Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group

A. Method

0. Ideas

Observation that for human actions, the relative geometry between various body parts (though not directly connected by a joint) provides a more meaningful description than their absolute locations (and this intuition was comfirmed in the lab).

[Problem to Discuss]

1. How to combine the joint location infromation with the joint angles information?
2. How to deal with the various length of different action sequence?
3. How to map Lie Group Space into linear space which suits for SVM to work?

1. Skeletal Representation (body part-based)

Thinking of e_n as a link of two joints v_{n1} and v_{n2}, e_n is treated as a body part. Calculate the rotation and translation (measured in the local coordinate system attached to e_n) required to take e_n to the position and orientation of e_m between each part. And set $P_{m,n}(t)$ a special Euclidean group SE(3), a 4 by 4 matrix, to combine these two values.

M is the number of body parts. C(t), the proposed representation at time instance t, is a set of pair-wise Pm, n(t), where 1 <= m, n <= M, and m != n.

2. The proposed Algorithm

Based on the above representation, a skeletal seqence of an action can be represented as a curve in SE(3) * SE(3) * … * SE(3), {C(t), 0 <= t <= T}. Then because SVM and Fourier analysis can't work in this curve space, the curve is mapped onto its tangent space, and we get a feature vector. The process maybe like the opeartion Feature(t) = [vec(log(Pm, n(t)), …].

1) Training

Using all training curves input the DTW(dynamic time warping, handle rate variations) algorithm to train a nominal curve for each action category. Apply FTP(Fourier temporal pyramid) for each dimension separately and concatenate all the Fourier coefficients to obtain the final feature vector. Finally, using SVM.

2) Action Recognition

Using one-vs-all SVM according the feature vector of the input skeletal seqence, choose the highest scoring class.

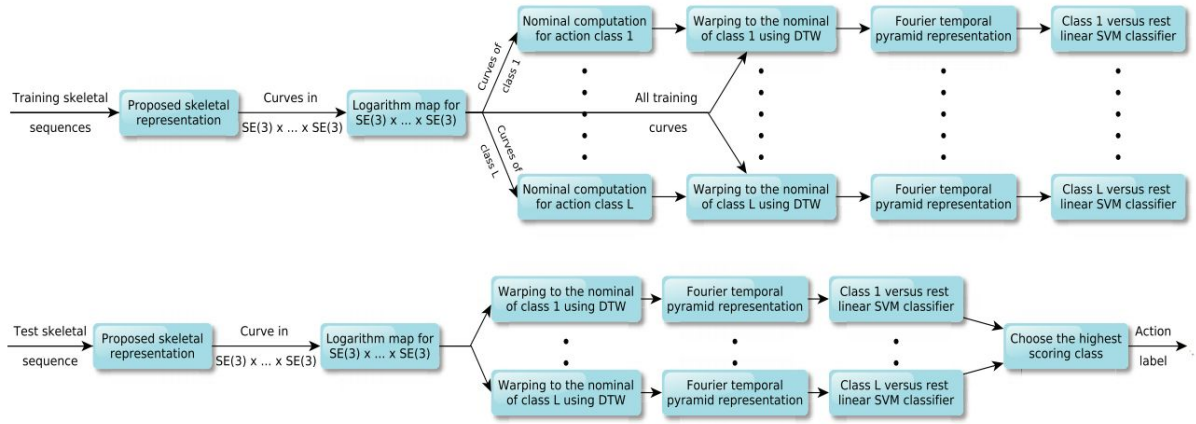

Figure 4: The top row shows all the steps involved in training and the bottom row shows all the steps involved in testing.

B. New things

1) Taking the relative geometry infomation instead the relative coordinate or absolute position. But I can't understand the power of the rotation and translation.

2) Using Lie group, which is a powerful mathmetical tool, but I don't know the theory of it now.

3) The data pre-processing is pretty good.

Placing the hip center at the origin to make the skeletal data invariant to absolute location of the human in the scene, all 3D joint coordinates were transformed from the world coordinate system to a person-centric coordinate system.

Normalization makes the skeletons scale-invariant.

Rotating the skeletons such that the ground plane projection of the vector from left hip to right hip is parallel to the global x-axis, which makes the skeletons view-invariant.

C. Shortcomings

1) Without action segmentation, it supposed a skeletal sequence is an action.

2) The algorithm is hard to understand due to its mathmetic.

Table 4: Comparison with the state-of-the-art results

MSR-Action3D dataset (protocol of [7])	
Histograms of 3D joints [20]	78.97
EigenJoints [22]	82.30
Joint angle similarities [13]	83.53
Spatial and temporal part-sets[18]	90.22
Covariance descriptors [5]	90.53
Random forests [27]	90.90
Proposed approach	**92.46**
MSR-Action3D dataset (protocol of [19])	
Actionlets [19]	88.20
Proposed approach	**89.48**
UTKinect-Action dataset	
Histograms of 3D joints [20]	90.92
Random forests [27]	87.90
Proposed approach	**97.08**
Florence3D-Action dataset	
Multi-Part Bag-of-Poses [14]	82.00
Proposed approach	**90.88**