

# Synthesized Articulated Behavior using Space-temporal On-line Principal Component Analysis

YUICHI MOTAI

*University of Vermont, USA, ymotai@uvm.edu*

## Abstract

This paper presents a new framework to synthesize humanoid behavior by learning and imitating the behavior of an articulated body using motion capture. The video-based motion capturing method has been developed mainly for analysis of human movement, but is very rarely used to teach or imitate the behavior of an articulated body to a virtual agent in an on-line manner. Using our proposed applications, new behaviors of one agent can be simultaneously analyzed and used to train or imitate another with a novel visual learning methodology. In the on-line learning phase, we have developed a new framework of principal component analysis (PCA) using an incremental technique. For traditional off-line PCA, it is necessary to store all the previous and new instances in order to reconstruct the eigenspace. Our proposed on-line PCA can add numerous new training instances while maintaining reasonable eigenspace dimensions. The proposed methodology is well matched to both behavioral training and synthesis, since it is automatically carried out as an on-line long-term learning of humanoid behaviors without the overhead of an expanding learning space. The final outcome of these methodologies is to synthesize multiple humanoid behaviors for the generation of arbitrary behaviors. The experimental results using a humanoid figure and a virtual robot demonstrate the feasibility and merits of this method.

*Keywords: Imitative robot, articulated behavior synthesis, visual motion capture system, learning from observation, human behavior model*

## 1. INTRODUCTION

Synthesis of humanoid/robotic behavior is a computationally demanding process. In the advanced robotics community, autonomous creation of motion, *i.e.*, behavior for a virtual avatar, is a very important research topic. Although the ultimate research goal is to establish an “autonomous” agent with any articulated behaviors, we are far from achieving this ideal level. Thus, here we concentrate our efforts towards making it possible for a robot to imitate or directly learn from a teacher some comprehensive behaviors, or to set up a framework to transfer behaviors by observing actual

behaviors in the real world. In many situations, teaching articulated behavior captured by a video camera is possible for humanoid robots. The use of vision-based motion analysis is not new; rather, it is standard in computational non-contact type measurement systems. For example, some existing 3D motion analysis systems have been already developed for various industrial applications, such as Vicon Motion Systems [1] A.P.A.S. [2], Motion Analysis [25], and [10, 11, 12] in academia.

The contribution for this paper is that we have address the development of a new framework for behavior imitation, in which dynamic behavior acquisition can be achieved by synthesizing principal behaviors. Using motion captured from video sequences, we have implemented an on-line learning framework in which observable instances are incremented. With previous methods, the feature space explodes when the degree of behavioral freedom is increased. Our proposed new method prevents this from happening by limiting the amount of data that must be stored. For this latest method, we extended principle component analysis (PCA) to classify the training behaviors in the iterative manner. More specifically, using an incremental learning technique, we attempt to replicate the best or optimal behavior of a particular agent that has been given direct instruction. Certain behaviors may follow a common pattern, and we can use information from this kinematics motion to build a common or principal task. We will explain how the principal components of articulated motions can be obtained and used together with classification and synthesis. At the beginning of the motion capture and behavior training process, the human-in-the-loop approach is developed when registering the behaviors in the form of basic motion 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 a number of problems in classifying the behaviors that will be automatically carried out in an on-line manner.

The organization of the rest of this paper is as follows: Section 2 describes related studies on learning by observation for humanoid robot. Section 3 illustrates the overall visual observation system to capture articulated movements for a humanoid robot. Our proposed methodologies for on-line learning using incremental PCA are presented in Section 4, and evaluated experimentally in Section 5. Section 6 reports some interesting experimental results to demonstrate the performance of synthesis. Finally, the synthesized performance using eigen behaviors is evaluated in Section 7. Concluding remarks are given in Section 8.

## 2. RELATED STUDIES

The field of visual imitation using motion capture is very active, and has evolved in not only the advanced robotic community, but also in biomechanics and computer graphics. Obviously, those fields have some interdisciplinary overlap; indeed, our project described in this paper has been inspired by and spans these fields. In the area of computer graphics, virtual agents are realized through avatars. The captured motion data are directly used for determining avatar behavior by

following typical biomechanics. Since they do not involve the actual mechanical structures of the robots, it seems simpler to apply forward kinematics to the graphical model of the avatar in the virtual world. 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 [3]. For example, patterns of each joint are selected and optimized under physical constraints [4]. Another extension is [5], in which behaviors of the avatar can be transferred from adults male/females to children by PD control with the fixed relevant parameters. Those extensions among avatars are retargeted if the physical models are clearly specified. To imitate behaviors from the real object in a virtual avatar, many further problems develop: *i.e.* it becomes necessary to estimate modified behaviors for new or different conditions [6]. For example, in the case of a modified run and walk behavior, the avatar can “slowly” run by relocating the two distinguished behaviors (run and walk). In [7], walking behaviors are synthesized by interpolating intermediate managements. Popović and his colleagues [32] developed an interactive system that gave the animator fine control over the motion of a single rigid body. Popović and Witkin [3] showed that significant changes to motion capture data can be made by manually reducing the character to the degrees of freedom for the task. The optimized motion was then mapped back up to the full character. Humanoid motion with many degrees of freedom can be optimized when the animator provides closely spaced key frames without exact timing information [29]. Short segments of motion can be computed for characters with many degrees of freedom as Rose and his colleagues [33] demonstrated when they computed optimal transitions between humanoid motion segments that began and ended with different but similar poses. Liu and Popović [30] showed that some dynamic effects can be preserved by enforcing patterns of linear and angular momentum that do not require the computation of such dynamic parameters as contact forces and joint torques. Outside the area of characterizing virtual robots, reduced order models of dynamics have been explored extensively in computer graphics and other fields [26,27,28,31,34]. In addition to the area of robotics, artificial intelligence, and physiology, numerical approaches of learning by observation robotics have been examined [13,14,15,16,17,18]. For those in the computational vision and machine learning communities, the task-level of learning needs to be constructed a priori based on the captured sensory data of the trajectories. The approaches described in this paper are in the line of those.

### 3. VISUAL OBSERVATION SYSTEM USING IMITATIVE FRAMEWORK THROUGH MOTION CAPTURE

We select articulated objects as targets since we wish to establish efficient and effective interaction among the actual target objects, both among virtual avatars and between the virtual and actual targets. Our overall learning system will be achieved by developing the following phases: 1) Presentation: The targeted articulated agent/human performs the desired behavioral tasks to the system from beginning

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) Recognition: The capturing system observes the performance by a color camera and constructs a spatio-temporal control-point description of the behavior/task. Recognition must be automatic and on-line in order to keep up with the increments of the behavior/task performance. 4) Training: For a given target environment, the system recognizes the initial configuration of the objects, and classifies and instantiates the behavior/task description. 5) Synthesis: The trained system is executed in order to generate the synthesized behaviors/tasks.

To realize these phases, the following required minimal protocols are established: First, humanoid movements are observed by a motion capturing device using a camera with a landmark marker (alternatively a gyroscope, or a dataglove). Then, using a uniquely developed GUI, the motion is registered in the computer incrementally. The registered behavior is retargeted to a virtual humanoid agent from the real world motion. The final step is to synthesize the motion in the agent by specifying how many behaviors are combined. In this proposed scenario, rather than registering the limited joint descriptions, the system learns the higher level behavior 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 target object. However, those descriptions required a labeling procedure in the learning phase. The exhaustive sample-labeling that is needed makes expensive human resources necessary and is often unrealistic. For example, in visually-based automatic gesture training, labeling data is a time consuming and difficult task. For other applications such as remote sensing or medical imaging, labeling data may require very expensive tests and so only a small set of labeled data may be available. Thus, only a few positive samples are available, while unlabeled examples are plentiful. In all these cases, we need to find ways to relieve the annotation burden for the users. Therefore, we develop here a new framework of training behavior with initial labeled instances using incremental learning methodologies, in which the rest of unlabeled training behaviors are incrementally executed in the on-line manner. In comparison to other approaches of incremental learning using each static image [19,20,21,22,23], our proposed method is new since we consider a sequence of images as a single unit of sensory data for the positions of markers. Our contribution in this system is that we have developed PCA to fit video sequences by incrementally updating eigen behaviors in an online manner. The system automatically reconstructs the eigenspace by updating a scatter matrix with the new unlabeled instances. We call these sequences eigenspace eigen behaviors. Another contribution of the system that we have developed is a unique synthesized process for generating autonomous behaviors based on the eigen behaviors. Please note that during the initial PCA supervised learning (before the online incremental PCA), the computer still needs to acquire the label of the initial behaviors. To label each behavior, we will describe a GUI developed as follows.

### 3.1. Graphical interface behavior editor for articulated objects

In the learning phase of behavior acquisition, the abstract models of Human Figure and Humanoid Robot have been prepared. We have developed an Open-GL based GUI editor for the registration process. A humanoid mannequin or robot Kondo KHR-1 is used for articulated objects in the real world. Each physical model, however, is not specified in detail. The number of joints is assigned by choosing from the menu in the behavior-learning editor. Figure 1 illustrates the registration process using our developed GUI editor. A human operator specifies the number of the articulated parts, which corresponds to the number of the highlighted landmarks. The humanoid structure follows Kondo KHR-1 specifications, similar to the standard of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) as FCD 19774 Humanoid virtual robot [8].

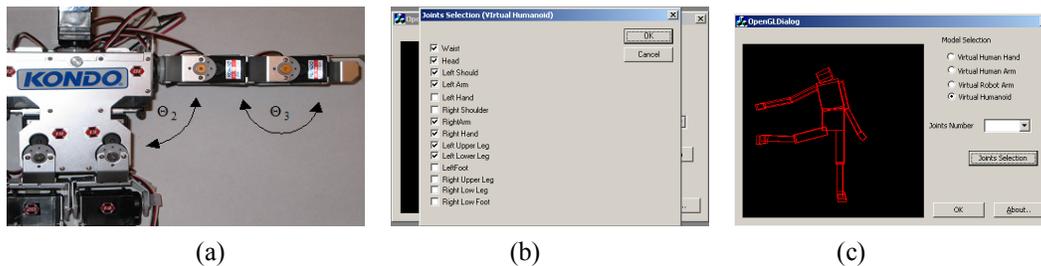


Figure 1: Learning editor (b) and (c) for a humanoid robot Kondo KHR-1 (a). The motion of the articulated objects is captured using the camera sensor.

## 4. ONLINE BEHAVIOR LEARNING USING INCREMENTAL PCA FOR SYNTHESIS

Our proposed learning system has several unique characteristics for capturing the behaviors of articulated agents. In order to limit the number of articulated behavior scenarios, the robotic agent is only allowed to learn a few typical operations with representative postural tasks, such as picking up an object. This provides examples for considering a small number of sequential articulated behaviors. Ultimately we would like to create efficient supervised classifiers, with the goal of requiring a minimal amount of supervision. As illustrated in Figure 2, initial behavior instances, typically numbering 40, are labeled for classification by a human operator using the GUI described in the previous section. This first step is called off-line learning, in which training behaviors are classified using a standard PCA method (Section 4.1). In the next step we develop an on-line learning methodology using a structured learning entity. In our learning frame, we initially apply supervised learning for small instances using PCA, and then we apply a new automated framework using an incremental learning technique (Section 4.2). Finally we will discuss the synthesis module by utilizing the PCA framework. (Section 4.3).

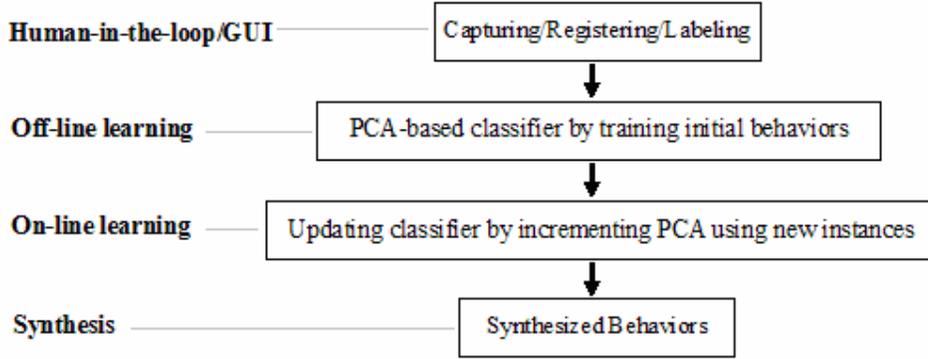


Figure 2: Overall learning procedure.

#### 4.1. Off-line PCA for time sequential data

The representation of the learned behaviors plays a very important role in determining how the learning algorithm works. Although we prepare the training instance by setting the beginning and ending of behavior sequences, each frame of the sequence is analyzed for the images from a camera sensor or from each data sampling of a data glove. We extend the data frames in the following way.

1) Rather than using the entire set of 2D image pixels to get an eigenimage (as in [26]), here the data are only extracted from the marker points. 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 in a 2D image.

2) For each kind of behavior, we will take a sequence of images “ $q$ ” to completely classify “ $p$ ” types of behaviors. More specifically, we reconstruct this measurement sequence by defining  $X_i$  to denote

the  $i$ -th sequence  $X_i = [x^l_{i1} \quad x^l_{i2} \quad \dots \quad x^l_{iq}]$ ,  $i=1 \dots p$ , 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. Initially

we calculate a  $2l \times q$  dimensional sample mean matrix  $m$  by computing  $m = \frac{1}{n} \sum_{i=1}^n X_i$ . We define

the scatter matrix  $S$  of image sequences  $X_i$  as  $S = \sum_{i=1}^n (X_i - m)(X_i - m)^T$ . Using the

eigenvectors  $e_j$  ( $j=1 \dots k$ ) of  $Se_j = \lambda_j e_j$ , the  $p$  types of behaviors  $X_i$  are decomposed

by  $X_i = m + Ea_i$ , where  $E = [e_1 \quad e_2 \dots \quad e_k]$  and  $i=1 \dots p$ . The principal component or scalar vector

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

In order to track the movement of the humanoid figure precisely and prevent interference from the environment, we put a color sticker on each joint of the humanoid. For example, in our experiment we have total of 11 color stickers; each color sticker has a vertical and horizontal position. Therefore, the dimensionality of each image has been reduced from 256\*256 to 22\*1. Since we have  $p$  types of behaviors and  $q$  image sequence for each behavior, 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.

#### 4.2. Incremental on-line PCA for time-sequential data

During the training phase, we train each behavior based on extracted sensory data. In the phase where new, unlabeled sequences are acquired, the system will update the eigenvectors and find the closest behavior which best represent the new sequence. We will apply a new, incremental PCA into the behavior classification.

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 is 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 process new images during learning, the entire set of original training image sequences must be stored. If the training images are very large, such a method will consume the storage of the agent. Instead, the use of incremental PCA to represent the training behavioral 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 we have obtained a eigen subspace  $E = [e_1 \ e_2 \dots \ e_k]$  from image sequences  $X_i$ ,  $i=1\dots n$ . The corresponding  $k$  largest eigenvalues are  $\lambda_j$ , the principal component scalar matrix is  $a_i = [a_{i1} \ a_{i2} \ \dots \ a_{ik}]$ ,  $i=1\dots n$ , 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:

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

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

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

The updated scatter matrix can be obtained:

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

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 summarized as follows. When a new image sequence  $X_{n+1}$  is received, we compute the new sample mean  $m'$  and scalar vector  $a_{n+1}$  for the old subspace  $E$ , then construct an updated subspace  $E'$ , but with no image representation. Let us denote that  $X_{i(n)}$ ' represents image sequences  $X_i$  in the old subspace of  $E$  and superscript  $'$  means that the image sequences are reduced by the eigenvectors. Also  $X_{i(n+1)}$ ' denotes the representation of previous image sequences  $X_i$  and the new image data  $X_{n+1}$  in the updated subspace  $E'$ . Then:

$$X'_{i(n)} = E a_{i(n)} + m \quad i=1 \dots n \quad (4)$$

In the new subspace:

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

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

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

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

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

After updating the knowledgebase, we need to represent all the sensory data of image sequences again in the new subspace, so that we do not need to keep the original image sensory data  $X_{i(n)}$  of images 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 Eq. (7).

In order to reduce the size of the representation, we attempt to limit the number of stored dimensions to  $k$ , but only if this number does not reduce the classification accuracy. These  $k$  new eigenvectors are sorted by decreasing order of the eigenvalues. We will define "Incremental On-line PCA" as the condition when the eigenspace  $k$  is expanded into  $k+1$  when the new eigenspace is computed with new instance  $X_{n+1}$ . We will designate the classification condition as "Non-incremental On-line PCA" if the eigenspace maintains the previous dimension after updating.

We now introduce some criteria to judge when the dimension of the eigen subspace should be expanded from  $k$  to  $k+1$  in order to maintain the desired balance between storage and accuracy. Two

criteria are evaluated by comparing thresholds to determine whether or not the eigenspace needs to be extended. Criterion 1: The new sensory data of image sequences at  $i=n+1$  cannot be represented with the old subspace. This occurs when the error between the original image and the reduced image  $\|X_{i(n+1)} - X'_{i(n)}\|$  has exceeded a pre-set threshold. Criterion 2: If the new sensory data of image sequences can be represented with the old subspace, but the overall sum of error for  $n+1$  sensory data from the images  $\sum_{i=1}^{n+1} \|X_{i(n)} - X'_{i(n)}\|$  has exceeded the threshold, we need to increase the dimension of the subspace. Therefore if one of the error calculations exceeds its threshold, the eigen space is extended, applying Incremental On-line PCA.

### 4.3. Behavior synthesis using PCA-based training for time-sequential data

The system synthesizes a new or blended behavior using the trained behaviors, which are labeled  $i=1\dots n$ . There are a number of ways to generate such a behavior. As a straightforward approach, we wish to establish a methodology based on the reduced PCA space. As shown in the previous subsection, we have computed a scatter matrix and a mean matrix over the entire set of training data:

$$S = \sum_{i=1}^n (X_i - m)(X_i - m)^T, \quad m = \frac{1}{n} \sum_{i=1}^n X_i \quad (8)$$

We decompose the scatter matrix into within-class matrix  $S_{wc}$  (scattering in each behavior at  $i=1,2,..c$ ) and between-classes matrix  $S_B$  (scattering across multiple behaviors). That is  $S = S_{wc} + S_B$ , and those are defined in the form:

$$S_{wc} = \sum_{i=1}^c S_i = \sum_{i=1}^c \sum_{j=1}^{n_i} (X_{ij} - m_i)(X_{ij} - m_i)^T \quad (9)$$

where  $n_i$  is the number of data in the  $i$ -th class

$$S_B = \sum_{i=1}^c n_i (m_i - m)(m_i - m)^T \quad (10)$$

$$\text{where } m_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} \quad \text{and} \quad a_i = E_B^T (m_i - m) \quad (11)$$

The decomposition of scatter matrix is not new, it is a standard method in linear discriminant analysis (LDA). We import this idea for synthesizing a new behavior by computing eigen behavior of between-class scatter matrix. Suppose we have total  $c$  behavior classes ( $i=1,2,..c$ ), the synthesized behavior can be reconstructed using between-class scatter matrix  $S_B$  by solving  $S_B e_j = \lambda_j e_j$ . Let re-denote  $E_B = [e_1 \ e_2 \ \dots \ e_{c-1}]$  for the eigenvectors. Now the system may generate the new behavior  $X'_{syn(i)} = E_B a_i + m_i$ . Note that since the behavior data format is the same as in the

previous section of  $X_i = [x_{i1}^l \quad x_{i2}^l \quad \dots \quad x_{iq}^l]$ , the spatio-temporal position of landmarks are already specified and can be used for the synthesized behaviors.

## 5. LEARNING PERFORMANCE OF AUTOMATED ON-LINE PCA TRAINING USING SEQUENTIAL DATA

In this experiment, we have used 11 color-point markers for all movable joints of a humanoid object shown in Figure 3. A Pulnix CCD color camera was used for capturing the color markers. We chose 6 representative human figure positions for partial behavioral sequential tasks. The task starts from a flat standing state, and completes the locomotion by one of the states chosen from 6 figure structures described below. These behaviors 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, and M6: raise both legs. Each behavior was captured during 4 seconds for 16 frames. From the color image, the 2D position of each marker point  $x_i^l = (u^l, v^l)^T$  is extracted, where  $u, v$  is a 2D point at each center position of the  $i$ -th joint (from  $i=1$  to 11). Thus, in the initial off-line PCA training space, the sensory image sequence  $X_i$  has 22 rows and 16 columns. Using the 6 behavior instances, the overall size of matrix  $[X_1, X_2, \dots, X_n]$  is  $22 \times 96$ . The dimension of the corresponding scatter matrix  $S$  is  $22 \times 22$ , and the eigenbehavior is at most 22, based on the matrix formation. In the incremental on-line PCA training phase, we may add any additional sensory sequence  $X_i$ . The computation time used for the incremental PCA is mainly due to computation of eigenvalues/vectors, and is short enough to not interfere with capturing the new image data frame.

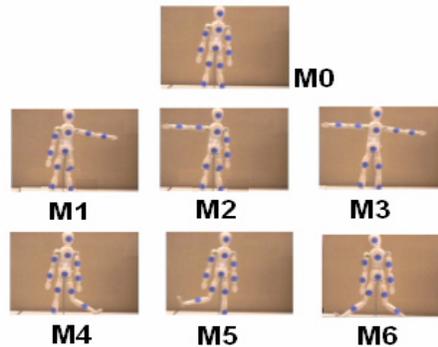


Figure 3: Six behavioral training tasks of a human figure, captured using a camera sensor. Behavior tasks are added to the training sequence for incremental learning.

Traditional off-line PCA was first conducted using the initial  $6 \times 5 = 30$  training behaviors; then we applied the following five different methods to handle the new behaviors:

- 1) the New Training method, which involves classifying the new behavior using the old eigenvectors.
- 2) the Non-incremental On-line method, which updates the eigenvectors using the new behavior and old reconstructed behaviors, maintaining the same eigenspace dimensions.
- 3) the Incremental On-line method, which updates the eigenvectors using both the new behavior and the old reconstructed behaviors and expands the eigenspace.
- 4) the Non-incremental Off-line PCA, which involves adding the new behavior to the original training behaviors and updating the eigenvectors, but keeping the same eigenspace dimension.
- 5) the Incremental Off-line PCA, which not only updates the eigenvectors using the new and old original behaviors, but also expands the eigenspace.

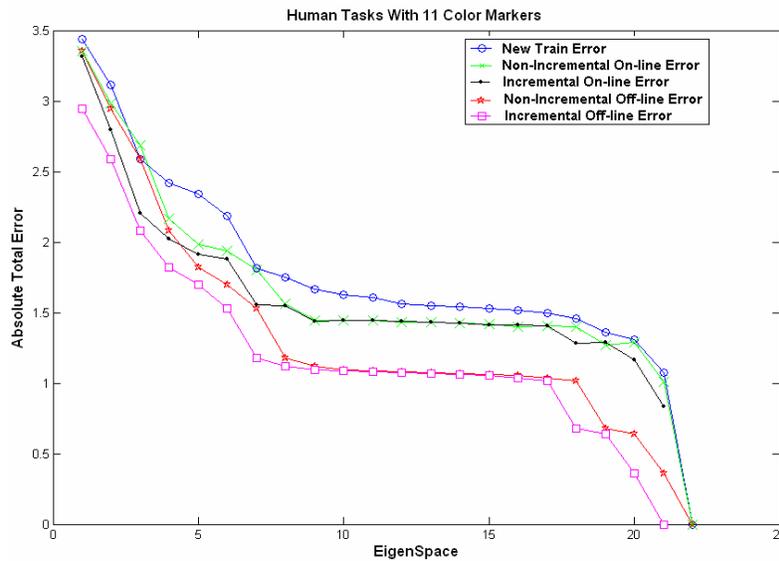


Figure 4: Total error transition with respect to eigenspace dimension across several PCA methods: New Training, Non-incremental On-line, Incremental On-line, Non-incremental Off-line, and Incremental Off-line.

The fitting measure was evaluated by comparing the measured distance between the raw measurement value  $X_i$  and the reconstructed value  $X_{i(n)}$  from Eq. (4) using traditional On-line PCA at  $n$ -th. Also, while the new data were acquired, the fitting measure was evaluated using  $X_{i(n+1)}$  from Eq. (5) for On-line PCA at the  $n+1$ -th sequence either incrementally or non-incrementally. To determine the new dimensionality of the eigen subspace, we applied criterion 1 and 2 described in Section 4.2. In other words, the error  $\|X'_{i(n)} - X_i\|$  or  $\|X'_{i(n+1)} - X_i\|$  was computed with respect to the number of input eigenvalues (as demonstrated on the X-axis in Figure 4). When the error was small, the reconstructed value was close to that of the instance, which indicated that the eigenspace expanded well to represent the sample instance. Figure 4 compares the errors of all five of the PCA methods described above (New Train, Non-incremental On-line, Incremental On-line,

Non-incremental Off-line, Incremental Off-line), across a number of eigenspace dimensions. The absolute total error for On-line PCA was larger than that for Off-line PCA for eigenspaces 8-22, but the difference between them was not large. The difference of absolute total error between the Non-incremental and Incremental On-line PCA methods is nearly zero for eigenspaces 8-22. However, for a small number of eigenspaces (1-8), the incremental methods perform better in terms of error reduction, both for On-line and Off-line PCA performs better. Unlike Off-line PCA, our proposed method does not store the previous instances, only the eigenspace; the space is reconstructed using the new instance. The On-line PCA allows the learning system to discard the acquired measurement data immediately after the update, reducing storage requirements. Figure 4 illustrates that our On-line PCA method maintains the reconstruction accuracy of traditional Off-line PCA.

## 6. SYNTHESIS PERFORMANCE OF AUTOMATED ON-LINE PCA TRAINING FOR HUMANOID MOTIONS

In this experiment, the system synthesized a new or blended behavior using the trained eigen behaviors as described in Section 4.3. We captured 50 image frames for each sequence and repeated 50 times for each category of behavior. The synthesized performance was evaluated by comparing the corresponding actual behavior  $X_{actual}$  with the new synthesized behavior  $X'_{syn} (= \sum_i E_B a_i + m_i)$ . Please note that the format for the spatio-temporal position of landmarks was identical for the actual and the synthesized behaviors, therefore the degrees of freedom and graphical models of the target humanoid stayed the same. The evaluation criterion was defined by the absolute total accuracy  $1 - (\| X_{actual} - X'_{syn} \| / \| X_{actual} \|)$ .

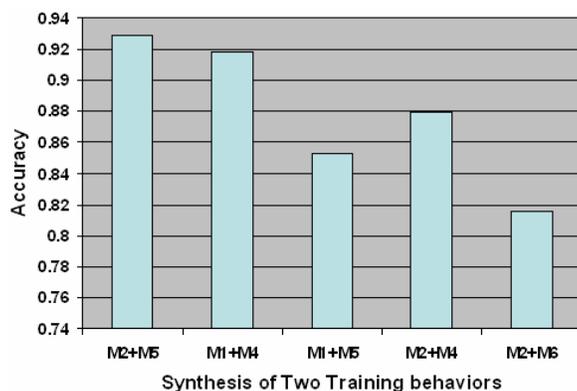


Figure 5: Overall accuracy when actual and synthesized behaviors are compared.

For example, Figure 5 shows that  $X'_{syn} (= X'_{syn(2)} + X'_{syn(5)})$  was the closest to  $X_{actual}$  in the accuracy degree close to 1.0. This result indicates that the chosen behaviors are dissimilar each

other; therefore the synthesized “combination” behavior would be expected to generate a new behavior. Note that our synthesis method decomposes scatter matrix  $S$  into  $S_{wc}+S_B$ . The extraction of discriminant factors between eigen behavior 2 and 5 are generated by minimizing (subtracting) the within-class portion and thus maximizing (enhancing) the between-class portion.

As shown in Figure 6, the output synthesized behavior for the combination of M2 and M5 confirmed the accuracy of the synthesis using our eigen behavior approach since we can easily compare it to the actual motion capture data.

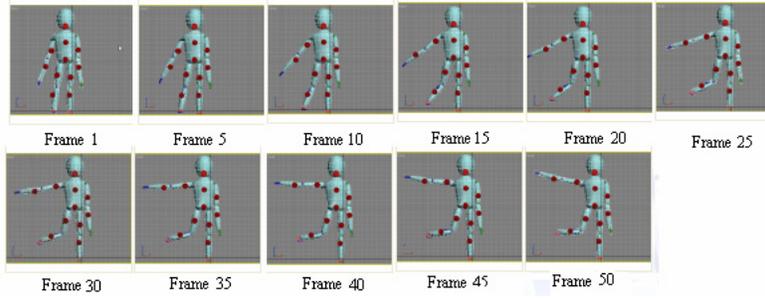


Figure 6: Synthesized virtual robot sequence for combining eigen behaviors using M2 and M5.

## 7. PERFORMANCE OF SYNTHESIS FOR AUTOMATED ON-LINE PCA TRAINING FOR HUMANOID MOTIONS

As an extension of combining two behaviors, we set an arbitrary interpolation by synthesizing across multiple (more than 2) eigen behaviors, that is  $X'_{syn} (= \sum_i^p E_B a_i + m_i)$ , where  $p$ , the number of behaviors to be synthesized, will be determined by a human operator/designer. In our synthesis framework, an arbitrary operation is restricted so that the between-classes can be maximized and the within-classes can be minimized. The criteria that we shall consider are the trace and the determinant of the scatter matrices described in Section 4.3. The trace criterion was chosen and represented by maximizing the following criterion, called the relative trace of the between-class

scatter matrix:

$$\arg \frac{\text{tr} [ S_B ]}{\text{tr} [ S_p ]} = \frac{\sum_{i=1}^p n_i (m_i - m)(m_i - m)^T}{\sum_{i=1}^p \sum_{j=1}^{n_i} (X_{ij} - m_i)(X_{ij} - m_i)^T} \quad (12)$$

This new criterion measures the relative square of the radius between eigen behaviors with respect to the choice of  $p$  (number of behaviors to synthesize) and  $i$  (which eigen behaviors). Please note that  $S_p = S_{wc} + S_B$ , and the trace is proportional to the sum of the variances in the coordinate directions. The proposed criterion is maximized in order to evaluate the degree of dissimilarity between the chosen eigen behaviors.

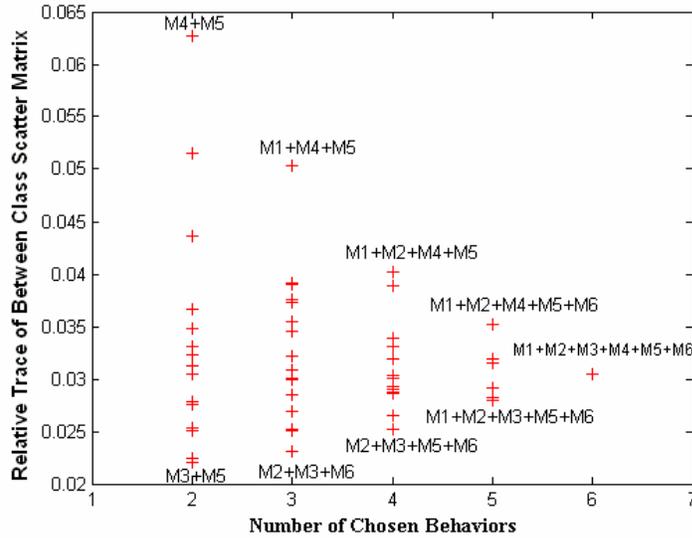


Figure 7: Relative Trace SB for comparison between chosen combinations of eigen behaviors.

Figure 7 illustrates that the relative trace was largest when the most dissimilar eigen behaviors were chosen. It is obvious that fairly well separated behaviors are expected. This measure of separation indicates which behaviors may be blended to synthesize natural synthesis. Such a measure is possible because highly correlated behaviors cannot be used to efficiently synthesize arbitrary behaviors. Figure 8 illustrates the synthesized results using 4 behaviors from 6 eigen behaviors. For example, M1-M2-M4-M5 behaviors are chosen among  ${}_6C_4 = 15$  cases because of the relative trace criterion. Since the actual behavior meets all required inverse kinematics for joints, the above results would confirm that the synthesized behavior follows the related constraints across the joints. In comparison to other studies [3, 25], our synthesized method is very efficient in terms of both computational time and data space requirements, because we do not solve the iterative constraint optimization problem, which always suffers from local minimum and convergence issues. Future work would include the quantitative comparison of our method with other methods that utilize kinematic constraint-based optimization. Also, the behavior synthesis resulting from our method (i. e. the example shown in Figure 8) should be evaluated through a subjective rating by human operators.

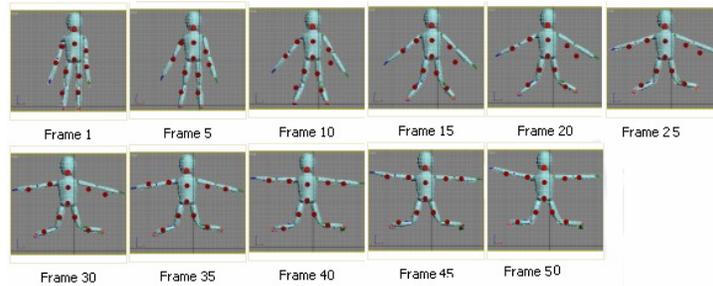


Figure 8: Synthesized virtual robot sequence for combining eigen behaviors using M1-M2-M4-M5.

## 8. CONCLUSION

In this paper, we have developed an on-line learning method for imitating articulated behaviors and utilized these behaviors to help synthesize new behaviors. We have applied an incremental learning framework by updating the classifiers from initial supervised PCA output. Typically, traditional PCA has primarily been used in an off-line manner. In contrast to off-line PCA method, which needs to store all the original previous instances and new instances in order to reconstruct the eigenspace, our proposed on-line method does not need to keep previous instances, only the eigenspace. The space is then reconstructed using each new input instance. Since video-based motions are our measurement data, the new instances arrive as a continuous stream that is too expensive to store. In our on-line PCA, we can add many new training instances while maintaining reasonable eigenspace dimensions. We have developed automated incremental learning methodologies using the positions of land markers in each image sequence as a unit of motion capture data. Each set of land marker data is used as a training behavior and incrementally executed as eigen behaviors in an on-line manner. The experimental results demonstrated the feasibility and merits of the reduction of the learning dimension using our on-line incremental PCA. We observed that a major factor in the design of the autonomous behavior is the availability of an initial, principle set of eigen behaviors, and the ability to expand the set by blending dissimilar (uncorrelated) eigen behaviors. Thus we set a framework to analyze the scatter matrix for eigen behaviors into between-classes and within-classes, and to synthesize the between-classes eigen behaviors. We introduced the criterion of the relative trace of between-class scatter matrix to determine which eigen behaviors should be chosen. The greatest advantage of our proposed synthesized approach is that it is extremely efficient in both computational time and data space requirements, while the other methods require solving constraint-based iterative optimization problems, which always suffers local minimum and convergence issues. As a remaining issue, the on-line training and synthesis performance using the eigen behavior should to be evaluated in a long-term training phase to acquire a larger database for training and testing.

## ACKNOWLEDGEMENTS

The author acknowledges J. Brooks Zurn for proof reading and comments.

## 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] Z. Popovic and A. Witkin, Physically Based Motion Transformation, in *Proc. of SIGGRAPH*, pp. 11-20 (1999)

- [4] J. K. Hodgins and N.S. Pollard, Adapting Simulated Behaviors For New Characters, in *Proc. of SIGGRAPH*, pp. 153-162, (1997)
- [5] A. Bruderlin and L. Williams, Motion Signal Processing, in *Proc. of SIGGRAPH*, pp. 97-104 (1995)
- [6] A. Witkin and Z. Popovic, Motion Warping, in *Proc. of SIGGRAPH*, pp. 1050-108 (1995)
- [7] H.C. Sun and D. N. Metaxas, Automating gait generation, in *Proc. of SIGGRAPH*, pp. 261-269 (2001)
- [8] <http://www.h-anim.net/> [On-line]
- [9] <http://www.is.aist.go.jp/humanoid/openhrp/> [On-line]
- [10] C. Stauffer and W.E.L. Grimson, Learning patterns of activity using real-time tracking, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8) pp. 747 – 757 (2000)
- [11] C.R.Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19** (7) pp. 780-785 (1997)
- [12] I. Haritaoglu, D. Harwood, L.S. Davis, W<sup>4</sup>: real-time surveillance of people and their activities, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8) pp. 809 - 830 (2000)
- [13] A. Fod, M. Mataric, and O. C. Jenkins, Automated Derivation of Primitives for Movement Classification, *Autonomous Robots*, **12** (1) pp. 39-54 (2002)
- [14] O. C. Jenkins, M. J. Mataric, and S. Weber, Primitive-Based Movement Classification for Humanoid Imitation, in *Proc. of International Conference of Humanoid Robotics* (2000)
- [15] Motion Analysis Inc, <http://www.motionanalysis.com/> [On-line]
- [16] 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, in *International Journal of Applied Artificial Intelligence*, **12**, pp. 701-729 (1998)
- [17] S. Schaal, Is imitation learning the route to humanoid robots, in *Trends in Cognitive Sciences*, **3** (6), pp. 233-241 (1999)
- [18] J. Demiris, G. Hayes, Imitative learning mechanisms in robots and humans, in *Proc. of European Workshop on Learning Robots*, pp. 9-16 (1996)
- [19] Y. Li, L. Q. Xu, J. Morphet, R. Jacobs, An integrated algorithm of incremental and robust PCA, in *Proc. of International Conference on Image Processing*, pp. 245-248 (2003)
- [20] M. Artac, M. Jogan, A. Leonardis, Incremental PCA for on-line visual learning and recognition, in *Proc. of International Conference on Pattern Recognition*, pp. 781 – 784 (2002)
- [21] M. Artac, M. Jogan, A. Leonardis, Mobile robot localization using an incremental eigenspace model, in *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1025 – 1030 (2002)
- [22] P. Hall, D. Marshall, and R. Martin, Incremental eigenanalysis for classification, in *Proc. of British Machine Vision Conference*, pp. 286-295 (1998)

- [23] S. L. Nayar, S. A. Nene, and H. Murase, Subspace methods for robot vision, in *IEEE Transactions on Robotics and Automation*, **12** (5), pp. 750 - 758 (1996)
- [24] P. Maes, T. Darrell, B. Blumberg, A. Pentland, The ALIVE System: Wireless, Full-body Interaction with Autonomous Agents, in *ACM Multimedia Systems*, **5** (2), pp. 105-112 (1997)
- [25] A. Safonova, J. K. Hodgins, N. S. Pollard, Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces, in *Proc. of SIGGRAPH*, pp. 514 - 521 (2004)
- [26] F. De La Torre, and M. J. Black, A framework for robust subspace learning, in *International Journal of Computer Vision*, **54**, pp. 117–142 (2003)
- [27] O. C. Jenkins and M. J. Mataric, Deriving action and behavior primitives from human motion data, in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2551–2556 (2002)
- [28] Y. Li, T. Wang, and H.-Y. Shum, Motion texture: a two-level statistical model for character motion synthesis, in *ACM Trans. on Graphics*, pp. 465–472 (2002)
- [29] Z. Liu, and M. Cohen, Decomposition of linked figure motion: Diving, in *Proc. of Eurographics Workshop on Animation and Simulation* (1994)
- [30] C. K. Liu, and Z. Popovic', Synthesis of complex dynamic character motion from simple animations, in *ACM Transactions on Graphics*, **21** (3), pp. 408–416 (2002)
- [31] A. Pentland, and J. Williams, Good vibrations: Modal dynamics for graphics and animation, in *Proc. of SIGGRAPH*, **23**, pp.215–222 (1989)
- [32] J. Popvic', S. M. Seitz, M. Erdmann, Z. Popovic', and A. P. Witkin, Interactive manipulation of rigid body simulations, in *Proc. of SIGGRAPH*, pp. 209–218 (2000)
- [33] C. F. Rose, B. Guenter, B. Bodenheimer, and M. F. Cohen, Efficient generation of motion transitions using spacetime constraints, in *Proc. of SIGGRAPH*, pp.147–154 (1996)
- [34] M. Santelio, M. Flanders, and J. F. Soechting, Patterns of hand motion during grasping and the influence of sensory guidance, in *Journal of Neuroscience*, **22**, pp. 1426–1435 (2002)