

Project 3 Deliverable 1

John Ripple
 CSCI
 Colorado School of Mines
 Golden, United States
 jripple@mines.edu

Abstract—This document discusses cleaning human skeleton data to create histograms with RAD and HJPD.

I. INTRODUCTION

A 3D camera was used to capture Human Skeleton data which included 20 joint positions shown in figure 1. To interpret what movement a person is making, the raw skeleton data was featurized. The featurization was done by turning each joint's data into histograms through the RAD and HJPD algorithms.

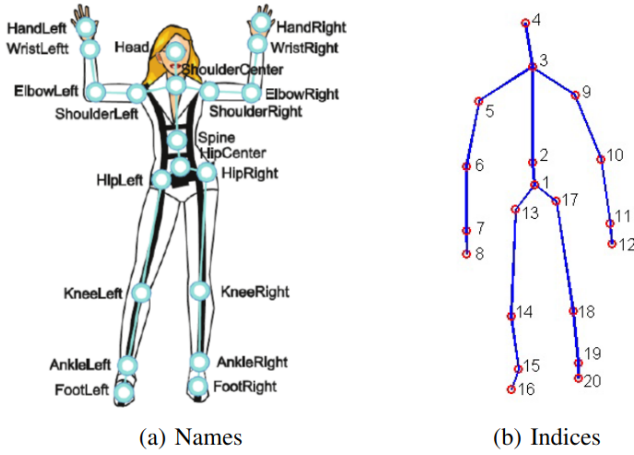


Fig. 1: Skeleton joint names and indices from Kinect SDK.

II. RAD ALGORITHM

The RAD Algorithm works by calculating the 3D distance from the hip center joint to each of the extremity joints. RAD also calculates the 3D angles between the vectors created from the hip center joint to each extremity. Both features are shown in figure 2. For every frame, both distance and angle are calculated and stored. The distance values for a joint for the entire file are aggregated and turned into a histogram. The angle values for a joint calculated from a file are also aggregated and turned into a histogram.

III. HISTOGRAM OF JOINT POSITION DISTANCES (HJPD)

The second featurization algorithm used was HJPD. HJPD works by finding the vector between the hip center joint and every other joint. This is done by subtracting the (x,y,z) coordinate of the hip center joint and the (x,y,z) of all other

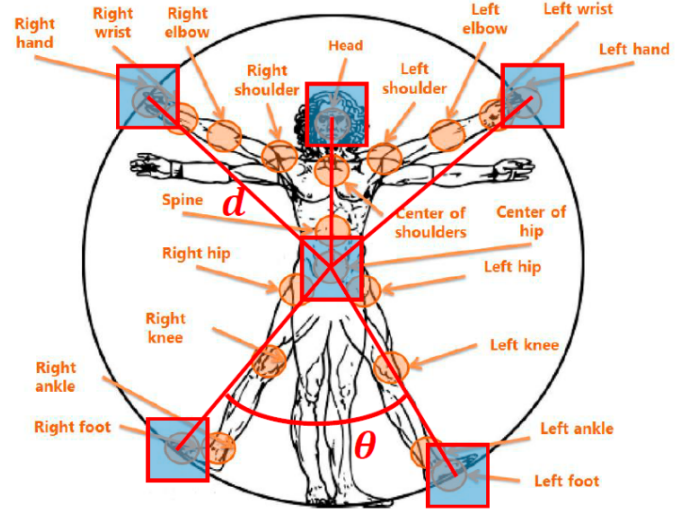


Fig. 2: Illustration of human representation based on relative distance and angles of star skeleton.

joints. Once the vectors are found, a histogram for the X position, Y position, and Z position for each vector is created.

$$(\Delta x, \Delta y, \Delta z) = (x, y, z) - (x_c, y_c, z_c) \quad (1)$$

IV. HISTOGRAM CREATION

For either algorithm used, feature data is generated and turned into a histogram. To turn 2D list feature data (table I) of joint data per frame into histograms, first, the data was transposed (table II). Transposing the data takes the data from every joint per frame into all data for a joint in a single list/row.

TABLE I:

Featurization data for joints per frame

Frame 1:	Joint 1 Data	Joint 2 Data	...
Frame 2:	Joint 1 Data	Joint 2 Data	...
Frame 3:	Joint 1 Data	Joint 2 Data	...
...			

Each row, representing a single joint's data, was then fed into `numpy.histogram()` with 20 bins. Every histogram has 20 bins, but the range for the histogram changes based on the data. The lower limit of the histogram is the minimum value found

TABLE II: Transposed featurization data

Frame 1:	Frame 2:	Frame 3:	...
Joint 1 Data	Joint 1 Data	Joint 1 Data	
Joint 2 Data	Joint 2 Data	Joint 2 Data	
...	

in the feature data (the row used in `numpy.histogram()`) while the upper limit is the maximum value found in the feature data. This allows every histogram to have the same number of bins while still allowing each histogram to not have large ranges with little to no data.