特輯

완전 디지탈 HDTV 방식의 움직임 처리 방식

崔 潤 植
現代電子産業(株) 産業電子研究所

I. 서 론

기존 color TV에 비해 수직, 수평 축으로 각각 2배씩, 전체적으로 4배 이상의 해상도를 가지는 HDTV의 신호는 기존의 제한된 채널을 이용하여 전송하기에는 너무 큰 대역폭을 가지게 된다. 따라서 HDTV 신호의 전송을 위해서는 대역 압축이 반드시 필요하게 된다. 이러한 대역 압축을 위해서 완전 디지탈 방식의 HDTV 시스템에서는 크게 3가지 방법을 사용하고 있는데, 첫째는 화상과 다음 화상 간의 관계, 즉 움직임벡타의 값을 이용하여 전송 정보량을 줄이는 기법, 둘째로 한 화상안의 정보 압축을 위한 DCT 변환과 양자화 기법, 그리고 세번째로 이들 변환된 정보를 정보의 확률적 분포에 의해 적절한 크기의 bit로 변환하는 VLC(variable length coding) 기법이 그것이다. 이 들 기법은 그림 1, 2의 완전 디지탈 방식 encoder, decoder 블럭도와 같은 형태로 조합되어 각각 사용되고 있다.

본 고에서는 이들 정보 압축 기법 중, 연속되는 화상 신호에서 현재 프레임의 화소들이 이전 프레임에 비해 어느 정도 움직였는 지를 벡타로 표시한 움직임벡타를 추정하여, 전체 영상의 전송 대신 이들 움직임 벡타 정보를 전송함으로써 전송 정보를 줄이는 기법에 대하여 중점적으로 서술하고자 한다. 실제적으로 수신단에서의 화상 복원은 이들 추정된 움직임벡타의 값에 의해 추정, 보상되어 만들어 지기 때문에, 정확한 움직임벡타의 추정이 중요하게 되고, 그 보상 방법 또한 중요한 문제가 된다. 더욱기 아무리 우수한 추정방법, 보상 방법이라도, HDTV와 같이 frame rate가 30 frame/sec에 이르면 실시간 구현이 문제가 되기 때문에 hardware적인 구현을 염두에 둔 추정, 보상 방법의 구현 또한 중요한 고려점이 될 수 밖에 없다.

본 고에서는 현재 미국 FCC에 제안해 놓고 있는 4가지의 완전 디지탈 방식들, 즉 General Instrument사의 digicipher system, ATRC의 adtv system, MIT의 CC-HDTV, Zenith AT&T사의 SC-HDTV 방식들에서 사용하고 있는 움직임 보상 방법들을 비교하여, 움직임 추정과 보상 방법을 통한 정보 감축 기법을 고찰하고자 한다.

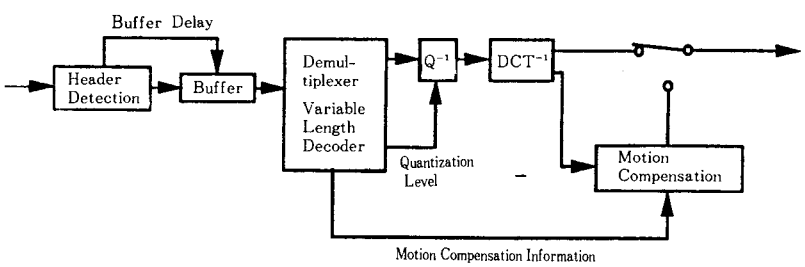


그림 1. 완전 디지탈 방식의 decoder 블럭도

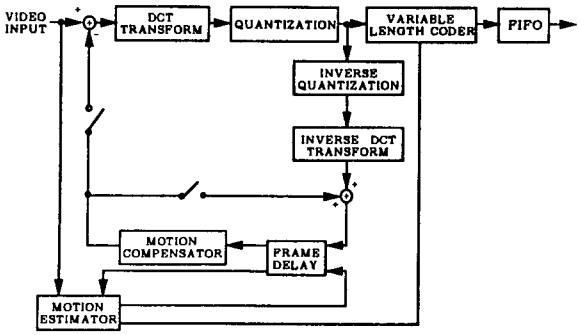


그림 2. 완전 디지탈 방식의 encoder 블럭도

Ⅱ. 움직임 추정 알고리듬

1. Block Matching Algorithm (Full Search 알고리듬)

움직임 벡타를 구하는 방법 중에서 가장 보편적으로 쓰이고 있고, 실제로 제안된 4가지 완전 디지탈 방식 중 3방식이 채택하고 있는 BMA는 화면의 움직임이 수평, 수직으로 평행 이동한 것으로 가정하여, 움직임이 일어난 frame의 block 화상이 움직임이 일어나기 전 frame의 어느 위치에 있는 block 화상과 가장 일치하는 가를 추정하여 그 위치를 통해 움직임벡타를 추정하는 방법이다. 일반적으로 block 크기는 8×8 이나 16×16을 쓰고 있는 것이 보통인데, 이는 block의 크기를 크게하면 전송해야 하는 움직임벡타의 bit 율이 떨어지는 대신, block 안에 여러가지 움직임이 있을 때는 신뢰도가 떨어지는 점을 고려한 block의 크기이다. 움직임의 범위는 이론적으로는 $-\infty$부터 $+\infty$까지 이지만, 실질적 계산 시간이나 H/W의 복잡도들을 고려하여 그 탐색 구간을 결정하게 된다.

수식적으로는, 가장 일치하는 block을 찾기 위하여, 연속된 두 frame의 화상 중에서, 이전 frame을 $f_1(x, y)$, 현재 frame을 $f_2(x, y)$라 했을 때, $f_2(x, y)$와 $f_1(x-\alpha, y-\beta)$에서 α, β를 변화 시켜가면서 $f_1(x-\alpha, y-\beta)$와 $f_2(x, y)$의 차를 구하여 그 차가 최소가 되는 (α, β)를 움직임벡타로 예측하는 방법을 쓰게 되는 데, frame 간의 차이를 구하는 식은 평균 절대 오차(E_{abs})를 주로 사용한다.

$$E_{abs} = \frac{1}{|B|^2} \sum_{x,y \in B} \sum |f_1(x-\alpha, y-\beta) - f_2(x,y)|$$

여기서 $|B|$는 block 크기를 뜻한다.

따라서 움직임 벡타는 위의 식을 최소로 하는 탐색 구간 내의 (α, β)를 찾아 그 값을 취한다. 이 방법은 정확한 방법인 반면, 연산 횟수가 많다는 단점이 있어 많은 고속 알고리듬이 발표되었다. 하지만, 최근 들어 VLSI chip 설계를 함에 있어, 연산 횟수 보다는 병렬처리 기법이 중요해 짐에 따라, 오히려 고속 알고리듬 보다 병렬 처리가 가능한 이 full search 방법이 채택되어 지고 있다. 실제로 실시간 구현이 필수적인 HDTV의 경우, 모든 BMA를 제안한 회사들이 굳이, 움직임 추정에 이 복잡한 full search 방법을 채택하고 있는 것이 바로 이러한 이유를 잘 말해주고 있는 것이다.

2. Spatio-Temporal Constraint 방식

이 방식은, 완전 디지탈 방식을 제안하고 있는 4가지 방식 중에서 유일하게 MIT가 사용하고 있는 방식인데, 화상의 변위가 frame간에 일정한 율로 일어났다고 가정하고, 그 변위율과 움직임벡타와의 상관 관계를 나타내는 미분 방정식인 Spatio-temporal constraint 방정식을 찾아내어 그 근을 구함으로써 움직임벡타를 추정하고자 하는 방식이다. 이러한 Spatio-temporal constraint 방정식은 아래와 같다.

$$\alpha \frac{\partial f(x,y,t)}{\partial x} + \beta \frac{\partial f(x,y,t)}{\partial y} + \frac{\partial f(x,y,t)}{\partial t} = 0$$

여기서 (α, β)는 움직임벡타이고 t는 시간축을 의미한다.

이 식에서 보는 것과 같이, $f(x,y,t)$의 편미분 값들은 어떤 순간에서의 변화율을 나타내므로 일정한 크기의 block 내에서도 각 순간마다 다른 값을 가지게 된다. 따라서 x,y 와 시간축 t 를 포함한 3차원적 단위 block 내의 모든 화소들에 대해 다른 계수값을 가지는 α, β에 관한 방정식이 성립하게 되어 움직임벡타를 구할 수 없는 과정의(overdetermined) 된 연립 방정식이 되고만다. 그러므로 이러한 상황에서 가장 근사한 값을 구할 수 있게 하기 위하여서는 아래와 같은 오차 함수를 최소화 시키는 방법이 사용된다.

오차함수=

$$\iiint_{x,y,t \in \psi} \left(\alpha \frac{\partial f(x,y,t)}{\partial x} + \beta \frac{\partial f(x,y,t)}{\partial y} + \frac{\partial f(x,y,t)}{\partial t}\right)^2 dxdydt$$

이때 이 오차 함수를 움직임벡타에 대해 미분하여 0으로 둠으로써 최소화 시켜서 얻어진 결과의 움직임벡타 $v = (\alpha, \beta)^{-1}$ 는 아래의 식을 만족시킨다.

$W_v = r$

단,

$$W = \begin{bmatrix} \iiint_{x,y,t\in\psi} \left(\frac{\partial f(x,y,t)}{\partial x}\right)^2 dxdydt \cdot & \iiint_{x,y,t\in\psi} \frac{\partial f(x,y,t)}{\partial x}\frac{\partial f(x,y,t)}{\partial y} dxdydt \\ \iiint_{x,y,t\in\psi} \frac{\partial f(x,y,t)}{\partial x}\frac{\partial f(x,y,t)}{\partial y} dxdydt \cdot & \iiint_{x,y,t\in\psi} \left(\frac{\partial f(x,y,t)}{\partial x}\right)^2 dxdydt \end{bmatrix}$$

$$r = \begin{bmatrix} \iiint_{x,y,t\in\psi} \frac{\partial f(x,y,t)}{\partial x}\frac{\partial f(x,y,t)}{\partial y} dxdydt \\ \iiint_{x,y,t\in\psi} \frac{\partial f(x,y,t)}{\partial x}\frac{\partial f(x,y,t)}{\partial y} dxdydt \end{bmatrix}$$

위의 공식으로 움직임 벡타 값을 추정함에 있어 한가지 유념할 것은, 움직임벡타 v가 존재하기 위하여서는 행렬식 W의 inverse 값이 존재하여야 한다. 즉, 행렬식 W가 0이 되어서는 안되는데, 이러한 경우는 함수 f(x,y,t)가 영역 ψ 안에서 상수일 경우에 해당한다. 따라서 이러한 경우를 고려하여, 움직임벡타의 추정을 아래의 몇가지의 경우로 나누어 계산하는 방법이 고안되었다.

(경우 1) 함수의 미분값이 0이 되는 경우 (상수값):

$v = 0,\ \lambda_1,\ \lambda_1 >$ threshold 값

(경우 2) 완전 계단 함수의 경우 (계단 경계방향으로의 미분치가 0인 경우):

$$v = \frac{[\eta_1^T r]\eta_1}{\lambda_1},\ \lambda_1 \gg \lambda_1$$

(경우 3) 일반적인 경우:

$$v = W^{-1} r$$

여기에서 λ_1, λ_2는 W의 Eigenvalue 들을 말하고, η_1, η_2은 이들에 해당하는 Eigenvector 들을 나타낸다.

이러한 Spatio-temporal constraint 방법은 앞서 설명한 BMA 방법에 비해 계산 방법이 간단하다는 장점을 갖고, 적절한 보상회로를 사용하면 noise가 많은 화상이나 깨끗한 화상 모두에, BMA 방식의 결과와 견줄만한 좋은 결과를 얻을 수 있는 것으로 알려져 있다.

Ⅲ. 각 방식의 움직임 추정 방법

1. ATRC 방식 (ADTV)

1) 개요

ADTV에 쓰이는 움직임 추정방식은 MPEG(motion picture expert group)에서 제안된 알고리듬인 half-pixel 정확도를 갖는 BMA(block matching algorithm)를 채택하고 있다. ADTV에서 말하는 움직임 추정은 움직임 정도를 나타내는 2차원의 벡터를 계산해내는 과정을 말하며 이 벡터들을 움직임 벡터라 한다. ADTV에는 2가지의 움직임 추정이 사용되는데 하나는 현재의 화면을 만들기 위해 과거의 화면을 이용하여 forward 움직임 추정을 하는 것이고 다른 하나는 현재의 화면을 만들기 위해 미래의 화면을 이용하여 backward 움직임 추정을 하는 것이다. 두가지 방법 모두 M×N으로 미리 정해진 탐색구간에서 왜곡이 가장 작은 block을 찾아내어 움직임 벡터를 알아내게 된다. 많이 쓰이는 왜곡 측정 방법으로는 MSE(mean square error)와 MAE(mean absolute error)가 있는데 계산상의 간편함과 hardware의 간단함으로 MAE가 주로 쓰인다. Half-pixel 움직임 추정은 두 단계로 이루어지는데 처음의 탐색은 integer-pixel 움직임 벡터를 얻기 위함이고 두번째 탐색으로 처음에 얻은 pixel 단위 움직임 벡터를 중심으로 half-pixel 움직임 벡터를 구하게 된다.

2) Frame별 움직임 추정

ADTV에서는 입력되는 영상을 몇가지 종류로 구분하여 frame 별로 나누는데 I-frame(intra-frame), P-frame(predicted-frame), B-frame(bidirectional-predicted frame) 이 3가지이고 각 frame 별 움직임 추정을 살펴보면

· I-frame : 이 frame은 움직임 추정을 하지않아 움직임 벡터는 없지만 다른 frame의 움직임 추정에 쓰인다.

· P-frame : 이 frame은 바로 전의 I-frame 또는 P-frame을 이용하여 forward 움직임 추정을 하고 다른 P-frame 이나 B-frame 움직임 추정에 쓰인다.

· B-frame : 이 frame은 전후의 I-frame 또는 P-frame을 이용하여 forward 움직임 추정과 back-

ward 움직임 추정을 하고 또 forward 움직임 보정된 frame과 backward 움직임 보정된 forame 둘을 합하여 둘로 나눈 보간된 frame을 만들어 이 3가지 움직임 추정 결과 중 DBD(displaced block difference)가 가장 작은 것을 택하여 움직임 벡터를 정하게 된다. DBD가 가장 작은 것이 보간된 것일 경우에는 forward 움직임벡터와 backward 움직임 벡터 둘 다를 전송하게 된다.

B-frame에서 움직임 벡터를 찾는 것을 그림으로 나타내면 다음과 같다.

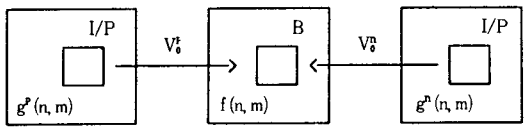


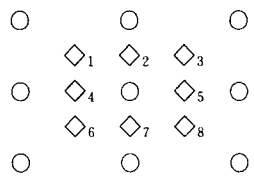
그림 3. B-frame 에서의 움직임 벡터 추정

DBD의 가장작은 값을 찾을 때 보간된 frame의 DBD는 다음과 같이 정의된다.

$$DBD^{pn} = \frac{1}{256}\sum_{m,n=0}^{8}\sum | f(m,n) - \frac{1}{2}$$
$$[g^p(n - v_{ox}^p,\ m - v_{oy}^p) + [g^n(n - v_{ox}^n,\ m - v_{oy}^n)]|$$

3) Half-pixel 움직임 추정

Integer-pixel 완전 탐색 후 8개의 근접 half-pixel 위치에 대해 BMA를 한다.



공간적 보간된 pixel의 값 계산은 다음과 같이 한다.

$$S(x+0.5, y) = (S(x, y) + S(x+1, y))/2$$
$$S(x, y+0.5) = (S(x, y) + S(x, y+1))/2$$
$$S(x+0.5, y+0.5) = (S(x, y) + S(x+1, y) + S(x, y+1) + S(x+1, y+1))/2$$

2. G.I 방식 (Digicipher)

G. I. 사의 Digicipher system 제안서에 나와있는 움직임 추정방식은 앞절에서 설명한 ATRC의 ATDV에 사용된 BMA와 유사하지만 정확도에 있어서 half-pixel 움직임 추정을 하지 않기에 ADTV보다 떨어진다. 그외의 다른점을 살펴보면 우선 backward 움직임 추정을 하지 않고 forward 움직임 추정만을 하여 ADTV의 B-frame이 없다. 즉, Digicipher의 frame은 I-frame과 P-frame만으로 이루어져 있다. 움직임 추정의 단위는 macro block으로 하는데 macro block은 4×2 Y신호 block, U신호 1개 block, V신호 1개 block으로 이루어지므로 이 모든 block들이 같은 움직임 벡터를 가지게 된다.

3. MIT 방식 (ATVA-Progressive HDTV)

MIT에서 제안하는 움직임 추정 방법은 2장에서 설명한 Spatio-temporal constraint 방정식에 바탕을 둔다. 이 방법은 균일한 속도의 전송을 가정으로 하며 물체의 회전, camera zoom, 움직임 물체 전송 영역, 다른 움직임 벡타를 갖는 다중 물체 움직임에서는 유효하지 않다. 작은 구간 내에서 균일한 움직임 전송을 가정하고 움직임 추정이 불명확한 구간에서 움직임 보상을 억제한다 하더라도 Spatio-temporal constraint 방정식에 바탕을 둔 움직임 추정과 보상에 의해 temporal correlation에 많은 감소가 이뤄진다. Block matching algorithm과 비교해서 이 방법은 block 크기에 따라 계산량이 크게 줄어들고 noise가 있을 때나 없을 때나 잘 수행된다.

움직임벡타는 휘도 신호로부터 추정되고 동일한 움직임 벡타는 휘도 신호(Y)와 색차 신호(U & C)에 대해 사용되며, 한 motion vector는 매 32×32 pixel size의 block마다 얻어진다. 각 motion vector는 12bit에 의해 표현되며 이에 요구되는 bit rate은 다음과 같다.

12bits/vector×23×40 vector/frame×60 frame/sec = .6624 Mbits/sec

이전 encoding된 frame과 motion vector로 부터 현재 frame에 대해 prediction이 이뤄지고 encoding된 현재 frame과 prediction된 것과의 차가 계산 되는데 이 차를 motion compensated residual이라 한다. 이것은 각각의 (Y, U, V)세 부분에 대해 얻어지며 상대적으로 image frame에 대해 충분히 큰 에너지를 갖을 때 (화면 변화가 있을 때) motion compensation이 무효화 되고 image frame 자체가 encoding 된다.

4. Zenith and AT &T 방식 (DSC-HDTV)

영상의 특성으로써, 근접한 화소들의 화소값은 일반적으로 매우 근사한 값들로 구성되어 있다. 그 이유는 물체의 경계 및 윤곽을 나타내는 선내의 화소들이 같은 물체를 표시하는데 사용되기 때문이다. Zenith사와 AT&T사는 이러한 영상의 특성을 이용하여 frame 영상 data의 불필요한 redundancy를 제거함으로써 효과적으로 움직임을 추정하는 hierarchical motion estimation (구조 계층적 움직임 추정) 방법을 사용하였다. 또한 transmission channel의 효율을 늘리기 위하여 아래의 그림과 같이 coarse motion estimation과 fine motion estimation의 2단계 motion estimation을 제안하였는데 그 제시된 방법을 구체적으로 분석하여 알아보기로 한다.

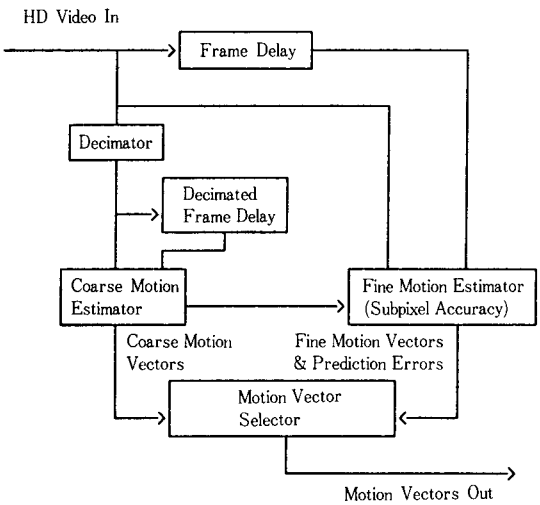


그림 4. 움직임 추정 방법

먼저 filter를 사용하여 frame 영상 신호와 이전 영상 신호의 redundancy를 없앤다. 이로써 resolution (해상도)는 수평,수직 방향으로 각각 1/2씩 줄어들게 된다. 전체 면적이 1/4로 줄어든 저 해상도의 영상 신호로부터의 coarse한 움직임 추정은 32×16 (수평×수직) 화소 크기의 block 단위로 현재 frame과 이전 frame의 화소값의 차이의 절대값인 추정 오차를 비교함으로써 이루어 지는데, 이때 휘도 신호 성분만을 이용하며, 이전 frame block을 탐색 구간 만큼 이동하며 얻은 추정 오차 중 가장 작은 것이 그 block의 움직임 vector가 된다. 이렇게 계층적 저 해상도의 화상에서의 block matching은 원래의 고 해상도 화상에 사용하였을때 보다 탐색 구간과 block size가 1/4로 줄어들어 계산량이 현저하게 줄어든다. 움직임 vector를 구하는 공식은 다음과 같다.

움직임 vector (α, β) 에 대한 추정 오차 :

$$= \sum_{i=0}^{\text{Block Width}} \sum_{j=0}^{\text{Block Height}} \left| I(i, j) - I(i+\alpha, j+\beta) \right|$$

고 해상도 움직임 추정은 저 해상도 화상에서 구한 움직임 vector를 이용하여 원래의 고 해상도 화상에서의 움직임 vector로 대치시키고 이를 다시 고 해상도 움직임 vector로 세분화 시킨다. 즉, 저 해상도 화상에서의 16×8 block size와 저 해상도 움직임 vector는 원래 해상도의 화상에서 수평, 수직방향으로 각각 1/2씩 줄어든 것이므로, 이들을 모두 2배 시키면 원래 화상의 block size와 움직임 vector가 되게 된다. 이는 다시 8×8의 작은 block size로 나뉘어지고, 각각의 작은 block은 저 해상도 움직임 vector를 기점으로 다시 탐색 구간 만큼 이동하여 고 해상도 움직임 vector를 그림 5와 같이 구한다. 움직임 vector encoder는 각각 저 해상도와 고 해상도에서 추정한 움직임 vector를 이용

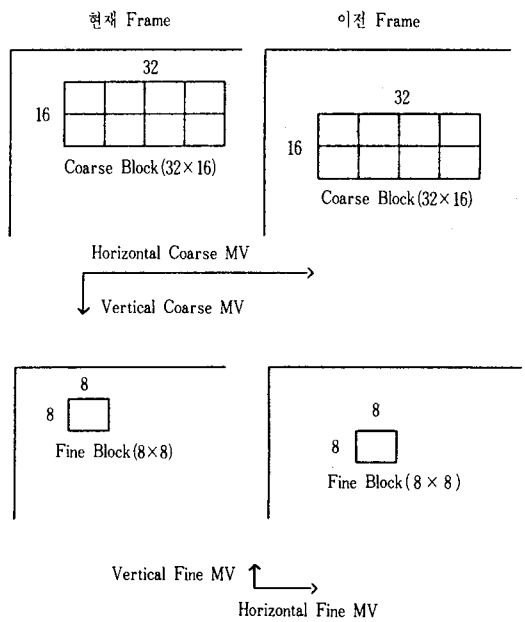


그림 5. 저 해상도와 고 해상도 화상에서의 움직임 추정

하여 다음 frame의 화면을 추정하고 그에 대응되는 움직임 vector를 선택한다. 이는 움직임 vector를 압축하는데 있어서 bit rate를 제한하기 위함인데, channel bandwidth의 여백에 따라 저 해상도 움직임 vector만을 보내기도 하고 저 해상도와 고 해상도 움직임 vector를 함께 보내기도 한다.

Ⅳ. 맺 는 말

이상에서, 완전 디지탈 HDTV의 움직임 추정과 보상 방법이 현재 미국 FCC에 제안되어진 4가지 디지탈 방식을 중심으로 설명되었다. 서술한 바와 같이, block matching algorithm이 계산량이 많기는 하지만 비교적 정확한 움직임 추정 효과를 나타내 주고 있고, 한 방향의 움직임 추정보다는 ATRC에서 제안하고 있는 bi-directional 움직임 추정 방법이 움직임 추정의 정확도를 높혀 주는 것을 볼 수 있다. 이러한 결과는 계산량이 많은 방식의 순으로 정확도가 결정되어 지기 때문에 당연한 귀결이라고 이야기 할 수도 있겠으나, 그만큼 hardware 구현 기술이 결정적 요소 기술임을 말해주고 있다고도 할 수 있다. 다시 말해서, 연산이 복잡해도 그 연산을 실시간에서 처리할 수 있는 device 만 설계할 수 있다면 움직임벡타는 보다 정확하게 추정, 보상할 수 있다는 이야기가 된다. 현재 제안되고 있는 움직임의 추정, 보상 방법이 모두 연산이 복잡하지만 parallel architecture나 pipeline architecture가 가능한 형태의 방식이라는 것을 유념할 필요가 있다. 특별히 중요한 것은, 움직임 보상 방법 그자체 보다도, 어떻게 움직임 추정과 보상을 필연적인 정보 압축에 이용할 것인가 하는 것과 어떻게 그 방법을 hardware로 구현할 것인가 하는 것이니 만큼, 이에 따른 system 차원에서의 연구와 분석이 계속되어져야 할 것이다.

參 考 文 獻

[1] Peter J. Burt, "The Laplacian pyramid as a compact image code," *IEEE Trans. on Communication*, vol. COM-31, no. 4, pp. 532-540, April 1983.

[2] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE*, vol. 73, no. 4, pp. 549-574, 1985.

[3] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[4] Jae S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, 1990.

[5] 이충웅, "HDTV를 위한 운동 검출 및 보상에 관한 기초 연구," 한국과학재단 목적기초연구 제 1차 중간보고서, 1990.

[6] Advanced Digital Television System Description, The Advanced Television Research Consotium, Feb. 1991.

[7] A. Koster, "MPEG Video Simulation Model III", Presented at '90 MPEG Meeting, PTT Research.

[8] 김태정, "HDTV 및 EDTV의 대역 압축 및 신호처리의 기초연구," 서울대학교 공학연구소, May 1991.

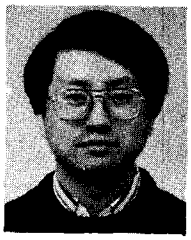
[9] Anil K. Jain, "Image data compression : a review," *Proc. IEEE*, vol. 69, no. 3, March 1981.

[10] "Digital Spectrum Compatible HDTV," Technical Description, Zenith and AT&T, Presented at the HDTV Symposium, Columbia University, April 1991.

[11] "ATVA-Progressice System," Advanced Television Research Program, Research Lab of Electronics, MIT, Feb. 1991.

[12] "Digicipher HDTV System," Technical Description, General Instrument Co., June 1990.

筆 者 紹 介



崔 潤 植

1957年 2月 12日生

1979年 연세대학교 전기공학과(학사)

1984年 미국 Case Western Reserve 대학 시스템공학과(석사)

1987年 미국 Pensylvania State Univ., University Park, 전기공학과(석사)

1990年 미국 Purdue Univ., West Lafayette, 전기공학과(박사)

1979年 ~ 1980年 대한전선주식회사 중전기 사업부

1984年 미국 Univ. of Illinois, Urbana-Champaign, 전기공학과 '핵융합연구실' 연구원

1990年 ~ 현재 현대전자산업(주) 산업전자연구소 연구4실 책임연구원

주관심분야 : HDTV, 영상신호처리, 패턴인식, 통계적 신호처리