Human Action Recognition using Spatio-Temporal Features from Kinect Petar Nikolov

Petar Nikolov,
PhD student from Technical University of Sofia

Abstract: This paper presents a person - motion analysis method by using the information provided by Microsoft’s Kinect sensor. Described are the Kinect’s raw data type, movement vector calculation and the data analysis used to compare the current data to the one determined beforehand for each defined action or the one dynamically calculated.

Keywords: Motion, analysis, Kinect

1. Introduction

Motion analysis is a topic related with computer vision – a subject of a deep research work in the recent years. The applications of such systems, capable to distinguish the human activities, are endless. Systems for security and video surveillance, personal assistants or other systems for human-machine interactions.

The human body consists of a set of non folding parts, connected with joints and a human motion could be represented like a continuous change of their positions [1]. The proposed method is using the joint positions to represent the poses of a human body. Microsoft Kinect sensor is providing a color image along with the labeled data of each joint of up to 6 persons. The second version of Kinect is providing the information of the positions of a 25 points of the human body (Fig.1).

The described algorithm aims to determine a position-invariant human activity in terms of a distance between the body and the sensor.

2. Proposed method description

Using of a color images for joint model description, which is known as “skeleton” requires a high computational power and is a prerequisite of a presence of a high amount of mistakes. Using the Kinect sensor, we acquire the body stance and joint positions with high accuracy. The proposed method offers a simplified representation of a body posture, based on the joint location in the depth map provided by the sensor.

The workflow of the algorithm could be separated in to two main stages – recording a sequence of motion vectors and comparing the resultant motion vector to classify human motion.

Fig. 1: 25 joints of a human body, provided by Kinect sensor

2.1 Tracking

In each frame, each point of interest is defined as:

\[ \vec{F} = [x, y, \text{distance}, \text{angle}] \]

where:

\( \vec{F} \) – intrinsic vector of a point of interest;

\( x \) – x coordinate of a point of interest;

\( y \) – y coordinate of a point of interest;

\( \text{distance} \) – distance between the points of interests

\( \text{angle} \) – angle between the locations of the points of interest in the current and the previous frame.

After the beginning of the recording state, for each frames the position of all points of interest is recorded. The positions are recorded right after the start of the algorithm. For method to
work correctly, before beginning the recording, we have to initialize the algorithm with N frames received from the sensor. This is necessary in order to determine the exact center of the coordinate system which would be used in motion vectors calculation. On determining the vector direction, the center of the coordinate system is the point of interest N frames before the beginning of recording. This leads to displacement of the center in each arrived frame. The maximum angle error is 15° - when are defined M = 24 displacements zone.

Once we have the information about the position of the points of interest, each of them is processed with a linear regression. This step is required due to the noise introduced in our measurements and in the data we receive from the sensor. Linear regression is using the following calculations: (Fig. 2):

\[ a = \bar{y} - b\bar{x} \]
\[ b = \frac{\Sigma xy - n\bar{xy}}{\Sigma x^2 - n\bar{x}^2} \]

where:
\[ \bar{x} = \frac{\Sigma x}{n} \] - mean value of coordinate x
\[ \bar{y} = \frac{\Sigma y}{n} \] - mean value of coordinate y
\[ \bar{xy} = \frac{\Sigma (xy)}{n} \] - mean value of the product of the two coordinates
\[ n \] - number of points used for regression calculation

In formula (1) parameter \( a \) represent the offset of the regression line to the x axys. If \( x=0 \), then \( y=a \).

Parameter \( b \) represents the slope of that line. Using regression line calculation we can calculate the displacement error for each point of interest compared to the previous frames. In longer records this realization would lead to a delay in the operation of the algorithm, because with each frame the points which considered to determine the straight line increase. To determine the recalculated location of the point of interest is used:

\[ \tilde{y} = a + bx \]

where \( \tilde{y} \) is recalculated value of \( y \) and \( x \) e x-value of the original coordinates of the point of interest.

For initial initialization of the parameters are chosen zero values. Then the vector is represented in a coordinate system, divided into M equal-sized segments to determine which direction is movement (Figure 3). These sectors divide the space of M possible directions of motion. If the position of the point of interest is located in a sector other than previous to that point of interest, it creates resultant vector.

Due to the sensor imperfections, the obtained data do not always correspond to the real data. If the movement takes place near the border between two sectors, even minor deviation could lead to displacement of the point from one sector to another. This would imply that new motion vector is created. In order to avoid the creation of new motion vectors, vector differences are compared with predefined thresholds.

If the difference is less than the threshold, then the resultant vector is combined with the previous motion vector according to Triangle law of vector addition to form a completely new motion vector in place of the old one.

For the creation of the new motion vector is used the direction determined in the previous step and the length, which is equal to the distance between the two points.

2.2 Recognition
This mode is similar to the preceding, in manner in which calculates vectors and describes movement. Obtaining new frame, the positions of all points of interests are calculated. Each of them is processed individually. The direction of the resultant vector is also calculated and a decision is taken whether it is necessary to create a new vector or the resultant vector should be merged with the one from the previous frame.

For the purpose of comparing motions, the vector obtained in the processing of a new frame is compared by its length and direction with all the vectors stored during the recording of movement.

If the initial point A is known with coordinates $A_x$ and $A_y$ and the final point B with coordinates $B_x$ and $B_y$ is also known, then the distance $r$ between them could be calculated using:

$$e = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$$

For the angle calculation, the possible variations are:

- $\Delta x = \Delta y = 0$, $\alpha = 0^\circ$
- $\Delta x = 0$ и $\Delta y > 0$, $\alpha = 270^\circ$
- $\Delta x = 0$ и $\Delta y < 0$, $\alpha = 90^\circ$
- $\Delta x > 0$ и $\Delta y = 0$, $\alpha = 180^\circ$
- $\Delta x < 0$ и $\Delta y = 0$, $\alpha = 0^\circ$
- $\Delta x > 0$ и $\Delta y < 0$, $\alpha = 90 + \tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) \times 180/\pi$
- $\Delta x < 0$ и $\Delta y > 0$, $\alpha = 270 + \tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) \times 180/\pi$
- $\Delta x < 0$ и $\Delta y < 0$, $\alpha = \tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) \times 180/\pi$
- $\Delta x > 0$ и $\Delta y > 0$, $\alpha = 180 + \tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) \times 180/\pi$

Where $\Delta x$ is the first difference in (4) and $\Delta y$ is the second one, $\alpha$ is the angle of interest.

If the calculated vector in the current frame has a direction match but not a length match, the vector is not considered for further calculations. The same consideration is valid also if there is a length match but not a direction one. The only option for two vectors to be considered equal is that both the direction and the distance are the same. This is valid under the condition that distance and directions matchings are done compared to a specified thresholds.

The decision about the vector recognition is done using :

$$|\text{saved}_i\text{angle} - \text{curr}_j\text{angle}| \lt \text{angleThreshold} \quad \cap \quad |\text{saved}_i\text{distance} - \text{curr}_j\text{distance}| \lt \text{angleThreshold}$$

Where:

- $j$ – index representing the point of interest;
- $i$ – index of the recorded resultant vector;
- angle – the angle of the correspondent vector;
- distance – the length of the correspondent vector;
- $\text{curr}_j$ – resultant vector between the current and the previous frame for the j-th point of interest;
- $\text{saved}_i$ – the i-th recorded vector for the j-th point of interest;

To be recognized, the vector must meet both conditions.

In order for the algorithm to be invariant of the position of the body in front of the sensor, in the above formula should be inserted a distance correction. During recording of the motion, a average distance of each of the points of interest to the sensor is stored. Kinect automatically provides distance information to the relevant points. After recording the average values are preserved. In order to be correctly used this information we receive from the sensor, we must comply with its specifications - working range of the body must be at least 1.37 meters and no further than 6m. In detection mode for each point of interest is measured coefficient, which is the ratio of the distance to the sensor in the current frame to the average distance to the sensor for the same point of interest during recording of motion.

### 3. Results of the experiment

The final tests were carried out on selected six movements - rebound, kick boxing, clapping, squatting, lifting. Filming is carried out by a single stationary sensor Microsoft Kinect v2. The movements are performed at a distance of 2.5 meters from the Kinect. The RGB and depth images are captured at a speed of 30 frames per second (30fps). The resolution of color sensor is 1920x1080 and the resolution of the IR depth sensor is 524x484. The six movements are performed by four people, 5 times each - a total of 120 samples. Examples of selected activities are shown in Fig. 4.
The accuracy of identification of the different movements varies in thresholds 74-80%.

Two of the key parameters in the algorithm are based on minimum error that caused in various activities.

- angleDiff – threshold for joining consecutive vectors;
- numberOfAngleDetectionZones – number of zones defining the change in the direction of movement;

The first parameter – threshold for resultant vector initialization in adjacent frames. The threshold should always be less than 360/ number of zones for motion detection. The provided results are achieved using the Kinect sensor at a distance of 2.5m.

Table 1: Maximum accuracy achieved in recognition of relevant activity

<table>
<thead>
<tr>
<th>Motion</th>
<th>Accuracy, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>handwave</td>
<td>74,48%</td>
</tr>
<tr>
<td>repetitive arcs with hands</td>
<td>80,44%</td>
</tr>
<tr>
<td>hand motion in a straight line</td>
<td>79,92%</td>
</tr>
</tbody>
</table>

Table 2: Results on determining the angle threshold.

<table>
<thead>
<tr>
<th>Angle diff</th>
<th>Accuracy, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>21,01%</td>
</tr>
<tr>
<td>5</td>
<td>78,25%</td>
</tr>
<tr>
<td>10</td>
<td>64,62%</td>
</tr>
<tr>
<td>15</td>
<td>59,25%</td>
</tr>
<tr>
<td>35</td>
<td>41,98%</td>
</tr>
</tbody>
</table>

Table 3: Results in the determination of “number of zones for determining the angle of the motion vector”

<table>
<thead>
<tr>
<th>Value</th>
<th>Accuracy, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>31,01%</td>
</tr>
<tr>
<td>10</td>
<td>64,25%</td>
</tr>
<tr>
<td>16</td>
<td>57,22%</td>
</tr>
<tr>
<td>24</td>
<td>76,25%</td>
</tr>
<tr>
<td>36</td>
<td>11,48%</td>
</tr>
</tbody>
</table>

Second parameter - the number of areas for motion detection - determining of how many zones will be divided coordinate system.

4. Conclusion

In this paper we proposed a method for motion recognition using a skeleton data provided by the Microsoft’s RGB-D sensor Kinect. The system is able to classify a motion performed from a person among multiple stored motion descriptions in a database. From the conducted experiments, the classification rate vary between 76% and 80%. The current system can be extended by including more skeleton points for tracking or by using additional Kinect sensors.

Acknowledgement: This paper was supported by Technical University – Sofia inner program to support PhD research projects under Contract № 162pd0025-7: “Human action estimation based on video image analysis”

References