

# 프레임간 히스토그램 차이를 이용한 개선된 대표프레임 추출 알고리즘

정지현, 전승철, 박성한  
한양대학교 공학대학 컴퓨터공학과

## An Improved key Frame Selection Algorithm Based on Histogram Difference Between Frames

Ji-Hyun Chung , Seung-Chul Jun, Sung-Han Park  
Department of Computer Science & Engineering, HanYang University  
{jhchung, scjun, shpark}@cse.hanyang.ac.kr

### Abstract

In this paper, we propose as new algorithm for the selection of key frames in a given video. For the selected key frames to be well defined, the selected key frames need to spread out on the whole temporal domain of the given video and guaranteed not to be duplicate. For this purpose, we take the first frame of each shot of the video as the candidate key frame to represent the video. To reduce the overall processing time, we eliminate some candidate key frames which are visually indistinct in the histogram difference. The key frames are then selected using a clustering processing based on the singly linked hierarchical tree. To make the selected key frames be distributed evenly on the whole video, the deviation and time difference between the selected key frames are used. The simulation results demonstrate that our method provides the better performance compared with previous methods.

### I. 서론

디지털화 된 대용량의 비디오들 중에서 흥미 있는 비디오의 부분이나 대략적인 핵심 내용을 찾기는 힘들다. 이때 이러한 문제를 해결하기 위해서 전체 비디오

중에서 비디오를 대표할 수 있는 한 개의 프레임을 이용한다. 전체 비디오를 대표하는 프레임은 잘 선택되어지면 대용량의 비디오 서버로부터 원하는 비디오를 찾거나 전체 비디오의 내용을 요약하거나 그것에 접근할 때 많은 도움을 줄 수 있다. 하지만 전체 비디오의 내용을 함축적으로 나타내는 하나의 프레임을 찾는 것은 쉬운 일이 아니다. 따라서 비디오 전체를 나타낼 수 있는 한 개의 대표프레임을 구하는 대신에 비디오의 내용을 부분적으로 나타낼 수 있는 여러 개의 샷(shot)으로 분할하고 각 샷을 비디오의 내용을 함축하는 대표하는 대표 이미지를 대표프레임의 집합으로 나타내는 연구가 많이 발표 되었다[1]-[3]. 그러나 샷의 대표 이미지를 대표프레임집합으로 추출하는 경우 대표프레임의 집합이 너무 크면 원하는 정보를 찾는 데 어려움이 있다. 따라서 샷의 개수를 최적화 하여야 한다. 일반적으로 대표프레임의 개수를 줄이는 방법은 중복을 제거하는 방법이 많이 사용된다[4]-[6].

.Yeung[4]은 비디오 샷으로부터 하나의 대표프레임만을 선택하였다. 이러한 대표프레임은 시간적 거리와 시각적인 유사성에 기반 하여 cluster화한다. 이러한 작업의 목적은 비디오 구조로부터 샷의 집합을 결정하는 것이다. 시간적인 제약을 제거해 주기 위해서 그룹과 시간 간격이 너무 멀리 떨어져 있는 프레임을 제거시킨다. Girgensoh[5]는 샷의 대표프레임을 계층적 트리에 의해서 분리 한 후 시간적인 제약을 위해서 너무 가까운 키 프레임을 제거하였다. 하지만 이러한 방법들의 경우에는 대표프레임의 집합이 한쪽으로 치우칠

가능성이 있다. 그리고 프레임을 제거시켜주는 기준을 heuristic하게 정해주어야 한다는 단점이 있다. 또 Sun[6]의 경우에는 비디오를 일정한 길이로 나눈 후 나누어진 비디오 중에서 유사도가 가장 떨어지는 프레임들을 대표프레임의 집합으로 하였다. 이 경우에도 전체 프레임에 유사도가 끌고루 분산되어 있는 비디오에 경우에만 적합하다. 이에 본 논문은 일정한 시간 경계와 함께 분산을 함께 적용함으로써 대표프레임을 부채살처럼 넓고 고르게 분포시킬 수 있는 방법을 제안한다.

## II. 제안하는 알고리즘

본 논문에서 제안하는 알고리즘은 여러 개의 대표프레임 중에서 비디오 클립을 잘 묘사할 수 있는 대표프레임의 집합을 결정한다. 대표프레임을 결정하기 위해서 먼저 샷의 경계를 찾은 후 그 샷의 첫 번째 프레임을 추출하여 대표프레임의 후보 프레임으로 놓는다. 샷의 경계를 결정하는 방법은 기존의 알고리즘[7]을 이용한다. 클러스터링을 하기 전에 대표 후보 프레임을 추출하는 부분과 실행시간과 클러스터링이 결과를 좋게 해주기 위한 전처리 과정을 거친다. 추출되어질 대표프레임의 집합을 좀더 명확하게 나타내기 위하여 후보 프레임들을 Singly Linked Hierarchical Tree[8]를 이용하여 클러스터 화한다. 이러한 알고리즘을 이용하여 원하는 개수의 대표프레임을 추출할 경우 각각의 클러스터로부터 하나의 프레임을 선택한다. 하지만 이러한 방법은 대표프레임의 전체적인 시간 분배에는 적합하지 않기 때문에 대표프레임들이 시간의 한 축으로 몰릴 가능성이 있다. 이러한 문제를 해결하기 위해서 찾아진 각 클러스터로부터 계속적으로 프레임들을 교체해 가면서 구성된 대표프레임의 집합의 분산 값과 시간 간격 값 및 각 프레임간의 차이를 이용하여 최적의 시간 분배 대표프레임의 집합을 추출한다. 그림 1은 제안하는 알고리즘의 대표프레임의 추출과정을 그림으로 나타낸 것이다.

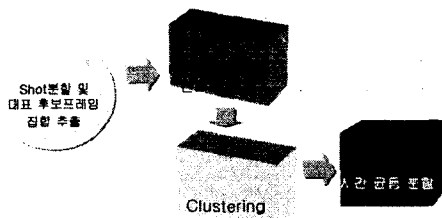


그림 1. 대표프레임 추출 처리과정

### 2.1 대표 후보 프레임집합 추출

후보 프레임의 집합을 추출하기 위해서 본 논문에서는 Jun 과 Park의 샷 분할 알고리즘[7]을 적용한다. 이 알고리즘은 샷 분할의 전 단계로 연속적인 장면의 변화 정도를 표시하는 히스토그램 차이와 픽셀 차이 값을 1차원 잡음 신호로 간주하여 이를 이용중간 값 필터를 적용하여 샷의 컷과 일반 프레임의 차이를 명확하게 한다. 이렇게 얻어진 데이터를 이용하여 인공 신경망을 사용하여 정확한 샷을 찾아낸다.

이렇게 구분되어진 샷의 프레임들 중에서 일단 비슷한 영상으로 클러스터화 되어진 샷 안의 프레임들은 영상의 차이가 모두 비슷하기 때문에 첫 번째 프레임을 후보 대표프레임 집합의 원소로 놓는다. 그림 2는 샷의 분할 과정을 그림으로 나타낸 것이다.

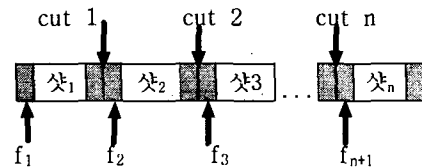


그림 2. 추출된 후보 대표프레임들의 집합

### 2.2 전처리 과정

비디오 데이터의 길이가 길거나 샷이 많이 발생하는 비디오 데이터의 경우 대표 후보 프레임 집합의 개수가 많아지기 때문에 클러스터링이나 시간 분할 처리시 많은 실행시간을 가지게 된다. 이에 프레임 집합의 개수를 최적화 시키므로 해서 실행시간을 줄이면서 보다 시각적으로 효율적인 대표프레임 집합을 생성하기 위하여 전처리 과정을 거친다. 이를 위하여 먼저 구해진 대표 후보프레임 집합 내의 각각의 프레임들간에 유사도를 비교한다. 유사도를 비교하는 비교단위로 클러스터링의 비교 방법으로 본 논문에서는 YCbCr 컬러모델의 Y값을 이용한다. 이때 Y값에 대한 유사도의 측정 도구로 히스토그램 차이를 이용한다. 식(2.1)은 두 프레임 간에 히스토그램 차이 값을 구하는 식이다.

$$HDM(f_i, f_{i+1}) = \frac{1}{M \times N} \sum_{j=1}^m \sum_{l=1}^n |H_i[j][l] - H_{i+1}[j][l]| \quad (2.1)$$

이렇게 해서 구해진 각각의 프레임 간의 히스토그램 차이 값을 가지고 하나의 대표 후보프레임에 대한 다른 모든 대표프레임과의 히스토그램 차이 값의 합 (SHD)을 구한다. 식 (2.2)는 그 과정을 식으로 나타낸

것이다.

$$SHD(f_i) = \frac{1}{n} \sum_{i=1}^n |H_i(i)| \quad (2.2)$$

$$AHD = \frac{1}{n} \sum_{i=1}^n |f_i| \quad (2.3)$$

위의 식을 사용하여 대표 후보프레임의 모든 원소들에 대해서 SHD를 구한 후 SHD의 평균값(AHD) 하나의 대표프레임에 대한 다른 모든 프레임간의 히스토그램 차이 값에 대한 평균값과 비교를 한다. 비교해서 AHD 보다 값이 작은 프레임들은 일차적으로 시각적인 변별력이 작은 프레임이라고 가정하고 이 프레임이 나오는 히스토그램 쌍은 제거시킨다. 그렇게 함으로써 다음에 있을 클러스터링 시에 좀더 명확한 시각적인 차이가 있는 대표프레임 집합을 생성할 수 있고, 전체적인 실행시간도 줄이는 효과를 가져온다.

### 2.3 시각적인 변별력을 위한 Clustering

비디오의 전체 내용들을 보다 적은 대표프레임으로 표현해 주기 위하여 비디오의 대표프레임 집합내의 프레임들 사이에 중복을 피해야 한다. 한 대표프레임 집합 내에 시각적으로 비슷한 프레임들이 존재하는 경우 비디오 데이터를 효율적으로 나타내는데 문제가 있다. 이를 위해서 비슷한 프레임끼리 묶는 클러스터링 방법을 이용한다. 본 논문에서는 이러한 클러스터링 방법으로 Singly Linked Hierarchical Clustering 방법 [8]을 사용한다. 이 방법을 사용하는 이유는 클러스터의 결과가 계층적으로 나타내어지기 때문에 시각적인 변별력이 큰 집합을 순서대로 추출할 수 있고 다른 계층적인 클러스터링 방법에 비해서 상대적으로 빠른 실행시간을 가지고 있기 때문이다.

### 2.3 대표프레임에 대한 시간 분할

일반적으로 대표프레임의 집합은 전체 비디오의 크기에 비례해서 시간적으로 넓게 분포되어야 한다. 넓은 시간에 걸쳐 분포된 대표프레임의 집합은 전체적인 시간흐름 따른 비디오 분석에 용이하다. 그 자세한 과정은 다음과 같다.

1) 추출되어진 대표프레임의 집합은 시간적인 순서가 아니므로 일단 시간순서에 맞게 정렬을 한다.

$$Dt = |frame\#of f_i - frame\#of f_j| \quad (2.4)$$

$$Average = \frac{1}{n-1} \sum_{i=1}^{n-1} Dt \quad (2.5)$$

ii) 시간순서에 따라서 정렬을 하고 난 후에 추출되어진 대표프레임 간의 분산을 구한다.

$$Var = \frac{1}{n-1} \sum_{i=1}^{n-1} |Average - Dt_i| \quad (2.6)$$

iii) 대표프레임 집합의 분산과 평균거리를 저장 한다.

vi) 프레임 간의 거리가 가장 먼 프레임과 가장 가까운 프레임을 제거한다.

v) 제거된 프레임과 같은 클러스터에 있는 클러스터에 있는 child frame으로 대체 시킨다.

vi) 임의의 클러스터의 프레임이 바닥이 될 때까지 이러한 과정을 반복한다.

vii) 모든 분산과 평균의 집합 중에서 가장 작은 분산과 비디오의 시간간격과 평균의 차가 가장 작은 분산과 평균이 들어가 있는 대표프레임 집합을 대표프레임으로 결정한다.

## III. 실험결과

실험에 사용한 데이터는 드라마 3편(27분, 52분, 55분)과 영화 2편(90분, 84분)이고 각각 대화가 많거나 액션이 많거나 화면의 전환이 빠르거나 등의 비디오의 전반적인 특징적인 데이터를 이용한다.

그림 3에서는 알고리즘의 전체적인 실행시간을 나타낸 것으로서 간편한 클러스터링 알고리즘과 전처리 과정을 거친 제안한 알고리즘의 결과가 좋게 나왔음을 보여주고 있다.

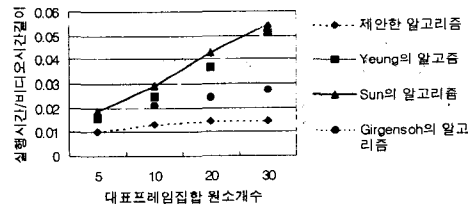


그림 3. 실행시간

그림 4는 대표프레임간의 평균 거리 차로서 그림5에 있는 분산그래프와 함께 대표프레임이 시간 영역에서 전체적으로 분포되어있는지를 알 수 있다.

평균 거리 차는 추출된 대표프레임 집합의 크기에 따라 집합의 대표프레임간의 거리 차이의 상대적인 값을 보여준다. 이 상대적인 거리 차(KD)는 집합의 크기를 k라 한 경우 다음 식(3.1) 및 식(3.2)에 의해서 구해진다.

$$SD = (\# \text{ of key frames in the set}) / (k-1) \quad (3.1)$$

$$KD = |SD - \text{Average}| / SD \quad (3.2)$$

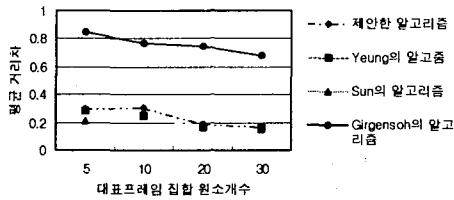


그림 4. 프레임간 평균 거리 차

그림에서 보는 바와 같이 평균 거리 차와 분산 값을 볼 때 다른 알고리즘과 비슷하거나 적은 값을 가짐으로써 좋은 결과를 보였다. 이는 제안한 알고리즘이 전체적으로 넓고 고르게 분포되어 있음을 나타낸다.

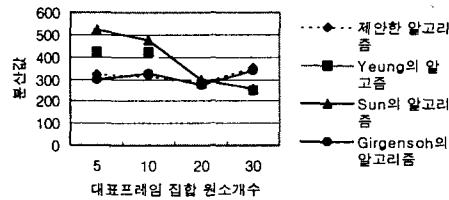


그림 5. 대표프레임 집합 크기별 분산

그림 6은 추출되어진 대표프레임 간의 히스토그램 차이 값을 계산한 것으로써 제안한 알고리즘은 기존의 알고리즘에 비해서 추출되는 원소가 증가함에도 고르면서도 높은 값을 가진다. 이는 추출된 대표프레임간의 시각적인 변별력이 높음을 알 수 있다.

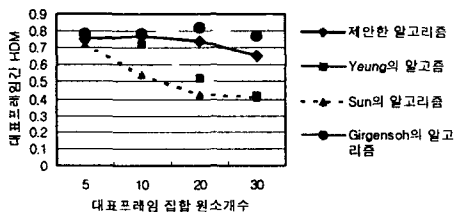


그림 6. 대표프레임간 히스토그램 차이값

## VI. 결론

본 논문에서는 시각적인 변별력을 위한 전처리 과정과 클러스터링, 프레임의 분산 값과 기준거리를 이용하여 비디오의 내용을 보다 정확하게 표현하는 방법을 제시한다. 개선된 전처리 과정을 거친 Singly-linked hierarchical clustering 알고리즘을 적용하기 때문에 시각적

으로 유사한 프레임이 대표프레임의 집합의 원소가 될 확률이 매우 적어지며 결과적으로 시간적 빠른 실행 시간을 가지고 시각적인 변별력이 큰 적절한 대표프레임의 집합을 추출한다. 실험의 결과에서 보는 것과 같이 제안한 알고리즘이 다른 알고리즘에 비해서 시각적으로 중복되지 않고 전체시간에 걸쳐서 고루 분포되는 대표프레임 집합을 생성하였음을 보여주고 있다.

향후 연구과제는 내용기반의 대표프레임의 추출을 위해서 Scene 기반의 대표프레임 추출 알고리즘에 대한 연구가 진행되어야 한다. 그렇게 함으로써 더욱 의미 있는 대표프레임의 추출이 가능해 지게 된다.

## 참고문헌

- [1] Y. Tonomura, A. Akutsu, K. Otsuji, T. Sadakata, "Video MAP and Video Space Icon: Tools for Anatomizing Video Content," Proc. ACM INTERCHI '93, pp.131-141, 1993.
- [2] H. Ueda, T. Miyatake, and S. Yoshizawa, "An Interactive Natural-Motion-Picture Dedicated Multi-media Authoring System," Proc. ACM SIGCHI 91, pp.343-350, New Orleans, 1991.
- [3] A.M.Ferman and A.M.Tekalp, "Multiscale Content Extraction and Representation for Video Indexing," Multimedia Storage and Archiving Systems II, SPIE, vol.3229, pp.22-31, Dallas, 1997.
- [4] M.M. Yeung, B.L. Yeo, W. Wolf and B. Liu, "Video Browsing using Clustering and Scene Transition on Compressed Sequences," SPIE Multimedia Computing and Networking, vol.2417, pp. 399-413, 1995.
- [5] A. Girgensohn and J. Boreczky "Time-Constrained Key frames Selection Technique," Proc. 1997 IEEE International Conference on Image Processing, vol. I, pp. 756-761, 1997.
- [6] X. Sun, M. S. Kankanhalli, Y. Zhu, and J. Wu, "Content Based Representative Frame Extraction for Digital Video," Proc. IEEE Conference of Multimedia Computing and Systems, Austin, 1998.
- [7] S.C. Jun and S.H. Park, "An Automatic Cut Detection Algorithm Using Median Filter and Neural Network," ITC-CSCC 2000, vol. II, pp. 1049-1052, 2000 July 11-13.
- [8] E. Rasmussen, "Information Retrieval: Data Structures and Algorithms," editors, W.B. Frakes and B. Baeza-Yates, pp. 419-442, Prentice Hall, 1992.