The 5th International Conference on Electrical Engineering and Informatics 2015
August 10-11, 2015, Bali, Indonesia

Inverse Kinematics and Gesture Pattern Recognition using Hidden Markov Model on *BeatMe! Project : Traditional Dance Digitalization*

Zahrotul Aisyah Ulfah*1, Aciek Ida Wuryandari*2, Yoga Priyana*3

School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
[1]zahrotul.aisyah@s.itb.ac.id
[2]aciek@lskk.ee.itb.ac.id
[3]yoga@lskk.ee.itb.ac.id

***Abstract*— Indonesian traditional dance preservation efforts that nowadays increasingly eroded by foreign culture needs to be improved and adapted to technological improvement. To answer that, BeatMe! Project developed for fulfil the needs of entertainment and traditional dance learning media by integrate 3D motion capture, processing data and visualization. This paper will explain the processing data detail implementation which include big data summarizing as in summarizing XYZ coordinate of joint to be angle between two joints. In addition, this paper also explain detail implementation of HMM learning system for gesture pattern learning and recognizing. Environment used is Kinect, Visual Studio and MATLAB. Result show summarized data of discrete value between joints angle, learning curve as the learning process output that tends to rise and converge on a value within limits of 700.000 symbols, new gesture pattern recognition show a good performance in one degree joint because its position nearest the main torso, and not to good performance in two degree joint because of randomized value that happen when the body position isn't aligned with Kinect.**

Keywords*— *inverse kinematics, Hidden Markov Model, recognition, Kinect, gesture, pattern

I. Introduction

Indonesia is a country that has so many diversity of cultures, such as music, traditional clothes, weapons and dance. However, this traditional dance are now starting to be eroded by foreign culture. Various conservative efforts is hard to do because of the lack of dance teaching facilities to be something interesting to learn and have a high selling price. Therefore, interest in traditional dance and documentation should be improved and adapted to technological improvement, such as using digital media so that every individual can improve their personal traditional dancing ability anywhere and anytime without location and time limit.

Beatme! Project: Traditional Dance Digitalization developed as a device that integrate 3D motion capture which capture dancer movement using Kinect, processing movement data, saving and visualizing in 3D model. In addition, movement data will processed into tutorial video and dance karaoke games.

One of the processes that handled by BeatMe! is data processing. Data processing covers data summarizing process, discretization and gesture pattern recognition. Raw data from motion capture form a big data contains XYZ coordinate joint position all over human body. For that, summarizing is necessary. Inverse kinematic will change the joint position to be parameter joint angle formed by two joints so that reduce data amount. Gesture pattern recognition developed using HMM which includes learning and recognition process. The system will be trained using a lot of gesture patterns data and then tested to recognize new pattern.

In this paper, given the detail design of summarizing data system and learning system. Then, the detail of Inverse Kinematics and HMM implementation will be explained too.

II. System Design

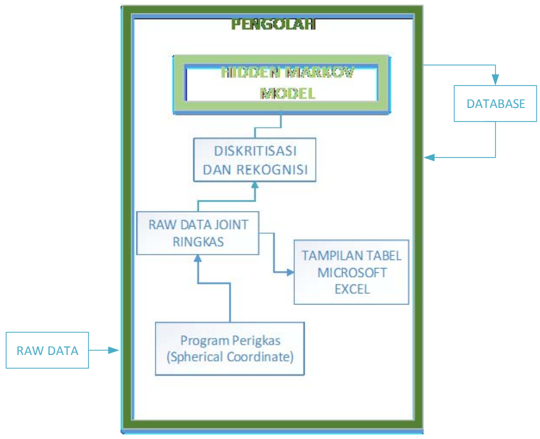

Fig. 1. Processing data module block diagram

978-1-4673-7319-7/15/$31.00 ©2015 IEEE

A. Data Summarizing

Joint that located on the main torso such as right shoulder, left shoulder, shoulder centre, spine, hip right and hip left tend not to perform the movement with a large angle, so it used as 3D orthogonal basis for rest joints. Components principle is as explained below

- $\bar{u}$, vector with direction from upper to lower body
- $\bar{r}$, vector with direction right to left of body
- $\bar{t}$, *cross product* of two principal component, $\bar{t}=\bar{u} \times \bar{r}$

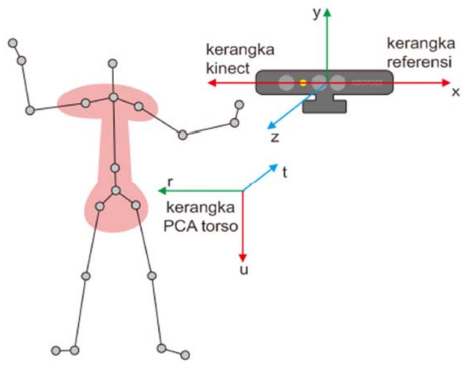

Fig. 2. Main torso and Kinect reference

1) One degree joint: Joint that directly adjacent to main torso such as right elbow, left elbow, right knee, left knee and head. On degree joint represented by elevation angle (θ) and azimuth angle (φ).

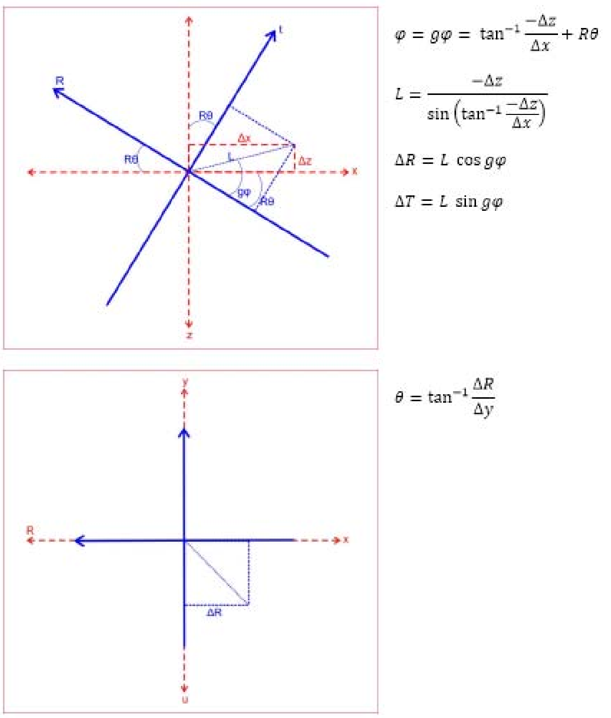

Fig. 3. Vector and one degree joint conversion formula

2) Two degree joint: Joint which is a continuation of the main torso but adjacent to the one degree joint, such as the right wrist, left wrist, right ankle and left ankle. Two degree joint represented by elevation angle (θ) relative to the reference plane of one degree joint.

$$\theta = \cos^{-1}\frac{\overline{b1}\,.\,\overline{b2}}{\|\overline{b1}\|\ \|\overline{b2}\|} \tag{1}$$

with

$$\overline{b1} = \begin{bmatrix} lututX - kakiX \\ lututY - kakiY \\ lututZ - kakiZ \end{bmatrix} \text{ and } \overline{b2} = \begin{bmatrix} lututX - pinggulX \\ lututY - pinggulY \\ lututZ - pinggulZ \end{bmatrix}$$

B. Modelling, Learning and Gesture Pattern Recognition

Hidden Markov Model is a probabilistic finite-state automata, where the transition between one states with another state is set using the probability function. In every transition, state emits an output with certain probabilities. Emission output can be multidimensional discrete symbol or continuous values. In a Markov process, transition probabilities are assumed to depend only on the previous transition (usually one), and can be modelled as a Markov chain. [1]

Complete specifications of HMM require two parameters models (number of states and number of symbols), and three kinds of probabilities (emission, transition and initialization) or commonly denoted as follows [2]

$$\lambda=(A,B,\pi) \tag{2}$$

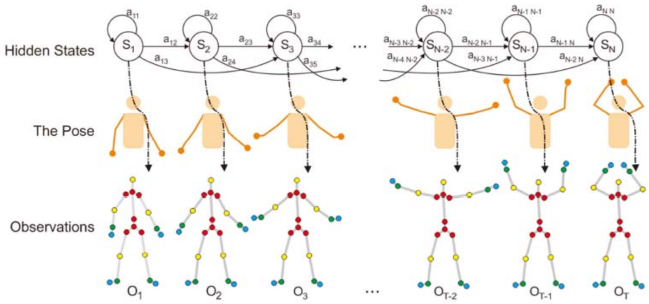

Fig. 4. HMM illustration for dance gesture pattern

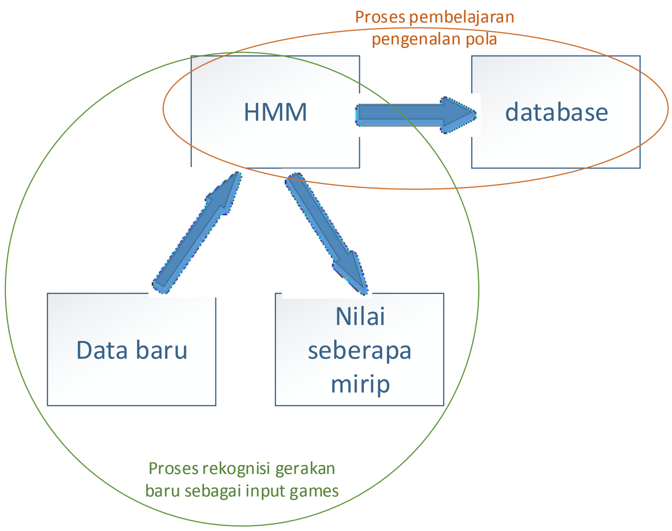

Fig. 5. HMM system diagram

1) Learning: Baum-Welch algorithm is used for learning algorithm [2]. In HMM, this algorithm is used to determine the parameters. The HMM learning process will give output three kinds matrices such as transition matrix (NxN) which represents the probability of state transition, the emission matrix (NxM) which represents the probability of observation and initialization matrix (1xN). The learning process executed iteratively until the similarity value (log-likelihood) be as convergent as possible. Convergence level determined using a threshold. The more data used for learning, the system will be more "smart" in recognizing a wide variety of new gesture pattern.

2) Recognition: Forward Algorithm is used for recognition algorithm [2]. This algorithm compare the new data with the output matrix of learning process previously. The comparison will be log-likelihood value similarly as in the learning process. If the learning process log-likelihood value is generated between the members of data training, then recognition log-likelihood value is between new data and data that has been previously recognized. If the recognition log-likelihood value is greater than the learning log-likelihood value, the new gesture can be said to be similar to the gesture that have been learned.

III. Implementation, Result and Analysis

Environment used in this implementation are Visual Studio 2013 and MATLAB R2011b. Mathematic operation and HMM implemented using library Accord.Net

A. Inverse Kinematics

The process is started by opening an external storage file of raw data, then the external file is read and assigned into an array so that make easier to access into small necessary part. The value that has been completely processed will be deleted and replaced with the value of the next timestamp so that the size of the array will always remain and do not overload the memory.

1) Input: CSV file contained XYZ coordinate of all joint at every second timestamps for a minute. Every timestamp write XYZ coordinate of 20 joints, so there are 12000 data lines for a minute recording.

```
0.08431186;0.007564744;2.444834;8308
0.08345801;0.06700049;2.496377;8308
0.0795724;0.4012219;2.482251;8308
0.07166871;0.562127;2.406232;8308
-0.07952555;0.3124072;2.471899;8308
-0.2162124;0.1741001;2.410239;8308
-0.04668151;0.1108105;2.28056;8308
0.009828798;0.08971401;2.241681;8308
0.2457982;0.3018486;2.471339;8308
0.38332;0.1305092;2.426958;8308
0.1984791;0.1111703;2.308959;8308
0.1345613;0.1406733;2.263155;8308
0.01500411;-0.06534108;2.431243;8308
-0.01952664;-0.4185228;2.348496;8308
0.0512718;-0.7417487;2.537083;8308
0.05109076;-0.7889843;2.459067;8308
0.1553428;-0.06504873;2.439198;8308
0.189171;-0.4143483;2.346116;8308
0.138358;-0.7225036;2.514903;8308
0.138177;-0.7697392;2.436888;8308
0.084046;0.008659185;2.444387;8408
0.08305166;0.06813091;2.49586;8408
0.07718657;0.4023378;2.481244;8408
0.07379061;0.5632215;2.405576;8408
-0.07964707;0.3140068;2.471836;8408
```

Fig. 6. CSV file of raw data

2) Output: .beatme formatted file contained the result of data summarizing from 12000 lines to be 600 lines for a minute recording. Every lines show discrete code of moving body part position based on its angle position and every second timestamp.

```
3;7;2;1;1;1;1;59428
2;7;2;1;1;1;1;59528
2;7;2;1;1;1;1;59628
2;7;2;1;1;1;1;59724
2;7;2;1;1;1;1;59820
2;7;2;1;1;1;1;59920
2;7;2;1;1;1;1;60024
2;7;2;1;1;1;1;60125
2;7;2;1;1;1;1;60228
2;7;2;1;1;1;1;60325
2;7;2;1;1;1;1;60424
2;7;2;1;1;1;1;60520
2;7;2;1;1;1;1;60620
1;7;2;1;1;1;1;60724
1;7;2;1;1;1;1;60824
1;7;2;1;1;1;1;60928
2;7;2;1;1;1;1;61024
2;7;2;1;1;1;1;61124
2;7;2;1;1;1;1;61220
3;7;2;1;1;1;1;61324
3;7;2;1;1;1;1;61424
2;7;2;1;1;1;1;61524
2;7;2;1;1;1;1;61624
```

Fig. 7. .beatme formatted file contained discrete value

The first digit indicates the angular position of the head. The second digit indicates the position of the right foot in terms of one degree azimuth and elevation. The third digit indicates the position of the left foot in terms of one degree azimuth and elevation. The fourth digit indicates the position combinations between right leg two degrees elevation and right hand one degree elevation. The fifth digit indicates the position combinations between left leg two degrees elevation and left hand one degrees elevation. The sixth digit is a combination between the position of the right hand two degrees elevation and one degree azimuth. The seventh digit is a combination of positions of the left hand two degrees elevation and one degree azimuth. The eighth digit indicates the timestamp changes every line as a pointer for every second sampling data.

B. Hidden Markov Model Learning

HMM learning process implemented in MATLAB because it contains iterative process that requires long time, this can interfere with real-time games process that are implemented in Visual Studio. Moreover, MATLAB can directly generate learning curve that indicate the success of the process. Another thing to consider is MATLAB has more optimal performance in processing numerical data in large quantities. This system use 5 states and 10 symbols.

1) Input: Discrete value pattern of a digit for approximately one minute recording. There are 70 data sample.

2) Output: Initialization matrix, transmission matrix, emission matrix, log-likelihood value and learning curve.

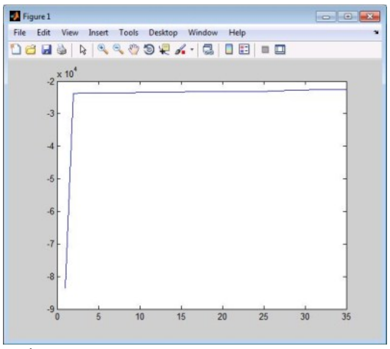

Fig. 8. Digit 1 learning curve

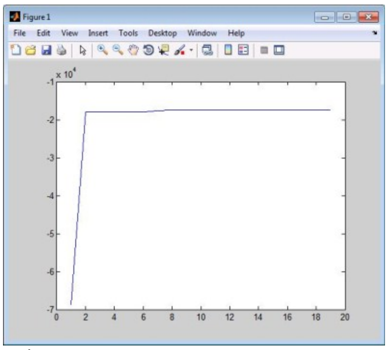

Fig. 9. Digit 2 learning curve

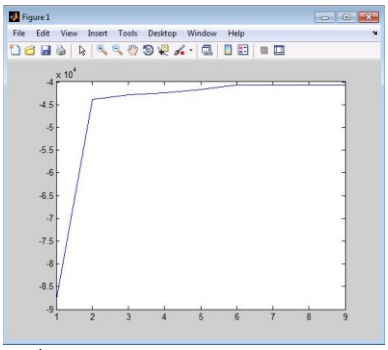

Fig. 10. Digit 3 learning curve

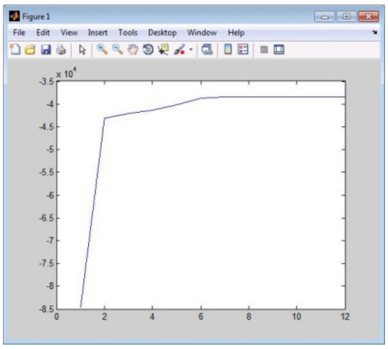

Fig. 11. Digit 4 learning curve

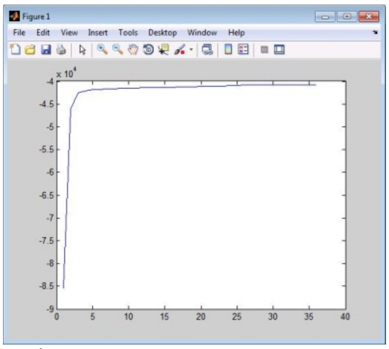

Fig. 12. Digit 5 learning curve

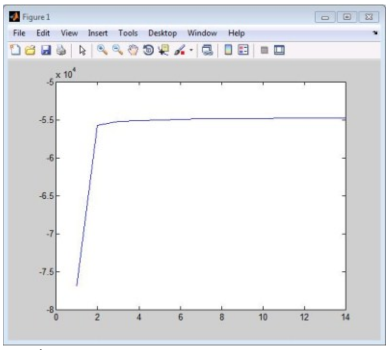

Fig. 13. Digit 6 learning curve

Generally, the learning curve tends to rise and converge towards a value. Iterative process occurs about 30-50 times. This shows that learning process has been as expected. In the learning process the number of symbols that should be used is 999,999 but the experiment show that MATLAB can only handle a maximum of 700,000 symbols. Therefore, the observation is split into per digit and can minimize the output matrix as well.

C. Hidden Markov Model Recognition

HMM recognition process implemented in Visual Studio so that the output value of this process can be directly processed further into scores in games development. In addition, the recognition process does not take too long time when compared with the learning process.

1) Input: Initialization matrix, transmission matrix, emission matrix from the learning process and discrete value pattern of a digit from new gesture that will be recognized

2) Output: Log-likelihood value of recognition

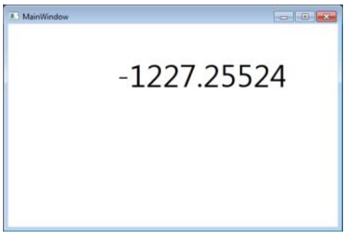

Fig. 14. Recognition log-likelihood value of digit 1

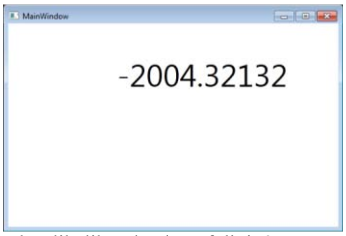

Fig. 15. Recognition log-likelihood value of digit 2

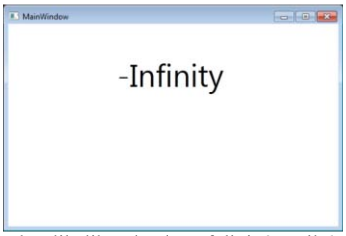

Fig. 16. Recognition log-likelihood value of digit 3 until 6

In the first and second digit log-likelihood value obtained is greater than the value of the learning log-likelihood so the new gesture patterns in the first and second digit is detected is similar to the pattern of previous gesture. In the third to the sixth digit, the value of the recognition log-likelihood generated is *-infinity*.

-infinity Value is shown as an indicator of dissimilarity between test data pattern with the previous known pattern. This can be happen because the variation in the third digit until the sixth digit observed to be very large and tend not to form a pattern, making it difficult to determine the similarities with the new data. Variations become so big and not developed a pattern because of two degree joint. When the body is not facing straight on Kinect, there will be parts of the body are covered by other parts. This causes the value of the acquisition data for the third to the sixth digit is often get randomized value and be not patterned. One degree joint produces discrete value variation much more stable because of its position closer to the main torso.

IV. Conclusions

This paper discussed about big data summarizing from XYZ coordinates of the joint into a more compact data but still representative. Then conducted by making learning system using Hidden Markov Model which includes the process of learning and recognition. The learning process has a good performance in MATLAB that can process large amounts of data as well as provide facilities to display a learning curve with limits of 700,000 symbols. Learning curve generated throughout the learning data digit has a tendency to rise and converge at a value. The new gesture pattern recognition process has a good performance within one degree joint because of its position close to the main torso as reference. Recognition on two degree joint is not always indicated similar to the learning data because it has large variations due to body parts covered with other body parts randomize value when the main torso is not facing straight on Kinect.

Acknowledgment

The author greatly appreciate the contributions of Mrs. Nurfitri Anbarsanti, Mrs. Aciek Ida Suryandari, Mr. Yoga Priyana and Mr. Ary Setijadi for the guidance and the teaching. The author is also appreciate the help of *BeatMe! Project* team for their help to developed and integrated the whole system.

References

[1] N. Anbarsanti, A. Setijadi. “Dance Learning and Recognition System based on Hidden Markov Model. A Case Study : Aceh Traditional Dance”. 2014.

[2] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE,* vol 77 no.2, Februari 1989.

[3] L. R. Rabiner, B. H. Juang. “An Introduction to Hidden Markov Models”. *IEEE ASSP Magazine*. January 1986.

[4] N. Freitas. “CPSC340 : Hidden Markov Models”.*University of British Columbia*. September 2012.

[5] G. Pettersson, J. Tysk, H. Vallgren. “Conversion of Hidden Markov Model computation to C#”. *Uppsala Universitet.* Januari 2013.

[6] C. Souza. (2010) Hidden Markov Models in C#. [On*line*]. Available: http://www.codeproject.com/Articles/69647/Hidden-Markov-Models-in-C/

[7] C. Souza. (2014) Sequence Classifiers in C# - Part I: Hidden Markov Models. [On*line*]. Available: http://www.codeproject.com/Articles/541428/Sequence-Classifiers-in-Csharp-Part-I-Hidden-Marko/

[8] (2009) Accord .Net Framework: HiddenMarkovModel Class. [On*line*]. Available: http://accord-framework.net/docs/html/T_Accord_Statistics_Models_Markov_HiddenMarkovModel.htm/

[9] (2009) Accord .Net Framework: BaumWelchLearning Class. [On*line*]. Available: http://accord.googlecode.com/svn-history/r667/docs/html/T_Accord_Statistics_Models_Markov_Learning_BaumWelchLearning_1.htm/

[10] J. C. Hall. (2011) How to Do Gesture Recognition with Kinect using Hidden Markov Models. [On*line*]. Available: http://www.creativedistraction.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/

[11] M. F. Iskander. “Electromagnetic Fields and Waves”. *Waveland Press, Inc*. 1992.

[12] J. Wood. (2013) Reading and Writing CSV *File*s in C#. [On*line*]. Available: http://www.blackbeltcoder.com/Articles/*files*/reading-and-writing-csv-*file*s-in-c/

[13] (2013) Morgan Tech Space: How to read data from csv *file* in c#. [On*line*]. Available: http://www.morgantechspace.com/2013/08/how-to-read-data-from-csv-*file*-in-c.html/