Our Goal:

Suppose a person is performing an action and another person tries to mimic the same action (probably when kids try to mimic dance sequences of their favorite stars). We all know that these two actions won't be exactly the same, hence the goal of our problem is to analyze the closeness of one's actions to other and then to transform mimicked action to original one.

Dataset:

Consider a scenario where person B tries to mimic person A's action. Then in order to make a new transformed video of person B which is more close to what A has performed we would require:

- Video of person A and B captured by Microsoft Kinect Device.

Why Kinect?

We will require skeleton data of both the videos and the state of the art technique used nowadays is using Kinect libraries.

Algorithm for Video Transformation

1. For each frame (pose) of person A, find closest frame (pose) of person B.
2. Transform the pose of B to match person A's current pose.
3. Repeat steps 1 and 2 for each frame of A and make new video of B.

Finding closest pose:

- First step in this approach is to segment out the person so that we have to only deal with human only and no backgrounds.

- Kinect gives us that Segmentation data in its depth image so first we have to align our original RGB image to its corresponding depth image and then segment out the person.

- For each such frame of video that we have recorded we have access to skeletal data of person in that frame as:

- Here in above we have shown our original frame and segmented out frame with skeleton overlaid on it. [Note that the above image is the RGB image aligned with the depth image].

- With Kinect v1 we get 20 skeleton joints and hence 19 bones. For finding closest pose of person B we have represented each of our poses with the help of a vector that contains the **angles of 19 bones (from 0 to 360 degree)**. We have taken positive X direction as our 0 degree.

- So now our pose is represented as a 19 length vector and hence we search for its nearest neighbor in other video. Note that this algorithm will find closest pose and will not differentiate where the person is standing. Following is the closest pose of B with respect to A's pose.

A's pose

Matching pose of B

Transforming A's pose to B:

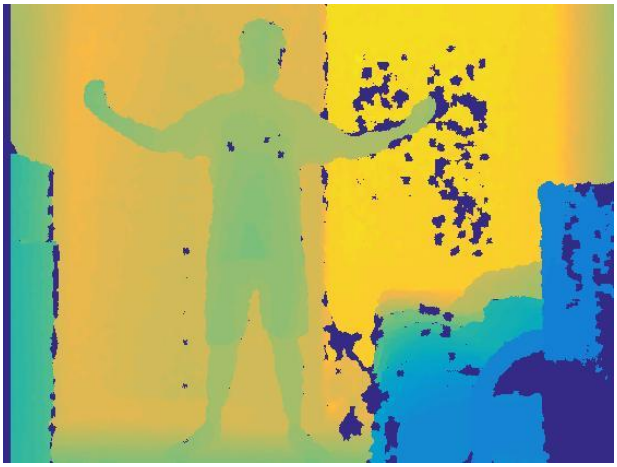
- Now since the obtained poses are similar but not exact we have to make them similar.
- We do this by making the Hip centers of B equal to A and then making skeleton of B on that place as shown here:

- Now here the transformed skeleton preserves the original bone lengths of B and bones have angles that are given to us by A's pose. So in a way we have B's new pose skeleton and now we need to shift the body to a new pose.
- In above figure each bone is made up of 2 points so we know the initial and final position of these 2 points. But in order to find an affine transformation Matrix that would lead to this transformation we need more points. So we took points 2 points near both joint points in a perpendicular direction to the bone. So we have a total of 6 points to calculate our matrix.
- From above step we obtain transformation matrix required for each bone. Now we apply same transformation matrix to ** **"THE BODY PARTS THAT ARE COVERED BY THAT BONE". **** To obtain our transformed image.

Unsolved Problems:

- As highlighted above I can't find a way to associate body parts to a particular bone. As of now I have taken rectangles of fixed size around every bone and have considered each and every pixel inside it to belong to that particular bone. This will work for some poses but not for all.

- We need to think of a method to associate body parts to a particular bone. And we will have to use depth information at each point. Following is one color coded depth image along with its RGB image and we have skeletal data to begin with.



- We need to somehow segment on depth to associate body parts to bones given skeletal data.
- Below I have shown the results that I get after drawing a rectangle around each bone and associating all the inside points of the rectangle with that bone.

Result:

1.

A’s pose

Matching pose of B

A’s pose

B aligned with A [Final Output]

2.

A’s pose

Matching pose of B

A’s pose

B aligned with A [Final Output]