Interactive Action Recognition using Relative Geometry Between Body Parts

Swami Sankaranarayanan and Kota Hara
Center for Automation Research, University of Maryland, College Park, MD 20742
{swamiviv,kotahara}@umiacs.umd.edu

Abstract

In this work, we propose

1. Introduction

Action recognition from the body skeleton has been very active due to the advance of motion capture system such as Kinect. Action recognition systems use the skeleton information as an input to predict the action taking place. Typically the skeleton information is given in the form of 3d locations of a set of body joints. Many action recognition systems directly use these information as the input or convert the location information into a set of joint angles and then use them as the input. The focus of the research has been on the recognition part.

[1] proposed a new skeleton representation based on relative geometry across body parts and achieve the state of the art results on several standard action recognition dataset. They compute the translation and rotation between each pair of body parts and map it to the tangent space to get a feature vector. The benefit of their approach compared to the previously adopted skeleton representation is that their method capture the relationship between body parts that are not directly connected each other. Thus, their method is more capable of representing actions in which the relationship between non-connected body parts, such as left arm and right arm, is important.

In this work, our task is to classify action performed between two people. We also use relative geometry across body parts but unlike their work, we also consider relative geometry between body parts across two people. This representation enables us to capture two people's interaction well.

2. Related Work

3. Action Localization

4. Skeleton Representation

The output from the Kinect sensor is 3d positions of 20 body joints for each person defined in the global coordinate. We define our skeleton representation for the first person as $S^{(1)} = (V^{(1)}, E^{(1)})$, where $V^{(1)} = \{v_1^{(1)}, v_2^{(1)}, \ldots, v_N^{(1)}\}, N = 20$ denotes the set of joints and $E^{(1)} = \{e_1^{(1)}, e_2^{(1)}, \ldots, e_M^{(1)}\}, M = 19$ denotes the set of body parts (See Fig.1). Let $e_{m1}^{(1)}, e_{m2}^{(1)} \in \mathcal{R}^3$ represents the starting and ending points of the body part $e_m^{(1)}$. Similarly we have $S^{(2)} = (V^{(2)}, E^{(2)})$ for the second person.

First, we define the person specific local coordinate at the first person's shoulder center computed as $(v_5 + v_6)/2$ such that the x coordinate is aligned with the person's body orientation. Then we update both $S^{(1)} = (V^{(1)}, E^{(1)})$ and $S^{(2)} = (V^{(2)}, E^{(2)})$ using this new coordinates. This process makes our representation invariant to the global translation and orientation.

Next we consider a set of body parts, $E = \{E^{(1)}, E^{(2)}\}$, and for each pair of body parts e_m and e_n, in E, we describe their relative geometry by the translation and rotation. The translation is computed as $T_{m,n} = e_{m1} - e_{n1}$. For the rotation, we compute the rotation axis r and the rotation angle θ between two vectors $e_{n2} - e_{n1}$ and $e_{m2} - e_{m1}$. From them we compute a quaternion $q_{m,n} \in \mathcal{R}^4$ by $q_1 = r_1 sin(\frac{\theta}{2}), q_2 = r_2 sin(\frac{\theta}{2}), q_3 = r_3 sin(\frac{\theta}{2}), q_4 = cin(\frac{\theta}{2})$.

Thus, each pair of parts can be represented as 7 dimensional vector. Since there are $\binom{38}{2}$ such pairs, we have $\binom{38}{2} \times 7 = 4921$ dimensional vector. We do the same step by using the second person's local coordinate and concatenate two vectors to obtain the final feature vector representing two people configurations at current frame.

We compute our skeleton feature across 41 frames which includes the action center frame, 20 frames before the action center frame and 20 frames after the action center frame. Finally we concatenate all the feature vectors into a final vector.

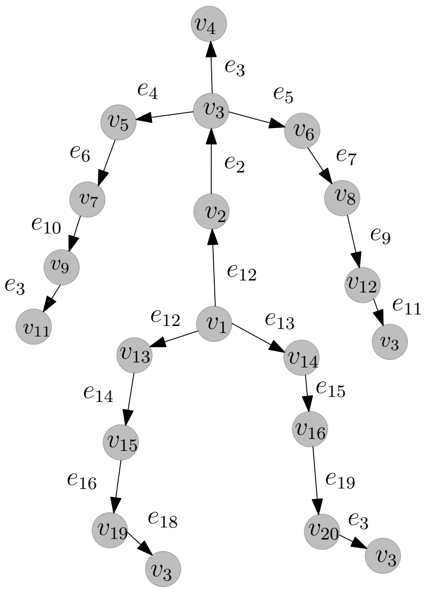


Figure 1: An illustration of the skeleton.

5. Classification Methods

We propose two different methods for classifying interactive actions from the skeleton features.

5.1. Generative Modeling

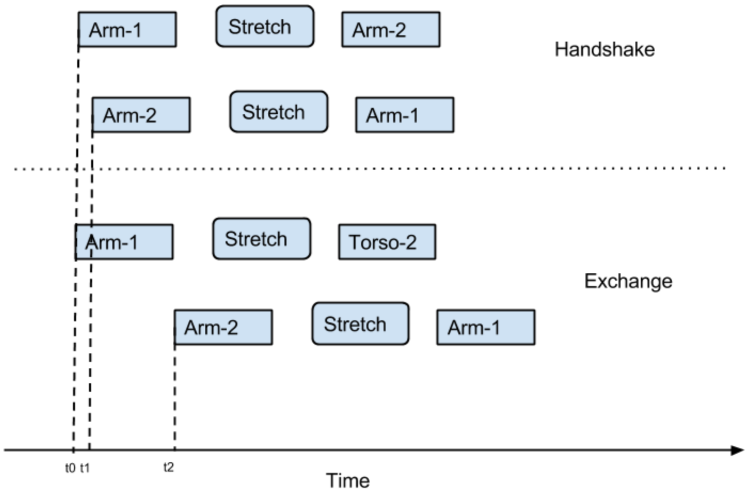
Given an input video containing the action, generate a semantic desription of the action as it occurs in the video. To describe an action would mean to generate a dictionary based representation for the action. A suitable middle ground between a complete linguistic action definition and a machine-derivable description, we borrow the action vocabulary decribed in [2]. Accordingly, a single person action in general can be described using an **operational triplet** as follows:

$$< Agent - Motion - Target > \tag{1}$$

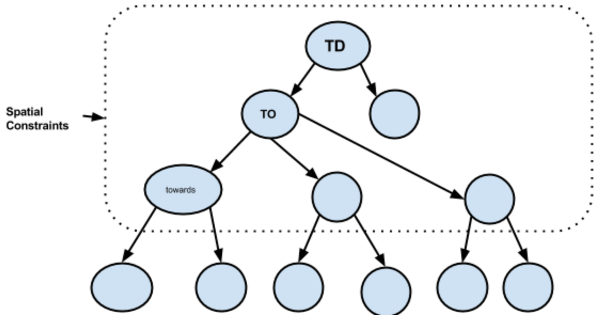
where $Agent$ is the person or the body part of the person which initiates the action; $Motion$ implies the movement of the $Agent$; $Target$ can be taken to be an object or another person or his body part that is involved in the interaction. Using this terminology some simple action representations can be generated as shown below:

- **Handshake**: <Arm1 - Outstretched - Arm2>
- **Kicking**: <Leg1 - Outstretched - Leg2>

It should be noted that for some actions that *Agent* and *Target* need not necessarily be a single person or body part but can be a collection of body parts that act on each other in a causal way. To see an example of this, consider the figure ??



Thus, we can see that actions that have similar semantic descriptions in terms of the operational triplet can be differentiated by means of proper temporal constraints. To look at how spatial constraints can decide the type of interaction, we incorporate that knowledge in the decision tree structure shown in figure ??.



5.2. Artificial Feed forward Neural Network

We use artificial feed forward neural network with softmax activation function as its final layer to estimate conditional probability, $P(y_t|x_t, t = t_{ac})$ at time instance t, where y_t is an action class label and x_t is the skeleton features computed from $t - 20$ to $t + 20$ frames and t_{ac} is the time instance of the action center frame. We manually localize the action center frame of the training sequences and train the model with 10 hidden layers.

In testing time, we apply the trained neural network in a time sliding window manner and obtain the conditional probability $P(y_t|x_t, t = t_{ac})$. We also apply our action center detector at each frame to obtain $P(t = t_{ac}|x_t)$. We then compute $P(y_t, t = t_{ac}|x_t) = P(y_t|x_t, t = t_{ac})P(t = t_{ac}|x_t)$ at each frame. Finally, we compute $P(y)$ for each sequence by $\sum_{t=t_s}^{t_e} P(y_t, t = t_{ac}|x_t)$ where t_s and t_e are the time instance where there are sufficient number of frames to apply the neural network. The final prediction is done by $\mathrm{argmax}_y\, P(y)$.

6. Experiments

6.1. Data Collection

The dataset that we have is an augmented version of the already available interaction dataset which consists of 8 actions :

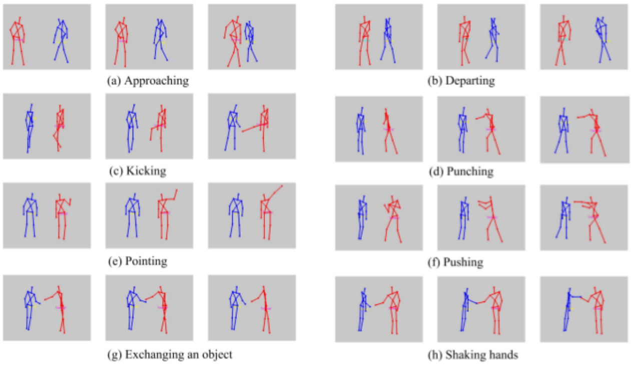


Figure 2: Available Interaction dataset

In addition to the available dataset, we create a new dataset of 10 different interactive actions using Kinect. The action classes we consider are summarized in Fig.3. For each pair of people, we collect 2 sequences per action class by switching people's locations. There are 12 sequences per action class.

The data is organized in the following form :

- : Depth data file corresponding to frame number xxx of activity.
- rgb$dummy$xxx.png: RGB data file corresponding to frame number xxx of activity.
- skeleton$dummy$pos.txt: Skeleton position data

Skeleton data consists of 15 joints per person. Each row follows the following format.

where,
$PA(i) \implies$ position of ith joint (x,y,z) for the subject A located at left
$PB(i) \implies$ position of ith joint (x,y,z) for the subject B located at right
x,y and z are normalized as [0,1].

Steps for performing Interaction Recognition using Kinect:

Step 0 : To organize the data in the above format. To do this, we extract skeleton data from the recorded OpenNI files that comes from the Kinect sensor.

Step 1 : For each frame, we have to spatially localize each person - We are trying to figure out how to extract this information directly from the Kinect output

Step 2 : Should perform an Object Detection routine to verify if there is any sign of Object exchange

Step 3 : Based on how each joint moves in the video per frame, the activity trees are grown and the corresponding activity is detected. Primarily we want to differentiate between Object/Non-object Interactions and further classify the Non-Object Interactions into one of the Action classes that were given to us during Training.

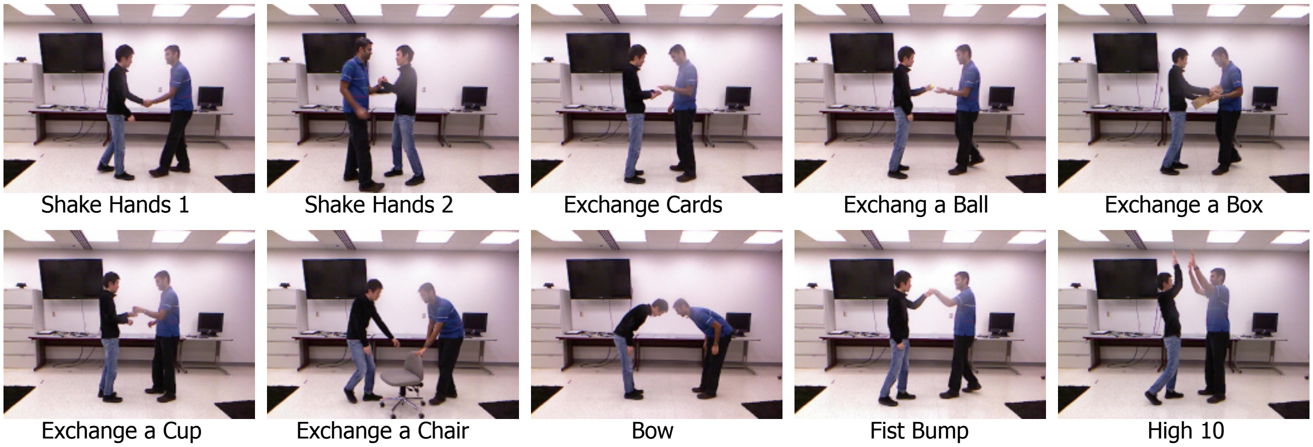
6.2. Results

7. Conclusion

References

[1] Raviteja Vemulapalli, Felipe Arrate, Rama Chellappa, *Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group*. CVPR, 2014. 1

[2] Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy, Sangho Park, J.K. Aggarwal, CVPR'04 2



Shake Hands 1

Shake Hands 2

Exchange Cards

Exchang a Ball

Exchange a Box

Exchange a Cup

Exchange a Chair

Bow

Fist Bump

High 10

Figure 3: Actions in newly collected dataset