

움직임 벡터의 신뢰성 향상을 위한 스무딩 방법에 관한 연구

(A Study on Motion Vector Smoothing for Reliability Improvement)

金 珍 台*, 金 基 賢**, 崔 宗 秀*

(Jin Tae Kim, Ki Hyun Kim and Jong Soo Choi)

要 約

본 논문에서는 움직임 벡터의 신뢰성 향상 및 움직임 벡터의 부호화 효율을 높이는 움직임 벡터 스무딩 기법을 제안한다. 스무딩 과정은 움직임 벡터 필드에서 움직임 벡터 상관도에 의해 영역을 형성시키고 각 영역내에서 스무딩 작업을 행하는데, 영역의 형성은 움직임 벡터의 각도와 크기를 이용한다. 움직임 벡터 필드에서 3×3 크기의 창내부에서 중심 벡터의 영역이 형성되면 중심 벡터를 영역내의 최소의 오차를 갖는 움직임 벡터로 대체한다. 스무딩 결과, 움직임 벡터의 신뢰성이 많이 향상되었으며, 부가적으로 0.967에서 6.773 bits/block 정도의 전송량 감소의 효과를 보았다.

Abstract

In this paper, we propose a new motion vector smoothing scheme which has high reliability and coding efficiency of motion vectors. In motion vector field, groupings are made using angle and magnitude of the motion vectors. In each group, the unreliable motion vectors are corrected by the motion vector smoothing. In 3×3 window, motion vector of the current block is made to group, and the motion vector smoothing is performed only in that group. Result of computer simulation shows much improvement of the reliability of motion vectors. Moreover, coding bits of the motion vector are diminished from 0.967 to 6.773 bits per block.

1. 서 론

영상 통신(visual communication)에 있어서 프

레이프간(interframe) 부호화는 높은 데이터 압축을 가능하게 한다. 동영상의 경우 공간 상관성(spatial correlation) 뿐만 아니라 시간 상관성(temporal correlation)을 갖고 있는데, 프레임간 부호화는 시간 상관성을 이용하여 시간축의 중복성을 제거하는 방법이다. 시간상의 중복성을 제거하기 위해서는 정확한 움직임 추정이 필요한데, 이를 프레임간 부호화에 적용한 방법이 움직임 보상 부호화(motion compensated coding: MCC)이다. 움직임 추정 방법에는 크게 화소 순환 알고리즘(pel recursive

* 正會員, 中央大學校 電子工學科
(Dept. of Elec. Eng., Chungang Univ.)

** 正會員, 東洋電子通信 中央研究所
(R&D Center, Oriental Telecommunication Co., Ltd.)

接受日字 : 1993年 5月 4日

algorithm: PRA)과 블럭 정합 알고리즘(block matching algorithm: BMA)이 있다.^[1] 블럭별로 움직임 벡터(motion vector)를 찾아 블럭내의 모든 화소에 같은 움직임을 적용하는 블럭 정합 알고리즘은 실시간 하드웨어 구현이 용이하기 때문에 HDTV (high definition television)^[2,3] 및 MPEG (moving picture experts group)^[4,5]의 비디오 압축 알고리즘 등에 널리 사용되고 있다. 블럭 정합 알고리즘을 근간으로 하는 움직임 보상 예측 부호화 방법은 움직임 벡터와 예측 오차(prediction error)가 함께 부호화된다. 이때 효율적인 부호화 방법을 사용하므로써 움직임 벡터와 예측 오차가 가능한 적은 정보량을 갖게 해주는 것이 매우 중요하다. 움직임 벡터에 대한 정보량은 사용하는 부호화 기법 및 전송률에 따라 다르지만, 예를 들어 384Kbits/s의 전송율을 갖는 코덱(codec)에서 전체 정보량의 약 40%가량을 차지하는 경우도 있다.^[6] 이렇게 움직임 벡터가 많은 정보량을 갖고 있는 경우에는 움직임 벡터의 효율적인 부호화는 매우 중요한 문제가 된다.

움직임 벡터 부호화 방법은 먼저 블럭 정합 알고리즘으로 추정된 움직임 벡터에 엔트로피(entropy) 부호화를 적용시키는 방법이 제안되었다.^[7,9] 블럭 정합 알고리즘으로 찾아낸 움직임 벡터는 영상내 움직임의 크기와 잡음 등의 영향을 받아 실제 움직임(real motion)을 표현해 주지 못하는 경우가 빈번히 발생하고, 움직임 벡터 상호간의 상관도도 떨어지기 때문에 엔트로피 부호화의 효과는 낮다. 움직임 벡터의 엔트로피 부호화 효과를 높이기 위해서 불연속적이고 신뢰성 없는 움직임 벡터를 스무딩(smoothing)하여 움직임 벡터를 교정하는 기법이 연구되어 왔다.^[10,13] CCITT의 RM 7^[11]에서는 배경 영역에 대해 움직임 벡터의 신뢰도를 높이는 스무딩 기법이 사용되었고, 조건부 전송을 가진 움직임 벡터 부호화 기법이 Choi 등^[12]에 의해 제안되었다. 그리고 움직임 추정을 확실적인 구조와 결합시켜 스무딩하는 방법^[14]과 계층적(hierarchical) 움직임 추정에서 움직임 벡터를 추정할 때 스무딩 항을 고려하는 방법^[15]들도 연구되었다.

본 논문에서는 움직임 벡터의 신뢰성(reliability)을 향상시켜 가능한 한 실제 움직임을 표현할 수 있도록 하고, 움직임 벡터의 엔트로피 부호화 효율도 높이는 움직임 벡터 스무딩 방법을 제안한다. 제안한 방법은 움직임 벡터에 의해 영역을 형성하고 각 영역에서 신뢰성이 없는 움직임 벡터를 스무딩 작업과 예측 오차를 고려하여 수정한다. 본 논문의 구성은 다음과 같다. II장에서 움직임 벡터의 효율적인 부호화

를 위해 개발된 엔트로피 부호화 방법들과 엔트로피 부호화의 효율성을 높이기 위해 블럭 정합 알고리즘에서 얻어진 움직임 벡터를 스무딩하는 기존의 방법을 소개한다. III장에서는 본 논문에서 제안한 움직임 벡터의 신뢰성 향상을 위한 스무딩 방법에 대해서 기술한다. IV장에서는 컴퓨터 시뮬레이션 결과 및 검토를 하며, V장에서 결론을 맺는다.

II. 움직임 벡터 부호화 기법과 기존의 스무딩 방법

1. 움직임 벡터의 엔트로피 부호화

블럭 정합 알고리즘으로 추정된 움직임 벡터는 일반적으로 가변 길이 부호화(variable length coding) 방법을 사용하여 부호화된다.^[17] 가변 길이 부호화 방법은 단순히 현재 프레임의 움직임 벡터를 전송하는 방법과 움직임 벡터의 시간적 상관성을 고려하여 이전 프레임의 움직임 벡터와 현재 프레임의 움직임 벡터의 차이를 전송하는 방법이 있다. Iinuma 등^[6]은 움직임 벡터 차의 분포가 제로(zero) 값에 집중되어 있는 성질을 이용하여 움직임 벡터 부호화에 엔트로피 부호화를 적용하였다. Koga 등^[7]은 움직임 벡터가 수평축 성분과 수직축 성분으로 구성되어 있는 성질을 이용하여 움직임 벡터 부호화에 다음과 같은 3가지 방법을 제안하였다.

- (1) 2차원 부호화 기법 : 한 블럭의 수평, 수직 움직임 벡터를 한 쌍으로 부호화 한다
- (2) 1차원 부호화 기법 1 : 한 블럭의 수평, 수직 움직임 벡터를 별도로 부호화 한다. 수평, 수직 성분에 대해 각각 별도의 부호 집합을 사용한다.
- (3) 1차원 부호화 기법 2 : 한 블럭의 수평, 수직 움직임 벡터를 별도로 부호화 한다. 수평, 수직 성분에 같은 부호 집합을 사용한다.

Koga 등은 Iinuma 등의 방법과 자신들이 제안한 방법들을 결합하여 비교, 검토하였다. 결과적으로 2차원 부호화 기법이 가장 효율적임을 알 수 있으나, 하드웨어 구현의 용이성을 고려할 때 성능은 약간 떨어지지만 1차원 부호화 기법 2가 가장 효과적이라는 결론을 내렸다.

한편 Choi 등^[12]은 움직임 벡터를 부호화하는데 다음과 같은 6가지 방법을 사용하였다. Choi 등도 Koga 등과 마찬가지로 방법 1)이 가장 우수한 성능을 보인다고 결론짓고 있다. 따라서 본 논문에서는 움직임 벡터를 부호화할 때 2차원 부호화 기법을 사용한다.

방법 1) 2차원 부호화 기법

- 방법 2) 움직임 벡터 차의 2차원 부호화 기법
- 방법 3) 1차원 부호화 기법 1
- 방법 4) 1차원 부호화 기법 2
- 방법 5) 움직임 벡터 차의 1차원 부호화 기법 1
- 방법 6) 움직임 벡터 차의 1차원 부호화 기법 2

2. 움직임 벡터의 스무딩과 부호화 방법

본 절에서는 블럭 정합 알고리즘을 통해 얻어진 움직임 벡터를 스무딩함으로써 움직임 벡터의 정보량을 감축시켜 움직임 벡터의 엔트로피 부호화 효율성을 높이는 기존의 기법들을 소개한다.

실시간 구현의 용이성 때문에 현재 광범위하게 사용되고 있는 블럭 정합 알고리즘을 이용하여 찾아낸 움직임 벡터는 물체의 실제 운동과는 관계 없이, 정해진 왜곡 평가 함수를 최소화시키는 방향으로 각 블럭별로 독립적으로 결정된다. 이와 같이 얻어진 움직임 벡터는 회전 운동이나 물체의 가까워짐 및 멀어짐 등의 운동을 제대로 반영하지 못한다. 또한, 하나의 블럭내에 2개 이상의 서로 다른 움직임을 갖는 물체가 포함되어 있는 경우에 BMA로 구해진 움직임 벡터는 신뢰성의 문제가 있다. 즉 실제의 움직임을 표현하는데 문제가 있다. 또한 잡음 등의 영향으로 물체의 실제 움직임과 관계없이 움직임 벡터가 불연속적으로 나타나기 때문에 인접한 블럭의 움직임 벡터 간에도 상관성이 존재하지 않아 엔트로피 부호화 기법을 적용하여도 정보량이 별로 줄어들지 않는다.¹⁶⁾ 이와 같은 문제점을 해결하려는 노력들이 있었다.

1) RM 7의 움직임 벡터 스무딩 방법¹¹⁾

CCITT의 RM 7에서는 배경 영역에서 불연속적이고 신뢰성 없는 움직임 벡터를 제거하기 위하여 BMA로 움직임 벡터를 찾은 다음 벡터 스무딩 과정을 거친다. 현재 프레임에서 교정중인 블럭의 움직임 벡터를 MV, 현재 블럭과 이전 프레임의 같은 위치에 있는 블럭과의 차의 절대값의 평균을 FD(frame difference), 현재 블럭과 움직임 벡터에 의해 이동된 위치에 있는 이전 프레임의 블럭과의 차의 절대값의 평균을 DFD(displaced frame difference)라 할 경우 RM 7의 움직임 벡터 교정 알고리즘은 아래와 같이 표현된다.

```

if (FD < 1) then MV = (0, 0)
else if 1 < FD < 3 then
  if (FD < DFD × 2) then MV = (0, 0)
else
  if (FD < DFD × 1.1) then MV = (0, 0)

```

여기서 움직임 벡터를 (0,0)으로 한다는 것은 RM 7에서의 'No MC' 모드(mode)를 의미하며, 이것은

움직임 벡터를 부호화(전송)하지 않는다는 것을 의미한다. 이런 경우에는 각 블럭별로 움직임 벡터를 전송하는지 여부를 알려 주어야 하므로 거기에 해당되는 비트가 필요하게 된다. 따라서 움직임 벡터를 엔트로피 부호화할 경우에는 (0,0)도 하나의 움직임 벡터인 것으로 취급하여 허프만 부호화기를 구성한 후 부호화시킨다고 볼 수도 있다. 단 움직임 벡터에서 (0,0)이 차지하는 비율이 높을수록 (0,0)을 'No MC' 모드로 처리하고 움직임 벡터를 전송하지 않는 것이 유리하다.

2) 조건부 전송을 가진 움직임 벡터 부호화 기법¹²⁾

Choi 등에 의해 개발된 이 방법은 움직임 벡터의 스무딩과 조건부 전송 두 단계를 결합시킨 것이다. 블럭 정합 알고리즘으로 구해진 움직임 벡터는 잡음 등의 영향으로 물체의 실제 움직임을 표현해 주지 못하여 불규칙 하고 인접 블럭의 움직임 벡터간에도 상관성이 거의 존재하지 않는다. 이와 같은 문제점을 해결하기 위해 먼저 움직임 벡터에 스무딩 기법을 적용한다. 움직임 벡터 스무딩 기법은 움직임 벡터간의 상관성을 증가시켜 움직임 벡터에 엔트로피 부호화를 적용시킬때 움직임 벡터의 정보량을 감축시킬 수 있다. 움직임 벡터 필드에서, 현재 블럭에 대한 움직임 벡터를 중심으로 3×3 창(window)을 띄운다. 그 안에 존재하는 움직임 벡터 중 그 수가 가장 많은 벡터를 MV_{max}, 그 수를 Max라 할 경우 Choi 등¹²⁾ 이 사용한 스무딩 기법은 다음과 같다

```

if MAX > T1 then MV' = MVmax
else MV' = (0, 0)

if FD < T2 then MV' = (0, 0)

```

여기서 FD의 정의는 앞에서와 같고, T₁과 T₂는 임계값, MV'는 스무딩된 움직임 벡터를 나타낸다. 이와 같은 스무딩 과정을 거치면 움직임 벡터의 엔트로피는 감소하여 움직임 벡터 부호화시 압축율은 높아지나, 대신 예측 오차 신호의 분산이 증가하여 같은 화질을 유지하는데 있어서 필요한 비트수는 오히려 증가할 수도 있다. 따라서 예측 오차의 분산을 양자화 간격내에서 증가시키면서 움직임 벡터의 엔트로피를 낮추는 목적으로 다음과 같은 조건부 전송 과정을 첨가하였다

- 단계 1) 현재 프레임의 블럭과 여기에 이전 프레임의 같은 위치의 블럭간의 평균 제곱 오차(mean squared error: MSE)를 구한다.
- 단계 2) 현재 프레임의 블럭과 이전 프레임의 움직임 벡터에 의해 이동시킨 위치에 있는 블럭간의 평균 제곱 오차를 구한다. 이 값은 MSE'로 표시한다.

단계 3) $|MSE - MSE'|$ 값이 임계값 Tr 보다 크면 움직임 벡터를 전송하고, 그렇지 않으면 움직임 벡터를 전송하지 않는다. 즉 움직임 벡터를 (0,0)으로 한다.

이와 같은 조건부 전송 기법은 RM 7 에서 처럼 'MC Coded' 모드와 'No MC' 모드로 나누어 처리되며 각 블록별로 움직임 벡터의 전송 여부를 알려주는 비트가 추가된다. 따라서 조건부 전송을 가진 움직임 벡터 부호화 기법도 RM 7 방법과 마찬가지로 엔트로피 부호화를 적용시킬 경우 움직임 벡터 (0,0)도 하나의 부호로 취급하여 허프만 부호화기를 구성한 후 부호화시킨다고 볼 수 있다.

III. 움직임 벡터 영역화에 의한 움직임 벡터 스무딩

RM 7에서 사용하는 움직임 벡터 교정 알고리즘은 이전 프레임과의 오차(FD)와 BMA로 구해진 움직임 벡터로 보상된 이전 프레임과의 오차(DFD)를 이용하여, FD의 크기가 충분히 작거나 DFD에 비해 너무 크지 않으면 움직임 벡터를 (0,0)으로 만들어 주는 것이다. 이 방법은 움직임 벡터를 (0,0)으로 만들어 주든지 그냥 유지하던지 두가지 중의 한 동작만 수행하므로 움직임 벡터의 정보량 감축에 있어서 효과적이지 못하다. 조건부 전송을 가진 움직임 벡터 교정 방법은 RM 7 기법과 비슷한 움직임 벡터 교정 과정에, 인접한 블록의 움직임 벡터들을 이용하여 움직임 벡터들의 일관성을 증가시키는 과정이 추가되어 있다. 따라서 이 방법은 인접한 움직임 벡터들간의 상관성이 증가하여 움직임 벡터의 정보량 압축에 있어서는 유리하지만, 예측 오차의 증가를 무시하고 벡터의 일관성을 향상시킴으로써 예측 오차의 분산이 크게 증가할 수 있다는 문제가 있다. 본 논문에서 제안하는 움직임 벡터 스무딩 기법은 위 방법들의 이러한 문제점들을 고려하며, 움직임 벡터의 신뢰성 향상에 목적이 있다. 즉, 예측 오차의 증가를 허용 범위 내에 고정시키면서 움직임 벡터의 신뢰성을 향상시키고, 부가적으로 정보량을 감축시키는 것을 목적으로 하고 있다. 본 논문에서 말하는 신뢰성이란 구해진 움직임 벡터가 실제 움직임에 가깝게 표현되는 것을 말한다. 제안한 움직임 벡터 교정 방법은 다음과 같으며 FD와 DFD의 정의는 II장과 동일하다

단계 1) 블록 정합 알고리즘을 이용하여 움직임 벡터 필드를 구성한다. 하나의 움직임 벡터는 다음과 같이 표시한다.

$$MV = (x, y)$$

단계 2) 움직임 벡터를 각도(angle)와 크기(magnitude)로 변환시킨다.

$$\text{angle} = \tan^{-1} \frac{x}{y}$$

$$\text{magnitude} = \sqrt{x^2 + y^2}$$

이 때, 각도는 -180° 에서 180° 의 조건을 갖도록 한다.

단계 3) 움직임 벡터 필드에서 현재 블록의 움직임 벡터를 중심으로 3×3 크기의 창을 씌움으로 움직임 벡터를 그룹화(grouping)한다. 그룹화는 움직임 벡터의 각도와 크기를 이용 하고, 비슷한 값을 갖는 움직임 벡터를 모아 영역을 구성한다.

단계 4) 현재 처리하고자 하는 움직임 벡터(3×3 의 창에서 중심 벡터)에 대해 영역 형성 유무를 판단한다. 영역은 2개 이상의 움직임 벡터가 하나의 그룹이라고 판명된 것을 말한다. 중심 벡터에서 영역이 형성되면 단계 5로 가고, 그렇지 않으면 단계 6으로 간다.

단계 5) 중심 벡터에 의한 예측 오차를 DFD라 하고, 중심 벡터를 같은 영역내의 모든 움직임 벡터로 대치한 다음 최소의 예측 오차를 DFD'라 한다. $|DFD - DFD'| < T$ 가 만족되면 중심 벡터를 해당 DFD'의 벡터로 대치 한다.

단계 6) 배경 영역을 위한 zero vector smoothing을 한다. 즉 $|DFD - DFD'| < T$ 가 만족하면 중심 벡터를 (0,0)로 대치한다.

그림 1은 제안한 알고리즘의 흐름도이다. 본 논문에서 추구하고자 하는 방향은 다음과 같다. 영상내에서 균일한 영역과 같은 물체(object)로 구성되는 블록은 비슷한 움직임을 가져야 하고, 비슷한 크기와 방향의 움직임 벡터로 표현되어야 한다. 또한 움직이는 물체의 경계를 제외하고는 인접 블록의 움직임 벡터는 비슷한 값을 갖는다. 즉 물체별로 영역을 구성하면 하나의 대표 벡터로 표시할 수 있다. 그러나 BMA의 제한성 때문에 하나의 블록에 서로 다른 움직임을 갖는 물체가 포함될 수 있고, 물체는 병진 운동 뿐만 아니라 회전 운동과 카메라에 의한 zooming, panning 등의 현상도 있으며 잡음의 영향도 무시할 수 없다. 따라서 BMA에 의한 움직임 벡터는 이런 경우에 많은 문제점을 내포하고 있다. 이러한 문제점을 극복하기 위해 움직임 벡터를 영역화하고 이를 통하여 같은 영역내에서 움직임 벡터를 스무딩한다.

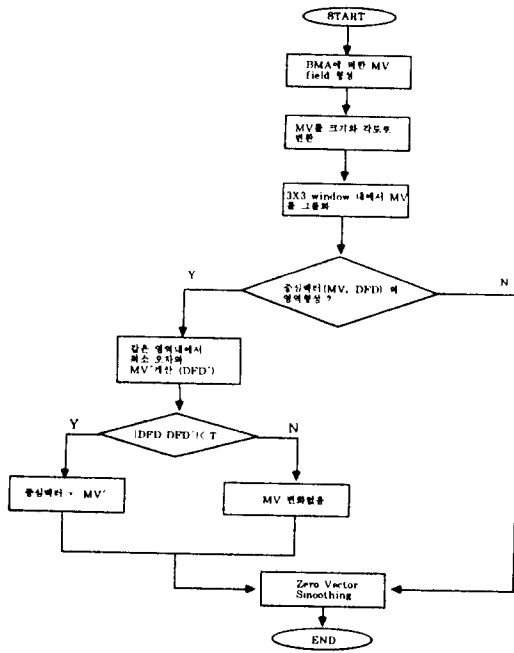
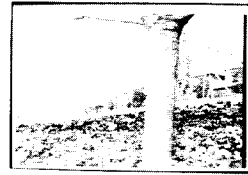


그림 1. 제안한 알고리즘의 흐름도
Fig. 1. Flow-chart of the proposed algorithm.

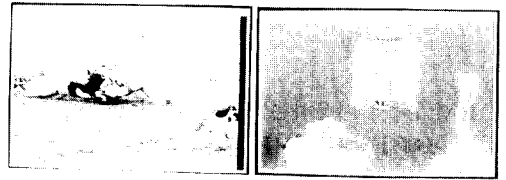
IV. 컴퓨터 시뮬레이션 결과 및 검토

본 장에서는 엔트로피 부호화의 효율성을 높이기 위해 움직임 벡터를 스무딩 시키는 기존 방법과 제안한 알고리즘을 컴퓨터 시뮬레이션을 하고, 그 결과를 기존의 방법과 비교 검토한다. 움직임 벡터 부호화 방법은 II장에서 설명한 가변 길이 부호화 방법을 이용한다. 가변 길이 부호화 방법으로는 허프만 부호화기^[17]를 사용한다. 실험 대상 영상은 MPEG 표준 영상인 'Flower Garden', 'Football', 'Popple'의 3종류를 사용하였으며 각 영상을 20 프레임씩 처리하였고, 휘도(luminance) 성분인 Y 신호만을 대상으로 하였다. 영상 크기는 720×480 이다. 원래의 영상은 비월 주사(interlaced scan)된 영상이나 본 실험에서는 홀수 필드를 프레임으로 보간하여 순차 주사(progressive scan)형태로 바꾸었다. 순차 주사 형태로 변환할 때 사용한 필터는 MPEG에서 해상도 변화에 사용되는 저역 통과 필터이다.^{4,5)} 그림 2는 실험에 사용한 원영상이다. 움직임 벡터를 구하는 움직임 추정 방법은 BMA의 전역 탐색(full search) 기법을 이용하였으며, 프레임간 최대 이동 변위는 ±

15로 하였다. 블록 크기는 8×8, 8×16, 16×8, 16×16 등으로 변화시켜 가면서 실험을 하였다. 제안한 알고리즘을 컴퓨터 시뮬레이션할 때 사용한 임계값 T는 3이다.



(a)



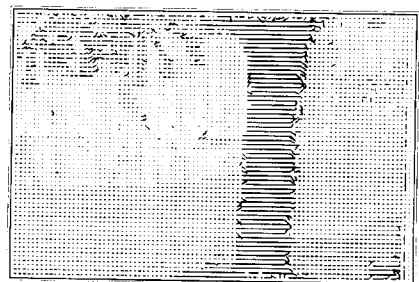
(b)

(c)

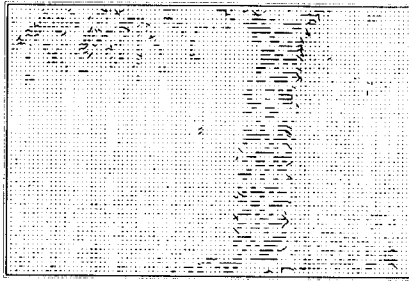
그림 2. 원 영상 (a) 'Flower Garden' 영상 (b) 'Football' 영상 (c) 'Popple' 영상
Fig. 2. Original Images (a) 'Flower Garden' image (b)'Football' image (c) 'Popple' image.



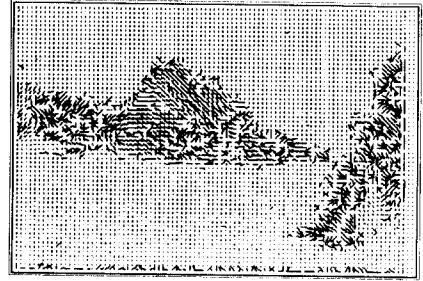
(a)



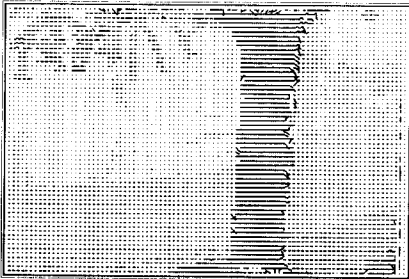
(b)



(c)



(b)



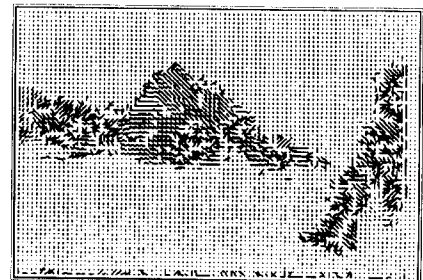
(d)



(c)

그림 3. 'Flower Garden' 영상의 움직임 벡터 필드. (a) 원래의 움직임 벡터. (b) RM 7 방법으로 스무딩된 벡터. (c) 조건부 전송 기법으로 스무딩된 벡터. (d) 제안한 방법으로 스무딩된 벡터.

Fig. 3. The motion vector field for 'Flower Garden' image. (a) Original motion vector. (b) Smoothed motion vector by RM 7. (c) Smoothed motion vector by conditional transmission. (d) Smoothed motion vector by the proposed method.



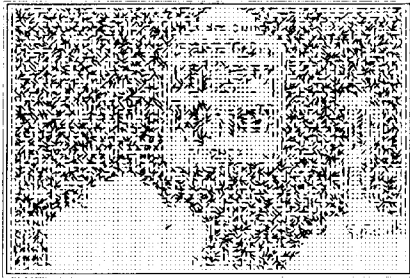
(d)

그림 4. 'Football' 영상의 움직임 벡터 필드. (a) 원래의 움직임 벡터. (b) RM 7 방법으로 스무딩된 벡터. (c) 조건부 전송 기법으로 스무딩된 벡터. (d) 제안한 방법으로 스무딩된 벡터.

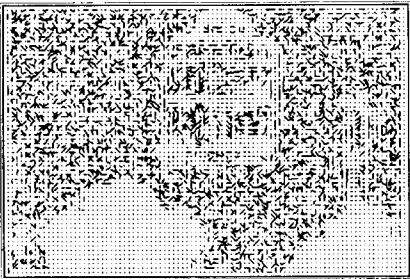
Fig. 4. The motion vector field for 'Football' image. (a) Original motion vector. (b) Smoothed motion vector by RM 7. (c) Smoothed motion vector by conditional transmission. (d) Smoothed motion vector by the proposed method.



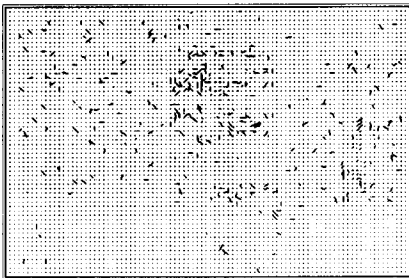
(a)



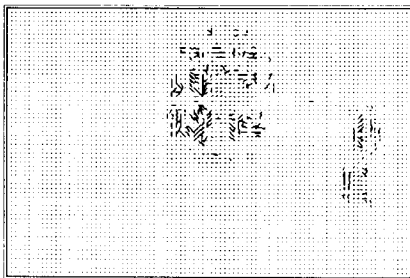
(a)



(b)



(c)



(d)

그림 5. 'Popple' 영상의 움직임 벡터 필드. (a) 원래의 움직임 벡터. (b) RM 7 방법으로 스무딩된 벡터. (c) 조건부 전송 기법으로 스무딩된 벡터. (d) 제안한 방법으로 스무딩된 벡터.

Fig. 5. The motion vector field for 'Popple' image. (a) Original motion vector. (b) Smoothed motion vector by RM 7. (c) Smoothed motion vector by conditional transmission. (d) Smoothed motion vector by the proposed method.

표 1. 예측 오차의 분산값 비교

Table 1. Comparison of variance of prediction error.

영상	블럭 크기	RM7	조건부 전송	전역 탐색	제안한 방법
Flower Garden	8 × 8	72.53	853.31	72.23	76.44
	8 × 16	88.64	963.64	88.29	92.98
	16 × 8	89.44	953.90	89.07	95.26
	16 × 16	105.47	1056.62	105.01	111.94
Football	8 × 8	108.01	409.62	107.53	118.99
	8 × 16	153.56	463.25	152.69	167.03
	16 × 8	152.22	459.95	151.44	165.05
	16 × 16	201.41	491.83	200.21	216.65
Popple	8 × 8	51.94	254.11	50.75	61.40
	8 × 16	62.28	273.82	61.47	71.06
	16 × 8	64.84	277.42	63.98	74.48
	16 × 16	75.12	287.76	74.38	84.36

표 2. 움직임 벡터의 비트율 비교

Table 2. Comparison of bit rate of motion vector. (bits/block)

영상	블럭 크기	RM7	조건부 전송	전역 탐색	제안한 방법
Flower Garden	8 × 8	4.313	2.073	5.333	3.867
	8 × 16	4.091	1.867	4.882	3.767
	16 × 8	4.187	1.929	4.921	3.801
	16 × 16	4.014	1.721	4.549	3.694
Football	8 × 8	5.402	2.829	5.632	4.551
	8 × 16	5.107	2.673	5.323	4.331
	16 × 8	5.091	2.654	5.300	4.324
	16 × 16	4.799	2.565	5.027	4.058
Popple	8 × 8	6.886	1.695	8.616	1.843
	8 × 16	6.880	1.376	8.228	1.659
	16 × 8	5.013	1.423	8.254	1.685
	16 × 16	6.609	1.288	7.737	1.542

그림 3, 그림 4, 그림 5는 각 영상의 2번째 프레임에 대해 BMA에 의해 구해진 움직임 벡터와 기존의 스무딩 방법에 의해 스무딩된 결과 그리고 제안한 방법에 의해 움직임 벡터를 스무딩한 결과를 보여 준다. 이 때 블럭의 크기는 8×8 이다. 그림 3, 그림

4. 그림 5에서 보면 제안한 방법으로 스무딩된 움직임 벡터가 물체의 움직임 별로 그룹화되었음을 알 수 있으며, 스무딩된 움직임 벡터가 실제의 움직임에 가깝게 표현되었다.

표 1과 표 2는 기존의 움직임 벡터 스무딩 알고리즘과 제안한 움직임 벡터 스무딩 알고리즘의 프레임당 평균 비트수와 예측 오차를 비교한 결과이다. 표 1과 표 2의 각 데이터는 20 프레임을 처리한 평균값이다. 비트율은 전역 탐색 움직임 보상 알고리즘으로 얻은 움직임 벡터를 2차원 허프만 부호화하여 얻은 값이다. 표 1과 표 2에서 전역 탐색은 전역 탐색 알고리즘으로 얻은 움직임 벡터에 의해 움직임 보상을 수행한 결과를 나타낸다. RM 7은 CCITT에 의해 권고된 방법을 적용시킨 결과를 나타내며, 조건부 전송은 Choi 등이 제안한 조건부 전송을 가진 움직임 벡터 교정 방법을 적용시킨 결과를 나타낸다. 실험 결과 움직임 벡터 정보량 감축 측면에서는 조건부 전송을 가진 움직임 벡터 부호화 방법의 성능이 가장 우수하나, 반대로 예측 오차가 크게 증가함을 알 수 있다. RM 7의 움직임 벡터 교정 방법은 예측 오차의 증가는 거의 없으나, 움직임 벡터의 정보량 감축 측면에서는 큰 효과를 발휘하지 못한다.

제한한 움직임 벡터 스무딩 기법을 적용하면 조건부 전송을 가진 움직임 벡터 부호화 방법과는 달리 예측 오차의 증가를 일정 범위 내로 제한할 수 있다. 이는 중심 벡터를 스무딩할 때 예측 오차가 일정값 이하로 만족되어야 스무딩된 움직임 벡터로 대처하기 때문이다. 표 1을 살펴보면 전역 탐색을 이용한 움직임 보상의 예측 오차 분산에 비해 조건부 전송을 가진 움직임 벡터 부호화 기법의 예측 오차 분산은 200에서 1000 정도가 증가했으나 제안한 움직임 벡터 스무딩 기법의 예측 오차 분산은 약 10 정도만 증가하였다. 또한 표 2를 보면 RM 7 기법에서는 움직임 벡터의 정보량을 최대 2 bits/block 정도 감축시키나 제안한 움직임 벡터 스무딩 기법은 최대 7 bits/block 정도를 감축시킬 수 있음을 알 수 있다.

실험 결과, 조건부 전송을 가진 움직임 벡터 부호화 방법을 사용하면 움직임 벡터에 대한 정보량은 많이 감축할 수 있지만 예측 오차가 크게 증가하여 복원된 영상 화질의 열화가 심각하게 나타난다. 또한 RM 7의 움직임 벡터 교정 방법을 사용하면 예측 오차의 증가를 일정 한도내로 제한시켜 복원된 영상의 화질에 심각한 열화가 나타나지 않으나 움직임 벡터의 정보량을 크게 감축시키지는 못한다. 기존의 움직임 벡터 교정 방법과는 달리 제안한 움직임 벡터 스무딩 기법을 적용하면, 주관적인 관점에서는 기존의

방법보다 실제의 움직임을 표현하는 움직임 벡터로 교정되고(그림 3, 그림 4, 그림 5), 객관적인 관점에서는 예측 오차의 증가를 일정 한도내로 제한시켜 복원된 화질에 심각한 열화를 방지하고 움직임 벡터의 정보량을 크게 할 수 있다(표 1, 표 2).

V. 결 론

본 논문에서는 동영상 부호화에 가장 널리 쓰이고 있는 블럭 정합 알고리즘을 이용한 움직임 보상 예측 부호화 방법에서 움직임 벡터를 스무딩하는 기법에 대해 연구하였다. 블럭 정합 알고리즘을 적용하여 얻은 움직임 벡터는 불연속적이며 상관성이 극히 적어 엔트로피 부호화를 적용해도 크게 효과를 보지 못한다. 이러한 문제점을 해결하고자 개발된 기존의 움직임 벡터 스무딩 기법은 예측 오차의 증가를 제한하는 방향만 강조하여 움직임 벡터의 정보량을 줄이는데 효과적이지 못하거나, 움직임 벡터의 정보량을 크게 줄이기 위해 예측 오차의 증가를 무시하는 문제점을 안고 있으며 실제의 움직임에 대한 고려가 없다.

본 논문에서는 기존의 움직임 벡터 교정 기법이 안고 있는 문제점을 해결하고자 하였으며, 교정된 움직임 벡터가 가능한 한 실제 움직임을 표현할 수 있도록 하였다. 그 결과 움직임 벡터의 신뢰성이 크게 향상되었으며, 또한 예측 오차의 증가를 일정한 범위내로 유지시키면서 움직임 벡터 부호화에 필요한 평균 비트수를 줄이는 부수적인 효과도 얻었다. 컴퓨터 시뮬레이션 결과 제안한 움직임 벡터 스무딩 알고리즘은 원래의 움직임 벡터에 비해 실제의 움직임을 표현하는 움직임 벡터로 교정됨을 알 수 있었다. 객관적인 화질은 약간 떨어지나 움직임 벡터의 부호화 비트를 감소시켰다.

앞으로의 연구로 움직임 벡터의 방향 뿐만 아니라 크기의 측면에서 신뢰성을 더욱 향상시키는 방법이 필요하며, 신뢰성이 향상된 움직임 벡터는 프레임간의 보간(interpolation)에 효율적으로 이용할 수 있을 것이다.

參 考 文 獻

- [1] H. G. Musmann, P. Pirsch, and H. J. Grallert, "Advanced in picture coding", *Proc. IEEE*, vol. 73, no. 4, pp. 523-548, 1985.
- [2] *Technical Details: Digital Spectrum Compatible*, Zenith and AT&T, Sep.

- 1991.
- [3] *DigiCipherTM HDTV System Description*, General Instrument Corporation, Aug. 1991
- [4] ISO-IEC/JTC1/SC29/WG11 MPEG 92/086, "Preliminary working draft of test model 0", Mar. 1992.
- [5] ISO-IEC/JTC1/SC2/WG8/N MPEG 90, "MPEG video simulation model three (SM3)", 1990.
- [6] K. Iinuma, T. Koga, K. Niwa, and Y. Iijima, "A motion-compensated interframe codec", in *Proc. Image Coding*, SPIE Vol. 594, pp. 194-201, 1985.
- [7] T. Koga and M. Ohta, "Coding of motion vector information", in *Proc. Picture Coding Symp.*, pp. 138-139, 1987.
- [8] C. T. Chen and F. C. Jeng, "On coding of motion vectors for motion-compensated predictive/interpolative video coder", in *Proc. Visual Communications and Image Processing*, SPIE Vol. 1605, pp. 812-821, 1991.
- [9] T. Koga and M. Ohta, "Entropy coding for a hybrid scheme with motion compensation in subprimary rate video transmission", *IEEE J. Selected Areas Comm.*, vol. SAC-5, no. 5, pp. 1166-1174, 1987.
- [10] W. Geuen, "Postprocessing of motion vectors", in *Proc. Picture Coding Symp.*, pp. 191, 1987.
- [11] CCITT SGXV, Description of Ref. Model 7 (RM7), 1988.
- [12] W. Y. Choi and R. H. Park, "Motion vector coding with conditional transmission", *Signal Processing*, vol. 18, pp. 259-267, 1989.
- [13] 김기현, 김진태, 장태규, 최종수, "효율적인 부호화와 예측오차를 고려한 움직임 벡터 스무딩 기법에 관한 연구", 제5회 신호처리 합동 학술대회 논문집, pp. 719-723, 1992.
- [14] C. Stiller, "Motion-estimation for coding of moving video at 8 kbit/s with Gibbs modeled vectorfield smoothing", in *Proc. Visual Communications and Image Processing*, SPIE Vol. 1360, pp. 468-476, 1990.
- [15] T. Reuter, "A modified blockmatching algorithm with vector reliability checking and adaptive smoothing", in *Proc. Image Processing and Its Applications*, pp. 295-299, 1989.
- [16] 이상욱, 최종수, 박래홍 등, 고도 통신망에 응용될 화상 처리 기술 연구, 1차년도 최종보고서, 한국전자통신연구소, 1992. 1.
- [17] R. W. Hamming, *Coding and Information Theory*, 2nd ed., Prentice-Hall, New Jersey, 1986.

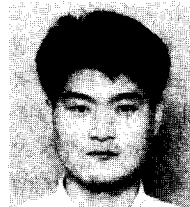
 著 者 紹 介

金 珍 台(正會員) 第 31 卷 B編 第 1 號 參照

현재 중앙대학교 기술과학연구소
연구원

崔 宗 秀(正會員) 第 28 卷 B編 第 5 號 參照

현재 중앙대학교 전자공학과 교수



金 基 賢(正會員)

1967年 9月 14日生, 1990年 2月
중앙대학교 전자공학과 졸업(공학사). 1992年 9월 중앙대학교 대학원 전자공학과 졸업(공학석사). 1992年 11월 ~ 현재 동양전자통신 (주) 중앙연구소 연구원, 주관심

분야는 영상신호처리, 동영상 데이터 압축, B-ISDN 망중단장치 연구 등임.