Learning by Observation of Robotic Tasks using On-line PCA-based Eigen Behavior *

Xianhua Jiang
Department of Electrical and Computer Engineering
University of Vermont
Burlington, VT 05405, USA
xjiang@cem.uvm.edu

Yuichi Motai
Department of Electrical and Computer Engineering
University of Vermont
Burlington, VT 05405, USA
ymotai@cem.uvm.edu

Abstract - **This paper presents a new framework for learning the behavior of an articulated body. The motion capturing method has been developed mainly for analysis of human movement, but very rarely used to teach a robot human behavior in an on-line manner. In the traditional teaching method, robotic motion is captured and converted into the virtual world, and then analyzed by human interaction with a graphical user interface. However such a supervised learning framework is often unrealistic since many real-life applications may involve huge datasets in which exhaustive sample-labeling requires expensive human resources. Thus in our learning phase, we initially apply the supervised learning to just small instances using a traditional principal component analysis (PCA) in the off-line phase, and then we apply the incremental PCA learning technique in the on-line phase. Our on-line PCA method maintains the reconstruction accuracy, and can add numerous new training instances while keeping reasonable dimensions of the eigenspace. In comparison to other incremental on-line learning approaches, which use each static image, our proposed method is new since we consider image sequences as a single unit of sensory data. The extensions of these methodologies include the robotic imitation of human behavior at the semantic level. The experimental results using a humanoid robot, demonstrate the feasibility and merits of this new approach for robotic teaching.**

Index Terms - incremental learning, motion analysis, principal component analysis, behavior editor, learning by observation.

I. Introduction

In the robotic community, learning by observation is a demanding approach to teach the robotic behavior. In many situations, robotic teaching captured by a video camera is possible in terms of either humanoid or robotic behavior. The use of visual-based motion analysis is not new; it is rather standard in computational non-contact type measurement systems. For example, some 3D motion analysis systems have been already developed in various industrial applications, such as Vicon Motion Systems [1] A.P.A.S.[2], Motion Analysis [3], and [4, 5, 6] in academia. In this paper, we argue that visual behavior acquisition can be utilized at a higher level. Since certain behavior may follow a common pattern, we can use this information to build a common task. We will explain how these visual perceptions of the robot can be obtained and used together with "virtual" robotic description. A human operator initially needs to supervise the learning system by labeling each behavior using the proposed graphical user interface (GUI). This unique motion editing process eliminates several further problems in classifying the behavior.

Our contribution to the learning phase is to develop incremental learning methodologies, in which the training behavior is incrementally executed in an on-line manner. Although traditional PCA is conducted off-line, our on-line PCA sequentially updates the classification without human's inputs. In this proposed framework, we do not need to separate the procedures for learning and testing, but we can add the testing instances as an incremental instance. Incremental learning approaches using PCA are found recently in [7, 8, 9, 10, 11] for the same purposes. We are developing a new method since we are using a sequence of images as a single unit of sensory data for positioning land markers. The proposed system simultaneously updates the eigenspace by updating the scatter matrix in an on-line manner.

The structure of this paper is as follows: Section II describes related studies on learning by observation. Section III describes our proposed system modules and graphical user interface editors. In Section IV, our main contribution in this paper, Incremental On-line PCA, is illustrated thoroughly. Section V reports some interesting experiment results to demonstrate the performance of the whole system. Finally, concluding remarks are given in Section VI.

II. Related Works

The studies on learning by observation are very active, and have evolved in the following existing fields: 1) biomechanics, 2) computer graphics, and 3) robotics. Obviously, those fields have some overlap to some degree, and indeed our project described in this paper spans the above three fields.

In the area of 2) computer graphics, virtual robots are realized through avatars. In those studies, the captured motion data is used for directly determining the avatar behavior. Since they do not involve the actual mechanical structures of the robots, in the virtual world it seems easier to apply forward kinematics to the graphical model of the avatar based on the captured data, which is in the area of 1) biomechanics. It is, however, more complicated if the structure of the avatar is not identical to the real object, or if the new behavior must be synthesized [12]. For example, patterns of each joint are selected and optimized under physical constraints [13]. Another extension is [14], in

* This work is partially supported by Vermont Experimental Program to Stimulate Competitive Research Program (EPS-0236976) and URECA Program at the University of Vermont.

which the avatar can now be transferred from adult males/females to children by PD control. Those extensions among avatars are allowed to retarget if the physical models are clearly specified. To retarget the virtual avatar to the real object, many further problems develop: i.e. it becomes necessary to estimate modified behavior in the new/different conditions [15]. In [16], walking behavior is synthesized by interpolating intermediate managements. In the area of 3) robotic, artificial intelligence, and physiological approaches [3,18,19,20,21,22] have been examined. Especially in the computational vision and machine learning communities, the task-level learning need to be constructed in advance, based on the captured sensory data of the trajectories, as we propose in the following section.

III. Visual Observation System Through Motion Capture

We take several articulated mechanical parts of a robot and the corresponding actual parts of a human body into consideration. We wish to establish the following protocols: First, human movements are observed by a motion capturing device, such as a camera, a gyroscope, or a data glove. Then, using a uniquely developed graphical user interface, the motion is registered in the computer incrementally. The registered behavior is retargeted both to a virtual human and to an actual target in the real world. The final step is to transfer its motion to another robot. Currently we have developed the kinematical model corresponding to human figure. More specifically, our overall learning by observation is achieved by developing the following procedure; 1) Presentation: The target robot performs the desired tasks to the system from start to end without pausing. 2) Initialization: The human-in-the-loop module specifies the abstract class of targeted objects and edits higher structures such as a goal hierarchy and dependency among the behavior tasks. 3) Learning: The capturing system observes the performance and constructs a semantic-level description of the behavior/task. Recognition must be automatic and on-line in order to keep up with the task incrementally. 4) Recognition: For a given target environment, the system recognizes the initial configuration of the objects, modifies and instantiates the task description. 5) Manipulation: The program is executed to accomplish the task with a humanoid robot.

During the initial supervised learning (before the online incremental PCA), the computer still needs to acquire the label of the initial behavior. To label each behavior, we will describe a graphical user interface shown in Fig. 1. In the phase of behavior acquisitions, a wood-made mannequin and a humanoid robot Kondo KHR-1 are used as articulated objects. Each physical model, however, is not specified in detail. The following snapshots of the graphical user interface (GUI) editor will illustrate the registration process. Using the GUI of Fig.1 (a), a human operator specifies the number of the articulated parts. The humanoid structure is standardized at the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) as FCD 19774 Humanoid animation [17]. We apply a kinematical model to restrict each joint so that the behavior can follow the adjacency constraints or relative joint relationships.

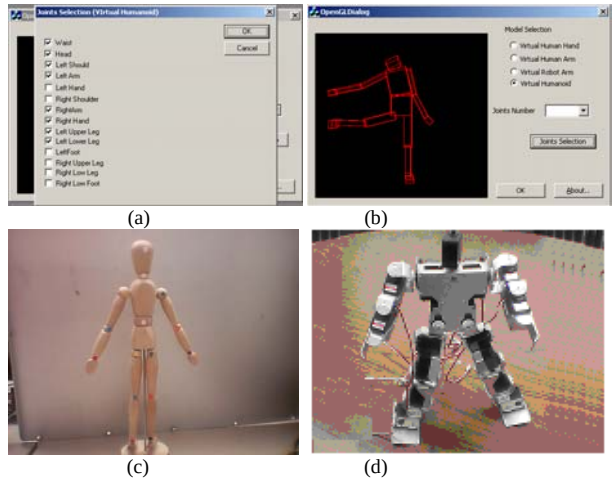

Fig. 1. Learning editor for a humanoid robot (a) (b). The motion of the articulated objects is captured using the camera sensor. The humanoid robot to be captured use either a mannequin (c) and Kondo KHR-1 (d).

VI. On-line Learning Using Incremental PCA

In order to limit the number of scenario of articulated behavior, the robot is only allowed to learn a few typical operations with representative postural tasks, such as picking up an object. We would like to create efficient supervised classifiers, hopefully with minimal supervision.

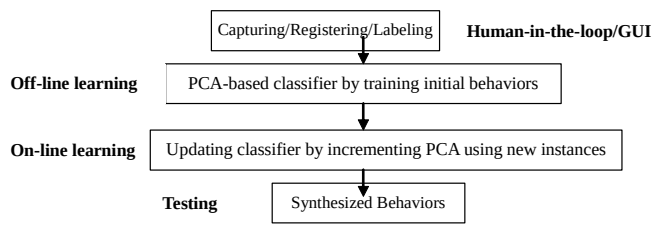

Fig.2. Overall learning procedure

As illustrated in Fig. 2, initial behavior instances are labeled for classifications by a human operator using the graphical user interface described in the previous section. This first step is called off-line learning, in which training behavior are classified using a standard PCA method. In the latter step we develop an on-line learning methodology using a structured learning entity. In the following subsections, we will explain the specific methodological procedure.

A. *Traditional Off-line PCA for Each Static Data*

In the traditional PCA, we use x_i to denote the *ith* sampled data (such as a captured image, output of dataglove, or gyroscope) in the form of a column $x_i \in R^{l\times 1}$, i=1…n, where l is the number of the data points in each data set, and n is the number of the data set or the number of images within one sequence (we will deal with sensory images as the acquired data). We reduce the dimensionality of the image by projecting the image to the k-dimensional eigen

space. Each image is approximated by $x_i^{'} = m + \sum_{j=1}^{k} a_{ij} e_j$, where superscript ' indicates that the measurement vector x_i is reduced by the eigen space, and m is the sample mean $m = \frac{1}{n}\sum_{i=1}^{n} x_i$. Eigenvectors e_j ($j=1\ldots k$) are computed by solving the scatter matrix $S = \sum_{i=1}^{n}(x_i - m)(x_i - m)^T$. We select the eigenvectors e_j corresponding to the k largest eigenvalues of the scatter matrix, that is $Se_j = \lambda_j e_j$. The scalar $a_{ij} = e_j^{\ T}(x_i - m)$ is found. It corresponds to the distance of any image x_i from the mean m along the e_j eigenvector. We use Jacob's method to solve the eigenvalues/vectors.

Thus, we get a least-squares solution by projecting the image x_i onto the subspace in the direction of e_j that passes through the sample mean. Each image is optimally approximated by taking into account the k most informative eigenvectors only.

B. *Time Sequential Data Representation*

In order to track the movement of the humanoid robot precisely and prevent interference from the environment, we put a color sticker on each joint of the humanoid robot. Therefore, here the data are only extracted from the marker points, instead of the entire set of 2D image pixels. These points are correlated to the geometrical model of the articulated object. Let us denote $x^l = (u^l, v^l)^T$ where l is the number of the marker position (u, v) in a 2D image.

For each kind of behavior, we will take a sequence of images "q" to totally classify "p" types of behavior. More specifically, we expand to reconstruct this measurement sequence by defining X_i as the i-th sequence: $X_i = \left[x^l{}_{i1} \quad x^l{}_{i2} \quad \ldots \quad x^l{}_{iq}\right]$, $i=1\ldots p$, where x^l_{ij} means the j-th image in the i-th behavior sequence. Thus we have a matrix X_i with dimensions of $2l$ rows by q columns. At first we calculate a $2l*q$ dimensional sample mean matrix $m = \frac{1}{n}\sum_{i=1}^{n} X_i$. As described in the above section A, using the eigenvectors e_j ($j=1\ldots k$), the p types of behavior X_i are decomposed by $X_i^{'} = m + Ea_i$, where $E= [e_1 \quad e_2 \ldots \quad e_k]$ and $i=1\ldots p$, by newly computing the eigenvectors e_j of the scatter matrix $S = \sum_{i=1}^{n}(X_i - m)(X_i - m)^T$. The scalar vector a_i is determined by $a_i = E^T(X_i - m)$.

Since we have p types of behavior and q images in one sequence, when p and q become large, we still require a lot of memory to store the data. By using PCA, we can reduce the dimensionality of each image, therefore reducing the amount of storage necessary.

C. *Incremental On-line PCA for Time Sequential Data*

The traditional PCA uses batch computation. That means the entire set of n training image sequences are needed to compute the knowledge representation. When a new image sequence has to be incorporated into the representation, we must discard the old representation and compute the $n+1$ image sequences to get the new representation. Therefore, in order to handle the new images during learning, all the original training image sequences must be stored. If the training images are very large, such a method will consume the storage of the robot. Instead, the use of incremental PCA to represent the training behavioral scenes allows the retention of only the most important features. In this way, we can discard the original image sequences once they have been used in updating. Since we only keep the reduced representation of the image sequences, the storage efficiency is increased.

We can assume we have obtained the subspace $E= [e_1 \; e_2 \ldots \; e_k]$ by a set of eigenvectors e_j, $j=1\ldots k$ from the training sensory data X_i, $i=1\ldots n$. The corresponding eigenvalues are λ_j, scalar vector $a_i = [a_{i1} \quad a_{i2} \quad \ldots \quad a_{ik}]$, $i=1\ldots n$, and the sample mean is m. Now, suppose a new image sequence X_{n+1} is found; we will update the knowledgebase to take into account this new image sequence.

First, we update the sample mean:

$$m' = \frac{1}{n+1}(nm + X_{n+1}) \tag{1}$$

We project the new image sequence to the old subspace E:

$$a_{n+1} = E^T(X_{n+1} - m) \tag{2}$$

The updated scatter matrix can be obtained through computations:

$$S' = S + \frac{n}{n+1}(X_{n+1} - m)(X_{n+1} - m)^T \tag{3}$$

In order to reflect the new data from image sequence X_{n+1}, we must update the eigenvectors. Let $X_{(n)}$' represents image sequences X_i in the old subspace E. Also $X_{i(n+1)}$' denotes the previous image sequences X_i and new image data X_{n+1} in the updated subspace E'. Then:

$$X^{'}{}_{i(n)} = Ea_{i(n)} + m \qquad i=1\ldots n, \tag{4}$$

In the new subspace:

$$X^{'}{}_{i(n+1)} = E^{'}a_{i(n+1)} + m^{'} \qquad i=1\ldots n+1 \tag{5}$$

We calculate the updated scalar value $a_{i(n+1)}$ by

$$a_{i(n+1)} = (E^{'})^T(X_i - m^{'}) \quad i=1\ldots n+1 \tag{6}$$

Please note that we do not store the previous image sequence X_i, $i=1\ldots n$, thus using Eq. (4), the new scalar vector Eq. (6) is represented by:

$$a_{i(n+1)} = \begin{cases} (E^{'})^T(Ea_{i(n)} + m - m^{'}) & i = 1\ldots n \\ (E^{'})^T(X_i - m^{'}) & i = n+1 \end{cases} \tag{7}$$

In order to reduce the size of the representation, we attempt to keep k dimensions, but only if it does not reduce

accuracy. These new k eigenvectors are sorted by decreasing order of the eigenvalues. We will define **Incremental On-line PCA** if the eigenspace k is expanded into $k+1$ when the new eigenspace is computed with new instance X_{n+1}. We will define **Non-incremental On-line PCA** if the eigenspace k keeps the same dimension when the new eigen space is computed.

We now introduce some criteria to judge when to expand the dimension of the eigenspace from k to $k+1$ in order to keep the balance between storage and accuracy. The two criteria are evaluated by comparing to thresholds. Criteria 1: The new sensory data of image sequences at $i=n+1$ can not be represented by the old subspace. Those occur when the error between the original image and the reduced image has exceeded the threshold. Criteria 2: If the new sensory data of image sequences can be represented by the old subspace, but the overall error of $n+1$ sensory data has exceeded the threshold, we need to expand the subspace. Therefore if the one of errors exceeds the threshold, the eigen space is extended, which means that the system applies Incremental On-line PCA.

V. EXPERIMENTAL ANALYSIS OF LEARNING BEHAVIOR THROUGH SEQUENTIAL DATA

We have applied the proposed GUI (described in Section III) and incremental PCA (described in Section IV) to the following two experiments using a mannequin. We measure the learning performance with quantitative accuracy/errors.

Experiment 1: Humanoid/Mannequin using Color Camera

In this experiment, we have used 11 color markers for all movable joints of a humanoid object shown in Fig. 3. A Pulnix CCD color camera was used for capturing the color markers. We chose 6 representative human movements, for partial behavioral sequential tasks. The task starts from a flat standing state, and completes the locomotion by one of the state chosen from M1 to M6 depicted in Fig.3.

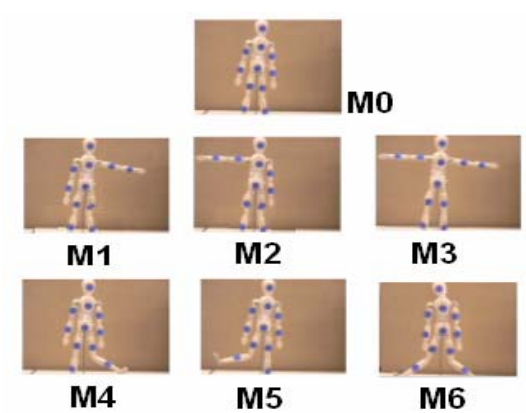

Fig. 3. Six training tasks of a human finger.

From the color image, the 2D position of each marker point $x^l{}_i = (u^l, v^l)^T$ is extracted, where u,v is a 2D point at the center of the *i-th* joint (i from 1 to 11). Thus, in the initial off-line PCA training space, the sensory image sequence X_i has the dimension of 22*16. Thus using the 6 behavior instances, the overall size of matrix $[X_i, X_2, ..X_n]$ is 22*96. The dimension of the corresponding scatter matrix S is 22*22, and the eigen behavior is at most 22. In the incremental on-line PCA training phase, we may add any additional sensory sequence X_i as described in Section V. The computation time for the incremental PCA is mainly for computing eigenvalues/vectors, which is less than the time for capturing the new image sequence.

Traditional off-line PCA was first conducted using the initial six training behavior, and then we applied the following five different methods to handle the new behavior:

1) New Training method <NewT> , which projects the new behavior to the old eigenvectors.

2) Non-incremental On-line method <NonON>, which updates the eigenvectors using the new behavior and old reconstructed behavior, and keeps the same eigenspace dimension.

3) Incremental On-line method <INON>, which updates the eigenvectors using the new behavior and old reconstructed behavior, and increase the dimension of the eigenspace.

4) Non-incremental Off-line PCA <NonOF>, which adds the new behavior to the original training behavior and updating the eigenvectors, but keeping the same eigenspace.

5) Incremental Off-line PCA <INOF> , which not only updates the eigenvectors using the new and old original behavior, but also expand the eigen dimension.

Table I shows the accuracy of each task by using five different methods. The training eigenspace is 12. The reconstruction ratios is evaluated for each Task in the traditional On-line PCA method by $1-\|X'_{i(n)} - X_i\| / \|X_i\| (= 1 - \|E\, a_{i(n)} + m - X_i\| / \|X_i\|)$, or in the our On-line (Non-) Incremental PCA by $1-\|X'_{i(n+1)} - X_i\| / \|X_i\| (= 1 - \|E a_{i(n+1)} + m' - X_i\| / \|X_i\|)$.

TABLE I
COMPARISION OF RECONSTRUCTION RATIO

ESP:12	Accuracy (%)				
	NewT	**NonON**	**INON**	NonOF	INOF
Task 1	97.44	97.63	97.63	98.51	98.51
Task 2	99.15	98.51	98.51	98.61	98.63
Task 3	97.67	97.90	97.89	98.90	98.90
Task 4	98.17	98.16	98.15	98.77	98.77
Task 5	97.59	97.23	97.21	97.94	97.95
Task 6	98.77	98.17	98.17	98.28	98.28
Task 7	95.98	98.56	98.57	98.56	98.57

Table I illustrated that our On-line PCA method still maintains the reconstruction accuracy like traditional off-line PCA. This result indicates that On-line PCA (both non-incremental and incremental) is very promising. It is obvious that Off-line PCA performs well but this method needs to keep all the original previous instances and new instances. The On-line PCA allows the learning system to discard the acquired measurement data immediately after the update. The error $\|X'_{i(n)} - X_i\|$ or $\|X'_{i(n+1)} - X_i\|$ is computed with respect to the dimension of eigenspace in

Fig. 4. The result of Table I is the specific case of the eigenspace at 12 in Fig. 4. Since the maximum dimension of our measurement space is 22, the error reaches zero at 22 in the eigenspace. Fig. 4 shows similar outcome of Table I. The absolute total error of On-line PCA worse than the Off-line PCA during eigenspace 8-22, but the difference between them is not large. The difference of absolute total error between Non-incremental and incremental is almost zero during eigenspace 8-22. But during 1-7, incremental method of both On-line and Off-line PCA performs better. Thus based on Fig.4, the incremental method is very effective especially in the low dimensional eigenspace. Again in the incremental framework, the result of both Table I and Fig. 4 confirmed that the proposed On-line method is very comparable to Off-line PCA.

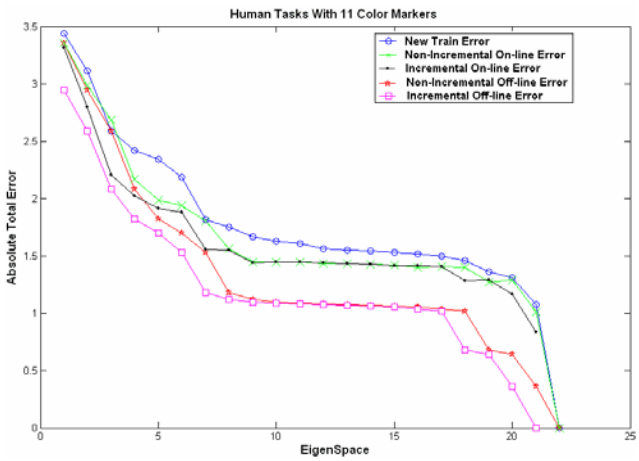

Fig. 4. Total error by changing the dimension of eigenspace among Off-line PCA, vs. On-line incremental PCA, vs. On-line non-incremental PCA.

Experiment 2: Classification Performance of automated on-line PCA training for Humanoid Behavior

Using two sub experiments, we evaluated the on-line PCA training performance for classifying unlabeled instances. In the first experiment, we applied On-line PCA to new instances distributed among M1-M6 (as shown in Fig. 3). The system automatically classified the new instances by measuring the distance in eigenspace between the existing classes M1 through M6. Off-line training data in one class had a normal distribution, therefore we calculated the expected value $u = \frac{1}{K}\sum_{j=1}^{K}(E'a'_j + m')$, where K was the amount of training data in one specific class. The standard deviation was $\sigma = \sqrt{\frac{1}{n-1}\sum_{j=1}^{K}(E'a'_j + m' - u)^2}$. If the new data was within the circle of $k\sigma$, we regarded this new data as belonging to this class. We adjusted k to examine the change in performance of the classifier (using values of k ranging from 0.5-1.0 in increments of .1 and from 1.25-3.0 in increments of .25). Four types of classifiers were used – TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). The Positive/Negative designations indicated whether the new instance was predicted to belong to the specified class, and True/False specified whether that prediction was correct. For example, a TN classifier would signify that the new instance was correctly predicted as not belonging to the class. We defined Accuracy as TP/(TP+FN), and the False Positive Rate to be FP/(FP+TN). Fig. 5 shows the receiver operation curve as an outcome of classification performance. The proportion of "true" classifications is shown as the false positive rate is relaxed through successive increases of the value k. The eigenspace was kept at 12. Fig. 5 demonstrates that the new unlabeled training instance was automatically classified with a high level of accuracy at a low rate of false positives.

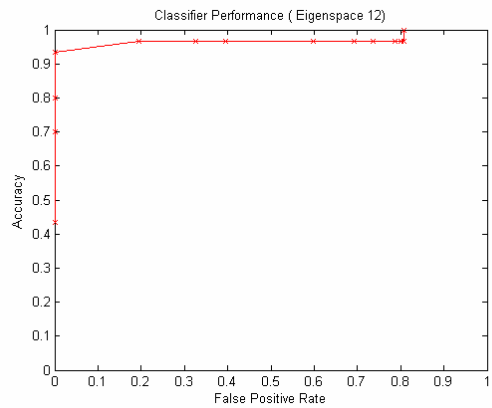

Fig. 5. The receiver operation curve for classifying existing trained behavior. The classification accuracy was evaluated by changing the value k.

In the second experiment, we applied new instances among M7-M12, which were not used for training before. That is, when a new instance x_{n+1} (not belong to any existing training class) was revived, the system regarded this new instance as belonging to a new class C_{p+1}. To this new class, we set its initial expected value $u_{p+1} = x_{n+1}$, and the initial $\sigma_{p+1} = 1.5 * \sqrt{\frac{1}{p}\sum_{i=1}^{p}\sigma_i^2}$. When the other new instances were in the $k\sigma_{p+1}$ range of the new class C_{p+1}, the On-line PCA updated the expected value and standard deviation, $u_{p+1} = \frac{1}{q+1}(q * u_{p+1} + x_{n+1})$, where q was the previous amount of data in the class C_{p+1}. The standard deviation $\sigma_{p+1} = \frac{M_q - q - 1}{M_q}\sigma_{p+1} + \frac{q+1}{M_q}\sqrt{\frac{1}{q}\sum_{i=1}^{q+1}(x'_i - u_{p+1})^2}$, where M_q was the constant value that presented the satisfying data amount in the new class. First, we set the initial value of M_q as 150, recorded the accuracy and false positive rate when increasing the threshold value k from 0.5 to 3.0 as shown in Fig. 6 (a). Then, we set the threshold value k at a constant value 1.0, recorded the accuracy and false positive rate when increasing M_q from 50 to 300 as shown in Fig. 6 (b). At last, we set M_q and k at constant values (M_q=150, k=1.0), using Non-incremental PCA and Incremental PCA to test the classifier performance, we found that the accuracy and false positive rate were the same in these two methods.

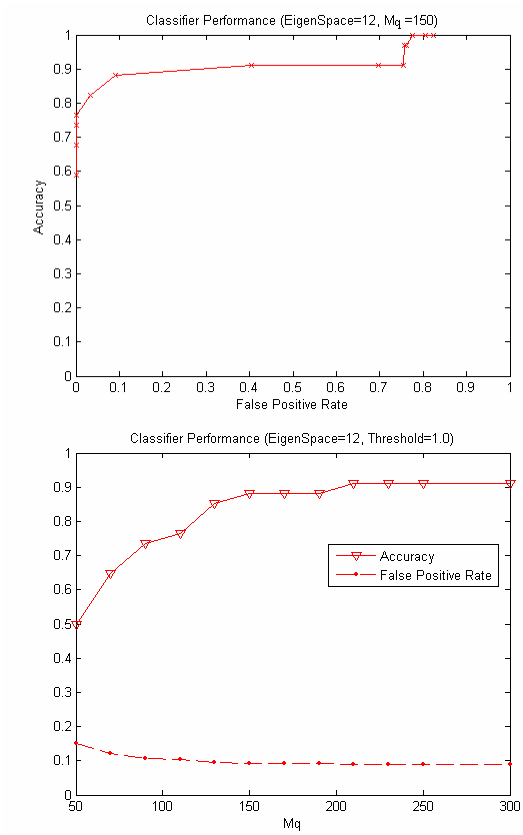

Fig. 6. The receiver operation curve for classifying new untrained behavior. The classified accuracy was evaluated (a) by changing the value *k* for False Positive Rate, and (b) by changing the value M_q .

Fig. 6 (a) shows good performance of the receiver operation curve as an outcome of On-line PCA training, in which the unlabelled new instance was classified to the new class as the system relaxed the false positive rate by increasing the threshold values. Fig. 6 (b) shows that if M_q was increased until 200, the classification accuracy reached to 0.9. Therefore our proposed method for unlabeled instances was well classified if the system chose appropriately large M_q.

VI. Conclusion

In this paper, we have applied an on-line learning method for training articulated behavior. Our on-line learning method needs no human supervision, since the system automatically expands initial classifiers using on-line PCA. Typical or traditional PCA has been mainly used in an off-line manner. Since video-based motions are our measurement data, the new instances arrive as a continuous stream that is too expensive to store. Thus we believe that the major factor in the design of the incremental learning system is the availability and expandability of the initial behaviors. We applied the new on-line method for a miniature human figure. The experiment results demonstrated the feasibility and merits of reducing learning dimensionalities using our on-line incremental PCA. In contract to the Off-line PCA method, which needs to keep all the original previous instances and new instances, our proposed On-line method does not need to keep the previous instances, but just keeps the eigenspace and reconstruct the space using the new instance. Our on-line PCA can add many new training instances while maintaining reasonable dimensions of the eigenspace. As a remaining issue, the on-line classification performance using the incremental PCA described in this paper needs to be evaluated through the long-term training phase.

Acknowledgment

The authors acknowledge J. B. Zurn and K. Gorman for their proof reading.

References

[1] Vicon Motion Systems, Oxford Metrics Ltd. http://www.vicon.com [On-line].
[2] A.P.A.S, Ariel Dynamics, http://www.arielnet.com/ [On-line].
[3] Motion Analysis Inc, http://www.motionanalysis.com/ [On-line]
[4] C. Stauffer and W.E.L. Grimson, Learning patterns of activity using real-time tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, Issue 8, pp. 747 - 757, 2000.
[5] C.R.Wren,.A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19 , Issue 7, pp. 780 - 785, 1997.
[6] I. Haritaoglu, D. Harwood, L.S. Davis, W4: real-time surveillance of people and their activities, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 22, Issue 8, pp. 809 - 830, 2000.
[7] Y. Li, L. Q. Xu,J. Morphett, R. Jacobs, An integrated algorithm of incremental and robust PCA, Proceedings of International Conference on Image Processing,, pp. 245-248, 2003.
[8] M. Artac, M. Jogan, A. Leonardis, Incremental PCA for on-line visual learning and recognition, Proceedings of International Conference on Pattern Recognition, pp. 781 - 784, 2002.
[9] M. Artac, M. Jogan, A. Leonardis, Mobile robot localization using an incremental eigenspace model, Proceedings of IEEE International Conference on Robotics and Automation, pp. 1025 - 1030, 2002.
[10] P. Hall, D. Marshall, and R. Martin, Incremental eigenanalysis for classification, Proceedings of British Machine Vision Conference, pp. 286-295, 1998.
[11] S. L. Nayar and S. A. Nene, and H. Murase, Subspace methods for robot vision, IEEE Transactions on Robotics and Automation, Vol. 12-5, pp. 750 - 758, 1996.
[12] Z. Popovic and A. Witkin, Physically Based Motion Transformation, SIGGRAPH 99, pp. 11-20, 1999.
[13] J. K. Hodgins and N.S. Pollard, Adapting Simulated Behavior For New Characters, SIGGRAPH97, pp.153-162, 1997.
[14] A. Bruderlin and L. Williams, Motion Signal Processing, SIGGRAPH95, pp.97-104, 1995.
[15] A. Witkin and Z. Popovic, Motion Warping, SIGGRAPH95, pp. 1050-108, 1995.
[16] H.C. Sun and D.N. Metaxas, Automating gait generation, SIGGRAPH2001, pp. 261-269, 2001.
[17] http://www.h-anim.net/ [On-line].
[18] A. Fod, M. Mataric, and O.C. Jenkins, Automated Derivation of Primitives for Movement Classification, Autonomous Robots, 12 (1), pp. 39-54, 2002.
[19] O.C. Jenkins, M.J. Mataric, and S. Weber, Primitive-Based Movement Classification for Humanoid Imitation, Proceedings of International Conference of Humanoid Robotics, 2000.
[20] P. Gaussier, S. Moga, J..P. Banquet, M. Quoy, N. CREARE, From perception-action loops to imitation processes: A bottom-up approach of learning by imitation, International Journal of Applied Artificial Intelligence, Vol. 12, pp. 701-729, 1998.
[21] S Schaal, Is imitation learning the route to humanoid robots, Trends in Cognitive Sciences, Vol.3, No.6, pp.233-241, 1999.
[22] J Demiris, G Hayes, Imitative learning mechanisms in robots and humans, Proceedings of the 5th European Workshop on Learning Robots, pp.9-16, 1996.