

이동객체 인식을 활용한 웹기반 차량이동 감지 시스템

주유성*, 이효종*

*전북대학교 전자정보공학부

e-mail: pristy@chonbuk.ac.kr

Web-based Vehicle Detection System by Recognition of Moving Image Objects

Yu-Sung Joo*, Hyo Jong Lee*

*Division of Electronics and Information Engineering,
Chonbuk National University

요 약

기존의 수동적인 차량관리 시스템은 관리자가 정해진 위치에 상시 대기하여 관리를 해주어야 하며 아날로그 녹화 시스템의 화질저하와 저장공간의 부족이라는 단점이 있으므로 생산성이나 관리적으로 크게 비효율적이다. 본 논문에서는 이러한 아날로그적인 방법을 탈피하여 영상처리 기법을 이용해 보안카메라에서 받아온 영상정보를 이용해 차량정보를 파악하고 실시간으로 처리하며 web이나 PDA 등 디지털 이동매체에 접속하여 이동환경에서 활용할 수 있도록 통합 시스템을 구축하는 것을 제안한다.

Keywords: 차량감지, 이동객체인식, JMF, 모폴러지

1. 서론

현재 사회 여러 분야에서 디지털 영상 처리(Digital Image Processing) 기술을 도입한 많은 시스템들이 사람들의 관심을 받고 있다. 디지털 영상 처리는 멀티미디어 시대에 있어 중요한 역할을 하는 영상을 컴퓨터를 이용하여 처리하는 분야이다. 얼굴을 인식하여 현금을 인출하여 주는 현금 지급기, 상대방의 얼굴을 보면서 통화하는 화상 전화나 주민등록증 갱신을 위해 손가락의 지문을 컴퓨터에 입력하는 등이 모두 디지털 영상처리 분야이다. 대부분 영상처리의 경우 카메라에 의해 획득된 영상을 하드웨어적인 연결에 의해 모니터에 전송하거나 단순한 기록만을 하고 있다.

그러나 진보된 영상 시스템에서는 획득한 영상을 분석하여 정보를 추출할 수 있어야 한다. 예를 들어서 주차장을 감시하는 카메라의 경우, 영상을 분석할 수 있다면 단순한 감시기능 이외에, 주차장에 접근하는 사람이나, 주차공간의 계산 등을 자동화시킬 수 있다.

본 논문에서는 보안 카메라에서 획득한 영상을 실시간 처리 기술을 적용하여 이동하는 객체를 인식하고, 필요한 정보를 생성 시킨 후 모니터에 결과를 나타내는데 필요한 기술을 구현한다[1]. 응용 사례로 주차장의 감시카메라에 적용하여 영상의 획득에서 실시간 처리에 이르기까지 필요한 사항들을 고찰하였다. 우수한 결과를 얻기 위해서는 잡음의 제거, 적절한 Threshold를 이용한 이진영상화 등을 전처리 기술로 적용하였고, 움직이는 객체는 사람과 자동차를 구분하여 특정 지점을 통과하는 사람이나 자동차의 수를 정확히 셀 수 있도록 하였다. 이러한 결과는 웹에서 연동할 수 있도록 하여 다수의 접속자들이 실시간 접속을 통하여 감시카메라로부터 가공된 정보를 공유할 수 있는 것이다.

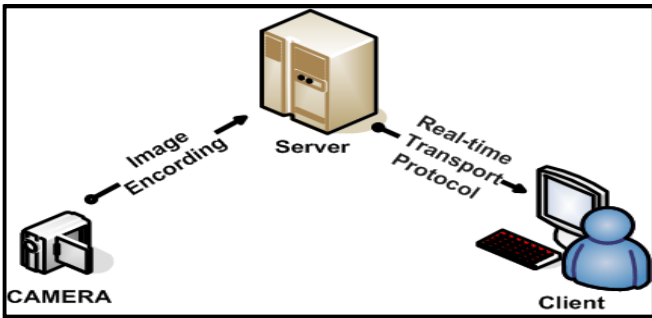
2. 영상의 획득

2.1 JMF Registry

디지털 캠코더 또는 감시카메라 에서 획득한 영상은 JMF library와 Video Capture Device를 사용해

입력받은 영상을 동영상 이미지로 변환 한다. Win32 환경 하에서 JMF media capture 기능을 이용하기 위해서는 JMF가 지원하는 capture device 조건에 충족해야 한다. 즉, capture device가 VFW Capture Driver를 제공해야 한다.

현재 논문에 테스트로 쓰이고 있는 BT878 칩셋을 사용하는 TV수신카드를 JMF상에서 VFW Capture Devices를 완벽하게 인식한다. BT878 칩셋의 최대 Capture 가능 Spec은 320*240 Pixel, 초당 15 frame 이다. 본 논문에서는 순수한 Capture기능만을 이용해 가공되지 않은 RAW Format의 데이터를 버퍼에 담아 데이터를 처리한다.



[그림 1] 시스템 구조

2.2 JMF를 통한 동영상 획득

JMF(Java Multimedia Frameworks)는 자바 기반으로 멀티미디어를 다루기 위한 클래스들을 모아놓은 라이브러리이다. 즉, 멀티미디어의 재생, 녹화, 검색, 전송 등의 기술을 자바 기반으로 만들어 이용하기 위한 도구인 것이다. 또한 JMF는 RTP(Realtime Transfer Protocol)를 지원함으로써 실시간으로 동영상과 음성을 전송하고, 이러한 기능을 통해서 인터넷 멀티채팅, 화상회의, 원격 감시등에도 그 이용분야가 상당히 높다. Capture된 영상을 다루기 위하여 프레임 단위로 영상을 잘라야한다.

JMF 라이브러리에서 FramGrabbingControl클래스의 grabFrame()함수를 이용하여 현재의 프레임을 Capture해서 버퍼에 저장한다. 버퍼에 저장되는 것은 벡터형태의 raw data이다. 이것을 이미지 처리와 저장을 하기 위해서 buffered Image로 변환해준다. 이것은 차분영상(differential image)을 위한 자료가 된다[2,3].

3. 영상처리

3.1 JAI를 이용한 차분 영상

JAI(Java Advanced Imaging)는 자바를 기반으로 이미지를 처리하는 라이브러리이다. JAI는 차분 영상의 계산, 다양한 영상 포맷의 읽고 쓰기, 이미지의 Pixel 정보의 처리, 압축 등 여러 가지 영상 처리에 관련된 도구들을 포함하고 있다[4].

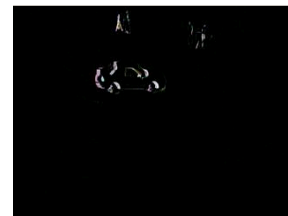
차분 영상 분석 방법은 주위 환경 변화에 민감하고 계산이 단순하며 다양한 유형의 움직이는 객체를 검출하는 데 장점이 있으나, 정지한 물체나 카메라 이동이 있을 경우에는 객체 인식에 오류가 있을 수 있다는 단점이 있다. 본 논문에서는 단일 영상 방법 중 차분 영상 획득 방법을 이용하였다[5].

[그림 2] 는 카메라로부터 입력받은 영상의 예를 보여주고 있다. 이와 같이 획득한 프레임을 JAI의 API 중의 하나인 RenderedOp를 이용하여 차분 영상(Differential Images)을 구할 수 있다.



(a) t_0 에서의 원 영상

(b) t_1 에서의 원 영상



(c) Gray channel의 차분영상

[그림 2] 카메라로부터 입력된 영상과 차분 이미지

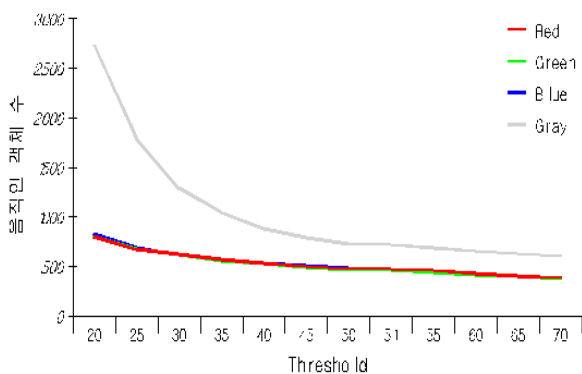
JAI에서 제공하는 RGB 형태의 영상은 각 화소당 32비트로 표현되고 있다. 이 32비트는 알파, 적색, 녹색, 그리고 청색 채널을 각각 8비트에 표현하고 있다. 실시간 처리에서 불필요한 계산의 제거는 중요한 요소가 되므로 본 연구에서는 RGB값을 평균한 gray영상과 더불어 각각의 독립된 채널의 효율성을 고찰하였다.

버퍼에 각 프레임이 저장되고, 이 이미지 영상에서 Pixel 정보를 추출하여 RGB값으로 변환 시킨다. 각 채널의 값은 인식 결과 측면에서 볼 때에 큰 차이가 없었다. Gray영상으로 처리하는 경우 화소당 2

회의 덧셈과 1회의 나눗셈 연산을 필요로 한다. 그러나, 이러한 연산을 이용할 경우 이동객체의 인식 영역은 넓게 나타났다. 특히 threshold의 값이 낮아질수록 개별 채널과 gray영상의 인식결과 값의 차이가 두드러졌다.

Pixel을 RGB값으로 변환 시키는 계산을 줄여 처리 속도를 향상시키기 위해, Red, Green, Blue 각 channel에서의 차분 영상을 구한 경우 육안으로는 그 차이의 식별이 거의 불가능하다.

[그림 3]은 동영상에서 움직이는 객체의 Pixel 수를 그래프로 나타낸 것이다.



[그림 3] 각 channel에서 인식한 객체 수

Threshold 값이 낮아질수록 인식하는 Pixel의 수는 많아지고, Gray channel에서 움직이는 객체를 가장 많이 인식하는 것을 알 수 있다. 실시간 처리를 위해서는 단일 채널을 선별하여 인식할 수도 있지만 본 논문에서는 Gray channel을 사용하여 연구를 진행하였다.

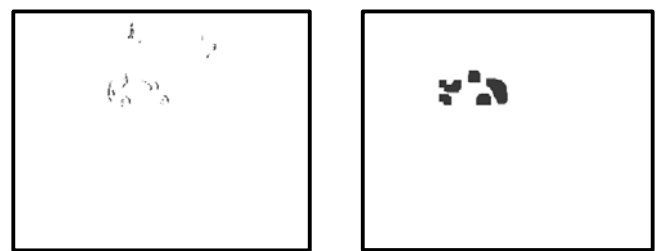
본 논문에서 Threshold값의 오차범위를 줄이기 위한 반복적인 테스트와 차량이 이동하는 시점에서의 화면상의 노이즈가 없는 최적의 값을 찾아 적용한다. Threshold 값이 낮을수록 움직이는 객체는 많이 인식 할 수 있으나, 움직이지 않은 객체가 민감하게 반응하여 원하는 결과와는 다르게 움직임으로 잘못 인식하는 객체 수도 상대적으로 늘어난다. 원치 않는 Pixel을 인식한 최소한의 Threshold 값은 Gray channel이 나머지 3개의 R,G,B 채널보다 약 2 배 정도 많았으며 오차범위 1~2 사이로 Red channel, Green channel, Blue channel 순으로 나타났다.

4. 객체 인식

객체의 인식은 앞서 기술한 적절한 Threshold값을 이용하여 구할 수 있다. 카메라의 움직임이 없다고 가정하면 이동하는 객체는 두개의 다른 시간, 즉 t_0 와 t_1 에 획득된 두 영상의 차분 영상으로 나타나게 된다. 이 차분 영상에서 Threshold값을 적용하여 영상을 구하면 이동하는 객체의 영역을 추출할 수 있다.

주차장 부근에서 발견되는 이동하는 객체는 주로 자동차와 보행자이다. 이러한 객체를 인식하기 위해 차분 영상으로 얻어진 이미지를 Pixel 단위로 처리한다. 이 Pixel들은 1차원 배열로 총 크기는 Width \times Height이다. 이미지를 Gray channel 값을 갖는 정보로 변환하여 가장 적절한 Threshold 값에 의해 이미지의 이동 객체에 대하여 이진화 영상으로 변환시킨다. 각 화소를 비교하여 Threshold 값보다 크면 1로, 작으면 0으로 처리하는 것이다. 최종적으로 '1'로 처리된 Pixel들은 확실하게 이동한 Pixel들이다.

'1'로 표현된 픽셀들은 대부분 객체가 형성되어 그룹을 짓게 되지만 객체의 길이가 긴 차량의 경우엔 차분 영상에 적용된 두 프레임의 시간 차이가 근소하기 때문에 윤곽선부근에서만 이동객체의 영역이 형성된다. 이와 같이 불확실한 윤곽선을 분명하게 나타내기 위해서 이진화 처리된 영상이 들어오면 Opening Operation을 수행해 침식과 확장연산을 수행한다. 그리하여 약간의 거리가 있는 차량 객체를 합칠 수 있는 최적의 모폴러지를 정하였고 그 결과 차분 영상 된 이미지에서 인접한 그룹들이 연결되었다. 즉 세션 상태에서 차량의 윤곽을 두드러지게 한 것이다. 본 영상에서는 3x3의 침식마스크와 9x9의 확장마스크를 사용하여 확산을 시켰을 때 양호한 결과를 보여주었다.



(a) 적용 전

(b) 적용 후

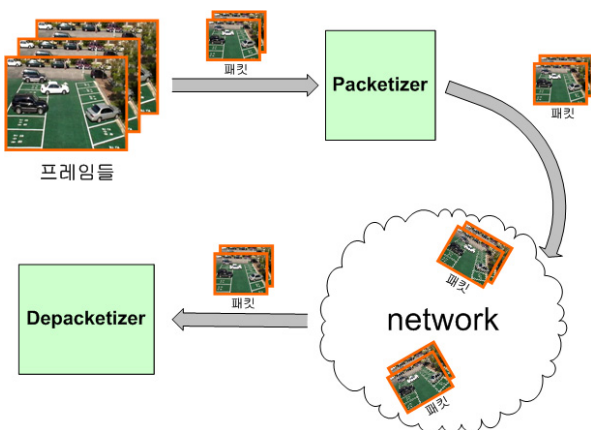
[그림 4] Morphology 적용

JAI상에서 이미지는 벡터형태를 이용하기 때문에 1차원 배열을 사용하여 2차원 이미지로 변환하는 연산을 수행한다. i 값은 Threshold값에 의한 1의 값이 채워진 Pixel의 인덱스를 나타낸다. 이렇게 완성된 객체의 Pixel의 수를 구하여 자동차는 약1600(80×20) Pixel을 차지하고, 사람은 약 160(8×20) Pixel을 차지한다. 그래서 차지하는 Pixel의 수 또는 가로세로의 비를 따져 보아서 대략적으로 자동차인지 사람인지 구분 할 수 있다. 실험결과 일반적으로 자동차는 4 : 1, 움직이는 사람은 0.4 : 1의 대조되는 비율을 나타내기 때문에 객체의 인식이 용이하였다. 하지만 고정된 카메라 시점에서 벗어난 영상을 획득하거나 사람과 차량이 겹쳐질 경우, 그리고 움직이는 차량이나 사람이 약간의 딜레이를 두고 움직일 경우 올바른 객체의 파악이 불가능하다.

5. 네트워크 환경 구축

최종적으로 객체를 인식하여 얻은 정보를 관리자가 언제 어디서든 확인할 수 있도록 하기 위해 네트워크를 이용한 송수신이 필요하다. 여기에 RTP를 이용, BroadCaster Class의 setProcessor 모드에 따라 미디어에 대한 Processor를 생성한 후 startBroadcast 메소드에서 SendStream을 생성, 미디어 정보를 전송한다.

[그림5]는 RTP에 대한 간략한 개념을 보여주며 미디어를 네트워크로 전송하기 위해 없어서는 안되는 부분이다. Packetizer는 미디어자료를 패킷 형태로 변환하며 Depacketizer는 반대로 다시 사용가능한 미디어의 형태로 변환하는 작업을 한다. 그림에서 보는것과 같이 몇 개의 프레임 덩어리로 패킷을 구성 후 목적지까지 도달하는 과정을 보여준다.



[그림 5] Packetizer와 Depacketizer

최종적으로 클라이언트에도 송수신을 위해 JMF가 깔려 있어야 하며 수신된 자료를 관리하기 위한 데이터베이스를 운용 해야만 한다. 정확한 관리를 위해 영상정보와 싱크를 맞추어 화면과 데이터베이스상의 정보가 어긋나지 않게 한다.



[그림 6] 클라이언트로 송신된 영상정보

[그림6]은 테스트를 위해 클라이언트 상에서 실시간으로 전송받은 영상과 모폴러지 연산을 통해 객체인식을 위한 이미지 처리결과를 수신 받은 화면을 보여준다.

6. 결론

움직이는 객체를 자동으로 인식하고, 이를 웹과 연동하여 다수의 접속자들이 주차장에 오고가는 사람은 어느 정도인지, 자동차는 어느 정도 주차가 되어 있는지, 주차장의 혼잡도를 미리 알고, 주차 가능한 공간의 여부를 알 수 있어 안내자가 없이도 주차를 손쉽게 할 수 있다.

이를 바탕으로 주차장에서 오가는 객체들, 즉 사람이나 자동차의 양이 얼마나 되는지를 정량적으로 측정할 수 있고, 이를 통계적으로 처리하여 해당 주차장의 혼잡도 등을 알아낼 수 있으며, 주차장의 주차 공간의 여부를 알 수 있을 것이다.

참고문헌

- [1] Ismail Haritaoglu, David Harwood, Larry S. Davis, "Real-Time Surveillance of People and Their Activities," IEEE Tans, Vol.22, No.8, Aug. 2000.
- [2] J. Hildebrandt, R. Hollamby, "Daynamic Geospatial Image mosaics using JAVA, JAI, RMiand CORBA,"
- [3] 임종석, 김옥현, "이동 카메라 영상에서 쉼터 정보를 이용한 다수 보행자 검출 및 추적", 한국 정보 처리학회, 2004.
- [4] 신창훈, 이주신 "다중 비디오에서 색 정보를 이용한 특정 이동 물체 추적 알고리즘" 한국 정보 처리학회, 2004
- [5] J. R. Parker, "Algorithms for Image Processing and Computer Vision,"