Primitive-Based Movement Classification for Humanoid Imitation

Odest Chadwicke Jenkins, Maja J Matarić, and Stefan Weber

Department of Computer Science, University of Southern California, Los Angeles CA 90089-0781, USA,
cjenkins|mataric|stefanwe@usc.edu,
http://www-robotics.usc.edu/~agents/imitation.html

Abstract. Motor control is a complex problem and imitation is a powerful mechanism for acquiring new motor skills. In this paper, we describe *perceptuo-motor primitives*, a biologically-inspired notion for a basis set of perceptual and motor routines. Primitives serve as a vocabulary for classifying and imitating observed human movements, and are derived from the imitator's motor repertoire. We describe a model of imitation based on such primitives and demonstrate the feasibility of the model in a constrained implementation. We present approximate motion reconstruction generated from visually captured data of typically imitated tasks taken from aerobics, dancing, and athletics.

1 Introduction

Imitation is a powerful mechanism for acquiring new skills. It involves an intricate interaction between perceptual and motor mechanisms, both of which are complex in themselves. Research into vision and motor control has explored the role of "subroutines", schemas [1], and other variants based on the notion of organizing the complex perceptual and motor repertoire into smaller sets of reusable modules. The coupling between visual perception and motor control remains a major challenge in robotics.

In this paper, we combine the notions of perceptual and motor routines into *perceptuo-motor primitives*. This notion is based on recent neuroscience evidence identifying "mirror neurons", a mechanism which directly couples the observation of certain movements and their motor execution [32]. Our primitives serve an analogous function, being a basis set for structuring motor control and a vocabulary for classifying and imitating observed movements. We describe a model of imitation based on such primitives [25]. Next, we demonstrate the feasibility of the model in a constrained implementation. We present approximate motion reconstruction generated from visually captured data of typically imitated tasks taken from aerobics, dancing, and athletics.

The rest of the paper is organized as follows. In Section 2, we discuss the role of primitives for humanoid imitation and outline the properties of perceptuo-motor primitives. Section 3 provides an overview of work related to imitation. In Section 4, we describe our imitation model. Section 5 presents an implementation of a constrained version of this model for imitation of a humanoid torso and state several objectives for this implementation are stated in this section. The use of this implementation for control of Adonis [27], a 20 DOF physically simulated humannoid torso, is also discussed in this section. Results of experiments for imitating a set of input motions inspired by athletics, aerobics, and dancing are presented in Section 6. Issues for and limitations of our model are discussed in Section 7. The paper is concluded in Section 8.

2 The Role of Perceptuo-Motor Primitives in Imitation

At the highest level, perceptuo-motor primitives are defined with two components: a set of motor controllers and an associated set of perceptual descriptions. The motor controllers are constrained by the kinematic and dynamic properties of the agent's motor system, and the commonly performed tasks. Each motor controller is then used for deriving descriptions (or models) of observed movement to facilitate motion perception (Figure 1). The motor controllers can either be manually derived, as in the current implementation, or learned, as in our other work [14]. Complex motor behaviors result from sequences and/or superpositions of the primitives.

Our model is based on neurosceince evidence for "motor primitives", which structure movement, and mirror neurons, which couple such movement with associated visual inputs [25], discussed in more detail below. We developed the notion of perceptuo-motor primitives as the unifying mechanism between the perceptual and motor systems, influenced by the constraints of each. These primitives encode "generic" movements, invariant to exact Cartesian position, rate of motion, size, and perspective, through parametrization of both the perceptual and motor components. The perceptual component allows for classifying a spectrum of varying but related movements using a given primitive, and motor controllers allow for the execution of a similarly large spectrum of motion using the same primitive.

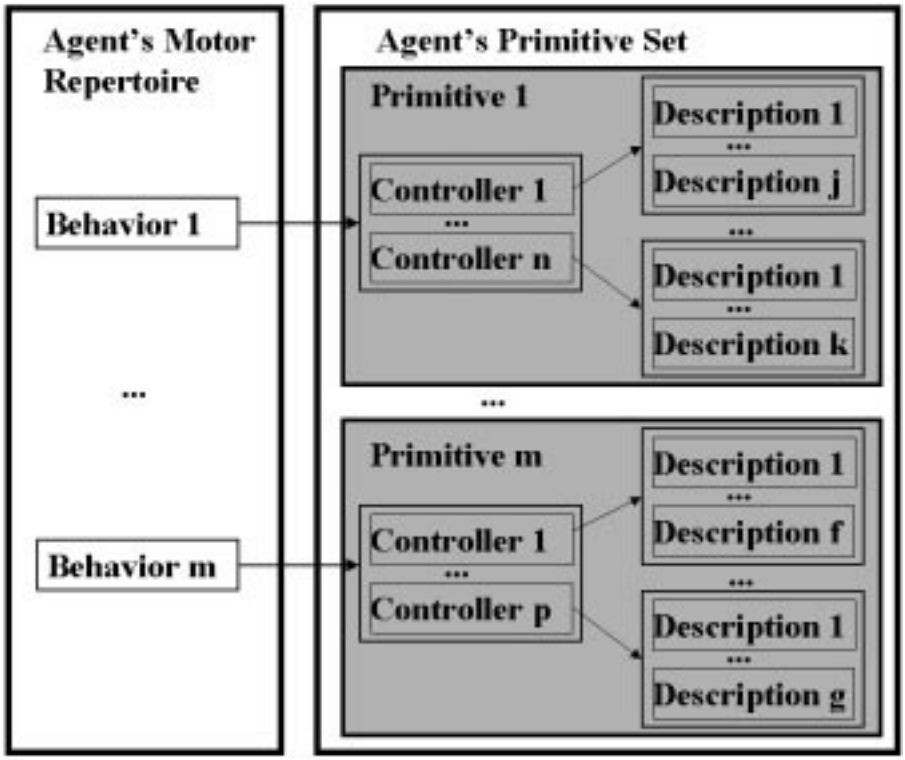

Fig. 1. *A diagram of a set of perceptuo-motor primitives. The arrows imply a "used to derive" relationship. Based on biological evidence, we assume that an agents motor repertoire consists of a set of behaviors, which can be sequenced and superimposed to create complex movment. These behaviors are used to derive a set of motor controllers for the agents primitive set. Analogously, these controllers can be sequenced and superimposed to create a similar set of complex movment. From each controller, a set of perceptual descriptions is generated to allow the agent to percieve movement that it can execute.*

Imitation, in our model, consists of automatically classifying observed movement into the set of primitives, resulting in a representation of the movement specified in terms of the primitives. The motor controller components can then execute an imitation of the observed movement using that representation. Our inspiration comes from biology, where precise, high-fidelity imitation requiring exact quantitative measurements of the observed movement is unlikely to be achieved. Thus, we aim at functional behavioral approximations of the observed movements.

2.1 Structuring the Motor and Perceptual Systems

Neuroscience evidence [8, 15, 28] lends support for an organization of the motor system in the form of a small set of additive behaviors. For example, the frog and rat motor systems are organized into a small (on the order of dozens) set of convergent and additive spinal vector fields [15, 28]. This provides the motivation behind structuring motor control in our model [25]. We use a small set of parametrized controllers in order to modularize and simplify control of articulated kinematic structures with a high number of degrees of freedom, in particular humanoids.

Ideally, the set of motor controllers and the set of perceptuo-motor primitives create a one-to-one mapping. This assumes a complete parametrization of the motor controllers; in its absence, the set of perceptuo-motor primitives is necessarily a subset of the agent's motor controllers. Namely, there is a many-to-one mapping between controllers and primitives. Primitives provide higher-level descriptions of a movement while the properties of these descriptions give parameterizations for the motor controllers. Ideally, a small number of general primitives to represent a large class of movements.

This approach stands in contrast to the explicit motor planning approach, which computes trajectories at "run time", whenever they are needed [30]. When using motor behaviors, stereotypical trajectories are looked up and parameterized for the specific task at hand, rather than computed *de novo* using inverse kinematics and dynamics computation. Because inverse kinematics computation can be costly [35], the notion of a small set of basis behaviors allows for the learning and reuse of an approximation of the inverse kinematics for specific areas in the workspace or a specific trajectory. Thus, a set of locally valid solutions for various problems are stored and recalled.

This approach provides a simpler mechanism than computing inverse kinematics anew time and again. However, unlike memory-based learning [5], which provides high-fidelity reconstruction but requires a large amount of memory, primitives are intended as a more parsimonious representation.

Neuroscience evidence reports on mirror neurons, areas in the motor cortex that trigger execution of specific movements, but also respond to observing the same movements executed by others [32]. This inspiration leads us to assume that the movement perception system is biased by the movement controller repertoire so as to perceive what it can execute. Thus, in our model, the perceptual component of a perceptuo-motor primitive is made to correspond to the type of movements the associated motor controller(s) can execute.

Choosing a good set of primitives is of key importance in this approach. In our other work [14], we explore learning a set of perceptuo-motor primitives. In this paper, we describe an implementation that manually derives primitive instances from actual motion data, as described in Section 5.

3 Related Work

The notion of primitives was inspired by neuroscience evidence, and has been applied to motor control of mobile robots in the form of schemas [2, 1, 3] and in our own work in the form of basis behaviors [26, 23]. Unlike previous work, which focused on either perceptual or motor primitives, we present a mechanism that combines the two, inspired by the function of mirror neurons. This provides a natural mechanism for imitation.

A key property of our percepuo-motor primitives is the direct coupling of perception and motor execution. [36] use a related notion in their computational theory in which formation and recognition are two aspects of a single function. [16] and related work is based on the use of via-point trajectories for motor control. In this paper, we also use via-point trajectories to simulate the functionality of motor controllers in primitive sets.

Notions related to movement primitives have been explored by Brand and Hertzmann [10] for editing of motion capture data and by Bregler [11] in the form of ”movemes” for labeling human motion in unsegmented video streams.

Using demonstration for task specification can be viewed as a coarse type of imitation. It has been studied in the area of robotics, where a series of visual images of a human performing an object stacking task was recorded, segmented, interpreted, and then repeated by a robotic arm [19, 22, 18, 21]. The major focus on this work was on the perceptual part of the problem, that of segmenting the visual stream and interpreting the sub-tasks.

Demonstration has also been used for priming learning, so as to provide an initial policy and thus greatly simplify the learning process. [33] applied this method to a model-based reinforcement learning system in the task of pole balancing by a 7 DOF robot arm.

Work with mobile robots has been more directly related to the underlying mechanisms in biological imitation. [17] employed a direct-mapping imitation mechanism to learn mobile robot maze navigation. [6] used a connectionist architecture to learn communication skills in a similar domain. In [7], this architecture was expanded and applied to imitation of human arm movements with a humanoid avatar.

Learning head movements by imitation has been another popular task [12]. The approach used a visual feature detector, which informed a built-in system that directly mapped a set of possible observed head movements to the robot’s own movements. The visual component of our system is similar to this work, namely the feature detector and the direct mapping we employ between the two kinds of primitives.

4 Our Imitation Model

Next, we describe in more detail our model for imitation using perceptuo-motor primitives. The model consists of five main subcomponents: Tracking, Attention, Learning, Classification, and Actuation. These components are divided into three layers: Perception, Encoding, and Action. Figure 2 illustrates the structure of the model.

4.1 Perception

The first layer, Perception, consists of two components, Tracking and Attention, that serve to acquire and prepare motion information for processing into primitives at the Encoding layer. The Tracking component is responsible for extracting the motion of features over time from the perceptual inputs.

In general, the choice of a primitive set is constrained by the types of information that can be percieved. Naturally, extending the dimensionality and types of information provided by the Tracking component increases the complexity of other components. For instance, if 3D location information is used, the primitive set generated by the Learning component must provide more descriptions to appropriately represent the possible types of motion.

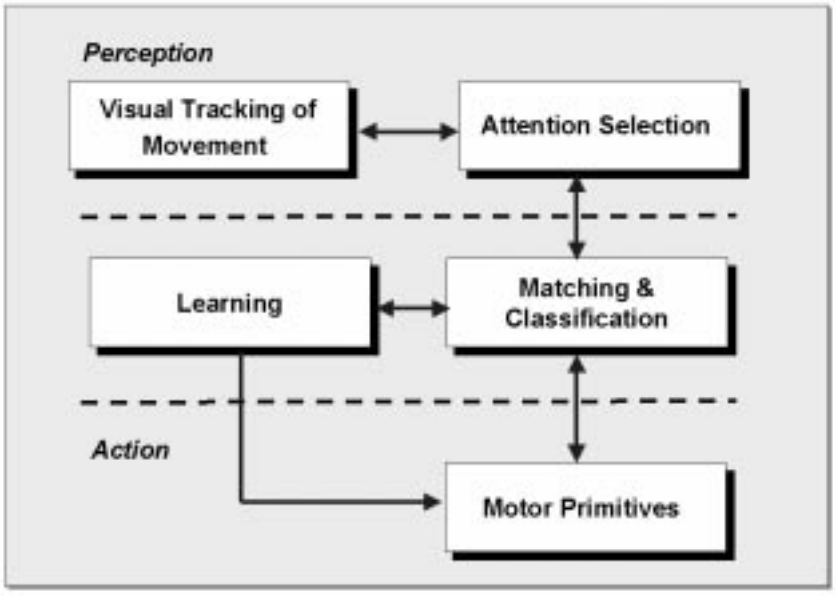

Fig. 2. *Our imitation model.*

The Attention component is responsible for selecting relevant information from the perceived motion stream in order to simplify the Classification and Learning components. This selection process is performed by situating the observed input into meaningful structures, such as a set of significantly moving features or salient kinematic substructures. Various attentional models can be applied [29]; we have also begun work on this problem [20].

4.2 Classification and Learning

The Encoding layer of the model encompasses Classification and Learning, which classify observed motions into appropriate primitives. The primitives, in turn, can be used to facilitate segmentation. The Learning component serves to determine and refine the set of primitives. The Classification component then uses the updated set of primitives for movement encoding. Thus, the classifier itself is a means for segmentating time based on the presence of primitives in the observed motion.

We address several issues regarding the Encoding layer. The first is how to represent motion segments such that they are amenable to classification and learning. The second is that each motion segment requires conversion into the chosen representation and may need to provide a means of segmentation. Additionally, the movement data can have invariances applied to it to account for variations in position and scale. The final issue is that the chosen representation and conversion must be consistent for all components used in this layer.

The Encoding layer provides two outputs to the Action layer: 1) the list of time segments representing a motion (from Classification), and 2) a set of constraints for creating motor controllers for each primitive in the primitive set (from Learning). The list of segments is especially important because it describes intervals in time where a certain primitive is active.

4.3 Action

The final layer, Action, consists of a single component, Actuation, which performs the imitation by executing the list of segments provided by the Classification component. Ideally, primitive controllers should provide control commands independently, which can then be combined and/or superimposed through motor actuation. The design of such controllers is an area of research we are persuing.

As noted above, the motor controller components of the primitives may be manually derived or learned. In both cases, to be general, they must be characterized in parametric form. We have developed a means of learning primitives directly from movement data [14]. We have also addressed the issue of dissimilar kinematics between performers and imitators [7].

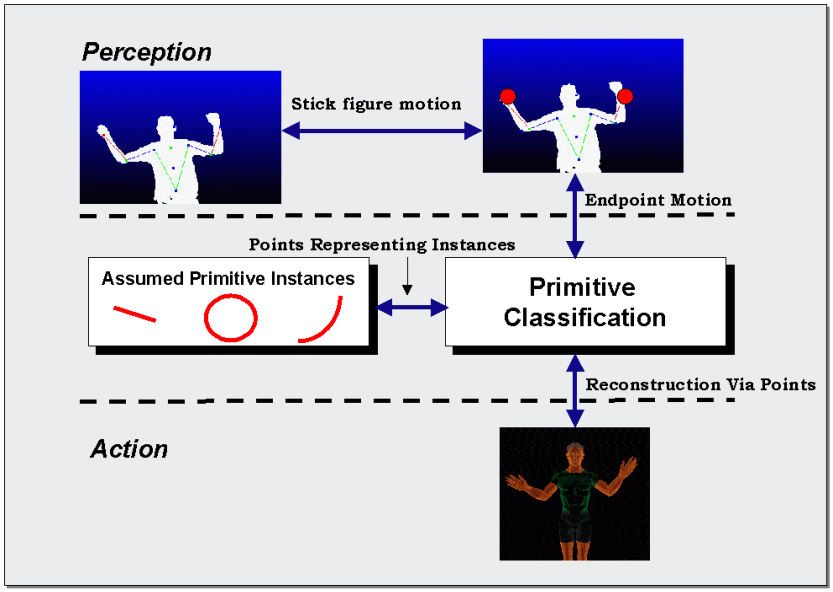

Fig. 3. *The layout of the system implementation of our imitation model.*

5 Imitation System Implementation

Based on the model described above, we have implemented a system for imitating the behavior of a human performer observed through video input by actuating a humanoid torso simulation. This implementation is a constrained instantiation of the imitation model, for purposes of feasibility validation. Figure 3 outlines the setup of our implementation in the context of our imitation model framework.

In general, classification and matching of arbitrary motion to a set of primitives is the foundation for using primitives. Consequently, the current implementation of our model focuses on the design and functionality of the Classification component. Because of the emphasis on classification, the other components in the system were not fully implemented in accordance with the model. Instead, the capabilities of these components either simulated or reasonably limited to achieve their functionality within the model.

5.1 Relating the Model and the Implementation

For our perception system, all of our motion data was acquired using a vision-based motion tracking system we developed [37], which provides a stick figure of the subject's upper body in 2D. In this tracking system, we consider a feature to be a fixed position on the body, such as the shoulder. An example of the output from this tracking system is shown in Figure 4.

Location information in 3D, potentially from motion capture, could be incorporated into the implementation by increasing the dimensionality of the data throughout the model. Feature orientation information could also be included.

Fig. 4. *Top: Snapshots of the output from tracking using [37]. Bottom: Snapshots of our humanoid simulation, Adonis.*

The Attentional component of the implementation assumes only the end-points of an arm are important and extracts only those features. The consideration of end-point motion (i.e. the motion of a hand or an endeffector) alone is based on the work of Matarić and Pomplun [24], demonstrating that humans tend to focus on end-point motion when visually observing arm movement. In our description of this implementation, we consider motion to relate to the movement of the end-point in a Cartesian space. The implementation could use a more flexible attentional module that extracts sets of significantly moving features or determines substructures of rigid movement in a kinematics of a performer. However, as the complexity of the output of this module increases, the complexity for implementing modules for the Encoding layer also increases.

The implementation we describe does not address learning of the primitive set. Instead, we manually derived a set of primitives based on simple 2D shapes to serve as a potential output of a Learning component. Our Classifier component uses vector quantization [4] to quantify windows of motion into a set of perceptuo-motor primitives, described in more detail in Section 5.4. We represent segments of motion as vectors of unit-length positional gradients, with a gradient being the change in position of a feature between frames. The output of the Classifier component is a list of segments. Each segment in the list specifies that one of the perceptual descriptions matches the observed movement over an interval of time. In general, any classifier in this model should return a list of segments describing time intervals where the execution of a given perceptual description has been observed.

In the purest form of the model, the list of segments provided from classification will serve to activate the appropriate motor controller in the set of primitives over the appropriate intervals of time. However, the generation of such controllers is difficult and a subject of our current research. In order to simulate this functionality, the perceputal descriptions of the primitives and the list of classification segments are used in conjunction to generate a via-point trajectory for the end-point of each arm. These via-points are traversed by the end-points of the arms of Adonis using an impedance controller developed by Matarić et al [27].

5.2 Deriving the Primitives

We manually acquired primitive descriptions from actual motion data, described in Section 5.2, using the a minimal 2D geometric set of shapes, consisting of: lines, circles, and arcs. Straight lines are inspired by and used to describe *reaching* movements of the hand. Circular and near-circular movements correspond to closed paths in Cartesian space. Finally, arcs correspond to open circles or portions thereof, thus completing the 2D projection of the possible trajectories. In our implementation, we used various descriptions of these primitives, varying in size, orientation, and direction, in order to facilitate classification. Paths representing 2D trajectories can be well approximated by this set of geometries. Thus, the purpose of this primitive set choice is to demonstrate the feasibility of our model. Ideally, the primitive set would be generated automatically by the Learning module from representative training motion. We address the automatic generation of primitive sets using principle components analysis in [14].

There are several goals associated with this implementation. First, this implementation demonstrates the feasibility of primitive representations for quickly generating approximate behaviors for articulated characters. Second, this implementation helps determine if end-point information alone is sufficient for behavior imitation and to what degree only end-point data is useful. Finally, the system will provide insight into how much utility our impedance controller can provide for developing motor controllers for primtive sets.

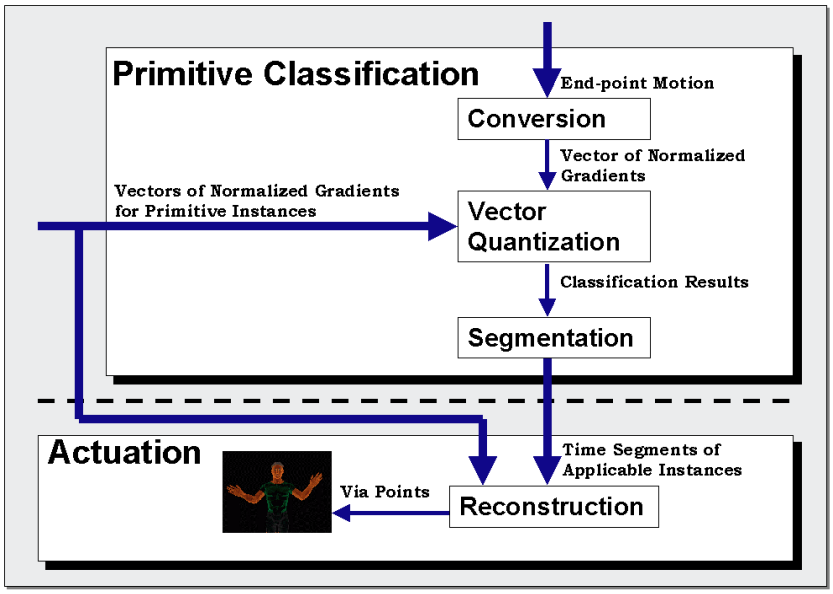

Fig. 5. *The subcomponents and interfaces for the Primitive-Based Classifier.*

5.3 Motion Representation

As stated previously, a common representation of motion is needed for modules in the Encoding layer. The most straightforward representation would simply use the locations across the trajectory of an end-point. However, such a representation would be sensitive to variations in the position and scale of related motions. Instead, motion in this layer is represented as the normalized gradients along the trajectory of the motion Equation 1 describes the conversion of a end-point motion segment, M, into an array of normalized gradients, E, for each frame t.

$$E(t) = (M(t+1) - M(t))/\|M(t+1) - M(t)\| \tag{1}$$

Given that the dimensionality of the gradient is d and the length of the segment is l, the motion segment can be viewed as a point in a ld-dimensional Euclidean space. Consequently, two motion segments, E_1 and E_2, can be compared by computing the distance between their corresponding points in this space space. Although, the comparison operator used in this implementation first computes the set of distances between each corresponding pair of gradients in d-space, D in Equation 2, and then the magnitude of the vector of gradient differences in ld-space computes the distance (or dissimilarity), R in Equation 3, between the segments. If length l_m of the longest motion segment is known, then all segments can be incorporated/projected into the same $l_m d$ dimensional space.

$$D(i) = \sqrt{\sum_{k=1}^{d} (E_1(id+k) - E_2(id+k))^2} \tag{2}$$

$$R = \sqrt{\sum_{k=1}^{l} (D(k))^2} \tag{3}$$

5.4 Primitive-Based Classification

Figure shows a high-level view of the primitive-based classifier. This classifier takes as one input a set of perceptual descriptions, denoted by P, and a stream of end-point motion, denoted by I. In this implemenatation, P represents the descriptions that would be computed by the learning module.

Each perceptual description in P is represented as an array of normalized gradients from an end-point motion trajectory. The motion trajectories needed to create the descriptions for P were generated from by a video sequence of a human performer executing movements representative of the manually-derived primitive set, with each arm simultaneously. We use a human performer in generating the primitive descriptions because a human has roughly the same kinematics and dynamics as our imitator, Adonis. Table 5.4 lists the actions executed by the performer in this training video sequence and their relationship to the set of primitives. The training video sequence was manually segmented in time, isolating each instance of a motion representing a perceptual description. After performing tracking and attentional processes on the training video sequence, the set of trajectories M is formed from the end-point motion segments of each arm, according to time intervals from manual segmentation. Given a unique identifier i (associated with a given time interval and arm) for each motion segment M_i from the training sequence, the corresponding perceptual description P_i is then defined as:

$$P_i(t) = (M_i(t+1) - M_i(t))/\|M_i(t+1) - M_i(t)\| \tag{4}$$

where t is the progression of time represented an offset from the beginning the motion segment. Figure 6 is a plot of the set of normalized gradients used for each perceptual description. The primitive description determination yielded 24 primitive descriptions, containing: 12 for the right end-point, 12 for the left end-point, 12 descriptions of lines, 4 descriptions of circles, and 8 descriptions of arcs.

Action in Training Motion	Extracted Primitive descriptions
Vertical Reach	Upward Line Downward Line
Horizontally Outward Reach	Outward Line Inward Line
Diagonally Outward Reach	Diagonally Upward Line Diagonally Downward Line
Large Circle	Large Circle
Small Circle	Small Circle
Figure Eight	4 Instances of Arcs

Table 1. The correspondences between training motion and manually determined primitive descriptions

Once the perceptual descriptions of the primitives P has been generated, the stream of input end-point motion, I, can now be encoded into the primitive set through classification. The classification algorithm that we use is based on the approaches to vector quantization presented in [4]. The first step in classification is to determine the response of each primitive instance P_i to an input motion I. The response serves as a measure of the presence of a given primitive in the input motion at a given time. This response, denoted $R_i(t)$ for perceptual description i and time t, is computed for the length of the input motion using the comparison operation from Section 5.3. The response $R_i(t)$ is computed by converting the motion of I over the interval t to $t + (l_i - 1)$ into I_c, a window of motion in the Encoding-layer representation, where l_i is the length of the perceptual description P_i, and performing the comparison operation for P_i and the converted window of motion.

The response computation procedure is summarized by the following equation:

$$R_i(t) = \frac{l_i}{\sqrt{\sum_{j=0}^{l_i-1}\left(\sqrt{\sum_{k=dj+1}^{(d+1)j}(I_c(k) - P_i(k))^2}\right)^2}} \tag{5}$$

As stated previously, the comparision operator for two motion segments computes a dissimilarity measure between the segments. The operator is reciprocated to make it a measure of similarity. Furthermore, the operator result is multiplied by the length of the description. This weighting is done to account for descriptions of varying length, assuming that probability of matching a longer description is lower than matching a smaller segment. The measurement capabilities of the responses is shown in Figure 6. The peaks in the responses of the figure accurately correspond the manually chosen segments with the appropriate perceptual description.

The set of response signals R are used to compute the signals that specify the active primitive description Q and maximal response L with respect to time. $R(t)$ specifies the similarities between each primitive description and the incoming motion of I at time t. We would like to classify the motion of I at this time into the description that

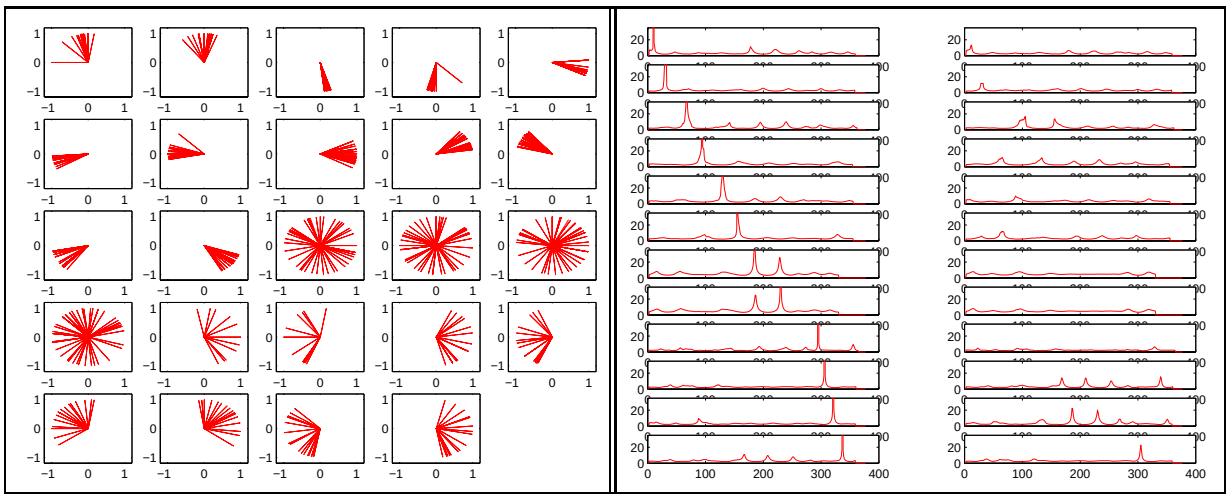

Fig. 6. *Left: The set of normalized gradients in (x,y) space for each of the primitive instances, consisting of 12 line, 4 circle, and 8 arc instances in left to right order. Right: The response $R(t)$ over time for the movement of the left end-point against the manually determined primitive set. The left and right columns present the response to the left and right arm perceptual descriptions, respectively. The first six rows represent line primitive instances, the next two rows are circle instances, and the last four rows are arc instances.*

maximizes the similarity measure. More specifically, $L(t) = R_x(t)$ and $Q(t) = x$, such that $R_x(t) = \texttt{max}(R_i(t))$ for all $i \in P$. The result of computing Q and L for the training motion are shown in Figure 7.

A connected components routine is used to group similar and adjacent classification results, based on Q. Specifically, two adjacent instances of time t and $t+1$ are grouped together if $Q(t) = Q(t+1)$. This grouping indicates the interval over the execution of a movememt matching one of the descriptions has occured. The result of this grouping operation is a segmentation of the input motion in time, expressed as a list of segments U. Each segment contains its start and end time (relative to I), and the index of the description executed in the segment. From figure **??**, the Q signal may appear cluttered and to provide no meaningful segmentation result; this occurance is the result of poor matches during times of transitions between valid segments.

We approach the problem of determining segments of importance from U by computing a weighting value corresponding to the confidence of each segment. A threshold is then applied to the weighting values based upon a percentage of the maximum weight. With little effort, this approach has yielded satisfactory results, but is clearly not the most optimal or elegant solution. For each segment $k \in U$, the weighting for U_k is computed as the length of the segment times the sum of response values of L within time interval of the segment,

$$U_k.\texttt{weight} = (1 + U_k.\texttt{end} - U_k.\texttt{start}) \sum_{j=U_k.\texttt{start}}^{U_k.\texttt{end}} L(j) \tag{6}$$

The maximum segment weight is stored as m. The segment U_k is then considered valid if its weight is greater than am, where a is a fractional threshold and $0 < a \leq 1$. Through the elimination of invalid segments, a new list W of validated segments is formed. Figure 11 shows a stair step plot of the list of valid segments computed from the training motion, described in the Section 5.4. Figure 10 shows a comparison of valid segments with their individual reconstructions from our implementation.

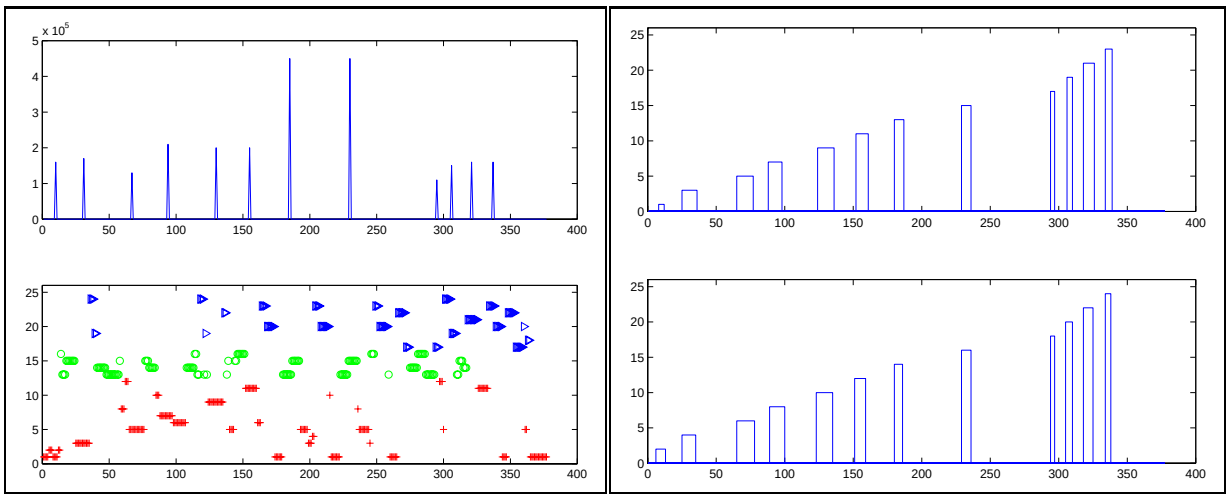

Fig. 7. *The results from classification of the training motion for the right and left arms. Left: plots of L(t) (above) and Q(t) (below) corresponding to the maximum valued response and active perceptual description for the left end-point with respect to time. In the plot of Q(t), Q(t) identifies the index of the active description, these indices are separated into crosses for lines, circles for circles, and triangles for arcs. Right: plot of Q(t) for the left and right arms after thresholding for valid segments.*

5.5 Reconstruction and Actuation

The execution of a motion is performed by invoking motor controllers within the primitives, and using control commands from the segments resulting from classification. In our current implementation, we simulate this functionality by overlaying primitive descriptions to reconstruct the end-point trajectory.

The basic procedure for this reconstruction is to place each sequence of normalized gradients at their appropriate position in time relative to the input motion I, given by W, and normalizing the sum of the set of gradients that occur at any instance of time. The summation of gradients is used to handle temporal overlapping of segments. Given an initial position for the input motion, a running sum of these overlayed gradients is used to generate a trajectory of end-point positions. After transforming these locations from image coordinates to the proper scale for the humanoid simulation, this trajectory of via-points can then be given to the impedance controller of Adonis for end-point traversal. From Figure 12, it can be seen that this approach to reconstruction appears to capture the essence of the input motion for the training sequence, i.e. the various executions of lines, circles and arcs (in the form of a figure-eight) can be easily observed. However, the scale and translational aspects of the input motion are not properly accounted for in this procedure. These aspects of the motion were lost when invariances to scale and translation were introduced for the Encoding layer.

This basic procedure was modified to incorporate limited information from the input motion stream to reintroduce appropriate positional and scaling properties to the reconstructed trajectories. Each segment in W is associated with an interval of time. For a given segment in W_i, the position at the beginning of its interval, $I(W_i.\texttt{begin})$, and the bounding box of the positions over the interval were computed. The starting position is used as a translation offset for the gradients in the segment. This procedure would be similar to reconstructing a function $f(x)$ using the Taylor approximation $f(0) + \Delta x f'(x)$. We assume the frequency of the sampling from the tracking system is fast enough (at least 10 frame/second) that subsequent terms in the Taylor series will be insignificant. The length and width of the computed segment bounding box is used to determine a scaling factor for each gradient in the the associated segment. By injecting this information, we can reconstruct a useful reconstruction of the input motion trajectory, as shown in Figure 8.

The general form of the previously described overlaying operations is:

$$T(i) = \frac{\sum_{j=1}^{|W|} ((S_{ij} B_j) + P_i)}{|W|} \tag{7}$$

where T is a 1 by l_I vector of gradients for reconstruction, S is a $|W|$ by l_I matrix of gradients, B is a $|W|$ by 1 vector of gradient scaling factors, and P is a 1 by l_I vector of positional offsets.

As stated previously, the reconstruction vector of gradients T can be converted into via-points using an unwraveling procedure. This procedure simply maintains a running sum of the gradients and recording the location resulting from adding each gradient to the sum. The reconstruction vectors for each end-point was transformed into via-points using the proportions of the simulation [27], specifically the ratios of the lengths of each arm and positions of shoulders. A via-point traversal controller was used to execute the reconstructed motion. The use of Adonis for actuation is described in detail in [38]. In brief, this control strategy allows the movement of an end-point to a desired (or equilibrium) point in Cartesian space by generating an end-point force based on a virtual spring and damper model applied between the end-point and the equilibrium point. This type of control serves to avoid costly inverse kinematics computation. Two parameters are required for each arm: a desired location for the endeffector and a desired orientation for the upper arm. The reconstruction via-points provide the desired trajectories for each hand of Adonis. The desired upper arm orientations are set at a constant posture to be perpendicular to the body and parallel to the line between the shoulders.

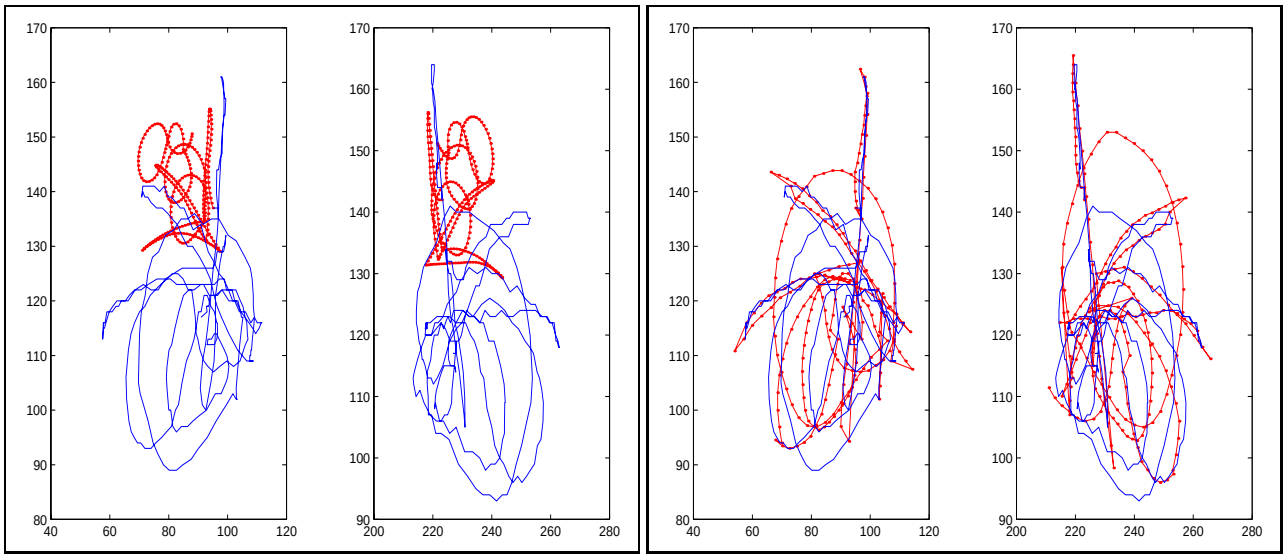

Fig. 8. *Reconstructed via-point trajectories against the training end-point trajectory in (x,y) space. The reconstructions using the basic reconstuction procedure and reconstuction with initial segment positions and bounding box scale factors.*

6 Experimentation and Evaluation

Using a video tracking system we developed, described in [37], six input motions, inspired from athletics, aerobics, and dancing, were captured. These captured motions are described in Table 5.5. The input motions were then given to our implementation for imitation. The following subsections describe four different methods we used for evaluating the results of our imitation system.

Error for Entire Motion The first idea that comes to mind for evaluating the effectiveness of the system is to compute the difference between the input and the reconstructed trajectories. This difference can be computed by the Root Mean Squared Error (RMS), described in [9]. RMS error serves as an estimator for the standard deviation between the two trajectories. In this evaluation procedure, RMS error was computed for the left and right endpoints of each input motion from our test motion set against the associated trajectory generated from our imitation system and two distortions of the original input motion. The distortions were generated by adding random noise

Motion Name	Alias	Description
Throwing	”Sky Hook”	Performed by moving one end-point at near full arm extension from waist level to a location above the head. Commonly used in Basketball and many other sports.
Aerobic1	”Arm Thrusts”	Performed by starting with the end-points at waist level and oscillating them inward and outward while gradually moving upward. Taken from [34].
Aerobic2	”Alternating Arm Waves”	Performed by having one end-point at the height of the head and the other at waist level and continuously alternating end-point locations with brief pauses. Taken from [34].
Egyptian	”Walk Like an Egyptian”	Performed by placing one end-point at shoulder height and one at waist height with bent elbows. The end-points move in the same direction with similar velocities. A popular dance from the 1980’s
Lin Roof	”Raising the Roof”	Linear version of raising the roof. Performed by linear end-point oscillations above the shoulders. Typically used as a form of celebration by tired athletes.
Cir Roof	”Raising the Roof”	Circular version of raising the roof. Performed by circular end-point motions above the shoulders. The result of raising the roof for those with less motor coordination.

Table 2. Descriptions of the input motion set.

offsets to the x and y components of the positions along end-point trajectory of the motion. Each invokation of random noise generator used provides a real number y, such that $0 \leq y \leq 1$. The level of distortion d is used to compute the distorted motion, such that $I_d(t) = dy(t) + I(t)$, where I is an input motion and I_d is a distortion of I at level d. For the evaluation of the trajectories, we used distortions of $d = 5$ and $d = 15$. These results are shown in Figure 9.

Error Per Segment The error for the entire motion trajectories is caused by a combination of the error of reconstructing the individual segment and the result of reintroducing aspects of translation and scale. Thus, another evaluation measure would be to compute the mean of the RMS error for the reconstruction of each individual segment from the classifier, W, with the trajectory of I over the associated interval of time. Figure 10 shows the reconstructions of the individual segments for the training motion. Figure 9 shows the results from computing mean RMS error for each motion in the input set and for distortions of the training motion at increasing levels of noise. The throwing motion incurred the worst reconstuction for both RMS measures. Yet, considering the plot of in Figure 12, however, it appears visually adequate.

Segmentation Validity Instead of using a distance metric to evaluate reconstructions, each segment can be evaluated visually, as in Figure 10, to determine if the segment is actually valid. For the purposes of segmentation, a valid segment is a segment correctly identified by the classifier. The classifier returns some amount of false positives (incorrectly returned segments) and false negatives (undetected actual segments). For this evaluation procedure, we manually determined whether each returned segment for a motion is a true or false positive. The result of accumulating these determinations is shown in Figure 9. As expected, the training motion returns all true positives because it was used for extracting the primitives. In addition, the throwing motion imitation performed well in this evaluation, contrasting the results from RMS error. In future work, we will use this evaluation procedure with impartial human evaluators to test motion believability.

Varied Levels of Noise Another evaluation procedure we used involve introducing distortions of the training motion. The imitation system was presented with input motions with increasing levels noise; the segmentation is shown in Figure 11 where the manual segmentation of the original motion is shown in the upper left box. This segmentation has a stair step structure because the perceptual descriptions are indexed by the order they appear in the training sequence. This structure is detected for the undistorted training motion. After applying a level 1 distortion, a few false negatives and false positives occur, but the structure of the segments is still apparent. As shown in the plots that follow, adding noise gradually removes the ability to return true positives and increases the number of false positive, thus distorting the structure of the segmentation. This degradation of the detected

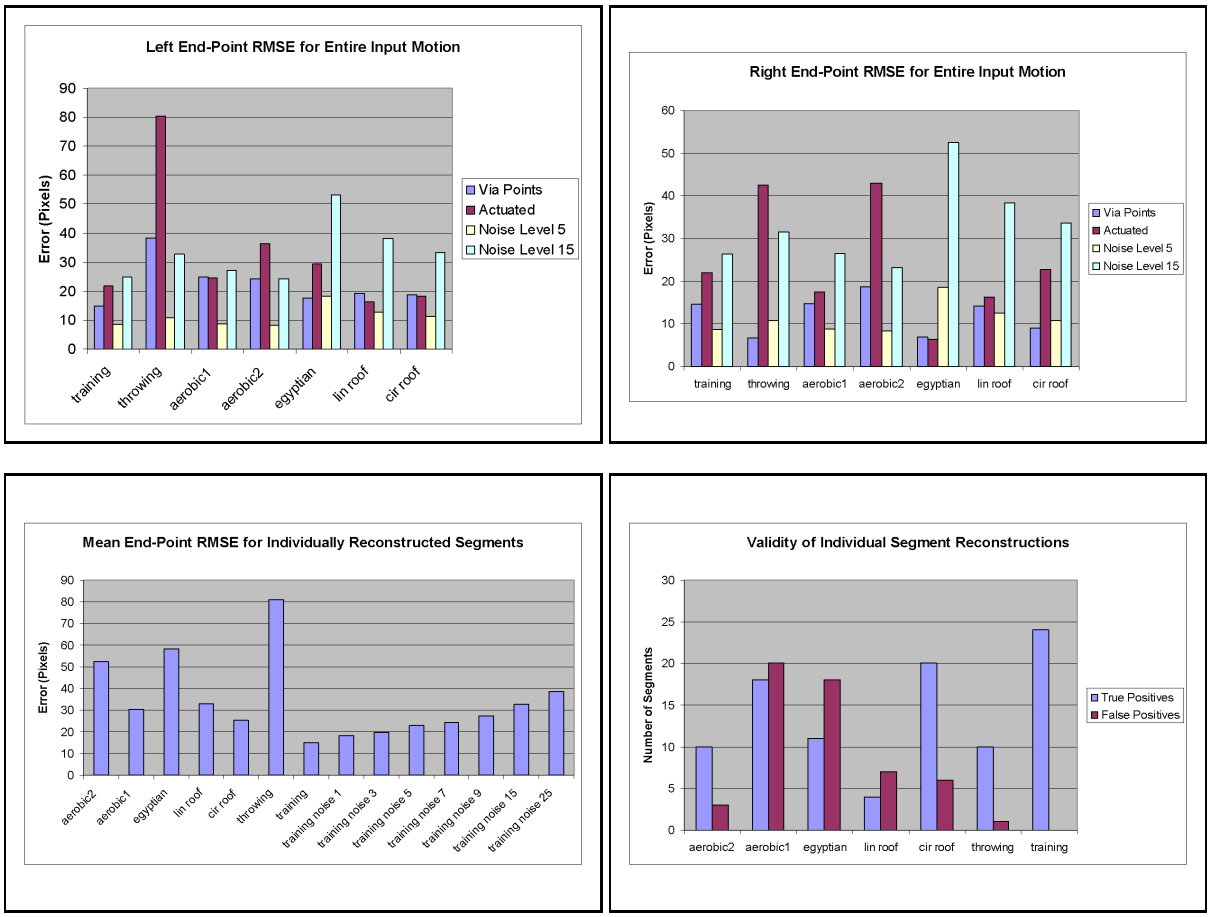

Fig. 9. *Plots of evaluations for entire motion RMS error, mean individual RMS error, and individual segment validity.*

segment structure is also indicated by the consistent increased in Mean individual segment RMS error shown in Figure 9. This increase in RMS error for greater distortions does not appear to reflect the magnitude that the segment structure has decayed.

7 Discussion

We have described a model for imitation and a constrained implemented validation of that model. As stated in Section 5, there were several objectives associated with this implementation that merit discussion.

Does this implementation demonstrate the ability for a primitives-based system to quickly approximate observed behaviors? From the classification and reconstruction results, shown in Figure 12, we believe that our current imitation system exhibits the potential for control based on perceptuo-motor primitives. However, no solid metrics for evaluating an imitator's performance are currently available, so the quality of the resulting imitation is still difficult to assess. Our goal is *believable* imitation. We presented three evaluation procedures applied to a set of input motion. The two procedures based on the automatically computed Root Mean Squared Error provide some insight into our system's ability to imitate. Such distance-based metrics, however, inherently concentrate on the exact reconstruction of the input motion and limit the ability to incorporate behavioral similarities of the motions. For example, reconstruction of the throwing motion was evaluated as poor by the RMS metrics, but generated a very believable reconstruction from examining the reconstruction in Figure 12. Furthermore, the validated segment reconstructions yield good imitation trajectories, but are evaluated poorly by distance-based metrics due to the difference in scaling. The analysis of segment validity through the determination of false positives and negatives has the potential for being a more reliable metric when applied by impartial human evaluators. However, this method

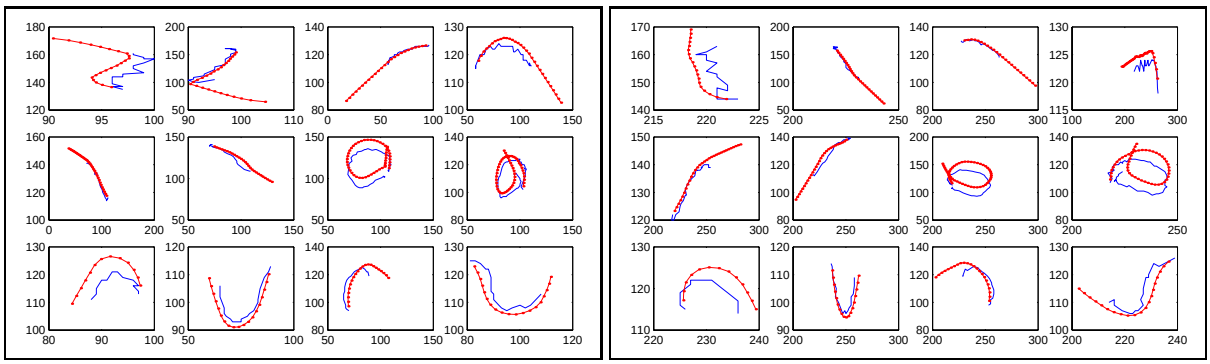

Fig. 10. *Reconstructions of individual valid segments in (x,y) space for the training motion using superimposition throughout for the right and left end-points. The solid and dotted curves plot the paths of the original and reconstructed motions, respectively.*

incurs the imprecision that accompanies empirical generalization. We are working on methods for evaluation [27], [31].

The limitations of segmentation have a direct impact on the quality of imitation and thus on the use and outcome of any evaluation procedure. The segmentation validation procedure we presented provides no guarantee of correct validation, but provides a practical means for eliminating irrelevant classifications. One problem with this validation procedure is that thresholding of segment weights is performed relative to the weight of highest weighted segment. This limitation makes it difficult to segment motions like the "Egyptian", in which the right end-point does not move significantly over the course of the motion. As a result, none of the segments are relelvant. However, there is no standard to eliminate all of these segments and, thus, irrelevant segments are identified as valid. A part of this problem could be addressed by introducing a primitive that represents idle, static, or stopped motion. Preferably, primitives should be generated in such a way that they can be parameterized or stereotyped to account for a significant range of motion.

We use a simple connected components routine in the segmentation process to group time intervals based on adjacency and identical classifications. Consequently, a segment may actually represent several occurrences of the same perceptual description or several segments could represent an occurrence of a single perceptual description. To account for this, another mechanism could be implemented to perform splitting of dissimilar segment parts and merging of like segments. Current work by Pomplun and Matarić [31] has begun to address mechanisms for split and merge motion segmentation. The detection of superimposed segments may make split and merge processing unnecessary.

Detecting and actuating superimposed primitive descriptions is a great challenge we face in the full realization of our imitation model. In our current implementation, we simulate the behavior of superimposed actuation of through overlaying, but do not address detection of superimposed descriptions. We have begun to consider approaches for this detection using multi-resolutional techniques and individual segmentation of response signals.

Is end-point information alone enough for behavior imitation? The described implementation focuses on human arm movement, where end-point information appears sufficient for most tasks we have considered. In general, end-point information is sufficient when the system is given an appropriate set of primitive descriptions. However, we can make no substantial claims for more complex kinematic structures where end-points with possibly conflicting control commands. Our implementation could be easily adapted to perform such imitation capabilities. However, we will addresses the problems that arise from incorporating multiple end-points through continued work on the Attentional component of our model. Furthermore, we will explore the use of primitives for motion involving object manipulation and reactions in physical environments.

Is impedance control a viable basis for primitive controllers? We chose impedance control [13] for reaching movements based on inspiration from human movement studies. The use of via-points with impedance control provided acceptable results. The implementation for this control system was relatively simple, but provided mod-

erately accurate traversal of the reconstruction trajectories. Because exact reconstruction and actuation is not our goal, the performance of this control system was sufficient. For primitive controllers in general, this approach is likely not the best means for control because due to the nature of the spring and damper model. This approch would be well suited for primitives that are linear or near-linear. In addition, we suspect that impedance control would be able to provide a means for combining control commands for superimposing motion. However, the current implementation handles only sequencing of primitives. This can be partially attributed to the abstract nature of the minimal 2D geometric primitive set and the method used for classification. The learning of primitive motion sets and superimposed classification are both subjects of our current research.

8 Summary and Continuing Work

We have described a primitives-based method for acquiring motor skills by imitation. This approach is based on neuroscience and cognitive science data on the organization of perceptual and motor systems, and our interpretation of those results for the specific purposes of imitation; the specific methods we describe are not intended to model any biological processes. The described model presents the current status of our on-going efforts in imitation learning, and we are improving it along several lines. We are pursuing the development of a more general, 3D version of the video motion tracking mechanism. We are also currently evaluating two different methods for automatic extraction of the perceptual primitives, as an alternative to the hand-selected and hand-coded ones we presented here [14]. Our continuing work will also address the mapping between the perceptual and motor components of the primitives. Our past work has developed several alternatives for motor control and, with the perceptual results presented here, we are looking toward a richer solution. Finally, we are applying the presented imitation model to a large battery of more general training data in order to learn various sequences and superpositions of primitives as encodings for complex skills such as dancing.
For videos comparing the human tracking and actuated movement movement, and more information about the vision system, visit the URL:
`http://www-robotics.usc.edu/~agents/imitation/`

9 Acknowledgements

This work is supported by the USC All-University Predoctoral Fellowship to Chad Jenkins, the NSF Career Grant IRI-9624237 to Maja Matarić, and the Fulbright Foundation Grant to Stefan Weber. We would also like to thank Sarah Loftus for her proofreading efforts and the use of her aerobics video.

References

1. M. Arbib. Schema theory. In S. Shapiro, editor, *The Encyclopedia of Aritificial Intelligence, 2nd Edition*, pages 1427–1443. Wiley-Interscience, 1992.
2. Ronald C. Arkin. Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 264–271, Raleigh, NC, April 1987.
3. Ronald C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, Massachusetts, USA, 1998.
4. S. Arya and D. M. Mount. Algorithms for fast vector quantization. In *The Proc. Data Compression Conference*, pages 381–390. IEEE Computer Society Press, 1993.
5. C. G. Atkeson. Memory-based approaches to approximating continuous functions. In *Proceedings, Sixth Yale Workshop on Adaptive and Learning Systems*, 1990.
6. A. Billard and G. Hayes. Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1):35–64, 1999.
7. A. Billard and M. Matarić. A biologically inspired robotic model for learning by imitation. In *Autonomous Agents*, pages 373–380, 2000.
8. E. Bizzi, F. Mussa-Ivaldi, and S. Giszter. Computations underlying the execution of movement: A biological perspective. *Science*, 253:287–291, 1991.
9. R. K. Bock and W. Krischer. *The Data Analysis Briefbook*. Springer, Berlin; New York, 1998.
10. M. Brand and A. Hertzmann. Style machines. In *The Proc. of ACM SIGGRAPH 2000*, pages 183–192, 2000.
11. C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recoginition*, 1997.
12. J. Demiris, S. Rougeaux, G. Hayes, L. Berthouze, and Y. Kuniyoshi. Deferred imitation of human head movements by an active stereo vision head. In *Proceedings of the 6th IEEE International Workshop on Robot Human Communication (RoMan97)*, Sendai, Japan, 1997. IEEE Press.

13. T. Flash and N. Hogan. The coordination of the arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 7:1688–1703, 1985.
14. A. Fod, M. Matarić, and O. Jenkins. Automated derivation of primitives for movement classification. In *Proc. of First IEEE-RAS International Conference on Humanoid Robots*, 2000.
15. Simon F. Giszter, Ferdinando A. Mussa-Ivaldi, and Emilio Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.
16. M. Haruno, D. M. Wolpert, and M. Kawato. Multiple forward-inverse models for human motor learning and control. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.
17. G. Hayes and J. Demiris. A robot controller using learning by imitation. In *Proceedings of the International Symposium on Intelligent Robotic Systems*, pages 198–204, Grenoble, France, 1994.
18. G. E. Hovland, P. Sikka, and B. J. McCarragher. Skill acquisition from human demonstration using a hidden markov model. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2706–2711, 1996.
19. K. Ikeuchi, M. Kawade, and T. Suehiro. Assembly task recognition with planar, curved, and mechanical contacts. In *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993.
20. O. Jenkins. An attention selection mechanism for moving human figures. Technical Report IRIS-00-381, Institute for Robotics and Intelligent Systems, University of Southern California, 2000.
21. M. Kaiser. Transfer of elementary skills via human-robot interaction. *Adaptive Behavior*, 5(3-4):249–280, 1997.
22. Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.
23. M. J Matarić. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2-3):323–336, 1997.
24. M. J Matarić and M. Pomplun. Fixation behavior in observation and imitation of human movement. *Cognitive Brain Research*, 7(2):191–202, 1999.
25. Maja J Matarić. Sensory-motor primitives as a basis for imitation: Linking perc eption to action and biology to robotics. In *Imitation in Animals and Artifacts*. The MIT Press, 2000.
26. Maja J. Matarić and Matthew J. Marjanović. Synthesizing complex behaviors by composing simple primitives. In *Self Organization and Life: From Simple Rules to Global Complexity, European Conference on Artificial Life (ECAL-93)*, pages 698–707, Brussels, Belgium, 1993.
27. Maja J Matarić, Victor B. Zordan, and Matthew Williamson. Making complex articulated agents dance: An analysis of control methods drawn from robotics, animation, and biology. *Autonomous Agents and Multi-Agent Systems*, 2(1):23–44, March 1999.
28. Ferdinando A. Mussa-Ivaldi, Simon F. Giszter, and Emilio Bizzi. Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences*, 91:7534–7538, 1994.
29. Harold E. Pashler. *The Psychology of Attention*. The MIT Press, 1999.
30. R. P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, MA., 1981.
31. M. Pomplun and M. Matarić. Evaluation metrics and results of human arm movement imitation. In *Proc. of First IEEE-RAS International Conference on Humanoid Robots*, 2000.
32. G. Rizzolatti, L. Gadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. Cognitive Brain Research, 3:131–141, 1996.
33. S. Schaal. Learning from demonstration. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 1040–1046. The MIT Press, 1997.
34. K. Smith. Kathy smith's body basics. Aerobics Video, 1985.
35. G. Tevatia and S. Schaal. Inverse kinematics for humanoid robots. In *IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000.
36. Yasuhiro Wada, Ysuharu Koie, Eric Vatikiotis-Bateson, and Mitsuo Kawato. A computational theory for movement pattern recognition based on optimal movement pattern generation. *Biological Cybernetics*, 73:15–25, 1995.
37. S. Weber. Simple human torso tracking from video. Technical Report IRIS-00-380, Institute for Robotics and Intelligent Systems, Universit y of Southern California, 2000.
38. S. Weber, M. Matarić, and O. Jenkins. Imitation using perceptuo-motor primitives. In *Autonomous Agents*, pages 136–137, 2000.

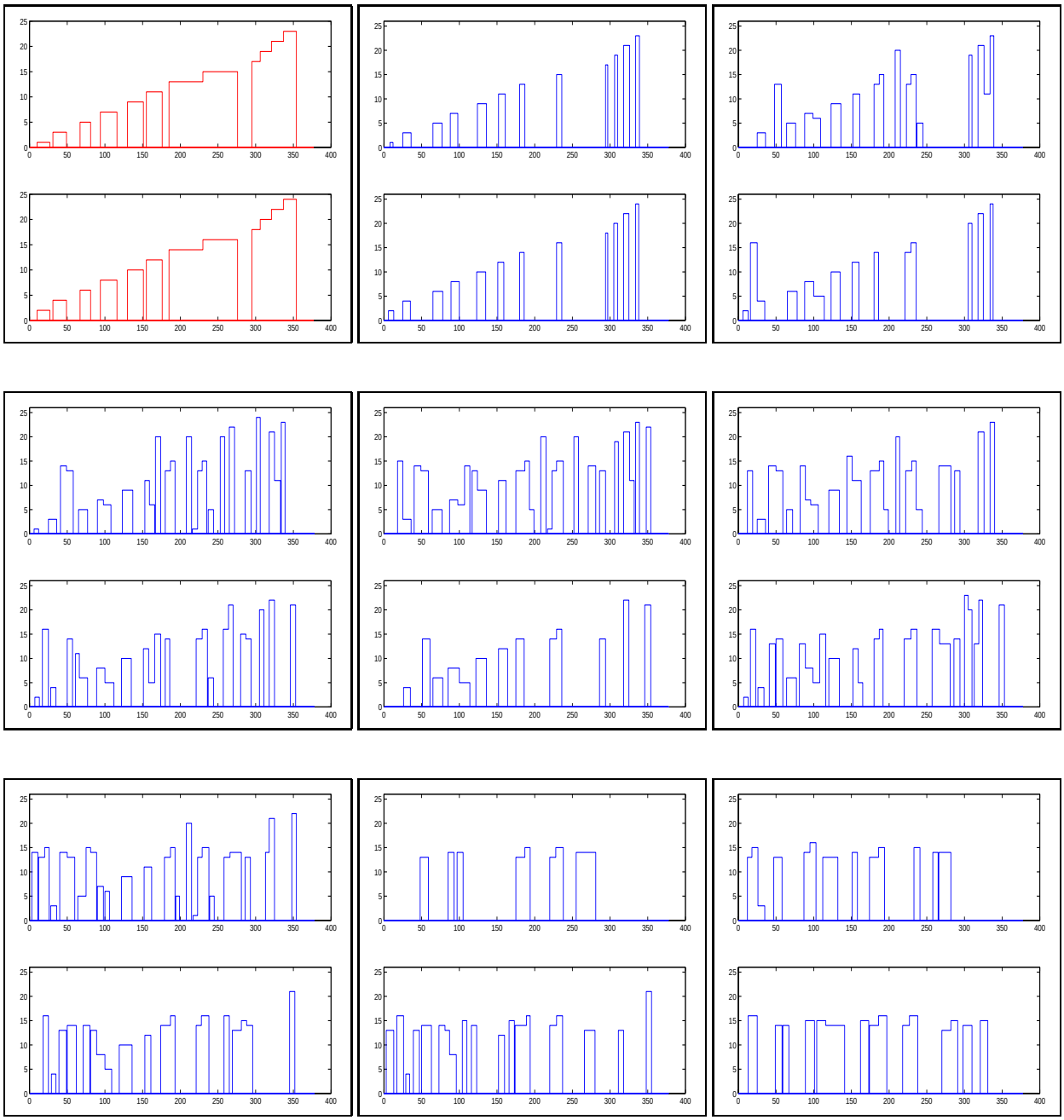

Fig. 11. *The upper left plot shows the manual segmentation of the training motion into the primitive descriptions for the left and right end-points. The succession of plots from left to right and top to bottom show the result after segment validation for distortions of increasing noise level corresponding to 0, 1, 3, 5, 7, 9, 15, and 25.*

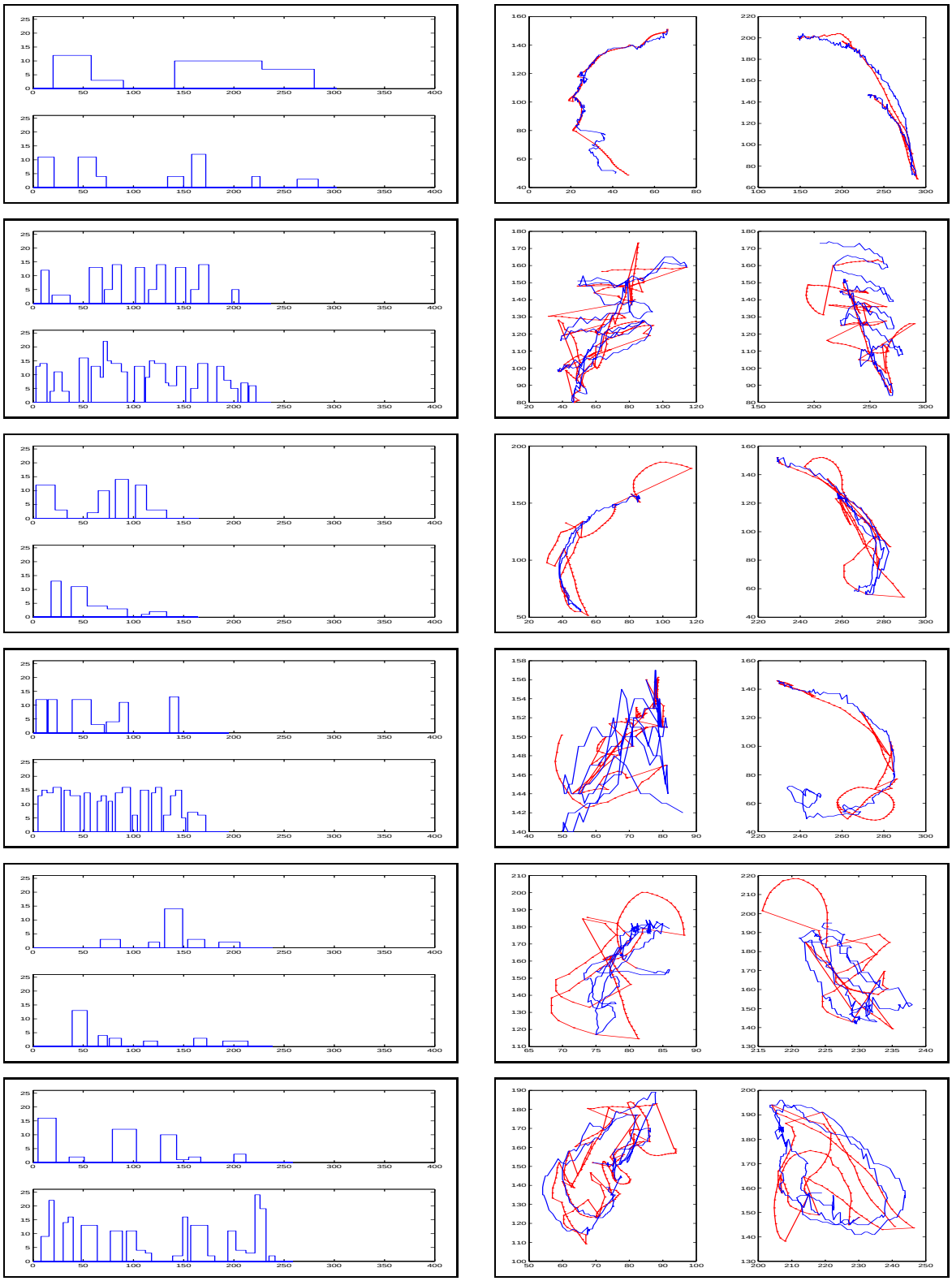

Fig. 12. *The rows of this figure provide the segments that result from classification and the reconstructed end-point trajectories for our set of test motion. The y-axis of the segmentation plots are the indices of the perceptual descriptions. The rows show the results for throwing, aerobic arm thrusts, alternating waves, "egyptian", and two versions of "raising the roof".*