

論文96-33B-1-18

신경 회로망 벡터 양자화를 이용한 움직임 탐색 영역의 예측

(Motion Search Region Prediction using Neural Network Vector Quantization)

柳 大 鉉 *, 金 在 昌 **

(Dae Hyun Ryu and Jae Chang Kim)

요 약

본 논문에서는 동영상 압축의 핵심 기술인 움직임 벡터 추정에 있어서 신경 회로망을 이용한 벡터 양자화에 의해 탐색 영역을 예측하는 방법을 제안한다. 훈련영상을 입력으로 하여 전역 탐색법 등에 의하여 구해진 움직임 벡터를 이용하여 움직임 벡터 코드 북을 생성하고 이를 예측 탐색 점으로 이용한다. 움직임 벡터 코드 북을 생성하기 위해서 병렬 처리 특성과 다양한 학습 알고리즘을 갖는 신경 회로망을 이용하였다. 제안된 방법은 움직임 벡터들의 높은 공간적 상관성을 이용하게 되고 결과적으로 적은 탐색 점으로 움직임 벡터를 추정할 수 있으므로 계산량을 줄일 수 있을 뿐 아니라 움직임 벡터를 표현하기 위해 소요되는 비트 수도 크게 줄일 수 있다. 모의 실험을 통하여 제안된 방식이 기존의 고속 블럭 매칭 알고리즘보다 우수함을 보였다.

Abstract

This paper presents a new search region prediction method using vector quantization for the motion estimation. We find motion vectors using the full search BMA from two successive frame images first. Then the motion vectors are used for training a codebook. The trained codebook is the predicted search region. We used the unsupervised neural network for VQ encoding and codebook design. A major advantage of formulating VQ as neural networks is that the large number of adaptive training algorithm that are used for neural networks can be applied to VQ. The proposed method reduces the computation and reduce the bits required to represent the motion vectors because of the smaller search points. The computer simulation results show the increased PSNR as compared with the other block matching algorithms.

I. 서 론

화상회의, 화상전화, 디지털 HDTV 등에서 동영상 압축 기술은 매우 중요한 부분을 차지한다. 동영상에서는 공간 상관성(spatial correlation) 뿐만 아니라 시

간 상관성(temporal correlation)도 매우 크기 때문에, 이를 이용하여 시간축의 중복성을 제거하는 방법인 움직임 보상 부호화(motion compensated coding: MCC)를 이용하면 높은 데이터 압축율을 얻을 수 있다.

움직임 보상 부호화는 정확한 움직임 추정(motion estimation)에 의해 움직임을 보상하는 부분과 그 예측 오차(prediction error)를 부호화하는 부분으로 이루어져 있다. 이와 같이 동영상 부호화 방식에 있어서 이동 정보량의 추정은 매우 중요한 문제이며 많은 연구가 이루어져 왔다^{[1][2]}.

움직임 추정 방법에는 크게 화소 순환 알고리즘(pixel

* 正會員, 韓國電子通信研究所 (ETRI)

** 正會員, 釜山大學校 電子工學科

(Dept. of Electronics Engineering, Pusan national Univ.)

接受日字: 1995年11月14日, 수정완료일: 1995年12月20日

recursive algorithm: PRA)과 블록 정합 알고리즘(block matching algorithm: BMA)이 있다^[3]. 블록별로 움직임 벡터(motion vector)를 찾아 블록내의 모든 화소에 같은 움직임을 적용하는 블록 정합 알고리즘은 실시간 하드웨어 구현이 용이하기 때문에 HDTV(high definition television) 및 MPEG(moving picture experts group)의 비디오 압축 알고리즘 등에 널리 사용되고 있다^{[4] [5]}.

블록 정합 알고리즘을 근간으로 하는 움직임 보상 예측 부호화 방법에서는 움직임 벡터와 예측 오차(prediction error)가 함께 부호화 된다. 이때 효율적인 부호화 방법을 사용함으로써 움직임 벡터와 예측 오차가 가능한 적은 정보량을 갖게 해주는 것이 매우 중요하다. 움직임 벡터에 대한 정보량은 사용하는 부호화 기법 및 전송률에 따라 다르다. 예를 들어 384 Kbits/s의 전송율을 갖는 코덱(codec)에서 전체 정보량의 약 40%가량을 차지하는 경우도 있다. 이렇게 움직임 벡터가 많은 정보량을 갖고 있는 경우에는 움직임 벡터의 효율적인 부호화는 매우 중요한 문제가 된다^{[6] [7]}.

본 논문에서는 움직임 벡터를 추정하기 위해 신경 회로망을 이용한 벡터 양자화에 의해 탐색 영역을 예측하고 예측 점에 대해서만 왜곡(distortion)을 계산함으로써 움직임 벡터를 구하는 방법을 제안한다. 시험 영상에서 전역 탐색(full search) 법을 이용하여 움직임 벡터를 구하고 이를 입력으로 FSCL(frequency sensitive competitive learning) 신경 회로망 벡터 양자화의 코드 북을 설계하고 이것을 움직임 예측 영역으로 이용하여 블록 정합 알고리즘을 수행하여 움직임 벡터를 구한다. 따라서 결과적으로 영상에서 움직임 벡터의 공간적 상관성(spatial correlation)을 이용하게 되어, 탐색 점의 수, 즉 계산량을 줄일 뿐만 아니라 움직임 벡터 전송을 위한 소오 비트 수를 줄임으로써 결과적으로 압축율을 높힐 수 있다. 모의 실험 결과 제안된 방법은 PNSR 측면에서 비교적 성능이 우수한 고속 정합 알고리즘으로 알려진 3 단계 탐색법(three step search: TSS)에 의한 것보다 향상됨을 보인다.

움직임 예측 영역을 나타내는 코드북을 학습시키는 데는 많은 계산량이 요구된다. 그러나 이러한 계산은 인코딩되지 않은 비디오에 대해 오프 라인(off-line), 또는 비교적 저속의 디지털 신호 처리기(digital sig-

nal processor) 등에 의해 병렬로 이루어 질 수 있으며, 망의 상태에 따라 코드 북의 크기, 즉 탐색 점의 수를 적응적(adaptive)으로 변화 시킴으로써 비트율을 제어할 수 있다.

이러한 방식은 폭주(congestion) 제어 방식을 사용하는 비동기 전송 모드(asynchronous transfer mode: ATM)에서 요구형 비디오(video on demand)와 같은 응용 분야에 적용될 수 있다. 이러한 응용 분야에서는 코딩되지 않은 영상 열(sequence)이 송신 단에 항상 저장되어 있으며 수신 장치에서 요구가 있을 때 송신 서버는 요구된 영상 열을 인코딩하여 송신한다. 비디오 코더는 망으로 부터 망의 상황에 대한 피드백 정보를 수신하여, 망에 폭주이 발생하였을 때는 패킷 손실을 줄이기 위해 출력 율을 자동으로 줄인다. 이러한 적응 방식은 우선 순위에 따라 패킷을 드롭시키는 방식에 비해 상대적으로 우수한 성능을 보인다^{[8] [9]}.

본 논문의 구성은 다음과 같다. II장에서 기존의 블록 정합 알고리즘을 간단히 소개하고 그 문제점을 설명한다. III 장에서 신경 회로망 벡터 양자화를 이용하여 움직임 영역을 예측하고 이를 이용하여 움직임 벡터를 검출하는 하는 방법을 기술한다. IV 장에서는 기존의 알고리즘과 제안한 방법을 모의 실험을 통해 그 결과를 비교 검토하며, V 장에서 결론을 맺는다.

II. 블록 정합 알고리즘에 의한 움직임 벡터의 추정

블록 매칭 알고리즘은 연속하는 화상(frame) 사이에서 물체의 이동량을 검출함에 있어서 시간축으로 인접한 화상들 내에서 부영상 사이의 상관 관계(correlation)를 비교하여 최대치를 이동 보상 위치로 이용하는 방법이다^[10]. 그림 1에 블록 정합 알고리즘에 의한 움직임 검출 방식이 나타나 있다.

블록 정합 알고리즘의 처리과정은 다음과 같다. 우선 화상을 고정된 크기의 부영상으로 나눈다. 이때 이전 화상내의 부영상과의 상관계수 값이 최대가 되는 위치(오차가 제일 작게 일어나는 위치)를 구하기 위해 식 (1)과 같은 함수 $D(\cdot)$ 를 정의한다.

$$D(i, j) = \frac{1}{NM} \sum_{m=1}^M \sum_{n=1}^N G(U(m, n) - U(m+i, n+j)), \quad (1)$$

$$-p \leq i, j \leq p$$

여기서,

- $G(.)$: error power를 구하는 비선형 함수
- U : 현 화상내에서 $M \times N$ 크기의 부영상으로 구성된 블럭
- U_s : 이전화상 내에서 $(M+2p)(N+2p)$ 크기의 탐색 영역
- p : 최대 이동가능거리

이다. 이때 이동치는 $D(i, j)$ 를 최소로 하는 (i, j) 로 주어진다. $D(.)$ 값은 최적위치(optimal position)에서 멀어질수록 증가한다. 즉 정확한 이동치를 갖는 위치에서 멀어지면 멀어질수록 오차가 커지므로 이동값을 구하는 것은 $D(.)$ 값을 적게하는 위치로 찾아가는 것이다. 이와같은 가정하에서, 계산량을 줄일 수 있을 뿐 아니라 탐색 속도를 향상시키기 위한 블럭 정합 방법으로 DMD(Direction of Minimum Distortion), TSS(three Step Search) 그리고 OTS(One at a Time Search) 등의 방법이 제안되었다^{[11] [12]}.

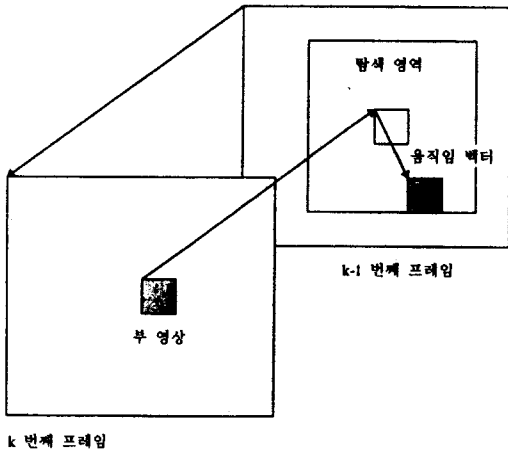


그림 1. 블럭 정합 알고리즘에 의한 움직임 검출
Fig. 1. Motion detection by block matching algorithm.

검출된 움직임 벡터를 이동 보상형 예측 또는 이동 보상형 내삽 등에 이용하기 위해서는, 운동체의 실제 이동 정보의 추출, SNR 등의 예측 성능, 전송될 이동 벡터의 데이터 량 등이 고려되어야 한다^{[13] [14]}. 위와 같은 측면을 고려할 때 블럭 정합 알고리즘에 의한 움직임 벡터 검출 방식은 널리 사용됨에도 불구하고 많은 단점을 갖고 있다. 즉 블럭 대 블럭으로 운동벡터를 검출하므로 그 블럭내에서 움직임이 균일하다는 가정이 필요하다. 이러한 가정은 상대적으로 작은 블럭(8*8 또는 16*16)에서만 만족한다. 그러나 블럭 크기가

가 작아진다는 것은 블럭의 수가 늘어나고 전송해야 할 운동벡터가 늘어나는 결과가 되므로 전송 부담이 생긴다. 또한 블럭 정합 알고리즘을 적용하여 구해진 이동벡터는 운동체의 국지적인 특성만을 이용하며 인접 벡터간들의 연관성을 고려하지 않았기 때문에 운동체의 실제 이동과 일치하지 않는 경우가 발생한다.

이런 문제점을 해결하기 위해 움직임이 일어나는 경계부분에 대해서 보다 섬세한 움직임 추정을 하기 위하여 블럭 크기를 가변시키는 등 블럭 정합 알고리즘에 대한 많은 연구가 있어왔으나, 블럭 크기를 크게 하여 움직임을 추정한 후 예측 오차가 크면 블럭을 나누는 방식의 경우에는 중복된 계산으로 인해 효율적이지 못하고, 또한 블럭 크기가 너무 다양한 경우에는 이동벡터량의 증가로 인해 부호화 시키기가 복잡하게 된다는 등의 단점을 피할 수가 없다.

Ⅲ. 신경 회로망 벡터 양자화를 이용한 움직임 벡터 추정 방식

일반적으로 블럭 정합 알고리즘에 의해 찾아진 움직임 벡터는 높은 시공간 상관성(spatiotemporal correlation)을 가지고 있기 때문에 이를 이용하여 움직임 벡터 검출의 성능을 높일 수 있다. 본 장에서는 훈련 영상(training image)에서 전역 탐색법을 이용하여 움직임 벡터를 구하고, 이를 입력으로 신경 회로망 벡터 양자화의 코드 북을 학습하여 얻어진 결과 코드북을 이를 움직임 예측 영역으로 이용하여 움직임 벡터를 추정하는 방법을 제안한다. 이는 영상에서 움직임 벡터의 공간적 상관성(spatial correlation)을 이용한 것으로써, 움직임 벡터 추정시 탐색 점의 수를 줄임으로써 계산량을 줄일 뿐 아니라 움직임 벡터 전송을 위한 추가 정보의 양을 줄여 데이터 압축율을 높일 수 있다.

1. 벡터 양자화(Vector Quantization)를 이용한 탐색 영역의 예측

벡터 양자화는 입력 데이터에 내재하는 통계적 특성을 이용하는 양자화 기법이라 할 수 있다. 먼저 입력 벡터 공간을 여러 개의 영역으로 나누고 각 영역에 대해 대표 벡터(representive vector)를 계산한다. 다음에 양자화될 데이터 벡터가 주어지면 그 벡터가 속한 영역을 찾고 그 영역에 대한 대표 벡터의 인덱스에 의해 그 입력 데이터 벡터를 표현한다. 다시 말하면

벡터 양자화란 적절한 거리 척도(distance measure)에 의해 이미 저장된 기준 벡터(reference vector) 중에서 가장 잘 일치하는 벡터로 매핑시켜주는 코딩 방식이다^[13]. 따라서 이 벡터 양자화 매핑은 식 (2)와 같이 k 차원 벡터 공간으로 부터 M 이라는 인덱스의 집합으로의 매핑으로 정의될 수 있다. 또한 인덱스 m 으로부터 재생 벡터 \hat{x}_m 을 찾을 수 있다.

$$VQ: x = (x_1, x_2, \dots, x_k) \rightarrow m, \quad (2)$$

여기서 $m \in \{M\}$ 이고, 집합 m 의 크기는 M 이다.

벡터 양자화에서는 먼저 시험 데이터에서 추출한 훈련 벡터들을 이용하여 입력 벡터의 대표 벡터들의 집합인 코드북을 만든다. 코드북을 구성하기 위해서는 모든 훈련 데이터와, 그와 가장 유사한 코드 어(code word) 사이의 평균 왜곡이 최소화되도록 하는 LBG (Lind, Buzo, and Gray) 알고리즘이 많이 이용된다^[14]. 코딩 시에는 입력 벡터와 가장 유사한 코드 어를 코드북에서 찾는다. 또한 전송시 이 코드 어를 전송하는 것이 아니라 코드 어의 인덱스(index)를 전송함으로써 데이터 량을 줄인다.

주어진 성능 기준에 대해 최적인 벡터 양자화 시스템의 설계는 주어진 기준에 대해 최적인 코드 북을 결정하는 것과 같다. 이것은 일반적으로 입력 데이터에 대한 통계적 특성을 알아야만 가능하지만 입력 데이터에 대한 통계적 특성은 보통 알려져 있지 않으므로 코드 북은 일반적으로 훈련(training)이라고 부르는 과정에 의해 구성된다. 따라서 훈련 과정에서는 실제의 데이터와 유사한 통계적 특성을 갖는 데이터를 사용하여만 최적의 코드북을 생성할 수 있다.

훈련 과정에서는 왜곡 척도 $d(x, \hat{x})$ 를 사용하여 입력 데이터가 어느 영역에 속하는지를 결정한다. 왜곡 척도는 x 를 \hat{x} 로 표현하는 비용(cost)으로 볼 수 있다. 훈련 데이터 벡터가 같은 영역에 속하는 지를 결정함으로써 k 차원의 입력 데이터 벡터를 셀들로 분할한다.

움직임 벡터들이 공간적 상관성을 갖는 일련의 영상 프레임 시퀀스의 움직임 벡터의 양자화에 이러한 벡터 양자화를 적용할 경우 훈련에 의해 생성된 코드 북은 대표 움직임 성분(representative motion component)을 나타낸다. 그리고 코드 북 설계시 코드 북의 크기를 적당히 조절하면 움직임 벡터 예측 오차에 의한 손실을 최소화하며 움직임 벡터의 데이터 량을 줄일 수 있다. 즉 코드북의 크기를 줄이면 탐색 영역이

줄어들므로 계산량이 적어지고 압축비를 높일 수 있으나, 움직임 예측 오차가 증가하므로 잉여신호의 전송시 데이터 량이 증가할 뿐 아니라 화질 저하를 초래할 수도 있다. 반면 코드북의 크기를 크게 하면 탐색 영역이 늘어나므로 계산 량이 증가하고 전송해야 할 움직임은 벡터의 정보량이 증가한다. ATM 망 등에서는 망의 상태에 따라 적응적으로 코드 북의 크기를 조절함으로써 성능을 최적의 상태로 유지하는 방법을 연구하고 있다^[15].

2. 신경 회로망을 이용한 움직임 정보의 벡터 양자화 앞 절에서 언급된 바와 같이 코드 북을 생성하기 위한 훈련과 인코딩 과정은 매우 복잡하고 계산량도 많다. 그리고 LBG 알고리즘과 같이 현재 사용되는 대부분의 알고리즘은 배치 모드 알고리즘(batch mode algorithm)이며, 훈련 과정에서는 훈련 데이터를 모두 액세스 해야할 필요가 있다^[15]. 또한 입력 데이터의 통계적 특성이 상황에 따라 변할 수 있는 경우에는 입력이 들어올 때 마다 코드 북 벡터를 변경하는 적응 벡터 양자화 방식을 사용하는 것이 유용하다. 신경 회로망에는 많은 우수한 훈련 알고리즘들이 있어 이러한 적응 벡터 양자화를 적용하는데 효과적이다.

실제로 벡터 양자화 인코딩과 코드 북 설계에 신경 회로망을 적용한 많은 연구가 있었다. Naylor와 Li는 Kohonen의 SOFM(self organizing feature map)을 이용하여 음성 처리 분야에서 벡터 양자화기의 코드 북의 설계에 응용하였다. Nasarabadi와 Feng는 영상 코딩에서 벡터 양자화 코드북을 설계하는데 이를 이용하였다. 또한 이들과 유사한 방법으로 Matsuyama에 의해 음성과 영상의 가변 영역 벡터 양자화에 대한 연구가 이루어 졌다^[16].

비교사(unsupervised) 경쟁학습(competitive learning) 알고리즘을 갖는 신경 회로망으로 벡터 양자화를 수행하면 신경 회로망의 병렬 처리 구조로 인해 실시간 처리가 가능하고 신경 회로망의 다양한 학습 방법에 의해 벡터 양자화 알고리즘이 개선될 수 있는 장점이 있다. 아래에서는 신경 회로망을 이용한 벡터 양자화에 대하여 간단히 설명한다.

가. 인코딩(encoding)

양자화될 벡터를 k 차원 벡터 공간으로 두고, 왜곡 척도 $d(x, y)$ 가 이 공간에서 정의된다고 하자. 코드북의 크기를 M 으로 두고 코드 어(code word)는 c_i ,

$i=1, \dots, M$ 으로 둔다. M 개의 신경 유닛을 갖는 신경 회로망에서 i 번째 코드어 c_i 를 신경 유닛 i 에 대한 가중치 벡터로 둔다. 인코딩될 어떤 벡터 x 가 주어질 때, x 는 M 신경 유닛에 병렬로 입력된다. 각 신경 유닛에서 입력 벡터와 그 가중치 벡터와의 왜곡이 식 (3)에 의해 계산된다.

$$d_i = d(x, c_i), \quad i = 1, \dots, M. \quad (3)$$

입력 벡터는 최소 왜곡을 갖는 신경 유닛(식 (4))의 인덱스 i^* 로 인코딩된다.

$$d_{i^*} = \min_j d_j \quad ; \quad (4)$$

최소 왜곡을 갖는 신경 유닛을 결정하는 과정과 그 인덱스를 부여하는 과정을 제외한 이러한 일련의 계산 과정은 모두 병렬로 수행될 수 있다. 최소 왜곡을 갖는 신경 유닛을 결정하는 데는 여러 가지의 신경 회로망 학습 알고리즘이 사용될 수 있다.

나. 훈련(training)

벡터 양자화를 신경 회로망으로 구성하는 가장 큰 장점은 벡터 양자화를 훈련하는데 CL(competitive learning), KSFM(Kohonen self-organizing feature map) 등 다양한 신경 회로망 학습 알고리즘을 적용할 수 있다는 것이다. 본 논문에서는 CL의 문제점을 개선한 FSCL(frequency sensitive competitive learning) 신경 회로망을 이용하여 움직임 벡터를 벡터 양자화하는 방법을 사용한다.

CL 신경 회로망에서는 먼저 가중치 벡터 w_i 를 랜덤한 값으로 초기화시킨다. 그리고 각 입력 벡터 x 가 주어질 때 마다 식 (5)에서 왜곡 $d(x, y)$ 를 최소로 하는 승리 출력 신경 유닛 z_i 와 승리 가중치 벡터(winning weight vector) w_{i^*} 를 찾는다.

$$z_i = \begin{cases} 1 & \text{if } d(x, w_i) \leq d(x, w_j), \quad j = 1, \dots, M \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

새로운 승리 가중치 벡터 $w_{i^*}(n+1)$ 는 식 (6)과 같이 계산된다.

$$w_{i^*}(n+1) = w_{i^*}(n) + \epsilon(x - w_{i^*}(n))z_i \quad (6)$$

위 식에서 파라미터 ϵ 은 학습 율(learning rate)이며 이는 학습이 진행됨에 따라 서서히 감소되어 영이

된다. 이 알고리즘은 학습 데이터를 반복 사용하여 훈련하며 이러한 학습 알고리즘의 문제점은 사용되지 않는 가중치 벡터가 발생하여 전체 평균 왜곡을 증가시킨다는 것이다. 이 문제를 해결하기 위한 방법으로써 KSFM은 학습 과정에서 승리 가중치 벡터들과 이웃하는 가중치 벡터들 사이에 상호 연관성을 두어 승리 가중치의 값 뿐아니라 이웃한 가중치 벡터들도 같이 그 값을 조정하도록 하였다. 그러나 이 방법은 계산 량이 많고 구조가 복잡하다는 단점이 있다.

FSCL 에서는 학습 과정에서 사용되는 왜곡 척도를 식(7)과 같이 수정하여 이러한 문제를 해결하였다.

$$d^*(x, w_i) = d^*(x, w_i(n)) * u_i(n) \quad (7)$$

여기서 u_i 는 신경 유닛 i 가 학습 과정에서 승리한 총 수를 나타낸다. 이 경우에서도 승리 신경 회로망 유닛은 학습 과정에서 최소 d^* 인 신경 유닛이 된다. 그림 2에 이러한 FSCL의 구조가 나타나 있다.

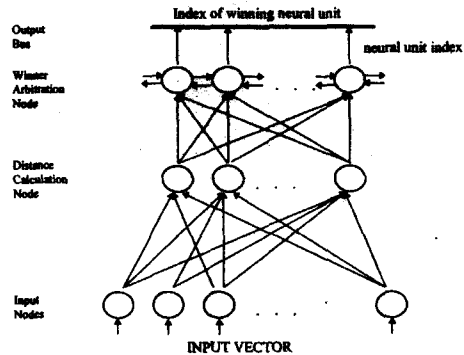


그림 2. FSCL의 구조
Fig. 2. Structure of FSCL.

3. 탐색 영역 예측에 의한 움직임 벡터의 추정
본 논문에서 제안한 신경 회로망 벡터 양자화를 이용한 움직임 벡터 추정 방식의 불력도는 그림 3 과 같다. 먼저 두 프레임의 시험 영상에 대해 전역 탐색 법에 의해 움직임 벡터를 구한다. 실시간으로 계산되지 않아도 되므로 보다 정교한 움직임 벡터 추정 방법을 사용한다면 더 좋은 결과를 얻을 수 있다. 이렇게 구해진 움직임 벡터를 훈련 벡터로 이용하여 신경 회로망 벡터 양자화를 훈련하여 코드 북을 생성한다. 코드 북의 코드 어들은 시험 입력 영상 열의 움직임 벡터들을 대표하게 된다. 영상 열에 대해서 이 코드 북을 탐색 점으로 사용하여 움직임을 추정 하므로 기존의 방

법에 비해 탐색점의 수가 현저히 줄어들고 계산 시간도 줄일 수 있다.

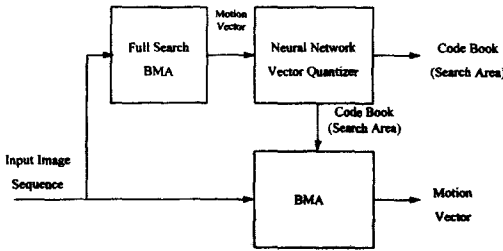
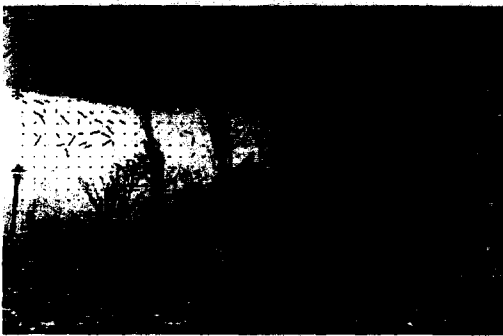
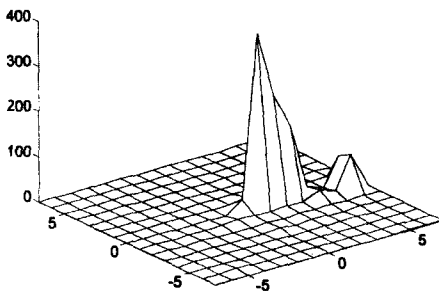


그림 3. 제안된 움직임 벡터 추정 블록도
Fig. 3. Block diagram of suggested motion vector estimation.



(a) 움직임 벡터



(b) 움직임 벡터의 분포

그림 4. 전역 탐색법에 의해 구해진 시험 영상의 움직임 벡터와 그 분포(블록크기=8, 탐색영역=±7)

Fig. 4. Motion vector and its distribution by full search.(block size=8, search area=±7)

그리고 움직임 벡터의 인덱스를 전송하므로 움직임 벡터를 전송하기 위한 정보량도 줄어들게 된다. 예를 들어 ±7의 탐색 영역을 갖는 블록 매칭 알고리즘을

사용할 경우 가능한 움직임 벡터의 수는 225 가지가 되고 이를 전송하려면 고정 길이로 인코딩하는 경우 움직임 벡터당 8비트가 필요하지만 제안한 방법에 의해 코드 북의 크기를 25로 설계한다면 4비트만 필요하게 되므로 압축 비를 높일 수 있다. 또한 탐색 점의 수가 225점에서 25점으로 줄어들므로 계산 량의 측면에서도 크게 성능 개선이 이루어진다.

그림 4(a)에 시험 영상의 초기 2 프레임을 사용하여 전역 탐색 법에 의해 움직임 벡터를 구한 결과가 나타나 있다. 그림 4(b)에 구해진 움직임 벡터들의 분포가 나타나 있고 그림에서, 동일한 영상 열의 움직임 벡터들은 큰 공간적 상관성을 가지고 있음을 알 수 있다. 이들 움직임 벡터들을 훈련 입력 데이터로 하여 앞 절에서 설명된 신경 회로망 벡터 양자화기에서 FSCL 학습 알고리즘에 의해 코드 북을 생성한다. 25 개의 크기를 갖는 코드북의 초기 값과 최종 생성된 코드 북이 그림 5에 나타나 있다.

IV. 모의 실험 결과 및 고찰

모의 실험에서는 'Flower Garden'(그림 4(a) 참조)의 SIF 형태를 시험 영상으로 사용하였다. 이 영상은 CCIR601의 1/2의 크기를 가지며 8비트로 양자화되었다. 블록 매칭 알고리즘에서의 블록 크기는 8 x 8로 하였다. MPEG에서는 탐색 영역을 수직과 수평 방향으로 각각 ±15를 권고하였으나 실험에서 사용한 시험영상이 CCIR601의 1/2의 크기인 SIF 형태이므로 본 실험에서는 탐색 영역을 ±7로 하였다.

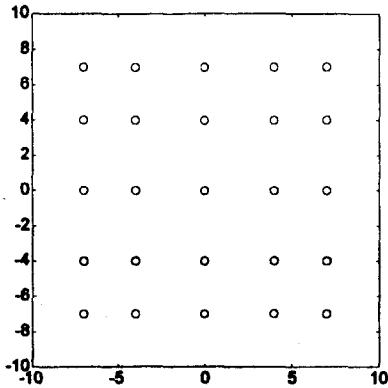
신경 회로망 벡터 양자화기를 이용하여 탐색영역을 예측하기 위해서는 코드 북을 훈련하기 위한 시험 영상이 필요한데 본 실험에서는 초기 두 프레임으로 부터 구해진 움직임 벡터를 사용하였다. 코드 북의 크기는 25로 하였다. 표 1에 나타난 바와 같이 고속 블록 매칭 방식 중 성능이 우수한 것으로 알려진 TSS의 경우 탐색 영역의 크기는 27 점이 되며, 전역 탐색법의 경우는 225점이 된다. 따라서 본 논문에서 제안한 방식은 탐색 점의 수를 줄이므로 계산 량을 감소시킬 뿐 아니라 움직임 벡터를 전송하는 데 필요한 비트 수도 크게 줄인다.

제안된 방식의 경우는 전역 탐색 법에 의한 블록 매칭 방식으로 움직임 벡터를 검출하는 경우와 비교하면 ±2 전역 탐색 블록 매칭 방식에 해당되므로 본 절에서

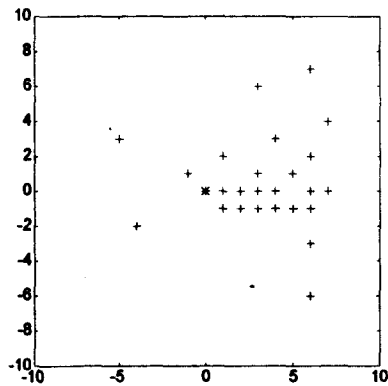
는 이들과도 성능을 비교하였다. 본 모의 실험에서는 객관적인 평가 기준으로 식 (8) 같이 정의되는 PSNR (peak signal to noise ratio)을 이용하였다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad [dB] \quad (8)$$

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |F(i, j) - F'(i, j)|^2 \quad (9)$$



(a) 코드 북의 초기값



(b) 출력 코드 북

그림 5. 코드 북의 초기값과 출력 코드 북
Fig. 5. Initial and output value of codebook.

그림 4에 초기의 두 화면을 훈련 영상으로 이용한 경우 전역 탐색 법에 의한 움직임 벡터와 그 분포를 보이고 있다. 그림 5는 위의 방법으로 구한 훈련 움직임 벡터를 신경 회로망의 입력으로 하여 훈련한 결과 생성된 가중치 벡터를 표시한 것이며 이는 결국 코드 북을 나타낸다.

표 2는 영상의 30 프레임에 대하여 제안된 방법과

전역 탐색 법 그리고 삼단계 탐색 법에 대해 평균 PSNR을 비교한 결과를 보여 준다. 제안된 방법은 전역 탐색법과 비교하여 평균 PSNR 측면에서 거의 차이가 없으나 삼단계 탐색 법에 비하여 1.5dB 이상 우수함을 보여 준다. 계산 량과 움직임 벡터를 전송하기 위한 부가 정보량의 측면에서 유사한 ± 2 전역 탐색 블록 매칭 방식과 비교할 때 1.5dB 이상 우수함을 보여 준다. 그림 6은 실험 영상의 30 프레임에 대한 PSNR 그래프이다.

표 1. 각 방법의 성능 비교
Table 1. Performance comparison.

방 식	탐색점의 수	움직임 벡터당 소요 비트수	평균 PSNR (db)
전역 탐색법 (± 7)	221 점	8 비트	22.6791
± 2 탐색 영역	25 점	4 비트	17.3795
삼단계 탐색법	27 점	10 비트	21.0388
제안된 방법	25 점	4 비트	22.6388

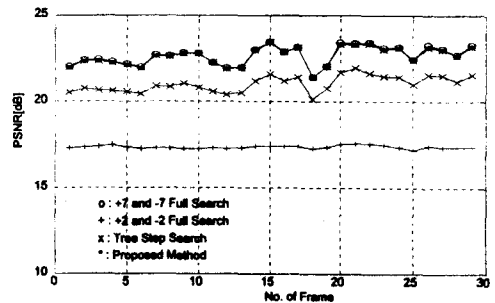


그림 6. PSNR
Fig. 6. PSNR.

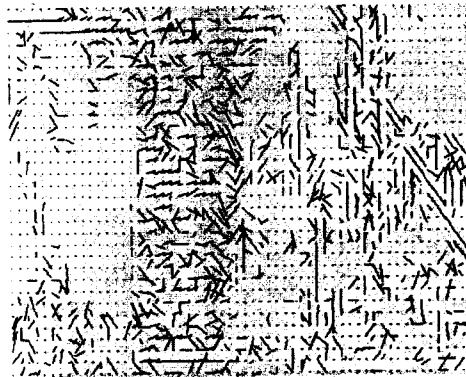
그림 7은 제안된 방식의 움직임 벡터 스무딩 효과를 보여준다. 실험에서는 움직임 벡터의 스무딩 효과를 효과적으로 나타내기 위하여 블록 크기를 4로 하고 탐색 영역은 ± 7 로 하여 각 방식에 의해 구해진 움직임 벡터의 일부를 확대하여 표시하였다. 그림에서 제안된 방식에 의해 구해진 움직임 벡터의 스무딩 효과를 분명하게 관찰할 수 있다. 이러한 스무딩 효과는 움직임 벡터의 양자화 과정에서 발생하는 것으로써 잡음 등에 의한 움직임 벡터를 잘못 추정하는 것을 방지하고 움직임 벡터의 엔트로피 부호화 효율도 높인다.



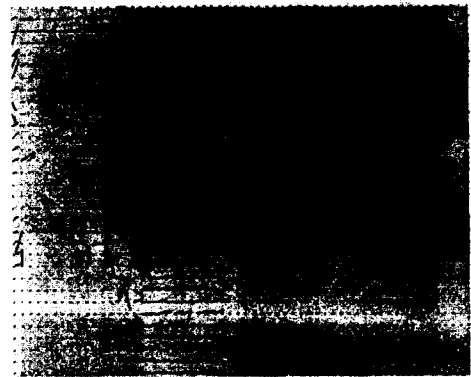
(a) 전역 탐색법



(b) ± 2 탐색 영역



(c) 삼단계 탐색법



(d) 제안된 방법

그림 7. 프레임 16의 움직임 벡터

Fig. 7. Motion vectors of frame 16.

V. 결 론

움직임 검출 기법은 동영상 전송 시스템에서 매우 중요한 역할을 하는 부분이다. 움직임 벡터를 얼마나 정확히 찾아내느냐에 따라 시스템의 성능이 좌우될 수 있으며 부분이며 지금까지 많은 연구가 진행되어 왔고 많은 움직임 검출 기법들이 제안되었다. 여러 가지 제안된 기법 중에서 삼단계 탐색 법이 하드웨어의 구현이 상대적으로 쉽고 계산 량이 적으며 성능도 비교적 우수하다. 그러나 보다 정확한 움직임 벡터를 검출하기 위해서는 탐색 영역내의 모든 점을 탐색하는 전역 탐색 법을 사용하여야 하며 이는 계산 량이 많으며 하드웨어 구현이 복잡해진다.

본 논문에서는 초기의 영상으로 전역 탐색 블록 매칭 알고리즘을 이용하여 움직임 벡터를 구하고, 이를

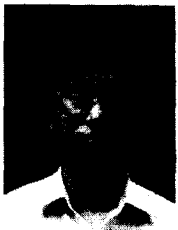
훈련 움직임 벡터로 FSCL신경 회로망에 적용하여 코드 북을 설계하였으며 이 코드 북을 움직임 예측 영역으로 이용하였다. 이러한 방법은 영상에서 움직임 벡터의 공간적 상관성을 이용한 것으로써 탐색 정의 수 즉 계산 량을 줄이고, 움직임 벡터 전송을 위한 소요 비트를 줄임으로써 압축 율을 높인다. 모의 실험 결과 제안된 방법은 PNSR 측면에서 삼단계 탐색 법에 의한 것보다 평균 1.5dB이상의 향상되는 결과를 보였다. 또한 제안된 방법은 움직임 벡터를 벡터 양자화하는 과정에서 움직임 벡터를 스무딩하는 것과 같은 효과를 얻기 때문에 잡음에도 강한 특성을 보였다.

참 고 문 헌

[1] A.K. Jain, "Image data compression: A

- review." *Proc. IEEE*, vol. 69, no. 3, pp. 384-389, Mar. 1981.
- [2] A.N. Netravali and J.O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, no. 3, pp. 366-406, Mar. 1980.
- [3] A.N. Netravali & J.D. Robbins, "Motion compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, no. 3, pp. 631-670, Mar. 1979.
- [4] ISO-IEC/JTC1/SC29/WG11 MPEG 92/086, "Preliminary working draft of test model 0", Mar. 1992.
- [5] ISO-IEC/JTC1/SC29/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] 김기현, 김진태, 장태규, 최중수, "효율적인 부호화와 예측오차를 고려한 움직임 벡터 스무딩 기법에 관한 연구," 제5회 신호처리 합동 학술대회 논문집, pp. 719-723, 1992
- [8] Y.Y. Lee and J.W. Woods, "Motion Vector Quantization for Video Coding," *IEEE Trans. Image Processing*, vol. 4, no. 3, Mar., 1995.
- [9] H. Kanakia, P. Mishra, and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport," in *ACM sigcomm'93 Conf. Proc. : Commun. Architectures, Protocols, Applications*, San Francisco, Sep. 13-17, pp. 20-31, 1993.
- [10] F. Giorda & A. Racciu, "Bandwidth reduction of video signal via shift vector transmission," *IEEE Trans. Commun.*, vol. COM 23, no. 3, pp. 1002-1003, Mar. 1977.
- [11] T. Koga et al., "Motion compensated interframe coding for video conferencing," *Nat. Telecom. Conf.*, Nov. 29-Dec. 3, 1981.
- [12] 이인홍, 박래홍, "이동벡터 추정을 위한 고속 Block Matching Algorithm에 관한 연구", *전자공학회지*, 제25권, 제2호, pp. 93-101, 1988
- [13] S. Ullman, "Analysis of Visual Motion by Biological and Computer Systems," *IEEE Computer*, vol. 14, no. 8, pp. 57-69, 1981.
- [14] M. Bierling, "Displacement Estimation by Hierarchical Block Matching," *Proc. of SPIE : Visual Communications and Image Processing*, 1001, pp. 942-951, 1988.
- [15] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1991.
- [16] S.C. Ahalt, A.K. Krishnamurthy, P. Chen and D.E. Melton, "Competitive learning algorithms for vector quantization," *Neural Networks*, vol. 3, no. 3, pp. 277-290, 1990.

— 저 자 소 개 —



柳大鉉(正會員)

1983년 2월 부산대학교 전기기
계공학과 졸업(공학사). 1985년
2월 부산대학교 전자공학과 졸
업(공학석사). 현재 부산대학교
전자공학과 박사과정 재학 중.
1987년 3월 ~ 현재 전자통신

연구소 재직 중. 주관심분야는 신경 회로망, 영상 처
리 등임

金在昌(正會員) 第 28卷 B編 4號 參照.

현재 부산대학교 전자공학과 교수