GENERATING STYLIZED DANCE MOTION FROM LABANOTATION BY USING AN AUTONOMOUS DANCE AVATAR

Worawat Choensawat and Kozaburo Hachimura
School of Science and Engineering, Ritsumeikan University, Kyoto, Japan

Keywords: Dance Notation, LabanEditor, 3D CG Animation, Associate Memory, Knowledge-based Systems.

Abstract: When producing the animation of a body motion from the dance notation, the dance knowledge is a key for accomplishing high-quality movement. This knowledge enables the dancer to know how to perform the correct movement from a movement notation score. This paper presents an approach for automatically simulating a CG animation from Labanotation scores. We achieve this goal by the integration of a CG animation with a dance-style interpretation module and it is called an autonomous dance avatar. In our experiment, we implemented an autonomous dance avatar to perform a Japanese stylized traditional dance such as Noh-Plays. The result shows that the autonomous dance avatar can reproduce Noh-Play correctly from Labanotation after it has been trained with the Noh-Play knowledge.

1 INTRODUCTION

Dance community, mainly in Western countries, has widely accepted Labanotation as a graphical notation scheme for describing human body movement. Similar to music score, Labanotation uses staff and symbols for the purpose of recording human movements in the fields of choreography and dance education (Hutchinson Guest, 1977).

Labanotation does not represent the nuances of a performance and exact movements of any particular dancer. However, it does capture the choreographer's creative idea, so that any person might interpret and perform those ideas again. Based on the aforementioned, with the same notation score, different dancers may perform a movement differently depend on their experience.

Labanotation is rich in symbols, and by using the full set of symbols; almost all of our body movements can be described. However, the resulting notation would become extremely complicated and difficult to comprehend. For that reason, the fundamental symbols have usually been used. The question is: how can we realize a method of describing peculiar features and nuances of artistic, traditional dance movements while suppressing the complexity in the notation score?

Hachimura and his research team have developed a system, named LabanEditor (Kojima et al., 2002), for editing Labanotation score and displaying the CG character animation of its score. LabanEditor uses a motion template for generating a CG animation from the fundamental elements of Labanotation (as illustrated in Section 2). The motion template describes the relationship between Labanotation symbols and the rotation and translation of the corresponding joint. However, using a single motion template, the system cannot reproduce slightly distinct poses that are sometimes defined with the same symbol.

For a current version of LabanEditor (Choensawat et al., 2010), the system uses the method of dynamic templates in order to represent the nuances of dance movements. With the dynamic template method, the system allows users to describe a single Labanotation score with multiple templates. However, this will load a user task for describing every single dance motion.

Because the creation of motion templates is a difficult task for users, this burden on the users can be lessened by having a knowledge based of dance styles embedded in a character model. In this paper, we present a dance-style interpretation module embedded in the character model, called *Autonomous Dance Avatar*. The embedded module enabled an autonomous dance avatar to encode the pattern of Labanotation score and select an appropriate dance movement to the pattern from the learned knowledge. The proposed module is built in LabanEditor.

Choensawat W. and Hachimura K. (2012).
GENERATING STYLIZED DANCE MOTION FROM LABANOTATION BY USING AN AUTONOMOUS DANCE AVATAR.
In *Proceedings of the International Conference on Computer Graphics Theory and Applications*, pages 535-542
DOI: 10.5220/0003946305350542

2 LABANOTATION

A Labanotation score is drawn in the form of vertical staff where each column corresponds to a body part. Figure 1(a) is an example of Labanotation scores corresponding to dance motion. Figure 1(b) shows the basic arrangement of columns in the staff. The horizontal dimension of the staff represents the parts of the body, and the vertical dimension represents time. The center line of the staff represents the center of the body: Columns on the right represent the right side of the body, and columns on the left, the left side of the body. Symbols are placed in the columns of the staff. The vertical length of a symbol shows the duration of the movement, from its beginning to its end (Hutchinson Guest, 1977).

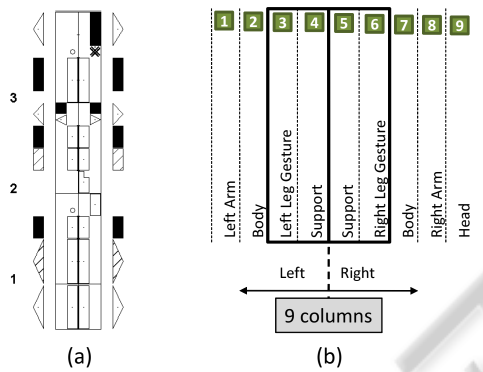

Figure 1: Labanotation scores: (a) example of Labanotation scores, (b) columns of Labanotation representing body parts.

Figure 2(a) shows direction symbols, used for describing the direction of movement of body parts. The shape of a symbol represents the horizontal direction of motion. Shading within a direction symbol shows the level of a movement, i.e. vertical direction of movement (low, middle, and high), as shown in Figure 2(a). Figure 2(b) shows the rotation signs and relationship pins respectively. The motion of each body part is expressed by a sequence of symbols placed in the corresponding column.

Labanotation is rich in symbols, all type of movement ranging from the simplest to the complex can be accurately described. Its usefulness are not limited to dancers and choreographer; the system has also been successfully applied to every field in which there is the need for recording human body motions e.g. athletics, anthropology, and physiotherapy.

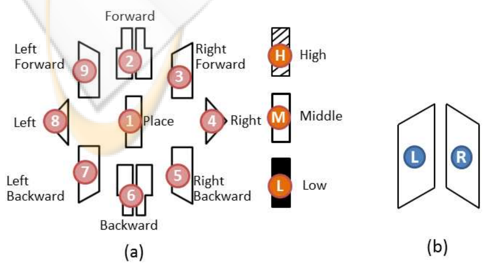

Figure 2: Symbols and signs used in Labanotation: (a) direction symbols, and (b) rotation signs.

3 RELATED WORK

3.1 Utilizing Labanotation for Dance Communities

To date, several graphics applications have been developed for preparing Labanotation scores and generating the body movement.

LabanWriter (Fox, 2000) is currently the most widely used Labanotation editor. The system is only for preparing Labanotation scores and recording them in digital form. It does not provide a function for displaying character animations corresponding to the notation. The latest version of LabanWriter can handle about 700 Labanotation symbols.

There have been several attempts to generate CG animation from Labanotation. The CG animation generator transforms Labanotation scores, which were prepared with LabanWriter, to the animation via the commercial software LifeForms (Coyle et al., 2002). However, LifeForms can only support the fundamental symbols of Labanotation.

LabanDancer (Wilke et al., 2005) is a LabanWriter scores to 3D animation translation tool. Like LifeForms, LabanDancer does not have any functions for preparing Labanotation scores and supports only a limited number of symbols.

LabanChoreographer (Zhang et al., 2006) is introduced for choreographing by retrieving the most similar motions from a motion capture database. Labanotation is used as an index tool for retrieval. Character animation is produced from motion capture data but not from the notation.

Practically, the above application software has some restrictions as follows. First, they lack of the integration of both creating Labanotation scores and producing 3D CG character animation. This is because they were separately designed and developed. Secondly, their software mainly focused on Western dances, and it takes no particular account of stylized dance motions of other cultures.

We decided to implement a dance-style interpretation embedded in the character model in LabanEditor because LabanEditor has the capability of preparing a Labanotation score and displaying the 3D CG character animation associated with the score.

3.2 Associative Memory

Simulating human-like learning or cognitive learning, we focus on an associative memory as referred to a content-addressable memory (Haykin, 1998). The content-addressable memory is a memory organization in which the memory is accessed by its content. If a pattern is presented to an associative memory, it returns whether this pattern coincides with a stored pattern. The coincidence need not be an exact match. An associative memory may also return a stored pattern that is similar to the presented one, so that noisy input can also be recognized.

An associative memory is used in information retrieval; for example, it is used for creating a memory of related keywords to produce thesauri (or knowledge bases) as introduced in (Lu et al., 2008; Chen et al., 2003; Chen et al., 1993). These automatic thesauri were then integrated with some existing manually-created thesauri for assisting concept exploration and query refinement. This associated memory was implemented with Hopfield networks (Adán-Coello et al., 2007). However, other models such as Boltzmann machine (Mairal et al., 2010), hypernetworks (Zhang and Kim, 2006; Zhang, 2008) have been applied to a search system of the word dictionary and a sentence completion when missing some words.

For the application of associative memories , we are interested in the string matching problem. If there is a given pattern, the problem consists in finding one or more usually all the occurrences of a pattern in a text. This problem has commonly occurred in many applications involving information retrieval such as bibliographic search and molecular biology. The concept of associative memories can be used for solving this problem. Such hypernetworks are used for learning the higher-order associations of the words from a text corpus. As described by Zhang et al. (Zhang, 2008), the hypernetwork memory is used for generating a text dialogue for a given movie scene image. The hypernetwork memory has the recall and recognition capability. For example, a training sentence "You need to wear it" and its source from the movie "24". In the recall task, the hypernetwork is given, say, "? need to wear it" and should complete the missing word to produce "You need to wear it". In the recognition task, the hypernetwork is to output "24" as the source of the sentence.

As suggested above, we can adopt an associative memory for generating a dance motion from Labanotation. In terms of Labanotation, a posture (or pose) is a position of the body as represented by a combination of Labanotation symbols. Then, a pose can be defined as a smallest unit of a Labanotation score. In an analogous manner, a Labanotation score would be comparable to a sentence of which words are equivalent of Labanotation units and characters are Labanotation symbols. Lastly, the source of sentences are comparable to a dance-style movement. The methodology of the dance-movement generation by using the associative memory will be described in Section 5.

4 LABANEDITOR

LabanEditor is an interactive graphical editor for editing Labanotation scores and displaying the 3D CG character animation associated with scores. The interactive interface for preparing Labanotation score allows users to input and edit the score by drag-and-drop techniques. When replaying the Labanotation score, users can observe the animation with a red horizontal line cursor moving upward corresponding to the animation progresses, as shown in Figure 3.

In the LabanEditor system, Labanotation scores can be represented as a simple format called *Labanotation Data (LND)*, which uses alphanumeric characters to represent basic symbols. The example of LND representation is shown in Figure 4(b). The lines that begin with "#" indicate the fundamental parameters of Labanotation. The movement of a body part is specified in the line followed by a command "direction", which corresponds to the Labanotation direction symbols. Figure 4 illustrates how a Labanotation score is convert to LND structure.

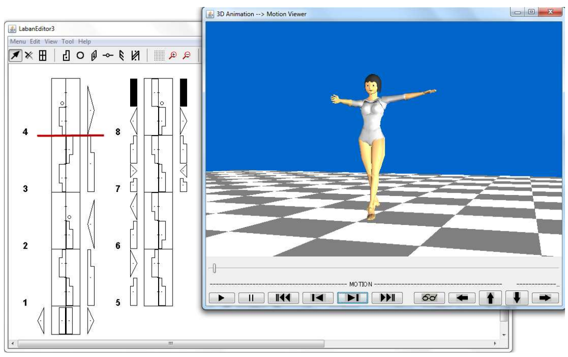

Figure 3: LabanEditor

LND describes a pose of the body at each timing just like key-frame body postures for animation, so

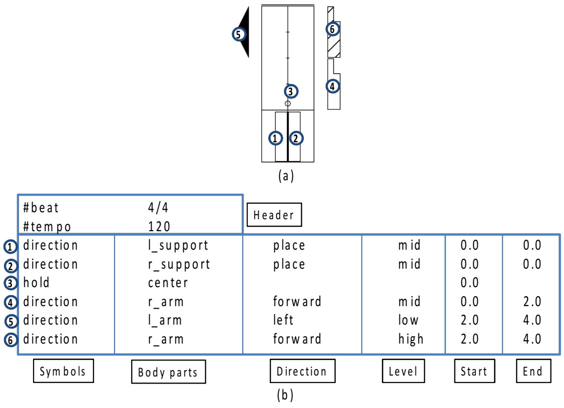

Figure 4: Relationship between Labanotation score and LND.

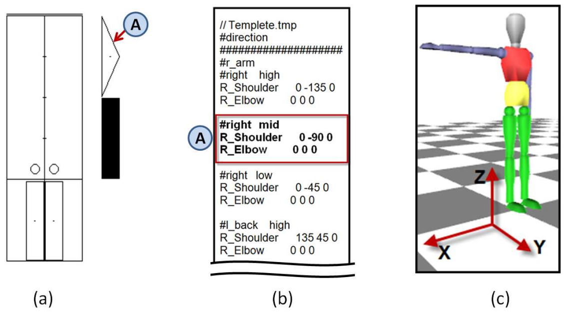

Figure 5: Relationship between user input symbols and a template file; (a) user input symbol, (b) part of a template file, and (c) target pose corresponding to the template in (b).

that we can produce motion of a body part by simply applying interpolation between start and end key-frame poses. A key-frame pose of a body part at a time corresponding to an end of a symbol is defined by a Labanotation symbol. The system converts direction symbols into animation key-frames by using a motion template for a mapping between the symbol and its corresponding pose of the body part.

For generating a animation, the system converts LND into animation key-frames by using a motion template file for a mapping between the symbol and its corresponding pose of the body part. The motion template file describes the relationship between a direction symbol at the particular column and the rotation and translation of the corresponding joint.

Figure 5 shows a notation and description in a motion template file, and the resulting pose. The symbol marked “A” in Figure 5 (a) is mapped to the description of the part marked “A” in the motion template file shown in Figure 5 (b), which indicates a target pose of the right arm achieved by rotating the right shoulder joint 90 degree counterclockwise around the y-axis from the standard pose as shown in Figure 5 (c).

5 AUTONOMOUS DANCE AVATAR

An autonomous dance avatar is a character model embedded with an capability of dance-style interpretation. The interpretation of dance styles is the recall and classification process of stored Labanotation scores and their associated motion templates. To achieve that, we started with teaching a dance avatar to have a dance-style memory, which is an associative memory between Labanotation scores and the corresponding movements. The dance-style memory is an associative function between scores and their movements as shown in Figure 6. Figure 6 is an example of stylized traditional dances, Noh-Plays where the leftmost column shows Labanotation scores related to four unit movements (called Kata) and the snapshots of the corresponding CG animation.

Figure 6: CG character animation of four Kata: (a) *Hiraki*, (b) *Tachi*, (c) *Shitai-tome*, and (d) *Ougi-kazashi*.

The dance-style memory can be designed and implemented with a two-layered, associative memory. The first layer involves a recall process of known poses. For example, as shown in Table 1, a combination of Labanotation symbols associated with pose#20 comprises three symbols of (l_support,place,mid), (r_support,place,mid), (r_arm,forward,mid), which these symbols are related to Labanotation symbols #1, #2, and #4 as shown in Figure 4 (a). After that, the second layer classifies a sequence of poses to a trained dance style; for example, a sequence of pose#20 and #10 is classified as motion#3.

Table 1: Dance-style interpretation with a two-layered, associative memory.

Layer	Query	Recall/ Recognition
	Labanotation unit	**Pose No.**
1^{st}	(l_support,place,mid) (r_support,place,mid) (r_arm,forward,mid)	pose#20
	(l_arm,left,low) (r_arm,forward,high)	pose#10
	Sequence of poses	**Movement**
2^{nd}	(#20, #10) (#5, #7, #30)	motion#3 motion#8

To summarize, we divide a task of autonomous dance avatar into two sequential subtasks as follows:

1. Decompose a Labanotation score into a number of units, and
2. Store and retrieve dance styles in/from the two-layered, associate memory.

The interpretation consists of storing and retrieving processes as described in Algorithms 2 and 2, respectively. First of all, both algorithms must start with the decomposition of Labanotation scores into units, and then follow by training or testing stages as shown in Algorithms 2 and 2), respectively. Algorithm 2 describes the implementation of an associative memory for storing Labanotation units and ,then, dance-style patterns which a pattern can be formed as a concatenation of units while Algorithm 2 shows the retrieving method of dance-style patterns.

Given an unknown Labanotation score, we can assign a set of motion templates to it by applying Algorithm 2. After decomposing the Labanotation score, Algorithm 2 starts with retrieving a stored unit from the 1^{st}-layer, associative memory for all units. Lastly, we adopt a concept of string matching for searching a

input : A set of Labanotation scores and its corresponding motion templates
output: A storage of dance-style patterns

Decompose Labanotation scores into units;
Create a two-layered, associative memory for storing dance-style patterns;
1^{st} : storing Labanotation units
2^{nd} : storing a sequence of Labanotation units and its associated motion tempates

Algorithm 1: Developing the dance-style pattern storage.

set of motion templates. A sequence of units is analogous to a text. That is to find an occurrence of a set of patterns (defined in Algorithm 2) in the text. This can be implemented by the 2^{nd}-layer, associative memory.

input : An unknown Labanotation score
output: A set of motion templates associated with its score

Decompose the Labanotation score into units;
for *each Labanotation unit* **do**
  Retrieve a most matching Labanotation unit from the 1^{st}-layer, associative memory;
end
Retrieve a set of motion templates from the 2^{nd}-layer, associative memory;

Algorithm 2: Retrieving a dance-style pattern from the storage.

5.1 Decomposition of a Labanotation Score

A Labanotation score is a set of symbols aligned along a time line as explained in Section 2. Given a set of symbols, we can find a minimum number of subsets where a subset must compose of coincident symbols. Each subset represents as a unit. Algorithm 3 shows how to break down a score into a number of units as similar to the minimum clique partition problem where the problem and its solution is described in the graph algorithms and applications as found in (Cenek and Stewart, 2003).

Given a score, vertices can be represented by symbols appearing in the score. Let l_i, c_i, u_i be lowest, center, and highest points of symbol i, respectively. Edges e_{ij} will be one if and only if

$$l_i \leq c_j \leq u_i \quad \text{OR} \quad l_j \leq c_i \leq u_j \quad .$$

For example, we will show a decomposition by a simple example. Given a Labanotation score as shown in Figure 7, we can find four cliques by applying Algorithm 3 as shown in Table 2.

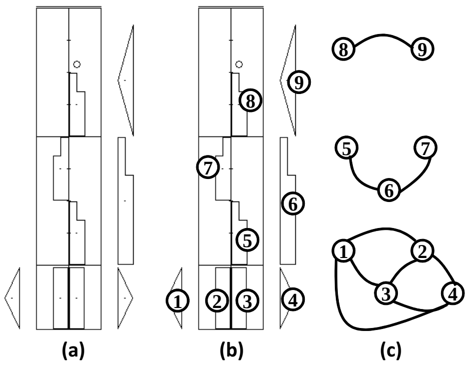

Figure 7: Transform a Labanotation score to an undirected graph $G(V,E)$: (a) a Labanotation score, (b) Labeling symbols with 1 to N where N is a number of symbols in the score, and (c) the corresponding graph $V = \{1,\dots,N\}; N = 9$.

5.2 Associative Memory

The purpose to use of an associative memory is for building a dance knowledge. The knowledge can be constructed by using a training set of input and target. After training, the memory is equivalent to a mapping function between inputs and the associative outputs. This can be accomplished by using the Bayesian statistical theory (Agrawal and Srikant, 1994; Liu et al., 1998). We are implementing an associative memory for interpreting Labanotation scores to motions. The interpretation from scores to motions must pass throughout poses as an in-between data as described in Table 1. Subsequently, our model is a two-layered, associative memory where the first layer is for matching between Labanotation symbols and a Labanotation unit, and the second is for assigning a sequence of Labanotation units with a motion template file.

input : A Labanotation score
output: A set of Labanotation units

1 *Draw an undirected graph $G(V,E)$, where V is a set of all symbols appearing in the score*;
2 *Determine $E = \{e_{ij}\}$* by using the equation below;

$$e_{ij} = \begin{cases} 1 & \text{if } c_i \in [l_j, u_j] \\ 0 & \text{otherwise} \end{cases}$$

3 Partition V into a minimum number of cliques;

Algorithm 3: Decomposing a Labanotation score to a set of minimum units.

Table 2: Example of a list of Labanotation units.

Labanotation score: *Figure* 7(*a*)	
No. of symbols: 9	
No. of units: 9 (minimum of cliques)	
Clique No.	**Set of symbols**
q_1	$\{1,2,3,4\}$
q_2	$\{5,6\}$
q_3	$\{6,7\}$
q_4	$\{8,9\}$
Motion template No.	**Pattern**
t_1	q_1q_2
t_2	q_3q_4

For the first-layer, the memory is used for converting a Labanotation score to a sequence of poses. We first store the Labanotation units referred as poses, after that the memory can recall the stored poses. For unknown pose in the recall process, the memory will try to search the most similar pose even if some missing symbols are occurred. The implementation is based on the statistical theory of the joint probability of a Labanotation symbol S and a Labanotation unit U as shown below:

$$P(S,U|W_1) = P(x_S, x_U|W_1) = P(x|W_1) \quad (1)$$

where W_1 is the training parameters and $x = (x_S, x_U)$ is the training pattern consisting of a Labanotation symbol x_S and a Labanotation unit x_U. The implementation can be achieved by using a Bayesian classifier (Witten et al., 2011; Hall et al., 2009).

The second-layer, associate memory is used for matching between a sequence of Labanotation units and its corresponding motion template. Similar to the first layer, the implementation of the second layer is as shown below and let a sequence of units be Q and a motion template be T.

$$P(Q,T|W_2) = P(y_Q, y_T|W_2) = P(y|W_2) \quad (2)$$

where W_2 is the training parameters and $y = (y_Q, y_T)$ is the training pattern consisting of a sequence of units x_Q and templates x_T.

6 EXPERIMENT AND PRELIMINARY RESULT

In experiment, we test the dance-style interpretation of Noh-Plays. For a brief description, Noh-plays are one of the most famous and characteristic Japanese traditional performing arts. Noh movements

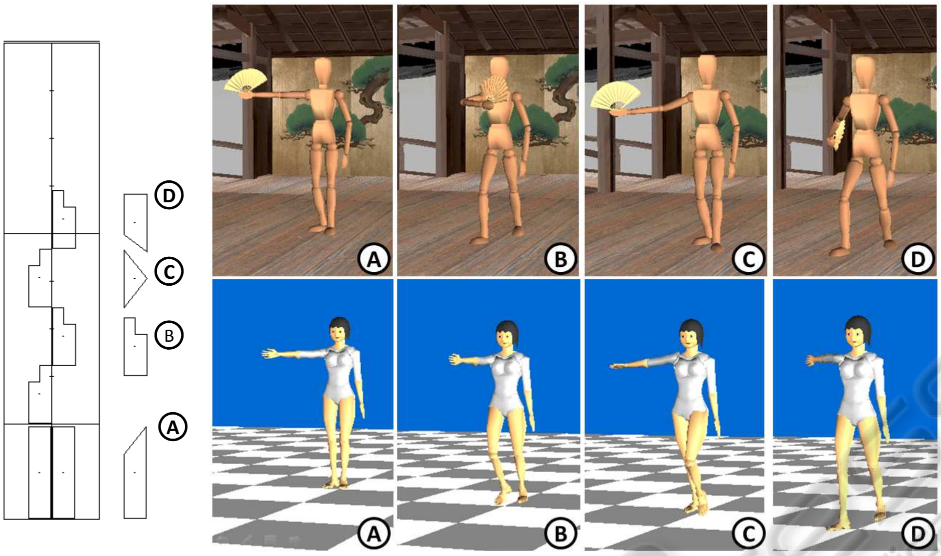

Figure 8: Snapshot of the CG animation for a Labanotation score comparing between the dance avatar with Noh knowledge and the normal avatar.

are highly stylized and unique. 'Shimai' is a short but principal performance extracted from the whole Noh play. In principle, each Shimai is composed of a number of prescribed movement units known as 'Kata', or form.

For the preparation of a dance-style database, we recorded the Noh-Plays (performed by Mr. Toyohiko Sugiura, who is the master of *Kanze* Noh School) by using three video cameras in the following angles: front, side, and perspective views, respectively. We have the videos of 6 Shimai with 32 unique Kata in total. By precisely observing the videos, we described each Kata with Labanotation and their associated motion templates. We use these Labanotation score and their associated motion templates to built the dance knowledge.

The aforementioned process involves with the preparation of training data that consists of Labanotation scores and the associated motion templates. We implemented the autonomous dance avatar embedded in LabanEditor and used Java classes of Bayesian classifier (Witten et al., 2011) for developing the two-layered, associate memory. After training a dance avatar, it can perform a Noh play correctly where an example is shown in Figure 8. Figure 8 shows the snapshots of the autonomous dance avatar embedded with a dance-style interpretation module comparing with a normal avatar. The autonomous dance avatar can move its body according to their stored motion patterns while the normal avatar just moved its body following a standard movement. Even though two avatars put their arm besides their body, they have different postures of their left arm. The autonomous dance avatar has its slightly bent left arm while the other has its left arm straight out. While their movement according to Labanotation unit D (Figure 8), the autonomous dance avatar rotating its right hand differ from that of the normal avatar.

7 CONCLUSIONS AND FUTURE WORK

Since using the fundamental description of Labanotation cannot describe a detail of human body movement, we have to create the motion template that describes the relationship between Labanotation symbols and the rotation and translation of the corresponding joint. In this paper, we present an autonomous dance avatar in which a dance-style interpretation module embedded. The embedded module enabled an dance avatar to encode the pattern of Labanotation score and select an appropriate dance movement to the pattern.

The contribution of this paper is a proposed framework for developing the autonomous dance avatar which includes the following mechanisms:

1. a mechanism based on a minimum clique partition for finding minimum independent units,
2. a mechanism incorporated an associative memory for storing and retrieving Labanotation units, and searching a motion template from a sequence of Labanotation units.

In our experiment, we test our approach against the stylized Japanese traditional dance, Noh-Plays. We create the database of Noh-Plays acquired from a recorded video of a Noh expert. The experimental results shows that the autonomous dance avatar can remember the Noh movement pattern. Comparing with a normal avatar, the autonomous dance avatar can pose its body according to Noh style. This is an preliminary result.

As beneficial for Noh players/learners, the autonomous avatar in LabanEditor can be used for the following goals:

- **Self Studying:** Noh beginners have the possibility of studying body motions on their own via the notation and CG animation.
- **Expressing Idea:** They can use the system as a presentation tool for their idea about the choreography of the performance and display in 3D CG animation.
- **Choreographing a Noh Play:** Ability to choreograph a Noh play without having to have the knowledge of Labanotation.

In future work, the dance-style knowledge of the autonomous dance avatar will expand to cover other dance styles. The system of a variety of dance-style knowledge will be implemented and evaluated in both user and expert domains. The scope of our evaluation will be related with the usefulness of the system, the accuracy and quality of 3D character animation.

REFERENCES

Adán-Coello, J., Tobar, C., de Freitas, R., and Marin, A. (2007). Hopfilter: an agent for filtering web pages based on the hopfield artificial neural network model. *Data Management. Data, Data Everywhere*, pages 164–167.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *20th International Conference on Very Large Data Bases*, pages 478–499. Morgan Kaufmann, Los Altos, CA.

Cenek, E. and Stewart, L. (2003). Maximum independent set and maximum clique algorithms for overlap graphs. *Discrete Applied Mathematics*, 131(1):77–91.

Chen, H., Lally, A., Zhu, B., and Chau, M. (2003). Helpfulmed: intelligent searching for medical information over the internet. *Journal of the American Society for Information Science and Technology*, 54(7):683–694.

Chen, H., Lynch, K., Basu, K., and Ng, T. (1993). Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE Expert*, 8(2):25–34.

Choensawat, W., Takahashi, S., Nakamura, M., Choi, W., and Hachimura, K. (2010). Description and reproduction of stylized traditional dance body motion by using labanotation. *Transactions of the Virtual Reality Society of Japan*, 15(3):379 – 388.

Coyle, M., Maranan, D., and Calvert, T. (2002). A tool for translating dance notation to animation. In *Proceedings of Western Computer Graphics Symposium*.

Fox, I. (2000). Documentation technology for the 21st century. *Proceedings of World Dance*, pages 136–142.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Pearson Education.

Hutchinson Guest, A. (1977). *Labanotation*. New York: Routledge, Chapman y Hall.

Kojima, K., Hachimura, K., and Nakamura, M. (2002). Labaneditor: Graphical editor for dance notation. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 59–64. IEEE.

Liu, B., Hsu, W., and Ma, Y. (1998). Integrating classification and association rule mining. In *Fourth International Conference on Knowledge Discovery and Data Mining*, pages 80–86. AAAI Press.

Lu, W., Lin, R., Chan, Y., and Chen, K. (2008). Using web resources to construct multilingual medical thesaurus for cross-language medical information retrieval. *Decision Support Systems*, 45(3):585–595.

Mairal, J., Bach, F., and Ponce, J. (2010). Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1–1.

Wilke, L., Calvert, T., Ryman, R., and Fox, I. (2005). From dance notation to human animation: The labandancer project. *Computer Animation and Virtual Worlds*, 16(3-4):201–211.

Witten, I., Frank, E., and Hall, M. (2011). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Zhang, B. (2008). Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory. *Computational Intelligence Magazine, IEEE*, 3(3):49–63.

Zhang, B. and Kim, J. (2006). Dna hypernetworks for information storage and retrieval. *DNA computing*, pages 298–307.

Zhang, S., Li, Q., Yu, T., Shen, X., Geng, W., and Wang, P. (2006). Implementation of a notation-based motion choreography system. *Interactive Technologies and Sociotechnical Systems*, pages 495–503.