

서베일런스 환경을 위한 압축 도메인에서 다채널 실시간 움직임 검출 및 저장 시스템

*오상건 *김영웅 *안용조 **김용성 **김승환 **조형준 *심동규
 *광운대학교 컴퓨터공학과 **SK telecom
 *{xjw0118, mipjji, yongjoahn, dgsim}@kw.ac.kr
 **{yns.kim, seunghwan.kim, hjcho1}@sk.com

Real-Time Motion Detection and Storage Method on a Compressed Domain for Multi-channel Video Surveillance Monitoring System

*Xiangjian wu, *Youngwoong Kim, *Yong-Jo Ahn, **Yong-sung Kim,
 **Seung-Hwan Kim, **Hyung-Jun Cho, and *Donggyu Sim
 *Kwangwoon University **SK telecom

요 약

본 논문에서는 압축 도메인에서 고속으로 움직임을 검출하고 해당 구간을 저장 하는 알고리즘을 제안한다. 제안하는 알고리즘은 H.264/AVC 기반의 압축 비트스트림에서 움직임 벡터와 참조프레임을 이용하여 움직임이 있는 프레임은 검출하고 움직임 유무에 따라 GOP 단위로 저장하는 과정을 수행한다. 압축도메인에서 움직임 검출과 구간 저장을 수행함으로써 복잡도를 낮추고 비디오 저장을 위한 공간을 절약해 실시간 다채널 영상 처리에 최적화 된 성능을 제공한다. 제안하는 움직임 검출 및 저장 시스템은 single thread 환경에서 실시간으로 평균 2957 프레임을 처리 가능하며, Multi thread의 경우 30 fps 영상 98개 채널을 실시간으로 처리 가능하다.

1. 서론

최근 다양한 영상 보안 기술들이 발전하면서 서베일런스 시스템은 범죄, 사고 해결에 중요한 단서를 제공 하고 있다. 동시에 높은 해상도의 서베일런스 비디오에 대한 수요가 높아지면서 취득한 비디오의 저장 공간과 부호화 및 복호화의 복잡도 역시 크게 증가하였다. 이러한 문제를 해결하기 위하여 비디오 내의 실제 움직임이 존재하는 구간만을 찾는 연구가 진행되어 왔다. 움직임이 있는 구간만을 저장할 수 있다면 부호화 및 복호화의 오버헤드뿐만 아니라 검출된 구간의 저장 공간 역시 크게 줄일 수 있다. 이러한 움직임 검출 알고리즘의 연구는 크게 픽셀 도메인 검출과 압축도메인 검출로 분류할 수 있다. 픽셀 도메인의 움직임 검출은 높은 정확도와 정교한 움직임 검출이 가능하지만 복잡도가 높고 복호화된 비디오를 저장하기 위해 많은 저장 자원을 필요로 한다. 반면 압축 도메인 움직임 검출은 픽셀 도메인 방식에 비해 움직임 검출 시 정확도와 정교함이 떨어지지만 낮은 복잡도와 검출된 비트스트림 자체를 저장 할 수 있어 저장 공간의 문제를 해결 할 수 있다. 그럼에도 불구하고 실시간 다채널 비디오를 처리하기 위해서는 복잡도를 더 낮추어야 한다. 제안하는 알고리즘에서는 연산 복잡도를 낮추기 위해 GOP 단위의 시스템을 설계하였고, 실제로 한 GOP내의 프레임에서 움직임이

검출되면 해당 GOP의 남은 프레임들에 대한 움직임 검출 알고리즘을 생략한다.

비디오 압축 과정에서의 움직임 예측은 최소 왜곡을 갖는 정보를 움직임 벡터로 선택하기 때문에 실제 움직임과 차이가 날 수 있다. 따라서 압축 도메인에서 움직임 검출 알고리즘의 기존 연구[1]에서는 실제 움직임과 움직임 벡터 사이의 유사도를 평가하는 방법으로 ‘confidence’ 값을 사용한다. ‘confidence’ 값은 시공간적 움직임의 유사도를 계산하여 0과 1사이에 값으로 나타내며, ‘confidence’ 값이 1에 가까울수록 실제 움직임 벡터와 유사함을 의미한다. 하지만 4x4 블록단위의 연산처리를 바탕으로 하기 때문에 높은 복잡도의 문제가 존재 한다. 일반적으로 서베일런스 환경에서는 촬영된 비디오의 저장능력을 극대화하기 위해 높은 압축률로 비디오 부호화를

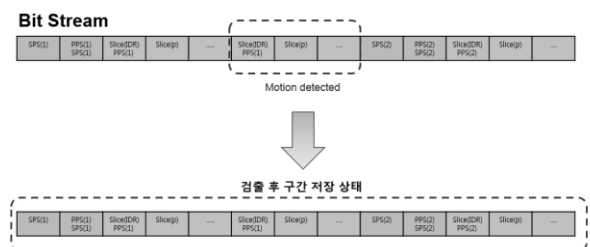


그림 1. 움직임 검출 구간 저장

수행한다. 선행 조사 결과 16x16, 8x8 과 같이 큰 블록 단위의 선택 비율이 99%로 높게 나타난다. 따라서 기본 연구 [2]와 같이 8x8 블록단위 움직임 검출 알고리즘을 적용하였다. 이를 기반으로 움직임 검출 되면 해당 GOP를 저장하고 검출 과정을 생략하는 알고리즘을 제안한다. 본 논문의 2장에서는 제안하는 시스템에 대해 설명하고, 3장에서는 실험 결과를 토대로 알고리즘을 검증한다. 마지막으로 4장에서 결론을 맺는다.

2. 제안하는 알고리즘

제안하는 알고리즘은 기존연구[2]를 바탕으로 다채널의 H.264/AVC 기반 압축 비트스트림들에서 움직임을 검출하고 저장하는 알고리즘이다. 입력 비트스트림은 네트워크 전송 단위인 NAL(Network Abstraction Layer) 단위로 전송되며, NAL 은 SPS, PPS, SLICE, SEI 등으로 구성된다. SPS(Sequence Parameter Set)과 PPS(Picture Parameter Set)은 비디오를 복호화 하기 위한 상위레벨 신텍스를 전송하는데 사용된다. Slice 에는 각각의 slice 를 복호화하기 위한 신텍스 정보를 포함하고 있으며, SEI 는 복호화를 위한 추가적인 정보를 전송하는데 사용된다. 실제로 움직임이 검출된 GOP(Group Of Picture)를 저장하기 위해서는 GOP 내의 slice 데이터와 해당 GOP 가 참조하는 SPS, PPS 를 함께 저장해야 한다. H.264/AVC 비트스트림 GOP 단위의 구조를 바탕으로 제안하는 움직임 검출 구간 알고리즘에서는 움직임이 검출된 구간에 해당하는 GOP 의 앞/뒤 GOP 를 추출 하여 저장한다. 그림 1 은 하나의 GOP 내에서 검출된 프레임들 저장하기 위한 구간 저장 방식을 나타낸다. 움직임 검출 여부에 따른 저장 방식은 다음과 같다.

1) 현재 GOP에서 움직임이 검출되지 않은 경우, 임시 버퍼에 해당 GOP 비트스트림을 임시 저장

2) 현재 GOP에서 움직임이 검출되면, 해당 GOP에서 아직 처리되지 않은 프레임에 대하여 조기 종료를 수행하고, 임시 버퍼에 저장된 이전 GOP와 이후 GOP에 해당하는 비트스트림을 저장

그림 2 에서는 GOP 내의 하나의 프레임에서 움직임이 검출되었을 때 나머지 프레임에서 움직임 검출을 생략하는 과정을 보여준다. 검출 된 프레임에서 다음 I 프레임을 수행 하기 전까지 검출 알고리즘은 생략되고 검출된 GOP 의 정상 복호화가 가능하도록 저장한다. 그림 3 에서는 제안하는 알고리즘의 다채널 병렬 처리 과정을 설명한다. N 개의 비디오 비트스트림을 입력으로 받아 N 개의 thread 를 생성하여 각각 독립적으로 병렬처리를 수행한다.

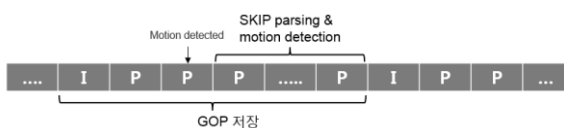


그림 2. GOP 기반 움직임 검출 알고리즘 스킵

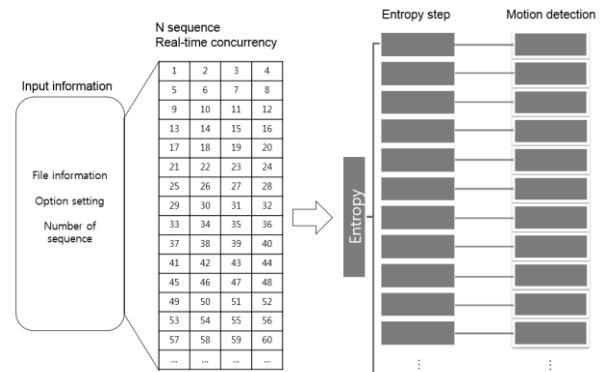


그림 3. 다채널 구성 블록도

3. 실험결과 및 분석

가. 테스트 셋

실험 비디오 촬영에 사용된 카메라 모델 HDS-S422 실외 카메라, Axis M1054 실내 카메라, NVT-4211R 실외/실내 카메라이며, 각 비디오 처리에 대한 single thread 에서 처리 속도는 표 1 과 같다. 또한, 제안하는 시스템으로 64 채널 병렬처리를 위한 비디오 구성은 다음 표 2 와 같다. 병렬 처리 비디오 셋은 각각 비트스트림 64 개로 구성되어 있으며, set1, set2 는 각각 CABAC 과 CAVLC 로 부호화된 비트스트림으로 움직임이 있는 구간과 없는 구간이 모두 존재한다. Set3 은 움직임이 없는 CAVLC 비트스트림으로 구성 되어있다. 제안하는 알고리즘에서는 비디오에 움직임이 존재하지 않을 때 움직임 검출 알고리즘을 계속 수행하기 때문에 이런 경우에도 실시간 병렬처리 가능 여부를 테스트 하기 위해 확인해야 한다

표 1. single thread 테스트 셋

Video name	해상도	fps	GOP 크기	Frame number	비디오 구성
Video1	1280x720	30	30	820	자동차/실외
Video2	1280x720	30	30	547	자동차/실외
Video3	1280x720	30	15	2010	사람/실외
Video4	1280x720	30	15	2755	사람/실내
Video5	1280x720	30	30	3037	사람/실내
NVT-4211R (Video1, Video2)					
HDS-S422 (Video3)					
Axis M1054 (Video4, Video5)					

표 2. 병렬 처리 비디오 셋

Video set name	해상도	fps	GOP 크기	Frame number	비디오 구성
set1	1280x720	30	30	92322*64	움직임이 존재하는 CLVLC 영상
set2	1280x720	30	30	10189*64	움직임이 존재하는 CABAC 영상
set3	1280x720	30	30	66532*64	움직임이 존재하지 않는 CLVLC영상

표 3. 싱글 스레드 움직임 검출 알고리즘 속도

Sequence name	기존방법 (fps)	제안하는 방법 (fps)	속도향상
Video1	469	2555	5.45
Video2	411	2914	7.06
Video3	365	3526	9.66
Video4	372	2206	5.93
Video5	473	3588	7.59
Average	418	2957	7.13

표 4. 멀티 스레드 움직임 검출 알고리즘 속도(64채널 기준)

Video set name	수행 시간(s)	fps
Set 1	944.23	98
Set 2	94.37	108
Set 3	885.397	75
Average	641.32	94

나. 성능 평가

제안하는 알고리즘의 움직임 검출 정확도는 recall 0.84, precision 0.94 로 높은 정확도를 보인다. 움직임 검출 및 저장 알고리즘의 성능은 Intel® Xeon® CPU E5-2687W v2 @3.40GHz 3.40GHz, 메모리 128GB, OS Windows 2008 R2 Standard 환경에서 Visual studio 2012 를 사용하여 측정했다. 다채널 실시간 처리 테스트를 위해 OpenMP 를 사용해 병렬처리를 구현하였다.

표 3 에서는 single thread 로 테스트 영상의 움직임 검출 및 저장 알고리즘의 속도를 측정한 결과이다. Single thread 로 평균 2957 fps 의 속도를 보인다. 이는 실시간으로 30 fps 의 비디오 98 채널을 동시에 처리 가능한 속도이다. 표 4 는 제안하는 알고리즘을 사용하여 64 개 채널을 병렬로 처리한 결과이다. 실험 결과 평균 94 fps 의 처리속도를 보이며, 병렬처리를 적용하여 실시간으로 64 채널 이상의 동시 처리가 가능함을 보여준다.

4. 결론

본 논문에서는 기존 알고리즘[2]을 바탕으로 서버일런스 환경에서 압축된 비디오 비트스트림의 움직임 벡터와 참조 프레임 정보를 이용하여 움직임을 검출하고 검출 구간을 비트스트림으로 저장하는 알고리즘을 제안하였다. 제안하는 알고리즘은 1280x720 해상도 비디오를 실시간으로 98 채널 동시 처리 가능한 속도를 보인다. 향후, 비트스트림의 크기 변화를 기반으로 비디오에서 움직임이 없는 구간을 판별하여 움직임 검출 및 저장 알고리즘의 속도를 향상시킬 계획이다.

감사의 글

이 논문의 일부는 2014 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2010-0025325).

참고문헌

- [1] K. Szczerba, S. Forchhammer, J. Støttrup-Andersen, and P. T. Eybye, "Fast Compressed Domain Motion Detection in H.264 Video Streams for Video Surveillance Applications," in Proc. Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS' 09), Sep. 2009, pp. 478-483. R. C. Baker and B. Charlie, "Nonlinear unstable systems," International Journal of Control, vol. 23, no. 4, pp. 123-145, May, 1989.
- [2] 김영웅, 오상진, 안용조, 김용성, 김승환, 조형준, 심동규, "A compressed domain fast motion detection for IP video surveillance monitoring system" 2014 년도 제 27 회 신호처리합동학술대회, 2012 년 10 월