Mining Mid-level Features for Action Recognition Based on Effective Skeleton Representation

A. Method

0. Ideas

This paper thinks that full skeletal description is a low-level feature, and it aims to propose a mid-level feature, which is based on the low-level feature, more discriminative. They try to apply Data Mining techniques to find frequent body part pose patterns, called FLPs(frequent local parts).

[Problems To Discuss]

1. How to develope mid-level features?
2. How to select the most frequent and relevant(discriminative, representative and non-redundant) patterns instead of the merely frequent patterns?

1. Skeletal Representation (body part-based)

1) As a matter of fact, among 20 skeleton joints captured by the Kinect, they find that there are five joints which are not reliable. Therefore, they just take the remaining 15 joints. And 15 joints forms 14 limbs.

2) Low-level feature 0.1, a unit difference vector(其实就是欧几里得空间中的单位方向向量) between two joints i and j of a limb L_{ij}, denoted as delta(i, j).

3) Low-level feature 0.2

Empirically set a threshold T, define a state of the limb L_{ij} {

```
    for e in delta(i, j):
        if abs(e) <= T:
            state = [state, 0]
        elif e > T:
            state = [state, 1]
        else:
            state = [state, -1]
```

}

As we can see that each dimension of a limb low-level feature, there are three different states. In total, a limb will have 27 states(3*3*3). And 14 limbs lead to a 27 * 14 = 378 dimension vector. What's

more, it is a boolean vector. For example, a limb's state is [0, 0, 1], so its feature vector is [0 1 0 0 0 … 0], 27 dimension. Obviously the 378 dimension boolean vector can be compressed into a 14 dimension vector, since the index of the 1's is suffcient to express its feature.
Up to now, we got a 14 dimension feature vector by mapping the low-level feature to a state for each limb. The next step is to combine the seperated limbs into 7 body parts, and assign different states according their DoF(Degree of Freedom in physics). The feature vector(FV) become 7 dimension now.
They want to add the temporal information into the FV, so they change each dimension of FV to represent the continuous C frames's state(but didn't show how they make it).
Finally, we get a 7 dimension mid-level feature vector, which contains both pose information and motion information.

4) Mid-level feature is the FLPs.

2. The proposed Algorithm
Pattern discovery in data mining is proposed to find the frequent pose states among all the train action samples. Remembering is among all the training action samples(see the Training part). Dispite of the middle process(because it needs some basic data mining concept, but it's hard, long and unnecessary to show them), assuming that we have found a set of the most frequent pose state, called FLPs. Note that each element of the FLPs is a feature vector of a frame, also called frequent pose pattern.

1) Training
Then for each training action, each frame is a transaction. In order word, an action is encoded as a series of transactions, say X. Then by counting the occurrences of FLPs(say K dimension) in X, we got a K dimension vector, the mid-level feature in legend, which is called bag-of-FLPs. Finally apply SVM to train these vectors, we got a model.

2) Action Recognition
SVM predict, input is the bag-of-FLPs of the new input action.

B. New things

1) Apply data mining methods to find the regular pattern in the global space.
2) Set a threshold to map the numerical value into discrete state, I think this is more powerful and compact because a pose's feature always swings in a small range of numerical value.

3) When selecting powerful frequent patterns rather than just frequent patterns, they add a property called representativity, which is difined the frequency among the training samples. This is a new view-point in removing noise pose, in my opinion.

C. Shortcomings

1) All parameters are choosen empirically instead of by theory. It makes people confused sometimes how to do better base on their work. And it is hard to apply the algorithm in new data set, because we have to test all the parameters one more time.
2) Mapping numerical value into discrete state will loss the original data. Therefore, it may usually go wrong when an action's key pose is a tiny pose.
3) Maybe the dataset is too challengeable, the accuracy is rather low. 78.8% on MSR-DailyActiviti dataset, while 75.56% on MSR-ActionRairs dataset.