Descriptive and Generative Representations for Modeling and Interpreting Human Movement

Paper ID 38

Abstract

Human movement analysis requires representations that allow for effective encoding and interpretation of spatio-temporally varying data. While several types of representation exist to encode elaborate movements (e.g. choreographies), they only bear a symbolic resemblance to what can be measured; trajectories of points on the body and joint angles. In this paper, we attempt to bridge the gap between the extremes of low-level encoding and high-level encoding. To facilitate this, we first introduce a representation based on dance notation. We then introduce a second notation based on L-systems. We show how the latter representation falls in the middle of the range of notations, allowing us to notate, encode, and synthesize various movements. We show the applicability of these representations by presenting animations created by an input of dance notation.

1. Introduction and Related Work

A fundamental need in machine analysis of human motion is an appropriate representation of the human body and all its motions over time. Human experts have relied on very elaborate representations to aid in encoding of human movements. Applications include gait analysis, posture analysis, dance notation, and physical therapy. These representations are very symbolic in nature, aiming more at the actions underlying the movements. Specific and very elaborate instances of these are dance notations [1][9][13][14] . These describe very detailed high-level movements, which sometimes have the goal of establishing an explicit and completely repeatable dance score. Gait analysts, on the other hand, rely more on several activity specific parameters to measure, for example, stride length, cadence etc., [6][4][5].

When we move to computerized measurement and tracking of human movement, the aim is to extract as detailed as possible a model of the person and the trajectories of limb movements and sometimes even joint angles. Computer vision research has made much progress in tracking whole-body movements from video , markerless and vision-based tracking of human articulation [8][7][15][10][22]. Such vision systems can provide trajectories of body parts when tracking reliably. Motion capture systems that use trackable markers attached to the body are more reliable in terms of extracting trajectories and joint angles of the articulated skeleton.

There are indeed higher level representations that try to use the captured measurements to model movements. Though they use different conventions, these high-level representations use an underlying model [11][2][3][17]. Despite effectively capture subtleties in motion, their connection to the metrics used in analysis of human movements by human experts is still very weak.

A representation that captures subtleties of motion and allows for ease in categorization and machine understanding will aid significantly in building systems that recognize how we move. In this respect, dance notations are promising, having been designed for elaborate, high-level, descriptions of choreography. Although this notation allows for easy categorization of motions, their use still has proven to be elusive for high-level categorization of human movements.

Computer representations and choreography are two extremes of the spectrum of human motion models. At one end of the spectrum lie abstract, high-level descriptions of motion primarily used by human experts. On the other end are the low-level descriptions of motion primarily used by computers. To bridge the gap between them, we examine a trait shared by most types of computer representations and present our preliminary efforts in this direction. All representations of the human body (with few exceptions), including various motion-capture standards, rely on joint angles to supply most, if not all, information. Other parameters such as stride length and speed (for walking) are estimated from this data.

We are interested in making a connection from data shared by computer representations to a high-level representation of human movement which is both compact and understandable. We build on concepts from dance notation, a representation designed by encoding very detailed, high-level and repeatable movements. Others have attempted to exploit those properties of dance notation by converting motion-capture data to Labanotation (a form of dance notation) for purposes of dance archiving [12][16]. Labanotation is a very verbose representation of motion. Due to the sheer volume of the notation, among other factors, they were forced to limit themselves to a small class of motions. As a result, Labanotation does not yield a representation with the properties we are seeking. There are choreographies which, like computer representations, use joint angles to define motion. To smoothly connect these high-level motion descriptions to the lower-level representations used by computers, we define an intermediate representation. Our proposed notation is based on L-systems.

In this paper we use Eshkol-Wachmann Movement notation to define high-level motions (Section 2). This motion is then converted into an L-System form (Section 3) for conversion into motion-capture style data. We present correspondences between these motion representations (Section 4). Finally, we show an application to computer animation by using dance notation to drive a character (Section 5).

2. Eshkol-Wachmann Movement Notation

The Eshkol-Wachmann (EW) movement notation is a mathematically oriented notation designed to accurately represent the range of motions of the human body. It was developed in the 1950's as a way for a choreographer to write down his movements such that could be accurately reproduced by dancers and understood by other people. They claim that "any event which has not been provided with some symbol will remain fortuitous and unrepeatable."[9].

Though the most common movement notation is Labanotation [14], EW-notation is more appropriate in this case. Where Labanotation has dozens of symbols to define motion, EW-notation only has three primary symbols. Furthermore, Labanotation is only concerned with abstract limb positions. Once one has defined the limb positions, additional symbols are added to capture subtlety. This causes the library of symbols to balloon out of control. EW-notation, on the other hand uses very precise symbols and, like computer representations, uses limb angles. It is thus able to capture the same subtlety by simply changing the angles without losing clarity. What we require is a notation that is compact and shares traits with other computer representations. EW-notation fits these criteria.

In EW-notation, the body is divided into eighteen separate parts according to the joints of the human body [9](Figure 1). Two additional parts are added representing the weight (contact with the ground) and the front (the section visible to the viewer) of the body. These parts are capable of two primary kinds of movement: rotational, and planar. Rotational movement allows a part of the body to

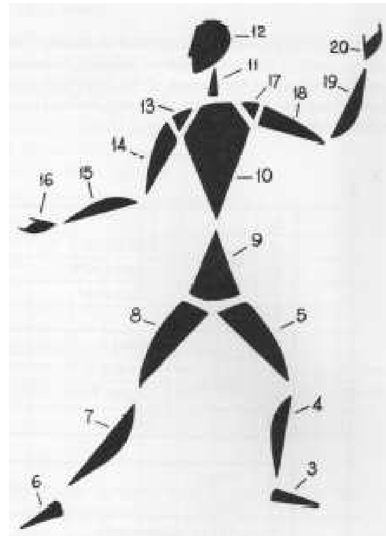

Figure 1. Parts Of the Human Body

rotate about its own axis, such as twisting a wrist. Planar movements are defined as movement parallel to a specified plane. Horizontal planar movement is defined parallel to a horizontal plane. There are three kinds of planar motion. They are movements in a horizontal plane, vertical plane and an intermediate plane. Given the horizontal plane, vertical plane is defined as a plane perpendicular to the horizontal plane, defined by the amount of rotation about the y-axis. Intermediate planes are defined by adding some value to rotate about the z-axis as well. Movements about the horizontal and vertical planes are merely cases of intermediate planar movement. In addition to the movements, one more constraint is added. It is called "The law of light and heavy limbs" and takes into account that body parts are connected and not wholly independent. The zero-position is defined as a stationary body with all limbs at rest [9](Figure 2). It is with respect to this position that movements and other positions are defined. In this paper, we assume a grounded and stationary body pose.

The primary advantage of EW-notation is its ease of visualization. With minimal training, any person can understand and perform motions defined in a page of choreography. This would, for example, allow an animator to define motions in dance notation with minimal effort and have a character repeat these motions perfectly using commercial packages. To facilitate this, we introduce an intermediate notation based on L-systems.

3. L-Systems

L-systems are a string rewriting system based on a set of production rules. These rules generate a series of com-

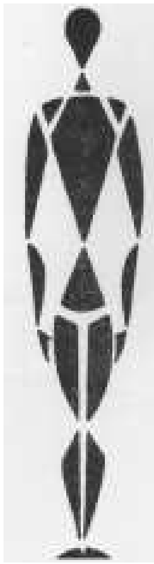

Figure 2. The Body At Rest

mands that can be interpreted by a LOGO-style interpreter. These commands are movement, represented by “F” and “f”. Rotations about the three principle axes are represented by “\”,“/”,“ˆ”,“&”,“+”, and “-”. The amount of rotation represented by each symbol is specified beforehand by some δ. A “[” represents pushing the current position and orientation onto a stack, and a “]” represents popping a position and orientation off the stack. The rules are recursive in nature. In the past, L-systems have primarily been used in the generation of fractals and plants [19][20]. They have been extended to generate more structured objects such as cities and buildings [18]. More recently, they have been combined with vision techniques to approximate the skeletons of trees [21]. These objects all share a trait in that they can grow to an unspecified size though cities are constrained by the amount of land available. The human body, however, is constrained to a single class of shapes. In order to ensure generation of valid skeletons, we constrain the number of recursions the L-system interpreter generates to two. We model a human body with a variant of bracketed L-systems presented in [19]. Our L-system for the human body follows (the turtle begins with an initial directional vector of [0 1 0]’):

$$
\begin{array}{lcl}
X_T & \rightarrow & [X_5 X_1 X_2 X_6] X_3 X_4 \\
X_1 & \rightarrow & F_1 \\
F_1 & \rightarrow & [\backslash 90 \delta_1 F_1 \backslash 90 \delta_2 F_1 \delta_3 F_1 \delta_4 F_1] \\
X_2 & \rightarrow & F_2 \\
F_2 & \rightarrow & [/90 \delta_5 F_2 /90 \delta_6 F_2 \delta_7 F_2 \delta_8 F_2] \\
X_3 & \rightarrow & F_3 \\
F_3 & \rightarrow & [/90 f_3 /90 \delta_9 F_3 \delta_{10} F_3 \hat{}90 \delta_{11} F_3] \\
X_4 & \rightarrow & F_4 \\
F_4 & \rightarrow & [\backslash 90 f_4 \backslash 90 \delta_{12} F_4 \delta_{13} F_4 \hat{}90 \delta_{14} F_4] \\
X_5 & \rightarrow & F_5 \\
F_5 & \rightarrow & \delta_{15} F_5 \delta_{16} F_5 \\
X_6 & \rightarrow & F_6 \\
F_6 & \rightarrow & \delta_{17} F_6 \delta_{18} F_6
\end{array}
$$

X_1 represents the production for the left shoulder and arm.
X_2 represents the production for the right shoulder and arm.
X_3 represents the production for the left leg.
X_4 represents the production for the right leg.
X_5 represents the production for the torso.
X_6 represents the production for the head and neck.

Two major changes to the standard L-system are worth noting. The first is that instead of having simply one generation representing forward motion of a LOGO turtle, we allow any number of such generations. The second is that we allow specification of the amount of rotation about an axis (this is equivalent to having a δ of 1). Each δ_i represents the parameter of motion for that particular limb. When all δ_i’s contain no motion, the body is in the zero-position. For the purposes of brevity and simplicity, the motions represented in the δ_i’s are always in the form of three rotations about the three principle axes, **X**,**Y** and **Z**. We will call this **XYZ** form.

Unlike other movement formats, L-systems can easily be understood by the casual observer and easily understood by computers (since they are simply LOGO commands). Due to the unstructured nature of L-system representation, it is trivial to convert data from other formats into the L-system format. Since other notations are more constrained, conversions between them are not always simple. These conversions can later be compressed into the **XYZ** form we have defined.

4. Representation of Motion and Pose

There is a mapping between EW-notation and the L-system representation. Given a pose and a movement in EW-notation, we can construct an equivalent L-system where all we need to change in the current L-system are the δ_i’s in the following manner:

When a rotation is specified for a body part, we add add “+θ” to the end of the δ_i for that body part in the L-system where θ is the amount of rotation specified.

When planar movement is specified for a body part, we add

“$\backslash\alpha + \beta\ \hat{}\gamma - \beta\ /\alpha$” to the start of the δ for that body part in the L-system where α is the specified rotation about the z-axis (for intermediate planar rotations), β is the specified vertical plane (for vertical planar rotations) and γ is the amount of rotation specified.

By definition, vertical planar movement has an α of 0. Horizontal planar movement has an α of 90 and a β of 0. The order of appearance of α and β depends on the body part specified. Limbs have a base of rotation at their top while the torso, head and neck, rotate from the bottom. When a body part is rotated from the bottom, the appearance of α and β are reversed.

Since these representations are mathematical in nature, we can define very precise movements. In order to prevent the L-system from getting large from long sequences, we put the δ_i's into **XYZ** form (See Appendix A). Through these precise movements, we can exactly represent any stationary pose of the human body.

To convert an L-system pose back into an EW-format, we first convert the rotation specified by each δ_i into quaternion format. The perpendicular to the vector defined is equivalent to the specified plane of rotation in EW-notation. Given a body in the zero-position, we simply find the vertical and horizontal shifts equivalent to this vector. The amount of rotation is also given by the quaternion representation.

5. Results

Using EW-Notation, we can animate a character by simply defining high-level motions. Given a time-step of motion, we first convert it into an L-System (described above), interpolating the motion across frames in order to go from a generative model to a descriptive one. From L-System form, it is then converted into the BioVision BVH standard (a list of joint angles in **ZYX** format). Here we present a few motions for the torso and the upper body as well as their corresponding L-systems. The initial choreography can be seen in [9] Figure 3. The entire animation, which shows the different kinds of motion of EW-Notation can be found at http://www.geocities.com/paper_writer2002/.

Figure 4 shows the body in the zero-position. The L-system for the arms and torso is:

$$\begin{array}{lcl} X_1 & \rightarrow & F_1 \\ F_1 & \rightarrow & [\backslash 90F_1\backslash 90F_1F_1F_1] \\ X_2 & \rightarrow & F_2 \\ F_2 & \rightarrow & [/90F_2/90F_2F_2F_2] \\ X_5 & \rightarrow & F_5 \\ F_5 & \rightarrow & F_5F_5 \end{array}$$

Figure 5 shows the body after one time-step. The L-system for the arms and torso is:

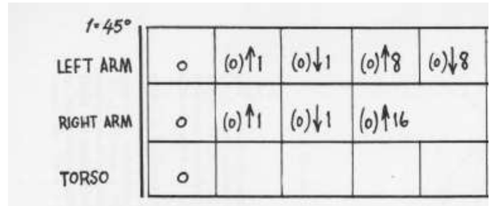

Figure 3. Some Simple Upper-Body Movements

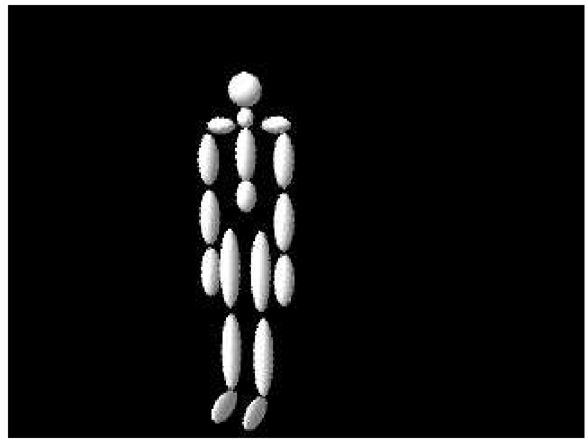

Figure 4. The Zero-Position

$$\begin{array}{lcl} X_1 & \rightarrow & F_1 \\ F_1 & \rightarrow & [\backslash 90F_1\backslash 90\hat{}\mathbf{45.0}F_1F_1F_1] \\ X_2 & \rightarrow & F_2 \\ F_2 & \rightarrow & [/90F_2/90\hat{}\mathbf{45.0}F_2F_2F_2] \\ X_5 & \rightarrow & F_5 \\ F_5 & \rightarrow & F_5F_5 \end{array}$$

Figure 6 shows the body taking a bow. The L-system for the arms and torso at the extreme is:

$$\begin{array}{lcl} X_1 & \rightarrow & F_1 \\ F_1 & \rightarrow & [\backslash 90F_1\backslash 90F_1F_1F_1] \\ X_2 & \rightarrow & F_2 \\ F_2 & \rightarrow & [/90F_2/90F_2F_2F_2] \\ X_5 & \rightarrow & F_5 \\ F_5 & \rightarrow & F_5\hat{}\mathbf{90.0}F_5 \end{array}$$

6. Conclusions and Future Work

We have presented two new representations of human body motions and a mapping between them. We have also shown how these can be converted into existing representations. This is made possible by the properties of the L-system. Since conversions into L-system format are simple, they can be used as a go-between for different, unrelated

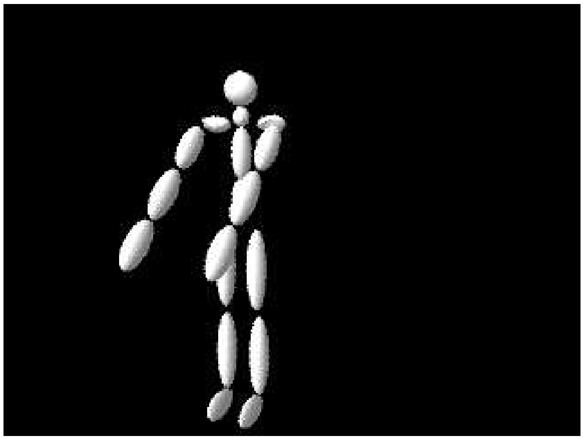

Figure 5. After One Time-Step

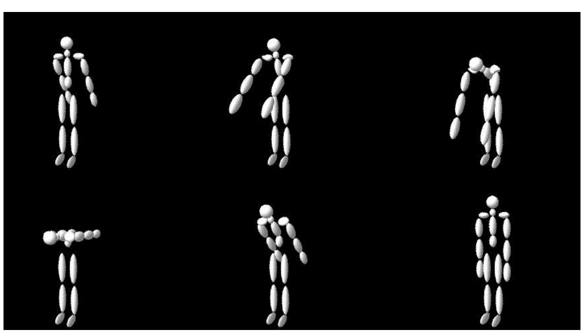

Figure 6. Taking a Bow

formats. For example, going from a motion-capture representation to dance notation can allow a vision algorithm to start with joint angles and use this conversion to classify them as specific high-level motions. We have discussed the limitations of Labanotation for body pose representation. Despite its limitations, it is a powerful tool. In the future, we hope to develop a conversion from Labanotation into the L-system format to facilitate other conversions.

We have also presented an immediate application of our system for purposes of character animation. Using our system, an animator can define high-level motions quickly and easily without having to key-frame every motion. Though motion-capture systems can allow an animator the same freedom, motions generated in this fashion are difficult to modify. Simple high-level motions generated with our system are also simple to change. Some issues do exist with the animations. The results show unconstrained motion. This can cause self-intersection of body parts. This problem, however, can easily be remedied by adding kinematic constraints to the character model, thus restricting motions to physically valid approximations. Further work in this direction will include interaction with the environment and the synthesis of more realistic motions.

References

[1] Action stroke dance notation. http://www.geocities.com/Broadway/Stage/2806/.

[2] J. Aggarwal and Q. Cai. Human motion analysis: A review. *Nonrigid and Articulated Motion Workshop*, pages 90–102, 1997.

[3] S. Bandi and D. Thalmann. A configuration space approach for efficient animation of human figures. *Nonrigid and Articulated Motion Workshop*, pages 38–45, 1997.

[4] T. Chau. A review of analytical techniques for gait data. part 1: Fuzzy, statistical and fractal methods. *Gait and Posture*, pages 49–66, 2001.

[5] T. Chau. A review of analytical techniques for gait data. part 2: Neural network and wavelet methods. *Gait and Posture*, pages 102–120, 2001.

[6] R. L. Craik and C. A. Oatis. *Gait Analysis: Theory and Application*. Mosby St. Louis, 1995.

[7] A. J. Davison, J. Deutscher, and I. D. Reid. Markerless motion capture of complex full-body movement for character animation. In *Eurographics Workshop on Computer Animation and Simulation*. Springer-Verlag, 2001.

[8] J. Deutscher and A. Blake. Articulated body motion capture by annealed particle filtering. In *Proceedings of Computer Vision and Pattern Recognition 2000*, 2000.

[9] N. Eshkol and A. Wachman. *Movement Notation*. Weidenfield & Nicholson London, 1958.

[10] D. M. Gavrila. The visual analysis of human movement - a survey. *Computer Vision and Image Understanding, vol. 73, no. 1*, 1999.

[11] M. Girard and A. A. Maciejewski. Computational modeling for the computer animation of legged figures. *SIGGRAPH 1985 Conference Proceedings*, pages 263–270, 1985.

[12] K. Hachimura and M. Nakamura. Method of generating coded description of human body motion from motion-captured data. In *Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication*, pages 122–127, 2001.
[13] F. Hagist and G. Politis. A computer program for the entry of benesh movement notation. *Dance Technology: Current Applications and Future Trends*, pages 73–81, 1989.
[14] Introduction to labanotation. http://www.rz.uni-frankfurt.de/∼griesbec/LABANE.HTML.
[15] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. cspress, 1996.
[16] T. Matsumoto, K. Hachimura, and M. Nakamura. Generating labanotation from motion-captured human body motion data. In *Proc. International Workshop on Recreating the Past - Visualization and Animation of Cultural Heritage*, pages 118–123, 2001.
[17] D. Ormoneit, T. Hastie, and M. Black. Functional analysis of human motion data. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000.
[18] Y. I. H. Parish and P. Müller. Procedural modeling of cities. *SIGGRAPH 2001 Conference Proceedings*, pages 301–308, 2001.
[19] P. Prusinkiewicz and J. Hanan. *Lecture Notes In Biomathematics*. Springer-Verlag New York, 1989.
[20] P. Prusinkiewicz, M. James, and R. Měch. Synthetic topiary. *Proceedings of the 21st annual conference on Computer graphics*, pages 351–358, 1994.
[21] I. Shlyakhter, M. Rozenoer, J. Dorsey, and S. Teller. Reconstructing 3d tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, pages 53–61, 2001.
[22] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision, vol. 1*, pages 784–800. Springer-Verlag, 2002.

A. Compressing an Arbitrary Set of Rotations

To convert sets of Euler angles, the following algorithm is adapted (based on the coordinate system). Here, the algorithm assumes conversion into **XYZ** format:

- Find the current rotation matrix, **C**
- Compute **T**=**Cz** (the projection of **z** into the new coordinate system)
- Rotate **T** into the **XZ** plane via a rotation about the X-axis learning **X** (the rotation about the X-axis)
- Compute **T**=**CXz**
- Rotate **T** into its proper position via a rotation about the Y-axis learning **Y**(the rotation about the Y-axis)
- Compute **T**=**CXYy**
- Rotate **T** into its proper position via a rotation about the Z-axis learning **Z** (the rotation about the Z-axis)
- The resulting set of rotations is (-**X**)(-**Y**)(-**Z**)