

## 적응적 휘도 성분 부호화와 블록의 움직임을 고려한 프레임을 증강 기법에서의 장면전환 기법

\*배창영, \*\*김정환, \*\*\*김일승 \*\*\*\*정제창

한양대학교 전자컴퓨터통신공학과

\*baepang@hanyang.ac.kr, \*\*jhkim07@hanyang.ac.kr, \*\*\*ghanjang@gmail.com, \*\*\*\*jjeong@hanyang.ac.kr

## Scene Change Detection for Frame Rate Up Conversion Considering Adaptive Luminance Coding and Block Motion Activity

\*Changyoung Bae \*\*Jeonghwan Kim \*\*\*Ilseung Kim \*\*\*\*Jechang Jeong

Department of Electronics and Computer Engineering, Hanyang University

### 요약

본 논문은 적응적 휘도 성분 부호화(Adaptive Luminance Coding) 알고리즘을 이용한 블록의 움직임을 고려한 장면 전환 감지 성능을 개선시키는 알고리즘을 제안하고자 한다. 기존의 장면전환 기법에서는 단순 물체의 움직임도 장면전환으로 인식하는 문제가 발생하는 점을 착안하여 본 논문에서는 적응적 휘도 성분 부호화와 블록 매칭 오류를 활용하여 보다 정확한 장면전환 기법을 제안하고자 한다. 실험결과를 통해 제안하는 방법이 기존 방법보다 높은 *Precision*과 *Recall*을 보이는 것을 확인할 수 있다.

### 1. 서론

최근 UHDTV에 많이 사용되는 LCD는 CRT(Cathode Ray Tube), PDP(Plasma Display Panel)와 같이 프레임 당 짧은 시간 화면을 유지하는 방식과 다르게 프레임 당 오랫동안 화면을 유지하는 Hold-type display 특성이 있다[1]. 이 특성으로 영상 내에서 현재 프레임에서 이전 프레임의 잔상이 나타나는 동작 흐림 현상(motion blur effect)같은 심각한 문제가 발생하고 이는 화질 저하의 원인이 된다.

이 현상의 해결책 중 하나로 프레임을 증강 기법(Frame Rate Up Conversion: FRUC)이 제안되었다. 초기 프레임을 증강 기법은 기존의 프레임을 복사하여 보간 프레임을 생성하는 방법 또는 두 개의 프레임의 평균값을 취하여 프레임 사이에 보간 프레임을 생성 하는 방법을 제안했다. 이 방법은 소프트웨어나 하드웨어로 설계가 간단하다는 장점이 있지만 영상의 움직임을 고려하지 않아 보간 프레임에 잔상효과가 발생한다[2].

이러한 단점을 보완하기 위해 움직임 보상 프레임을 증강 기법(Motion Compensation Frame Rate Up Conversion: MC-FRUC)이 제안되었다 [3-5]. 움직임 보상 프레임을 증강 기법은 두 가지 기술의 조합으로 이루어진다. 하나는 우선 움직임 예측(Motion Estimation: ME)기술로 연속된 프레임에서 물체의 움직임 벡터(Motion Vector: MV)를 찾는 과정이고, 또 다른 하나는 보간 프레임 움직임 보상(Motion Compensated Interpolation: MCI)기술로 움직임 예측에서 찾은 움직임 벡터를 활용하여 보간 프레임을 생성하는 과정이다. 이 두 가지 과정을 통해 움직이는 물체가 포함된 프레임 사이에서

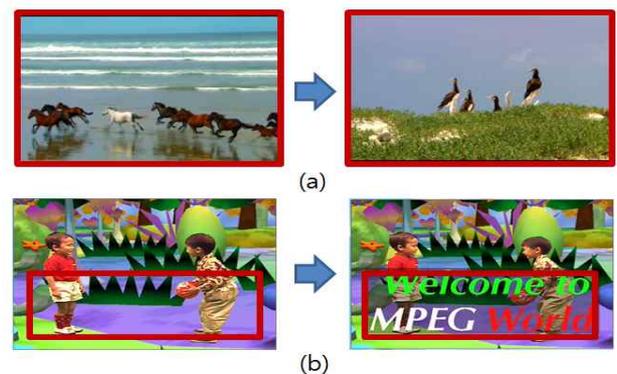


그림 1 장면전환의 종류 (a) 전역 장면전환 (b) 지역 장면전환

도 보간 프레임을 정확하게 생성할 수 있다. 하지만 그림 1에서 보이는 장면전환과 같은 문제는 움직임 예측으로 해결할 수 없어 보상 프레임의 많은 화질 열화를 야기 시킨다.

이를 해결하기 위해 장면전환 감지 알고리즘이 추가적으로 제안되었다[2, 6]. [2]에서는 멀티 히스토그램을 이용하며 장면전환을 감지하는 알고리즘을 제안하였다. 하지만 이 방법은 히스토그램 생성 과정과 Otsu 방법[7]을 사용해 최적의 문턱치를 찾는 과정에서 많은 연산량이 요구된다. [6]에서는 Boolean 연산을 이용하여 복잡도를 대폭 줄이는 장면전환 감지 기법을 제안하였다. 이러한 알고리즘들은 빠르고 복잡한 움직임이 있는 경우 움직임 예측을 할 수 있음에도 장면전환 영역으로 구분하는 오류를 범하는 경우가 있다.

본 논문에서는 블록의 움직임 고려한 장면전환 감지 알고리즘을 제안하고자 한다. 적응적 휘도 성분 부호화를 이용하여 장면전환 후보군을 만들고, 복원된 참조 프레임의 정보를 활용하여, 움직임 예측이 가능한 부분에서의 장면전환 선택을 방지함으로써 Precision과 Recall을 대폭 향상시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 알고리즘인 적응적 휘도 성분 부호화 알고리즘을 분석한다. 또한 제안한 알고리즘과 그에 관한 실험 결과를 각각 3, 4장에서 설명하고, 5장에서 결론으로 본 논문의 끝을 맺는다.

## 2. 기존 알고리즘 (적응적 휘도 부호화 알고리즘)

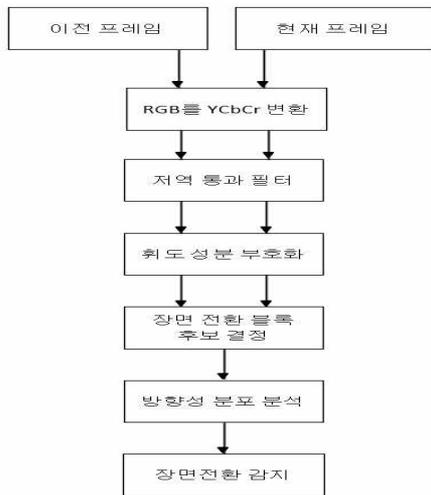


그림 2 적응적 휘도 부호화 알고리즘 흐름도

적응적 휘도 성분 부호화 알고리즘은 휘도성분 부호화를 통해 블록을 이전 부호화 된 픽셀들과 픽셀들의 평균값의 차 두 가지를 이용하여 장면전환을 감지하는 알고리즘이다. 우선 이전 프레임과 현재 프레임의 휘도 성분만 뽑아 저역 통과필터를 적용하여 주위 픽셀 간의 갑작스런 변화를 제거한다.

다음은 휘도 성분 부호화로 블록들의 특징을 찾아 이전 블록과 현재 블록의 유사성을 판단하기 위해 이전 부호로 변환시키는 과정이다. 식 (1)을 사용하여 블록 내의 모든 픽셀의 평균값을 구하고 이 평균값을 기준으로 블록 내 픽셀을 0 또는 1로 지정해준다.

$$\begin{aligned}
 p_{i,j}^A &= \frac{1}{NM} \sum_{x=1}^N \sum_{y=1}^M p_{i,j}(x,y) \\
 p_{i,j}^{bit}(x,y) &= \begin{cases} 1, & \text{if } p_{i,j}^A < p_{i,j}(x,y) \\ 0, & \text{otherwise} \end{cases} \quad (1)
 \end{aligned}$$

여기서  $p_{i,j}^A$ 은 (i, j)번 블록의 휘도 성분 픽셀의 평균값,  $x$ 와  $y$ 는 블록 내 픽셀의 위치,  $N$ 과  $M$ 은 블록 내 수직 수평 픽셀 수,  $p_{i,j}^{bit}$ 는 평균값을 기준으로 0 또는 1을 할당하는 이전부호이다.

다음은 장면전환 블록 후보 결정과정으로 식 (1)에서 구한 이전 평균값과 블록을 활용하여 이전 블록과 현재 블록의 유사성을 찾는다. 식 (2)을 통해 구한 이전 블록과 현재 블록의 평균값과 Xor로 구한 이전 블록의 왜곡 정도를 특정 임계값과 비교하여 장면전환 후보 블록을 식

(3)과 같이 구한다.

$$\begin{aligned}
 D_{i,j}^A &= |p_{i,j}^{A,prev} - p_{i,j}^{A,curr}| \\
 D_{i,j}^S &= \frac{1}{NM} \sum_{x=1}^N \sum_{y=1}^M p_{i,j}^{bit,prev}(x,y) \otimes p_{i,j}^{bit,curr}(x,y) \quad (2) \\
 CB_{i,j} &= \begin{cases} 1, & \text{if } D_{i,j}^A > T_1 \text{ and } D_{i,j}^S > T_2 \\ 0, & \text{otherwise} \end{cases} \quad (3)
 \end{aligned}$$

여기서  $D_{i,j}^A$ 는 이전 블록과 현재 블록의 평균의 차의 절대 값,  $D_{i,j}^S$ 는 이전  $p_{i,j}^{bit}$ 와 현재  $p_{i,j}^{bit}$ 의 Xor연산 평균,  $CB_{i,j}$ 는 장면전환 후보 여부로 '1'일 때 장면전환 후보로 판별하고 '0'일 때 장면전환 아님으로 결정한다.

마지막으로 방향성 분포 분석으로 그림 3과 같이 주위 블록들의 장면전환 비율을 따져 장면전환 후보 블록을 수정하는 과정이다. 식 (4)에서 찾은 장면전환 후보 블록의 주위의 좌 상단, 좌 하단, 우 상단, 우 하단 블록들의 장면전환 유무 확인으로 확실하게 장면전환을 확정한다.

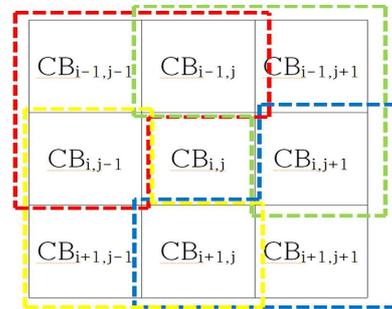


그림 3 방향성 분포 분석 블록 예

$$\begin{aligned}
 P_{i,j}^{TL} &= \frac{1}{3}(CB_{i-1,j-1} + CB_{i-1,j} + CB_{i,j-1}) \\
 P_{i,j}^{TR} &= \frac{1}{3}(CB_{i-1,j+1} + CB_{i-1,j} + CB_{i,j+1}) \\
 P_{i,j}^{BL} &= \frac{1}{3}(CB_{i+1,j-1} + CB_{i+1,j} + CB_{i,j-1}) \\
 P_{i,j}^{BR} &= \frac{1}{3}(CB_{i+1,j+1} + CB_{i+1,j} + CB_{i,j+1}) \\
 FB_{i,j} &= \begin{cases} 1, & \text{if } (P_{i,j}^{TL} \text{ or } P_{i,j}^{TR} \text{ or } P_{i,j}^{BL} \text{ or } P_{i,j}^{BR}) > T_3 \\ 0, & \text{otherwise} \end{cases} \quad (4)
 \end{aligned}$$

여기서  $P_{i,j}$ 는 블록을 중심으로 좌 상단, 좌 하단, 우 상단, 우 하단의 장면전환 유무에 대한 비율,  $FB_{i,j}$ 는 장면전환 감지의 최종 결과 값으로 '1'일 때 장면전환으로 '0'일 때 장면전환 아님으로 판별한다.

## 3. 제안하는 알고리즘

기존의 알고리즘에서 움직임에 의한 장면전환 감지의 문제의 해결책으로 본 논문에서는 적응적 휘도 성분 부호화와 블록의 움직임을 고려한 장면전환 기법 알고리즘이 제안하고자 한다. 제안된 알고리즘의 흐름도는 그림 4과 같다.

우선 기존의 적응적 휘도 부호화 알고리즘을 활용하여 장면전환 블록을 찾는다.

이렇게 구한 장면전환 후보 블록 중 움직임 보상기술 충분히 보강

프레임 생성이 가능한 후보 블록은 장면전환 후보에서 제외할 필요가 있다. 움직임에 의한 장면전환 블록을 탐색하는 방법은 이전 프레임 탐색범위 안에서 장면전환 후보 블록과 제일 유사한 블록을 식 (5)을 사용해 최소의 SAD 값을 갖는 유사 블록을 찾는다[8].

$$SAD(dx, dy) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |f_{cur}(x, y) - f_{ref}(x + dx, y + dy)| \quad (5)$$

여기서  $x, y$ 는 블록의 위치,  $dx$ 와  $dy$ 는 이전 프레임의 탐색 범위 안에서 움직임을 나타낸다. 이제 찾은 블록과 장면전환 블록의 상관성을 찾기 위해 이전 프레임의 최소의 SAD 블록과 현재 프레임의 장면전환으로 판별된 후보 블록을 다시 휘도 성분 부호화를 진행한다. 이 과정을 수행한 결과 두 블록이 유사하다면 움직임에 의한 장면전환의 경우로 판별되어 장면전환 아님으로 판명난다.

마지막으로 방향성 분포 분석을 사용하여 공간적으로 주위 블록들의 장면전환 비율을 분석하여 잘못 찾은 장면전환을 보정해준다. 또한 오차를 최소한으로 줄이기 위해 중간 값 필터(Median Filter)를 사용하여 소금과 후추 잡음(Salt and Pepper Noise)라고 불리는 임펄스 잡음을 효과적으로 제거해준다.

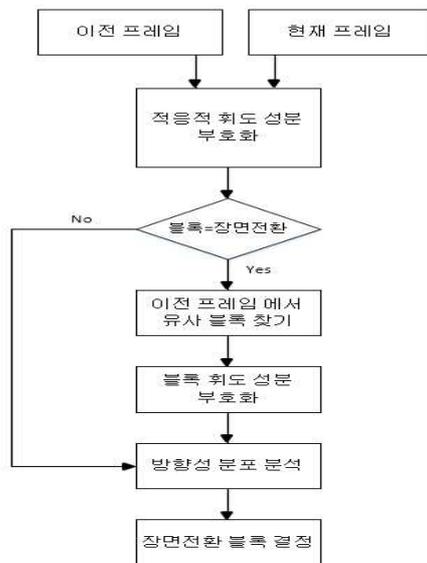


그림 4 제안된 알고리즘 흐름도

#### 4. 실험결과

제안된 알고리즘의 성능 비교를 위해 그림 5와 같이 장면전환이 많이 나타나는 Foreman, Wild\_life, Big\_bang 그리고 Family\_guy의 CIF(352x288) 4개의 영상을 이용하여 실험했고, 표 1과 같이 실험 환경을 설정하였다.

장면전환의 객관적 평가를 위해 식 (6)의 Precision, Recall 과 F1 score 을 사용했고, 이 값이 1에 가까울수록 좋은 성능을 나타낸

표 1 실험 환경 및 설정

프로세서	Intel(R) Core i5-3570 CPU @ 3.40GHz
메모리(RAM)	8.00GB
소프트웨어	Visual Studio 2013
기본 블록 크기	8x8 픽셀
탐색 범위	± 16 픽셀
[2] 임계값	d=3, λ=0.5
[6] 임계값	T <sub>1</sub> =5, T <sub>2</sub> =0.5

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}}$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

다[9].

여기서  $T$ 는 실제 장면전환 부분,  $F$ 는 실제 장면전환 아닌 부분,  $P$ 는 장면전환 부분으로 판정,  $N$ 는 장면전환 아님으로 판정을 나타낸다.

표 2, 3을 분석하면 제안된 알고리즘은 [6]알고리즘에 비해 Precision, Recall, F1 score가 평균적으로 각각 0.24, 0.01, 0.16 향상된 것을 확인할 수 있다. 이를 통해 제안하는 알고리즘이 기존의 장면전환 알고리즘보다 좋은 성능을 보이는 것을 확인할 수 있다.



(a) (b) (c) (d)

그림 5 실험 영상과 제안된 알고리즘의 장면전환 탐색 결과 (a) Foreman, (b) Wild\_life, (c) Big\_bang, (d) Family\_guy

#### 5. 결론

본 논문에서는 적응적 휘도 성분 부호화와 블록의 움직임을 고려한 프레임움 증강 기법에서의 장면전환 기법 알고리즘을 제안하였다.

표 2 Precision, Recall, F1 score, 처리시간 비교

Sequence	Method	장면전환 프레임 수	장면전환 감지 평가 요소						
			$N_{TP}$	$N_{FP}$	$N_{FN}$	Precision	Recall	F1 score	프레임 당 처리시간 (frame/ms)
Foreman (300)	Method 1	138	107	159	31	0.40	0.77	0.53	115.66
	Method 2		126	24	12	0.84	0.91	0.87	20.47
	Proposed		127	5	10	0.96	0.92	0.94	54.97
Wild_life (300)	Method 1	123	101	162	22	0.38	0.82	0.52	113.05
	Method 2		110	62	13	0.64	0.89	0.74	20.29
	Proposed		118	11	5	0.92	0.96	0.93	33.89
Big_bang (300)	Method 1	36	33	185	3	0.15	0.92	0.26	115.24
	Method 2		33	23	3	0.59	0.92	0.71	20.43
	Proposed		31	1	5	0.97	0.86	0.91	26.72
Family_guy (300)	Method 1	87	79	89	8	0.47	0.91	0.62	114.86
	Method 2		82	24	5	0.77	0.94	0.85	20.36
	Proposed		84	4	3	0.96	0.97	0.96	30.46

표 3 평균 Precision, 평균 Recall, 평균 F1 score, 평균 처리시간

Method	Precision	Recall	F1 score	프레임 당 처리시간 (frame/ms)
[2]	0.35	0.85	0.48	114.70
[6]	0.71	0.92	0.78	20.39
Proposed	0.95	0.93	0.94	36.51

움직임 보상이 가능한 영역에서의 장면전환을 찾아 수정함으로써 정확도를 높여 주었고, 이를 통해 정확한 장면전환 감지가 되어 프레임율 증강기법 후의 보간 프레임의 화질 저하를 줄였다.

제안된 알고리즘은 이전의 알고리즘에 비해 Precision, Recall, F1 score 값이 평균 0.24, 0.01, 0.16 향상되었고 특히 Precision 값이 높게 나와 정확한 장면전환 감지가 된다는 것을 알 수 있었고 이를 통해 성능이 우수함을 증명하였다.

### 감사의 글

“본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학 ICT 연구센터 육성지원사업의 연구결과로 수행되었음” (IITP-2015 - H8501-15-1005)

### 참고문헌

[1] T. Kurita, “Moving picture quality improvement for hold-type AM-LCDs,” in *SID Int. Symp. Dig. Tech. Papers*, pp.986-989, Jun. 2001

[2] S.-J. Kang, S. I. Cho, S. Yoo, and Y. H. Kim, “Scene change detection using multiple histograms for motion-compensated frame rate up conversion,” *Journal of Display Tech.*, vol.8, no.3, pp.121-126, Mar. 2012.

[3] G. Dane and T. Nguyen, “Motion vector processing for frame rate up conversion,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 309-312, May. 2004.

[4] S.-J. Kang, S. Yoo, and Y. H. Kim, “Dual motion estimation for frame rate up-conversion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1909 - 1914, Dec. 2010.

[5] S. -J. Kang, K. R. Cho, and Y. H. Kim, “Motion compensated frame rate up-conversion using extended bilateral motion estimation,” *IEEE Trans. Consumer Electron.*, vol. 53, no. 4, pp. 1759 - 1767, Nov. 2007.

[6] S.-J. Kang, “Adaptive luminance coding-based scene-change detection for frame rate up-conversion,” *IEEE Transactions on Consumer Electronics*, vol.59, No. 2, pp. 370-375, May. 2013.

[7] Jianzhuang, Liu, Li Wenqing, and Tian Yupeng, “Automatic thresholding of gray-level pictures using two-dimension Otsu method,” *Conference Proceedings, China.*, 1991 International Conference on. IEEE, pp. 325-327, Jun. 1991.

[8] Shilpa Metkar, Sanjay Talbar, “Motion Estimation Techniques for Digital Video Coding,” *Springer Briefs in Applied Sciences and Technology Computational Intelligence*, pp. 17-19, 2013.

[9] Y. Yang and X. Liu, “A re-examination of text categorization methods,” *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, pp. 42-49, Aug. 1999.