




Real-time dance evaluation by markerless human pose estimation

Yeonho Kim¹ · Daijin Kim¹ 

Received: 9 August 2017 / Revised: 27 March 2018 / Accepted: 29 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract This paper presents a unified framework that evaluates dance performance by markerless estimation of human poses. Dance involves complicated poses such as full-body rotation and self-occlusion, so we first develop a human pose estimation method that is invariant to these factors. The method uses ridge data and data pruning. Then we propose a metric to quantify the similarity (i.e., timing and accuracy) between two dance sequences. To validate the proposed dance evaluation method, we conducted several experiments to evaluate pose estimation and dance performance on the benchmark dataset EVAL, SMMC-10 and a large K-Pop dance database, respectively. The proposed methods achieved pose estimation accuracy of 0.9358 mAP, average pose error of 3.88 cm, and 98% concordance with experts' evaluation of dance performance.

Keywords Human pose estimation · Dance performance evaluation

1 Introduction

Pose estimation methods have been improved dramatically since the introduction of Kinect, which is a reasonably-priced depth sensor. These methods include using an interest-point detector to localize body parts in a single depth image [16], and consulting a database to

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11042-018-6068-4>) contains supplementary material, which is available to authorized users.

✉ Daijin Kim
dkim@postech.ac.kr

Yeonho Kim
beast@postech.ac.kr

¹ Computer Science and Engineering, Pohang University of Science and Technology, Pohang, South Korea

obtain intermediate representations of body parts [20]. A method that uses random tree walk for each joint [12] is fast; although it cannot distinguish front from back views of a human, it considerably improved the accuracy compared to related methods.

Convolutional neural networks (CNNs) have been successfully applied to the task of pose estimation. They have been used to directly regress the 2D Cartesian coordinate of body joints in a holistic manner [1], to intermediate representations (heat-maps) for all joints to refine the joint positions in 2D pose estimation [3], and to use depth images to regress joint positions for 3D pose estimation [9].

Since pose estimation methods have become reliable, many real-world applications are now realistic. To recognize human actions, the spatial differences between detected joints are encoded by several features along with temporal differences [23] [22]. Proposed methods to recognize human behaviors use a hidden Markov model (HMM) [21] or dynamic time warping (DTW) [18]. Moreover, various aspects of activity-recognition have been explored, such as off-line activity recognition [10], on-line activity recognition [11], and human-to-human interaction recognition [25]. Most applications have focused on classifying human activities, but few methods have been introduced to evaluate the precision of human actions, such as dance gestures [17], music-conducting gestures [19], or performance-evaluation in health care [15].

In this paper, we propose a unified framework that uses markerless estimation of human poses to evaluate dance performance. We develop a method to estimate human pose by using ridge data and data pruning. The method is invariant to rotation and occlusion. After estimating the poses, we extract the dance feature to measure the similarity of timing accuracy and pose accuracy between dance performances of teacher and learner. To the best of our knowledge, this is the first time to evaluate the qualitative and quantitative aspect of human performance by using a large K-Pop dance database.

The rest of the paper is organized as follows. Section 2 describes the proposed human pose estimation. Section 3 describes the proposed dance evaluation method. Section 4 validates the proposed methods using a benchmark database and a large K-Pop dance database. Finally, Section 5 presents conclusions.

2 Human pose estimation

We develop a pose estimation method that is invariant to rotation and occlusion, because dance involves complicated poses such as full-body rotation and self-occlusion. The proposed method takes a depth image that has been captured by a Kinect camera, where the image has a size of 640×480 , and each pixel $X_i = (x_i, y_i, z_i)$ represents 3D coordinates.

Our method (Fig. 1) first removes the floor and extracts human depth silhouettes from a sequence of depth images. Then it extracts ridge data from the silhouette by finding

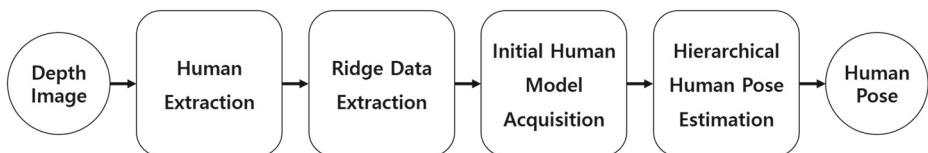


Fig. 1 Overall process of the proposed human pose estimation

local maxima in the distance map; these ridge data approximate the positions of the long bones. Then it generates a set of initial human model parameters for body parts with a tree-structured kinematic model and a specially-proposed head-shoulder-torso structure. Finally, the model estimates the positions of human body joints by tracking human body parts in a hierarchical top-down manner after eliminating invalid ridge data by consulting predefined model parameters and by constraining the local area of search for each body part.

2.1 Human extraction

The process of human extraction consists of four steps: floor removal, object segmentation, human detection, and human identification (Algorithm 2).

First, we use a floor-removal technique that disconnects all objects from the floor. For this process, we model the floor as a plane

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \quad (1)$$

where d_i denotes the distance from the floor plane, for which the normal vector is $(a, b, c)^T$ and n is the number of floor candidates.

We exploit the fact the floor has smaller y than objects. (1) We obtain the y values of the raw depth data by setting (a, b, c) to $(0, 1, 0)$. (2) We collect the depth data $X_i = (x_i, y_i, z_i) (i = 1, \dots, n)$ that have the smallest y values and identify them as floor candidates. (3) We obtain the normal vector (a, b, c) from the floor candidates by using the least squares method in (1). (4) We compute the distances $d_i (i = 1, \dots, N)$, where N is the number of pixels in the depth data, by taking the inner product of all depth data and the normal vector (a, b, c) . (5) We remove the floor by eliminating pixels that have distances less than a threshold.

Second, we use the 3D-connected component-labeling (CCL) technique [5] to segment the objects in the original depth image. The CCL algorithm uses a binary connectivity criterion to assign a unique label to each connected component. To process the depth image, we modify the binary connectivity criterion that compares a threshold depth value ϵ_z to the difference in the depth values $d(\cdot)$ of neighboring pixels (X_i, X_{i+1}) ; when $|d(X_i) - d(X_i + \mathbf{i})| \leq \epsilon_z$, two pixels are included in the same object.

Third, we identify human objects among the segmented objects by assuming that only humans move. The motion information can be obtained by computing the depth difference between two consecutive frames $t - 1$ and t as $\delta d(X_i) = d^t(X_i) - d^{t-1}(X_i)$, where subscripts $t - 1$ and t denote the frame indices. We define the sum of depth differences within a bounding box of the segmented object ($\Delta d(B_i) = \sum_{X_i \in B_i} \delta d(X_i)$) as a measure to discriminate moving objects from stationary objects. If $\Delta d(B_i)$ exceeds the threshold motion value ϵ_m , the segmented object is identified as a human.

Finally, we assign a unique ID to each detected human by applying a simple association rule that computes the distance between a specific human in the current frame and all humans in the previous frame. We assume that this human in the current frame is the one for whom this distance is smallest.

Algorithm 1 Human Extraction

Input: X : depth pixels, A_0 : initial floor normal, d_0 : initial floor distance
Output: Segmented human image *labels*

```

/* Floor removal                                     */
 $\mathbf{X}_f = \{X_n | X_n A_0 - d_0 < \delta_f\};$ 
 $i = i + 1;$ 
repeat
     $\mathbf{d}_{i-1} = \mathbf{X}_f A_{i-1};$ 
     $A_i = (\mathbf{X}_f^T \mathbf{X}_f)^{-1} \mathbf{X}_f^T \mathbf{d}_{i-1};$ 
     $\mathbf{X}_f = \{X_n | X_n A_i - \bar{\mathbf{d}}_{i-1} < \delta_f\};$ 
     $i = i + 1;$ 
until  $\mathbf{X}_f$  is changed;
 $X = X \setminus \mathbf{X}_f;$ 

/* 3D connected components labeling                 */
for  $j = 1$  to  $|X|$  do
    if  $X_j$  is not zero vector then
        neighbors = connected depth pixels with the similar depth value;
        if neighbors is empty then
            linked[NextLabel] = labelsj = NextLabel;
            Nextlabel += 1;
        else
            labelsj = min(neighbors labels);
            linked[label] = union(linked[label], L);
        end
    end
end

/* Human identification                             */
for  $j = 1$  to  $|X|$  do
    if  $X_j$  is not zero vector then
        labelsj = find(labelsj);
    end
end
end

```

2.2 Ridge data extraction

The ridge data are composed of a set of depth pixels that are located in the local maximal region of a distance transform map (DTM) [13]. The process of generating ridge data from the depth image involves four steps (Fig. 2).

- (1) Build the depth edge image that is obtained by thresholding the depth difference between two neighboring pixels (Fig. 2a).
- (2) Build a DTM that is computed by the pixel distance $D_T(X_i)$ at the pixel position X_i in the DTM (Fig. 2b).
- (3) Find the local maximum in the distance map by drawing a circle $C(X_i)$ with a radius of $D_T(X_i)$ at X_i in the DTM, compute the average of D_T along the circumference of the circle, and compute the ratio R_T of the average D_T over the pixel's $D_T(X_i)$ (Fig. 2c).

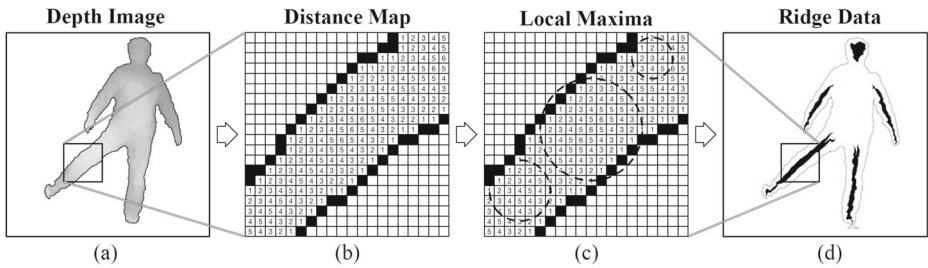


Fig. 2 Overall process of generating ridge data from the depth image.

- (4) Take pixels for which R_T is smaller than a given threshold value as the local maximal pixels, i.e., ridge data (Fig. 2d).

Mathematically, we can define the ridge data $R(I)$ of the input depth image I as

$$R(I) = \left\{ X_i \in I \mid \frac{\sum_{X \in C(X_i)} D_T(X)}{\sum_{X \in C(X_i)} I(X)} \leq \epsilon_r \cdot D_T(X_i) \right\}, \quad (2)$$

where $D_T(X_i)$ is the pixel distance in the DTM at the pixel position X_i , $I(X)$ is an indicator function for which value is 1 if the pixel position X is located on the circle $C(X_i)$ and 0 otherwise, and ϵ_r is a parameter that controls the amount of ridge data; the richness of the ridge data increases as ϵ_r increases. In this work, we chose 0.3 empirically as an optimal value of ϵ_r . The process converts depth edge images (Fig. 3a) to corresponding ridge data images (Fig. 3b).

The intuitive interpretation of the ridge data can be explained as follows. In the ridge region of DTM, the distances $D_T(X')$ to the closest edge from the circle $C(X_i)$ centered on X_i are smaller than the distance $D_T(X_i)$ between the center X_i and the closest edge. Therefore, the distance value of the center is greater than the average value of the distances on the circle. In the slope region of DTM, half of the distances to the closest edge $D_T(X')$ from the circle $C(X_i)$ centered on the X_i are larger than the distance $D_T(X_i)$ between the center X_i and the closest edge, in the other half, these distances are smaller than $D_T(X_i)$. Therefore, the distance value of the center is similar to the average value of the distances on the circle.

2.3 Initial human model acquisition

For the model-based human pose estimation, we must acquire initial human model parameters such as the lengths and angles between adjacent joints. The model uses 14 joints to represent the human body: J_H (center of head), J_{LH} (left hand), J_{LE} (left elbow), J_{LS} (left shoulder), J_{RH} (right hand), J_{RE} (right elbow), J_{RS} (right shoulder), J_T (center of torso), J_{LP} (left hip), J_{LK} (left knee), J_{LF} (left foot), J_{RP} (right hip), J_{RK} (right knee), and J_{RF} (right foot), and considers two angles: θ_{LS} between J_H and J_{LS} , and θ_{RS} between J_H and J_{RS} . In this work, we propose a semi-automatic method that acquires the initial human model parameters hierarchically. The proposed method requires that each human initially hold a Y-shaped pose [14].

To locate $\{J_{LS}, J_{RS}, J_{LP}, J_{RP}, J_T, J_H\}$, we model the region of the torso as a maximal rectangle that contains the upper body. The positions of the upper-left lower-right

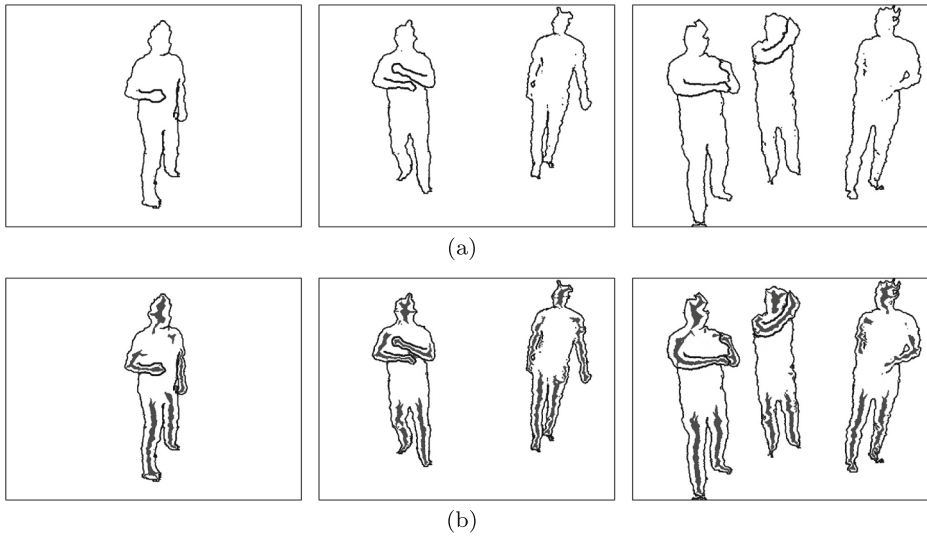


Fig. 3 Conversion of **a** depth edge images to **b** ridge data

corner of the maximal rectangle can be determined by column-wise and row-wise scanning, respectively [13], where column-wise and row-wise scanning are the maximal lengths of connected valid depth pixels in the column and row, respectively. Then we estimate the positions (J_{LS} , J_{RS}), (J_{LP} , J_{RP}) and J_T sequentially. We divide the maximal rectangle into six sub-parts to cover J_{LS} , J_{RS} , J_{LP} , and J_{RP} . Then we estimate the positions of J_{LS} , J_{RS} , J_{LP} , and J_{RP} by averaging the depth pixels within the corresponding sub-parts. We estimate the position of J_T by averaging the positions of shoulders and hips. Then we estimate the position of the head center by using a mean-shift algorithm [4] over the ridge data above the torso maximal rectangle.

Then we construct a head-shoulder-torso (HST) structure which consists of ten lengths and two angles. The lengths are $l_{H \perp S}$ perpendicular from the head center to the line that joins the left shoulder to the right shoulders, $l_{LS, RS}$ between the left shoulder and the right shoulder, $l_{H, T}$ between the head center and the torso center, $l_{LS, T}$ between the left shoulder and the torso center, $l_{RS, T}$ between the right shoulder and the torso center, $l_{(T \perp P)}$ perpendicular from the torso center to the line that joins the left hip to the right hip, $l_{LS, LP}$ between the left shoulder and the left hip, $l_{LS, RP}$ between the left shoulder and the right hip, $l_{RS, LP}$ between the right shoulder and the left hip, and $l_{RS, RP}$ between the right shoulder and the right hip. The angles are θ_{LS} and θ_{RS} .

We estimate the positions of limbs that contain either two arm parts or two leg parts. We propose to use the Hough transform [8] over the ridge data (Fig. 4a) within the left-arm bounding box and the right-arm bounding box. Among many straight lines of the Hough transform (Fig. 4b), we apply the k -means algorithm with $k = 2$ over parameter space to select two dominant straight lines that correspond to the upper and lower arm. We estimate the positions of the left and right elbows (Fig. 4c) to be at the intersection points of the two straight lines within the arm bounding boxes, then estimate the positions of the left and right hands (Fig. 4d) by averaging the ridge data near the distal ends within the arm bounding boxes. We use a similar process to identify the legs and to estimate the positions of the knees and feet.

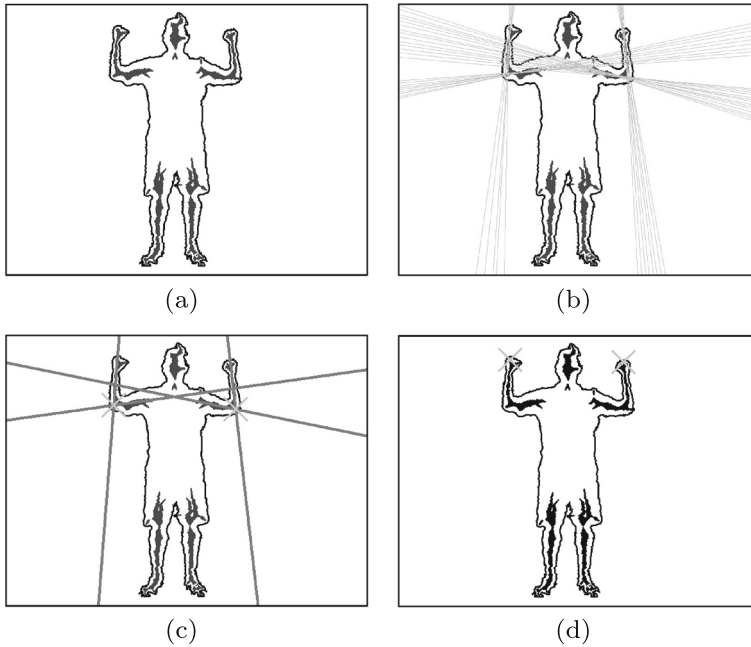


Fig. 4 Overall procedure of locating elbows and hands (procedures are described in the text)

Then we compute the lengths of limbs. l_{LUA} and l_{RUA} are the lengths between the shoulder and the elbow of the left arm and the right arm, respectively. l_{RLA} and l_{RUA} are the lengths between the elbow and the hand of the left arm and the right arm, respectively. l_{LUL} and l_{RUL} are the lengths between the hip and the knee of the left leg and the right leg, respectively. l_{RLL} and l_{RLL} are the lengths between the knee and the foot of the left leg and the right leg, respectively.

2.4 Hierarchical human pose estimation

We propose a hierarchical human pose-estimation method that determines the human joints in a top-down manner from head through torso to limbs (Algorithm 2). Estimation of the location of each joint involves four steps: joint prediction, candidate collection, invalid-data pruning, and joint estimation.

2.4.1 Step 1: joint prediction

First, we predict each joint position in the current frame from the joint position in the previous position as

$$\tilde{J}_j \leftarrow J_j^{t-1} + \Delta J_j^{t-1}, \quad (3)$$

where \tilde{J}_j is the predicted joint position and $\Delta J_j^{t-1} = J_j^{t-1} - J_j^{t-2}$ is the increment of the velocity model of the joint position. This prediction may reduce the computation time by limiting the search space for joint tracking.

2.4.2 Step 2: candidate collection

Second, we examine ridge data $R(I)$ to collect the candidates for the joint position as

$$C_j(I) = \{X_i \in R(I) \mid D_E(X_i, \tilde{J}_j) \leq r_j\}, \quad (4)$$

where $D_E(X_i, \tilde{J}_j)$ is the Euclidean distance between the pixel position X_i and the predicted joint position \tilde{J}_j and r_j is the radius of search space.

2.4.3 Step 3: invalid-data pruning

Third, we eliminate implausible candidates to leave only the valid set of joint data according to the constraints of the initial human model. We collect the valid data for head center as

$$V_H(I) = \{X_i \in C_H(I) \mid |D_E(X_i, \tilde{J}_T) - l_{H,T}| \leq \epsilon_j\}, \quad (5)$$

where \tilde{J}_T is the predicted position of torso center and $l_{H,T}$ is the length parameter in the HST structure.

For the shoulder joints J_s ($s \in \{LS, RS\}$), we collect the valid data as

$$V_s(I) = \left\{ X_i \in C_s(I) \mid \begin{cases} |D_E(J_{s'}, X_i) - l_{LS,RS}| \leq \epsilon_j \\ \left| \frac{\|\overrightarrow{J_{s'}X_i} \times \overrightarrow{J_HJ_{s'}}\|}{\|\overrightarrow{J_{s'}X_i}\|} - l_{H\perp S} \right| \leq \epsilon_j \end{cases} \right\}, \quad (6)$$

where $\overrightarrow{J_{s'}X_i}$ is a vector from the opposite shoulder $s' \in \{RS, LS\}$ to the candidate data X_i , $\overrightarrow{J_HJ_{s'}}$ is a vector from the head center to the opposite shoulder position, and $l_{H\perp S}$ and $l_{LS,RS}$ are the length parameters in the HST structure.

For the torso center, we collect the valid data as

$$V_T(I) = \left\{ X_i \in C_T(I) \mid \begin{cases} |D_E(J_H, X_i) - l_{H,T}| \leq \epsilon_j \\ |D_E(J_{LS}, X_i) - l_{LS,T}| \leq \epsilon_j \\ |D_E(J_{RS}, X_i) - l_{RS,T}| \leq \epsilon_j \end{cases} \right\}, \quad (7)$$

where $l_{H,T}$, $l_{LS,T}$, and $l_{RS,T}$ are the length parameters in the HST structure.

For the hip joints J_p ($p \in \{LP, RP\}$), we collect the valid data as

$$V_p(I) = \left\{ X_i \in C_p(I) \mid \begin{cases} |D_E(J_{LS}, X_i) - l_{LS,p}| \leq \epsilon_j \\ |D_E(J_{RS}, X_i) - l_{RS,p}| \leq \epsilon_j \\ \left| \frac{\|\overrightarrow{J_{p'}X_i} \times \overrightarrow{J_TJ_{p'}}\|}{\|\overrightarrow{J_{p'}X_i}\|} - l_{T\perp P} \right| \leq \epsilon_j \end{cases} \right\}, \quad (8)$$

where $\overrightarrow{J_{p'}X_i}$ is a vector from the opposite pelvis joint $p' \in \{RP, LP\}$ to the candidate data X_i , $\overrightarrow{J_TJ_{p'}}$ is a vector from the torso center to the opposite pelvis position, and $l_{H,T}$, $l_{LS,T}$, $l_{RS,T}$ are the length parameters in the HST structure.

For the limb joints J_k ($k \in \{LE, RE, LH, RH, LK, RK, LF, RF\}$), we collect the valid data as

$$V_k(I) = \left\{ X_i \in C_k(I) \mid \begin{cases} |D_E(X_i, J_p) - l_k| \leq \epsilon_j \\ P_L(X_i, J_p) > \epsilon_L \end{cases} \right\}, \quad (9)$$

where l_k are the length constraints between the limb joint J_k and its parent joint J_p ($p \in \{LS, RS, LE, RE, LP, RP, LK, RK\}$), $P_L(X_i, J_p)$ is the degree of straightness that is defined by the ratio of the Euclidean distance over the geodesic distance, and ϵ_L is a

straightness threshold that controls the tolerance of straight lines. In this work, we chose 0.7 empirically, as an optimal value of ϵ_L . The degree of straightness is computed as

$$P_L(X_i, J_i^t) = \frac{D_E(X_i, J_i^t)}{D_G(X_i, J_i^t)}, \quad (10)$$

where J_i^t is the position of the parent limb joint of the limb joint J_i^t at the t th frame and $D_G(X_i, J_i^t)$ is the geodesic distance between X_i and J_i^t .

We consider the degree of straightness to avoid data ambiguity when arms or legs are bent. For example, we eliminate invalid candidate data around the elbow. When we bend the arm, the lower arm's candidate data can be identified as part of the valid elbow's ridge data because two candidate data are close together due to the arm bending. The degree of straightness separates effectively the lower arm from the elbow because the degree of straightness from the shoulder to the elbow is ≈ 1 (Fig. 5a) but that from the shoulder to the lower arm is < 1 (Fig. 5b).

2.4.4 Step 4: joint estimation

Finally, we aggregate the collected valid data into the estimated position for the joint as

$$J_j = \frac{1}{|V_j(I)|} \sum_{X_i \in V_j(I)} X_i, \quad (11)$$

where $|V_j(I)|$ is the total number of valid ridge data.

If insufficient valid ridge data remain after eliminating invalid data, we use the raw depth data to estimate the joint position. We restart step 2, but use the raw depth data to collect the candidates for the joint position as

$$C'_j(I) = \{X_i \in I \mid D_E(X_i, \tilde{J}_j) \leq r_j\}, \quad (12)$$

where C'_j is the candidate data for joint J_j . Then we eliminate implausible candidates from C'_j according to the same constraints of each body part, such as head, torso, and limb.

The parameter r_j of (4) governs the search space for collecting the candidate data. A large r_j collects abundant candidates, but it increases the time to prune out the invalid data.

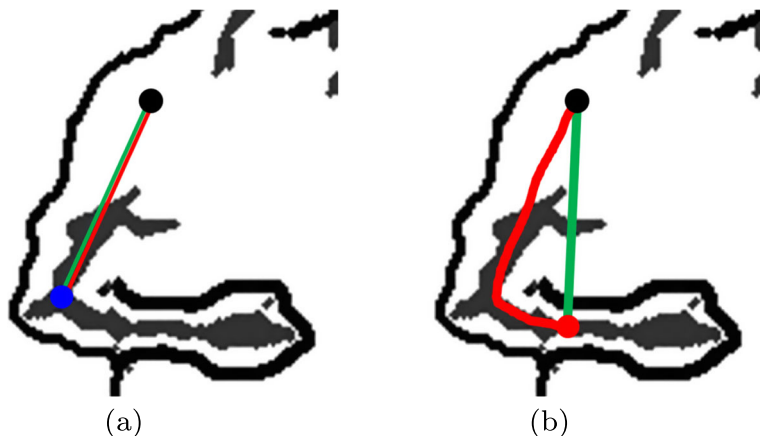


Fig. 5 Examples of the degree of straightness; **a** strong straightness and **b** weak straightness, green line: Euclidean distance; and red line: geodesic distance

The parameter ϵ_j (5–9) controls the tolerance of difference between the estimated length and the length constraint of the human model. We empirically chose these parameters and Table 1 shows the mean Average Precision (mAP) and frame per second (fps) with different search space r_j and threshold ϵ_j .

Algorithm 2 Hierarchical Human Pose Estimation

Input: $R(I)$: ridge data, I : raw depth image

Output: Estimated joint position J_j

```

/* Joint Prediction */
 $\tilde{J}_j = J_j^{t-1} + \Delta J_j^{t-1}$ 

/* Candidate Collection */
 $C_j = \{X_i \in R(I) \mid D_E(X_i, \tilde{J}_j) \leq r_j\}$ 

/* Invalid data Pruning */
switch j do
  case H /* head */
     $V_j = \{X_i \in C_j \mid |D_E(X_i, \tilde{J}_T) - l_{H,T}| \leq \epsilon_j\}$ 

  case LS or RS /* shoulders */
     $V_j = \left\{ X_i \in C_j \mid \begin{cases} |D_E(J_{j'}, X_i) - l_{LS,RS}| \leq \epsilon_j \\ \left| \frac{\|\vec{J_{j'}X_i} \times \vec{J_HJ_{j'}}\|}{\|\vec{J_{j'}X_i}\|} - l_{H\perp S} \right| \leq \epsilon_j \end{cases} \right\}$ 

  case T /* torso center */
     $V_j = \left\{ X_i \in C_j \mid \begin{cases} |D_E(J_H, X_i) - l_{H,T}| \leq \epsilon_j \\ |D_E(J_{LS}, X_i) - l_{LS,T}| \leq \epsilon_j \\ |D_E(J_{RS}, X_i) - l_{RS,T}| \leq \epsilon_j \end{cases} \right\}$ 

  case LP or RP /* hips */
     $V_j = \left\{ X_i \in C_j \mid \begin{cases} |D_E(J_{LS}, X_i) - l_{LS,j}| \leq \epsilon_j \\ |D_E(J_{RS}, X_i) - l_{RS,j}| \leq \epsilon_j \\ \left| \frac{\|\vec{J_{j'}X_i} \times \vec{J_TJ_{j'}}\|}{\|\vec{J_{j'}X_i}\|} - l_{T\perp P} \right| \leq \epsilon_j \end{cases} \right\}$ 

  otherwise /* limb */
     $V_j = \left\{ X_i \in C_j \mid \begin{cases} |D_E(X_i, J_p) - l_j| \leq \epsilon_j \\ P_L(X_i, J_p) > \epsilon_L \end{cases} \right\}$ 
endsw
endsw

/* Joint Estimation */
if  $|V_j(I)|$  is sufficient then
   $J_j = \frac{1}{|V_j(I)|} \sum_{X_i \in V_j(I)} X_i$ 
else
   $C_j = \{X_i \in I \mid D_E(X_i, \tilde{J}_j) \leq r_j\}$ 
  repeat from line 3
end
```

Table 1 Influences of r_j and ϵ_j on mean average precision (mAP) and frames per second (fps)

ϵ_j	r_j									
	80		100		110		120		140	
	mAP	fps	mAP	fps	mAP	fps	mAP	fps	mAP	fps
40	0.912	223.1	0.935	219.6	0.935	202.1	0.931	183.6	0.924	160.3
45	0.910	227.2	0.934	220.7	0.934	198.3	0.930	180.7	0.925	154.5
50	0.906	220.8	0.931	221.5	0.936	199.6	0.935	182.3	0.907	155.3
55	0.904	221.6	0.930	218.9	0.935	200.0	0.935	182.0	0.907	158.4
60	0.900	220.5	0.927	218.7	0.935	199.5	0.935	181.8	0.907	155.4

3 Dance performance evaluation

To evaluate dance performance, we developed a dance teacher program (Fig. 6) that can help people to learn dances from examples in five steps: (1) It shows the dance performed by a virtual dance teacher, and the learner tries to follow the dance as closely as possible, (2) The human pose estimator extracts the joint positions of the learner, (3) The dance feature generator makes the dance features for dance learner from the joint positions in the extracted learner’s joint positions, then (4) performs dynamic time warping of the learner’s dance features to the teacher’s dance features which are pre-generated from the teacher’s K-Pop dance database, and (5) evaluates the learner’s dance performance by comparing the learner’s dance features to the teacher’s dance features.

The learner can select one K-Pop dance among 100 popular K-Pop dances, which are listed in a menu window (Fig. 7, left). To show the dance performed by a dance expert, we use standard computer graphics techniques to render the realistic virtual teacher (Fig. 7, upper right). While the learner tries to mimic the dance teacher, the program represents an instant dance evaluation score and dance similarity of each body part (Fig. 7, lower right).

3.1 Dance feature extraction

Human poses are commonly represented by 15 joint positions with the Cartesian 3D coordinate system. Because of the variation in camera position and orientation, or in human body

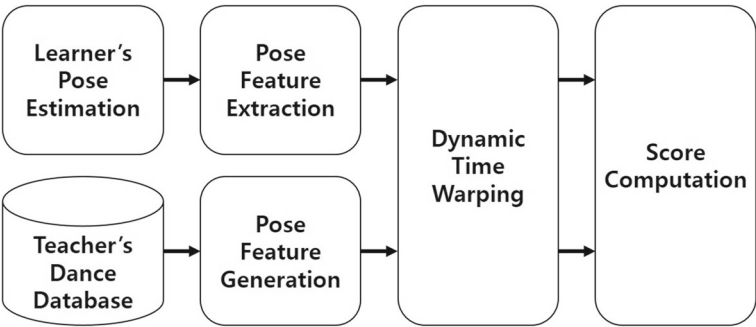


Fig. 6 Overall process of the dance teacher

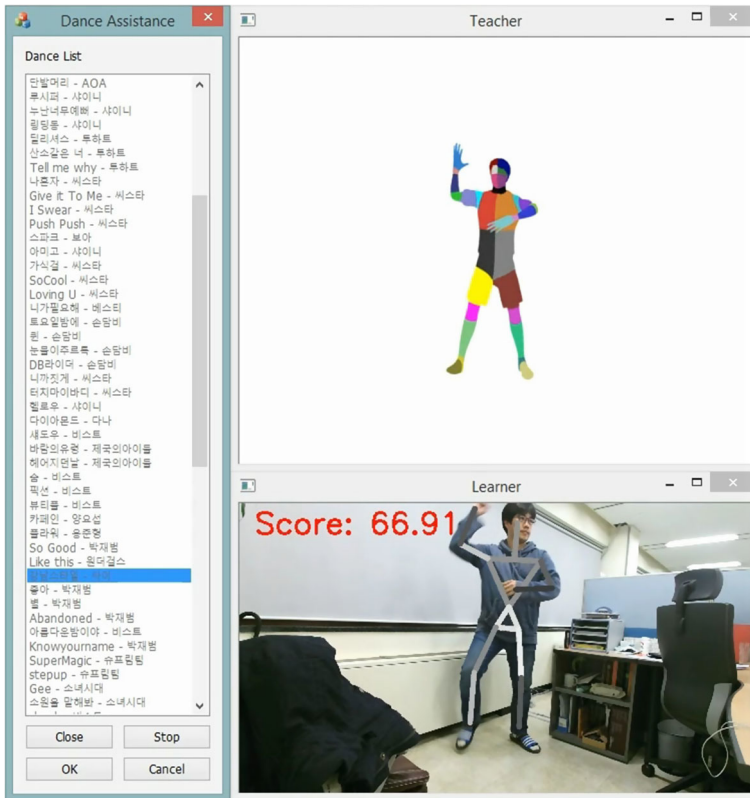


Fig. 7 Screen shot of the dance teacher program. Left window: list of 100 dance sequences; upper-right window: dance teacher; lower-right window: learner with an instant dance evaluation score and dance similarity of each body part represented by an intensity value (bright intensity = high similarity)

shape and size, the traditional representation of human pose is not suitable to compare dance performances of teacher and learner. We expand a previous method [17] to design a new 22-dimensional feature: it is composed of a six-dimensional torso feature, an eight-dimensional first-degree feature, and an eight-dimensional second-degree feature.

3.1.1 Torso feature

In [17], the human torso is represented as one component, and the angles of yaw, pitch, and roll are encoded with respect to the world coordinate. However, K-Pop dance includes very complex torso poses, so we design a six-dimensional torso feature by dividing the torso joints into an upper-torso group and a lower-torso group. The upper-torso group forms a plane that contains the torso center, the left shoulder, and the right shoulder; the lower-torso group forms another plane that contains the torso center, the left hip, and the right hip. Each plane has its own three-dimensional base axis. We obtain three-dimensional joint angles by computing the dot product between the upper and lower base axes at the current time, and obtain other three-dimensional joint angles by computing the difference between the

three-dimensional joint angles of the previous and current times. The proposed torso feature can represent torso orientation, and the bend, twist, and lean of torso and full-body rotation.

3.1.2 First- and second-degree feature

The first-degree feature has eight-dimensional joint angles that represent the movement of two elbows and two knees, where each joint has an inclination and an azimuth with respect to the adjacent parent joint, such as shoulder and hip [17]. Similarly, the second-degree feature has eight-dimensional joint angles that represent the movement of two hands and two feet, where each joint has an inclination and an azimuth with respect to the adjacent parent joint, such as elbow and knee [17].

3.2 Dance similarity

The timing accuracy of each subsequence is measured by the timing similarity of the dance sequences of the learner and teacher as follows (Algorithm 3). The dance teacher program (1) obtains the learner's dance sequence by accumulating it for a given period (2 s in this work), (2) finds the corresponding teacher's dance sequence by dynamic time warping, and (3) computes the timing similarity of the dance sequences between learner and teacher as

$$S_t = 1 - \min \left(1, \exp \left(\frac{|T_t - T_l| - \tau}{\alpha \tau} \right) \right), \quad (13)$$

where $T_t = \frac{T_t^s + T_t^e}{2}$ is the middle time of the teacher's dance sequence, and $T_l = \frac{T_l^s + T_l^e}{2}$ is the middle time of the learner's dance sequence, where the superscripts s and e denote the start and end point of dance sequence, respectively. α is a parameter that governs the slope of (13) and τ is a tolerance parameter that controls maximum deviation of timing.

The pose accuracy is measured by the posture similarity of the dance sequences between learner and teacher as

$$S_p = \frac{1}{T_t^e - T_t^s + 1} \sum_{i=T_t^s}^{T_t^e} \exp \left(-\frac{\|\mathbf{f}_l^i - \mathbf{f}_t^i\|}{\beta} \right), \quad (14)$$

where vectors \mathbf{f}_l^i and \mathbf{f}_t^i denote the dance features of the learner and teacher respectively at the i th frame, and β is a parameter that controls the amount of deviation from the teacher's dance.

The dance similarity of the dance sequences between learner and teacher is defined by the sum of partial scores of timing and pose accuracies as

$$S = \frac{1}{N} \sum_{j=1}^N S_t^j S_p^j, \quad (15)$$

where N is the number of subsequences of the learner's dance sequence and S_t^j and S_p^j denote the timing accuracy and the posture accuracy respectively at the j th subsequence.

Algorithm 3 Dance Performance Evaluation

Input: Learner's dance feature $\mathbf{f}_l^i, i = [1, n]$, teacher's dance feature $\mathbf{f}_t^j, j = [1, m]$
Parameters: Timing tolerant value τ , slope control parameters α, β
Output: Timing accuracy S_t , pose accuracy S_p

```

/* Dynamic time warping                                     */
foreach  $i = 1$  to  $n$  do  $DTW[i, 0] = \inf$  foreach  $j = 1$  to  $m$  do  $DTW[0, j] = \inf$ 
 $DTW[0, 0] = 0$ ;
for  $i = 1$  to  $n$  do
    for  $j = 1$  to  $m$  do
         $DTW[i, j] =$ 
             $\|\mathbf{f}_l^i - \mathbf{f}_t^j\| + \min(DTW[i - 1, j], DTW[i, j - 1], DTW[i - 1, j - 1]);$ 
    end
end

/* Timing accuracy                                           */
 $T_t^s = \arg \min_j DTW[1, j]$ ;
 $T_t^e = \arg \min_j DTW[n + 1, j]$ ;
 $T_t = (T_t^e - T_t^s + 1)/2$ ;
 $S_t = 1 - \min\left(1, \exp\left(\frac{(|T_t - n/2| - \tau)}{\alpha\tau}\right)\right)$ ;

/* Pose accuracy                                             */
 $S_p = \exp\left(-\frac{\min_j DTW[n+1, j]}{\beta T_t}\right)$ ;

```

4 Experimental results and discussions

4.1 Pose estimation accuracy

We validate the pose-estimation accuracy of the proposed human pose estimation method by comparing to the benchmark dataset EVAL [7]. The EVAL dataset was recorded using a Microsoft Kinect depth camera at a speed of 30 fps and a resolution of 320×240 pixels, and consists of 24 real-world depth image sequences, each of which had a variety of sizes and complexities from 288 frames to 488 frames. The true position of each joint was labeled using a commercial marker-based motion-capture system.

We compared the pose estimation accuracy of the proposed method with those of the state-of-the-art methods [12, 24]. We define mAP as the ratio of the number of true positives over the total number of joints, where a joint position is considered as a true positive when the position is located within the distance threshold $\delta = 0.1$ m of the true joint position.

The proposed human pose estimation method achieved the best pose estimation accuracy 0.9358 mAP on the EVAL dataset (Fig. 8), which contains complicated poses such as crossed punch, hand standing, and sitting on the floor. The proposed human pose estimation method successfully tracks human body joints because we use the ridge data instead of raw depth data to prune the invalid data. This improvement is achieved because (1) ridge data remain within the body parts, but the raw depth data drift over the overlapped body parts and (2) when raw depth data are used, data that meet the requirement for data pruning may come from the wrong points.

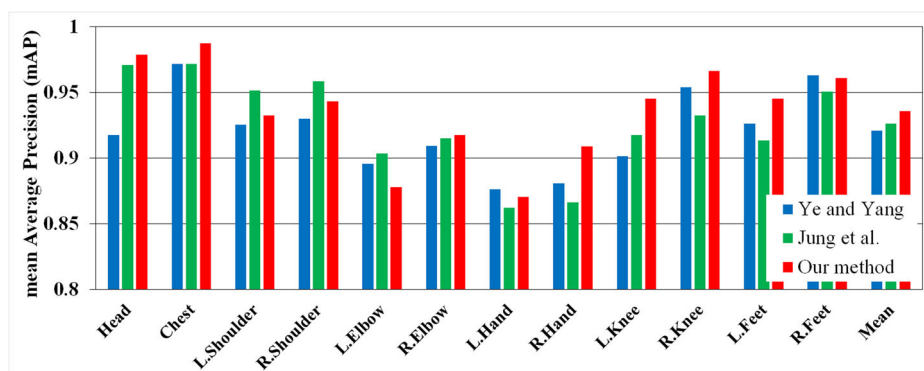


Fig. 8 Comparison of mean average precision (mAP) using the EVAL dataset, with two state-of-the-art methods (Ye and Yang [24] and Jung et al. [12]) and ours

To show the effectiveness of the ridge data, we conducted additional experiments using the CNN which is proposed by Huang and Altamar [9]. As the input of CNN, we took the depth image for the first additional experiment and took the depth + ridge image for the second additional experiment, where the ridge image represents a probability map of ridge data. We used 800,000 synthetic images for training, 300,000 synthetic images for validation, and EVAL dataset for testing. We achieved 0.9566 and 0.9667 mAP of pose estimation accuracy for the first and second experiment, respectively. The experimental result shows that the ridge image makes a large improvement in pose estimation accuracy over the limb parts because the limb parts contain the rich and strong ridge data (Fig. 9).

4.2 Pose estimation error

We validate the proposed human pose estimation method in terms of pose estimation error using the benchmark dataset SMMC-10 [6]. The SMMC-10 dataset was recorded using a

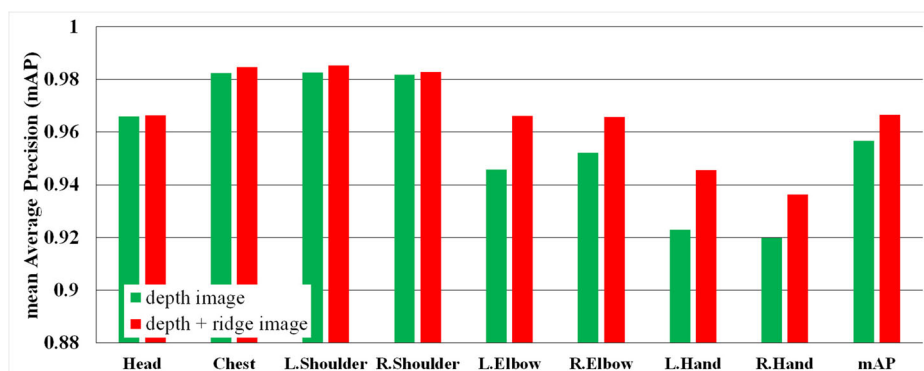


Fig. 9 Comparison of pose estimation accuracy between depth and depth + ridge image using the EVAL dataset

Mesa SwissRanger time-of-flight camera at a speed of 25 fps and a resolution of 176×144 pixels, and consists of 28 real-world sequences of depth images, each of which had a variety of sizes and complexities from 100 frames to 400 frames. The true position of each joint was labeled using a commercial marker-based motion-capture system.

We compared the pose estimation error of the proposed method with those of the state-of-the-art methods (Ganapathi et al. [6], and Baak et al. [2]). We define average pose error as

$$\epsilon_{avg} = \frac{1}{\sum_i^N K_i} \sum_{i=1}^N \sum_{k=1}^{K_i} \|J_k^i - \tilde{J}_k^i\|, \quad (16)$$

where N is the number of total frames, K is the number of visible motion capture markers, J_k^i is the true 3D position of the k th joint in i th frame, and \tilde{J}_k^i is the corresponding 3D position of the estimated k th joint in the i th frame.

To show the effect of the proposed ridge data feature, we conducted an additional experiment that is almost the same as the proposed method except at the candidate collection phase of each estimation step. The original method tries to collect candidates from ridge data first, but the experiment without ridge data collected the candidates from raw depth data directly.

The proposed method achieved the lowest pose errors (3.88 cm) in most of the benchmark sequences (Fig. 10). Especially, the proposed method achieved the lowest pose errors in sequences 24 to 27. These sequences contain complex motions such as fast kicks and swings, self-occlusions, and full-body rotations. This result shows that the HST structure and ridge data capture the fast movements, occlusions, and full-body rotations, effectively. Our method using the raw depth data shows large standard deviation, which means that the estimated positions are not localized well inside the body. But, our method using the ridge data reduces the standard deviation significantly because the ridge data prevent the drift of joint positions.

The proposed method successfully estimated the complicated poses such as occlusion, sitting on floor, and handstand in the EVAL dataset (Fig. 11, left column) but in some cases missed the head, or double-counted joints (Fig. 11, right column).

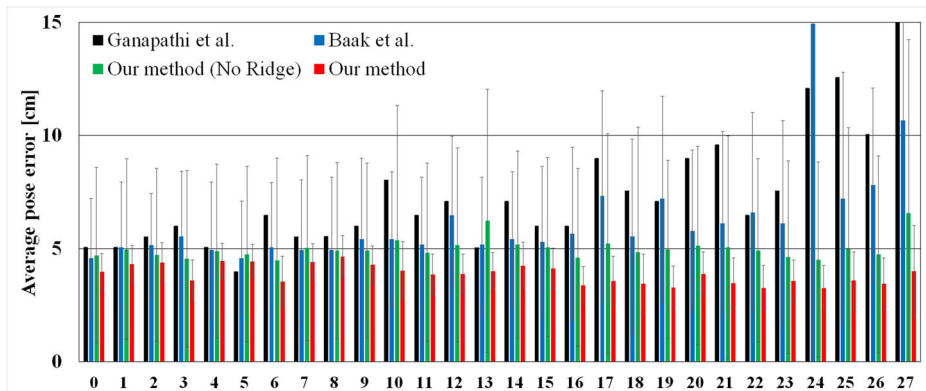


Fig. 10 Comparison of average pose error and standard deviation (bars) using the SMMC-10 dataset, with two state-of-the-art methods (Ganapathi et al. [6] and Baak et al. [2]) and ours.

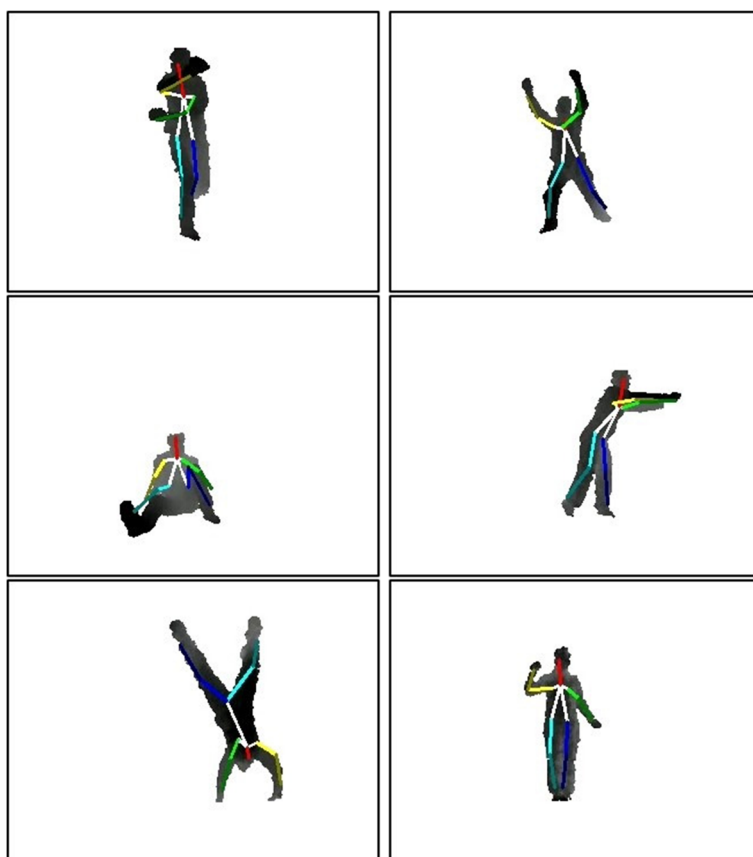


Fig. 11 Human pose estimation results on EVAL dataset. Successful cases (left column) and Failed cases (right column)

4.3 Dance evaluation accuracy

To validate the dance performance evaluation, we constructed a large K-Pop dance dataset¹ that consists of 100 popular K-Pop dances. For each K-Pop dance, we used a commercial motion capture system to construct a teacher's K-Pop dance database, and used a Microsoft Kinect 2 camera to construct a learners' K-Pop dance database that was recorded by four learners who had various dance skill levels. Learner's dance sequences were labeled subjectively by a group of dance experts as *best*, *good*, *bad*, and *worst*.

We validate the dance evaluation accuracy of the proposed dance teacher program by observing whether the proposed method correctly evaluates the four learners' dance performances. Evaluation scores of *best*, *good*, *bad*, and *worst* learners are denoted by S_1 , S_2 , S_3 , and S_4 , respectively. We considered that the evaluation scores are correct if and only if $S_1 > S_2 > S_3 > S_4$. Evaluation scores for 100 sets of dance sequences were assigned

¹The K-Pop database is available from <https://goo.gl/NoVDm4>.

Table 2 Four learners' examination scores that is measured by the dance teacher program

Seq	Scores based on [17]				Scores using dance feature based on traditional HPE				Scores using dance feature based on CNN-HPE			
	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4
1	63.4	58.3	55.5	52.4	67.0	60.8	56.2	52.2	62.4	56.2	51.9	47.6
2	58.9	60.1	53.3	51.5	62.7	62.5	54.2	51.4	58.2	57.8	49.5	46.8
3	62.8	57.7	55.5	51.4	66.4	60.2	56.4	51.3	61.9	55.9	51.8	46.6
4	63.6	60.3	57.8	54.0	66.6	62.4	58.6	53.8	62.1	58.0	53.6	49.2
5	63.1	59.3	55.6	53.0	66.6	61.2	56.9	52.9	62.0	56.6	52.4	48.4
6	65.2	59.8	57.7	53.9	68.9	61.9	58.5	53.7	64.1	57.9	53.7	49.3
7	62.4	60.3	54.4	53.8	66.0	62.7	55.2	53.4	58.0	52.7	48.1	45.1
8	58.7	54.9	51.8	49.6	61.9	56.7	52.2	49.3	60.1	56.0	52.4	46.7
9	60.4	60.9	57.0	53.1	63.4	62.7	57.8	52.8	61.3	58.2	53.4	45.7
10	61.1	58.0	56.2	51.0	64.5	60.3	56.8	51.0	60.7	54.7	52.9	46.5
11	61.7	56.8	56.4	50.7	65.0	59.0	57.4	50.9	61.7	56.4	54.0	46.4
12	62.4	60.4	57.2	50.4	65.7	62.5	58.1	50.1	62.8	58.7	54.5	47.2
13	62.5	58.5	57.7	50.8	66.1	60.7	58.6	50.8	60.2	58.3	53.0	47.7
14	64.2	61.5	58.5	51.8	67.1	63.4	58.9	51.4	63.0	59.8	52.9	49.2
15	59.2	53.1	48.4	50.4	62.6	55.2	49.3	50.3	40.5	37.9	35.9	48.0
16	63.6	58.7	55.7	51.3	67.2	61.0	56.6	51.0	62.4	56.3	51.9	46.5
17	61.9	57.2	55.7	51.2	65.6	59.5	56.7	51.2	62.1	57.4	53.8	46.5
18	64.2	58.5	55.4	51.5	67.7	61.2	56.5	51.8	60.3	55.2	50.4	46.2
19	59.1	54.7	56.2	49.8	62.8	57.5	57.4	50.3	58.4	52.9	52.6	45.7
20	60.4	58.2	57.4	52.7	63.8	60.5	58.3	52.5	62.2	53.1	53.1	47.8
21	63.1	59.5	57.1	50.8	66.9	62.0	58.7	51.1	63.2	56.7	52.0	47.0
22	60.9	57.4	54.0	51.0	64.4	59.6	54.7	50.7	62.3	55.9	51.9	47.9
23	63.4	55.4	56.6	52.5	66.5	57.4	57.3	52.3	58.7	51.5	45.6	46.2
24	61.0	59.5	55.5	53.3	64.2	61.5	55.7	52.4	61.2	55.0	52.1	46.6
25	61.5	58.1	54.3	52.5	64.7	60.1	55.3	51.9	60.2	56.8	50.1	46.1
26	63.5	61.2	54.8	53.5	67.0	63.3	55.9	52.9	63.0	56.9	50.5	45.6
27	57.2	56.5	51.7	47.8	60.4	58.5	52.8	47.5	60.1	58.6	50.3	46.8
28	62.6	58.7	55.7	53.5	65.9	60.8	56.6	52.9	61.5	57.6	51.0	48.1
29	62.9	60.6	54.9	52.7	65.9	62.6	55.8	52.5	57.9	53.5	50.8	47.4
30	61.0	55.2	55.7	52.6	64.6	56.0	56.0	50.6	59.9	54.1	50.1	46.4
31	63.1	59.7	56.2	53.1	66.6	62.1	57.0	52.6	60.8	56.7	52.8	46.3
32	63.3	58.9	55.1	51.0	66.5	60.9	56.0	50.8	61.9	57.4	52.5	47.8
33	63.0	58.5	55.5	51.8	66.4	60.7	56.6	51.9	62.2	56.8	50.8	47.1
34	61.4	55.0	55.7	51.0	64.6	56.5	56.4	50.3	57.7	56.0	52.6	46.7
35	58.6	55.7	53.8	52.1	61.8	57.6	54.9	51.9	60.3	57.1	50.2	48.4
36	61.1	56.2	53.6	51.0	64.1	58.4	54.3	50.8	54.4	49.3	44.9	42.6
37	62.0	59.4	56.6	51.4	65.1	60.9	57.3	50.6	61.6	57.2	51.0	46.6
38	62.9	59.7	55.9	53.1	66.3	62.1	57.1	52.4	49.2	46.9	40.7	38.4
39	62.8	59.1	54.1	52.0	66.5	61.1	55.2	51.4	61.9	56.7	51.7	45.7
40	59.9	58.2	56.1	52.3	62.0	60.2	57.0	51.3	63.0	57.4	53.1	47.5

Table 2 (continued)

Seq	Scores based on [17]				Scores using dance feature based on traditional HPE				Scores using dance feature based on CNN-HPE			
	S ₁	S ₂	S ₃	S ₄	S ₁	S ₂	S ₃	S ₄	S ₁	S ₂	S ₃	S ₄
41	61.8	59.7	53.6	53.6	64.8	61.6	54.6	53.0	59.9	56.1	53.8	47.9
42	55.4	51.2	47.8	47.0	58.4	53.1	48.8	46.7	62.1	56.3	51.6	47.0
43	62.5	59.4	54.2	51.6	65.8	61.5	55.2	51.2	60.1	57.1	51.5	48.1
44	50.1	49.2	44.0	42.6	52.7	50.2	44.2	42.1	60.3	55.9	51.1	47.5
45	63.4	59.2	55.8	51.0	66.1	60.7	55.9	50.1	62.7	59.2	51.8	48.4
46	64.3	60.0	57.3	53.1	67.2	61.6	57.6	52.0	56.4	54.3	48.6	43.3
47	63.2	52.6	53.3	51.3	66.8	54.8	54.3	51.3	61.7	56.6	52.4	48.4
48	63.3	58.1	55.6	52.5	66.6	60.4	56.3	52.3	61.7	58.2	51.6	47.9
49	61.4	59.3	54.5	51.3	64.4	61.1	54.5	50.5	63.5	55.6	50.5	46.0
50	64.4	59.5	54.6	50.7	67.5	61.2	54.9	49.9	59.7	56.4	52.1	48.1
51	61.9	61.2	54.3	52.1	64.5	62.8	54.5	51.0	64.0	55.2	52.3	48.5
52	62.8	60.2	54.8	53.4	65.9	62.0	55.1	52.6	61.7	56.6	53.7	47.3
53	60.9	58.5	57.7	53.0	63.9	60.6	58.0	52.2	63.0	57.6	52.5	47.7
54	62.8	57.9	55.0	51.6	66.8	60.9	56.0	51.5	62.4	57.6	51.9	46.9
55	64.9	57.5	53.7	50.2	68.6	60.4	55.4	50.7	63.3	52.8	52.3	49.6
56	60.6	58.7	55.3	52.5	64.0	60.9	56.6	52.7	60.4	56.1	50.9	45.9
57	65.4	57.3	55.5	53.2	68.7	59.4	56.7	53.0	62.5	57.1	53.7	47.2
58	62.9	58.9	57.3	51.9	66.1	60.9	58.3	51.8	59.7	56.2	53.7	47.9
59	64.0	60.0	56.0	52.6	67.5	62.0	56.9	52.5	59.4	54.8	52.3	46.6
60	61.3	60.3	57.0	52.2	64.4	62.6	57.6	52.2	57.7	55.8	51.7	44.3
61	58.6	57.9	55.0	48.7	61.6	59.9	55.8	48.3	59.2	55.5	52.7	45.0
62	64.1	61.9	56.3	53.9	67.4	64.5	57.4	53.8	59.7	57.4	51.9	47.7
63	61.9	58.1	55.6	50.5	65.0	59.2	55.7	50.1	60.3	52.7	52.3	46.2
64	63.3	59.4	56.5	51.7	66.9	61.7	57.7	51.7	57.1	54.8	50.7	47.1
65	63.4	59.9	55.0	51.5	66.9	62.3	56.4	51.6	61.7	58.3	51.0	48.6
66	60.3	59.5	54.0	53.0	63.0	61.4	54.4	52.3	53.3	52.2	51.2	45.1
67	61.1	58.2	54.1	50.2	64.6	60.3	55.0	50.2	62.0	57.0	52.5	48.8
68	63.6	59.3	57.4	51.9	67.0	61.5	58.3	51.6	60.5	58.1	52.0	47.5
69	60.1	57.0	55.6	50.9	63.7	59.2	57.0	51.2	60.2	52.6	51.9	46.5
70	60.0	57.0	55.7	49.4	63.1	59.4	56.7	49.2	62.2	57.7	52.6	48.0
71	60.1	59.7	55.2	52.2	63.6	61.9	56.1	52.2	62.1	56.6	51.7	46.5
72	57.9	56.6	54.0	51.4	61.3	59.0	55.4	51.6	61.9	56.3	52.2	47.4
73	63.2	57.5	53.2	49.7	66.6	59.3	53.5	49.7	62.2	55.2	49.5	45.3
74	64.4	54.6	55.8	54.3	68.1	56.8	56.9	54.1	60.8	54.9	51.5	45.8
75	61.2	56.5	55.4	48.7	64.4	58.6	55.9	49.0	62.5	57.3	53.1	47.0
76	61.7	60.6	55.7	52.6	64.9	62.6	56.4	52.1	60.8	56.9	48.0	45.4
77	61.8	59.0	51.4	49.8	65.1	61.1	52.4	49.9	62.5	56.7	49.9	48.0
78	62.3	57.7	54.3	52.2	65.6	59.6	55.1	52.1	61.5	55.3	50.7	47.5
79	62.2	59.3	55.2	50.0	65.4	61.6	56.0	49.7	61.0	57.2	51.6	45.6
80	63.0	59.7	56.1	53.9	66.5	61.7	56.9	53.5	61.3	58.6	51.4	46.6

Table 2 (continued)

Seq	Scores based on [17]				Scores using dance feature based on traditional HPE				Scores using dance feature based on CNN-HPE			
	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4
81	64.3	59.0	53.4	52.7	66.9	61.0	54.2	52.3	62.0	50.7	49.7	46.7
82	62.2	61.0	54.7	51.2	65.6	63.1	55.6	51.1	60.0	54.4	51.5	44.6
83	60.9	55.6	52.2	47.7	64.5	58.4	53.6	48.4	62.8	56.0	52.2	45.8
84	54.4	54.3	54.5	49.8	57.0	56.1	55.3	49.4	60.1	54.0	49.0	44.2
85	63.8	58.4	55.7	50.2	67.2	60.2	56.7	50.3	63.2	55.0	50.7	44.8
86	62.2	60.0	56.9	52.7	64.6	61.5	57.4	52.2	59.3	58.4	53.5	48.3
87	63.0	59.7	53.5	53.9	66.7	61.8	54.9	53.7	59.5	57.2	52.3	49.4
88	62.6	58.6	54.5	53.3	66.2	61.3	55.9	53.0	62.3	57.5	50.5	49.2
89	41.2	39.9	38.5	53.3	43.4	41.0	39.0	52.9	62.0	57.2	51.1	48.3
90	64.1	57.0	54.0	49.2	67.5	59.0	54.8	49.1	60.3	57.0	52.9	47.7
91	60.4	59.7	55.8	54.1	63.8	61.4	56.8	53.9	56.1	52.8	46.9	43.6
92	57.4	55.0	50.2	48.3	60.2	56.9	50.9	48.1	60.1	58.3	53.0	47.8
93	60.5	60.8	56.7	52.5	64.3	63.2	57.7	52.4	61.8	57.2	50.8	48.4
94	59.4	56.0	54.6	49.5	63.2	56.8	54.6	49.6	58.8	52.7	50.2	45.2
95	63.8	58.9	55.2	52.8	67.1	61.2	56.2	52.7	62.6	56.8	51.5	48.0
96	59.8	58.9	53.4	52.5	62.6	60.8	54.1	52.1	58.6	56.4	49.7	47.7
97	62.8	59.7	54.3	53.4	66.0	61.4	55.0	52.9	59.2	57.2	50.2	48.0
98	62.2	59.6	53.4	50.8	65.8	61.9	54.6	51.0	61.5	57.4	49.9	46.5
99	60.8	60.7	53.6	51.6	64.5	63.2	54.8	51.7	60.0	58.7	50.4	47.2
100	62.3	58.9	54.8	51.4	65.4	60.7	55.5	51.1	61.3	56.4	51.3	46.9

Bold scores within row and section disagree with expert evaluation

by the program. Evaluation scores were calculated by a conventional feature in [17], by the proposed dance feature using the traditional pose estimation method, and by the proposed dance feature using the CNN-based pose estimation method.

Dance evaluation scores from the proposed dance feature using the CNN-based pose estimation method agreed with 98 times out of 100 with the evaluations of dance experts (Table 2); the exceptions were the 15th and 23rd sequences, whereas [17] agreed 86 times and the proposed dance feature using the traditional pose estimation method agreed 97 times.

5 Conclusion

We proposed a unified framework that evaluates dance performance by markerless estimation of human poses. This framework uses ridge data and data pruning to estimate human poses, and uses a dance teacher program to evaluates learner's dance timing and accuracy.

The main contributions of this paper are (1) The proposed human pose estimation method achieved higher pose-estimation accuracy (0.9358 mAP) and lower average pose error (3.88 cm) than the current state-of-art tracking methods. (2) The method tracks each body joint efficiently by using ridge data and data pruning. (3) We constructed a large K-Pop dance

database, which contains 100 dance experts' sequences and 400 dance learners' sequences for the dance teacher program. (4) The proposed dance teacher program achieved 98% agreement with experts' dance evaluation.

Acknowledgments This research was partially supported by the MSIT (Ministry of Science, ICT), Korea, under the SW Starlab support program (IITP-2017-0-00897) supervised by the IITP (Institute for Information & communications Technology Promotion).

This work was partially supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (IITP-2014-0-00059, Development of Predictive Visual Intelligence Technology).

References

1. Alexander T, Szegedy C (2014) Deeppose: Human pose estimation via deep neural networks. In: The IEEE conference on computer vision and pattern recognition, pp 1653–1660
2. Baak A, Müller M, Bharaj G, Seidel H-P, Theobalt C (2011) A data-driven approach for real-time full body pose reconstruction from a depth camera. In: IEEE international conference on computer vision, pp 1092–1099
3. Cao Z, Simon T, Wei S, Sheikh Y (2017) Realtime multi-person 2D pose estimation using part affinity fields. In: The IEEE conference on computer vision and pattern recognition
4. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. *IEEE Trans Pattern Anal Mach Intell* 25(5):564–577
5. Dillencourt M, Samet H, Tamminen M (1992) A general approach to connected-component labeling for arbitrary image representations. *J ACM* 39(2):253–280
6. Ganapathi V, Plagemann C, Koller D, Thrun S (2010) Real time motion capture using a single time-of-flight camera. In: IEEE conference on computer vision and pattern recognition, pp 755–762
7. Ganapathi V, Plagemann C, Koller D, Thrun S (2012) Real-time human pose tracking from range data. In: European conference on computer vision, pp 738–751
8. Hough P (1959) Machine analysis of bubble chamber pictures. In: International conference on high energy accelerators and instrumentation, pp 73
9. Huang J, Altamar D (2016) Pose estimation on depth images with convolutional neural network
10. Jalal A, Kamal S, Kim D (2014) A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments. *Sensors* 14(7):11735–11759
11. Jalal A, Kamal S, Kim D (2015) Depth Silhouettes Context: A new robust feature for human tracking and activity recognition based on embedded HMMs. In: 12th international conference on ubiquitous robots and ambient intelligence, pp 294–299
12. Jung H, Lee S, Heo Y, Yun I (2015) Random tree walk toward instantaneous 3D human pose estimation. In: The IEEE conference on computer vision and pattern recognition, pp 2467–2474
13. Kim Y, Kim D (2015) Efficient body part tracking using ridge data and data pruning. In: IEEE-RAS 15th international conference on humanoid robots, pp 114–120
14. Lee M, Nevatia R (2009) Human pose tracking in monocular sequence using multilevel structured models. *IEEE Trans Pattern Anal Mach Intell* 31(1):27–38
15. Ofli F, Kurillo G, Obdrzalek S, Bajcsy R, Jimison H, Pavel M (2015) Design and evaluation of an interactive exercise coaching system for older adults: lessons learned. *J Biom Health Inform* 20(1):201–212
16. Plagemann C, Ganapathi V, Koller D, Thrun S (2010) Real-time identification and localization of body parts from depth images. In: IEEE international conference on robotics and automation, pp 3108–3113
17. Raptis M, Kirovski D, Hoppe H (2011) Real-time classification of dance gestures from skeleton animation. In: The 2011 ACM SIGGRAPH/Eurographics symposium on computer animation 147–156
18. Reyes M, Dominguez G, Escalera S (2011) Featureweighting in dynamic timewarping for gesture recognition in depth data. In: IEEE international conference on computer vision workshops, pp 1182–1188
19. Schramm R, Jung C, Miranda E (2015) Dynamic time warping for music conducting gestures evaluation. *IEEE Trans Multimedia* 17(2):243–255
20. Shotton J, Girshick R, Fitzgibbon A, Sharp T, Cook M, Finocchio M, Moore R, Kohli P, Criminisi A, Kipman A, Blake A (2013) Efficient human pose estimation from single depth images. *IEEE Trans Pattern Anal Mach Intell* 35(12):2821–2840

21. Sung J, Ponce C, Selman B, Saxena A (2011) Human Activity Detection from RGBD Images, plan, activity, and intent recognition, vol 64
22. Xia L, Chen C, Aggarwal J (2012) View invariant human action recognition using histograms of 3d joints. In: IEEE computer society conference on computer vision and pattern recognition workshops, pp 20–27
23. Yang X, Tian Y (2014) Effective 3d action recognition using eigenjoints. *J Vis Commun Image Represent* 25(1):2–11
24. Ye M, Yang R (2014) Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In: IEEE conference on computer vision and pattern recognition, pp 2345–2352
25. Yun K, Honorio J, Chattopadhyay D, Berg T, Samaras D (2012) Two-person interaction detection using body-pose features and multiple instance learning. In: IEEE computer society conference on computer vision and pattern recognition workshops, pp 28–35



Yeonho Kim received the B.S. degree in computer engineering from Catholic University of Korea, Seoul, Korea, in 2008. Now, he is working toward the M.S. leading to Ph.D. degree in computer science and engineering at Pohang University of Science and Technology (POSTECH), Pohang, Korea. His current research interests include computer vision, human computer interaction, and human pose estimation.



Daijin Kim received the B.S. degree in electronic and engineering from Yonsei University, Seoul, South Korea, in 1981, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejeon, 1984. In 1991, he received the Ph.D degree in electrical and computer engineering from Syracuse University, Syracuse, NY. During 1992–1999, he was an Associate Professor in the Department of Computer Engineering at DongA University, Pusan, Korea. He is currently a Professor in the Department of Computer Science and Engineering at POSTECH, Pohang, Korea. His research interests include face and human analysis, machine intelligence and advanced driver assistance systems.