 응용이 제한적인 수밖에 없다. 이러한 문제점들을 해결하기 위해서 3단계 탐색(Three-Step Search : TSS), 2차원 로그 탐색(2-D log search, cross search), 새로운 3단계 탐색 알고리즘(New Three Step Search : NTSS), 1차원 전역 탐색(One-Dimensional-Full Search: ODFS), 계층적 탐색(hierarchical search)등 많은 알고리즘들이 개발되어 왔다. 3단계 탐색 알고리즘은 모션 벡터를 추정하는 과정에서 국부최소에 빠지는 단점이 있으며, 새로운 3단계 알고리즘은 움직임이 적은 영상인 경우에서는 비교적 정확하게 모션 벡터를 추정하지만 움직임이 빠른 영상인 경우에서는 전역최소(global minimum)에 빠져서 부정확한 모션 벡터를 추정하는 단점이 있다.

이러한 fast 알고리즘들은 error-surface 함수가 블록함수라는 가정에서 출발했다. 그러나 실제 영상에 있어서 이러한 가정은 거의 맞지 않기 때문에 FBMA에 비해서 최적의 모션벡터를 찾을 수 없다.

최적의 모션벡터를 찾기 위해서 제안된 다른 방법은 successive elimination algorithm (SEA)이다.

SEA는 위의 방법들과는 달리 error-surface가 블록함수라는 가정을 하지 않기 때문에 최적의 모션벡터를 찾을 수 있고, 일반적으로 FBMA에 비해서 많은 양의 계산을 줄일 수 있다. 그러나 rejection비율이 입력 영상마다 다르기 때문에, 속도면에서 입력 영상의 특성에 영향을 많이 받고, 장면전환이 있는 영상이나 전체적으로 콘트라스트가 낮은 영상의 경우 FBMA보다 나쁜 성능을 보일 가능성이 있다.

지금까지 개발된 알고리즘이 시간 도메인에서 계산을 수행해 모션 벡터를 찾았던 데 반해, 이 논문에서는 제안하는 방법은 주파수 도메인에서 모션 벡터를 찾는다. 시간 도메인에서는, error-surface 함수를 구해서 이 함수가 최소값을 가지는 위치를 모션 벡터로 선언한다. 그러나 주파수 도메인에서는, 현재의 프레임에서의 매크로 블록과 참조 프레임의 매크로 블록의 상관관계가 최대인 위치가 모션벡터가 된다. 이 논문에서는 상관관계를 이용하는 방법과 error-surface를 이용하는 방법이 동등하다는 것을 수학적으로 증명하고 모든 입력 영상에 대해서 동일한 속도로, 최적의 해를 구할 수 있는 새로운 알고리즘을 제안하고자 한다.

II. 증 명

모션 벡터를 구하는 과정은 다음의 조건을 만족시키는 인덱스 (i,j)의 쌍을 찾는 것과 같다.

$$d = \min_{i,j} \|X - Y(i, j)\|_p \tag{1}$$

여기서 $\|\cdot\|_p$ 는 l_p 노름(norm)을 나타낸다. X는 모션벡터를 얻고자 하는 현재 프레임에서 매크로 블록이 있다고 할 때, 이 블록의 화소값들로 이루어진 벡터이다. Y(i,j)는 탐색 범위 안에 있고 모션 벡터가 될 수 있는 조건을 가지는 참조 프레임의 매크로 블록의 화소값들로 이루어진 벡터를 나타낸다. l_2 노름(norm) 공간에서, (2)를 전개하면 식 (2)는 (3)과 수학적으로 동등관계를 가진다는 것을 알 수 있다.

$$\|Z - W_1\|_2 < \|Z - W_2\|_2 \tag{2}$$

$$2Z^T W_1 - \|W_1\|_2^2 > 2Z^T W_2 - \|W_2\|_2^2 \tag{3}$$

위의 동등관계를 식 (1)에 적용시키면

$$d = \min_{i,j} \|X - Y(i, j)\|_2 \tag{4}$$

$$= \max_{i,j} \{ 2X^T Y(i,j) - Y(i,j)^T Y(i,j) \} \quad (5)$$

식 (4) 에서 얻어진 결과로부터, 모션 벡터가 될 수 있는 모든 매크로 블록과 현재 모션 벡터를 구하고자 하는 매크로 블록에 대해서, 반복적으로 차에 의한 distortion을 구한 다음 가장 적은 distortion을 보이는 매크로 블록의 위치를 찾았던 FBMA 대신에, 각각의 매크로 블록들 사이의 상관관계를 구해서 공식 (4)의 값이 최대가 되는 위치를 모션벡터로 선언을 할 수 있다.

III. Number Theoretic Transform

Number Theoretic Transform을 도입하기 위해서 몇 가지 기호를 정의해야 한다. 두 개의 값 a, b의 차가 정확히 M의 배수가 될 때 a, b는 합동(congruent)이라고 하고 ≡ 를 사용해서 나타낸다. 또 어떤 값을, 자기와 합동이고 집합 0, 1, 2, ... , M 에 속해 있는 다른 값으로 대체할 때 ((×)) 기호를 사용한다.

여기서 우리는 N개의 정수로 이루어진 시퀀스 x(n)의 새로운 Number Theoretic Transform을 변환된 시퀀스인 x(k)와의 관계를 이용해서 정의한다.

$$X(k) = \left(\sum_{n=0}^{N-1} x(n) \alpha^{nk} \right) \quad (6)$$

여기서 α 는 정수이고, M, N 과는 서로 공통인수를 가지고 있지 않아야 한다.

이 공식의 형태는, DFT에서 $W = \exp(j2\pi/N)$ 이 했던 역할을 α가 한다는 것을 제외하면 DFT와 동일하다. DFT에서 $W^N = 1$ 이었던 것과 마찬가지로 α^N 역시 1과 합동(congruent)이고 모든 계산은 모듈로 M이다. FFT를 구현하기 위해서 $W^N = 1$ 이라는 조건이 만족 되어야 한다. DFT 마찬가지로, FNT를 구현할 고속(fast) 알고리즘이 존재하기 위해서 N은 반드시 합성수이어야 한다. 또한 모듈로 M 과 시퀀스의 길이 N 은 서로 공통된 인수를 가지고 있지 않아야 한다. DFT를 해서 얻어진 결과가 주파수 영역에서 신호의 특성을 보여줬던 것에 반해, FNT변환은 상관관계와 convolution을 계산하기 위한 중간 과정으로만 이용하게 될 것이다. 상관관계를 구하기 위해서 두개 시퀀스를 FNT변환을 한 다

음 서로 곱한다. 곱해진 시퀀스는 원래 시퀀스들의 circular convolution을 계산하기 위해서 역변환을 거치게 된다. 따라서 역변환이 반드시 정의되어야 한다. DFT와의 유사성에 의해서 역변환은 다음과 같다는 것을 기대할 수 있다.

$$x(m) = \left(\left(N^{-1} \sum_{k=0}^{N-1} x(k) a^{-mk} \right) \right) \quad (7)$$

여기서 N의 음수의 지수가 있는 것은 가능하다. 왜냐하면 어떤 수 p가 있다고 할 때 p와 M이 서로 소이면 모듈로 M에 대해서 p는 역수를 가지게 된다는 것이 정수론에서 증명이 되어있기 때문이다. Number Theoretic Transform의 정의에서 N과 M 서로소이기 때문에 N의 역수는 존재한다.

IV. Fermat Number 변환

위의 조건을 만족시키는 많은 변환 중에서, 페르마의 수라 불리는 $M = 2^{2^m} + 1$ 을 만족시키는 변환을 Fermat Number Transform (FNT)라 한다. 이때 다른 파라미터들은 각각 $N = 2^{m+1}, \alpha = 2$, 를 만족시킨다. 따라서 FNT변환은 다음과 같이 정의된다.

$$X(k) = \left(\sum_{n=0}^{N-1} x(n) 2^{nk} \right) \quad (8)$$

$$x(m) = \left(\left(N^{-1} \sum_{k=0}^{N-1} x(k) 2^{-mk} \right) \right) \quad (9)$$

FNT에서는 α=2를 사용하고, FFT 구조에서 필요로 하는 계산이, 미리 정해진 α의 거듭제곱들을 입력 시퀀스에 곱한 다음 결과들을 서로 더하거나 빼는 과정으로 이루어져 있다. 컴퓨터로 계산을 할 때, 2의 거듭제곱을 곱하는 것은 간단한 ROTATE-LEFT 연산으로 대체가 가능하기 때문에, FFT를 구현하기 위해서 사용되었던 모든 곱셈연산을 ROTATE-LEFT연산으로 대체 가능하다.

컴퓨터가 제한된 범위의 값만을 저장할 수 있기 때문에, FNT 연산 도중에, 중간 결과가 저장 공간이 가지는 범위를 벗어날 가능성이 있다. 이런 상황들을 방지하기 위해서 연산 중간 과정에서, 적절히 residue reduction을 해주어야 할 필요가 있다. 여기서 32 비트를 하나의 워드로 쓰는 시스템에서, $M = 2^{16} + 1$ 로 사용한다고 가정하자. 어떤 32 비트의 변수가 xxxxyyyy의 값을 가진다고 하면, 간단히

계산에 의해서 위드의 상위 16 비트, xxxx0000,는 모듈로 M계산에서 대해서 양수 0000xxxx와 같은 크기의 음수의 값을 가진다는 것을 알 수 있다. 따라서 xxxxyyyy의 실제값은 모듈로 M에 대해서 0000yyyy-0000xxxx와 같다. 여기로부터 우리는 residue reduction을 보다 간단한 형태로 계산할 수 있다. 즉, 상위 16 비트를 하위 16 비트에서 뺀 결과가 원래의 32 비트 위드와 모듈로 M에 대해서 congruent 하다. 이런 과정은 나눗셈과 모듈로 연산에 의해서 남은 나머지를 구하는 것과 같다. 그러나 나눗셈 연산은 많은 수의 CPU 클럭을 필요로 하기 때문에 계산 속도를 떨어뜨리는 문제가 있다. 일반적인 모듈로 연산이 나눗셈을 포함하는데 반해, FNT 과정에서 필요로 하는 모듈로 연산은 상위 16 비트를 하위 16 비트로 보내기 위한, shift-right연산이다. (표 1). 이때 주의할 점은, 하위 16 비트에서 상위 16 비트를 빼는 과정에서 음수가 나올 수 있다는 것이다. FNT는 모두 자연수로만 이루어져 있기 때문에, 음수로 나온 결과에 M을 더함으로써 양수로 만들어 주어야 한다.

표 1. Numbers of arithmetic operations for various algorithms required for 16x16 current block with search window 17x17.

	ADD	SUB	MUL	ABS	SHIFT	AND
FNT	48999	32913	2048	0	32168	32168
TSS	6400	6400	6375	0	0	0
FBMA-11	73695	73984	0	73984	0	0
FBMA-12	73695	73984	73984	0	0	0

V. 알고리즘

우선 N을 시퀀스의 길이, W를 탐색 범위로 정의하자.

- 1) 모션벡터를 찾고자 하는 $N \times N$ 크기의 매크로 블록을 현재 프레임으로부터 가져온 다음, 제로패딩을 해서 크기가 $(N+2W) \times (N+2W)$ 인 블록이 되도록 한다. 또한, 현재프레임에서의 매크로 블록과 동일한 위치의 참조 프레임에서, 매크로 블록을 탐색 범위 W만큼 확장시킨 $(N+2W) \times (N+2W)$ 크기의 블록을 가져온다.
- 2) 각각의 행렬을 2차원 FNT변환을 한 다음, 결

과를 같은 위치에 있는 컴포넌트들을 서로 곱셈한 다음 역변환을 한다.

- 3) 2가지 얻어진 결과는 circular 상관 관계를 한 결과이기 때문에, 이 중에서 full 상관관계의 결과값을 가지는 $(N+2W) \times (N+2W)$ 크기의 블록만을 선택한다.
- 4) 공식 (4)에 따라서 정규화 과정을 거친 다음, 최대값을 가지는 음셋을 찾으면, 이 값이 모션 벡터가 된다.

VI. 시뮬레이션

FNT 알고리즘의 성능을 테스트하기 위해서 350x240의 크기를 가지는 연속된 30프레임의 table tennis 비디오 시퀀스를 사용하였다. 매크로 블록의 크기를 일반적인 크기인 16x16으로 설정했다. 수직, 수평 방향의 최대 서치 범위를 8로 했기 때문에 하나의 매크로 블록에 대해 17x17에 서치 윈도우를 가지게 된다. FNT의 인자들은 $N = 2^{16} + 1, N = 32$ 이다. 표 2와 표 3에서 볼 수 있듯이, FNT를 구현하기 위해서 필요로 하는 계산의 복잡도가 FBMA를 11에서 구현하기 위해서 필요한 복잡도의 23퍼센트, FBMA를 12에서 구현하기 위한 복잡도의 8퍼센트에 불과하다는 것을 확인할 수 있었다. 이러한 이점들은 FNT를 사용하는 모션벡터

예측 알고리즘이, 많은 수의 CPU 클럭을 소모하는 곱셈연산과 나눗셈 연산을 각각 간단한 ROTATE, AND 그리고 ADD/SUB 연산으로 대체했기 때문에 가능하다.

공식 (4)에 의하면, FNT를 사용하는 경우와 FBMA를 사용하는 경우가 서로 동등한 관계를 가지고 있기 때문에 동일한 PSNR이 얻어져야 하지만, 그림 1에서는 FBMA의 PSNR이 FNT에 비해서 조금 높게 나왔다. FNT의 PSNR이 더 낮게 나온 이유는 다음과 같다. 우리가 두 개의 32x32 행렬에 대해서 상관관계를 했고, 만약 각각의 행렬 요소들이 자기의 최대 DC값을 가진다고 가정하면 상관관계를 했을 때의 결과는 M의 값보다 크게 된다. 따라서 모든 연산이 모듈로 M에 대해서 행해지기 때문에, 최종 결과는 실제 상관관계 결과와 합동(congruent)인 관계를 가질 뿐 동일한 결과는 아니다. 이러한 현상을 방지하기 위해서, 각각의 블록의 DC레벨을 적절한 수준으로 낮추어서 최종결과에서 오버플로우가 일어나지 않도록 했다.

이때 DC값을 낮추는 방법으로는 오른쪽으로 shift하는 방법을 사용한다. 그러나 인텔 계열의 프로세서에서 효율적인 계산을 하기 위해서 발생하는 이와같은 문제들은, 다른 레지스터 구조를 가지는 프로세서에서 N = 64로 설정하면 쉽게 해결할 수 있다.

표 2. Clocks per operation for Pentium Processor [5]. (ABS is an optimal combination of certain operations).

	ADD	SUB	MUL	ABS	SHIFT	AND
FNT	1/3	1/3	10	3	1/3	1/3

표 3. Total number of clock cycles for various algorithms required for 16x16 current block with search window 17x17

	ADD	SUB	MUL	ABS	SHIFT	AND	TOTAL
FNT	16333	10971	20480	0	10722	5493	63999
TSS	2133	2133	63750	0	0	0	68016
FBMA- _{l1}	24565	24662	0	221952	0		271178
FBMA- _{l2}	24565	24662	739840	0	0		789066

VII. 결론

본 논문에서는 Fermat Number 변환에 의해 주파수 도메인에서 현재의 프레임의 매크로 블록과 참조 프레임의 매크로 블록의 상관관계가 최대인 것을 찾아서 이것을 모션벡터로 인식하는 방식에 대하여 제안하였다. 또 상관관계를 이용하는 방법과 error-surface를 이용하는 방법이 동등하다는 것을 수학적으로 증명하였다. 또한 모든 입력 영상에 대해서 동일한 속도로, 최적의 해를 구할 수 있는 새로운 방법에 대하여 살펴보았다. 정리해보면, l_2 노름(norm)의 차이를 최소화하는 모션벡터가 참조 프레임과 현재 프레임 간의 상관관계를 최대화시키는 모션 벡터와 동일하다는 것을 알 수 있었다. Number theoretic 변환인 FNT 가 상관관계를 효율적으로 계산할 수 있기 때문에, 모션 벡터 예측을 FNT에 기반해서 할 수 있었다. 매크로 블록의 크기가 16x16 이고 탐색 윈도우의 크기를 17x17로 했을 경우, FNT기반의 모션벡터 예측 알고리즘의 복잡도는 FBMA- l_1 와 FBMA- l_2 알고리즘에 비해

서 각각 23퍼센트와 8퍼센트의 크기에 불과했다. 실제로 FNT의 복잡도는 PSNR 값의 손실이 없고 최적화 되어있는 상태에서 sub-optimal 알고리즘인 TSS와 비슷한 수준이다. 결론적으로 이 논문에서 제안한 FNT에 기반한 접근 방법은 FBMA 방법에서 필요로 하는 계산량을 PSNR의 손실이 없이도 줄일 수 있다. 그리고 입력 영상의 영향을 받는 알고리즘들은, 탐색을 하는 도중에 계속 비교를 하면서, 특정 조건을 만족시키면 모션 벡터로 선언을 하고 그렇지 않은 경우에는 다음 후보(candidate) 위치에서 동일한 연산을 반복하는 형태를 취하고 있기 때문에 영상의 조건에 따라서 알고리즘의 성능이 영향을 받지만, 이 방법은 모든 입력영상에 대해서 동일한 계산을 하기 때문에 입력영상에 관계없이 항상 수행시간이 일정하다는 장점이 있다.

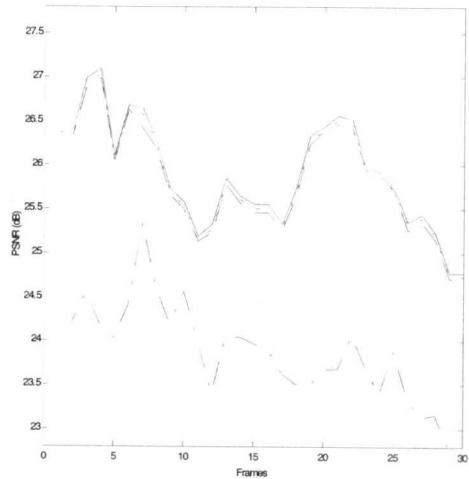


그림 1. PSNR of reconstructed images for the table tennis sequence. Solid line: FBMA-l2, dotted line: FNT, dashed line: FBMA-l1, dash-dot line: TSS

참고 문헌

- [1] Yih-Chuan Lin and Shen-Chuan Tai, Fast full-search block-matching algorithm for motion-compensated video compression, *IEEE Trans. Commun.*, vol. 45, pp. 527-531, May, 1997
- [2] Huan-Sheng Wang and R. M. Mersereau, Fast algorithms for the estimation of motion vectors, *IEEE Trans. Image Processing*, vol. 8, pp.435-438, March, 1999.
- [3] Lawrence R. Rabiner and Bernard Gold, Theory

and Application of Digital Signal Processing, *Prentice-Hall*, 1975.

[4] Boaz Porat, A Course in Digital Signal Processing, *John Wiley, NewYork*, 1997.

[5] Barry B. Brey, The Intel 32-bit Microprocessors : 80386, 80486, and Pentium, *Prentice-Hall*, 1995.

[6] K. Suguri et al., A real-time motion estimation and compensation LSI with wide-search range for MPEG 2 video encoding, *IEEE International Solid-state Circuits Conf.*, 1996

[7] K.W Cheng and S.C Chan, Fast block matching algorithm for motion estimation, *Proc. Of Circuits and System*, 1996

[8] Motion Picture Experts Group, MPEG Committee Draft, *ISO-IEC JTC1/SC29/WG11/602*, Nov. 1993

[9] C.H. Hsieh, P.C.Lu, J. S. Shyu, and E.H. Lu, Motion estimation algorithm using interblock correlation, *IEE Elctronic Letters*, vol. 26, no. 5, pp. 276-277 March 1990.

[10] B. Lie and A. Zaccarin, New fast algorithms for the estimation of block motion vectors, *IEEE Trans. on Circuits and Systems for Video Technology*, vol.3, no.2 pp. 148-157, April 1993.

[11] C. Horne, Improving block based motion estimation by the use of global motion, *Proc. of SPIE VCIP'93*, vol. 2308, pp.1818-1825, Sept. 1994

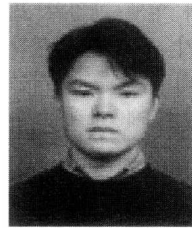
[12] D. LeGall, MPEG : A video compression standard for multi-media communication, *ACM*, vol. 34, no. 4, pp. 47-58, April 1991

[13] ISO/IEC JTC1/SC29/WG11, Moving Picture Expert Group. MPEG-2 test model 4, 1993.

[14] CCITT SG XV, Recommendation H.261, Video codec for audiovisual services at p*64 kbit/s, *Tech. Rep. COM XV-R37-E*, August 1990.

김 남 호(Nam-Ho Kim)

학회회원



1999년 2월 : 고려대학교
전기공학과 졸업
2001년 8월 : 고려대학교
전파공학과 석사
2001년 9월~현재 : LG산전
중앙 연구소

<주관심 분야> 영상신호처리, 통신신호처리, 영상압축

성 주 승(Ju-Seung Sung)

학회회원

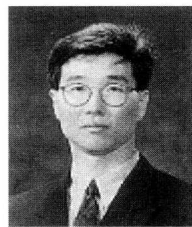


1999년 3월 : 고려대학교
공과대학 공학부 입학
2000년 7월~현재 : 고려대학교
전기전자전파공학부
2002년 6월~현재 : 고려대학교
통신신호처리 연구실

<주관심 분야> 멀티미디어 신호처리용 ASIC설계, 컴퓨터 네트워크

송 문 호(S. Moon-Ho Song)

정회원



1982년 2월 : MIT 전자공학과
졸업
1991년 2월 : USC 전자공학과
박사
1995년 6월~ : 고려대학교
전파공학과 부교수
2000년 9월~현재 : 서울대학교
기계항공공학부 부교수

<주관심 분야> 의료영상신호처리, 영상압축, 통신신호처리