

개선된 SOM 을 이용한 칼라 이미지 벡터 양자화

남미영*, 이종희**, 김광백**

*신라대학교 전자계산학과

**신라대학교 컴퓨터정보공학부

e-mail : bk4004@orgio.net

Color Image Vector Quantization using Enhanced SOM

Mi-Young Nam*, Jong-Hee Lee**, and Beak-Kwang Kim**

*Dept. of Computer Science, Silla Univ.

**Division of Computer Information and Engineering, Silla Univ.

요 약

정보 산업의 발달에 의해 여러 가지 형태의 정보 전달이 가능하게 되었으며 최근에 칼라 화상의 정보 전달에 있어서 빠른 전송과 압축기법이 필요하게 되었다. 본 논문에서는 칼라 화상을 압축하는데 있어 개선된 SOM 알고리즘을 이용하여 칼라 이미지에 대한 벡터 양자화 기법을 제안한다. 제안된 방법은 기존의 LBG 알고리즘을 이용한 벡터 양자화 기법에 비해 블록화 현상을 줄일 수 있었으며 이미지 전체에 대해 블록의 수만큼 계속해서 반복하지 않고 동적으로 코드북을 생성시킴으로써 실행 시간도 줄일 수 있었다. 또한 웨이블릿을 칼라 화상에 적용시켜 화상의 특징을 더욱더 두드러지게 함으로써 개선된 SOM 을 적용시 재생의 효과를 높일 수 있었다.

1. 서론

정보 산업의 발달에 의한 멀티미디어, 화상회의, 종합정보통신망 서비스(ISDN service) 등의 발달로 여러 가지 형태의 정보 전달이 가능하게 되었고, 최근에 칼라 화상은 정보 전달의 빠른 전송과 압축 기법이 절실히 필요하게 되었다. 칼라 화상은 2 개의 주요소로 이루어져 있다. 신경망에 적용하기 위해서는 화상의 각 픽셀을 red(R), green(G), blue(B) 3 개의 요소를 각각 계수화(digitize)하여 사용한다.

각 칼라 요소는 8 비트를 사용하여 한 화소를 표현하는데 24 비트가 필요하다. 24 비트로 표현될 수 있는 팔레트(palette)의 가지 수는 2^{24} 칼라이지만 4096×4096 화소보다 작은 화상에서는 실제적으로 그 모든 칼라가 사용되지 않는다. 따라서 실제 화상의 화소를 압축하는 것이 가능해지며, 저장 장치 영역과 전송매체 대역폭의 제한에 의해서도 압축이 필요하게 된다.

자료 압축은 화상의 질을 어느 정도 유지하면서 전송이나 저장 장치 영역에 대한 비트율을 감소시키는 것이 목표이다. 이를 위해 DPCM, Transform Coding, Hibrid Coding 등과 같은 대역폭 압축 기법이 연구되어 왔다[1]. 이 기법들에서는 압축이 화상 화소의 개별적인 값들 사이에서 진행된다. 그러나 연속되는 화소들

은 어느 정도 중속되어 있고 상관 관계가 있기 때문에 이 방법들은 최적이 되지 못한다. 그리고 shannon의 이론에 따르면 이론적으로 스칼라 코딩보다는 벡터 코딩에 의해서 더 좋은 성능을 얻을 수 있다고 알려져 있다[2].

이에 따라 벡터 양자화에 대한 연구가 활발히 진행되어 왔다. 벡터 양자화를 사용한 대부분의 방법들은 린드(Linde), 부조(Buzo), 그레이(Gray)에 의해 개발된 LBG 알고리즘을 사용하고 있다[3]. 그러나 LBG 알고리즘은 전체 화상을 여러 번 반복해 각 단계에서 코드 벡터들을 최적의 위치로 옮기는 방법으로 알고리즘의 계산 복잡도로 인해서 많은 시간이 요구된다.

인공 신경망을 이용한 방법은 적용되는 데이터로부터 적응적으로 학습되기 때문에 시간의 흐름에 따라 원 데이터의 통계 분포가 달라지는 응용에 적합하다 [4, 5, 6]. 그리고 신경망 구조는 거대한 병렬구조를 가지기 때문에 고속의 처리기에 적합하고, 이 신경망 구조를 H/W로 구현시 색인에 의한 코드북의 탐색이 O(1)을 제공하게 때문에 특별히 코드북의 설계를 위해 신경 쓸 필요가 없게 된다. 그리고 칼라 화상에 대한 벡터 양자화는 코드북 결정을 위해 화상에서 화소들의 분석이 필요하고, 그 칼라 이미지에 대한 코드북은 미리 알려져 있지 않기 때문에 자율학습 방법이

적합하다.

2. 본론

경쟁 학습 신경망은 주어진 입력 패턴들에 대해서 유사성을 가지는 특징들을 클러스터링하여 분류하는 방법으로 하나의 입력 패턴에 대한 대표 클래스를 나타내는 승자 노드를 선택하여 그 승자 노드에 입력 패턴의 정보를 저장하는 방식이다

입력 패턴과 저장 패턴 차이의 정도를 저장 패턴에 반영하여 저장 패턴이 입력 패턴에 유사하도록 조정된다. 따라서 여러 개의 입력 패턴이 하나의 클래스에 해당될 경우 그 저장 패턴은 자신에게 포함되는 모든 입력 패턴의 정보를 포함하면서 적응적으로 조정된다. 이러한 저장 패턴인 가중치를 조정해 나가는 과정이 벡터 양자화에서는 제시되는 입력 패턴들에 대해서 코드북을 생성해 나가는 과정에 해당된다.

2.1 개선된 SOM 을 이용한 칼라 이미지 벡터 양자화

제안된 칼라 이미지 양자화 알고리즘은 개선된 SOM 을 이용함으로써 칼라 이미지에 대해서 벡터 양자화가 가능하도록 하였으며, 실시간 처리로 코드북을 생성하며 LBG 보다 향상된 재생 능력을 내는 경쟁 학습 알고리즘이다.

기본적으로 경쟁 학습 구조를 가지고 빈도수를 이용한 왜곡 척도와 가중치 수정 방법을 도입한 경쟁 학습 알고리즘을 기반으로 승자 노드의 대표 벡터와 입력 벡터간의 오류값에 비례하도록 가중치 조정을 하고 블록(block) 효과를 개선하기 위해 초기 코드북 생성시 입력 벡터의 인접한 화소에도 승자 노드의 영향을 받도록 한다.

학습은 두 단계로 구성되고 각 단계에서 전체 학습 벡터들을 한번씩만 적용한다. 첫번째 단계에서는 주어진 학습 벡터들의 분포를 잘 나타내 주도록 초기 코드북을 생성한 후에 두 번째 단계에서는 초기 코드북을 이용하여 현재의 각 결정 영역 내의 중심으로 옮겨 다시 코드북을 생성한다.

2.2 개선된 SOM 경쟁 학습 알고리즘

일반적으로 경쟁 학습 신경망에서 가중치 조정에 사용되는 식은 다음과 같다.

$$W_{ij} = W_{ij} + \alpha(x_i(W_{ij})) \quad \text{식(1)}$$

α 는 학습률로 0 과 1 사이의 값으로 임의로 설정한다. 여기서 α 는 오류 값으로 입력 벡터와 코드 벡터간의 차이를 의미한다. 위식에 의하면 코드북 생성을 위한 가중치가 입력 벡터와 코드 벡터간 차이만큼 조정되어 코드 벡터에 입력 벡터의 정보를 반영하게 된다. 여기에 그 정보를 반영할 때 입력 벡터와 코드

터간에 차이가 많이 나면 코드 벡터에 입력 벡터를 기억시키기 위해 입력 벡터의 정보를 더 많이 반영해야 한다.

위 문제를 해결하기 위해 승자노드에서 오류 값을 0 과 1 사이의 값으로 정규화된 출력 값인 출력 오류를 학습률로 사용한다. 제시된 출력 오류에 의한 가중치 조정으로 오류의 크기에 따라서 그 비율 만큼의 조정이 가능하다. 그리고 코드 벡터의 저활용 문제를 방지하기 위해서 학습 중에 각 코드 벡터들이 수정되어지고 선택되어지는 횟수가 가능한 한 동등해지도록 각 노드가 승자가 된 빈도수를 가중치 조정에 사용한다. 즉 학습이 끝날 때 각각의 노드들이 승자 노드로 선정된 빈도수가 유사하도록 하자는 것이다. 이를 위해서는 코드 벡터들과 학습 벡터간의 실제적인 왜곡 정도도 충분히 고려 되어야 한다. 따라서 경쟁 학습 알고리즘에서 승자노드를 선정하기 위한 왜곡의 척도를 승자노드로 선정된 빈도수와 유클리디안 거리의 곱으로 수정한다. 제시된 방법을 바탕으로 한 가중치 조정에 대한 알고리즘은 다음과 같다.

$$\begin{aligned} \text{Weight}[i][\text{winner}] &+= \\ &\quad \text{Alpha} * (\text{double})\text{Input}[i] * \text{Dwgt}[i][\text{winner}]; \\ \text{Alpha} &= (\text{Now_err}[\text{win}] / \text{COL_NUM}) + ((\text{double})1.0 \\ &\quad / ((\text{double})\text{Frequency}[\text{win}])); \end{aligned}$$

여기서 $(\text{double})1.0 / (\text{double})\text{Frequency}[\text{win}]$)는 i 번째 노드가 승자가 된 빈도수를 0 과 1 사이의 값으로 바꿔주는 정규화시켜 주는 루틴이고, $\text{Now_err}[\text{win}]$ 은 현재 노드의 오류값, $\text{Frequency}[\text{win}]$ 는 i 번째 노드가 승자가 된 빈도수를 의미한다.

위의 오류값을 이용한 방법은 현재의 가중치 변화량만 고려하고 그 전의 변화량은 고려되지 않는다. 따라서 비록 적응적 가중치 조정이 되지만 현재의 변화량에 의해 가중치의 값이 많이 좌우된다. 이러한 문제점을 해결하기 위해 가중치 조정시 현재의 가중치 변화량뿐만 아니라 그 이전의 가중치 변화량에 비례하여 가중치를 조정하도록 한다. 즉 오류 역전과 알고리즘에서 정의하는 모멘텀 개념을 적용한 것으로 이에 따른 가중치 조정식 알고리즘은 다음과 같다.

$$\begin{aligned} \text{Weight}[i][\text{winner}] &+= \text{Dwgt}[i][\text{winner}]; \\ \text{Dwgt}[i][\text{winner}] &= (\text{Alpha} * ((\text{double})\text{Input}[i] - \\ &\quad \text{Weight}[i][\text{winner}])) + \text{Alpha} * \text{Dwgt}[i][\text{winner}]; \\ \text{Weight}[i][\text{winner}] &+= \text{Dwgt}[i][\text{winner}]; \end{aligned}$$

위의 알고리즘에서 첫 번째 항은 현재의 가중치 변화량의 의미를 나타내고 두 번째는 하이 모멘텀을 적

용한 것으로 이전의 변화량을 의미하는 것이다.

제안된 SOM 학습 알고리즘에 대해서 재생률을 높이기 위해 웨이블릿을 적용한다. 웨이블릿의 주파수를 적용시킨 후 각각의 RGB 값을 분리하여 제안된 학습 알고리즘을 적용하여 칼라 이미지를 벡터 양자화한다. 웨이블릿의 저주파수를 이용하기 때문에 이미지의 특징들이 더 잘 나타남으로 재생시 재생률이 좋아진다. 웨이블릿을 적용시킨[7] 값으로 벡터를 구성하고 이 벡터를 제안된 SOM 학습 알고리즘에 적용시킨다. 승자 노드에 해당하는 인덱스가 구해지면 그 인덱스에 해당하는 웨이블릿을 적용한 벡터의 값이 아닌 원래 영상의 값을 코드북에 저장하고 재생한다.

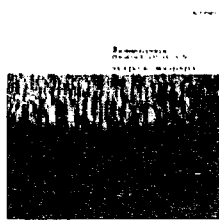
3. 실험 및 결과

실험 환경은 IBM 586 펜티엄 III 에서 C++ Builder 로 구현하였으며, 실험에서 사용한 이미지는 128 x 128 크기의 칼라 비트맵 이미지이다. 16byte 의 크기의 벡터로 코드북을 생성하기 위해 영상을 4 x 4 크기의 블록으로 나누어 총 1024 개의 블록에 대한 코드북을 생성하여 성능을 비교하였다.

LBG 알고리즘은 전체 화상을 여러번 반복해서 각 단계에서 코드벡터를 최적의 위치로 옮기며 코드북을 생성한다. 그러므로 코드북을 생성시키는데 시간이 오래 걸렸으며 재생시 영상의 블록화 현상도 나타났다. 다양한 칼라가 사용되는 칼라 이미지에는 적합하지 않았다. 그러나 개선된 SOM 을 이용하여 코드북을 생성시키고 영상을 압축했을 경우 속도가 4 배나 향상되었으며 재생시 영상의 블록화 현상도 현저하게 줄어들었다. 그리고 웨이블릿을 적용함으로써 재생률을 높일 수가 있었다.



a) Image1 : 초정장 그림



b) Image2 : 앨범 표지



c) Image3 : 암세포 이미지

그림 1. 사용한 영상 이미지의 예
위의 영상과 같은 경우 LBG 알고리즘을 이용한 경우 압축률은 아주 좋았으나 재생률은 다른 방법에 비해 좋지 않았다. 따라서 기존의 SOM 알고리즘을 이용한 경우와 제안된 SOM 을 이용한 경우의 영상의 압축 정도, 코드북의 크기 정도 등을 비교하여 표 1 에 나타내었다.

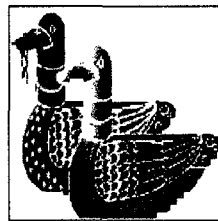
표 1. 벡터 양자화된 코드 파일의 크기

단위 : byte

방법	SOM 알고리즘	개선된 SOM 알고리즘	Wavelet 과 개선된 SOM 알고리즘
Image1	10736	10656	12064
Image2	16272	11224	16528
Image3	17360	17216	17376

* 원래 이미지의 크기 : 128 * 128 * 3 = 49152byte

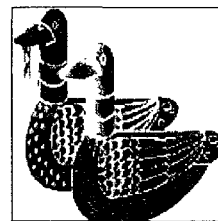
Image1 에 대한 각 실험 방법에 따른 재생 이미지는 아래와 같다.



a) SOM



b) 개선된 SOM



c) 웨이블릿과 개선된 SOM

그림 2. 재생된 이미지

일반적인 SOM 알고리즘을 이용하였을 때 보다 개선된 SOM 알고리즘을 사용하여 코드북을 동적으로 생성하였을 경우 압축률이 향상되었으며 이미지를 재생했을 때의 재생률도 좋았다. 그리고 여기에 웨이블릿을 적용하였을 경우 압축률은 개선된 SOM 만 사용했을때보다 조금 낮지만 재생률은 좋았다. 웨이블릿의 적용으로 이미지의 특징들이 좀 더 잘 구분되어졌기 때문이다. 따라서 이 값을 이용하여 승자 노드를 정하고 코드북을 생성하였으므로 압축률은 조금 떨어지나 재생률은 향상되었다.

4. 결론 및 향후 연구과제

최근에 근접한 코드북을 생성해주는 LBG 알고리즘은 전체 화상을 반복 적용해서 코드북을 생성하기 때문에 큰 화상일 경우에는 많은 시간이 요구된다. 이에 비례하여 제안된 방법은 전체 화상을 두 번 적용하여 코드북을 생성함으로써 실시간 응용에 적합하다.

그리고 칼라 이미지에 대해 벡터 양자화를 하는데 있어서 사전에 코드북을 먼저 생성한다는 것은 어려운 일이다. 따라서 벡터 양자화하기 위한 코드북을 동적으로 생성시켰으며, 개선된 SOM 학습 알고리즘을 적용함으로써 기존의 SOM 학습 알고리즘에 비해 압축률과 재생률을 높일 수 있었다. 그리고 여기에 웨이블릿을 적용한 결과 재생률이 향상되었다.

앞으로의 연구 방향은 다양한 화상에 적용하여 다른 방법들과의 비교가 필요할 것이고, 입력 벡터와 코드북의 차이를 더 잘 반영해 줄 수 있는 퍼지 기법을 이용한 함수에 관한 연구가 필요하다.

참고문헌

- [1] M.Rabbani and P.W.Jones, "Digital Image Compression Technique," Spie Optical Engineering Press, PP.144-169, 1991.
- [2] R.M.Gray, "Vector Quantization," IEEE ASSP Magazine, Apr., pp.4-29, 1984.
- [3] Y.Linde, A.Buzo, and R.M.Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. On Communications, Vol. Com-28, No-1, pp.84-95, Jan., 1980.
- [4] K.R.L.Godfrey and Y.Attaliuzel, "Self-Organized Color Image Quantization for Color Image Data Compression," Proceedings of IJCNN, Vol.3, pp.162-1626, 1993.
- [5] Bruce E. Watkins and Murali Tummala, "Classification Vector Quantization of Image Data using Competitive Learning," Proceedings of ICIP, 1984.
- [6] Liza Jo and M.R.Kaimal, "A Self Organizing Neural Network for Second Generation Image Coding," Proceedings of TAINN, pp.130-137,1999.
- [7] 김진아, 정성환, "Wavelet 기반의 질감 특성을 이용한 영상 검색," 한국정보과학회 가을학술발표논문집, Vol.24.No2,pp.379-382,1997.