Learning and Recognition of Object Manipulation Actions Using Linear and Nonlinear Dimensionality Reduction

Isabel Serrano Vicente, Danica Kragic and Jan-Olof Eklundh
Computational Vision and Active Perception Lab and Centre for Autonomous Systems, KTH, Stockholm, Sweden
isasevi, danik, joe@nada.kth.se

***Abstract*— In this work, we perform an extensive statistical evaluation for learning and recognition of object manipulation actions. We concentrate on single arm/hand actions but study the problem of modeling and dimensionality reduction for cases where actions are very similar to each other in terms of arm motions. For this purpose, we evaluate a linear and a nonlinear dimensionality reduction techniques: Principal Component Analysis and Spatio-Temporal Isomap. Classification of query sequences is based on different variants of Nearest Neighbor classification. We thoroughly describe and evaluate different parameters that affect the modeling strategies and perform the evaluation with a training set of 20 people.**

I. Introduction

In robotics, recognition of human activity has been used extensively for robot task learning through imitation and demonstration, [1], [2], [3], [4], [5], [6], [7], [8]. It has been shown in [9] that an action perceived by a human can be represented as a sequence of clearly segmented *action units*. This motivates the idea that the action recognition process may be considered as an interpretation of the continuous human behaviors which, in its turn, consists of a sequence of action primitives [6] such as *reaching*, *picking up*, *putting down*. In relation, learning *what* and *how* to imitate has been recognized as an important problem, [8]. It has been argued that the data used for imitation has statistical dependencies between the activities one wishes to model and that each activity has a rich set of features that can aid both the modeling and recognition process. In [4], a framework for acquiring hand-action models by integrating multiple observations based on gesture spotting is proposed. The work presented in [5] proposes a gesture imitation system where the focus is put on the coordinate system transformation (*View-Point Transformation*) so that the teacher induced gesture is transformed into the robot's egocentric system. This way the robot *observes* the gesture as it was generated by the observer himself. The work in [6] approaches the task learning problem by proposing a system for deriving behavior vocabularies or simple action models that can be used for more complex task extraction and learning. A learning system for one and two-hand motions where the robot's body constraints are considered as a part of the optimal trajectory generation process has been presented in [8]. Our work differs from the work above in that we perform a thorough analysis of how the dimensionality of the data as well as the number and placement of sensors affect the recognition rate. In addition, the above cited work studies actions which are very different in nature while we concentrate on actions that are very similar to each other and thus difficult to disambiguate. The studied actions are basic building blocks of any imitation based learning system and the contribution of our work is in the evaluation of the suitability of the known methods. An interesting trend to note here is that most of the studies are based on a **single** user generated motion. A natural question to pose here is how the underlying modeling methods scale and apply for cases when the robot is supposed to learn from multiple teachers. The experimental evaluation conducted in our work is based on 20 people.

We perform an extensive statistical evaluation and stress that achieving the high recognition rates is not the focus of our study but the evaluation of the existing techniques and their suitability for modeling and recognition of object manipulation actions. Such an evaluation, considering a training set of 20 people, has previously not been performed. Single arm/hand actions are considered with a specific focus on the problem of modeling and dimensionality reduction for cases where actions are very similar to each other in terms of arm motions. For this purpose, we evaluate a linear and a nonlinear dimensionality reduction techniques: Principal Component Analysis and Spatio-Temporal Isomap. Classification of query sequences is based on a combination of clustering and different variants of Nearest Neighbor classifiers. For both methods, we thoroughly describe and evaluate different parameters that affect the modeling strategies and perform the evaluation with a training set of 20 people. To our knowledge, there are no examples in the field of robotics where such a large set of people was considered. The results can be used to enable a more sophisticated probabilistic modeling and recognition of actions for methods such as those presented in [6], [8].

In Section II we describe the experimental setting and data collection. In Section III we give a short overview of dimensionality reduction techniques and present details of their implementation in Section IV. Experimental evaluation is summarized in Section V and paper concluded in Section VI.

II. Data Collection and Preprocessing

We follow the classical approach to activity recognition through training and testing steps. The system learns a model for each activity which is then used for the classification of new actions in the testing step. The four activities considered in this work are:

1) Push forward an object placed on a table (P);
2) Rotate an object placed on table (R);
3) Pick up the object placed on the table (PU) and
4) Put down an object on a table (PD).

Notations P, R, PU, PD are used to denote different actions in the experimental evaluation in Section V.

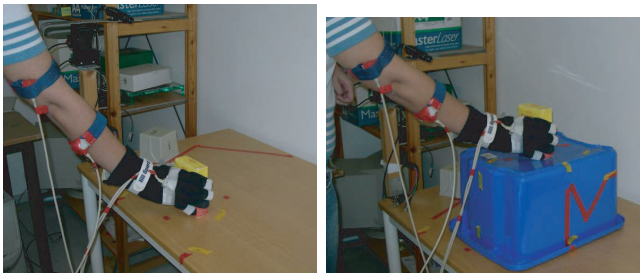

Fig. 1. Left) An example of pushing forward an object on the table and Right) An example of pushing forward an object on the box

Fig. 1 shows two example images stored during a push activity training - the activity is performed with the object being placed at two different heights. To motivate the choice of these activities, let us consider a robot being a part of a coffee drinking scenario. A *pick up* activity could be representing the fact of picking up the cup to take a swig of coffee; *put down* an object could represent leaving the cup of coffee after taking a swig, *rotate* an object would be similar to fold a napkin placed on the table, and finally, let us suppose that the person who sat down in front of you taking a coffee asks for the sugar bowl close to you and you *push* the bowl sliding over the table to bring it closer to him/her. The activities considered in this work are major building blocks of any similar task.

To generate the measurements for the training data, a Nest of Birds sensors were used which track the position and orientation of four sensors, referred to transmitter emitting pulsed DC magnetic field. The placement of the sensors is shown Fig. 1: thumb, hand, lower arm and upper arm. Apart from the variation in their height and velocity with which an action was performed, the following variations were introduced to the training data:

- The objects were put on two different heights
- The person was standing at three different angles with respect to the table: 0, 30 and 60 degrees

Each action was performed three times for all combinations of the above heights and orientations resulting in total 18 training sequences per person and action thus 360 training sequences for each action.

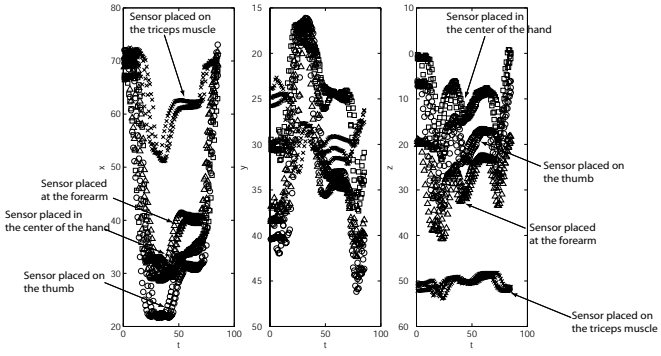

Fig. 2. Sensor measurements retrieved for three trials of a "rotate" activity.

III. Dimensionality Reduction

Finding low-dimensional representation of high-dimensional observations is one of the key problems in the area of activity modeling and recognition. In the current study, we have evaluated two dimensionality reduction methods. The first is the classical PCA, [10] where each data point is reconstructed by a linear combination of the principal components. For cases where the data represents essential nonlinear structures, PCA and similar techniques fail to detect the intrinsic dimensionality and model for the data. Therefore, we also evaluate a nonlinear dimensionality reduction approach proposed in [6] which is based on the isometric feature mapping or Isomap, [11].

A. Principal Component Analysis (PCA) and Isometric Feature Mapping (Isomap)

PCA method retains those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components. The idea is that such low-order components often contain the "most important" aspects of the data if the assumption of linearity holds.

The main idea of Isomap, [11] is to find the intrinsic geometry of the data by computing the geodesic manifold distances between all pairs of data points. Once the geodesic distances are estimated, multidimensional scaling is applied which removes nonlinearities in the data and produces a coordinate space intrinsic to the underlying manifold. Since the training data in our system are represented in a global coordinate system (robot centered), the system should be able to perform disambiguation of spatially proximal data that are structurally different (*pick up* and *put down*) as well as model the correspondence of spatially distal data points that share common structure (actions performed at different heights). An extension of the classical Isomap, the ST-Isomap, proposed in [6] is a method that satisfies the above requirements. Implementation details are presented in Section IV-C.

B. Clustering Methods

We have evaluated two clustering techniques in connection to PCA based action classification: *k*-means clustering and Gustafson-Kessel clustering. *k*-means clustering [10] is a partitioning method in which clusters are mutually exclusive (hard partitioning method). Clustering algorithms group sample points, $\mathbf{m}_j$ into c clusters. The set of cluster prototypes or centers is defined as $\mathbf{C} = \left[\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(c)}\right]$ where

$$\mathbf{c}^{(i)} = \frac{\sum_{j=1}^{d} u_{ij} \ \mathbf{m}_j}{\sum_{j=1}^{d} u_{ij}} \qquad i = 1, 2, \ldots, c \tag{1}$$

where $u_{ij} \in \mathbf{U}$ denotes the membership of $\mathbf{m}_j$ in the ith cluster and $\mathbf{U}$ is known as the *partition matrix*. For the classical *k*-means clustering, the hard partitioning space is defined as:

$$M_h = \{\mathbf{U} \in V_{cd} : u_{ij} \in \{0,1\}, \forall (i,j); \sum_{i=1}^{c} u_{ij} = 1; 0 < \sum_{i=1}^{d} u_{ij} < d, \forall i\} \tag{2}$$

The objective function we have to minimize is:

$$J_h(\mathbf{M}; \mathbf{U}, \mathbf{C}) = \sum_{i=1}^{c} \sum_{j=1}^{d} u_{ij} d_{\mathbf{A}}^2 \left(\mathbf{m}_j, \mathbf{c}^{(i)}\right) \tag{3}$$

where A is a norm-inducing matrix and $d^2_{\mathbf{A}}$ represents the distance measure

$$d^2_{\mathbf{A}} = \left(\mathbf{m}_j, \mathbf{c}^{(i)}\right) = \|\mathbf{m}_j - \mathbf{c}^{(i)}\|^2_{\mathbf{A}} = \left(\mathbf{m}_j - \mathbf{c}^{(i)}\right)^T \mathbf{A} \left(\mathbf{m}_j - \mathbf{c}^{(i)}\right) \tag{4}$$

The above condition of hard membership can be relaxed so that each sample point has some graded or "fuzzy" membership in a cluster. The incorporation of probabilities (or graded memberships) may improve the convergence of the clustering method compared to the classical k-means method. In addition, we do not have to assume anymore that the samples belong to spherical clusters. We shortly describe the method used in our work also known as Gustafson-Kessel (GK) clustering. First, we define a fuzzy partition space as:

$$M_f = \{\mathbf{U} \in V_{cd} : u_{ij} \in [0,1], \forall (i,j); \sum_{i=1}^{c} u_{ij} = 1; 0 < \sum_{i=1}^{d} u_{ij} < d, \forall i\} \tag{5}$$

Here, fuzzy objective function is a least-squares functional:

$$J_f(\mathbf{M};\mathbf{U},\mathbf{C}) = \sum_{i=1}^{c} \sum_{j=1}^{d} (u_{ij})^w d^2_{\mathbf{A}} \left(\mathbf{m}_j, \mathbf{c}^{(i)}\right) \tag{6}$$

where w is a weighting factor $w = [1, \infty)$. Gustafson-Kessel method is a variation of fuzzy clustering algorithms which allows the samples to belong to several clusters simultaneously, with different degrees of membership. It employs an adaptive distance norm in order to detect clusters of different geometrical shapes in the data set. Specifically, each cluster has its own norm-inducing matrix $\mathbf{A}^{(i)}$:

$$d^2_{\mathbf{A}^{(i)}} = \left(\mathbf{c}^{(i)}_l - \mathbf{m}_j\right)^T \mathbf{A}^{(i)} \left(\mathbf{c}^{(i)}_l - \mathbf{m}_j\right) \tag{7}$$

where

$$\mathbf{A}^{(i)} = (|\mathbf{F}^{(i)}|)^{1/(r+1)} (\mathbf{F}^{(i)})^{-1} \tag{8}$$

$$\mathbf{F}^{(i)} = \frac{\sum_{j=1}^{d} (u_{ij})^w \left(\mathbf{m}_j - \mathbf{c}^{(i)}\right) \left(\mathbf{m}_j - \mathbf{c}^{(i)}\right)^T}{\sum_{j=1}^{d} (u_{ij})^w} \tag{9}$$

IV. Implementation

We give a short overview and implementation details for the methods used in this study.

A. PCA without temporal dependencies

These measurements are gathered by 1,2,3 and/or 4 sensors where each sensor provides a full pose estimate. The assumption is that each action consists of a set of discrete poses represented in a high-dimensional space. Each rotation angle is represented by its sine and cosine value resulting in 9 measurements in total per sensor. Our reasoning here was that different actions will vary differently along different directions. If we are able to find this directions, each action may be represented only with those ones along which the data varies the most, precisely what PCA gives us. The implementation follows the classical PCA approach: we first estimate the mean of the data, subtract it from all the samples, estimate the covariance matrix and estimate its SVD, [10]. Finally, we keep only the eigenvectors that for which eigenvalues $\lambda_n > 0.005\lambda_{max}$. In our evaluation, the dimensionality reduction was following: single sensor (from 9 to 3), two sensors (18 to 5), three sensors (27 to 6) and four sensors (36 to 7). These values are easy to understand due to the constraints posed by the kinematic structure of the arm. Once the basic set of eigenvectors is chosen, the training data is projected to this reduced space. This is done for each action separately. To ease the classification, we cluster each action representation space. For this purpose, we have used k-means and GK clustering presented in Section III-B. In the classification stage, each testing sequence is first projected to the reduced action representation space. For each sample point in an action, the distance to the closest cluster center is estimated and the classification is based on the Euclidean distance.

B. PCA with Temporal Dependencies

We have also evaluated a PCA approach where we took into account the temporal dependencies in the data. To be able to estimate the covariance matrix using whole sequences, we normalized them to equal length - 85 sample points per sequence. According to the procedure described in the previous section, the dimensionality reduction was following: single sensor (from 765 to 17), two sensors (1530 to 22), three sensors (2225 to 24) and four sensors (3060 to 26). Training sequences are then projected to separate decreased spaces where each represents one of the actions. Classification of a new sequence is performed based on the Euclidean distance.

C. ST-Isomap

For the implementation of Isomap, we adopted the approach proposed in [6]. As in the case of temporal PCA, the sequences are first normalized to equal length of 85 sample points. We shortly explain the basic idea behind the method.

- Calculate a distance matrix D^l between N local neighbors using Euclidean distances. In the current implementation, $N = 10$. For each data sample, identify common temporal neighbors (CTN) and adjacent temporal neighbors (ATN). We refer to [6] and [12] for a more detailed definition of these.
- Reduce the distances in the original matrix taking into account spatio-temporal correspondences

$$D^0_{S_i,S_j} = \begin{cases} D^l_{S_i,S_j}/(c_{CTN}c_{ATN}) & \text{if } S_j \in CTN(S_i) \text{ and } j = i+1, \\ D^l_{S_i,S_j}/c_{CTN} & \text{if } S_j \in CTN(S_i), \\ D^l_{S_i,S_j}/c_{ATN} & \text{if } j = i+1, \\ penalty(S_i,S_j) & \text{otherwise.} \end{cases} \tag{10}$$

  where c_{ATN} and c_{CTN} are scalar parameters and $CTN()$ denotes common temporal neighbors. We set $c_{ATN} = 1$ and vary value for $c_{CTN} = [2 \quad 5 \quad 10 \quad 100]$. Fig. 3 shows the effect of c_{CTN} parameter to the resulting embedding of the activities.
- Use D_0 to compute shortest path distance matrix D_g using Dijkstra's algorithm, [13]
- Use Multidimensional Scaling [14] to embed D_g to a lower dimensional space. We have evaluated the system for $[3 \quad 4 \quad 5 \quad 6]$ dimensions.

V. Experimental evaluation

We present the results for i) PCA without temporal dependencies, ii) PCA with temporal dependencies and iii) ST-Isomap.

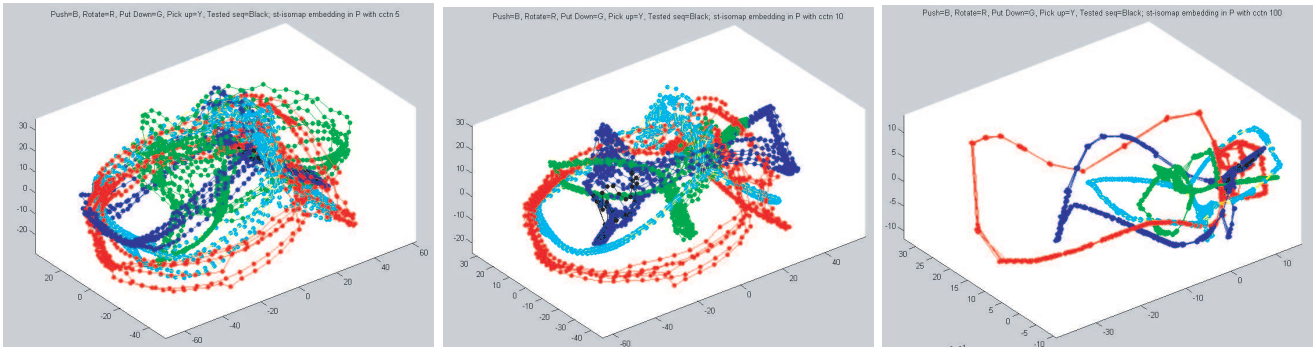

Fig. 3. Training data after estimating ST-Isomap and MDS embedding in 3 dimensions. The figures show the influence of the c_{CTN} parameter to the embedding: higher c_{CTN} brings sequences closer to each other.

A. PCA without temporal dependencies

We have trained the system with 1, 5, 10 or 20 people. In case of 1,5, 10 persons, we split the data in three possible combinations of two trials for training and one trial for evaluation. For 20 people, we split the trials in three possible combinations of two for training the system and one for testing it, so we test the system three times with the demonstrations of all people. In all cases we clustered the data using both *k*-means and GK-clustering algorithms using three, five and eight clusters. Here, we show the resulting average of all the experiments and refer to [12] for a more detailed evaluation. In the forthcoming tables, the actions in the upper row are the tested sequences and the actions in the left column are the result of the classification. The results are expressed in percentage.

As explained in Section II, for each action, we have varied the position of the object (two heights) and the relative orientation of the person with respect to the table. The first experimental evaluation considered only two actions (push and rotate) where training and testing was performed on sequences captured under the same conditions (same orientation and height of the object). The average results considering different number of people in the training set as well as different numbers of sensors are summarized in Table I. We note here that we present the results for 5 clusters in more detail since it gave the highest classification rate on average. It can be seen that for only two actions, a classification rate of close to 90% is achieved. The presented results use are based on *k*-means clustering. GK-clustering gave approximately the same classification rate.

The second experiment to conduct was to consider all four actions, again considering the same conditions for training and testing. Due to the limited space, we show only the average classification rates for all four actions. In Table II we show how the size of the training set affects the rate given that the number of clusters is kept constant. In Table III we show how the number of clusters affect the classification rate given that the training set consist of all 20 people. Compared to the previous experiment, we can see that by adding two additional actions, the recognition rate is 30% lower on average. Again, similar results are obtained for both clustering methods.

Finally, we have evaluated the method considering all the variance in the data, namely that each action was performed on two different heights and in three orientations. The results are summarized in Table IV. It is obvious that, with the

5 clusters

	push	rot	push	rot	push	rot	push	rot
1pers	1s		2s		3s		4s	
push	**91.8**	1.6	**90.5**	1.3	**91.5**	2	**92.1**	2
rot	8.2	**98.4**	9.5	**98.7**	8.5	**98**	7.8	**98**
5pers	1s		2s		3s		4s	
push	**80**	34.4	**83.3**	27.8	**83.3**	19	**87.8**	30
rot	20	**65.6**	16.7	**72.2**	16.7	**81**	12.2	**70**
12pers	1s		2s		3s		4s	
push	**79.6**	18.5	**74.5**	14.8	**82.4**	18	**82.4**	14.8
rot	20.4	**81.5**	25.5	**85.2**	17.6	**82**	17.6	**85.2**
20pers	1s		2s		3s		4s	
push	**83**	14.7	**91.4**	16.7	**92.5**	11.9	**93.1**	10.8
rot	17	**85.3**	8.6	**83.3**	7.5	**88.1**	6.9	**89.2**

3 clusters

20pers	1s		2s		3s		4s	
push	**89.7**	28.9	**88.6**	21.7	**93.1**	26.4	**91.7**	21.1
rot	10.3	**71.1**	11.4	**78.3**	6.9	**73.6**	8.3	**78.9**

8 clusters

20pers	1s		2s		3s		4s	
push	**88.1**	15.6	**86.9**	12.5	**90.6**	10.6	**91.1**	8.9
rot	11.9	**84.5**	13.1	**87.5**	9.4	**89.4**	8.9	**91.1**

TABLE I: Classification rates two actions (push, rotate) when the training and testing was done under same conditions (object height, persons orientation) using *k*-means clustering using different number of sensors (1-4s is 1 to 4 sensors).

1 pers	1 sensor	2 sensors	3 sensors	4 sensors
average	**91.4**	**91.1**	**90.2**	**90**
5 pers	1 sensor	2 sensors	3 sensors	4 sensors
average	**61.9**	**65**	**68.6**	**61.1**
12 pers	1 sensor	2s	3 sensors	4 sensors
average	**60.8**	**60.8**	**63.1**	**61.7**

TABLE II: Classification rates for four actions trained and tested in same conditions (height and orientation), with varying size of the training set.The number of clusters in *k*-means is 5.

3 clusters	1 sensor	2 sensors	3 sensors	4 sensors
average	**59.4**	**61.4**	**62.2**	**64.1**
5 clusters	1 sensor	2 sensors	3 sensors	4 sensors
average	**64.7**	**68.4**	**70.6**	**69.8**
8 clusters	1 sensor	2 sensors	3 sensors	4 sensors
average	**66.5**	**68**	**68.9**	**70**

TABLE III: Classification rates for four actions and 20 people trained and tested in the same conditions (height and orientation), with varying number of clusters.

1 pers	1 sensor	2 sensors	3 sensors	4 sensors
average	**37.5**	**30.6**	**37.5**	**37.5**
5 pers	1 sensor	2 sensors	3 sensors	4 sensors
average	**34.7**	**33.9**	**38.1**	**38.9**
12 pers	1 sensor	2 sensors	3 sensors	4 sensors
average	**34.3**	**33.7**	**37.5**	**35.6**
20 pers	1 sensor	2 sensors	3 sensors	4 sensors
average	**35.4**	**37.2**	**37.3**	**37.4**

TABLE IV: Classification rates for four actions trained and tested in different conditions, with varying size of the training set. The number of clusters used in *k*-means is fixed to five.

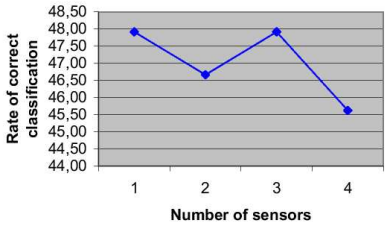

Fig. 4. The effect of number of sensors used to the classification rate.

the recognition rates of about 40%, the simple approach considered here is not able to scale accordingly with the variation in the data. The next section presents the results of the method where temporal dependencies between the data points are taken into account.

B. Temporal PCA

We present the results with all four actions, where the training and testing was performed given all 20 people and actions performed in all combinations of orientations and heights. As above, as each action sequence was performed three times in all conditions, we evaluated the system taken all combinations of two testing and one training action sets. Table V summarizes the results for one (1s, hand), two (2s, thumb and hand), three (3s, thumb, hand, forearm) and all four (4s) sensors considered. Important to note is that the recognition rate is somewhat higher compared to the results in the previous section but the system still has the difficulty of discriminating between some of the actions. We believe that this is an important result. Implementing PCA with temporal dependencies requires aligned and equal length sequences which may be difficult to obtain in an online process where we would like to perform recognition during and not after an action has been executed. A simple "voting" approach presented in the previous section may be as suitable. Another issue that we have investigated was if the number of sensors affects the classification rate. The results are summarized in Fig. 4. We note that the difference is only marginal and that almost equal results are obtained with a single or all four sensors. This means that for actions which are very similar using only a single sensor on the hand or tracking only the pose of the hand may be sufficient.

	1 sensor				2 sensors			
	P	R	PD	PU	P	R	PD	PU
P	**50.1**	42.5	12.5	29.2	**50**	43.3	12.5	32.5
R	8.3	**33.3**	3.3	10	9.2	**35**	3.3	15
PD	15	3.3	**69.2**	22.5	15	5.8	**69.2**	20
PU	25.8	20.8	15	**38.3**	25.8	15.8	15	**32.5**
	3 sensors				4 sensors			
	P	R	PD	PU	P	R	PD	PU
P	**51.7**	42.5	12.5	30	**51.7**	42.5	12.5	29.2
R	7.5	**35**	3.3	1.5	8.3	**30**	3.3	11.7
PD	14.2	5	**69.2**	21.7	14.2	4.2	**66.7**	25
PU	26.7	17.5	15	**35.8**	25.8	23.3	17.5	**34.2**

TABLE V: Classification rates for PCA with temporal dependencies for four actions and 20 people in the training set.

C. ISOMAP

A non-linear dimension reduction, ST-Isomap was applied to extract a low dimensional representation for the activities. Shepard interpolation [15] was used map a query sequence to the estimated embedding and Euclidean distance was used for classification. From the training set of 20 people, we formed subsets of one, two and three persons. For each person, all four activities were considered using three trials for all combinations of three orientations and two heights. The classification was performed with the queries not included in the training set. As before, we have evaluated the system with different numbers and sensors placements. In the forthcoming tables, this is denoted as: sensors placed on the i) hand (s1), ii) hand and thumb (s14), iii) hand, thumb and forearm (s142). Thorough experimental evaluation with different values for c_{CTN} parameter and dimensionality of the embedding space was conducted.

Fig.5 shows the results obtained by ST-Isomap with training based on a single person. The results show how the dimension of the embedding and sensor number affect the classification result. Here, parameter $c_{CTN} = 2$. Fig.6 shows a similar experiment, but here the size of the training set was three. It is interesting to notice that best results are obtained based on the sensor placed on the hand. For the future, this would motivate that only the position of the user's hand and not the complete arm joint motion is needed to recognize object manipulation sequences when ST-Isomap is used. The effect of changing the values of parameter c_{CTN} is shown in Table VI. It can be seen that, compared to the PCA, ST-Isomap gives better classification results.

		push	rot	pd	pu
ct = 2	push	**88.9**	22.2	29.2	36.1
ct = 2	rot	11.1	**70.9**	8.3	16.7
ct = 2	pd	0	0	**51.4**	16.7
ct = 2	pu	0	6.9	11.1	**30.5**
ct = 5	push	**88.9**	6.9	19.4	25
ct = 5	rot	0	**79.2**	4.2	9.7
ct = 5	pd	1.3	0	**62.5**	27.8
ct = 5	pu	9.7	13.9	13.9	**37.5**
ct = 10	push	**90.3**	18.1	25	29.2
ct = 10	rot	1.4	**72.2**	8.3	13.9
ct = 10	pd	2.8	2.8	**50**	30.5
ct = 10	pu	5.5	6.9	16.7	**26.4**
ct = 100	push	**84.7**	8.3	6.9	23.6
ct = 100	rot	5.6	**80.6**	5.6	4.2
ct = 100	pd	2.8	6.9	**65.3**	36.1
ct = 100	pu	6.9	4.2	22.2	**36.1**

TABLE VI: Classification results using a single sensor placed on the hand. Training was performed with 3 persons. The recognition rates show the dependency on the parameter c_{CTN}.

		s1				s14				s142			
		p	r	pd	pu	p	r	pd	pu	p	r	pd	pu
ct=2	p	55,6	0,0	11,1	11,1	61,1	11,1	5,6	22,2	50,0	50,0	33,3	50,0
3dimensions	r	5,6	77,8	0,0	0,0	5,6	61,1	61,1	38,9	44,4	50,0	50,0	16,7
3dimensions	pd	16,7	11,1	38,9	44,4	16,7	27,8	5,6	38,9	5,6	0,0	16,7	33,3
3dimensions	pu	22,2	11,1	50,0	44,4	16,7	0,0	27,8	0,0	0,0	0,0	0,0	0,0
ct=5	p	77,8	0,0	0,0	0,0	33,3	0,0	0,0	0,0	50,0	55,6	5,6	44,4
3dimensions	r	0,0	88,9	5,6	5,6	5,6	94,4	22,2	16,7	33,3	44,4	66,7	38,9
3dimensions	pd	22,2	0,0	66,7	11,1	61,1	5,6	33,3	33,3	0,0	0,0	0,0	0,0
3dimensions	pu	0,0	11,1	27,8	83,3	0,0	0,0	44,4	50,0	16,7	0,0	27,8	16,7
ct=10	p	83,3	0,0	11,1	27,8	38,9	33,3	11,1	5,6	77,8	33,3	55,6	83,3
3dimensions	r	16,7	61,1	5,6	0,0	0,0	50,0	5,6	16,7	5,6	44,4	22,2	16,7
3dimensions	pd	0,0	0,0	0,0	0,0	61,1	0,0	38,9	33,3	5,6	22,2	5,6	0,0
3dimensions	pu	0,0	38,9	83,3	72,2	0,0	16,7	44,4	44,4	11,1	0,0	16,7	0,0
ct=100	p	100,0	0,0	0,0	0,0	27,8	11,1	0,0	22,2	61,1	11,1	50,0	50,0
3dimensions	r	0,0	88,9	0,0	11,1	0,0	50,0	5,6	0,0	33,3	50,0	0,0	0,0
3dimensions	pd	0,0	5,6	61,1	16,7	33,3	0,0	5,6	0,0	0,0	0,0	11,1	0,0
3dimensions	pu	0,0	5,6	38,9	72,2	38,9	38,9	88,9	77,8	5,6	38,9	38,9	50,0

		s1				s14				s142			
		p	r	pd	pu	p	r	pd	pu	p	r	pd	pu
ct=2	p	94,4	5,6	11,1	5,6	94,4	0,0	0,0	11,1	94,4	44,4	33,3	44,4
6dimensions	r	0,0	94,4	0,0	5,6	5,6	100,0	72,2	72,2	0,0	55,6	16,7	27,8
6dimensions	pd	0,0	0,0	38,9	0,0	0,0	0,0	0,0	0,0	5,6	0,0	50,0	22,2
6dimensions	pu	5,6	0,0	50,0	88,9	0,0	0,0	27,8	16,7	0,0	0,0	0,0	5,6
ct=5	p	94,4	0,0	11,1	0,0	77,8	0,0	0,0	0,0	11,1	0,0	5,6	0,0
6dimensions	r	0,0	88,9	11,1	33,3	11,1	100,0	0,0	22,2	5,6	55,6	16,7	0,0
6dimensions	pd	0,0	0,0	27,8	0,0	11,1	0,0	50,0	38,9	0,0	0,0	50,0	50,0
6dimensions	pu	5,6	11,1	50,0	66,7	0,0	0,0	50,0	38,9	83,3	44,4	27,8	50,0
ct=10	p	83,3	0,0	0,0	0,0	100,0	0,0	11,1	5,6	55,6	22,2	0,0	0,0
6dimensions	r	0,0	88,9	0,0	50,0	0,0	83,3	5,6	22,2	11,1	77,8	33,3	77,8
6dimensions	pd	0,0	0,0	11,1	0,0	0,0	0,0	38,9	0,0	0,0	0,0	44,4	16,7
6dimensions	pu	16,7	11,1	88,9	50,0	0,0	16,7	44,4	72,2	33,3	0,0	22,2	5,6
ct=100	p	100,0	0,0	0,0	0,0	72,2	0,0	0,0	0,0	83,3	0,0	0,0	0,0
6dimensions	r	0,0	100,0	0,0	16,7	0,0	55,6	0,0	0,0	11,1	77,8	11,1	0,0
6dimensions	pd	0,0	0,0	38,9	22,2	0,0	0,0	33,3	5,6	5,6	0,0	33,3	0,0
6dimensions	pu	0,0	0,0	61,1	61,1	27,8	44,4	66,7	94,4	0,0	22,2	55,6	100,0

Fig. 5. ST-Isomap results with training based on one person and testing it with another one. The results show how the dimension of the embedding and sensor number affect the classification result.

		s1				s14			
		p	r	pd	pu	p	r	pd	pu
ct=2	p	72,2	33,3	33,3	16,7	16,7	33,3	27,8	11,1
3dimensions	r	27,8	55,6	11,1	44,4	83,3	55,6	44,4	72,2
3dimensions	pd	0,0	0,0	50,0	11,1	0,0	5,6	11,1	0,0
3dimensions	pu	0,0	11,1	5,6	27,8	0,0	5,6	16,7	16,7
ct=5	p	77,8	5,6	22,2	33,3	61,1	50,0	27,8	50,0
3dimensions	r	0,0	77,8	5,6	16,7	0,0	11,1	5,6	5,6
3dimensions	pd	5,6	0,0	50,0	33,3	38,9	16,7	50,0	11,1
3dimensions	pu	16,7	16,7	22,2	16,7	0,0	22,2	16,7	33,3
ct=10	p	94,4	50,0	50,0	44,4	77,8	22,2	22,2	16,7
3dimensions	r	0,0	38,9	11,1	11,1	0,0	50,0	0,0	0,0
3dimensions	pd	0,0	0,0	33,3	27,8	11,1	22,2	55,6	44,4
3dimensions	pu	5,6	11,1	5,6	16,7	11,1	5,6	22,2	38,9
ct=100	p	77,8	16,7	16,7	38,9	77,8	11,1	61,1	50,0
3dimensions	r	16,7	66,7	11,1	11,1	5,6	55,6	33,3	16,7
3dimensions	pd	5,6	11,1	50,0	27,8	16,7	22,2	5,6	22,2
3dimensions	pu	0,0	5,6	22,2	22,2	0,0	11,1	0,0	11,1

		s1				s14			
		p	r	pd	pu	p	r	pd	pu
ct=2	p	100,0	22,2	16,7	50,0	66,7	22,2	16,7	11,1
6dimensions	r	0,0	77,8	0,0	0,0	16,7	61,1	5,6	27,8
6dimensions	pd	0,0	0,0	66,7	27,8	5,6	16,7	55,6	22,2
6dimensions	pu	0,0	0,0	16,7	22,2	11,1	0,0	22,2	38,9
ct=5	p	100,0	5,6	11,1	22,2	27,8	0,0	0,0	11,1
6dimensions	r	0,0	88,9	5,6	5,6	0,0	61,1	38,9	27,8
6dimensions	pd	0,0	0,0	72,2	27,8	50,0	5,6	44,4	11,1
6dimensions	pu	0,0	5,6	11,1	44,4	22,2	33,3	16,7	50,0
ct=10	p	88,9	11,1	16,7	22,2	55,6	0,0	0,0	5,6
6dimensions	r	0,0	77,8	5,6	11,1	0,0	44,4	0,0	0,0
6dimensions	pd	11,1	5,6	50,0	27,8	38,9	27,8	44,4	0,0
6dimensions	pu	0,0	5,6	27,8	38,9	5,6	27,8	55,6	94,4
ct=100	p	88,9	5,6	0,0	11,1	83,3	5,6	0,0	16,7
6dimensions	r	0,0	83,3	0,0	0,0	0,0	61,1	55,6	5,6
6dimensions	pd	0,0	5,6	61,1	33,3	16,7	16,7	5,6	22,2
6dimensions	pu	11,1	5,6	38,9	55,6	0,0	16,7	38,9	55,6

Fig. 6. ST-Isomap results with training based on 3 persons and testing it with another one. The results show how the dimension of the embedding and sensor number affect the classification result.

VI. Conclusion

We have presented a study on recognition of object manipulation actions: pick up, put down, rotate and push. The first contribution of the work is that training and testing are performed with 20 people where the manipulated object was placed on two different heights with people performing the actions multiple times at three different orientations. Most of the current systems that utilize robot imitation learning use a single person to train or teach tasks to the robot. Since the intention for the future is that robots will be able to learn from observing *different* and *multiple* people, we believe that it is important to study how different methods scale with respect to this.

In this work, we have concentrated on evaluation of dimensionality reduction using linear and nonlinear techniques. Although the techniques have been known for sometime, we have shown how the number of sensors, their placement and different modeling parameters affect the classification rate. PCA and nearest neighbor classification have been used frequently for action classification but our study shows that this techniques are not suitable for cases where the actions are very similar to each other. We believe that for recognition of such actions, dimensionality reduction has to be performed with care in order to preserve the true variance in the data. Even if the non-linear dimensionality reduction is more appropriate, the number of common and adjacent temporal neighbors have to be chosen carefully. The results also show that using the explicit knowledge of kinematic chains (arm model) may not be necessary in order to achieve satisfactory recognition rates. Finally, for most actions it is enough to provide only the measurements of the hand motions while distinguishing between *pick-up* and *put-down* would gain from including the knowledge of the object as well.

Acknowledgment

This work has been supported by EU through the project PACO-PLUS, FP6-2004-IST-4-27657. We also thank Odest Chadwicke Jenkins for the help with ST-Isomap.

References

[1] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching," in *IEEE Trans. on Robotics and Automation*, 10(6), 799-822, 1994.

[2] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[3] A. Billard, "Imitation: A review," *Handbook of brain thory and neural network, M. Arbib (ed.)*, pp. 566–569, 2002.

[4] K. Ogawara, S. Iba, H. Kimura, and K. Ikeuchi, "Recognition of human task by attention point analysis," in *IEEE/RSJ Int. Conf. on Intelligent Robot and Systems*, pp. 2121–2126, 2000.

[5] M. C. Lopes and J. S. Victor, "Visual transformations in gesture imitation: What you see is what you do," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2375– 2381, 2003.

[6] O. C. Jenkins and M. J. Mataric, "Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion," *International Journal of Humanoid Robotics*, vol. 1, pp. 237–288, Jun 2004.

[7] S. Ekvall and D. Kragic, "Grasp recognition for programming by demonstration tasks," in *IEEE Int. Conf. on Robotics and Automation*, pp. 748 – 753, 2005.

[8] S. Calinon, A. Billard, and F. Guenter, "Discriminative and adaptative imitation in uni-manual and bi-manual tasks," 54(5), 2005.

[9] D. Newton et al, "The objective basis of behavior unit," *Journal of Personality and Social Pshychology*, 35(12), 847-862, 1977.

[10] R. Duda, P. Hart, and D. Stork, *Pattern classification*. New York: Wiley-Interscience, 2001.

[11] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[12] I. S. Vicente, *Human action recognition based on linear and nonlinear dimensionality reduction using PCA and Isomap*. KTH, Stockholm, Sweden, http://cogvis.nada.kth.se/ danik/Isabel.pdf: Master thesis, 2006.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Second Edition. MIT Press and McGraw-Hill, 2001.

[14] T. Cox and M. Cox, *Multidimensional Scaling*. 2001.

[15] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *23rd National Conference ACM*, pp. 517–524, 1968.