Real Time Surface Modeling Using Human Pose Estimation

Anja Kempf[1], Jiaqi Wang[1], Qiao Qiao[1], and Yang Liu[1]

[1]Technical University of Munich

Introduction

This project is aimed to develop a pipeline that uses Kinect sensor to first estimatethe body pose as input, and uses the body pose to control the movement of an virtual avatar.

Gamedesign

The game consists of two main parts, first the player stands in front of the Kinect camera, a pose will be estimated and we use the gener ted skeleton (from pose) to calculatethe rough size of the player's body, and change the shape parameters on the virtualavatar accordingly.The second part is the dancing game, we use 5 pre-defined poses as the targets, and they will appear in the game sequentially with certain time intervals in between. user can also record 5 new target poses in another menu.The player have to perform the corresponding pose. We then compare the performedpose and target pose by calculating the angles of some chosen joints.

Kinect

We used the KinectSDK Package (`https://www.assetstore.unity3d.com/en/?stay/content/115950#!/content/7747`) for Unity to handle the input. From the depthmap Kinect returns a skeleton, that can be displayed in Unity too.

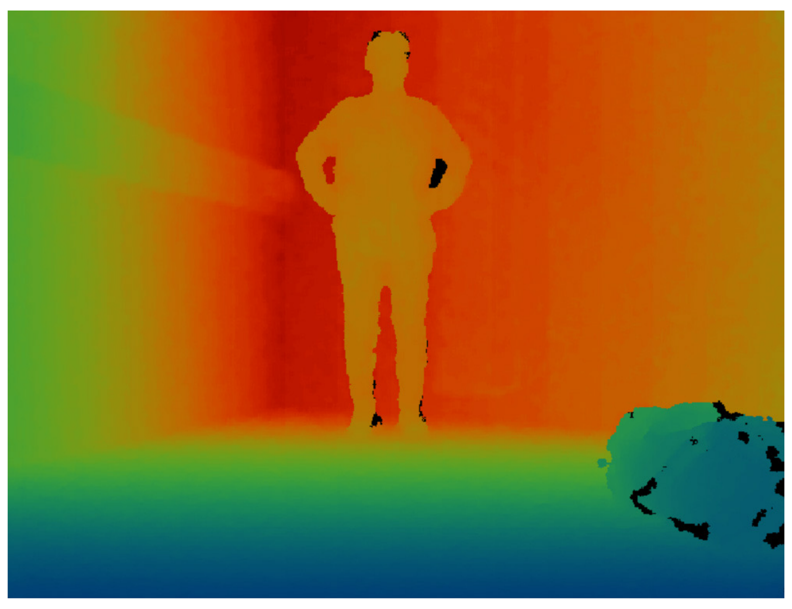

Fig. 1: Depthmap.

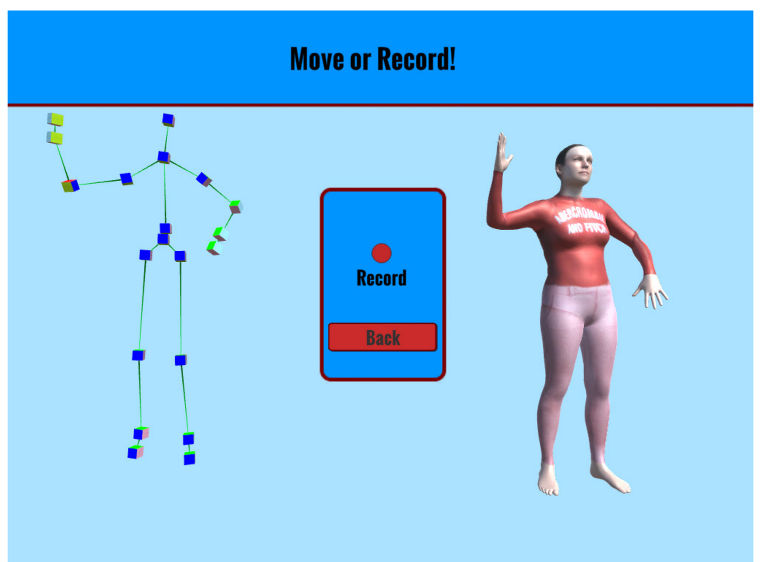

Fig. 2: DisplayScene.

SMPL

We used the smpl model for Unity (`https://smpl.is.tue.mpg.de/`). SMPL, Skinned Multi-Person Linear Model, is a realistic 3D model of the human body that is based on skinning and blend shapes

In figure 3 you can see the female smpl model and the result of applied shape blendshapes.

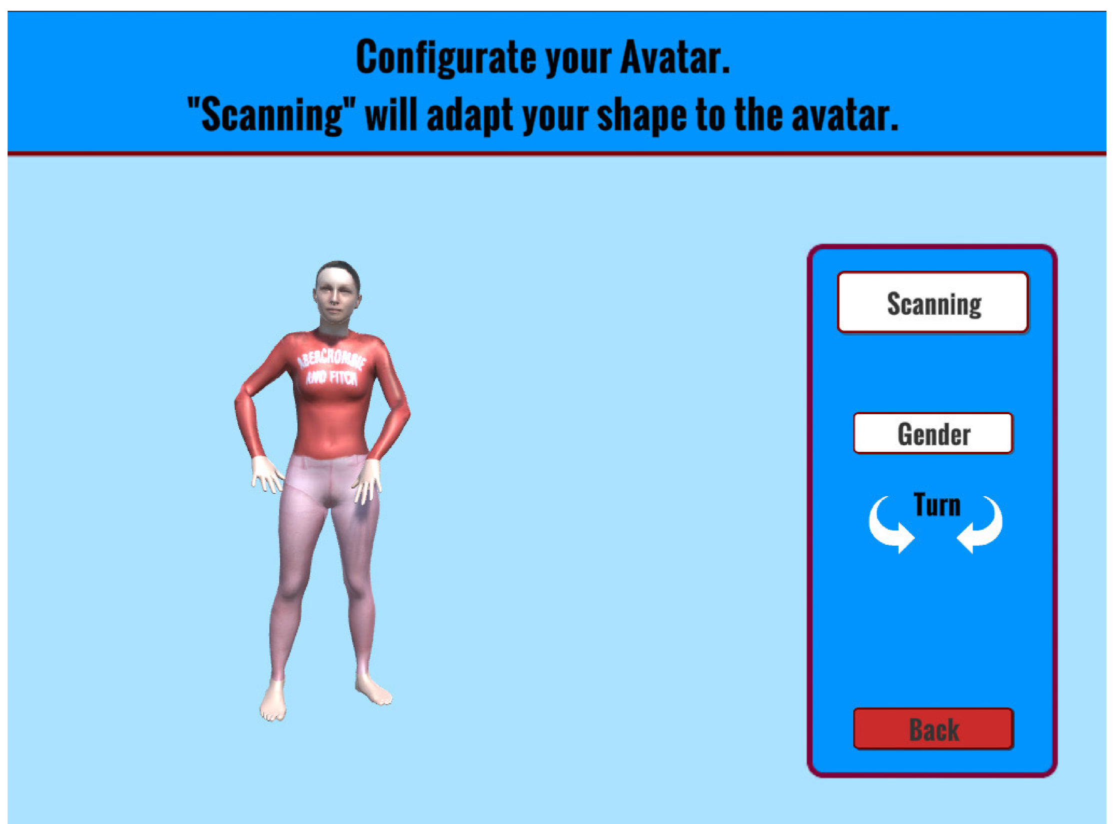

Fig. 3: Shape Changing.

Filtering

Double Exponential Smoothing is another formulation of Exponential Smoothing, which is more reliable when applying it to data that shows a trend.

$$S_t = \alpha p_t + (1-\alpha)(S_{t-1} + bt - 1) \quad 0 \le \alpha \le 1$$

$$b_t = \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1} \quad 0 \le \gamma \le 1$$

In our case, since joint positions in a time series shows the trend of the movement, this smoothing works better.

Pipeline

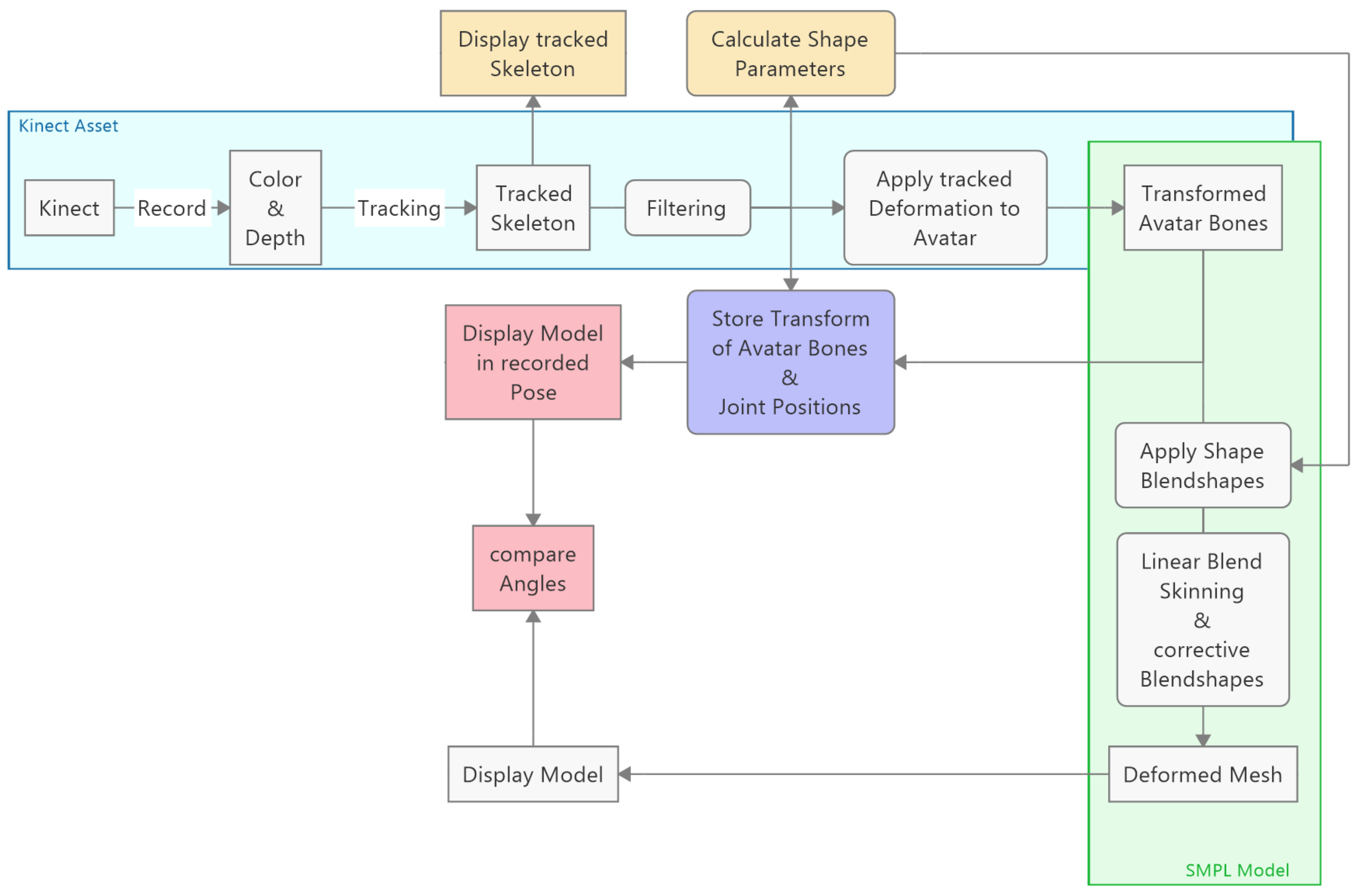

Fig. 4: Method overview.

Avatar Control

With the KinectSDK Package in Unity we can easily transfer the movement of the human skeleton to the avatar of the smpl model, which is real-time. We can also first store the data locally and then load it into the scene to controll our avatar, for example, to use predefined pose to controll the targetmodel in the dance-menu.

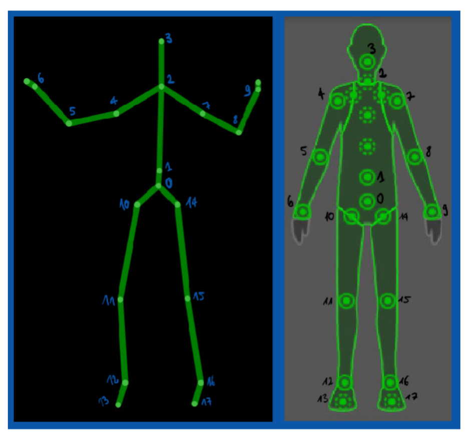

Fig. 5: Skeleton Mapping.

Pose Compare

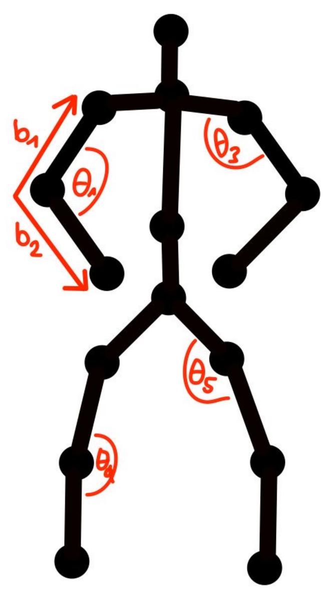

Fig. 6: Angles.

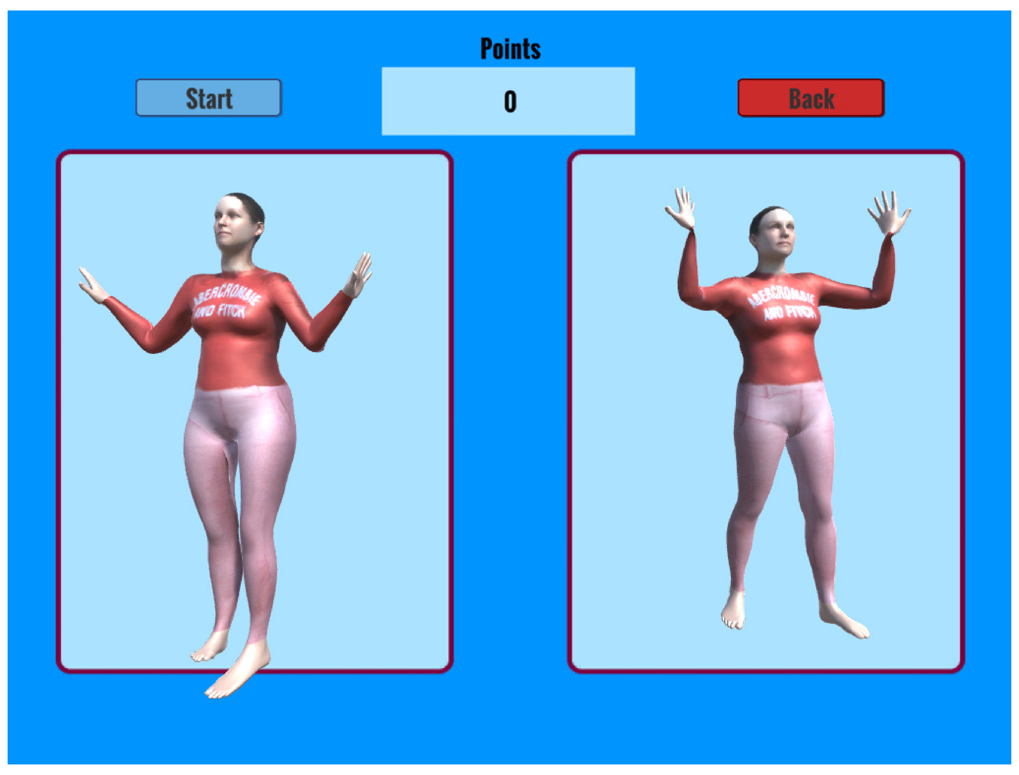

Fig. 7: DanceScene.

For comparing two poses we calculate the angles between 8 joints of the Kinect data. We are using $\cos\theta = \frac{b1 \cdot b2}{|b1| \cdot |b2|}$. To match the given pose the player has to get under a certain threshold for every angles.

Shape Evaluation

In order to reshape the model we calculate body height and width. The body height is estimated by the torso length plus the average of the length of both legs. 15 centimeters offset, nearly half of the head length is additionally added. The body width is approximated by the euclidean distance bewteen two elbows. See figure 8. These values are then mapped to the $[-5, 5]$ range of the blendshape parameters, for the height we assume a range of $[1.2, 2.2]$ m and for the width $[0, 1]$ m. We are just using the first two for overall height and width of the model.

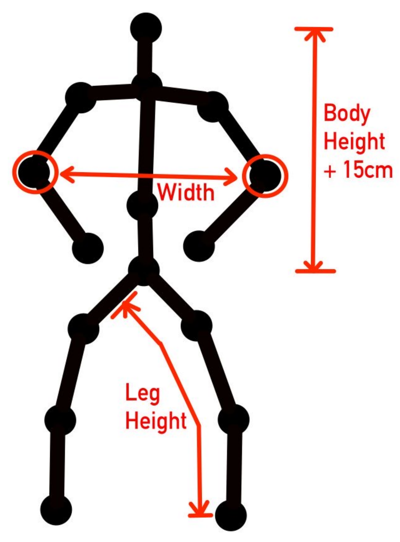

Fig. 8: Shape parameters.