

論文98-35S-1-15

동영상 압축을 위한 모델 기반 부화소 단위 움직임 추정 기법

(Model-Based Subpixel Motion Estimation for Image Sequence Compression)

徐廷旭*, 鄭濟昌*

(Jeong Wook Suh and Jechang Jeong)

요 약

본 논문에서는 움직임 보상 에러 신호의 수학적 모델에 기반하여 정수 단위의 움직임 벡터로부터 더이상의 화소간의 보간없이 부화소 단위의 움직임 벡터를 구하는 방법을 제안한다. 먼저 정화소단위로 움직임 벡터를 전역탐색 방법에 의하여 추정하고 전역탐색 과정에서 함께 구해진 주변 화소의 에러 차분의 합들을 이용하여 움직임 추정에러의 수학적 모델의 계수값들을 결정한다. 계수값이 결정된 모델의 방정식으로부터 부화소의 에러 차분의 합의 근사값을 추정하도록 하였다. 본 논문에서는 단계적인 수학적 모델로서 모델 1, 모델 2, 모델 3과 수정모델 2, 수정모델 3, 부분 보간 모델 3을 제안한다. 모델 1에서는 3×3 화소들로 구성된 평면을 미지계수가 9개인 2차 방정식으로 모델링한 것으로부터 부화소 단위의 움직임 벡터를 구한다. 모델 2와 모델 3에서는 모델 1을 단순화하여 미지계수가 각각 6, 5개인 2차 방정식으로 모델링한 것으로부터 부화소 단위의 움직임 벡터를 구한다. 수정모델 2와 수정모델 3은 일부 화소에 가중치를 주어 성능을 향상시킨 모델이다. 부분 보간 모델 3은 모델 3에 의해 구해진 부화소의 주위만을 부분적으로 보간하여 정확성을 더욱 향상시킨 방법이다. 실험 결과는 제안한 모델들이 더 정밀할수록 좋은 성능을 보여주고 있으며 단순화한 모델들을 수정한 모델들은 단순하면서도 성능이 우수함을 보여준다.

Abstract

This paper presents a method to estimate subpixel accuracy motion vectors using a mathematical model without interpolation. The proposed method decides the coefficients of mathematical model, which represents the motion estimation error, using motion-compensated prediction errors around the pixel accuracy motion vector which is achieved by full search. And then the proposed method estimates subpixel accuracy motion vector from achieved mathematical model. Step by step mathematical models such as type 1, type 2, type 3, modified type 2, modified type 3, and Partial Interpolation type 3 are presented. In type 1, quadratic polynomial, which has 9 unknown coefficients and models the 3 by 3 pixel plane, is used to get the subpixel accuracy motion vectors by inverse matrix solution. In type 2 and 3, each quadratic polynomial which is simplified from type 1 has 5 and 6 unknown coefficients and is used by least square solution.

Modified type 2 and modified type 3 are enhanced models by weighting only 5 pixels out of 9. P.I. type 3 is more accurate method by partial interpolation around subpixel which is achieved by type 3. These simulation results show that the more delicate model has the better performance and modified models which are simplified have excellent performance with reduced computational complexity.

* 正會員, 漢陽大學校 電子通信工學科

(Dept. of Elec. Comm. Eng., Hanyang Univ.)

接受日字: 1997年5月13日, 수정완료일: 1997年11月19日

지틀 영상 데이터의 압축은 크게 3가지 방법을 이용하고 있다. 시간적인 중복성(temporal redundancy)을 줄이는 방법과 공간적인 중복성(spatial redundancy)을 줄이는 방법 그리고 발생부호의 통계적 특성을 이용하여 줄이는 방법이 주로 이용된다. 따라서 이러한 방법들은 MPEG-1이나 MPEG-2, HDTV와 같은 국제표준에서 모두 채택되어 사용되고 있으며 계속 연구되고 있다. 공간적인 중복성을 줄이는 방법으로는 이산여현변환(Discrete Cosine Transform)이 가장 많이 사용되고 발생한 부호의 통계적 특성을 이용하는 방법은 허프만 부호화나 산술(Arithmetic)부호화가 많이 사용된다. 시간적인 중복성을 줄이는 대표적인 방법이 바로 움직임 추정 및 보상 방법이다. 움직임 추정방법은 크게 화소단위(pel-by-pel basis)의 추정방법과 블록단위(block-by-block basis)의 추정방법이 있는데 블록단위의 움직임 추정방법이 더 많이 사용되고 있다. 바로 이전 화면과 가장 유사한 부분을 찾고 그 유사한 부분을 나타내는 벡터와 그 부분과의 차이 성분만을 전송하는 움직임 추정 및 보상방법은 가능한 정밀하게 찾을수록 전송할 차이성분이 줄어들어 데이터를 더욱 효과적으로 줄일 수 있지만 이전 화면에서 가장 유사한 부분을 찾기 위해서는 상당한 추정시간과 계산량이 요구된다. 따라서 동영상의 부호화시간이 가장 많이 소요되는 움직임 추정시간을 줄이는 노력이 계속되고 있다. 먼저 정화소(integer pixel) 단위로 움직임 벡터를 찾는 방법으로 2-D 대수적 탐색법(Logarithmic Search)^[1], 3단계 탐색법(Three Step Search)^[2] 등이 일찍부터 제안되었지만 국부 최소값(local minimum)에 빠질 수 있는 가능성과 부화소(sub pixel) 단위까까지 더 세밀하게 움직임 벡터를 찾아야 할 필요성, 규칙적인 형태의 병렬처리를 사용하여 효율적인 하드웨어의 구현이 가능하다는 점 때문에 MPEG-2 등에서는 보통 전역탐색법(Full Search)을 사용하고 있다. 정화소와 정화소 사이를 보간에 의해 더욱 정밀하게 추정하는 부화소 단위의 움직임 추정 방법은 현재 MPEG-2 등에서 채택되어 사용되고 있지만 보간에 의한 계산량과 부호화시간의 증가는 개선되어야 할 여지가 많은 부분이다. 부화소 단위로 움직임 벡터를 찾아서 정밀도를 높이면서도 소요시간과 계산량을 줄이는 방법들^[3-6]이 제안되고 있지만 기존의 방법들이 에러 모델을 지나치게 단순화하여 계산량은 획기적으로 줄었지만 부호화 효율의 손실이

커지는 단점이 있다.

이에 본 논문에서는 단계적인 수학적 모델을 제시하여 부화소단위의 움직임 벡터를 좀더 정확히 찾을 뿐만 아니라 보다 넓은 응용력을 갖도록 하였다. 본 연구에서 제안된 모든 움직임 추정 에러의 모델에서, 먼저 정화소단위로 움직임 벡터를 전역 탐색법에 의하여 추정하고 그때 구해진 움직임 벡터와 전역 탐색과정에서 함께 구해진 주변 화소의 에러 차분의 합들을 이용해 모델의 계수값들을 결정한 후 부화소의 에러 차분의 합의 근사값을 추정하도록 하였다. 단계적인 수학적 모델로서 모델 1, 모델 2, 모델 3과 수정 모델 2, 수정 모델 3, 부분 보간(partial interpolation) 모델 3이라 이름 붙이고 이러한 순서로 기술해 나가도록 한다. 모델 1은 역행렬풀이(inverse matrix solution) 방법을 이용하여 부화소값을 추정하고 모델 2는 최소자승법(least square solution)을 이용하여 부화소값을 추정하며 모델 3은 가장 간단화된 알고리즘을 사용한다. 수정 모델 2와 수정 모델 3은 일부 화소에 가중치를 주어 성능을 향상시킨 모델이고 부분 보간 모델 3은 모델 3에 의해 구해진 부화소의 주위만을 부분적으로 보간하여 정확성을 더욱 향상시킨 방법이다.

본 논문의 구성은 다음과 같다. II장에서는 표준안에서의 움직임 추정 방법을 간략히 살펴보고 부화소 단위의 움직임 추정 방법에 있어서의 계산량과 이에 따른 문제점을 살펴본 후 기존의 부화소 단위의 움직임 추정 방법에 대해 소개한다. III장에서는 제안하는 모델기반 부화소단위 움직임 추정방법으로 모델 1, 모델 2, 모델 3, 수정모델 2, 수정모델 3, 부분 보간 모델 3에 대해서 기술하고 이에 따른 부화소 단위 움직임 벡터의 추정방법을 제시한다. IV장에서는 각 모델에 따른 실험결과를 기술하고 V장에서는 결론을 맺는다.

II. 표준안과 기존의 부화소 단위 움직임 추정 방법

먼저 현재 동영상의 국제표준으로서 많이 이용되고 있는 H.261, MPEG-1, MPEG-2, H.263의 경우를 살펴보기로 한다. H.261은 정화소 단위의 움직임 추정을 하고 있으며 MPEG-1에서는 반화소단위의 움직임 추정이 선택사항으로 되어있고 MPEG-2부터는 반화소단위의 움직임 추정이 필수적으로 사용되고 있다. 반화소단위 움직임 보상은 움직임 추정의 정밀도를 높

이는 역할뿐만 아니라 H.261에 쓰이는 루프 필터 대신 영상을 부드럽게 평활화하는 공간적 필터의 역할도 한다. H.261에서는 과거의 재생 영상을 저장하는 프레임 메모리를 한 장만 사용하여 순방향 예측만을 행한다. 이에 비해 MPEG-1에서는 프레임 메모리를 두 장 사용하여 과거 재생 영상으로부터의 순방향 예측과 미래 재생 영상으로부터의 역방향 예측을 함께 사용하는 구조, 즉 쌍방향 예측을 수행한다. MPEG-2에서의 예측 방법으로서 순방향 예측, 역방향 예측, 쌍방향 예측이 있는 것은 MPEG-1과 같지만 이들 예측 방법은 다시 프레임 예측과 필드 예측으로 나눌 수 있다.

H.263의 복호기에서는 매크로블록당 하나의 움직임 벡터를 사용하거나 고급 예측 모드(Advanced Prediction mode)에서는 네개의 벡터를 사용한다. 움직임 벡터의 수평성분이나 수직성분은 $[-16, 15.5]$ 사이의 반화소 단위의 값을 갖는다. 일반의 H.263 예측 모드에서는 움직임 벡터가 부호화된 화상 내의 화소까지로 제한되지만 비제한 움직임 벡터 모드(Unrestricted Motion Vector mode)에서는 움직임 벡터가 화상 밖까지 가리킬 수 있다. 움직임 벡터에 의해서 참조되어지는 화소가 부호화된 화상의 밖에 있다면 화상의 모서리값이 대신 사용된다. 고급 예측 모드(Advanced Prediction mode)에서는 중첩(Overlapped)블록 움직임 보상을 하고 한 매크로블록에 4개의 움직임 벡터를 사용하는 것이 가능하다. 동영상의 국제표준들이 갈수록 성능이 우수하고 다양한 방법들을 사용함에 따라 이에 따른 부호화 방법과 과정들이 복잡해지고 계산량이 많아지게 된다. 그러므로 부호화 과정의 각 단계나 요구되는 방법들을 구현함에 있어 부분적으로나 혹은 전체적으로 간단히 하거나 빠른 속도로 구현하는 것이 더욱 필요하게 되었다. MPEG에서부터 사용되고 있는 반화소 단위의 움직임 추정의 경우 16×16 화소의 매크로 블록 단위 휘도성분으로 움직임을 추정, 보상하는데 먼저 인접한 매크로 블록에서 전역탐색법에 의해 찾아 최적의 정화소 단위의 움직임 벡터를 찾는다. 찾은 정화소 단위의 움직임 벡터 근처에서 다시 반화소 단위로 움직임 벡터를 추정하게 되는데 이때 찾은 최적의 반화소 단위 움직임 벡터의 후보가 되는 반화소들은 모두 정화소들간의 보간(반올림 평균)에 의해 구해지게 된다. 따라서 16×16 화소단위의 매크로 블록에서는 8번의 16×16 의 반화소들을 보간에 의해 계산해야 하므로 정화소

단위의 움직임 벡터 추정외에도 많은 계산량과 시간이 추가로 요구된다. 이렇게 많은 계산량과 시간을 줄이고자 많은 방법들이 제안되었다. Y. Senda와 H. Harasaki^[46]는 반화소 단위의 움직임 벡터를 찾는 데 있어서 보간을 이용하지 않고 주변의 정수 단위의 에러차분을 이용하였다. 반화소 단위의 에러차분을 구하기 위해 L2 Norm과 L1 Norm의 관계를 이용하여 MSE(Minimum Square Error)를 MAE(Minimum Absolute Error)로 근사화하고 그 후에 가정에 근거하여 구현에 용이하도록 간략화함으로써 주변의 정수 단위 에러차분으로 반화소 단위의 움직임 벡터를 구하는 것을 제안하였다. K. M. Nam et al^[13]은 평균 피라미드(Mean Pyramid)를 이용하여 계층적으로 움직임을 추정하는 방법을 제안하였다. 정화소 단위의 움직임 벡터를 먼저 찾기 위해 이미지의 화소를 서브샘플링(subsampling)하고 평균하여 피라미드 이미지를 몇 단계로 만든 후 최상위 단계에서 움직임 벡터를 찾고 점점 단계를 낮추어감에 따라 정밀하게 움직임 벡터를 찾는 방법이다. 성능개선을 위해 각 단계에서의 후보벡터의 개수를 몇 개 더 늘여 각 하위 단계마다 독자적으로 움직임 벡터를 찾는 방법도 함께 제시되었으며 그렇게 찾아진 정화소 단위의 움직임 벡터 주변을 기존의 보간을 이용하여 반화소 단위의 움직임을 추정하였다. 따라서 이 방법은 정화소 단위로 움직임을 찾는 단계만을 간단히 하는 방법에 속하며 부화소 단위의 움직임은 기존의 보간을 이용하였기 때문에 부화소 단위의 움직임 추정시 요구되는 많은 계산량을 줄이지는 못했다고 할 수 있다.

III. 제안하는 모델 기반 부화소 단위 움직임 추정 방법

1. 모델 1(Generalized Inverse matrix solution)

그림 1에서 ○표시된 곳은 정화소의 위치들이고 ×표시된 곳은 반화소의 위치들을 나타내고 있다. 이중 (0,0)좌표는 이미 전역탐색법에 의해서 찾아진 정수 단위의 움직임 벡터이다. 따라서 반화소 단위의 움직임 벡터는 (0,0)좌표 자신일 수도 있고, (0,0) 좌표 주변의 8개의 반화소의 좌표 중 하나일 수 있다.

3×3 화소들로 구성된 평면은 식(1)과 같은 2차 방정식으로 모델링될 수 있으며 9개의 미지계수로 구성된다.

$$f(x, y) = c_1x^2y^2 + c_2x^2y + c_3x^2 + c_4xy^2 + c_5xy + c_6x + c_7y^2 + c_8y + c_9 \quad (1)$$

이 방정식은 x나 y 어느 한 값을 고정하면 그 지점의 2차 방정식이 된다. 고정된 x나 y가 변할 때마다 그래프의 기울기가 변하는 2차 곡선이 될 것이다. 전역탐색 과정에서 이미 찾은 움직임 벡터인 (0,0)에서의 에러 차분의 합과 주변 8개의 정화소에서의 에러 차분의 합이 모두 9개이므로 9개의 계수 값을 구할 수 있다. 왼쪽 위(좌표(-1,-1))부터 오른쪽 아래(좌표(1,1))까지 차례로 방정식에 대입했을 때 행렬 형식으로 나타내면 식(2)와 같이 나타낼 수 있다.

$$\begin{bmatrix} f(-1, -1) \\ f(0, -1) \\ f(1, -1) \\ f(-1, 0) \\ f(0, 0) \\ f(1, 0) \\ f(-1, 1) \\ f(0, 1) \\ f(1, 1) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{bmatrix} \quad (2)$$

여기서 f()은 에러차분의 합 값을 나타낸다. 이 방정식을 풀기 위해서는 식(2)의 9×9 행렬의 역행렬을 구해야 하는데 이 행렬은 특이하지 않으므로(Non-singular) 식(3)과 같이 역행렬이 구해진다. 매크로블록(MB)에서 언제나 그림 1과 같이 정수 단위의 움직임 벡터가 (0,0)으로 설정되고 주변은 그 상대적인 값(-1,0,1)으로 설정되기 때문에 매번 역행렬을 구할 필요 없이 식(3)과 같은 고정된 값을 사용할 수가 있어서 9×9 행렬의 역행렬로 인한 계산량의 증가는 전혀 고려할 필요가 없게 된다. 게다가 식(3)의 9×9 행렬은 성긴 행렬(sparse matrix)이고 규칙적이어서 계산량에 있어서 유리하게 작용할 수 있다.

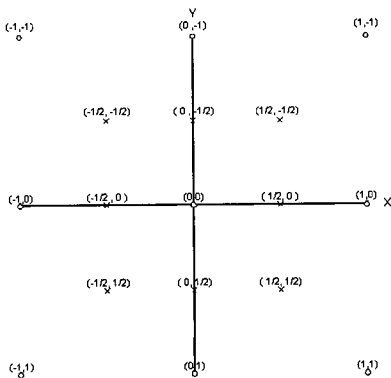


그림 1. 정화소과 반화소의 상대적 좌표
Fig. 1. Relative coordinates of pixels and halfpixels.

식(3)에 의해 구해진 9개의 계수 $c_1 \sim c_9$ 을 가지고 식(1)을 완성하여 이제는 그림 1의 반화소의 각 좌표들을 대입하여 그 좌표에서의 에러 차분의 합의 값을 추정한다. 식(4)에서 보여진 것처럼 이때도 각 반화소들에 의해 구성되는 9×9 행렬은 고정적이고 성긴 행렬이며 어느 정도 규칙성이 있음을 알 수 있다.

$$\begin{bmatrix} 1/4 & -1/2 & 1/4 & -1/2 & 1 & -1/2 & 1/4 & -1/2 & -1/4 \\ -1/4 & 1/2 & -1/4 & 0 & 0 & 1/2 & 0 & 1/4 & -1/2 & 1/4 \\ 0 & 0 & 0 & 1/2 & -1 & 1/2 & 0 & 0 & 0 & 0 \\ -1/4 & 0 & 1/4 & 1/2 & 0 & -1/2 & -1/4 & 0 & 1/4 & 0 \\ 1/4 & 0 & -1/4 & 0 & 0 & 0 & -1/4 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & -1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & -1 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & -1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f(-1, -1) \\ f(0, -1) \\ f(1, -1) \\ f(-1, 0) \\ f(0, 0) \\ f(1, 0) \\ f(-1, 1) \\ f(0, 1) \\ f(1, 1) \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{bmatrix} \quad (3)$$

$$f(x, y) = c_1x^2y^2 + c_2x^2y + c_3x^2 + c_4xy^2 + c_5xy + c_6x + c_7y^2 + c_8y + c_9$$

$$\begin{pmatrix} (-0.5, -0.5) \\ (0, -0.5) \\ (0.5, -0.5) \\ (-0.5, 0) \\ (0, 0) \\ (0.5, 0) \\ (-0.5, 0.5) \\ (0, 0.5) \\ (0.5, 0.5) \end{pmatrix} \begin{pmatrix} 0.0625 & -0.125 & 0.25 & -0.125 & 0.25 & -0.5 & 0.25 & -0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & -0.5 & 1 \\ 0.0625 & -0.125 & 0.25 & 0.125 & -0.25 & 0.5 & 0.25 & -0.5 & 1 \\ 0 & 0 & 0.25 & 0 & 0 & -0.5 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.25 & 0 & 0 & 0.5 & 0 & 0 & 1 \\ 0.0625 & 0.125 & 0.25 & -0.125 & -0.25 & -0.5 & 0.25 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.5 & 1 \\ 0.0625 & 0.125 & 0.25 & 0.125 & 0.25 & 0.5 & 0.25 & 0.5 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{pmatrix} \quad (4)$$

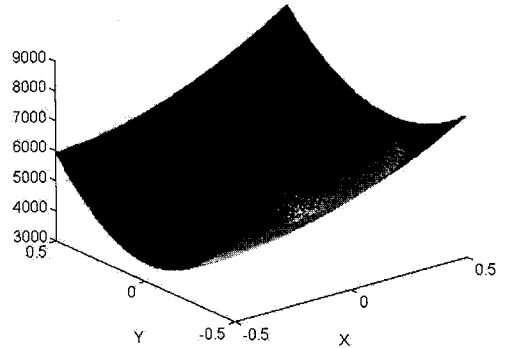


그림 2. 임의의 매크로블록에서 구해진 9개의 계수를 가진 에러 방정식의 3차원적 표현
Fig. 2. 3-Dimensional display of error polynomial which has 9 coefficients in arbitrary MB.

그림 2는 임의의 매크로블록에 대해 이렇게 해서 구해진 식을 3차원적으로 나타낸 그림이고 그림 3은 실제 9점만을 표시한 것이며 그때의 고정된 x와 y에서의 2차원 그래프를 그림 4에 나타내었다. 그림 2와 그림 4에서의 x와 y는 각각 화소의 수평방향의 성분과 수직방향의 상대적인 좌표를 의미한다. 즉, 좌표(0, 0)은 이미 전역탐색법에 의해서 찾은 정수 단위의 움직임 벡터를 의미한다. 예를들어 임의의 매크로블록에서 반화소단위의 에러차분의 합을 나타낸 그림 2, 3, 4에서 (-0.5, 0)일 때 에러차분의 합이 가장 작은 것으로 나

타나 있는데 이때 정수 단위의 움직임 벡터가 (-3, 4) 였다면 본 방법에 의해 구해진 반화소 단위의 움직임 벡터는 (-3.5, 4)가 되는 것이다.

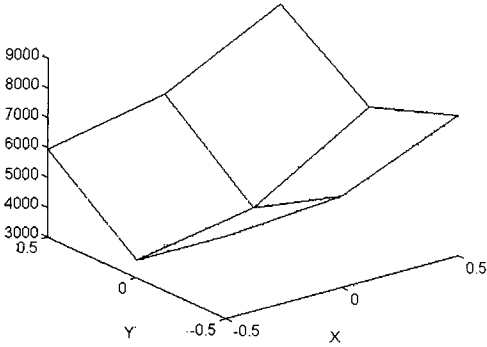


그림 3. 임의의 매크로블록에서 구해진 에러 방정식에 실제 9개의 점에서의 에러값
Fig. 3. Error values of real 9 points from error polynomial in arbitrary MB.

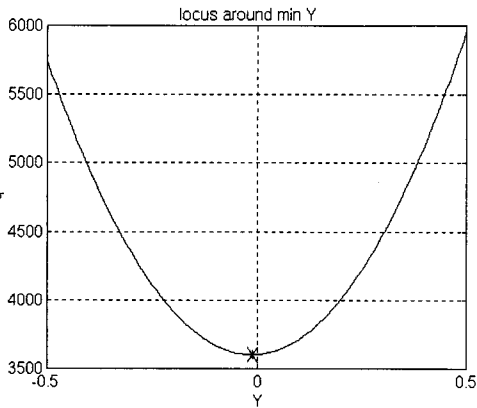
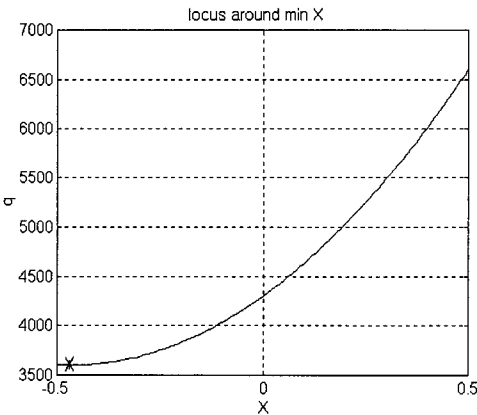


그림 4. 그림2에서 구해진 그래프를 고정된 x와 y에서 본 2차원 그래프
Fig. 4. 2-Dimensional graph in fixed x and y from Fig 2.

2. 모델 2 (least square solution)

모델 1에서는 9개의 미지수를 가진 방정식을 풀기 위해 9좌표의 에러 차분의 합을 이용하였는데 좀 더 간단화된 모델로서 식(1)에서 x^2y^2 , x^2y , xy^2 항을 없앤 모델을 다음과 같이 제시한다.

$$f(x, y) = c_1x^2 + c_2xy + c_3y^2 + c_4x + c_5y + c_6 \quad (5)$$

식(1)에 의한 식이 그림 5와 같이 3차원적 에러 차분의 합을 등고선처럼 표시할 때, x, y축에 대하여 대각방향으로 기울어진 타원형까지 나타낼 수 있다면 식(5)는 그림 6과 같이 x축 혹은 y축에 수평한 방향으로 찌그러진 형태의 타원만으로 표시되는, 좀 더 간단화된 식이라고 할 수 있다. 이 식은 미지수가 6개이므로 필요한 좌표의 수는 6개이지만 주어진 9개의 좌표 중 어느 것을 선택해야 더 정확한 결과를 낼지는 알 수가 없다. 따라서 미지수가 6개이고 주어진 값이 9개 일 때(식(6)) 사용하는 최소자승법(least square solution)(식(7))을 사용하게 된다. 여기서 행렬 A는 그림 1에서 왼쪽 위인 좌표(-1, -1)에서부터 오른쪽 아래인 좌표(1, 1)까지의 정화소 좌표들을 식(5)에 대입했을 때의 각 항들의 값들을 나타내는 9×6행렬이고, c는 식(5)의 계수들을 나타내는 6×1행렬이며, f는 전역탐색과정에서 이미 찾은 좌표(-1, -1)에서 (1, 1)까지의 각각의 에러차분의 합을 나타내는 9×1행렬이다. 식(7)과 같은 최소자승법을 사용했을 때의 의사역행렬(pseudo inverse)된 결과를 식(8)에 나타내었다.

$$\begin{matrix} f(-1, -1) \\ f(0, -1) \\ f(1, -1) \\ f(-1, 0) \\ f(0, 0) \\ f(1, 0) \\ f(-1, 1) \\ f(0, 1) \\ f(1, 1) \end{matrix} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} \quad (6)$$

$$A \cdot c = f \quad (7)$$

$$c = (A^T A)^{-1} A^T \cdot f$$

이렇게 해서 구해진 계수 값을 가지고 모델 1에서 했던 것과 동일하게 식(5)를 완성하고 식(9)와 같이 반화소의 각 좌표들을 대입하여 그 좌표에서의 에러 차분의 합의 값을 추정한다.

$$\begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{matrix} = \begin{bmatrix} 1/6 & -1/3 & 1/6 & 1/6 & -1/3 & 1/6 & 1/6 & -1/3 & 1/6 \\ 1/4 & 0 & -1/4 & 0 & 0 & 0 & -1/4 & 0 & 1/4 \\ 1/6 & 1/6 & 1/6 & -1/3 & -1/3 & -1/3 & 1/6 & 1/6 & 1/6 \\ -1/6 & 0 & 1/6 & -1/6 & 0 & 1/6 & -1/6 & 0 & 1/6 \\ -1/6 & -1/6 & -1/6 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 \\ -1/9 & 2/9 & -1/9 & 2/9 & 5/9 & 2/9 & -1/9 & 2/9 & 1/9 \end{bmatrix} \begin{matrix} f(-1, -1) \\ f(0, -1) \\ f(1, -1) \\ f(-1, 0) \\ f(0, 0) \\ f(1, 0) \\ f(-1, 1) \\ f(0, 1) \\ f(1, 1) \end{matrix} \quad (8)$$

$$f(x, y) = c_1x^2 + c_2xy + c_3y^2 + c_4x + c_5y + c_6$$

$$\begin{matrix} (-1/2, -1/2) \\ (0, -1/2) \\ (1/2, -1/2) \\ (-1/2, 0) \\ (0, 0) \\ (1/2, 0) \\ (-1/2, 1/2) \\ (0, 1/2) \\ (1/2, 1/2) \end{matrix} \begin{bmatrix} 1/4 & 1/4 & 1/4 & -1/2 & -1/2 & 1 \\ 0 & 0 & 1/4 & 0 & -1/2 & 1 \\ 1/4 & -1/4 & 1/4 & 1/2 & -1/2 & 1 \\ 1/4 & 0 & 0 & -1/2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1/4 & 0 & 0 & 1/2 & 0 & 1 \\ 1/4 & -1/4 & 1/4 & -1/2 & 1/2 & 1 \\ 0 & 0 & 1/4 & 0 & 1/2 & 1 \\ 1/4 & 1/4 & 1/4 & 1/2 & 1/2 & 1 \end{bmatrix} \quad (9)$$

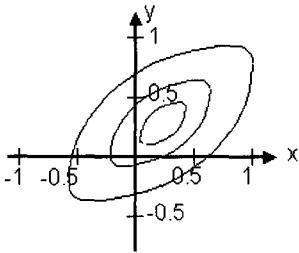


그림 5. 모델 1의 에러값을 평면적으로 나타낸 그림
Fig. 5. Error values of type 1 expressed by contour line.

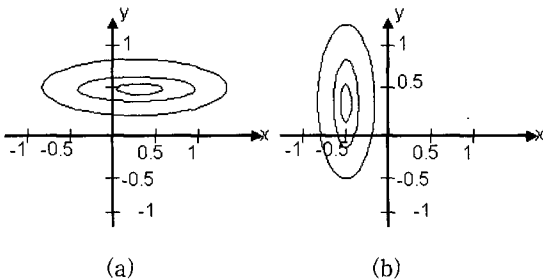


그림 6. (a),(b) 모델 2의 에러값을 평면적으로 나타낸 그림
Fig. 6. (a),(b) Error values of type 2 expressed by contour line.

3. 모델 3 (Simplified fast algorithm)

이 모델은 모델 2보다도 더 간단히 xy항을 없애 버린 것으로 식(10)과 같다.

$$f(x, y) = c_1x^2 + c_2x + c_3y^2 + c_4y + c_5 \quad (10)$$

이 식은 그림 7과 같이, 이제는 타원형이 아니라 원형의 등고선(에러 차분의 합)의 형태가 된다. 또한 이

것은 그림 1에서 (0,0) 자신과 (0,0)으로부터 가장 가까운 거리에 있는 4점((0,-1), (-1,0), (1,0), (0,1))만을 이용하므로 정방행렬(square matrix)이 된다. 이 행렬을 모델 1이나 모델 2와 같은 방법으로 해를 구하는 것은 기존의 방법^[3]의 모델과 일치한다. 기존의 방법^[3]은 본 논문 전체의 특수한 경우가 되며 기존의 방법^[3]을 일반화한 것이 본 논문이다. 모델 3에서는 모델 1이나 모델 2에서처럼 역행렬을 직접 사용하지 않고 기존의 방법^[3]에서처럼 다음과 같은 방법으로 계산량을 많이 줄일 수 있다. 즉, x, y가 분리가능(separable)하다고 가정하면 움직임 추정 에러의 수학적 모델을 식 (11)로 놓을 수 있다.

$$E(x) = a|x - b|^r + c \quad (11)$$

(r=2, a>0, |b|<1, c≥0)

여기서 b는 정확한 움직임 벡터이며 x는 b 주변의 벡터 중 하나라고 한다면 그림 8(γ=2)처럼 나타낼 수 있다.

$$P_0 = \frac{1}{256} \sum_{m=0}^{15} \sum_{n=0}^{15} |Y(m, n) - Y'(m + m_0, n + n_0)|^2 \quad (12)$$

그림 8에서 P₀는 식(12)와 같이 정확소 움직임 벡터 (m₀, n₀)의 MC 에러의 자승평균(square mean)이고 Y(m, n)과 Y'(m, n)은 각각 현재 프레임과 이전 프레임의 휘도(luminance)값이다. 움직임 벡터의 수평 방향(즉, x축 방향)만을 예로 설명하고 수직방향(즉, y축 방향)은 x, y가 분리가능하기 때문에 같은 방법으로 구하면 된다. 그림 8에서 정확소단위의 움직임 벡터 주변의 MC 에러 평균인 P₁과 P₋₁을 정의하면 MC 예측 에러(prediction error) 모델과 MC 예측 에러인 P₀, P₁, P₋₁을 이용하여 다음과 같이 나타낼 수 있다.

$$\frac{P_1 - P_0}{P_{-1} - P_0} = \frac{|1 - b|^r - |b|^r}{|1 + b|^r - |b|^r} = g(b) \quad (13)$$

반화소 단위의 움직임 벡터의 경우 식(13)은 다음과 같이 표현될 수 있다.

$$\begin{aligned} MV' &= (m_0 - 0.5, n_0) \text{ if } b = g^{-1}\left(\frac{P_1 - P_0}{P_{-1} - P_0}\right) < -\frac{1}{4} \\ MV' &= (m_0 + 0.5, n_0) \text{ if } b = g^{-1}\left(\frac{P_1 - P_0}{P_{-1} - P_0}\right) > \frac{1}{4} \\ MV' &= (m_0, n_0) \text{ if } -\frac{1}{4} \leq b \leq \frac{1}{4} \end{aligned} \quad (14)$$

여기서 MV'는 수평방향의 새로운 움직임 벡터이고

또다른 수직방향의 MC 에러 2개를 사용하여 같은 방법으로 새로운 움직임 벡터를 구한다. 이와 비슷한 방법에 의해서 4분화소(quarter-pixel)단위와 같은 다른 정밀도(precision)의 부화소도 쉽게 유도될 수 있다.

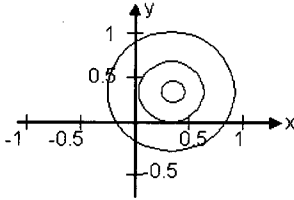


그림 7. 모델 3의 에러값을 평면적으로 나타낸 그림
Fig. 7. Error values of type 3 expressed by contour line.

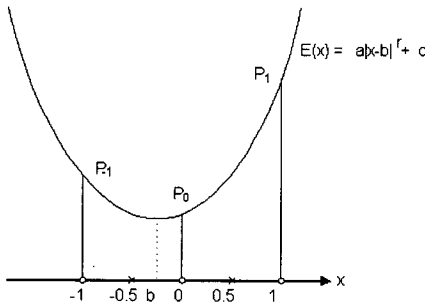


그림 8. MC 예측 에러의 모델
Fig. 8. Model of MC prediction error.

$\gamma=2$ 일 때 식(14)는 식(15)로 간단히 표현되어진다.

$$\begin{aligned} MV' &= (m_0 - 0.5, n_0) \text{ if } 3(P_{-1} - P_0) < (P_1 - P_0) \\ MV' &= (m_0 + 0.5, n_0) \text{ if } (P_{-1} - P_0) > 3(P_1 - P_0) \\ MV' &= (m_0, n_0) \text{ otherwise.} \end{aligned} \quad (15)$$

또한 $\gamma=1$ 일 때 식(14)는 식(16)으로 간단히 표현되어진다.

$$\begin{aligned} MV' &= (m_0 - 0.5, n_0) \text{ if } 2(P_{-1} - P_0) < (P_1 - P_0) \\ MV' &= (m_0 + 0.5, n_0) \text{ if } (P_{-1} - P_0) > 2(P_1 - P_0) \\ MV' &= (m_0, n_0) \text{ otherwise.} \end{aligned} \quad (16)$$

2에 의한 곱셈은 1비트 이동 연산자(shift operation)에 의해 쉽게 구현될 수 있고 3에 의한 곱셈도 1비트 이동 연산자에 의한 2배 곱셈후 1번 더하면 되므로 이 모델에 의한 곱셈은 단 한번도 없게 된다.

4. 수정 모델 2 (weighted least square solution)
그림 1에서 식(5)의 해를 구하는 데 있어 9개의 좌

표를 그냥 이용하기 보다는 (0,0)에서 더 가까운 좌표에 가중치를 주어서 계수 값을 구하는 방법이다.(그림 9) 이는 전역 탐색법에 의해서 이미 찾은 정화소의 좌표로부터 가까울수록 에러가 작은 것이므로 가까운 좌표에 의한 영향을 더 크게 하여 정확성을 높이는 생각에서 착안되었다.

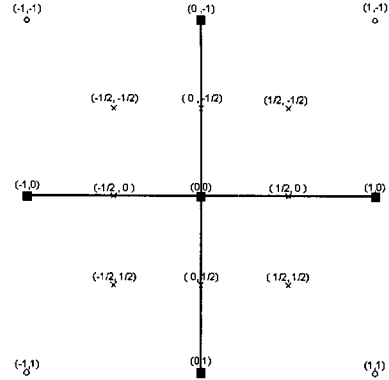


그림 9. (0,0)을 비롯한 5개의 점에 가중치를 준 것 (■ 부분이 가중치를 준 부분)
Fig. 9. 5 weighted points including (0,0) (■ is weighted point).

이것은 모델 2의 식(6)에 가중치 행렬 W를 곱하는 것이 추가된다. 우선 (0,-1), (-1, 0), (1, 0), (0, 1)에는 s(side)배의 가중치를 적용하고 (0, 0)에는 c(center)배의 가중치를 적용하면 식 (17)과 같다. 이 때 최소자승법은 식 (18)과 같다.

$$\text{대각행렬(diagonal matrix) } W = \begin{pmatrix} 1 & & & & & & & & \\ & s & & & & & & & \\ & & 1 & & & & & & \\ & & & s & & & & & \\ & & & & c & & & & \\ & & & & & s & & & \\ & & & & & & 1 & & \\ & & & & & & & s & \\ & & & & & & & & 1 \end{pmatrix} \quad (17)$$

$$\begin{aligned} w \cdot b &= (w \cdot A)x \\ \Rightarrow x &= (A^T w^T w A)^{-1} A^T \cdot w^T \cdot w \cdot b \end{aligned} \quad (18)$$

5. 수정 모델 3 (weighted least square solution)
모델 3의 식(10)을 사용함에 있어 이용할 수 있는 9개의 좌표를 모두 이용하되 모델 3에서 사용되었던 5개의 좌표((0,-1), (-1, 0), (0, 0), (1, 0), (0, 1))에 가중치를 주어 최소 자승법을 사용하는 방법이다. 구하는 과정은 수정 모델 2와 같다.

6. 부분 보간(partial interpolation) 모델 3

곱셈에 의한 복잡성(complexity)이 전혀 없는 모델 3에 의해 우선 반화소를 먼저 찾은후 정확도를 높이기 위해 바로 그 반화소에서 제일 거리가 가까운 반화소와 모델 3에 의해 찾아진 반화소만을 보간에 의해 다시 에러차분의 합을 구한 후, 그 값들을 비교한다. 따라서 그림10과 같이 3가지 경우가 있을 수 있다. 그림 10에서 정확소의 위치(0, 0)에서의 에러차분의 합은 이미 정확소 단위로 움직임을 찾을 때 구했으므로 그대로 이용한다. 반화소 추정을 위해 전역탐색시 필요한 보간은 정확소 주변의 8개의 반화소인데 비해 제안하는 방법에 의해 필요한 보간은 그림10의 (a)와 (b)의 경우 3개, (c)의 경우 4개가 필요하므로 보간을 이용한 전역 움직임 추정과 거의 같은 성능에 비해 계산량은 절반 이하가 된다.

IV. 실험 결과 및 고찰

제시한 방법의 성능평가를 위하여 flower garden 22번째 프레임(352×240), table tennis 2번째 프레임(352×240), football 17번째 프레임(352×240)에 대해 실험하였다. 여기서 경계(boundary)를 어떻게 처리하느냐에 따라 PSNR값의 차이가 있으므로 경계 부분의 처리에 의한 효과를 배제하기 위해 경계에 닿아 있는 매크로블록에서는 제안된 방법을 사용하지 않고 실험하였다. 따라서 이 값들은 상대적인 성능 비교에만 의미 있고 실제로 경계 부분에서도 사용한다면 PSNR값이 더 올라감을 볼 수 있었다.

실험결과에서, 제안된 모든 모델이 정확소단위로 탐색(search)했을 때와 보간을 사용해서 반화소단위로 움직임 벡터를 찾았을때 사이의 PSNR값을 나타내고 있다. 모델 1, 2, 3만을 놓고 볼 때 대체적으로 9개의 계수값을 사용하는 모델 1이 가장 좋고, 5개의 계수를 사용하는 모델 3의 PSNR이 좀 떨어지지만, 모델 2의 경우 flower garden에서는 모델 3보다 PSNR이 작고 football에서는 모델 1보다 근소하게 좋은 것을 볼 수 있다. 이것은 모델 2에서 사용한 최소자승법(least square solution)이 정확한 해(correct solution)가 아니라 최선의 해(best solution)라서 (0,0)좌표로부터 많이 떨어진 좌표의 영향들을 많이 받기 때문이다.

이것을 가중치를 주어서 개선한 수정 모델 2는 모델 1보다 더 좋거나 근소하게 PSNR이 떨어지고, 같

은 방법으로 가중치를 준 수정 모델 3도 모델 3보다 성능이 더 좋아지는 것을 볼 수 있다. 물론 계산량은 모델 3이 가장 적고 수정 모델 3은 가중치만큼의 계산만이 더 첨가될 뿐이다. 부분 보간 모델 3의 경우 PSNR이 보간을 이용한 반화소 단위의 전역 탐색과 거의 같은 정도의 화질을 나타내고 있다. 부분 보간 모델 3은 보간을 이용한 반화소 단위의 전역 탐색보다 계산량은 절반이하이면서도 성능은 거의 같다는 것을 알수 있다.

표 1. 실험 결과(PSNR)

Table 1. Simulation Results(PSNR).

(MSE)

	flower garden	table tennis	football
integer full search	24.4797dB	32.6099dB	23.4561dB
half full search (using interpolation)	25.2660dB	33.4284dB	24.1335dB
모델 1	25.2281dB	32.9716dB	23.8678dB
모델 2	25.1100dB	32.9675dB	23.8699dB
모델 3	25.1634dB	32.9663dB	23.8328dB
수정 모델 2 (s : 2 c : 2)	25.2137dB	33.0072dB	23.9149dB
수정 모델 3 (s : 2 c : 2)	25.1700dB	32.9699dB	23.8482dB
부분보간모델 3	25.2647dB	33.3533dB	24.0062dB

표 2. 영상에 따른 최적 가중치

Table 2. Optimum weights for images.

수정 모델 2	flower garden 22th fr.	table tennis 2nd fr.	football 17th fr.
PSNR 최적가중치 (s, c)	25.2258dB (s: 2.9, c: 2.8)	33.0881dB (s: 1.5, c: 1.4)	23.9226dB (s: 2.4, c: 1.2)

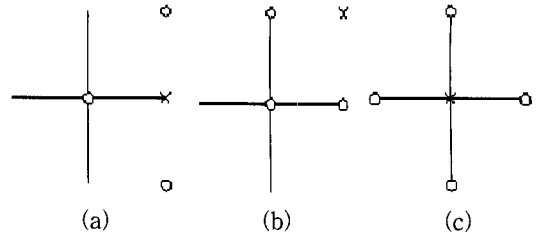
표 1에서는 가중치를 9좌표 중 5좌표에 2배씩 준 것을 나타내었다. 실제 최적의 가중치는 각 영상마다 다를 수 있는데 실험에 의하면 표 2와 같다. 또한 이 실험에서 에러 차분의 합을 구할 때 MSE(Minimum Square Error)를 사용하였다. MAE(Minimum Absolute Error)로 할 때는 실험결과가 모델에 따라 일정하지 않았는데 이는 MSE가 PSNR공식의 log속의 분모의 형태에 더 잘 맞고, 본 연구에서 모델로 설정한 에러 방정식(식(1), 식(5), 식(10))이 모두 자승함수(square function)의 형태이므로 더 잘 맞기 때문이라고 생각된다.

표 3. 제안한 방법의 계산량 비교
Table 3. Comparison of the computational complexity of the proposed methods.

	덧셈 횟수 (횟수/sec.)	곱셈 횟수 (횟수/sec.)
정화소 단위 전역탐색	21,212,430,000	10,606,215,000
반화소 단위 전역탐색	222,486,790	82,861,056
모델 1	2,751,246	0
모델 2	2,670,327	890,109
모델 3	161,838	0
수정 모델 2	2,670,327	890,109
수정 모델 3	2,184,813	525,974
부분 보간 모델 3 (평균값)	124,165,710	32,223,744

(MSE, CCIR601:720×480, 29.97 Hz)

표 3에서는 제안하는 방법의 계산량을 비교하였다. 이 비교에서는 CCIR 601 포맷(740×480)을 기준으로 하였고 프레임율(frame rate)은 29.97Hz로 하였다. 탐색 영역은 -16 ~ +15 (정화소 단위), -16 ~ +15.5 (반화소 단위)이며 매크로블록의 크기는 16×16으로 하였다. 표 1에서 사용한 에러차분의 합이 MSE이므로 표 3에서 계산된 에러차분의 합에도 MSE의 계산에 필요한 만큼의 곱셈수가 늘어났다. 부분 보간 모델3은 그림 10과 같이 3가지 경우가 섞여 나오므로 평균한 계산량이다. 덧셈과 뺄셈은 같은 연산으로 취급하여 덧셈으로 표시하였고 곱셈과 나눗셈도 곱셈으로만 표시하였으며, 비트 이동 연산자(shift operation)에 의해 쉽게 구현이 가능한 2의 제곱 형태의 곱셈과 나눗셈의 횟수는 표 3에서 비교된 모든 방법과 모델에서의 곱셈 횟수에서 제외하였다. 모델 2의 경우 의사 역행렬된 결과가 2의 제곱 형태가 아니어서 모두 그대로 곱셈을 해야 하므로 모델 1이나 모델 3에 비해 곱셈 횟수가 많아지게 되었으며, 수정 모델 2의 경우는 이미 가중치가 곱해져서 의사 역행렬이 된 행렬을 에러차분의 합들에 곱하게 되므로 덧셈과 곱셈의 횟수는 모델 2의 횟수와 동일하게 된다. 부분 보간 모델 3의 경우는 표 1의 결과에서 보여지듯이 성능은 반화소 단위 전역탐색법과 거의 비슷하지만 덧셈과 곱셈의 횟수는 반화소 단위 전역탐색법에 비해 절반 정도밖에 되지 않는 것을 볼 수 있다.



× : 모델 3 방법에 의해 구해진 반화소
○, × : 보간에 의해 다시 에러차분의 합을 구해야 하는 반화소
× : halfpixel achieved by type 3
○, × : halfpixel supposed to calculate the sum of difference by interpolation

그림 10. 부분 보간 모델 3에 의해 필요한 보간의 횟수
Fig. 10. Needed number of interpolation by P.I. type 3.

V. 결론

본 연구에서는 동영상의 부호화시 시간이 가장 많이 소요되는 움직임 추정에서 부화소 단위로 움직임을 추정, 보상하여 더욱 정밀하게 보상을 할 뿐만 아니라 소요되는 부호화 시간을 줄이는 알고리즘에 관하여 연구하였다. 단계적인 수학적 모델로서 첫 번째 모델은 정화소와 그 주위의 8점을 이용하여 9개의 계수를 가지는 2차 방정식을 구하고 그것을 이용하여 반화소 단위의 움직임을 추정하였고, 두 번째와 세 번째 모델은 첫 번째 모델에서 일부 항들을 제거하여 좀 더 간단화된 모델로서 반화소 단위의 움직임을 추정하였다. 네 번째와 다섯 번째 모델은 앞의 모델에서 정화소 단위의 움직임 벡터로부터 가까운 몇 개의 점에 가중치를 주어 계산함으로써 성능을 향상시켰다. 마지막 방법으로는 모델 3에 의해 구해진 반화소의 주위만을 부분적으로 보간하여 정확성을 더욱 향상시켰다.

제안한 방법을 동영상에 적용한 결과 보간을 이용한 반화소 단위 전역탐색법에 비해 소요시간과 계산량을 줄이면서도 정밀도는 거의 비슷한 수준까지 단계적으로 가까워지는 것을 알 수 있었다. 이는 원하는 하드웨어의 요구사항에 맞추어 정밀도와 복잡도를 조절하여 적용할 수 있음을 나타낸다. 제안한 방법은 반화소 단위의 움직임 추정 및 보상을 하는 MPEG-2와 그것을 기반으로 하는 HDTV나 각종 동영상의 압축의 부호화와 전송에 기여할 것으로 기대된다.

※ 본 연구는 한국전자통신연구원(ETRI)의 “개방형 대용량 통신처리시스템 개발” 과제의 연구비 지원하에 수행되었음

참 고 문 헌

- [1] J. R. Jain, and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE trans. on comm.*, Vol. COM-209, No. 12, Dec. 1981.
- [2] T. Koga, et al, "Motion compensated interframe coding for video conferencing," *National Telecommunication Conference*, Nov. 29- Dec. 3, 1981, New Orleans, LA, USA.
- [3] J. Jeong, and W. Ahn, "Subpixel-accuracy Motion Estimation Using a Model for Motion Compensated Error," *PCS(Picture Coding Symposium)'93*, 13.4. March. 1993.
- [4] Y. Senda, H. Harasaki, and M. Yano, "A Simplified Motion Estimation Using an Approximation for the MPEG-2 Real-Time Encoder," *ICASSP'95*, Vol. 4, pp. 2273-2276, 1995.
- [5] K. T. Choi, S. C. Chan and T. S. Ng, "A New Fast Motion Estimation Algorithm using Hexagonal Subsampling Pattern and Multiple Candidates Search," *Proc. ICIP'96*, Vol. 2, pp. 497-500, 1996.
- [6] Yuzo Senda, Hidenobu Harasaki and Mitsuhara Yano, "Theoretical Background and Improvement of a simplified Half-Pel Motion Estimation," *Proc. ICIP'96*, Vol. 3, pp. 263-266, 1996.
- [7] X. Li and C. Gonzales, "A Locally Quadratic Model of the Motion Estimation Error Criterion Function and its Application to Subpixel Interpolations," *IEEE Trans. Circuits and Systems for Video Tech.* Vol. 6. No. 1, pp. 118-122, Feb., 1996.
- [8] Motion Picture Expert Group, "MPEG II Test Model 5," *Doc. ISO-IEC/JTC1/SC29/WG11/MPEG93*, Mar. 1993.
- [9] X. Lee and Y. Zhang, "A Fast Hierarchical Motion-Compensation Scheme for Video Coding Using Block Feature Matching," *IEEE Trans. Circuits and Systems for Video Tech.* Vol. 6. No. 6, pp. 627-635, Dec., 1996.
- [10] L. Liu and E. Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Trans. Circuits and Systems for Video Tech.* Vol. 6. No. 4, pp. 419-422, Aug., 1996.
- [11] B. Liu and A. Zaccain, "New Fast Algorithm for the Estimation of Block Motion Vectors," *IEEE Trans. Circuits and Systems for Video Tech.* Vol. 3. No. 2, pp. 148-157, April, 1993.
- [12] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438-442, Aug. 1994.
- [13] K. M. Nam, J. S. Kim, and R. H. Park, "A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 344-351, Aug. 1995.

저 자 소 개



徐廷旭(正會員)

1996년 2월 : 한양대학교 전자통신 공학과 공학사. 1996년 3월 ~ 현재 : 한양대학교 대학원 전자통신 공학과 석사과정. 주관심 분야: 디지털 신호처리, 영상 통신 및 압축, 디지털 통신 등



鄭濟昌(正會員)

1980년 2월: 서울대학교 전자공학과 공학사. 1982년 2월: 한국 과학 기술원 전기전자공학과 공학석사. 1990년 8월: 미시간대학교(앤아버) 전기공학과 공학박사. 1982년 2월 ~ 1986년 7월: 한국방송공사 기술 연구소 연구원(뉴미디어 연구개발). 1990년 9월 ~ 1991년 1월: 미시간대학교(앤아버) Postdoctorial Research Fellow. 1991년 2월 ~ 1995년 2월: 삼성 전자 멀티미디어 연구센터 신호처리연구소 수석연구원 (HDTV 및 멀티미디어 연구 개발). 1995년 3월 ~ 현재: 한양대학교 전자통신공학과 조교수. 주관심 분야: 영상 및 음성 압축, 영상처리, 디지털 신호처리, 디지털통신, VLSI설계 등.