

프레임 블록화와 객체의 이동을 이용한 샷 전환 탐지 알고리즘

김 승 현*, 황 두 성**

A shot change detection algorithm based on frame segmentation and object movement

Seung-Hyun Kim *, Doosung Hwang**

요 약

본 논문에서는 프레임 블록화와 이동블록 간 객체 이동을 이용한 샷 전환 탐지 알고리즘을 제안한다. 객체의 급격한 이동을 탐지하기 위해 연속 프레임의 현재 블록의 대각선상에 위치한 이동 블록을 정의하고 블록 히스토그램을 비교한다. 제안 방법은 두 연속 프레임 내 블록 간 객체 이동 여부를 검사하며, 객체 이동 블록 정보를 가지고 프레임 간 샷 전환 탐지를 예측한다. 현재 프레임의 블록이 다음 프레임의 이동 블록과 비교 시, 블록 히스토그램이 사용되며 샷 전환 임계값은 Otsu 임계값 방법을 이용하여 자동으로 선정한다. 제안 방법은 영화, 드라마, 애니메이션, 국가기록원 소장 영상 등과 같이 다양한 흑백 또는 칼라 영상에 대해 테스트되었다. 실험결과 제안하는 방법은 기존의 알고리즘과 비교 시 탐지율을 높일 수 있었다.

▶ Keywords : 샷 전환 탐지, 블록화, 객체 이동, Otsu 임계 값

Abstract

This paper proposes a shot change detection algorithm by using frame segmentation and the object changes among moving blocks. In order to detect the rapid moving changes of objects between two consecutive frames, the moving blocks on the diagonal are defined, and their histograms are calculated. When a block of the current frame is compared to the moving blocks of the next frame, the block histograms are used and the threshold of a shot change detection is automatically adjusted by Otsu's threshold method. The proposed algorithm was tested for the various types of color or gray videos such as films, dramas, animations, and video tapes in National Archives of Korea. The experimental results showed that the proposed algorithm could enhance the detection rate when compared to the studied

•제1저자 : 김승현 •교신저자 : 황두성

•투고일 : 2015. 4. 4, 심사일 : 2015. 4. 13, 게재확정일 : 2015. 4. 28.

* 단국대학교 컴퓨터과학과(Dept. of Computer Science, Dankook University)

** 단국대학교 운동의과학과(Dept. of Kinesiology Medical Science & Computer Science, Dankook University)

methods that use brightness, histogram, or segmentation.

▶ Keywords : shot change detection, segmentation, object movement, Otsu's threshold

I. 서 론

현대 사회는 첨단 영상 장비 사용의 확대에 의해 사회 전반에 걸쳐 방대한 양의 비디오 데이터를 생산하고 있으며, 이러한 대용량 비디오 데이터를 효율적으로 관리하기 위해서는 컴퓨팅 자원의 활용이 필요하다. 비디오 데이터는 영상과 음향으로 구성된 일정한 형식의 구조화 된 멀티미디어 데이터이다. 비디오는 프레임(frame)들의 집합이고, 하나의 카메라에서 찍힌 연속적인 프레임의 모음이 샷이 되며, 다른 카메라의 샷과 연결되는 샷 사이의 전환되는 부분을 샷 경계(shot boundary) 또는 컷(cut)이라 한다. 이러한 샷 경계 또는 컷을 탐지하는 샷 전환 탐지(shot boundary detection)는 비디오 데이터의 분할(segmentation), 색인(index), 검색(search), 저장(storing)과 같은 효율적인 관리를 위한 기반 기술이다.

샷 전환 간 자동 탐지를 위해서는 영상을 구성하고 있는 각 프레임들의 특징을 추출하여 비교해야 한다. 비교에 사용되는 특징은 화소 값, 화소 값들의 분포를 나타내는 히스토그램 등이 사용되었다. 화소 값이나 히스토그램과 같은 특징을 가공하지 않고 비교하는 방법[1-3]은 샷 전환 탐지에 보편적으로 사용되는 방법이며, Otsu 임계 값 기반 히스토그램 분포를 이용한 방법[4], 로그정규분포를 이용한 방법[5], 히스토그램과 통계학적 접근을 합성하는 방법[6], 프레임의 수직과 수평의 화소 정보를 이용한 방법[7] 등이 연구되었다.

제안하는 샷 전환 탐지 알고리즘은 두 프레임의 블록 분할, 이동블록의 히스토그램 비교, Otsu 임계 값 설정 방법을 이용한다. 이동블록을 이용한 비교는 영상 내 객체의 급격한 움직임에 영향을 덜 받으며, 샷 전환 탐지 과정에서 발생하는 오 탐지를 줄일 수 있다. 임계 값 설정의 자동화는 영상마다 알맞은 임계 값을 이용한 비교가 가능하여 샷 전환 탐지의 정확도를 높일 수 있다는 장점이 있다.

본 논문의 2장에서는 샷 전환 탐지에 대한 관련 연구를 살펴본다. 3장에서는 제안하는 알고리즘을 설명

한다. 4장에서는 제안하는 알고리즘과 기존의 샷 전환 탐지 방법으로 실험한 결과를 보여준다. 마지막으로 5장에서는 제안하는 알고리즘의 장점 및 단점의 토의와 앞으로의 개선방안을 제시한다.

II. 관련 연구

샷 전환 탐지 알고리즘은 N 개의 프레임으로 구성된 비디오 $V = \{F_1, F_2, F_3, \dots, F_N\}$ 에 대해 샷 전환의 발생 여부를 나타내는 $D = \{d_1, d_2, d_3, \dots, d_{N-1}\}$ 를 출력한다. d_k 는 F_k 와 F_{k+1} 사이의 샷 전환 여부를 나타내며 탐지 알고리즘을 이용하여 결정된다.

화소 값 비교 방법은 기존 샷 전환 탐지 방법에서 널리 사용된 방법이다.

$$DP_k(x, y) = \begin{cases} 1, & \text{if } |F_k(x, y) - F_{k+1}(x, y)| > \epsilon_1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$d_k = \begin{cases} 1, & \text{if } \frac{1}{X_f Y_f} \sum_{x=0}^{X_f-1} \sum_{y=0}^{Y_f-1} DP_k(x, y) > \epsilon_2 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Zhang 등[1]과 Xiaoquan Yi 등[2]은 식 (1)과 (2)와 같이 연속된 프레임 간의 대응되는 좌표의 화소 값 차이를 이용해 샷 전환을 탐지하였다. 화소 값 차이 DP (different pixel)는 식 (1)을 통해 계산한다. 식 (1)의 $F_k(x, y)$ 는 k 번째 프레임의 (x, y) 에 위치한 화소의 값이며, 식 (2)의 X_f, Y_f 는 프레임의 가로와 세로 크기를 의미한다. $DP_k(x, y)$ 는 k 번째 프레임과 $k+1$ 번째 프레임 간의 화소 변화 값에 의해 결정된다. 화소 값의 차이가 임계 값 ϵ_1 보다 크다면 $DP_k(x, y)$ 는 1이 되고, 그렇지 않으면 0이다. k 번째 프레임 내 모든 화소에 대하여 $DP_k(x, y)$ 를 구한 후, 프레임 간 평균 화소 변화율을 계산한다. 평균 화소 변화율이 임계 값 ϵ_2 보다 크다면, 기존 프레임 F_k 와 참조 프레임

F_{k+1} 사이에 샷 전환이 발생했다고 판단한다. ϵ_1 과 ϵ_2 는 사전 실험에서 선정한다.

화소 값 비교와 더불어 샷 전환 탐지에 많이 사용되는 방법에는 히스토그램의 차이를 이용한 방법이 있다. 히스토그램 차이를 이용한 방법에는 GRAY 히스토그램을 이용한 방법 [1]과 RGB, HSV, YCrCb 등 다양한 색 공간의 히스토그램 정보를 이용한 방법[3]이 있다. Zhang 등[1]이 제안한 히스토그램 비교방법은 식 (3), (4)와 같다.

$$DH_k(j) = \begin{cases} 1, & \text{if } |H_k(j) - H_{k+1}(j)| > \epsilon_1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$d_k = \begin{cases} 1, & \text{if } \frac{1}{J} \sum_{j=0}^{J-1} DH_k(j) > \epsilon_2 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

식 (3)에서 $H_k(j)$ 은 k 번째 프레임의 j 번째 히스토그램 빈을 의미한다. 식 (4)의 J 는 히스토그램 빈의 크기를 의미한다. 식 (3)에서 $|H_k(j) - H_{k+1}(j)|$ 가 허용 임계 값 ϵ_1 보다 크다면 $DH_k(j)$ 는 1이 된다. 샷 전환 여부를 나타내는 d_k 는 히스토그램 빈의 차이를 구한 뒤, 평균을 구하여, 변화율이 임계 값 ϵ_2 보다 크다면, 기준 프레임 F_k 와 참조 프레임 F_{k+1} 사이에 샷 전환이 발생했다고 판단하여 $d_k=1$ 이 된다. ϵ_1 과 ϵ_2 는 사전실험에서 선택해야하는 어려움이 있다.

언급한 관련 연구는 모두 프레임의 전체 정보를 이용하여 비교하는 방법이다. 프레임의 전체 정보를 이용한 비교방법은 밝기의 급격한 변화나 객체의 움직임 등에 민감하게 반응 하였으며, 샷 전환이 발생한 프레임을 탐지하지 못하거나 샷 전환이 아님에도 샷 전환이 발생했다고 오 탐지하는 등 정확도가 낮았다. 이에 샷 전환 탐지의 정확도를 높이기 위해 프레임 임을 분할하여 분할된 영역끼리 비교를 하는 방법들이 연구되었다[4,8,9].

Lee 등[8]과 Go 등[9]은 프레임을 분할하여 블록 간의 비교를 이용하여 샷 전환을 탐지하였다. Lee 등[8]은 두 프레임의 블록을 비교할 때, 현재 프레임의 블록과 다음 프레임의 비교대상 블록의 확장된 영역을 비교를 하여 블록 내에서 발생하는 객체의 이동에 의한 오 탐지율을 줄이는 방법을 제안하였고, 블록 간의 화소 값 차이를 이용하여 샷 전환을 탐지한다.

$$DB_k = \frac{1}{B} \sum_{i=0}^{B-1} \left[\frac{1}{X_b Y_b} \right. \quad (5)$$

$$\left. \sum_{x=0}^{X_b-1} \sum_{y=0}^{Y_b-1} 1(|b_{k,i}(x,y) - b_{k+1,i}(x,y)| > \epsilon_1) \right]$$

$$d_k = \begin{cases} 1, & \text{if } DB_k > \epsilon_2 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

식 (5)의 B 는 블록의 수를, X_b 와 Y_b 는 블록의 화소 크기를 의미하며, $1(x)$ 는 x 가 참이면 1, 거짓이면 0이다. 블록의 차이가 임계 값 ϵ_1 보다 크다면, 해당 블록에서 샷 전환이 발생했다고 판단한다. 샷 전환 여부를 나타내는 식 (6)의 d_k 는 DB_k 가 임계 값 ϵ_2 보다 크면, 샷 전환이 발생했다고 판단하며, ϵ_1 과 ϵ_2 의 선택에 따라 탐지율이 의존된다.

Kang 등[4]은 프레임을 일정 크기의 블록으로 분할 한 뒤, Otsu 임계 값 설정 방법을 이용하여 블록 내 히스토그램의 Otsu 임계 값을 계산 후, 두 블록 간 Otsu 임계 값의 차이를 구하였다. 여기서 Otsu 임계 값 설정 방법은 자동 임계 값 설정 방법으로 히스토그램의 분포를 이용하여 히스토그램을 두 그룹으로 나누는 임계 값을 구하는 방법이다[10,11]. 식 (7)에서 두 블록 간 Otsu 임계 값 차이가 ϵ_1 이상이라면 $B_{i,j}$ 는 1이 되며, 샷 전환 여부를 나타내는 d_k 는 식 (7)과 같이 모든 블록의 $B_{i,j}$ 값에 의해 결정된다.

$$d_k = \begin{cases} 1, & \text{if } \frac{1}{B_r B_c} \sum_{i=1}^{B_r} \sum_{j=1}^{B_c} B_{i,j} > \epsilon_2 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

식 (7)에서 B_r 과 B_c 는 각각 분할 된 프레임 내 블록의 행과 열의 수를 의미한다. 프레임 내 모든 블록에 대해 $B_{i,j}$ 값을 구한 뒤, 샷 전환이 발생되었다고 판단 된 블록이 임계 값 ϵ_2 보다 크다면, 식 (7)에 의해 두 프레임 사이에서 샷 전환이 발생한 것으로 판단한다[4].

화소와 히스토그램만을 이용한 방법[1-3] 외에도 로그 정규분포(log-normal distribution)를 이용하여 임계 값을 영상마다 자동으로 설정하여 정확도를 높인 방법[5]도 있으며, 히스토그램과 통계학적 접근을 합성하는 방법[6]도 있다. 이 방법[6]은 급진적인 샷 전환뿐만 아니라 점진적 샷 전환도 검출이 가능하다고 보고되었다. 프레임의 일부 정보만을 이용하여 샷 전환 검출을 시도한 연구[7]에서는 프레임의 수직, 수평의 중심 부분을 한 줄씩 계속적으로 수집한 뒤, 시간을 기

준으로 수집 한 프레임의 중심 부분을 이어 붙여서 한 장의 이미지로 재구성한다. 그리고 재구성 한 이미지를 수평방향 또는 수직방향으로 이동하며, 한 줄씩 화소 값 차이를 측정하여 샷 전환을 탐지한다.

III. 샷 전환 탐지 알고리즘

본 논문에서는 기준 프레임의 블록과 참조 프레임의 이동 블록을 이용한 히스토그램 비교방법을 제안하여, 샷 전환 탐지 과정에서 발생하는 객체의 급격한 이동에 의한 오 탐지율을 줄이는데 목적이 있다.

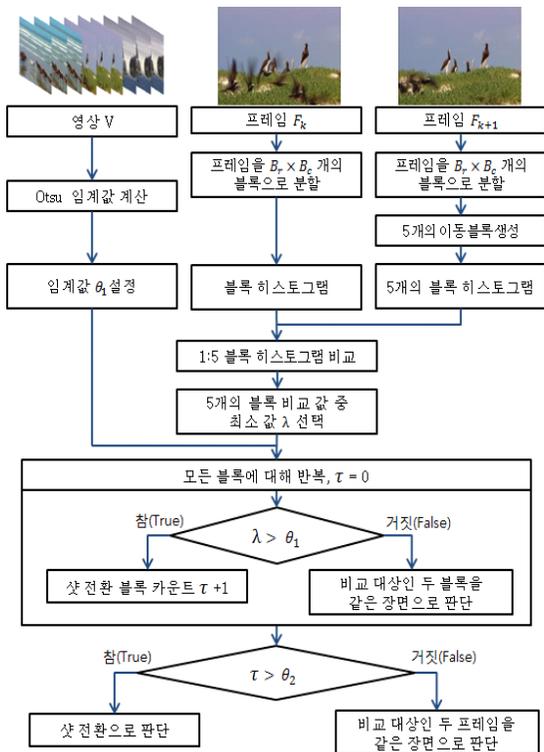


그림 1. 제안하는 샷 전환 탐지 알고리즘
Fig. 1. The proposed shot change detection algorithm

그림 1은 알고리즘의 전체 처리 과정을 나타낸다. 영상 V 마다 다르게 설정되는 임계 값 θ_1 은 모든 프레임에 대한 Otsu 임계 값을 계산한 후, Otsu 임계 값의 차이를 이용하여 샷 전환 후보군을 설정, 후보군의 Otsu 임계 값 평균으로 설정된다. θ_1 은 F_k 의 블록과 F_{k+1} 의 이동블록 간 샷 전환

탐지에 사용된다. θ_1 보다 블록 간 차이가 크면 샷 전환이 발생한 블록이며, 그렇지 않으면 샷 전환이 발생하지 않았다고 판단한다. 프레임 내 모든 블록에 대한 샷 전환 여부를 계산한 후, 샷 전환이 나타난 블록의 수와 임계 값 θ_2 를 비교하여 프레임의 샷 전환 여부를 결정한다.

제안 알고리즘은 샷 전환 탐지의 정확도를 높이기 위해 프레임의 전역 히스토그램 정보를 사용하지 않고 그림 2와 같이 $B_r \times B_c$ 개의 일정한 크기로 프레임을 분할하여 각 블록의 히스토그램 정보를 사용한다.

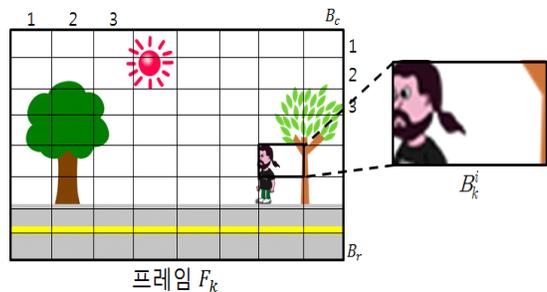


그림 2. 프레임 블록화의 예
Fig. 2. An example of a frame segmentation

그림 2에서 B_r 은 블록의 행의 수를, B_c 는 블록의 열의 수를, B_k^i 는 i 번째 블록을 의미한다. 프레임을 일정 크기의 블록으로 분할 한 뒤, 동일 인덱스 블록끼리 비교하기 위해서 각 블록별로 인덱스를 설정한다. 프레임 F_k 의 (x, y) 에 위치한 화소 $F_k(x, y)$ 의 블록 인덱스 i 는 식 (8)로 계산한다.

$$i = \left\lfloor \frac{y}{Y_b} \right\rfloor \times B_c + \left\lfloor \frac{x}{X_b} \right\rfloor + 1 \quad (8)$$

식 (8)에서 X_b 와 Y_b 는 블록의 화소 크기이다. 프레임 F_k 를 분할 한 뒤, 각 블록의 화소 값을 통해 히스토그램을 계산하여 H_k^i 에 저장한다.

블록 간 비교는 F_k 프레임의 블록과 F_{k+1} 프레임의 이동블록들의 히스토그램 차이로부터 계산된다. B_k^i 블록의 비교 대상인 이동블록은 F_{k+1} 의 $B_{k+1}^i(1)$ 블록을 대각선 방향으로 이동시켜 생성한다. 그림 3은 B_k^i 에 대해 F_{k+1} 의 비교 이동블록 $B_{k+1}^i(l)$, $l = 1, 2, 3, 4, 5$ 의 이동블록 생성과정을 보여 준다.

이동블록의 이동거리는 사전 실험을 통해 표본 영상에서 객체의 급격한 이동으로 인해 오 탐지가 발생하는 경우의 평균 이동 비율인 블록 크기의 20%로 설정 하였다. 블록의 이동거리가 평균 이동 비율보다 낮은 경우, 급격한 움직임을 보이는 객체가 이동블록 내에 포함되지 않았으며, 큰 경우에는 배경의 변화로 인해 오 탐지가 발생하였다. 그림 3의 빗금으로 채워진 영역은 $B_{k+1}^i(1)$ 블록을 의미하며, 각 이동 블록의 이동 방향은 화살표로 나타내었다. 이동블록의 인덱스는 $B_{k+1}^i(1)$ 을 기준으로 시계방향으로 지정하였다.

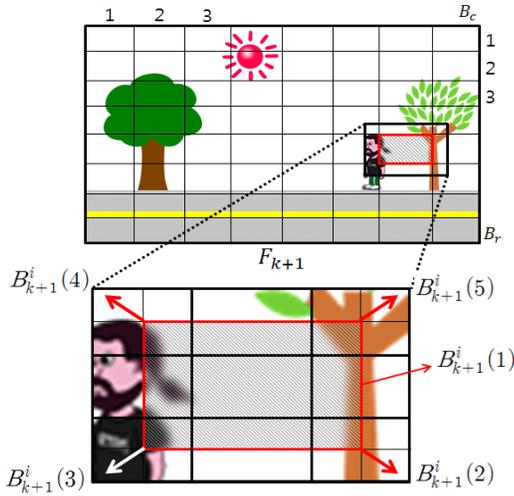


그림 3. F_{k+1} 의 $B_{k+1}^i(1)$ 을 이용한 이동블록
Fig. 3. An example of moving blocks for F_{k+1} over B_k^i

그림 3과 4의 B_k^i 와 $B_{k+1}^i(1)$ 를 1:1로 비교 할 경우, 사람이 좌측으로 이동하여 B_k^i 영역을 벗어났기 때문에 히스토그램에서 큰 차이를 보인다. 그러나 B_k^i 을 $B_{k+1}^i(4)$ 이동블록과 비교 하게 되면, 차이는 샷 전환을 판단하는 임계 값 θ_1 보다 작을 가능성이 높다. $B_{k+1}^i(1)$ 블록으로부터 생성된 이동블록의 히스토그램은 $H_{k+1}^i(l)$ 에 저장한다. B_k^i 와 $B_{k+1}^i(1)$ 블록의 샷 전환 여부를 결정하는 $h_i(k, k+1)$ 는 식 (9)로 계산한다.

$$h_i(k, k+1) = \min_{(0 < l \leq 5)} \left(\frac{|H_k^i - H_{k+1}^i(l)|}{X_b Y_b} \right) \quad (9)$$

식 (9)의 l 은 현재 비교에 사용된 이동블록을 의미한다. 샷 전환 여부를 나타내는 $h_i(k, k+1)$ 는 H_k^i 와 이동블록의 히스토그램인 $H_{k+1}^i(l)$ 을 모두 비교하여 최솟값으로 결정한다. $h_i(k, k+1)$ 값은 클수록 두 블록 간의 히스토그램 차이가 크다는 것을 의미한다.

블록의 샷 전환 여부를 판단하는 임계 값 θ_1 은 Otsu 임계 값 설정 방법[10,11]을 이용해 자동으로 설정한다. Otsu 임계 값의 계산에 사용되는 클래스 내 분산은 식 (10)을 이용하여 계산한다.

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad (10)$$

가중치 ω_1 과 ω_2 는 해당 화소가 나타날 확률이며, σ_1 과 σ_2 는 각 클래스의 분산 값이다. 히스토그램을 두 클래스로 나누는 임계 값 t 는 히스토그램의 빈 값이다. 모든 t 에 대해 $\sigma_w^2(t)$ 값을 구하고 난 뒤, $\sigma_w^2(t)$ 값이 가장 작을 때의 t 를 Otsu 임계 값으로 설정한다[10,11].

θ_1 을 설정하기 위해 영상을 구성하는 N 개의 모든 프레임에 대해 식 (10)을 이용하여 Otsu 임계 값을 계산한 후, $T = \{t_1, t_2, \dots, t_{N-1}\}$ 에 저장하였다. Kang 등[4]과 같은 방법으로 연속 프레임 간 두 블록의 Otsu 임계 값을 비교하여 차이가 3보다 큰 경우, 두 블록 사이에서 샷 전환이 발생한 것으로 판단하고 있다. 제안하는 샷 전환 탐지 알고리즘에서는 연속 프레임 간 Otsu 임계 값 차이가 3보다 큰 경우, 샷 전환 후보군으로 분류한다. Otsu 임계 값 차이에 대한 허용치를 3으로 설정한 경우, 실제 모든 샷 전환 프레임이 샷 전환 후보군으로 분류되어 적절한 θ_1 을 얻을 수 있었다. θ_1 은 샷 전환 후보군의 Otsu 임계 값들의 평균으로 설정한다. 블록의 샷 전환 여부를 나타내는 b_i 는 식 (11)을 이용해 계산한다.

$$b_i = \begin{cases} 1, & \text{if } h_i(k, k+1) > \theta_1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$h_i(k, k+1)$ 가 임계 값 θ_1 보다 크면, B_k^i 블록의 샷 전환 여부를 나타내는 b_i 는 식 (11)에 의해서 샷 전환이 발생한 블록으로 판단하여 1로 설정된다. 모든 블록의 샷 전환 탐지 결과 b_i 를 계산한 후, 프레임의 샷 전환을 탐지하는 d_k 는 식 (12)에 의해 계산된다.

$$d_k = \begin{cases} 1, & \text{if } \sum_{i=0}^{B_r \times B_c} b_i > \theta_2 \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

식 (12)에서 θ_2 는 전체 블록 중 샷 전환 블록의 비율을 나타내는 임계 값이며, 사전 실험을 통해 표본 영상에서 샷 전환이 발생한 경우의 블록 간 히스토그램 차이가 θ_1 이상인 블록 수의 평균으로 설정하였다. b_i 값이 1인 블록의 수를 통해 연속 프레임 간 샷 전환이라고 판단된 블록의 비율을 계산한 후, θ_2 이상이면 식 (12)에 의해 $d_k=1$ 이 되고 F_k 와 F_{k+1} 프레임 사이에서 샷 전환 탐지를 결정한다.

IV. 실험

제안하는 샷 전환 탐지 알고리즘의 성능을 평가하기 위하여 다양한 타입의 영상을 사용하였다. 1960년대부터 1980년대 사이에 촬영 된 흑백 또는 컬러의 국가기록원 디지털화 영화필름, 2005년 이후 촬영 된 드라마와 영화 그리고 사람이 인위적으로 색을 입힌 애니메이션 영상으로 실험을 진행하였다. 국가기록원 영상은 텔레시네(Telecine) 장치를 사용하여 아날로그 영상을 디지털로 변환 한 영상이다.

본 논문의 실험에서는 앞서 준비한 다양한 영상들을 중복 되지 않게 4분 단위로 분할하였으며, 분할 된 영상의 앞 5,000 프레임으로 실험을 진행하였다. 실험에 사용 된 영상의 타입과 프레임 그리고 실제 샷 전환의 수는 표 1과 같다.

표 1. 실험에 사용된 비디오
Table 1. The tested videos

| 타입 | 비디오 | 프레임 수 | 샷 전환 수 |
|-------------|-----|-------|--------|
| 영화 (흑백) | V1 | 5,000 | 59 |
| | V2 | 5,000 | 79 |
| 영화 (컬러) | V3 | 5,000 | 51 |
| | V4 | 5,000 | 79 |
| 드라마 (흑백) | V5 | 5,000 | 61 |
| | V6 | 5,000 | 67 |
| 드라마 (컬러) | V7 | 5,000 | 55 |
| | V8 | 5,000 | 67 |
| 기록원 (흑백) | V9 | 5,000 | 44 |
| | V10 | 5,000 | 47 |
| 기록원 (컬러) | V11 | 5,000 | 28 |
| | V12 | 5,000 | 21 |
| 애니 (컬러) | V13 | 5,000 | 46 |
| | V14 | 5,000 | 40 |

그리고 실험에 사용된 PC의 사양은 Intel Core 2 Duo E7500 CPU 2.93GHz, 4GB RAM, 32비트 Windows 7 운영체제이며, OpenCV 2.4.9를 사용 하였다.

제안 알고리즘과 다른 알고리즘의 성능을 비교하기 위해서 다음 4가지 방법을 사용하여 실험을 진행하였다. 제안 알고리즘, 1:1 블록매칭, 전역 화소비교, 전역 히스토그램비교 방법을 각각 실험 영상에 적용하여 성능을 평가하였다. 실험에서는 영상을 블록으로 분할하는 B_r 과 B_c 를 $2^k, k=1,2 \dots 5$ 까지 변화시켜, 샷 전환 탐지 비율이 높은 $k=3$ 으로 설정하

표 2. 테스트 비디오의 샷 전환 탐지 수
Table 2. The number of detected cuts for the tested videos

| 타입 | 비디오 | Cut | 제안 알고리즘 | | | 블록 기반 비교 | | | 화소 비교 | | | 히스토그램 비교 | | |
|-------------|-----|-----|---------|---|---|----------|---|----|-------|---|-----|----------|----|----|
| | | | D | M | N | D | M | N | D | M | N | D | M | N |
| 영화 (흑백) | V1 | 59 | 59 | 0 | 2 | 58 | 1 | 29 | 58 | 1 | 5 | 44 | 15 | 0 |
| | V2 | 79 | 79 | 0 | 0 | 79 | 0 | 6 | 77 | 2 | 5 | 53 | 26 | 0 |
| 영화 (컬러) | V3 | 51 | 51 | 0 | 0 | 51 | 0 | 0 | 51 | 0 | 3 | 43 | 8 | 0 |
| | V4 | 79 | 79 | 0 | 0 | 79 | 0 | 1 | 78 | 1 | 6 | 56 | 23 | 0 |
| 드라마 (흑백) | V5 | 61 | 61 | 0 | 0 | 60 | 1 | 8 | 61 | 0 | 6 | 54 | 7 | 0 |
| | V6 | 67 | 66 | 1 | 0 | 67 | 0 | 7 | 66 | 1 | 6 | 59 | 8 | 0 |
| 드라마 (컬러) | V7 | 55 | 55 | 0 | 0 | 54 | 1 | 2 | 54 | 1 | 3 | 51 | 4 | 1 |
| | V8 | 67 | 67 | 0 | 0 | 67 | 0 | 1 | 67 | 0 | 2 | 59 | 8 | 0 |
| 기록원 (흑백) | V9 | 44 | 41 | 3 | 1 | 44 | 0 | 8 | 39 | 5 | 200 | 33 | 11 | 2 |
| | V10 | 47 | 47 | 0 | 7 | 47 | 0 | 17 | 44 | 3 | 76 | 45 | 2 | 13 |
| 기록원 (컬러) | V11 | 28 | 28 | 0 | 0 | 27 | 1 | 3 | 28 | 0 | 45 | 27 | 1 | 3 |
| | V12 | 21 | 21 | 0 | 0 | 21 | 0 | 2 | 21 | 0 | 11 | 20 | 1 | 2 |
| 애니 (컬러) | V13 | 46 | 46 | 0 | 0 | 46 | 0 | 0 | 46 | 0 | 16 | 42 | 4 | 0 |
| | V14 | 40 | 40 | 0 | 0 | 40 | 0 | 0 | 40 | 0 | 0 | 36 | 4 | 3 |

표 3. 비디오에 대한 성능 평가

Table 3. The performance evaluation of the tested videos

| 타입 | 비디오 | 제안 알고리즘 | | | 블록 기반 비교 | | | 화소 비교 | | | 히스토그램 비교 | | |
|-------------|-----|---------|------|------|----------|------|------|-------|------|------|----------|------|------|
| | | rec | pre | F-m | rec | pre | F-m | rec | pre | F-m | rec | pre | F-m |
| 영화 (흑백) | V1 | 1.00 | 0.97 | 0.98 | 0.98 | 0.67 | 0.79 | 0.98 | 0.92 | 0.95 | 0.75 | 1.00 | 0.85 |
| | V2 | 1.00 | 1.00 | 1.00 | 1.00 | 0.93 | 0.96 | 0.97 | 0.94 | 0.96 | 0.67 | 1.00 | 0.80 |
| 영화 (컬러) | V3 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.94 | 0.97 | 0.84 | 1.00 | 0.91 |
| | V4 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 | 0.93 | 0.96 | 0.71 | 1.00 | 0.83 |
| 드라마 (흑백) | V5 | 1.00 | 1.00 | 1.00 | 0.98 | 0.88 | 0.93 | 1.00 | 0.91 | 0.95 | 0.89 | 1.00 | 0.94 |
| | V6 | 0.99 | 1.00 | 0.99 | 1.00 | 0.91 | 0.95 | 0.99 | 0.92 | 0.95 | 0.88 | 1.00 | 0.94 |
| 드라마 (컬러) | V7 | 1.00 | 1.00 | 1.00 | 0.98 | 0.96 | 0.97 | 0.98 | 0.95 | 0.96 | 0.93 | 0.98 | 0.95 |
| | V8 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 0.99 | 1.00 | 0.97 | 0.99 | 0.88 | 1.00 | 0.94 |
| 기록원 (흑백) | V9 | 0.98 | 0.96 | 0.97 | 1.00 | 0.76 | 0.86 | 0.96 | 0.42 | 0.58 | 1.00 | 0.70 | 0.83 |
| | V10 | 1.00 | 0.87 | 0.93 | 1.00 | 0.73 | 0.85 | 0.94 | 0.37 | 0.53 | 0.96 | 0.78 | 0.86 |
| 기록원 (컬러) | V11 | 1.00 | 1.00 | 1.00 | 0.96 | 0.90 | 0.93 | 1.00 | 0.38 | 0.55 | 0.96 | 0.90 | 0.93 |
| | V12 | 1.00 | 1.00 | 1.00 | 1.00 | 0.91 | 0.95 | 1.00 | 0.66 | 0.79 | 0.95 | 0.91 | 0.93 |
| 애니 (컬러) | V13 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.74 | 0.85 | 0.91 | 1.00 | 0.95 |
| | V14 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.90 | 0.92 | 0.91 |
| 평균 | | 1.00 | 0.99 | 0.99 | 0.99 | 0.90 | 0.94 | 0.99 | 0.79 | 0.86 | 0.87 | 0.94 | 0.90 |

었다. 그 결과 총 64개의 블록으로 프레임이 분할되었다. 더 작은 크기로 분할하게 되면 샷 전환 탐지에 소요되는 시간이 증가하게 되고, 큰 크기로 분할하게 되면 블록 내에서 객체 이동을 반영하는 히스토그램의 비율이 낮아져 샷 전환 탐지율이 낮아졌다.

실험에 사용된 θ_1 은 Otsu 임계 값 설정 방법[10,11]을 사용하여 영상마다 자동으로 설정되게 하였다. θ_2 는 사전 실험을 통해 실험 영상과 별개로 각 타입별로 하나의 표본 영상을 생성하여, θ_1 보다 차이가 큰 블록 수의 평균인 49로 설정하였다. 이는 전체 블록의 77%에 해당하는 수치이다. 설정한 임계 값은 제안하는 샷 전환 탐지 방법뿐만 아니라, 동일한 조건에서 성능을 측정하기 위해 다른 샷 전환탐지 방법에도 동일하게 적용시켰다.

실험 결과 표 2의 D (detection)는 알고리즘이 탐지 한 샷 전환의 수, M (miss cut)은 실제로는 샷 전환이 발생했지만 알고리즘이 탐지하지 못한 샷 전환의 수, N (not cut)은 실제 샷 전환이 발생하지 않았지만 샷 전환이 발생했다고 오 탐지된 수이다. 화소 값을 이용한 방법은 프레임 간 작은 변화에도 민감하게 반응하여 샷 전환이라고 잘 못 판단하는 경우가 많이 발생하였다. 히스토그램을 이용한 방법은 프레임의 전체 화소 값의 분포를 이용해 비교하기 때문에, 두 프레임 간 대응되는 화소의 값이 바뀌어도 오 탐지를 하는 경우가 적었다. 그러나 프레임 내 색상의 분포가 비슷한 경우에 샷 전환으로 탐지하지 못하였다. 블록을 이용한 방법은 프레임을 분할하지 않은 히스토그램과 화소 비교방법에 비해 정확도가

높으며 샷 전환을 오 탐지한 컷의 수도 적다. 제안하는 방법은 이동블록과의 비교를 통해 히스토그램을 이용한 방법의 단점이었다던 색상의 분포로 인한 M (miss cut)을 낮출 수 있었다. 프레임을 분할함으로써, 하나의 블록이 픽셀의 역할을 하여, 프레임 전체의 히스토그램을 이용한 비교방법보다 민감하게 프레임 간 변화에 반응한 결과로 분석된다.

제안하는 샷 전환 탐지 방법은 4개의 이동 블록을 이용하여, 기존의 블록을 이용한 비교방법의 객체의 움직임과 같은 급격한 변화로 인해 발생하는 오 탐지를 줄였다. 이동 블록뿐만 아니라, 이동시키지 않은 블록도 포함하여 비교를 함으로써, 기존 블록 비교 방법이 갖는 정확도를 유지 하였다.

실험 결과의 일반화 성능 평가를 위해 식 (13), (14), (15)를 이용해 재현율(recall), 정밀도(precision),

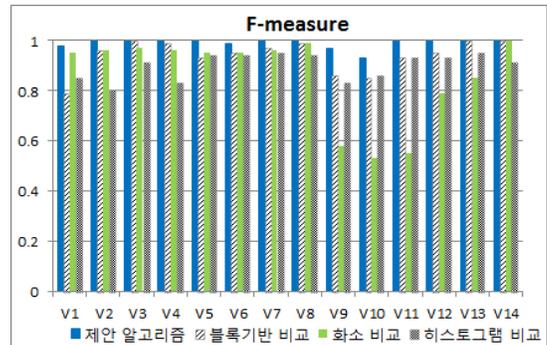


그림 4. 테스트 비디오의 F -measure 평가 결과
Fig. 4. The F -measure result of the tested videos

F -measure로 평가하여 표 3에 제시하며, F -measure 평가 결과는 그림 4에 나타낸다.

$$rec = \frac{D}{D+M} \quad (13)$$

$$pre = \frac{D}{D+N} \quad (14)$$

$$F\text{-measure} = 2 \times \frac{pre \times rec}{pre + rec} \quad (15)$$

식 (13)에서 rec 값이 1이면, 모든 샷 전환을 탐지했음을 의미하며, 식 (14)에서 pre 값이 1이면 샷 전환 알고리즘의 예측이 오 탐지 없이 정확했음을 의미한다. 식 (15)에서 F -measure의 값이 1에 가까울수록 제안하는 샷 전환 탐지 알고리즘의 정확도와 예측율이 높다고 평가 된다. 결과로부터, 제안 알고리즘은 화소 값 비교방법과 히스토그램 비교 방법보다 정확도를 높였으며, 이동 블록을 사용하여 기존의 블록 기반 비교방법보다 오 탐지율을 낮췄다.

V. 결 론

본 논문에서는 이동 블록을 사용하여 샷 전환을 탐지하는 방법을 제안하였다. 기존의 샷 전환 탐지 알고리즘은 객체의 급격한 이동이나 변화로 인해 오 탐지가 많이 발생한다. 또한 임계 값이 사전 실험을 통해 고정 된 값으로 설정되기 때문에, 다양한 타입의 영상에 있어서 모두 좋은 결과를 얻을 수 없는 문제점이 있었다.

제안하는 샷 전환 탐지 알고리즘은 이동블록을 사용하여 오 탐지율을 줄이는데 목적이 있으며, Otsu 임계 값 설정 방법을 사용하여 각 영상에 알맞은 임계 값을 자동으로 추출 한다. 실험에서 제안하는 샷 전환 탐지 알고리즘은 기존의 샷 전환 탐지 알고리즘보다 정확도가 높았으며, 오 탐지한 샷의 수도 줄일 수 있었다. 또한 영화, 드라마, 흑백영상이나 애니메이션 등의 다양한 타입의 영상에 대해서 모두 비슷한 정확도를 보여주었다.

제안하는 알고리즘은 비교 연구방법에 비해 샷 전환 탐지에 소요되는 시간이 증가하였다. 샷 전환 탐지 이전에 임계 값 θ_1 을 설정하는 과정, 이동블록의 1:5 히스토그램 비교는 전체 연산시간을 증가시키는 요인이다. 제안 알고리즘의 히스토그램 연산, 이동블록 생성과 같은 연산은 병렬 처리 가능성이 높아, CUDA 또는 OpenCL과 같은 GPU 기반의 병렬 아키텍처를 이용하면 전체 연산 시간을 단축 할 수 있다.

REFERENCES

- [1] H. J. Zhang, A. Kankanhalli, S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, Vol. 1, No. 1, pp. 10-28, 1993.
- [2] Xiaoquan Yi, Nam Ling, "Fast Pixel-Based Video Scene Change Detection," *IEEE Int. Symposium on circuits and Systems*, Vol. 4, pp. 3343-3346, May 2005.
- [3] S. Y. Tak, S. Yoo, B. R. Lee, W. J. Lee, K. S. Ryu, T. G. Kim, H. C. Kang, "Scene change detection of various color space using difference of histogram," *Proceedings of the Spring Conf. on The Korea Contents Association*, pp. 466-468, May 2010.
- [4] S. J. Kang, S. I. Cho, S. J. Yoo, Y. H. Kim, "Scene Change Detection Using Multiple Histograms for Motion-Compensated Frame Rate Up-Conversion," *Journal of Display Technology*, Vol. 8, No. 3, pp. 121-126, March 2012.
- [5] S. G. Hong, B. K. Cho, Y. S. Choe, "Adaptive Thresholding for Scene Change Detection," *IEEE Third Int. Conf. on Consumer Electronics - Berlin*, pp. 75-78, Sept. 2013.
- [6] S. Y. Shin, S. E. Baik, S. B. Pyo, Y. W. Rhee, "Scene Change Detection Using Local χ^2 ," *Journal of The Korea Society of Computer and Information*, Vol. 15, No. 1, pp. 203-207, June 2007.
- [7] J. K. Jin, J. H. Cho, J. H. Jeong, D. S. Jeong, "Fast Scene Change Detection Using Macro Block Information and Spatio-temporal Histogram," *Journal of The Institute of Electronics Engineers of Korea*, Vol. 48-SP, No. 1, pp. 141-148, Jan. 2011.
- [8] J. G. Lee, K. S. Han, B. M. Yoo, D. S. Hwang, "Shot Boundary Detection Algorithm by compensating Pixel Brightness and Object

- Movement," Journal of The Korea Society of Computer and Information, Vol. 18, No. 5, pp. 35-42, May 2013.
- [9] S. M. Go, H. G. Kim, M. S. Oh, "A Study on block histogram's comparison for cut detection," Journal of The Korean Institute of Maritime Information and Communication Sciences, Vol. 5, No. 7, pp. 1301-1307, Dec. 2001.
- [10] J. R. Parker, "Algorithms for Image Processing and computer vision 2nd Edition," wiley publishing, pp. 149-151, 2011.
- [11] Puneet, Naresh Kumar Garg, "Binarization Techniques used for Grey Scale Images," International Journal of Computer Applications, Vol. 71, No. 1, pp. 8-11, June 2013.

저 자 소 개



김 승 현
 2013: 단국대학교
 컴퓨터과학과 공학사
 현 재: 단국대학교
 전자계산학과 석사과정
 관심분야: Image Processing
 Parallel Processing
 Email : pcpp13@naver.com



황 두 성
 1986: 충남대학교 이학사
 1990: 충남대학교 이학 석사
 2003: Wayne State University 박사
 현 재: 단국대학교 컴퓨터과학과,
 운동의과학과 부교수
 관심분야: Machine Learning
 Parallel Processing
 Semantic Web
 Email : dshwang@dankook.ac.kr