Identification of Odissi Dance Video Using Kinect Sensor

Sriparna Saha[1], Shreya Ghosh[2], Amit Konar[3]
[1,3] Electronics and Telecommunication Engineering Dept.
[2] School of Bioscience and Engineering
Jadavpur University
Kolkata, India
{sahasriparna, shreyaghosh215}@gmail.com,
konaramit@yahoo.co.in

Ramadoss Janarthanan
Computer Science & Engineering Dept.
TJS Engineering College
India
srmjana_73@yahoo.com

***Abstract*— This paper introduces an algorithm for identification of dance video by recognizing posture from each frame for the purpose of e-learning. We are taking Indian classical dance 'Odissi' as the input. The twenty videos 'Chowkh' and 'Tribhangi' of 'Odissi' dance have been recognized using Kinect sensor, which is used for visual sensing. With the help of Kinect, we obtain a set of twenty body junction coordinates out of which only sixteen are required for our proposed work. A unique and simple methodology has been adopted to distinguish between the postures based on the distance and angle between the different joint coordinates. The average joint information values from each hand and leg are processed and they form the four vertices of a 4-sided polygon. The value of the four edges and four angles are to obtain from the polygon to train SVM. The experimental details show that this algorithm performs with a high recognition rate of 92.7% using SVM.**

Keywords—4-sided polygon; Euclidean distance; motion; posture; multi-class SVM

I. INTRODUCTION

In our proposed work, we are taking dance videos of Indian classical dance. 'Odissi', an Indian classical dance form, involves intricate footwork, graceful body movements and expression of feelings. It is a combination of 'Ragam' (melody), 'Talam' (rhythm), 'Nritta' (dance) and 'Rasa' (feelings). The languid grace of 'Odissi' is contributed by the different postures in it. The most important postures in 'Odissi' are 'Chowkh' and 'Tribhangi'. The posture 'Chowkh' resembles a square, thus forming four right angles in the body and ensuring equal weight distribution on both the legs of the subject. 'Tribhangi' on the other hand is a three-body-bend position, where one half of the lower body remains static along a vertical axis while the other leg crosses the first. The part of the body corresponding to the other leg moves the torso in the opposite direction. At the same time the head and the neck moves in the same direction. This is characterized by unequal distribution of weights on the legs. The permutation and combination of the above mentioned postures forms the basis of the 'Odissi' dance form.

In this paper, we propose a methodology or algorithm to recognise the whole body postures from each frame [1] in the dance videos of 'Odissi'. 'Chowkh' and 'Tribhangi' each consists of 10 videos. Thus 20 videos are taken as the inputs. Each of the video length is 20 sec. Each frame in the video corresponds to a specific posture.

The motion analysis has been carried out by the Kinect sensor, which tracks the skeleton of the dancer, while performing [2]. The 3D image representation of a human being is detected by Kinect sensor. It tracks the skeleton of the person standing in front of it using visible and IR cameras. A total of sixteen joint coordinates have been considered for our proposed work. They include the hip, shoulder, elbow, hand, wrist, knee, ankle and foot joints.

In today's world, internet plays a very important role. Hence the proposed algorithm helps to globalize an Indian dance form for the purpose of e-learning. 'Chowkh' and 'Tribhangi' are highly specialized features of 'Odissi'. So anybody in any part of the world can learn these videos and evaluate if it is correct via an e-learning program. The purpose of this algorithm is to serve as an e-learning program, as it offers flexibility, accessibility and cost effectiveness for the dancers world-wide.

In this proposed work, the recognition of dance video is carried out by Support Vector Machine (SVM). It is basically a binary classifier, but multi-class recognition is also possible, as it gives a very good result there also. Data-mining and pattern recognition are the two important aspects of SVM [3]. It has many applications in pattern classification like military, economic and many other fields.

Few works in other dance forms have already been carried out previously. 'Ballet' poses [4] are modelled using stick figure diagrams. The postures from videos of 'Ballet' are sequenced based on the skeleton obtained using motion capturing devices. Three angles of any leg are taken as a feature and the other leg is kept static. The authors of [5] uses Markov model along with K-means algorithm for removal of ambiguity in posture recognition. For posture recognition, background image plays an important role. Noisy background, improper texture of the dress and an inappropriate distance of the dancer from the camera leads to erroneous results [6]. In our case, since we are using Kinect sensor, the above

978-1-4673-6217-7/13/$31.00 ©2013 IEEE

mentioned problems are easily removed. In [7], the authors detect basic human postures with the help of active contours and neural networks. Here background subtraction technique is implemented at the pre-processing stage. However, the effectiveness of the algorithm is limited to only three postures-sitting, bending and squatting. Another approach used for human posture recognition uses decision tree and is discussed at length in [8]. Decision tree is based on entropy which measures information gain [9]. A fuzzy theory based approach to posture recognition uses a genetic fuzzy finite state machine. Here inputs to the state machine are provided and outputs are obtained using fuzzy inference rules [10]. Comparison of supervised and unsupervised learning classifiers for human posture recognition is undertaken in [11].

With reference to human-computer interaction [12], posture recognition plays an important role. In [13], a condensation algorithm implements a visual tracking system. In [14], real time MPEG video inputs are provided and AC-DCT coefficients are used to obtain Eigen space representation of human silhouettes. Two cameras are required for this work each along with a standard camera and video processing board. According to [15] view-invariant posture recognition is much more complex than view-variant posture recognition. Authors of [16] deal with common postures like standing, sitting, bending etc. using accelerometer and show that accelerometers are better than location sensor for posture recognition purpose.

Kinect sensor generates depth as well as RGB data. In [17], we find that a time-of-flight depth camera along with a pulse coupled neural network (PCNN) is being used for recognition of human body in distorted background. Here hierarchical decision tree is used for the classification purpose. Inverse kinematics is applied for upper body tracking in [18]. As head and hand region are normally well-defined and less amount of noise is present in those parts, so tracking is much more reliable. Here the authors achieved more than 90 percent accuracy. Punching, waving, clapping and dumbbell are the main gestures taken as the inputs [18]. Thus the algorithm works for simple body movements while our proposed work takes complex body postures as inputs.

Datasets are prepared for 7 dancers and training to testing ratio is 4:1. The time complexity of the proposed algorithm is 4.726 sec in an Intel Pentium Dual Core processor running Matlab R011b for a video length of 20 sec. Recognition rate is as high as 92.7% using multi-class SVM. So the work gives a high level of accuracy. Secondly the proposed work in our paper is applicable in different lighting conditions. In various illumination conditions, where human eye can detect object, Kinect sensor also detects the skeleton of the human body. This work is not only useful for Indian dance forms but also can be used for other international dance types for identification of a video [19]. Moreover, since we are using average value and not the exact one, noise is greatly reduced.

In this paper, Section II elaborates the experimental setup used for this proposed work while Section III shows the problem formulation and approach. Section IV explains the experiment details and section V concludes with some idea about future work.

II. EXPERIMENTAL SETUP

The experimental setup is shown in Fig. 1. Here the dancer needs to stand in front of the Kinect sensor within a particular range of distance, such that the whole body skeleton of the dancer is recognized correctly. The distance within which the subject should stand or perform in front of the Kinect is roughly 1.2 to 3.5m or 3.9 to 11ft. A white background is used to remove any sort of sort noise which can affect the skeleton detection process. Two standing lights are placed in front of the background to ensure pleasant lighting condition. Kinect sensor detects human body skeleton in almost all lighting conditions, but to obtain good quality RGB video, the standing lights are placed.

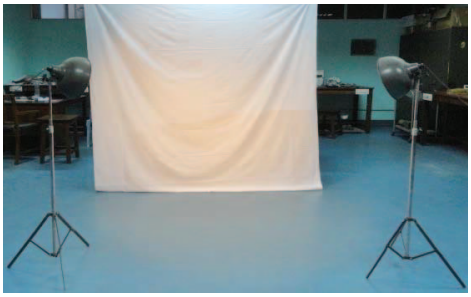


Fig. 1. Experimental Setup

The Kinect sensor basically looks like a webcam as shown in Fig. 2. It has the appearance of a long horizontal bar with a motorized base. It has the ability to track the skeleton of the person standing in front of it. It has visible, as well as IR cameras. The IR cameras make the device capable of high resolution depth sensing. The depth sensor is a combination of Infrared Laser and a CMOS sensor. This depth sensing technology can detect and capture 3D video data under any light conditions. The range of sensing in the depth sensor can be adjusted over a specified range with the Kinect sensor being able to calibrate it based on the surrounding physical conditions and the presence of furniture or any other wooden object.

Based on the resolution, the Kinect sensors have an output video frame rate of about 9 Hz to 30 Hz. An 8-bit VGA resolution of 640 × 480 pixels with a Bayer color filter is used by the RGB videos. However, the hardware can also be used for resolutions up to 1280x1024 pixels and other formats such as UYVY. A VGA resolution of 640 × 480 pixels having 11-bit depth is possessed by the monochrome depth sensing video stream. This provides 2,048 levels of sensitivity.

Fig. 2. Kinect Sensor

III. Problem Formulation and Approach

In this part, we describe the features which have been used to design the proposed algorithm. Sixteen junction coordinates have been used for the purpose of designing the algorithm. We have taken average co-ordinate value for each hand and leg of the dancer. Since body proportion of different dancers can vary widely, considering average value reduces noise and hence the recognition rate increases.

A. Formation of Vertices of 4-sided Polygon

Each video is 20 seconds long and the Kinect captures at 30 frames per second rate. Hence a total of 600 frames is obtained. For each frame sixteen joint co-ordinates are processed to obtain four vertices of the 4-sided polygon. The average value of shoulder right (SR), elbow right (ER), wrist right (WR) and hand right (HR) forms the first vertex (Vertex_1). Similarly, four left hand co-ordinates, shoulder left (SL), elbow left (EL), wrist left (WL) and hand left (HL) produce the second vertex (Vertex_2). For right leg, hip right (HR), knee right (KR), ankle right (AR) and foot right (FR) co-ordinates are used to obtain third vertex (Vertex_3). In the similar fashion, left leg co-ordinates, such as hip left (HL), knee left (KL), ankle left (AL) and foot left (FL) form fourth vertex (Vertex_4). As Kinect takes 3D information of the human body, so each co-ordinate has 3D values, x, y and z. Here z direction shows the distance of the subject from the camera. The equations used for vertex formation are shown in (1-4).

$$Vertex\_1 = \frac{SR + ER + WR + HR}{4} \tag{1}$$

$$Vertex\_2 = \frac{SL + EL + WL + HL}{4} \tag{2}$$

$$Vertex\_3 = \frac{HR + KR + AR + ER}{4} \tag{3}$$

$$Vertex\_4 = \frac{HL + KL + AL + EL}{4} \tag{4}$$

B. Calculation of Length of Edges

Now, the Euclidean distance between any two vertices is calculated. Let the two vertices be V_i and V_j having co-ordinates (x_i,y_i,z_i) and (x_j,y_j,z_j) respectively. The distance between these two vertices is calculated by (5). As we have created four vertices, thus we obtain four distances. These Euclidean distances denote the length of the edges of the polygon.

$$Dist = \left\| V_i - V_j \right\| \tag{5}$$

where, Dist is the Euclidean distance between the vertices

First edge is formed between first and second vertices. Second and fourth vertices are joined by second edge and third edge is obtained by joining fourth and third vertices. The remaining edge of the polygon, i.e., and fourth edge is produced using third and first vertices co-ordinate information.

C. Determination of Angles

Vital information about any polygon is its angle values. While structuring a 4-sided polygon, four angles values are produced. Let, the co-ordinates of first three vertices be (a_1, b_1,c_1), (a_2,b_2,c_2) and (a_3,b_3,c_3) respectively. The vectors v1 and v2 formed by these three vertices are shown in Fig. 3.

$$v1 = (a_1 - a_2)\vec{i} + (b_1 - b_2)\vec{j} + (c_1 - c_2)\vec{k} \tag{6}$$

$$v2 = (a_3 - a_2)\vec{i} + (b_3 - b_2)\vec{j} + (c_3 - c_2)\vec{k} \tag{7}$$

The angle between two vectors is calculated by (8).

$$\text{angle} = \frac{\text{atan2}(\text{norm}(\text{cross}(v1,v2),\text{dot}(v1,v2)))}{\text{pi}} \times 180^0 \tag{8}$$

where norm function returns a matrix.

Cross product or vector product is defined by

$$\vec{M} \times \vec{N} = mn \sin\theta \vec{P} \tag{9}$$

where m and n are the magnitudes of $\vec{M}$ and $\vec{N}$ vectors correspondingly, θ is the angle between the two vectors and $\vec{P}$ be the unit vector normal to the plane $\vec{M}$ and $\vec{N}$.

Dot product or scalar product is defined by

$$\vec{M} \bullet \vec{N} = mn \cos\theta \tag{10}$$

atan2 is the arctangent angle whose range is $\left(-\pi/2, +\pi/2\right)$.

$$\text{atan}\,2(y,x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \\ \arctan\left(\frac{y}{x}\right) + \pi & x > 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & y \geq 0, x < 0 \\ & y < 0, x < 0 \\ +\pi/2 & y > 0, x = 0 \\ -\pi/2 & y < 0, x = 0 \\ \textit{undefined} & y = 0, x = 0 \end{cases} \tag{11}$$

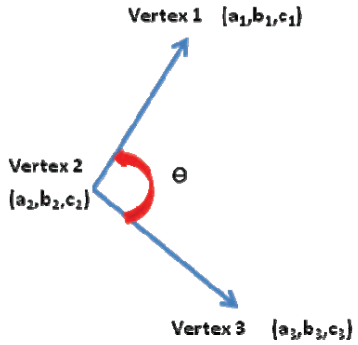


Fig. 3. Calculation of angle between first three vertices

In the above mentioned process, all the other three angles are calculated. Fig. 4 demonstrates the modelling of the polygon in the skeleton view of 1st video of 'Chowkh' for frame no 150. Skeleton of the subject has twenty joints, but in this proposed work we are using only sixteen joints. Those are shown by the magenta color cubes. The other four joints are marked by circles. All the twenty joints are joints via blue line. While the four vertices of the polygon are shown by black stars, the red dotted lines are the four edges of the polygon. We can easily see that the noise due to variation of human body structure is neglected as we are taking an average value and not the actual ones. The elegant hand movements of 'Odissi' dance is modeled by this 4-sided polygon.

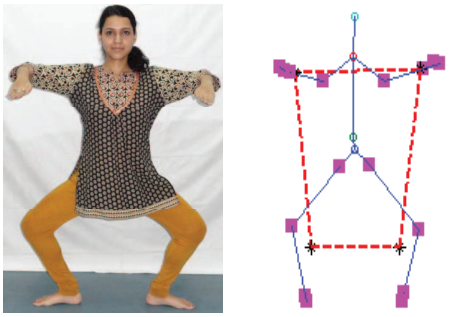


Fig. 4. RGB image with skeleton showing 4-sided polygon

IV. Experiment Details

Algorithm for the Proposed Work

Step 0 Create a dataset of skeletons for all the training videos.

BEGIN

Step 1 Separate the sixteen joints based on whether they belong to hand or leg body parts for 1st frame of the unknown video.

Step 2

If

Joints belong to right hand, then call the average function and form Vertex_1.

Else if

Step 3 Joints belong to left hand, then call the average function and form Vertex_2.

Else if

Step 4 Joints belong to right leg, call the average function and produce Vertex_3.

Else

Step 5 All the other remaining joints belong to left leg, thus Vertex_4 is formed after calling the average function.

End if

Step 6 Calculate Euclidean distance between Vertex_1 and Vertex_2 obtained from Step 2 and 3.

Step 7 Calculate Euclidean distance between Vertex_2 and Vertex_4 produced Step 3 and Step 5.

Step 8 Calculate Euclidean distance between Vertex_4 and Vertex_3 obtained from Step 5 and 4.

Step 9 Calculate Euclidean distance between Vertex_3 and Vertex_1 produced Step 4 and Step 2. Hence the four sided polygon is formed.

Step 10 Measure the four internal angles of the polygon.

Repeat steps 1-10 for 600 frames of the unknown video length of 20 sec.

Step 11 Run multi-class SVM to identify the correct video based on the features extracted by the above steps.

END

TABLE I. Feature Extracted for 5th Video of Travangi (Result showing for frame no 100, 200, 300)

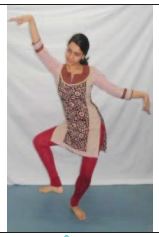
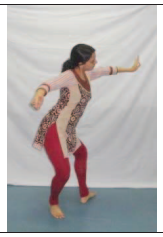
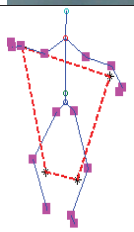
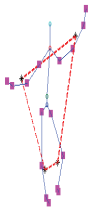
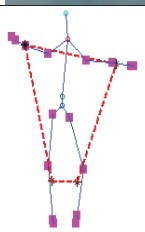
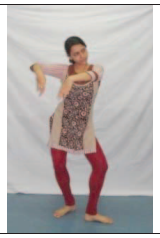
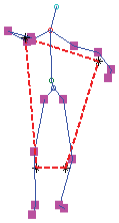
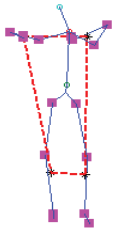
Information	Frame No 100	Frame No 200	Frame No 300
RGB Image			
Skeleton View			
Value of Edges	0.680903 0.829513 0.257619 0.717603	0.799684 0.745134 0.186736 0.971968	0.725144 0.900993 0.194994 0.821953
Value of Angles	63.1703 110.095 99.4253 85.4642	88.302 97.5855 119.693 51.9851	66.3282 103.246 108.057 76.8539

TABLE II. FEATURE EXTRACTED FOR 5TH VIDEO OF TRAVANGI (RESULT SHOWING FOR FRAME NO 385, 456)

Information	Frame No 385	Frame No 456
RGB Image		
Skeleton View		
Value of Edges	0.575317 0.84982 0.207326 0.725662	0.365402 0.875204 0.188503 0.881825
Value of Angles	66.5126 98.1452 108.184 87.0843	80.8539 103.98 87.3599 87.3403

The 5th video of 'Tribhangi' is presented in Table I and II. frame no 100, 200, 300, 385 and 546 are shown each with RGB image and their corresponding skeleton view. The length and the angle value of the four edges for those particular frames are also given. From Table I and II, it is very prominent that modelling of 'Odissi' dance, which consists of a wide change in hand and leg movements, is achieved by this polygonal approach. Also we are taking an average value and not the actual ones, so noise is reduced.

The time complexity of the proposed algorithm is 4.726 sec in an Intel Pentium Dual Core processor running Matlab R011b for a video length of 20 sec. A high accuracy rate of 92.7% is obtained using multi-class SVM. We have taken datasets from 7 dancers, out of which 80% datasets are used as training purpose and remaining part is considered for testing. Each video is taken at 30 frames per sec rate, thus for 20 sec video, 600 frames are produced. In each frame, a specific skeleton is obtained, which is modeled by a 4-sided polygon. Each polygon formed in each frame has four edges and four angles. Thus we are having eight parameters for each frame in the video. For each video of twenty sec, total 600 × 8 features are obtained. As twenty total videos are taken as inputs so 600 × 8 × 20 total no of parameters are processed for each person.

High change in intensity of light does not affect our system, as Kinect can detect human body in wide change of lighting conditions. Thus this algorithm is apt for e-learning of Indian classical dance. The promotion of the rich Indian culture world-wide is the objective of our work. Though we are using 'Odissi' as our dataset, our algorithm is not limited to a particular dance form. This proposed logic is applicable for any type of dance forms, be it Indian or Western. A wide range of hand movements is tackled by this algorithm. Hence it is applicable for any type of dance form to promote it via e-learning.

V. CONCLUSION AND FUTURE WORK

In this is the proposed work, we have achieved a high recognition rate of 92.7% with a very less timing complexity of 4.726 sec in an Intel Pentium Dual Core processor running Matlab R011b for a video length of 20 sec. For the recognition purpose we are using multi-class SVM. 7 dancers have performed for the preparation of datasets. 80% data are taken for training process and 20% for testing.

In [4], sequencing of 'Ballet' is done using human motion sensing device but this algorithm takes only the leg joint information and is designed when only one leg is moving and other is kept constant. On the other hand, our algorithm does not have this type of restriction; the whole body movement is modelled by the proposed work simultaneously. This algorithm is applicable for any type of dance form. We are using 'Odissi' dance for its elegant and graceful movements. As our algorithm deals with large change in hand and leg movements, so any other type of dance forms can easily be tackled by this proposed work. There are variations in the body proportions of different dancers, but this does not affect our system, as we are taking average value. Thus noise is reduced considerably. A low cost Kinect sensor has been used to obtain information, which has played an intricate role in designing the algorithm. So promotion of Indian dance style can be achieved cost effectively. As Kinect sensor is used to obtain the body co-ordinates, thus there is no scaling effect.

In this paper we have worked with twenty different dance videos of single dancer performing 'Odissi'. However, there are many other complex dance forms, like 'Salsa', where two dancers performing at the same time and interacting with each other. With further development in the methodology we are trying to recognize the entire dance video with two persons in which we are currently working on.

ACKNOWLEDGMENT

We would like to thank University Grant Commission, India, University of Potential Excellence Programme (Phase II) in Cognitive Science, Jadavpur University.

REFERENCES

[1] L. Snidaro, G. L. Foresti, and L. Chittaro, "Tracking human motion from monocular sequences," *International Journal of Image and Graphics*, vol. 8, no. 03, pp. 455–471, 2008.

[2] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using Kinect," in *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*, 2012, pp. 185–188.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[4] A. LaViers, Y. Chen, C. Belta, and M. Egerstedt, "Automatic sequencing of ballet poses," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 3, pp. 87–95, 2011.

[5] Y. Lee and K. Jung, "Non-temporal Mutliple Silhouettes in Hidden Markov Model for View Independent Posture Recognition," in *Computer Engineering and Technology, 2009. ICCET'09. International Conference on*, 2009, vol. 1, pp. 466–470.

[6] P. Silapasuphakornwong, S. Phimoltares, C. Lursinsap, and A. Hansuebsai, "Posture recognition invariant to background, cloth textures, body size, and camera distance using morphological geometry," in *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, 2010, vol. 3, pp. 1130–1135.

[7] F. Buccolieri, C. Distante, and A. Leone, "Human posture recognition using active contours and radial basis function neural network," in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, 2005, pp. 213–218.

[8] N. M. Tahir, A. Hussain, S. A. Samad, and H. Hussin, "On the Use of Decision Tree for Posture Recognition," in *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*, 2010, pp. 209–214.

[9] A. Konar, *Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain*, vol. 1. CRC press, 1999.

[10] A. Alvarez-Alvarez, G. Trivino, and O. Cordón, "Body posture recognition by means of a genetic fuzzy finite state machine," in *Genetic and Evolutionary Fuzzy Systems (GEFS), 2011 IEEE 5th International Workshop on*, 2011, pp. 60–65.

[11] K. K. Htike and O. O. Khalifa, "Comparison of supervised and unsupervised learning classifiers for human posture recognition," in *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, 2010, pp. 1–6.

[12] T. Winkler, "Creating interactive dance with the very nervous system," in *Proceedings of the 1997 Connecticut college symposium on art and technology, New London, Connecticut*, 1997, pp. 212–217.

[13] A. Chella, H. Dindo, and I. Infantino, "A system for simultaneous people tracking and posture recognition in the context of human-computer interaction," in *Computer as a Tool, 2005. EUROCON 2005. The International Conference on*, 2005, vol. 2, pp. 991–994.

[14] L. B. Ozer and W. Wolf, "Real-time posture and activity recognition," in *Motion and Video Computing, 2002. Proceedings. Workshop on*, 2002, pp. 133–138.

[15] C. M. Mak, Y. Lee, and Y. H. Tay, "Causal Hidden Markov Model for view independent multiple silhouettes posture recognition," in *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*, 2011, pp. 78–84.

[16] H. Gjoreski, M. Lustrek, and M. Gams, "Accelerometer placement for posture recognition and fall detection," in *Intelligent Environments (IE), 2011 7th International Conference on*, 2011, pp. 47–54.

[17] H. Zhuang, B. Zhao, Z. Ahmad, S. Chen, and K. S. Low, "3D depth camera based human posture detection and recognition Using PCNN circuits and learning-based hierarchical classifier," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, 2012, pp. 1–5.

[18] C. Tran and M. M. Trivedi, "3-D posture and gesture recognition for interactivity in smart spaces," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 178–187, 2012.

[19] M.-H. Lee, U.-M. Kim, and J.-I. Park, "An Analysis Methodology on Emotion of Korean Traditional Dance Using a Virtual Reality System," in *Culture and Computing (Culture Computing), 2011 Second International Conference on*, 2011, pp. 149–150.