

장면전환 검출 알고리즘의 구현 및 비교

김경욱*, 이효종

*전북대학교 전자정보공학부

전북대학교 공업기술연구소

e-mail:{kwkim, hlee}@sel.chonbuk.ac.kr

Comparison of Algorithms for Shot Change Detection

Kyong-Wook Kim*, Hyo Jong Lee

*Division of Electronics and Information Engineering,

Chon-buk National University

Research Institute of Industrial Technology,

Chon-buk National University

요 약

동영상의 장면 전환의 검출은 특정 객체의 검출, 비디오 압축 또는 비디오 문서의 군집화, 비디오 데이터베이스 시스템 등 많은 응용프로그램에서 유용하게 다루어진다. 특히 멀티미디어 데이터베이스에서 이미지를 검출하는 처음 단계로서 Shot Change 검출은 아주 중요하다. 이미 장면 전환의 검출을 위한 여러 알고리즘이 개발되어 발표되었다. 본 논문에서는 대용량의 영상 데이터 사이클을 고려하여 검출에 소요되는 시간과 검출의 정확도의 상쇄관계를 알아보기 위해서 히스토그램의 분포에 의한 알고리즘과 이미지의 평균과 분산을 이용한 알고리즘을 구현하고 그 알고리즘 간의 성능의 차이를 비교하였다.

1. 서론

멀티미디어가 발달함에 따라서 멀티미디어를 취급하는 응용프로그램의 발달 또한 활발해졌고 앞으로 계속 요구되고 있다. 특히 멀티미디어 데이터를 취급하는 멀티미디어 데이터베이스 시스템이 그 중 하나이다. 멀티미디어 데이터베이스 시스템의 디자인 과정 중 맨 처음 단계라 할 수 있는 장면 전환의 검출은 비디오 데이터베이스 시스템에서 뿐만 아니라 비디오 검출, 비디오 압축, 비디오 문서의 군집화 등 여러 분야에서 유용하게 다루어지고 있다. 이따 여러 알고리즘들이 개발되어 있다.

비디오의 장면전환은 대상 자료의 크기가 방대하여 정확한 장면 전환 검출을 위해서는 영상의 다양한 특성들을 모두 비교하여야 한다. 즉, 정확한 검출을 위해서는 오랜 계산량이 필요하고, 이의 감소를 위해서 단순한 알고리즘을 구현할 경우에는 장면 전환 검출률이 낮아진다. 검출시간과 검출의 정확도에

는 상쇄관계가 존재하는 것이다. 간단한 계산량을 통하여서도 가급적 검출의 높은 정확도를 이루는 방법을 알아보기 위해서, 본 논문에서는 장면전환(SC:Shot Change) 검출에 관한 두 개의 알고리즘을 구현하였다. 히스토그램을 이용한 알고리즘과 영상을 구성하는 화소들의 평균과 분산을 이용한 알고리즘을 구현하고 그 결과를 비교하였다.

2. 히스토그램을 이용한 알고리즘

2.1 칼라 히스토그램

각각의 프레임에서 R, G, B 각각에 대하여 히스토그램을[1] 구한 다음, 이웃하는 두 프레임 사이에서 화소 값이 k인 경우의 R, G, B 각각의 히스토그램의 차를 구하고, 모든 경우의 k값(0-255)에 대하여 그 차이들의 합을 구한 값 D를 SC 검출의 기준으로 삼는다. 이 방법은 지역적인 정보를 SC 검출에 반영할 수 없다.

$$D_{color}(F_i, F_j) = \sum_{k=0}^{255} (|H_i^r(k) - H_j^r(k)| + |H_i^g(k) - H_j^g(k)| + |H_i^b(k) - H_j^b(k)|) \quad (식1)$$

2.2카이스퀘어 히스토그램

카이스퀘어 히스토그램 방법[1]은 R, G, B 각각의 히스토그램을 이용한다는 면에서는 2.1과 같지만 검출의 기준이 되는 공식에 차이가 있다. (식2)에 보이는 바처럼 (식1)의 \sum 오른쪽 부분을 제곱을 하고 그 값을 이웃하는 두 프레임에서의 R, G, B 각각의 화소 값 k일 때의 히스토그램 값들 중 큰 값들의 합으로 나눈 결과를 모든 경우의 k값(0-255)에 대하여 합을 구한다.

$$D_x(F_i, F_j) = \sum_{k=0}^{255} \frac{(|H_i^r(k) - H_j^r(k)| + \max(H_i^r(k), H_j^r(k)) + |H_i^g(k) - H_j^g(k)| + \max(H_i^g(k), H_j^g(k)) + |H_i^b(k) - H_j^b(k)| + \max(H_i^b(k), H_j^b(k)))^2}{\max(H_i^r(k), H_j^r(k)) + \max(H_i^g(k), H_j^g(k)) + \max(H_i^b(k), H_j^b(k))} \quad (식2)$$

(식1)과의 차이는, (식1)이 이웃하는 두 프레임 사이에서 히스토그램의 절대적인 차이를 의미한다면, (식2)는 상대적인 차이를 의미한다고 할 수 있다. 예를 들어 아래 그림 1-a에서 총 1600 픽셀중 왼쪽 길쭉한 사각형이 400 픽셀, 오른쪽 위의 검정색 사각형이 100 픽셀을 차지하고 있다고 할때, 두 이미지 사이의 히스토그램 차이값은 200이다. (식2)로 두 이미지의 차이를 구하면 28.3이다.

다시 말해 (식2)는 지역적인 정보를 어느 정도 반영한다고 할 수 있다.

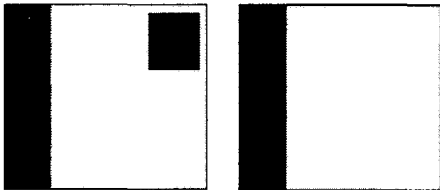


그림 1-a 그림 1-b
지역정보를 반영하는 예

그리고 (식2)의 분자에 있는 제곱에 의해 SC가 일어난 경우와 그렇지 않은 경우의 차이가 크게 벌어진다. 즉 SC 검출을 위한 문턱치를 결정하는데 있어서 (식1)에 비해 훨씬 여유롭다고 할 수 있다. 또 분모가 두 프레임의 히스토그램 값 중 큰 값들의 합으로 이루어져 있기 때문에 전체적으로 D 값이 작아진다.

2.3 두 가지 방법의 혼용

위 두 가지 방법에 대한 실험 중에 2.1방법에서만 독립적으로 발생하는 오류(false)와 2.2방법에서

만 독립적으로 발생하는 간과(miss)를 인지하였다. 이렇게 한 가지 방법에서만 독립적으로 발생하는 오류나 간과를 없애기 위해, 또 (식2)가 어느 정도 지역적인 정보를 반영한다는 점을 감안하여 문턱치를 결정하는데 있어서 (식1)과 (식2)를 동시에 사용하였다.

```

if(D_chi(i, j) >= Th1) True;
elseif(D_chi(i, j) >= Th2 and
      D_color(i, j) >= Th3) True;
else False;
    
```

먼저 (식2)로 높은 문턱치(Th1)를 설정하여, 만족하는 경우 SC로 인정하고, 다음에 간과하였던 부분들을 보정하기 위해 낮은 문턱치(Th2)를 설정하고 그 문턱치와 (식1)에 사용되는 문턱치(Th3)를 동시에 만족하는 경우 SC로 인정하였다.

3. 평균과 분산(표준편차)을 이용한 알고리즘

3.1 평균의 차이를 이용한 방법

이미지의 평균 역시 연속적인 프레임 이미지에서의 갑작스러운 변화를 인지하는데 비중 있는 요소가 될 수 있다. (식3)에서처럼 R, G, B 각각의 평균을 구한 뒤 각각의 차이를 합한 결과를 SC 검출의 기준으로 삼았다.

$$D_{mean}(F_i, F_j) = |\mu_i^R - \mu_j^R| + |\mu_i^G - \mu_j^G| + |\mu_i^B - \mu_j^B| \quad (식3)$$

$$D_{std.dev}(F_i, F_j) = |\sigma_i^R - \sigma_j^R| + |\sigma_i^G - \sigma_j^G| + |\sigma_i^B - \sigma_j^B| \quad (식4)$$

(식4)는 두 이웃하는 프레임 이미지 사이의 표준편차의 차이를 나타낸 식이다. 실험 결과를 개선하기 위하여 문턱치 설정에 (식4)를 도입하였다.

```

if(D_mean(i, j) >= Th1) True;
elseif(D_mean(i, j) >= Th2 and D_chi(i, j) >=
      (D_mean(i, j) + D_std.dev(i, j)) * C) True;
else False;
    
```

아래 그림을 보면 판이하게 다른 그림이지만 두 이미지의 평균값($\mu_R + \mu_G + \mu_B$)은 495로 같다.

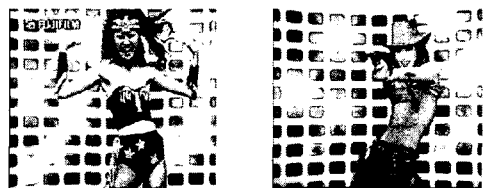


그림 2-a 그림 2-b
평균은 같지만 판이하게 다른 두 이미지

이 방법 또한 2.1에서처럼 지역정보를 반영하지 못한다. 그래서 3.2절에서는 이미지를 몇 개의 블록으로 나누어서 실험하였다.

3.2 블록별 평균의 차이를 이용한 방법

지역적인 정보를 SC 검출에 반영하기 위해서 이미지를 4 x 4 총 16 블록으로 나누어서 각각의 블록에 대하여 3.1의 방법을 적용하였다.

먼저 각 블록의 평균을 구하고, 이웃하는 두 프레임 이미지 사이에서 같은 위치에 대응하는 두 블록간의 평균의 차이를 구한다. 그런 다음 각각의 블록에 대하여 특정한 문턱치를 설정한다. 그리고 그 문턱치를 넘어서는 블록의 수를 계산하여 전체 이미지의 SC검출의 기준으로 삼았다.

$$D_{mean}^{l,m}(F_i, F_j) = \left| \mu_i^{R(l,m)} - \mu_j^{R(l,m)} \right| + \left| \mu_i^{G(l,m)} - \mu_j^{G(l,m)} \right| + \left| \mu_i^{B(l,m)} - \mu_j^{B(l,m)} \right| \quad (식5)$$

(식5)은 (l, m) 블록에서의 두 이웃한 프레임 사이의 평균의 차를 나타낸 식이다.

각 블록의 문턱치를 설정하는데 있어서 전체 이미지의 표준편차를 도입하였다. 이미지의 표준편차가 크다는 말은 화소 값의 분포가 평균값에서 멀게 분포되어 있다는 말이다. 즉 이미지에 다양한 색이 분포되어 있다는 말이고, 이것은 블록별 평균의 차이 값 (식5)에 비례하여 영향을 미친다. 다시 말해서 이미지의 표준편차가 크면 작을 때 보다, SC가 아님에도 불구하고 블록별 평균의 차이 값이 커지기가 쉽다. 따라서 블록별 문턱치를 전체이미지의 표준편차에 비례하도록 설정하였다.

```
for (l = 0, count = 0; l < 4; l++)
    for (m = 0; m < 4; m++)
        if (Dmean,blockl,m ≥ 1/3 × C1)
            count++;
if (count ≥ C2) True;
else False;
```

3.3 유사도(likelihood ratio)를 이용한 방법

유사도[2]는 비디오를 구성하는 영상들의 화소값의 밝기 정보와 화소들의 분산을 이용하여 계산될 수 있다. 즉, 밝기의 변화나 분산의 특성이 다르면 장면전환이 발생했다고 볼 수 있는 것이다.

$$L_{k-1,k} = \frac{\left[\frac{S_k + S_{k-1}}{2} + \left(\frac{m_k - m_{k-1}}{2} \right)^2 \right]^2}{S_k \times S_{k-1}} \quad (식6)$$

L값은 S_k = S_{k-1}, m_k = m_{k-1}일때 최소값 1을 갖는다. L값이 작으면 작을 수록 두 이미지의 유사도는 높아지며 더욱 비슷하다고 할 수 있다.

본 논문에서도 S_k와 m_k 값을 k번째 프레임 이미지 전체의 분산과 평균값으로 설정하고 실험하였다. 다시 말해, L값이 L값 자체만으로 연속적인 프레임 이미지 사이에 있는 갑작스러운 변화를 검출하는 지표로 사용될 수 있는지를 실험하였다.

실험은 R, G, B 각각에 대해 L값을 구한 다음 그 합을 SC의 기준으로 삼았다. 그리고 문턱치 설정에 있어서 지역정보를 반영해 실험 결과를 개선하기 위해 아래와 같이 (식4)를 도입하였다.

$$D_L(F_i, F_j) = L_{i,j}^R + L_{i,j}^G + L_{i,j}^B \quad (식7)$$

```
if (DL(i, j) ≥ Th1) True;
elseif (DL(i, j) ≥ Th2 and Dchi(i, j) ≥ (Dmean(i, j) + Dstd.dev(i, j)) × C) True;
else False;
```

4. 실험 및 결과

총 10개의 데이터 파일을 가지고 실험하였다. 데이터들은 MPEG 포맷으로 구성된 영상들이며 이미지 사이즈는 320x240 2개, 352x240 2개, 352x288 1개, 480x480 4개, 720x480 1개이다. 총 5128 프레임으로 데이터 당 평균 512.8 프레임, 총 162초 분량에 해당되며 데이터 당 평균 16.2초, 정확한 장면전환은 총 75회에 걸쳐 발생한다.

4.1 히스토그램을 이용한 방법

칼라 히스토그램 알고리즘의 실험 과정에서 데이터들의 이미지 사이즈가 달라짐에 따라서 D_{color}의 값이 크게 변하는 것을 인지하였다. 그래서 320x240의 사이즈를 기본 사이즈로 하였고, 이보다 큰 이미지의 경우에는 이미지의 비율 계산하여 320x240의 Th 값에 그 비율 곱한 값으로 Th 값을 설정하였다.

	D _{color}	D _{chi}	D _{color} + D _{chi}
Th	68000	60000	Th ₁ =75000 Th ₂ =34000 Th ₃ =67800
SCD	88	79	88
Hit	69(78.4%)	70(88.6%)	70(79.5%)
간과	6(8%)	5(6.7%)	5(6.7%)
오류	19(21.6%)	19(11.4%)	18(20.5%)

표 1 : 히스토그램 알고리즘의 결과

표 1에 실험시에 사용했던 문턱치와 실험 결과를 나타내었다. 실제 실험에서는 세 번째 방법에서 어느 정도 효과를 보았지만, 다른 실험에서는 사용하지 않은 데이터들을 제외하는 바람에 결과가 다른

방법과 별다른 차이가 없는 것처럼 보인다.

4.2 평균과 분산을 이용한 방법

평균과 유사도는 그 자체만으로는 SC를 검출하는 기준으로서 적합하지 못했다. 특히 지역정보를 반영하지 못해서 많은 오류와 간과를 유발했다. 이런 결과를 개선하기 위해 어느 정도 지역정보를 반영하는 카이스퀘어 알고리즘을 같이 사용하였다.

표2에 평균과 분산을 이용한 방법을 실험할 때 사용한 문턱치 및 상수와 실험 결과를 나타내었다.

	D_{mcan}	$D_{mcan,block}$	D_L
Th	Th ₁ = 50 Th ₂ = 10 C = 1000	C ₁ = 1/5 C ₂ = 10	Th ₁ = 3.4 Th ₂ = 3.05 C = 1000
SCD	81	76	74
Hit	66(81.5%)	74(97.4%)	65(87.8%)
간과	9(12%)	1(1.3%)	10(13.3%)
오류	15(18.5%)	2(2.6%)	9(12.2%)

표 2 : 평균과 분산을 이용한 알고리즘의 결과

평균을 사용한 방법과 유사도를 사용한 실험 결과는 그 자체만을 가지고 판단했을 때 보다는 훨씬 양호해졌지만 블록을 사용한 방법에 비하면 그다지 좋은 알고리즘이라 할 수 없었다. 그러나 SC 검출을 위한 기준으로서의 가치는 충분하다고 여겨지고, 지역정보를 반영하는 어떤 다른 알고리즘과 같이 사용된다면 좋은 결과를 낼 수도 있을것이다.

표에서 보는 바와 같이 블록을 사용한 방법이 가장 좋은 결과를 나타내었다. 이미지를 블록으로 나누어서 각각 SC의 여부를 판단함으로써 지역정보를 반영할 수 있었다. 이미지 내부의 특정한 위치에서의 급격한 변화로 인한 오류는 거의 없앨 수 있었다.

5. 결론

장면 전환 검출을 위한 여러 알고리즘을 구현하고 그 결과를 비교하여 보았다.

히스토그램을 이용한 방법은 평균과 분산을 이용한 방법보다(블록별 평균을 이용한 방법을 제외하고) 오류는 많았지만 간과는 더 적었다. 히스토그램을 이용한 방법들 중에서는 문턱치 설정에 있어서 두가지 방법을 혼용한 방법에서 제일 좋은 결과를 얻었다. 이 결과는 지역적인 정보가 장면 전환 검출에 중요함을 나타낸다.

평균과 분산을 이용한 방법에서, 오직 두 이미지 사이의 평균의 차이를 이용한 방법과, 두 이미지 사

이의 유사도만을 이용한 방법은, 그 자체만으로는 장면 전환 검출을 위한 기준으로서 부적합하였다. 다른 지역정보를 반영할 수 있는 요소와 함께 사용하면 어느정도 원하는 결과를 얻을 수 있을 것이다.

블록별 평균을 사용한 방법이 제일 좋은 결과를 나타내었다. 지역정보를 반영할 수 있었기에 좋은 결과를 얻을수 있었다. 다른 방법들 보다 시간은 오래 소요되었지만 그만큼 좋은 결과를 얻을 수 있었다. 히스토그램에 비해 계산량을 대폭 줄이고서도, 히스토그램 방법의 결과에 맞먹는 결과를 얻을 수 있을 거라 생각하고 평균을 사용하여 구현했는데, 결국 평균 한가지 만으로는 적합한 결과를 얻을 수 없었고, 지역 정보를 반영하는 카이스퀘어 알고리즘을 같이 사용한 후에 비로소, 히스토그램과 견줄만한 결과를 얻을수 있었다. 그 과정에서 시간 비용이 증가하여, 큰 의미를 부여할 만한 알고리즘이 아니라고 판단하게 되었다.

장면 전환을 검출하는데 있어서 지역정보를 얼마얼마나 반영할 수 있는 지가 알고리즘 성능을 크게 좌우 하였다.

장면 전환 검출 뿐만 아니라 이미지를 검출하는 과정에서도 이미지의 지역정보를 잃지 않는 것이 중요한 문제가 될 것이다.

참고문헌

[1] Haitao Jiang, Abdelsalam Helal, Ahmed K. Elmagarmid, Anupam Joshi "Scene change detection techniques for video database systems", Multimedia Systems (1998) 6 : p186-195
 [2] Rakesh Dugad, Krishna Ratakonda, Narendra Ahuja "Robust video shot change detection", 1998 IEEE second workshop on Multimedia Signal Processing , CA 7-9 Dec. 1998 : p376-381