

KD-Trees 와 PCA를 이용한 MPEG 비디오 검색

김대일, 장혜경, 홍종선, 김영호, 강대성
동아대학교 전자공학과

MPEG Video Retrieval using KD-Trees and PCA

Daeil Kim, Hye-Kyoung Jang, Jong-Sun Hong, Young Ho Kim, Dae-Seong Kang

Dept. of Electronic Eng., Dong-A Univ.

E-mail: daeilu@hanmail.net

요약

본 논문은 동영상 압축 부호화에 대한 표준안인 MPEG 기반의 압축 비디오 stream에서 [1, 2], 질의 영상에 대한 효율적인 검색 기법을 제안한다. 비디오 검색은 높은 차원의 색인 정보를 이용하는데, 높은 차원의 data set을 색인 정보로 하여 효율적인 검색 능력을 보여주는 KD-Trees(K Dimensional-Trees) 알고리즘 [3]을 비디오 검색 기법에 적용하고자 한다. 먼저, key frame에 PCA(Principal Component Analysis) 알고리즘 [4]을 이용하여 색인 정보를 추출한 다음, 추출된 색인 정보를 KD-Trees에 적용하여 효율적인 검색을 가능하게 한다. 실험 결과, 기존의 검색 기법보다 상당한 양의 처리 시간과 메모리 공간을 줄일 수 있음을 보였다.

I. 서론

최근 디지털 기술의 발전은 영상과 음향 데이터를 통신 및 컴퓨터와 결합하여 새로운 멀티미디어로 발전하고 있다. 이들 중에서 비디오와 관련된 연구로는 비디오 색인 및 검색에 관한 연구가 많이 진행되고 있다 [5, 6]. 비디오 데이터는 매우 복잡한 계층 구조를 가지는데 이를 효과적으로 표현하기 위해서는 여러 단계의 계층으로 나눌 필요가 있다. 비디오 데이터의 가장 기본이 되는 단위는 프레임(frame)이며, 그 다음 상위 단위는 필름이 끊기지 않고 시간적으로나 공간적으로 연속해서 촬영된 일정 구간의 프레임들로 구성된 shot이다. 먼저, shot 단위로 비디오 stream을 분석하여, 각 shot마다 key frame들을 추출하였다. 여기에 PCA 알고리즘을 적용하여 색인 정보를 저장한다. 색인 정보를 바탕으로 검색하는 Tree 구조에는 여러 가지가 있는데, 본 논문에서는 KD-Trees

구조를 이용하여 검색하였다. 고 차원의 point들을 색인 정보로 이용하여 검색하는 많은 방법들 중 하나인 KD-Trees는 향상된 검색 능력을 보여준 알고리즘이다. 한 data point의 좌표는 vector로 표현되기 때문에 차원이 증가하면 entry size가 점점 더 커지게 된다. 그 결과로써, node capacity가 줄어들게 되고, 이것은 결국 검색 경로를 늘어나게 하여, 검색능력을 떨어뜨리는 원인이 된다. 비디오 검색은 매우 높은 차원의 색인 정보를 다루기 때문에 이런 문제의 발생은 검색 성능에 지대한 영향을 미치게 된다. KD-Trees는 이런 문제의 발생을 막고, 효율적인 검색을 가능하게 한다. 본 논문에서 제안하는 검색 기법은 실험을 통해 유사성만을 이용한 기존의 검색 기법보다 성능의 우수함을 보였다.

II. PCA를 이용한 색인 정보 저장

본 논문에서는 key frame에서 색인 정보를 추출할 때, 주성분 해석 기법(PCA)을 이용한다. 주성분 해석 기법은 입력 공간내의 변동량이 많은 방향으로 주축을 발견하여 데이터의 중복성을 제거하는 기법으로 주성분 분석 기법이라고도 한다. 내용기반 검색은 영상 특징 정보의 차이 값을 척도로 하여 데이터베이스 내의 모든 영상들과 검색 대상 영상을 비교하는 것으로 방대한 계산량을 요구한다. 따라서 주성분 해석을 이용한 고차원 데이터의 차원 감소는 영상 분석에 있어 유용하다. 즉, $n \times n$ 의 2차원 영상을 n 개의 1차원 데이터 요소로 축소함으로써 상당한 양의 처리 시간과 메모리 공간을 줄일 수 있다. 또한 주성분 해석에 의하여 계산된 주축은 입력 벡터의 최대 고유치의 고유 벡터의 방향과 일치하며 입력 벡터의 변동량이 가장 큰 방향이다. 다음으로 영역 분할 알고리즘을 이용해서 주축과 수직을 이루면서 노드의 평균을

지나는 지점을 기준으로 영역을 2개로 분할한다. 각 영역의 평균 오차를 계산하여 평균 오차가 가장 큰 영역을 선별한 후 선택된 영역에 영역 분할 알고리즘을 재 적용한다. 다음은 주축을 구하는 알고리즘과 영역 분할 알고리즘이다.

1. 주축을 구하는 알고리즘

① 입력벡터에 대한 기하학적 평균(geometrical mean) M 을 구한다.

$$M = \begin{bmatrix} M_x \\ M_y \end{bmatrix}$$

$$M_x = \frac{1}{q} \sum_{i=0}^{q-1} I_{x_i}, \quad M_y = \frac{1}{q} \sum_{i=0}^{q-1} I_{y_i}$$

여기서, q 는 총 입력벡터의 개수이고, I_{x_i}, I_{y_i} 는 입력벡터의 x축, y축 I_x, I_y 의 i 번째 요소를 나타낸다.

② 입력 벡터에 대한 공분산 행렬(covariance matrix) C 를 구한다.

$$C = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}, \text{ where}$$

$$C_1 = \frac{1}{q} \sum_{i=0}^{q-1} (x_i^2 - M_x^2)$$

$$C_2 = C_3 = \frac{1}{q} \sum_{i=0}^{q-1} (x_i y_i - M_x M_y)$$

$$C_4 = \frac{1}{q} \sum_{i=0}^{q-1} (y_i^2 - M_y^2)$$

③ 가장 큰 고유치(the largest eigenvalue)를 구한다.

$$A = C_1 + C_4$$

$$B = A^2 - 4(C_1 C_4 - C_2 C_3)$$

$$\lambda_{\max} = \frac{A + \sqrt{B}}{2}$$

④ 주축의 각도(θ)를 계산한다.

$$\theta = \text{atan}\left[\frac{\lambda_{\max} - C_1}{2}\right]$$

2. 영역 분할 알고리즘

① 같은 영역에 할당된 입력 벡터의 평균 좌표를 계산한다.

② 주축을 구하는 알고리즘을 이용하여 주축과 수직이면서 노드의 평균 좌표를 지나는 직선과의 각도(Φ)를 계산한다.

③ 다음의 식에 의해서 해당 영역의 각 데이터는 두 영역으로 분할된다.

$$\text{If } (\cos(\Phi) \leq 0) \text{ then 영역 1} \\ \text{else 영역 2}$$

그림 1은 주축을 구하는 알고리즘과 영역 분할 알고리즘을 적용하여 전체 입력 벡터를 두 부분으로 나누는 과정을 나타낸 것이다.

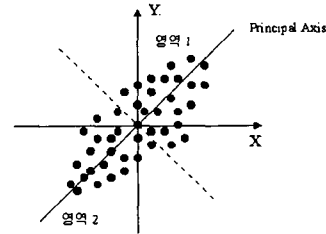


그림 1. 주축에 의한 영역 분할

이상의 알고리즘에 의해 전체 입력 벡터는 여러 개의 영역으로 분할되고 각 영역의 평균값을 코드워드로 사용한다. 그리고 추출된 key frame은 각 코드워드에 할당된 값을 카운트하여 코드북을 만든다. 이 카운트 값을 MPEG 비디오 stream의 색인 정보로 사용된다. 그림 2는 카운트 값을 조절한 각각의 색인 정보를 보여주고 있다.

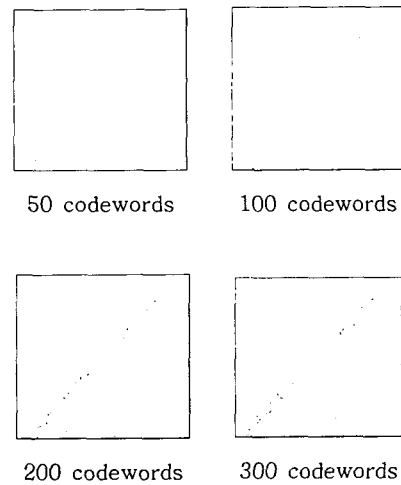


그림 2. 카운트 값에 따른 색인 정보

III. KD-Trees

KD-Trees의 기본 개념은 tree 구조의 중복 정보를 제거하는 것에 있다. 중복 정보의 제거는 색인 node들의 capacity를 증가시켜서 결국은 검색 능력의 향상을 가져온다. 그림 3은 세 개의 색인 영역을 보여준다. KD-Trees에서, 영역 R1에 대한 색인 entry는 <1, 0, 0.4, cp1>이다. 또, 영역 R2에 대한 색인 entry는 <2, 0.4, 0.5, 1, 1, cp2>이고, 영역 R3에서는 <2, 0.4, 0, 1, 0.5, cp3>이다. 여기서 cp1, cp2, cp3는 자식 node pointer들이다.

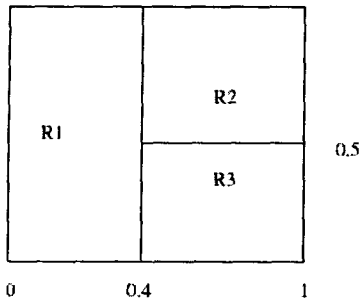


그림 3. 세 개의 색인 영역

0.4의 값이 모든 영역에서 공유되고, 0.5의 값은 두 개의 색인 entry들에서 공유됨을 알 수 있다. 그림 4는 이를 바탕으로 하여 KD-Trees 구조를 나타내었다.

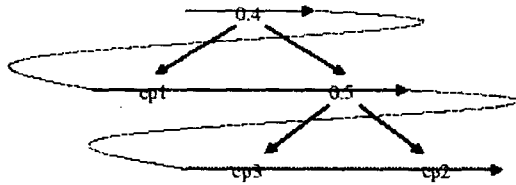


그림 4. KD-Trees 구조

질의 객체가 삽입되면, 그림 4와 같이 tree는 leaf page 부터 root까지 가로질러 검색하게 된다. 각각의 division node에서 node 값이 질의 값보다 크면 왼쪽 node를 선택하고, 그렇지 않으면 오른쪽 node를 선택한다. 질의 객체가 leaf node에 도달하면 검색을 마치게 된다. 그림 5는 개념을 확장하여 비디오 stream에서 추출된 색인 정보 일부를 KD-Trees 구조로 나타내었다.

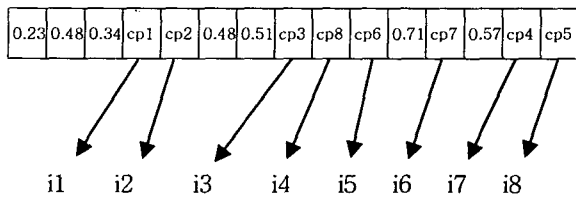


그림 5. KD-Trees 구조의 예

IV. 전체 알고리즘

그림 6은 본 논문에서 제안한 검색 기법을 적용한 전체 알고리즘이다. MPEG 비디오 stream에서 DC extractor로 DC image를 추출한 후 shot 경계를 검출한다. 각 shot마다 key frame을 추출한 다음 PCA 알고리즘을 적용하여 색인 정보를 저장하고, 영역 분할 알고리즘을 통해 각 색인 영역들을 지정한다. 이를 바탕으로 KD-Trees를 구성한 후 검색한다.

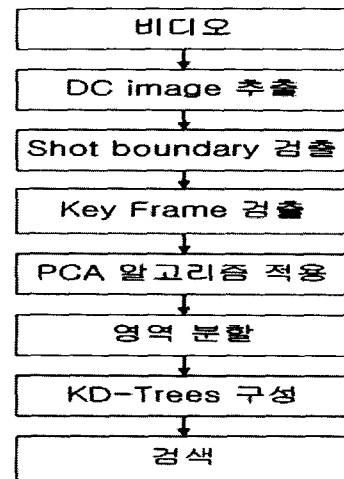


그림 6. 전체 알고리즘

V. 실험결과

본 논문에서는 제안한 검색 알고리즘과 기존의 알고리즘을 비교하기 위해 테스트 영상으로 352x240 size인 뮤직비디오들을 사용하였다. 검색을 하기 전에 먼저 비디오 stream에서 DC 영상을 추출하고, shot 경계 검출 후 key frame들을 찾는다. 그림 6(a), (b)는 DC 영상 추출 시스템이고, (c), (d)는 검출된 DC 영상이다.

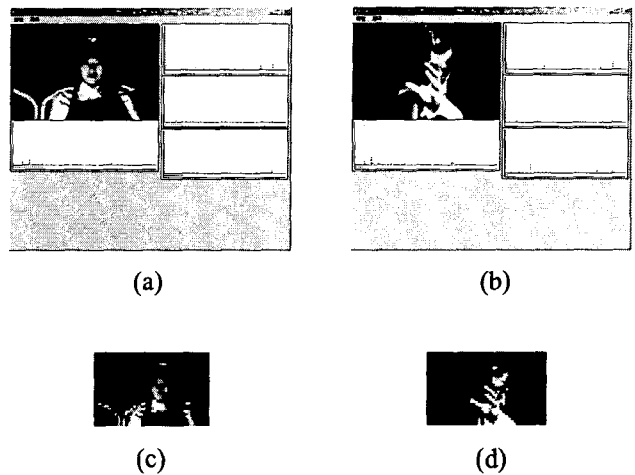


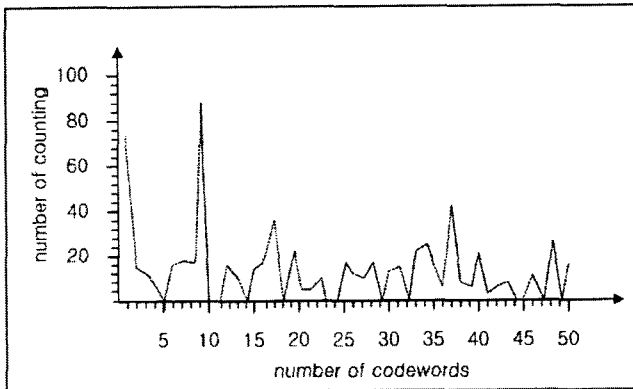
그림 6. (a), (b) DC image extractor와 (c), (d) 결과 DC image

추출된 key frame들의 feature를 위 장에 설명한 PCA 알고리즘을 통해 추출하여, 기존의 검색 기법과 KD-Trees를 이용한 검색 기법을 비교, 분석한다. 그림 6(a)는 추출된 key frame이며, 이 key frame에 PCA를 적용하여 코드워드를 추출한 후, 6(b)에서 보여주는 바와 같이, 생성된 50개의 코드워드와 대표 프레임에 할당된 카운트 값을 나타낸다. x축은 코드워드를 나타내고, y축은

각 코드워드에 할당되는 카운트 회수를 나타낸다.

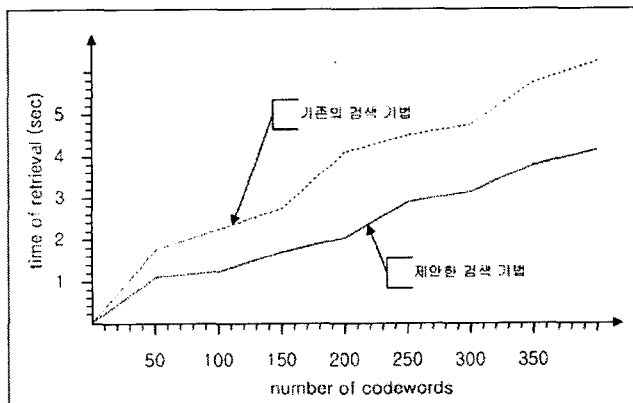


(a)

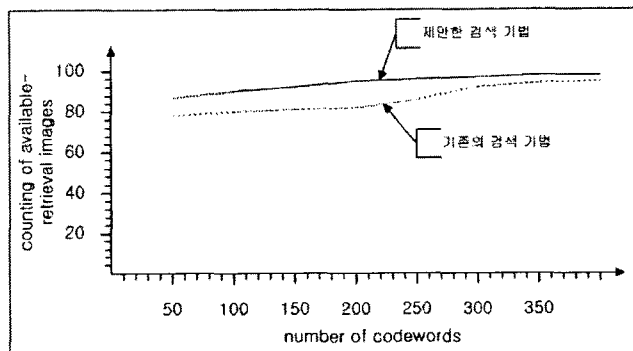


(b)

그림 6. (a) Key frame의 DC image 와 (b) 코드북 생성 결과



(a)



(b)

그림 7. (a) 검색 시간과 (b) 검색 성능

그림 7(a)에서는 각각의 코드워드 수에 대한 기존 검색 기법과 제안한 검색 기법의 검색 시간을 비교하였다. 코드워드의 수가 늘어남에 따라 제안한 검색 기법의 검색 시간이 기존의 그것보다 감소함을 알 수 있다. 7(b)는 코

드워드 수에 대한 검색 성능을 비교하였다. 검색 성능은 질의 영상과 결과 영상의 명도 성분 비교를 통해 5% 이내의 오차만 허용하였다. 100개의 질의 영상에 대해 검색한 결과 영상이 오차 범위 이내의 영상일 경우를 카운트 하였다.

VI. 결론

기존의 검색 방법에서는 MPEG 비디오 stream으로부터 DCT DC 계수를 추출하고, 이로 구성된 DC 영상에 대해 shot 경계를 검출한 다음, key frame들을 정한다. 그리고, 2차원 입력 벡터로 나타낸 후 PCA에 기반한 주축을 구하고, 영역을 분할하여 유사도 측정 함수를 통해 검색하였다. 본 논문에서는 단순한 유사성을 넘어 시간적, 공간적으로 효율적인 KD-Trees를 이용한 영상 검색 기법을 제시하였다. 실험 결과의 비교 분석으로부터 제안한 기법은 기존의 검색 기법보다 상당한 양의 처리 시간과 메모리 공간을 줄일 수 있음을 보였다. 향후 tree 구조의 분할과 병합에 관한 연구를 통해 검색과정 중의 page capacity를 감소시키고자 한다.

VII. 참고 문헌

- [1] Wei Xiong and Chung-Mong Lee, "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification", Computer Vision and Image Understanding Vol.71, No. 2/2, August, pp. 166-181, 1998.
- [2] Hong Heather Yu, "A hierarchical Multiresolution Video shot Transition Detection Scheme, Computer Vision and Image Understanding Vol.75, No. 1/2, July/ August, pp. 196-213, 1999.
- [3] J. L. Bentley, "Multidimensional binary search trees used for associative searching", Communications of ACM 18(9), pp. 509-517, 1975.
- [4] Oja E., "A simplified neuron model as a principal component analyzer", Journal of Mathematical Biology 15, pp.267-273, 1982.
- [5] Wei Xiong and Chung-Mong Lee, "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification", Computer Vision and Image Understanding Vol.71, No. 2/2, August, pp. 166-181, 1998.
- [6] Hong Heather Yu, "A hierarchical Multiresolution Video shot Transition Detection Scheme", Computer Vision and Image Understanding Vol.75, No. 1/2, July/ August, pp. 196-213, 1999.