# RGB센서와 Depth센서를 이용한 유동인구 수 측정 시스템

*김기용 **윤경로

*건국대학교

*kky284@gmail.com

# Pedestrian Counting System Using RGB-Sensor and Depth-Sensor

*Kim, Ki-Yong **Yoon, Kyoungro

*Konkuk University

## 요약

컴퓨터 비전이나 감시영상 시스템에서 유동인구 수 측정은 보안 및 경제적 측면에서 중요한 과제이다. 본 논문에서는 RGB 센서와 Depth센서를 이용해서 유동 인체를 식별하고 추적하며, 유동인구 수를 측정하는 방법을 제안한다. 제안된 방법은 받아온 영상에서 유동인체를 검출하고, 검출된 인체를 추적한다. 추적된 정보는 이동방향 인식을 하기 위해 사용된다. 중첩에 대한 문제를 해결하기 위해 Merge&Separate 방법과 Kinect Merge&Separate방법을 제안한다. 검출된 유동인체에 대한 정보는 다양한 변수들로 저장된다. 또한 RGB센서와 Depth센서의 장단점을 상호 보완할 수 있는 인체 매칭 시스템을 제안한다. 본 논문의 실험은 자체적으로 제작한 영상을 사용하여 실험을 진행하였다. 실험을 통하여 제안된 유동인구 수 측정 시스템이 유동인구 검출률을 향상시킬 수 있음을 보여준다.

## 1. 서론

오늘날 공공장소에서 감시 카메라를 쉽게 접할 수 있을 것이다. 유동인구 측정 기술은 감시영상 시스템에서 중요한 사안이다. 사람을 인식하고 유동인구의 수를 세는 것은 보안, 안전이나 경제적인 목적으로서 유용한 정보로 사용될 수 있을 것이다. 예를 들어, 공공장소에서 행사나 축제 같은 상황에서 갑작스러운 사람 수 변화는 위험한 상황을 야기할 수 있다. 또한 상업적인 목적으로서 백화점이나 대중교통 시스템은 유동인구의 측정을 통해서 일정이나 행사와 같은 계획을 세우는데 효과적인 정보로 사용될 수 있다.

컴퓨터 비전이나 감시영상 시스템 분야에서 사람의 수를 세는 것은 개념적으로 크게 두 가지 다른 방법이 있다. 직접적인 방법에서는(검출기반), 영상에서 각각의 사람은 개별적으로 검출된다. 영상에서 사람은 배경과 분리되고 검출된다. 검출된 영역에 대해서 쉽게 사람의 수를 셀 수 있다. 간접적인 방법에서는(회귀 기반), 검출 기반과는 다르게 영상에서 전체적인 특징점을 추출하여 사람의 수를 센다. 각각의 사람에 대한 분할이나 검출을 요구하지 않고 추출된 특징점의 수와 실제 영상에서의 사람의 수에 대하여 관계를 정의하여 수를 측정한다.

두 방법 모두 사람을 검출하는데 각각 장단점을 가지고 있다. 직접적인 방법은 사람의 크기나, 밀도, 부분적인 폐색, 원근법 등에 대해서 큰 영향을 받지 않는다. 또한 각각의 사람에 대해서 위치와 거리를 측정할 수 있다. 하지만 정확하게 각각의 사람을 검출하는 것은 이 자체로 많은 연산량을 요구한다. 아주 복잡하고 사람이 밀집한 상황에서 부정확한 검출을 야기하기도 한다. 간접적인 방법은 복잡하고 밀집된 상황에서 직접적인 방법보다 좀 더 강인하게 측정할 수 있다. 그러나 추출된 특징점과 실제 사람의 수의 관계에 대해 아주 정교한 정의가 필요하다. 영상에서 나타나는 사람마다 다른 거리와 밀도 등을 고려하는 것은 쉽지 않은 일이다.

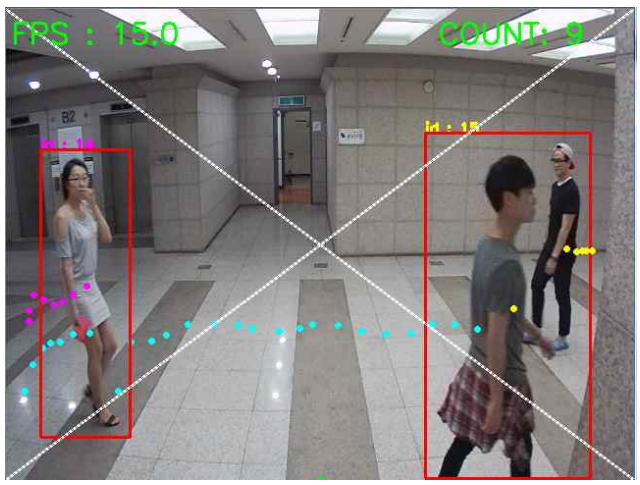


그림 1 제안하는 시스템의 검출 및 추적

두 방법 모두 복잡한 상황에서 중첩에 대한 문제를 해결하는데 어려움을 가진다. 본 논문에서는 Depth센서의 깊이 정보를 이용하여 RGB센서의 중첩문제를 보완하는 방법을 제안한다. RGB센서와 Depth센서에서 들어온 카메라의 FOV(Field of View)내에서의 영상을 분석하여 유동인체를 식별하고 추적하며, 유동인구의 수를 세는 방법을 제안한다. Depth센서에서 유동 인체추적 방법은 깊이(depth) 정보에 기반하여 추적한다. 깊이(depth) 정보는 중첩된 상황에서 효율적으

로 유동인체를 추적할 수 있어 RGB센서에서 중첩문제에 대한 오검출률을 낮출 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 제안된 방법에 대한 구체적인 접근방법을 살펴본다. 3절에서는 자체적으로 제작한 유동인구 검출 데이터셋을 통해 제안된 방법을 평가한다. 마지막으로 4절에서는 본 논문에 대해 결론짓는다.

## 2. 유동인구 분석 시스템 모델

본 절에서 제안된 방법은 RGB센서와 Depth센서로 나뉜다. 공통적인 분석 시스템 흐름은 객체를 검출하고 객체에 대한 추적을 한다. 추적된 정보는 이동방향 인식을 하기위해 사용된다. 검출된 객체정보는 객체ID, 입장시간, 퇴장시간, 체류시간, 입장위치, 퇴장위치, 이동방향으로 구성된다. 마지막으로 유동인구의 수를 측정한다.

### 2.1 RGB센서기반 분석 모듈

RGB센서에서 객체 검출은 배경 분리 방법에 의존한다. 검출된 객체는 각각 고유 ID를 할당하여 분리힌다. 객체 ID할당은 이전 프레임과 비교를 통하여 검출된 객체들 사이에 거리를 비교하여 견적한다. 할당된 ID는 추적 알고리즘을 이용해 고유ID를 유지할 수 있게 한다. 그림 2는 유동인구 분석 시스템 모델의 흐름도를 보여준다.

그림 2 유동인구 분석 시스템 모델 흐름도

객체검출 방법은 가우시안 혼합 기반(Gaussian Mixture-based) 알고리즘을 사용한다. 고정된 카메라에서 순차적으로 들어오는 입력영상을 배경으로 갱신시켜 현재 프레임과의 차영상을 이용하여 객체를 검출한다. 검출된 객체는 잡음(noise)와 홀(hole)을 제거하기 위해 모폴로지(morphology) 연산을 수행한다. 검출된 객체는 이전 프레임과 현재 프레임 간에 거리 비교를 통해서 각각 ID를 할당 받는다.

$$Objs = \begin{cases} extantObjs & pObjs.point - cObjs.point < threshold \\ allocOjbs & otherwise \end{cases} \quad (1)$$

수식(1)에서 Objs는 검출된 객체를 나타내고, extantObjs는 존재하는 객체이고 allocObjs는 새로 할당된 객체이다. pObjs.point는 이전 프레임에서의 객체위치이고 cObjs.point는 현재 프레임에서의 객체위치이다. 각각의 객체는 이전 프레임의 객체와 현재 프레임의 객체 사이에 거리변화가 최소일 것이다. threshold 내에 있는 객체는 존재하는 객체라 보고 새로운 ID를 할당하지 않고 좌표만 갱신된다. 갱신된 객체 이외에 검출된 객체는 새로운 ID를 할당 받는다. 또한 이전에 있던 객체가 현재 프레임에서 검출되지 않으면 할당된 ID는 해제된다.

검출된 객체는 추적을 함에 따라 상/하/좌/우 이동방향을 측정할 수 있다. 상/하/좌/우 방향은 영상의 중심을 기준으로 대각선 4등분 지표로 영역을 나눈다. 검출된 객체는 중심좌표를 이용하여 객체의 입장위치와 퇴장위치를 측정한다. 각각의 객체가 영상에서 어느 방향에서 입장과 퇴장을 했는지 정해진 지표를 통해서 분석될 수 있다.

복잡한 상황에서 각각 검출된 객체는 반복적으로 합쳐지고 분리되는 상황이 발생할 것이다. RGB센서기반 모듈은 매 프레임 검출된 객체 간 거리로 추적을 이용했기 때문에 객체 간 중첩이 발생되면 그 중 하나의 객체를 잃어버리는 경우가 발생한다.

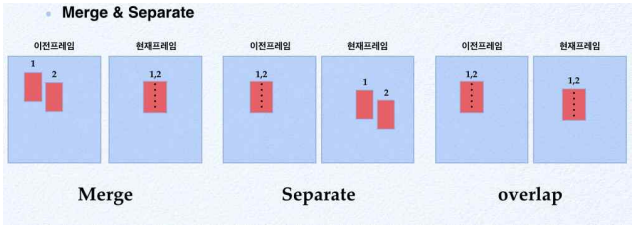


그림 3 Merge&Separate 원리

이러한 중첩문제를 해결하기 위해 Merge&Separate 방법을 제안한다. 그림3은 Merge&Separate 방법의 전체적인 원리이다. 두 개의 객체가 있다고 가정하고, 현재 프레임에서 합쳐진 경우(Merge), 거리에 대한 임계치(threshold)를 주어 현재 프레임에서 검출된 객체에 이전에 검출된 두 개의 객체를 모두 할당한다. 이전 프레임에서 합쳐진 객체가 현재 프레임에서 분리된 경우(Separate), 현재 프레임에서 두 개의 객체가 검출되었다면 임계치 영역 안에 각각 검출된 객체를 할당한다. 이전 프레임과 현재 프레임에서 객체가 오버랩된 경우(overlap), 현재 프레임에서 임계치 영역 안에 두 개의 객체 밖에 없다면 그대로 할당한다.

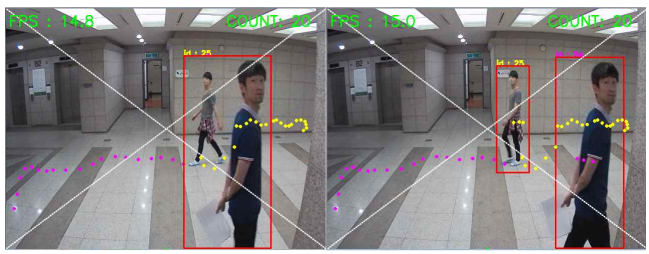


그림 4 Merge&Separate 적용 예

### 2.2 Depth센서기반 분석 모듈

Depth센서에서 객체 검출은 Kinect API에서 제공하는 Skeleton Data를 이용한다. Kinect 카메라의 인체 검출 수는 최대 6명의 사람을 인지할 수 있으며, Skeleton Tracking을 통해서 잡을 수 있는 사람은 최대 2명이다. 나머지 4명은 각 객체에 대한 중심좌표를 반환한다. Skeleton Data를 검출할 수 있는 최대거리는 4m로 제한된다. Kinect는 사람의 관절을 20개 영역으로 나눠서 처리하며, 여러 사람이 들어와도 Tracking ID에 따라 추적을 계속할 수 있다. Depth센서기반 분석 모듈은 RGB센서와 동일한 흐름도를 보이며, 객체 검출부분과 중첩문제 해결부분에서 차이를 보인다. 그림2에서 객체검출과 객체 추적

및 중첩문제 부분은 다른 알고리즘이 적용된다.

Kinect API를 이용해서 최대 6명의 사람을 검출할 수 있지만, 검출된 객체 간 중첩문제는 해결되지 않는다. 만약 두 명의 사람이 중첩된 경우 Kinect카메라로부터 가까운 객체(depth가 더 가까운 객체)는 계속 추적을 할 수 있지만, 뒤로 가려진 객체는 추적하던 Tracking ID를 잃어버리고 다시 새로운 ID를 할당 받게 된다.

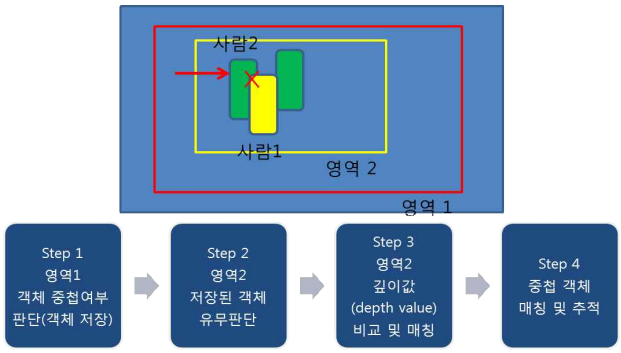


그림 5 Kinect Merge&Separate 원리

이러한 중첩문제를 해결하기 위해 Kinect Merge&Separate 방법을 제안한다. 그림5는 Kinect Merge&Separate 방법의 전체적인 원리이다. 전체적인 방법은 영역을 지정해서 객체의 생성 및 소멸에 대하여 관리를 하는 것이다. 화면 내에 영역1(붉은색)은 추적된 객체가 화면 밖으로 나갔는지 다른 객체에 의해 가려졌는지를 판단하는 지표로 사용된다. 이 영역 안에서 소멸된 객체는 모듈 내에서 객체에 대한 정보를 저장한다. 영역2(노란색)는 영역1안에서 생성된 객체 주변에서 사라진 객체가 있는지를 확인하기 위한 지표이다. 영역2안에서 사라진 객체가 있다면 깊이 값(depth value)를 비교하여 가장 가까운 객체를 할당하여 추적을 계속할 수 있게 ID를 부여한다.

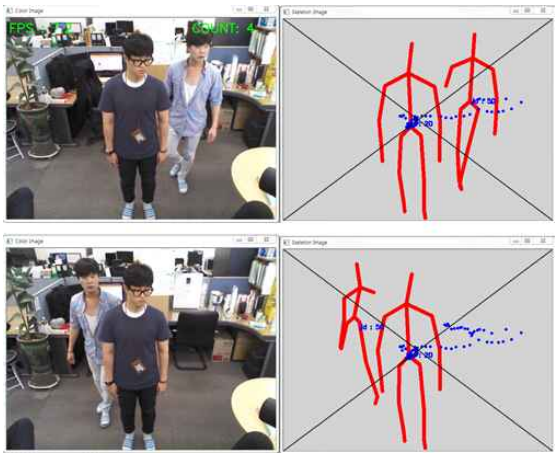

그림 6 Kinect Merge&Separate 적용 예

## 2.3 RGB센서와 Depth센서의 인체 매칭

제안된 분석 모듈에서 RGB센서와 Depth센서는 검출 및 추적을 통하여 유동인구의 수를 측정하였다. RGB센서는 모든 RGB카메라에 적용될 수 있는 장점이 있고, 단점은 3차원 좌표정보를 2차원 좌표정보로 이용하여 추적할 때 Merge&Separate방법이 오차를 발생시킬 수 있다. Depth센서는 RGB센서와 비교하여 깊이 정보(depth)를 획득할 수 있는 장점이 있다. 단점은 Kinect API는 최대 6명의 사람을 검출할 수 있고 최대거리가 4m로 제한된다. RGB센서와 Depth센서에서 각각 검출된 객체를 매칭하고, 또한 Depth센서의 깊이 정보를 이용하여 RGB센서의 Merge&Separate방법을 보완할 수 있다.

RGB센서에서 적용된 Merge&Separate방법은 이전 프레임과 현재 프레임을 비교하여 영상에서 2차원 좌표영역의 거리 비교에 따라 합병 및 분리를 한다. 실제 3차원 영역에서 유동인구가 교차한 경우, 2차원 좌표영역을 이용하는 Merge&Separate방법은 객체를 분리할 때 오차를 발생시킬 수 있다. 이러한 오차를 Depth센서와 인체 매칭을 통하여 보완한다.

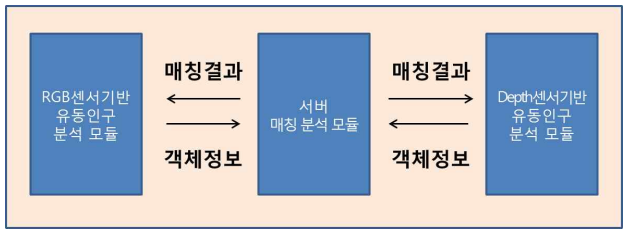


그림 7 RGB센서와 Depth센서 인체 매칭

RGB센서와 Depth센서는 검출된 객체간에 중첩 후 분리된 객체정보를 서버 매칭 분석모듈에 전송한다. 서버 매칭 분석모듈은 객체 수신 및 매칭을 담당한다. 수신된 객체정보는 객체 ID, 이동방향, 누적포인트 등으로 구성되어 있다. 서버 매칭 분석모듈은 수신된 객체들에 대한 누적포인트를 비교하여 RGB센서와 Depth센서의 인체 매칭을 수행한다. RGB센서와 Depth센서에서 전송된 객체는 중첩 후 분리된 상황의 정보이다. 이 각각의 객체는 누적포인트 중에서 시작점과 중간점, 마지막점의 비교를 통해 분리된 상황을 판별한다. Depth센서에 적용된 Kinect Merge&Separate는 깊이 값(depth value)를 적용해 RGB센서의 Merge&Separate의 오차를 보정할 수 있다. 서버 매칭 분석모듈은 객체 비교를 통해 RGB센서의 분리 상황에 대한 오차를 판별하면 RGB센서에 객체 보정에 대한 결과를 다시 전달하여 보정한다.

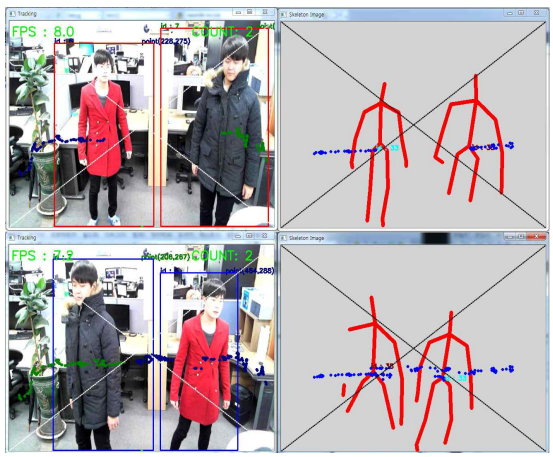

그림 8 인체 매칭 적용 예

## 3. 실험적 평가

유동인구 시스템 인구 수 측정 평가는 자체적으로 제작한 유동인구 검출 영상들을 이용하여 평가를 수행하였다. 실험에 사용된 데이터 셋은 RGB센서와 Depth센서로 동일한 영상을 촬영했으며, 무작위로 돌아다니는 환경을 조성한 복잡한 상황의 복도이다. 모든 데이터 셋에 모듈은 동일한 조건으로 실험이 진행되었고, 데이터 셋의 크기는 640x480 픽셀로 수행되었다.

표1은 RGB센서와 Depth센서에 대한 데이터셋별 유동인구 수 측정 결과이다. S4 와 S5는 카메라의 높이가 약 2m로서 평균적으로 좌에서 우로 움직이는 유동인구가 많은 영상으로서 외부 조명에 대한 영향이 덜하다. S4와 S5는 다양한 거리의 유동인구가 이동하는 환경이며, RGB센서기반 검출모듈은 중첩된 상황에서도 각 객체를 강인하게 추적하며 약 90%의 검출률을 보였다. Depth 중첩 보정 상황에 대한 RGB 유동인구 수는 많은 오검출을 보여 적용하지 않은 결과이다. Depth센서기반 검출모듈은 RGB센서보다 좁은 화각(FOV)을 보인다. Kinect API는 사람을 검출할 때 카메라와 평행한 옆모습은 검출하지 못하고 어느 정도 정면을 보여야 검출한다. 또한 카메라로부터 거리가 4m 이상의 객체에 대한 검출을 하지 못하여 근접한 유동인구에 대하여 보정한다. Depth센서기반 검출모듈은 S4와 S5에 대해 Merge&Separate 보정을 통하여 RGB 유동인구 수 검출률에 대한 증가를 보인다.

| 데이터 셋 | 실측한 유동인구 수 | 측정된 유동인구 수 | 검출률 | Depth 중첩 보정 수 | 보정후 유동인구 수 | 증가율 |
|---|---|---|---|---|---|---|
| 유동인구 SET 4 (S4) | 123 | 110 | 89.4% | 4 | 114 | 92.7% (△3.3%) |
| 유동인구 SET 5 (S5) | 141 | 127 | 90.0% | 7 | 134 | 95.0% (△5.0%) |

〔표 1〕 데이터셋별 유동인구 수 실험 결과

## 4. 결론 및 향후과제

본 논문에서는 RGB센서와 Depth센서를 이용해서 유동 인체를 검출하고 추적하며, 유동인구 수를 측정하는 방법을 제안하였다. 검출된 유동인체에 대한 정보를 정의하고, 복잡한 상황에서 중첩문제를 해결하였다. 자체적으로 제작한 데이터 셋에 대한 평가는 좋은 성능을 보인다. RGB센서의 중첩문제를 보완하기 위한 Depth센서의 인체 매칭을 제안했다. 제안된 인체 매칭 모듈은 영상 내에 2명 이하의 전제조건이 붙는 완성된 모듈이 아니기 때문에, 향후 RGB센서와 Depth센서의 인체 매칭 모듈은 일반적인 상황에 대해 객체 보정을 통해 좀 더 효율적이고 좋은 성능을 보일 것으로 보인다.

## 참고문헌

[1] D. Conte, P. foggia, G. Percannella, F. Tufano and M. Vento, “A method for counting people in crowded scenes,” in International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2010.

[2] V.B Subburanman, A. Descamps, C. Carincotte, "Counting people in the crowd using a generic head detector". AVSS 2012. Beijing: IEEE Press, 2012: 470-475

[3] H. Fu, H. Ma and H. Xiao. "Real-time accurate crowd counting based on RGB-D information," In Proceedings of International Conference on Information Processing, pp.2685-2688, 2012.

[4] C.-T. Hsieh, H.-C. Wang, Y.-K. Wu, L.-C. Chang, and T.-K. Kuo, “A kinect-based people-flow counting system,” International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), pp.146-150, 2012.

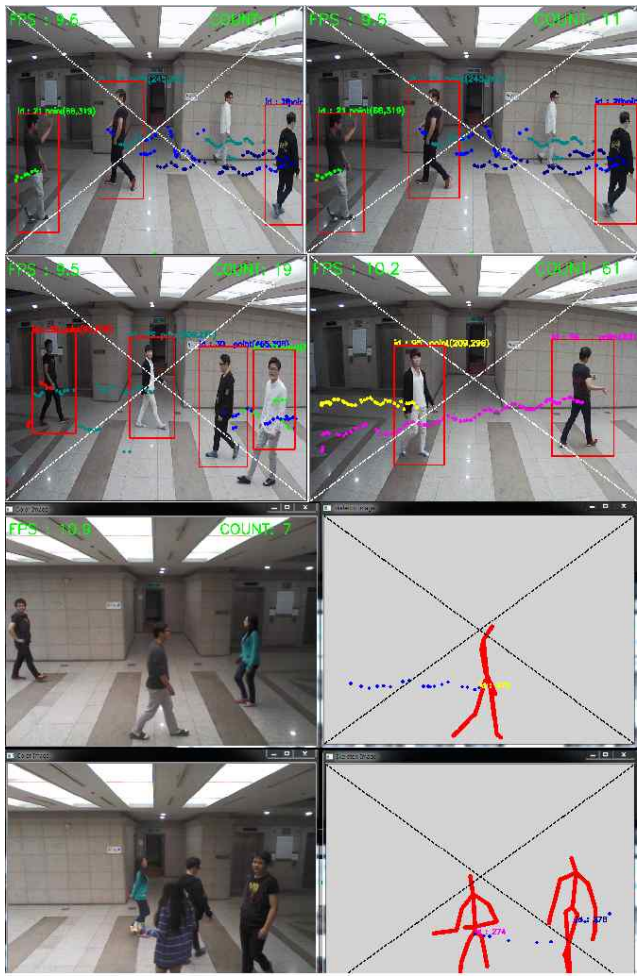

그림 9 S4 와 S5에 대한 시스템 적용 예