Journal of The Korea Society of Computer and Information
Vol. 21 No. 9, pp. 37-43, September 2016

www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.9.037

Shot Group and Representative Shot Frame Detection using Similarity-based Clustering

Gye-Sung Lee *

Abstract

This paper introduces a method for video shot group detection needed for efficient management and summary of video. The proposed method detects shots based on low-level visual properties and performs temporal and spatial clustering based on visual similarity of neighboring shots. Shot groups created from temporal clustering are further clustered into small groups with respect to visual similarity. A set of representative shot frames are selected from each cluster of the smaller groups representing a scene. Shots excluded from temporal clustering are also clustered into groups from which representative shot frames are selected. A number of video clips are collected and applied to the method for accuracy of shot group detection. We achieved 91% of accuracy of the method for shot group detection. The number of representative shot frames is reduced to 1/3 of the total shot frames. The experiment also shows the inverse relationship between accuracy and compression rate.

▸ Keyword : Shot Detection, Video Clustering, Key Frame, Similarity Distance

I. Introduction

명시적으로 구조화되지 않았으나 복합적인 형태를 지닌 비디오 데이터는 컴퓨터 네트워크와 하드웨어의 발전과 함께 광범위하게 사용되어 왔다. 특히 Web 2.0의 폭발적인 사용 확대와 함께 온라인 비디오가 광범위하게 사용되었고 YouTube, Google Video, Dailymotion 등 많은 온라인 서비스 제공자를 통해 비디오 공유가 확대돼 가고 있다[1]. 다양한 시각적 특징을 가진 비디오 콘텐츠는 사용자에게 편리한 정보를 손쉽게 제공하지만 비디오 자료에 대한 분석 및 관리는 그만큼 용이하지는 않다.

비디오 검색엔진을 통해 비디오를 검색할 때 사용자는 질의 키워드를 제공하고 이에 대하여 검색엔진은 연관도 점수를 계산하고 그에 따라 검색된 비디오를 서열화하여 리스트로 나열한다. 많은 검색엔진은 이를 위해 사전처리로 각 비디오에 대한 태그와 주석(annotation)을 붙여 이것을 조사하여 검색하는 키워드 방식이 주를 이룬다. 관심 비디오를 빠르게 검색하여 적절하게 추천해 주는 장점이 있지만 비디오 검색을 위한 키워드, 메타 데이터 등 주석정보가 저장되어야 한다. 키워드 방식이 아니고 특정 장면의 샷 프레임을 제시하여 검색해야 하는 경우와 같은 내용기반 검색은 사용자의 심도 있는 요구를 충족하고 사용자에게 편리함을 제공하지만 비디오 자료의 방대함에 비해 비디오 자체의 내부 구조는 특별한 것이 없어 검색이 어려워진다. 비디오의 특정 이미지를 통한 시각적 검색이 가능하려면 각 비디오를 전수조사를 해야 하는데 이 방법은 매우 비효율적이다. 보다 효율적인 방법은 비디오 클립을 비교 가능한 프레임 단위로 축약하는 방법을 제공하거나 비디오를 대표 샷 프레임으로 재구성하여 이들과 비교하여 검색하는 방안을 통해 해결할 수 있다[2,4]. 이와 같은 비디오 요약은 유사하거나 반복되는 비디오 클립을 제거하여 비디오를 간략하게 축소할 수 있는 장점을 제공한다.

본 연구에서는 비디오 클립을 세부 구성요소인 샷으로 구분하고 이들 샷 간의 유사도를 조사하여 하나로 묶는 클러스터링을 수행하고 각 클러스터를 대표하는 샷을 찾아 이들 대표 샷 프레임으로 비디오를 요약하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 비디오 데이터를 샷으로 나누거나 샷 그룹으로 나누는데 사용되는 유사도 측정 방법과 샷, 장면 검출에 관한 관련연구를 소개하고, 3장에서는 비디

• First Author: Gye-Sung Lee, Corresponding Author: Gye-Sung Lee
*Gye-Sung Lee(gslee@dankook.ac.kr), Dept. of Computer Science, Dankook University
• Received: 2016. 07. 26, Revised: 2016. 08. 18, Accepted: 2016. 09. 07.

오 데이터를 세부적으로 나누어 그룹화 하는 클러스터링 방법을 제시하고 대표 샷을 찾는 방법에 대해 설명한다. 실험을 통한 그룹 검출 결과도 제시한다. 4장에서 결론을 맺는다.

II. Related Studies

비디오는 크게 프레임, 샷, 장면, 비디오 클립의 4가지의 계층적이며 순차적인 데이터의 모음으로 구성되어 있다. 여기서 프레임이란 비디오 구성의 최하위 계층이며 하나의 정지영상이다. 프레임의 상위 계층인 샷은 하나의 카메라로 기록된 연속적인 프레임의 모음이다. 장면은 주제가 같은 내용을 가진 인접한 샷의 모음이다. 비디오 클립은 연관된 장면들로 구성되어 진다. 연속된 프레임으로 이뤄진 샷을 중심으로 유사한 것들을 묶어 그룹으로 지정하여 의미 있는 장면을 구성하는데 이 과정에 비디오 클러스터링이 적용된다. 클러스터링의 주요 연산은 프레임 또는 샷 간 유사도를 계산하는 것이다.

1. Similarity

비디오 데이터를 비교할 때 비디오 데이터를 구성하는 데이터의 수준에 따라 프레임 간 비교가 될 수도 있고 또는 더 높은 수준인 샷 간, 장면 간, 또는 비디오 간 비교가 될 수 있다. 비교 수준이 결정되면 이들을 비교할 수 있는 측도가 정의되어야 한다. 프레임 단위의 유사도를 생각한다면 프레임별로 유사도를 계산하여 이들의 합을 구하여 최대 유사도를 계산한다. 비디오 또는 샷 단위로 비교할 시에는 스트링 비교방법의 최대 공통 서열을 찾는 문제(LCS: Longest Common Substring)로 해결할 수 있다. 만일 길이가 서로 다른 임의의 비디오 수에 대하여는 NP-hard 문제이지만 시퀀스 수가 고정되어 있을 경우 동적 프로그래밍으로 다항 시간에 해결할 수 있다. 그러나 이마저도 프레임이 보통 다차원 데이터이기 때문에 이와 같은 스트링 비교방법은 효과적이지 않다.

프레임 수준에서 공통된 부분을 찾기보다는 비디오 데이터의 부분을 대표할 수 있는 대표 속성을 찾아 비교하는 방법을 고려할 수 있다[2,3,4]. 이 방법은 샷으로 구분되는 데이터의 대표되는 프레임을 대상으로 유사도를 측정하기 때문에 매우 유사하거나 반복 검출된 프레임 수준의 유사도 검사 대신 샷 간 유사도를 계산할 수 있어 고수준의 비디오 데이터 검색에 대해 빠른 처리가 가능해 진다[2]. IBM의 QBIC(Query by Image Content)[9]는 이미지의 컬러 히스토그램을 사용하여 유사한 이미지를 검색하는 시스템이다. 컬러 이외에도 질감, 형태, 키워드 등을 이용하여 검색하는 시스템으로 빠른 검색을 보장한다. 초기형태의 이미지 검색 시스템인 QBIC는 질의 이미지를 제시하면 데이터베이스에 있는 정지 이미지나 비디오의 특징들을 비교하여 일치도에 따라 리스트를 화면에 제시하는데 주로 데이터베이스에 있는 이미지나 비디오를 전처리하여 특징들을 추출하여 메타 데이터 형식으로 저장해 놓는다.

본 연구에서는 유사도 측정에 저수준 특징인 컬러 히스토그램을 사용한다. 같은 길이의 두 비디오 X, Y 간 거리 $d(X, Y)$는 식(1)과 같이 정의되고 비유사도 정도를 의미한다.

$$d(X, Y) = \|H^X - H^Y\| = \sqrt{\sum_k (x_k - y_k)^2} \quad (1)$$

H^X, H^Y는 비디오 X, Y의 모든 프레임에 대한 히스토그램 대푯값을 나타내고 x_k 는 비디오 X에 대한 개별 히스토그램 bin의 값을 나타낸다.

2. Keyframe Detection and Video Shot Clustering

하나의 비디오 클립이 주어질 때 비디오는 계층 구조로 표현될 수 있다. 프레임에서 장면으로 이어지는 계층구조가 비디오에 내포되어 있기 때문이다. 계층적 비디오 표현 방식을 채택하여 비디오 표현, 인덱싱, 브라우징, 검색을 용이하게 하는 연구가 많이 발표되어 있다[3,6,7]. 비디오 파싱 절차를 통해 비디오 클립이 장면(scene)으로 나눠지고 장면은 몇 개의 키 프레임으로 구성된 샷으로 나눠진다. 가장 높은 수준에 있는 장면은 해변가, 식당에서의 대화, 결혼식 등으로 구분되어지는 샷으로 이뤄져 있다. 키 프레임은 샷의 의미를 내포하는 최소한의 프레임으로 이뤄지는데 샷 검출 알고리즘이 적용되어 샷을 구분한다[5,6]. 보통 하나의 비디오 클립은 다수의 샷으로 구성되는데 비디오에 따라 다르지만 영화나 드라마의 경우 5분 분량의 비디오에는 평균 50~100개의 샷이 있다고 알려져 있다. 샷 당 1개에서 3개의 키 프레임을 가정하면 총 50~300개의 키 프레임이 있다. 이들을 유사도 기반 클러스터링으로 묶으면 이보다 훨씬 적은 샷으로 비디오를 표현할 수 있으며 장면도 계층적으로 구성하여 표현할 수 있다[7].

비디오 데이터의 구조적 분석에는 샷 경계 검출, 키 프레임 추출, 장면 검출을 포함한다. 샷 검출을 위한 프레임의 시각적 특징에 모션 벡터[10], 컬러 히스토그램, 블록 컬러 히스토그램[1,2,16], 윤곽선 변화율[4,5,18]을 사용한다. 조명이 바뀌는 장면이나 큰 모션이 일어나는 화면의 경우 윤곽선 분석이 좀 더 유리할 수 있다. 샷 검출에 상호정보(Mutual Information, MI)와 결합(joint) 엔트로피를 사용한 연구에서는 한 프레임에서 다른 프레임으로 전이하는데 요구되는 정보량을 측정하여 급격히 변화하는 프레임을 찾는다[11]. 두 프레임 사이에 내용상 큰 변화가 있을 경우 상호 의존도가 낮아 MI 값이 작아진다.

샷을 구성하는 프레임 중 샷을 대표할 수 있는 키 프레임을 추출하는 여러 알고리즘이 있다. [12]는 샷의 모든 프레임을 다차원 공간에서 점으로 표현하여 이들을 잇는 곡선을 생성한다. 이 곡선을 가장 잘 표현할 수 있는 일련의 제어점들로 비디오를 요약하는 방식이다. 이 제어점에 해당하는 프레임들로 키 프레임을 지정한다. 다른 방법으로 클러스터링을 이용하여 키 프레임을 찾는 방법이 있다[1,2,7,8,13,16]. 샷이 검출된 이후 샷에 포함된 프레임들로 클러스터의 중심을 찾고 이 중심에 가

장 가까운 프레임을 키 프레임으로 설정하는 방식이다[13]. 이 방법의 장점은 샷의 길이에 관계없이 단일 프레임으로 샷을 요약할 수 있다는 점이다. 더불어 불필요한 반복 프레임을 제거할 수 있다.

샷 그룹 검출은 스토리와 의미를 담은 구간을 찾는 것이며 보통 여러 샷으로 연계된 그룹을 형성한다. 이 샷 그룹은 의미상의 수준을 형성한다. 연속적인 샷으로 역어지며 샷을 대표하는 키 프레임을 비교하여 유사한 것으로 묶어 샷 그룹을 검출하는 방식이 주를 이룬다. 모션 궤적을 추적하여 이를 시간적으로 나열하여 일련의 키 프레임으로 설정하고 변화가 급격히 이뤄지는 시점에서 샷 그룹을 마감하는 방식이 [14]에 소개되어 있다. 이 경우 샷의 모든 차원의 내용을 모두 포함시키지 못할 경우가 발생한다. 배경을 분석하여 장면으로 인식하는 방식도 [15]에 제시되었다. 모자이크 기술로 프레임을 쪼개서 배경 부분을 인식한다. 이때 사용하는 시각적 특징으로 색상, 질감, 등을 사용하고 배경에 대한 이 특징의 분포를 추정하여 샷 그룹에 속할지 여부를 결정하는 방식이다. 장면 탐지를 위해 다단계 과정을 거치거나 복합적인 특징을 활용하는 경우가 있다. [16]에서는 컬러 히스토그램을 통해 샷을 일차로 분류하고 프레임에 속한 객체를 식별하여 객체별 컬러 히스토그램 통해 2차로 분류하여 장면을 구분하는 방법을 제시하였다.

Fig. 1은 본 연구에서 수행하는 비디오 데이터 클러스터링 시스템의 블록 다이어그램을 보여준다. 시스템에 프레임들이 입력되면, 프레임별로 특징을 추출하여 이웃하고 있는 프레임과 비교한다. 이때 사용하는 것이 유사도 측정이고 이를 계산하여 샷을 검출한다. 샷들 간 유사도를 구해 샷 그룹을 검출하는데, 이는 샷들의 그룹은 비디오 내의 한 장면을 형성할 수 있다.

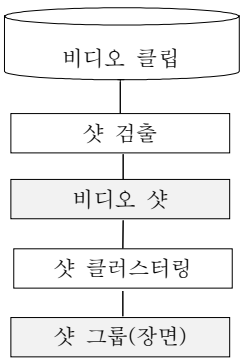

Fig. 1. Steps of Video Clustering

III. Scene Detection by Clustering

1. Shot Detection

샷 검출은 샷 간의 경계를 찾아야하기 때문에 샷 경계 검출이라고도 불린다. 샷의 경계는 비디오 특징 값이 급격히 변화하는 프레임으로 결정된다. 프레임 n에 대한 특징 값$f(n)$을 구하는데 바로 이전 프레임 $n-1$과의 비유사도(dissimilarity)를 측정하는 거리 함수 $d(n-1,n)$로 구한다. 프레임 사이의 비유사도 계산에 프레임의 256차원 HSV 컬러 히스토그램을 사용한다. 두 프레임, f와 g사이의 비유사도인 거리 d는 식(2)와 같이 정의된다[1,2].

$$d(H(f),H(g)) = \sqrt{\sum_{i=0}^{255}(H(f)(i)-H(g)(i))^2} \quad (2)$$

여기서 $H(f)(i)$는 프레임 f의 i번째 bin의 색조 값이다. 3가지 속성값에 가중치를 적용한 비유사도 거리는 식(3)과 같다 [3,6].

$$d(f,g) = w_h d(H(f),H(g)) + w_s d(S(f),S(g)) + w_v d(V(f),V(g)) \quad (3)$$

여기서 w_h, w_s, w_v는 각 속성별 가중치로 명도가 낮은 영역은 유사도 측정 시 색조는 의미가 적고, 반대의 경우에는 색조의 의미가 커지므로 가중치를 조절하여 유사도를 계산한다[6]. 모든 속성 값들은 0과 1사이의 값으로 정규화 된다.

2. Shot Group Detection

샷이 구분되면 유사 샷을 묶어 샷 그룹을 형성해야 한다. 이때 샷의 대표 프레임인 키 프레임을 선정하여 이들 간의 유사도를 계산하여 샷 간 유사성을 검사한다[2,4,19]. 샷 사이의 유사도 계산은 샷 검출 시 사용하였던 식을 키 프레임 간의 유사도 계산에 적용한다. 샷 i와 j 사이의 유사도는 식 (4)를 이용하여 계산한다. 식(3)과 차이점은 비유사도를 유사도 함수로 변형시켰다는 점이다.

$$sim(S_i,S_j) = \sum_l (1-(w_h\sqrt{(S_{i,l}^H - S_{j,l}^H)^2} + w_s\sqrt{(S_{i,l}^S - S_{j,l}^S)^2} + w_v\sqrt{(S_{i,l}^V - S_{j,l}^V)^2})) \quad (4)$$

w_h, w_s, w_v: 색조, 채도, 명도에 대한 가중치

$S_{i,l}^H, S_{i,l}^S, S_{i,l}^V$: 샷 S_i의 l번째 bin의 색조, 채도, 명도 값

유사도 함수에 의해 샷 클러스터링을 수행하는데 시간적 요소에 의해 연관되는 시간적(temporal) 클러스터링과 시간적 요소에 무관하게 유사한 샷을 묶는 공간적(spatial) 클러스터링 2가지로 구분하여 그룹을 결정한다. 샷의 특징 값만 이용하여 유사한 샷을 묶는 일반적인 클러스터링은 식(4)로 유사도를 계산하여 묶는 공간적 클러스터링이다. 시간적인 의미를 포함하는 시간적 클러스터링은 샷들의 시간적 인접성을 고려하여 그룹으로 묶는 방법이다. 샷이 인접한 경우 이들은 동일한 이벤트 상황에서 촬영한 화면일 개연성이 높기 때문이다. 비록 인접한 샷의 내용이 다를지라도 하나의 의미상 그룹으로 묶을 수 있다. 한 공간에서 여러 카메라가 서로 다른 각도에서 촬영이 일어날 수도 있고 대화의 상황에서 두 출연자가 번갈아 나타나는 경우를 예로 들 수 있다[5,6]. 따라서 인접한 샷을 서로 연관된 샷 그룹으로 묶는 클러스터링 방법을 고려할 필요가 있다. Fig. 2는 이런 상황을 나타내는 샷의 나열을 보여주고 있다.

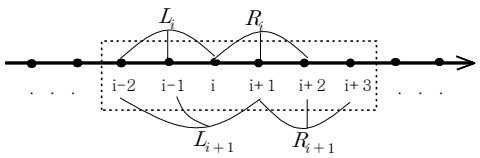

Fig. 2. Temporal Clustering

샷 S_i를 중심으로 좌우 5개 샷, 총 6개 샷으로 구성된 윈도우(점선 구간)을 지정하여 시간적 클러스터링을 수행한다. 시간적으로 인접한 다수 개의 샷을 중심으로 이들을 하나의 클러스터로 결합할 수 있는지 여부를 계량화하기 식(5)을 정의한다.

$$\begin{aligned} L_i &= Max\{sim(S_i, S_{i-1}), sim(S_i, S_{i-2})\} \\ R_i &= Max\{sim(S_i, S_{i+1}), sim(S_i, S_{i+2})\} \\ L_{i+1} &= Max\{sim(S_{i+1}, S_{i-1}), sim(S_{i+1}, S_{i-2})\} \\ R_{i+1} &= Max\{sim(S_{i+1}, S_{i+2}), sim(S_{i+1}, S_{i+3})\} \end{aligned} \quad (5)$$

샷 S_i가 왼쪽 샷 보다 오른쪽의 샷과 유사할 경우 새로운 그룹으로 정할 수 있다. L_i는 현재 샷 S_i기준으로 이전 2개의 샷과 비교해서 유사도가 높은지 여부를 가늠하는 값이다. R_i는 우측의 2개의 샷과 비교하여 유사도가 높은지 여부를 나타낸다. L_{i+1}과 R_{i+1}은 샷 S_{i+1}을 기준으로 좌우 인접한 2개의 샷과의 유사도 정도를 나타내는 값이다. 시간적 샷 클러스터링은 비디오 샷 S_i가 주어질 때 만약 그것이 새로운 그룹의 첫 번째 샷이라면 그것은 그것의 왼쪽에 있는 샷들보다 오른쪽에 있는 샷들과 더 큰 상관관계를 가질 것이다. 동일한 그룹에 있는 샷들은 다른 그룹에 속해 있는 샷들과의 관계보다 더 큰 상관관계를 가질 것이라고 가정하기 때문이다. 그룹의 잠재적인 경계를 결정하기 위해서 샷 S_i의 분리(Partition)수준을 나타내는 $P(i)$를 식(6)과 같이 정의한다.

$$P(S_i) = Min(R_i, R_{i+1}) / Min(L_i, L_{i+1}) \quad (6)$$

$P(S_i)$는 샷이 하나 걸러 유사 샷이 검출될 경우에 이를 그룹으로 묶는 효과도 발휘한다. 샷 그룹을 결정하기 위해서 각 샷의 우측에 인접한 샷과의 비교를 통해 유사도가 큰 경우 좌우측 인접한 샷을 비교하는 $P(S_i)$값을 구한다. 이 값이 설정된 특정 임계값을 넘어서면 샷 그룹의 시작으로 인정한다. 다음은 샷 그룹의 시작을 찾는 알고리즘이다.

```
for S_i  i=[1..M]
    if R_i ≥ T_1 {
        if P(S_i) ≥ T_2
            S_i: first shot of the new group
        else
            continue
    }
    else if (R_i < T_1 && L_i < T_1) {
        S_i: first shot of the new group
    }
```

새로운 그룹의 첫 번째 샷인 경우에 샷 S_i의 R_i와 $P(S_i)$는 미리 정의되어진 임계값보다 커야 한다. 그룹 검출 첫 번째 절차는 바로 이러한 상황을 처리하기 위해 제안되어진다. 두 번째 그룹 시작 검출의 경우는 비슷하지 않은 샷이 그룹의 양측에 있는 경우, 즉 자기 자신이 그룹을 분리하는 역할을 수행하는 경우를 위해 제안된 것이다.

여기서 사용된 T_1, T_2는 사전에 계산된 임계값으로 샷을 이루는 키 프레임의 엔트로피를 통해 계산되는 방법을 적용하였다[3,5,6,8]. 샷 그룹은 물리적으로 유사한 샷들과 의미상 장면을 구성하는 두 가지 종류의 그룹을 내포하고 있다. 즉, 시간적으로 관련이 있는 샷들과 공간상으로 관련이 있는 샷들로 나누어 생각할 수 있다. 시간상으로 관련이 있는 샷들은 인접한 샷들이 시각적으로는 비교적 낮은 유사도를 가지지만, 유사 샷이 엇갈려 나타나는 경우 의미 있는 하나의 장면을 구성할 것으로 볼 수 있어 그룹으로 묶는 것이 유효하다. 시간적 클러스터링을 통한 그룹은 장면으로 규정할 수 있는 좋은 방법을 제공한다. 인접한 시간에 비록 샷의 비유사도가 커 별도의 그룹으로 생성될 가능성이 크나 시간적 요소가 더 크게 작용할 수 있어 의미 있는 장면으로 묶을 수 있다. Fig. 3은 그룹으로 묶여진 비디오 그룹의 예를 보여주는 그림이다.

Fig. 3. Result for Shot Group Detection

3. Shot Group Clustering and Representative Shot Detection

시간적 클러스터링은 장면을 정의할 수 있는 의미상의 샷 그룹을 결정할 수 있다. 여기에는 유사 샷들이 다수 개 포함될 수 있다. 샷 그룹을 대표할 수 있는 몇 개의 샷을 클러스터링을 통하여 선별하여 묶어 하위 샷 그룹으로 나눌 수 있다. 다음 알고리즘은 그룹 내의 샷을 분류하는 클러스터링 알고리즘이다.

```
k=1, C_k ← S_1   // first cluster(C_1)
for S_i ∈ G_m   i=[1..g]   {
   simVal_j = max_j(ClSim(S_i, C_j))
   if (simVal < Th) {
      k=k+1   // new cluster
      C_k ← S_i
   }
   else
      C_j ← S_i
}
```

그룹(G_m) 내의 샷의 개수가 2개 이상인 경우 위의 클러스터링 알고리즘이 시작된다. 첫 번째 샷은 새로운 클러스터 C_1으로 정의되고 두 번째 샷부터 기존의 클러스터와 비교하여 가장 가까운 클러스터를 찾는다. 샷(S_i)과 클러스터(C_j) 간 유사도 $CSim(S_i, C_j)$는 식 (7)로 계산된다.

$$CSim(S_i, C_j) = \max_l(sim(S_i, S_l \in C_j)) \quad (7)$$

최대로 유사한 클러스터를 찾기 위해 여러 클러스터와 유사도를 계산하여 가장 유사한 클러스터에 샷을 배정한다. 단, 최대 유사도 값이 임의로 설정된 적정 수준(Th)보다 작으면 새로운 클러스터를 생성하는 것이 좀 더 나은 클러스터링 결과가 될 것이므로 이 샷으로 새로운 클러스터를 생성한다. 그룹 내 클러스터링이 끝나면 각 클러스터의 대표 샷을 선정하여 그룹을 간략하게 재구성할 수 있다. 다음 알고리즘은 대표 샷을 구하는 알고리즘이다.

```
for C_j, j=1..k {
  if (|C_j| > 1) {
    S_avg = avgShot(H, S, V) for C_j
    S_rep = max_l(sim(S_l, S_avg)), S_l ∈ C_j
  }
}
```

클러스터(C_j)를 구성하는 샷이 2개 이상인 경우 각 클러스터의 샷 평균 S_{avg}를 구한다. S_{avg}는 클러스터 C_j내의 각 샷에 대하여 색조, 채도, 명도 각 성분을 bin 별로 합산하여 샷의 개수로 나눠 각 성분의 평균값을 구한 값이 된다. 이 평균 샷은 클러스터의 중앙 지점을 정의하게 되고 이에 가장 유사한 샷S_l을 찾아 이를 대표 샷 S_{rep}로 정한다.

4. Shot Group Detection and Experiment Result

실험용 비디오 클립을 중심으로 샷 그룹 검출 실험을 시행하였다. 비디오 클립은 주제별로 스포츠, 광고, 뮤직 비디오, 드라마, 뉴스 등 다양한 유형 별로 수집하여 이들을 대상으로 올바른 그룹으로 나눠지는지 여부를 실험하였다. Table 1은 대상 비디오 클립에 대한 정보를 나타낸 것이다.

Table 1. Video clips for Shot Group Detection

주제	파일	시퀀스(초)	프레임 수
뉴스	2	277	5862
광고	1	19	575
스포츠	2	77	2510
뮤직비디오	1	26	628
다큐멘터리	1	167	4033

그룹 검출 단계에서 검출된 결과의 질을 판단하기 위해 정확도를 식 (8)과 같이 정의한다.

$$P = \frac{CG}{TG} \quad (8)$$

CG(Correct Group)는 올바로 검출한 그룹 수를 가리키고 TG(Total Group)는 비디오 클립 내에서 검출된 그룹의 총수를 가리킨다. 여기서 올바른 그룹은 관측자에 의해 그룹 내의 모든 샷이 동일한 이벤트를 내포하는 경우에 한해 그룹이 올바르게 검출되었다고 판단한다.

장면으로 요약 압출될 수 있는 정도를 표시하는 수치로 압축률(CR: Compression Rate)을 식(9)와 같이 정의한다.

$$CR = \frac{(TS - TG)}{TS} \quad (9)$$

여기서 TS(Total Shot)는 샷의 총 수를 나타낸다. 이 식은 개별 샷들이 모여 그룹을 구성한 비율을 측정하는 것으로 이 값이 클수록 압축률이 높다고 할 수 있다. 압축률이 높다는 것은 소수의 샷으로 전체 비디오를 표현할 수 있음을 나타내는 것이다. 실험을 통해 측정된 정확도와 압축률이 Table 2에 표시되었다.

Table 2. Experiment Result for Group Detection

주제	TS	P			CR
		TG	CG	P	
뉴스	109	76	74	0.97	.30
광고	15	8	7	0.88	.47
스포츠	29	16	13	0.81	.45
뮤직비디오	6	4	4	1.0	.33
다큐멘터리	11	7	5	0.71	.36

Table 2에 표기된 대로 그룹 검출은 대체로 70% 이상의 정확도를 나타내고 있음을 보여준다. 이 표는 그룹 검출 정확도가 높을수록 압축률은 반비례하여 낮음을 보여주고 있다. 예로 뉴스의 경우 그룹 검출 정확도는 97%에 이르나 샷 그룹의 수가 많아 압축률은 낮게 나타났다. 전체의 내용을 표현하기 위해서는 다수의 대표 샷이 필요하다는 점으로 해석된다.

SDCEO 시스템[16]은 샷 그룹에 해당하는 장면을 검출하는 방법을 제시하였고 실험결과를 발표하였다. 3개의 실험 비디오를 통해 얻은 장면 검출 정확도는 86.5%, 80.1%, 93.9%이었다. 본 연구의 정확도와 큰 차이는 없었으나 실험을 위한 샷을 2명의 실험자가 수동으로 샷과 장면으로 나누어 정의하였고 오류를 최소화하기 위해 다른 1명이 그 결과를 결정하게 하였다. 실험용 데이터 수가 제한적이고 샷 검출 시 인접한 프레임들을 중심으로 인위적으로 샷을 구분하였기 때문에 본 연구와 직접적인 비교는 어렵다고 판단된다. 또한 상향식 클러스터링을 사용하여 계층적 클러스터링을 수행하였으나 인접한 샷을 통한 그룹형성에는 제약이 있는 것으로 판단된다. 장면 검출 정확도를 개선하기 위한 방법으로 컬러 히스토그램 외에 코너 에지와 객체 컬러 히스토그램을 추가적으로 사용하여 일정 수준 정확도를 높이는 결과를 산출하였으나 인접한 샷을 그룹으로 묶지 못하는 결과를 초래할 수 있다는 점이 있다.

Yeung의 연구[17]에서는 검출된 샷들을 클러스터링을 통해 묶되 시간 제약을 통해 인접성에 의한 그룹을 강제하였다. 클러스터들 간의 시간적 관점에서 전후관계를 연결하는 장면 전환 그라프(Scene Transition Graph, STG)를 작성한 후 그라프에서 cut edge를 찾아 이들로 장면의 경계를 결정하였다. 수많은 노드와 edge로 구성된 STG는 쉽게 소단위 그룹으로 분리되기는 어려워 생성되는 클러스터의 비디오 시간 분량이 수초에서 수분을 넘나드는 그룹을 형성한다. 이 장면 구성은 본 연구에서는 시간적 클러스터링을 통한 그룹에 해당된다고 할 수 있으나 Yeung의 연구에 있어서는 의미 있는 장면을 찾는 것이 주목적이기 때문에 본연구와 차별된다. 본 연구의 공간적 클러스터링을 통한 샷 그룹은 Yeung과 같은 연속적인 의미 있는 장면을 정의하기는 어려울 수 있으나 비디오 전체를 대표할 수 있는 샷으로 사용될 수 있을 것이다.

IV. Conclusions

본 논문은 비디오 클립의 내용을 클러스터링을 통하여 구분하는 방법에 관한 연구이다. 프레임의 저수준 특징들을 이용하여 샷을 구분하였고 현재의 샷과 이웃하고 있는 샷을 대상으로 샷 윈도우를 이동하면서 이들 간의 연관성을 찾아 시간적 클러스터링을 수행하였다. 시간적 클러스터링의 결과는 의미상으로 서로 연결된 장면을 구성할 수 있는 샷 그룹을 형성한다. 이 그룹에는 시각적 속성이 다른 샷들이 두 개 이상 섞여 있어 이들을 다시 클러스터링하여 하위 샷 그룹을 형성한다. 그룹에 속해 있는 두 개 이상의 샷에 대하여 이들의 중심에 해당되는 대표 샷을 구해 이들로 그룹을 대표하도록 선별하였다. 비디오 클립을 이런 그룹의 대표 샷으로 표현할 수 있다면 소수의 샷 프레임으로 전체 내용을 압축하여 표현할 수 있을 뿐만 아니라 비디오 검색도 이들을 통해 빠른 시간에 검색할 수 있는 방안을 제공할 수 있다. 다수 개의 비디오 클립을 대상으로 샷을 생성하고 이들을 앞에서 소개한 알고리즘을 적용해 샷 그룹으로 분류하였다. 생성된 샷과 생성된 그룹 사이의 비율로 압축률을 정하였는데 압축률과 그룹 검출 정확도 사이에는 서로 반비례하는 특징이 있다는 것을 실험을 통해 확인하였다.

본 연구는 시각적 특성인 색상의 배합과 비율에만 의존함으로 인해 올바른 샷 검출이 되지 못하는 경우도 발생한다. 같은 상황의 화면이더라도 조명효과에 변화가 발생했을 때 결과가 달라질 수도 있는 문제를 안고 있다. 향후 주도적 색상, 평균 명도, 모멘트, 윤곽선, 모션과 같은 다른 시각적 특징을 고려한 클러스터링을 연구할 필요가 있다고 본다. 의미 있는 샷 그룹과 장면을 대표하는 샷 프레임을 선별하여 이를 비디오 인덱싱, 내용기반 비디오 검색 응용에 활용할 예정이다.

REFERENCES

[1] S. Liu, M. Zhu, Q. Zheng, “Mining similarities for clustering web video clips,” International Conference on Computer Science and Software Engineering, pp. 759-762, 2008.

[2] S. Cheung, A. Zakhor, “Fast similarity search and clustering of video sequences on World Wide Web,” IEEE Trans on Multimedia, Vol. 7, No. 3, pp.524-537, 2004.

[3] Asghar, M.N., Hussain, F., Manton, R., “Video indexing: a survey,” International Journal of Computer and Information Tech. Vol. 3, No. 1, pp. 148-169, 2014.

[4] S.T. Dhagdi, P.R. Deshmukh, “Keyframe based video summarization using automatic threshold & edge matching rate,” International Journal of Scientific and Research Publications, Vol. 2, Issue 7, pp.1-12 2012.

[5] J. Baber, N. Afzulpurkar, S. Satoh, “A framework for video segmentation using global and local features,” International Journal of Pattern Recognition and AI, Vol. 27, No. 5, Article ID 1355007, pp.1-29, 2013.

[6] X. Zhu, J. Fan, A.K. Elmagarmid, X. Wu "Hierarchical video summarization and content description joint semantic and visual similarity," ACM/Springer Multimedia Systems Journal, Vol. 9, No.1, pp.31-53, 2003.

[7] A. Vailaya, A.K. Jain, H.J. Zhang, “Video Clustering,” Technical report MSUCPS:TR96-64, Michigan State University, 1996.

[8] J. Foo, J. Zobel, R. Sinha, “Clustering Near-Duplicate Images in Large Collections,” Proceedings of the International Workshop on Multimedia Information Retrieval, pp.21-30, 2007.

[9] M. Flickner, H. Sawhney, et. al., “Query by image and video content: the QBIC system,” Computer Vol. 28, Issue 9, pp.23-32, 1995.

[10] S.V. Porter, “Video segmentation and indexing using motion estimation,” Ph.D. dissertation, U. of Bristol, 2004.

[11] Z. Cernekova, I. Pitas, “Information theory based shot cut/fade detection and video summarization,” IEEE Proceedings in circuit and systems for video technology, Vol. 16, No. 1, pp.82-91, 2006

[12] J. Calic, E. Izuierdo, "Efficient key-frame extraction and video analysis," Proceedings of International Conference on IEEE, pp.28-33, 2002

[13] Y. Zhuang, Y. Rui, T.S. Huan, S. Mehrotra, "Adaptive key frame extracting using unsupervised clustering," Proceedings of International Conference on Image Processing, Vol. 1, pp.582-585, 2002

[14] A. Hanjalic, R.L. Lagendijk, J. Biemond, "Automated high level movie segmentation for advance video retrieval systems," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 9, No. 4, pp. 580-588, 1999

[15] L.H. Chen, Y.C. Lai, H.Y. Liao, "Movie scene segmentation using background information," Pattern Recognition, Vol. 41, No. 3, pp.1056-1065, 2008.

[16] D-W Chin, T-W Kim, J-M Choi, "Video scene detection using shot clustering based on visual features," Journal of Intelligent Information Systems, Vol. 18, No. 2, pp.47-60, 2012

[17] M. Yeung, B-L Yeo, "Segmentation of video by clustering and graph analysis," Computer Vision and Image Understanding, Vol. 71, No. 1, pp.94-109, 1998

[18] S-H Kim, D-S Hwang, "A shot change detection algorithm based on frame segmentation and object movement," Journal of the Korea society of computer and information, Vol. 20, No. 5, pp.21-29, 2015

[19] S-Y Shin, S-B Pyo, "Video abstracting using scene change detection and shot clustering for construction of efficient video database," Journal of the Korea society of computer and information, Vol. 11, No. 2, pp.111-119, 206

Author

Gye-Sung Lee received the B.S. degree in Electronics Engineering from Sogang University in 1980, the M.S. degree in Computer Science from KAIST in 1982, and the Ph.D. degree in Computer Science from Vanderbilt University in 1994. Dr. Lee joined the faculty of the Computer Science Department at Dankook University, Chungnam Korea, in 1996. He is currently a Professor in the Department of Computer Science, Dankook University. He is interested in data mining, intelligent systems, and bio-informatics.