

3D 모션 캡처 참고 자료

학과:컴퓨터공학부

학번:201802907

이름:이정우

일반적인 단일 모션캡처 앱은 시중에 많이 존재, 다만 사용자가 재미를 느낄 수 있는 엔터테인먼트적인 요소와 결합이 필요.

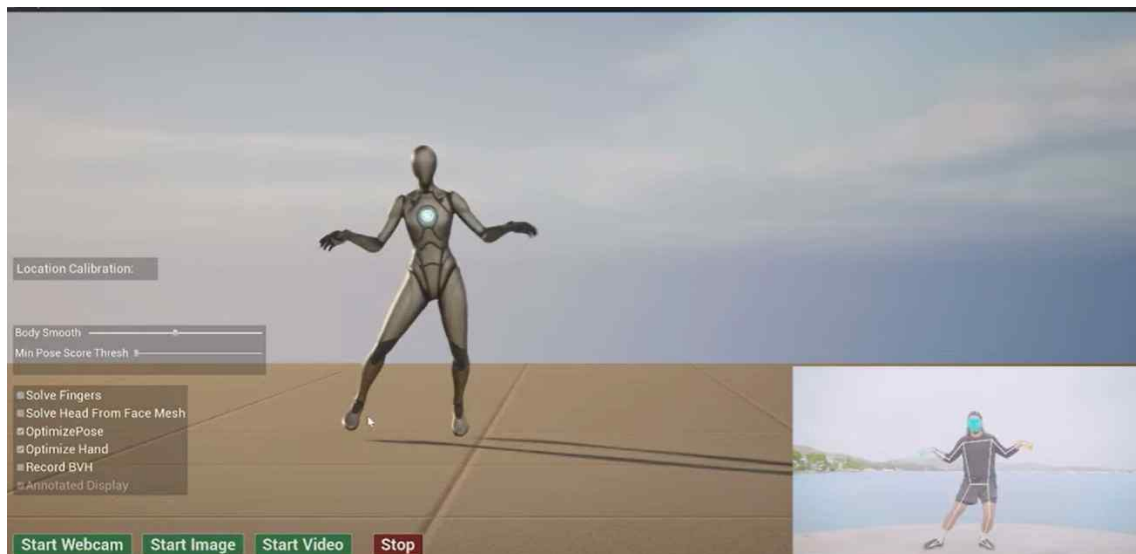
예) 복싱 게임, 버추얼 유튜버, 운동 자세 교정 어플리케이션,

1. 언리얼 엔진 (MediaPipe4U-Demo)

링크

데모 영상:https://www.youtube.com/watch?v=cop7_kCaDn4

깃허브:<https://github.com/endink/Mediapipe4u-plugin/tree/main?tab=readme-ov-file>



언리얼 엔진에서 제공하는 미디어파이프4u 플러그인 사용 rigging 방식

장점: 압도적인 그래픽 성능, 부드러운 모션 캡처

단점: 무거운 용량, 모바일 x, 인공지능 모델을 빌드하기 어려움.

2. 유니티 (UnityPythonMediaPipeAvatar)

링크

데모 영상: <https://www.youtube.com/watch?v=icS2yforZpw>

깃허브: <https://github.com/ganeshsar/UnityPythonMediaPipeAvatar>



스크립트 언어로 파이썬을 사용, opencv + mediapipe 파이썬으로 카메라를 받아와서 스켈레톤 좌표를 바탕으로 유니티 화면에다 아바타 렌더링, rigging 방식

장점: 준수한 아바타 퀄리티, 인공지능 모델 연동 용이

단점: 유료화 기능 제한, 모바일 연동이 안되는 것은 아니지만 다소 어려움, 되더라도 실행이 느릴 것으로 예상

3. three.js (kalidokit)

데모영상:<https://www.youtube.com/watch?v=UegatFGY4TQ>

깃허브:<https://github.com/yeemachine/kalidokit>



장점: 모바일 연동 가능, 인공지능 모델 빌드 용이

단점: 다소 아쉬운 캐릭터 퀄리티

이외 알아낸 사항

1. avatar mocap 프로젝트 중 opencv로 받아온 것과 그렇지 않은 것이 존재한다.
2. 카메라 함수를 게임 엔진 내장 카메라와, opencv를 동시에 사용했을 때(혹은 three.js 자바스크립트 프론트엔드 카메라와 opencv) 권한 문제가 발생한다. (메소드 충돌로 인한 failed to load camera 오류)
3. 인공지능 모델은 파이썬으로 작성되므로 opencv와 아바타 코드 동시에 쓸 수 없다.
4. opencv는 pc에 최적화, 이미지 인코딩 방식, three.js 상에 3d 공간에 opencv로 받은 미디어파이프 스켈레톤 좌표를 실시간으로 나타내는 것이 불가능하다.
5. 미디어 파이프는 자바스크립트와 파이썬 모두 지원한다.

흐름도



AJAX: 비동기적으로 웹 서버와 브라우저 간에 데이터를 교환하는 기술.
웹 페이지를 새로 고치지 않고도 동적으로 데이터를 로드하고 표시

API 정의서

AJAX

(자바 스크립트 코드) function onResults(results) 함수에 해당 코드 삽입

```
$.ajax({
  url: '{% url 'test' %}', // URL을 템플릿 태그로 설정 해당 url로 데이터 전송
  type: 'POST',
  headers: {
    'X-CSRFToken': '{{ csrf_token }}' // CSRF 토큰 설정
  },
  data: dataX, // 좌표 등의 데이터를 장고 서버로 보냄

  success: function(data) {
    console.log('Data sent successfully');
    var jsonData = data.json_data; // 장고 서버에서 받은 데이터 정의
    console.log(jsonData);

    //HTML에 텍스트 로드
    var receivedDataElement = document.getElementById('received_data'); // HTML 요소
    선택
    receivedDataElement.innerHTML = jsonData; // HTML

  },
  error: function(xhr, status, error) {
    console.error('Error sending data:', error);
    // 오류가 발생한 경우 처리
  }
});
```

장고 Views.py

```
def test(request):
    if request.method == 'POST':
        leftShoulderXCoordinate = request.POST.get('leftShoulderXCoordinate')

        (데이터 받는 예시)

        # 받은 데이터 input_data로 변환

        predictionsTest = model.predict(input_data) #인공지능 모델 계산 후
        probability_class1 = predictionsTest[0][1] 자세 1

        # 클래스1에 해당하는 확률

        probability_class2 = predictionsTest[0][2] 자세 2

        # 클래스2에 해당하는 확률
```

```
DBModel.objects.create(label_1=probability_class1)

latest_entry = DBModel.objects.latest('id') # 데이터 베이스
json_data = latest_entry.label_1[:5] #5째 자리수까지 표현 레이블 1의 데이터

return JsonResponse({'json_data': json_data})
```

Models.py

#데이터 베이스 모델 추후 개선 예정

```
class DBModel(models.Model):
    label_1 = models.CharField(max_length=100)
```

+ 인공지능 자세 교정 모델