

적응적 부분 정합 방법을 이용한 영상 비틀림 방법

正會員 林 東 根*, 扈 堯 盛*

Image Warping using An Adaptive Partial Matching Method

Dong-Keun Lim*, Yo-Sung Ho* *Regular Members*

요 약

본 논문에서는 가변 탐색 영역을 가진 부분 정합 방법을 적용한 새로운 움직임 추정 방법을 제안한다. 거친 움직임을 추정할 때에 탐색 영역을 고정하지 않고 인접 영상간의 차이(FD; frame difference)에 대한 신호 대 잡음 비(PSNR; peak signal-to-noise ratio)를 사용하여 탐색 영역의 크기를 조정한다. 육각형 정합 방법은 영상 비틀림에서 미세 보정을 위한 방법 중의 하나인데, 이것은 최적의 영상 화질 개선을 가져오지만 계산량이 많은 것이 단점이다. 제안된 부분 정합 방법은 육각형 정합 방법에 대하여 영상 화질면에서는 구분할 수 없을 정도이면서도 계산량은 약 50% 이하로 줄었다. 움직임 추정 및 보상 방법의 성능은 어파인 변환과 쌍선형 변환, 그리고 제안한 움직임 추정 알고리즘들을 함께 사용하여 계산량, 전송해야 될 비트량, 복원 영상의 화질의 3가지 점을 동시에 고려하여 이전의 방법과 비교하였다. 제안된 방법을 사용하여 복원된 영상의 화질은 기존의 블럭 정합 방법과 비교할 때 월등히 우수하고, 육각형 정합 방법과 비교될 만하였다. 또한 계산량과 부호화에 사용된 비트의 양은 블럭 정합 방법과 이미 개발된 육각형 정합 방법같은 영상 비틀림 방법들보다 상대적으로 많이 감소하였다.

ABSTRACT

This paper proposes a new motion estimation algorithm that employs partial matching in a variable search area. Instead of using a fixed search range for coarse motion estimation, we examine a varying search range, which is determined adaptively by the peak signal to noise ratio (PSNR) of the frame difference. The hexagonal matching method is one of the refined methods in image warping. It produces improved image quality, but it requires a large amount of computations. The proposed adaptive partial matching method reduces computational complexity below about 50% of the hexagonal matching method, while maintaining the image quality comparable. The performance of two motion compensation methods, which combine the affine or bilinear transformation with the proposed motion estimation algorithm, is evaluated on the following criteria: computational complexity, number of

*광주과학기술원 정보통신공학과
論文番號:97238-0721
接受日字:1997年 7月 21日

coding bits, and reconstructed image quality. The quality of reconstructed images by the proposed method is substantially improved relative to the conventional BMA method, and is comparable to the full hexagonal matching method; in addition, computational complexity and the number of coding bits are reduced significantly.

I. 서 론

영상의 움직임 추정과 움직임 보상은 영상 부호화의 발전에 크게 기여하였으며, 점점 그 중요성이 더해가고 있다. [1, 19-23]. 물체의 움직임을 추정하기 위해 이를 수학적으로 모델화하는데, 계산의 간편함과 실시간적인 처리에 대한 요구 때문에 지금까지는 주로 병진 운동만을 고려한 움직임 추정 방법[3-7]이 연구되었다. 병진 운동은 연속된 두 개의 영상 사이에 움직임을 편차를 발생시킨다. 연속한 두 영상간의 편차를 예측하기 위해 여러 가지 방법이 제안되었으며, 편차를 표현하는 움직임 벡터에 기반한 움직임 보상 예측, 움직임 보상과 변환 부호화, 그리고 적응적인 움직임 보간 방법이 사용되고 있다.

최근에는 원래 음성이나 단순한 문자 정보만을 전송하기 위해 사용되던 저속전송율의 통신 채널을 통하여 영상 통신을 하려는 요구가 늘어나고 있다. 이러한 저속 채널은 범용 통신 채널(PSTN), 라디오 채널, 64kbps 이하의 전송율을 요구하는 전자 메일을 전송하는 컴퓨터 네트워크까지도 포함한다. 따라서 이와같이 다양한 통신 채널을 모두 고려하기 위해 넓은 범위의 전송율을 지원하는 영상 부호화기의 개발이 점점 더 중요해지고 있다.

저속 전송율의 영상 통신을 위한 주된 기술적인 요소는 충분한 영상 화질을 얻으면서도 가능하면 전송할 비트 수를 줄이고 구현에 적절한 하드웨어 구조를 가지는 영상 압축 방법이다. 지금까지는 블럭 정합 움직임 추정 방법[3-7, 23]과 DCT[8, 18-22]를 사용하여 충분한 정도의 성능을 얻을 수 있었다. 블럭 정합 움직임 추정 방법은 영상을 균일한 블럭으로 나누어 각 블럭 단위로 움직임을 추정하는 움직임 모델을 사용한다. 그런데 이 방법은 실세계를 표현할 만큼의 충분한 정보를 전송하지 못할 때 블럭간의 경계면에서 불연속성과 블럭 현상을 발생시킨다. 따라서 저속 통신에서 원하는 저속 전송율에 맞추다 보면 영상 화질의 많은 저하를 가져온다. 이러한 단점을 보완하기 위하여 좀 더 정밀한 움직임 모델을 사용하는 다양한

부호화 방법이 제안되었다. 3차원 모델은 그 중에서 가장 정밀한 것으로서 모델에 기반한 부호화 방법[21, 23]에 사용된다. 실제로 3차원 모델은 실세계를 아주 적절하게 표현할 수 있지만 오늘날의 영상 분석 기술로는 실세계로부터 정확한 움직임 정보를 추출하는 것은 여전히 어려운 문제이다.

실제적인 응용을 위해서는 일반적으로 사용되는 모델보다는 단순하지만 기존의 블럭 정합 방법보다는 좀 더 정확한 모델이 사용된다. 움직임 모델로서 어파인 변환(affine)[17, 21, 23]과 쌍선형 변환(bilinear)[17]이 많이 사용된다. 어파인 변환을 사용한 영상 비틀림 방법(image warping)[14-16, 24]은 영상을 삼각형으로 쪼갠 다음, 각 삼각형 꼭지점 위치의 움직임 벡터를 예측하고 이 값을 어파인 변환에 적용하여 삼각형 내부의 움직임 벡터를 예측 보상한다. 따라서 이 방법은 삼각형의 변형을 가져오며 인접한 삼각형들과의 연결 구조(topology)가 매우 중요하다. 이와 같은 연결 구조 덕분에 기존의 블럭 정합 방법에서 나타날 수 있었던 블럭간의 불연속성은 나타나지 않고 블럭 현상도 감소된다. 삼각형 내의 움직임을 표현하는 어파인 변환은 3차원 공간상 물체의 움직임을 2차원 영상에 투영(projection)한 것과 같다. 해당 움직임 정보를 전송하는 방법으로는 어파인 변환의 6개 변수를 전송하는 방법이 있지만, 제한된 비트를 할당하여 표현상의 정밀도가 많이 감소하므로 세 꼭지점의 움직임 벡터를 전송하는 것이 더 유리하다. 비슷한 방법으로 쌍선형 변환에서는 사각형 구조를 사용하여 움직임을 추정한다[15].

본 논문에서 제안한 것은 영상 비틀림을 사용하는 데 거친(coarse) 움직임을 추정한 후, 좀 더 미세(refine) 움직임 추정을 추가적으로 적용하며 그 과정에서 계산량을 줄이고 압축율을 증가시키면서도 영상 화질은 크게 떨어뜨리지 않는 방법이다. 또한 수학적 선계를 통하여 고속의 어파인 변환을 이루는 방법을 연구하였는데, 결과적으로는 벡터 관계를 이용한 변환과 동일한 형태를 이루는데 유도 과정에서 몇 가지의 실수를 추가로 고려하면 어파인 변환보다 더 정밀

한 변환이 쉽고 빠르게 이루어질 수 있음을 시사한다. 아울러 지금까지는 변환의 관점에서 당연하다고 간과되었던 사항을 증명하였다. 또한 지금까지의 관련 논문들은 영상 화질의 개선에만 관심을 두었기 때문에 계산량은 상대적으로 엄청나게 증가하여 실시간적인 처리를 요구하는 저속 영상 통신에서 사용을 꺼리게 만들고 단지 미래에 구현 가능한 속제로 남겨졌다. 이에 반하여 본 논문에서 주안점을 둔 것은 영상 화질, 전송 정보량, 계산량의 3가지 관점에서 모두 적절한 결과를 얻는데 있으며, 나아가서는 현재의 H.261, H.263이나 MPEG의 구조[2]에 약간의 변형만을 취하여 실시간적인 응용하는데 그 목적을 두었다. 이 점은 연구 과정에서 많이 고려되었던 부분으로서 다양한 방법이 검토되었지만, 3가지 관점을 모두 만족시킬 수 있는 방법은 찾지 힘들어서 적절한 결과를 가져오는 구조를 제시하였다.

논문의 구성은 2장에서는 영상 비틀림 방법의 개념과 적용 방법에 관한 간략한 설명을 하였고, 3장에서는 수학적 전개와 제안한 방법에 대한 설명을 하였다. 4장에서는 3가지 관점에서 제안한 방법의 성능을 보여주며 실시간적인 처리에 사용될 수도 있음을 보여준다.

II. 영상 비틀림 방법

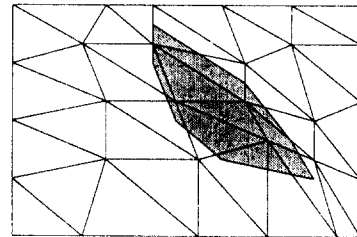
2.1 영상 비틀림 방법의 개념 (Concept of Image Warping)

움직임 보상을 이용한 영상 부호화의 구조에서 현재의 예측 영상은 이전의 영상을 사용하여 기하학적인 변환[11-13, 17]을 취함으로써 구해지는데 이러한 방법을 통칭하여 영상 비틀림 방법이라고 부른다. 변환은 이전의 영상과 예측되는 영상간의 화소 위치에 대한 공간적인 관계를 규명한다. 이론적으로는 하나의 변환 방법이 전체 영상에 대하여 적용될 수도 있지만 모든 화소에 대하여 적절한 정밀도를 가지는 변환 방법을 찾는 것은 계산량도 많고 그 결과의 정확성도 자신할 수 없다. 게다가 다항식의 형태로 표현되는 움직임 필드는 전형적으로 많은 불연속성을 가지고 있다. 따라서, 보다 나은 접근 방법으로는 상대적인 불연속성 감소를 위해 영상을 여러 개의 블럭으로 나누고 각각의 영역에 대하여 서로 다른 변환 계

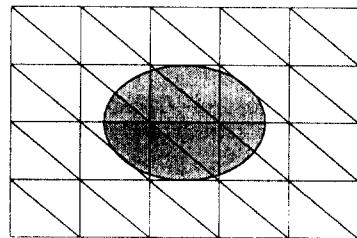
수를 전송하는 것이다. 즉, 격자점 혹은 꼭지점이라고 불리는 영상의 화소점 위치 값에 대한 공간적인 관계, 즉 상대적 편차를 나타내는 움직임 벡터를 복호 화기에 전송한다. 이러한 화소점들의 위치는 변환에서는 고정된 것으로 사용한다. 영상을 서로 중복되지 않는 블럭으로 나눌 때, 격자점은 인접한 몇 개의 블럭에서 공유되며 해당되는 격자점 짝들간의 편차는 움직임 보상 과정에서 예측된다.

위에서 설명한 방법은 변환을 수행할 때, 다음과 같은 장점을 가진다.

- (1) 작게 나눈 블럭에 대해서 변환을 적용하는 것은 계산량을 감소시킨다.
- (2) 변환 계수를 찾는 것은 개개의 영상 화소에 대한 편차를 추정하는 문제로 간략화된다.
- (3) 변환 계수를 전송하는 것은 현존하는 코덱 (codec)에 간단한 변화를 주어서 격자점의 편차를 전송하는 것과 동일하다.



(a) 불규칙한 조각으로 쪼개진 이전 영상
Irregular Patched Previous Frame



(b) 규칙적인 조각으로 쪼개진 현재 영상
Regular Patched Current Frame

그림 1. 영상 비틀림에서 변형된 조각
Fig 1. Deformed Patch in Image Warping

그림 1은 예측 방법에 대하여 보여준다. 현재의 영상은 균일한 간격으로 격자점(꼭지점)을 선택하고 이전의 영상에서 해당 격자점이 어느 위치로 이동하였는지 추정한다. 이 그림에서 보듯이 이전 영상에 있는 삼각형이 변형된다. 기하학적인 변환은 예측 영상을 만들기 위하여 이전 영상의 각 블록에 대하여 적용된다. 균일한 병진운동만을 가진 꼭지점의 편차는 고전적인 블록 정합 방법에서 구한 것과 비슷한 결과를 보이지만, 영상 비틀림의 경우에 블록 꼭지점들의 움직임의 상대적인 차이는 블록의 찌그러짐을 가져오며 물체의 모양을 자유자재로 변형시킬 수 있다. 예측 과정은 크게 격자점의 선택, 화소값을 사용한 정합, 블록 단위로 변환의 3단계이다.

그림 2는 실제적인 변환의 예인데, 하나는 영상 전체에 어파인 변환을 적용하여 병진운동과 회전운동, 그리고 확대축소 운동을 한 것이고, 다른 하나는 몇 개의 블록에만 3가지 운동을 적용한 것인데, 특히 얼굴의 코와 입 부분이 많이 변형되었음에 주목하라. 어떤 의미에서 보면 영상 비틀림 방법은 꼭지점에 대하여 광류(optical flow) 방법[9-10]을 적용하여 간략화시킨 형태로도 볼 수 있다.

2.2 수학적인 표기법 (Mathematical Notation)

움직임 보상은 블록이나 조각이라고 불리는 지역적인 영역으로 영상을 쪼갬 다음 각 영역의 움직임 요소들을 추정한다. 영역은 삼각형이나 사각형등의 다양한 블록으로 분류할 수 있다. 앞에서 설명한 과



(b) 특별히 얼굴 부분만 어파인 변환 적용
Affine Transformation on a Particular Part of Face

그림 2. 어파인 변환을 사용한 영상 비틀림의 예
Fig. 2. An Example of Image Warping using Affine Transformation

정은 복원된 이전 화면인 (k-1) 번째 영상 $\tilde{I}_{k-1}(x', y')$ 을 사용하여 현재 화면인 k 번째 화면 $\tilde{I}_k(x, y)$ 를 예측하는 것으로서 다음과 같이 수식으로 표현할 수 있다.

$$\tilde{I}_k(x, y) = \tilde{I}_{k-1}(x', y') = \tilde{I}_{k-1}(f(x, y), g(x, y)) \quad (1)$$

여기서 $\tilde{I}_k(x, y)$ 와 $\tilde{I}_{k-1}(x', y')$ 의 기하학적인 관계는 $x' = f(x, y)$, $y' = g(x, y)$ 으로 표현된다.

Ⅲ. 수학적인 분석과 제안한 구조

3.1 벡터 관계를 이용한 고속 어파인 변환 (Fast Affine Transformation using Vector Relation)

영상 비틀림에서 행렬을 사용한 변환[1]은 어떤 경우는 그것의 역행렬이 존재하지 않을 수도 있다. 즉 행렬의 판별식 값이 0인 경우가 있는데, 이때는 해가 존재하지 않게 된다. 또한 역행렬을 구한 다음 이것과 움직임 벡터의 값을 알고자 원하는 위치 값과의 곱셈 연산을 수행하므로 많은 계산 시간이 걸린다. 그래서, 벡터 관계를 이용한 고속 어파인 변환을 유도한다.

먼저 변형 후에 비율을 나타내는 값 p, q가 보존되는 지를 검토한다. 만약 이것이 변하면 현재 화면에



(a) 영상 전체에 어파인 변환 적용
Affine Transformation on Whole Image

대하여 벡터 관계를 이용한 어파인 변환을 통해 구해진 값과 이전 화면에 대하여 동일한 방법으로 구한 값과 합치되지 않는 빈 공간이 존재하므로 지금의 변환이 무의미하게 된다. 따라서 이 점을 확인하는 것이 매우 중요하다.

그림 3의 이전 화면의 삼각형 조각으로부터 다음과 같은 관계를 얻을 수 있다.

$$q_{k-1} = \frac{\sqrt{(u_q - u_1)^2 + (v_q - v_1)^2}}{\sqrt{(u_0 - u_1)^2 + (v_0 - v_1)^2}} \quad (2a),$$

$$q_k = \frac{\sqrt{(x_q - x_1)^2 + (y_q - y_1)^2}}{\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}} \quad (2b)$$

그리고 어파인 변환은 다음과 같이 표현할 수 있다.

$$\begin{aligned} x &= a_{11}u + a_{21}v + a_{31} \\ y &= a_{12}u + a_{22}v + a_{32} \end{aligned}$$

혹은 $[x, y, 1] = [u, v, 1] \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix} \quad (3)$

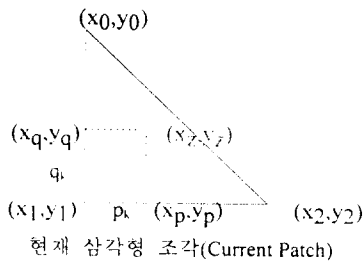
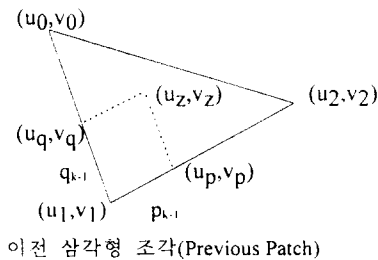


그림 3. 변형된 삼각형 조각
Fig 3. Deformed Triangular Patch

각 꼭지점에 대하여 어파인 변환을 적용하면 다음과 같다.

$$\begin{aligned} x_q &= a_{11}u_q + a_{21}v_q + a_{31} & x_1 &= a_{11}u_1 + a_{21}v_1 + a_{31} \\ y_q &= a_{12}u_q + a_{22}v_q + a_{32} & y_1 &= a_{12}u_1 + a_{22}v_1 + a_{32} \\ x_0 &= a_{11}u_0 + a_{21}v_0 + a_{31} \\ y_0 &= a_{12}u_0 + a_{22}v_0 + a_{32} \end{aligned} \quad (4)$$

여기서 변형 후의 비율을 나타내는 값인 q_k 를 구하면 다음과 같이 표현된다.

$$q_k = \frac{(a_{11}^2 + a_{12}^2)(u_0 - u_1)^2 + 2(a_{11}a_{21} + a_{12}a_{22})(u_0 - u_1)(v_0 - v_1) + (a_{21}^2 + a_{22}^2)(v_0 - v_1)^2}{(a_{11}^2 + a_{12}^2)(u_q - u_1)^2 + 2(a_{11}a_{21} + a_{12}a_{22})(u_q - u_1)(v_q - v_1) + (a_{21}^2 + a_{22}^2)(v_q - v_1)^2} \quad (5)$$

병진 운동, 회전 운동, 확대축소 운동을 행렬 연산을 통하여 수행하는 어파인 변환을 검토해보자. 일반적으로 말해서 행렬 연산을 할 때 연산 순서는 매우 중요한데 서로 다른 결과를 만들기 때문이다. 병진 운동 행렬을 D, 회전 운동 행렬을 R, 그리고 확대축소 운동 행렬을 S라고 하면, 가능한 행렬 연산의 경우는 다음과 같다.

$$M_{comp1} = D. R. S \quad (6a),$$

$$M_{comp2} = R. S. D \quad (6b),$$

$$M_{comp3} = R. D. S \quad (6c),$$

$$M_{comp4} = S. D. R \quad (6d),$$

$$M_{comp5} = D. S. R \quad (6e),$$

$$M_{comp6} = S. R. D \quad (6f)$$

(6)과 같이 6가지의 서로 다른 경우가 존재하지만 처음 2x2의 부 행렬의 값은 다르지 않다. 그 값은 두개의 군을 이루는데, 하나는 (6a)에서 (6c)까지이고, 다른 하나는 (6d)에서 (6f)까지이다.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} S_u \cos \theta & S_v \sin \theta \\ -S_u \sin \theta & S_v \cos \theta \end{bmatrix} \quad (7a),$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} S_u \cos \theta & S_u \sin \theta \\ -S_v \sin \theta & S_v \cos \theta \end{bmatrix} \quad (7b)$$

이 결과로부터,

$$a_{11}^2 + a_{12}^2 = S_u^2 \cos^2 \theta + S_v^2 \sin^2 \theta, a_{21}^2 + a_{22}^2 = S_u^2 \sin^2 \theta + S_v^2 \cos^2 \theta$$

$$a_{11} a_{21} + a_{12} a_{22} = \sin \theta \cos \theta (S_v^2 - S_u^2) \quad (8a)$$

$$a_{11}^2 + a_{12}^2 = S_u^2 \cos^2 \theta + S_u^2 \sin^2 \theta = S_u^2,$$

$$a_{21}^2 + a_{22}^2 = (-S_v \sin \theta)^2 + S_v^2 \cos^2 \theta = S_v^2 \quad (8b)$$

$$a_{11} a_{21} + a_{12} a_{22} = -S_u \cos \theta \cdot S_v \sin \theta + S_u \sin \theta \cdot S_v \cos \theta = 0$$

그리고나서 (8)식을 (5)식에 대입하고, 만약 $S_u = S_v$ 인 경우, 즉 각 축에 대하여 동일한 비율을 적용하면 다음의 (9)식과 같이 된다. (이 조건 때문에 어파인 변환은 쌍선형 변환의 특별한 경우가 되며 다른 조건을 적용하여 여러 변환을 유도할 수도 있지만 여기에서는 언급하지 않는다.)

$$q_k^2 = \frac{(S_u^2 \cos^2 \theta + S_v^2 \sin^2 \theta) (u_q - u_1)^2 + (S_u^2 \sin^2 \theta + S_v^2 \cos^2 \theta) (V_q - V_1)^2}{(S_u^2 \cos^2 \theta + S_v^2 \sin^2 \theta) (u_0 - u_1)^2 + (S_u^2 \sin^2 \theta + S_v^2 \cos^2 \theta) (V_0 - V_1)^2}$$

$$= \frac{S_u^2 (u_q - u_1)^2 + S_v^2 (V_q - V_1)^2}{S_u^2 (u_0 - u_1)^2 + S_v^2 (V_0 - V_1)^2} = q_{k-1}^2 \quad (9a)$$

$$q_k^2 = \frac{S_u^2 (u_0 - u_1)^2 + S_v^2 (V_0 - V_1)^2}{S_u^2 (u_0 - u_1)^2 + S_v^2 (V_0 - V_1)^2}$$

$$= \frac{(u_q - u_1)^2 + (V_q - V_1)^2}{(u_0 - u_1)^2 + (V_0 - V_1)^2} = q_{k-1}^2 \quad (9b)$$

비율을 나타내는 값 p에 대해서도 동일한 유도가 가능하며 변형 후에도 비율 p, q는 보존된다.

이제부터 벡터 관계를 이용한 어파인 변환을 유도하자.

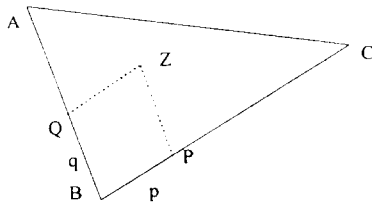


그림 4. 3개의 선분으로 구성된 삼각형
Fig 4. Triangle with Three Lines

그림 4에서 각 점은 다음과 같이 표현된다.

$$A = (x_A, y_A), B = (x_B, y_B), C = (x_C, y_C), P = (x_P, y_P),$$

$$Q = (x_Q, y_Q), Z = (x_Z, y_Z)$$

$$x_p = p x_c + (1-p) x_b \quad x_q = q x_A + (1-q) x_B \quad (10)$$

$$y_p = p y_c + (1-p) y_B \quad y_q = q y_A + (1-q) y_B$$

직선 BC와 직선 QZ은 동일한 기울기를 가지고, 직선 BA와 직선 PZ은 동일한 기울기를 가짐을 알 수 있다. 따라서 다음과 같이 전개할 수 있다.

$$a_{PZ} = \frac{y_Z - y_P}{x_Z - x_P} = \frac{y_B - y_A}{x_B - x_A} = a_{BA} \quad (11a)$$

$$a_{QZ} = \frac{y_Z - y_Q}{x_Z - x_Q} = \frac{y_B - y_C}{x_B - x_C} = a_{BC} \quad (11b)$$

$$(y_B - y_A) x_Z - (x_B - x_A) y_Z = (y_B - y_A) x_P - (x_B - x_A) y_P \quad (12a)$$

$$(y_B - y_C) x_Z - (x_B - x_C) y_Z = (y_B - y_C) x_P - (x_B - x_C) y_P \quad (12b)$$

이 식은 다음과 같이 정리된다.

$$x_Z = p(x_C - x_B) + q(x_A - x_B) + x_B,$$

$$y_Z = p(y_C - y_B) + q(y_A - y_B) + y_B \quad (13)$$

이것은 다음과 같이 각 꼭지점을 벡터로 취급하였을 때의 결과와 동일하다.

$$Z = p(C - B) + q(A - B) + B = (1-p-q)B + pC + qA \quad (14)$$

3.2. 적응적 부분 정합 방법 (Adaptive Partial Matching Method)

Nakaya와 Harashimaga 제안한 육각형 정합 방법 [14]은 지금까지 개발된 블럭 정합 방법이나 영상 비틀림 방법 중에서 복원 영상 화질이 가장 좋다[24]. 그렇지만 이것은 많은 계산량을 요구하므로 실시간적으로 부호화하기는 어렵다. 그래서 이 방법과 비교하여 복원 영상의 화질은 약간 감소하지만 계산량을 줄이는 방법인 적응적 부분 정합 방법을 제안한다.

먼저 탐색 영역을 고려해보자. 블럭 정합 방법의 PSNR은 탐색 영역을 크게하면 할수록 계속 증가하지만 어느 지점에 이르면 아주 조금씩 증가하거나 포화된다. 블럭 정합 방법과 다르게 영상 비틀림 방법

은 실제적인 예측 오차, 즉 움직임 보상의 결과로서의 예측 오차를 사용하여 움직임 벡터를 추정하지 않기 때문에 블럭 정합 방법의 경우와 다른 결과를 가져온다. 블럭 정합 방법과 달리 불필요하게 탐색 영역을 증가시킨다고 하여 반드시 복원 화질이 개선되지는 않는다. 이 점을 감안하면 적절한 탐색 영역이 결정되며 그림 5과 같이 거친 움직임 추정이 적용된다.

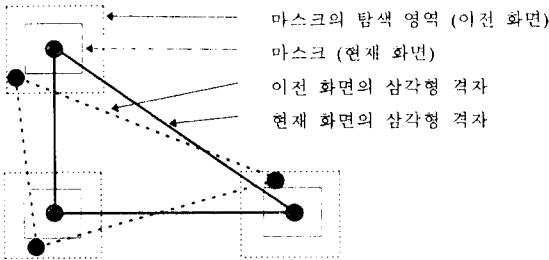


그림 5. 거친 움직임 추정에 의한 변형 (마스크는 블럭 정합 방법의 블럭과 비슷한 의미)
 Fig 5. Deforming Process for Coarse Motion Estimation (Mask has the Same Meaning of Block in BMA)

그림 6은 영상 비틀림 방법에서 탐색 영역을 무조건적인 증가가 반드시 PSNR을 증가시키지는 않음을 보여준다. 그리고 인접한 영상간의 차이인 FD의 PSNR이 작으면 탐색 영역은 더 커야함을 보여준다. (Miss America만 예를 들었지만 다른 영상 군도 유사한 결과를 보인다.)

격자 구조에 기반한 부호화에서 기하학적인 변환의 목적은 각 꼭지점의 움직임 벡터값들을 사용하여 보간하므로 예측된 움직임 벡터의 오차를 줄이는데 있다. 그런데, 한 꼭지점에 대해서만 오차를 줄여가다 보면 인접한 꼭지점에 너무 근접하고 심한 경우 그림 7과 같이 엇갈리며 겹침이 발생하여 격자의 연결 상태를 파괴한다. 따라서 그림 6과 같은 결과가 발생한다.

그림 8과 같이 FD값으로부터 적용적인 탐색 영역을 선택할 수 있다. 본 논문에서는 CIF (352x288) 크기를 가진 Miss America (missa), Claire, Salesman (gs)의 세가지 영상군을 사용하여 탐색 영역에 따른 성능을 검토한 후 최적값만으로 된 본포를 얻었다. 적절한 탐색 영역에 대하여 모델화를 위하여 LMS (least-mean square) 방법을 사용하였는데 1차나 2차 근사

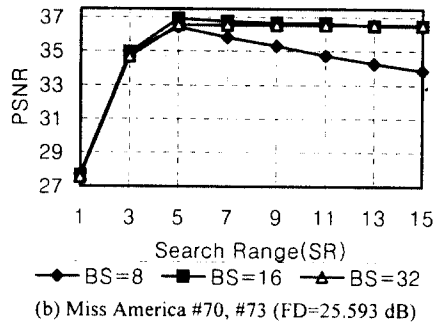
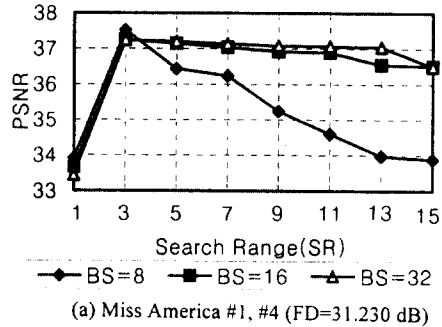


그림 6. 블럭 크기에 따른 PSNR 과 탐색 영역(SR)의 관계 (Miss America)
 Fig 6. PSNR vs. Search Range(SR)

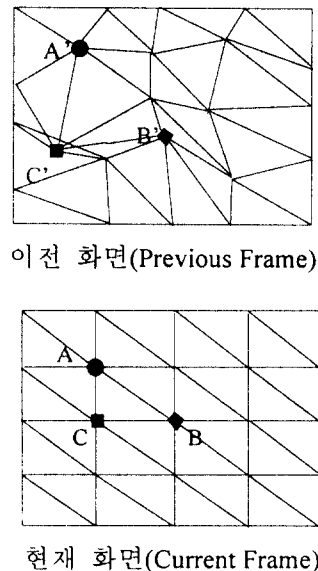
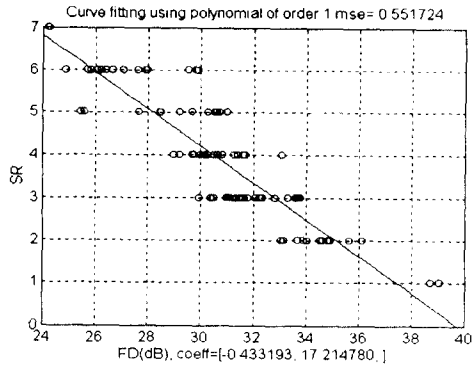
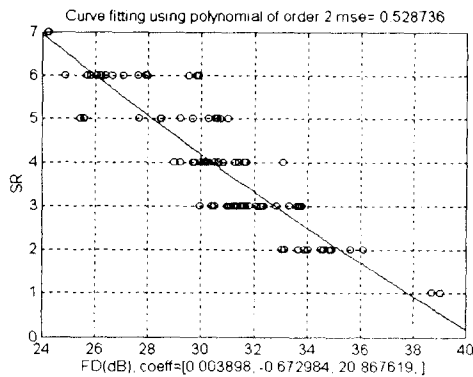


그림 7. 겹침에 의하여 깨어진 연결 상태
 Fig 7. Broken Topological Structure by Flip-over



(a) 1 차 근사

$$SR = -0.433193 FD + 17.214780$$



(b) 2 차 근사

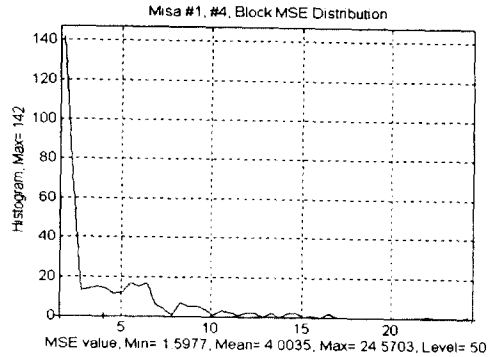
$$SR = 0.003898 FD^2 - 0.672984 FD + 20.867619$$

그림 8. 적절한 탐색 영역을 위한 여러 차수의 다항식 근사
Fig 8. Optimal Search Range(SR) for Several Order Polynomial Fitting

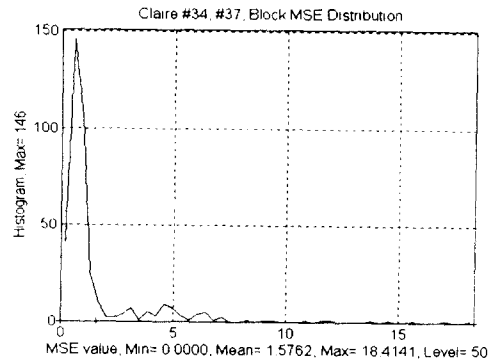
정도로 충분하다.

이제 앞의 성질을 이용한 적응적 부분 정합 방법을 제안한다. 그림 9에서 보듯이 FD 영상에 대하여 블럭 단위로 평균 MSE (Block MSE) 분포를 보면 대부분의 FD영상은 값이 작은 부분이 값이 큰 부분보다 많다는 것을 알 수 있다. BMSE 값이 작은 부분은 인간 시각이 잘 느끼지 못하는 부분이므로 이전 화면을 그대로 사용하고 값이 큰 부분에 대하여만 정합을 수행한다.

send_MSE_map, send_CP_map, Refine_CP_map 등 여러 가지 지도를 만드는 예를 그림 10에서 보여



(a) Miss America #1 and #4간의 차 영상



(b) Claire #34 and #37간의 차 영상

그림 9. 차 영상의 블럭 MSE 분포

Fig 9. Block MSE Distribution for Image Sequences

준다. 그림 (a)는 FD 영상을 동일한 블럭으로 쪼갠 후 각각의 BMSE를 구하여 큰 순서로 번호를 붙여둔다. 채워진 점은 블럭 보상 시에 한 개 이상의 블럭에서 사용하는 꼭지점이고 다른 비어있는 점은 하나의 블럭에 대해서만 사용되는 꼭지점을 의미한다. 그림 (b)는 그림 (a)에 대하여 send_CP_rate에 따라서 큰 블럭부터 지도 부분에 'I'로 표시한다. 그림 (c)는 그림 (b)의 블럭에 대하여 미세 보정을 할 때, 추가로 사용되는 꼭지점을 포함한 지도이다. 그림 (d)에서 'O'표시는 그림 (c)의 send_CP_map에 의하여 설정되는 전송될 꼭지점을 의미하고 'X' 표시는 미세 보정을 할 때 사용되지만 전송하지 않는 꼭지점을 의미한다. 그림 11은 제안한 능동 부분 정합 방법의 흐름도이다.

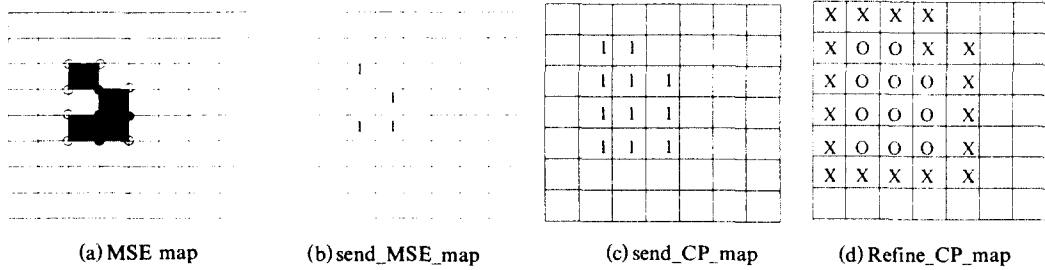


그림 10. 사용된 여러 가지map의 예
Fig 10. Several Used Map Diagram Case

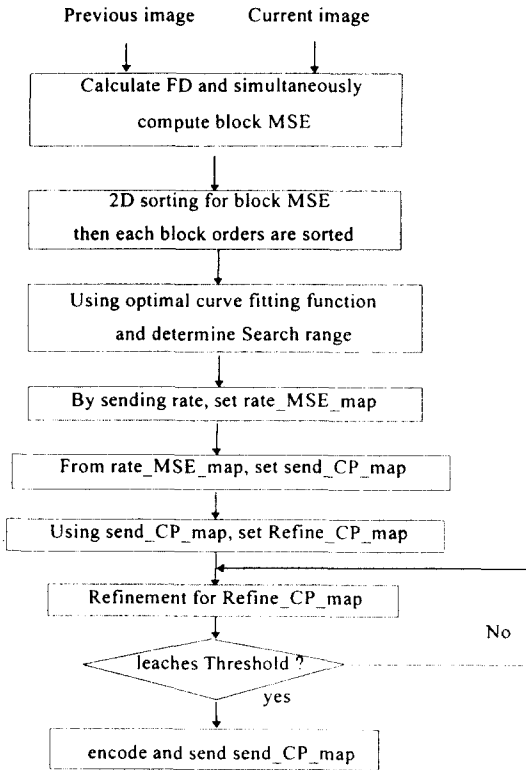


그림 11. 제안한 적응적 부분 정합 방법의 흐름도
Fig 11. Flowchart of Proposed Adaptive Partial Matching Method

IV. 실험 결과 및 분석

4.1 실험 환경 (Simulation Environment)

이 실험은 CCITT 단색조(gray scale) CIF (352x288)

형식 크기의 시험 영상인 Miss America (Missa), Claire, Salesman 영상 군 각 88장에 대하여 수행되었다. 원래 30Hz의 율을 가진 영상을 실험에서는 10Hz 율로 만들기 위해서 1번째, 4번째, ..., 88번째의 영상만을 사용했다. 복원된 영상의 화질을 비교하기 위해서는 다음과 같은 PSNR(peak signal-to-noise ratio)을 사용하였다.

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_k(m, n) - \tilde{I}_k(m, n))^2,$$

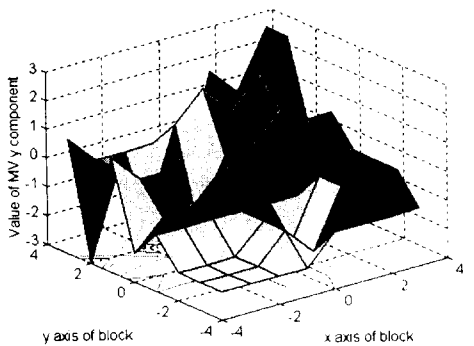
$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ [dB]} \quad (M = 352, N = 288)$$

4.2 거친 움직임 추정과 적응적 부분 정합 방법을 사용한 미세 보정의 결과

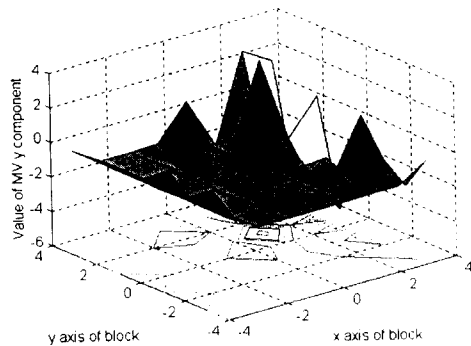
(Coarse Motion Estimation and Results after Refinement using Adaptive Partial Matching Method)

지금까지 다양한 블럭 정합 방법이 제안되었는데 전역 탐색법(FSM)이 최고 성능을 가지며 MPEG, H.263등에 사용된다. 본 논문의 목적은 그것을 대체할 방법의 일환으로 전개되며 성능 비교도 포함한다. 그림 12는 몇 개의 8x8블럭을 뽑아서 화소 단위로 실제 움직임 벡터를 구한 것인데 움직임 자체는 불규칙하게 분포한다. 그런데, 약간은 선형적인 형태를 취할 수 있다. 삼각형 조각을 사용하는 어파인 변환은 선형 변환의 일종이므로 이런 경우는 어파인 변환을 사용한 영상 비틀림 방법이 FSM보다 성능이 뛰어나지만 블럭에 병진 운동의 성분이 더 많은 경우는 오히려 FSM이 나은 성능을 보인다. 그렇지만 이러한 블럭은 일반적으로 많지 않다.

육각형 정합 방법[14]은 반복을 거듭할수록 성능을 향상시키지만 계산량을 고려해서 1번의 수행만을 하는데 이것으로도 충분하다. 이 논문을 통하여 여러 종류의 영상 비틀림 방법의 성능이 나오는데 영상 화질, 계산량, 전송 정보량의 관점에서 충분한 성능을 가져오는 경우가 많지만 최선의 결과는 3가지 관점을 모두 만족하지 못한다. 왜냐하면 전송 채널의 제한, 실시간처리(계산량), 여러 계층 분류 때 적용될 수 있는 영상 화질등과 같은 제약이 다르므로 타협점(trade-off)을 고려해야 한다. 여기서 표기 방법을 보면 거친 움직임 추정에서 어파인 변환을 사용하고 적응적 부분 정합 방법을 적용하여 미세 보정을 한 경우는 AFMC라고 하였고 쌍선형 변환을 사용하고 적응적 부분 정합 방법을 적용한 경우는 BLMC로 쓴다.



(a) x 방향 움직임 요소(x Component of Motion)



(b) y 방향 움직임 요소(y Component of Motion)

그림 12. (17, 21) 위치에 있는 8x8 크기 블록의 실제 움직임 분포(Missa #1, #4)

Fig 12. Original Motion Vector Field in Location (17, 21) Block which has 8x8 Size

그림 13은 여러 전송율에 따른 결과를 보여준다. Miss America 경우는 50%로 충분하며 Claire 경우는 25%만으로도 충분하였다. 여기서 100% 전송율의 경우가 최고의 성능을 가져오지만 50%이상의 전송율을 사용하더라도 충분한 정도의 성능을 가져옴을 알 수 있다. 나중에 또 설명하겠지만 50%정도가 일반적인 경우 적절한 타협점이 되리라고 생각할 수 있다.

지금까지는 주로 영상 화질의 측면에서 성능을 검토하였다. 대부분의 논문들이 여기까지 전개하며 이전의 것보다 월등하다고 하지만 그것들은 그에 상응하는 많은 계산량을 요구하며 추가 정보를 전송하므로 구현이 불가능하다고 생각되는 것도 많다. 그래서 본 논문은 여러 관점에서 모두 비교한다. 계산량과 전송 정보량은 적응적 부분 정합 방법에 의해서 효과적으로 감소된다.

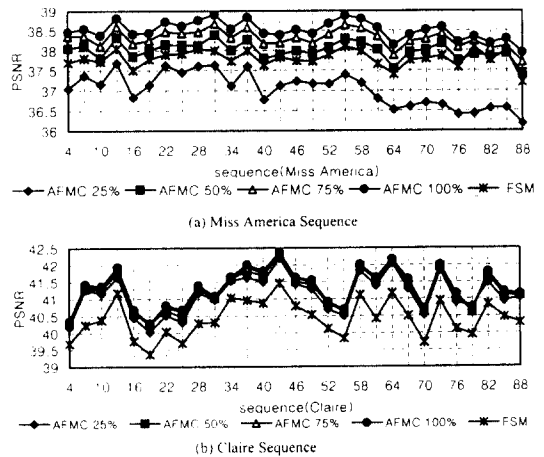


그림 13. 어파인 변환을 사용한 적응적 부분 정합 방법의 성능

Fig 13. Adaptive Partial Matching using Affine Transformation

움직임 벡터의 차에 대하여 가변장 부호화 방법(VLC)을 사용하면 더욱 압축할 수 있지만 본 논문에서는 비교의 관점에서 쉽게 하기 위해 고정 부호화 방법을 사용하였다. (기존의 H.263, MPEG 등의 표준들의 장점을 모두 수용할 수 있다.) 그림 14는 부호화하는 형식을 보여준다. header_bit 정보로부터 이전

화면과 현재 화면간의 움직임을 추정하기 위해 사용된 탐색 영역의 크기를 알 수 있다. 여기서는 이전의 방법과 비교를 위해 탐색 영역을 최대 ± 17 로 제한시켰으므로 2~4 비트 크기 중 하나를 취한다. 만약 이전의 방법들이 최대 탐색 영역 크기를 증가시키면 제한된 방법도 동일하게 증가시키며 전송 정보량의 더 많은 절약을 가져올 수 있다. send_map은 어느 블록이 예측된 것이고 어느 블록이 이전 화면에서 복사될 것인지를 나타내는 위치 정보이다. mv_bits는 부호화되는 움직임 벡터를 나타내며 몇 비트 사용 여부는 header_bit가 결정한다.

여기까지 하고 나서 현재 화면과 이전 화면간의 차이 영상은 DCT와 양자화(Quantization)를 통하여 부호화된다. 이것을 DCTQ라고 표기하며 그림 14와 MPEG의 부호화 방법을 따른다[2].

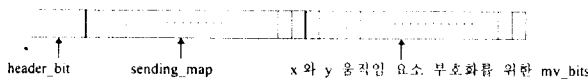


그림 14. 부호화를 위한 bit 할당
Fig 14. Encoding Bit Location

계산량은 다음의 2가지 관점에서 비교된다.

1. 블럭이나 꼭지점의 움직임 벡터를 추정하기 위하여 사용되는 MAD(Mean Absolute Difference)를 들 수 있는데 이것은 탐색 영역의 크기(SR)에 좌우된다.
2. 어파인 변환이나 쌍선형 변환을 사용한 미세 보정 시 요구되는 계산 시간

표 1과 표 2는 FSM, AFMC, BLMC의 결과들을 평균한 것을 정리한 것이다. 표 1을 보면 50% 근방에서 영상 화질과 계산량은 적절하면서 25%와 75%의 전송 정보량 중 DCTQ들과 비교할 때 이 부분이 최소 점임을 볼 수 있다. 여러 다른 경우가 가능하지만 본문에서는 50%정도를 적절한 전송율이라고 본다. Refine_CP는 부분 정합 방법에서 사용되는 꼭지점의 갯수이고 Trans_CP는 전송되는 꼭지점의 갯수이다. Overhead는 header_bit(4비트)와 sending_map(396비트)를 합하여 400비트이다. Opt_SR은 적절한 탐색 영역의 크기인데, 적용적 탐색 영역의 경우 평균을 취하면 실수값이 나오는데 이것보다 큰 인접 정수값을 기록해놓았다. 이것도 비교의 관점에서 처리한 것이므로 실제 적용 시에는 더욱 줄일 수 있다. Total_Trans는 총괄한 전송 정보량이고 Total_Comp는 총괄한 곱셈 연산량이다. 괄호 안의 c는 거친(coarse) 움직임 추정과 FD따위를 구하는데 소요된 것을 의미하며 r은 미세 보정(refinement)을 하는데 소요된 것을 의미한다. DCTQ의 bit는 원 화면과 예측 화면의 차이를 부호화할 때 소요되는데, BLMC(BIL + DCTQ)가 다른 방법들보다 양이 적다. 그렇지만 AFMC(TBM + DCTQ)가 BLMC보다 약간 계산량이 감소한다. 즉 뒤의 첨자 α, β 는 FSM에서 움직임 보상에 걸리는 시간에 대한 상대적으로 어파인 변환이나 쌍선형 변환에 소요되는 시간에 대한 비율인데 보통 1~1.5사이의 값을 가지며 α 가 β 보다 약간 작다. 그러므로 어파인 변환과 쌍선형 변환중 어느 것이 나은지를 결정하기는 힘들다. 이 결정은 영상 군의 특성, 탐색 영역, 정합에 사용된 판정식인 MAD, MSE에 따라 달라진다.

표 1. Miss America에 적용시 평균 성능(AFMC)
Table 1. Average Performance for Affine Transformation

	FSM	AFMC 25%	AFMC 50%	AFMC 75%	AFMC 100%
PSNR(dB)	37.7961	37.0281	37.9972	38.2938	38.5093
DCT/Q size(bytes)	1618	1689	1510	1498	1492
Refine_CP	396	155	269	357	357
Trans_CP	396	99	198	297	357
Overhead(bits)	0	400	400	400	400
Opt_SR	7 (Fixed)	4.31	4.31	4.31	4.31
Total_Trans	16112	14704	14064	14760	15192
Total_Comp(c)		3773538	6474354	8559194	8559198
Total_Comp(r)	22809600	11176704 α	22353408 α	33530112 α	40303872 α

표 2. Miss America에 적용시 평균 성능(BLMC)

Table 2. Average Performance for Bilinear Transformation

	FSM	BLMC 25%	BLMC 50%	BLMC 75%	BLMC 100%
PSNR (dB)	37.7961	37.0660	38.0638	38.3718	38.6019
DCT/Q size(bytes)	1618	1676	1491	1478	1471
Refine_CP	396	155	269	357	357
Trans_CP	396	99	198	297	357
Overhead(bits)	0	400	400	400	400
Opt_SR	7 (Fixed)	4.31	4.31	4.31	4.31
Total Trans	16112	14600	13912	14600	15024
Total_Comp(c)		3773538	6474354	8559194	8559198
Total_Comp(r)	22809600	19869696β	39739392β	59609088β	71651328β

표 3. 여러가지 영상 비틀림 방법에 따른 성능의 평균값

Table 3. Average Performance for Several Image Warping

Methods

	Hexa7	Gradient	Forward	HGI	Proposed
PSNR (dB)	38.3894	37.0660	38.3729	37.8021	38.6019
DCT/Q size(bytes)	1491	1676	1505	1723	1471
Refine_CP	396	396	396	515	357
Trans_CP	396	396	396	515	357
Overhead(bits)	0	0	0	396	400
Opt_SR	7	.	7	7	4.31
Total Trans(bits)	15096	16576	15208	17904	15024
Total_Comp(c)	22809600	792x792 역행렬	22910976	22995879	8559198
Total_Comp(r)	71651328β	71651328β	71752704β	97491968β	71651328β

Hexa7 : 육각형 정합방법 (탐색영역 ±7) [14]

Gradient : Gradient Constraint 방법[14]

Forward : Forward Matching 방법[15]

HGI : Hierarchical Grid Interpolation 방법[16]

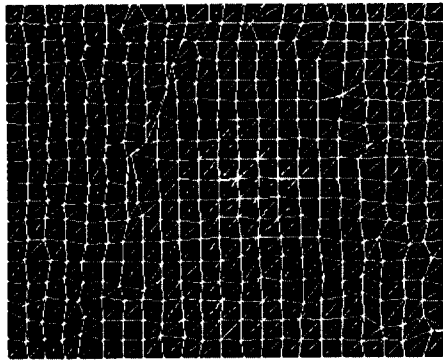
Refine_CP는 미세 보정에 사용된 꼭지점의 갯수를 보여주는데 이것이 실제적인 계산량의 대부분을 결정한다. 사각형 블럭의 FD분포를 이용하였으므로 BLMC와 AFMC의 Refine_CP의 갯수가 동일하게 된 것이다. 만약 AFMC에 삼각형 블럭의 FD 분포를 적용하면 갯수가 달라진다.

표 3에서 보듯이 기존의 영상 비틀림 방법과 비교할 때, 제안한 방법은 영상 화질, 전송 정보량, 계산량의 관점에서 모두 적절한 해답이 될 수 있음을 알 수 있다. 각 방법에서 계산량에 대한 부분은 서로 다른 연산을 가지므로 정확히 측정하기는 어렵다. 즉 Hexa7, Forward, HGI방법은 블럭의 MSE를 계산하는데 반해서 Gradient 방법은 역행렬의 계산이 필요하므로 비교의 관점이 정확히 일치하지는 않지만 곱셈의 양에 비중을 많이 두고 정리하였다. 제안한 방법이 다른 방법과 비교하여 우수하게 된 이유는 다음과 같다.

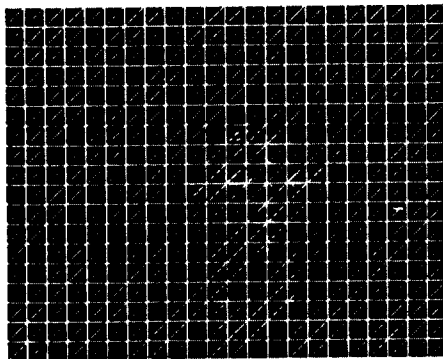
기존 방법과 달리 적용적으로 가변하는 탐색영역을 적용함과 배경같은 움직임이 적은 영역은 적은 값의 잡음이 섞이더라도 움직임 예측에 영향을 받기 쉬우므로 영상간의 차이가 미소한 부분을 이전 영상의 해당되는 값으로 적절히 대체함, 그리고 우선 순위에 따른 정보 전송에 의하여 격자 겹침에 의한 성능 손실을 줄이고 계산량도 감소시킬 수 있었다.

본 연구에서는 적응적 부분 정합 방법을 적용하였을 때 적절한 타협점을 잡을 수 있었다. 그리고 응용적인 측면에서 다양한 확장이 가능하다. 작은 채널 용량을 사용하거나 실시간적인 응용이 요구된다면 거친 움직임 추정과 움직임 보상만으로도 충분한 해답이 될 수도 있다. 만약 좀 더 정밀함을 요구한다면 미세 보정을 추가하면 된다. 선형 근사화를 적용한 영상 비틀림 방법은 블럭 정합 방법보다 블럭간의 불연속성과 블럭 효과가 적기 때문에 눈에 거슬림이 적다. 그림 15에서 100% 전송율일 때 꼭지점의 연결 구

조를 보여주는데 현재 화면은 규칙적이고 이전 화면은 불규칙적으로 변형되었다. 또한 배경 부분에는 잡음에 의한 큰 변형이 일어나서 어떤 부분은 겹쳐 넘어가고 있다. 그림 7에서 이 문제에 관하여 언급했었다. 이러한 문제들이 표 3에 있는 다른 방법들에서도 발생하는데 이러한 변형은 영상 화질의 저하를 가져온다. 그림 16은 25%와 50%의 꼭지점만을 사용한 영상 비틀림 방법에서의 격자 구조인데 상대적으로 변형이 적다. 또한 부분 정합 방법이 보다 효과적으로 움직이는 물체를 추적할 수 있음을 알 수 있다.



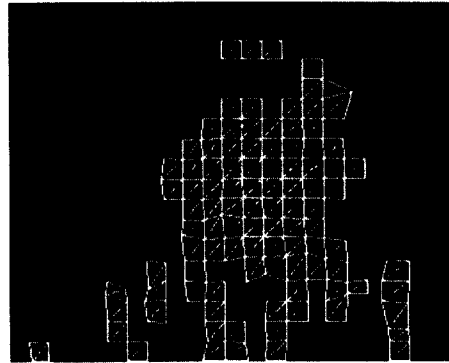
(a) 이전 화면의 격자 연결 구조(# 76)
Grid Topology in Previous Frame



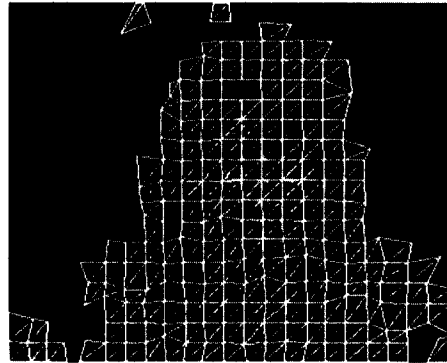
(b) 현재 화면의 격자 연결 구조(#79)
Grid Topology in Current Frame

그림 15. 100% 꼭지점을 사용한 영상 비틀림 후의 격자 연결 구조(Miss America)

Fig 15. Grid Topologies after Image Warping using 100% Grid Points (Miss America)



(a) 이전 화면의 격자 연결 구조(25%)
Grid Topology for 25% Grid Points



(b) 이전 화면의 격자 연결 구조(50%)
Grid Topology for 50% Grid Points

그림 16. 적용적 부분 정합 방법을 사용한 영상 비틀림 후의 격자 연결 구조(Miss America)

Fig 16. Grid Topologies after Proposed Adaptive Partial Matching (Miss America)

V. 결 론

본 논문에서는 통계적으로 조사된 어파인 변환과 쌍선형 변환에 기반한 새로운 움직임 추정 및 보상 방법을 제시하였고, 벡터 관계를 이용한 고속 어파인 변환을 유도하였으며, 다른 고속 변환을 유도할 수 있는 여지를 남겨두었다. 예측 영상은 이전 화면과 현재 화면간에 선택된 화소를 기하학적으로 변환에 의해서 구성되는데, 본 논문에서 주로 제안한 것은 적용적 부분 정합 방법이다. 제안한 알고리즘은 계산량의 측면에서 기존의 육각형 정합 방법보다 향상되

었으며 영상 화질은 비교될 정도이다. 그리고 전송 정보량도 줄었으며 전송율을 조절하여 여러 가지로 응용할 수 있다.

제안한 알고리즘에 DCTQ를 결합시킨 부호화기를 시험해 본 결과, 제안한 방법은 초저속 영상 통신에 효과적임을 알 수 있었다. 이러한 성능은 비록 꼭지점의 갯수를 50% 이하로 줄인다 하더라도 어떠한 변환과 쌍선형 변환은 받아들일 수 있을 정도의 결과를 나타냄에 기인한다. 비록 객관적인 평가대상인 PSNR이 약간의 차이 밖에 없지만 주관적인 평가 대상인 복원 영상은 다른 방법들보다 훼손의 정도가 훨씬 적다. 그 이유는 격자 겹침에 의한 구조 파괴를 최소화시켰기 때문이다.

참 고 문 헌

1. M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*, SPIE Press, Volume TT 7, 1991.
2. ISO/IEC 13818-2: "Information Technology-Generic Coding of Moving Pictures and Associated Audio: Video," International Standard, March 1996.
3. J. R. Jain and A. K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1806, Dec. 1981.
4. R. Srinivasan and K. R. Rao, "Predictive Coding Based on Efficient Motion Estimation," *IEEE Trans. Commun.*, vol. COM-33, no. 8, pp. 888-896, Aug. 1985.
5. S. Kappagantula and K. R. Rao, "Motion Compensated Interframe Image Prediction," *IEEE Trans. Commun.* vol. COM-33, no. 9, pp. 1011-1014, Sep. 1985.
6. H. G. Musmann, P. Pirsch and H. Grallert, "Advances in Picture Coding," *IEEE Proceeding*, vol. 73, no. 4, pp. 523-548, Apr. 1985.
7. J. Wang and D. Zhang, "Improved Motion Estimation Algorithm of Television Signals," *HDTV'95 Proceeding*, pp. 8A33-8A40, Nov. 1995.
8. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, London; Academic Press, 1990.
9. F. Bergholm and S. Carlsson, "A Theory of Optical Flow," *Computer Vision, Graphics, and Image Processing*, vol. 53, no. 2, pp. 171-188, Mar. 1991.
10. T. S. Denney Jr. and J. Prince, "Optimal Brightness Functions for Optical Flow Estimation of Deformable Motion," *IEEE Trans. Image Processing*, vol. 3, no. 2, pp. 178-191, Mar. 1994.
11. A. Watt, *3D Computer Graphics*, Addison Wesley, 1993.
12. D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, 1986.
13. D. Hearn and M. P. Baker, *Computer Graphics*, Prentice-Hall, 1986.
14. Y. Nakaya and H. Harashima, "Motion Compensation Based on Spatial Transformations," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, no. 3, pp. 339-356, June 1994.
15. J. Nieweglowski, T. G. Gampbell and P. Haavisto, "A Novel Video Coding Scheme Based On Temporal Prediction Using Digital Image Warping," *IEEE Trans. Consumer Electronics*, vol. 39, no. 3, pp. 141-150, Aug. 1993.
16. C. L. Huang and C. Y. Hsu, "A New Motion Compensation Method," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 4, no. 1, pp. 42-52, Feb. 1994.
17. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press Monograph, 1990.
18. K. R. Rao and J. J. Hwang, *Techniques and Standards For Image Video and Audio Coding*, Prentice-Hall, 1996.
19. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.
20. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.

