

논문 2013-50-12-16

멀티 코어 프로세서 기반의 영상 감시 시스템을 위한 침입 탐지 처리의 가속화

(Acceleration of Intrusion Detection for Multi-core Video Surveillance Systems)

이 길 범*, 정 상 진*, 김 태 환**, 이 명 진**

(Gil-beom Lee, Sang-jin Jung, and Tae-Hwan Kim[Ⓢ], and Myeong-jin Lee)

요 약

본 논문은 멀티 코어 프로세서 기반의 영상 감시 시스템을 위한 침입 탐지 처리의 가속화를 제안한다. 침입 탐지 처리의 가속화를 위해 병렬화를 진행하였고, 이를 위해 기존 침입 탐지 알고리즘을 분석하고 데이터 의존성을 고려하여 프레임 단위의 병렬화된 처리 구조를 설계하였다. 병렬화된 침입 탐지 처리의 유효성을 검증하기 위하여 다중 쓰레드 기반의 프로그램으로 구현하여 침입 탐지의 가속화 정도를 측정하였다. 구현한 침입 탐지 처리 프로그램의 탐지 속도는 논리적 쓰레드를 8개까지 구현할 수 있는 환경에서 기존 단일 쓰레드 처리 대비 최대 353.76%가 향상되었다.

Abstract

This paper presents a high-speed intrusion detection process for multi-core video surveillance systems. The high-speed intrusion detection was designed to a parallel process. Based on the analysis of the conventional process, a parallel intrusion detection process was proposed so as to be accelerated by utilizing multiple processing cores in contemporary computing systems. The proposed process performs the intrusion detection in a per-frame parallel manner, considering the data dependency between frames. The proposed process was validated by implementing a multi-threaded intrusion detection program. For the system having eight processing cores, the detection speed of the proposed program is higher than that of the conventional one by up to 353.76% in terms of the frame rate.

Keywords : Intrusion detection, Video surveillance, Acceleration, Multi-core, Multi-thread

I. 서 론

최근 영상 감시 시스템에서 영상의 해상도가 증가하고

시스템의 다채널화가 이루어지고 있다. 다채널화가 이루어진 영상 감시 시스템에서는 모니터링을 위하여 많은 인력이 필요하다. 최근의 영상 감시 시스템은 지능형 영상 감시 알고리즘^[1]을 내장하여 영상 감시를 자동화함으로써, 영상 감시를 위한 인력을 줄이고 있다. 자동화된 영상 감시 시스템은 특정 상황을 검출해내는 복잡한 연산을 수행해야하기 때문에 실시간 영상 감시를 수행할 수 있도록 가속화할 필요성이 증가하고 있다.

* 학생회원, ** 정회원, 한국항공대학교 항공전자 및 정보통신공학부
(School of Electronics, Telecommunication & Computer Engineering, Korea Aerospace University)

Ⓢ Corresponding Author (E-mail: taehwan.kim.kr@gmail.com)

※ 본 연구는 경기도지역협력연구센터 (GRRC) 프로그램에 따른 한국항공대학교 차세대방송미디어 연구센터 (GRRC항공2013-B04) 지원으로 수행되었음.

접수일자: 2013년7월29일, 수정완료일: 2013년11월27일

본 논문에서 가속화의 대상인 침입 탐지 알고리즘^[2] 또한 다양한 상황에서 물체 탐지 및 추적과 침입 탐지의 정확도를 유지하기 위해 높은 연산 복잡도를 가지고 있

다. 고속의 물체 탐지 및 추적을 위한 선행 연구들에서는 높은 탐지 속도를 달성하기 위하여 탐지 정확도를 희생하고 알고리즘의 연산 복잡도가 높은 부분을 다른 방식으로 변경하는 방법이 다음과 같이 연구되었다. 지능형 영상 시스템에서 배경 생성에 이용되는 가우시안 혼합 모델^[3-4]은 여러 배경 모델 중에서 비교적 정확한 성능을 갖지만 연산이 복잡하다는 단점이 있다. 그래서 실시간 요구를 만족하기 위해 연산은 간단하지만 잡음에 민감한 차분 영상 방식과 가우시안 혼합 모델을 적응적인 방식으로 함께 이용하여 배경을 생성하는 방법이 연구되었다^[5]. 또한, 연산 복잡도가 높은 무게중심 추적^[6], 물체의 경계선 추적^[7]과 같은 기존의 물체 추적 알고리즘 대신에 적응적 FIR 필터, 템플릿 정합 기법을 사용하여 실시간 물체 추적을 가능하게 한 연구가 있다^[8]. 앞서 언급한 것처럼 위 연구들은 연산 복잡도의 감소로 처리 속도는 증가하지만 탐지 정확도는 기존 방식보다 감소한다.

최근 연구에서 영상 감시 알고리즘의 가속화를 위한 다른 방법으로 멀티 코어 프로세서를 기반의 병렬화를 이용한 가속화가 제시되고 있다. 그 중 하나로 물체 탐지 알고리즘 중 하나인 WaldBoost 알고리즘^[9]을 병렬화하여 가속화한 연구가 있다^[10]. 또한 Histogram of Gradients 기반의 물체 탐지 알고리즘^[11]과 물체 추적을 위한 Earth Mover's Distance 알고리즘^[12]을 병렬화해서 가속화한 사례도 있다^[13]. 위와 같은 사례들은 지능형 영상 감시 알고리즘 전체가 아니라 일부분인 물체 탐지 및 추적 부분만 단편적으로 병렬화하여 가속화한 연구들이다.

본 논문에서는 지능형 영상 처리 알고리즘의 가속화를 위한 방법으로 멀티 코어 프로세서 기반의 병렬화를 이용하며 침입 탐지 알고리즘의 전체 과정을 병렬 처리 구조로 설계했다. 본 논문에서 제안하는 내용의 요점은 다음과 같다.

- 1) 침입 탐지 알고리즘의 전체 처리 과정을 병렬 처리 구조로 설계했다. 병렬화 과정에서 기존 침입 탐지 알고리즘^[2]의 데이터 의존성을 분석하여 해결하고 멀티 코어 프로세서 기반의 다중 쓰레드로 구현하여 검증했다.
- 2) 병렬화한 침입 탐지 처리를 구현한 결과 논리적 쓰레드를 8개까지 구동할 수 있는 환경에서 최대 353.76%의 속도 향상이 이루어졌고 알고리즘의 변경이 없으므로 침입 탐지의 정확도는 그대로 유지 할 수 있다.

본 논문은 다음과 같이 구성된다. 먼저 II장에서 병렬화의 대상인 침입 탐지 알고리즘을 분석한다. III장에서는 침입 탐지 알고리즘의 병렬 처리 구조를 제시하고 데이터 의존성의 원인과 해결 방법을 설명한다. 그리고 IV장에서 제안하는 방법의 가속화 결과를 실험을 통해 검증하고 V장에서 결론을 맺는다.

II. 기존 침입 탐지 알고리즘의 분석

1. 침입 탐지 알고리즘의 문제 정의

침입 탐지 알고리즘^[2]은 접근을 금지하는 Region of Interest (ROI)를 설정하여 영상 내 물체에 대한 탐지 및 추적을 수행함과 동시에 물체의 ROI 침입 여부를 판단한다. 그림 1은 프레임 내에 설정된 ROI와 물체 탐지 및 침입 탐지 상황을 나타낸다. 그림 1에서 보는 바와 같이, 프레임 내에 물체가 감지되면 녹색 실선 Bounding Box로 표시하고 ROI에 침입하는지를 지속적으로 감시한다. 탐지된 물체가 ROI 영역에 침입하면 침입 상황을 발생시킨 물체로 인지하고 붉은색 점선 Bounding Box로 표시하며 추적한다.

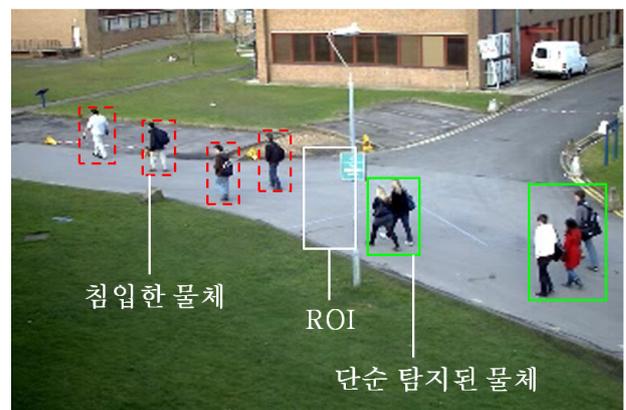


그림 1. 침입 탐지 처리의 예.
Fig. 1. Example of the intrusion detection.

2. 침입 탐지 알고리즘의 움직임 분할 기법

침입 탐지 알고리즘은 물체 탐지 및 추적 알고리즘^[14-15]을 기반으로 한다. 물체 탐지 및 추적 알고리즘은 영상에서 움직이는 물체를 탐지하고 물체의 이동을 추적하는 알고리즘이다. 물체 탐지 및 추적 알고리즘의 핵심은 배경과 움직이는 물체를 구분해내는 움직임 분할 처리이며 기존 연구에서는 주요 기법을 크게 세 가

지로 분류한다^[16].

1) 배경 차분 (Background Subtraction)^[17~18]. 배경 차분 방식은 특정 배경 모델을 기반으로 현재 영상을 배경에 반영하고 배경 영상과 현재 영상의 차이를 이용하여 전경을 분리하는 방법이다. 이 방법은 순간적인 큰 변화에 민감하다는 단점이 있지만 카메라의 촬영 각이 고정된 CCTV 영상에는 효과적이다.

2) 차 영상 (Temporal Difference)^[19]. 차 영상 방식은 처음부터 배경을 누적시키는 것이 아니라 연속된 프레임 간의 차분을 이용하여 그 순간의 배경과 전경을 구분하는 방식이다. 순간적인 큰 변화에는 강인하나 영상의 변화가 적을 때는 비효율적이다.

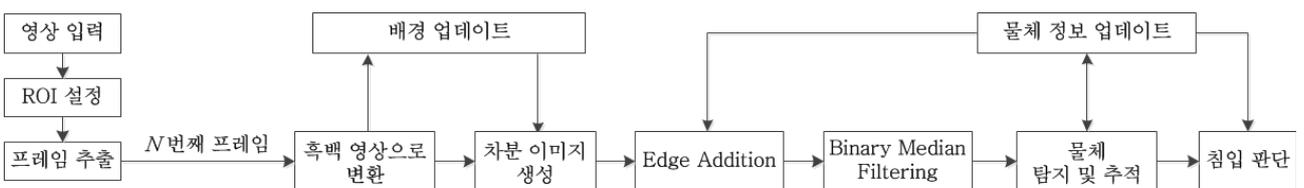
3) 광류 흐름 (Optical Flow)^[20]. 광류 흐름 방식은 영상의 단위 영역 또는 각 픽셀의 움직임 벡터를 파악하여 물체를 추출하는 방법이다. 광류 흐름 방식은 PTZ 카메라 영상처럼 촬영 각이 변화하는 상황에서도 물체를 감지할 수 있다는 장점이 있지만 많은 연산량이 단점이다.

본 논문에서 가속화하고자 하는 침입 탐지 알고리즘^[2]은 고정된 CCTV 영상을 입력 영상으로 가정한 알고리즘이다. 고정된 CCTV 영상에 대해서는 배경 차분 방식이 차 영상 방식보다 배경의 정확도 측면에서 좋은 성능을 보이고 광류 흐름 방식에 비해서는 연산량 측면에서 좋은 성능을 보인다. 침입 탐지 알고리즘에서 사용하는 움직임 분할 처리 방식은 가우시안 혼합 모델^[3~4] 기반의 배경 차분 방식이며 침입 탐지 알고리즘의 자세한 동작 흐름은 다음 절에서 설명한다.

3. 침입 탐지 알고리즘의 동작 흐름

가. 초기 설정 단계

침입 탐지 알고리즘의 전체적인 수행 순서는 그림 2와 같다. 침입 탐지 처리는 입력 영상으로부터 프레임을 추출하는 작업과 침입 상황을 감시할 ROI를 설정하는



$N = 0, 1, 2, 3, \dots$

그림 2. 침입 탐지 알고리즘의 동작흐름.

Fig. 2. Flow of the intrusion detection.

것으로 시작한다. 설정된 ROI는 이후 처리되는 모든 프레임에 대해 동일하게 적용된다.

나. 배경과 전경의 분리

침입 탐지 알고리즘에서는 앞서 언급한 대로 영상에 존재하는 물체를 파악하기 위해 가우시안 혼합 모델^[3~4] 기반의 배경 차분 방식을 이용하며 매개 변수들은 표 1에 정리했다. 침입 탐지 알고리즘은 가우시안 혼합 모델을 이용하여 매 프레임마다 업데이트된 배경 영상을 생성하고 배경 영상의 (i, j) 위치에 있는 픽셀 값 $BG(i, j)$ 와 현재 처리할 흑백 영상의 (i, j) 위치에 있는 픽셀 값 $I(i, j)$ 간의 차분을 수행한다. 그리고 각 픽셀의

표 1. 침입 탐지 알고리즘의 GMM 매개 변수.

Table 1. GMM parameter of intrusion detection algorithm.

GMM 매개 변수	값
λ (Standard Threshold)	2.5
ω (Weight)	0.05
σ^2 (Variance)	900
K (Number of Gaussian)	5
α (Learning Rate)	200



그림 3. 이진 영상에서 발생하는 물체 분리 현상.

Fig. 3. Example of the fragmentation for a binary image.

차분 결과를 Threshold T 와 비교하여 (1)과 같이 전경 영상의 (i, j) 위치의 픽셀 값 $F(i, j)$ 를 결정한다.

$$F(i, j) = \begin{cases} 1, & \text{if } |I(i, j) - BG(i, j)| > T \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

전경 영상을 생성한 이후에는 그림 3과 같이 물체가 분리되어 따로 인식되는 Fragmentation 현상을 막기 위해 이전 프레임들에서 탐지한 물체들의 Edge를 더하는 과정을 수행한다. 이 과정에서 실제 물체의 영역을 벗어나 추가된 Edge는 Binary Median Filtering을 통해 제거한다. Binary Median Filtering은 이진 영상에서 Mask를 가지고 픽셀들을 검사하여 1과 0의 개수를 파악하고 빈도수가 더 높은 값으로 중심 픽셀 값을 결정한다. Binary Median Filtering까지 완료된 영상은 최종적인 전경 영상으로 확정된다.

다. 물체 정보 파악과 침입 탐지

전경 추출이 완료되면 Contour Labeling^[21]을 통해 전경 영상 내에 존재하는 물체들을 파악하고 각 물체의 무게중심을 구한다. Contour Labeling은 이진 영상에서 8방향 탐지를 통해 픽셀 값이 1인 픽셀들을 Label Number로 연결해가며 물체를 파악한다. 무게 중심은 수평성분에 대한 무게중심과 수직 성분에 대한 무게중심으로 나뉜다. 수평성분에 대한 무게 중심 C_x 는 (2)와 같이 물체 내 존재하는 픽셀의 수평 위치 X_t 의 합을 해당 물체 내의 총 픽셀 수 N 으로 나눈 값이다.

$$C_x = \left(\sum_{t=1}^N X_t \right) / N \quad (2)$$

수직 성분에 대한 무게 중심도 동일한 방식으로 계산한다. 이렇게 구한 무게 중심은 ROI 위치와 비교하여 침입 상황을 판단하는 기준으로 사용한다. 침입 상황에 대한 판단이 끝나면 하나의 프레임에 대한 모든 처리는 끝나게 된다. 처리 결과 중 프레임 내에 존재하는 물체의 정보는 프레임 간의 참조를 위해 저장한다. 이는 앞서 언급했던 Fragmentation 현상을 방지하기 위해 Edge를 더하는 과정과 물체 탐지 및 추적, 침입 판단에 사용된다.

III. 제안하는 가속화 방법

본 장에서 제안하는 가속화 방법은 침입 탐지 알고리즘의 전체 과정을 병렬 처리 구조로 설계하는 것이다. 병렬화 과정에서 발생하는 데이터 의존성은 원인을 분석하고 멀티 코어 프로세서 기반의 다중 쓰레드로 구현하는 과정에서 쓰레드의 동기화와 쓰레드 처리 순서의 제어를 이용하여 해결했다.

1. 멀티 코어 프로세서 기반의 병렬 처리

멀티 코어 프로세서 환경에서 다중 쓰레드를 이용한 병렬 처리는 응답성, 자원 공유, 속도 측면에서 성능 향상을 기대할 수 있다. 이를 위해서는 다중 쓰레드를 이용한 병렬화 과정에서 각 쓰레드의 Load Balancing과

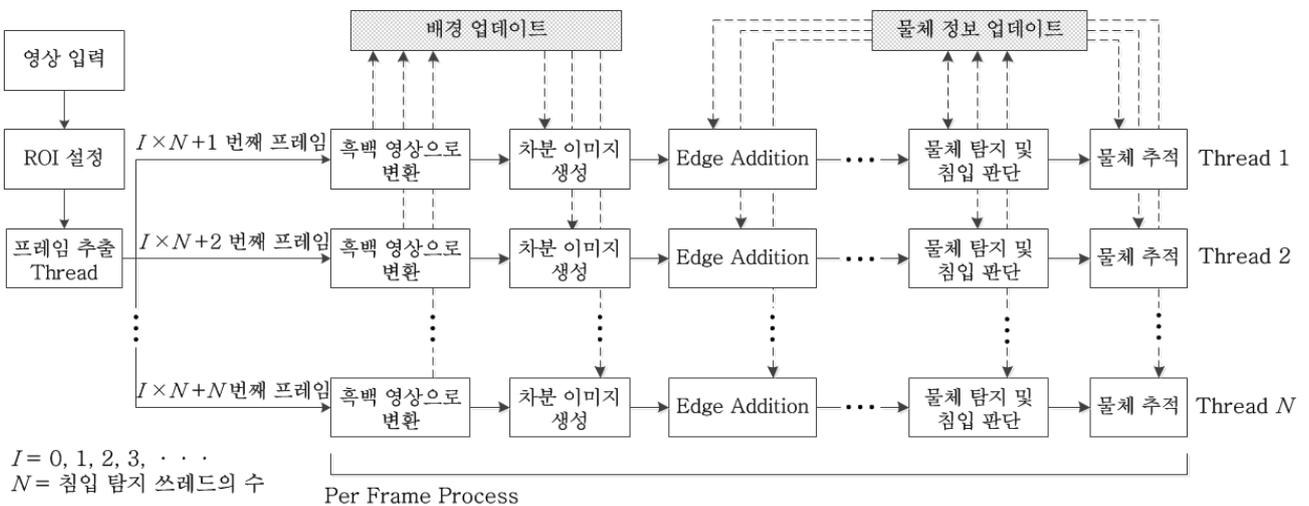


그림 4. 제안하는 병렬화된 침입 탐지 처리 (N은 쓰레드 수)

Fig. 4. Parallelization of the proposed intrusion detection, where N denotes the number of threads.

데이터 의존성 문제를 해결해야 한다.

본 논문에서는 침입 탐지 알고리즘을 분석하여 침입 탐지 처리 과정 전체를 그림 4와 같이 병렬 처리 구조로 설계하였다. Load Balancing 문제를 해결하기 위해 침입 탐지 처리 과정에서 I/O-bound Process와 CPU-bound Process를 구분했다. I/O-bound Process는 CPU 연산보다는 데이터 입출력 과정에 많은 시간을 소비하는 작업이고 CPU-bound Process는 대부분의 시간을 CPU 연산에 소비하는 작업이다. 침입 탐지 알고리즘에서 입력 영상으로부터 프레임을 추출하는 과정은 I/O-bound Process이고 프레임 추출 후 배경 업데이트와 물체 탐지를 비롯한 후처리 과정은 CPU-bound Process에 해당한다. Load Balancing 문제를 해결하기 위해 제안하는 방법은 I/O-bound Process인 프레임 추출 과정에 별도의 쓰레드를 할당하고 나머지 쓰레드들이 각각 한 프레임에 대하여 침입 탐지 처리 연산 과정을 수행하는 것이다. 이후 전자는 프레임 추출 쓰레드, 후자는 침입 탐지 쓰레드로 언급한다. 침입 탐지 알고리즘에 존재하는 데이터 의존성과 그 해결방법은 다음 절에서 설명한다.

2. 침입 탐지 알고리즘의 데이터 의존성 분석

가. 데이터 의존성에 따른 문제

침입 탐지 처리 과정에 데이터 의존성이 없다면 프레임 추출 과정과 프레임 단위의 각 침입 탐지 처리 과정들이 독립적으로 수행될 수 있다. 하지만 침입 탐지 알고리즘에는 데이터 의존성이 존재한다. 데이터 의존성을 해결하지 않으면 각 프레임 처리 결과 값들 간에 충돌이 발생하여 물체 탐지 과정에서 물체의 Fragmentation 현상을 제대로 해결하지 못하거나 침입 상황을 발생시킨 물체를 새로운 물체로 인식하여 추적이 중지되는 문제가 발생한다.

나. 프레임 입력에 따른 데이터 의존성

프레임 추출 후 각 침입 탐지 쓰레드로 프레임 전달이 이루어져야하기 때문에 프레임 추출 쓰레드와 침입 탐지 쓰레드 사이에 데이터 의존성이 발생한다. 본 논문에서는 다중 쓰레드로 침입 탐지 처리를 구현하는 과정에서 프레임 입력 여부에 따른 각 쓰레드의 동기화로 데이터 의존성을 해결했다. 해결방법에 대한 자세

한 내용은 3절에서 설명한다.

다. 각 프레임 처리 간에 존재하는 데이터 의존성

침입 탐지 처리 과정 내의 데이터 의존성은 하나의 프레임에 대한 처리 결과가 다음 프레임의 처리에 반영되어야 하는 부분에서 발생한다. 이 부분의 데이터 의존성을 해결하기 위해서는 프레임의 처리 순서를 제어하는 것이 필요하다. 배경 생성 모델로 사용하는 가우시안 혼합 모델은 시간에 따라 픽셀 값의 가중치를 다르게 누적시키기 때문에 배경 업데이트 과정에서 각 프레임들이 순차적으로 적용되어야 한다. 전경 추출의 결과로 생성된 이진 영상에서 물체를 파악하기 위한 과정은 이전 프레임의 물체 탐지 결과가 그 다음 프레임에 반영되어야 하므로 이 부분에서도 처리 순서의 제어가 필요하다. 위 두 가지 경우를 제외하고도 알고리즘 후반부에서 물체 탐지 및 침입 판단에 대한 데이터를 참조하는 부분에 데이터 의존성이 존재한다. 위와 같은 데이터 의존성들은 프레임 입력에 따른 데이터 의존성과 마찬가지로 구현 과정에서 쓰레드의 동기화와 처리 순서의 제어로 해결했으며 자세한 방법은 3절에서 설명한다.

3. 침입 탐지 알고리즘의 다중 쓰레드 구현

가. 제안하는 다중 쓰레드 구성

병렬화한 침입 탐지 처리를 다중 쓰레드로 구현하면서 Load Balancing을 위해 CPU-bound Process인 프레임 단위의 침입 탐지 처리들을 각각 쓰레드로 구성하고 I/O-bound Process에 해당하는 프레임 추출 과정은 별도의 쓰레드로 동작하도록 설계했다. 데이터 의존성은 쓰레드의 상태 변화와 각 침입 탐지 쓰레드의 처리 순서를 제어함으로써 해결했다.

나. 제안하는 데이터 의존성의 해결 방법 및 구현

제안하는 병렬화한 침입 탐지 처리 구조는 데이터 의존성이 있는 입출력 작업들을 파악하여 해당 과정을 Critical Section으로 설정했다. 그림 4에서 음영으로 표시된 부분이 데이터 의존성이 있는 부분이며 점선으로 표시한 부분은 의존성이 있는 데이터의 입출력 관계를 나타낸다. 프레임 추출 쓰레드는 각 침입 탐지 쓰레드로 프레임을 넘겨주는 순서를 그림 4에 표현된 것처럼 구성된 침입 탐지 쓰레드의 개수 N 과 Iteration 횟수

I , 침입 탐지 쓰레드의 번호에 따라 일정하게 유지한다. 그리고 침입 탐지 쓰레드가 처리하고 있는 프레임 번호가 낮을수록 데이터 입출력에 대한 높은 우선순위를 갖는다. 즉, 먼저 들어온 프레임부터 먼저 처리하는 구조를 가진다. 이는 물체 탐지 및 추적, 침입 탐지를 하는 과정이 물체의 순차적인 이동을 감지해야하기 때문이다. 위에서 Critical Section으로 지정된 처리 과정은 각 침입 탐지 쓰레드의 동기화와 처리 순서의 제어를 통해 데이터 의존성을 해결한다. 그림 5는 침입 탐지 처리 과정에서 Critical Section의 동기화 상황에 따른 침입 탐지 쓰레드의 동작 흐름을 나타낸 것이다. 침입 탐지 처리를 수행하는 쓰레드들은 쓰레드 생성시의 오버헤드를 줄이기 위해 쓰레드 풀 형태로 구성된다. 쓰레드 풀에서 Ready 상태에 있는 쓰레드는 프레임 추출 쓰레드로부터 프레임을 전달 받으면 Wait 상태로 전환되어 진행할 부분에 Lock이 걸려있는지 확인한다. 이 때 Lock이 걸려있지 않으면 침입 탐지 처리를 진행하고 진행 도중에 Lock 걸려있는 부분이 있으면 다시 Wait 상태로 복귀하여 대기한다. 하나의 프레임에 대한 처리를 모두 마치면 다시 Ready 상태로 돌아가서 다음 프레임을 전달 받을 때까지 기다린다. 침입 탐지를 최적으로 수행 가능한 쓰레드의 수 N 은 CPU에서 구동 가능한 논리적

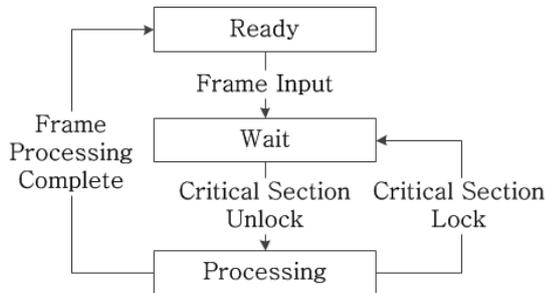


그림 5. 제안하는 시스템에서 침입 탐지 쓰레드의 상태도.

Fig. 5. State diagram of the intrusion detection thread.

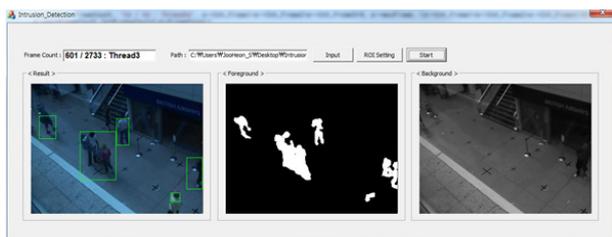


그림 6. 제안하는 다중 쓰레드 침입 탐지 프로그램.

Fig. 6. Demonstration of the proposed intrusion detection program.

쓰레드의 수에 의존적이기 때문에 실험환경에 따라 유동적이다. 다중 쓰레드로 구현한 침입 탐지 프로그램은 그림 6과 같다.

IV. 실험

본 논문에서 제안하는 가속화 방법의 성능을 검증하기 위해 실험에 사용한 입력 영상은 IEEE의 PETS^[22]에서 제공하는 Data Set이다. 실험에 사용한 영상의 물체 상황과 복잡도는 표 2에 정리했으며 각 Data Set의 해상도는 QVGA (320×240) 이다. 속도 측면에서의 성능 비교를 위해 침입 탐지 수행 시간을 측정하고 단위 시간당 침입 탐지 처리를 완료하는 프레임 수 (Frame Rate)를 구했다. 제안하는 가속화 방법은 기존의 침입 탐지 알고리즘을 변경하지 않고 처리 과정을 병렬화한 것이므로 기존 침입 탐지^[2]와 동일한 검출 정확도^[23]를 갖는다. 실험을 진행한 환경은 표 3에 정리했다. 각 실험환경에서 침입 탐지를 수행하는 쓰레드 수 N 에 따른 성능 측정 결과는 표 4에 정리했다. 결과로 표기된 수치들은 동일한 영상에 대해 10번 수행한 결과의 평균값이다. 표 4의 Conventional은 단일 쓰레드로 구현한 침입 탐지 처리이고 Proposed는 다중 쓰레드로 구현한 침입 탐지 처리이다. 표 4에서 다중 쓰레드 적용 시 침입 탐지 쓰레드의 개수가 증가함에 따라 침입 탐지 처리 속도가 향상되는 것을 확인할 수 있다. 그러나 쓰레드의 개수가 많아짐에 따라 CPU에서 동작하는 쓰레드 전환 시에 발생하는 Context Switching 오버헤드와 데이터

표 2. 실험에 사용한 영상의 특성.

Table 2. The property of the data set used in experiment.

	물체 상황	복잡도
S0_RF_View_001	Regular Flow ^{a)}	낮음
S0_RF_View_003	Regular Flow ^{a)}	중간
S06_BAG_STEAL	Crowded Flow ^{b)}	높음

^{a)} Regular Flow : 비교적 일정한 속도, 간격의 물체 이동

^{b)} Crowded Flow : 혼잡한 물체 이동

표 3. 실험환경 I 과 실험환경 II.

Table 3. Experimental environment.

	실험환경 I	실험환경 II
CPU	Intel Pentium Dual Core, 2.3GHz	Intel i7 Quad Core, 3.3GHz
Logical Thread	4 개	8 개

표 4. 쓰레드 개수에 따른 침입 탐지 처리 속도.
Table 4. Intrusion detection speed.

		S0_RF_View_001		S0_RF_View_003		S06_BAG_STEAL	
		Exec.Time [sec] ^{a)}	Frame Rate [fps] ^{b)}	Exec.Time [sec] ^{a)}	Frame Rate [fps] ^{b)}	Exec.Time [sec] ^{a)}	Frame Rate [fps] ^{b)}
실험환경 I	Conventional	26.861	14.836	25.695	15.567	105.971	18.807
	Proposed (N=2)	16.955	23.592	17.948	22.286	87.171	22.863
	Proposed (N=3)	16.633	24.048	16.360	24.449	77.290	25.786
	Proposed (N=4)	12.868	31.085	12.580	31.796	74.025	26.923
실험환경 II	Conventional	17.953	22.280	17.062	23.444	83.697	23.812
	Proposed (N=2)	15.311	26.125	15.238	26.250	77.382	25.755
	Proposed (N=3)	12.181	32.838	12.231	32.703	61.938	32.177
	Proposed (N=4)	9.128	43.821	9.130	43.811	46.472	42.886
	Proposed (N=6)	6.157	64.967	6.143	65.114	31.575	63.119
	Proposed (N=8)	5.075	78.817	5.179	77.235	27.649	72.082

a) Execution Time [sec] = 침입 탐지 처리에 소요된 시간.

b) Frame Rate [frame per second (fps)] = 전체 프레임 수 / 침입 탐지 처리에 소요된 시간.

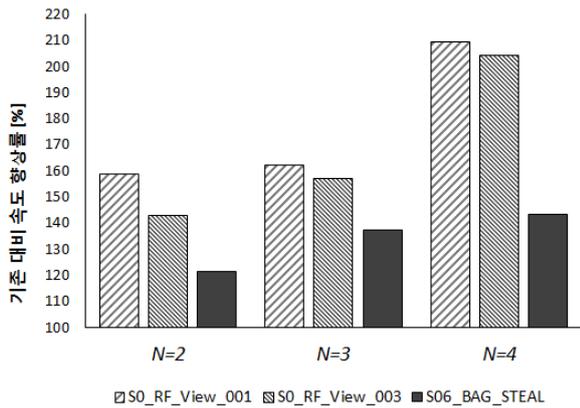


그림 7. 실험환경 I 에서 쓰레드 수 (N) 에 따른 기존 방식 대비 속도 향상률.

Fig. 7. Improvement ratio of the proposed intrusion detection speed according to the number of threads for the experimental environment I.

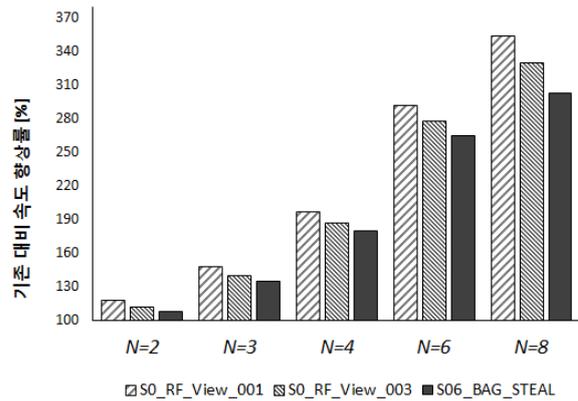


그림 8. 실험환경 II 에서 쓰레드 수 (N) 에 따른 기존 방식 대비 속도 향상률.

Fig. 8. Improvement ratio of the proposed intrusion detection speed according to the number of threads for the experimental environment II.

의존성이 존재하는 부분의 동기화 오버헤드 때문에 이상적인 성능 향상 수치에는 못 미치는 것을 확인할 수 있다. 그림 7과 8은 기존 단일 쓰레드 침입 탐지 대비 다중 쓰레드 침입 탐지의 속도 향상률을 그래프로 나타낸 것이다. 각 실험 환경에서 사용할 수 있는 쓰레드를 모두 활용한 경우 실험환경 I 에서는 다중 쓰레드로 구현된 침입 탐지가 평균 85.64%의 성능 향상을 보였고 실험환경 II 에서는 평균 228.64%의 성능 향상을 보였다.

V. 결론

지금까지 멀티 코어 프로세서 환경에서 병렬화를 이용한 침입 탐지 처리의 가속화를 제안했다. 본 논문에서는 기존 알고리즘을 변경하지 않고 침입 탐지 알고리즘의 전체 과정을 멀티 코어 프로세서 기반의 병렬 처리 구조로 설계하여 검출 정확도 성능의 저하 없이 가속화했다. 가속화 효과의 검증을 위해 실험한 결과 최대 353.76%까지 속도 향상이 이루어졌다. 병렬화 과정에서 발생하는 데이터 의존성은 구현 과정에서 쓰레드의 동기화와 처리 순서의 제어를 통해 해결했다. 본 논문이 제안한 방법은

알고리즘의 변경 없이도 고속의 침입 탐지 처리를 수행할 수 있다는 것이 장점이며 침입 탐지 알고리즘이 아닌 다른 영상 감시 알고리즘에도 적용이 가능하다.

REFERENCES

- [1] Tae-kyung Kim, Joon-ki Paik, "Video analysis and tracking for intelligent surveillance systems," *Journal of the Institute of Electronics Engineers of Korea on Signal Processing*, 39(2), pp. 55-65, Feb 2012.
- [2] Joo-heon Park, Youn-chul Shin, Jae-won Jeong, Myeong-jin Lee, "Detection and tracking of intruding objects based on spatial and temporal relationship of objects," in *Proc. IWIT Intl. Conf. ISA*, 2013.
- [3] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Intl. Conf. CVPR*, vol. 2, pp. 246-252, 1999.
- [4] Ying-Li Tian, *et al.*, "Robust and efficient foreground analysis for real time video surveillance," in *Proc. IEEE Intl. Conf. CVPR*, vol. 1, pp. 1182-1187, 2005.
- [5] Jeong Hwan Choi, Young Min Baek, Jin Hee Na, Jin Young Choi, "Effective moving object detection algorithm for surveillance systems," in *Proc. KIEE Conf. ICS*, pp. 457-458, Oct 2007.
- [6] Anil Kumar, Yaakov Bar-Shalom and Eliezer Oron, "Precision tracking based on segmentation with optimal layering for image sensors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 182-188, Feb 1995.
- [7] Perona, P., Malik, J., "Scale-space and edge detection using an isotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 629-639, Jul 1990.
- [8] Seok-Hwan Jang, In-haeng Kim, Hyoung-geun Song, Sang-ho Song, Tong-rae Cho, Ki-back Kim, Woong-soo Kim, Whoi-yul Kim, "Real-time multi-target tracking system," in *Proc. IEK Winter Conference 11(1)*, pp.499-502, Jan 1998.
- [9] J. Sochman, J. Matas, "WaldBoost - Learning for time constrained sequential detection," in *Proc. IEEE Intl. Conf. CVPR*, vol. 2, pp. 150-156, Jun 2005.
- [10] Herout, Adam, *et al.* "Real-time object detection on CUDA," *Journal of Real-Time Image Processing*, vol 6(3), pp. 159-170, Sep 2011.
- [11] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Intl. Conf. CVPR*, vol. 1, pp. 886-893, Jun 2005.
- [12] Q. Zhao, S. Brennan, and H. Tao, "Differential EMD tracking," in *Proc. IEEE Intl. Conf. ICCV*, pp. 1-8, Oct 2007.
- [13] Xie, Di, Lu Dang, and Ruofeng Tong, "Video based head detection and tracking surveillance system," in *Proc. IEEE Intl. Conf. FSKD*, pp. 2832-2836, May 2012.
- [14] G. L. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1045-1062, 1999.
- [15] G. L. Foresti, "A real-time system for video surveillance of unattended outdoor environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 697-704, 1998.
- [16] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst. Man, Cynern., Pt. C*, vol. 34(3), pp. 334-352, Aug 2004.
- [17] L. Maddalena, A. Petrosino, "A self-organization approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, pp. 1168-1177, Jul 2008.
- [18] D.-M. Tasi, S.-C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.*, vol. 18 pp. 158-167, Jan 2009.
- [19] A. J. Lipton, H. Fujiyoshi, and R.S. Patil, "Moving target classification and tracking from real-time video," in *Proc. IEEE Workshop on WACV*, pp. 8-14, Oct 1998.
- [20] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, pp. 42-77, 1994.
- [21] Walczyk Robert, Armitage Alistair and Binnie David, "Comparative study on connected component labeling algorithms for embedded video processing systems," in *Proc. Intl. Conf. IPCV*, 2010.
- [22] PETS, <http://pets2013.net>
- [23] H. Grabner, P. M. Roth and H. Bischof, "Is pedestrian detection really a hard task?" in *Proc. IEEE Intl. Workshop on PETS*, pp 1-8, Oct 2007.

저 자 소 개



이 길 범(학생회원)
2008년~현재 한국항공대학교
정보통신공학과
학사 과정.
<주관심분야 : 영상 신호 처리,
병렬 프로그래밍>



정 상 진(학생회원)
2007년~현재 한국항공대학교
정보통신공학과
학사 과정.
<주관심분야 : 영상 신호 처리,
마이크로 컨트롤러>



김 태 환(정회원)-교신저자
2005년 연세대학교 전기전자
공학과 학사 졸업.
2007년 한국과학기술원 전기 및
전자공학과 석사 졸업.
2010년 한국과학기술원 전기 및
전자공학과 박사 졸업.

2013년 현재 한국항공대학교 항공전자 및
정보통신공학부 교수.
<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC
설계>



이 명 진(정회원)
1994년 KAIST 전기및전자공학과
학사.
1996년 KAIST 전기및전자공학과
석사.
2001년 KAIST 전자전산학과
박사.

2001년~2004년 삼성전자(주) SYSTEM LSI
사업부 책임.

2004년~2007년 경성대학교 전기전자공학과
조교수.

2007년~현재 한국항공대학교 항공전자 및
정보통신공학부 부교수.

<주관심분야 : 영상부호화, 영상처리, 영상통신,
컴퓨터 비전, 영상 회로 및 시스템 설계>