# Interactive Action Recognition from 3D Skeleton Data

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

## Abstract

*In this work, we deal with the problem of recognizing actions that involve interaction between two people. For this purpose, we have created a dataset that involves ten different interactive actions using the Kinect Sensor. We extract discriminative features from the Skeleton data obtained from Kinect by extending the framework of [1] to two person interactions. Further, we propose a simultaneous action detection and localization method using the latent structural SVM framework, where the latent variables are the start and end frames of an action in the given input video. We demonstrate the effectiveness of our approach over simple multi-class classfication methods.*

## 1. Introduction

Human action recognition has been a major problem in computer vision community for many years. There are many applications of action recognition such as surveillance, health-care and human computer interaction. Most of the previous works focus on action recognition of a single person. The action classes considered in those works are limited to simple actions such as walking, running and jumping.

There are several works which deal with action classes involved with multiple people. These classes are more complex to recognize than the single-person actions due to the complexity of the actions. The actions are defined not only by the motion of one person but also their interactive motions. We call these multiple-people actions as interactive actions.

The examples of interactive actions we consider in this work are exchanging objects and greetings between two people. For exchanging objects, we consider different types of the objects such as cards, a ball and a chair. Depending on the object being exchanged, their interactive movement come out in a different form. For greetings, we consider interactive actions such as shaking hands and bowing. Our goal is to recognize these different types of interactive actions from the 3d skeleton information.

Action recognition from the body skeleton information has been actively studied due to the advancement of motion capture systems such as Kinect. 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 most of the previous research has been on the recognition part.

[1] proposed a new skeleton representation based on relative geometry across body parts that achieves state of art results on several standard single-person action recognition datasets. The benefit of their approach compared to the previously adopted skeleton representation is that their representation can directly capture the relationship between body parts that are not directly connected each other. Thus, it is more capable of recognizing actions in which the relationship between non-connected body parts, such as left arm and right arm, is more important. In this work, we extend [1] to classify actions performed by two people.

For the recognition part, we develop an algorithm which not only predicts the action class but also localizes frames where actions are taking place. We perform this using two methods.

- (Baseline), we use a heuristic action localization method described in Section 2. We extract the features on the localized frames and use popular techniques like SVM, K-nn and Classification trees in a mutli-class classification setting to get the action class labels.

- (Proposed method) we use a Latent Structured SVM framework for localizing action and recognizing the action label simultaneously. Specifically, we treat the staring and ending frames of the action as latent variables and jointly optimize the cost function in a max-margin manner. Note that this method does not require annotation for latent variables as they are automatically determined in the optimization process.

Our experiments indicate the effectiveness of the latent structured SVM based method over the baseline method.

The paper is organized as follows: Section 2 describes the proposed skeleton representation at each frame. Section 3 explains our heuristic action localization method. Section 4 presents the Joint Action Localization and Detection method that we propose, using a Latent Structural SVM framework. The comparison to some state-of-art approaches is done in Section 5. The Results are shown in Section 6, followed by Conclusion in Section 7.

## 2. Heuristic Approach for Action Localization

The action sequences that are collected from the Kinect consists of several irrelevant frames which does not contain the action that we are looking for. Thus, given a video sequence it becomes important to approximately localize the action. This is a mandatory pre-processing step to extract realiable features that completely describe the action that we are interested in. We alos assume throughout this work that there is only one continuous action that takes place throughout a given video sequence. In this section, we decribe a heuristic way to localize the action in a given video. A given video sequence consists of the following parts:

$$< Init.Action - MainAction - Finish > \qquad (1)$$

The heuristic described here tries to acquire the frames corresponding to the Main Action in an automated way. We perform the localization in two steps:

- Computing the action center frames
- Computing the start and end frames

The action center frame is computed as the frame where the distance between the interacting persons is the least. If there are multiple frames with the same distance value (this could occur since some interactions can last for a few frames), the action center frame is assigned as the minimum of those frames.

The start(end) of any interaction in the video could be implied by a constant decrease(increase) in distance between two persons. Thus, we compute the distance between the persons at each frame and identify when the rate of change of distance goes above(start) or below (end) a given threshold $\delta$, which is a user input parameter.

As can be observed, this method does not yield us exact localization but an approximate one. In the subsequent section, we take care of this by imposing temporal smoothness on the features that we extract from each frame.

## 3. Skeleton Representation

The underlying notion of using a skeleton representation for human actions is that the human motion can be represented as a set of rigid body motions, where the rigid bodies are the different body joints. The output from the Kinect sensor is 3d positions of 20 body joints for each person defined in the global coordinate. Since most of our action classes only involve with upper body motion, in this work we use only 12 upper body joints ( See Fig.1).

We define our skeleton representation for the first person as $S^{(1)} = (V^{(1)}, E^{(1)})$, where $V^{(1)} = \{v_1^{(1)}, v_2^{(1)}, \ldots, v_N^{(1)}\}, N = 12$ denotes the set of joints and $E^{(1)} = \{e_1^{(1)}, e_2^{(1)}, \ldots, e_M^{(1)}\}, M = 11$ denotes the set of body parts 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 $\theta$ 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 $\binom{22}{2}$ such pairs, we have $\binom{22}{2} \times 7 = 1617$ 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 a current frame.

To extract features from a sequence of frames specified by a starting frame and ending frame, we first compute our skeleton feature vectors at each frame between the starting frame and ending frame. In order to obtain a fixed length vector regardless of the number of frames, we apply temporal interpolation at each dimension of the feature vector to sample values at 20 uniformly distributed time instances. Finally we concatenate all the feature vectors into a final vector. Thus, the length of the final vector extracted from frames between the given starting frame and ending frame is $20 \times 3234 = 64680$.

## 4. Joint Action Localization and Classification

In Sec.2, we proposed a heuristic approach for action localization where the aim is to automatically determine when the action starts and ends. The problem of this approach is that it is not designed to achieve better classification accuracy. Also it is purely subjective to define the start and the end of the action.

Thus, we propose a method which can do both tasks si-

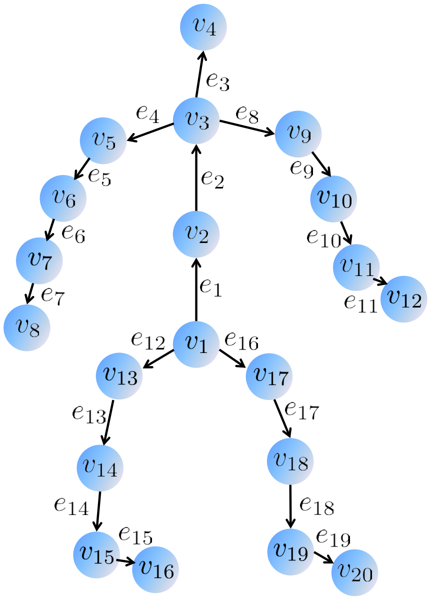


Figure 1: An illustration of the skeleton.

multaneously. In the proposed method, the training does not require manual labeling of the starting and ending frame of each action sequence. The model is trained in such a way that the starting and ending frame are determined in order to maximize the discriminative power of the model.

Our method is based on a latent structured SVM framework [6]. In this framework, we are given a set of training data

$$\mathcal{S} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \in (\mathcal{X}, \mathcal{Y})^N$$

where in our case, $x$ is skeleton feature extracted from a whole sequence and $y$ is an action class label. Let $h$ denote a latent variable, which specifies the starting and ending frame of the sequence.

The prediction is done by finding $h$ and $y$ which maximize the following scoring function:

$$f_{\mathbf{w}}(x) = \operatorname*{argmax}_{(y,h)\in\mathcal{Y}\times\mathcal{H}} [\mathbf{w}\cdot\Phi(x,y,h)] \qquad (2)$$

where $\Phi(x,y,h)$ is a joint feature map. As a result, we not only obtain the predicted action class but also the starting and ending frame of the action in the given sequence.

In this work, the joint feature map $\Phi(x,y,h)$ is computed as follows. We first extract skeleton features explained in Sec.3 from frames between starting frame and ending frame specified by $h$. We then construct $\Phi$ by setting $y$-th section of $\Phi$ to $x^*$ and remaining part to 0:

$$\Phi = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ x^* \\ \vdots \\ 0 \end{pmatrix} \in R^{|\mathcal{Y}|\times dim(x^*)} \qquad (3)$$

This way, for each action class, different part of $\mathbf{w}$ is used to compute the score and the class which has the highest score is chosen by Eq.2.

The training is done by solving the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}[\max_{(\tilde{y},\tilde{h})\in(\mathcal{Y},\mathcal{H})}[\Delta(y_i,\tilde{y}) + \mathbf{w}\cdot\Phi(x_i,\tilde{y},\tilde{h})] - \max_{h\in\mathcal{H}} \mathbf{w}\cdot\Phi(x_i,y_i,h)] \qquad (4)$$

where $\Delta(y_i, y_j)$ is a loss function defined as

$$\Delta(y_i, y_j) = \begin{cases} 0 & (y_i = y_j) \\ 1 & (otherwise) \end{cases}$$

By rewriting Eq.4,

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}[\max_{(\tilde{y},\tilde{h})\in(\mathcal{Y},\mathcal{H})}[\Delta(y_i,\tilde{y}) + \mathbf{w}\cdot\Phi(x_i,\tilde{y},\tilde{h})]] - C\sum_{i=1}^{N}[\max_{h\in\mathcal{H}} \mathbf{w}\cdot\Phi(x_i,y_i,h)]$$

it is clear that the objective function is non-convex as it is defined as the difference between two convex functions. We thus find a local optimum by EM-like algorithm where we first find, for each training sample, $h$ which maximize $\mathbf{w}\cdot\Phi(x_i,y_i,h)$. In our setting, this corresponds to finding the starting and ending frame for a training sample which make the score as high as possible.

In the second step, we fix $h$ and minimize $||\mathbf{w}||^2 + C\sum_{i=1}^{N}\max_{(\tilde{y},\tilde{h})\in(\mathcal{Y},\mathcal{H})}[\Delta(y_i,\tilde{y}) + \mathbf{w}\cdot\Phi(x_i,\tilde{y},\tilde{h})] - \mathbf{w}\cdot\Phi(x_i,y_i,h^*)$ where $h^*$ is found in the previous step. Note that this optimization problem is convex. The maximization in the objective function corresponds to finding the most strict constrain for the optimization problem. To put it simply, we find a class ($\tilde{y}$ and $h$ where ($\tilde{y}$ is different from the true class but its score is high.

The training is done by iteratively executing the above two steps while the decrease of the objective value becomes marginal.

## 5. Comparison to State-of-Art

The problem of Action Recongition has been studied for different purposes by different communities including

Computer Vision and Robotics. Most of the earlier approaches concentrate on recogizing Single Person Actions. Being a tough problem in its own right, the more interesting and informative actions are one that take place between a group of people. As a specific case, this work concentrates on Two-person interactions. There have been quite a few works that approach the same problem. These could be braodly divided into two categories: Semantic and Discriminative. An example of a Semantic approach is [2], which builds a discriminative model by describing an action as an operational triplet : $< Agent - Motion - Target >$. They enumerate all possible combinations of the operational triplet in terms of different body parts and the different poses that they take. To impose temporal consistency, a dynamic Bayesian network is used and action classification is performed for each frame over a given video. Another work closest to the Action Localization decribed in the previous section is the Similarity Constrained Latent SVM, where they present an algorithm that learns from input videos that have action labels, and produces a classifier that can label test videos and mark a discriminative region of interest.

The novelty in our work comes from the fact that we extend the Skeleton Representation work of [1] to two-person Interaction sequences. In our skeleton representation, in addition to the relative geometry across body parts within one person, we have also considered relative geometry between body parts across two people. We believe that this representation enables us to capture two person interactions well. For instance, our representation can capture the geometric relationship between a left arm of a person A and a right arm of a person B, which might be helpful to recognize the 'shaking hands' actions. The Latent SVM based approach that has been proposed in this work to perform Joint Action localization and Recognition has not been done previously for actions involving more than one person.

# 6. Experiments

## 6.1. Data Collection

We create a new dataset named Maryland Interactive Action Dataset (MIAD) which consists of 10 interactive action classes captured by Kinect. The action classes we consider are summarized in Fig.2 with their representative images. For each pair of people, we collect 2 sequences per an action class by switching their locations. Since we collect data from 6 pairs of people, there are 12 sequences per action class.

Our new data, MIAD differs from the existing two people interactive dataset such as K3HI [3] and SBU Kinect Interaction Dataset [4]. First, our skeleton is represented by 20 joints whereas K3HI and SBU have only 15 joints. Second, the action classes we consider are more 'fine-grained'. In K3HI and SBU, there is only one action class for 'exchanging an object' class while MIAD contains 5 unique action classes for it, differentiated by the object being exchanged. Also K3HI and SBU have only one action class for the greeting action, namely, shaking hands, while MIAD includes 5 different action classes. These fine-grained actions are generally more challenging to discriminate as actions are more similar each other.

## 6.2. Action Localization Results

In this section we show some sample results of our Action localization procedure described in Section **??**.

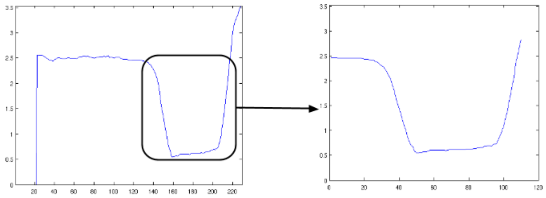


Figure 3: Distance Profile before(left) and after(right) localization

Figure 3 shows the difference in the distance profiles between the raw action sequence and the processed action sequence. As can be seen from Figure 3, the processed distance profile shows an approximately zoomed in portion of the original profile. This localized region represents the frames in the given video as to where the interaction takes place.

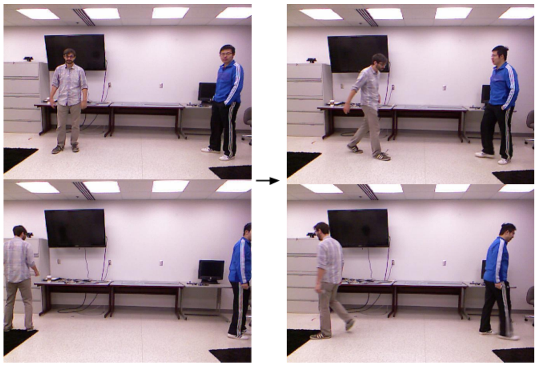


Figure 4: Start(Top) and End(Bottom) Frames

Figure 4 shows the start and end frames corresponding to the above action profiles. Thus, we use this simple heuristic method to extract only the relevant portion of the action sequence and provide this as input to the later processing.

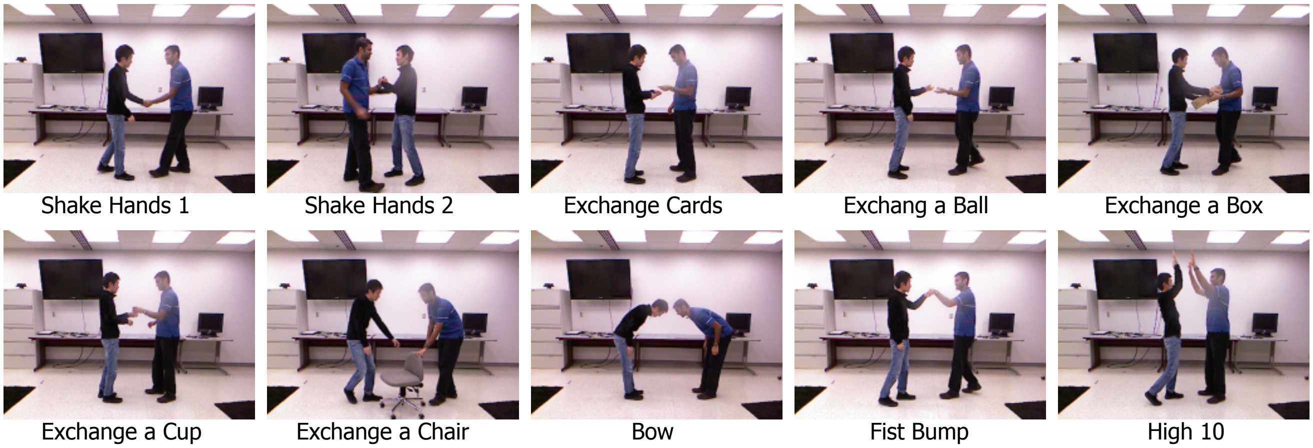


Figure 2: Action classes in newly collected MIAD dataset.

## 6.3. Action Classification Results

### 6.3.1 Baseline

We first conduct experiments using a baseline method explained below. We compute skeleton features from frames localized by the heuristic action localization method. To classify a given test input to one of the action classes, we learn a discriminative mapping between the input space and the output action label space. In the training stage, we learn the parameters of this mapping/classifier and apply them to classify the test input. In this work, we have used the following classification methods:

- Support Vector Machines (SVM): one (vs) one, one (vs) all
- k- Nearest Neighbour: k = 1,4
- Classification Trees

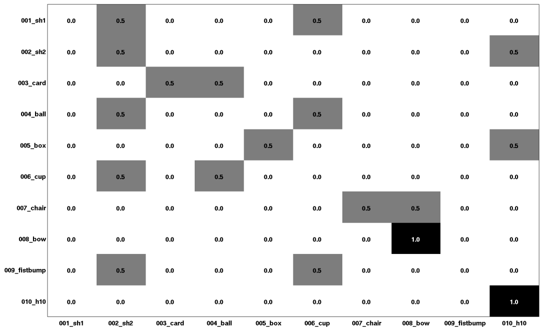
SVM's are desgined to handle binary classification problems. To use them for multi-class classification, we use two paradigms which are explained as follows:

- **1(vs)1**: In this case, for each class combination $(i, j)$ one binary classifier is learnt. Thus for a $N$ class problem, the total number of classifiers becomes $\frac{N(N-1)}{2}$.
- **1(vs)all**: In this case, for each class $i$ in the training set, one classifier is learnt. The positive examples for this classifier are the inputs from class $i$ while the ngative examples are the rest of the inputs from all other classes. Thus, $N$ classifiers are learnt, one per class.

Since SVM's produce only the decision values for a multi-class setting, we use Platt scaling as described in [5] to convert these decision values to valid posterior probability estimates. Thus, for each classifier, for each test input $x_t$, we obtain the posterior probability measure $p(y = i|x_t)$, where $i$ is any class label between $\{1, 10\}$ and assign that $i$ to $x_t$ which maximizes the probability measure. For each of these methods, we use different experimental settings by using different amounts of training and test data for each setting. The parameters for SVM are tuned using cross-validation on the training set.

Table 1: Classification accuracy of the baseline methods

| Split-Ratio | SVM(1-1) | (1-all) | knn-1 | knn-4 | CT |
|---|---|---|---|---|---|
| 20:80 | 24.28 | 34.85 | 26.32 | 23.06 | 10.61 |
| 40:60 | 29.85 | 43.71 | 32.17 | 30.72 | 32.46 |
| 50:50 | 33.33 | 46.46 | 37 | 20.73 | 36.06 |
| 60:40 | 35.51 | 51.9 | **41.63** | 32.24 | 38.36 |
| 80:20 | **43** | **57** | 37 | **33** | **47** |



| | 001_sh1 | 002_sh2 | 003_card | 004_ball | 005_box | 006_cup | 007_chair | 008_bow | 009_fistbump | 010_h10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 001_sh1 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| 002_sh2 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 |
| 003_card | 0.0 | 0.0 | 0.5 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 004_ball | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| 005_box | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 |
| 006_cup | 0.0 | 0.5 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 007_chair | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.5 | 0.0 | 0.0 |
| 008_bow | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 |
| 009_fistbump | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 |
| 010_h10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |

Figure 5: Confusion matrix for the best performing classification method using heuristic action localization(SVM one-vs-all). Darker color indicates larger values

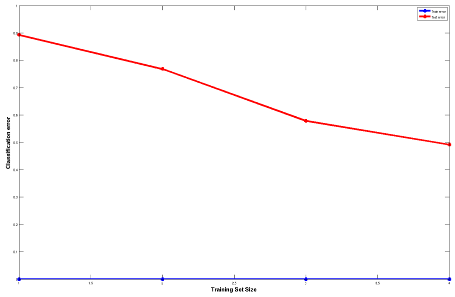


Figure 6: Learning Curve for SVM one-vs-all method

The results for our classification method are shown in Table 1. Figure 5 shows the confusion matrix for the best performing classification method at 80:20 split. The learning curve for the best performing classifier is shown in Figure 6. The learning curve shows that our model has very high variance, since the test error is much higher than the error on the training set. This implies that we can bridge the gap and thus improve the accuracy of our method by using more training data.

Both labels 1, 2 correspond to variations of the shaking hands action, one being the normal way and the other being an upright shake hands action. The scenario is shown in Figure 7. Given that these actions are very difficult to distinguish and that different people perform these actions in different ways (such that the disriminative aspect is lost), the features extracted from the limited training data that we have could not clearly distinguish between these actions.

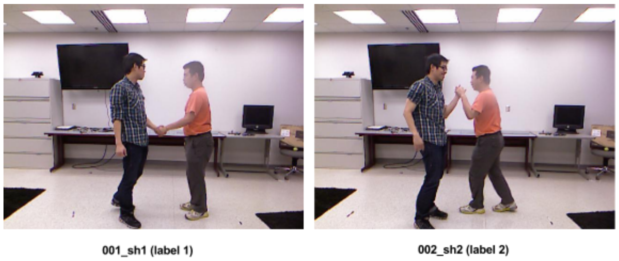


Figure 7: Variants of the shake hands action (Labels 1 and 2)

#### 6.3.2 Joint Action Localization and Recognition

We evaluate our proposed joint action localization and recognition method. The classification accuracy for 50:50 split is 58%, which outperforms all the baseline method. More over, our method can also localize the action frames. The confusion matrix is shown in Table 2 and some sample localization results are shown in Fig.8.

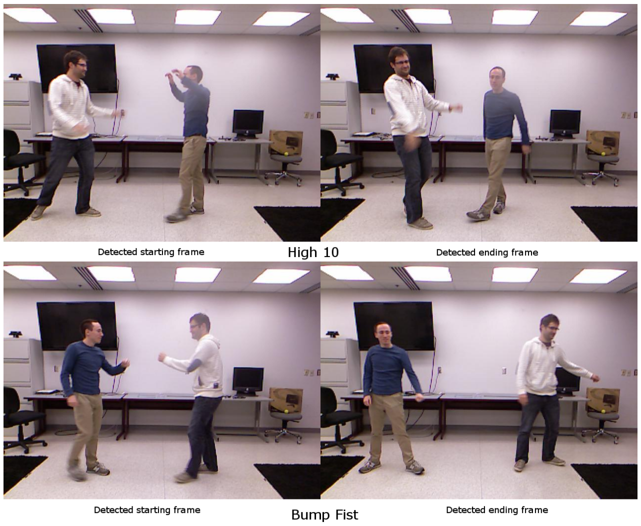


Figure 8: Sample action localization results

From the confusion matrix in Table 2, we can observe that the most difficult action to recognize is 'Exchange a Box'. By inspecting the recorded sequences, we found out that many of the upper body joints are occluded by the box, which we believe makes the recognition hard. This problem is also observed for 'Exchange a Chair' class, which makes the accuracy low for this class.

## 7. Future Work

Actions involved in exchanging objects such as box, ball and cup could be more accurately classified by giving an additional information to the classifier involving the description of the object that appears in these frames. This part is a work under progress where we are trying to find an effective way to obtain an object description that would not be very computationally expensive.

## 8. Conclusion

In this work, we have attempted to classify Interactions involving two people. For this purpose, we have collected a database of Interactions that involve exchange of object and greeting actions. We have extended the existing state-of-art skeleton representation technique to two-person interaction sequences. The results were obtained by building a discriminative mapping between the action sequences and the action labels. We used several classifiers including SVM, k-nearest neighbour and Classification Trees. We also propose a method based on the latent structured SVM where action localization and recognition is jointly addressed. Furthermore, we are also exploring the use of attributes as a tool for classification, while also deriving suitable interpretation of an action sequence as would be described by a human.

Table 2: Confusion matrix of the proposed joint action localization and recognition method

| | Shake Hands 1 | Shake Hands 2 | Exchange Cards | Exchange a Ball | Exchange a Box | Exchange a Cup | Exchange a Chair | Bow | Fist Bump | High 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Shake Hands 1 | 33 | 0 | 0 | 17 | 0 | 50 | 0 | 0 | 0 | 0 |
| Shake Hands 2 | 0 | 50 | 0 | 0 | 0 | 33 | 0 | 0 | 0 | 17 |
| Exchange Cards | 0 | 0 | 67 | 17 | 0 | 17 | 0 | 0 | 0 | 0 |
| Exchange a Ball | 17 | 0 | 0 | 33 | 0 | 33 | 0 | 0 | 17 | 0 |
| Exchange a Box | 17 | 0 | 17 | 50 | 0 | 0 | 0 | 0 | 17 | 0 |
| Exchange a Cup | 0 | 0 | 25 | 0 | 0 | 75 | 0 | 0 | 0 | 0 |
| Exchange a Chair | 0 | 0 | 0 | 17 | 0 | 33 | 17 | 33 | 0 | 0 |
| Bow | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| Fist Bump | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| High 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

## References

[1] Raviteja Vemulapalli, Felipe Arrate, Rama Chellappa, *Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group*. CVPR, 2014. 1, 4

[2] Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy, Sangho Park, J.K. Aggarwal, CVPR'04 4

[3] Sangho Park, J.K. Aggarwal, *K3HI: Kinect-based 3D Human Interaction Dataset* 4

[4] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras, *Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning*. CVPRW, 2012. 4

[5] John C. Platt, *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. ADVANCES IN LARGE MARGIN CLASSIFIERS, MIT Press, 1999. 5

[6] Learning structural SVMs with latent variables, Chun-Nam John Yu and Thorsten Joachims, ICML'09. 3