Incremental online PCA for automatic motion learning of eigen behaviour

Xianhua Jiang and Yuichi Motai*

Department of Electrical and Computer Engineering,
University of Vermont, Burlington, VT 05405, USA
E-mail: xjiang@cems.uvm.edu E-mail: ymotai@uvm.edu
*Corresponding author

Abstract: This paper presents an online learning framework for the behavior of an articulated body by capturing its motion using real-time video. In our proposed framework, supervised learning is first utilised during an offline learning phase for small instances using principal component analysis (PCA); then we apply a new incremental PCA technique during an online learning phase. Rather than storing all the previous instances, our online method just keeps the eigenspace and reconstructs the space using only the new instance. We can add numerical new training instances while maintaining the reasonable dimensions. The experimental results demonstrate the feasibility and merits.

Keywords: incremental learning; motion analysis; principal component analysis; PCA; behaviour editor; learning by imitation.

Reference to this paper should be made as follows: Jiang, X. and Motai, Y. (200x) 'Incremental online PCA for automatic motion learning of eigen behaviour', *Int. J. Intelligent Systems Technologies and Applications*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: X. Jiang obtained a BS in Control Science and Engineering from Zhejiang University, China in 2001, a Master of Automation from Zhejiang University in 2004. She is currently pursuing Doctor Degree in Electrical and Computer Engineering at the University of Vermont. Her research areas include artificial intelligence, computer vision, robotics, and signal processing.

Y. Motai obtained a Bachelor of Engineering Degree in Instrumentation Engineering from Keio University in 1991, a Master of Engineering Degree in Applied Systems Science from Kyoto University in 1993, and a PhD Degree in Electrical and Computer Engineering from Purdue University in 2002. He is currently an Assistant Professor in Electrical and Computer Engineering Department at the University of Vermont. His research interests are computer vision, human-computer interaction, ubiquitous computing, sensor-based robotics, and mixed reality.

1 Introduction

Automatic learning is a demanding approach to vision-based real-time interfaces that is facilitated by observing the motions and/or behaviour of tracked objects. The target application is sometime called learning by imitation. In visual-based interface communities, automated training methodology for articulated behaviour is very important. Offline based training behaviour using a video camera is possible, and the use of visual-based motion analysis is now standard in non-contact type measurement systems. For example, some existing 3D motion analysis systems have been already developed in various industrial applications, such as Vicon Motion Systems (http://www.vicon.com/), A.P.A.S. (http://www.arielnet.com/), Motion Analysis (http://www.motionanalysis.com/), and (http://www.is.aist.go.jp/humanoid/openhrp/; Wren et al., 1997; Haritaoglu et al., 2000) in academia.

In this paper, we argue that visual behaviour acquisition can be extended from the existing static image-based learning methodologies to video-based motion sequences. In the learning phase for training behaviour, we would like to implement an online learning framework, in which observable instances are incremented. When the degree of behavioural freedom is increased, the feature space tends to explode in a long-term learning phase. Our new method prevents this from happening. We use PCA to classify the training behaviours. Certain behaviours may follow a common pattern, and we can use information from this kinematical motion to build a common or principal task, which we call eigen behaviour. We will explain how the principal components of articulated motions can be obtained and used together with classification.

The organisation of this paper is as follows: Section 2 describes our proposed system. In Section 3, our main contribution, incremental online PCA, is illustrated thoroughly. Section 4 contains some interesting experiment results to demonstrate the performance. Concluding remarks are given in Section 5.

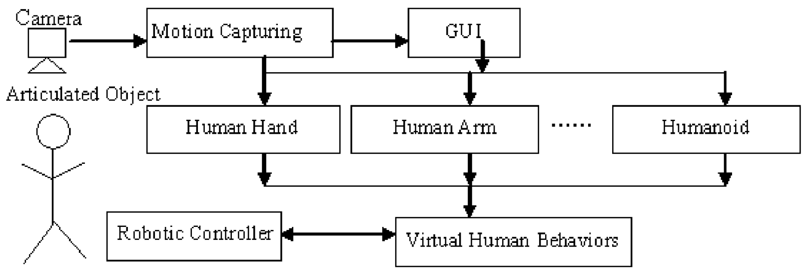
2 Visual observation system through motion capture

We take several articulated parts of a humanoid and the corresponding actual parts of a human body into consideration. Currently we have developed the following four kinematical models to be used to correspond to articulated target objects:

- human hand
- human arm
- robotic arm
- humanoid robot.

Figure 1 shows a flow diagram of the developed modules. We select these objects as targets since we wish to establish efficient and effective interaction among the actual target objects, the virtual avatars, and eventually retarget both the virtual and actual targets. More specifically, we wish to establish the following protocols: First, human movements are observed by a motion capturing device using a camera. Then, using a uniquely developed Graphical User Interface (GUI), the motion is registered in the computer by a human-in-the-loop. The registered behaviour is retargeted to a virtual human from an actual object in the real world. The final step is to train its motions in the learning phase.

Figure 1 Modules for learning by visual observation



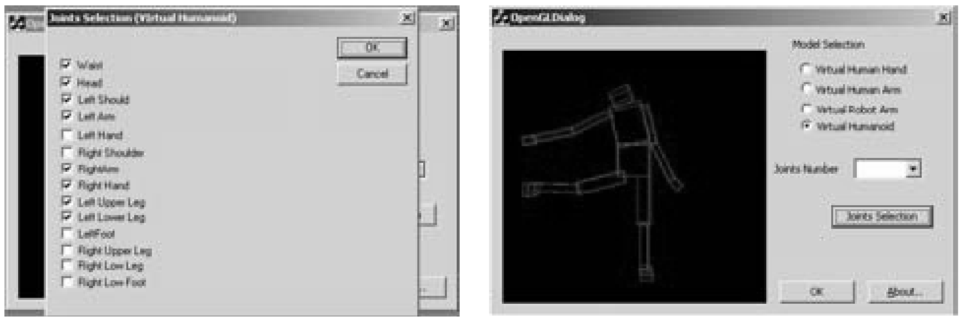
In this proposed scenarios, rather than registering the limited joint descriptions, the system learns the principal behaviour of each articulated object part. Moreover, determining optimal values of motion parameters such as trajectory and velocity of the manipulator depend on the configuration of the workspace and the structure of the manipulator. Task plans described in terms of more abstract and qualitative representation of assembly operations such as pick and place are of high reusability. However those descriptions required a labelling procedure in the learning phase. The exhaustive sample-labelling that is needed makes expensive human resources necessary and is often unrealistic. For example, in visual-based automatic gesture training, labelling data is a time consuming and difficult task. For other applications, labelling data may require very expensive tests so that only a small set of labelled data may be available. In some domains, only a few positive samples are available while unlabelled examples are plentiful. In all these cases, we need to find ways to relieve the users from the annotation burden. Therefore we develop here a new framework of training behaviour using incremental learning methodologies, in which the training behaviours are incrementally executed in the online manner. Compared to other approaches of incremental learning using each static image (Li et al., 2003; Artac et al., 2002a, 2002b; Hall et al., 1998; Nayar et al., 1996), our proposed method is new since we consider image sequences as one unit of sensory data for the positions of markers. Thus our main contribution in this paper is that we have developed PCA to fit video sequences by incrementally updating eigen behaviour in an online manner. The system automatically reconstructs the eigenspace by updating a scatter matrix with the new instances. Another contribution of the system that we have developed is a unique registration process of the labelling for behaviour learning by initial inputs from the human-in-the-loop (the details of which are beyond the scope of this paper). During the initial PCA supervised learning (before the online incremental PCA), the computer still needs to acquire the label of the initial behaviours. To label each behaviour, we will describe a GUI developed as follows:

2.1 Behavioural graphical interface editor

In the learning phase of behaviour acquisitions, humanoid robot Kondo KHR-1 and a wood-made mannequin are used for articulated objects. Each physical model, however, is not specified in detail. The number of joints is assigned by choosing from the menu in the learning editor. The following snapshots of the GUI editor will illustrate the registration process.

Using the GUI of Figure 2(a), a human operator specifies the number of the articulated parts, which highlights the landmarks. The humanoid structure is standardised at the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) as FCD 19774 Humanoid animation (http://www.h-anim.net/). In this format, the joints of the humanoid are defined as a tree structure. We apply a kinematical model to restrict each joint so that the system can register the behaviour to follow the adjacency constraints or relative joint relationships.

Figure 2 (a) Learning editor for a humanoid robot and (b) The motion of the articulated objects is captured using the camera sensor



3 Online learning using incremental PCA

In our proposed learning system, we apply several unique characteristics for capturing behaviours of articulated robotic movements. In order to limit the number of scenarios of articulated behaviours, the robot is only allowed to learn a few typical operations with representative postural tasks. This provides examples for considering a small number of sequences of the articulated behaviours. We would like to create efficient classifiers, hopefully with minimal supervision. Initial behaviour instances shown in Figure 3 (typically 50 behaviour instances), are labelled for classifications by a human operator. This first step is called offline learning, in which training behaviours are classified using a standard PCA method. In the latter step we develop an online learning methodology using unlabelled instances. Thus, in our overall learning frame, we initially apply supervised learning for small instances using traditional PCA (Section 3.1), and then we apply a new framework to PCA using an incremental learning technique (Section 3.3). In the following subsections, we will explain the specific methodological procedures.

3.1 Traditional off-line PCA for each static data

For the traditional PCA, we represent x_i as all sensory data at each sampling frame i in the form of a column of vectors $x_i \in R^{l \times 1}$, $i = 1 \ldots n$, where l is the number of data points in each data set, and n is the number of the data sequence or the number of images. We reduce the dimensionality of the image by projecting the image to the k-dimension 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 eigenspace, 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 of $Se_j = \lambda_j e_j$. 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 eigen values/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 to the degree of error by taking into account the k most informative eigenvectors only.

3.2 Time sequential data representation

The representation of the learned behaviours plays a very important role in the learning algorithm. Our learning behaviours can be classified into two type of structures:

- semantic labelled behaviour (such as raise right hand etc.)
- time sequential instances (i.e., the progression of movement through the beginning middle, and end of a task sequence).

Although we prepare the training instance by setting the beginning and ending of behaviour sequences, each sequence is analysed using the entire set of image frames from a camera sensor. We extend the data frame into the following way.

- Rather than using the entire set of 2D image pixels with an eigenimage approach (Nayar et al., 1996), here the data are only extracted from the marker points. These points are correlated to the geometrical model of the articulated object. Let us define $x^l = (u^l, v^l)^T$ where l is the number of markers with position (u, v) in a 2D image. In order to track the movement of the humanoid robot precisely and prevent interference from the environment, we put colour stickers on the each joint of the humanoid robot. For example, in our humanoid experiment we have a total of 11 colour stickers; each colour sticker has a vertical and horizontal position. Therefore, the dimensionality of each image has been reduced from 256 × 256 to 22 × 1.
- In order to completely classify 'p' types of behaviours, we take 'q' images as a sequence for each kind of behaviour. More specifically, we expand to reconstruct this measurement sequence by defining X_i to denote the ith sequence:

  $$X_i = (x^l{}_{i1} \quad x^l{}_{i2} \quad \ldots \quad x^l{}_{iq}), \quad i = 1 \ldots p,$$

where x_{ij} means the jth image in the ith behaviour sequence. Thus we have a matrix X_i with *dimension* of $2l$ row by q columns. At first we calculate a $2l \times q$ dimensional sample mean m by newly computing

$$m = \frac{1}{n}\sum_{i=1}^{n} X_i .$$

As described in the above, using the eigenvectors $e_j (j = 1 \ldots k)$, the p types of behaviours X_i are decomposed by $X_i' = m + Ea_i$, where $E = [e_1 \quad e_2 \quad \ldots \quad e_k]$ and $i = 1 \ldots p$. Eigen space E is computed from the scatter matrix S of image sequences X_i by defining

$$S = \sum_{i=1}^{n} (X_i - m)(X_i - m)^T .$$

The scalar vector a_i is determined by $a_i = E^T(X_i - m)$.

In this way, we consider time sequences as a single unit of sensory data for positioning landmarks. Since we have p types of behaviours and q images in a sequence for each behaviour, when p and q become large, and we still require a lot of memory to store the data at the long-term training phase. By using PCA, we can reduce the dimensionality of each image, therefore reducing the amount of storage necessary.

3.3 *Incremental online PCA for time sequential data*

In this phase, we train each behaviour using extracted sensory data from an image sequence described above. When a new sequence is demonstrated, the system will update the eigenvectors and find the closest behaviour which best represent this new sequence by extracting eigen behaviour of articulated motions. We would then like to apply a new incremental PCA into the robotic behaviour classification.

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 in the traditional PCA. If the size of the training images is very large because of time sequence, such a method will consume the storage of the system. Instead, the use of incremental PCA to represent the training behavioural scenes allows the retention of only the most important features. We can update the knowledge by combining the old representation of training and the new image sequences. 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 that we have obtained a set of eigenvectors e_j, $j = 1 \ldots k$ from the training sensory data X_i, $i = 1 \ldots n$. The eigenspace $E = [e_1 \quad e_2 \quad \ldots \quad e_k]$. The corresponding eigenvalues are λ_j, $j = 1 \ldots k$, scalar matrix $a_i = [a_{i1}\ a_{i2} \ldots a_{ik}]$, $i = 1, \ldots n$, 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:

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

In order to reflect the new data from image sequence X_{n+1}, we must update the eigenvectors by solving $S'e_i' = \lambda_i' e_i'$. The updating process can be summarised as follows. When a new image sequence X_{n+1} is received, we compute the new sample mean m' and scalar matrix a_{n+1} on the old subspace E, then construct an updated subspace E', but without image representation. Let us use $X'_{i(n+1)}$ to represent image sequences X_i in the old subspace of E and superscript $'$ indicates that the image sequences are reduced by the eigenvectors. Also $X'_{i(n+1)}$ represents the previous image sequences X_i and the new image sequence X_{n+1} in the updated subspace E'. Then:

$$X'_{i(n)} = Ea_{i(n)} + m \quad i = 1 \ldots n. \tag{4}$$

In the new subspace:

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

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 equation (4), the new scalar matrix equation (6) is represented by:

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

After updating the knowledgebase, we need to represent all sensory data of image sequences again in the new subspace, so that we do not need to keep the original image sensory data in memory. Since we only store the reduced sensory data of image sequences, it is important to update these images and also keep the approximations provided by equation (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 Online PCA* if the eigenspace k is expanded into $k + 1$ when the new instance X_{n+1} comes. We will define *Non-incremental online PCA* if the eigenspace k keep the same dimension k when the new eigenspace is computed when considering the new instance X_{n+1}.

We now introduce some criteria to judge when to expand the dimension of eigen subspace from k to $k + 1$ in order to maintain the balance between storage and accuracy. The two criteria are evaluated to determine whether the eigenspace needs to be extended or not.

Criteria 1: The new sensory data of image sequences at $i = n + 1$ can not be represented by the old eigenspace satisfactorily. This occurs when the difference between the original image and the reduced image $\| X_{i(n+1)} - X'_{i(n)} \|$ 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 of the images $\sum_{i=1}^{n+1} \| X_{i(n)} - X'_{i(n)} \|$ has exceeded the threshold, we need to span the subspace. Therefore, if one of thresholds is exceeded, the eigenspace is extended, and the system applies incremental online PCA.

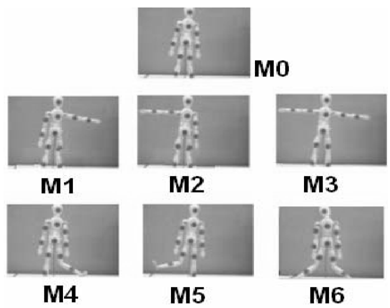
4 Experimental analysis of learning behaviours through sequential data

We have applied the proposed incremental PCA to the following three experiments using a mannequin. We measured the learning performance with quantitative accuracy/errors by changing the dimensions of the eigenspace (Experiment 1) and by changing the number of training instances for online PCA (Experiment 2). We also evaluated classification performance of unlabelled instances for the proposed online PCA (Experiment 3).

4.1 Experiment 1: dimension analysis of the eigenspace representations for humanoid/mannequin motions

In this experiment, we have used 11 colour-point markers for all movable joints of a humanoid object shown in Figure 3. A Pulnix CCD colour camera was used for capturing the colour markers. We chose six representative human figure positions, for partial behavioural sequential tasks. The task started from a flat standing state M0 and completed the locomotion to a new state chosen from six figure structures described in Figure 3.

Figure 3 Six behavioural training tasks of a human figure using a camera sensor. New behavioural tasks are added to the training sequence for incremental learning



These behaviours are described as

M1: raise the right hand
M2: raise the left hand
M3: raise both hands
M4: raise the right leg
M5: raise the left leg
M6: raise both leg.

Each behaviour was captured during 4 seconds for at least 16 frames. From the colour image, the 2D position of each marker point $x_i^l = (u^l, v^l)^T$ is extracted, where *u*, *v* is the 2D centre position of each figure joint from $i = 1$ to 11. Thus, in the initial offline PCA training phase, the sensory image sequence X_i has the dimension of 22 for the row vectors, and column dimensions of at least 16. Thus using the 6 behaviour instances, the size of each X_i is 22×16, and the overall size of matrix $[X_1, X_2, \ldots, X_n]$ is 22×96. The dimension of the corresponding scatter matrix *S* is 22×22, and the eigen behaviour is at most 22, based on the matrix formation. In the incremental online PCA training phase, we may add any additional sensory sequence X_i as described in Section 3. The computation time for the incremental PCA is mainly for computing eigenvalues/vectors, which is less than the time for sequentially capturing the new image data frame.

Traditional offline PCA was first conducted using the initial $6 \times 5 = 30$ training behaviours (6 training behaviours, and each behaviour was repeated by 5 times). We then applied the following five different PCA methods to handle the new behaviours by considering the 31*th* training instance:

- the New Training method <NewT>, which involves projecting the new behaviour to the old eigen subspace
- the Non-incremental online method <NonON>, which updates the eigenvectors using the new behaviour and old reconstructed behaviours, and keeps the same eigenspace dimensions
- the Incremental online method <INON>, which involves the eigenvectors using the new behaviour and old reconstructed behaviours, and spans the eigenspace by incrementing the eigen dimension
- the Non-incremental Off-line method <NonOF>, which means adding the new behaviour to the original training behaviours and updating the eigenvectors, but keeping the same dimension of eigenspace
- the Incremental Off-line method <INOF>, which not only updates the eigenvectors using the new and old original behaviours, but also span the eigenspace by incrementing the eigen dimension.

Table 1 shows the accuracy of each Task 1–7 when the training instances were represented by the eigenspace using five different methods discussed. The eigenspace dimension was 12.

Table 1 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

In the result of Table 1, Task 1–7 were reconstructed first. The fitting measure, which we call the *reconstruction ratio*, was evaluated by comparing the measured distance between the raw measurement value X_i and the reconstructed value $X'_{i(n)}$ by equation (4) using traditional online PCA at *n*th. Also when the new data is coming, the fitting measure was evaluated using $X'_{i(n+1)}$ by equation (5) of online PCA at n + 1th sequence with either incremental or non-incremental way. That means, the reconstruction ratio is evaluated for each Task in the non-incremental online and non-incremental Offline PCA (NonON and NonOF) by $1-\| X'_{i(n)} - X_n \| / \| X_n \| (= 1 - \| Ea_{i(n)} + m - X_n \| / \| X_n \|),$ or in our online and Offline incremental PCA (INON and INOF) with additional instance by

$$1-\| X'_{i(n+1)} - X_n \| / \| X_n \| (= 1 - \| E' a_{i(n+1)} + m' - X_n \| / \| X_n \|).$$

For example in the online PCA, we setup a threshold for the criteria to determine whether eigenspace with the dimension of 12 is expanded into 13 or not; that is, a non-incremental PCA (NonON) or incremental PCA (INON).

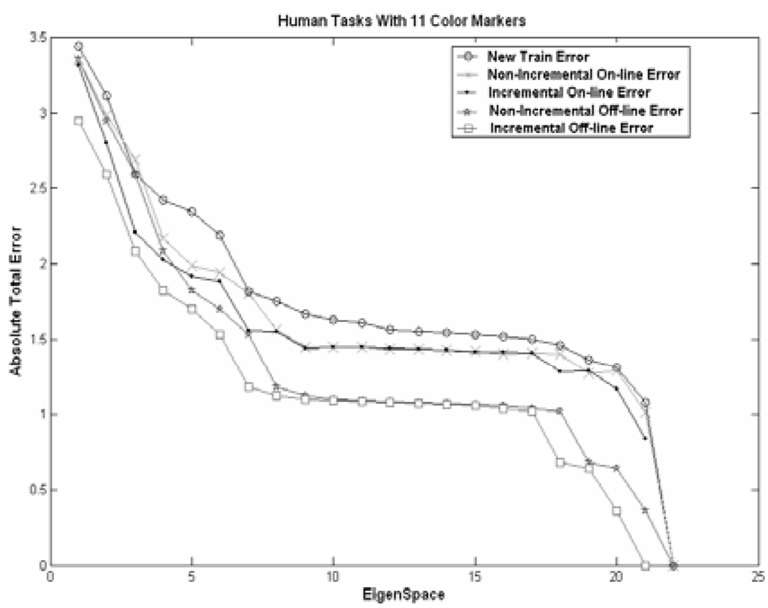
Table 1 illustrated that our online PCA method still maintains the reconstruction accuracy like traditional Offline PCA. This result indicates that online PCA (both non-incremental and incremental) is very promising. It is obvious that Offline PCA performs well, but this method needs to keep all of the original previous instances and new instances, in order to reconstruct the eigenspace using all the instances. Our proposed Online method does not keep the previous instances, but only retains the eigenspace and reconstructs the space using just the new instance and the reduced old instances. The Online PCA allows the learning system to discard the acquired measurement data immediately after the update. Since video-base motions are our measurement data, the eigenspace for the new instances arrives as a continuous stream and therefore is too expensive to store completely.

If the dimension of eigenvalues is changed, the total error of

$$\sum_{i=1}^{n+1} \| X'_{i(n)} - X_i \| \text{ or } \sum_{i=1}^{n+1} \| X'_{i(n+1)} - X_i \|$$

is computed with respected to the number of eigenvalues in Figure 4. When the error is small, the reconstructed one is close to the instance, which means the eigenspace spans well to represent the sample instance.

Figure 4 Absolute total error by changing the dimension of eigenspace for reconstruction; comparison among Offline incremental PCA, vs. Offline non-incremental PCA, vs. online incremental PCA, vs. online non-incremental PCA, and New Train method



As seen in Figure 4, by changing the dimension of eigenspace, we compared the errors of all five PCA methods. The result of Table 1 is the specific case of the eigenspace at 12 in Figure 4, which means that Table 1 was only evaluated at the largest 12 eigenvalues for reconstructions. Since the dimension of our measurement space is 22, the error reaches zero at 22 in the eigenspace. Figure 4 shows similar outcome of Table 1; the absolute total error of online PCA was larger than Offline PCA during eigenspace 8–22, but the difference between them was not large. The difference of absolute total error between Non-incremental and incremental online PCA is almost zero during eigenspace 8–22. But during 1–8, incremental method of both online and Offline PCA performs better; the total error is reduced by the incremental process. Thus based on the result of Figure 4, especially in the low dimensional eigenspace, the incremental method is very effective. Again in the incremental framework, the result of both Table 1 and Figure 4 confirmed that the proposed online method is very comparable to the Offline PCA.

4.2 Experiment 2: accuracy performance evaluation of incremental or non-incremental online PCA

In this experiment, we evaluated the performance of online incremental PCA by adding many new training instances using the online method. The old training tasks are the same as Experiment I, but new training behaviour is increased. The new tasks are combined tasks of Figure 3, starting from M0 in Figure 3 and ending at the one of following state:

M7: raise the left hand and left leg
M8: raise the right hand and right leg
M9: move the head to the right side
M10: raise the right leg and left hand
M11: raise the left leg and right hand
M12: raise the left hand and both leg.

Each sequential task was captured 5 times, so we have 6×5 new instances in addition to previous 30 instances. The rest of the sensory settings is all the same as Experiment I. We measure the total error

$$\sum_{i=1}^{n+1} \| X_i - X'_{i(n+1)} \|$$

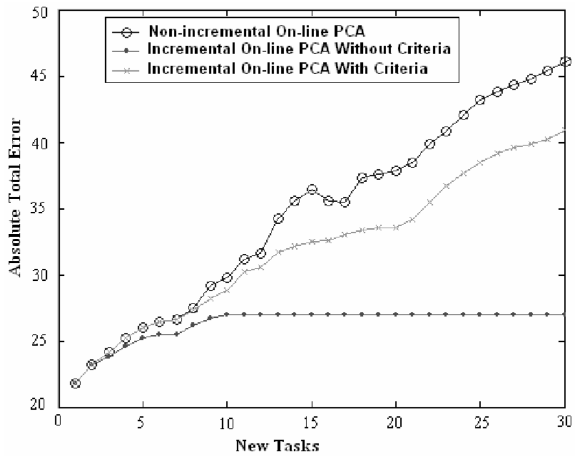
by observing the changes across the new 30 tasks.

We compared the proposed online PCA with several different options:

- non-incremental
- incremental without criteria
- incremental with criteria.

The beginning (New Task 0) in Figure 5. The dimension of Eigenspace was 12, which was the size as in Experiment I.

Figure 5 Absolute total error by training extended tasks as new instances using online PCA; comparison online Non-incremental PCA, vs. online incremental PCA without Criteria, vs. online incremental PCA with Criteria



- The Non-incremental method did not increase the number of eigenvalues; even when New Tasks were captured, the eigenspace was kept at 12. As shown in Figure 5, Non-incremental online PCA has an increased total error as New Tasks were added. This case demonstrated the worst case for accumulating the differences between eigen representation and actual new instances. This method should be chosen only when the system does not allow an increase in the eigenspace dimensions.

- Incremental online PCA without criteria method was the opposite of the first method, and was chosen to demonstrate how error was reduced by increasing the dimension of eigen subspace as much as possible. The system increased the dimension of eigensapce without applying any criteria. Since the system already used up to 12 eigenspace, the remaining eigenspace in Experiment II is 22–12 = 10. As shown in Figure 5, the total error was smaller than the other two methods (Non-incremental and incremental with criteria). It should be noted that after *10th* New Tasks, the error was not increased, since the eigenspace was already filled up and can represent the new instances completely. We observed that

  $$\sum_{i=1}^{n+1} \| X_{i(n+1)} - X'_{i(n+1)} \|$$

  was not increased because of the following reason:

  Using equations (5) and (7) at over *10th* New Tasks in Figure 5,

  $$\begin{aligned} X'_{i(n+1)} &= E' a_{i(n+1)} + m' = E'(E')^T (X_i - m') + m' \\ &= I(X_i - m') + m' = X_i . \end{aligned}$$

  Since the dimension of E' has been 22, $E'(E')^T = I$ Thus after the 10th New Tasks, total error of Incremental online PCA in Figure 5 maintains the same error values by updating the eigenspaces. The drawback of this method was that if the dimension of the measurement space was very large, the dimension of the eigenspace should be too large to cover the new instances.

- Incremental online PCA with criteria method was our optimal solution providing a compromise between the first method and the second method. Our criteria for deciding whether the system increases the eigenspace was that the absolute error of the new task projecting to the previous eigenspace was larger than 0.8. That is at $i = n + 1$, the system checks whether $\| X'_{n+1(n)} - X_{n+1} \|$ was larger than the threshold value. In our experiment, we chose this value as 0.8. The dimension of eigenspace was increased from 12 to 18 after adding up to 30 new tasks in Figure 5. Thus the incremental online PCA was our proposed choice for satisfying both the desire to limit the eigenspace dimension and still maintained sufficient accuracy. The absolute total error was controllable if the system set the threshold values before the system reached the saturated dimension of available eigenspace. In this automated manner, this incremental online PCA with criteria could be applied for long-term training with new instances.

4.3 Experiment 3: classification performance of automated online PCA training for unlabelled instances

We evaluated the Online PCA training performance for classifying unlabelled instances using two sub experiments. In the first experiment, we applied online PCA to 30 new instances distributed among M1–M6 (as shown in Figure 3). The system automatically classified the new instances by measuring the distance in eigenspace between the existing classes M1–M6. To each existing clas, there are 40 training data. We suppose these data had a normal distribution in the class. Therefore, we calculated the expected value

$$u = \frac{1}{n}\sum_{j=1}^{n}(E'a'_j + m'),$$

where n was the amount of training data in one specific class. The standard deviation was

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(E'a'_j + m' - u)^2}.$$

If the new data was within the circle of $r\sigma$, we regarded this new data as belonging to this class. We adjusted scaler value r to examine the change in performance of the classifier (using values of r ranging from 0.5–1.0 in increments of 0.1 and from 1.25–3.0 in increments of 0.25).

Four notions were used to measure the performance of the classifier – TP, FP, TN, and FN. The True/False specified whether that prediction was correct. The Positive/Negative indicated whether the new instance was predicted to belong to the specified class. Therefore, TP (True Positive) means that the classifier predicts the new data belong to the class while actually it really belongs to this class. FP (False Positive) means the classifier predicts the new data belong to the class while actually it really does not belong to this class. TN (True Negative) means the classifier predicts the new data doesn't belong to the class while actually it really does not belong to this class. FN (False Negative) means the classifier predicts the new data does not belong to the class while actually it really belongs to this class. We defined Accuracy as TP/(TP + FN), and the False Positive Rate as FP/(FP + TN).

Figure 6 shows the Receiver Operation Curve (ROC) as an outcome of classification performance. The proportion of 'true' classifications is shown as the false positive rate is relaxed through a successive increase of the parameter value r. For this scenario, the eigenspace was kept at 12, and the system did not expand at all. Figure 6 demonstrates that the new unlabelled training instances were automatically classified with a high level of accuracy at a low rate of false positives.

In the second experiment, we applied 34 new instances from M7 class and 34 new instances from M8 class, which were not used for training before. That is, when a new instance x_{n+1} (not belong to any existing training class) was received and can't be assigned to any existing class, 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

$$\sigma_{p+1} = 2.5 \times \sqrt{\frac{1}{p}\sum\nolimits_{i=1}^{p}\sigma_i^2},$$

where p is the number of classes. When the other new instances were in the $r\sigma_{p+1}$ range of the new class C_{p+1}, the classifier updated the expected value and standard deviation,

$$u_{p+1} = \frac{1}{q+1}(q \times u_{p+1} + x_{n+1}),$$

where q was the previous amount of data received 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 is the parameter value that presented the satisfying data amount in the new class. In this experiment, first we set the M_q at a constant value 500, recorded the accuracy and false positive rate when increasing the threshold value r from 0.8 to 2.0 as shown in Figure 7(a). Figure 7(a) shows good performance of the receiver operation curve as an outcome of online 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 value r. Then as shown in Figure 7(b), we set the threshold value r at a constant value 1.0, recorded the accuracy and false positive rate when increasing M_q from 100 to 500. This results show that our proposed method for unlabelled instances was well classified if the system chose appropriately large M_q.

Figure 6 ROC for classifying existing trained behaviours. The classification accuracy was evaluated by changing the value r

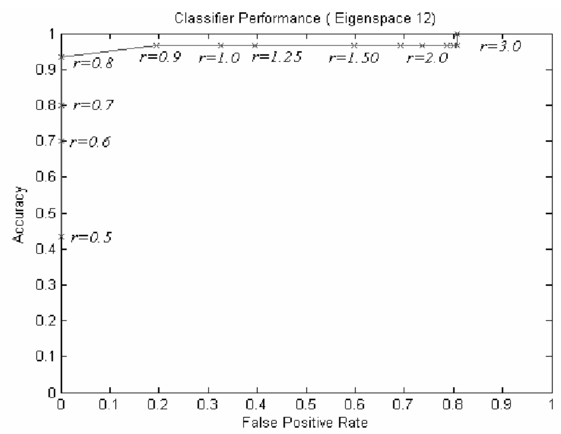
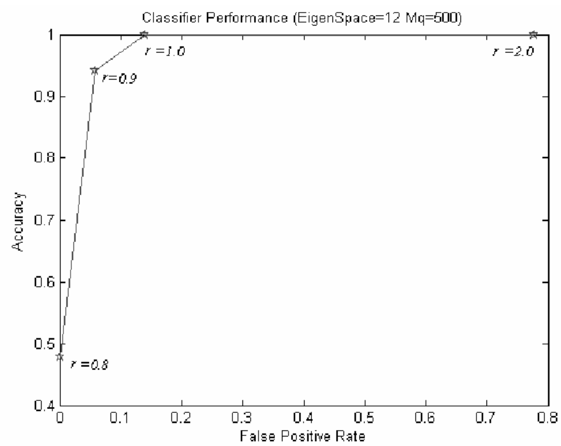
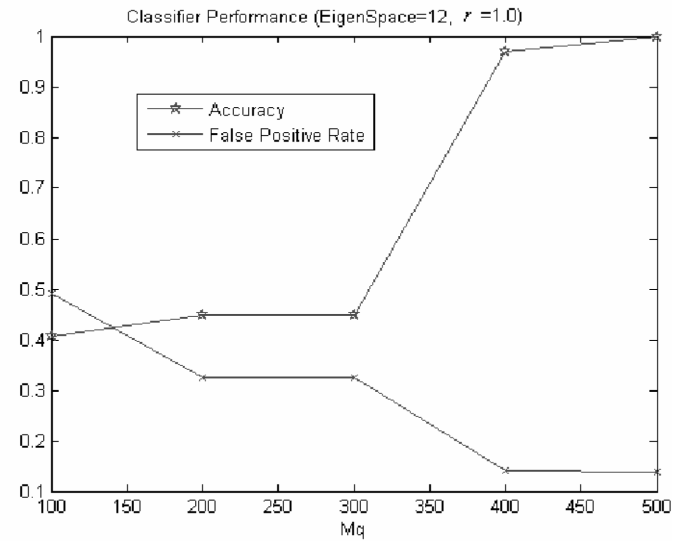


Figure 7 ROC for classifying unlabelled new trained behaviours. The classification accuracy was evaluated: (a) by changing the value r and (b) by changing the value M_q



(a)

Figure 7 ROC for classifying unlabelled new trained behaviours. The classification accuracy was evaluated: (a) by changing the value *r* and (b) by changing the value M_q (continued)



(b)

5 Conclusion

In this paper, we have applied an online learning method for training articulated behaviours. Our contribution to the learning phase was to develop incremental learning methodologies, in which the training behaviours were incrementally executed in an online manner. Although traditionally PCA was conducted off-line, our online PCA sequentially updated the classification without human's inputs since human operation of the system should be minimised as much as possible. In this proposed framework, we did not need to separate the procedures for learning and testing, but we can add the testing instances as an incremental instance. We defined a sequence of images as a single unit of sensory data for positioning land markers. The proposed system simultaneously updated the eigenspace by updating the scatter matrix in an online manner. Since video-based motions were our measurement data, the eigenspace for the new instances arrived as a continuous stream that was too expensive to store. Thus we believe that the major factor in the design of the incremental learning systems was the availability and expandability of the former knowledge based initial behaviours that were acquired. We applied the proposed method for a miniature human figure (replaced by a humanoid robot). The experiment results demonstrated the feasibility and merits of reducing learning dimensionalities using our incremental online PCA. We have developed the incremental learning methodologies for extracting eigen behaviours of articulated motions in an online manner using each image sequences as a unit of sensory data for the positions of land markers. In contrast to the Off-line PCA method, which needed to keep all the original previous instances and new instances in order to reconstruct the eigenspace, our proposed online method did not keep the previous instances, but just kept the eigenspace and reconstruct the space using the new instance. In our online PCA, we were able to add lots of new unlabelled training instances while maintaining the reasonable dimensions of the eigenspace. As a remaining issue, the online classification performance using the incremental PCA described in this paper needs to be evaluated through the long-term training phase.

Acknowledgment

This work is partially supported by Vermont NSF Experimental Program to Stimulate Competitive Research Program (EPS-0236976), Grants-in-Aid of Research of Sigma Xi, and URECA! Program at the University of Vermont. The authors acknowledge J.B. Zurn, K. Gorman, and M. Stommel for their comments.

References

A.P.A.S. (2005) *Ariel Dynamics*, accessed August 2, http://www.arielnet.com/.

Artac, M., Jogan, M. and Leonardis, A. (2002a) 'Mobile robot localization using an incremental eigenspace model', *Proceedings of IEEE International Conference on Robotics and Automation*, pp.1025–1030.

Artac, M., Jogan, M. and Leonardis, A. (2002b) 'Incremental PCA for online visual learning and recognition', *Proceedings of International Conference on Pattern Recognition*, pp.781–784.

Hall, P., Marshall, D. and Martin, R. (1998) 'Incremental eigenanalysis for classification', *Proceedings of British Machine Vision Conference*, pp.286–295.

Haritaoglu, I., Harwood, D. and Davis, L.S. (2000) 'W4: real-time surveillance of people and their activities', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp.809–830.

Humanoid Animation Working Group (2005) Accessed August 2, http://www.h-anim.net/.

Li, Y., Xu, L.Q., Morphett, J. and Jacobs, R. (2003) 'An integrated algorithm of incremental and robust PCA', *Proceedings of International Conference on Image Processing*, pp.245–248.

Motion Analysis Inc. (2005) Accessed August 2, http://www.motionanalysis.com/.

Nayar, S.L., Nene, S.A. and Murase, H. (1996) 'Subspace methods for robot vision', *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp.750–758.

Open HRP (2005) Accessed August 2, http://www.is.aist.go.jp/humanoid/openhrp/.

Vicon Motion Systems (2005) *Oxford Metrics Ltd*, accessed August 2, http://www.vicon.com/.

Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) 'Pfinder: real-time tracking of the human body', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp.780–785.