

VDCluster : 대용량 비디오 시퀀스를 위한 비디오 세그멘테이션 및 클러스터링 알고리즘

(VDCluster : A Video Segmentation and Clustering Algorithm for Large Video Sequences)

이 석 룡[†] 이 주 홍^{**} 김 덕 환^{***} 정 진 완^{****}
 (Seok-Lyong Lee) (Ju-Hong Lee) (Doek-Hwan Kim) (Chin-Wan Chung)

요 약 본 논문에서는 비디오 저장이나 검색과 같은 비디오 정보 처리를 위한 중요한 기초 연구로써 비디오의 표현을 위한 효과적인 기법을 제안한다. 비디오 데이터 세트는 수초에서 수분 사이의 상연 시간을 갖는 비디오 클립들의 집합이며, 각 비디오 클립은 연속된 비디오 프레임들로 구성되어 있다. 이 비디오 클립은 다차원 데이터 시퀀스(multidimensional data sequence: MDS)로 표현될 수 있으며, 프레임 사이의 시간적인 정보를 고려하여 비디오 세그먼트로 나누어 지고, 한 클립 내에서 서로 유사한 세그먼트들은 다시 비디오 클러스터로 군집화된다. 따라서, 각 비디오 클립은 소수 개의 비디오 클러스터로 표현되어 진다. 본 논문에서 제안한 비디오 세그멘테이션 및 클러스터링 알고리즘 *VDCluster*는 사전에 정의된 일정 수준의 클러스터링 품질을 보장하고 있으며, 다양한 비디오 시퀀스에 대한 실험을 통하여 알고리즘의 효과를 입증한다.

키워드 : 다차원 데이터 시퀀스, 비디오 세그멘테이션, 클러스터링

Abstract In this paper, we investigate video representation techniques that are the foundational work for the subsequent video processing such as video storage and retrieval. A video data set is a collection of video clips, each of which is a sequence of video frames and is represented by a multidimensional data sequence (MDS). An MDS is partitioned into video segments considering temporal relationship among frames, and then similar segments of the clip are grouped into video clusters. Thus, the video clip is represented by a small number of video clusters. The video segmentation and clustering algorithm, *VDCluster*, proposed in this paper guarantee clustering quality to such an extent that satisfies predefined conditions. The experiments show that our algorithm performs very effectively with respect to various video data sets.

Key word : Multidimensional Data Sequences, Video Segmentation, Clustering

1. 서 론

비디오 정보 처리는 대용량의 저장 공간과 고속의 컴퓨터 처리 능력이 전제되어야 하기 때문에 그 효율성에

도 불구하고 다루기 어려운 분야 중의 하나이다. 이러한 어려움을 극복하기 위하여 비디오 데이터를 효과적으로 표현하고, 저장하고, 또한 검색하는 기술은 필수적이다. 비디오 데이터베이스는 수많은 비디오 클립을 포함하고 있으며, 각 비디오 클립은 MDS로 표현될 수 있다. 이전의 연구[1]에서, n 차원 공간에서 K 개의 점들로 이루어진 MDS S 를 다음과 같이 정의하였다.

$$S = (S[1], S[2], \dots, S[K])$$

여기에서, 각 벡터 요소 $S[j]$ ($1 \leq j \leq K$)는 n 개의 스칼라 항목으로 구성되어 있으며, 다음의 식으로 표시된다: $S[j] = (S^1[j], S^2[j], \dots, S^n[j])$. 비디오 클립은 복수 개의 프레임이 시간 순으로 연속되어 있으며, 각 프레임은 RGB나 YCbCr 칼라 공간과 같은 특징 공간(feature

· 이 논문은 2000년도 한국학술진흥재단의 지원에 의하여 연구되었음.
(KRF-2000-041-E00262)

† 정 회 원 : 한국외국어대학교 산업정보시스템공학부 교수
slee@hufs.ac.kr

** 종 신 회 원 : 인하대학교 컴퓨터공학부 교수
juhong@inha.ac.kr

*** 비 회 원 : 한국과학기술원 정보통신공학과
dhkim@islab.kaist.ac.kr

**** 종 신 회 원 : 한국과학기술원 전자전산학과 교수
chungcw@islab.kaist.ac.kr

논문접수 : 2001년 8월 3일
심사완료 : 2002년 2월 18일

space)에서 다차원 벡터로 표시된다. 따라서, 비디오 클립은 다차원 공간에서 각 프레임들 하나의 점으로 사상함으로써 연속된 점들의 시퀀스로 모델링할 수 있다. 비디오 클립을 MDS로 모델링함으로써, 비디오 클립의 프레임들을 클러스터링하는 문제는 다차원 공간에서 MDS에 속해있는 점들을 클러스터링하는 문제로 변형될 수 있다. 각 MDS는 비디오 세그먼트로 나누어지고 유사한 세그먼트들은 다시 비디오 클러스터로 군집화된다. 그림 1은 비디오 자료의 계층 구조를 나타낸다.

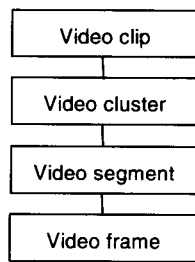


그림 1 비디오 자료의 계층 구조

클러스터링 문제는 판매 분석(sales analysis), 패턴 인식, 유사성 검색 등의 데이터베이스 응용 분야에서 많이 연구되어 왔으며, 다차원 공간의 주어진 점들에 대하여, 유사한 특성을 갖는 점들은 같은 클러스터에, 상이한 특성을 갖는 점들은 다른 클러스터에 군집화 하는 것이다. 이때, 어느 클러스터에도 속하지 않는 점들은 주변점(outlier)이 된다.

다양한 클러스터링 기법들이 연구되어 왔지만, 비디오 데이터의 클러스터링은 기존의 클러스터링과는 여러 측면에서 다르다. 첫째, 비디오 클러스터링에서는 프레임, 즉 점들 간의 시간적 순서가 고려되어야 한다. 둘째, 기존의 방법에서는 클러스터링의 대상이 되는 한 객체가 하나의 점으로 표현되어 하나의 클러스터링에 속하였으나, 비디오 클러스터링에서는 한 객체(비디오 클립)가 여러 개의 점들로 이루어져 있고, 따라서 여러 개의 클러스터에 나뉘어져 포함된다. 셋째로, 클러스터의 형태에 있어서도 차이가 발생할 수 있다. 기존의 방법들은 클러스터가 추후 어떻게 사용될 것인가 보다는 클러스터 자체의 정량적인 특성에 초점을 맞추는 경향이 있다. 즉, MSE(mean square error)와 같은 주어진 조건을 최적화하는 클러스터를 구하는 문제에 집중하고 있다. 그러므로 클러스터의 형태는 데이터 공간에서 점들의 분포에 따라 대개 임의의 모습이 되게 된다. 반면에, 비디오 데이터의 클러스터링에서는 클러스터링 자체 뿐만

아니고 차후의 비디오 저장 및 검색 등도 염두에 두어야 하므로, 이러한 사항들도 클러스터링에 반영되어야 한다. 또한, 클러스터의 형태는 기존의 인덱싱 메커니즘에도 적합해야 한다. 현재 널리 사용되고 있는 R-tree [2]나 그 외 여러 인덱스 구조들[3,4,5]이 노드의 형태로써 하이퍼 사각형(hyper-rectangle)을 채택하고 있으므로, 본 논문에서는 클러스터의 형태를 하이퍼사각형으로 제한한다. 본 논문에서의 클러스터링 문제는 주어진 비디오 클립 데이터 세트, 비디오 세그먼트당 최소의 점수 $minPts$ 에 대하여, 사전에 정의된 요건을 최적화하는 비디오 클러스터의 집합 및 주변점들의 집합을 구하는 것으로 정형화 될 수 있다. 입력 인수 $minPts$ 는 주변점을 결정하기 위하여 사용된다. 세그멘테이션 후의 비디오 세그먼트 내의 점들의 수가 $minPts$ 보다 적으면 그 세그먼트 내의 모든 점들은 주변점으로 간주된다.

제안한 기법의 첫번째 단계에서는 비디오 클립을 파싱하여 MDS 데이터 세트를 생성한다. 비디오 클립의 각 프레임으로부터 특징 값들(feature values)이 추출된다. 이것은 한 프레임의 픽셀들, 또는 한 프레임을 일정한 구획으로 나눔으로써 생성된 각 블록의 픽셀들의 색상 값을 평균하여 구한다. 다음은 세그멘테이션 단계로써, 생성된 MDS가 사전에 정의된 기하학적, 의미적 요건을 충족시키는 범위 내에서 비디오 세그먼트로 나누어진다. 주변점들은 이 단계에서 식별된다. 마지막으로 클러스터링 단계에서는 한 MDS 내의 유사한 세그먼트들이 보다 나은 클러스터링 품질을 확보하기 위하여 클러스터로 군집화된다. 이렇게 하여 주어진 비디오 클립은 소수 개의 비디오 클러스터로 표현되며 전체적인 과정이 그림 2에 표시되어 있다.

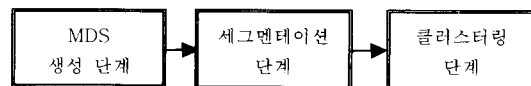


그림 2 제안된 기법의 전체 구조

클러스터링 단계에서 생성된 비디오 클러스터는 하이퍼사각형 형태이며 따라서 차후의 검색 처리를 위하여 R-tree [2]나 다른 다차원 인덱스 구조들[3,4,5]을 사용하여 인덱싱되어 데이터베이스에 저장되게 된다. 검색 과정에서, 질의 비디오는 하나의 프레임 혹은 비디오 클립의 형태로 주어지게 되며, 이 질의 비디오는 역시 MDS 생성 단계와 세그멘테이션 단계를 거쳐서 하나 혹은 복수 개의 비디오 세그먼트로 표현된다. 이 비디오 세그먼트의 하이퍼사각형을 기초로 데이터베이스에 저

장된 비디오 클러스터를 탐색하여 후보 클러스터를 구하고, 이 후보 클러스터로부터 원하는 비디오 세그먼트를 검색하게 된다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구와 함께 데이터 시퀀스의 클러스터링에 대하여 간단하게 언급한다. 3 장에서는 클러스터링에 관련된 기본 정의와 클러스터링의 특성 및 클러스터링의 품질을 측정하기 위한 여러 가지 기준에 관하여 논의한다. 비디오 세그멘테이션 과정과 알고리즘이 4 장에서 기술되고, 5 장에서는 유사한 세그먼트들을 병합함으로써 비디오 클러스터를 생성하는 알고리즘에 대하여 기술한다. 6 장에서 실험 결과 및 간단한 분석을 제공하고, 7 장에서 결론을 맺는다.

2. 관련 연구

다차원 공간에서 데이터 포인트들을 클러스터링하기 위하여 다양한 기법들이 제안되었다. CLARANS[6]는 임의의 검색에 기반하고 있고, 두 개의 사용자 입력 인수를 사용하여 검색 공간을 축소함으로써 효율성을 높이고 더욱 설계한 클러스터링 기법이다. BIRCH[7]는 데이터 베이스를 읽어 *CF-tree*라 불리는 인덱스를 생성하는 다단계 클러스터링 기법이며, 트리의 단말 노드를 군집화하기 위하여 임의의 클러스터링 알고리즘을 사용한다. DBSCAN[8]의 기본 아이디어는 클러스터내의 각 포인트에 대하여 주어진 반경 안에 정해진 최소 수의 포인트들을 포함하게 하는 것이다. 따라서 이 기법은 두 개의 입력 인수 (반경과 최소 포인트 수) 만을 필요로 한다. CLIQUE[9]는 주어진 고차원 데이터 공간의 서브 공간에서 밀도가 높은 클러스터를 자동으로 식별할 수 있는 기법이다. 즉, 주어진 공간에서는 클러스터가 발견되지 않아도 그 서브 공간에서는 클러스터가 존재할 수 있으며 이 경우에 적합한 기법이다. 서브 공간에서의 클러스터링에 관한 아이디어는 [11]에서 사상된 클러스터링(projected clustering)의 개념으로 확장되었으며, 이것은 데이터 포인트가 밀접하게 연관되어 있는 특정 서브 차원을 선택하고 이 차원에 대한 클러스터를 찾아내는 기법이다. CURE[10]는 구형이 아니면서 다양한 크기를 갖는 클러스터를 생성한다. 이 기법에서는 각 클러스터를 잘 산재하여 분포하고 있는 복수 개의 포인트들로 나타낸다. 구형이 아닌 클러스터는 여러 개의 포인트를 사용하면 그 모양을 보다 잘 표현할 수 있다. 그러나 위에 언급된 기법들은 여러 개의 입력 인수를 필요로 하며, 데이터 포인트들간의 시간적, 의미적 관계를 고려하고 있지 않다. 따라서, 이 기법들은 프레임 간의 시간

순서와 의미가 중요한 비디오 클립과 같은 데이터 시퀀스의 클러스터링에는 적용할 수 없다.

한편, 영상처리 분야에서는 여러 가지 비디오 분석 방법 [12, 13, 14, 15]이 제안되었다. 이들 방법은 먼저 카메라 이동으로 인한 샷 경계의 검출 및 검출된 샷 내에서의 키 프레임의 선택, 그리고 의미상으로 연관이 있는 샷들을 시간을 고려하여 장면(scene)으로 추상화하는 기법들을 제안하고 있다. 이 기법들에 기반한 비디오 검색에서는 주로 각 비디오 샷 마다 한두 개의 키 프레임이 선택되고 질의는 선택된 프레임들에 대하여 행해진다[16]. 그러나 이러한 검색 방식은 키 프레임이 샷 내의 모든 프레임들을 효과적으로 요약하지 못하므로 정확성(correctness)을 보장하지 못한다. 따라서, 엄밀한 의미에서 다차원 공간에서 시퀀스로 표현되는 비디오 스트림에 관한 검색이라기 보다는 키 프레임에 의한 정지 영상 검색의 측면이 강하다. 즉, 기존의 접근 방식은 비디오 분석을 통하여 비디오를 세그멘테이션하고 표현하는 것에 중점을 두며, 검색은 검출된 키 프레임에 의거하여 수행되는데 비하여, 제안한 방법에서는 기존의 방식을 반영한 의미적 요소 외에도 클러스터의 기하학적 요소(볼륨과 에지)를 고려하여 다차원 공간 상에서 정확성을 보장하고 효율적인 검색이 가능하도록 설계했다는 점에서 차이가 있다. [1]의 연구에서 우리는 과오 누락(false dismissal)을 방지하기 위하여 한두 개의 대표 점에 기반하지 않고, 세그먼트 내의 모든 점들을 포함하는 최소의 경계 사각형(minimum bounding rectangle)을 기초로 한 다차원 데이터 시퀀스의 유사성 검색 기법을 제안하였다. 정확성을 보장하는 것은 유사성 검색에서 중요한 문제이다.

데이터 시퀀스에 관한 최초의 알고리즘은 [17]에서 제안되었으며, 이 기법은 한 시퀀스를 모양이 하이퍼 사각형인 복수 개의 서브 시퀀스로 군집화한다. 이 알고리즘은 [18]에서 샷 경계를 검출하기 위해 알고리즘의 수행이 전방향과 후방향의 두 방향으로 이루어지도록 약간 수정되었으며, 또한 다차원의 하이퍼 사각형 질의를 지원하기 위하여 [1]에서 약간 수정되었다. 이 알고리즘들은 평균 디스크 접근 수를 클러스터 내의 포인트 수로 나눈 값인 한계 비용(marginal cost: MCOST)에 근거하여 클러스터링을 수행한다. 즉, MCOST를 결정하기 위하여 이 기법들은 한 포인트가 클러스터에 포함될 때 포함되는 포인트로 인한 클러스터의 볼륨의 증가를 중요한 클러스터링 요인으로 고려한다. 그러나 볼륨 요인만 고려하는 것은 부족하다. 클러스터의 에지의 길이와 클러스터 내의 점들 사이의 유사성도 중요한 요인으

로써 고려되어야 한다. 이 문제는 다음 장에서 다루어진다.

3. 클러스터링 특성

이 장에서는 비디오 세그먼트와 비디오 클러스터를 정의하기 위해 사용되는 하이퍼사각형의 여러 가지 특성들에 대하여 논의하고, 또한 효과적인 세그멘테이션 및 클러스터링을 수행하기 위한 클러스터링 요소들에 대하여 고려한다.

n 차원공간에서 k 개의 점 P_j ($j = 1, 2, \dots, k$)를 갖는 하이퍼사각형 HR 은 다음과 같이 두 개의 끝점 L (low point)과 H (high point), 사각형내의 점 n 으로 표현된다: $HR = \langle L, H, k \rangle$. 여기에서 $L = \{(L^1, L^2, \dots, L^n) \mid L^i = \min_{1 \leq j \leq k} (P_j^i)\}$, $H = \{(H^1, H^2, \dots, H^n) \mid H^i = \max_{1 \leq j \leq k} (P_j^i)\}$ ($i = 1, 2, \dots, n$) 이다. 또한 점 P_j 를 모든 차원에 대하여 $L^i = H^i = P_j^i$ 로 놓음으로써 편의상 하이퍼사각형 형태로 나타낼 수 있다. 즉, $\langle P_j, P_j, 1 \rangle$ 이 된다. 이러한 사각형은 볼륨과 에지가 0이 되며 $HR(P_j)$ 로 표시된다. 하이퍼사각형 HR 의 볼륨 $Vol(HR)$ 과 에지의 길이 $Edge(HR)$ 은 다음의 식으로 계산된다:

$$Vol(HR) = \prod_{1 \leq i \leq n} (HR.H^i - HR.L^i)$$

$$Edge(HR) = 2^{n-1} \cdot \sum_{1 \leq i \leq n} (HR.H^i - HR.L^i)$$

따라서 HR 의 한 점 당 볼륨과 에지, $VPP(HR)$ 와 $EPP(HR)$ 는 각각 다음과 같이 표시된다:

$$VPP(HR) = \frac{Vol(HR)}{HR.k} = \frac{\prod_{1 \leq i \leq n} (HR.H^i - HR.L^i)}{HR.k}$$

$$EPP(HR) = \frac{Edge(HR)}{HR.k} = \frac{2^{n-1} \cdot \sum_{1 \leq i \leq n} (HR.H^i - HR.L^i)}{HR.k}$$

두 개의 하이퍼사각형은 비디오 세그멘테이션 및 클러스터링 과정에서 병합될 수 있다. 이 병합 연산을 위하여 다음의 병합 연산자를 정의한다:

정의1 (병합 연산자 \oplus). 두 개의 하이퍼사각형 HR_1 과 HR_2 사이의 병합 연산자 \oplus 은 다음과 같이 정의된다: $HR_1 \oplus HR_2 = HR_3$. 여기에서 $HR_3.L = \{(HR_3.L^1, HR_3.L^2, \dots, HR_3.L^n) \mid HR_3.L^i = \min(HR_1.L^i, HR_2.L^i)\}$, $HR_3.H = \{(HR_3.H^1, HR_3.H^2, \dots, HR_3.H^n) \mid HR_3.H^i = \max(HR_1.H^i, HR_2.H^i)\}$ (단, $i = 1, 2, \dots, n$), $HR_3.k = HR_1.k + HR_2.k$.

점 P_j 가 하이퍼사각형 $HR = \langle L, H, k \rangle$ 에 병합되는 경우를 고려해 보자. 이 병합 과정은 하이퍼사각형의 볼륨, 에지, 그리고 점의 수에 영향을 줄 것이다. 이 변화의 양은 클러스터링의 중요한 요소가 되며, 다음의 식으

로 표현될 수 있다.

$$\Delta Vol(HR, P) = Vol(HR \oplus HR(P)) - Vol(HR)$$

$$\Delta(Edge(HR, P)) = Edge(HR \oplus HR(P)) - Edge(HR)$$

3.1 두 점 사이의 유사성(similarity)

다차원 공간 상에서 다차원 벡터로 표현되는 두 점간의 유사성은 일반적으로 두 점간의 유클리디안 거리(Euclidean distance)의 함수로써 정의된다. 비디오 프레임간의 유사성도 각각의 특징 벡터 사이의 거리의 함수로 정의된다. n 차원 시퀀스 S 에서 두개의 인접한 점들 사이의 거리는 다음의 식으로 주어진다. :

$$dist(S[j], S[j+1]) = \sqrt{\sum_{1 \leq i \leq n} (S^i[j+1] - S^i[j])^2}$$

여기에서 $S^i[j]$ 는 시퀀스 S 의 j 번째 점의 차원 i 의 좌표값이다.

3.2 클러스터링 요소

본 절에서는 MDS를 클러스터링하기 위하여 고려해야 할 두 가지의 클러스터링 요소, 즉 기하학적(geometric) 요소와 의미적(semantic) 요소에 관하여 논의한다. 전자는 하이퍼 사각형의 기하학적 특성에 관련된 것이며 세그멘테이션과 클러스터링 모두에 적용된다. 반면에 후자는 하이퍼 사각형 내의 각 점들 사이의 의미적인 관계에 관련된 것이며 세그멘테이션에만 적용된다.

기하학적 요소: 클러스터의 기하학적 특성은 검색 효율에 중대한 영향을 미치므로 이 요소를 중요하게 고려해야 한다. 검색 공간에서 볼륨이 큰 클러스터는 부피가 작은 클러스터 보다 질의에 의해 액세스될 확률이 높은 것은 명백하다. 그러나, 클러스터의 볼륨뿐만 아니라 클러스터의 에지 또한 중요한 요소로 고려되어야 한다. 클러스터링에서 에지가 미치는 영향과 구체적인 예는 [19]에 기술되어 있다.

의미적 요소: 하나의 비디오 세그먼트 내에 있는 연속된 두 점은 의미적으로 밀접한 관련이 있으므로 연속된 두 점 사이의 거리 또한 중요한 요소로써 고려되어야 한다. 그림 3에서 보인 것처럼 연속된 점 P_j ($j = 1, 2, \dots, k, k+1, \dots$)로 구성되어 있는 MDS에서, 점 P_{k-1} 가 비디오 세그먼트 VS_1 에 포함되어야 할 지, 아니면 이 점으로부터 새로운 비디오 세그먼트가 시작되어야 할 지를 결정하는 문제를 고려해보자. 점 P_{k-1} 가 VS_1 에 포함될 때의 볼륨과 에지의 증가량을 각각 $\Delta Vol(VS_1, P_{k-1})$, $\Delta Edge(VS_1, P_{k-1})$ 라고 할 때 이 양은 그림에서 전하게 표시되어 있는 부분과 관련된다. 클러스터링 요

소로 볼륨과 에지 만을 고려한다면 ΔVol 과 $\Delta Edge$ 이 작기 때문에 P_{k+1} 은 VS_1 에 포함될 가능성이 높다. 그러나 P_{k+1} 은 공간적으로 P_k 와 멀리 떨어져 있기 때문에, 즉 의미적으로 서로 다르므로 P_{k+1} 부터 새로운 비디오 세그먼트 VS_2 가 시작되는 것이 바람직하다. 이것은 두 개의 연속된 점 사이의 거리 또한 중요한 클러스터링 요소로 고려해야 한다는 것을 의미한다.

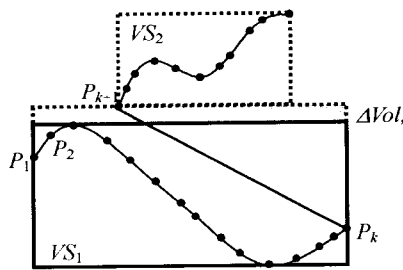


그림 3 클러스터링에서의 의미적 요소

3.3 클러스터링 품질 측정 기준

이 절에서는 클러스터링의 품질을 측정하기 위하여 사용되는 측정 기준을 사용한다. 품질을 측정하기 위한 정량적인 기준으로써 다음의 3가지가 사용된다: Volume of clusters per point (VPP), Edge of clusters per point (EPP), Number of points per cluster (PPC). p 개의 하이퍼 사각형 HR_1, \dots, HR_p 에서 VPP , EPP , 그리고 PPC 는 다음과 같이 정의된다.

$$VPP = \frac{\sum_{1 \leq j \leq p} Vol(HR_j)}{\sum_{1 \leq j \leq p} HR_{j,k}},$$

$$EPP = \frac{\sum_{1 \leq j \leq p} Edge(HR_j)}{\sum_{1 \leq j \leq p} HR_{j,k}}, \quad PPC = \frac{\sum_{1 \leq j \leq p} HR_{j,k}}{p}$$

4. 비디오 세그멘테이션

비디오 클립으로부터 MDS가 생성된 후에는 각 시퀀스는 비디오 세그먼트로 분할된다. 점 P 가 단위 공간 $[0,1]^n$ 에서 하이퍼 사각형 $HR = \langle L, H, k \rangle$ 에 포함되는 경우를 고려해보자. 세그멘테이션은 점 P 의 HR 로의 병합이 주어진 조건을 만족하면 이 점은 현재의 세그먼트의 HR 에 포함되고 그렇지 않을 경우에는 그 점으로부터 새로운 세그먼트가 시작되는 방식으로 진행된다. 이 과정에서 병합의 주체는 하이퍼 사각형이고 병합의 대상은 항상 점이 된다. 다음은 비디오 세그먼트의 정의이다.

정의2 (비디오 세그먼트). k 개의 점들 P_j ($j = 1, 2,$

\dots, k)를 시간 순서로 포함하고 있는 비디오 세그먼트 VS 를 다음과 같이 정의한다: $VS = \langle sid, SP, HR \rangle$, 여기에서 sid 는 segment-id, SP 는 VS 의 시작점, $HR = \langle L, H, k \rangle$. 단, $L = \{(L^1, L^2, \dots, L^n) \mid L^i = \min_{1 \leq j \leq k} (P_j^i)\}$, $H = \{(H^1, H^2, \dots, H^n) \mid H^i = \max_{1 \leq j \leq k} (P_j^i)\}$, $i = 1, 2, \dots, n$.

세그멘테이션 과정 중에서 어떤 점이 특정 세그먼트에 포함되기 위해서는 사전에 정의된 기하학적, 의미적 조건들을 만족시켜야 한다. 다음 절에서 이러한 조건들과 알고리즘을 기술한다.

4.1 기하학적 조건

기하학적 조건은 비디오 세그먼트의 볼륨과 에지에 관한 조건이며 이를 위하여 먼저 단위 하이퍼 큐브의 개념을 소개한다. 단위 하이퍼 큐브는 [19]에서 소개되었으며, 여기서는 본 논문에서 알고리즘을 기술하기 위하여 적합한 형태로 약간 수정하였다. 비디오 세그먼트 S 의 모든 K 개의 점들을 포함하는 최소의 하이퍼 사각형을 HR_S 라 하면, 단위 하이퍼 큐브 $uCUBE$ 는 S 내의 모든 점들이 HR_S 의 하이퍼 사각형 내에 균일하게 분포된다고 가정했을 때, $[0,1]^n$ 공간에서 한 점이 차지하는 큐브로써 정의된다. 이 큐브의 한 변의 길이가 e 라 하면 볼륨과 에지의 길이는 다음의 식으로 표현된다:

$$Vol(uCUBE) = e^n = \frac{Vol(HR_S)}{K},$$

$$Edge(uCUBE) = 2^{n-1} \cdot n \cdot e = 2^{n-1} \cdot n \cdot \sqrt[n]{\frac{Vol(HR_S)}{K}}$$

S 의 모든 점들이 HR_S 내에서 균일하게 분포한다면, 한 점이 하나의 $uCUBE$ 에 할당된다고 생각할 수 있다. 즉, 직관적으로 S 의 각 점이 형태가 단위 하이퍼 큐브인 하나의 비디오 세그먼트를 형성한다고 볼 수 있다. 그러나 여기에서 가정한 균일 분포는 일반적으로 실제 세계에서는 거의 존재하지 않는다. 동일한 비디오 샷 내에 있는 프레임들은 상당한 유사성을 보이고 있으며, 따라서 각 프레임들 한 점으로 나타냈을 때 이 점들은 군집화되는 특성을 보인다. 이 균일 분포가 두 클러스터의 병합을 허용할지의 여부를 결정하는 기하학적 조건을 제공한다. 즉, 볼륨과 에지에 대한 바운딩 한계 값 (bounding threshold) - τ_{vol} 과 τ_{edge} - 은 다음의 식으로 표현한다.

$$\tau_{vol} = Vol(uCUBE) = e^n,$$

$$\tau_{edge} = Edge(uCUBE) = 2^{n-1} \cdot n \cdot e$$

정의 3 (기하학적 바운딩 조건). 공간 $[0,1]^n$ 에서 점 P 가 세그먼트 VS 에 병합될 때, 기하학적 바운딩 조건은

P 를 세그먼트 VS 에 병합하기 위해 만족시켜야 하는 조건이며 다음과 같이 정의된다:

$$\Delta Vol(VS, P) \leq \tau_{vol} \wedge \Delta Edge(VS, P) \leq \tau_{edge}$$

정리 1. 기하학적 바운딩 조건을 만족하는 클러스터링은 VPP 와 EPP 에 대하여 균일 분포의 경우 보다 항상 나은 클러스터링 품질을 보장한다.

증명. [19] 참조.

4.2 의미적 조건

3.2절에서 논의된 또 하나의 중요한 기준은 의미적 요소이다. 어떤 점이 현재의 비디오 세그먼트에 병합될지의 여부를 결정하기 위하여, 그 점과 시퀀스 상의 이전 점과의 거리를 검사한다. 그 거리가 사전에 정의된 한계 값(threshold)을 초과하게 되면 새로운 세그먼트가 그 점으로부터 시작된다. K 개의 점 P_j ($j = 1, 2, \dots, K$)를 갖는 MDS를 고려해 보자. 한계 값 τ_{dist} 는 시퀀스 내의 모든 인접한 두 점 쌍의 거리의 평균으로 정의되며, 다음의 식으로 표시된다.

$$\tau_{dist} = \frac{1}{K-1} \cdot \sum_{1 \leq j \leq K-1} dist(P_j, P_{j+1})$$

정의 4 (의미적 조건). 단위 공간 $[0,1]^n$ 에서 점 P_{k+1} 이 비디오 세그먼트 VS 에 병합되는 경우, 의미적 조건은 P_{k+1} 을 VS 에 병합하기 위해 만족시켜야 하는 조건이며, 다음과 같이 정의된다:

$$dist(P_k, P_{k+1}) \leq \tau_{dist}$$

위의 조건을 만족하면 비디오 세그먼트 내의 임의의 두 개의 인접한 점들의 쌍 사이의 거리는 시퀀스 내의 모든 인접한 두 점 쌍의 거리의 평균 보다 작거나 같게 된다. 이것은 비디오 세그먼트 내의 인접한 프레임들은 상당한 정도의 유사성을 가져야 함을 의미한다.

4.3 비디오 세그멘테이션 알고리즘

비디오 세그먼트에 어떤 점을 병합하는 것은 기하학적 바운딩 조건과 의미적 바운딩 조건을 만족하는 경우에만 허용된다. $sid \leftarrow NewSID()$, $SP \leftarrow P_i$, $HR \leftarrow HR(P_i)$ 로 치환함으로써 편의상 점 P_i 를 비디오 세그먼트 형식, $\langle NewSID(), P_i, HR(P_i) \rangle$,으로 나타낼 수 있다. 여기

알고리즘 MERGE_POINT

입력: 비디오 세그먼트 VS_{in} , 점 P_i

출력: 비디오 세그먼트 VS_{out}

단계 0: $VS_{out} \leftarrow VS_{in}$ (비디오 세그먼트에 점을 병합)

$VS_{out}.sid \leftarrow VS_{in}.sid$

$VS_{out}.SP \leftarrow VS_{in}.SP$

$VS_{out}.HR \leftarrow VS_{in}.HR \oplus HR(P_i)$

단계 1: return VS_{out}

그림 4 알고리즘 MERGE_POINT

알고리즘 VIDEO_SEGMENTATION

입력: MDS S (K 개의 점 포함), 비디오 세그먼트 당 최소 점 수 $minPts$

출력: 비디오 세그먼트 집합 \mathcal{VS} , 주변점 집합 \mathcal{O}

단계 0: 초기화

$\mathcal{VS} \leftarrow \emptyset$, $\mathcal{O} \leftarrow \emptyset$

compute τ_{vol} , τ_{edge} , and τ_{dist} for S .

$VS_{current} \leftarrow VS(First\ point\ P_i\ of\ S)$

단계 1: 비디오 세그먼트 생성

for each successive point P ($2 \leq j \leq K$) of S ,

if $\Delta Vol(VS_{current}, HR, HR(P)) \leq \tau_{vol} \wedge$

$\Delta Edge(VS_{current}, HR, HR(P)) \leq \tau_{edge} \wedge$

$dist(P_{j-1}, P) \leq \tau_{dist}$ then

$VS_{current} \leftarrow MERGE_POINT(VS_{current}, P)$

else

if $VS_{current}.HR.k \leq minPts$ then

$\mathcal{O} \leftarrow \mathcal{O} \cup \{all\ points\ in\ VS_{current}\}$

else

$\mathcal{VS} \leftarrow \mathcal{VS} \cup \{VS_{current}\}$

$VS_{current} \leftarrow VS(P)$

end if

end if

end for

단계 2: return set \mathcal{VS} , set \mathcal{O}

그림 5 알고리즘 VIDEO_SEGMENTATION

에서 $NewSID()$ 는 비디오 세그먼트의 sid 를 생성하는 함수이다. 점 P_i 로 이루어진 비디오 세그먼트는 $VS(P_i)$ 로 표시된다. 한 점의 병합에 관한 알고리즘 $MERGE_POINT$ 와 비디오 세그멘테이션 알고리즘 $VIDEO_SEGMENTATION$ 이 그림 4와 5에 기술되어 있다.

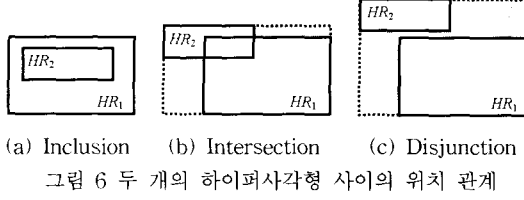
5. 비디오 클러스터링

MDS로부터 비디오 세그먼트를 생성한 후, 공간적으로 가까운 세그먼트 들은 3.3 절에서 정의된 클러스터링 품질을 높이기 위해 병합될 필요가 있다. 비디오 세그먼트나 비디오 클러스터를 표현하는 두 개의 하이퍼 사각형들을 병합할지의 여부를 결정하는 문제는 중요하다. 이러한 병합은 사전에 정의된 조건을 만족하는 경우에만 허용된다. 다음은 비디오 클러스터의 정의이다.

정의 5 (비디오 클러스터). r 개의 비디오 세그먼트 VS_j ($j = 1, 2, \dots, r$)를 시간 순서로 포함하고 있는 비디오 클러스터 VC 를 다음과 같이 정의한다: $VC = \langle cid, slist, HR \rangle$. 여기에서, cid 는 cluster-id이고, $slist$ 는 클러스터 내의 VS 들의 시간 순서를 고려한 리스트이며, $HR = (L, H, k)$ 이다. 단, $L = \{(L^1, L^2, \dots, L^n) \mid L^i = \min_{1 \leq j \leq r} (VS_j.HR.L^i)\}$, $H = \{(H^1, H^2, \dots, H^n) \mid H^i = \max_{1 \leq j \leq r} (VS_j.HR.H^i)\}$ ($i = 1, 2, \dots, n$), $k = \sum_{1 \leq j \leq r} (VS_j.HR.k)$.

5.1 두개의 하이퍼사각형사이의 위치 관계

두 개의 하이퍼사각형을 병합할지의 여부를 결정하기 위하여 그들 사이의 위치 관계를 파악하는 것이 중요하다. 이러한 위치 관계에는 다음의 3가지가 있다: inclusion, intersection, disjunction. 그림 6은 이러한 위치 관계를 보여 준다.



두 개의 하이퍼사각형 HR_1 과 HR_2 을 병합하여 HR_m 이 생성된다고 가정하자. (즉, $HR_m = HR_1 \oplus HR_2$) 그리고 VPP_m 과 EPP_m 은 각각 병합된 경우의 VPP와 EPP 이고, VPP_n 과 EPP_n 은 병합되지 않은 경우의 VPP와 EPP라 하면, 3.3절의 식에 의해 다음이 성립한다.

$$VPP_m = \frac{Vol(HR_1 \oplus HR_2)}{HR_{m,k}}, \quad EPP_m = \frac{Edge(HR_1 \oplus HR_2)}{HR_{m,k}} \quad (a)$$

$$VPP_n = \frac{Vol(HR_1) + Vol(HR_2)}{HR_{1,k} + HR_{2,k}},$$

$$EPP_n = \frac{Edge(HR_1) + Edge(HR_2)}{HR_{1,k} + HR_{2,k}} \quad (b)$$

(a) **Inclusion** ($HR_1 \supseteq HR_2$)

이 경우에는 $Vol(HR_m) = Vol(HR_1)$ 이므로 위의 식 a, b를 사용하여 다음을 유도할 수 있다.

$$VPP_n = \frac{Vol(HR_1) + Vol(HR_2)}{HR_{1,k} + HR_{2,k}} = VPP_m + \frac{Vol(HR_2)}{HR_{m,k}}.$$

$$\text{따라서, } VPP_m = VPP_n - \frac{Vol(HR_2)}{HR_{m,k}}.$$

$$HR_1 \supseteq HR_2 \text{이므로, } 0 \leq \frac{Vol(HR_2)}{HR_{m,k}} \leq \frac{Vol(HR_1)}{HR_{m,k}} \\ = \frac{Vol(HR_m)}{HR_{m,k}} = VPP_m.$$

$$\text{따라서, } \frac{1}{2}VPP_n \leq VPP_m \leq VPP_n.$$

유사하게 $Edge(HR_m) = Edge(HR_1)$ 이므로, 다음을 유도할 수 있다.

$$EPP_n = \frac{Edge(HR_1) + Edge(HR_2)}{HR_{1,k} + HR_{2,k}} \\ = VPP_m + \frac{Edge(HR_2)}{HR_{m,k}}.$$

$$\text{따라서, } EPP_m = EPP_n - \frac{Edge(HR_2)}{HR_{m,k}}, \quad 0 \leq \frac{Edge(HR_2)}{HR_{m,k}} \\ \leq EPP_m \text{이므로 다음이 성립한다.}$$

$$\frac{1}{2}EPP_n \leq EPP_m \leq EPP_n$$

위의 식에 의하여 $VPP_m \leq VPP_n$, $EPP_m \leq EPP_n$ 이므로 병합이 일어난 후의 VPP와 EPP는 일어나지 않은 경우보다 항상 개선된다는 것을 알 수 있다. 그러므로 한 하이퍼사각형이 다른 사각형에 포함되는 경우에는 당연히 병합해야 한다. $VPP_m = VPP_n/2$ 와 $EPP_m = EPP_n/2$ 의 경우는 두개의 하이퍼사각형이 동일한 경우이다.

(b) **Intersection** ($HR_1 \cap HR_2 \neq \emptyset$)

VPP와 EPP에 관하여 병합이 일어나지 않는 경우에 비하여 개선된 클러스터링 품질을 얻기 위해서는 $VPP_m \leq VPP_n$, $EPP_m \leq EPP_n$ 이 성립되어야 한다. 식 a, b에 의하여 다음의 식을 유도할 수 있다.

$$VPP_m = \frac{Vol(HR_m)}{HR_{m,k}} = \frac{Vol(HR_1 \oplus HR_2)}{HR_{m,k}} \\ \leq \frac{Vol(HR_1) + Vol(HR_2)}{HR_{1,k} + HR_{2,k}} = VPP_n \\ EPP_m = \frac{Edge(HR_m)}{HR_{m,k}} = \frac{Edge(HR_1 \oplus HR_2)}{HR_{m,k}} \\ \leq \frac{Edge(HR_1) + Edge(HR_2)}{HR_{1,k} + HR_{2,k}} = EPP_n$$

위의 식에 의해서 개선된 클러스터링 품질을 얻기 위한 조건은 다음과 같다: $Vol(HR_1 \oplus HR_2) \leq Vol(HR_1) + Vol(HR_2)$, $Edge(HR_1 \oplus HR_2) \leq Edge(HR_1) + Edge(HR_2)$. 예지에 관한 조건을 고려해 보자. 두개의 하이퍼사각형이 겹치는 경우에는 두 사각형의 모든 예지는 각 차원에서 겹쳐야 한다. 하이퍼사각형의 기하학적 특성에 따라 다음의 정리가 성립한다.

정리2. 두개의 하이퍼사각형 HR_1 과 HR_2 이 겹치는 경우, 예지에 관한 다음 식이 항상 성립한다

$$Edge(HR_1 \oplus HR_2) \leq Edge(HR_1) + Edge(HR_2)$$

증명. $e_i(HR)$ 을 차원 i 에서 HR 의 예지의 길이라고 하면, 하이퍼사각형이 겹치는 경우에 두 사각형의 모든 예지는 각 차원에서 겹쳐야 하므로 $e_i(HR_1 \oplus HR_2) \leq e_i(HR_1) + e_i(HR_2)$ (단, $i = 1, 2, \dots, n$)이 성립한다. 모든 차원을 고려하면

$$Edge(HR_1 \oplus HR_2) = \sum_{1 \leq i \leq n} e_i(HR_1 \oplus HR_2) \\ \leq \sum_{1 \leq i \leq n} (e_i(HR_1) + e_i(HR_2)) \\ = \sum_{1 \leq i \leq n} e_i(HR_1) + \sum_{1 \leq i \leq n} e_i(HR_2) \\ = Edge(HR_1) + Edge(HR_2).$$

이 되므로 정리2가 성립한다.

정리2에 의해 $Edge(HR_1 \oplus HR_2) \leq Edge(HR_1) + Edge(HR_2)$ 이 항상 성립하므로 개선된 클러스터링 품질을 얻기 위한 조건은 다음과 같이 축소된다: $Vol(HR_1 \oplus HR_2) \leq Vol(HR_1) + Vol(HR_2)$

(c) **Disjunction** ($HR_1 \cap HR_2 = \emptyset$)

두개의 하이퍼사각형이 겹치지 않고 서로 떨어져 있다면 그림 6에 의하여 $Vol(HR_1 \oplus HR_2)$ 의 값이 $Vol(HR_1) + Vol(HR_2)$ 의 값보다 항상 크게 된다. 반면, $Edge(HR_1 \oplus HR_2)$ 의 값과 $Edge(HR_1) + Edge(HR_2)$ 의 값은 상황에 따라 변한다. 식 a, b를 이용해서 다음의 식을 유도할 수 있다.

$$VPP_m = \frac{Vol(HR_m)}{HR_{m,k}} = \frac{Vol(HR_1 \oplus HR_2)}{HR_{m,k}}$$

$$> \frac{Vol(HR_1) + Vol(HR_2)}{HR_{1,k} + HR_{2,k}} = VPP_n$$

클러스터링 품질로써 VPP 와 EPP 둘 다 고려하므로 두개의 하이퍼사각형이 겹치지 않고 서로 떨어져 있는 경우, 이 둘을 병합하면 EPP 의 값에 관계없이 클러스터링 품질이 저하됨을 알 수 있다. 따라서 이 경우에는 병합이 허용되지 않는다. 이상의 3가지 경우를 모두 고려하여 최종적으로 다음의 정리가 성립한다.

정리3. 비디오 세그먼트나 비디오 클러스터의 두개의 하이퍼사각형 HR_1 과 HR_2 이 병합하는 경우, 다음의 조건이 성립하면 VPP , EPP , PPC 에 관하여 병합하지 않는 경우 보다 항상 개선된 클러스터링 품질을 보장한다:

$$Vol(HR_1 \oplus HR_2) \leq Vol(HR_1) + Vol(HR_2)$$

증명. 위의 조건이 성립하면 다음의 식 역시 성립한다:

$$VPP_m = \frac{Vol(HR_1 \oplus HR_2)}{HR_{1,k} + HR_{2,k}}$$

$$\leq \frac{Vol(HR_1) + Vol(HR_2)}{HR_{1,k} + HR_{2,k}} = VPP_n$$

HR_1 과 HR_2 이 겹칠 경우 $Vol(HR_1 \oplus HR_2) \leq Vol(HR_1) + Vol(HR_2)$ 이 성립하므로 정리2로부터 $Edge(HR_1 \oplus HR_2) \leq Edge(HR_1) + Edge(HR_2)$ 가 항상 성립함을 알 수 있다. 따라서 다음의 식을 유도할 수 있다:

$$EPP_m = \frac{Edge(HR_1 \oplus HR_2)}{HR_{1,k} + HR_{2,k}}$$

$$\leq \frac{Edge(HR_1) + Edge(HR_2)}{HR_{1,k} + HR_{2,k}} = EPP_n$$

PPC_m 을 병합된 하이퍼사각형의 PPC , PPC_n 을 병합되지 않은 경우의 PPC 라 하면, 3.3절의 조건식에 의해 PPC_n 은 $(HR_{1,k} + HR_{2,k}) / 2$ 이 되고, PPC_m 은 $(HR_{1,k} + HR_{2,k})$ 이 됨은 명백하다. 따라서, $PPC_m > PPC_n$ 이 되므로 정리3이 성립한다.

정리3은 두개의 하이퍼사각형이 병합될 때, 병합에 의하여 확장된 볼륨 $ExpVol$ (그림 6에서 진하게 표시된

부분)은 두 사각형의 겹친 부분의 볼륨보다 작거나 같아야 함을 의미한다. 즉,

$$Vol(HR_1 \oplus HR_2) = Vol(HR_1 \cup HR_2) + ExpVol$$

$$= Vol(HR_1) + Vol(HR_2) - Vol(HR_1 \cap HR_2) + ExpVol$$

$$Vol(HR_1 \oplus HR_2) - (Vol(HR_1) + Vol(HR_2))$$

$$= (Vol(HR_1 \cap HR_2) + ExpVol$$

정리 3의 식에 의하여 $- Vol(HR_1 \cap HR_2) + ExpVol \leq 0$ 이 되므로 결국 $ExpVol \leq Vol(HR_1 \cap HR_2)$ 이 성립함을 알 수 있다.

5.2 비디오 클러스터링 알고리즘

두개의 하이퍼사각형을 병합하는 것은 정리 3의 조건이 만족될 때에만 허용된다. 비디오 세그먼트 VS 를 $cid \leftarrow NewCID()$, $slist \leftarrow AddItem(VS.sid)$, $HR \leftarrow VS.HR$ 로 치환함으로써 비디오 클러스터 형식 $\langle NewCID(), AddItem(VS.sid), VS.HR \rangle$ 로 표현할 수 있다. 여기에서 $NewCID()$ 는 비디오 클러스터의 cid 를 생성하는 함수이며, $AddItem(VS.sid)$ 는 $slist$ 에 VS 의

알고리즘 MERGE_CLUSTERS

입력: 비디오 클러스터 VC_{IN1} , VC_{IN2}

출력: 비디오 클러스터 VC_{OUT}

단계 0: * 두 비디오 클러스터를 병합 *

$VC_{OUT}.cid \leftarrow NewCID()$

for each item $VS_a.sid$ in $VC_{IN1}.slist$

$VC_{OUT}.slist \leftarrow AddItem(VS_a.sid)$

end for

for each item $VS_b.sid$ in $VC_{IN2}.slist$

$VC_{OUT}.slist \leftarrow AddItem(VS_b.sid)$

end for

$VC_{OUT}.HR \leftarrow VS_{IN1}.HR \oplus VS_{IN2}.HR$

단계 1: **return** VC_{OUT}

그림 7 알고리즘 MERGE_CLUSTERS

알고리즘 VIDEO_CLUSTERING

입력: MDS S_i에 대한 비디오 세그먼트 V_iS 의

출력: 비디오 클러스터 집합 \mathcal{VC}_i

단계 0: * 초기화 *

$\mathcal{VC}_i \leftarrow \emptyset$

for each item VS_i in V_iS :

$VC_i \leftarrow VC(VS_i)$

$\mathcal{VC}_i \leftarrow \mathcal{VC}_i \cup \{VC_i\}$

end for

단계 1: * 비디오 클러스터 생성 *

for each pair (VC_a, VC_b) in \mathcal{VC}_i :

if $Vol(VC_a.HR \oplus VC_b.HR) \leq Vol(VC_a.HR) + Vol(VC_b.HR)$ **then**

$VC_c \leftarrow MERGE_CLUSTERS(VC_a, VC_b)$

$\mathcal{VC}_i \leftarrow \mathcal{VC}_i - \{VC_a, VC_b\}$

$\mathcal{VC}_i \leftarrow \mathcal{VC}_i \cup \{VC_c\}$

end if

end for

단계 2: **return set** \mathcal{VC}_i

그림 8 알고리즘 VIDEO_CLUSTERING

sid를 추가하는 함수이다. 비디오 세그먼트 VS로부터 생성된 비디오 클러스터는 VC(VS)로 표시된다. 그림 7의 알고리즘 MERGE_CLUSTERS는 두개의 클러스터를 병합하는 알고리즘이며, 그림 8의 알고리즘 VIDEO_CLUSTERING은 알고리즘 MERGE_CLUSTERS를 이용하여 클러스터링을 수행한다.

5.3 통합된 알고리즘과 복잡도

그림 9의 알고리즘은 그림 5에서 서술된 VIDEO_SEGMENTATION 알고리즘과 그림 8에서 서술된 VIDEO_CLUSTERING 알고리즘을 통합한 알고리즘으로서 MDS들의 데이터 집합에서 클러스터들을 구한다. 이 알고리즘은 두개의 입력 인수으로써 데이터 집합과 주변점을 결정하는 데 사용하는 minPts를 취하여 주변점 집합과 비디오 클러스터 집합을 출력한다.

```

알고리즘 VDCluster
입력: 시퀀스 집합 MDS's, minPts
출력: 비디오 클러스터 집합 VC, 주변점 집합 O
단계 0: * 초기화 *
        VC ← ϕ, O ← ϕ
단계 1: * 비디오 클러스터링 과정 *
        for each MDS Si in the data set (1 ≤ i ≤ N)
            VSi: Oi ← VIDEO_SEGMENTATION(Si, minPts)
            O ← O ∪ Oi
            VCi ← VIDEO_CLUSTERING(VSi)
            VC ← VC ∪ VCi
        end for
단계 2: return set VC, set O
    
```

그림 9 알고리즘 **VDCluster**

그림 9의 알고리즘VDCluster의 복잡도는 데이터베이스의 시퀀스들의 개수 N에 대해서 분명히 O(N)이다. 그림 5의 VIDEO_SEGMENTATION 알고리즘에서 보면, 각 시퀀스의 K개의 점들에 대해서, 단계0에서는 한계값(threshold value)을 구하는데 O(K)의 계산이 필요하고, 단계1에서도 역시 for-루프에서 O(K)의 계산이 필요하다. 비디오 세그먼트나 클러스터의 개수는 K에 비하면 무시할 수 있을 정도로 작고 가변이므로 고려하지 않아도 된다. 결론적으로 전체 통합된 알고리즘의 복잡도는 NK에 비례하므로 O(NK)이다.

6. 실험

본 논문에서 제안된 방법의 효용성을 평가하기 위해서 TV 뉴스, 드라마, 만화 영화와 같은 다양한 실제 비디오 데이터를 사용하여 실험하였다. 또한 제안된 방법이 얼마나 양질의 클러스터를 찾는지 측정하는 척도로서 3.3절에서 정의된 측정 기준을 사용하였다. 본 절에서는 실험

환경을 서술하고 실험 결과를 보이고 분석한다.

6.1. 실험 환경

실험 데이터로서 사용하기 위해서 다양한 비디오 데이터로부터 서로 길이가 다른 비디오 클립들을 만들고 각 클립의 프레임에서 RGB 색상 특징을 추출했다. RGB 색상 특징은 다차원의 벡터로서 표현되고, 한 개의 프레임은 다차원 공간의 한 점으로 사상(mapping)되므로 하나의 비디오 클립은 MDS로 표현된다. 실험에서는 편의를 위해서 3차원 데이터를 사용하였으나 본 논문에서 제안한 방법은 데이터의 차원에 제약을 두지 않고 있다. 본 실험의 데이터는 30 프레임에서 9000 프레임까지의 서로 다른 길이의 비디오 클립으로 구성되어 있으며 (1초에서 5분까지의 비디오 클립), 각 클립은 3차원 시퀀스로 구성되어 있고, 전체 6,984개의 비디오 클립이 사용되었다. 표 1은 이들 데이터들을 요약한 내용을 보여주고 있다.

표 1 실험에 사용된 데이터

데이터명	비디오 클립의 길이 (프레임수)	클립의 개수
v1	30 < L ≤ 500	1,527
v2	500 < L ≤ 1000	1,216
v3	1000 < L ≤ 2000	1,032
v4	2000 < L ≤ 3000	836
v5	3000 < L ≤ 4000	645
v6	4000 < L ≤ 5000	508
v7	5000 < L ≤ 6000	415
v8	6000 < L ≤ 7000	336
v9	7000 < L ≤ 8000	262
v10	8000 < L ≤ 9000	207
합		6,984

6.2. 실험 결과

본 실험에서는 MDS를 클러스터링 하기 위해서 두가지의 클러스터링 요소(기하학적 요소와 의미적 요소)를 고려하였다. 비디오 세그먼트와 클러스터의 기하학적인 특성을 고려한 클러스터링의 효과를 측정하기 위해서, 3.3절에서 정의된 클러스터링 품질 측정 기준인 VPP, EPP, PPC를 사용하였다. 기하학적인 특성 외에도 세그먼트 안의 연결된 점들 사이의 평균 거리인 MeanDist를 사용하여 비디오 세그먼트 안의 점들의 의미적 관계를 측정하였다. k개의 점 P_j(j=1,2,...,k)을 가진 비디오 세그먼트 VS에서 인접한 점들 사이의 유클리드 거리의 합(DistSum_{VS})은 다음과 같이 된다:

$$DistSum_{VS} = \sum_{1 \leq j \leq k-1} dist(P_j, P_{j+1})$$

MDS S가 p개의 비디오 단편 VS_h ($h = 1, 2, \dots, p$) 들로 구성되어 있을 때, S에 대한 MeanDist 는 다음과 같이 정의된다:

$$MeanDist_S = \frac{\sum_{1 \leq h \leq p} DistSum_{VS_h}}{\sum_{1 \leq h \leq p} (VS_h \cdot HR \cdot k - 1)}$$

본 논문과 관련된 알고리즘들 [1, 18]은 [17]에서 제안된 MCOST에 기초하여 약간 수정된 것들이기 때문에, 본 실험에서는 제안된 방법과 MCOST를 서로 비교하였다. 그림 10-13은 실험 결과를 보여 주고 있다. 본 논문에서 제안한 세그멘테이션과 클러스터링의 결과는 각각 V_SEQ 와 V_CL 로 나타내고, MCOST에 의한 결과는 $MCOST$ 로 나타낸다.

한 점 당 볼륨(VPP): 그림 10은 서로 다른 길이의 비디오 데이터에 대해서 각 알고리즘에 의해서 생성된 하이퍼사각형에 대해서 한 점 당 볼륨(VPP)을 보여주고 있다. 그림에서 비디오 클립의 길이가 증가함에 따라서 V_SEQ 와 V_CL 의 VPP가 $MCOST$ 의 것보다 현저히 감소하는 것을 볼 수 있다. 길이가 긴 시퀀스에 대해서 (v7~v10), V_CL 의 VPP가 $MCOST$ 의 VPP의 29%에서 42% 정도 된다. 이것은 제안한 방법이 세그멘테이션 과정에서 각각의 시퀀스의 특성을 잘 반영하기 때문이다. 즉, 세그멘테이션 과정에서 기하학적 경계 조건을 구하기 위해 사용되는 단위 하이퍼큐브는 시퀀스의 길이가

증가함에 따라 밀도가 높아지는 경향이 있기 때문이다.

그 이유는 비디오 클립이 길어질수록 비슷한 세그먼트를 가질 확률이 높아지는 현상이 있기 때문이다. 예를 들면, 긴 뉴스 비디오 클립은 같은 앵커가 나타나는 유사한 장면이 많을 수 있다. 그리고 비디오 클립의 볼륨이 세그멘테이션을 위한 경계 값으로 사용되기 때문에, 비디오 클립의 단위 하이퍼 큐브의 밀도가 높아지면 그 비디오 클립이 밀도가 높은 세그먼트를 많이 가지는 것은 당연하다. 뿐만 아니라 제안된 기법은 주변점을 적절히 다룸으로써 밀도가 높은 세그먼트들을 생성해 내는데 기여하고 있다. 한편, V_CL 의 VPP는 비디오 클러스터가 정리 3에 기초하여 생성되므로 V_SEQ 의 VPP보다 대체로 좋다는 것을 알 수 있다.

한 점 당 에지의 길이(EPP): 그림 11은 한 점 당 하이퍼사각형의 에지의 길이(EPP)에 대해서, $MCOST$, V_SEQ , V_CL 간에 현저한 차이를 보여 주고 있다. V_CL 의 EPP는 $MCOST$ 의 EPP의 52%에서 79% 정도이고, V_SEQ 의 EPP는 $MCOST$ 의 EPP의 78%에서 89% 정도이다. 이러한 향상된 결과는 제안한 방법이 볼륨뿐만 아니라 중요한 클러스터링 요인 중의 하나인 에지의 길이도 고려하기 때문이다. 또한 제안한 방법이 주변점을 적절히 다룸으로써 더욱 향상된 결과를 보여 주고 있다. V_SEQ 와 V_CL 의 EPP는 비디오 클립의 길이가 길어지면 작아지는데, 이것은 VPP의 경우에서와 같은 이유 때문이다. V_CL 의 EPP는 V_SEQ 의 EPP의

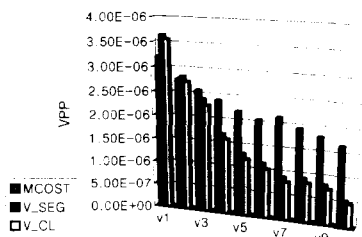


그림 10 한 점 당 볼륨의 비교

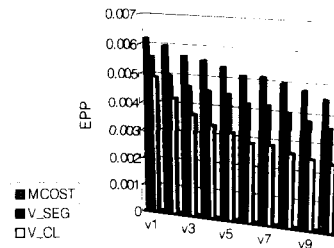


그림 11 한 점 당 에지 길이의 비교

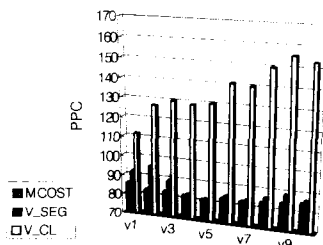


그림 12 한 클러스터 당 점의 개수

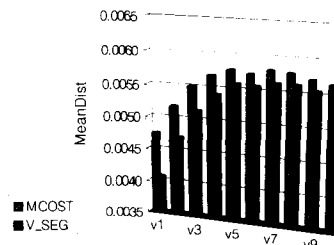


그림 13 세그먼트 안의 연속된 점들 사이의 평균 거리

66%에서 88% 정도이다. 반면에 V_CL 의 VPP는 V_SEQ 의 VPP에 비해서 약간 좋아진 것을 볼 수 있는데 이것은 비디오 클러스터링 과정에서 비디오 세그먼트들을 합치는 것이 VPP보다는 EPP를 더 많이 개선시킨다는 것은 시사한다.

한 클러스터 당 점의 개수(PPC): 앞에서 언급한 VPP와 EPP와 관련하여, 그림 12는 한 클러스터당 점의 개수(PPC)를 보여준다. 그림에서 볼 수 있듯이, V_SEQ 의 PPC는 모든 비디오 데이터에 대해서 $MCOST$ 의 PPC에 비해 약간 향상되어 있는 것을 볼 수 있다. 그러나 여러 개의 세그먼트들을 한 개의 클러스터로 합치는 클러스터링 과정은 PPC를 매우 향상시킨다. V_CL 의 PPC는 $MCOST$ 의 것 보다 1.30배에서 1.85배 정도 더 좋다.

VPP, EPP와 PPC에 관련된 실험 결과로부터, 제안된 방법이 불륨과 에지의 길이에 관하여 $MCOST$ 방법보다 밀도가 높은 클러스터를 생성해 낸다는 결론을 내릴 수 있다. 그리고 클러스터의 밀도는 비디오 클립의 길이가 길어질수록 더 높아진다.

세그먼트 안의 점 사이의 평균 거리: 그림 13은 세그먼트 안의 점들의 의미적인 관계를 나타내 주는 척도로 사용되는 $MeanDist$ 를 보여주고 있다. 세그먼트 안의 점들간의 $MeanDist$ 가 작다는 것은 그 값이 큰 경우보다 점들 간의 의미적인 관계가 더 가깝다는 것을 가리킨다. V_SEG 의 $MeanDist$ 는 모든 비디오 데이터에 대해서 $MCOST$ 의 85%에서 97% 정도인데, 이 사실은 제안된 방법으로 생성한 비디오 세그먼트의 점들 간의 의미적인 관계가 $MCOST$ 보다 더 가깝다는 것을 설명해 주고 있다. 이러한 향상된 결과는 제안한 방법이 정의 4에서 서술된 중요한 클러스터링 요소인 의미적 경계 조건을 고려하기 때문이다. 이에 비해서 $MCOST$ 는 이 조건을 전혀 고려하고 있지 않다.

7. 결론

비디오 데이터의 검색은 비록 폭 넓게 연구되어 있지는 않지만, 데이터 베이스 응용에서 가장 연구 가능성이 많은 영역 중의 하나이다. 비디오 데이터를 효율적으로 검색하기 위해서, 클러스터링 과정은 비디오 데이터를 효과적으로 표현하고 인덱싱하고 저장하는 기초 과정으로써 필수적이다. 본 논문에서는 큰 규모의 비디오 데이터에 대한 세그멘테이션과 클러스터링에 대하여 연구하였다. 이 문제를 해결하기 위해서 먼저 클러스터들의 기하적인 특성과 의미적인 특성을 고려한 클러스터링 요소들을 논의하였다. 그리고 클러스터링의 품질을 평가하기 위한 기준을 정의하였다. 이러한 클러스터링 요소와

품질 기준에 기초하여, 본 논문은 다음과 같은 바람직한 특성을 가지는 효과적인 클러스터링 방법을 제안하였다:

- 비디오 클러스터 안의 요소들 사이의 시간적이고 의미적인 관계를 유지 관리한다.
- 불륨과 에지에 관하여 미리 정의된 기준을 만족하는 밀도가 높은 클러스터를 생성한다.
- 주변점을 적절히 찾아내고 이후의 검색 작업의 효율을 높이기 위하여 주변점들을 비디오 클러스터와 별도로 다룰 수 있다.
- 클러스터링에 필요한 대부분의 입력 인수들이 비디오 자체의 특성으로부터 결정되고 사용자가 제공하지 않는다.

제안된 방법에 대하여 다양한 비디오 데이터를 사용하여 실험을 하였고, 클러스터링의 품질을 평가하였다. 제안된 방법이 VPP, EPP, PPC 그리고 $MeanDist$ 에 관하여 상당히 효율성이 있다는 사실을 실험 결과에서 확인하였다. 향후의 연구 과제로서 본 논문에서 제안한 비디오 세그먼트와 클러스터에 기초하여 대용량 비디오 데이터에 대한 유사성 검색 문제를 연구할 계획을 가지고 있다.

참고 문헌

- [1] S. L. Lee, S. J. Chun, D. H. Kim, J. H. Lee, and C. W. Chung, Similarity search for multidimensional data sequences, *Proceedings of IEEE Int'l Conference on Data Engineering*, pages 599-608, 2000.
- [2] A. Guttman, R-trees: a dynamic index structure for spatial searching, *ACM SIGMOD*, pages 47-57, 1984.
- [3] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, The R⁺-tree: an efficient and robust access method for points and rectangles, *ACM SIGMOD*, pages 322-331, 1990.
- [4] S. Berchtold, D. Keim, and H. Kriegel, The X-tree: an index structure for high-dimensional data, *Proceedings of VLDB*, pages 28-39, 1996.
- [5] T. Sellis, N. Roussopoulos, and C. Faloutsos, The R⁺ tree: a dynamic index for multi-dimensional objects, *Proceedings of VLDB*, pages 507-518, 1987.
- [6] R. T. Ng and J. Han, Efficient and effective clustering methods for spatial data mining, *Proceedings of VLDB*, pages 144-155, 1994.
- [7] H. J. Zhang, C. Y. Low, S. W. Smoliar, Video parsing and browsing using compressed data, *Multimedia Tools and Application 1*, pages 89-111, 1995.

- [8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, *Int'l Conference on Knowledge Discovery in Databases and Data Mining*, pages 226-231, Portland, Oregon, 1996.
- [9] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, *ACM SIGMOD*, pages 94-105, 1998.
- [10] S. Guha, R. Rastogi, and K. Shim, CURE: An efficient clustering algorithm for large databases, *ACM*, pages 73-84, 1998.
- [11] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park, Fast algorithms for projected clustering, *ACM SIGMOD*, pages 61-72, 1999.
- [12] D. DeMenthon, V. Kobla, and D. Doermann, Video summarization by curve simplification, *ACM Multimedia*, pages 211-218, Bristol, UK, 1998.
- [13] B. Günsel, A. M. Ferman, and A. M. Tekalp, Video indexing through integration of syntactic and semantic features, *Proceedings of IEEE Workshop on Applications of Computer*, pages 90-95, 1996.
- [14] A. Hampapur, R. Jain, and T. Weymouth, Digital video segmentation, *ACM Multimedia*, pages 357-364, 1994.
- [15] T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: An efficient data clustering method for very large databases, *ACM SIGMOD*, pages 103-114, 1996.
- [16] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, Query by image and video content: the QBIC system, *IEEE Computer*, Vol. 28, No. 9, pages 23-32, 1995.
- [17] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, Fast subsequence matching in time-series databases, *ACM SIGMOD*, 1994.
- [18] V. Kobla, D. Doermann, and C. Faloutsos, Video Trails: Representing and visualizing structure in video sequences, *Proceedings of ACM Multimedia*, pages 335-346, Seattle, Washington, 1997.
- [19] S. L. Lee, and C. W. Chung, On the effective clustering of multidimensional data sequences, *Information Processing Letters*, Vol.80, pages 87-95, 2001.



이 석 홍

1984년 연세대학교 기계공학과 학사. 1993년 동대학원 산업공학과 전자계산전공 석사. 2001년 한국과학기술원 정보및통신공학과 박사. 1984년 ~ 1995년 한국IBM 선임연구원. 현재 한국의국어대학교 산업정보시스템공학부 조교수. 관심분야는 데이터웨어하우스, 데이터마이닝, 멀티미디어 데이터베이스, 정보검색



이 주 홍

1983년 서울대학교 컴퓨터공학과 학사. 1985년 동대학원 석사. 2001년 한국과학기술원 박사. 한국통신 전임연구원, 한국IBM 선임연구원, 현재 인하대학교 컴퓨터공학부 조교수. 관심분야는 데이터웨어하우스, 데이터마이닝, 멀티미디어, XML, 정보검색



김 덕 환

1987년 서울대학교 계산통계학과(학사). 1995년 한국과학기술원 정보및통신공학과(석사). 1995년 ~ 현재 한국과학기술원 박사과정. 1987년 ~ 1997년 2월 LG전자(주) 통신기기연구소 선임연구원. 1997년 3월 ~ 현재 동양공업전문대학 인터넷정보과 조교수. 관심분야는 멀티미디어 데이터베이스, 데이터마이닝, 웹정보검색



정 진 완

1973년 서울대학교 공과대학 전기공학과(학사). 1983년 University of Michigan 컴퓨터공학과(박사). 1983년 ~ 1993년 미국 GM 연구소 전산학과 선임연구원 및 책임연구원. 1993년 ~ 현재 한국과학기술원 전산학과 부교수 및 교수. 관심분야는 XML, 멀티미디어 데이터베이스, GIS, 웹 정보검색, 객체지향 데이터베이스