Exemplar-Based Primitives for Humanoid Movement Classification and Control

Evan Drumwright, Odest Chadwicke Jenkins, and Maja J Matarić
Interaction Lab/USC Robotics Research Labs
941 West 37th Place, SAL 300, Mailcode 0781
University of Southern California
Los Angeles, CA 90089-0781
Email: {drumwrig|cjenkins|mataric}@robotics.usc.edu

Abstract—We present a unified methodology for humanoid robot control and activity classification using motor primitives [1], computationally efficient behaviors capable of perception and control. These primitives constitute a vocabulary for humanoid control capable of generating a large variety of complex movement through sequencing and superposition. We demonstrate how such primitives can be automatically derived from human motion-capture data, how they can be used to construct upperbody controllers, and how they can be applied to classification of observed humanoid behavior in real time.

I. INTRODUCTION

Real and simulated humanoid robots have been programmed to successfully catch balls [2], vault [3], juggle [4], dance [5], etc. However, there has been little success in getting humanoid robots to perform tasks in dynamic, human-populated environments. A humanoid that is to be effective in such environments must be able to react to unforeseen, exigent situations. It must also interact with humans, necessitating the ability to understand human activity. Motor primitives [1], modular and computationally efficient behaviors capable of perception and control, aim to provide these capabilities.

Our implementation of primitives provides the means for humanoid control with minimal or no planning and the ability to classify observed human movement. The primitive structure acts as a model of humanoid and human behavior, thereby providing a mechanism for control. The same model serves as a reference for classifying observed humanoid and human movement. It follows that primitives can serve as a foundation for imitation learning [1]: primitives act both in control and recognition of activity.

We implement motor primitives in this paper as sets of exemplars coupled with an interpolation mechanism; the implementation is drawn from Rose et al.'s Verbs and Adverbs work [6]. The collective set of exemplars acts as a movement model from which novel motion can be synthesized. Additionally, These sets intrinsically define probability distributions over robot configurations, given the model class. Similarly, primitives are dynamical systems, allowing their use for movement prediction for a given model class. We demonstrate the utility of exemplar-based primitives for humanoid control and human movement classification.

II. RELATED WORK

There has been much work to date in the area of humanoid control. Matarić et al. [5] used three different control strategies to make a dynamically simulated humanoid dance. Hodgins and Wooten [3] used state-machine based algorithms for animating dynamically simulated human athletes running, bicycling, and vaulting. Faloutsos et al. [7] controlled dynamically simulated humanoid characters rolling, rising, and performing other complex activities. Ijspeert et al. [8] used non-linear dynamical systems and locally linear learning to store and execute trajectories for swinging a tennis racket on a humanoid robot. These methods are predicated on primitive movements and some utilize modular control architectures. Our method differs from these in generating parametric joint angle trajectories, unlike the nonparametric trajectories in [8] and in operating at the kinematic level, unlike [3] and [7]. Additionally, Murray et al. [9], among others, have presented dynamic level motor primitives for robot control, but did not attempt to use their representation for perception. In contrast, we use the same representation for both control and perception, thereby reducing the complexity of the control architecture and ensuring that each function is an inverse of the other.

There exist several methods that operate at the kinematic level and are usable for humanoid control. Rose et al. [6] developed a system for interpolating motion data that is usable for both humanoid robots and non-humanoid animated agents. Kavraki et al. [10] planned paths for agents in high-dimensional spaces (e.g., configuration spaces) around obstacles. Kovar and Gleicher [11] splice clips of generalized motion-capture data together to create new motions. Our method draws from these areas, particularly from interpolating motion data.

Little research has been conducted on human action recognition using joint angle or joint-position data. Mori et al. [12] used a fuzzy rule-based system, with rules determined by human judges, in order to classify human joint angle data. Campbell and Bobick [13] successfully recognized ballet movements by first converting joint-position trajectories into a high-dimensional phase space and then operating within that space. We believe that our method for classification in

joint-space is faster and more robust than [12], and that our Cartesian-space classification method is simpler to implement than [13].

III. PRIMITIVE REPRESENTATION

Motor primitives are parametric, kinematic models of movement for individual behaviors, where a behavior is defined as a class of structurally identical yet stylistically varied movements. Tennis forehand strokes are good examples of behaviors; each type of stroke (e.g., forehand, backhand, etc.) is a recognizably different behavior and would thus compose a separate primitive. In contrast, there are many ways to perform each stroke; varying the model parameters alters stroke target and style. A primitive model for a behavior is constructed from an interpolation function trained on a set of exemplar joint angle trajectories; the trajectories correspond to defining movements for the behavior.

The remainder of this section discusses determining the primitive vocabulary (Section III-A), interpolation algorithms (Section III-B), and primitive flowfields (Section III-C), a dynamical systems representation of primitives used throughout this paper.

A. Development of primitives

The first problem that must be resolved before employing motor primitives is that of determining an appropriate movement vocabulary. This can be performed automatically as in [14] or through manual choreography as in [15], [6]. Both methods are appropriate in specific situations. Automated derivation, as summarized below, allows for behavior design and implementation to be dictated by capabilities demonstrated by humans, but requires a significant amount of representative motion capture data. Manual determination of the exemplar trajectories is much slower, more tedious, and susceptible to errors in human judgement, but results in precisely choreographed trajectories. It is quite feasible in limited domains, such as sports.

Our method for automatic primitive determination takes as input kinematic motion from some source (e.g., motion capture [16], robot control, manual animation). The automated derivation procedure consists of three main subprocedures: interval segmentation, dimension reduction, and clustering. The input motion is first segmented into intervals, producing a data set of motion segments. Each segment is considered to be atomic in that no relevant "events" occur within it. To segment, relevant events are found in the input motion and used as segment boundaries. Because no ground truth is available for deciding relevant events, heuristics, such as a thresholding z-function [17], limb pendulum swings [14], or manual inspection, are used to find relevant events.

Spatio-temporal dimension reduction and clustering are performed on the data set of motion segments to cluster exemplars of the same behavior. We assume that every motion segment represents an exemplar of some underlying behavior. Dimension reduction is used to transform the motion segments so that exemplars of the same underlying behavior are placed

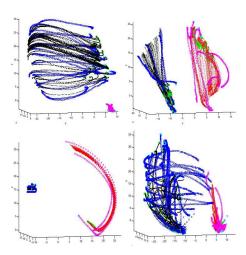


Fig. 1. Each plot shows hand trajectories for motion segments grouped into primitive sequence groups (right hand trajectories are displayed in dark bold marks, left hand in light bold marks). Interpolated trajectories are also shown (right hand in dark marks, left hand in light marks). The primitive sequence groups shown are for (top left) waving an arm across the body, (top right) dancing the "monkey", (bottom left) punching, and (bottom right) an unintentionally determined sequence group.

into clusterable proximity (i.e., significantly closer to each other than any other motion segment). We perform this transformation with augmented Isomap [18]. Isomap is a technique for nonlinear dimension reduction, which we extended to treat temporal as well as spatial characteristics of the data set [14]. The result from applying this *spatio-temporal Isomap* are separable clusters, each corresponding to a behavior and containing a set of exemplar trajectories.

B. Interpolation

Primitives use interpolation to produce novel trajectories from a few reference exemplars. We have experimented with multiple interpolation algorithms, including Shephard's [19] and the Verbs and Adverbs system [6]. The actual method of interpolation is not relevant to our methodology, but the running time of the interpolator should be low to facilitate real-time control.

C. Primitive flowfields

It is possible to sample trajectories from a primitive and to use those to construct nonlinear dynamical systems. Those samples lie in joint-space on a low-dimensional manifold that describes the temporal flow of each primitive. Such manifolds, or *flowfields*, provide update vectors from a given posture by evaluating nearby gradients on the temporal manifold. These update vectors can be used to predict kinematic motion. Figure 2 illustrates a flowfield for an arm waving primitive.

Primitive flowfields are practical for both movement control and classification. For control purposes, a set of flowfields with a high-level controller can produce smooth trajectories offline or incremental online control. The same underlying mechanism is used for both online and offline methods by using the flowfields as predictors that encode the joint space dynamics of the primitives from a current kinematic posture.

In the offline procedure, the prediction from an activated primitive is used to incrementally update a synthesized motion trajectory. In the online procedure, the prediction from an activated primitive resets the desired joint angle values for motor level control.

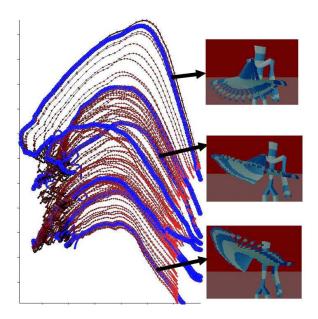


Fig. 2. A primitive fbwfi eld for a horizontal arm waving behavior. The fbwfi eld moves forward from right to left, with exemplars shown in bold. Motion for selected exemplars of this primitive are shown.

IV. CONTROL USING PRIMITIVES

We tested primitives as trajectory formation mechanisms for two physically-simulated humanoid robots, *Ares* and *Adonis*, in order to demonstrate the efficacy of our approach for realtime control. The two testbeds are similar in size, mass, and appearance to a male human; the primary difference between them is in the number of DOF.

Adonis [5] is a humanoid torso, consisting of 20 upper body DOF with fixed legs. Adonis' joints include a 3 DOF waist, 3 DOF neck, 3 DOF shoulders, 1 DOF elbows, and 3 DOF wrists. The simulated humanoid is actuated using proportional-derivative (PD) controllers. Ares [15] contains 44 Euler joints (132 DOF), including 72 DOF in the spine, 15 DOF in each arm, 12 DOF in each leg, and 3 DOF in the neck. Each arm is composed of five joints located at the clavicle, shoulder, elbow, wrist, and fingers; the fingers are treated as a single limb. The legs include joints at the hip, knee, ankle, and toes. Ares' joints are controlled by a feedforward plus feedback controller.

We used a behavior-based architecture to control these humanoids. Each primitive is encapsulated by a behavior module, which generates joint angle trajectories. The control architecture (Figure 3) is augmented by ancillary behaviors that serve to arbitrate conflicts between behaviors or send commands to behaviors. Our implementation was quite usable for real-time control and did not induce a significant computational load. Additionally, the Verbs and Adverbs system

[6], which inspired our motor primitive implementation, has proven to be quite effective at creating animation via trajectory formation.

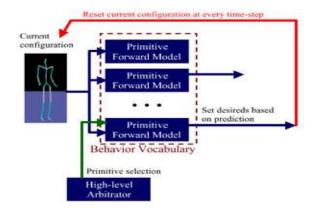


Fig. 3. Flowchart for humanoid motion synthesis using flowfi eld primitives with an arbitrator. Motion formation using this mechanism requires only an initial pose and incrementally updates desired configurations using a primitive selected by an arbitrator.

V. MOVEMENT CLASSIFICATION

Motor primitives are parameterized models of behavior. In this section, we present two methods that use these models to classify observed movement. The methods differ in their inputs. The first operates on joint angle data, and is well suited when motion capture data are available. The second method operates on Cartesian data, which are typically more readily obtained from machine vision.

A. Bayesian-based movement classification in intrinsic coordinates

We use a set of exemplars to build a Bayesian classifier which attempts to determine the probability that an observed movement is an instance of a behavior, C. Put another way, it calculates the probability of C, given a time series X of length n, where X_i is a vector of joint angle values sampled at time i:

$$P(C|X) = P(C|X_n, X_{n-1}, \dots, X_1)$$

By applying Bayes rule we transform the left hand side:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X|C)P(C) + P(X|\overline{C})P(\overline{C})}$$

In the absence of *a priori* knowledge, it must be assumed that a behavior is as likely to occur as not. Thus, the prior probabilities P(C) and $P(\overline{C})$ can be eliminated from the above equation. Now we need to calculate the probability of seeing behavior C given the time series and the probability of not seeing C given the time series.

1) Determining the conditional distributions:

a) Determining $P(X|\overline{C})$: We approximate $P(X|\overline{C})$ with a uniform distribution, because it is somewhat fair to assume that the human or humanoid's configuration can be arbitrary if it is not executing primitive C. This distribution is evaluated, with a small tolerance to allow for sensor malfunction, at X.

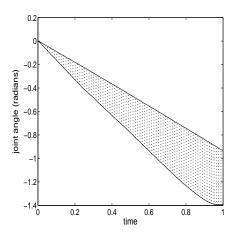


Fig. 4. Plot of the mixture space (the area produced by varying p and τ in $f(p,\tau)$) for "joint 2" of a primitive. The exemplars are plotted with solid lines. $f(p,\tau)$ assumes value -0.9 for $\tau\in[0.57,1]$

b) Determining P(X|C): Calculating P(X|C) is quite difficult. The joint random variables of this distribution are interdependent; the joints of the human or humanoid are tightly coupled as is each point in the time series to its neighbor. Therefore, the distribution cannot be factored into manageable univariate distributions. There is also no basis to choose a parametric distribution, which could simplify evaluation of P(X|C) considerably. For the following discussion, assume that $f(p,\tau)$ is a function with multidimensional outputs that takes as inputs p, an interpolation parameter, and τ , a time index that satisfies the property $\tau \in [0,1]$. $f(p,\tau)$ is the functional form of C.

Our way to solve this problem is to factor P(X|C) into manageable distributions that can be evaluated. Consider the equation below:

$$P(X_t, X_2|C) = P(X_1|X_2, C)P(X_2|C)$$
 (1)

In Equation 1, a multivariate distribution is factored into two univariate distributions. $P(X_2|C)$ is trivial to compute: sampling from $f(p,\tau)$ yields a nonparametric distribution of joint angles for C that is then evaluated at X. However, $P(X_1|X_2,C)$ is far more difficult to calculate, although C makes this computation tractable.

Assume that X_1 and X_2 are observed data from joints 1 and 2 respectively, and that $X_2 = -0.9$ radians. Figure 4 illustrates the output of the primitive for joint 1; the output assumes the value -0.9 radians during the time interval [0.57,1] time units. Thus, $t \in [0.57,1]$, because X_1 and X_2 are obtained from the primitive at the same value of τ . We can now determine the nonparametric distribution $P(X_1|X_2,C)$ by sampling from $f(p,\tau)$ over the interval [0.57,1]; Figure 5 shows the mixture space that from which samples are drawn. Note that only the component of the samples that corresponds to joint 1 is used in the resulting one-dimensional distribution. Figure 6 depicts this distribution.

This technique for evaluating multivariate distributions has been extended to observations with time-series data in the

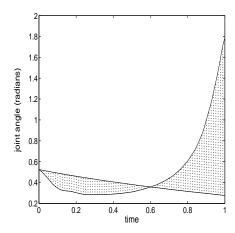


Fig. 5. Plot of the mixture space for "joint 1" of the same primitive as in Figure 4. The exemplars are plotted with solid lines.

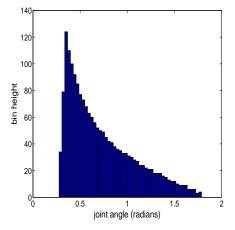


Fig. 6. Histogram produced by sampling Figure 5 between [0.57, 1].

same manner. Properties of the primitives (e.g., $\tau \in [0,1]$, nondecreasing τ for executing trajectories, etc.) are used to avoid computationally expensive multidimensional integration.

2) Complexity analysis: We require that the perceptual system be fast so that it, like the actuation system, can be used in a situated robot. Our algorithm exhibits a complexity of $\Theta(n)$, where n is the number of data points to be classified. This translates into a running time of approximately 20 ms for a 12-joint primitive on a 2 Ghz Pentium 4 processor when n=2. This performance precludes real-time recognition of multiple primitives, unless multiple, faster processors are used. Alternatively, speed can be increased significantly at the expense of accuracy by reducing sampling and interval search granularities.

B. Flowfield-based movement classification in extrinsic coordinates

The presented Bayesian-based method is simple and fast but relies on joint angle data as input, which are not always available in robotics applications. Thus we also implemented a flowfield-based classifier that operates on end-effector locations in Cartesian coordinates. We call this method of

TABLE I
CLASSIFICATION ACCURACY

Data Set	n = 1	n=2
Ares movements	96.61%	96.61%
Choreographed animations	99.94%	99.94%
Motion capture	99.97%	99.98%

classification trajectory encoding (Figure 7) because the result is a single joint space trajectory comprised of subtrajectories that predict best matching primitives. A trajectory encoding classifier works by continually comparing predictions from each primitive with observed future states from the input movement. The current state in this context is the posture at the instance of time after the last classification decision was made. Each primitive produces a predicted trajectory from the current state over a user-specified duration horizon. Classification decisions are made based on the Euclidean distance between observed and predicted end-effector locations at intervals defined by the user. The matching operator provides a scalar value indicating the similarity between a primitive's prediction and the observed future state of the input movement. Motion within the decision interval is classified into the primitive with the greatest similarity value produced from the matching operator. These classifications are winnertake-all and do not account for motion that fails to match any primitives. The encoded trajectory is formed by concatenating the predicted trajectory of the classified primitive with the previously encoded motion.

C. Evaluation

We evaluated the Bayesian classifier using three data sets consisting of 209,188 data points, where each data point is a vector of joint positions. The first data set contains movements performed on Ares and is primarily composed of movements generated by the motor primitives jab, hook, etc.; it also contains movements for transitioning between primitives. The second data set consists of over 50 animations (e.g., swim, reach, swat bees, etc.) choreographed by Credo Interactive [20]. All animations consist of many movements with substantial activity of the arm. The third data set is composed of 500 motion capture files that represent behaviors such as tennis, scrubbing, bowling, and walking; it and descriptions of the included motions are available from Credo Interactive [21]. We altered animations and motion capture data so that the postures of the hands in the animations matched those of the motor primitives. This alteration was made to handicap our classifier; recognizing a movement as a negative instance of a punch is considerably simplified if the hand is not clenched.

The classifier operates very well on all data sets, as shown in Table I. It is apparent that few of the movements included in the second or third data sets enter the mixture space of either behavior. This is unfortunate because we are almost entirely prevented from observing the additional classification power gained by increasing n; considering previous joint data is useful only in reducing false-positive classifications. The

additional power of greater n might prove to be unnecessary. We synthesized only five behaviors, which are highly similar semantically and structurally, but there is good separability between them, as evidenced by the high accuracy in classifying the first data set. Furthermore, there is almost complete separability between these behaviors and those encoded in the second and third data sets, even though they span much of the same Cartesian space.

The trajectory encoding classifier was tested against selected motions from the Credo motion capture set. The object of the test was to classify the observed motion as the primitive that, albeit subjectively, resembled the motion the most. Classification was complicated by several factors, including dissimilar kinematic structures and postures that differed between the start of the motion and the start of the primitive. Each motion was applied to four different sets of primitives, representing vertical waving (waving up and waving down), punching (punching and returning to neutral), and dancing "the twist" (twisting left and twisting right). Primitive-generated trajectories were not exact replicas of the observed testing motions, but rather coarse imitations that were synthesized by the flowfield's dynamical system. The classifier proved successful for all motions categorized in this manner. Future research will validate this method of classification on larger data sets.

VI. CONCLUSIONS

We have described an exemplar-based representation representation for motor primitives, a means for their realization as temporal flowfields, two methods for movement classification, and their use for control of humanoid robots. Our goal was to demonstrate that primitives are a powerful substrate onto which mechanisms for activity classification, prediction, and control can be built. Our future work will apply primitives towards facilitating interaction and imitation learning between humans and humanoids.

VII. ACKNOWLEDGEMENTS

This work was funded by the Defense Advance Research Projects Agency (DARPA) grant MARS DABT63-99-1-0015, DARPA MARS 2020 grant 5-39509-A, and the Office of Naval Research (ONR) MURI grant N0001-01-1-035.

REFERENCES

- M. Matarić, "Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics," in *Imitation in Animals* and Artifi acts, K. Dautenhaun and C. Nehaniv, Eds. MIT Press, 2002, pp. 391–422.
- [2] M. Riley and C. Atkeson, "Robot catching: Towards engaging humanhumanoid interaction," *Autonomous Robots*, vol. 12, pp. 119–128, 2002.
- [3] J. Hodgins and W. Wooten, "Animating human athletes," in *Robotics Research: The Eighth Int. Symposium*, Y. Shirai and S. Hirose, Eds. Berlin: Springer-Verlag, 1998, pp. 356–367.
- [4] C. Atkeson, J. Hale, M. Kawato, S. Kotosaka, F. Pollick, M. Riley, S. Schaal, S. Shibata, G. Tevatia, and A. Ude, "Using humanoid robots to study human behavior," *IEEE Intelligent Systems*, vol. 15, pp. 46–56, July 2000.
- [5] M. Matarić, V. Zordan, and M. Williamson, "Making complex articulated agents dance: an analysis of control methods drawn from robotics, animation, and biology," *Autonomous Agents and Multiagent Systems*, vol. 2, no. 1, pp. 23–44, March 1999.

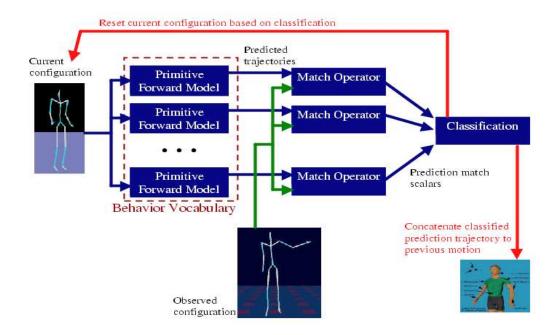


Fig. 7. Flowchart for fbwfi eld movement classification into a trajectory. The classifier uses an given initial configuration to update internally based on the best matching prediction from a primitive. The internal configuration is continually updated until the observed motion is completed.

- [6] C. Rose, B. Bodenheimer, and M. Cohen, "Verbs and adverbs: Multidimensional motion interpolation using radial basis functions," *IEEE Computer Graphics and Applications*, vol. 18, no. 5, pp. 32–40, 1998.
- [7] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "The virtual stuntman: dynamic characters with a repertoire of motor skills," *Computers and Graphics*, vol. 25, no. 6, pp. 933–953, December 2001.
- [8] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15, 2002, pp. 1547–1554.
- [9] R. M. Murray, D. C. Deno, K. S. J. Pister, and S. S. Sastry, "Control primitives for robot systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 1, pp. 183–193, 1992.
- [10] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [11] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," ACM Transactions on Graphics, vol. 21, no. 3, pp. 473–482, July 2002.
- [12] T. Mori, K. Tsujioka, M. Shimosaka, and T. Sato, "Human-like action recognition system using features extracted by human," in *Proc. of* the 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems, Lausanne, Switzerland, October 2002, pp. 1214–1220.
- [13] L. Campbell and A. Bobick, "Recognition of human body motion using phase space constraints," in *Proc. of 5th Intl. Conf. on Computer Vision*, 1995, pp. 624–630.
- [14] O. C. Jenkins and M. J. Matarić, "Automated derivation of behavior vocabularies for autonomous humanoid motion," in *Proc. of Autonomous Agents and Multiagent Systems (AAMAS 2003)*, Melbourne, July 2003, pp. 225–232.
- [15] E. Drumwright and M. Matarić, "Generating and recognizing free-space movements in humanoid robots," in 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 2003, pp. 1672–1678.
- [16] C.-W. Chu, O. Jenkins, and M. Matarić, "Markerless kinematic model

- and motion capture from volume sequences," in *Proc. Computer Vision and Pattern Recognition (CVPR)* 2003, vol. 2, 2003, pp. 475–482.
- [17] A. Fod, M. Matarić, and O. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, January 2002.
- [18] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [19] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the ACM national conference*. ACM Press, 1968, pp. 517–524.
- [20] "Credo interactive, lifeforms studio 4.0," http://www.credo-interactive.com, 2001.
- [21] "Credo interactive, mega mocap v2," http://www.credo-interactive.com, 2002.