

데이터 기반 실시간 의류 시뮬레이션

지도교수: 이인권 교수님

지도조교: 이도해 조교님

팀: Tailor

전영주 김성하 이규식

1. 연구주제

CPU를 활용하여 사실적인 실시간 의류 시뮬레이션을 개발 및 구현한다.

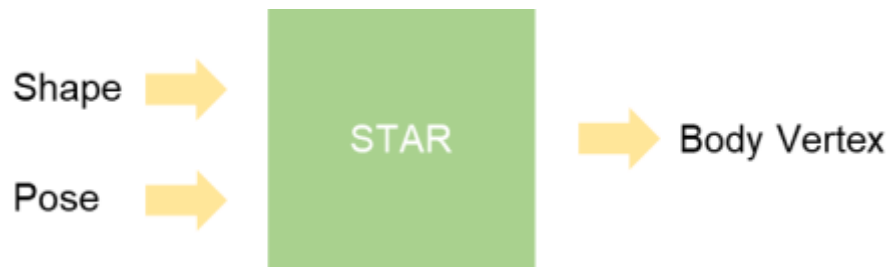
2. 연구의 필요성

미래에는 메타버스와 게임과 같은 VR/AR 애플리케이션들이 많이 사용될 것이다. 이러한 애플리케이션을 사용할 때 사용자가 잘 몰입하고 현존감을 느끼기 위해서는 사실적인(realistic) 사람의 묘사가 필요하다. 또한, 옷이 몸의 80% 이상을 차지한다는 점을 생각하면 사실적인 옷 묘사는 매우 중요하다.

하지만 사실적인 옷 묘사를 하려면 물리 기반 시뮬레이션이 가장 좋지만, 많은 자원과 시간이 필요하다. 이는 실시간성(real-time)이 필수적인 애플리케이션의 성격을 고려해 보았을 때 부적합하다. 따라서 많은 연구가 사실성을 일부 포기하는 대신 실시간성을 확보한 기법들을 소개해왔다.

본 연구진은 위와 같이 사실성을 일부 포기한 최첨단 연구 중에서 실시간성을 손해 보지 않고도 사실성을 크게 더하는 방법을 제시한다.

2.1. 기존 연구



[그림 1]

기존의 cloth simulation을 한 연구들은 정형화되어 있는 사람의 Body Vertex 위에 옷을 입혔다. 이런 정형화되어있는 사람의 Body Vertex를 구하기 위해 STAR 모델을 사용한다(그림 1). STAR는 몸의 모양을 나타내는 Shape와 몸의 자세를 나타내는 Pose를 받으면 Body Vertex를 output 한다. 기존의 연구는 STAR의 input인 Pose와 Shape를 입력했을 때 몸의 모양에 맞는 옷의 vertex를 출력하는 학습모델을 구현하였다.

2.2. 기존 연구의 한계점

STAR가 정형화되어 있어 사용에 편리하지만, 표현할 수 있는 몸의 범위가 제한되어있다는 치명적인 단점이 있다. 예를 들어 STAR는 동양인을 잘 표현하지 못하는데 이는 STAR를 학습시킬 때 사용되었던 샘플들의 대다수가 서양인이기 때문이다. 또 정형화된 몸을 표현하기 때문에 비대칭 등의 real-human의 섬세한 표현을 잘하지 못한다.

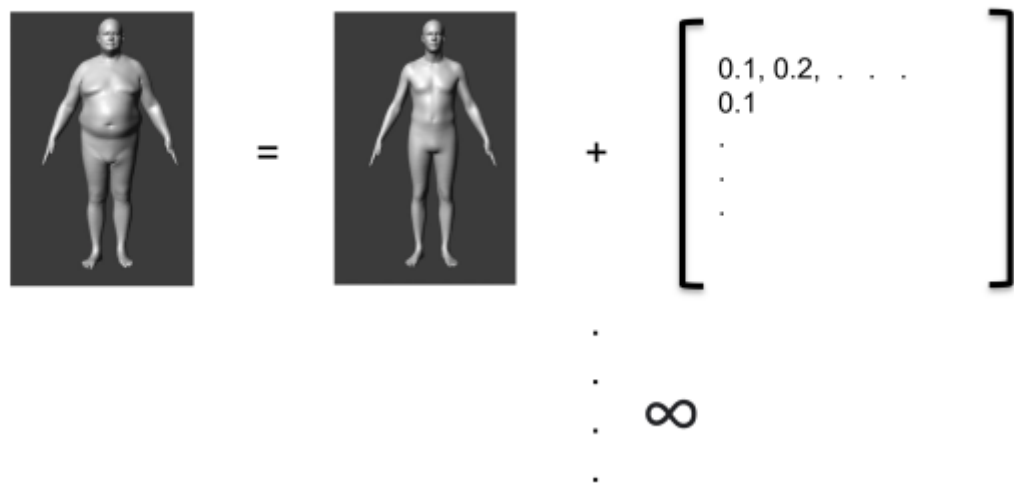
2.3. 초기 연구

본 연구진은 초기 연구에서 옷의 vertex를 추론하는 학습모델에 Shape, Pose 외에 Displacement를 input으로 추가하여 더 세밀하게 표현된 몸에 옷 표현을 하려 했다. Displacement란, 우리가 실제로 표현하고 싶은 사람의 몸 Vertex와 이를 STAR로 표현한 Body Vertex와의 차이를 의미한다. 이를 이용하여 기존의 STAR보다 더 다양한

사람의 몸을 표현할 수 있다. 기존의 STAR에서 Displacement를 추가한 이 표현법을 STAR-D라고 명명한다.

3. 연구내용

$$\text{STAR}(\text{Shape}, \text{Pose}) = \text{STAR}(\text{Shape}^*, \text{Pose}) + \text{Displacement}$$



[그림 2]

3.1. Shape/Pose/Displacement 조합

새로운 모델을 학습시키려면 input으로 넣을 Shape/Pose/Displacement의 조합을 만들어야 한다. 기존에 Shape와 Pose의 쌍은 주어진 것을 쓰면 되지만, Displacement로 임의의 값을 넣으면 사람의 Body Vertex가 자연스럽지 않기 때문에 적절한 값들을 알아내는 것이 관건이었다. 본 연구진은 STAR를 이용하여 기존의 Body Vertex를 이루는 Shape와 Pose의 쌍과 Shape는 다르지만, Pose는 동일한 Shape와 Pose의 쌍으로 적절한 Displacement를 알아내어 무수히 많은 Shape/Pose/Displacement의 조합을 만들 수 있음을 발견하였다[그림 2].

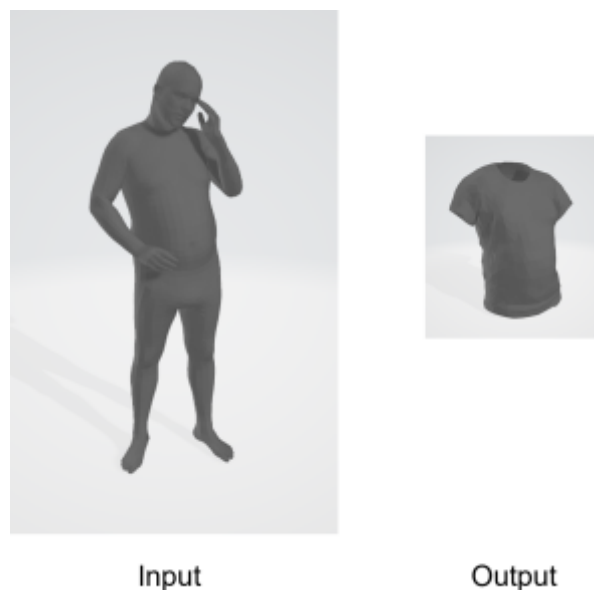
이를 이용하여 Shape/Pose/Displacement를 입력받았을 때 해당하는 Body Vertex에 맞는 Cloth Vertex를 출력하는 모델을 구현하였다. Displacement를 생성하는 구체적인 내용은 다음과 같다.

3.2. Displacement 생성

먼저 기존의 Body Vertex와 그 Body Vertex에 입힌 Cloth Vertex가 필요하다. 기존의 Body Vertex를 만드는 데에 사용한 Shape를 조금 변경하여 새로운 Shape를 만든다. 새로운 Shape와 기존의 Pose를 이용하여 새로운 Body Vertex를 얻어낼 수 있고, 이것과 기존의 Body Vertex의 차이를 구하면 Displacement가 된다.

3.3. 초기 모델 학습

위와 같은 방법으로 다수의 Shape/Pose/Displacement를 생성하였고, 이를 통해 학습을 진행하였다. [그림 3]은 은 학습이 완료된 후에 모델을 테스트한 결과이다.



[그림 3]

3.4. 초기 연구의 한계점

초기 연구에 대해서 반팔 상의는 적절한 옷을 출력하지만, 긴팔 상의는 손에 가까울수록 잘 맞지 않은 옷을 출력하였다. [그림 4]에서 왼쪽은 초기 연구 모델로 예측을 진행한 결과이고, 오른쪽은 물리 기반 시뮬레이션으로 출력한 결과이다. 왼쪽 옷의 왼쪽 소매 끝부분이 매끄럽지 못하게 표현되었다.

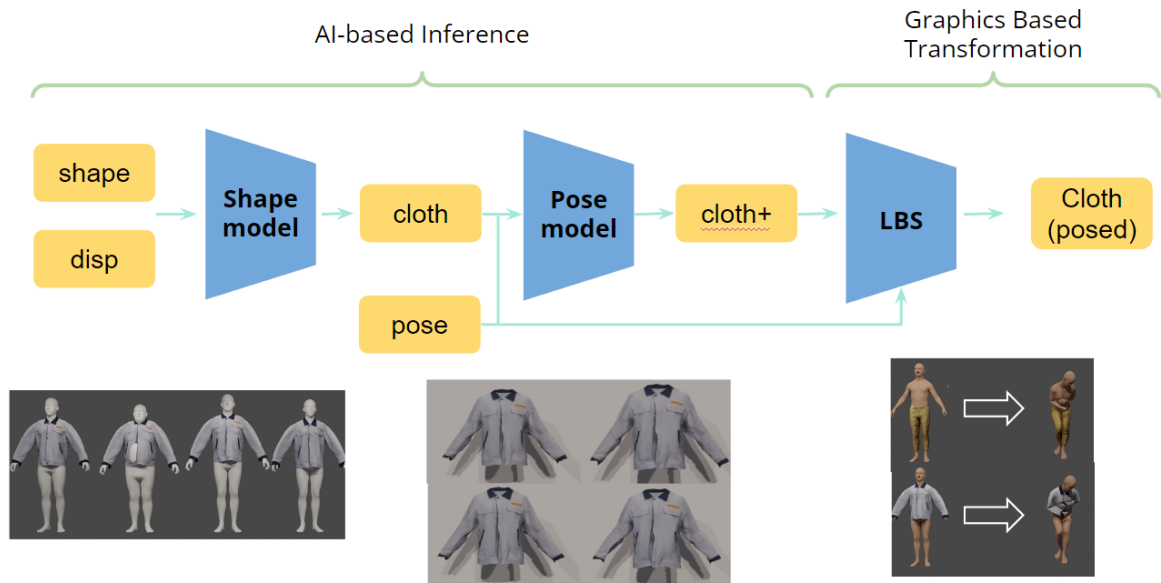


[그림 4]

팔은 다른 부위와는 다르게 몸의 중심에서 멀기 때문에 pose에 따라 달라지는 양이 매우 크고, 이 때문에 A.I. 모델이 이를 적절히 학습하지 못한 것으로 보인다. 따라서 이를 보완하기 위해 개선된 모델을 고안하였다.

3.5. 개선된 연구

초기 연구의 한계점이 명확하여 개선된 연구에서는 다양한 포즈에 대해서도 예측을 정확하게 할 수 있도록 모델을 개선하였다. 기존의 A.I 모델은 다양한 pose를 잘 표현하지 못했기 때문에, A.I 모델은 shape와 옷의 detail을 담당하고, 그 후에 Graphics 기술인 LBS를 통해 다양한 포즈를 표현 가능하게 하였다.



[그림 5]

[그림5]는 최종 모델의 구조를 나타낸다.

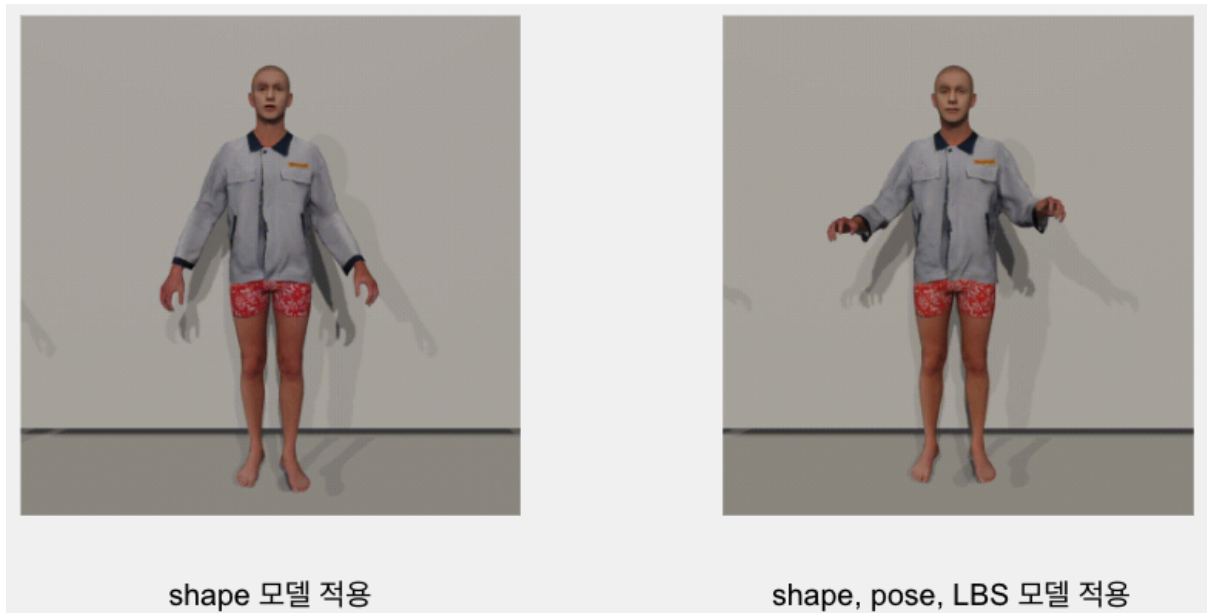
첫 번째 모델인 Shape model에서 shape와 displacement를 이용해 A-pose로 된 cloth를 출력한다. Shape model은 shape와 displacement에 따라 옷이 어떻게 바뀌는지를 학습한다.

두 번째 pose model은 첫 번째 단계에서 출력한 옷을 포즈에 맞게 변형하였을 때 자연스러울 수 있도록 디테일을 더한다. 예를 들어 주름같이 포즈에 따라 달라지는 옷의 변형을 예측한다.

마지막 LBS 단계에서는 앞선 단계에서 출력한 A-pose의 옷을 원하는 pose에 맞게 변형한다. 이 단계에서 옷의 vertex에 가상의 뼈를 더하고, 그 뼈를 회전함으로서 변형을 진행한다. 옷을 회전시키기 위해서 각 vertex는 인접한 뼈에 대한 weight가 필요하다. 이 weight는 이미 계산되어있는 몸의 weight를 거리에 반비례하게 가중합하여 설정하였다.

4. 연구 결과

4.1. 최종 모델로 학습한 결과

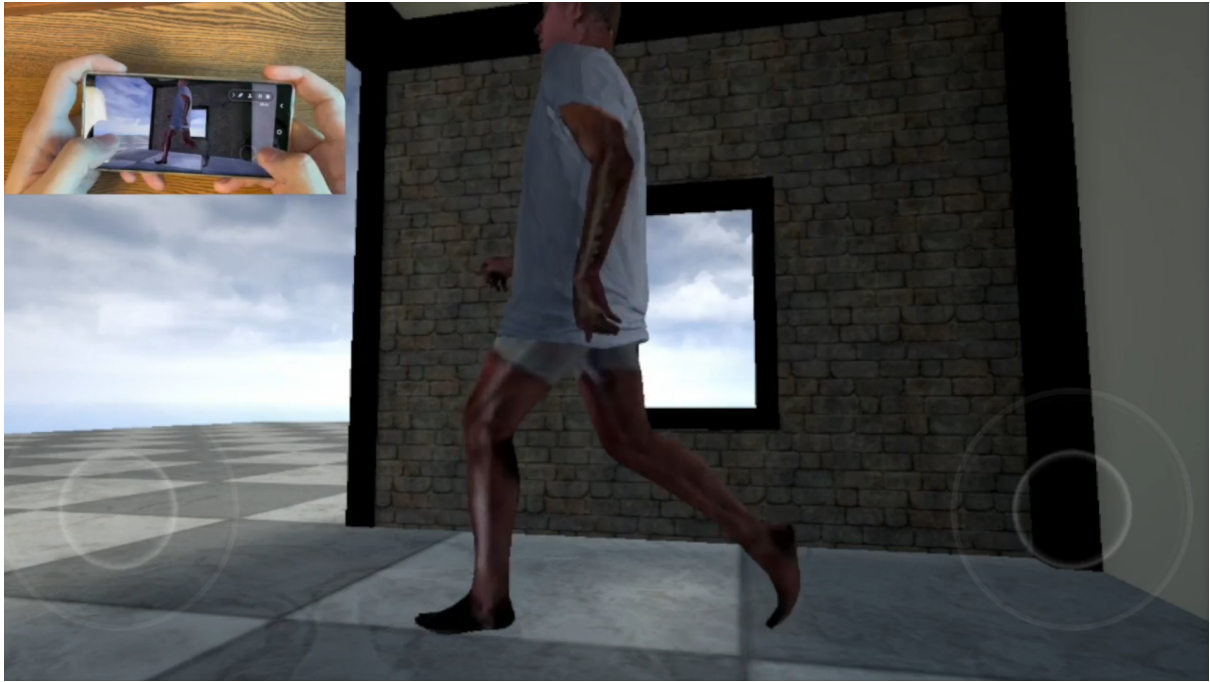


[그림 6]

<https://youtu.be/6ecoxuVkcSE?t=217>

위의 [그림 6]는 최종 모델로 학습한 결과와 초기 모델로 학습한 결과를 비교하였다. 오른쪽에 해당하는 것이 최종 모델로 학습한 결과인데, 배꼽 쪽의 옷을 보면 사람의 포즈에 따라 맞게 옷이 움직인 것을 확인할 수 있다. 사진보다는 동영상으로 사람의 움직임과 옷을 함께 보았을 때에 더 명확한 차이가 보인다. 링크를 통해 동영상으로 옷의 움직임을 더 자세히 관찰할 수 있다.

4.2. 안드로이드에서 구동해본 결과



<https://youtu.be/kBsLl9QZzSA>

안드로이드는 시간 부족으로 개선된 모델이 아닌 중간 모델로 진행하였다.

해당 앱에서는 화면 아래의 양 옆에 있는 조이스틱을 통해 카메라의 위치와 방향을 바꿔가며 실시간으로 렌더링 되는 옷을 관찰할 수 있다.

데모에서는 정해진 애니메이션을 하는 것 처럼 보이지만, 실제로는 매 프레임마다 옷의 vertex를 계산하고 있다.

4.3. 성능 평가

본 연구진은 개발한 모델의 성능을 평가하기 위해 다음과 같은 실험을 하였다.

4.3.1. 배터리 발열 평가

먼저 핸드폰을 100%로 충전을 한다. 충전 때문에 온도가 올라갈 수 있으므로 충전이 완료된 후에 30분 정도 두었다. 그 후에 해당 앱을 1시간 동안 실행하였다.

1시간 후에 배터리는 83%가 되고 모바일 폰의 온도는 30도에서 34도로 단 4도만 높아졌다.

4.3.2. 렌더링 속도 평가

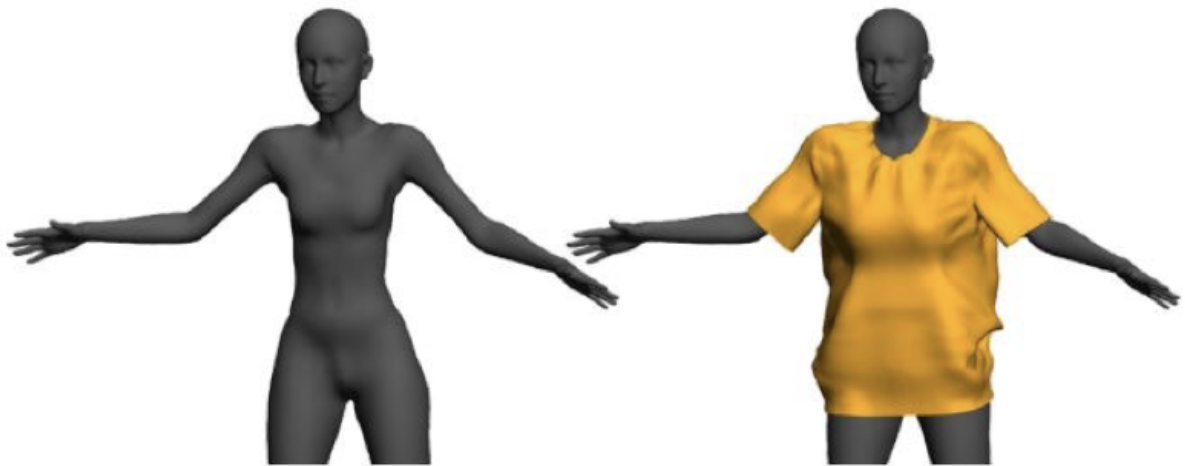
해당 애플리케이션에서 한 프레임의 옷을 렌더링하는데에 필요한 시간을 측정하였다.

한 프레임에 대략 60 milliseconds 정도 소요되었다. 이는 A.I. 훈련에 사용된 ground truth 옷을 만들 때 사용한 Intel i9-12900 CPU와 비교하면 무려 33배나 빠른 속도이다.

따라서 옷의 렌더링 속도 측면에서 물리 기반 시뮬레이션을 사용한 Desktop 용 CPU보다 월등히 빠른 속도를 보여주었다.

5. 논의

5.1. 한계점



[그림 7]

개선된 모델은 몸에 딱 맞는 옷은 잘 표현하지만, [그림 7]과 같이 펄럭이는 옷들은 표현을 잘하지 못한다. 그 이유는 몸에 딱 맞는 옷은 그전에 어떤 모션을 취했는지가 옷의 표현에 큰 영향을 주지 않지만, 펄럭이는 옷은 그전에 어떤 모션을 취했는지에 따라 옷의 펄럭이는 정도가 달라지기 때문이다. 현재 개선된 모델에서는 그전에 어떤 모션을 취했는지에 대한 정보를 고려하지 않아 펄럭이는 옷을 잘 표현하지 못한다.

5.2. 개선방안

그전에 어떠한 모션을 취했는지에 대한 context를 모델의 input data로 추가하면 해결할 수 있을 것으로 보인다. 또한 연속적인 데이터에 강점이 있는 RNN, LSTM과 같은 기법을 사용하면 모델을 더 폭넓은 범위의 옷에 적용하는 것을 기대해볼 수 있을 것이다.