

코너 정보와 Adaboost 인식 기술을 이용한 비디오 내의 블록 오류 고속 검출 방법

*하명환 *이문식 *박성춘 **안기욱 **김민기

*한국방송공사, **미디어코러스

*mhha@kbs.co.kr

Fast block error detection method in video using a corner information and Adaboost recognition technology

*Ha, Myunghwan *Lee, Moonsik *Park, Sungchoon

**Ahn, Kiok **Kim, Min-Gi

*Korean Broadcasting System, **Mediachorus Inc.

요약

방송 콘텐츠 제작에는 카메라, VCR, NLE, 인코더 등의 장비가 사용되고 있으며, VCR 헤더 불량, 테이프 노후화/보관불량, NLE 편집 오류, 인코더 장비 불량 등의 다양한 이유로 콘텐츠에 예기치 않은 비디오 및 오디오 오류가 발생할 수 있다. 이러한 문제점을 해결하기 위하여 콘텐츠에 포함된 다양한 비디오 및 오디오 오류를 자동으로 검사할 수 있는 자동 검사 시스템이 요구된다.

본 논문에서는 이러한 다양한 오류를 자동으로 검사할 수 있는 방법 중 특히 비디오 내에 종종 포함되는 블록 오류를 대상으로 하는 고속 오류 검출 방법을 설명한다. 제안한 방법은 비디오 내의 매 프레임의 코너 수를 계산하고, 시간 증가에 따른 코너 수의 변화량을 검사하여 블록 오류가 포함될 것으로 예상되는 후보 프레임을 찾는 1단계 과정과, 후보 프레임을 대상으로 Adaboost 인식 기술을 사용하여 학습한 분류기를 통해 최종 블록 오류가 포함된 프레임을 검출하는 2단계 과정으로 구성된다.

시스템 구현 실험 결과, 비디오 내에 포함된 블록 오류를 프레임 단위로 정확하게 고속 검출 하는 것이 가능함을 확인하였다. SD급의 경우 실시간 대비 2.3배속 가량의 고속 검사가 가능하고 HD의 경우에도 0.8배속 수준의 고속 검사가 가능하였다.

1. 서론

방송 콘텐츠 제작에는 카메라, VCR, NLE, 인코더 등의 장비가 사용되고 있으며, VCR 헤더 불량, 테이프 노후화/보관불량, NLE 편집 오류, 인코더 장비 불량 등의 다양한 이유로 콘텐츠에 예기치 않은 비디오 및 오디오 오류가 발생할 수 있다. (<그림 1> 참조) 콘텐츠에 포함된 오류로 인해 일차적으로는 방송 이후 품질에 대한 시청자들의 불만이 발생하고, 오류를 가진 채 비디오 아카이브에 저장된다면 이를 이용한 인터넷 다시보기 서비스 등 콘텐츠 재사용 시에도 여전히 오류가 남아 콘텐츠 서비스의 품질을 저하시킨다. 이러한 오류를 찾는 가장 단순한 방법은 사람이 눈으로 직접 검사하는 것이다.

하지만, 비디오 아카이브와 같은 방대한 자료를 육안 검사하는 것은 사실상 불가능하며, 특히 1 프레임 오류의 경우 1/30 초, 정말 눈 깜박할 사이에 발생하므로 사람의 눈으로 찾지 못하는 경우도 많고, 오류를 다행히 찾았다 하더라도 오류로 분류할지 아닐지에 대한 판단 또한 주관적이어서 일관성을 보장할 수 없는 단점이 있다. 이러한 문제점을 해결하기 위하여 콘텐츠에 포함된 다양한 비디오 및 오디오 오류를 자동으로 검사할 수 있는 자동 검사 시스템이 개발되었다. [1][2]

방송 장비는 기본적으로 블록 기반의 압축 방법을 사용하므로 오

류가 발생하는 경우 네모난 블록 모양의 오류가 많이 발생한다. (<그림 1>) 본 논문에서는 이러한 블록 오류를 고속으로 검출하는 방법을 제안한다. 관련 분야의 연구로는 blocking artifact 혹은 blockiness 라고 명명되는 블록 오류 (<그림 2>) 를 검출하는 연구가 다수 수행되었다. Blocking artifact의 경우, 8x8 혹은 16x16 블록 단위로 압축을 수행하는 경우, 블록과 블록 경계면이 직선 모양으로 선명하게 드러나는 현상이며, 이러한 현상은 결국 비디오 화질을 저하시키는 현상이다. [3][4][5]

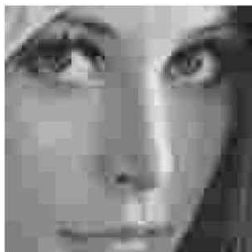
본 논문에서 검출하고자 하는 블록 오류는 <그림 3>에서 예시한 것과 같이, 8x8 혹은 16x16 의 정방형으로 국한하지 않으며, 블록과 블록 경계면의 선명도를 검사하는 것이 아닌, 블록 형태의 의도하지 않은 이미지가 영상 내에서 존재하는 지를 검사한다. 본 논문에서 검사하고자 하는 비디오 오류는 비디오 내에 포함된 물체의 윤곽을 따라서 긴 네모 형태로 분포하여, 에지 블록 오류 (edge blockiness) 라고 명명하였으며, 본 논문에서는 간단히 블록 오류로 기술하였다.

본 논문의 목적은 비디오 내에 포함된 블록 오류를 프레임 단위로 정확하게 검출하기 위한 방법을 제공하는 것이다. 이 방법은 비디오 내의 전체 프레임을 일일이 검사하여 블록 오류를 검사하는 대신, 고속의 코너 검출 후, 검출된 코너 수의 변화량을 검사하여 후보 프레임을 선

정하고, 선정된 후보 프레임에 대해서만 Adaboost 인식 기술을 이용하여 정확하게 블록 오류가 포함된 프레임을 검출해 내는 2단계를 거침으로써 고속의 블록 오류를 검출하는 것이 가능하다.



<그림 1> 비디오 오류 예시



<그림 2> blocking artifact
혹은 blockiness 오류

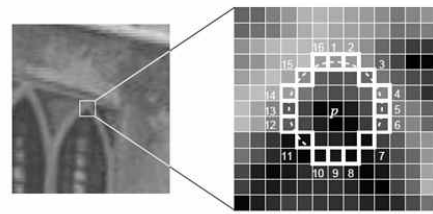


<그림 3> 블록 오류 예시

2. 코너 수 변화량 임펄스 검사 단계

고속 블록 오류 검출을 위하여 비디오 내의 매 프레임의 코너 수를 계산하고, 시간 증가에 따른 코너 수의 변화량을 검사하여 블록 오류가 포함된 후보 프레임을 찾고, 후보 프레임을 대상으로 Adaboost 인식 방법으로 학습한 분류기를 통해 최종 블록 오류가 포함된 프레임을 검출하는 2 단계를 거친다.

고속 코너 검출 방법은 참고 문헌에 표시하였다. [6][7] <그림 4>에서는 흑백영상에서의 코너 후보 영역 설명도를 나타내었다.

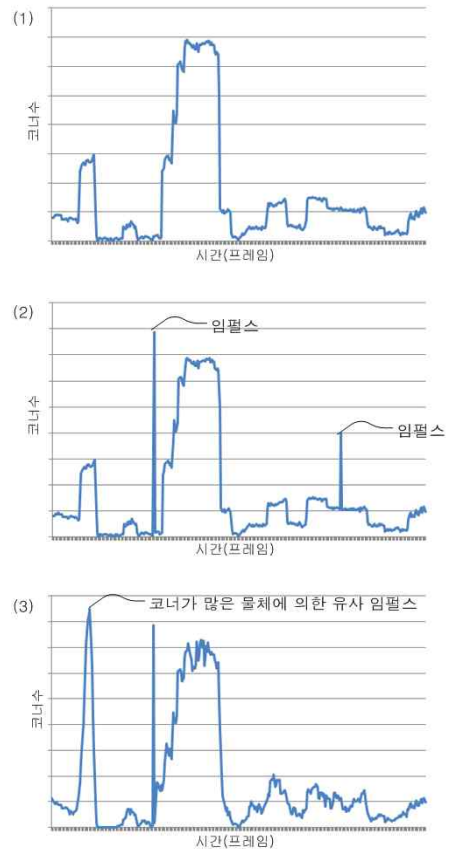


<그림 4> 흑백영상에서의 코너 후보 영역 설명도

p 를 중심으로 한 주변 화소 1~16은 p 를 둘러싼 대략적인 원을 만들었을 때 그 원의 근사적인 테두리 화소이다. 중심 p 보다 밝거나 어두운 연속적인 12개의 화소가 존재하는 경우, p 는 코너가 될 수 있다. <그림 4>의 경우 270°의 연속적인 밝은 화소가 존재하고, 90°의 연속적인 어두운 화소가 존재하여 코너가 된다.

이 외에도 고속 코너 검출 방법으로 1, 5, 9, 13에 위치한 화소값을 우선 검사하거나, 학습 방법을 통한 고속 코너 검출 방법 등 다양한 코너 검출 방법이 연구되었다.

블록 오류가 포함된 비디오 프레임에서는 코너 수의 변화량이 순간적으로 급격하게 증가한다. 이러한 현상은 시간상으로 코너 수를 열거한 그래프 상으로는 임펄스의 형태로 나타나며, <그림 5>의 (2)에서 설명하고 있다.



<그림 5> 시간의 증가에 따른 코너 수 변화량 그래프

<그림 5>의 (1)은 블록 오류가 없는 정상적인 비디오의 경우 시간 증가에 따른 코너 수 그래프이다. <그림 5>의 (2)는 블록 오류가

포함된 비디오의 경우 시간 증가에 따른 코너 수 그래프이다. 임펄스를 확인할 수 있다. <그림 5>의 (3)은 블록 오류가 포함되어 있지 않지만, 코너가 많이 포함된 물체가 포함된 경우 (2)의 경우와 유사한 임펄스가 발생하는 경우를 설명하고 있다.

코너가 많은 물체가 포함된 예시 영상은 <그림 6>에 제시하였다. 이러한 영상은 오검출(False Detection)의 원인이며, 코너 수 변화량을 조사하는 1단계에서는 구분해 낼 수가 없다. 따라서, 1단계에서는 블록 오류가 포함된 프레임과 코너가 많이 포함된 물체가 포함된 프레임이 후보 프레임으로 모두 선정될 수 있다.



<그림 6> 코너가 많은 물체가 포함된 영상

3. Adaboost 블록 오류 인식 단계

일반적인 인식 알고리즘의 경우 학습된 결과만을 적용하더라도 많은 연산이 요구되며 HD급 콘텐츠와 같은 고품질 비디오에서의 고속 처리에 부적합하다. 이전에 설명한 임펄스를 검출하는 과정에서 검출된 프레임은 전체 비디오에서 극히 일부만 해당되어, 이러한 최소한의 검출된 후보 프레임에 대하여 인식 알고리즘을 사용하도록 하였다.

본 논문에서는 블록 오류 영상 집단과 유사 영상 집단을 판별하기 위하여 Adaboost 알고리즘을 사용하였다. Adaboost는 학습에 의하여 단지 두 집단에 대한 인식을 가능케 하는 알고리즘이다. <그림 7>에서는 학습에 사용된 영상을 예시하였다. Adaboost 인식 방법은 참고문헌 [8]에서 설명되며 다음으로 요약된다.

- ① 집합 $x_i \in X, y_i \in Y = \{-1, +1\}$ 에 속하는 $(x_1, y_1), \dots, (x_m, y_m)$ 이 주어질 때,
- ② 초기화 $D_1(i) = \frac{1}{m}$
- ③ For $t = 1, \dots, T$:
 - 분포 D_t 에 따라 학습 데이터 생성
 - 예러가 최소인 약 분류기 $h_t : X \rightarrow \{-1, +1\}$ 선택

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$$
 - 가중치 계산

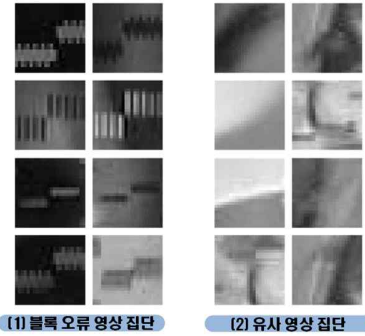
$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$
 - 분포 갱신

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) e^{\alpha_t h_t(x_i)}}{Z_t}$$

④ 강분류기 생성

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$



<그림 7> 학습에 사용된 영상 예시

또한, Adaboost의 입력 파라미터로 자주 사용되는 Haar-like features은 참고문헌 [9]의 방법으로 요약된다. Haar-like features를 고속 연산하기 위하여 고안된 Integral Image의 방법에 관련된 것 역시 [9]에 설명된다.

여기서 Haar-like features로는 <그림 8>, <그림 9>에 도식화 된 다섯 가지 형식에 대하여 다양한 크기와 위치를 갖는 2832 개를 사용한다. 각각의 경우에 대한 Haar-like feature 값은 아래의 수식을 통하여 결정된다.

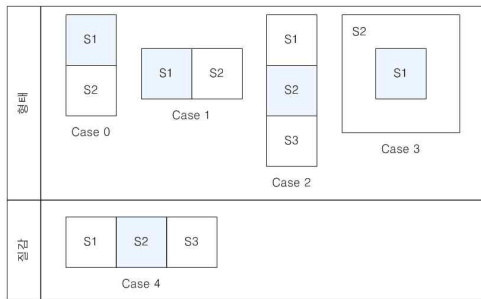
- Case0 : $f = S_2 - S_1$
- Case1 : $f = S_2 - S_1$
- Case2 : $f = S_1 - S_2 + S_3$
- Case3 : $f = S_2 - S_1$
- Case4 : $f = S_1 - S_2 + S_3$

다음은 S_n 을 생성하는 수식으로 $I(x, y)$ 는 흑백 영상을 나타낸다.

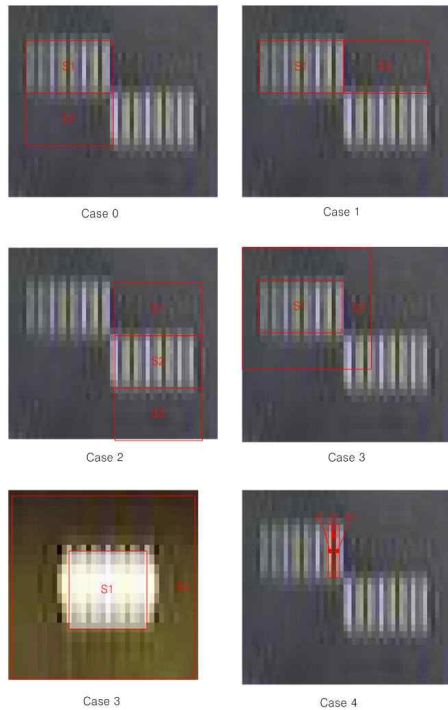
$$S_n(x, y) = \sum_x \sum_y I(x, y)$$

S_n 은 각 블록 안에서의 에너지를 나타내는데, 블록 내부의 모든 화소 값의 합에 해당 된다. 이때 case 0 ~ case 4 Haar-like features를 32x32 블록 내에 크기와 위치를 달리 배치한 2832개의 특징을 산출하는 데는 엄청난 연산을 필요로 한다. 훈련 과정에서의 연산량은 실제 인식 과정에서 영향을 미치지 않는으나, Haar-like feature의 경우 인식 과정에 입력인자로 사용되어야 함으로 Integral Image를 사용하는 방법으로 연산한다.

인식과정에서의 학습 예로 인식 블록 크기인 32x32 블록 오류 화상 5000개와 일반 화상의 32x32 화상 8000개에 대하여 case 0 ~ case 4의 2832 차원 특징 값을 입력하여 학습을 시킨다. 여기서 만들어진 강분류기에 비디오 흑백영상을 입력으로 32x32로 나누어 산출된 case 0 ~ case 4의 2832 차원 특징 값을 입력하여, 블록 오류인 경우와 아닌 경우를 인식한다.



<그림 8> Haar-like features의 설명도 (1)



<그림 9> Haar-like features의 설명도 (2)

4. 실험 및 결과

본 논문의 제안 방법을 통해 비디오 내에 포함된 블록 오류를 프레임 단위로 정확하게 고속 검출 하는 것이 가능하게 된다. 이것은 비디오 내에 블록 오류가 있는지를 사람이 일일이 육안 검사하는 방법보다 훨씬 정확하며 고속으로 검사하는 것이 가능하다. 육안 검사의 경우 VCR 등을 통해 재생하면서 오류 발생 의심 구간은 다시 플레이백하여 천천히 재생하는 등의 방법으로 검사할 수 밖에 없어 검사 시간이 길어진다. 하지만, 개발한 시스템을 사용하면 이러한 불편 없이 고속검사가 가능한 것으로 확인되었다. SD급 300초 길이, HD급 2438초 길이의 샘플 콘텐츠를 대상으로 인텔 제온 2.66GHz 6코어 2CPU, 12GB 메모리, 윈도우즈 서버 2008 컴퓨터에서 알고리즘을 구현하고, 그 실험 결과를 <표 1>에 나타내었다. SD급의 경우 실시간 대비 2.3배속 가량의 고속 검사가 가능하고 HD의 경우에도 0.8배속의 수준의 고속 검사가 가능함을 확인하였다.

<표 1> 검사 속도 실험 결과

해상도	길이 (sec)	검사 시간 (sec)	검사 속도 (실시간 대비)
SD (720x480)	300	131	2.3 = 300/131
HD (1920x1080)	2438	3086	0.8 = 2438/3086

5. 결론

본 논문에서는 비디오 내에 포함되는 블록 오류를 대상으로 하는 고속 오류 검출 방법을 설명하였다. 제안한 방법은 비디오 내의 매 프레임의 코너 수를 계산하고, 시간 증가에 따른 코너 수의 변화량을 검사하여 블록 오류가 포함될 것으로 예상되는 후보 프레임을 찾는 1단계 과정과, 후보 프레임을 대상으로 Adaboost 인식 기술을 사용하여 학습한 분류기를 통해 최종 블록 오류가 포함된 프레임을 검출하는 2단계 과정으로 구성하였다. 시스템 구현 실험 결과, 비디오 내에 포함된 블록 오류를 프레임 단위로 정확하게 고속 검출 하는 것이 가능함을 확인하였다.

향후, KBS 비디오 아카이브에 대응량으로 축적되는 미디어 파일에 대응할 수 있도록 속도 향상과 검출율을 높이기 위한 노력이 계속될 것이고, 오류 복원 기술까지 확장할 계획이다.

6. 참고문헌

- [1] Baton, File-based Quality Check Solution, <http://www.interrasystems.com>
- [2] VidChecker, File-based Quality Check Solution, <http://www.vidcheck.com>
- [3] Tecktronix Inc, "비디오 신호 장애의 고조파 측정 방법", 대한민국 등록 특허 10-0680922, 2007
- [4] Janko et al, Tecktronix Inc Beaverton, OR(US), "Harmonic Measurement of Blockiness in video signals-United States Patent", Aug. 2002
- [5] George A. Triantafyllidis, Dimitrios Tzovaras, Michael G. Strintzis, "Blocking artifact detection and reduction in compressed data", Journal: IEEE Transactions on Circuits and Systems for Video Technology - TCSV, vol. 12, no. 10, pp. 877-890, 2002
- [6] M. Trajkovic and M. Hedley, "Fast corner detection", Image and Vision Computing 16 (2): 75-87, 1998
- [7] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", European Conference on Computer Vision, May. 2006
- [8] Yoav Freund, Robert E. Schapire, "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting", 1995
- [9] Viola, P.; Jones, M., "Rapid Object Detection Using a Boosted Cascade of Simple Features", IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), ISSN: 1063-6919, Vol. 1, pp. 511-518, December 2001