

최 종 연 구 보 고 서

지능형 영상 감시 시스템 개발에 관한 연구

수탁기관 : 한양대학교 산학협력단

휴먼알에이치

최 종 연 구 보 고 서

지능형 영상 감시 시스템 개발에 관한 연구

수탁기관 : 한양대학교 산학협력단

휴먼알에이치

제 출 문

휴먼알에이치 귀하

본 보고서를 지능형 영상 감시 시스템 개발에 관한 연구
의 최종연구보고서로 제출합니다.

2008 년 12 월 30 일

수탁기관 : 한양대학교 산학협력단
수탁기관장 : 박재근 (인)
연구책임자 : 김회율
참여연구원 : 곽재호,
김병수,
조동찬,
김철문,
U. Habib

요 약 문

1. 제 목

지능형 영상 감시 시스템 개발에 관한 연구

2. 연구의 목적 및 중요성

1) 연구의 목적

본 연구는 지능형 영상 감시 시스템(Intelligent Video Surveillance System, IVSS) 개발을 목적으로 한다. 카메라에서 입력되는 실시간 영상을 분석하여 움직이는 물체를 자동으로 탐지/분류/추적하고, 움직임 정보를 통해 이벤트를 추출하여 보안 정책에 위배되는 이벤트가 발생했을 경우 관리자에게 통보하는 기능을 갖춘 인공 지능 감시 시스템 개발을 목적으로 한다.

2) 연구의 중요성

지난 2001 년 발생한 9.11 테러 사태는 미국을 중심으로 세계 각국이 개인 및 공공의 안전을 위하여 보다 강화된 보안 기술 개발에 많은 투자를 하게 되는 계기가 되었다. 이와 더불어 영상보안 솔루션의 수요는 2002 년부터 기업과 정부를 중심으로 증가하여 급속한 성장을 거듭하고 있으며, 이러한 추세는 향후에도 지속될 전망이다. 또한 영상보안 기술은 2006 년 미국 정부가 중점 투자할 4 대 기술분야 중의

하나였으며, 높은 시장 잠재력과 성장이 예측되는 분야이다. 국내에서도 최근 들어 일련의 문화재에 대한 화재 및 강력범죄의 증가에 따른 감시/예방 차원에서 영상보안 기술의 적용확대를 추진하고 있다. 이와 함께 영상보안은 CCTV 카메라, 디지털 저장장치, 영상분석 및 인식 기술 등의 발전과 시장 확대로 보다 진화된 기술의 개발을 요구하고 있다.

3. 연구의 내용 및 범위

- ✓ 기본 기능
 - 이동 객체 자동 감지
 - 이동 물체 추적
 - 물체 자동 분류
 - 이동 물체 방향 인식
- ✓ 확장 기능
 - 영역 침입 감시
 - 통행 방향 위반 감시
 - 배회 물체 감시
 - 무단 방치 물체 감시
 - 무단 파손 행위 감지
 - 무단 이동물체 감지
 - 군중 밀집 상태 감지
 - People Counting
 - 카메라 방해 감지

4. 연 구 결 과

- ✓ 노이즈(noise) 환경에 강인한 물체 감지 및 추적
- ✓ 다양한 이벤트 감지를 위한 영상 분석 알고리즘
- ✓ PTZ 카메라 연동
- ✓ 다양한 네트워크 카메라 지원
- ✓ 비디오/이벤트 데이터 관리 및 검색 기능
- ✓ 실시간 원격지 모니터링 시스템
- ✓ 다양한 형태의 이벤트 알람 출력

5. 활 용 분 야

영상보안의 진화는 지능형 영상감시, 생체인식, CCTV/DVR 영상보안 기술 등의 각기 다른 배경과 응용분야로 발전되어 왔다. 지능형 영상 감시 기술은 초창기 군사적 응용을 위하여 목표물 검출 및 추적 등 영상분석 기술 위주로 발전되어 왔으며, 영상분석에서 요구되는 엄청난 계산을 위한 하드웨어적 제약이나 비용보다는 영상분석의 성능을 높이는 데 많은 노력을 기울여 왔다. 생체인식 기술은 지문인식을 위주로 발전되어 왔으나 최근 들어 원거리 얼굴인식이나 홍채인식, 걸음걸이 인식 등 새로운 기술의 출현으로 영상보안에서 이들 기술을 접목하려는 시도가 계속되고 있는 분야이다. 그리고, 영상보안의 주류 시장을 형성하고 있는 CCTV/DVR 기반의 영상보안 기술은 산업체를 중심으로 보안장비 측면에서 지속적인 발전을 거듭하고 있는 분야이다.

전통적인 CCTV 영상보안 기술은 카메라를 통하여 획득된 영상을 전송망을 통하여 전달받아 모니터를 이용하여 보안상황을 사람이 직접 감시하

거나 DVR 등으로 저장한 후 사후에 저장된 영상을 검색하여 대응하는 다소 수동적인 물리적 보안 시스템을 의미한다. 그러나 이러한 영상보안 시스템은 급속한 CCTV 카메라의 보급에 따른 사람에 의한 직접감시 및 분석의 한계, 사후처리가 아닌 실시간 감시 시스템에 대한 요구 증대, IT 기술의 급속한 발전 등에 따른 이들 기술들이 통합된 지능형 영상보안 시스템으로 새로운 진화를 하고 있다.

지능형 영상 보안 기술의 응용 분야는 공항, 군사, 항만, 도로, 교량 등 주요 국가시설, 지하철, 버스, 빌딩, 경기장, 주차장, 카지노, 응급상황 감시에서 자동차 및 모바일 기기에 이르기까지 매우 다양한 분야로 그 응용 영역을 확대하고 있는 추세이다.

- 목 차 -

제 1 장 연구 개요	10
제 1 절 연구의 필요성 및 목표	10
1. 연구의 필요성	10
2. 국·내외 기술 동향.....	11
3. 연구의 목표	16
제 2 절 연구의 내용 및 범위	17
제 2 장 시스템 구성	18
제 1 절 시스템 구성도	18
제 2 절 시스템 흐름도	20
제 3 장 관련 연구	21
제 1 절 물체 분할	21
1. 차영상법	21
2. 배경영상의 추정	25
3. 전역 움직임 보상	33
제 2 절 물체 추적	38
1. Template Matching.....	38
2. Kalman Filtering.....	43
제 4 장 알고리즘 상세 설명	48
제 1 절 배경 모델 생성 및 갱신	48
제 2 절 전경 영역 분리	51
제 3 절 이동 객체 추적	53

제 4 절 금지 구역	55
제 5 절 도난/불법 방치	57
제 6 절 카메라 방해	59
제 5 장 시스템 상세 설명	61
제 1 절 카메라	61
1. USB 카메라	61
2. Axis 210	63
3. Axis 214 PTZ	64
제 2 절 서버(Server)	64
제 3 절 클라이언트(Client)	67
제 4 절 미디어 서버(Media Server)	71
제 6 장 향후 연구 과제	72
제 7 장 결 론	76

- 그 림 목 차 -

그림 1. 지능형 영상 감시 시스템 구성도	18
그림 2. 지능형 영상 감시 시스템 흐름도	20
그림 3. 차영상과 에지 정보를 활용하는 추적 시스템	23
그림 4. 움직이는 물체와 정지물체의 검출[18]	27
그림 5. 전경 영역 검출 영상	52
그림 6. 이동 객체 추적 흐름도	54
그림 7. 금지구역 침범 감시	56
그림 8. 도난/불법 방치	57
그림 9. 도난/불법 방치 감시 알고리즘	58
그림 10. 서버 구성도	65
그림 11. 서버 사용자 화면	66
그림 12. 클라이언트 구성도	68
그림 13. 클라이언트 사용자 화면	69
그림 14. 미디어 서버 사용자 화면	71

- 표 목 차 -

표 1. 연구 개발 내용 및 범위	17
표 2. 카메라 방해 를 위해 추출하는 특징량	60
표 3. 시스템 한계점	72

제 1 장 연구 개요

제 1 절 연구의 필요성 및 목표

1. 연구의 필요성

9.11 사태 이후, 세계 각국은 개인 및 공공의 안전을 위한 보안에 대한 중요성을 한층 더 실감하고 이에 대한 노력을 강화하고 있으며, 이러한 과정에서 지능형 영상보안 기술의 개발 및 활용에 대한 중요성이 날로 증대되고 있다.

아날로그 CCTV에서 디지털 저장장치, IP 기반기술과의 결합과 지능형 영상인식에 이르기까지 영상보안 기술은 지속적인 진화를 거듭하고 있다. 이에 따라 기존의 시설물과 출입자에 대한 수동적인 녹화 및 감시에서 실시간으로 상황을 인지하고 자율 대응할 수 있는 네트워크 기반의 지능형 시스템으로 발전하고 있다. 또한 공항, 군사, 항만, 도로, 교량 등 주요 국가시설, 지하철, 버스, 사무실, 카지노에서 산불감시에 이르기까지 매우 다양한 분야로 그 응용 영역을 확대하고 있는 추세에 있다.

그러나 기존의 감시 시스템은 그러한 추세를 반영하는데 그 한계가 있다. 즉, 기존의 CCTV 기반 아날로그 감시시스템은 사람이 무슨 일이 발생했는지 장시간 입력되는 비디오를 관찰하고 있어야 하는 문제점이 있다.

이러한 문제점을 해결하기 위해 실시간으로 입력되는 영상을 사람이 직접 보고 판단하지 않더라도 자동적으로 위급한 상황이 발생할 경우 경보를 알려주는 지능형 감시시스템의 개발이 필요하다.

2. 국·내외 기술 동향

□ H/W 기술 동향

영상보안의 주류시장을 형성하고 있는 CCTV/DVR 기술은 2000년도부터 급속한 성장을 거듭하고 있는 분야이다. 초창기 특정공간에 설치된 CCTV 카메라를 통하여 획득된 영상정보를 폐쇄적인 유선 전송을 이용하여 관제센터로 전송하고 관리자가 모니터를 통하여 직접 감시하던 영상보안 시스템은 IT 기술의 진보로 VCR 저장장치 기반의 아날로그 시대에서 DVR 기반의 영상압축 및 디지털 전송기술을 사용하는 디지털 시대로 발전하였으며, 최근에는 광-대역 통신망 및 개발 프로토콜을 사용하는 IP 기반의 네트워크와 영상자동분석 및 인식 기술이 접합된 지능형 영상보안 기술로 진화하고 있다.

영상보안 시스템의 기본구성은 영상촬영을 위한 CCTV 카메라, 영상을 볼 수 있는 디스플레이 장치, 카메라와 디스플레이 장치를 연결시켜주는 전송 망, 영상을 저장할 수 있는 저장매체로 요약할 수 있다. CCTV 카메라는 흑백 CCD 소자에서 메가 픽셀 급의 고해상도 디지털 IP 카메라로 발전하고 있으며, PTZ 제어장치에 탑재되어 관제센터에서 PTZ 제어기를 이용하여 원격으로 사람이 직접 디스플레이를 보면서 영상확대, 물체 또는 영역 추적 등의 기능을 수행할 수 있다. 영상 저장매체는 1997년 국내 업체에서 세계 최초로 상업화에 성공한 DVR이 대세를 이루고 있으며, 평균 16채널 이상의 다수 CCTV 카메라를 통하여 획득되는 영상을 압축하여 저장한다. 영상압축은 표준규격을 사용하며, MPEG-4 방식에서 높은 압축성과 유연성을 갖고 있는 H.264 규격으로 전환되고 있는 추세이다.

DVR은 보안장비의 특성상 제품의 신뢰성이 가장 중요한 요소로 365일 24시간 오류 없이 동작해야 한다. 또한 시간 별, 날짜 별, 카메라 별, 이벤트 별로 검색 기능 등을 제공하는 것이 일반적이다. 현재의 DVR은 PC 기반에서 stand-alone 방식을 거쳐 휴대전화기, GPS 등을 이용하는 모바일 DVR 등 서비스 결합 형 제품들이 출현하고 있는 추세이다. 그리고 전송망은 Ethernet 기반의 네트워크 전송 망으로 진화하고 있다.

□ S/W 기술 동향

비디오 영상 감시시스템에 실시간으로 입력되는 비디오 영상시퀀스 속에서 물체를 검출하고 추적하는 기술은 여러 가지 기준에 따라 분류될 수 있다. 일반적인 물체 추적시스템은 다음과 같은 기본적인 구성을 가지고 있다.

- ① 카메라로부터 영상을 획득하고 필요한 전처리를 하는 부분
- ② 영상 속에서 물체의 존재를 검출하고 물체가 존재하는 영역을 배경으로부터 분할 한다. 이때 분할된 영역은 대상이 되는 물체가 존재하는 영역과 반드시 일치하지는 않는다.
- ③ 이전 영상 프레임까지 얻은 정보로부터 추적중인 각 물체의 상태 정보에 근거하여 ②에서 분할 된 추정 영역에 속한 후보 영역과의 일치(match)를 판정하고 현재 프레임에서 각 물체의 상태를 갱신(update)하여 계속적으로 추적한다.

여기서 이동체 영역의 분할 방법으로서 다음과 같은 차영상 기법들이 주로 사용되고 있다. 이 기법들은 고정된 카메라를 사용하는 경우에 적용할 수 있다.

차영상에 근거한 분할 기법들

- 인접영상끼리의 차영상법(difference method)[53]
- N-시간만큼 떨어진 이전영상과의 차영상법[50]
- 배경과의 차영상법(background subtraction method)
 - 1) 고정된 배경을 사용하는 방법
 - 2) 배경 추정(background estimation)
 - 이동물체만을 고려
 - 이동물체, 정지물체, 정지그림자 고려[18]
 - Kalman filter 사용[44][16]
 - Temporal Median filter 사용[45]
 - 배경영상의 각 픽셀을 Gaussian으로 모델[48]
 - 배경영상의 각 픽셀을 Gaussian Mixture로 모델[44]

인접영상끼리의 차영상법[53]은 구현이 가장 간단하고 배경의 완만한 변화에 대해서 강인하므로 많이 사용된다. 하지만, 물체의 움직임이 없는 경우에는 물체의 대상영역을 분할해 낼 수 없는 문제점이 있다. 물체의 작은 움직임을 잡아내기 위하여 차영상법을 바로 인접한 영상 대신 N단계 이전의 영상프레임과 차영상을 적응적으로 이용하는 기법[50]을 사용하기도 한다.

만약, 배경영상을 알고 있다면 현재 영상과 배경 영상과의 차영상을 통하여 물체의 영상 영역을 쉽게 분할해 낼 수 있다. 추적대상이 되는 물체가 없는 경우의 영상을 한번 찍어 두었다가 계속해서 사용하는 방법은 구현이 쉬우나 시간, 계절, 조명 조건에 따라 배경영역의 조도가 크게 바뀌므로 실제로 사용하기 어렵다. 따라서, 적응적으로 배경을 추정하는 방법이 많이 연구되어 왔으며 이동물체만을 고려하는 일반적

인 방법과 이동물체, 정지물체, 그림자 등을 고려하는 방법, Kalman filter를 사용하는 방법, Median filter를 사용하는 방법, Gaussian 분포를 갖는 영상을 이용하는 방법 등이 알려져 있다. 그러나, 위에서 열거한 차영상 방법은 고정카메라가 아닌 움직이는 카메라를 사용하는 경우에는 그대로 적용할 수 없다. 왜냐하면, 카메라의 움직임에 의하여 고정된 배경영역이 움직이는 물체처럼 차영상 된 결과 영상에 그대로 나타나 버리기 때문이다.

이러한 문제점을 해결하기 위하여 카메라가 움직이는 동안에는 차영상을 계산하지 않는 소극적 대처법과, 움직이는 카메라에 의한 전역움직임을 적극적으로 보상하는 전역움직임 보상기법을 사용한다. 전자는 구현이 단순하지만 카메라가 움직이는 동안 물체의 추적이 불가능하므로 얻을 수 있는 성능에 한계가 존재하며, 후자의 방법은 복잡한 계산이 필요하지만 많이 사용된다.

한편 현재 추적 대상인 물체와 예측을 통해 얻은 후보 물체가 존재하는 영역 내에서 후보물체까지의 거리를 측정하여 이 거리가 가장 작은 물체를 연속적으로 추적하는 방법에는 어떤 형태든 차영상에 의존하는 방법을 전혀 사용하지 않으므로 근본적으로 배경의 움직임을 보상하려는 시도가 불필요하다. 이 경우는, 차영상 동작을 통해 후보 물체가 존재하는 영역을 상대적으로 쉽게 얻을 수 있는 차영상 방법들에 비하여 탐색하여야 하는 후보영역을 줄일 수 있는 다른 추적 기법이 사용된다.

능동카메라를 위한 대처 법

- 고정카메라 기법을 시간 간격을 두어서 제어하는 방법
- 전역 움직임 벡터의 추정 및 보상법

1) 전 서브블록 영역에서의 Template Matching 이용[14]

2) 일부 서브블록의 모서리/특징 정보 이용[50]

- 배경의 움직임에 상관없이 목표 물체와 거리를 최소화하도록 추적

1) Template matching[46]

2) Mean-shift[3][6]

3) Condensation[8]

고정된 카메라를 사용하는 경우에는 바로 배경으로부터 분할된 영역을 얻을 수 있으며, 움직이는 카메라의 경우 먼저 전역 움직임 보상을 수행한 후 배경으로부터 분할된 영역을 얻을 수 있다. 이전 영상 프레임까지 얻은 정보로부터 추적중인 각 물체의 상태 정보에 근거하여 분할된 추정영역에 속한 후보 영역과의 일치 정도를 판정하고 현재 프레임에서의 각 물체의 상태를 갱신하여 계속적으로 추적한다. 다음과 같은 물체 추적 기법들이 있다.

추적(Tracking) 기법

- Template Matching에 기반한 방법
 - 1) BMA(block matching algorithm)[46]
 - 2) Correlation 기반[46]
 - 3) Color 기반[1]
- Mean-shift 기반[3]
- Kalman Filtering[13]
- Condensation 알고리즘[8]

3. 연구의 목표

- 노이즈(noise) 환경에 강인한 물체 감지 및 추적 시스템 개발
- 다양한 이벤트 감지를 위한 영상 분석 알고리즘 개발
- PTZ 카메라 연동 모듈 개발
- 다양한 CCD 카메라 지원 모듈 개발
- 비디오/이벤트 데이터 관리 기능 개발
- 실시간 원격지 모니터링 기능 개발
- 다양한 형태의 이벤트 경보 출력 기능 개발

제 2 절 연구의 내용 및 범위

본 연구의 연구 개발 내용 및 범위는 표 1 과 같다.

표 1. 연구 개발 내용 및 범위

구분 연차	연구개발 내용	연구개발 범위
1차 년도	<ul style="list-style-type: none"> ■ 이동 물체 감지 ■ 이동 물체 추적 ■ 물체 분류 ■ 금지 구역 감시 ■ 도난/불법방치 감시 ■ 카메라 방해 ■ People Counting ■ 군중 밀집도 측정 	알고리즘 알고리즘 알고리즘 알고리즘 알고리즘 알고리즘 알고리즘 알고리즘
	<ul style="list-style-type: none"> ■ CCD 카메라 연동 ■ PTZ 카메라 연동 ■ 모니터링용 Client 	연동모듈 연동모듈 연동모듈
	<ul style="list-style-type: none"> ■ 감시용 Server 개발 ■ 실시간 영상 전송 모듈 ■ 이벤트 기반 비디오 저장(H.264) ■ 전송된 데이터 저장 및 검색 	시스템개발 시스템개발 시스템개발 시스템개발

제 2 장 시스템 구성

제 1 절 시스템 구성도

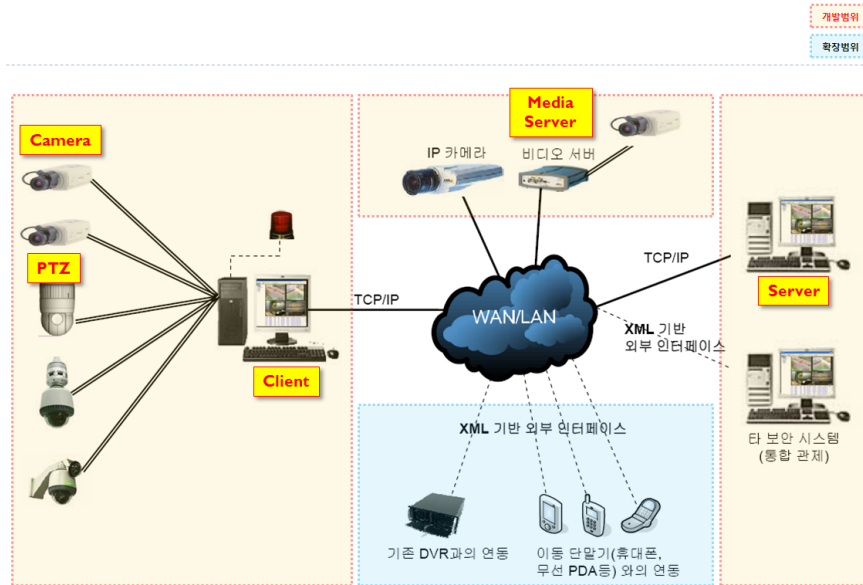


그림 1. 지능형 영상 감시 시스템 구성도

그림 1은 지능형 영상 감시 시스템의 전체 시스템 구성도를 나타낸 것이다. 지능형 영상 감시 시스템은 크게 세 부분으로 구성된다.

- 영상 입력 부 : CCD Camera, PTZ, 영상분석 클라이언트
- 서버 : 영상 입력 단으로부터 들어오는 영상을 실시간으로 모니터링
- 미디어 서버 : 이벤트 발생 구간의 영상을 실시간으로 저장

영상 입력 부는 영상을 실시간으로 획득하는 CCD 카메라, 입력된 영상을 실시간으로 분석하는 영상 분석 클라이언트, 그리고 PTZ 카메라로 구성된다. 실시간으로 입력되는 영상에 대해 영상 분석 기능을 수행하기 위해 CCD 카메라는 영상 분석 클라이언트 Workstation과 네트워크 기반으로 연결되어 있다. 보다 효율적인 감시를 위해서 PTZ 카메라도 함께 구성이 되어 있어, CCD 카메라가 감시하는 영역을 확대하여 보다 명확하게 화면을 모니터링 할 수 있다.

서버는 다중 클라이언트가 접속되는 중앙 통제실과 같은 역할을 한다. 여러 대의 클라이언트가 서버에 접속하고 중앙 서버는 각 클라이언트에서 보내오는 영상을 실시간으로 모니터링 할 수 있다. CCD 카메라의 영상 뿐만 아니라 PTZ 카메라의 영상도 함께 전송되어 효율적인 모니터링이 가능하다.

미디어 서버는 각 클라이언트에서 보안 정책에 위배되는 이벤트가 발생했을 경우 해당 구간의 영상을 H.264로 인코딩 하여 실시간으로 저장하는 역할을 담당한다. 이렇게 저장된 영상을 중앙 서버에서 검색 기능을 통해서 검색할 수 있고 이를 사후 대처에 활용할 수 있다. 그림 1에서 빨간 사각형 테두리 부분은 기본적인 감시 시스템의 시스템 구성을 보인 것이고 파란색 테두리 부분은 확장 가능한 보안 시스템 구성을 보인 것이다. 보안정책에 위배되는 이벤트가 발생했을 때 해당 화면을 사용자의 모바일 폰이나 다른 Display 기기로 사용자에게 실시간으로 전송된다면 보다 효율적인 보안이 가능할 것이다. 이러한 보안적 접근법은 현대 보안 솔루션에서 많이 시도되고 있다.

제 2 절 시스템 흐름도

그림 2는 지능형 영상 감시 시스템의 시스템 흐름도를 나타낸 것이다.

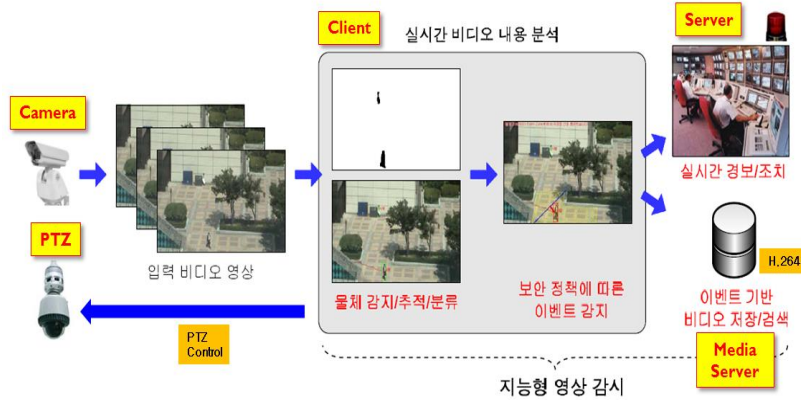


그림 2. 지능형 영상 감시 시스템 흐름도

CCD 카메라로부터 실시간으로 입력되는 영상은 카메라와 연결된 영상 분석 클라이언트 시스템으로 전송된다. 클라이언트의 실시간 영상 분석 프로그램은 실시간으로 입력되는 영상을 기반으로 영상을 분석한다. 즉, 입력되는 영상에서 움직이는 물체를 감지하고 감지된 물체를 추적 및 분류한다. 움직이는 물체의 움직임 정보를 기반으로 보안 정책에 따른 이벤트를 감지하고 발생한 이벤트를 기반으로 보안 정책에 위반되는 이벤트에 대해 관리자에게 위험성을 통보한다. 이때 이상한 행동을 보이는 물체에 대해서는 보다 명확한 모니터링을 위해 PTZ 카메라를 움직여 해당 물체를 확대 모니터링 한다. 보안 정책에 위배되는 이벤트가 감지되었을 경우 해당 영상은 미디어 서버에 실시간 저장된다. 저장된 비디오는 중앙 서버의 검색 기능을 통해 검색할 수 있고 사후 처리에 활용된다.

제 3 장 관련 연구

제 1 절 물체 분할

1. 차영상법

1.1. 연속 영상간의 차영상을 이용한 이동체 영역 결정

무인 감시기능을 수행하는 가장 기본적인 무단 침입자 추적 시스템이 [53]에 제시되었다. 기본적인 알고리즘의 동작은 다음과 같다. 먼저 CCD 카메라로부터 포착되는 영상 프레임 사이에 시간미분을 수행하여 움직임 에너지를 추출한다. 이 움직임 에너지를 구하는 과정은 다음의 식(1)에 의해 차영상을 구하는 것으로 대신할 수 있다.

$$\frac{dI(x, y, t)}{dt} \cong I(x, y, t) - I(x, y, t - \Delta t) \quad (1)$$

여기서 $I(x, y, t)$ 은 시간 t 에 입력된 영상 F 의 (x, y) 에 위치한 픽셀의 명암 값을 나타내며, 이때 픽셀의 위치를 나타내는 (x, y) 는 $x, y \in F$ 를 만족한다. 그리고 Δt 는 연속된 두 영상의 시간 간격이다.

식(1)과 구한 차영상에 절대값을 취한 후, 적절한 임계값으로 이진화 한다. 이때 차영상에는 시간 t 의 영상과 시간 $(t-1)$ 의 영상에서 서로 공통 부분인 배경의 영상이 완전히 제거되지만, 움직이는 사람의 모습은 시간 t 에서의 영상과 시간 $(t-1)$ 의 영상이 이중으로 겹쳐져서 나타나게 된다.

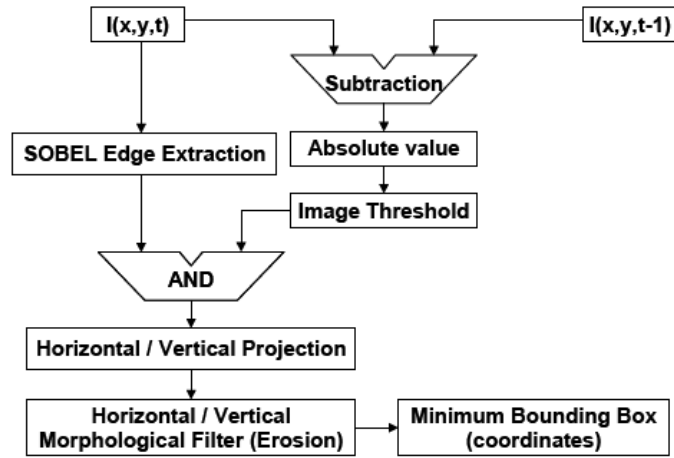
다음으로 차영상의 결과로부터 이동체의 존재 및 움직임 여부를 판

단하기 위하여 차영상에 존재하는 움직임 에너지 성분의 크기를 임계값과 비교하여 이동체를 배경으로부터 분할해 내고 이동체 영역을 표시하는 MBB(Minimum Bounding Box)를 결정한다.

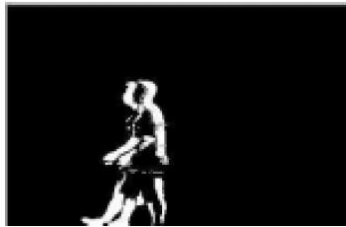
다음으로 이동체로 추정된 물체의 중심점을 계산하고 블록정합 과정을 수행하여 이동체의 정확한 중심위치를 계산해 낸다.

과거에 측정된 이동체 영역의 중심좌표로부터 다음 단계의 이동체 영역의 중심좌표가 영상의 가운데 위치하도록 Pan/Tilt 드라이버를 구동시킨다.

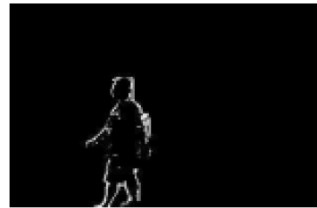
간단한 차영상에 기초한 이동물체 영역 추출 방법은 과거 두 프레임 사이의 이동체 동시에 표현되는 단점을 가진다. [5][64]에서는 이를 개선하여 그림 3과 같이 현재영상의 에지(edge)영상 추출결과와 차영상을 논리 AND 연산을 통하여 좀더 정확한 이동물체의 위치를 검출할 수 있음을 보였다. 그림 3 (b)에서 보인 일정 방향으로 걷고 있는 사람의 이진 차영상을 현재 프레임의 영상으로부터 추출한 에지 영상과 AND 논리연산을 수행하여 그림 3 (c)의 움직임 에지 영상을 얻는다. 이 방법은 단순히 차영상만을 사용하여 움직임 물체 영역을 예측하는 방법에 비하여 훨씬 더 정확하게 물체의 현재 이동위치를 확인 할 수 있음을 알 수 있다.



(a). 알고리즘



(b) 이진 차영상 추출 결과



(C) 움직임 에지 영상 추출결과

그림 3. 차영상과 에지 정보를 활용하는 추적 시스템

1.2. 속도 적응적인 차영상법

연속 영상에 나타난 이동 물체의 속도에 따라 정확한 이동 물체 영역의 검출이 불가능한 경우가 있다. 일정한 시간 간격의 연속 영상의 차영상을 얻으므로 이동 물체가 너무 빠르면 물체의 영역뿐만 아니라 배경 영역도 검출될 수 있으며, 반면 이동 물체가 너무 느리면 물체의

일부분만 검출된다. 따라서 이동물체의 속도에 적응적으로 적절한 이전 영상을 선택할 수 있는 방법이 요구된다.

일반적인 차영상법에서는 매 프레임마다 이전 영상과의 차영상을 구하여 움직임 에너지를 구한다. 움직이는 물체의 속력이 낮으면 차영상에 나타나는 움직임 에너지가 너무 작아지는 문제점이 있다. 반면, [50]에서 제시된 이동물체 속도에 적응적인 차영상법은 물체속도가 빠를 때는 바로 직전의 영상프레임과의 차영상을 구하지만 물체의 속도가 느린 경우는 멀리 떨어진 영상과의 차영상을 구해 움직임 에너지가 큰 값이 되도록 하였다.

1.3. 배경영상과 현재 영상간의 차영상을 이용한 이동체 영역 결정

이전 영상과의 차 연산에 의해 영상 내에 존재하는 움직임 에너지를 추출하여 이동물체를 검출하는 방법은 이동 중 일시 정지한 사람이 두 팔을 아래위로 흔드는 경우의 예와 같이 팔, 다리의 움직임이 두 개 이상의 서로 독립적인 물체로 검출될 수 있는 단점이 발생할 수 있다.

한편, 이러한 문제점은 만약 감시대상이 되는 지역의 배경영상을 알고 있다면 배경영상과 현재 영상과의 차영상 결과로부터 쉽게 해결될 수 있다. 차영상에는 배경영상이 아닌 모든 물체의 영상이 나타나게 되므로 움직이는 물체는 물론이고 배경이 아닌 정지된 어떤 물체라도 검출이 된다. 따라서 정지한 사람이 팔·다리만을 움직이는 경우에 다수의 물체로 검출되는 오류 없이 단일 물체로 검출할 수 있게 된다.

하지만 배경영역의 픽셀 값과 물체의 픽셀 값의 차이가 거의 없으면 배경 영상과의 차 연산 방법으로는 이동물체를 거의 검출할 수 없

거나 충분한 움직임 에너지를 얻을 수 없어 후에 그 물체의 추적이 불가능하게 되는 단점이 존재한다.

배경 영상으로는 검출, 추적 대상이 되는 물체가 존재하지 않는 경우에 미리 획득해 둔 영상을 사용하는 방법과 배경 외 물체가 존재하는 경우라도 이를 적극적으로 추정해 내어 사용하는 방법 [18][29][48]이 있다. 전자의 경우 배경영상의 획득이 용이하고 비용이 적게 드는 장점을 가지지만 시간의 경과와 계절의 변화, 기후의 변화에 따른 콘트라스트(contrast)의 변화 등을 포함한 배경 영상의 변화를 고려해 줄 수가 없으므로 현재 영상과의 차영상을 구할 때 존재하지 않은 물체가 존재하는 것으로 잘못 인식될 수가 있다. 이러한 단점은 후자의 배경 영상 추정방법으로 극복할 수 있다.

2. 배경영상의 추정

2.1. IIR 필터에 의한 배경 추정

배경 영상은 초기에 이동물체가 존재하지 않는 경우 입력되는 현재 영상으로 추출한다. 이후의 배경영상은 식(2)와 같이 IIR 필터를 사용하여 재귀적으로 매 프레임 연산 시 픽셀단위로 갱신(update)한다 [18].

$$\begin{aligned} B_{t+1}(x, y) = \\ S_t(x, y) \times (1 - \exp(-1/T)) + B_t(x, y) \times \exp(-1/T) \end{aligned} \quad (2)$$

여기서 $B_{t+1}(x, y)$ 은 다음 프레임 연산 시 사용될 배경 영상의 추출 결과이고, $B_t(x, y)$ 은 현재 프레임의 배경영상이다. $S_t(x, y)$ 는 검출

된 이동물체의 영역을 현재 프레임의 배경영상 $B_t(x, y)$ 의 픽셀 값으로 대체한 현재 프레임의 영상이다. [18]에서는, 배경 영상의 추출과정에 있어, 이동물체의 영역과 더불어 정지물체의 영역을 검출하여 이 두 영역에 해당하는 픽셀들을 배경 영상 $B_t(x, y)$ 의 픽셀 값으로 대체함으로써 얻은 현재 프레임의 영상을 $S_t(x, y)$ 로 사용하였다. 이에 따라, 이동물체와 정지한 물체가 차지하고 있는 영역을 제외한 나머지 부분만을 다음 프레임 연산에 사용될 배경 영상으로 반영시키게 된다.

안정적인 배경 영상의 추출 및 업데이트는 이동물체 및 정지물체 영역의 정확한 추출에 달려있다. [18]에서는 이동물체, 정지물체, 그리고 조명변화에 따른 그림자 등의 변화를 구분하는 방법이 제시되었다. 먼저, 현재 영상과 직전 영상의 차영상(time difference image)을 구하면 이 차영상에는 연속된 영상간의 차영상이므로 움직이는 물체의 영상이 나타난다. 다음으로 추정된 배경 영상과 현재 영상의 차영상(background subtraction image)을 구한다. 이때 정지한 물체의 영역 영상(stationary area image)을 배경 영상과의 차영상에는 나타나지만 연속된 영상간의 차영상에는 나타나지 않는 부분을 구하여 얻을 수 있으며 이동물체 영역의 영상은 두 차영상의 공통인 부분을 취하여 얻을 수 있다.

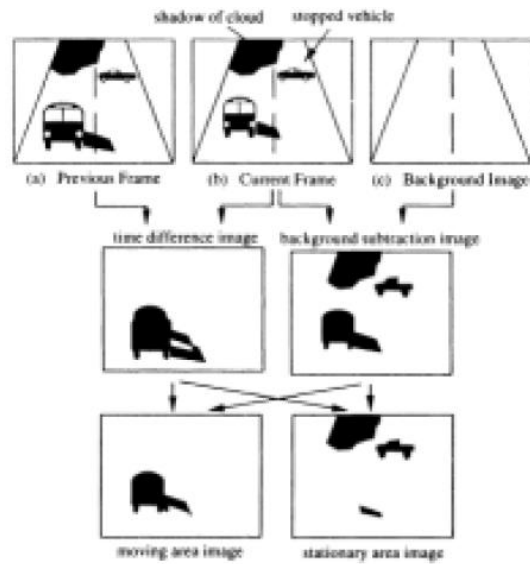


그림 4. 움직이는 물체와 정지물체의 검출[18]

정지물체 영역 영상의 각 영역에 대해서는 픽셀(pixel)의 평균과 표준편차를 계산하고 구름이나 조명의 변화에 의한 일시적인 그림자 등의 부분 등의 부분은 표준편차가 적은 반면 실제 정지 물체(object)인 경우는 표준편차가 큰 값을 보인다는 사실을 이용하여 정지물체인지 그 이외의 정지 영역인지를 분간해 낸다.

2.2. Kalman Filter를 사용한 배경 추정

[16]에서는 백그라운드 이미지 시퀀스를 각 픽셀 단위의 신호처리 시스템으로 기술하고 조명의 변화를 백그라운드 영상에 속하는 경우에는 빠르게 적응하도록 하고 움직이는 물체 영역에 속하는 경우에는 느리게 적응하도록 Kalman filter를 사용하여 제어하였다. 각 픽셀에

대해 사용된 백그라운드 운동은 다음과 같다. 시간 t_i 에서의 픽셀의 Intensity를 $s(x, y, t_i)$ 라하고 시스템 상태는 추정된 $\hat{s}(x, y, t_i)$ 와 1차 미분 $\dot{\hat{s}}(x, y, t_i)$ 로 나타내어 추정 식은

$$\begin{bmatrix} \hat{s}(x, y, t_i) \\ \dot{\hat{s}}(x, y, t_i) \end{bmatrix} = \begin{bmatrix} \tilde{s}(x, y, t_i) \\ \tilde{\dot{s}}(x, y, t_i) \end{bmatrix} + \mathbf{K}(x, y, t_i) \left(s(x, y, t_i) - \mathbf{H}(x, y, t_i) \begin{bmatrix} \tilde{s}(x, y, t_i) \\ \tilde{\dot{s}}(x, y, t_i) \end{bmatrix} \right) \quad (3)$$

Prediction 식은

$$\begin{bmatrix} \tilde{s}(x, y, t_i) \\ \tilde{\dot{s}}(x, y, t_i) \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} \hat{s}(x, y, t_{i-1}) \\ \dot{\hat{s}}(x, y, t_{i-1}) \end{bmatrix} \quad (4)$$

이며, 여기서 시스템 Matrix는 상수로

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} \\ 0 & a_{2,2} \end{bmatrix}, \quad a_{1,2} = a_{2,2} = 0.7 \quad (5)$$

이다.

Measurement matrix는 $\mathbf{H} = [1 \quad 0]$ 가 사용되었다. Kalman Gain

$$K(x, y, t_i) = \begin{bmatrix} k_1(x, y, t_i) \\ k_2(x, y, t_i) \end{bmatrix} \quad (6)$$

은 $k_1(x, y, t_i) = k_2(x, y, t_i) = \alpha m(x, y, t_{i-1}) + \beta([1 - m(x, y, t_{i-1})])$ 이고, $m(x, y, t_{i-1})$ 은 (x, y) 위치의 픽셀이 t_{i-1} 시간에 배경인지 아닌지를 나타내는 Segmentation을 나타내는 Binary map으로 다음과 같이 주어진다.

$$m(x, y, t_{i-1}) = \begin{cases} 1 & \text{if } d(x, y, t_{i-1}) \geq th(x, y, t_{i-1}) \\ 0 & \text{else} \end{cases} \quad (7)$$

1은 전경(foreground)로의 선택을, 0은 배경(background)로의 선택을 나타낸다. 여기서 $d(x, y, t_{i-1}) = |s(x, y, t_{i-1}) - \hat{s}(x, y, t_{i-1})|$ 이고 임계값(threshold)는 상수가 사용되었다. [16]의 방법을 확장하여 [29]에는 사람을 배경으로부터 분할하는 방법이 제시되었고, [39] 등에서 수정된 방법이 자동차를 배경으로부터 분할하는데 사용되었다.

2.3. Temporal Mean Filter의 사용

[45]에서는

$$B_{t+1}(x) = \begin{cases} B_t(x) + 1, & \text{if } B_t(x) < I_t(x) \\ B_t(x) - 1, & \text{otherwise} \end{cases} \quad (8)$$

식 (8)에 의하여 배경 영상을 갱신한다. 일정한 시간이 지나면 배경 픽셀은 Temporal median 값을 가지게 된다. 갱신된 배경 영상과 입력된 영상 I_i 의 차영상을 구한 뒤 임계값으로 Thresholding 하여 움직이는 물체의 Blob 영역을 나타내는 마스크 M_i 를 얻는다. 모션 해석을 통해 얻은 정보를 통하여 배경 영상의 갱신을 On/Off 하도록 제어한다.

2.4. 배경영상을 Gaussian mean 영상과 Standard deviation으로 모델

[48]에서는 배경 영상을 하나의 Gaussian mean 영상과 Standard deviation 영상으로 표현하였다. 먼저, 다음과 같은 가중치 누적 (weighted accumulation)을 생각한다.

$$\hat{x}_{i+1} = \phi(\hat{x}_i, x_i, \tau) = x_i(1 - e^{-\frac{1}{\tau}}) + \hat{x}_i e^{-\frac{1}{\tau}} \quad (9)$$

여기서 τ 는 가중치 누적을 하는 시간의 길이를 나타내는 상수이다. 누적 연산이 수행될 Subset $R \subset I$ 이 주어져 있을 때, 이러한 누적 연산을 영상 프레임의 Subset R 에 속한 각 픽셀에 대하여 적용하도록 함수를 다음과 같이 확장하여 정의한다.

$$\forall (x, y) \in I$$

$$\phi(\hat{f}_i, f_i, \tau, R)(x, y) = \begin{cases} \phi(\hat{f}_i(x, y), f_i(x, y), \tau), & \text{if } (x, y) \in R \\ \hat{f}_i(x, y), & \text{if } (x, y) \in \Lambda R \end{cases} \quad (10)$$

이 함수를 사용하여 Temporal change를 나타내는 영역을 다음과 같이 나타낼 수 있다.

$$D_i = \{(x, y) \in I \mid d_i(x, y) > \mathcal{G}_d\} \quad (11)$$

여기서 $d_i = \phi(d_{i-1}, \Delta f_i, \tau_d, I)$ 이고 $\Delta f_i \equiv f_i - f_{i-1}$ 로 정의되는 차영상을 나타낸다. 즉, d_i 는 Δf_i 영상을 τ_d 의 상수로 누적한 영상을 나타내고 D_i 는 d_i 값이 Threshold \mathcal{G}_d 보다 큰 영역을 나타낸다.

배경 영상은 Mean과 Standard deviation 영상을 각각 나타내는 $\{\mu_i, \sigma_i\}$ 로 나타내고 현재 영상 프레임 f_i 의 픽셀 값 중 평균보다 1 standard deviation 더 떨어진 값을 갖는 픽셀 영역을 배경 변화 영역 B_i 로 감지할 수 있다. 즉,

$$B_i = \{(x, y) \in I \mid |\mu_i(x, y) - f_i(x, y)| > \sigma_i(x, y)\} \quad (12)$$

현재의 배경 영상은 다음의 Iteration을 위하여 식(13)에 의해 갱신된다.

$$\begin{aligned} \mu_{i+1} &= \phi(\mu_i, f_i, \tau_b, A_i), \\ \nu_{i+1} &= \phi(\nu_i, f_i^2, \tau_b, A_i) \end{aligned} \quad (13)$$

여기서 $\sigma_{i+1} = \sqrt{\nu_{i+1} - \mu_{i+1}^2}$ 로부터 구해지고, A_i 는 최근 일정한 기간 (a 프레임) 동안 변화하지 않는 영역(즉, 배경영역)의 추정 영역이다. 즉, 시간적 변화를 나타내는 영역을 식(15)로 배제하여 얻는다.

$$A_i = \bigcap_{k=\max(0, i-a)}^i (\Delta D_k) \quad (14)$$

[48]에는 수정된 Kalman filter를 사용한 방법과의 비교결과를 제시하였으며 비슷한 계산시간이 요구되지만 정확성에서 12%이상 더 앞서는 것으로 나타났다.

2.5. 배경 영상 픽셀을 Gaussian Mixture로 모델

기타의 배경추정 방법으로 배경에 속한 각 픽셀의 칼라 분포를 Gaussian 분포를 갖는 다수의 분포의 결합체(mixture)로 모델링 하여 각 Gaussian 분포의 파라미터들 Weight와 Mean, Covariance를 갱신 하여 배경영상을 결정한다. Weight는 다음 식으로 갱신된다.

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha M_{k,t} \quad (15)$$

여기서 α 는 상수이며, $M_{k,t}$ 는 현재의 RGB 입력 값이 k -번째의 Gaussian과 일치하면(평균에 대하여 표준편차의 정해진 배수 내에 들어가면) 1이 되고 그렇지 않으면 0이 된다. Gaussian이 일치되면 해당 Gaussian의 평균과 Variance는 현재 RGB 입력 값을 이용하여 갱신된다. 만약 현재의 RGB 입력 값에 일치되는 Gaussian이 없

으면 가장 낮은 Weight를 갖는 Gaussian을 평균이 현재 입력 값이고 Variance가 큰 새로운 Gaussian으로 대체한다. 이렇게 관리되는 Gaussian들 중에서 Weight가 높은 순서로 정한 다음 신뢰도가 높은 처음 B개의 분포를 선택하여 Background 값으로 사용한다. [44]에는 수정된 Gaussian mixture 방법이 사용되었다.

3. 전역 움직임 보상

카메라가 고정되어 있지 않고 목표물을 따라 움직이는 경우 카메라의 움직임에 의한 전역 움직임(global motion)이 추적 물체의 움직임과 함께 나타나므로 이를 보상해 주어야 한다. 왜냐하면, 전역 움직임을 보상하지 않으면 움직이는 카메라에 대하여 차영상 기법을 사용할 수 없기 때문이다.

[14]에서는 전체의 영상을 8x8로 나눈 서브블록(sub-block)에 대하여 각 블록(block)의 Local 모션 Vector를 각각 구한다. Local motion vector는 BMA(block matching algorithm)에 의한 Template matching을 일정한 거리 내에 있는 주변 영역에 대하여 전 영역 탐색을 수행함으로써 구할 수 있다.

다음에 각 Local vector들의 Average operation을 취하여 Error vector를 효과적으로 제거할 수 있다. 그러나, 일정한 밝기를 갖는 단조로운 배경 영상의 영역에서 Local vector는 Global motion vector의 움직임을 반영하지 못하고 큰 Error를 보이게 된다. 이러한 문제점을 해결하기 위하여 [14]에서는 각 Local vector에 대하여 신뢰도(confidence measure)를 정의하고 이에 따라 전역 모션 Vector를 정한다. 신뢰도는 각 8x8 서브블록(sub-block)에 대하여 Cornerness와

distinctness를 측정함으로써 계산된다. 먼저, 8x8 block 내의 각 64개 pixel에 대하여 Plessy corner detection 알고리즘을 사용하여 $c_i = \text{Det}(A) - k\text{Trace}^2(A)$, $i=0.1, \dots, 63$ 을 계산하고 그 중 최대값

$C_n = \max\{c_i\}$ 를 구한다. 이때 $A = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}$ 는 각 pixel 위치에서

구하며 g_x, g_y 는 각각 x-축과 y-축 방향의 Gradient를 나타낸다. 다음으로 최대 Cornerness를 갖는 Position p_n 과 블록내의 다른 픽셀과의 Distinctness를 측정한다. 즉,

$$L_n = \sum_{i=0}^{63} \left\{ 1 - \exp \left(- \left(\frac{I(p_n) - I(i)}{th} \right)^6 \right) \right\} \quad (16)$$

이렇게 측정된 두 값을 조합하여 신뢰도를 나타내는 계수가 된다.

$$w_n = (C_n + \lambda L_n) / \max_k (C_k + \lambda L_k) \quad (17)$$

[14]에서는 전역 움직임 보상을 위한 Affine transform 모델을 가정하고 측정된 Local vector들과 Affine transform 후에 얻어지는 추정되는 Global motion vector와의 오차가 블록의 신뢰도 계수를 고려할 때 가장 작아지도록 이 Affine 모델에서 필요한 6개의 파라미터를 [55]에서 사용된 동일한 기법으로 구하였다. 즉, 다음 오차함수를 만족하는 상수 a_i 들을 구한다.

$$E(a) = \min \sum_{i=1}^N w_i \left\{ \begin{aligned} & \left[v_X(x_i, y_i) - (a_1 x_i + a_2 y_i + a_3) \right]^2 \\ & + \left[v_Y(x_i, y_i) - (a_4 x_i + a_5 y_i + a_6) \right]^2 \end{aligned} \right\} \quad (18)$$

이 방법에서는 신뢰도가 높은 블록의 중요도가 더 크게 고려되지만 많은 모션 블록이 계산에 사용되므로 계산 량이 많다.

한편, 적은 수의 블록을 사용하여 전역 움직임 보상을 이루는 방법이 [50]에 제안되었다. [50]에서는 수평 방향의 에지 성분이 강한 블록(Bh)과 수직방향의 에지 성분이 강한 블록(Bv)을 동시에 이용하여 정합을 수행한다. Bh와 Bv를 선택하기 위해 임의의 블록에 대하여 수평 수직 방향의 Gradient magnitude의 합을 계산한 뒤 이들을 비교한다. Bh(Bv)를 이용하면 수직(수평) 방향으로 급한 경사를 가지는 MAD(maximum absolute difference) 곡선을 얻을 수 있으므로, 이들을 동시에 이용할 때 모든 방향에서 급한 경사를 가지는 MAD 곡선을 얻을 수 있으며 GMV(global motion vector)를 유일하게 결정할 수 있다.

$$GMV = \arg \min_{(mv_x, mv_y)} \left(\begin{aligned} & \sum_{x, y \in B_h} w_h |I_t(x, y) - I_{t-\delta t}(x + mv_x, y + mv_y)| \\ & + \sum_{x, y \in B_v} w_v |I_t(x, y) - I_{t-\delta t}(x + mv_x, y + mv_y)| \end{aligned} \right) \quad (19)$$

단, $I_t(x, y)$ 와 $I_{t-\delta t}(x, y)$ 는 각각 현재 영상과 이전 영상의 (x, y) 에서의 휘도를 나타내며 w_h 와 w_v 는 각각 B_h 와 B_v 의 Gradient magnitude의 합의 역수로 정의되는 값으로서 각 블록의 MAD 곡선이

최종 MAD 곡선에 동등하게 기여하도록 하기 위한 것이다.

선택된 정합 블록들이 이동 물체를 포함하지 않도록 하기 위해 바로 이전 이동 물체 영역의 위치에 존재하는 블록은 선택하지 않으며, 또 움직임 추정 후 다음의 경우에는 계산된 GMV는 실제 전역 움직임을 구하지 못하고 국부 움직임 방향으로 치우쳤다고 판단한다. 즉, 만일 일부 블록이 이동 물체를 포함하는 경우 각 블록의 움직임 벡터가 서로 달라 “ B_h 블록만을 이용해 정합하는 MAD값”과 “ B_v 블록만을 이용해 정합하는 MAD값”을 독립적으로 구한 합이 “모든 (B_h , B_v)를 동시에 고려하여 블록 정합을 하는 경우의 MAD 값”에 비하여 크게 증가할 것이다. 만일 잘못된 결과로 판단되면 새로운 블록 쌍을 선정하여 다시 추정을 시도한다. [50]에는 또 전역 움직임 추정이 정확하지 않은 경우 이를 보상하기 위한 방안이 제안되었다. 전역움직임이 보상되어도 현재 영상과 이전영상과의 차영상에는 전역 움직임 추정의 오차 등에 따라 원하지 않는 배경의 에지 부가 나타나 있다. 이 에지 부를 제거하기 위하여 현재영상의 Gradient 정보를 이용하였다. 현재 영상 블록 내부의 Gradient 크기의 합을 나타내는 BGM(block gradient magnitude)과 차영상 블록 내부 밝기 값의 합을 나타내는 DBD(displaced block difference)의 (DBD, BGM) 순서쌍이 배경영역의 한 블록의 경우 직선 1의 하단부에 주로 분포한다는 관찰을 통해 차영상에 속한 어떤 블록이 이동물체에 속하는지 배경에 속하는지를 최종적으로 결정한다.

실험에 따르면 이 방법에서 4개의 블록만을 사용하여 전역 움직임을 추정하여도 234개의 전체블록을 사용하여 정합하는 경우에 대해 98%가 일치하였고 불일치 하는 경우에도 1 pixel 이하의 차이를 보였다. 따라서, 전역의 움직임을 보상하기 위하여 전체 블록을 사용하는

경우보다 훨씬 낮은 계산 량이 필요하다. [63]에서는 특징 점을 이용하여 Global 모션을 보정한 후 Pan/Tilt에 의해 움직이는 카메라로 이동물체를 추적하였다. 첫 단계에서 Pan/Tilt 구동에 따른 카메라의 이동 량을 계산하기 위하여 현재영상 I_t 와 이전영상 I_{t-1} 간의 영상 정합을 두 번의 Template 비교를 통해 시도한다. 각 영상에 대해 1/3 서브샘플링 된 이전 영상과 현재 영상을 구하고 이들의 정합을 통해 개략적인 이동 량을 구한 다음, 이 위치의 일정 반경 내에서 Subsample하지 않은 원래 영상을 사용하여 정교한 정합을 수행한다. 이때 Template measure로는 픽셀간 차이의 절대값이 사용되었다.

두 번째 단계인 특징 점 추적단계에서는 계산된 카메라 이동 량을 현재 특징 점 좌표에 더하여 카메라 움직임에 의한 광역 움직임을 제거한 다음, 움직임이 보상된 좌표주변의 일정영역에서 특징 점을 선택하고 상관정합기법을 이용하여 다음 특징 점의 움직인 위치를 추적한다. 특징 점으로는 꼭지점(corner) 추출기법을 사용하였는데 7x7 혹은 9x9로 분할한 작은 사각형 내부의 x-축과 y-축 방향의 gradient

g_x , g_y 를 각각 구한 후 다음의 행렬
$$\begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}$$
의 고유 값의

크기가 임계값 이상인지를 판정하여 사용하였다. 이 방법은 [14]에서 사용한 Cornerness 측정과 비슷한 방법이다. 다음에는 현재 이미지와 이전 이미지의 특징 점들의 움직임 거리를 계산하고 특징 점들의 평균움직임 거리를 계산하여 평균 이동 량을 계산한다. 특징 점으로부터 이동체를 배경과 분리하기 위하여 계산된 평균 이동 량을 사용하여 움직임 모션과 정지모션을 구분한 후 정지모션은 제거하게 된다. 마지막으로 특징 점들의 중심위치를 구한 다음 카메라 중심으로 이동하도록 카메라의 Pan/Tilt를 구동시킨다.

제 2 절 물체 추적

1. Template Matching

Template를 사용하는 Tracking method의 기본적 동작은 다음과 같다.

- ① 초기화: 처음에 물체의 검출을 통하여 위치를 파악하고 Template를 계산하여 둔다.
- ② 추적:
 - (a) 현재 영상에서 Template에 가장 잘 Matching 되는 영상의 영역을 찾아내고 물체의 새로운 위치로 정의한다.
 - (b) 최근의 측정(measurement) 값을 반영하는 Smoothing filter를 물체의 새로운 좌표에 적용하여 좌표의 잡음을 제거한다.
 - (c) 다음 영상에서의 물체의 위치를 예측하기 위한 움직임 예측을 수행한다.

Step ②(c)의 예측단계가 효율적이지 못하면 Step ②(a)에서 탐색 영역을 줄일 수 없으므로 많은 영역을 탐색해야 하고 시간과 비용이 많이 들게 된다. 따라서 효율적인 예측기법이 매우 중요하다.

BMA(block matching algorithm)을 전적으로 사용하여 자동차의 추적에 사용하는 방법이 [46]에 보고되었다. 현재의 영상 프레임을 많은 작은 블록으로 나누고 이전 프레임의 블록과 정규화 된 Cross-correlation을 비교하여 연속되는 두 작은 블록 사이의 변위를 추정해 낸다. BMA의 결과로부터 각 블록의 Local motion vector를 구할 수 있으며 Local motion vector가 가지는 Noise를 제거하기 위하여 통상의 Median 필터와 마찬가지로 동작하는 VMF(vector median filter)

를 사용하여 잘못된 Local motion vector 추정치를 제거한다.

다음에 비슷한 Displacement vector를 가지는 8-개의 연결된 Cluster를 하나의 그룹으로 묶는 과정을 거치면 Labeling이 된(같은 물체에 속한) 연결된 블록들을 얻게 된다. 이 Label 정보를 두 연속되는 영상 사이에서 비교함으로써 자동차를 추적할 수 있다.

이 방법은 전적으로 BMA에 의존하므로 계산량이 매우 많다는 문제점을 가진다. 또 계산블록의 크기보다 자동차가 카메라에 가까워 매우 크게 잡힐 때에는 한 자동차가 너무 지나치게 많은 블록으로 분할되어 잘못 추적될 확률이 높아진다.

[15]에서는 영상 피라미드(pyramid)를 이용한 다해상도(multi-resolution) Correlation matching 방법을 제시하였다. 먼저 두 개의 연속되는 영상은 각각 영상 피라미드(pyramid) 구조로 표현된다. 영상 피라미드에서 상위 층은 영상의 해상도가 바로 아래층에 비하여 2배 낮지만 반대로 Template matching은 Full-resolution의 경우보다 빠르게 진행된다. 영상 피라미드에서의 Correlation 정합은 다음과 같이 이루어진다. 먼저,

- ① 연속되는 두 영상을 영상 피라미드로 표현한다.
- ② 가장 해상도가 낮은 상위 층에서부터 움직이는 목표물체의 변위를 계산한다.
- ③ 상위 층에서 구한 목표물체의 변위를 다음 해상도가 높은 하위 층으로 Mapping시킨 다음 현재 해상도 레벨에서 변위를 계산한다.
- ④ 가장 해상도가 높은 최하위층에 도달하였으면 알고리즘을 끝내고 아니면 ③으로 가서 반복한다.

4개 Level의 영상 피라미드를 구축하고 수행한 실험결과에 따르면

통상의 Mono-resolution algorithm에 비하여 약 160배의 계산 시간 개선이 있는 것으로 보고되었다.

Template matching에 의하여 움직이는 물체를 정확히 추적하기 위하여 정확한 Template의 추출은 매우 중요하다.

[18]에서는 추가적으로 더 정확한 Template를 추출하기 위한 움직이는 물체(자동차)의 그림자를 제거하기 위한 방법을 제안하였다. 배경 영상과 입력 영상의 차영상을 DCT 하면 움직이는 물체의 그림자 부분은 비슷한 픽셀값을 넓은 범위에 걸쳐 가지므로 큰 DC 성분을 갖는 반면, 움직이는 자동차 부분은 상대적으로 큰 AC 성분을 갖는다는 점을 착안하여 그림자를 제거한 후 움직이는 자동차의 윤곽을 더 정확히 얻어낸다.

그러나 만약 움직이는 자동차의 상대적으로 넓은 범위에 거의 비슷한 값을 갖는 픽셀이 존재하게 되면 그림자 부분과 비슷한 특성을 보이게 되어 역시 큰 DC 성분을 가지게 되므로 그림자로 인식되어 제거되는 문제가 있다. 이를 보완하기 위하여 이전 영상에서 추출된 움직이는 물체의 모션 벡터를 이용하고, Affine transform을 이용하여 현재 프레임에서의 물체 정보를 추정하고 그림자를 제거한 영상과 보완적으로 사용한다.

추출된 템플릿 영상을 사용하여 현재 영상 내에서 찾아낸 모든 이동물체의 최소 경계영역 영상을 대상으로 템플릿 비교연산을 수행한다. 기존에 사용하던 정규화된 상관도를 이용하는 등의 방법 이외에 Hausdorff distance를 이용한 연산이 근래에 많은 주목을 받고 있다.

* Hausdorff distance를 이용한 Template 비교

주어진 유한한 점(point)의 집합

$$A = \{a_1, a_2, \dots, a_p\}, B = \{b_1, b_2, \dots, b_q\} \quad (20)$$

에 대해 Hausdorff 거리는 다음과 같이 정의된다.

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (21)$$

여기서

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (22)$$

이때, $\|\cdot\|$ 은 A 와 B 에 속한 점들 사이에 적용되는 norm(혹은 Euclidean norm)을 말한다. $h(A, B)$ 는 집합 A 에 속한 각 점에 대하여 집합 B 에 속한 모든 점들 중 가장 가까운 점까지의 거리를 구한 후, 그 거리 값들 중에서 가장 큰 값을 출력한다. 만약 $h(A, B) = d$ 라면, 집합 A 에 속한 각 점은 집합 B 에 속한 모든 점과 거리 d 내에 있다는 뜻이다. 그러나, Hausdorff distance는 outlier point가 존재하는 경우 잡음이나 가려짐(occlusion)에 민감하므로 [26]에서는 Partial Hausdorff distance를 제안하고 Correlation에 근거하여 영상 속에서 모델을 찾아내는 것 보다 성능이 더 우수함을 보이는 실험 결과를 소개하였다.

$$h_K(A, B) = K_{a \in A}^{th} \min_{b \in B} \|a - b\| \quad (23)$$

여기서 $K_{a \in A}^{th}$ 는 집합 B 까지의 거리 $\min_{b \in B} \|a - b\|$ 를 계산한 후 순서를 지어 나열하였을 때 K -번째 거리의 값을 의미한다. K 개의 Best matching point 만을 사용하므로 Outlier point가 존재하거나 가려짐이 발생할 때 성능이 원래의 정의보다 더 우수하다.

다양한 변형된 형태의 Hausdorff distance가 물체 추적에 사용되었다. 문헌 [32]에서는 여러 가지 정의에 따른 Hausdorff distance를 level=3인 피라미드(pyramid) 구조의 영상에 대하여 수행한 결과를 비교 설명하였다. 피라미드 구조는 영상의 해상도가 단계가 높을 수록 낮아지므로 낮은 해상도에서의 영상 Matching을 시도한 후 필요에 따라 더 높은 해상도에서의 Matching을 시도하므로 연산량을 획기적으로 줄일 수 있다. 실험결과에 따르면 피라미드 영상구조를 사용하지 않는 방법에 비하여 약 40배의 계산시간 감소를 얻을 수 있다. 또, Olson [34]는 단순 천이(translation) 뿐만 아니라 회전(rotation)과 크기변화(scale)를 고려하고, 위치와 Orientation을 고려하도록 변형된 Hausdorff distance를 정의하여 사용하였다. [42]에도 모델과 영상 사이의 Partial Hausdorff distance를 이용한 추적결과를 제시하였다.

[32][34][42] 등에 제시된 Hausdorff distance에 의한 모델과 영상 Matching에 의한 추적은 비록 다른 Template measure 보다 정확하게 모델영역과 매칭되는 영상 영역을 검출할 수는 있으나 계산에 너무 많은 시간(피라미드 구조[32]를 이용한 경우조차도 한 영상의 Matching에 수초가 소요)이 필요하다는 단점이 있다.

2. Kalman Filtering

[23]은 복잡한 배경에 대해 움직이는 복수의 사람들의 움직임을 고정된 Monochrome 카메라 영상으로부터 검출하고 얼굴부위를 추적하는 시스템이다. 이 시스템에는 시간에 따른 물체의 움직임 영역을 비교적 빠른 시간 내에 보여줄 수 있지만 Motion-based tracker로서 Kalman filter를 채용하여 움직임이 있는 Cluster list에 관한 정보를 얻은 후, 이를 이용하여 Model-based tracker인 Neural networks을 이용한 얼굴 검출기가 탐색해야 하는 공간의 크기를 줄이도록 하였다.

움직임 정보는 식(24)와 같이 영상 $I(x, y, t)$ 을 Gaussian function $G(t)$ 의 2차 시간미분과 Convolution하여 얻은 영상 $S(x, y, t)$ 의 Temporal zero-crossing 정보로부터 얻는다.

$$S(x, y, t) = \frac{\partial^2 G(t)}{\partial t^2} * I(x, y, t) \quad (24)$$

입력 영상 열(sequence)에서의 에지는 시간적 Convolution을 수행하는 중간 프레임 $S(x, y, t)$ 에서 시간적인 Zero-crossing으로 나타나게 되며, 이를 검출하여 움직이는 물체가 존재할 수 있는 Cluster를 검출한다.

검출된 Cluster는 움직이는 물체에 해당되지만 실제로는 항상 그렇게 되는 것은 아니다. 하나의 움직이는 물체는 여러 개의 Cluster로 나타날 수 있으며 반대로 여러 개의 움직이는 물체가 하나의 Cluster로 뭉쳐져서 보일 수도 있다. [23]에서는 각 Cluster를 중심점(centroid)과 Zero-crossing 점들의 표준편차를 기준으로 Euclidean

distance에 근거하여 Model하였다. 이러한 Cluster들을 추적하기 위하여 Cluster position, Motion, Shape의 Measurement에 근거한 Kalman filter가 채용되었다.

Cluster의 중심점은 Position, Velocity, 그리고 가속도를 사용하여 모델이 된다. x-축에 대하여 다음의 시스템 모델이 사용되었다.

$$\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_k + \begin{bmatrix} 0 \\ 1/2 \\ 1 \end{bmatrix} v_{acc} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_{pos} \quad (25)$$

여기서 v_{acc} 와 v_{pos} 는 각각 가속도와 위치 추정에 대한 Noise covariance이다. Measurement model은 다음과 같다.

$$z_{k+1} = x_{k+1} + w \quad (26)$$

여기서 z_{k+1} 와 x_{k+1} 은 각각 t_{k+1} 에서의 관측(observation)벡터와 상태 벡터이고 w 는 Noise covariance vector이다. Cluster는 일정한 높이와 폭을 가지는 것으로 모델이 되었다. 시스템 모델과 관측모델은 식(27)과 같다.

$$\begin{aligned} h_{k+1} &= h_k + v_h, & w_{k+1} &= w_k + v_w \\ z_{k+1} &= h_{k+1} + u, & z_{k+1} &= w_{k+1} + u \end{aligned} \quad (27)$$

여기서 v_h, v_w 는 시스템 Noise이고 u 는 관측 노이즈(noise)이다.

Cluster는 적어도 4 프레임 동안 추적이 되는 경우에 Cluster bounding box로 초기화되고, $p=10$ 프레임 동안 Matching cluster의 추적이 되지 않더라도 Cluster의 정보를 유지한다. 이것은 짧은 기간 동안 어떤 물체에 대해 움직임 검출이 실패하여 Clustering error가 발생하더라도 지속적으로 추적이 지속될 수 있도록 하기 위한 것이다. 이상의 Kalman filtering 결과로 움직이는 물체의 Cluster를 추적할 수 있으며 추적중인 Cluster의 Bounding box를 얻을 수 있다.

추적 중인 물체가 사람이므로 얼굴의 위치와 크기는 Cluster의 Bounding box로부터 유추할 수 있다. 얼굴의 검출(detection)은 Neural networks을 이용하여 이루어 졌다. Neural network은 정면 혹은 거의 정면 영상들로 이루어진 9000여장의 얼굴영상을 사용하여 Training하였다. 실험[23]에 따르면 이렇게 훈련된 Neural network 라도 주어진 영상의 전체 영역에서 사람의 얼굴을 정확히 찾아 내는 데는 많은 문제가 있으며 얼굴과 비슷한 배경 영역을 얼굴로 잘못 찾아내는 일이 많다.

그러나, Kalman filter에 의한 Motion prediction으로 인해 얼굴을 검출하기 위한 탐색 후보 영역이 크게 줄어들게 되므로 잘못된 검출을 하는 일을 막을 수 있다. 또, Neural network을 이용한 얼굴 검출에서 일치하는 정도에 대한 정도를 Kalman filter로 Feedback시켜 더 나은 Motion prediction을 하게 할 수 있는 장점이 있다.

“Intelligent Train Platform” 프로젝트의 일환으로 3차원 모델을 사용하여 사람을 검출과 위치를 추적하는 자동 영상 감시장치가 [41][33]에 소개되었다. 영상처리 기술을 응용하여 지하철 역사에서 기차의 진입을 감시하고, 플랫폼의 승객의 승 하차를 감시하여 기차의 정지시간을 줄이기 위해 사용되었다. 비용을 절감하기 위하여 고정된

초점거리를 갖는(zoom이 없는) 고정(static) 카메라를 한 개 사용하여 3-D로 사람을 검출하고 추적한다. 시스템은 전처리 부, 사람 검출 부, 사람 추적 부, 그리고 모델 부로 이루어져 있다. 전처리부는 [48]에서 소개된 배경영상 추정 기법을 이용하여 변화가 있는 부분을 검출하고 검출된 변화영역(ROI: region of interest)의 에지를 Sobel 알고리즘으로 검출하여 사람 검출 부에 공급한다.

사람 검출 부에서는 전처리부에서 제공된 에지 Feature를 사용하여 3D로 모델이 되는 사람의 상태를 검출한다. 상태 공간에서 사람은 3개의 좌표와 1개의 회전각도 그리고 이것들의 1차 미분 값을 포함한 8개의 파라미터로 표현되며 해당되는 모델의 형태가 계산된 영상 에지에 일치하는 정도를 상대적인 겹침(overlap)의 정도로 나타낸 후 Condensation 알고리즘을 사용하여 각 사람에 대한 Hypothesis를 얻는다. 이 Hypothesis를 영상 내의 3D 공간으로 투영하여 Cluster 위치를 얻은 후 Smoothing 하여 Local maximal이 되는 좌표를 구하고 이 위치를 Cluster의 초기 위치로 사용한다.

복수의 사람 추적은 보편적으로 채용되는 Kalman filter를 이용하여 이루어진다. Kalman filter는 1차 움직임 모델이 채용되어 사람이 갑작스럽게 방향을 바꾸는 경우 속도(velocity) 성분의 잡음으로 모델이 되므로 Forgetting factor를 도입하였다. 사람 검출 부에서 얻어진 Hypothesis와 추적되는 사람과의 결합(data association)을 위하여 Hypothesis set에 속한 Vertex와 추적되는 사람의 Set에 속한 Vertex를 결합시키는 비용을 최소화 시키는 Bipartite graph matching model이 사용되었다.

영상 기반 자동차감시를 위한 시스템인 [45]에서도 Kalman filter가 자동차의 추적에 사용되었다. [45]에서는 주행중의 가려짐

(occlusion)을 검출하여 처리하였다. 두 개 이상의 물체가 Merging 되었다가 다시 분리되는 경우 물체의 Feature(형상, 중심 위치, MBB, 속도)에 따라 유사도를 정합하여 물체가 가려짐에서 벗어나 다시 분리되는 경우 각 물체의 경로를 정확히 추적한다.

제 4 장 알고리즘 상세 설명

제 1 절 배경 모델 생성 및 갱신

연속된 영상에서 이동 객체를 분할하기 위해서는 우선 입력 영상으로부터 배경 영상을 생성하여야 한다. 생성된 배경 모델을 이용하여 입력된 영상에서 배경 영역을 제거함으로써 이동 객체를 분리할 수 있다. 본 과제에서는 동영상에서 배경 모델 생성 및 갱신을 위하여 I. Haritaoglu 등에 의해 제안된 방법을 사용하였다. 이 방법은 연속된 영상에서 누적된 픽셀 값의 최대, 최소, 차분 값을 사용하여 배경 모델을 생성한다.

영상 V 에서 위치 x 의 초기 배경 모델 $[m(x), n(x), d(x)]$ 은 식 (28)과 같이 생성한다. 식 (28)에서 V 는 연속된 N 개의 영상의 배열을 의미한다. $V^z(x)$ 는 V 에서 z 번째 영상의 위치 x 의 픽셀 값이며 $V^z(x)$ 은 고정된 배경으로 선택된 픽셀 값을 의미하고, $\sigma(x)$ 과 $\lambda(x)$ 는 전체 영상의 위치 x 의 픽셀의 표준 편차와 중간 값을 나타낸다.

$$\begin{bmatrix} m(x) \\ n(x) \\ d(x) \end{bmatrix} = \begin{bmatrix} \min_z \{V^z(x)\} \\ \max_z \{V^z(x)\} \\ \max_z \{|V^z(x) - V^{z-1}(x)|\} \end{bmatrix}, \quad (28)$$

$$\text{where } |V^z(x) - \lambda(x)| < 2 \times \sigma(x)$$

배경 모델은 조명의 변화와 같은 영상의 변화에 대응하기 위하여

지속적인 갱신을 필요로 한다. 배경 모델 갱신을 결정하기 위하여 change map을 구성하게 되며 각 map의 구성은 아래와 같다.

Detection support map (gS) : 이전 N 프레임 영상 동안 픽셀이 배경이었던 횟수를 의미한다.

$$gS(x,t) = \begin{cases} gS(x,t-1)+1 & \text{if } x \text{ is background pixel} \\ gS(x,t-1) & \text{if } x \text{ is foreground pixel} \end{cases} \quad (29)$$

Motion support map (mS) : 픽셀이 움직이는 픽셀이었던 횟수를 의미한다. 움직이는 픽셀의 판단은 연속된 3장의 영상 차이를 통해 계산한다.

$$mS(x,t) = \begin{cases} mS(x,t-1)+1 & \text{if } M(x,t) = 1 \\ mS(x,t-1) & \text{if } M(x,t) = 0, \end{cases} \quad (30)$$

where

$$M(x,t) = \begin{cases} 1 & \text{if } (|I(x,t) - I(x,t+1)| > 2*\sigma) \wedge \\ & (|I(x,t-1) - I(x,t)| > 2*\sigma) \\ 0 & \text{otherwise.} \end{cases} \quad (31)$$

Change history map (hS) : 마지막으로 전경 영역으로 판단되어진 시간을 의미한다.

$$hS(x, t) = \begin{cases} 255 & \text{if } x \text{ is foreground pixel} \\ hS(x, t-1) - \frac{255}{N} & \text{otherwise.} \end{cases} \quad (32)$$

위와 같은 change map을 이용하여 배경모델의 갱신이 이루어지며, 이 과정은 식 (33)에 나타난 것과 같다. 배경 모델을 이용하여 전경과 배경 영역을 분리하고 이동 객체를 추적하는 동안 영상의 모든 픽셀에서 배경 모델의 갱신이 필요한지를 결정하게 된다. 위치 x 의 배경 모델을 갱신할 경우 전경 또는 배경 영역의 값으로 배경 모델 변경하며 갱신이 필요 없다고 판단된 경우 원래의 값을 유지한다. 식 (33)에서 $(m^f(x), n^f(x), d^f(x))$ 는 전경으로 판단된 픽셀의 모델 의미하고, $(m^b(x), n^b(x), d^b(x))$ 는 배경으로 판단된 픽셀의 모델을 의미한다. 마지막으로 $(m^c(x), n^c(x), d^c(x))$ 는 기존의 값을 의미한다.

$$[m(x), n(x), d(x)] = \begin{cases} [m^b(x), n^b(x), d^b(x)] & \text{if } (gS(x) > k * N) \text{ (pixel-based)} \\ [m^f(x), n^f(x), d^f(x)] & \text{if } (gS(x) < k * N \wedge mS(x) < r * N) \\ & \text{(object-based)} \\ [m^c(x), n^c(x), d^c(x)] & \text{otherwise,} \end{cases} \quad (33)$$

이때 k 와 r 의 값은 경험적으로 선택되었으며, 이 값에 따라 분리되는 전경 영역을 조절할 수 있다.

제 2 절 전경 영역 분리

전경 영역 분리는 연속된 영상에서 움직이지 않는 영역을 배경이라 가정하고 이렇게 생성된 배경 영상과 입력 영상의 차이를 이용하여 연속된 영상에서 움직임이 있는 영역을 분리하는 방법이며 영상 감시 시스템 등 움직이는 물체를 검출하여 사용하는 시스템에 중요하게 이용 될 수 있다.

생성된 배경 모델 $[m(x), n(x), d(x)]$ 를 이용하여 입력된 영상에서 움직이는 물체를 식 (34)에 나타난 것과 같이 분리할 수 있다. 식 (34)에서 $I'(x)$ 는 입력 영상에서 위치 x 의 픽셀 값을 나타내며 상수 k 는 실험적인 결과를 통해 얻어 낼 수 있다. 상수 k 가 적은 경우 보다 많은 전경 영역을 분리해 내며 반대로 k 가 커질수록 분리되는 전경 영역이 적어지게 된다.

$$B(x) = \begin{cases} 0, background & \begin{cases} (I'(x) - m(x)) < kd_u \\ \vee (I'(x) - n(x)) < kd_u \end{cases} \\ 1, foreground & otherwise. \end{cases} \quad (34)$$

위와 같은 방법을 이용하여 전경 영역을 분리한 결과는 그림 5에서와 같이 분리된 전경 영역에 그림자가 포함된다. 이를 해결하기 위해서는 입력된 영상에서 그림자 영역을 제거해야 한다.



(a) 입력영상 (b) 배경영상 (c) 전경영상

그림 5. 전경 영역 검출 영상

그림자 영역의 픽셀들은 다음과 같은 특징을 갖는다고 가정하였다.
 1) 그림자 영역에서는 배경 모델과 동일한 색상을 유지하며 2) 그림자 영역의 픽셀은 배경 모델의 픽셀보다 낮은 밝기 값을 갖는다. 이러한 가정을 이용하여 분리된 전경 영역에서 그림자를 제거한다.

식 (35)는 그림자 영역의 경우 픽셀의 밝기 값을 변화시킨다는 첫 번째 가정에 의하여 각 픽셀이 그림자 영역에 속할 확률을 표현한 것이다.

$$P_{color}(i) = \exp \left(- \left(\frac{(I_{r_i} - m_r)^2}{2 \times s_r^2} + \frac{(I_{g_i} - m_g)^2}{2 \times s_g^2} \right) \right) \quad (35)$$

식 (35)에서 m_r, m_g, m_b 는 각 채널에 대해 정규화된 평균이고, s_r, s_g, s_b 는 정규화된 RGB 각 채널에 대한 표준 편차이다. 즉, 특정 위치의 픽셀과 배경 영상의 색상 차이가 적을수록 해당 픽셀은 그림자 영역에 속한 픽셀일 확률이 높아진다. 또한 그림자는 배경 영상에서 밝기 값이 낮아진다는 두 번째 가정에 기반하여 각 픽셀이 그림자

영역에 속할 확률을 식 (36)와 같이 나타낼 수 있다.

$$P_{brightness}(i) = \exp\left(-\frac{(I_d - M_d)^2}{2 \times S_d^2}\right) \quad (36)$$

식 (36)에서 I_d 는 배경 영상과 입력 영상의 밝기 변화이고, M_d, S_d 는 실험적으로 얻어진 그림자가 갖는 밝기 변화의 평균과 표준편차이다. 식 (35), (36)에서 얻어진 확률 분포를 이용하여 입력된 영상에서 각 픽셀이 그림자 영역 안에 속한 픽셀일 확률을 식 (37)과 같이 나타낼 수 있다.

$$P_{shadow}(i) = P_{color}(i) \times P_{brightness}(i) \quad (37)$$

식 (37)과 같이 입력 영상에서 각 픽셀이 그림자 영역에 포함될 확률을 계산하고 임계값을 기준으로 영상의 그림자를 제거하였다.

제 3 절 이동 객체 추적

이동 객체 추적은 영상 분석에 있어서 매우 중요한 요소이며 이를 통하여 현재 영상의 물체와 이전 영상의 영역의 관계를 알 수 있고 이러한 정보를 이용하여 이동 객체의 이동, 생성, 소멸 등에 대한 정보를 얻을 수 있다. 이러한 이동 객체 추적은 영상에서 사람의 움직임 분석, 침입자 감지 등 다양한 분야에 이용할 수 있다.

이동 객체를 추적하기 위하여 전경 영역 분리 기법으로 전경 영역

을 분리하고 Labeling 과정을 거쳐 각각의 영역으로 구분한다. 이렇게 구분된 각각의 영역을 하나의 이동 객체로 가정하고 연속된 영상에서 각각의 영역을 비교하여 이동 객체를 추적 한다. 연속된 영상에서 분리된 영역은 단순히 위치 이동만이 존재할 시에는 매우 쉽게 영역을 추적할 수 있지만 시간이 지남에 따라 물체의 겹침 현상으로 인하여 두 영역이 병합 될 수도 있고 이와는 반대로 하나의 영역이 둘 이상의 영역으로 분리되는 문제가 발생할 수 있다. 이 경우 영역간의 물체 추적이 어렵게 된다. 이러한 문제를 해결하기 위하여 분리된 전경 영역의 추적은 Tao Zhao에 의해 제한된 Blob tracker를 사용하였다. Blob tracker는 시간적으로 인접한 두 영상을 비교하여 위치 이동, 분리, 병합 등에 대한 추적을 할 수 있으며 전체 흐름도는 그림 6와 같이 구성된다.

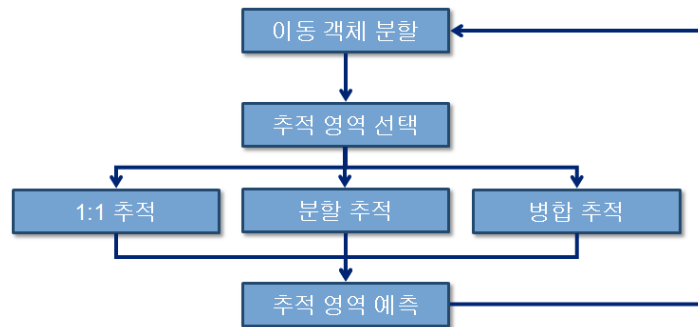


그림 6. 이동 객체 추적 흐름도

일치 (Perfect-match)

시간 t 에서 분리된 전경 영역을 식 (38)에서와 같이 표현할 수 있다. 이전 영상의 전경 영역과 현재 영상의 전경 영역이 기준보다 차이가 작은 경우 (영역의 중점의 차이가 적고 크기의 차가 일정 임계 값

보다 작은 경우) 우리는 이 두 영역을 일치했다라고 하고 분리, 병합에 대한 계산을 하지 않는다.

$$B^{(t)} = \{B_1^{(t)}, \dots, B_n^{(t)}\} \quad (38)$$

☞ 분리 (Split)

시간 t 에서 분리된 둘 이상의 전경 영역이 이전 영상의 전경 영역과 일치 할 경우 우리는 영역이 분리 되었다고 한다. 이때 둘 이상의 전경 영역과 이전 영상의 전경 영역의 비교는 현재 영상의 가능한 조합을 구성하여 조합된 영역의 중점과 크기를 기준으로 일치 (Perfect-match)과 같은 방법으로 비교를 한다.

☞ 병합 (Merge)

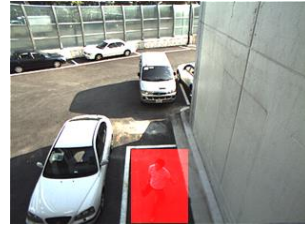
시간 t 에서 분리된 하나의 영역이 이전 영상의 둘 이상의 전경 영역과 일치 할 경우 우리는 영역이 병합 되었다고 한다.

제 4 절 금지 구역

금지 구역 감시는 사용자가 시스템에 금지구역을 설정하고 움직이는 물체가 이 금지구역에 침범했을 경우 사용자에게 경고를 통보하게 된다. 금지구역에 대한 침범 여부는 앞서 설명한 배경 추출과 물체 추출 알고리즘을 통해 추출된 물체의 특정 위치(Ref. Point)가 금지구역에 들어오는 경우로 설정하였다.

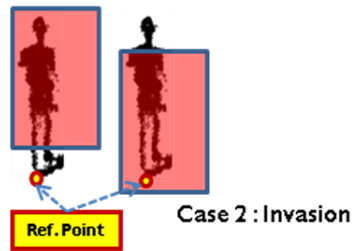


(a) 금지구역설정



(b) 금지구역침범

Case 1 : Not Invasion



(c) 금지구역 침범 판단 근거

그림 7. 금지구역 침범 감시

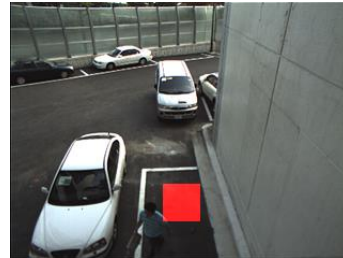
그림 7은 금지구역이 설정된 예와 금지구역에 물체가 침범한 경우를 보여주고 있다. (a)는 설정된 금지구역을 보여주고 있다. (b)는 금지구역에 사람이 침범한 경이이다. (c)는 금지구역에 침범이 발생했는지에 대한 판단의 근거를 보여주고 있다. (c)에서 보듯이 앞서 설명한 알고리즘을 통해 추출된 물체의 최 하단(Ref. Point)이 금지구역을 통과했을 경우를 금지구역에 대한 침범으로 간주하게 된다. 그 이유는 사람의 시각의 위치에 따라 몸통이 금지구역에 들어갔을 경우 이는 실제로는 금지구역을 침범하지 않은 경우가 많다. 그림 7 (c)의 case1이 여기에 해당한다. 따라서 case2처럼 물체의 최 하단이 금지구역에 들어왔을 경우에만 금지구역 침범으로 간주한다.

제 5 절 도난/불법 방치

도난 방지란 사용자가 임의의 물체에 대해 감시영역을 설정해 두고 이 영역 내에서 감시 물체가 사라졌을 경우 사용자에게 경고를 알리는 기능을 말한다. 불법 방치는 이와 반대의 경우라고 할 수 있다. 즉 사용자가 설정한 임의의 영역에 불법 방치 물이 놓여졌을 경우 사용자에게 경고를 알리는 기능이다.



(a) 감시영역설정(도난)



(b) 도난발생



(c) 감시영역설정(불법방치)



(d) 불법방치발생

그림 8. 도난/불법 방치

그림 8은 도난/불법 방치에 대한 예를 보여주는 것이다. (a), (b)에서 보듯이 사용자가 설정한 감시 영역 내에 감시하고자 하는 물체가 있고 이 물체가 도난 되었을 경우 경고가 관리자에게 통보된다. (c),(d)에서는

사용자가 불법 방치를 감지하고자 하는 영역을 설정하고 이 영역에 불법 방치 물이 발생했을 경우 역시 경고를 관리자에게 통보하게 된다.

도난 및 불법 방치 감시에 대한 알고리즘은 다음과 같다.

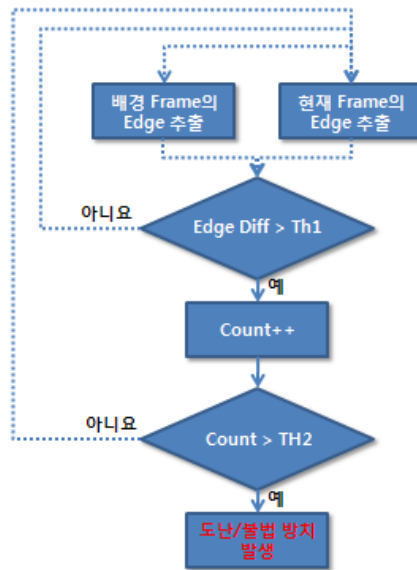


그림 9. 도난/불법 방치 감시 알고리즘

그림 9에서 보듯이 도난 및 불법 방치 감시를 위해 실시간으로 입력되는 현재 Frame과 동시에 만들어 지는 배경 Frame 각각에 대해 에지 정보를 추출한다. 추출된 에지 정보를 이용하여 두 Frame 간에 에지의 차를 구한다. 이때 구한 차가 임의의 임계값 $Th1$ 을 넘게 되면 설정된 영역 내에 이상 징후가 발생할 가능성이 생긴 것이다. 이 때 이 차 값 만으로 도난 이나 불법 방치 여부를 판단하게 되면 오 경보 율이 높아지게 된다. 에지의 차에 의해 생긴 정보는 우선 영역 내에 문제가 발생할 가능성이 있다는 것 뿐 아직 문제가 발생한 것은 아니기 때문이다.

따라서 에지의 차가 임의의 임계값보다 높을 경우 발생 빈도수에 해당하는 Count를 1 증가 시켜둔다. Count가 증가 되는 시점에는 증가된 Count를 이용하여 실제로 도난 및 불법 방치 횟수가 얼마나 발생했는지를 판단하게 된다. 즉 시간이 지날수록 에지 차가 계속 발생하고 이때 마다 Count 가 증가하므로 Count의 값이 임의의 임계값 Th2 보다 높다는 것이 현재 감시 영역에 이상 징후가 계속 발생하고 있다는 것이 된다. 따라서 Count가 임의의 임계값 Th2보다 크다면 실제로 도난 및 불법 방치가 발생했다고 판단 관리자에게 경보를 알리게 된다.

결론적으로 현재 Frame과 배경 Frame 간의 에지의 차가 계속 발생하고 이 차의 발생이 일정 시간 계속 발생한다면 도난 및 불법 방치 이벤트가 발생했다고 판단하게 되는 것이다.

제 6 절 카메라 방해

카메라 방해란 감시 지역에 설치된 카메라에 누군가가 카메라 앞을 가렸을 경우 이 이벤트를 감지해 내는 기능을 말한다. 실시간으로 감시 중인 카메라가 외부의 요소에 의해 인위적으로 방해를 받는다면 시스템은 카메라가 방해를 받은 그 시점부터 올바른 감시를 수행할 수 없게 된다. 따라서 카메라 방해가 발생했을 경우 관리자에게 즉각적으로 알려 빠른 대처가 되도록 할 필요가 있다.

여러 번의 테스트를 거쳐 분석한 결과 시스템에 카메라가 방해를 받을 경우 다음과 같은 증상 및 영상이 취합되게 된다.

- 입력된 현재 Frame이 전체적으로 검은색을 나타낸다.
- 현재 입력되는 Frame과 배경 Frame 이 큰 차이를 나타낸다.

- 순간적으로 전경픽셀의 수가 증가하는 현상을 나타낸다.

위의 정보를 바탕으로 카메라 방해 판단하기 위해 중요한 2 가지 특징량(feature)를 추출한다. 한가지는 현재 시스템에 추출된 전경픽셀의 양이고 다른 하나는 현재 입력되는 Frame에서 검은색(black) 픽셀이 차지하는 비율이다. 앞에서 정리한 카메라 방해에 대한 증상에서 보듯이 카메라가 가려졌을 경우 입력되는 영상이 전체적으로 검은색을 나타내는 경향이 많았었다. 따라서 현재 입력되는 현재 Frame에서 각 픽셀에 대해 특정 임계값(threshold)보다 낮은 검은색 픽셀의 개수를 구한다. 이렇게 구해진 검은색 픽셀 수를 전체 픽셀의 수로 나누어 검은색 픽셀이 차지하는 비율을 구하게 된다. 이 비율이 높다는 것은 현재 카메라가 가려졌을 가능성이 높다는 것을 의미한다. 검은색 픽셀의 비율과 동시에 앞서 설명한 물체 추출 알고리즘을 통해 추출되는 전경픽셀의 수를 구한다. 따라서 현재 Frame과 배경 Frame 간의 차를 구해서 전경픽셀로 인식되는 픽셀의 수를 구하고 이를 전체 픽셀의 수로 나눠서 전체 픽셀에서 전경픽셀이 차지하는 비율을 구한다. 이렇게 구해진 두 개의 특징량을 이용 두 개의 특징량이 임계값을 넘는 경우 카메라 방해가 발생했다고 간주하고 이를 관리자에게 통보하게 된다. 카메라 방해 판단을 위해 추출하는 특징량을 표 2에 정리하였다.

표 2. 카메라 방해를 위해 추출하는 특징량

- ① 현재 Frame에서 검은색 계열의 픽셀이 차지하는 비율
- ② 배경 Frame 과 현재 Frame 사이의 차를 구해서 전경픽셀이라고 판단되는 픽셀의 수가 전체 영상 크기에서 차지하는 비율

제 5 장 시스템 상세 설명

제 1 절 카메라

본 프로젝트에서는 크게 3종류의 카메라를 사용하였다. 감시 영역을 전체적으로 촬영하기 위해 사용한 고정 카메라로 USB 카메라와 Axis 210 네트워크 카메라를 사용하였고 움직이는 객체를 추적하기 위해 Axis 214 Pan-Tilt-Zoom(PTZ) 카메라를 사용하였다.

1. USB 카메라

PC에서 손쉽게 영상을 얻기 위해 USB로 연결되는 카메라를 사용하였다. USB카메라는 총 두 종류의 카메라를 사용하였다.

1)ZECA MV-402

제품 사진	
이미지 센서	1/3" Color CMOS
화소수	33만 화소
전송속도	VGA : 15 프레임/초 CIF : 30프레임/초
출력신호	USB format

ZECA MV-402 USB카메라는 프로젝트 초반에 사용한 영상 Capture 장비이다. 이 장비에서 촬영한 영상을 얻기 위해 Video for Windows(VFW) 라이브러릴 이용하였다. 하지만 ZECA MV-402 USB 카메라의 경우 노출과 화이트밸런스를 고정시킬 수 없었고 영상의 크기가 너무 작아 실사용에 무리가 있었다.

2)ARTCAM-200MI

제품 사진	
이미지 센서	1/2" Color CMOS
화소수	200만 화소
전송속도	UXGA : 10 프레임/초 SXGA : 15 프레임/초 XGA : 21 프레임/초 VGA : 46 프레임/초
출력신호	USB 2.0

ARTCAM-200MI 카메라는 USB 2.0의 전송 속도를 이용하여 프레임 그라버 없이 고해상도의 영상을 입력 받을 수 있는 카메라이다. 1600x1200크기의 200만화소 영상을 전송 받을 때 초당 10장의 영상

을 전송 받을 수 있다. 하지만 프레임 그래버가 없다 보니 영상을 전송 받기 위해 CPU의 50%를 사용하게 된다. 이에 따라 영상을 처리하는 부분에서 상대적으로 사용 가능한 CPU의 대역폭이 줄어든다.

2. Axis 210

제품 사진	
이미지 센서	1/4" RGB CCD
화소수	630x480
전송속도	Motion JPEG : 30 프레임/초 MPEG-4 : 25 프레임/초
출력신호	HTTP

Axis 210카메라는 Network을 통하여 원격으로 영상을 전송 받을 수 있는 카메라이다. USB가 아닌 네트워크를 통하여 영상을 전송 받을 수 있기 때문에 PC와 다른 장소에 설치되어 있어도 충분히 사용이 가능하다. 하지만 PC와 카메라 사이에 영상을 전송 받을 수 있는 충분한 속도와 대역폭이 보장되어야 한다. ARTCAM 200MI에 비하여 사용 가능한 영상의 크기가 작다. 네트워크로 카메라를 컨트롤 할 수 있는 API를 제공하기 때문에 손쉽게 카메라를 컨트롤 할 수 있다.

3. Axis 214 PTZ

제품 사진	
이미지 센서	1/4" CCD
화소수	704x480
전송속도	Motion JPEG : 30 프레임/초 MPEG-4 : 21 프레임/초
출력신호	HTTP

Axis 214 PTZ카메라는 Pan-Tilt-Zoom이 가능한 카메라로 감시 영역에서 움직이는 물체를 추적하는데 용이하다. 네트워크를 통하여 감시 영역에 대한 영상을 획득할 수 있고 카메라를 컨트롤 할 수 있다. Axis 에서 제공하는 SDK를 통하여 Pan-Tilt-Zoom을 컨트롤 할 수 있지만 명령에 대한 반응 속도가 느리기 때문에 빠르게 물체를 추적하기는 힘들다.

제 2 절 서버(Server)

Sever모듈은 중앙 관제실에서 다수의 클라이언트를 컨트롤하기 위한 프로그램이다. Server모듈에서는 네트워크를 통해 다수의 클라이언트에 접속하고 해당 클라이언트에 PTZ 카메라가 연결되어 있는 경우 해당

PTZ 카메라에도 연결하여 PTZ 카메라의 영상을 표시한다. 접속한 클라이언트에서는 각 종 이벤트에 대한 정보와 실시간 Thumbnail 영상을 전송 받고 화면에 표시해준다. 각 클라이언트에 대한 접속 정보는 데이터베이스에서 관리하고 Client Data Management모듈을 이용하여 해당 데이터베이스에 접근한다. Sever의 전체적인 구성은 아래의 그림과 같다.

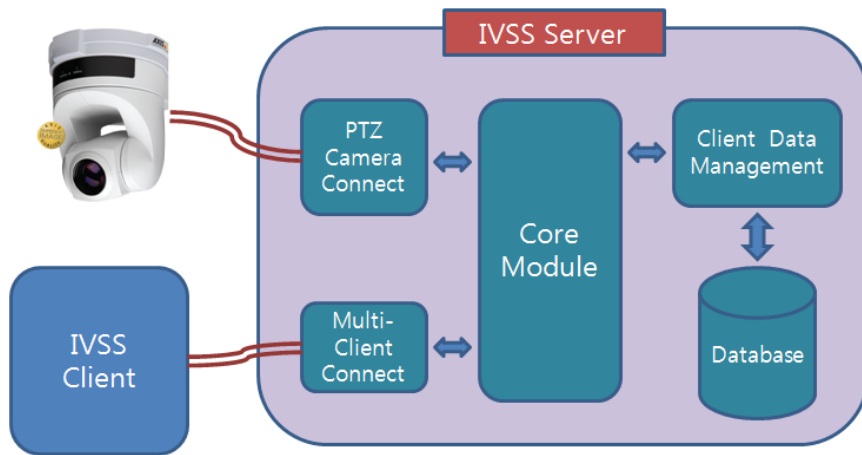


그림 10. 서버 구성도

서버 프로그램의 실제 실행 모습은 아래의 그림과 같다. 클라이언트의 접속 목록이 왼쪽에 트리 형태로 나오고 중앙에 PTZ 카메라 영상과 클라이언트의 Thumbnail 영상이 나온다. 아래에는 연결한 클라이언트들에서 전달받은 Thumbnail 목록이 나오고 각 클라이언트에서 전달받은 이벤트가 나타난다.

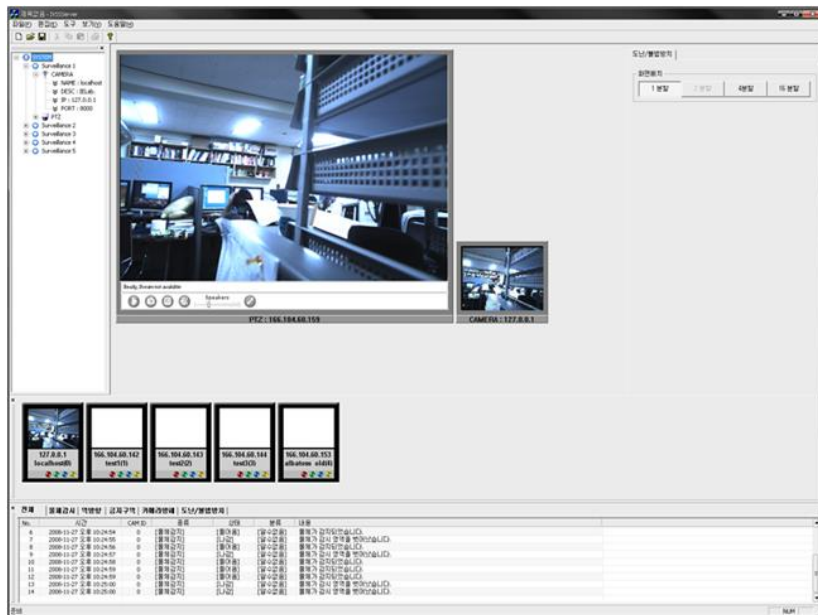


그림 11. 서버 사용자 화면

1. PTZ Camera Connect

클라이언트가 접속하고 있는 PTZ카메라에 접속하기 위한 모듈이다. 이 때 PTZ와 Server는 네트워크로 연결되어 있어야 한다. Axis SDK를 이용하여 접속한 PTZ카메라에서 실시간 영상 정보를 얻어 올 수 있다. 실제 PTZ카메라에 대한 컨트롤은 클라이언트에서만 가능하다.

2. Multi-Client Connect

하나의 서버에는 총 64개의 클라이언트를 동시에 연결할 수 있다. 각각의 클라이언트에서 보내오는 이벤트 정보와 Thumbnail 영상을 동시에 처리하기 위해 Thread를 이용하여 각각의 클라이언트를 관리한다. 클라이언트와의 연결이 끊겼을 경우 자동으로 해당 클라이언트와 관련

된 자원들을 정리한다. 이 모듈을 통하여 클라이언트와 연결을 생성하고 명령을 전달할 수 있다. 또한 명령에 대한 응답과 각종 정보를 전달 받는다.

3. Core

Core모듈에서는 전반적으로 시스템을 관리한다. 프로그램이 초기 기동할 때 Database에서 클라이언트 연결 정보를 얻어와서 각 클라이언트들과의 연결을 생성한다. 새로운 클라이언트와의 연결을 생성하는 기능을 하고 기존 클라이언트와의 연결이 끊어졌을 때 자원을 해제하는 기능도 한다. 클라이언트들을 배치하기 위한 layout매니저를 가지고 있어서 각각의 영상을 1,4,16,64 분할 할 수 있다.

4. Client Data Management & Database

클라이언트에서 필요로 하는 각종 정보들을 검색하거나 저장하기 위한 모듈이다. 데이터의 성질에 따라 외부 Database에 저장하거나 레지스트리에 저장한다. Database에 저장되는 데이터는 mdb를 이용하여 저장된다.

제 3 절 클라이언트(Client)

Client모듈은 USB카메라와 PTZ카메라에서 입력 받은 영상을 이용하여 물체를 추적하고 각종 이벤트를 판단하여 HTTP로 연결된 Server 모듈에 영상과 이벤트를 전달한다. 이 때 발생한 이벤트와 관련된 영상을 H.264로 인코딩하여 Media Server에 저장한다. Client의 전체적인 구성은 다음과 같다.

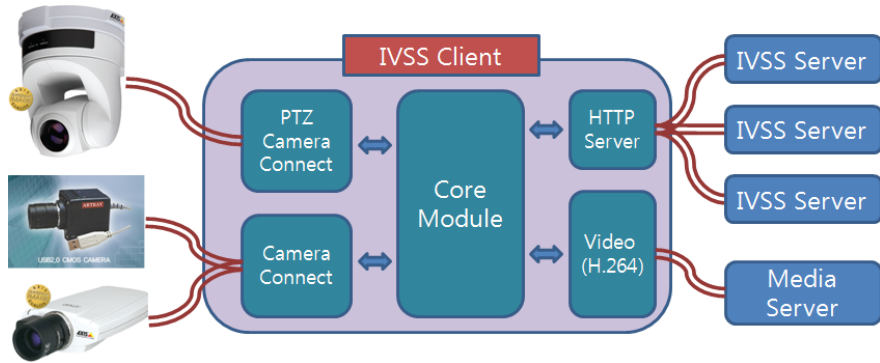


그림 12. 클라이언트 구성도

클라이언트의 실행화면은 아래의 그림과 같다. 연결된 고정 카메라의 영상이 클라이언트의 오른쪽에 나타나고 중앙에는 PTZ 카메라의 영상이 나타난다. 오른쪽 중앙에는 고정 카메라에서 입력 받은 영상을 이용하여 객체를 추적하는 영상이 나온다. 아래에는 발생하는 각종 이벤트를 표시하고 있다.

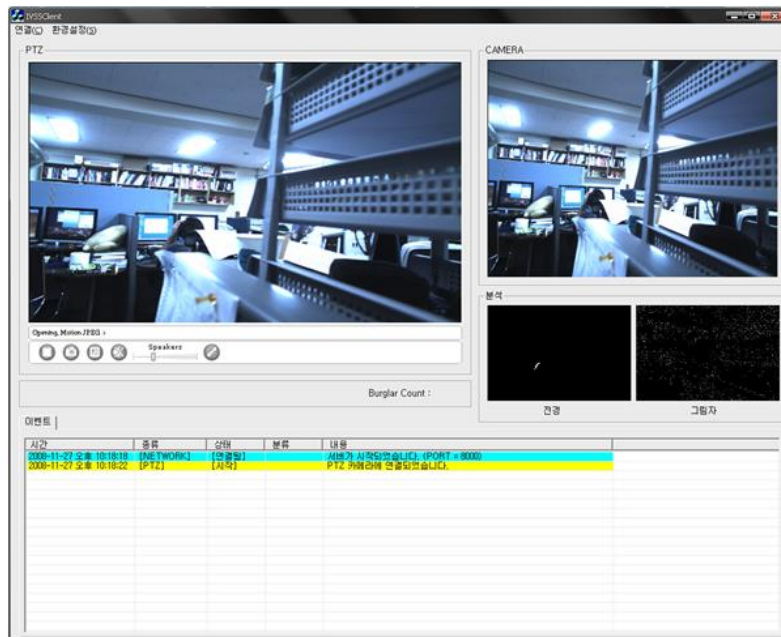


그림 13. 클라이언트 사용자 화면

1. Core Module

Core Module에서는 USB카메라에서 받은 영상을 분석하여 객체를 추적하고 각 종 이벤트 발생 여부를 판단한다. 또한 풀사이즈 영상을 메모리에 저장하고 있다가 이벤트 발생시에 H.264로 인코딩하여 Media서버로 전송한다. Thumbnail 크기의 작은 영상은 영상을 취득할 때마다 HTTP Server를 이용하여 각 Server모듈로 전송한다. 영상 내에서 관심 객체가 이동할 경우 PTZ카메라를 이동시켜 실시간으로 객체를 추적한다.

2. Camera Connect

카메라 연결 모듈은 고정된 카메라에서 취득한 영상을 읽어오고

Core Module에 전달하는 역할을 한다. 이 때 사용할 수 있는 카메라는 USB로 연결된 카메라와 네트워크로 연결된 카메라가 있다. 서버에 저장되어 있는 설정에 따라 둘 중 하나의 카메라에 연결하여 영상을 취득한다. 이 때 Thread를 이용하여 영상을 취득하기 때문에 Core Module과 별개로 작동하고 전체적인 시스템 성능이 향상된다.

3. PTZ Camera Connect

PTZ 카메라 연결 모듈은 Axis 214 PTZ 카메라에 연결하기 위한 모듈이다. 이 모듈에서는 PTZ 카메라에서 취득한 영상을 출력하는 역할 뿐만 아니라 Core module에서 명령을 전달받아 PTZ 카메라의 pan-tilt-zoom 명령을 처리한다. 또한 필요에 따라 PTZ 카메라의 영상을 취득한다.

4. HTTP Server

HTTP Server모듈은 여러 개의 IVSS Server에 대응하기 위해 멀티 Thread로 작동하는 웹 서버이다. 클라이언트 프로그램을 시작하면 8000번 포트로 HTTP Server가 시작한다. IVSS Server에서 연결 요청이 들어오면 새로운 Thread를 생성하여 요청을 처리한다. 이 연결을 통하여 실시간 Thumbnail을 보내고 이벤트가 발생하였을 때 이벤트 정보를 보낸다. IVSS Server와의 연결이 끊겼을 경우 자동으로 자원을 해제해준다.

5. Video(H.264)

USB카메라 모듈을 통하여 취득한 영상을 저장하기 위해 H.264 코덱을 이용하여 인코딩하고 Media 서버로 전송하는 모듈이다. H.264 코덱

은 공개 라이브러리인 x264를 이용하였다. 이 모듈은 Thread로 작동하여 다른 프로세스의 작업을 방해하지 않는다. H.264로 동영상 인코딩이 끝나면 Media서버로 동영상을 전달하여 저장한다.

제 4 절 미디어 서버(Media Server)

Media 서버는 각 클라이언트에서 이벤트가 발생하였을 때 녹화된 동영상을 저장하는 서버이다. 각 클라이언트에서는 H.264로 인코딩 된 동영상과 이벤트 정보를 Media Server로 전달한다. Media 서버에서는 Server의 요청에 따라 저장되어 있는 영상을 검색하여 전송해준다. Media 서버의 실행 영상은 아래의 그림과 같다. 오른쪽에는 저장되어 있는 영상의 목록이 나타나고 왼쪽 달력을 이용하여 범위 검색이 가능하다.

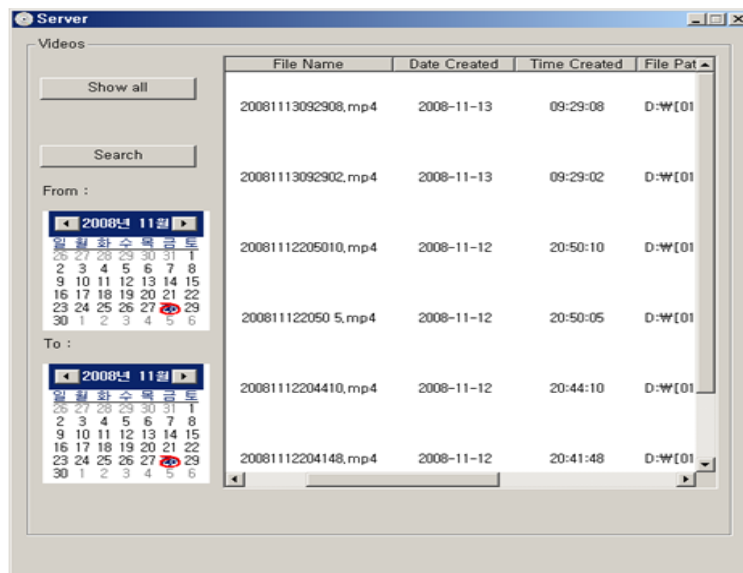


그림 14. 미디어 서버 사용자 화면

제 6 장 향후 연구 과제

본 프로젝트를 통해 개발된 지능형 영상 감시 시스템은 기존의 CCTV를 이용한 수동적 감시시스템 보다 효율적인 감시 기능을 제공한다고 할 수 있다.

그러나 외부 환경적인 요인에 따라 많은 종류의 잡음 때문에 몇 가지 시스템적인 한계를 가지고 있다. 본 시스템이 가지고 있는 한계점(문제점)을 표 3에 정리하였다.

표 3. 시스템 한계점

분류	문제점/내용
Tracking	Initialize Problem
	시스템이 동작하기 시작하는 시점에 배경을 생성하는 과정에서 초기에 움직이는 물체가 포함되어 있는 경우 초기 배경이 올바르게 생성되지 않음
	Similar Color Problem
	움직이는 물체가 배경과 색깔이 유사할 경우 물체를 추출해 내지 못함
	Small Object Problem
	카메라로부터 거리가 멀리 있는 물체의 경우 물체를 효율적으로 추출하지 못함
	Shadow Problem
	다양한 조명 환경에 의해 생기는 그림자를 효과적으로 제거하지 못하여 물체로 인식되고 있음

	Occluded Object Problem
	여러 개의 물체가 겹쳐져서 움직이는 경우 분리해서 추적하지 못함
	Camera Shaking Problem
	카메라가 외부 요인으로 인해 흔들림이 발생하는 경우 감시 효율이 떨어짐
금지구역	Invasion Scope Problem
	감시 영역을 필요 이상으로 넓게(또는 좁게) 설정할 경우 정확도가 떨어짐(오경보 발생률이 높아짐)
도난 불법방치	Object Movement Problem
	물체의 도난을 감시하는 경우 감시하는 물체가 감시 영역에서 움직이는 경우 도난으로 잘못 판단하는 경우가 있음
	Proper Area Problem
	감시하고자 하는 물체 보다 감시 영역을 필요 이상으로 넓게(또는 좁게) 설정할 경우 오경보 발생률이 높아짐

시스템의 문제점은 기능상의 분류를 기반으로 크게 세가지로 분류할 수 있다.

Tracking의 경우 본 시스템의 가장 기본이 되는 기능이다. 이 Tracking 기능의 경우 크게 여섯 가지의 문제점을 가지고 있다. 우선 Initialize Problem은 시스템이 초기에 동작할 때 배경을 생성하게 되는데 이 때 생성하는 배경은 시스템이 시작되는 시점으로부터 입력되는 Frame의 평균을 이용하여 생성하게 된다. 이때 초기에 입력되는 Frame에 움직이는 물체가 포함되어 있는 경우 움직이는 물체가 초기 배경으로 생성에 영향을 끼치게 되

며 이러한 잘못된 배경 생성이 초기에 물체를 검출함에 있어 Ghost 현상을 발생하게 된다. 즉 움직이는 물체가 배경으로 포함되어 이 부분에 계속된 차가 발생하여 물체가 있지도 않은 영역에 물체가 있는 것처럼 검출되게 된다.

두 번째로 Similar Color Problem의 경우 추적하고자 하는 물체의 색깔(픽셀 값)이 생성된 배경의 색깔(픽셀 값)과 유사한 경우 그 차이가 현저히 낮아서 물체를 추출할 수 없게 된다.

세 번째로 Small Object Problem은 카메라부터 현저히 멀리 있는 물체의 경우 크기가 너무 작아서 물체를 제대로 인식하지 못하게 된다. Shadow Problem은 카메라로부터 입력된 영상에 그림자가 포함되어 이를 물체로 인식함으로써 잘못된 추적을 발생하게 된다.

본 시스템에서 움직이는 물체라 함은 자의적으로나 타의적으로 인간이 인식함에 있어 분명히 움직임을 갖는 물체를 정의한다. 이 때 외부의 다양한 조명 환경 때문에 움직이는 물체에는 자연스럽게 그림자가 생기기 마련이며 그림자는 본 시스템과 같은 감시시스템에서 추출하고자 하는 감시 정보와는 아무런 상관이 없는 정보가 된다. 따라서 물체를 추출 추적함에 있어 이러한 그림자는 최대한 줄일 필요가 있다. 본 시스템에서도 그림자를 제거하는 알고리즘이 적용되어 있으나 다양한 조명 환경에 대처하지 못하는 단점이 있다. Occluded Object Problem은 입력되는 영상에서 여러 개의 물체가 겹쳐져 나타날 때 본 시스템은 이를 하나의 물체로 인식하고 추출 추적한다.

그러나 사용자는 이를 별개의 물체로 인식하길 원하고 있다. 따라서 겹쳐진 물체의 추적을 위한 보다 고급스런 알고리즘 개발이 필요하다. 마지막으로 Camera Shaking Problem은 시스템에 설치된 감시 카메라가 외부 요인에 의해 인위적으로 움직여 졌을 경우 입력되는 영상에 심한 왜곡이 발생한다. 이렇게 발생한 왜곡은 시스템에 잘못된 정보를 전달하여 감시 시스템이

올바로 동작하지 않게 된다. 따라서 카메라가 외부 요인에 의해 전체적으로 흔들림이 발생했음을 인지 이에 대한 보안을 위한 개발이 필요하다.

금지구역의 경우 단 하나의 문제점을 안고 있다. 사용자가 필요 이상으로 금지구역을 설정할 경우 금지구역의 침범을 인식함에 있어 그 정확도가 떨어지는 단점이 있다. 이는 알고리즘 상으로도 보완이 가능하겠지만 근본적으로는 사용자가 적정 크기의 감시 영역을 설정하는 것이 우선 되어야 한다.

도난 및 불법 방치 물 감시의 경우 금지구역과 마찬가지로 적정 크기의 영역 설정이 필요하다. 감시 하고자 하는 물체보다 필요 이상으로 큰 영역을 설정할 경우 영역 내에 변화 정보가 느리게 감지되어 감시 효율이 떨어지는 단점이 있다. 따라서 이 부분은 금지구역과 마찬가지로 적정 크기의 영역 설정이 전제되어야 하겠다. 도난 및 불법 방치 물 감시의 경우 또 다른 문제점을 한가지 가지고 있는데 Object Movement Problem이 그것이다. 감시 영역 내에 물체가 감시 영역 내에서 인위적으로 이동했을 경우 본 시스템에 적용된 알고리즘으로는 이 것이 도난으로 간주되고 있다. 그러나 실제로는 감시 영역에 존재 하므로 이는 도난으로 볼 수 없다. 따라서 도난 및 불법 방치 물 감시에 대한 기능도 이에 대한 보안 연구가 필요하다.

제 7 장 결 론

본 연구에서는 기능형 영상 감시 시스템 개발을 목표로 하고 이에 상응하는 알고리즘 및 시스템 개발을 수행하였다. 입력되는 영상의 평균값을 이용하여 배경을 생성하는 배경 모델링 기법을 이용하여 배경을 생성하고 이를 이용 현재 입력되는 영상과 차를 구하고 차 영상에서 움직이는 물체를 추출하였다. 추출된 물체는 Motion Vector Analysis 기법을 활용 물체의 경로를 추적하였고 물체의 이동 경로 정보를 이용하여 다양한 형태의 이벤트를 추출하였다.

본 연구에서 시스템 개발은 크게 세가지로 이루어 졌다. 첫 번째는 영상을 입력 받고 입력된 영상으로부터 알고리즘을 적용하는 클라이언트 부분이다. 클라이언트 부분은 실시간 감시를 위한 카메라와 영상처리 어플리케이션이 동작하는 workstation으로 구성되어 있다. 클라이언트에서 추출된 영상 정보는 보관 및 모니터링을 위해 서버로 전송되며 서버는 이 기능이 중심을 이루고 있다.

영상 감시 서버는 여러 대의 클라이언트를 동시에 접속할 수 있으며 각 접속된 카메라의 영상을 실시간으로 모니터링 할 수 있다. 뿐만 아니라 Media Server에 저장된 영상을 실시간으로 검색할 수 있고 검색된 결과를 재생할 수 있다. Media Server는 클라이언트에서 이벤트가 발생했을 경우 해당 구간에 대한 영상을 실시간으로 저장하는 시스템이다. H.264 기반으로 영상이 인코딩 되면 실시간으로 Media Server에 저장된다.

결론적으로 본 연구는 지능형 영상 감시 시스템을 개발함을 목적으로 했으며 이를 위한 알고리즘 개발 및 시스템 개발을 완료 하였다. 향후 효율적인 알고리즘을 보강을 통하여 시스템의 성능을 개선이 필요하다.

참고 문헌

- [1] Gary R. Bradski, "Computer Vision Face Tracking for Use in a Perceptual user Interface," Intel Technology Journal Q2, 1998.
- [2] F.J.Aherne, N.A.Thacker, and P.I.Rockett, "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data," Kybernetika, Vol.32, No.4, pp.1-7, 1997.
- [3] D. Comaniciu, V. Ramesh, "Robust Detection and Tracking of Human Faces with an Active Camera", IEEE Int. Workshop on Visual Surveillance, Dublin, Ireland, 11-18, 2000.
- [4] D. Comaniciu, P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis", IEEE Trans. Pattern Analysis Machine Intell., Vol. 24, No. 5, 603-619, 2002.
- [5] Don Murray, Anup Basu, "Motion Tracking with an Active Camera", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16, no.5 May 1994
- [6] D. Comaniciu, V. Ramesh, P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, South Carolina, Vol. 2, 142-149, 2000
- [7] Alper Yilmaz, Khurram Shafique, "Target Tracking in FLIR Imagery Using Mean-Shift and Global Motion Compensation," proceedings of IEEE Workshop on Computer Vision Beyond Visible Spectrum, Hawaii, 2001.
- [8] Michael Isard and Andrew Blake, "CONDENSATION - conditional

- density propagation for visual tracking " Int. J. Computer Vision, 29, 1, 5-28, (1998)
- [9] Michael Isard and Andrew Blake, "ICondensation: Unifying low-level and high-level tracking in a stochastic framework" Proc 5th European Conf. Computer Vision, 1998
 - [10] David Reynard, Andrew Wildenberg, Andrew Blake, "Learning Dynamics of Complex Motions from Image Sequences" In Proc. European Conf. Computer Vision 1996. pp357-368
 - [11] Michael Isard and Andrew Blake, "Contour tracking by stochastic propagation of conditional density" Proc European Conf. Computer Vision, 1996, pp.343-356.
 - [12] Michael Isard and Andrew Blake, "A smoothing filter for Condensation" Proc 5th European Conf. Computer Vision, 1998
 - [13] Greg Welch and Gary Bishop, " An Introduction to the Kalman Filter," Department of Computer Science University of North Carolina at Chapel Hill., TR 95-041, March 11, 2002
 - [14] J.S.Lee, K.Y.Rhee, S.D.Kim, "Moving Target Algorithm based on the confidence measure of motion vectors", International Conference on Image Processing(ICIP) 2001, Thessaloniki, Greece, pp.369~372
 - [15] Jason Z. Zhang, Q. M. Jonathan Wu, "A Pyramid Approach to Motion Tracking", Real-Time Imaging, Vol. 7, No. 6, Dec 2001, pp. 529-544
 - [16] K.-P. Karmann, A. von Brandt, "Moving Object Recognition using an Adaptive Background memory,"in V. Cappellini (ed), Time-

- varying image processing and moving object recognition, 2, pp.297-307, Elsevier Pub. B.V., Amsterdam, 1990.
- [17] Stephen S. Intille, James W. Davis, and Aaron F. Bobick, "Real-Time Closed-World Tracking" Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 697-703, June 1997.
 - [18] Naohiro Amamoto and Akihiro Fujii, "Detecting Obstructions and Tracking Moving Objects by Image Processing Technique" , Electronics and Communications in Japan, Part 3, Vol. 82, No. 11, 1999
 - [19] S. Huwer and H. Niemann, "2D-Objecttracking based on Projection-Histograms" 5th European Conference on Computer Vision- ECCV 98, Freiburg, Germany Springer Publishers, Lecture Notes in Computer Science 1406, Vol 1, p. 861-876
 - [20] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen. "An Image Processing System for Driver Assistance". Image and Vision Computing (Elsevier) , 18(5):367-376, 2000.
 - [21] Stan Birchfield, "An Elliptical Head Tracker". Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, pages 1710-1714, November 1997
 - [22] Stan Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, p. 232-237, June 1998

- [23] S. McKenna, S. Gong, "Tracking Faces," Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Page(s): 271-276, 1996.
- [24] Vu,V.T., Bremond,F., Thonnat,M. "Human Behaviour Visualisation and Simulation for Automatic Video Understanding", WSCG, p.485-492, February 2002
- [25] Sminchisescu,C., Telea,A. "Human Pose Estimation from Silhouettes - A Consistent Approach Using Distance Level Sets". WSCG p.413-420. February, 2002
- [26] Daniel P. Huttenlocher, Gregory A. Klanderman, William J. Rucklidge, "Comparing Images Using the Hausdorff Distance", IEEE Transactions on Pattern Analysis and Machine Intelligence (1993)
- [27] Ce Wang, Michael S. Brandstein, "A Hybrid Real-Time Face Tracking System" Proc. ICASSP'98, 1998.
- [28] C.S.Kim, J.B.Lee and S.D.Kim, "A Novel Global Motion Estimation, Compensation for Motion Compensated Coding", Int. Picture Coding Symposium '96, P. 595 - 598, 1996, 3
- [29] Christof Ridder, Olaf Munkelt, Harald Kirchner, " Adaptive Background Estimation and Foreground Detection using Kalman-Filtering " , in Proceedings of International Conference on recent Advances in Mechatronics, ICRAM (1995), O. Kaynak, M. Özkan, N. Bekiroglu, I. Tunay (Editors), UNESCO Chair on Mechatronics, 193-199
- [30] Rita Cucchiara, Massimo Piccardi, Paola Mello, "Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System", IEEE

TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 1, NO. 2, JUNE 2000

- [31] Stephen S.Intille, Aaron F. Bobick, "Visual tracking using Closed-worlds", M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 294, November 1994
- [32] O.-K. Kwon, D.-G. Sim, and R.-H. Park, " Robust Hausdorff distance matching algorithms using pyramidal structures " , Pattern Recognition, vol. 34, no. 10, pp. 2005-2013, Oct. 2001.
- [33] Stefan Huwer , Heinrich Niemann, "3D Modelbased Detection and Tracking of People in Monocular Video Sequences", IASTED Conference on Signal and Image Processing, Las Vegas, Nevada, USA, November 2000, pp.172-177
- [34] Clark F. Olson and Daniel P.Huttenlocher, "Automatic Target Recognition by Matching Oriented Edge Pixels", IEEE Transactions on Image Processing, 6(1): 103-113, January 1997.
- [35] Clark F. Olson, "A Probabilistic Formulation for Hausdorff Matching", Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition , p.150-156, 1998
- [36] K.Daniilidis, Ch.Krauss, M.Hansen, G.Sommer, "Real Time Tracking of Moving Objects with Active Camera", REAL TIME IMAGING, 1997
- [37] Bernd Heisele, Thomas Serre & Sam Prentice, "Real-Time Face Detection", MIT AI Lab. <http://www.ai.mit.edu>
- [38] Margrit Betke, Esin Haritaoglu, Larry S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle" Machine Vision and Application, Vol.12, p.69-83, 2000.

- [39] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning" , In Proc. of ECCV 94, Stockholm, Sweden, pages 189-196, 1994.
- [40] Chris Stauffer & Eric Grimson, "Similarity Templates for Detection and Recognition", MIT AI Lab. <http://www.ai.mit.edu>
- [41] Stefan Huwer, Heinrich Niemann, "3D Modelbased Detection and Tracking of People in Monocular Video Sequences", Proc. of the IASTED Int. Conf. Signal and Image Processing, Nov. 19-23 2000, Las Vegas, Nevada, USA
- [42] Rafael Murrieta-Cid, "target tracking method based on a comparison between an image and a model", <http://underdog.stanford.edu/tmr/>
- [43] Chris Stauffer & Eric Grimson, "Tracking-Based Automatic Object Recognition", MIT AI Lab.
- [44] D R Magee, "Tracking Multiple Vehicles using Foreground, Background and Motion Models", University of Leeds, SCHOOL OF COMPUTER STUDIES RESEARCH REPORT SERIES Report, 2001.21, Submitted to European Conference on Computer Vision, May 2002
- [45] Y.K. Jung and Y.S. Ho, "Traffic Parameter Extraction using Video-based Vehicle Tracking", IEEE Int'l Conference on ITS, pp. 764-769, Oct. 1999.
- [46] L. Di Stefano, E. Viarani. "Vehicle Detection and Tracking Using the Block Matching Algorithm", Proc. of "3rd IMACS/IEEE Int'l Multiconference on Circuits, Systems, Communications and Computer" Athens, Greece, Vol. 1, pp. 4491-4496, July 4-8, 1999,

available only on cd-rom.

- [47] Gian Luca Foresti, Vittorio Murino, Carlo Regazzoni, "Vehicle Recognition and Tracking from Road Image Sequences", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 48, NO. 1, JANUARY 1999
- [48] Stefan Huwer, Heinrich Niemann, "Adaptive Change Detection for Real-Time Surveillance Applications", Third IEEE International Workshop on Visual Surveillance - VS-2000, Dublin, Ireland, July 2000, pp. 37-45
- [49] 장기정, 권인소, "적응 칼라 모델을 이용한 실시간 얼굴 추적", 제 12회 영상처리 및 이해에 관한 워크샵, 2000
- [50] 강동구, 나종범, "차영상을 사용한 자동 움직임 검출 알고리즘" 제15회 신호 처리 합동 학술 대회 논문집, p.47-50, 2000.
- [51] 이철훈, 김형태, 설성욱, 남기곤, 이장명, "칼만 필터를 이용한 다중 차량 추적 알고리즘", 전자공학회 논문집 제 36권, p.89-95, 1999
- [52] 설성욱, 이희봉, 김효성, 남기곤, 이철현, " 히스토그램 프로젝션을 이용한 움직이는 카메라로부터의 이동물체 추적 알고리즘", 신호처리 시스템 학회 논문지 제 2권 4호 p.38-44, 2001
- [53] 장국렬, 박순영, "지능화된 무인감시시스템을 위한 침입자 추적 알고리즘에 관한 연구", 목포대학교 정보산업연구지 제5집 p.11-26, 1997
- [54] 이상욱, 설성욱, 남기곤, 권태하, "특징 기반 다중 물체 추적 시스템에 관한 연구," 대한전자공학회논문지, vol. 36-S, no. 11, pp. 95-101, 1999. 11.
- [55] 이종원, 김옥중, 김성대, 이상지, "새로운 전역 움직임 보상 판단 기법", 제9회 신호처리합동학술대회 논문집 제9권 1호, 1996

- [56] J.S.Lee and H.C.Lee and S.D.Kim, "Target Detection and Tracking Algorithm Using Moving Region Detection in FLIR Sequences", THE ACOUSTICAL SOCIETY OF KOREA, 185-189, 1999, 10
- [57] H.C.Choi, M.S.Shin, J.S.Lee and S.D.Kim, "Target Extraction Algorithm in infrared image using Model information and edge linking", , THE ACOUSTICAL SOCIETY OF KOREA, 229-232, 1999, 10
- [58] H.C.Shin, H.C.Choi, J.S.Lee, J.H.Cho and S.D.Kim, " Target Recognition with Intensity-Boundary Features " , Proceeding of the Korean Signal Processing, 411-414, 2001, 9
- [59] J.H.Cho, H.C.Choi, J.S.Lee, H.C.Shin and S.D.Kim, " Object Region Detection using Multi-Sensor Fusion and Background Estimation " , roceeding of the Korean Signal Processing, 443-446, 2001, 9
- [60] H.C.Choi, J.S.Lee, H.C.Shin, J.H.Cho and S.D.Kim, " Boundary Extraction Using Statistical Edge and Curvature Model " , Proceeding of the Korean Signal Processing, 443-446, 2001, 9
- [61] 김휘용, 김성대, " 화상감시를 위한 이동물체 추출 기법, " 2002년 제 14회 영상처리 및 이해를 위한 워크샵 발표 논문집, pp.37-42, 2002, 1
- [62] 박수형, 염동훈, 고기영, 김두영, " 레벨 세트와 히스토그램을 이용한 이동물체의 추적, " 2002년도 대한전자공학회 하계종합학술대회 논문집 제25권 1호, pp.137-140, 2002.
- [63] 오종안, 정영기, " 능동카메라 환경에서의 특징기반의 이동물체 추적, " 2002년도 대한전자공학회 하계종합학술대회 논문집 제25권 1호, pp.141-144, 2002.

- [64] 류성희, "무인 영상 감시를 위한 이동 물체 검출 및 추적 시스템의 실시간 구현에 관한 연구", 수원대학교 석사학위논문, 2001