Project Report
Motion Classification based on Mocap Data

Victor Liu, linxiol@g.clemson.edu

I. INTRODUCTION

With the huge development of computer animation, Human Motion Capture(mocap) shows remarkable performance in many fields, such as science fiction filmmaking, VR applications, sports science and robotics and so on. To label different varieties of mocap data is increasingly needed. However, manually generated annotations for raw mocap data is obviously time consuming and tedious. So, this project introduced a way to predict raw mocap data automatically based on Machine Learning methods.

Dataset for this project comes from CMU mocap database[1] including thousands of various motions. One category motion inside the dataset consists of one Acclaim Skeleton File(ASF) and several Acclaim Motion Capture(AMC) data. In this project, I focused on classification of walk motion against the other motions using Random Forest classifier.

II. RELATED WORK

Kadu, H. et al.[2] propose a way by translating a segmented frames(human poses) of data from a motion into a codeword recorded in a codebook, therefore, one entire motion can represent as a single string and the string sequences be stored in a tree based on the tree-structured vector quantization(TSVQ). Then they reached the result by a fusion method based on SVM. They approach to a high accuracy, although, they restricted human motion into a finite number of cases, which may not be flexible. And they trained their model only in a limited number of mocap data which reduce the reliability of their model.

III. METHOD AND IMPLEMENTATION

A. Data Acquisition

As the CMU mocap database website does not support batch download, I constructed a web crawler to download all the files including .asf, .amc and index file through the source page(see fig 1). The principle of the web crawler is content matching based on python3 standard module re. Matching pattern see table I.

TABLE I
MATCHING PATTERN

match goal	match pattern
subject name	r'Subject (.*?)'
asf url	r'¡A HREF="(.*?)"¿asf¡/A¿'
index file url	r'#(\d+) (.*?) file index'

CMU mocap website is simple that make it easy to applied the match pattern to find the information needed, though, I still need to consider a lot of details. For example, the number of frames for each motion is different(120 or 60) which is a very important detail. I have to modify the match rule over and over again to dig all the information and files I need.

```
        </TR>
        <TR>
          <TD>Image</TD>
          <TD>Trial #</TD>
          <TD>Motion Description</TD>
          <TD></TD>
        </TR><TR BGCOLOR=#FF0000><TD></TD><TD>1</TD><TD>playgro
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_01.c3d">c3d</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_01.mpg">mpg</A></
HREF="badtrial.php?subjectnumber=1&motion=1">Feedback</A></TD></T
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_02.tvd">tvd</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_02.amc">amc</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_02.avi">Animated<
</TD><TD>3</TD><TD>playground - climb, hang, swing</TD><TD><A HRE
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_03.c3d">c3d</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_03.mpg">mpg</A></
HREF="badtrial.php?subjectnumber=1&motion=3">Feedback</A></TD></T
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_04.tvd">tvd</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_04.amc">amc</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_04.avi">Animated<
</TD><TD>5</TD><TD>playground - climb, go under</TD><TD><A HREF="
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_05.c3d">c3d</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_05.mpg">mpg</A></
HREF="badtrial.php?subjectnumber=1&motion=5">Feedback</A></TD></T
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_06.tvd">tvd</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_06.amc">amc</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_06.avi">Animated<
</TD><TD>7</TD><TD>playground - climb, sit, dangle legs, jump dow
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_07.c3d">c3d</A></
HREF="http://mocap.cs.cmu.edu:80/subjects/01/01_07.mpg">mpg</A></
```

Fig. 1. CMU Mocap Website Source Page

B. Features Selection

This project concentrated on motion capture data in .amc file which record the translation and main joints' rotation of humans. The original whole features are root, lowerback, upperback, thorax, lowerneck, upperneck, head, rclavicle, rhumerus, rradius, rwrist, rhand, rfingers, rthumb, lclavicle, lhumerus, lradius, lwrist, lhand, lfingers, lthumb, rfemur, rtibia, rfoot,

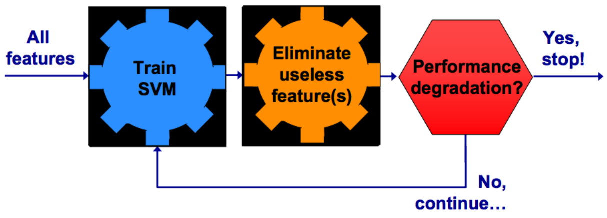

Fig. 2. Embedded Methods

TABLE II
RESULTS FROM EMBEDDED METHODS

joints	root	lowerback	upperback	thorax	...	ltoes
keep(0)	0	1	1	1	...	0

rtoes, lfemur, ltibia, lfoot, ltoes. I tried to select the most useful features by embedded methods [3](see fig 2). During the procedure of the method, for each feature, this method drop the feature to find the change in result performance. If the result got worse than keep. I applied embedded methods five times, however, each time the accuracy remained generally the same. I have to keep them all as you can find in table II.

Although, I selected some features based on my common sense:root, thorax, head, rhumerus, rradius, lhumerus, lradius, rfemur, rtibia, lfemur, ltibia. I also made a comparison with the keeping the whole features. As fig 3 shown, when the number of dataset is small, my selection of features result in a higher accuracy. For this reason, during the following, I used the features I selected.

Fig. 3. My Selection of Features vs. All Features

C. Data Processing

For the spatial domain I generally concatenated all the values within one frame. As for the temporal relationship, I tried three different ways to process raw data: 1) Evenly pick the key frame during the whole time sequence. 2) Divide the whole time sequence into segments equally and pick the first key frame in each segments. 3) Calculate the change with the previous key frame as motion vector.

The results shows in fig 4. I prefer the second method as it seems a little outperform to the other methods.

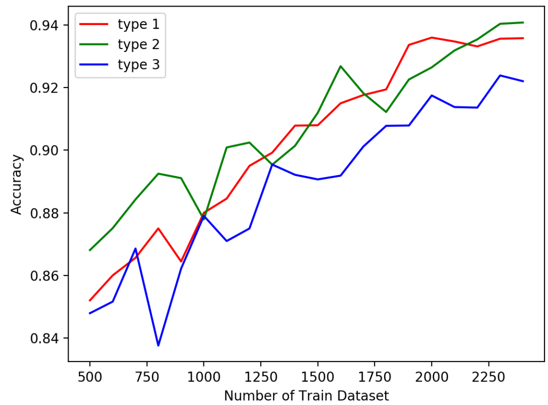

Fig. 4. Comparison of Three Type Data Processing

D. Parameter Adjustment

In this part, I mainly focused on two super-parameter adjustment: number of estimator in Random Forest classifier and number of key frame to pick.

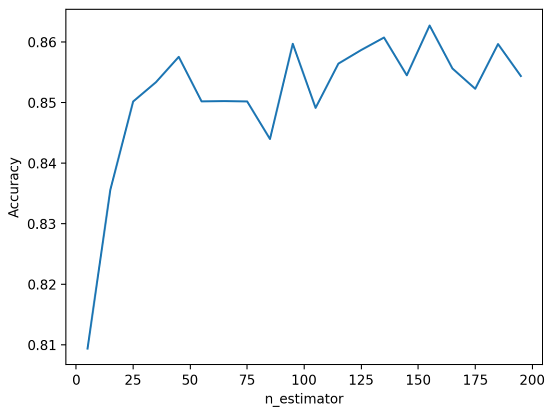

Fig. 5. n_estimator vs. Accuracy

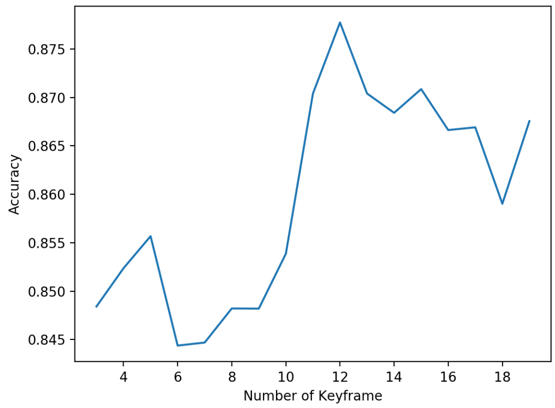

Fig. 6. Number of KeyFrame vs. Accuracy

TABLE III
FINAL RESULT

Accuracy of Random Forests	0.97			
	precision	recall	f1-score	support
0	0.98	0.96	0.97	676
1	0.96	0.98	0.97	672
avg/total	0.97	0.97	0.97	1348

The results shows in fig 5 and fig6 that the best choice for n_estimator is 50 or 100 and best key frame number is 12.

E. Data Augment

Due to the limited number of mocap data from CMU mocap database is far from satisfied by machine learning model, I extend the dataset by plus a start frame index when sampling key frames from a mocap data. So that sampling from different beginning will achieve to a different train data. Finally the positive(walk motion) dataset can reach up to about 2500 from 500.

IV. RESULTS

Result detail see table III.

V. CONCLUSION AND FUTURE WORK

After data processing, features selection and data augment, the accuracy reach up to 97% which is quite honorable. Although, this project still exists some issues. First, I labeled the dataset into walk motion and the other manually only depend on if their file name contained 'walk'. However, walk motion can be expressed as another way. In the error case of this project, for example, existed a motion named 'very highly step' which was apparently a walk motion but labeled as 'other'. Second, some motions may contained a small part of 'walk' or even though belonged to other motion but walking meanwhile. Next, walk motion but stand for a while before walking. These are some detail issues. To a higher level, how to detect a motion to be within a specific time; how to deal with motion data with various time length; how to determine different motions with a little difference. These are all the problem needed to be discussed. And I believe that with the explosive growth of number of mocap data, both machine learning and deep learning will play a significant role in computer animation. There will be one day the effect of these problem fade away.

REFERENCES

[1] http://mocap.cs.cmu.edu/search.php?subjectnumber=
[2] Kadu, H., & Kuo, C. C. J. (2014). Automatic human mocap data classification. IEEE Transactions on Multimedia, 16(8), 2191-2202.
[3] Recursive Feature Elimination (RFE) SVM. Guyon-Weston, 2000. US patent 7,117,188