

## 애니메이션의 효과적인 장면경계 검출 알고리즘

장석우<sup>1</sup>, 정명희<sup>1\*</sup>

<sup>1</sup>안양대학교 디지털미디어학과

## An Effective Detection Algorithm of Shot Boundaries in Animations

Seok-Woo Jang<sup>1</sup> and Myunghee Jung<sup>1\*</sup>

<sup>1</sup>Department of Digital Media, Anyang University

**요약** 셀 애니메이션은 배경이 하나의 셀로 표현되고, 장면이 변화될 경우에는 배경이 변경되기 때문에 장면전환 시 비교적 큰 변화가 일어난다. 그리고 실제로 카메라를 이용하여 촬영한 영상과는 달리 사람이 직접 그린다 보니 사용된 색상의 종류 또한 그렇게 많지 않다. 본 논문에서는 애니메이션의 이러한 특성을 최대한 반영하고 보다 효과적으로 셀 애니메이션의 장면전환을 검출하기 위해서 색상과 블록 단위의 히스토그램을 단계적으로 활용하는 새로운 애니메이션의 장면전환 검출 기법을 제안한다. 제안된 알고리즘은 연속적으로 입력되는 애니메이션 영상을 받아들인 후 먼저 칼라공간을 HSI 공간으로 변형하고, 두 영상 사이의 색상 값의 차연산을 수행하여 인접한 영상이 장면전환 후보인지를 1차적으로 판단한다. 만일, 인접한 영상이 장면전환 후보군으로 판단되면 부 영역별로 색상 히스토그램을 작성하고, 여기에 가중치를 적용하여 장면전환이 발생했는지의 유무를 최종적으로 판단한다. 본 논문의 실험에서는 제안된 애니메이션의 장면전환 검출 방법이 기존의 장면전환 검출 방법에 비해 보다 우수하다는 것을 보인다.

**Abstract** A cell animation is represented by one background cell, and there is much difference of images when its shot is changed. Also, it does not have a lot of colors since people themselves draw it. In order to effectively detect shot transitions of cell animations while fully considering their intrinsic characteristics, in this paper, we propose a animation shot boundary detection algorithm that utilizes color and block-based histograms step by step. The suggested algorithm first converts RGB color space into HSI color one, and coarsely decides if adjacent frames contains a shot transition by performing color difference operation between two images. If they are considered to have a shot transition candidate, we calculate color histograms for 9 sub-regions of the adjacent images and apply weights to them. Finally, we determine whether there is a real shot transition by analyzing the weighted sum of histogram values. In experiments, we show that our method is superior to others.

**Key Words** : Animations, Shot Boundaries, HSI Color Model, Features

### 1. 서론

최근 들어 초고속의 유무선 인터넷, 스마트 폰, 태블릿 PC 등과 같은 대중화된 기술 및 기기의 급속한 발전으로 인해 영화, 광고, 게임 등의 엔터테인먼트 분야에서 새로운 부가가치 창출을 위한 다양한 종류의 콘텐츠에 대한 수요가 점점 더 늘어나고 있다. 이러한 콘텐츠 중에서 애니메이션은 어린이에서부터 중·장년층에 이르기까지 인

기가 많은 콘텐츠이므로 널리 보급되고 있다.

일반적으로 애니메이션(animation)은 사람이 직접 그린 창작물이다. 따라서 애니메이션을 그린 사람에 따라서 각기 다른 모양의 주인공이 나오게 되지만 전체적인 구도는 기본 형식에 따르기 때문에 대부분 동일하다. 보통 애니메이션은 크게 모델 애니메이션, 3차원 애니메이션, 그리고 셀 애니메이션의 3가지 주요 부류로 분류될 수 있다[1]. 모델 애니메이션은 점토나 흙 등으로 사물을 제작

\*교신저자 : 정명희(mhjung@anyang.ac.kr)

접수일 11년 05월 31일

수정일 11년 06월 21일

게재확정일 11년 08월 11일



[그림 1] 애니메이션의 장면전환  
 [Fig. 1] Shot boundaries of animations

하고, 제작된 사물의 모션을 캡처하여 생성한 애니메이션이다. 3차원 애니메이션은 컴퓨터 그래픽 작업을 수행하여 3차원 공간에서 만든 애니메이션이다. 그리고 셀 애니메이션은 보통 TV나 극장에서 볼 수 있는 애니메이션으로 여러 장의 셀을 이용하여 배경, 주인공, 그리고 사물 등을 표현한다. 특히, 셀 애니메이션은 다른 방법에 비해서 애니메이션의 제작에 소요되는 시간이나 인력, 그리고 비용 등을 절감하기 위해서 많이 사용된다.

이런 애니메이션은 시간이 지남에 따라 그 양이 기하급수적으로 꾸준히 증가하고 있으므로 애니메이션 콘텐츠를 효과적으로 관리하고 색인하기 위한 영상처리 관련 기술의 필요성이 제기되고 있다. 특히, 애니메이션을 보다 체계적으로 관리하기 위해서는 무엇보다도 먼저 애니메이션에 대한 장면전환 검출이 선행되어야 한다. 보통 장면전환은 급진적인 장면전환인 컷과 점진적인 장면전환인 페이드 인, 페이드 아웃, 디졸브, 와이프 등으로 구성된다[2,3].

일반적인 장면전환을 검출하는 기존의 방법들은 관련 문헌에 소개되어 있다. [4]에서는 화소 단위의 차이값을 분석하여 장면 사이의 경계를 검출하였다. 즉, 인접한 두 영상 사이의 화소값 또는 칼라값이 차이가 나는 화소의 전체 개수가 일정한 임계치(threshold)를 넘으면 장면전환이 발생한다고 판단한다. 이 방법의 단점은 잡음이나 모션 정보에 민감하다는 점이다. [5,6]에서는 영상에 대한 명암값 또는 칼라값의 히스토그램을 작성하고, 두 개의 인접한 영상에 대한 히스토그램의 빈(bin) 단위의 차이가 미리 정의된 임계치를 초과하면 장면전환이 존재한다고 인식한다. 이 방법은 대체적으로 잘 동작한다고 알려져 있으나 유사한 명암값 분포를 가지는 서로 다른 장면이 동시에 존재할 경우에는 오류를 산출한다. [7]에서는 새롭게 발생하는 에지와 기존에 있다가 사라지는 에지의 비율을 이용하여 여러 가지 장면전환을 검출하는 방법을 고안하였다. 그런데 이 방법은 조명 등의 영향으로 인해 밝기값의 변화가 존재할 경우에는 에지를 검출하는 정확

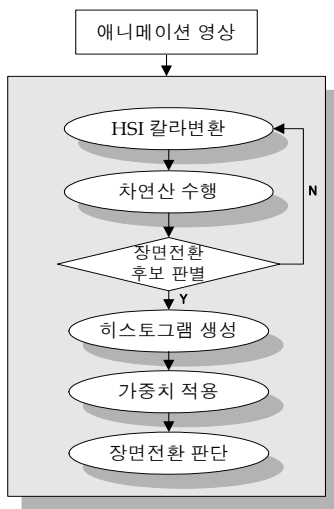
도가 상당히 저하되므로, 이런 상황에서는 강건하게 장면전환을 검출하지 못한다고 알려져 있다. [8,9]에서는 영상을 크기와 모양이 동일한  $N \times N$  크기의 사각형의 블록 단위로 분할하고, 블록 단위의 모션 정보를 추출한 후 모션 정보의 차이값을 이용해서 장면전환을 검출하는 방법을 제안하였다. 이 방법은 지역적 또는 전역적인 모션이 존재할 경우에는 프레임 단위의 비교 방법에 비해 보다 강건하다고 알려져 있으나 명암값의 변화가 존재할 경우에는 다소 오류가 발생한다.

일반적으로, 셀 애니메이션은 그림 1과 같이 배경이 하나의 셀로 표현되고, 애니메이션의 장면이 전환될 경우에는 배경이 변경되기 때문에 장면전환 시 비교적 큰 변화가 일어난다. 그리고 실제로 카메라를 이용하여 촬영한 영상과는 달리 사람이 직접 그리다 보니 사용된 색상의 종류 또한 그렇게 많지 않으며, 애니메이션의 장면전환은 대부분 컷으로 형성된다. 따라서 본 논문에서는 이러한 특성을 최대한 반영하고 보다 효과적으로 셀 애니메이션의 장면전환을 검출하기 위해서 색상과 블록 단위의 히스토그램을 단계적으로 활용하는 새로운 애니메이션의 장면전환 검출 기법을 제안한다.

2장에서는 장면전환의 후보군을 검출하는 방법에 대해 기술한다. 3장에서는 애니메이션을 부영역으로 분할하고, 분할된 영역 단위로 색상 히스토그램을 구하여 최종적인 애니메이션의 장면전환을 검출하는 기법에 대해 설명한다. 그리고 4장에서는 제안된 방법의 성능을 평가하기 위해 수행한 실험결과를 보이며, 5장에서는 결론 및 향후 연구방향을 제시한다.

## 2. 장면경계의 후보 검출

제안하는 색상과 블록 단위의 히스토그램을 단계적으로 활용하는 애니메이션의 장면전환 검출 시스템의 전체적인 개요도는 다음의 그림 2와 같다.



[그림 2] 전체 개요도  
[Fig. 2] Overall flow

그림 2에서와 같이 제안된 시스템은 연속적으로 입력 되는 애니메이션 영상을 받아들인 후 먼저 칼라공간을 HSI 공간으로 변경하고, 두 영상 사이의 색상 값의 차연산을 수행하여 인접한 영상이 장면전환 후보인지를 판단한다. 만일, 인접한 영상이 장면전환 후보군으로 판단되면 부 영역별로 색상 히스토그램을 작성하고, 여기에 가중치를 적용하여 장면전환이 발생했는지의 유무를 최종적으로 판단한다.

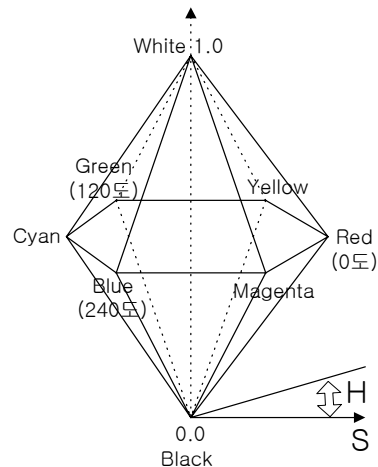
제안된 장면전환 검출 알고리즘은 먼저 입력된 영상의 RGB 칼라공간을 HSI 칼라공간으로 변환한다. 일반적으로, 칼라 영상을 코딩하기 위해서 RGB 칼라 공간을 많이 사용하지만, 영상처리를 수행하려 할 때 세 가지의 R, G, B 칼라 값을 모두 처리해야 하므로 RGB 칼라 공간을 사용하여 영상 처리를 수행하는 것은 비효율적이다. 그리고 RGB 칼라 공간은 영상의 밝기 변화에 민감하게 작용할 뿐만 아니라 동일한 영상의 동일한 영역에 대해서도 조명의 크기나 방향에 따라서 칼라 값의 차이가 매우 크게 발생할 수 있다. 이런 이유로 RGB 칼라 공간은 영상 처리에 적합하지 못하다고 알려져 있다[10]. 따라서 본 논문에서는 RGB 칼라 공간 대신 HSI 칼라 공간을 사용한다. 식 (1)은 RGB 칼라공간을 HSI 칼라공간으로 변환하는 수식을 보여준다. 식 (1)의 HSI 칼라 공간에서 색상 H(hue)는 0도에서 360도의 범위를 가지는 각도로 표현된다. 그리고 명도 I(intensity)는 검은색을 나타내는 0에서 흰색을 나타내는 255사이의 범위를 가지며, 채도 S(saturation)는 0에서 1사이의 범위를 가지는 값으로서 칼라의 순수한 정도를 나타낸다[11].

$$H = \cos^{-1} \left( \frac{[(R-G) + (R-B)]}{2 \sqrt{(R-G)^2 + (R-B)(G-B)}} \right) \quad (1)$$

$$I = \frac{1}{3} \times (R + G + B)$$

$$S = 1 - \frac{3}{(R + G + B)} \times [\min(R, G, B)]$$

그리고 그림 3은 HSI 칼라공간의 구조를 시각적으로 보여준다.



[그림 3] HSI 칼라공간  
[Fig. 3] HSI color space

본 논문에서는 장면전환 후보군을 1차적으로 검출하기 위해서 식 (2)와 같은 색상 척도를 활용한다.

$$\Phi_{t,t-1}(x,y) = \sum_{x,y} NoH_{t,t-1}(x,y) \quad (2)$$

$$NoH_{t,t-1}(x,y) = \begin{cases} DoH_{t,t-1}(x,y) & \text{if } DoH_{t,t-1}(x,y) \leq 180 \\ 360 - DoH_{t,t-1}(x,y) & \text{if } DoH_{t,t-1}(x,y) > 180 \end{cases}$$

$$DoH_{t,t-1}(x,y) = |H_t(x,y) - H_{t-1}|$$

식 (2)에서  $H_t(x,y)$ 와  $H_{t-1}(x,y)$ 는 t와 t-1시점 영상의 칼라 공간을 HSI 칼라공간으로 변환한 후 H 값만을 취한 영상이다. 따라서  $DoH_{t,t-1}(x,y)$ 은 두 영상에 차연산을 적용하여 생성한 영상이 되는데, RGB 칼라공간을 HSI 칼라공간으로 변환한 후 H 만을 사용하였기 때문에 값이 0에서 360 사이의 값으로 표현된다. 그런데 색상이 1이라는 값과 360이라는 값은 거의 동일한 색상임에도 불구하고 차연산을 수행하면 359라는 큰 값이 나온다.

따라서 이러한 현상을 보정하기 위해서 식 (2)에서와 같이  $NoH_{t,t-1}(x,y)$ 를 생성한다. 그런 다음, 생성된  $NoH_{t,t-1}(x,y)$ 를 모든 화소에 대해서 합산하여 색상 척도인  $\Phi_{t,t-1}(x,y)$ 를 생성하는데, 이 값이 정해진 임계치를 넘으면 애니메이션의 1차적인 장면전환 후보로 판단한다.

### 3. 애니메이션의 장면경계 검증

일반적으로, 애니메이션의 영상에서 장면이 변화하지 않을 경우에는 인접한 영상 내부에 포함된 색상의 수가 거의 동일하며, 색상들이 위치한 영역 또한 유사하다. 그리고 애니메이션 영상은 실세계와는 달리 인간이 구분하기 쉽게 하기 위해서 색상의 차가 명확하다. 따라서 색상의 히스토그램을 이용한 기존의 장면전환 검출 방법들은 애니메이션의 동일한 장면을 마치 장면전환이 발생한 것으로 잘못 인식하는 경우가 종종 발생한다. 예를 들어, 그림 4와 같이 동일한 배경일지라도 캐릭터가 차지하는 영역이 비교적 크고, 한 프레임 사이에서 많은 이동이 진행된다면 이러한 현상이 발생하기 쉽다. 특히, 그림 4와 같이 캐릭터가 한 프레임 안에서 완전히 회전하는 경우에는 자주 오 동작을 일으키는 경향이 있다.



(a)



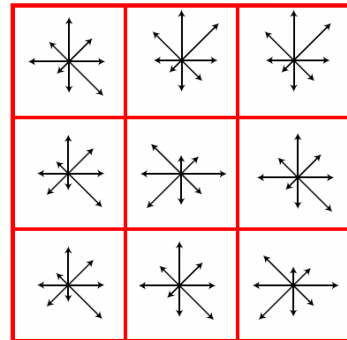
(b)

[그림 4] 동일한 장면  
[Fig. 4] Same scene

본 논문에서는 이러한 문제를 해결하기 위해서 두 가지의 방법을 사용한다. 첫째, 장면전환이 발생할 확률이 높은 후보로 판단된 애니메이션 영상을 받아들인 후 전체 영상을 크기가 동일한 사각형으로 구성된 3×3 블록으로 분할한다. 그런 다음, 각 블록에 대해서 색상 H 값의 히스토그램을 작성한다. 이 때, H 값은 0도에서 360도 사이의 값을 가지므로 본 논문에서는 360도를 45도씩 8등분하고, 각각의 분할된 범위에 대해서 그림 5와 같이 색상 히스토그램을 작성한다. 그림 5에서 화살표의 길이는 8개의 분할된 범위에서 색상의 누적합을 의미한다.

그리고 식 (3)과 같이 인접한 두 개의 영상에서 획득한 히스토그램의 차이의 합을 9개의 블록별로 각각 계산한다. 식 (3)에서  $H_t^k(i)$ 는 t 시점 영상의 k번째 블록에 대한 히스토그램이다.

$$SoH_{t,t-1}(k) = \sum_{i=1}^8 |H_t^k(i) - H_{t-1}^k(i)| \quad (3)$$



[그림 5] 부영역의 히스토그램  
[Fig. 5] Histogram of sub-regions

둘째, 애니메이션 영상의 경우 대체적으로 주인공이 영상의 중앙에 위치하고 배경은 그 나머지에 위치하므로, 그림 6과 같이 설정한 가중치를 분할된 영역의 해당되는 히스토그램의 합인  $SoH_{t,t-1}(k)$ 에 적용한다.

3	2	3
2	1	2
3	2	3

[그림 6] 영역별 가중치  
[Fig. 6] weighting factors

식 (4)는 이 과정을 수식으로 보여주는데,  $M(t, t-1)$ 은 인접한 영상 사이의 장면전환 유무를 판단하는 척도가 된다. 그런 다음,  $M(t, t-1)$  값이 미리 정해진 임계치를 넘으면 장면이 전환되었다고 판단하고, 임계치를 넘지 못하면 장면이 전환되지 않았다고 판단한다.

$$M(t, t-1) = \sum_{k=1}^9 SoH_{t,t-1}(k) * w(k) \quad (4)$$

#### 4. 실험결과

본 논문의 실험을 위해 인텔 Pentium-4의 3.0GHz CPU와 1GB의 메모리를 사용하였고, 운영체제로는 Windows XP를 사용하였다. 그리고 장면전환 구현을 위해서는 Visual Studio C++ 6.0을 이용하였으며, 영상처리 오픈 라이브러리인 OpenCV 1.1도 사용하였다. 입력으로 사용된 영상으로는 해상도가 640× 480이고, 초당 프레임수가 23.976이며, 편당 24분짜리 애니메이션 영상 3편을 실험에 사용하였다.

본 논문에서는 제안된 장면전환 검출 방법의 성능을 기존의 장면전환 검출 방법과 비교 평가하기 위해서 기존의 방법 중 블록 기반의 장면전환 검출 방법[8]과 히스토그램 기반의 장면전환 검출 방법[5]도 구현하였다. 기존의 두 방법 모두 비교적 알고리즘이 복잡하지 않으며, 장면전환을 추출하기 위해서 널리 사용되는 방법들이다.

본 논문의 실험에서 사용한 애니메이션 영상의 프레임수는 총 103,548 프레임이며, 검출된 장면의 변화 개수는 792 프레임이었다. 그리고 장면전환 검출의 정확도는 약 97%의 검출률을 보여 주었다. 장면전환에 대한 정확도는 식 (5)와 같이 애니메이션 영상에 존재하는 장면전환을 정확하게 검출했는지 아니면 검출하지 못했는지 만을 백분율로 측정하였다.

$$Ratio = \frac{\text{number of detected shots}}{\text{number of existing shots}} \quad (5)$$

본 논문의 실험을 통해 결과적으로 평가해 볼 때 기존의 방법은 입력되는 애니메이션 영상들이 실제 영상과 비슷하게 복잡할 경우에는 실제 영상에서의 장면전환 검출 결과와 유사한 정확도를 보였으나, 애니메이션 영상이 복잡하지 않고 단순할 경우에는 오히려 정확도가 저하되는 것으로 파악되었다.

표 1, 표 2, 그리고 표 3은 블록 기반의 장면전환 검출 방법, 히스토그램 기반의 장면전환 방법, 제안된 검출 방

법으로 실험한 결과를 각각 표로 보여준다. 그리고 그림 7은 기존의 방법과 제안된 방법의 애니메이션 장면전환 검출 결과의 정확도를 그래프로 보여주는데, 그림 7에서 확인할 수 있듯이 제안된 장면전환 검출 방법이 다른 방법에 비해 보다 우수한 성능을 보임을 알 수 있다.

그리고 그림 8은 제안된 장면전환 검출 방법을 이용해서 장면전환이 발생하는 애니메이션 영상을 정확하게 검출하는 실행 화면의 예를 보여준다.

[표 1] 블록 기반의 방법

[Table 1] Block-based method

애니메이션	프레임수	검출된 장면변환	정확도(%)
1	34525	223	80.29
2	34500	175	77.66
3	34523	276	88.63

[표 2] 히스토그램 기반의 방법

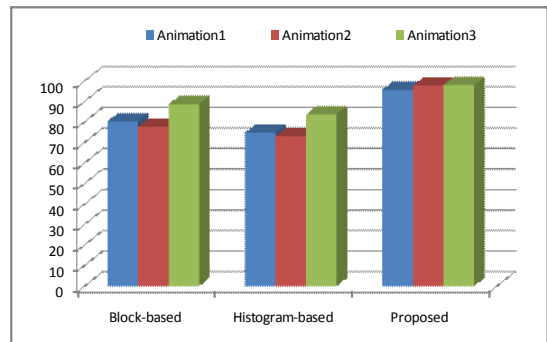
[Table 2] Histogram-based method

애니메이션	프레임수	검출된 장면변환	정확도(%)
1	34525	207	74.75
2	34500	164	73.01
3	34523	260	83.66

[표 3] 제안된 방법

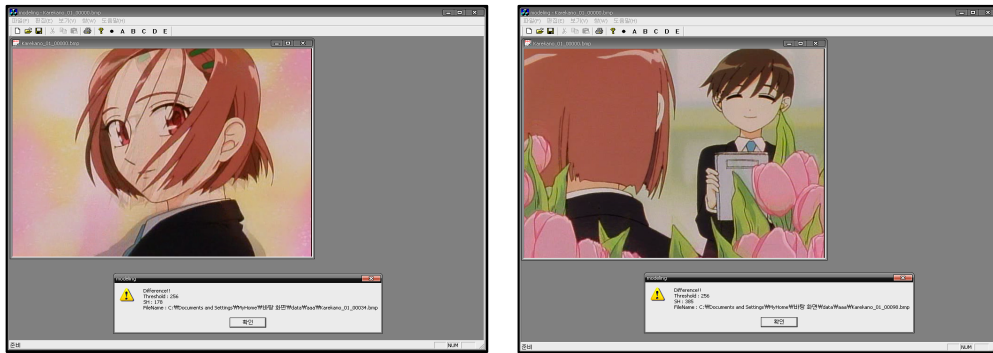
[Table 3] Proposed method

애니메이션	프레임수	검출된 장면변환	정확도(%)
1	34525	266	95.68
2	34500	220	97.77
3	34523	306	98.01



[그림 7] 성능 평가

[Fig. 7] Performance evaluation



[그림 8] 장면전환 검출  
[Fig. 8] Shot boundary detection

### 5. 결론

영화, 게임, UCC, IPTV 등의 엔터테인먼트 분야에서 새로운 부가가치 창출을 위한 다양한 종류의 콘텐츠에 대한 수요가 점점 더 늘어나고 있다. 이러한 콘텐츠 중에서 애니메이션은 어린이에서부터 중·장년층에 이르기까지 인기가 많은 콘텐츠이므로 널리 보급되고 있는데, 애니메이션 콘텐츠를 효과적으로 관리 및 색인하기 위한 기술이 필요한 실정이다.

본 논문에서는 색상과 영역별 히스토그램을 단계적으로 적용하는 애니메이션의 장면경계를 검출하는 알고리즘을 새롭게 제안하였다. 제안된 알고리즘은 연속적으로 입력되는 애니메이션 영상을 받아들인 후 칼라공간을 HSI 공간으로 변경하고, 두 영상 사이의 색상 값의 차연산을 수행하여 인접한 영상이 장면전환 후보인지를 1차적으로 판단한다. 만일, 인접한 영상이 장면전환 후보군으로 판정되면 부 영역별로 색상 히스토그램을 작성하고, 여기에 가중치를 적용하여 장면전환이 발생했는지의 여부를 최종적으로 검증한다. 실험에서는 제안된 방법이 기존의 방법에 비해 비교적 정확하게 애니메이션의 장면전환을 검출함을 보인다.

향후 연구로는 다른 애니메이션 영상을 입력받아 제안된 방법의 성능을 다양하게 비교 평가할 예정이며, 제안된 알고리즘에 사용된 여러 가지 인수를 안정화하고, 자동화하기 위한 방안을 모색할 예정이다.

### Reference

[1] S.-H. Lee, S.-O. Yang, and Y.-T. Paik, "Multimedia," Jungil Books, March 2005.  
[2] S.-W. Jang, G.-Y. Kim, and H. I. Choi, "Shot Transition

Detection by Compensating for Global and Local Motions," In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1061-1066, 2005.

[3] A.F. Smeaton, P. Over, A.R. Doherty, "Video Shot Boundary Detection: Seven Years of TRECVID Activity," Computer Vision and Image Understanding, Vol. 114, No. 4, pp. 411-418, 2011.  
[4] M.-S. Lee, Y.-M. Yang, and S.-W. Lee, "Automatic Video Parsing Using Shot Boundary Detection and Camera Operation Analysis," Pattern Recognition, Vol. 34, No. 3, pp. 711-719, March 2001.  
[5] C. C. Lo and S. J. Wang, "A Histogram-based Moment-Preserving Clustering Algorithm for Video Segmentation," Pattern Recognition Letters, Vol. 24, No. 14, pp. 2209-2218, October 2003.  
[6] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-motion Video," Multimedia Systems, Vol. 1, No. 1, pp. 10-28, 1993.  
[7] T. Y. Liu, K. T. Lo, X. D. Zhang, and J. Feng, "A New Cut Detection Algorithm with Constant False-Alarm Ratio for Video Segmentation," Journal of Visual Communication and Image Representation, Vol. 15, No. 2, pp. 132-144, 2004.  
[8] C. A. Dhawale and S. Jain, "Motion Compensated Video Shot Detection Using Multiple Feature Experts," ICGST International Journal on Graphics, Vision, and Image Processing, Vol. 8, No. 5, pp. 1-11, 2009.  
[9] Y.-U. Han, S.-I. Jung, S.-J. Kim, S.-Y. Lee, and S.-H. Kim, "Improving Histogram Scene Change Detection Method Using Motion Vector," In Proceedings of the Fall Conference of the Korea Institute of Information Scientists and Engineers, Vol. 26, No. 2, pp. 410-412, 1999.

- [10] F. Perez and C Koch, "Toward Color Image Segmentation in Analog VLSI: Algorithm and Hardware," International Journal of Computer Vision, Vol. 12, No. 1, pp. 17-42, 1994.
- [11] K. S. Tana and N. A. M. Isa, "Color Image Segmentation Using Histogram Thresholding - Fuzzy C-Means Hybrid Approach," Pattern Recognition, Vol. 44, No. 1, pp. 1-15, 2011.

---

**장 석 우(Seok-Woo Jang)**

[정회원]



- 1995년 2월 : 숭실대학교 전자계학과 (공학사)
- 1997년 2월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2000년 8월 : 숭실대학교 컴퓨터학과 (공학박사)
- 2011년 6월 ~ 현재 : 안양대학교 디지털미디어학과 조교수

<관심분야>

로봇비전, 증강현실, HCI, 비디오 색인 및 검색, 등

---

**정 명 희(Myunghee Jung)**

[정회원]



- 1989년 : 서울대학교 계산통계학과 (공학사)
- 1991년 : University of Texas, Austin, (공학석사)
- 1997년 : University of Texas, Austin, (공학박사)
- 2011년 6월 ~ 현재 : 안양대학교 디지털미디어학과 교수

<관심분야>

통신, 시뮬레이션, 원격탐사 데이터 처리, 영상처리 멀티미디어 등