

GPU 기반 스크래치 탐지 알고리즘의 설계

이준구⁰, 한기선^{**}, 유병문^{***}, 황두성^{*}

⁰단국대학교 컴퓨터학과, ^{**}강동대학교 방송영상미디어과, ^{***}(주)엘앤와이비전

e-mail:leejg01679@gmail.com⁰, ksdream1@hanmail.net^{**},

byoungmoon.you@gmail.com^{***}, dshwang@dankook.ac.kr^{*}

Design of Scratch Detection Algorithm based on GPU

Joon-Goo Lee⁰, Ki-Sun Han^{**}, Byoung-Moon You^{***}, Doo-Sung Hwang^{*}

⁰Dept. of Computer Science, Dan-Kook University

^{**}Dept. of Visual Broadcasting Media, Gang-Dong College

^{***}L&Y Vision Technologies, Inc.

● 요약 ●

영상의 스크래치 탐지는 프레임 간 화소 데이터의 비교에 있어서 많은 처리 시간을 필요로 한다. 본 논문은 스크래치 탐지 알고리즘이 GPU에서 수행할 수 있도록 병렬 설계를 제안하고, 국가 기록원 소장 디지털화 영상에 대해 실험하였다. 실험에서 제안하는 방법은 순차적 스크래치 탐지 방법과 비교하여 약 5배의 처리 시간을 단축했으며, 탐지율은 각 방법 모두 60% 정도로 유사함을 보였다.

키워드: 스크래치 탐지(scratch detection), 병렬처리(parallel processing), CUDA, GPU

I. 서론

디지털 기술이 발전하며 효율적인 멀티미디어 서비스를 위해 과거에 만들어진 아날로그 영상들을 디지털화 하고 있다. 하지만 아날로그 필름의 영사 또는 복사 등의 과정에서 원본 영상에 훼손이 발생하여 디지털화된 영상에도 훼손이 나타난다.

선형 스크래치는 디지털화된 영상에서 많이 발생하는 훼손이다. Kokaram[1]은 선의 윤곽(line profile) 모델을 이용하고, Bruni 등[2]은 Kokaram's 모델과 프레임 횡단면의 휘도 증감률을 이용했으며, Gullu 등[3]은 시간적으로 연속한 프레임에 일관적인 스크래치 후보를 이용하여 스크래치를 탐지하였다.

한편, CUDA(Compute Unified Device Architecture)를 이용해 GPU(Graphics Processing Units)를 범용적인 분야에 적용하는 기술이 각광받고 있으며, 이를 이용해 본 논문에서 소개하는 스크래치 탐지 알고리즘을 병렬화할 수 있다.

본 논문에서는 CUDA를 이용한 GPU 기반 병렬 설계를 통해 고속 스크래치 탐지 방법을 제안한다. 2장에서는 스크래치 탐지 알고리즘과 병렬 설계 방법을 소개하고, 3장에서는 소개한 알고리즘들을 국가기록원 소장 디지털화 영상에 적용하고 결론을 도출한다.

II. 제안 알고리즘

1. 스크래치 탐지

선형 스크래치는 연속한 프레임들에서 주변 화소와 비교해 급격하게 밝은 선의 모양을 나타내므로 기준 프레임과 연속한 참조 프레임을 흑백으로 변환한다. 선형 스크래치는 프레임 내에서 연속적이지 않을 수 있기 때문에 먼저 두께에 따른 스크래치 세그먼트를 탐지한다. 스크래치 세그먼트가 표시된 연속한 두 프레임을 상하로 연결 하여 실제 선형 스크래치를 탐지한다(그림 1).

기준 화소로부터 x축으로 인접한 화소의 거리를 나타내는 $\Delta = \{-2, -1, 1, 2\}$ 이고, $p(x,y)$ 가 해당 좌표 화소의 밝기 값일 때, 주변 화소와의 비교를 통해 스크래치 화소를 계산하는 $S_d(x,y)$ 는 다음과 같다.

$$S_d(x,y) = \sum_{i \in \Delta} 1(p(x,y) - p(x+i,y) - \theta_1) \text{ -----(1)}$$

여기서, θ_1 은 화소 값의 차이가 스크래치를 나타내는지 판단하는 임계값이고, $x > 0$ 이면 $1(x)=1$ 그 외는 $1(x)=0$ 이다. $S_d(x,y)$ 의 값이 Δ 의 요소 개수와 같으면 해당 화소를 스크래치 화소로 설정한다. $S_d(x,y)$ 가 스크래치 화소이면서, 세로축으로 인접한 화소들도 스크래치 화소인 경우, 해당 영역을 스크래치 세그먼트로 설정한다. $S_d(x,y)$ 를 중심으로 하여 스크래치 세그먼트를 계산하는

이 논문은 행정안전부 국가기록원 재원으로 2013년 기록보존기술 연구개발사업의 지원을 받아 수행된 연구임.

$S_n(x,y)$ 는 다음과 같다.

$$S_n(x,y) = \sum_{j=y-\tau/2}^{y+\tau/2} 1(S_d(x,j) - |\Delta|) \text{ -----(2)}$$

여기서, τ 은 세그먼트의 길이를 나타내며, $S_n(x,y)$ 가 τ 의 70% 이상이면 $p(x,y-\tau/2)$ 부터 $p(x,y+\tau/2)$ 영역을 스크래치 세그먼트로 설정한다.

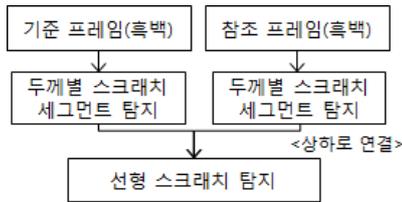


그림 1. 스크래치 탐지 과정
Fig. 1. Scratch Detection Process

스크래치 세그먼트가 표시된 프레임을 상하로 연결한 후, 프레임 입과 크기가 같은 마스크를 이용해 동일한 라인에서 스크래치 세그먼트들의 길이가 기준치 이상일 경우 해당 영역의 세그먼트들을 선형 스크래치로 표시한다.

2. 스크래치 탐지 병렬 알고리즘

스크래치 세그먼트 탐지 모듈은 대량의 각 화소에 대해 동일한 연산을 수행하며 순차성이 없으므로 병렬처리에 매우 적합한 구조이다. 따라서 스크래치 세그먼트 탐지 모듈에 대해 병렬 알고리즘을 설계하며, 상대적으로 느린 글로벌 메모리의 접근을 공유 메모리[4]로 대체하여 효율을 높인다.

프레임의 화소 수만큼 스레드를 구성하여 하나의 스레드가 하나의 화소를 계산 한다. 원본 프레임 src로부터 스크래치 세그먼트를 계산하여 seg 행렬에 저장하는 GPU 커널 findSegment는 다음과 같다.

```

kernel findSegment(Mat src, Mat seg){
    Num count = 0;
    shred_memory s_src[H][W] = src[H][W];
    Loop k from 0 to  $\tau$  step 1
        Index y = blockIdx- $\tau/2+k$ ;
        Index x = threadIdx;
        if(s_src[y][x]-s_src[y][x+ $\Delta$ ]>theta_1)
            count++;
        seg[blockIdx][threadIdx]=count;
    }
  
```

여기서, Mat은 행렬을 나타내고, s_src는 공유 메모리이며, blockIdx와 threadIdx는 스레드의 고유 아이디를 결정하는 요소이다.

III. 실험 및 결론

본 논문에서는 스크래치가 존재하는 디지털화 영상으로부터 CUDA를 사용한 GPU 기반의 고속 병렬처리 스크래치 탐지 방법을 제안하였다. 제안한 알고리즘의 성능을 평가하기 위하여 국가 기록원의 디지털화 필름들 중 스크래치의 출현 빈도가 높은 1500 프레임을 추출하여 사용하였다. 실험을 위해 각 식에서 사용한 θ_1 은 스크래치와 정상 화소의 차이 값의 평균인 10을 사용했고, τ 와 Δ 는 실험을 통한 적정 수로 각각 9와 좌우 2씩 설정했다.

프레임 한 장을 처리하기 위해 순차 기반 알고리즘은 54ms, 병렬 기반 알고리즘은 11ms가 소요되었으며, 각 알고리즘 모두 60%에 근접한 탐지율을 보였다. 결과적으로 GPU 기반의 병렬 알고리즘은 순차 기반의 알고리즘[5]과 비교하여 5배의 처리시간을 단축하면서도 유사한 검출률을 보였다.

참고문헌

- [1] A. C. Kokaram, "Detection and Removal of Line Scratches in Degraded Motion Picture Sequences," in *Proc. Signal Processing VIII : Theories and Application*, pp. 5-8, Trieste, Italy, Sept. 1996.
- [2] V. Bruni, D. Vitulano, "A Generalized Model for Scratch Detection," *Image Processing, IEEE Transaction on*, vol. 13, no. 1, pp. 44-50, Jan. 2004.
- [3] M. K. Gullu, O. Urhan, S. Erturk, "Scratch Detection via Temporal Coherency Analysis and Removal using Edge Priority Based Interpolation," in *Proc. IEEE Intl. Symposium on Circuits and Systems*, pp. 92-96, Island of Kos, Greece, May 2006.
- [4] D. B. Kirk, W. M. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach," Elsevier, pp. 83-90, 2010.
- [5] B. M. You, K. T. Jung, S. K. Kim, D. S. Hwang, "Detection and Restoration of Vertical Non-linear Scratches in Digitized Film Sequences", in the 2012 Intl. Conf. on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, NV, July 2012.