

TMO 기반 감시 로봇의 설계 및 구현

Design and implementation of a surveillance robot

정유진, 박선식, 이재호
한국외국어대학교 컴퓨터공학과

Yoojin Chung, Sunsik Park, Jaehyo Lee
Department of Computer Engineering,
Hankuk Univ. of Foreign Studies

요약

본 논문에서는 빈 사무실의 침입을 감시하는 로봇을 설계하고 구현한다. 실시간 감시를 위해 TMO-Linux 커널을 사용하였고 로봇으로 X-Bot 플랫폼을 이용하였다. 로봇의 카메라 움직임을 제어하고 영상을 처리하는 영상서버를 설계하고 구현하였다. 또한 영상서버와 TMO 서버와 통신하며 로봇과 카메라를 제어하는 클라이언트를 설계하고 구현하였다.

Abstract

In this paper, we design and implement a surveillance robot to detect an intruder in an empty office. We use a TMO-Linux kernel for a real-time surveillance and use a X-Bot platform for a robot. We design and implement an image server to process images and to detect an intruder. And we design and implement a client to communicate with a image server and TMO server and control a camera on a surveillance robot.

I. 서론

본 논문에서는 빈 사무실의 침입을 감시하는 로봇을 설계하고 구현한다. 실시간 감시를 위해 TMO-Linux 커널을 사용하였고 로봇으로 X-Bot 플랫폼을 이용하였다. 로봇의 카메라 움직임을 제어하고 영상을 처리하는 영상서버를 설계하고 구현하였다. 또한 영상서버와 TMO 서버와 통신하며 로봇과 카메라를 제어하는 클라이언트를 설계하고 구현하였다.

II. TMO 기반 감시 로봇의 설계 및 구현

본산 실시간 객체 모델 TMO는 실시간 시스템의 시간 보장성 컴퓨팅을 지원하기 위한 설계 패러다임으로 제안된 것이고, TMO_Linux 의 스케줄러는 기존 리눅스 커널의 스케줄러 부분에 주기적 실시간 프로세스와 실

시간 서버 프로세스를 위한 데드라인 기반 스케줄링이 추가된 것이다.

본 논문의 TMO 기반 감시 로봇의 개발환경은, 운영체제로 TMO-Linux를 사용하였고, EV2440 Board [8]을 개발 보드로 사용하였고, 카메라로 Tilt Cam [9]을 사용하였다.

1. Xbot

본 논문의 감시 로봇은 TMO-Linux 커널에서 TMO Application을 이용한 로봇제어에 유진로봇의 X-Bot 청소로봇 플랫폼 [10]을 활용하여 개발하였다. X-Bot은 적외선 센서(전방/바닥)와 충격감지 센서(전방), 엔코더 카운터 센서를 가지고 있는데 이번 개발에서 사용한 센서는 바닥 적외선 센서를 제외한 3개 센서를 사용했다. X-Bot은 8비트 단위로 처음과 끝을 나타내는 2개와 크게 6개의 센서상황들을 포함한 총 32비트 값을 시리얼포트를 통하여 전송한다. 이는 TMO Application

에서 배열에 저장하여 필요한 센서값을 확인하여 활용하였다.

전방 적외선 센서는 총 7개의 센서로 전방 3개 센서는 하나만 인지되면 동일한 값을 출력한다. 각각 단위 센서 값 개수는 5개 이고 단위 센서들 동시에 인지될 경우 다양한 방향에서 장애물을 인지 할 수 있다. 충격 감지 센서는 3개의 단위 센서로 그 개수는 3개이고 동시에 인지될 경우 적외선과 같이 다양한 방향에서 장애물을 인지할 수 있다.

엔코더 카운터 값은 로봇의 이동 방향과 회전 방향에 따라 각각 바퀴의 펄스 값의 증감형태가 달라진다. 0에서부터 255까지 256개의 펄스 값이 발생하는데 증가형태일 때는 적은 수부터 출력되고 감소형태일 때는 높은 수부터 출력된다. 로봇으로부터 데이터가 10 msec 단위 출력되기 때문에 일정하게 순차적인 형태로 펄스 값이 출력되지 않고 또한 속도에 따라서 출력된 펄스 값의 폭이 달라진다. 누적펄스 값은 오른쪽 바퀴 기준으로 하였고 증감형태가 변할 때 값이 큰 쪽으로 변화하는데 이를 감지하여 감소하는 것을 증가형태로 변화시켜 계산하였다.

로봇의 바퀴가 한번 회전할 때 1초에 600개의 펄스가 발생할 경우 속도는 22.6cm/sec이 된다. 단위 펄스 당 이동거리는 22.6cm/sec을 600으로 나눈 값이 된다. 속도에 따라 단위시간당 발생하는 펄스 값이 변화하므로 이동 거리는 누적된 펄스 값에 단위 펄스 당 이동거리를 곱하면 결과가 나온다. 이동거리를 이용해서 회전시 회전 각도를 구할 수 있다. 로봇이 회전하기 시작한 시점부터 회전한 거리를 부채꼴 호의 길이를 구하는 공식에 대입하여 회전한 각도를 구하여 원하는 각도만큼 이동하게 하였다. 이러한 것을 바탕으로 로봇이 이동하는 알고리즘을 개발하였다.

2. TMO Application

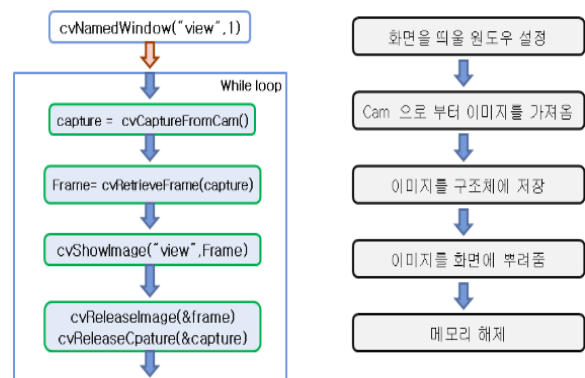
시리얼 통신을 담당하는 RS232C 클래스와 로봇에게 명령을 내리는 Controller 클래스, 주로 이동 알고리즘이 들어가 있는 Navigation 클래스를 만들어 3개의 TMO 클래스를 사용한다. 이 3개의 클래스는 채널을 할당하여 서로 통신하게 되는데 채널 할당 시 Read 또는 Write 형태로 정하는 옵션이 있다.

RS232C 는 각각 1개씩 SpM과 SvM을 갖는다. SpM

에서는 계속 로봇에서 센서 상황 데이터를 받아오고 SvM에서는 Read로 설정된 채널로 들어오는 메시지, 즉 로봇제어 명령을 로봇에 전송한다. SpM에서는 3개의 센서 값을 제어하기위해 3개의 채널을 할당하여 각각 로봇으로부터 읽은 데이터를 넘겨 줄 수 있게 한다. 채널을 통해 32비트 값 모두를 넘겨주고 특정 센서에 대한 데이터 활용은 3개의 채널과 연결된 Navigation에서 한다. SvM에서는 로봇제어 데이터 값을 읽어오는 것이므로 Controller와 연결된다. Controller 는 1개의 SvM을 갖는데 Navigation 으로부터 받은 데이터를 활용하여 제어명령을 설정및 변경한다. 이렇게 만들어진 로봇 제어 명령어는 할당된 채널을 통해 RS232C 로 전달된다. Navigation 은 2개의 SpM과 3개의 SvM을 갖는다. SpM (drive)는 이동알고리즘을 구현하고 SpM (remotedControl)은 원격제어S를 하기 위해 소켓을 통해 클라이언트가 접속을 해서 보낸 메시지를 제어한다. 이 두 개의 할당된 Controller 채널을 이용해 메시지를 전달한다. 3개의 SvM은 Navigation에서 센서값을 활용하기 위해 RS232C에서 채널을 통해 받은 데이터를 적외선, 충격감지, 엔코더카운터 값으로 가공한다. SvM (handleWheelEN) 은 펄스값의 증감 형태를 모두 증가형태로 변화시키고 이동거리를 실시간으로 계산한다.

3. OpenCV

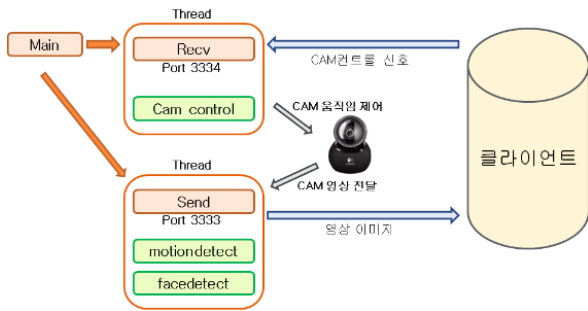
OpenCV [5]는 인텔사의 주도 하에서 개발된 실시간 컴퓨터 비전 라이브러리인데, 본 논문에서는 이를 사용한다. OpenCV의 기본 패턴은 다음 그림과 같다.



▶▶ 그림 1. OpenCV 기본패턴

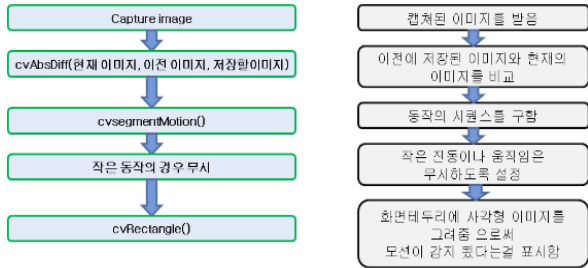
4. 영상 서버

영상서버의 구조는 아래 그림과 같다.



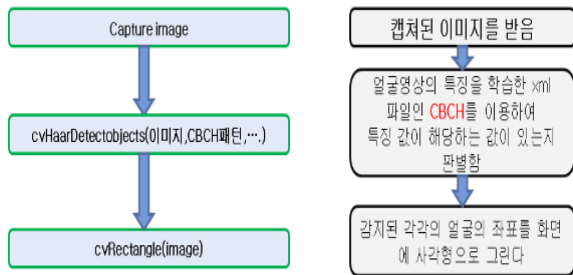
▶▶ 그림 2. 영상처리 서버 구조

영상서버에서 동작을 인식하는 절차는 다음과 같다.



▶▶ 그림 3. 동작인식 구조

영상 서버에서 얼굴을 인식하는 절차는 다음과 같다.



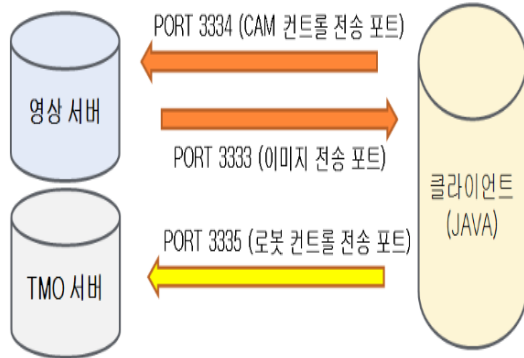
▶▶ 그림 4. 얼굴 인식 구조

5. Tilt / Pan CAM 컨트롤

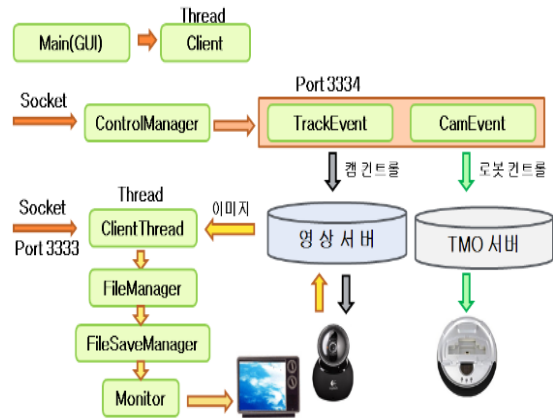
CAM의 회전을 제어하기 위해서는 ioctl함수를 이용하여 직접 제어하였다.

6. 클라이언트 구현

클라이언트는 영상서버 / TMO 서버 총 2개의 서버와 연결한다. 이 서버들과 통신하기 위하여 3개의 포트를 사용하며 그림 5와 그림 6에서 보듯이 각 포트는 각각의 일을 수행한다.



▶▶ 그림 5. 클라이언트 동작 구조



▶▶ 그림 6. 클라이언트 동작 클래스

6.1 클라이언트 구성 요소

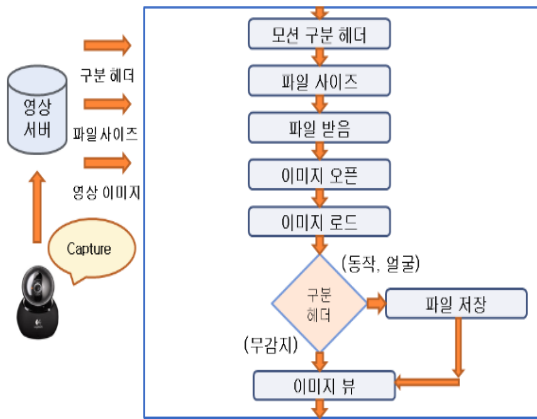
클라이언트 구성 요소는 다음과 같은 7개의 클래스로 이루어지며, 각 클래스는 각각의 역할을 수행한다.

- 1) Main - GUI담당
- 2) Monitor - 이미지 패널 제어
- 3) Component - ListComponent 객체관리
- 4) ControlManager - 서버 컨트롤 제어
- 5) TrackEvent / CamEvent - CAM제어 / 로봇 제어
- 6) Client / ClientThread - 서버간의 통신
- 7) FileManager - 서버에서 전송되는 이미지 및 데이터 제어

6.2 이미지의 처리 절차

이미지의 처리 절차는 다음과 같다.

- 1) 클라이언트의 이미지처리는 서버로부터 .JPG파일을 받아서 모니터에 뿌려준다.
- 2) 이미지 데이터를 받기 전에 영상서버로부터 모션감지여부 및 파일사이즈 헤더를 받고 파일을 받는다.
- 3) 받은 파일은 모션감지여부에 따라서 각기 다른 뷰로 보내지게 된다.



▶▶ 그림 7. 클라이언트 이미지 처리 구조

6.3 클라이언트 GUI 구조

클라이언트의 GUI는 다음과 같은 기능들로 구성되어 있다.

- 1) 서버연결 / 세부설정 : 각 모니터 기능 및 파일저장을 ON / OFF 할 수 있다.
- 2) 영상 모니터 : 실시간 CAM 영상을 비롯하여 모션 / 얼굴탐지 / 파일 뷰 의 모니터기능을 가지고 있다.
- 3) 파일 저장 리스트 : 설정에 따라서 탐지된 모션 과 얼굴을 파일로 저장한다.
- 4) 로봇 / 카메라 컨트롤 : 로봇제어 및 카메라를 수동 또는 자동으로 제어할 수 있다.
- 5) 모션탐지 로그 : 모션 및 얼굴 인식 시간을 로깅 한다.

구현된 GUI는 그림 8과 같다.



▶▶ 그림 8. 클라이언트 GUI

6.4 얼굴 탐지 인식

얼굴 탐지는 그림 9와 같이 인식되는 얼굴의 수에 따라서 사각형의 색이 변하도록 하였으며, 각각의 파일로 저장되도록 하였다.



III. 결론 및 향후계획

본 논문에서 TMO-Linux Kernel 2.6.10 을 사용하였는데, TILT 조절이 되는 CAM이 사용하는 UVCDriver 모듈설치가 TMO-Linux Kernel 2.6.18에 만 정상적으로 되고 사용한 커널에는 되지 않았다. 아

직 TMO-Linux Kernel 2.6.10에 맞게 해당 Driver를 수정하지 못해서 캠 제어는 별도의 C코드로 작성하여 테스트하였다. Driver를 수정하고 보드에 포팅하여 현재 시스템을 경량화할 것이다. 또한 영상 처리시 jpeg로 변환하여 클라이언트로 보내게 되는데 서버에서 jpeg로 변환하는 오버헤드가 커서 딜레이가 생기게 된다. 영상을 jpeg로 저장하지 않고 RAW 포맷으로 빠르게 전송하는 방법을 추가적으로 보완해야 할 것과, OpenCv 에서 동작 감시 하는 소스에서 메모리를 최적화시키지 않아 메모리를 상대적으로 많이 소비되는것 또한 보완해야 될 사항이다.

■ 참고 문헌 ■

- [1] 김정국, "분산 실시간 객체를 지원하는 리눅스 기반 운영체제 TMO-Linux", 전자공학회지 제29권11호 pp.91-98, 2002. 11
- [2] 김정국, 김문희, "분산 실시간 객체 TMO의 실행 지원을 위한 미들웨어 및 실시간/임베디드 커널의 구현 동향", 정보과학회지 제24권 제8호 p.53-60, 2006. 8
- [3] Kim, J. G., et al, "TMO-Linux: A Linux-based Real-time Operating System Supporting 2.3. Execution of TMO's," Proc. IEEE Int'l Symposium, ISORC2002, Washington DC, Apr. 28, 2002.
- [4] 정성환, 이문호, "컴퓨터 비전 실무프로그래밍 - 기본편", 홍릉과학출판사, 2007
- [5] OpenCV - <http://sourceforge.net/projects/opencvlibrary/>
- [6] Motion-3.2.9 - http://sourceforge.net/project/showfiles.php?group_id=13468&package_id=153327
- [7] CAM UVC Driver -<http://linux-uvc.berlios.de/>
- [8] EV2440 Board - <http://contest.kesic.or.kr> , <http://www.minirobot.co.kr/>
- [9] Logitech QuickCam - <http://www.logitech.com>
- [10] X-Bot (유진로봇) - <http://yujinrobot.com/>