

수정된 K-means 알고리즘

Modified K-means Algorithm

조 제 황*

(Che-Hwang Cho*)

* 동신대학교 전기전자공학과

(접수일자: 2000년 8월 11일; 채택일자: 2000년 10월 11일)

기존의 방법보다 우수한 성능의 코드북을 설계할 수 있는 방법을 제시한다. 기존 방법에서는 학습 반복에 의하여 얻어지는 새로운 벡터가 분할된 영역의 중심벡터가 되지만, 제안된 방법에서는 학습 반복의 초기에는 새로운 벡터와 전 벡터간의 거리에 의하여 조절되는 벡터이다. 실험 결과는 제안된 방법에 의하여 얻어지는 코드벡터가 국부적으로 기존의 방법보다 더 최적인 코드북을 구성할 수 있음을 보인다.

핵심용어: 코드북, 코드벡터, 알고리즘

투고분야: 음향 통신기술 분야 (6.1)

We provide an useful method to design codebooks with better performance than conventional methods. In the proposed method, new codevectors obtained from learning iterations are not the centroid vectors which are the representatives of partitions, but the vectors manipulated by the distance between new codevectors and old codevectors in the early stages of learning iteration. Experimental results show that the codevectors obtained by the proposed method converge to a locally better optimal codebook.

Key words: Codebook, Codevector, K-means algorithm

Subject classification: Acoustic communication (6.1)

I. 서 론

영상에서 얻어지는 n 차원의 벡터는 n 차원의 벡터 공간에서 군집을 구성할 수 있다. 그러한 군집의 대표벡터는 코드북의 코드벡터와 동일한 성질을 갖는다. 즉 코드북을 설계하는 것은 군집의 대표벡터를 찾는 것으로 볼 수 있다. 코드북을 설계하는 전형적인 방법은 K-means 알고리즘으로써 LBG (Linde, Buzo, and Gray) 알고리즘으로도 알려져 있다[1]. 그러나 K-means 알고리즘은 국부적으로 최적인 코드북으로 수렴하고, 초기벡터에 의해 성능이 크게 좌우되는 단점을 갖고 있다. 이러한 문제를 완화하는 성능을 높이기 위한 다양한 초기 코드북 결정 방법이 발표되었다[2,3,4]. 초기 코드북 결정 방법 중 splitting 방법이 학습 시간이 많이 소요되는 단점이 있지만 다른 방법에 비하여 보다 우수한 성능의 코드북 설계를 보장한다. K-means 알고리즘은 두 가지 조건을 만족하는데, 하나는 학습벡터와 코드벡터 간의 유클리드 거리가 최소일 때 그 학습벡터는 고려된 코드벡터에 대응되는 분할에 속한다는 최단거리 이웃조건(Nearest Neighbor Condition)과, 새로운 코드벡터는 분할을 구성하는 학습벡터의 중심벡

터라는 중심조건(Centroid Condition)이다. 참고문헌 [5]에서 제안된 방법은 중심조건을 제외하고 기존의 K-means 알고리즘과 거의 같지만 보다 우수한 성능의 코드북을 설계할 수 있음을 보여준다. 제안된 방법으로 얻어지는 새로운 벡터는 중심벡터와 전 벡터간의 거리에 1.8배되는 가중치로 연장선상에서의 벡터로 구해지며, 기존의 K-means 알고리즘에서는 가중치가 1 임을 의미한다. 이러한 가중치는 모든 학습 반복에서 고정되는데, 이러한 점은 보다 우수한 성능의 코드북 설계에 적절하지 않다. 왜냐하면 학습 반복이 진행됨에 따라 새로운 벡터와 중심벡터 간의 거리는 점점 짧아지고, 따라서 가중치가 최종 수렴하는 코드북에 미치는 영향이 감소되기 때문이다. 본 논문에서는 기존의 K-means 알고리즘과 참고문헌 [5]에서 제안된 방법에 의하여 설계된 코드북과 본 연구에서 제안된 방법에 의한 코드북의 성능을 각각 비교한다.

II. 본 론

그림 1은 k 번째와 $k+1$ 번째 학습 반복에서 고려되는 i 번째 벡터를 나타내며, 이 벡터간에는 다음 방정식이 만족된다.

$$y_{k+1}^i = y_k^i + \delta (m_{k+1}^i - y_k^i) \quad (1)$$

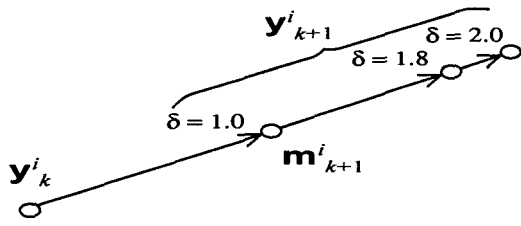


그림 1. 학습반복에 대한 코드벡터간의 관계
Fig. 1. The relationship between codevectors for learning iterations.

여기서 y_{k+1}^i 와 y_k^i 는 각각 k 번째와 $k+1$ 번째 학습 반복에서 고려되는 i 번째 코드벡터이고, m_{k+1}^i 는 코드벡터에 대응되는 분할의 중심벡터를 의미한다. 가중치 δ 는 기존 K-means 알고리즘에서 1이며, Jancy 방법[6]에서는 2이고, 참고문헌 [5]에서는 1.8이다. 이 방법들 중 참고문헌 [5]에서 제안한 방법에 의해 설계된 코드북의 성능이 가장 우수함이 알려졌다. 그러나 지금까지 발표된 방법 모두 δ 는 모든 학습 반복에서 고정된다. 학습 반복이 후반부로 갈수록 전 코드벡터와 새로운 코드벡터 간의 거리는 학습 반복의 초기 단계에 비해 월등히 작은 거리를 갖게되므로 새로운 코드벡터의 결정을 위한 δ 의 영향은 줄어든다. 그러나 학습 반복의 초기 단계에서 그 거리가 크기 때문에 δ 값이 새로운 코드벡터의 결정에 중요한 요소가 된다. 따라서 모든 학습 반복에서 특히 초기 단계에서 δ 를 고정시키는 것은 적절하지 않다.

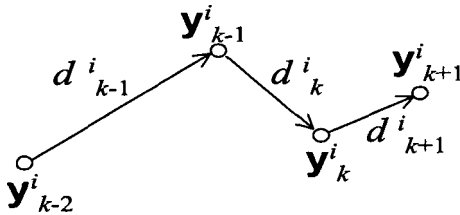


그림 2. 학습반복에 대한 임의의 코드벡터 궤적
Fig. 2. One kind of trajectory of a codevector for learning iterations.

그림 2는 $k-2$ 번째에서 $k+1$ 번째 학습 반복 시 i 번째 코드벡터 y^i 의 움직임을 나타낸다. d_k^i 는 $k-1$ 번째와 k 번째 학습 반복에서 i 번째 코드벡터 간의 거리로써 다음과 같이 주어진다.

$$d_k^i = |y_k^i - y_{k-1}^i| \quad (2)$$

그림 2에 주어진 거리 d_{k-1}^i , d_k^i , d_{k+1}^i 간에 만족되는 부등식은 아래와 같이 총 6가지가 있다.

$$d_{k+1}^i < d_k^i < d_{k-1}^i \quad (3)$$

$$d_{k-1}^i < d_k^i < d_{k+1}^i \quad (4)$$

$$d_k^i < d_{k+1}^i < d_{k-1}^i, d_k^i < d_{k-1}^i < d_{k+1}^i \quad (5)$$

$$d_{k-1}^i < d_{k+1}^i < d_k^i, d_{k+1}^i < d_{k-1}^i < d_k^i \quad (6)$$

식 (3)은 i 번째 코드벡터가 수렴함을 의미하는 것으로 학습 반복의 후반부에서는 대부분의 코드벡터가 식 (3)을 만족한다. 그러나 코드벡터가 학습 반복의 초반부에 식 (3)을 만족하면 지역적으로 보다 더 최적인 코드북의 구성 기회를 놓칠 수 있다. 식 (4)는 i 번째 코드벡터가 발산함을 의미한다. 학습 반복이 거듭됨에 따라 식 (4)를 만족하는 코드벡터의 수는 감소한다. 초기 코드북은 국부적으로 최적인 최종 코드북에 대한 정보 없이 선택되기 때문에 학습 반복의 초기에 대부분의 코드벡터는 식 (5)와 식 (6)을 만족한다. 본 논문에서 제안되는 방법은 학습 반복의 초기에 식 (3)이나 식 (4)를 만족하는 코드벡터를 식 (5)나 식 (6)을 만족하도록 변경하는 것이다. 이러한 조건을 만족시키기 위해서는 아래에 주어지는 식을 만족해야 한다.

$$d_{k+1}^i = \frac{d_{k-1}^i + d_k^i}{2} \quad (7)$$

식 (1)과 식 (7)에 의해 아래에 주어진 것과 같이 δ 를 유도할 수 있다.

$$\delta = \frac{d_{k-1}^i + d_k^i}{2 |m_{k+1}^i - y_k^i|} \quad (8)$$

즉 코드벡터 y_{k+1}^i 가 학습 반복의 초반에 식 (3)이나 식 (4)를 만족하면 식 (8)에 의해 구해지는 δ 를 식 (1)에 적용함으로써 코드벡터 y_{k+1}^i 가 식 (5)나 식 (6)을 만족하게 된다.

III. 실험 및 고찰

본 논문에서는 256개 코드벡터로 구성되는 초기 코드북을 얻기 위해 splitting 방법을 사용한다. 학습벡터는 16개 성분으로 구성되며, 512×512 화소와 256 그레이레벨의 20개 흑백 영상을 사용하여 얻어진다. 설계된 코드북의 성능을 시험하기 위해 512×512 화소와 256 그레이레벨의 흑백 영상 Lena, Peppers, 그리고 Zeida를 사용한다. 학습 반복의 수는 모든 시험에서 30회로 제한한다. 설계된 코드북을 사용하여 재구성된 영상의 화질 정도는 아래에 주어지는 PSNR(Peak Signal to Noise Ratio)을 사용하여 비교된다.

$$\text{PSNR} = 20 \log_{10} \frac{255}{\sqrt{\frac{1}{512^2} \sum_{i=1}^{512} \sum_{j=1}^{512} (f_{ij} - g_{ij})^2}} \quad (9)$$

여기서 f_j 와 g_j 는 각각 원영상과 복원된 영상의 i 번째 열과 j 번째 행의 화소값이다.

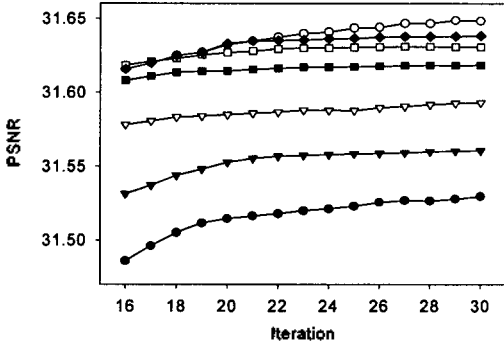


그림 3. 가변 δ 에 대해 학습반복에 대한 PSNR
Fig. 3. PSNR versus leaning iterations for a variable δ .

그림 3은 16번째 학습반복 후의 코드북에 대해 Lena 영상을 복원했을 때의 PSNR을 나타낸다. 기호 ●와 ○는 모든 학습반복에서 각각 $\delta=1.0$ 와 $\delta=1.8$ 로 고정시킬 때의 결과를 나타낸다. 기호 ▼, ▽, ■, □, ◆은 순서대로 2, 6, 10, 14, 18번째까지는 $\delta=1.8$ 로 하고, 이후의 학습반복에서는 $\delta=1.0$ 로 한 경우의 결과를 나타낸다. 이 결과는 δ 가 학습반복의 초기 단계에서는 설계되는 코드북 성능에 큰 영향을 미치지만 후반부로 갈수록 그 영향은 감소한다는 점이다. 본 논문에서는 학습반복의 초반부로 볼 수 있는 10회까지 제안된 방법을 적용하고, 10회 이후는 $\delta=1.8$ 를 적용하여 학습한다. 그림 4, 5, 6은 각각 Lena, Peppers, Zelda 영상에 대한 실험결과를 나타낸다. 그림 4(a), 5(a), 6(a)는 수렴을 의미하는 식 (4)를 만족하는 코드벡터의 수를 나타내고, 그림 4(b), 5(b), 6(b)은 발산을 의미하는 식 (5)를 만족하는 코드벡터의 수를 나타낸다. 그림 4(c), 5(c), 6(c)는 각 시험영상에 대한 PSNR을 나타내며, 제안된 방법이 다른 방법보다 우수함을 나타낸다.

IV. 결 론

기존의 방법보다 우수한 성능의 코드북을 설계하는 새로운 방법이 제안된다. 학습반복에서 새로운 코드벡터는 전 코드벡터와 중심벡터의 연장선상에서 결정되며, 학습반복의 초기단계에서 가중치 δ 의 영향이 크기 때문에 δ 는 기존의 방법에서와 같이 학습 시 고정되는 것이 아니라 가변된다.

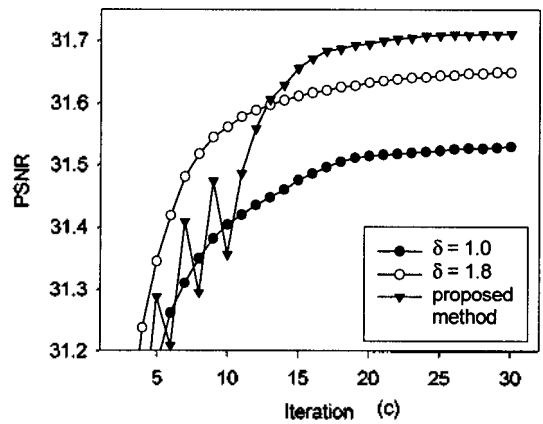
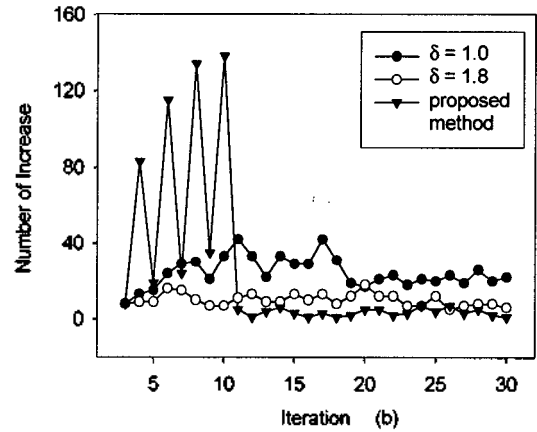
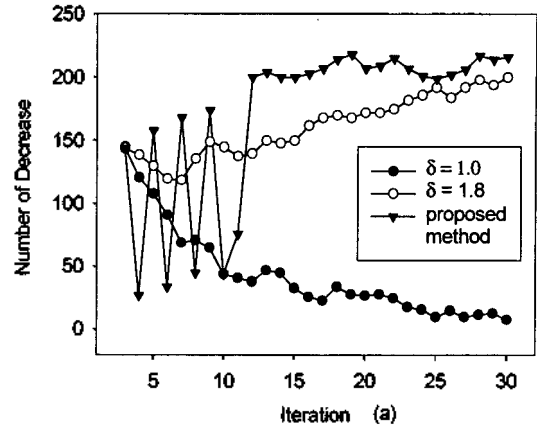


그림 4. Lena 영상에 대해

- (a) 수렴하는 코드벡터의 수
- (b) 발산하는 코드벡터의 수
- (c) 학습반복에 대한 PSNR

Fig. 4. (a) The number of convergent codevectors, (b) The number of divergent codevectors, (c) PSNR versus leaning iterations for Lena image.

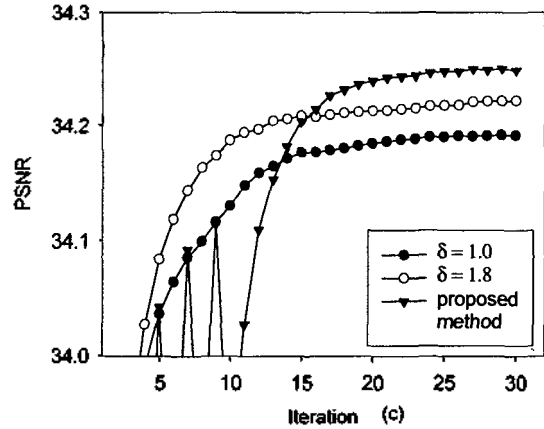
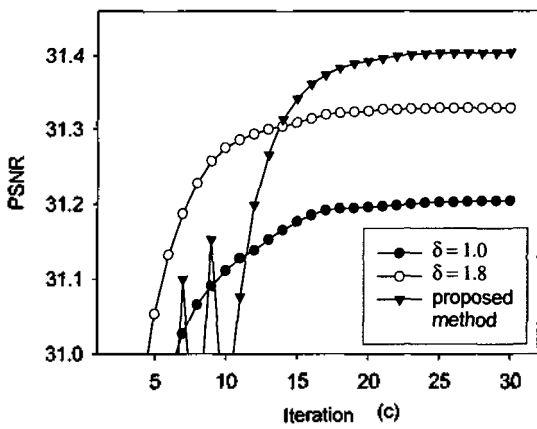
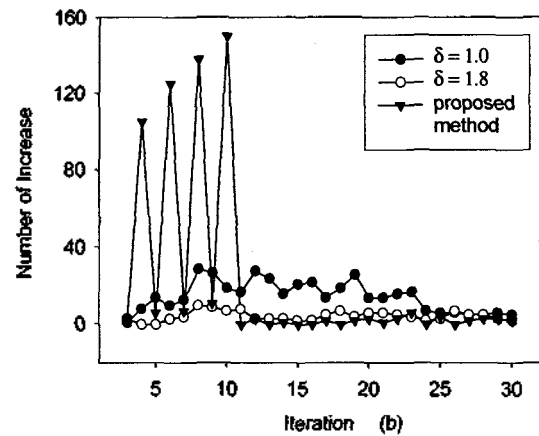
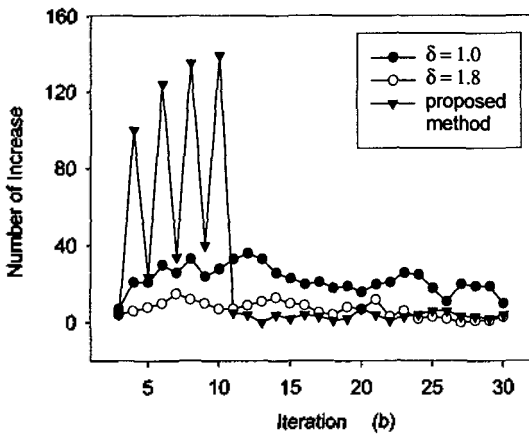
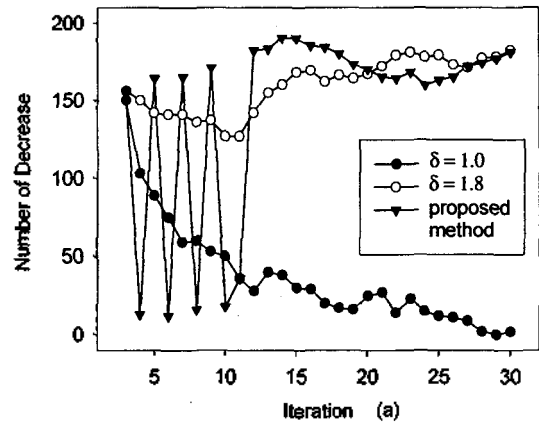
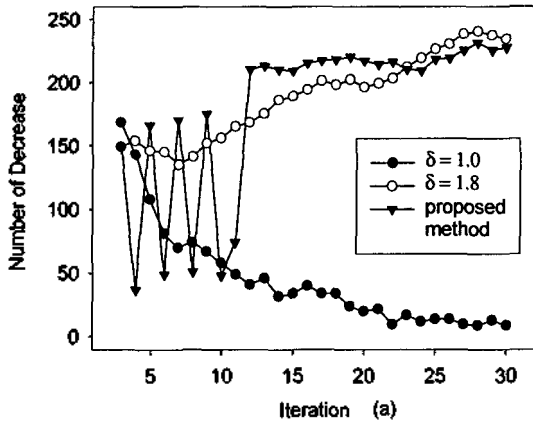


그림 5. Peppers 영상에 대해

- (a) 수렴하는 코드벡터의 수
- (b) 발산하는 코드벡터의 수
- (c) 학습반복에 대한 PSNR

Fig. 5. (a) The number of convergent codevectors, (b) The number of divergent codevectors, (c) PSNR versus leaning iterations for Peppers image.

그림 6. Zelda 영상에 대해

- (a) 수렴하는 코드벡터의 수
- (b) 발산하는 코드벡터의 수
- (c) 학습반복에 대한 PSNR

Fig. 6. (a) The number of convergent codevectors, (b) The number of divergent codevectors, (c) PSNR versus leaning iterations for Zelda image.

참 고 문 헌

1. Y.Linde, A.Buzo, and R.M.Gray, "An algorithm for vector quantizer design," IEEE Trans.Commun, COM-28, 84-95, 1980.
2. W.H.Equitiz, "A new vector quantization clustering algorithm," IEEE Trans.Acoust.Speech and Signal Proc., 7, 1568-1575, 1989.
3. I.Katsavounidis, C.C. Jay Kuo, and Z.Zhang, "A new initialization technique for generalized Lloyd iteration," IEEE Signal Processing Letters, 1, 144-146, 1994.
4. M.Rabbani and P.W.Jones, *Digital image compression techniques*(SPIE Press, 1991)
5. D.Lee, S.Baek, and K.Sung, "Modified K-means algorithm for vector quantizer design," IEEE Signal Processing Letters, 4, 2-4, 1997.
6. M.R.Anderberg, *Cluster analysis for applications*(Academic, New York, 1973)

▲ 조 제 황(Che-Hwang Cho)



1984년 2월 : 광운대학교 전자공학과 (학사)
 1986년 2월 : 광운대학교 대학원 전자공학과(석사)
 1990년 2월 : 광운대학교 대학원 전자공학과(박사)
 1989년 3월~현재 : 동신대학교 전기·전자공학부 부교수

※ 주관심분야: 적응신호처리, 영상처리, 패턴인식, MEMS