

MARKER 추적 기반 IoT 디바이스 제어 증강현실 인터페이스

순천향대학교 컴퓨터공학과

황대영, 안승훈, 이운기, 이에녹, 오진선, 고금주, 임지은, 천인국*

IoT device control via augmented reality interface based on tracking MARKER

DaeYoung Hwang, SeungHoon Ahn, WoonGi Lee, E Noch Yi, JinSeon Oh, GeumJoo Ko, JiEun Lim

InGook Chun*

Dept. of Computer Science & Engineering, SoonChunHyang University, Asan 31538, Korea

= Abstract =

Currently, most IoT devices are not integrated or controlled because they have non-intuitive interfaces such as dedicated applications or dedicated controllers. This paper describes home IoT technology that controls devices through AR interface. Each IoT device is assigned its own MARKER. Once the MARKER given to each device is recognized through the smartphone user application, the user interface that manipulates the IoT device is provided as AR on the recognized MARKER. Using this interface, the datas sent from the user application go through a Web server connected by a TCP socket and are formatted with its own protocol and passed to the Aduino. Markers were recognized through coordinate calibration using the sub-pixel algorithm and the integration of original images from the smartphone camera and AR was implemented by calculating the correction matrix on the marker object. The part of device control was implemented through reverse engineering, which analyzes raw data of the remote control and finds internal rules. This IoT system provides a more intuitive interface and more organic connection between device.

I. 서 론

오늘날 4차 산업혁명은 기술의 침단으로서 주목을 받고 있다. 그 중 사물인터넷은 인간과 가장 밀접한 기술로서 급격한 성장세를 보이고 있으며 산업시설, 건설, 건설사 등 다양한 분야에서 연구되고 상용화 되고 있는 추세이다.

특히 가전기기들이 IoT화 되면서 스마트 홈을 구성하였고 최근 국내에서도 홈 네트워크에 관련된 시

장이 크게 활성화 되었다(통계자료). 시장이 활성화 되면서 사물인터넷 환경을 접목시킨 기기의 수가 증가함에 따라 다수의 기기종 기기를 관리하기 위한 통합시스템의 필요성이 대두되고 있다. 또한 IoT 환경을 구축하기 위해서는 여러 가지 기술적인 환경 구축이 선행되어야 하는데, 특히 입력 데이터를 처리하고 저장하고 변환하는 것과 같은 인터페이스 역할을 수행하는 기술이 필요하다. 인터페이스 기술에는 미들웨어 기술, 데이터마이닝, 상황 인식, 정보 인지 등 다양한

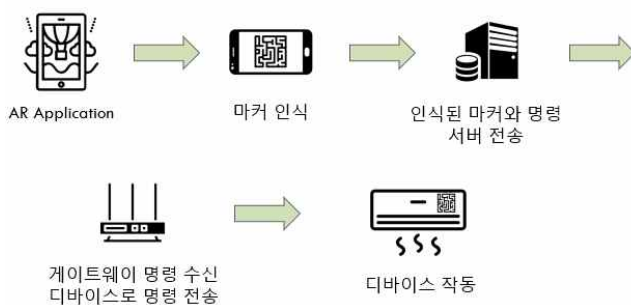
기술들이 있다. 그 중에서도 정보의 인지는 영상, 사운드, 텍스트 등과 같은 다양한 데이터 수단을 이용하여 인지할 수 있는데, 영상데이터를 이용하여 정보를 인지하는 기술이 영상 인식이다. 이는 인간이 외부정보의 70%이상을 시각을 통하여 받아들인다는 점에서 영상 인식을 통한 정보 인지는 보다 직관적인 인터페이스를 제공한다고 볼 수 있다.

그러나 아직까지 일반적인 홈 IoT 디바이스는 전용 어플리케이션이나 컨트롤러와 같이 통합되어 있지 않거나 비직관적인 인터페이스를 통해 제어하는 방식이다. 이에 인터페이스를 시각화 하여 직관적인 인터페이스의 구현에 주목하였다. 따라서 본 논문에서는 직관적인 인터페이스, 그리고 디바이스와 사용자 간의 유기적인 연결을 제공할 것을 목적으로 하는 증강현실 IoT 컨트롤 시스템을 제안한다.

II. 본론

1. MARKER 추적 기반 IoT 디바이스 제어 증강현실 인터페이스

1.1 시스템 구성도



1.2 Android

본 논문에서는 인터페이스를 시각화 하여 직관적인 인터페이스를 구축하기 위해 서브픽셀 알고리즘을 통한 좌표 보정을 통하여 MARKER를 인식하고 보정 및 회전 행렬을 계산함으로써 원본 영상 데이터와 AR을 통합하는 시스템을 통해 구현하였다.

MARKER를 기반으로 한 AR 시스템을 구현하기 위해서는 MARKER 인식을 통하여 AR 시스템이 카메라 위치를 추정을 해야 한다. 위치를 추정하기 위해서는 사전에 영상에서 MARKER를 인식하여 MARKER의 좌표를 찾아야 하는데, MARKER 인식을 하기 위해서는 우선적으로 원본영상을 이진화 시켜야 한다.

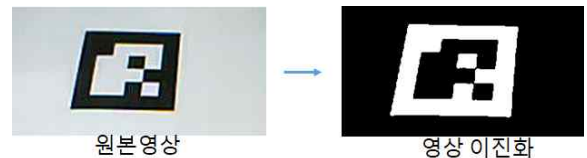


그림 1

이진화 시킨 영상을 가지고 MARKER의 외곽선을 검출하여 검출된 외곽선을 따라 근사화 시켜줘야 한다. 이때 나온 결과를 통하여 사각형 도형을 검출할 수 있게 되는데 이 때 검출된 사각형 도형으로 마진의 크기를 확인해 MARKER를 검출할 수 있게 된다.

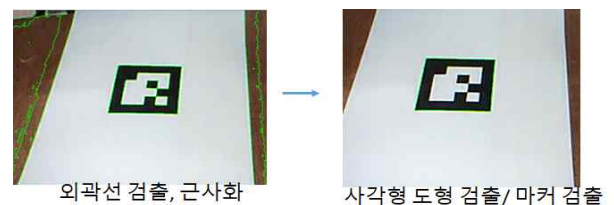


그림2

검출된 MARKER를 캘리브레이션하여 7×7 로 분할하여 바이트코드로 변환을 하는데 바이트코드를 사전에 정의한 MARKER 바이트코드 패턴과 비교하여 해밍거리가 가장 작은 코드의 ID를 반환하게 된다.

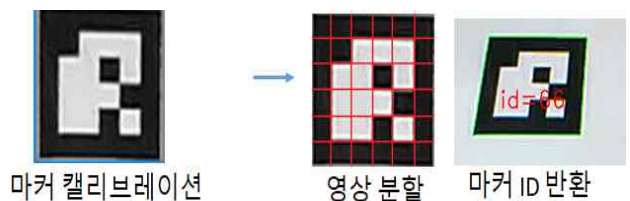


그림 3

MARKER를 인식한 후 AR 시스템의 오브젝트의 위치를 계산해야 한다. 그런 다음 Unity를 사용하여 AR을 구현하기 위해서 MARKER의 좌표를 검출하

여 좌표를 원본 영상과 K-Mean 알고리즘으로 보정을 해 준다.

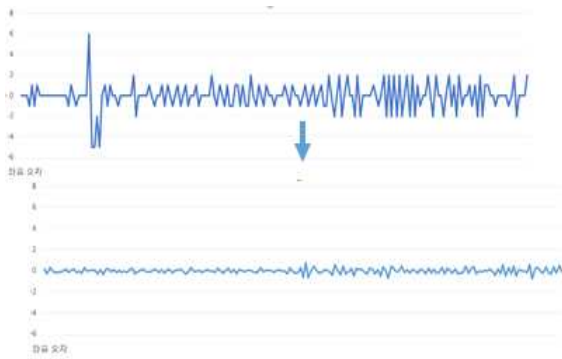


그림 4. K-Means 알고리즘 보정

보정된 결과 행렬을 아래와 같은 행렬을 이용하여 변환행렬 T와 회전행렬 R을 계산해줘야 한다.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\downarrow$$

$$sp_c = K[R|T]p_w$$

$$\downarrow$$

$$sp_c = K[R|T](p_w G) * \text{invertZ} * \text{invertY}$$

그림 7

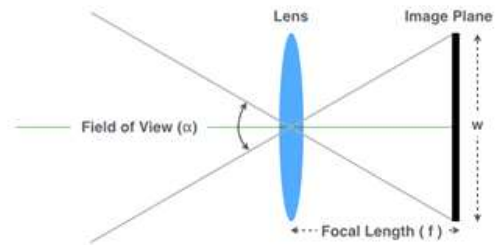
카메라 좌표계의 점 uv와 월드좌표계의 xyz 는 위와 같은 관계가 성립된다. K는 카메라 외부인수로 보정을 위한 행렬을 나타내며, R과 T는 월드좌표계에서 카메라 좌표계로 Projection 하기 위한 변환 회전 행렬로 두 행렬을 구함으로 카메라 위치 추정이 가능하다. Unity에서는 left-handed OpenCV는 right-handed 좌표 시스템이므로 Z와Y로 인버스 시켜 보정시켜준다. 오브젝트 보정행렬 G를 월드좌표계에 계산하여 오브젝트의 크기를 화면에 비례하여 조절시켜준다.

그림 8

Unity의 메인카메라의 FoV는 물리카메라에



중속되어 있지만 AR카메라는 Unity월드 좌표계에 중속되어 있다. 따라서 AR카메라의 Field of View를 계산하여, 영상 Width와 Height에 중속되도록 계산해 주어야 한다.



$$FOV = 2 \cdot \arctan \left(\frac{d_{ImagePixel} \cdot d_{Object}}{2 \cdot r_{Object} \cdot d_{ObjectPixel}} \right)$$

```
double fovScale = (2.0 * Mathf.Atan ((float)(imageSize.width / (2.0 * fx))) /
(Mathf.Atan2 ((float)cx, (float)fx) + Mathf.Atan2 ((float)(imageSize.width - cx), (float)fx)));
double fovScale = (2.0 * Mathf.Atan ((float)(imageSize.height / (2.0 * fy))) /
(Mathf.Atan2 ((float)cy, (float)fy) + Mathf.Atan2 ((float)(imageSize.height - cy), (float)fy)));
```

그림 9

그렇게 하여 AR카메라와 메인카메라의 영상을 통합시킬 수 있게 된다.

그림 10은 본 논문에서 구현한 MARKER을 기반으로 한 AR 시스템에 대한 결과를 보이고 있다.

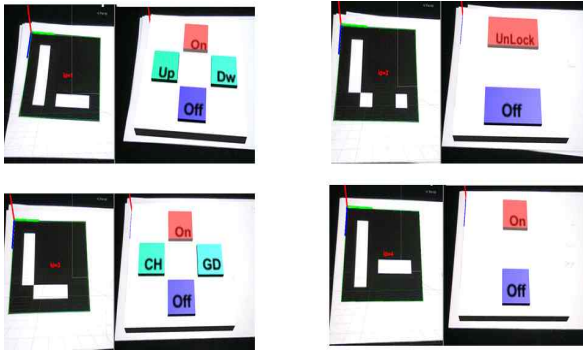


그림 10

1.3 Server

사용자의 명령을 애플리케이션으로부터 받아서 Arduino 프로그램을 통해 디바이스를 제어하기 위하여 본 논문에서는 서버시스템을 구축하였다.

서버 시스템은 JavaScript의 런타임인 NodeJS로 구현하였으며 TCP 서버로서의 역할을 수행함으로써 TCP클라이언트들과의 상호작용을 통해 데이터를 공유하도록 하였다.

TCP 클라이언트 유형은 Android와 Arduino이다. Android로부터 사용자의 명령을 TCP 소켓을 통하여 전달받고, Arduino에게 마찬가지로 TCP 소켓을 통하여 다시 전달해주게 되는데, 이 때, 두 클라이언트에 대한 작동은 ip 대역으로 구분한다.

Android 어플리케이션에서 사용자에게 명령을 실행할 하드웨어 종류와 명령을 문자열로 받아오면 복합 명령인지 단일 명령인지 확인한 후에 해당 데이터베이스에 하드웨어와 명령어 존재 여부를 확인한다. 본 IoT System에서는 연계 동작 기능 또한 구현하였는데, 단일 명령인지 복합명령인지에 따라 분기문을 거쳐서 작동한다. 또한 사용자가 복합 명령을 추가하면 복합명령 데이터베이스에 명령을 추가하는 기능도 구현하였다. 만약 MARKER를 통해 인식된 디바이스에 해당하는 명령을 확인하였으나 단일명령과 복합명령 모두 존재하지 않는 명령에 대하여 입력받게 된 경우 오류메시지를 사용자에게 보낸다. 디바이스에 해당하는 명령을 확인한 경우 Arduino에게 자체 프로토콜에 의해 포맷된 데이터를 전송한다. 이 때 Arduino에 부착된 Wifi Shield를 이

용해 TCP/IP 프로토콜을 통하여 Arduino와 통신한다.

1.4 Arduino

사용되는 Arduino 보드는 두 가지이다. 하나는 리모컨 역할을 하는 보드이며 이 보드의 플래시 메모리 안에는 각 명령에 대한 기준이 되는 RAW 데이터가 저장되어 있으며 웹 서버로부터 TCP 통신을 통해 전달 받은 명령을 인식하여 각 명령에 맞는 RAW 데이터를 이용해 리버스 엔지니어링을 통하여 분석된 정보에 따라 적절한 비트를 조작하여 새로운 RAW 데이터를 생성한다. 그 후 적외선 송신기 모듈을 통해 다른 하나인 게이트웨이 역할을 하는 보드에게 전송한다. 적외선 수신기 모듈을 통해 전달 받은 명령을 확인하고 각 명령에 대해 다른 동작을 한다.

III. 결론

본 논문에서는 직관적인 인터페이스, 그리고 디바이스와 사용자 간의 유기적인 연결을 제공할 것을 목적으로 하는 증강현실 IoT 컨트롤 시스템을 제안한다. 영상인식을 통한 더욱 직관적인 인터페이스로 구현된 본 IoT System은 기기 간 보다 효율적인 연결을 가능하게 한다.