

MPEG 비디오 스트림에서의 대표 프레임 추출 및 멀티미디어 검색 기법

김영호^o, 강대성

동아대학교 전기전자컴퓨터공학부

Key Frame Detection and Multimedia Retrieval on MPEG Video

Young Ho Kim^o, Dae-Seong Kang

School of Electrical, Electronic and Computer Eng., Dong-A Univ.

E-mail: dskang@daunet.donga.ac.kr

요약

본 논문에서는 MPEG 비디오 스트림을 분석하여 DCT DC 계수를 추출하고 이들로 구성된 DC 이미지로부터 제안하는 robust feature를 이용하여 shot을 구하고 각 feature들의 통계적 특성을 이용하여 스트림의 특징에 따라 weight를 부가하여 구해진 characterizing value의 시간변화량을 구한다. 구해진 변화량의 local maxima와 local minima는 MPEG 비디오 스트림에서 각각 가장 특징적인 frame과 평균적인 frame을 나타낸다. 이 순간의 frame을 구함으로서 효과적이고 빠른 시간 내에 key frame을 추출한다. 추출되어진 key frame에 대하여 원영상을 복원한 후, 색인을 위하여 다수의 parameter를 구하고 사용자가 질의한 영상에 대해서 이들 파라미터를 구하여 key frame들과 가장 유사한 대표영상들을 검색한다.

I. 서론

현재 많은 정보들이 비디오 데이터로 전송 또는 저장되고 있으며 고성능 PC의 보급과 internet과 같은 통신망의 대중화로 비디오 데이터는 급속도로 증가하고 있다. 이에 많은 비디오 스트림들의 데이터베이스화를 위해 고용량의 비

본 연구는 한국과학재단 특정기초연구 (1999-2-3-2-001-2)지원으로 수행되었음.

디오 데이터를 효과적으로 색인하고 검색할 수 있는 기술이 연구되어 지고 있다. 검색은 구문기반검색과 내용기반검색으로 나눌 수 있으며[1,2] 모든 과정을 인간의 주관적인 관점이 배제되고 객관적으로 처리되는 자동화된 내용기반검색이 주로 연구되어 지고 있다. 본 논문에서는 이런 내용기반검색을 위하여 비디오 스트림을 분석하여 shot을 찾아내고 이들 중 key frame을 찾아 색인하여 사용자의 질의에 부합하는 비디오를 검색한다[3,4].

본 논문의 구성은 다음과 같다. I장 서론에 이어 II장은 shot 및 key frame 검출하고 III장에서는 key frame 색인 및 검색을 한다. IV장에서는 실험한 결과와 이에 대한 고찰을 하였고 V장에서 결론을 맺었다

II. shot 및 key frame 검출

1. Shot boundary frame 추출

본 연구에서는 shot을 검출하기 위하여 MPEG에서 I picture의 계수만으로 영상을 구성한 DC image를 사용함으로써 video 스트림의 decoding 과정을 없앴으로서 처리시간을 대폭 감소시켜 효율적인 shot boundary frame 추출이 가능하게 하였다. shot boundary frame를 추출하기 위한 feature로서 4개의 feature를 조합하여 사용한다. 식(1)은 이전 영상과의 pixel간의 차분값의 feature로서 서로 다른 영상의 유사도를 측정하

는데 가장 기본이 되며, 전체적인 휘도 변화를 나타낸다.

$$DiffImg_i = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |I_{i-1}(x, y) - I_i(x, y)|}{NM} \quad (1)$$

여기서 M은 DC image의 열의 개수이고, N은 행의 개수이다.

식(2)는 shot 검출을 위하여 일반적으로 사용되어 지고 있는 feature로서 DC image의 히스토그램에 대한 chi-square 값이다.

$$X_i^2 = \sum_{k=0}^{n-1} \frac{(H_{i-1}(k) - H_i(k))^2}{H_i(k)} \quad (2)$$

여기서, $H_i(k)$ 는 i 번째 DC image 히스토그램의 k 번째 bin의 값을 나타낸다. n은 8bit 영상을 사용할 경우 256이다.

식(3)은 이전 DC image와 현재 DC image와의 feature로서 히스토그램 분산의 차분값이다.

$$DiffD_i^2 = \frac{(D_{i-1} - D_i)^2}{D_i} \quad (3)$$

여기서, D_i 는 i 번째 DC image의 히스토그램 분산 값이다.

식(4)(5)(6)(7)은 본 논문에서 새로이 제안하는 feature로서 양자화한 영상의 히스토그램의 각 bin값들의 열과 행의 위치에 대한 분산의 chi-square 값이다. 이 feature는 히스토그램의 bin 값들을 이용함으로써 객체의 움직임에 강인하고, 각 bin 값들의 열과 행의 위치에 대한 분산의 chi-square 값을 구함으로써 칼라의 변화에 둔감하다.

$$PX_i^2 = \sum_{k=0}^{n-1} \frac{(\rho X_{i-1}(k) - \rho X_i(k))^2}{\rho X_i(k)} \quad (4)$$

$$PY_i^2 = \sum_{k=0}^{n-1} \frac{(\rho Y_{i-1}(k) - \rho Y_i(k))^2}{\rho Y_i(k)} \quad (5)$$

$$\rho X_i(n)^2 = \epsilon(|BX_i(n) - \overline{BX_i(n)}|^2) \quad (6)$$

$$\rho Y_i(n)^2 = \epsilon(|BY_i(n) - \overline{BY_i(n)}|^2) \quad (7)$$

식(4)는 열에 대한 분산의 chi-square 값이고, 식(5)는 행에 대한 분산의 chi-square 값이다. 식(6)은 i번째 DC image의 n 번째 bin의 열에 대한 분산이고, 식(7)은 i번째 영상의 n 번째 bin의 행에 대한 분산이다. 비슷한 pixel의 개수를 갖는 서로 다른 객체에 대해서 히스토그램의 chi-square 값은 서로 같은 객체로 인식하나, 새로이 제안한 feature의 경우는 이를 다른 객체로

분석해내는 것이 가능하다. 이상과 같이 구해진 DC image의 feature들로부터 아래와 같은 단계로 shot boundary frame을 검출한다.

Step 1. 각 파라미터들의 전체 프레임에 대한 평균을 구한다.

$$\overline{DiffImg}, \overline{X}, \overline{DiffD}, \overline{PX}, \overline{PY}$$

Step 2. DC image의 각 feature 값이 아래의 조건식에서 condition 1과 2를 동시에 만족하면서 condition 3 혹은 4가 만족할 경우 shot boundary frame으로 검출한다.

$$\text{condition 1. } DiffImg_i > \alpha \overline{DiffImg}$$

$$\text{condition 2. } X_i > \beta \overline{X}$$

$$\text{condition 3. } DiffD_i > \gamma \overline{DiffD}$$

$$\text{condition 4. } PX_i > \delta \overline{PX} \text{ and } PY_i > \delta \overline{PY}$$

여기서, 각 상수 $\alpha, \beta, \gamma, \delta$ 는 실험을 통해서 2.0, 2.0, 2.0, 2.0으로 설정되었다.

2. Key frame 추출 알고리즘

key frame 검출은 구해진 shot들 중 그 video 스트림을 가장 잘 표현할 수 있는 대표 frame을 찾는 과정이다.

다음은 key frame 추출을 위한 shot의 파라미터들이다.

① 휘도의 평균

$$f_1 = AveShot_i = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} Shot_i(x, y)}{MN}$$

② 히스토그램의 분산값

$$f_2 = DisShot_i^2 = \epsilon(|H_i(n) - \overline{H_i(n)}|^2)$$

③ 양자화한 shot 히스토그램의 각 bin값들의 열과 행의 위치에 대한 분산의 평균

$$\rho X_i(n)^2 = \epsilon(|BX_i(n) - \overline{BX_i(n)}|^2)$$

$$\rho Y_i(n)^2 = \epsilon(|BY_i(n) - \overline{BY_i(n)}|^2)$$

$$f_{3x} = AveDisX_i = \frac{\sum_{k=0}^{n_1} \rho X_i(k)}{n_1}$$

$$f_{3y} = AveDisY_i = \frac{\sum_{k=0}^{n_2} \rho Y_i(k)}{n_2}$$

④ 이전 shot과의 휘도 차

$$f_4 = DiffShot_i$$

$$= \frac{\sum_{x=0, y=0}^{M-1, N-1} |Shot_i(x, y) - Shot_{i-1}(x, y)|}{MN}$$

⑤ 휘도 차의 누적에 대한 f_4 의 비율 .

$$f_5 = AccDiffShot_i = \frac{\sum_{k=0}^i DiffShot_k}{DiffShot_i}$$

위의 파라미터들을 각 검출된 shot에 대해서 구하고 난 후 key frame을 검출하기 위하여 아래와 같은 단계를 거쳐 characterizing value를 구한다.

Step 1. shot으로 검출된 모든 프레임에 대한 각 feature들의 분산을 구한다.

$$\rho_{f_n}^2 = \epsilon(|F_n - \overline{F_n}|^2)$$

Step 2. 각 feature들의 전체 평균에 대한 차를 구한다.

$$f_n' = |f_n - \overline{f_n}|$$

Step 3. 각 feature들에 대해 분산에 대한 비율만큼 weight를 부가하여 각 프레임의 특징 값 (C_m)을 구한다.

$$C_m = \omega_1 f_1' + \omega_2 f_2' + \dots + \omega_n f_n'$$

여기서, m 은 shot으로 검출된 프레임의 개수, 가중치 ω_n 는 아래와 같다.

$$\omega_1 = \frac{\rho_1}{\sum_{i=1}^n \rho_i}, \omega_2 = \frac{\rho_2}{\sum_{i=1}^n \rho_i}, \dots, \omega_n = \frac{\rho_n}{\sum_{i=1}^n \rho_i}$$

위의 단계를 거쳐 계산되어진 characterizing value의 시간에 대한 변화량을 구하여 local maxima와 local minima를 구한다.

III. Key frame의 색인과 검색

색인을 위하여 key frame으로 추출된 DC image를 원 영상으로 복원한다. 그리고 key frame의 색인을 위하여 파라미터들을 구한다. 원 영상에 대한 파라미터를 구함으로 보다 정확한 색인이 가능하고, 검색 시 입력되는 질의 영상과 좀 더 객관적인 유사도 측정이 가능하다. 색인과 검색알고리즘의 단계는 아래와 같다.

Step 1. key frame으로 추출된 frame을 원 이미지로 복원한다.

Step 2. frame의 색인을 위한 파라미터를 구한다.

① 영상의 휘도 평균(p_1)

② 히스토그램의 분산(p_2)

Step 3. Step2에서의 색인 값과 질의 영상의 색인 파라미터 값의 차이가 적은 순으로 유사도 평가를 한다.

$$Sim_i = \frac{\sum_{x=0, y=0}^{M-1, N-1} |Img_{query}(x, y) - Img_i(x, y)|}{NM}$$

Step 4. Sim_i 의 값이 일정 임계치(S) 이하가 될 때까지 Step 3을 반복한다.

Step 5. Sim_i 값이 작은 순으로 10개의 응답영상을 출력한다.

IV. 실험결과 및 고찰

실험 결과들은 Visual C++ 6.0 에 의해 Window 98 상에서 구현하였다. 테스트 영상으로 384 X 288 크기를 갖는 1618프레임의 영화와 946프레임의 뮤직비디오를 사용하였다. 그림 1은 전체 알고리즘의 구성도이다.

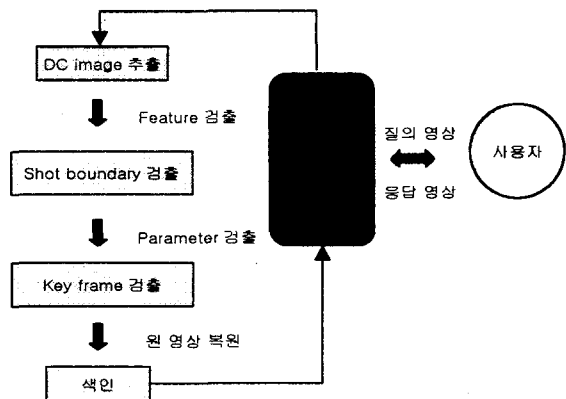


그림 1. 전체 알고리즘 구성도

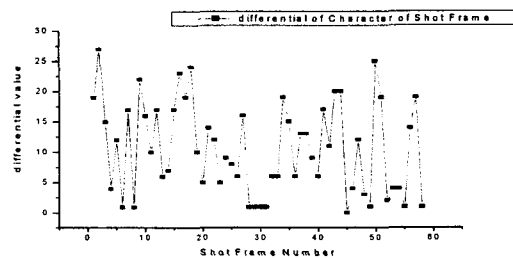


그림 2. characterizing value의 시간에 대한 미분

그림 2는 characterizing value의 시간에 대한 미분치이다. 각 local maxima와 local minima는 그 비디오 스트림의 가장 특징적인 frame과 평균적인 frame을 나타내고 있다. 이 순간의 frame을 key frame으로 추출하였다.

표 1은 shot 검출의 기준으로는 직접 사람의 시각으로 shot이 바뀌는 frame을 검출하여 비교 기준으로 사용하였다

표 1. shot 검출 결과

result feature	correct	miss	extra	total
f1, f2	45/56 (80%)	11	12	57
f1, f2, f3, f4	51/56 (91%)	5	9	60

표 1에서 correct는 사람의 시각으로 뽑은 shot과 정확히 일치하는 shot을 검출한 경우이고, miss 항목은 사람이 shot으로 지각하였으나 검출하지 못한 경우이고, extra 항목은 사람이 shot으로 지각하지 않았으나, shot으로 검출한 경우이다. 표 2는 제안한 방법으로 색인하여 150개의 영상중 1순위에 정확한 검색이 이루어진 경우는 98%로서 신뢰도 높은 검색이 가능하다.

표 2. 검색결과

	1st out	2nd out	3rd out	others
accuracy	147/150	1/150	1/150	1

V. 결론

본 연구에서는 객체의 움직임에 강인하면서 shot내에서의 칼라의 변화에 둔감한 제안한 robust feature를 이용하여 shot을 검출하였고, shot에서 구한 각 feature들의 통계적 특성을 이용하여 video 스트림의 특징에 따라 weight를 부가하여 구해진 제안한 characterizing value를 적용하여 key frame을 추출하였다. 실험 결과 일반적으로 사용되어지는 feature에 제안한 feature를 적용함으로써 보다 정확한 shot 추출이 가능하였다. 또한 제안된 characterizing value의 시간에 대한 미분치의 local maxima와 local minima에

서 video 스트림의 중요한 대표 frame을 key frame으로 추출하였고, 특히 가장 대표되는 스트림의 중요 부분에서 뚜렷한 분별력을 보였고, 이로서 주어진 video 스트림은 key frame으로서 충분히 대표될 수 있었다. 색인과 검색에 있어서는 1차적으로 key frame을 색인하고 질의 영상 입력 시 질의 영상의 색인 파라미터를 구해 파라미터 값에 따라 순차적으로 유사도를 비교함으로써 보다 빠르고 효율적인 검색이 가능하게 한다.

VI. 참고문헌

- [1] "Techniques and Standards for Image · Video and Audio Coding," K.R.Rao, J.J.Hwang. Prentice Hall.
- [2] "그림으로 보는 최신 MPEG," 후지와라 히로시, ASCII Corporation.
- [3] "A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases," Yunis S. Avrithis, Anastasios D. Doulamis, Nikolaos D. Doulamis and Stefanos D. Kollias, Computer Vision and Image Understanding Vol.75, Nos. 1/2, July/August, pp. 3~24, 1999.
- [4] "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification," Wei Xiong and Chung-Mong Lee, Computer Vision and Image Understanding Vol.71, Nos 2/2, August, pp. 166-181, 1998.
- [5] "Scene Change Detection in a MPEG Compressed Video Sequence," Jianhao Meng, Yujen Juan, Shin-Fu Chang, IS&T/SPIE Symposium Proceedings Vol. 2419, February 1995, San Jose, California.
- [6] "A feature-based algorithm for detecting and classifying production effects," Ramin Zabih, Justin Miller, Kevin Mai, Multimedia Systems 7, pp. 119-128, 1999.
- [7] "A hierarchical Multiresolution Video shot Transition Detection Scheme," Hong Heather Yu, Computer Vision and Image Understanding Vol.75, Nos 1/2, July/August, pp. 196-213, 1999