



메타버스에 모션인식을 활용한 사용자 인증

팀명: 하이나비
팀원: 이창열, 차정현, 이진현
지도교수님: 서승현

2021년 캡스톤디자인2 결과발표회

구상동기

가상세계와 현실 넘나들다... '메타버스' 열풍

Z세대 놀이터 뛰어넘어 글로벌 시장 선도할 미래 신산업으로 급부상
정부, 디지털 뉴딜 '메타버스'로 확장...초연결·초실감 산업 집중 육성

코로나 시대 급부상한 '메타버스', 들어보셨나요?

| 비대면의 일상화 기대, 상호 소통 가능한 가상공간의 확산

 디지털조선일보

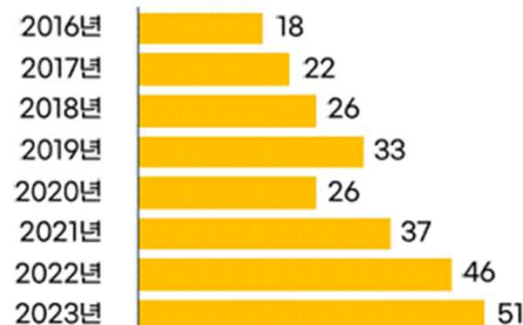
'부캐'로 나를 알리는 시대... 가상 현실 플랫폼 '제페토'



네이버 제페토

가상현실 산업 규모

* 2020년부터 전망치



메타버스 관련 기술시장 규모

(단위: 억달러)



LAB.M 메타버스사업현황

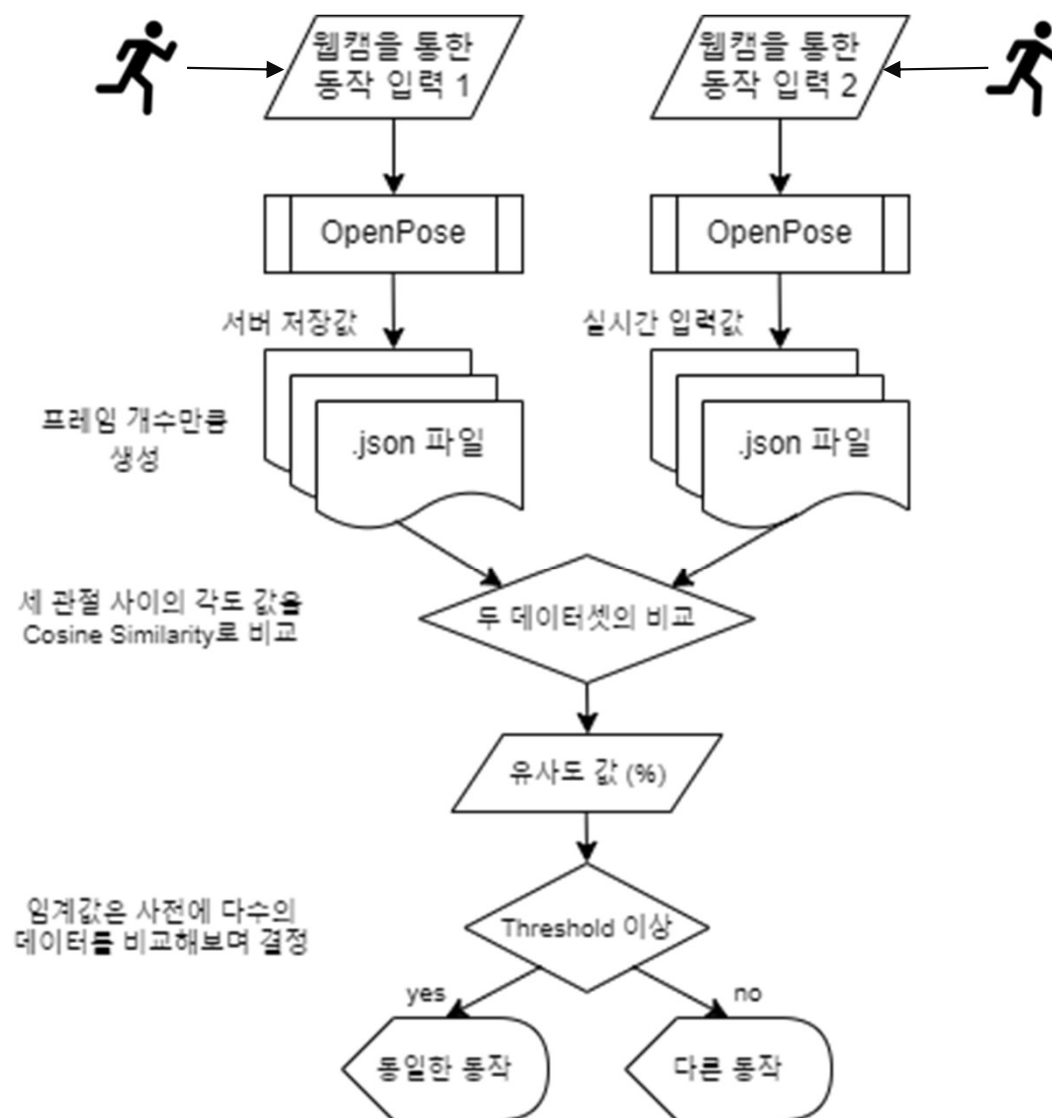
아이뉴스24

[新 미래 메타버스] ① 성큼 다가온 "레디 플레이어 원"



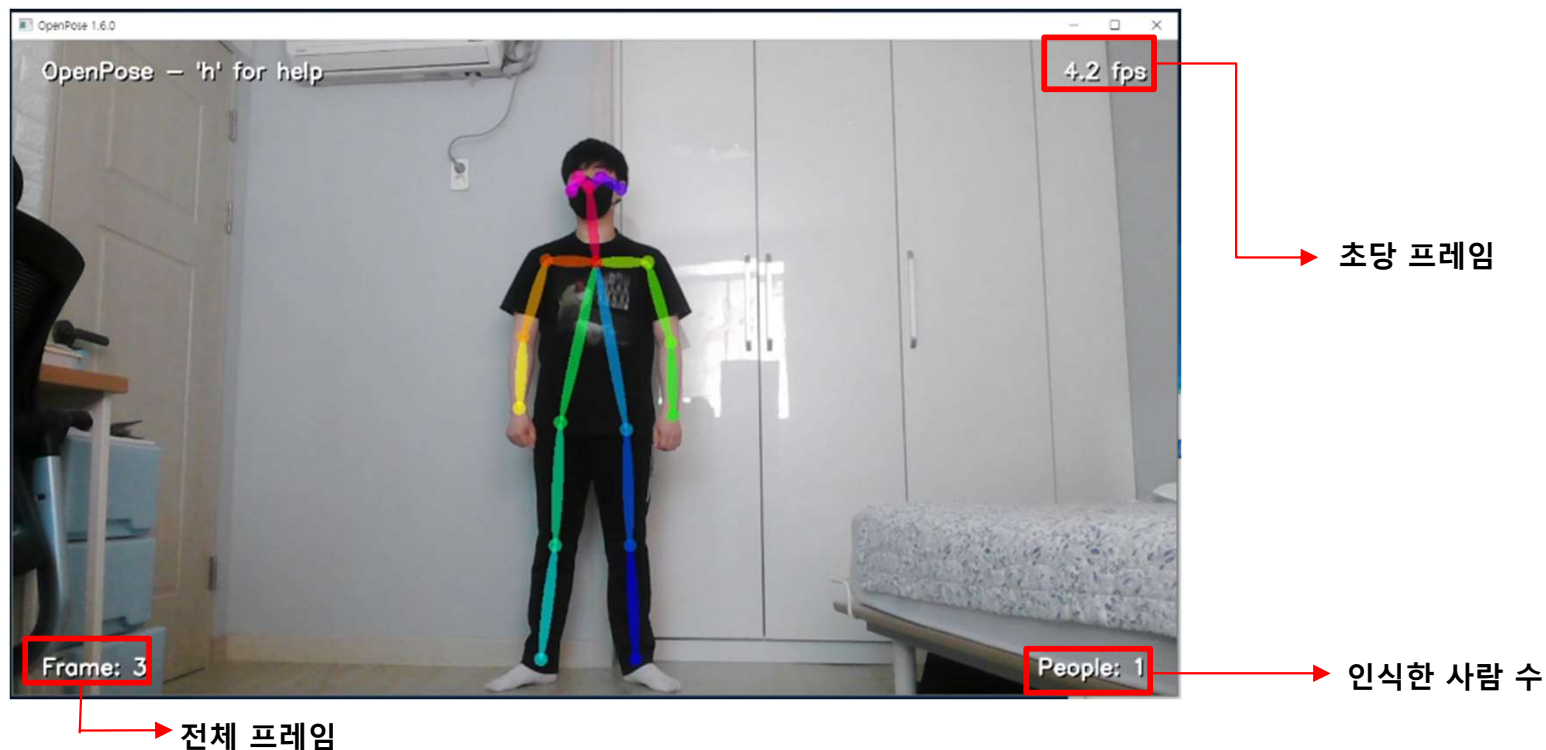
영화 <레디 플레이어 원 (2017)>

Flowchart

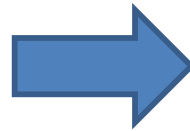
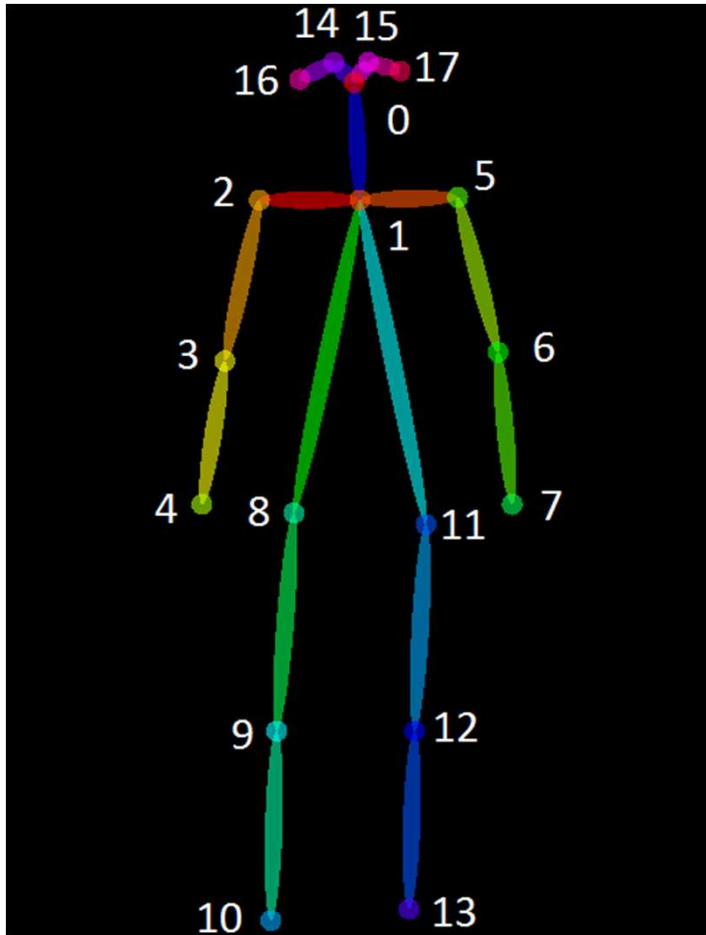


제목


- OpenCV 기반의 모션인식 코드이다. 실시간 영상에 대하여 모션인식이 가능하며 사용가능한 모델은 Body25, COCO, MPI가 있다.
- OpenPose에서 인식한 모션의 좌표값을 Json파일로 추출이 가능하다.



COCO model

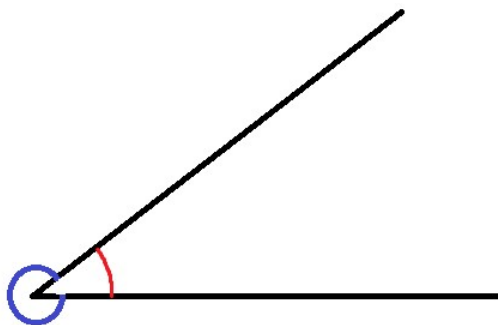


| ?? | x 좌표 | y 좌표 |
|----|---------|---------|
| 0 | 402.516 | 99.6382 |
| 1 | 403.818 | 161.036 |
| 2 | 365.938 | 161.028 |
| 3 | 352.902 | 215.849 |
| 4 | 349.02 | 265.433 |
| 5 | 444.262 | 161.039 |
| 6 | 463.886 | 215.857 |
| 7 | 484.703 | 265.49 |
| 8 | 381.578 | 274.586 |
| 9 | 364.632 | 362.046 |
| 10 | 347.647 | 425.977 |
| 11 | 433.819 | 275.877 |
| 12 | 442.96 | 360.756 |
| 13 | 462.535 | 423.369 |
| 14 | 393.346 | 95.7466 |
| 15 | 411.641 | 90.6015 |
| 16 | 381.672 | 108.834 |
| 17 | 423.353 | 102.324 |

 000000000094_keypoints.json

Similarity

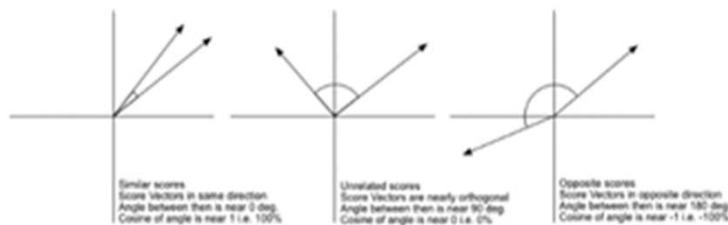
■ 외적



- 2개의 벡터사이 각도를 구할 경우 두개의 각도 중에 작은 각도를 출력한다.
- 항상 같은 방향의 각도를 출력하기 위하여 외적의 특성을 사용하였다.

■ Cosine Similarity

$$\text{similarity} = \cos\theta = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2} \sqrt{\sum B_i^2}}$$



- 두 벡터간의 각도 차이를 이용하여 유사도를 판별하는 방법
- 두 벡터가 같을 경우 1, 다를 경우 -1이 된다.

Similarity

```
double external_csin(double x1, double y1, double x2, double y2, double x3, double y3) {
    //external_product(외적 계산)
    double x_move = x1 - x2;
    double y_move = y1 - y2;
    double x_standard = x3 - x2;
    double y_standard = y3 - y2;
    double external = (x_standard * y_move) - (x_move * y_standard);
    double R = 2.0;
    if (external >= 0) {
        double result = acos(cosin(x_move, y_move, x_standard, y_standard));
        return result;
    }
    else if (external < 0) {
        double revers_csin = 6.28378 - acos(cosin(x_move, y_move, x_standard, y_standard));
        return revers_csin;
    }
}

//유사도 계산
double sim0 = cos(fabs(server_csin0 - live_csin0));
double sim1 = cos(fabs(server_csin1 - live_csin1));
double sim2 = cos(fabs(server_csin2 - live_csin2));
double sim3 = cos(fabs(server_csin3 - live_csin3));
double sim4 = cos(fabs(server_csin4 - live_csin4));
double sim5 = cos(fabs(server_csin5 - live_csin5));
double sim6 = cos(fabs(server_csin6 - live_csin6));
double sim7 = cos(fabs(server_csin7 - live_csin7));
double sim8 = cos(fabs(server_csin8 - live_csin8));

double avesim = (((sim0)+(sim1) + (sim2) + (sim3) + (sim4) + (sim5) + (sim6) + (sim7) + (sim8) ) / 9) * 100;
```

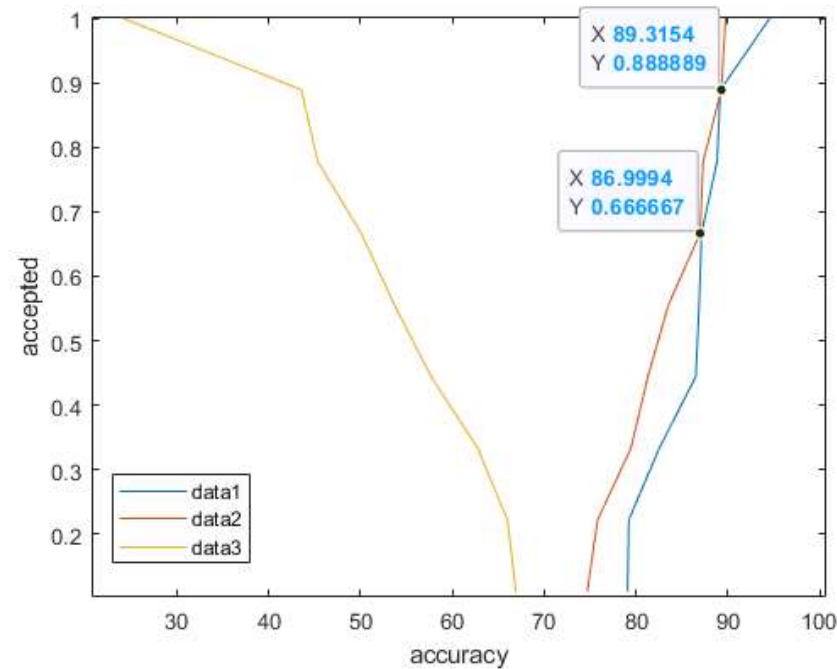
- 외적을 이용하여 3점의 사이의 각도를 구한다.
- 서버에 저장된 값과 실시간 각도 값의 차를 이용하여 유사도를 구한다.
- 각 프레임에 대한 유사도 평균을 구한다.
- 전체 프레임에 대한 유사도를 구한다.

수행결과

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 동일 인물 동일 행동 | 79.2382 | 88.8495 | 89.2086 | 86.8950 | 94.5827 | 79.0971 | 87.1580 | 82.5556 | 86.5308 |
| 다른 인물 동일 행동 | 79.4667 | 89.7771 | 75.8408 | 74.7081 | 83.5522 | 86.9994 | 87.3281 | 89.3154 | 81.3169 |
| 동일 인물 다른 행동 | 53.7106 | 43.5820 | 45.3653 | 24.1548 | 62.7897 | 57.7109 | 66.9424 | 66.0200 | 50.0963 |

- 위 데이터를 통해서 Threshold를 정해야 합니다.
- 위 데이터에서 다른 사람이 동일 행동을 했을 때 또한 유사도가 높음을 확인 가능하다.

수행결과 및 결론



Data1 : 동일 사용자 동일 행동
Data2 : 다른 사용자 동일 행동
Data3 : 동일 사용자 다른 행동

- 결과 값에 따라 Threshold 값을 80으로 정했다.
- 행동을 구분이 가능하나 사용자를 구분하지 못하다
- Secondary 인증으로서 역할이 가능하다.

Thank you.

