

# 다중물체추적을 이용한 유동인구 행태 분석

\*최규형      \*\*최영주      \*\*\*정지홍      \*\*\*\*\*서용덕  
\*서강대학교   \*\*서강대학교   \*\*\*국민대학교   \*\*\*\*\*서강대학교  
\*kyu@sogang.ac.kr

## Population Movement Analysis Using Visual Object Tracking

\*Choi, Kyuhyoung      \*\*Choi, Youngju      \*\*\*Jung, Ji hong      \*\*\*\*\*Seo, Yongduck  
\*Sogang University   \*\*Sogang University   \*\*\*Kookmin University   \*\*\*\*\*Sogang University

### 요약

비디오에서의 물체 추적은 컴퓨터비전(computer vision)의 주요 연구 분야로 지능형 로봇, 무인 감시 체제 등의 영역의 핵심 기술로 여겨지고 있다. 본 논문에서는 다중물체추적을 통해 카메라로 부터 입력된 동영상에서 특정 장소를 지나가는 사람들을 추적함으로써, 그 지역에서의 인구의 이동 패턴을 추출하고자 한다. 물체 추적은 블롭 추적(blob tracking) 방식을 이용하며, 이를 위해 정확한 전경물체 추출, 추출된 이미지 블롭(blob)과 기존 트랙과의 연결, 새로운 물체(사람)의 등장과 퇴장의 작업을 수행한다. 추적된 물체들이 궤적을 통해, 시간의 변화에 따른 그 지역에서의 인구의 밀도, 주 이동 경로, 방향 등의 변화를 추출한다. 이러한 통계치는 해당 지역의 개발 정책 수립 및 시장성 조사를 위한 2차 데이터로 활용할 수 있다.

### 1. 서론

특정지역에 대해, 시간에 따른 인구의 변화를 분석하는 것은 학문적으로도 가치 있는 일이지만, 산업적으로도 그 지역에 새로운 사업 및 개발을 수행하기 위한 사전 작업으로써 수행되어야 하는 일이다. 도시 지역의 대부분의 시설들의 규모는 그 지역의 인구수에 따라 결정되기 때문이다. 하지만 일일이 사람들의 움직임, 인구수의 변화, 주 이동 경로, 밀집도의 변화 등을 관찰하고 파악하는 일은 쉽지 않을 뿐만 아니라 다분히 시간 소모적이다. 게다가 장기간에 대한 분석이 이루어져야 할 경우에는 더욱더 작업이 어려워진다. 이를 위해 본 논문에서는 단순히 원하는 지역에 설치된 카메라로 부터 장기간 동안 촬영한 비디오 입력을 받아, 처리하여 자동으로 유동인구의 행태를 분석함으로써 쉽게 원하는 결과를 얻는 방법을 제안하고자 한다.

이러한 인구의 변화는 결국 해당지역의 모습을 담은 동영상에서 사람들을 찾고, 찾은 사람들을 추적함으로써 알아낼 수 있다. 물론 삼차원 공간상에서의 추적이 가능하다면 이상적이겠지만, 2차원 이미지 상에서의 추적도 인구수의 변화, 주 이동 경로 등을 파악하기에 충분한 데이터를 제공해 준다. 이러한 동영상에 대한 처리를 통해, 영상 내의 이미지에 대한 정보를 얻어내는 것이 결국 컴퓨터비전(computer vision)이 추구하는 목표이기 때문에, 위의 문제는 컴퓨터비전의 주된 연구주제 들이기도 하다. 한 장 또는 여러 장의 이미지 내에서 특정한 물체를 찾는 것은 물체인식(recognition)으로서, 무인 보안 시스템과 관련하여 주로 사람의 얼굴에 대한 인식이 연구 분야로서 많은 관심을 받아왔다.

동영상 내에서 지정된 물체의 움직임을 알아내는 것, 즉 좌표의 변화를 알아내는 것이 물체추적(target tracking)으로서, 이 또한, 영상기반 무인 감시 시스템, 로봇 주행 등의 구현에 있어서 필수적인 기술 분야이다. 일반적인 영상에서 하나가 아닌 여러, 그것도 서로 상호작용하거나, 시각적으로 겹치는 물체들을 추적하기란 쉬운 일이 아니다.[1]

다시 말해서, 추적 대상의 움직임이나 외양의 변화를 모델링할 수 있는 일반적인 방법이 없다는 것이다. 삼차원 공간상에서 추적한다면, 관성의 법칙, 속도 가속도의 법칙, 탄성의 법칙, 빛 반사 모델, 조명 모델 등을 적용해서 어느 정도 단순한 움직임을 보이는 물체를 모델링하겠지만, 이차원 이미지 상에서는 연속된 아주 짧은 기간이 아니고서는 위의 법칙 등을 적용하는 것이 별다른 의미가 없다. 물체추적에서 가장 힘든 부분은 다른 비슷한 물체(clutter)의 등장이다. 결국 헛갈리지 않고, 초기에 지정해준 대상을 쫓아가는 일은 특별한 처리를 하지 않고서는 불가능하다. 예를 들어 추적 대상의 특징 정보를 초기에 주고 추적하는 경우에, 시간이 흘러감에 따라 대상물체는 초기와는 다른 특징을 나타내고, 다른 비슷한 물체가 나타나서, 점점 주어진 특징 정보를 나타낸다면, 컴퓨터는 당연히 원래 물체가 아닌 다른 물체를 쫓아갈 수밖에 없는 것이다.

본 논문의 경우와 같이, 어떤 지역에서의 사람들을 추적한다고 하면 카메라는 어느 정도 넓은 영역을 담기위해서 장면에서 먼 곳에 설치되거나 줌아웃(zoom-out)을 해야 한다. 이럴 경우 대부분의 사람들은 가로 세로의 비율이 1 : 4 정도의 외양이 조금씩 변화하는 물체(blob)일 뿐, 상하의를 특정한 컬러의 옷으로 입지 않고 서는, 이 사람과 저 사람이 다르다 라고 말할 수 있는, 즉, 서로 겹쳤을 때, 확실히 구분해서 분리시켜 생각할 수 있는 여지가 거의 없다. 또한 영상신호 취득 과정에서의 노이즈로 인해, 동일한 물체가 이전 프레임에서의 관측 정보, 예를 들어 색이나 모양과 다른 특징을 보이거나, 아예 신호가 잡히지 않을 수도 있다. 즉, 입력 신호를 100% 신뢰할 수 없기 때문에, 필터링(filtering)이 필요하고, 사전지식(prior knowledge)과 맞물려서, 현재의 상태, 즉, 물체의 위치를 추정하는 것이다.

근래에 들어, 영상처리나 컴퓨터비전에 대한 학계나 산업계 뿐만 아니라 대중들의 관심이 커짐에 따라, 컴퓨터비전 분야에서 자주 쓰이는 알고리즘이나 오피레이션 등을, 컴퓨터 프로그래밍시 함수로 가져다 쓸 수 있게 만든 라이브러리 등이 등장하게 되었다. OpenCV가 [3]

그 중 하나인데, 최근 릴리즈(release)된 버전에 보면, 영상기반 감시시스템(visual surveillance) 구현을 위한 라이브러리가 추가되었음을 볼 수 있다.[7] 고정된 카메라, 즉, 고정된 배경을 가정했을 때, 사람을 하나의 이미지 블롭으로 보고 그러한 블롭들의 움직임의 일관성으로부터 사람의 궤적(trajecory)을 구하는 비교적 간단하면서도 실용적인 툴(tool)이다. 본 논문에서는 이러한 OpenCV 라이브러리를 이용하여 비교적 넓은 지역의 사람들을 추적하고, 추적 결과를 바탕으로 유동인구의 변화에 관련된 통계치들을 구하고자 한다.

2장에서는 이 작업의 전체적인 흐름을 보이고 3장에서는 실제로 사람 및 차량을 추적하는 방법을 설명한다. 마지막으로 4장에서는 이렇게 추적된 움직이는 물체에 대한 데이터를 분석하는 방법 및 분석 결과를 보인다.

## 2. 물체추적

### 가. 입력영상

일단 분석하고자 하는 지역에 카메라를 설치하는 작업을 한다. 본 논문의 목적에 적합한 입력영상을 만들기 위해, 이론적으로 가장 적합한 카메라의 위치는 머리 위 하늘이다. 즉 대상지역에 수직으로 공중에 카메라를 매달아 아래로 쳐다 보게 하는 것이다. 이렇게 되면 겹침 문제가 현저하게 줄어들게 된다. 즉, 사람들 사이에 완전한 신체적 접촉(포옹이나 어깨동무)이 일어나지 않는 이상, 대부분의 경우 사람들은 독립된 이미지 블롭으로 검출되게 된다. 현실적으로 이러한 카메라 위치 설정은 불가능하므로, 되도록 높은 곳, 즉, 고층 건물에 설치한다. 분석하려는 지역이 넓어 하나의 카메라로 수용을 할 수 없을 경우 두 대 이상의 카메라로 카메라 네트워크를 구성해, 전체적인 부분이 다 보일 수 있도록 설치하면 된다. 실제 촬영 시에 카메라의 위치는 12층의 빌딩 옥상에서 거리를 바라보게끔 하였다. 그리고 촬영 시간은 약 70분 정도의 분량으로 촬영을 수행하였다.

### 나. 블롭추적(Blob Tracking)

다음으로 수행되는 단계는 이렇게 얻은 동영상 데이터를 바탕으로 움직이는 물체를 추적하는 것이다. 본 논문에서는 움직이는 유동인구로 자동차와 오토바이 등의 교통수단을 한 사람의 데이터라 가정하고 추적을 수행 하였다. 만약 오토바이와 자동차의 데이터를 제외한 실제 거리를 걷는 사람에 대한 데이터만을 얻고자 한다고 하면 오토바이와 자동차에 대한 데이터를 제외하고 얻은 추적결과 만을 얻을 수도 있다.

추적 시 주어지는 입력 데이터는 한 시간 이상의 DV 파일로 된 동영상이다. 동영상을 읽는 작업은 FFmpeg 라이브러리를 사용하였다.[6] FFmpeg은 리눅스(linux)기반의 동영상 및 이미지를 처리할 수 있게끔 제공하는 라이브러리로 OpenCV에서 제공하는 동영상 디코딩에 비해 좀더 다양한 포맷(format)의 동영상을 읽을 수 있게 제공 하며 동영상을 control하는데 좀더 안정적으로 작업이 수행되도록 하고 있다. 특히 카메라로 촬영된 동영상의 경우 DV 포맷의 파일이 대부분인데 이 파일을 입출력 할 수 있게 함수를 제공하고 있다.

추적은 OpenCV에서 제공하는 함수 "blobtrack"을 사용하여 이루어지며, 블롭 추적의 과정은 그림 1과 같이 5개의 모듈(module)로 이루어져있다. 동영상에서 얻은 각 프레임(frame)을 순차적으로 입력하여 주변 결과로서 해당 프레임상의 각 블롭 들의 ID와 위치(Pos), 크기(Size)를 얻게 된다. 각 단계별 수행 순서는 아래와 같다.

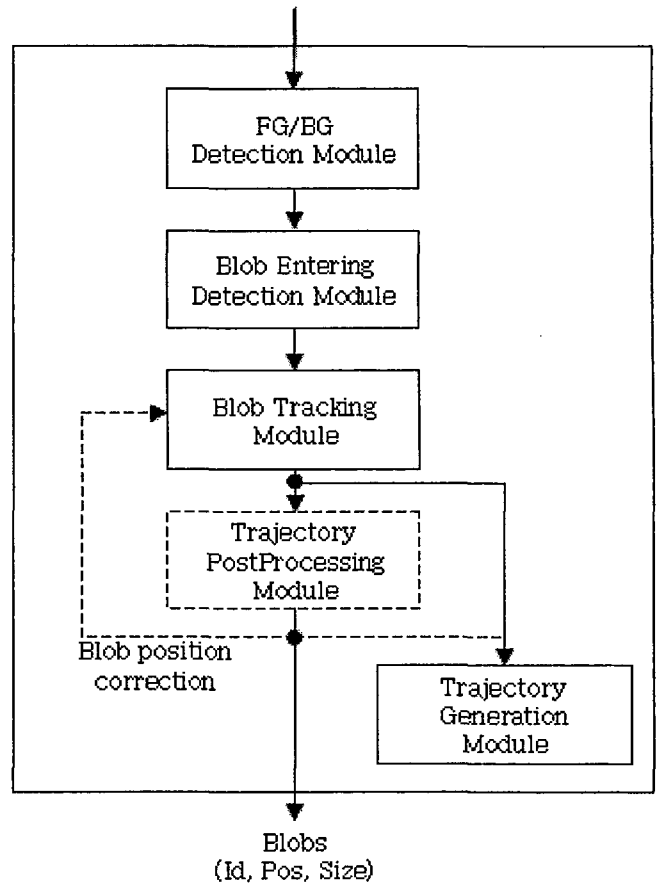


그림 1. 블롭추적 처리과정

"FG/BG Detection" 모듈은 각 픽셀(pixel)에 대한 전경 / 배경 분리(foreground / background segmentation) 을 수행한다. 여기에는 세 가지 옵션(option)이 있는데, 하나(FG\_0)는 [2]에 소개되었던 방법으로 각 픽셀에 대해 전경일 확률을 구해, 이것이 0.5보다 크면 전경, 작으면 배경으로 처리하는 방법이다. 다른 하나(FG\_1)는 [4, 5]의 방법으로 각 픽셀마다 프레임의 진행에 따라 입력되는 색상정보로 컬러 히스토그램을 만들고, 이것을 가우시안 혼합(Gaussian mixture)로 모델링 하여, 주성분(major element)들에 해당하는 픽셀을 배경, 나머지는 전경으로 처리하는 것이다. 결국, 이미지에서 배경은 움직이지 않거나, 조금씩 흔들거리는 것에 반해, 전경은 여기 저기 움직인다는 것을 고려한 분리 방법이다.

"Blob Entering Detection" 모듈은 "FG/BG Detection" 모듈의 결과를 이용하여 각 프레임에 들어가는 새로운 블롭을 검출(detection)한다. 즉, 이전 프레임에서 검출되었던 블롭들로부터 근치가 아닌 곳에서 현재 프레임의 블롭이 검출되었다면 이것은 화면으로의 새로운 사람의 등장이거나, 가려졌던 사람의 출현이다. "FG/BG Detection" 으로부터의 이진화된 영상에 CCL(connected component labelling)을 하여 블롭들의 리스트로 만든다.

"Blob Tracking" 모듈은 "Blob Entering Detection" 결과들에 의해 초기화되며, 이를 통해 이전 프레임까지의 추적결과에 현재 프레임의 블롭을 할당(association)시켜 궤적을 이어 나가고, 새롭게 삼입되는 블롭들에 대해서는 궤적을 초기화 한다. 여기에는 여러 가지 옵션이 있는데, 첫째(MS1)가 Mean Shift 알고리즘을 이용한 방식으로 컬러히스토그램의 모드(mode)를 추적하는 방식이다. 둘째(MS2)는 MS1

와 같은 방식이나 스케일(scale)에 대해서도 고려하는 방식이다. 세 번째 옵션(CC)은 connected component 분석을 기반으로 한 것으로 현재 프레임의 블롭이미지와 이전 프레임의 블롭이미지를 겹쳐 보았을 때, 어느 두 블롭이 하나의 connected component로 형성되면 동일한 물체라고 간주하는 간단한 방식이다. 네 번째 방식(PF)는 파티클 필터(particle filter)를 이용한 방식으로, 일정한 수의 샘플들을 가지고 추적대상의 이미지 상의 확률분포를 근사화하는 방법이다. 확률적인 방식이라 정확도가 높은 반면에 시간이 오래 걸린다.

“Trajectory Generation” 모듈은 궤적의 저장을 수행한다. 이것은 모든 블롭들의 위치(Pos)를 모으고, 각 블롭들에 대해 한 blob tracking이 끝나는 지점에서 그 blob의 trajectory를 hard disk에 저장한다.

“Trajectory PostProcessing” 모듈은 블롭 궤적에 대한 smoothing을 수행한다. 이는 Kalman smoothing을 이용하여 이루어지며, 위치(Pos)와 크기(Size)에 대해 Kalman 필터링을 한다. 이 모듈은 선택사항이며 꼭 이 수행과정에 포함시킬 필요는 없다.

이렇게 각 모듈의 옵션 설정에 따라 블롭추적의 결과도 조금씩 다른 양상을 보인다. 본 논문에서는 “FG/BG Detection”을 “FG\_1”으로 설정하였으며, “Blob Entering Detection”은 “BD\_Simple”, “Blob Tracking”은 “PF” 방식, 그리고, “Trajectory PostProcessing”에는 “Kalman filtering”을 적용하여 수행하였다.

이렇게 블롭추적을 수행하고 나면 우리는 연속된 프레임별로 ID가 부여된 블롭들의 크기와 위치 데이터를 얻게 된다. 이렇게 얻게 되는 블롭들은 이전 프레임에서 존재하는 ID일 경우 동일한 물체의 움직임으로 간주되어 동일한 궤적(trajectory)을 갖게 된다. 만약 전 프레임에 존재하지 않는 ID의 블롭일 경우에는 새로운 궤적을 설정해주게 된다. 이런 식으로 연속된 프레임 간에 존재하는 모든 블롭들의 궤적을 구성하게 된다. 이로써 움직이는 물체가 동영상에 나타나는 순간부터 사라지는 순간까지, 즉 그 블롭이 움직인 전체적인 동선을 파악할 수 있게 된다. 또한 모든 블롭들의 ID설정에 따라 얼마나 많은 수의 움직임을 갖는 객체들이 존재했는지를 파악할 수 있게 된다. 이를 통해 우리는 촬영된 지역 유동인구의 형태분석을 수행할 수 있다. 좀 더 구체적인 결과 데이터 분석 과정은 다음 장에서 논한다. 그림 4는 입력 데이터와 그것을 블롭추적한 전경(foreground) 동영상과, 원본동영상에 추적 결과를 입력한 동영상 중 몇 장의 이미지들이다.

### 3. 데이터 분석

블롭추적 수행 후 우리는 각 프레임별로 ID가 부여된 블롭들의 크기와 위치 데이터를 얻게 되고, 이는 궤적테이블(trajectory table)을 구성하는 요소가 된다. 궤적테이블은 동적 테이블로써 매 블롭들이 검출될 때마다 이 블롭이 기존의 테이블에 존재하는 ID를 갖고 있는지 그렇지 않으면 새롭게 생성된 ID의 것지를 파악한다. 만약 기존에 존재하는 ID의 블롭이라면 기존에 만들어진 궤적에 새롭게 공간을 생성하여 삽입시킨다. 새롭게 생성된 블롭일 경우엔 새로운 궤적을 만들어 준다. 이렇게 동일한 물체에 대한 궤적을 만들어 가면서 전체 블롭들의 모든 궤적을 구성해 간다. 결국 모든 프레임의 블롭추적 수행 후에 우리는 궤적의 개수 또는 ID의 번호로 실제로 얼마나 많은 수의 움직이는 물체가 존재했는지를 파악할 수 있으며, 매 프레임별 블롭들의 위치와 데이터를 통해 어떤 동선을 그리며 움직였는지 알 수 있게 된다. 또한 블롭 크기를 통해 이 물체가 사람인지 아니면 자동차인지를 파악하고 움직임 속도를 통해 사람과 교통수단(오토바이, 자가용 등)과의 구

분을 수행할 수 있게 된다. 물론 프레임별로 넘버링(numbering)이 수행됨으로 시간에 대한 정보도 얻을 수 있게 된다.

이러한 결과 데이터를 통해 여러 행태의 유동인구 형태 분석을 수행할 수 있다. 먼저, 그림 2에서 보는 바와 같이 시간대별 인구의 변화량을 파악할 수 있다.

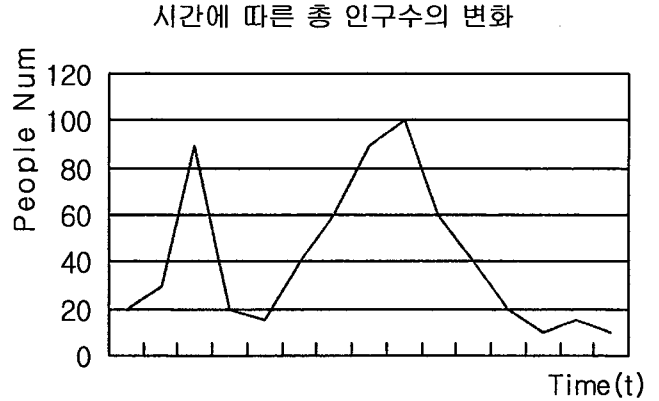


그림 2. 시간에 따른 총 인구 수의 변화 플롯

또한 그림 3처럼 인구 밀도의 히스토그램(Histogram)을 통해 특정 시간대의 유동인구의 동선을 파악할 수 있으며 그와 동시에 어느 지역으로 인구의 흐름이 더 많이 생기는지를 파악할 수 있다.



Min Value = 1

Max Value = 12714

그림 3. 특정 시간 동안의 유동인구의 흐름 분포도

좀더 깨끗한 빨강색상(255, 0, 0)으로 갈수록 이 시간대에 가장 많은 인구의 흐름이 있었던 곳을 말하며 순수파랑(0, 0, 255)으로 갈수록 가장 적은 인구의 수를 말하고 있다. 본 논문에서는 실험 결과 사람이 수가 많았던 지역에 비해 상대적으로 적은 부분이 넓게 분포하면서 그 수가 너무 적어 로그 변환(Log Transformation)을 이용해서 분포도를 조정하였다.

$$\text{value} = 255 * \log(\text{Num} + 1) / \log(\text{Max} + 1)$$

그 외에도 시간에 따른 인구 분포 현상을 파악할 수 있으며 인구의 출입과 퇴장과 움직임의 방향성 등도 함께 고려할 수 있는 데이터를 얻을 수 있다. 이러한 여러 데이터 분석을 통해 유동인구 형태 분석을 편리하고 빠르게 수행할 수 있다

#### 4. 결론

본 논문에서 제안한 특정지역을 촬영한 동영상 분석을 수행하는데 블롭추적(blob tracking) 방법을 적용하여 움직이는 물체를 추적함으로써 얻은 데이터를 기반으로 유동인구의 분석 방법을 제안함으로써 기존에 수동적인 방법에서 발생하는 시간적인 어려움, 노동적인 어려움 그리고 비용적인 어려움을 해결하는 방법을 제안하였다. 성공적인 블롭추적은 강건한(robust) 전경 / 배경 분리를 전제로 하는데, 입력 영상이 야외 장면이기 때문에 조명의 변화가 심한 경우, 예를 들어, 해돋이나 해질녘, 구름이 갑자기 해를 가리는 경우 등에는 픽셀마다의 배경 색상정보를 업데이트(update)할 시간이 없기 때문에, 전경 추출이 제대로 이루어지지 않는데, 이에 대한 조치가 이후의 추후 과제가 될 수 있다.

본 논문은 서울시 산학연 협력사업에서 시행한 [전략산업 혁신클러스터 육성 지원 사업(과제명 : 청계천권역 산업활성화를 위한 U환경 기반 선행디자인 클러스터 구축)]의 지원을 받은 결과물입니다.

또한, 본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 문화콘텐츠기술연구소(CT)육성사업의 연구결과로 수행되었습니다.

#### 5. 참고문헌

- [1] Andrew Senior, Arun Hampapur, Ying-Li Tian, Lisa Brown, Sharath Pankantim Ruud Bolle, "Appearance Models for Occlusion Handling ", Proc. 2nd IEEE Int. Workshop on PETS, Kauai, Hawaii, USA., December 9, 2001
- [2] L. Li, W. Huang, I.Y.H. Gu, Q. Tian, "Foreground object detection from videos containing complex background," ACM Multimedia, 2003.
- [3] Intel , "Computer Vision Workload Analysis : Case Study of Video Surveillance Systems", May 19, 2005
- [4] Shotaro Akaho\*, Hilbert J.Kappen, "Nonmonotonic generalization bias of Gaussian mixture models", Neural Computation archive Volume 12 , Issue 6 (June 2000)
- [5] Thanarat Horprasert, David Harwood, Larry S. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection", ICCV'99 Frame-Rate Workshop
- [6] FFmpeg, .[http://arrozcu.no-ip.org/ffmpeg\\_wiki/tiki-index.php](http://arrozcu.no-ip.org/ffmpeg_wiki/tiki-index.php)
- [7] Intel Open Computer Vision Library, <http://sourceforge.net/projects/opencvlibrary>

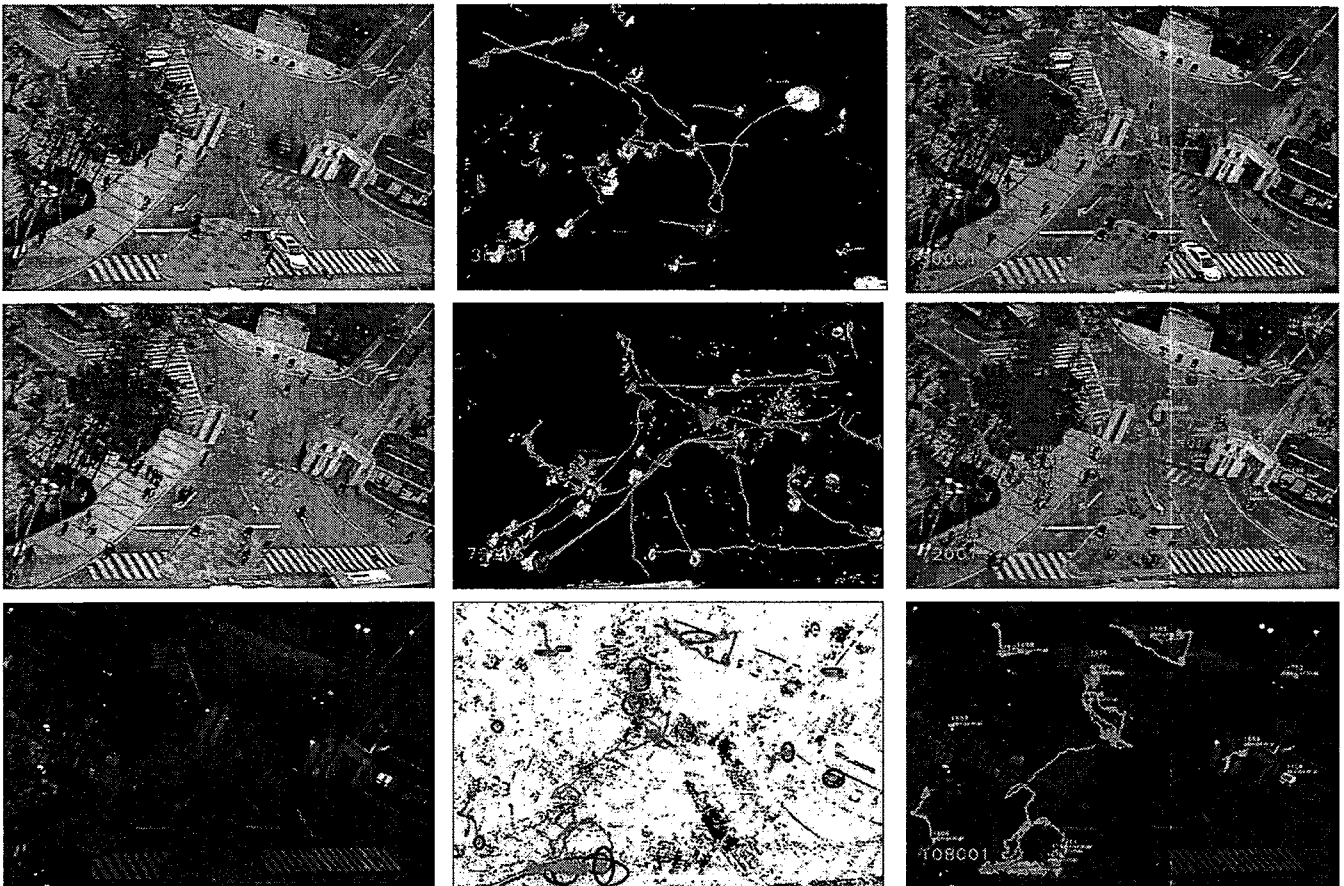


그림 4. 각 행은 시간 즉, 20분, 40분, 60분이 지났을 때의 상황이고, 첫째 열은 원본, 둘째 열은 전경 + 추적 결과, 셋째 열은 원본 + 추적 결과를 나타낸다.