

영상의 적응적 워핑 예측을 이용한 웨이브렛 기반의 동영상 부호화

김상준, 지인호

홍익대학교 전자전기컴퓨터공학부

A WAVELET-BASED VIDEO CODING USING ADAPTIVE IMAGE WARPING PREDICTION

Sangjun Kim, Innho Jee

Department of Electronics, Electrical, and Computer Engineering, Hongik University

ijee@wow.hongik.ac.kr

Abstract

기존에 MPEG-1, 2나 H.261/263 등에 사용되고있는 움직임 예측 방법은 블록 기반의 코딩 방식으로 단순히 영상의 움직임을 선형 모델로 간주하고 영상을 일정한 크기의 사각형으로 나누어, 각 사각형이 어느 곳으로 이동하는지를 추정하는 방식이었다. 그러나 이러한 방식은 블록 당 하나의 움직임 벡터만으로 처리하기 때문에 블록 내의 복잡한 움직임은 추정할 수 없을 뿐만 아니라, 보다 일반적인 움직임인 회전, 뒤틀림, 확대, 축소 등을 추정할 수 없다. 또한, 영상의 내용에 상관없이 일정한 크기의 블록으로 나누어 처리하기 때문에 주어진 영상에 최적화된 움직임 추정을 수행하기가 어렵다. 특히 저속 비트율에서는 이러한 점들이 크게 부각된다.

이러한 점들을 극복하기 위해서 여기에서는 삼각형 매쉬를 이용한 공간 변환 방법을 이용하였다. 여기에 영상을 영상에 따라 적응적으로 분해하여 처리할 수 있는 웨이브렛 패킷 부호화를 사용하여 에너지가 많고 적응에 따라 초기 제어점의 격자를 조절하여 좀 더 우수한 성능을 얻을 수 있다.

이 논문은 (2000)년 충남테크노파크 교수실험실창업 연구개발비에 의하여 지원되었음

1. 서론

ITU-T 단기표준화 권고안인 H.263은 기존의 CCITT 표준 영상 압축 알고리즘인 H.261[4]를 공중전화망이나 이동통신망과 같은 협대역망에 적합하도록 개량한 것으로 64 kbps 이하의 초저속 동영상 부호화 권고안이다. 협대역망에서 동영상을 전송하기 위해서 압축률이 높은 부호화 알고리즘을 사용한다. 기존의 H.261 권고안을 64 kbps 환경에 적합하도록 개량한 H.263 권고안은 움직임 추정 및 보상을 이용하여 시간적 중복성을 제거하며 변환 부호화(transform coding)를 이용하여 공간적 중복성을 제거하고 가변길이 부호화를 함으로써 통계적 중복성을 제거하는 복합적인 방법으로 영상을 압축한다. 장기표준화에서는 영역기반 부호화, 웨이브렛 변환 부호화, 객체기반 부호화, 모델기반 및 프렉탈 기반 부호화 등의 고등 알고리즘에 대한 연구가 활발히 진행되고 있다. 단기표준화 권고안인 H.263은 초저속 비트율을 위하여 압축할 때 발생하는 블록화 현상 및 모스키토 현상에 의한 극심한 열화를 피할 수 없기 때문에 초저속 동영상 서비스를 확산시키기 어렵다. 이러한 문제점을 해결하기 위하여 C-cube, LSI-Logic, IIT, CLI, AT&T 등과 같은 세계적인 기업들뿐만 아니라 대학 연구소 등에서 초저속 비트율에 적합한 동영상 압축 알고리즘 개발이 활발히 연구되어 지고 있다.

2. 공간 변환

공간변환은 원래의 좌표계로 표현된 함수로 하나의 좌표계에서 또 다른 좌표계로 변환하는 것을 말한다. 이를 식으로 표현하면 식 (1)과 같다.

$$\hat{I}(x, y) = I(u, v) = I(f_1(x, y), f_2(x, y)) \quad (1)$$

$$\begin{cases} u = f_1(x, y) \\ v = f_2(x, y) \end{cases}$$

블록정합 알고리즘을 공간 변환으로 표현하면, 사상 함수는 다음과 같다.

$$\begin{cases} u = x + a_{00} \\ v = x + b_{00} \end{cases} \quad (2)$$

a_{00} 와 b_{00} 는 각각 수평, 수직 방향의 천이를 의미한다. 따라서 블록정합 알고리즘은 움직임 표현시에 가로, 세로 방향의 2개의 파라미터만을 사용하므로 병진 운동만을 다룰 수 있어 실제의 삼차원 동영상을 정확히 표현하는데 문제가 있다. Affine 변환은 블록정합 알고리즘보다는 더 우수한 자유도를 보장한다. Affine 변환의 사상 함수는 다음과 같다.

$$\begin{cases} u = a_{10}x + a_{01}y + a_{00} \\ v = b_{10}x + b_{01}y + b_{00} \end{cases} \quad (3)$$

Affine 변환은 각 방향에 대해서 3개의 파라미터를 필요로 한다. 이로 인해 affine 변환은 천이는 물론 회전, 확대, 축소, 변형등의 운동까지도 표현이 가능하다[2]

자유도가 더 우수한 변환으로는 bilinear 변환, 2차 공간 변환등이 있으나 자유도가 높을수록 사상함수의 차수가 증가함으로 인해 계산량이 증가하는 단점을 가지고 있다. 따라서 본 논문에서는 공간변환 방법 중 가장 많이 쓰이는 affine 변환을 이용하였다.

3. 적응적 patch 분할

영상을 triangle patch들로 분할하는 방법으로는 영상내의 내용과는 무관하게 일정한 모양으로 patch를 분할하는 방법[8]과 영상내의 물체의 경계나 움직임 정도를 고려하여 적응적으로 다양한 모양으로 patch를 분할하는 방법이 있다. 복잡한 측면에서는 고정된 patch 모양 분할 방법[6]이 유리하나 물체의 경계등에서 왜곡이 발생할 수 있으며 인간의 시각적 측면에서도 좋지 않다. 적응적 patch 모양 분할 방법은 구현하기에는 복잡하지만 움직임을 더 효율적으로 추정할 수 있고 물체의 경계부분을 잘 표현할 수 있으므로 전송률이나 화질측

면에서 우수한 성능을 가진다. 따라서 본 논문에서는 초저속 전송률에 적합한 적응적 patch 모양 분할 방법을 이용하였다. 적응적 patch 분할 과정을 간략하게 보면 다음과 같다.

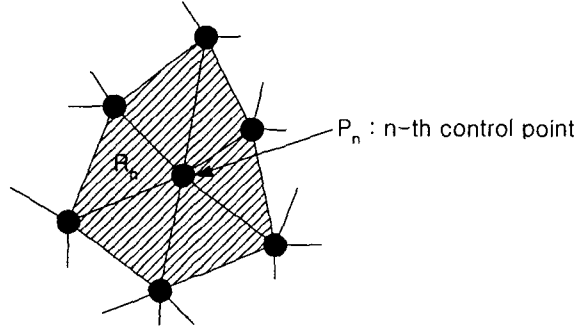


그림 1 : n 번째 제어점에 접하는 영역 R_n

- i) 균일하게 임의의 갯수의 제어점을 영상에 분배한다.
- ii) 식 (4)에 의해 정의된 patch 표현 오차의 차이가 가장 최소인 제어점을 선택하여 제거한다.
- iii) patch 재구성 알고리즘에 의해 제어점이 제거된 영역을 삼각형 patch로 재구성한다.
- iv) 원하는 제어점의 수가 될 때까지 i)~iii)의 과정을 반복한다.

선택된 제어점에 접하는 patch들의 영역 R_n 에서 patch 표현 오차는 다음과 같이 정의 된다.

$$A_n = \frac{1}{N_n} \sum_{p \in R_n} \{f(p) - \hat{f}_{rep}(p)\}^2 \quad (4)$$

where $n = 1, \dots, N$

N_n 은 영역 R_n 안의 화소 수이고, $\hat{f}_{rep}(p)$ 는 $p = (x, y)$ 에서 다음과 같이 계산되는 affine 보간된 화소 값이다.

$$\hat{f}_{rep}(p) = ax + by + c \quad (5)$$

각 patch에서 계수인 (a, b, c) 는 각 patch를 구성하는 제어점과 다음의 선형 연립 방정식에 의해 구해진다.

$$\begin{aligned} f(p_A) &= a x_A + b y_A + c, & p_A &= (x_A, y_A) \\ f(p_B) &= a x_B + b y_B + c, & p_B &= (x_B, y_B) \\ f(p_C) &= a x_C + b y_C + c, & p_C &= (x_C, y_C) \end{aligned}$$

이렇게 구해진 선택된 제어점에 접하는 영역 R_n 에서의 patch 표현 오차와 선택된 제어점을 제거하였을 경우의 표현 오차와의 변화량을 다음과 같이 구한다.

$$\Delta A_n = A'_n - A_n \quad (6)$$

만약, 변화량이 없거나 음수이면 선택된 제어점을 영구히 제거하고 영역 R_n 을 patch 재구성한다. 그리고 재구성 방법은 다음과 같다.

- i) 제거된 제어점과 직접 연결되어 있던 제어점들을 버퍼에 저장한다.
- ii) 이들 중 연속된 제어점 3개로 patch로 구성해 본다.
- iii) 가장 표현 오차가 적은 patch를 새로운 patch를 만든다.
- iv) 새롭게 patch로 구성된 연속된 3점 중, 가운데 제어점을 버퍼에서 제거한다.
- v) 버퍼에 남아있는 제어점이 3개가 될 때까지 과정 i)~iv)을 반복한다.

4. 웨이브렛 변환 부호화

웨이브렛 변환은 최근에 비정체적 동영상 신호를 표현하는데 유용하다는 장점 때문에 동영상 처리 분야에 서 부각되어지고 있다. 동영상 압축에서 DCT와 비교해서 웨이브렛 변환의 장점은 블록화 현상과 반점 현상이 없다는 것이다. 이러한 복원된 영상의 향상된 화질은 초 저속 전송률을 갖는 분야에 적합하다.

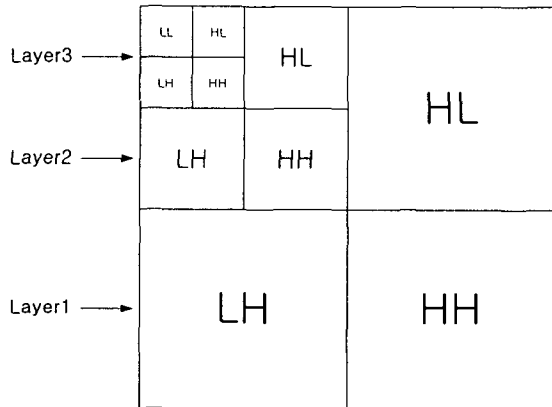


그림 2 : 옥타브 트리 분할과 각 대역의 명칭

웨이브렛 변환을 이용한 동영상 압축 기법은 H.263 권고안에서 DCT/IDCT 대신 웨이브렛 기반으로 대체함으로써 쉽게 적용될 수 있다[5-8]. 웨이브렛 기반 부호화는 DCT 기반 부호화 알고리즘의 단점인 초저속 전송률에서 블록화 현상이 나타나지 않으며, 점진적 전송에 효과적으로 적용될 수 있기 때문에 MPEG-4 장 기표준화로 채택되었다. 일반적으로 이 방법은 각 대역 별로 적절한 부호화 방법과 비트 할당을 통하여 독립적으로 부호화 할 수 있다.

5. 공간 변환을 이용한 움직임 보상

이전 프레임의 영상을 우선 웨이브렛으로 분해한 후 가장 최저 대역의 부영상을 적응적 patch분할을 한다. 이전 프레임의 영상의 patch 분할이 다 이루어지면, 각 제어점을 중심으로 기존의 블록정합 알고리즘의 반화소 법으로 탐색을 한다. 여기서 구한 patch의 구조와 움직임 벡터는 계층적으로 다른 부영상에 적용하게 된다.

그리하여 얻어진 각 제어점의 움직임은 affine 공간 변환식에 따라 다음과 같이 표현 되어진다. 수평방향에 해당하는 변환 계수를

$$d_a = [a_{10}, a_{01}, a_{00}] \quad (7)$$

$$d_b = [b_{10}, b_{01}, b_{00}] \quad (8)$$

이라 하고, 움직였을 것으로 추정되는 좌표를

$[u_1, u_2, u_3], [v_1, v_2, v_3]$ 이라 할 때 다음의 행렬 연산으로 계산이 가능하다[2]

$$\begin{bmatrix} a_{10} \\ a_{01} \\ a_{00} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (9)$$

이렇게 얻어진 공간 변환 계수를 이용하여 화소들을 공간 변환 후, 다음의 식에 의해 보간을 수행한다.

$$I_d(m, n) = \frac{\sum_{p=1}^l r^{-1}(a, b_p) I_d(u_p, v_p)}{\sum_{p=1}^l r^{-1}(a, b_p)}$$

$r(a, b) = \sqrt{(m-u)^2 + (n-v)^2}$ 이고 l 은 보간 될 점으로부터 가장 가까운 점들의 수이다.

6. 실험 및 결과

제안된 동영상 코덱의 성능을 측정하기 위한 테스트 동영상으로 QCIF 포맷(176*144)의 Clair 영상과 Carphone 영상을 사용하였다. QCIF 포맷은 공중전화망(PSTN)과 같은 저 전송률에서의 화상 통신에 사용되는 포맷이다. 각 실험에 사용된 동영상은 초당 5 프레임으로 설정하고 60 프레임을 가지고 실험을 하였다.

Clair 영상은 배경이 존재하지 않는 단순한 움직임의 영상이고, Carphone은 빠르게 지나가는 창문 뒤의 배경과 인물의 움직임이 복합적으로 존재하는 보다 복잡한 구조를 가지고 있는 영상이다. Patch 분할에 의한 affine 공간 변환 방법은 초기 제어점의 개수와 간격은 성능에 많은 영향을 끼친다. 초기 제어점의 수를 360개로 하고 목표 제어점의 수는 99개로 하였다. H.263의 움직임 벡터의 개수가 99개이므로 이 두 가지의 호환을 위해 목표 제어점의 수를 99개로 하였다. 다음은 성능 비교를 위하여 제어점을 균등하게 분배하여 움직임 보상을 수행한 경우와 비교하였다.

| | Clair | Carphone |
|-------------|---------|----------|
| 균등한 patch구조 | 37.33dB | 36.51dB |
| 적응적 patch구조 | 38.64dB | 36.82dB |

표 2. 움직임 예측 오류(PSNR)

실험 결과인 표 1.에서 전체적으로 균등한 patch 구조에 비해 적응적 patch 구조의 움직임 예측 오류가 적은 것을 알 수 있다. 복잡한 동영상인 Carphone의 경우 움직임이 적은 Clair영상에 비하여 성능의 향상이 적은데 그것은 웨이브렛 영역에서의 최상위 부영상의 크기가 매우 작기 때문에 작은 부영상 내에서의 큰 움직임 예측의 오류가 다른 계층에서 누적되기 때문에 발생하는 현상이라고 생각할 수 있다. 그러나 그 만큼 처리 속도가 빨라지므로 처리 속도의 향상이라는 장점이 존재한다. 따라서 제안된 움직임 예측 방법은 공간 변환이 가지는 장점인 회전, 확대, 축소 등의 움직임이 추정 가능하다는 장점과 대신 처리 속도가 느린 점을 절충한 방법이라 할 수 있다.

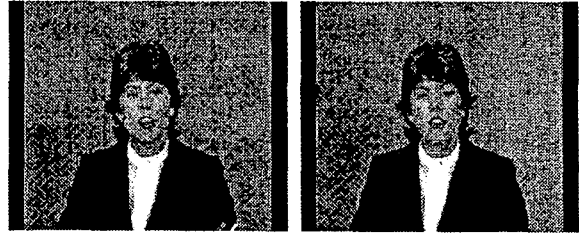


그림 3 : Clair 원 영상 그림 4 : 보상한 영상
참고 문헌

- [1] J. Nieweglowski, T. G. Campbell, and P. Haavisto, "A novel video coding scheme based on temporal prediction using digital image warping", IEEE Trans. on Consumer Electronics, Vol. 39, no. 3, pp. 141-150, Aug. 1993.
- [2] Y. Nakaya and H. Harashima, "Motion Compensation based on spatial transformations," IEEE Trans. on Circuit and Systems for Video technology, vol. 4, no. 3, June 1994.
- [3] Yuichiro Nakaya and Hiroshi Harashi Harashima, "An iterative estimation method using triangle patches for motion compensation," Proc. SPIE VCIP'95, Vol. 1605, pp. 546-557, 1991.
- [4] CCITT Recommendation H.261, "Video codec for audio-visual service at Px64kit/s," July 1990.
- [5] O. Johsen, O. V. Shentov and S. K. Mitra, "A technique for the efficient coding of the upper bands in subband coding of images," Proc. of ICASSP/ pp. 2097-2100, April 1990.
- [6] Stephane G. Mallat, "A theory for multiresolution signal decomposition : The wavelet representation," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11, no. 7, July 1989.
- [7] R. de Querioz, C. Choi, Y. Huh, J. Hwang and K. R. Rao, "Wavelet transforms in a JPEG-like image coder," Proc. SPIE VCIP'94, Vol. 2308, pp. 1662-1673, Sep. 1994.
- [8] Ya-Qin Zhang and Sohail Zafar, "Motion compensated transform coding for color video compression," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 2, no. 3, Sep. 1992.