Data-driven Skill Derivation from Natural Human Motion

Odest Chadwicke Jenkins, Chi-Wei Chu, and Maja J Matarić

Interaction Lab, Center for Robotics and Embedded Systems, Computer Science Department, University of Southern California, Los Angeles CA 90089, USA,
cjenkins,chu,mataric@usc.edu,
WWW home page: `http://robotics.usc.edu/~agents`

Abstract

We present a data-driven methodology for automatically creating skills from a human demonstration for a humanoid robot. Our learning from demonstration (or performance) methodology consists of two components: markerless motion capture and behavior derivation from motion. The motion capture component uses images from multiple calibrated cameras to estimate both the kinematic model and motion of an uninstrumented subject. The behavior derivation component estimates the underlying spatio-temporal structure of a subject's kinematic motion such that underlying behaviors can be identified and constructed into forward models. The forward models can then be used for classification, prediction, and synthesis of motion for robot/agent control as a substrate for robot programming by demonstration.

1 Introduction

A variety of approaches to robot control have been proposed, including planning, behavior-based, reactive, and hybrid architectures. All of these either require or can be enhanced with a *skill-level behavior repertoire*. This repertoire serves as both a modularization of motor capabilities and a parsimonious interface for task-level control, as a compact and purposeful set of basic capabilities. We view this repertoire as a set of parameterized programs for a mechanism at a lower motor/dynamics layer without task-dependent or semantic knowledge. The skill repertoire provides a set of capabilities for a higher task-level control mechanism that incorporates world semantics and task objectives.

The utility of a skill repertoire for robot control is greatly influenced by the selection of behaviors included in the repertoire. Determining an appropriate set of skill behaviors can be done manually or automatically. Initially, a manually developed skill repertoire may be more intuitive, but is susceptible to design and scalability problems. Additionally, such a repertoire is at most a hypothesis of actual human capabilities or a guess at a robot's necessary capabilities. An automatically derived repertoire could develop a robot's capabilities from repeated trial and error of actuation possibilities, but the space of control possibilities is likely too large for this to be feasible.

Our approach aims to provide *motion behavior vocabularies* as skill repertoires that are automatically estimated from human performance, while allowing for manual refinement. To derive behavior vocabularies we use unsupervised learning techniques to identify behaviors present in the motion of a human subject. This approach relies on two main features: the ability to capture human performance kinematically and the ability to find the underlying spatio-temporal structure in human motion. We recently developed two *model-free* unsupervised techniques for providing these capture and structure estimation functionalities. The first technique, *Kinematic Model and Motion Capture (KMMC)* [5], allows for the capture of *natural subject motion* by estimating both a subject's kinematic model and motion from calibrated passive sensors (e.g., cameras). The second technique, *Performance Derived Behavior Vocabularies (PDBV)* described in [14], uses motion as input to automatically derive a behavior vocabulary and corresponding motion controllers.

In this paper, we describe these two techniques within our greater data-driven methodology for automatically producing behavior vocabularies from human performance. This "performance-to-vocabulary" methodology emphasizes the inherent connection between techniques for capturing "natural" motion by methods similar

in aim to KMMC and deriving behaviors present in captured motion similar to PDBV. We describe the implementation of these two techniques and motivate their usage to learn robot skills from human performance as a substrate for robot programming by demonstration. Although, we typically cast our performance-to-vocabulary methodology in the context of the NASA Robonaut [1], the same methodology could be used with other primitives-based control and imitation models for humanoid [18, 25, 2] and mobile [20] robots .

2 Motivation and Overview

The main purpose of our methodology is to endow a humanoid robot with a useful set of skill-level capabilities. Examples of such skills include reaching, waving, grasping, batting, etc. Instinctively, manual selection and implementation of these skills may appear to be the more straightforward route, especially when focusing on a specific task with available domain knowledge (e.g., athletics). While this holds true in the short term, we believe that the utility of a manually-derived repertoires will decline with time due to several factors, including manual design miscalculations, scalability for new and modified behaviors, the complexity of controller implementations, and the absence of definitive ground truth about human behaviors.

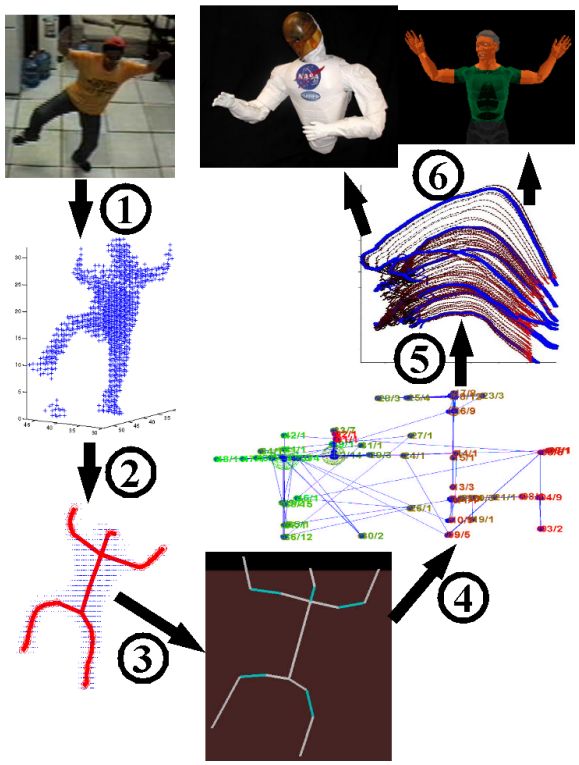

Figure 1: *Flowchart of our performance-to-vocabulary methodology. 1) Voxel carving creates 3D volumes of moving subject. 2) NSS skeletonizes volume sequence. 3) KMMC estimates kinematic model and motion from subject volumes and skeletons. 4) PDBV produces behaviors from captured motion data. 5) A flow field forward model is created from each behavior. 6) Forward models are used to issue control commands to a robot (e.g., NASA Robonaut or Adonis simulation).*

Instead, we automatically derive a skill repertoire from human demonstration, while allowing for manual refinement. In our approach, we assume that human motion is structured through a set of low-level primitives and higher-level task goals.

Using global spectral dimension reduction [29], the underlying structure of a kinematic motion is estimated as clusters of motion, or *primitive feature groups*. Each group is a set of motion segments, trajectories in joint angle space, with a common theme. As a parameterization mechanism, interpolation is used within each primitive feature group to produce new motions that are variations on the group theme. By densely sampling within a primitive feature group, the cluster trajectories form a velocity field that can be used as a nonlinear dynamical system.

A nonlinear dynamical system is a forward model with perceptual and control capabilities, defined as a *perceptual-motor primitive* [17, 18]. In the context of derived behavior vocabularies, forward models are referred to as *primitive behaviors*. A primitive behavior is a time-update model for proposing the next posture of a kinematic system given the current posture. This time-update (or next-state) function can be used to produce control trajectories for a robot in joint space or predict future motion of a human performer. Additionally, every forward model makes predictions for a performer's motion. Thus, classification of motion can be performed by selecting the forward model with the best prediction over an interval of time. With the ability to classify motion and synthesize control trajectories, a vocabulary of primitive behaviors is a substrate for imitating the structure of previously unseen motion and allowing for variations on the motion to be produced.

Our perspective on deriving behavior vocabularies begs the question of how motion data should be collected to produce usable skills. Most commercially available motion capture systems are suited to collecting *constrained*

subject motion, where a performer's movements are scripted or subject to predetermined constraints. Such capture systems typically require conspicuous instrumentation in a prepared environment. Collecting motion data from such a system introduces a bias into the derived behavior vocabulary. Instead, we aim to collect *uninstrumented, natural subject motion* which, we hypothesize, will yield a rich vocabulary of underlying skills.

Recent computer vision research has addressed *markerless motion capture* for uninstrumented subjects [19, 10, 4, 8], but most of these techniques are *model-based* in requiring an a priori kinematic model. These methods do not explicitly handle non-standard kinematics due to object manipulation, multi-person interaction, or non-standard subjects (e.g., amputees). In an effort to provide *model-free* markerless motion capture, we have developed KMMC to estimate the kinematic model and motion as sensed from a set of calibrated cameras. In KMMC, a volumetric representation is built from sensor data and skeletonized via spectral dimension reduction. The volumes and skeletons produced from a sequence of motion are used to build a common kinematic model across frames with frame-specific joint angles.

3 Related Work

The goals of our methodology compliment those of the Verbs and Adverbs (V-A) approach to building behavior vocabularies [24]. In the V-A approach, a skilled user manually: *i)* segments scripted motion data, *ii)* partitions motion segments into exemplars of "verbs", *iii)* positions exemplars of a verb in an "adverb" space, and *iv)* expresses a "verb graph" for smooth and valid transitions between verbs. If constructed properly, V-A adverb spaces provide a human intuitive parameterization of the verb through interpolation. Non-skilled users should be able to select parameters and sequence performances of verbs such that a desired motion results. Additionally, the V-A vocabulary can be used to produce motion at run-time for interactive environments.

Our methodology can be used to automatically derive V-A behavior vocabularies from real-world human motion. Our methodology trades off the ease in creating the vocabulary for the loss of human-intuitive behaviors with semantic parameters. However, our automatically derived vocabularies maintain the flexibility of V-A vocabularies that can be manually refined after creation. In addition, our methodology aims for vocabulary construction to be accessible to users without technical prowess by automating both the performance-to-capture and capture-to-vocabulary procedures. Of other methods proposed for learning primitive behaviors from motion performances, the method closest to PDBV is *motion textures* [16]. Motion textures take as input a single motion performance, containing performances of various unknown behaviors, and outputs a collection of "motion textons" representing primitive behaviors and their transitions. However, the method relies on linear dynamical systems (LDS) to segment the input motion and derive the motion textons. By using LDS, motion textons serve to accurately represent the input motion with respect to an error function. In contrast, PDBV primitive behaviors are derived based on the spatio-temporal structure of the input motion. Consequently, primitive behaviors can have observable themes other than linear dynamics.

Related to these two methods are techniques for automatically building *motion graphs* [15]. These techniques take as input a set of motion "clips" and build a single conglomerated graph of transitions between subsequences of the original clips. A motion graph generalizes sequences of the input motions, but, unlike a primitive behavior or motion texton, does not intrinsically generalize to new motion. Instead, constraint-based optimization mechanisms use the motion graph to produce new motions.

Alternatively, control modules could be learned incrementally from robot experience. These methods produce forward models [7] or paired forward and inverse models [31] by creating new modules for experiences not represented in the current set of control modules. However, modification of these control modules for manual refinement is not straightforward.

Complimentary to PDBV are methods for generalizing a single behavior from demonstration, which take motion exemplars of a behavior and generalize them through interpolation, as in [24], or by learning a nonlinear dynamical system represented by the exemplars, as in [13].

Behavior vocabularies derived by PDBV could serve as a substrate for task-level control mechanisms. [12] have proposed a hybrid architecture for reinforcement learning on top of a *control basis* [11] of robust controllers. The basis could be replaced with a vocabulary derived by PDBV with the correct input data representation. Additionally, our derived vocabulary could be used with behavior-based task-level architectures [20] or within autonomous agent architectures [23].

Previous methods for markererless motion capture require the use of a kinematic model for fitting onto a 3D volume of the subject [19, 4] or to search pose hypotheses matching 2D image features [10, 8]. For markerless

kinematic motion capture, we offer a model-free approach that estimates the kinematic model and motion of an uninstrumented subject.

4 Deriving Behavior Vocabularies from Motion

Derived vocabularies consist of *primitive behaviors* and *meta-level behaviors*. A meta-level behavior is a model of the sequential composition of a subset of the primitive behaviors with respect to a higher-level behavior found in the input motion. Analogous to linguistic grammars, primitive behaviors would be considered terminals and meta-level behaviors non-terminals, representing of sequences of primitives.

The PDBV takes as input a single continuous motion, in the form of joint angles over time. For each underlying behavior, this structure should be present through multiple performances of the behavior that spans a range of variations. In addition, there should be no disconnection between subsets of the variations (or *instances*) of the behavior (i.e., the performances of the behavior should realize a single connected component). The purpose of the derivation process is to isolate instances of each underlying behavior into a primitive feature group and generalize these instances into a behavior primitive. For isolating instances, we assume the input motion has an underlying *spatio-temporal* structure. Spatially, instances of a behavior are related by similar spatial signature. Temporally, instances of a primitive feature group B typically precede and follow the other instances of other primitive feature groups A and C (i.e., a performance sequence of $A \rightarrow B \rightarrow C$), respectively.

The input motion is processed by four essential subprocedures of PDBV: interval segmentation, dimension reduction, clustering, and interpolation. Interval segmentation divides the input motion into a data set of motion segments. Each segment is an instance of some behavior being performed. Segmentation is a complex problem because it is dependent on knowing subgoals of a typically unknown intention or task domain. Because this problem is underdetermined, we approach interval segmentation by heuristically looking for salient events. One segmentation method, *z-function segmentation* [9], serves as a "stop detector" by looking for common zero velocity crossings across DOFs. This method, applied [22] to segmenting motion data from Robonaut into "episodes", works well for discrete "point-to-point" motions. For large dynamic motions, we developed *Kinematic Centroid Segmentation* [14] to treat limbs as pendulums and look for boundaries between pendulum swings across limbs.

We use *spatio-temporal dimension reduction* to transform the data set motion segments into clusterable primitive feature groups. Each motion segment is a point in a $d \times n$ dimensional "segment" space, where d is the number of DOFs and n is the number of frames in a motion. Motion segments are time-normalized to n frames using cubic spline interpolation. The dimensionality of segment space can be high (e.g., a motion segment of 20 DOFs normalized to 50 frames exists as a point in a 1000 dimensional segment space). Methods for *global spectral dimension reduction*, such as Isomap [29], are able to reduce spatial data (i.e., independent samples from an identical underlying manifold structure) to its intrinsic (potentially nonlinear) dimensions. The core embedding mechanism of these methods is the transformation of pairwise distances into distance-preserving coordinates through eigendecomposition. This mechanism can be explained in terms of kernel methods and multidimensional scaling [30]. The intrinsic underlying structure of the data is unraveled, given that the pairwise distance metric is consistent with this underlying structure. However, these methods only consider spatial distance metrics where the input motion has an unincorporated sequential order. By augmenting the spatial distance metric for temporal dependencies between pairs, our method for spatio-temporal dimension reduction is able to uncover the structure of the input motion by positioning instances of the same behavior group into *clusterable proximity* in the resulting embedding. Hence, we define a set of points in clusterable proximity to have a substantially smaller distance between any intra-cluster data pair than any inter-cluster data pair.

For *spatio-temporal Isomap*, we incorporate *common temporal neighbors (CTN)* with spatial geodesic distances to isolate primitive feature groups into clusterable proximity. In spatial Isomap, geodesic distances is used as the spatial distance metric, computed through all-pairs shortest paths. More specifically, Dijkstra's algorithm is computed from an initial graph where each data point places as distance-weighted edge between itself and its *local spatial neighborhood* (i.e., nearest neighbors). For incorporating temporal dependencies, we leverage these neighborhoods to identify common temporal neighbors (i.e., data pairs with locally similar spatio-temporal patterns). A point t_y is a common temporal neighbor of a point t_x if $t_y \in nbhd(t_x)$ and $t_{y+1} \in nbhd(t_{x+1})$, where $nbhd(t_x)$ is the spatial neighborhood of t_x, t_{x+1} and t_{y+1} are data points temporally adjacent to t_x and t_y, respectively. CTN is symmetric: $t_y \in CTN(t_x) \Leftrightarrow t_x \in CTN(t_y)$. If $t_y \in CTN(t_x)$, we reduce their neighborhood distance $D(t_x, t_y)$, by some given constant c_{CTN} before shortest-paths computation. Additionally, we set c_{offset} as a distance offset between temporally adjacent points, that are not spatial neighbors, to put

additional space between feature groups and c_{atn} as a scaling for temporal neighbors with common temporal neighbors (set to 1 for primitives). We assume c_{CTN} is set large (or small) enough such that *CTN transitivity* holds. The CTN transitivity property is if $t_y \in CTN(t_x)$ and $t_z \in CTN(t_y)$, then t_x, t_y, t_z will all be in clusterable proximity. We define the set of points related by CTN transitivity as a *CTN connected component*.

In the reduced dimension embedded space, members of a CTN connected component will be identifiable as a primitive feature group through simple clustering and generalized to a primitive behavior through interpolation. By performing an eigendecomposition on the matrix of pairwise distances, the resulting embedded space places intra-feature motion segments very proximal coordinates and inter-cluster segment very distal coordinates by preserving distances in a least squares fashion. Using the separation between clusters, primitive feature groups are found by a simple *axis-aligned bounding box clustering* method, called "sweep-and-prune" [6]. This method separates points into clusters by iteratively partitioning data on each coordinate dimension by looking for points exceeding some threshold separation distance. Probabilities for transitioning from one primitive to another are computed from a feature group sequence of the input motion. The embedding procedure preserves the correspondence between points in input (segment) space and the reduced-dimension (embedding) space. Using this correspondence, we treat each primitive feature group as *exemplars* for interpolating [27] across the span of variations of the primitive.

A *primitive behavior* is a family of motion trajectories defined by its exemplars and without semantic parameters. Through interpolation, an infinite number of variations on the exemplar motion can be produced. However, parameterization provided by interpolation cannot be easily indexed into for yielding a motion with particular properties. In V-A [24], interpolation provides an easily indexable verb through manual positioning of exemplars such that the adverb space has semantically meaningful axes. A selected point in adverb space will roughly index to a desired motion using interpolation as a *lazy evaluation* mechanism. In the absence of a semantic parameterization, we take an *eager evaluation* route by densely sampling a primitive feature group for a representative set of the range of variations. A fixed number of random samples are placed within an ellipsoid fitting the feature group, found through PCA. The samples are interpolated to produce a set of motions representing the primitive. If the sampling is dense enough, sampled motion trajectories should approximate the (probably nonlinear) underlying spatial manifold of the primitive in joint angle space.

Because the sampled trajectories are sequential, the spatial manifold they comprise can easily be used as a *flow-field* to convert a location in joint angle space to a flow vector (i.e., joint angle displacement). Using this flow field as a dynamical system, a primitive can synthesize motion from an initial posture to use as a desired trajectory for a robot. The use of primitives as dynamical systems eliminates the necessity for trajectory smoothing for transitioning between primitives. In terms of activity perception, a primitive can make predictions on future postures given the current posture of an observed motion. However, predictions from an appropriate model will likely diverge from an observed motion due to factors such as noise. We have begun experiments using Kalman filters to provide the optimal representation of an observed motion with respect to a primitive. Using the Kalman gains from each primitive over the observed motion, we segment the observed motion into intervals where a motion interval is best modeled by a single primitive.

Meta-level behaviors are formed by primitives that are typically performed in sequence, as part of a compound motion. For instance, if primitives A, B, and C are always performed in succession, then they could be collapsed into a meta-level behavior ABC. At the primitive level, we performed a spatial collapsing of varying instances of a primitive. At a meta-level, we temporally collapse groups of successive primitive instances into a *behavior feature group*. We use the same spatio-temporal Isomap except use c_{atn} to collapse instances of structurally sequential primitives and c_{ctn} to maintain the integrity of the primitive feature group. We hypothesize that additional iterations of spatio-temporal Isomap can be applied to find successively higher meta-levels of behaviors until converging, although we have not acquired motion data explicitly expressing multiple meta-levels yet. Meta-level behaviors are useful in the context of motion synthesis and movement classification because they encode compound motion across primitives and other meta-level behaviors. The meta-level behavior does not produce motion directly, but indexes into and arbitrates between its component primitives. We hypothesize that an interpolation mechanism or flow-field can be constructed to execute a meta-level behavior without indexing into component primitives.

5 Kinematic Model and Motion Capture

The capture of subject motion for input into the behavior vocabulary derivation procedure can be accomplished with a variety of motion capture systems. Here, we focus on one approach to motion capture that we developed,

called Kinematic Model and Motion Capture (KMMC). The motivation for developing KMMC stems from the lack of reliable, inexpensive, and streamlined motion capture system.

KMMC is a method for estimating a tree-structured kinematic model and joint angle motion from a motion sequence of a subject from mutiple calibrated cameras. KMMC uses *voxel carving* techniques [21, 28, 26] from computer vision that reconstruct volume points that make up the subject. These techniques work by finding the 3D intersection of volumes cast by segmented silhouettes of the subject in each camera image. KMMC works on point volumes that are currently created from camera systems, but could also be provided from or in combination with other passive sensing mechanisms (e.g., laser range scanners, thermal cameras, acoustics). The KMMC procedure consists of the following main subprocedures: volume skeletonization, frame-specific kinematic model estimation, skeleton curve alignment, normalized kinematic model estimation, and kinematic model reapplication.

The first step in KMMC is to find a *skeleton curve* that runs through the "middle" of the subject's volume and is conceptually similar to a *principal curve*, a medial-axis restricted to a 1-manifold. While a variety of skeletonization techniques exist, we require a method that can provide accurate results without substantial initialization and parameter tuning, and remain computationally reasonable for application to a large number of volumes. Therefore, we hypothesized that nonlinear dimension reduction could be used to remove nonlinearities in a volume due to rotations about the subject's DOFs. By removing joint nonlinearities, the volume is transformed such that finding a skeleton curve is simple.

We developed *Nonlinear Spherical Shells (NSS)* as a skeletonization method that uses Isomap to transform a subject's pose-dependent volume points to pose-invariant locations. The locations that comprise a subject's volume vary by the kinematic posture of the subject. We found that applying Isomap to a set of volume points with an underlying kinematic posture transform these points such that they become structured by the zero-posture of the same underlying kinematic model. For human subjects, the transformation to the zero-posture is visually similar to a "Da Vinci" posture, where the center of mass is at the origin and limbs are straight and pointing away from the origin. Using Isomap, the transformation of the volume to the zero-posture structure is performed without an explicit model of the subject's kinematics. The zero-posture volume is partitioned into concentric spherical shells starting at the volume center of mass and moving outward. "Sweep-and-prune" axis-aligned bounding box clustering is performed on the points of each shell partition. The centroid of each cluster is considered a point on the underlying principal curve of the zero-posture volume. These shell cluster centroids are connected into a tree-structured principal curve by connecting centroids of overlapping clusters on adjacent shells. A *skeleton curve* for the original pose-dependent volume is found by mapping the points of the zero-posture principal curve through interpolation and maintaining the tree-structured connectivity.

Using the captured volumes and skeletons via NSS, we estimate frame-specific kinematic models for a volume sequence. The NSS skeleton curve is an estimate of the underlying nonlinear axis for the capture subject at a specific frame. In estimating the kinematic posture for the capture subject, the skeleton curve must be partitioned by joints so each partition is reflective of a rigid body with a linear underlying axis (e.g., a cylinder). The tree-structure of the skeleton curve provides some of this partitioning in the root and branches of the skeleton curve tree. The final set of partitions are skeleton points that divide nonlinear sections of the skeleton curve into linear subsections. To find these points, we consider each skeleton curve section, defined between two existing joints or terminal skeleton points, and iterative include points of the section while checking for nonlinearity. Nonlinearity is measured by skewness of the volume points associated with included skeleton points. If including a new point causes skewness to exceed a given threshold, a partition is placed before the new point and section partitioning proceeds again.

To find a common kinematic model across all frames, the frame-specific models are aligned and common joints are identified. For alignment, temporally adjacent kinematic models are aligned to collapse all models together. We experimented with two methods for alignment. The first minimizes the squared error between the positioning of two models. The second, a simpler and more accurate alternative, assumes temporal coherence between skeleton curves to use Isomap for collapsing models together. The Isomap alignment procedure defines skeleton point local neighborhoods for spatial intra-frame neighbors and temporal inter-frame neighbors in adjacent frames. By setting the distance between inter-frame neighbors to zero, corresponding points across frames will collapse to the same location in the embedding. The structure of the skeleton curve is retained by using spatial distance for intra-frame neighbors.

Considering only skeleton points that are joints, density estimation is performed on the aligned models to find a common kinematic model. Locations of high density indicate locations of common joints across frames.

Density was computed using a Gaussian kernel and thresholded to identify common joint locations. These joints and their connectivity define the common kinematic model across the motion sequence. This model is then reapplied to the skeleton points in each frame to find joint angles through the local coordinate systems for each joint.

6 Integration with Robonaut project

Robonaut [1] is a robotic humanoid torso, developed by NASA. A current effort on the Robonaut project is to endow Robonaut with the ability to autonomously act and learn from humans in various collaborative situations. We envision a technique similar to PDBV being used to develop a set of skill-level controllers to serve as a "control basis" interface below task-level learning mechanism and above the Robonaut API.

In our estimation, two main technical obstacles need to be addressed to enable Robonaut to learn from human performance in an automated fashion. The first is the derivation of *perceptual-motor primitives* [17, 18] from robot experiences. These primitives differ from those we have discussed and derived so far in that we assumed perception is provided in kinematic form and motion is performed in free-space with no object interactions. Robonaut, however, will have significant environment and object interaction using only on-board sensing, not limited to kinematic information. By including sensory information in the input data [3, 22], the derived vocabulary will contain forward models with an association between information from on-board sensing and proprioception. Such primitives will allow Robonaut to act from sensory input, predict future sensory and motion states, and explain its motion and sensory input in terms of skills.

The second problem centers on incorporating automatically derived skills with task-level behavior learning mechanisms. Behavior vocabularies derived by a method such as PDBV provide basic capabilities at a skill-level, without explicit semantic or task-oriented information. Task-level learning mechanisms allow robots to autonomously learn tasks from human demonstration and supervision, assuming the presence of a set of basic skills. Our goal is enable the task-level by automatically deriving basic skills from a collection of human performed tasks off line. For Robonaut, the collection of human performed tasks will be taken as the sensory-motor information from teleoperation. Task-level learning, using an approach such as [12] and [20], can be performed online using a derived skill vocabulary. Such task-level mechanisms will benefit from this parsimony provided by skills grounded with meaning from human performance.

7 Conclusion

We have presented a methodology for deriving robot skills as behavior vocabularies from natural human performance and an approach for model-free markerless motion capture of such natural human performance. Together, these methods form a means for endowing a robot with skills without explicit programming. Techniques used for this methodology will be applied in developing sensory-motor skills and integration of task-level learning for robot platforms such as the NASA Robonaut.

References

[1] R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems*, 15(4):57–63, July-Aug. 2000.

[2] Christopher G. Atkeson, Josh Hale, Mitsuo Kawato, Shinya Kotosaka, Frank Pollick, Marcia Riley, Stefan Schaal, Tomohiro Shibata, Gaurav Tevatia, Ales Ude, and Sethu Vijayakumar. Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, 15(4):46–56, 2000.

[3] M. E. Cambron and R. A. Peters II. Determination of sensory motor coordination parameters for a robot via teleoperation. In *2001 IEEE International Conference on Systems, Man and Cybernetics*, Tucson, Arizona, USA, October 2001.

[4] Kong Man Cheung, Takeo Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, pages 714 – 720, June 2000.

[5] Chi-Wei Chu, Odest Chadwicke Jenkins, and Maja J Matarić. Markerless kinematic model and motion capture from volume sequences. In *To appear in the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, Madison, Wisconsin, USA, June 2003.

[6] Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, and Madhav K. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–196, 218, 1995.

[7] Yiannis Demiris and Gillian Hayes. Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model. In K. Dautenhahn and C. Nehaniv, editors, *Imitation in Animals and Artifacts*, chapter 13, pages 327–361. MIT Press, 2002.

[8] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133, Hilton Head, SC, USA, 2000.

[9] A. Fod, M. Matarić, and O. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, January 2002.

[10] D.M. Gavrila and L.S. Davis. 3d model-based tracking of humans in action: A multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recoginition*, pages 73–80, San Francisco, CA, USA, 1996.

[11] Roderic A. Grupen, Manfred Huber, Jefferson A. Coelho Jr., and Kamal Souccar. A basis for distributed control of manipulation tasks. *IEEE Expert, Special Track on Intelligent Robotic Systems*, 10(2):9–14, 1995.

[12] Manfred Huber and Roderic A. Grupen. A hybrid architecture for learning robot control tasks. *Robotics Today*, 13(4), 2000.

[13] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, pages 752–757, Maui, Hawaii, USA, 2001.

[14] Odest Chadwicke Jenkins and Maja J Matarić. Automated derivation of behavior vocabularies for autonomous humanoid motion. In *To appear in the Second International Joint Conference on Autonomous Agents and Multiagent Systems (Agents 2003)*, Melbourne, Australia, July 2003.

[15] Lucas Kovar, Michael Gleicher, and Frdric Pighin. Motion graphs. *ACM Transactions on Graphics (TOG)*, 21(3):473–482, 2002.

[16] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture: a two-level statistical model for character motion synthesis. *ACM Transactions on Graphics (TOG)*, 21(3):465–472, 2002.

[17] Maja J Matarić. Getting humanoids to move and imitate. *IEEE Intelligent Systems*, pages 18–24, July/August 2000.

[18] Maja J Matarić. Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics. In Chrystopher Nehaniv and Kerstin Dautenhahn, editors, *Imitation in Animals and Artifacts*, pages 392–422. MIT Press, 2002.

[19] Ivana Mikić, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Articulated body posture estimation from multi-camera voxel data. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 455–460, Kauai, HI, USA, December 2001.

[20] Monica Nicolescu and Maja J Matarić. A hierarchical architecture for behavior-based robots. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 227–233, Bologna, Italy, July 2002.

[21] Simon G Penny, Jeffrey Smith, and Andre Bernhardt. Traces: Wireless full body tracking in the cave. In *Ninth International Conference on Artificial Reality and Telexistence (ICAT'99)*, December 1999.

[22] Richard Alan Peters and Christina L. Campbell. Robonaut learning through teleoperation: Automatic acquisition of trajectories for articulated motion. Unpublished report, August 2002.

[23] J. Rickel, S. Marsella, J. Gratch, R. Hill, D. Traum, and W. Swartout. Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, 17(4):32–38, July/August 2002.

[24] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics & Applications*, 18(5):32–40, September - October 1998. ISSN 0272-1716.

[25] Stefan Schaal. Movement planning and imitation by shaping nonlinear attractors. In *Proc. of 12th Yale Workshop on Adaptive and Learning Systems*, 2003.

[26] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1067–1073, 1997.

[27] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the ACM national conference*, pages 517–524. ACM Press, 1968.

[28] Richard Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 58(1):23–32, July 1993.

[29] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[30] Christopher K. I. Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.

[31] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329, 1998.