저성능 디바이스를 이용한 자세추정 기반 3D 모델 움직임 제어1)

장재훈°, 최유주*
서울미디어대학원 대학교 인공지능 응용소프트웨어학과
jaehoon0518b@gmail.com, yjchoi@smit.ac.kr

Pose estimation-based 3D model motion control using low-performance devices

Jae-Hoon Jang°, Yoo-Joo Choi*
AI Software Engineeering, Seoul Media Institute of Technology University
*Corresponding Author

요 약

본 논문에서는 저성능 컴퓨터나 스마트폰의 카메라를 통해 입력받은 영상을 기반으로 사용자의 포즈를 추정하고, 실시간으로 사용자의 포즈에 따라 3D 모델의 모션이 제어되어 가시화 될 수 있는 클라이어트-서버 구조의 “자세추정 및 3D 모델 모션 제어 시스템”을 제안한다. 제안 시스템은 소켓통신 기반의 클라이언트-서버구조로 구성되어, 서버에서는 실시간 자세 추정을 위한 딥러닝 모델이 수행되고, 저성능 클라이언트에서는 실시간으로 카메라 영상을 획득하여 영상을 서버에 전송하고, 서버로부터 자세 추정 정보를 받아 이를 3D 모델에 반영하고 렌더링 함으로써 사용자와 함께 3D 모델이 같은 동작을 수행하는 증강현실 화면을 생성한다. 고성능을 요구하는 객체 자세 추정 모듈은 서버에서 실행하고, 클라이언트에서는 영상 획득 및 렌더링만을 실행하기 때문에, 모바일 앱에서의 실시간 증강현실을 위한 자세 추정 및 3D 모델 모션 제어가 가능하다. 제안 시스템은 “증강현실 기반 영상 찍기 앱” 에 반영되어 사용자의 움직임을 따라하는 3D 캐릭터들의 영상을 쉽게 생성할 수 있도록 할 수 있다.

1. 서론

자세추정 기술(Pose Estimation)은 컴퓨터비전(Computer Vision)의 분야중 하나로, 카메라로 촬영된 이미지나 영상에서 객체의 위치와 방향을 추정하여 자세(pose)를 예측하는 기술이다. 이러한 자세추정 기술은 동물, 로봇, 객체등 다양한 대상에 적용할 수 있으며 이 중에서 가장 연구가 활발히 이루어지고 있는 분야는 인간의 자세를 추정하는 인간의 자세추정(Human Pose Estimation) 분야이다. 인간의 자세추정을 위해선 사용되는 사진이나 영상에서 주로 인간의 신체관절의 위치와 자세를 추정하는 작업이 필요하다. 인간의 자세추정을 위해 사용된 전통적인 방법 중 하나는 키넥트(Kinect)와 같은 모션캡처(Moticon Capture) 장비나 센서를 부착하여 파악하는 방법이 사용되었다. 이 방법은 실시간으로도 인간의 정교한 움직임을 파악할 수 있지만 장비를 구축하기 위해 높은 비용이 필요할 수 있으며 실생활에선 센서장비를 착용하고 다니지 않는다는 점, 키넥트 장비의 경우 일정한 크기의 공간이 있어야만 사용가능하다는 점[1]과 같은 어려움으로 인해 실생활에서 일반적으로 사용하기에는 어려움이 있다. 그러나 딥러닝 기술발전의 영향으로 이러한 장비를 사용하지 않고도 단일 2D이미지나 영상만을 사용해서 2차원 또는 3차원의 자세도 추정이 가능해지면서 자세추정에 대한 연구가 활발해질 수 있었고 다양한 논문들과 쉽게 사용가능한 자세추정 오픈소스 라이브러리 및 프레임워크들이 나올 수 있게 되었다. 딥러닝 기술발전의 영향으로 자세추정기술을 사용하기 위해 더 이상 센서와 같은 물리적인 장비를 필수로 요구 하진 않지만 많은 계산량과 데이터세트를 필요로 하기 때문에 자세추정기술을 사용하기에는 여전히 사용되는 시스템의 성능이나 GPU를 사용하지 않는 모바일기기의 경우에는 제한점이 있다.

본 논문에선 서버-클라이언트 구조를 적용하여 사용되는 장비성능의 영향을 최소화 하여 자세추정기

1) 본 연구는 문화체육관광부 “관광서비스 혁신성장 연구개발사업” (R2022020105)의 지원에 의하여 수행되었음

술을 사용할 수 있는 방법을 제안한다. 2절에서는 딥러닝 기반의 오픈소스 인간자세추정 기술들에 대해 설명하고, 3절에서는 서버-클라이언트 구조를 적용하여 자세추정기슬을 사용한 방법에 대해 설명한다.

2. 자세추정 관련 오픈소스 라이브러리

2.1 인간 자세추정의 방법

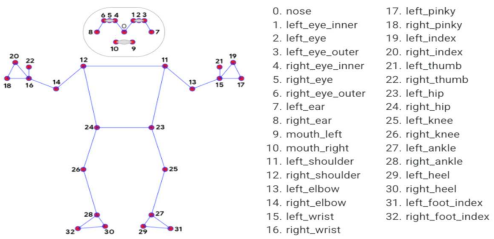
인간의 자세추정기술의 방법으론 크게 2가지 하향식(Top-Down)방법과, 상향식(Bottom-Up)방법으로 분류된다[2] 하향식 방식은 이미지나 영상에서 먼저 인간의 위치를 감지한 뒤 감지된 영역 안에서 인간의 관절에 해당하는 키포인트(keypoint)의 구조를 추정하는 방식으로 먼저 인간의 위치를 인식하고 추정이 이뤄지기 때문에 정확도는 높지만, 영상에 여러명의 인간이 있을 경우 속도가 느리다는 단점이 있다. 상향식 방식은 인간의 키포인트를 먼저 찾은 뒤 상관관계를 분석하여 키포인트를 연결한 뒤 인간의 자세를 추정하는 방식으로 사람을 인식하는 과정이 없기 때문에 속도는 빠르지만 하향식에 비해 정확도가 떨어진다는 단점이 있다.

2.2 자세추정 오픈소스 라이브러리 OpenPose

상향식 방식의 자세추정 오픈소스 라이브러리로는 오픈포즈(OpenPose)[3]가 있다. 오픈포즈는 2017 CVPR에서 발표된 딥러닝 기반의 자세추정 오픈소스로 라이브러리로, 상향식 방식이기 때문에 몸,손, 얼굴 등 총 135개 신체 키포인트를 먼저 추정한 뒤 이를 연결하여 실시간으로 시각화 한다.

2.3 자세추정 오픈소스 라이브러리 MediaPipe

하향식 방식을 사용하는 자세추정 오픈소스라이브러리로는 미디어파이프(MediaPipe)[4]가 있다. 미디어파이프는 구글(Google)에서 개발한 오픈소스 비전 및 AI 프레임워크로 컴퓨터 비전에 관련된 다양한 이미지 데이터 처리 및 얼굴인식, 객체인식, 손의 제스처 인식등 다양한 기능을 제공한다.

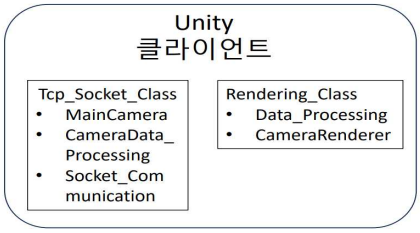


(그림 1) MediaPipe Pose의 키포인트 목록

이중 신체의 포즈를 추정하는 모델의 키포인트는 33개의 키포인트를 사용한다. (그림 1)은 미디어파이프를 통하여 추정할 수 있는 33개의 관절 키포인트를 보여주고 있다.

3. 클라이언트-서버 구조의 자세추정과 렌더링

본 논문에선 TCP 프로토콜(Transmission Control Protocol)기반의 소켓(Socket)을 적용하여 구축된 클라이언트(Client)와 서버(Server)의 구조에서 자세추정을 적용한다. 인간의 자세추정을 위한 오픈소스 라이브러리는 미디어파이프를 사용하였고 클라이언트 프로그램의 작성은 유니티(Unity)엔진을, 서버 프로그램의 작성은 파이썬(Python)으로 작성하였다. 소켓은 소프트웨어로 구현시킨 추상적인 포트를 말한다[5]. 소켓을 통해 IP주소와 포트번호를 사용하여 디바이스간의 통신 및 클라이언트와 서버간의 통신을 구축할 수 있다. 클라이언트 프로그램을 유니티 엔진으로 사용한 이유는 크로스 플랫폼을 지원하기 때문에 PC가 아닌 IOS, Android, ,콘솔, 태블릿과 같은 다양한 디바이스에서의 사용이 가능하기 때문이다. 클라이언트 프로그램의 주요 기능은 크게 3가지로, 첫째는, 자세추정을 적용하기 위한 입력영상을 실시간으로 받아와야할 카메라 기능, 둘째는, 입력받는 카메라 영상과 영상에서 추출된 메타정보를 서버와 주고받을 수 있는 통신기능, 셋째는, 서버로부터 전송받은 정보를 바탕으로 추정된 자세를 시각화할 렌더링기능이 필요하다.



(그림 2) 클라이언트 주요 클래스 구성

실제 구현을 위해 주요 기능들을 (그림 2)와 같은 클래스와 모듈단위로 구성하였다. Tcp_Socket_Class의 주요모듈의 구성은 다음과 같다. MainCamera모듈은 카메라관련 모듈로 클라이언트 프로그램이 설치된 디바이스에서 사용가능한 카메라 목록을 가져온 뒤 초기화 후 사용자가 지정한 카메라를 구동 시킨다. CameraData_Processing 모듈은 입력되는 카메라 데이터의 전처리를 담당하는 모듈이다. 유니티 엔진은 웹캠 영상을 텍스처(Texture)형식으로 받아

오기 때문에 텍스처의 이미지포맷은 RGB24로 지정하여 이미지의 픽셀(pixel)정보를 가져온 뒤 Jpeg타입으로 인코딩(Encoding)하는 전처리를 진행한다.

Socket_Comunication 모듈은 서버프로그램과의 통신구축 및 데이터의 송수신을 담당하는 모듈이다. 통신설정을 위해 서버와 동일한 IP주소와 포트번호, 프로토콜(TCP)을 설정하여 소켓을 생성한 뒤 전처리된 데이터를 소켓을 통해 서버로 전송하기 위해 바이트(Byte)타입으로 인코딩한 뒤 프레임단위로 나눠서 전송 하게 된다. 이후 서버프로그램에서 미디어파이프를 통해 추정된 신체의 키포인트 위치(좌표)값을 다시 소켓을 통해 수신 받게 된다.

Renderering_Class는 서버로부터 전송받은 데이터를 사용하여 입력영상위로 렌더링 작업을 하기 위한 클래스이다. Data_Processing모듈은 Tcp_Socket_Class를 참조하여 서버로부터 전송된 데이터를 유니티엔진에서 사용하기 위한 전처리를 담당하는 모듈이다. 서버로부터 전송받은 위치데이터를 문자열로 디코딩(decoding)한 뒤 유니티엔진에서 사용되는 위치데이터 포맷인 벡터(Vector)타입으로 변환하는 작업을 담당한다. CameraRenderer모듈은 렌더링을 통한 자세추정의 시각화작업을 담당하는 모듈이다. Data_Processing모듈을 통해 전처리된 데이터를 실시간으로 실행되고 있는 카메라 영상위로 렌더링을 통한 시각화를 해줌으로서 실시간으로 움직이는 인간의 자세를 추정할 수 있게 된다.

서버 프로그램의 주요 기능은 첫째, 클라이언트 프로그램과의 통신의 형성 및 클라이언트 프로그램과의 데이터 송수신 기능 둘째, 클라이언트로부터 받은 입력영상기반의 자세정보 추정 관련 기능이 필요하다. 서버 프로그램의 Socket_Comunication 모듈도 클라이언트와 동일한 IP주소와 포트번호 프로토콜을 (TCP) 설정한 소켓을 생성하여 클라이언트와의 통신을 구축한 뒤 , 연결된 소켓을 통해 클라이언트로부터 전송되어지는 영상을 실시간으로 수신하게 된다. 수신된 데이터는 바이트타입으로 인코딩된 데이터이기 때문에 파이썬에서 사용하기 위해 넘파이(numpy)타입의 데이터로의 디코딩 및 변환이 이루어지는 전처리가 이루어진다. FindPosition모듈은 Socket_Comunication모듈에서 전처리가 이루어진 데이터를 기반으로 실제 자세추정의 작업이 이뤄지는 모듈로, 미디어파이프에서 제공되는 자세 감지기를 사용하여 사람의 자세로 추정되는 33개의 키포인트 위치값에 대한 각각의 x,y,z 값을 구하여 소켓으로 넘겨주게 되면 소켓을 통해 클라이언트 프로그램으로의 전송이 이루어진다.

4. 구현 결과

제안 시스템은 서버를 위한 시스템으로 Geforce RTX 3090 GPU가 탑재된 시스템에서 파이썬 프로그램을 통하여 미디어파이프를 수행하여 객체의 포즈를 추정하도록 구현하였다. 또한, 클라이언트는 Unity3D를 통하여 구현하여, 웹캠이 설치된 고사양 GPU가 없는 일반 데스크탑과 모바일 장비에서 구동하도록 하였다. (그림 3)은 클라이언트 시스템에서 캡처한 서로 다른 포즈의 자세추정 결과에 따른 스켈리톤 3D 모델의 움직임 결과 화면이다.

(그림3) 실시간으로 이뤄지는 자세추정

5. 결론

본 논문에선 실시간으로 이뤄지는 딥러닝 기반의 자세추정기술에 클라이언트-서버구조를 적용하였다. 서버 프로그램이 자세추정의 필수정보인 키포인트의 추정 및 관련 메타정보의 계산을 담당하고 , 클라이언트프로그램은 카메라 영상의 입력 및 렌더링의 기능만 담당하게 함으로서 서버(PC)와 통신연결만 되어있다면 클라이언트로 사용되는 장비의 성능은 상대적으로 부족하더라도 실시간으로 자세추정기술을 사용할 수 있도록 하였다.

참고문헌

[1] 안희권 , "MS 키넥트의 몰락...그 이유는?" , 아이뉴스24 , 2017.

[2] Yang, Sen, et al. "Detecting and grouping keypoints for multi-person pose estimation using instance-aware attention." Pattern Recognition 136 (2023): 109232.

[3] Cao, Zhe, et al. "Realtime multi-person 2d pose estimation using part affinity fields." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[4] "google/mediapipe", github , last modified Sep 13.2023 , accessed Sep 13.2023 , https://github.com/google/mediapipe.

[5] “Socket 소켓”, 정보통신기술용어해설, 2022년10월21일 수정, http://www.ktword.co.kr/test/view/view.php?m_temp1=280