

중고 스마트 폰을 이용한 얼굴 추적 및 영상 전송 시스템

황준호, 김연중, 최한슬
우종호, 정순호
부경대학교 컴퓨터공학과

Facial Tracking & Image Transporting System Using Used Smart Phone

Jun-Ho Hwang, Yoen-Jung Kim, Han-Seul Choi
Chong-Ho Woo, Soon-Ho Jung
Dept of Computer Engineering, Pukyong National University

요 약

요즈음 주변에서 쉽게 구할 수 있는 중고 스마트 폰으로 안면 인식 앱과 모터제어 기술을 합작하여 식별된 사람의 영상을 또 다른 스마트 폰으로 전송하는 기초적인 기술을 구현하였다. 이 기술을 발전시키면 인공지능 제품을 중고 스마트 폰으로 저렴하게 구현할 수 있을 것으로 기대된다.

1. 서론

인공지능의 기술을 이용하여 사람의 얼굴을 추적하는 애플리케이션이 구현되어 적용되어지고 있다. 예를 들어 ‘주인을 따라다니는 가방’, ‘안면 인식을 통한 호텔 체크인’ 등이 있다.[1]

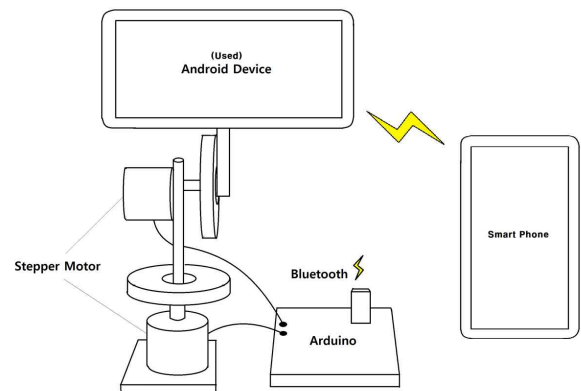
대부분의 얼굴 인식 기술을 활용한 시스템에서는 독립적인 디바이스에 한 해 완제품으로 구성하고 있어서, 개인이 사용하기에는 고가거나, 접하기에는 거리가 먼 경우가 대부분이다.

본 논문에서는 이러한 문제를 해결하기 위해 중고 스마트 폰을 이용한 얼굴 추적 시스템으로 얼굴을 인식하고 확인된 영상을 또 다른 스마트 폰으로 전송하는 기술을 구현하였다.

이 논문의 구성은 2장에서 시스템의 개요, H/W 그리고 S/W를 설명하고 3장에서 실험 및 구현을, 4장에서는 결론을 언급한다.

2. 얼굴 추적 시스템

이 시스템은 크게 두 부분으로, 첫 번째는 서버로서 스마트 폰의 카메라를 이용해서 사람의 얼굴을 감지하고 인식된 얼굴이 화면의 중앙으로 이동할 수 있도록 스텝모터로 구성된 관절을 아두이노로 제어하는 부분이다. 두 번째는 서버에서 구성된 영상을 클라이언트인 또 다른 스마트 폰으로 전송하여 알려 주는 부분이다. 전체적인 구성은 (그림 1)과 같다.



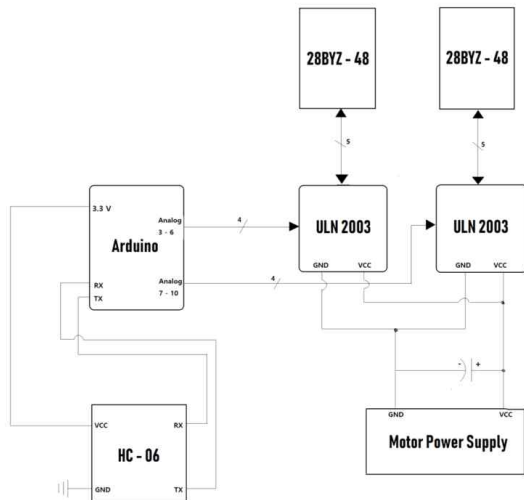
(그림 1) 개 요

2.1. H/W

이 시스템의 하드웨어는 <표 1>과 같이 구성하였다. 상하(이하 회전각), 좌우(이하 회전 방향)를 제어하기 위해 각각 한 개의 스텝모터(이하 모터)를 사용하였다. 모터의 제어를 위해서 오픈소스 하드웨어인 아두이노를 사용했고 모터와 아두이노를 연결시키기 위해서 드라이버를 사용하였다. 서버-액츄에이터 간 통신은 전송 거리가 짧고 간단한 명령만을 송수신하기 때문에 블루투스 통신방식을 채택했다. 서버-클라이언트 간 통신은 영상을 송수신 해야하므로 인터넷으로 통신한다.

<표 1> 하드웨어 구성

	하드웨어	동작	기종
서버	스마트 폰	카메라 동작	갤럭시 S8
	아두이노	액추에이터 제어	Arduino Uno
	스텝모터	회전각/회전 방향 표현	28BYJ-48
	모터드라이버	스텝모터 제어	ULN2003
	블루투스	장치 간 통신	HC-06
클라이언트	스마트 폰	전송된 영상의 확인	LG G7



(그림 2) 액추에이터 회로도

액추에이터의 회로도는 (그림 2)와 같다. 스텝모터는 아두이노 보드의 전원만으로 동작시키기에 무리가 있어서, 충분한 토크 압력을 위해서 외부전원을 인가하였다.[2]

2.2. S / W

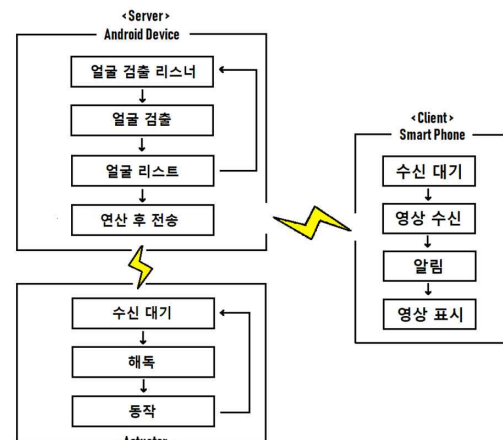
이 서버의 소프트웨어는 <표 2>와 같이 스마트 폰과 액추에이터로 구성된다.

<표 2> 소프트웨어 구성

	장치	기능
서버	스마트 폰	U I
		얼굴 검출
		위치 판별
		명령 송신
	액추에이터	명령 수신
클라이언트	스마트 폰	모터 제어
		영상 수신
		영상 Display

(그림 3)과 같이 스마트 폰 내부의 얼굴 검출 리스너가 얼굴에 대한 리스트를 갱신하는 동작을 반복한다. 얼굴 리스트를 이용하여 위치판별 후에 액추에이터로 명령을 전송한다. 액추에이터는 스마트 폰으로부터 명령을 수신하여서 회전각, 회전 방향 이동 명령을 판단하여 모터에 적절한 신호를 인가함으로써 동

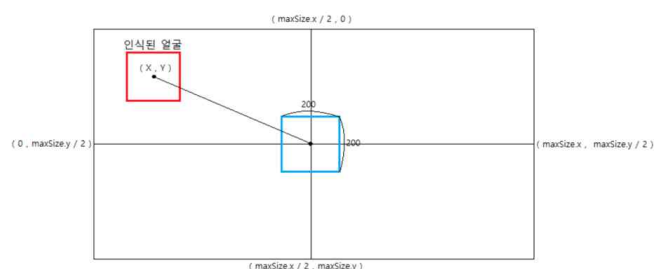
작하게 된다.



(그림 3) 소프트웨어 흐름

2.2.1.스마트 폰의 얼굴 검출과 위치판별

Android API를 이용해서 얼굴을 검출한다. (그림 4)는 인식된 얼굴을 화면의 중앙으로 위치시키기 위해서 현재의 위치가 화면의 중앙 지점을 기준으로 어느 위치에 있는지 판별하는 과정을 화면으로 나타낸 예시이다. 얼굴의 좌표 값이 중앙 영역 내에 있다면 화면의 중앙에 위치한다고 판별한다. maxSize는 스마트 폰의 최대 너비와 최대 높이를 가진 객체이다. 그러므로 maxSize의 x와 y를 2로 나눈 값이 스마트 폰의 정중앙의 좌표가 된다. 이를 기준으로 움직여야 할 방향을 판별하게 된다. 판별한 방향으로 움직이도록 액추에이터에게 지시한다. 이 동작을 반복하면서 화면의 중앙 지점으로 스마트 폰이 이동하게 된다.[3][4]



(그림 4) 얼굴 인식 상황 예시

Left : $X < \text{maxSize.x}/2 - C$

Right : $X > \text{maxSize.x}/2 + C$

Up : $Y < \text{maxSize.y}/2 - C$

Down : $Y > \text{maxSize.y}/2 + C$

Stop : $\text{maxSize.x}/2 - C < X < \text{maxSize.x}/2 + C$

Stop : $\text{maxSize.y}/2 - C < Y < \text{maxSize.y}/2 + C$

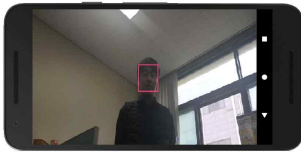
2.2.2. 액추에이터

액추에이터의 소프트웨어는 Arduino IDE를 이용하여 작성했고 라이브러리는 stepper를 사용했다. 먼저, setup 함수를 통해서 모터와 블루투스 사용 준비를 하고, 준비가 완료되면 액추에이터는 시리얼 통신을 열어두고 수신을 위한 대기 상태에 들어간다. 그 후, 스마트폰 측으로부터 명령을 전달받으면 해당 명령을 통해 입출력 핀에 적절한 신호를 인가함으로써 모터를 제어하고 다시 수신 대기 상태로 들어간다.

2.2.3. 클라이언트 스마트 폰

클라이언트로 동작하는 스마트 폰의 애플리케이션은 서버로부터 영상을 전송받아서 사용자에게 알려주고 화면에 표시한다. 동작 시, 인터넷을 이용해서 통신하고 서버에서 인식된 사람의 영상을 즉각적으로 클라이언트에 전송한다.

(그림 5)는 클라이언트 동작 화면을 예시로 보였다.



(그림 5) 클라이언트 동작 화면 예시

3. 실험 및 구현

앞선 내용을 바탕으로 실제 동작을 실험했다. 얼굴이 검출되면 동작이 시작되고 얼굴 추적을 통한 위치 보정을 반복한 다음, 얼굴이 중앙에 위치하게 된다. 그러면 서버가 클라이언트에게 영상을 전송하고 클라이언트는 사용자에게 알리고 영상을 표시한다.

최종적으로 아래와 같이 시스템을 구성했다. (그림 6)과 같이 얼굴을 인식하고 얼굴이 중앙에 위치하면 클라이언트로 영상을 전송한다.



(그림 6) 시스템 동작 모습

4. 결론

본 논문에서는 중고 스마트 폰을 재활용하여 얼굴 추적과 영상 전송 시스템을 구현했다.

Android FaceDetector API를 이용해서 얼굴 검출이 이루어졌는데, 안경을 착용하거나 머리카락으로 눈썹을 가리는 경우에는 문제점이 있었다.

중고 스마트 폰을 이용한 얼굴 인식 어플을 인공지능적인 기능 강화와 클라이언트의 UI를 간편하게 하여 사용의 편리성을 향상시킬 수 있다면 더 나은 성능을 기대할 수 있을 것이다.

우리는 이 실험을 통해 고가의 장비 대신 중고 스마트 폰을 이용해 얼굴추적시스템을 개인이 쉽게 구현할 수 있기를 기대해본다.

참고문헌

- [1]Youtube "[글로벌 경제] 안면 인식 기술, 어디까지 왔나? / KBS뉴스(News)" 2019, <https://www.youtube.com/watch?v=MCitKp2nedY>
- [2]Youtube MIMLAB "17.스테퍼모터 활용하기", 2016, <https://www.youtube.com/watch?v=U050R2ek1zc>
- [3]android"FaceDetector", <https://developer.android.com/reference/android/media/FaceDetector>
- [4]Github "FaceDectector", 2018, <https://github.com/RedApparat/FaceDetector>