프로젝트 포트폴리오

시각장애인을 위한 보행 보조 시스템

성명: 김태헌

목 차

• 프로젝트 개요

• 개발 배경

한국에 시각장애인들은 맹인 안내견을 제외하고는 사회 기반시설의 부족으로 인하여 외출 및 사회활동에서 소극적인 참여에 그치고 있습니다. 본 프로젝트는 맹인 안내견과 같은 혜택을 보지 못하는 시각장애인분들에게 조금이나마 주변 환경에 대한 정보를 제공하여 외출 및 사회활동의 참여를 장려 하려는 의도를 가지고 출발하게 되었습니다.

• 개발 목표

프로젝트에서 서버의 목적은 관리자에게 반드시 필요한 기능만 넣는 것과 IOCP를 이용한 고속 입출력을 구현하는 것을 목표로 두었습니다.

• 개발 범위

MFC, IOCP를 사용한 서버, 캠, 라즈베리 파이를 사용한 클라이언트, openCV를 사용한 영상 분석 알고리즘

- ## 참여 분야

  MFC 기반 IOCP Server

- ## 개발 환경 및 사용 기술

  - **IOCP**

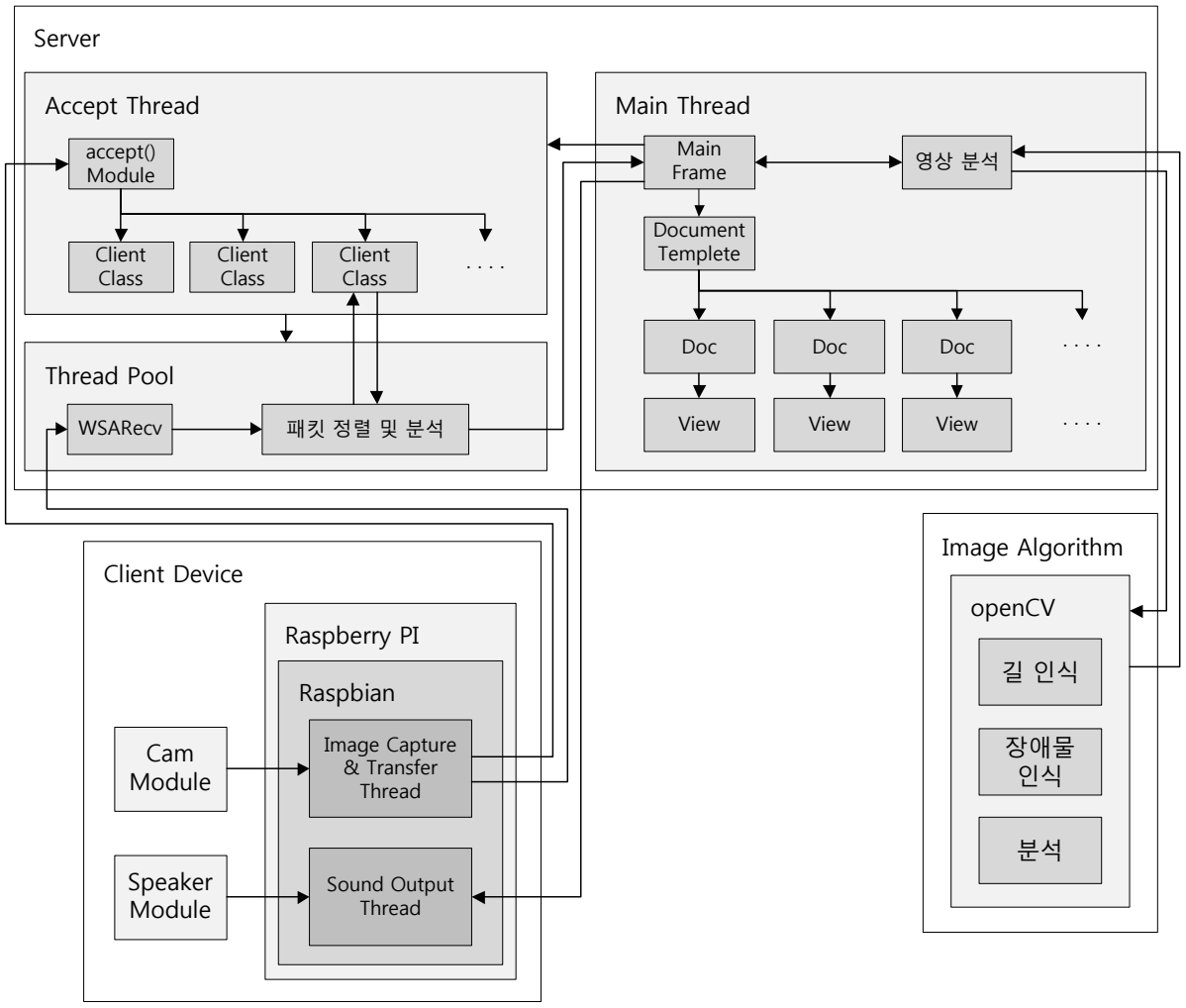
    프로그램 설계를 끝내고 prototype 제작에는 MFC에서 제공하는 CAsyncSocket을 사용하였지만 시스템의 한계를 느꼈습니다. 이후 방향을 바꿔 Windows의 IO 방식 중 가장 빠르다는 IOCP를 적용하였습니다.

  - **MFC**

    C++ 기반으로 서버에서 중요시 되는 속도에서 C# 등의 다른 언어보다 이점이 있고, 서버에서 이미지 처리에 사용되는 openCV가 C++로 호환이 가능하기 때문에 MFC를 선택하였습니다.

- ## 프로젝트 설명

  - **System Architecture**



크게 PC에서 사용할 서버와 라즈베리 파이로 제작된 클라이언트가 있으며 서버에 이미지 처리 알고리즘이 있습니다.

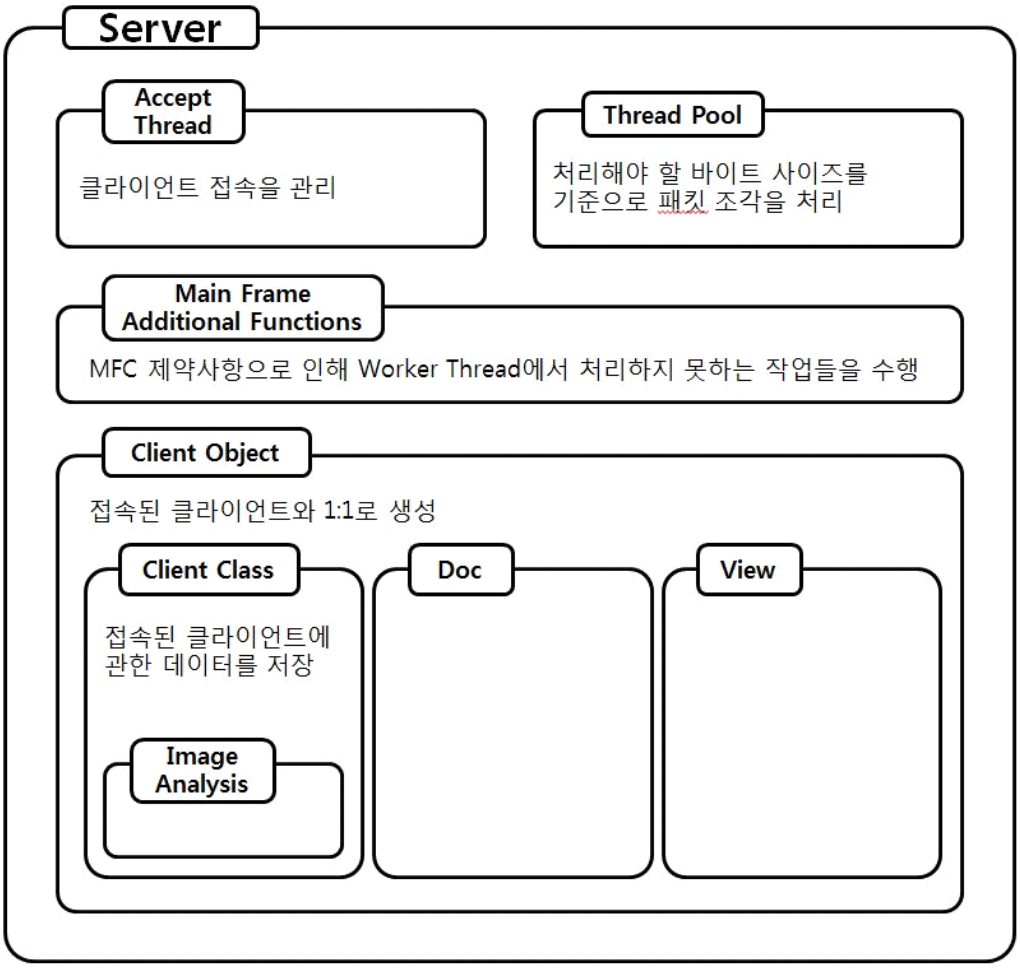
서버에서는 각 클라이언트와의 통신과 데이터 저장, 이미지 처리 알고리즘으로부터 반환받은 피드백 전송, 클라이언트에 할당된 자식 창 관리 등의 기능을 합니다. 클라이언트는 캠에서 찍은 이미지 전송 및 서버에서 보낸 피드백을 분석하여 해당 경고를 출력하는 기능이 있습니다. 이미지 처리 알고리즘에서는 이미지 내 길과 장애물의 유무를 판별하여 서버 측에 클라이언트에 피드백으로 보낼 구조체를 넘겨주는 기능을 합니다.

- **참여 분야(Server) 상세 설명**
  - **IOCP**

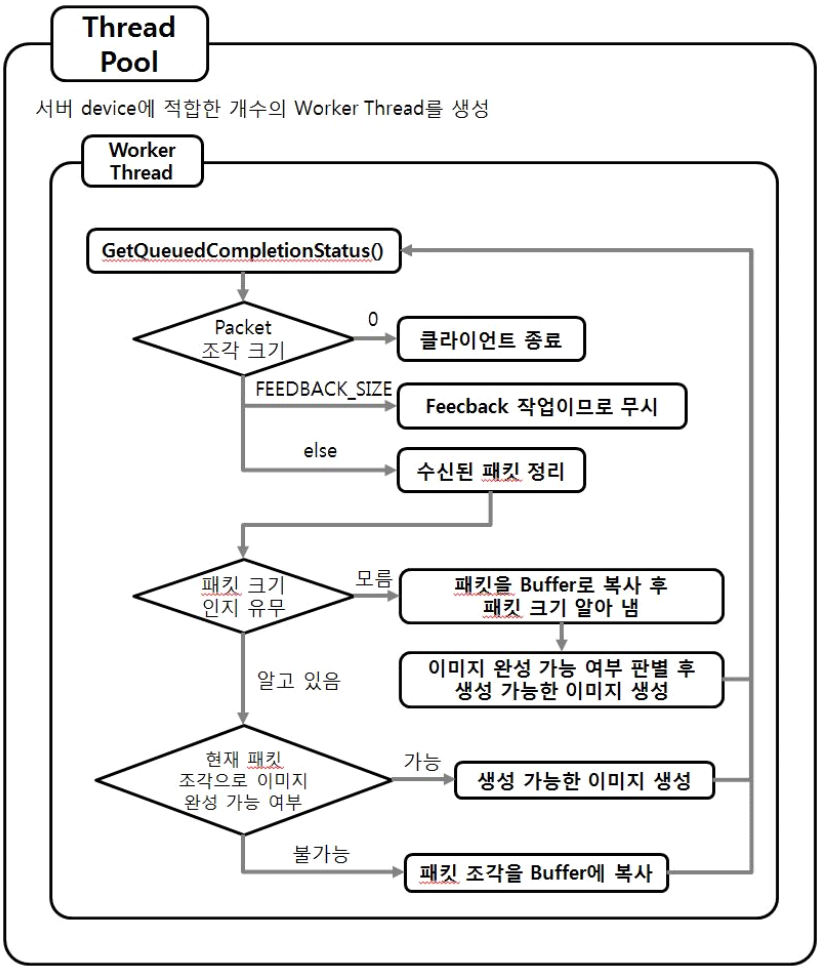
    IOCP의 기본 개념에 스레드 풀이 활용 되므로 각 클라이언트 별 패킷을 정리하는 문제가 관건이었습니다. 본 프로젝트에서는 CP 오브젝트에 클라이언트 소켓을 Completion Key로 등록하여 어느 클라이언트에서 보낸 패킷인지를 판단합니다. 이후 GetQueuedCompletionStatus() 함수의 두 번째 매개변수인 lpNumberOfBytes로 클라이언트가 접속 종료 했는지를 판별하고 접속 종료가 아닐 경우 해당 패킷 처리 방식을 선택합니다. lpNumberOfBytes가 피드백 구조체의 크기와 일치할 경우 무시하게 되며, 이외의 경우 클라이언트에서 전송된 패킷으로 인지하고 개별 클라이언트에 대한 처리를

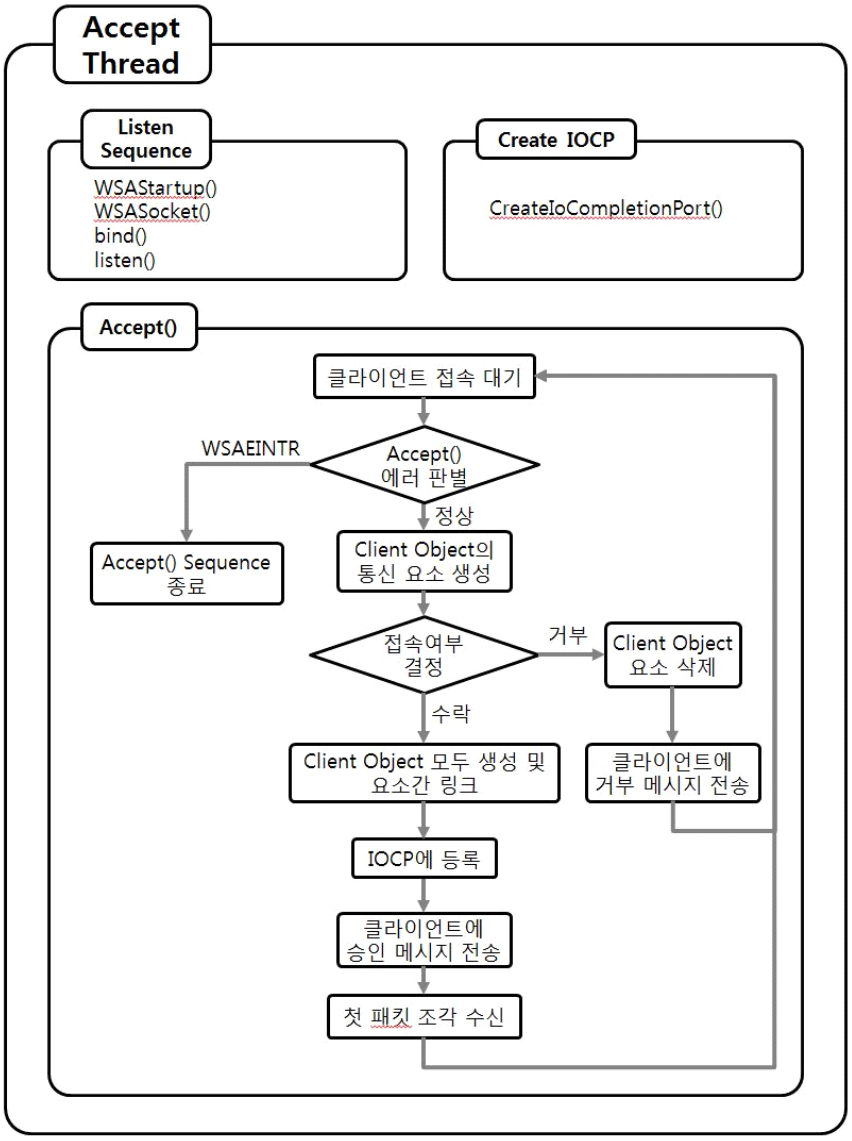
하게 됩니다.

다음은 서버 프로그램의 모듈별 구성도입니다.



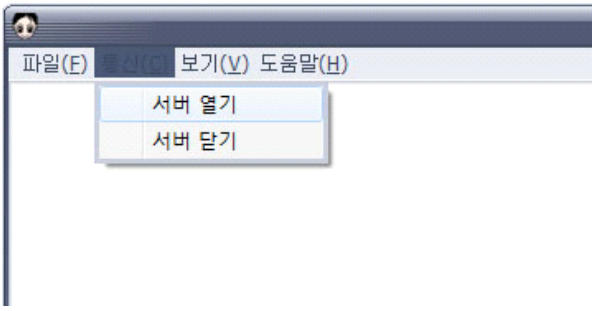
(다음 장에 계속)


Thread Pool
서버 device에 적합한 개수의 Worker Thread를 생성
Worker Thread
GetQueuedCompletionStatus()
Packet 조각 크기
0
클라이언트 종료
FEEDBACK_SIZE
Feecback 작업이므로 무시
else
수신된 패킷 정리
패킷 크기 인지 유무
모름
패킷을 Buffer로 복사 후 패킷 크기 알아 냄
이미지 완성 가능 여부 판별 후 생성 가능한 이미지 생성
알고 있음
현재 패킷 조각으로 이미지 완성 가능 여부
가능
생성 가능한 이미지 생성
불가능
패킷 조각을 Buffer에 복사

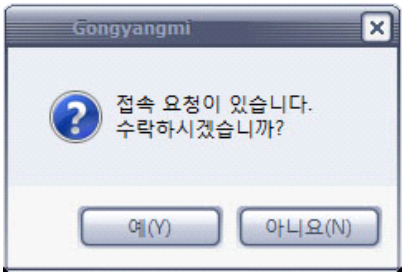

Accept Thread
Listen Sequence
WSAStartup()
WSASocket()
bind()
listen()
Create IOCP
CreateIoCompletionPort()
Accept()
클라이언트 접속 대기
Accept() 에러 판별
WSAEINTR
정상
Accept() Sequence 종료
Client Object의 통신 요소 생성
접속여부 결정
거부
Client Object 요소 삭제
수락
Client Object 모두 생성 및 요소간 링크
클라이언트에 거부 메시지 전송
IOCP에 등록
클라이언트에 승인 메시지 전송
첫 패킷 조각 수신

UI

실행파일을 실행 시키면 그림과 같은 화면을 볼 수 있습니다. 메뉴 바의 구성으로는 '파일' '통신' '보기' '도움말' 순서로 구성되어 있습니다. 이것은 관리자가 직관적으로 프로그램을 사용할 수 있게 디자인하였습니다. 파일 안에는 '끝내기' 메뉴만을 넣었으며 서버의 실행과 종료는 통신메뉴에서 사용됩니다. '보기'에는 접속한 클라이언트의 IP주소가 순차적으로 나타납니다.

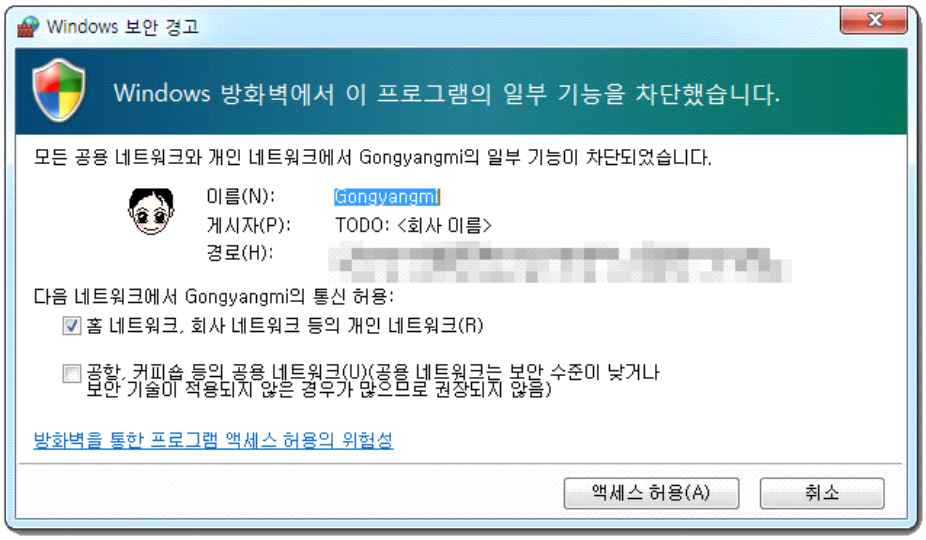


서버를 구동시키기 위해서는 이와 같이 '메뉴 바'의 '통신'에 있는 '서버 열기'를 선택하면 클라이언트가 접속 할 수 있는 환경이 구성됩니다.



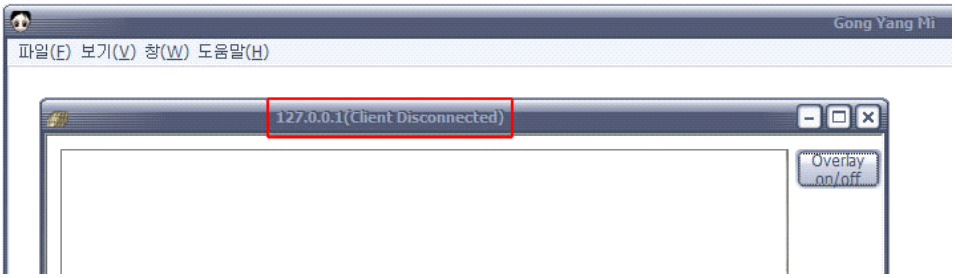
클라이언트가 접속을 시도하면 그림과 같은 '접속 승인 확인 창'이 뜨며 '예(Y)'를 누르게 되면 클라이언트와의 통신과 영상 분석이 시작되게 됩니다.

Child Window 상단의 타이틀 바에 접속한 클라이언트의 IP가 보이게 됩니다. 오른쪽 위의 'overlay on/off' 버튼을 누르면 Child Window에 출력되는 영상에서 분석된 이후 추가되는 색 영역을 끄고 켤 수 있습니다.

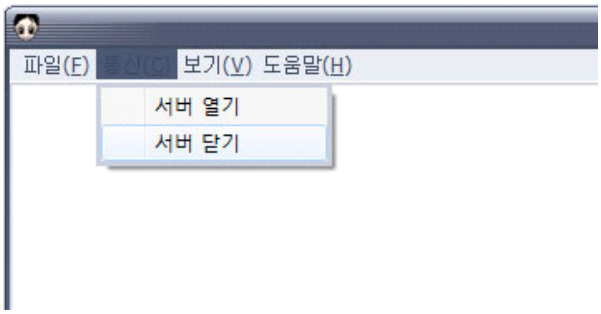


만약에 해당 PC에서 처음으로 서버를 가동하는 경우 그림과 같은 경고 창이 뜰 수 있으나 본 프로그램이 사용하려는 '7777'번 포트를 개방하려는 의도이므로 '액세스 허용(A)'을 눌러주면 윈도우 방화벽의 인바운드 규칙에 프로그램이 등록됩니다.

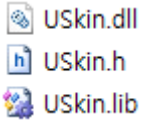
개발당시에 PC의 네트워크 환경이 IP공유기에 속해있었기 때문에 외부 IP에서 서버에 접속하기 위해서 IP 공유기의 '포트포워드' 기능이 필요하였습니다. 해당 기능을 설정하기 위해서는 공유기 설정 페이지인 '192.168.0.1'(공유기 설정에 따라 변경될 수 있음)에 접속하여 그림과 같이 포트포워딩 설정을 해주면 해당 포트로 공유기에 할당된 공인 IP로 접속하게 되면 IP 공유기에서 설정된 사설 IP로 패킷을 전달하게 됩니다.



클라이언트에서 먼저 접속을 종료시킨 경우 그림과 같이 Child Window 타이틀 바에 'Disconnected' 문구가 뜨며, 서버에서 먼저 접속을 종료시켜야 할 경우 Child Window 오른쪽 위의 닫기 버튼을 누르면 됩니다.



서버를 닫으려면 모든 Child Window를 닫은 후 그림과 같이 '메뉴 바'의 '통신'에 있는 '서버 닫기'를 선택하게 되면 서버 프로그램의 통신 기능이 꺼지게 됩니다.



MFC에서 UI를 구현하기 위해서는 구현하려는 모든 부분들에 대하여 직접 코딩을 해주어야 했습니다. 너무나 많은 시간과 노력을 소비하여야 했기에

시간과 노력을 단축하기 위하여 외부라이브러리를 가지고와 사용하기로 결정하였습니다. 위 USkin 라이브러리의 장점은 간단한 코딩으로 전체적인 UI를 개선할 수 있다는 장점이 있고, msstyle이라는 파일들은 모두 가져다가 쉽게 사용할 수 있다는 것이었습니다.