Doctoral Dissertation

3D Human Pose Estimation Using Ridge Data in Depth Image

Yeonho Kim (김 연 호)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2019

깊이 영상내 산등성이 데이터를 이용한 3차원 사람 자세 추정

3D Human Pose Estimation Using Ridge Data in Depth Image

3D Human Pose Estimation Using Ridge Data in Depth Image

by

Yeonho Kim

Department of Computer Science and Engineering

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

A dissertation submitted to the faculty of Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science and Engineering

Pohang, Korea

1. 4. 2019

Approved by

Daijin Kim Daijin Kim

Academic Advisor

3D Human Pose Estimation Using Ridge Data in Depth Image

Yeonho Kim

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a doctoral degree from POSTECH

1/4/2019

Committee Chair	Daijin Kim	(Seal)
Member	Minsu Cho	(Seal)
Member	Suha Kwak	(Seal)
Member	Soon-Yong Park	(Seal)
Member	Heeyoul Choi	(Seal)

DCSE 김 연 호, Yeonho Kim,
20090876 3D Human Pose Estimation Using Ridge Data in Depth Image,
깊이 영상내 산등성이 데이터를 이용한 3차원 사람 자세 추정,
Department of Computer Science and Engineering, 2019,
61 p, Advisor: 김대진(Daijin Kim). Text in English

ABSTRACT

Human pose estimation is the most fundamental technology to understand human behavior in various practical areas such as action recognition, human-computer interaction, entertainment. This thesis proposes ridge data that is the local maxima in the distance transform map of a single depth image and shows the selective representation of body skeleton although there have occlusion, full-body rotation, and fast movement.

We need to segment human silhouette from depth image in order to extract the ridge data. The process of human segmentation consists of four steps; floor removal, object segmentation, human detection, and human identification. Then, we compute the distance transform map from the edge image of the segmented human silhouette. The ridge data is extracted by finding the local maxima in the distance transform map. To show the effectiveness of the ridge data, we consider two types of human pose estimation using the ridge data: (1) feature-based hierarchical human pose estimation and (2) convolutional neural networks-based human pose estimation.

The feature-based method hierarchically tracks the human joints by pruning the invalid data according to an initial human model. The parameters of the initial human model are the lengths and angles of body parts and they are either generated from an initial pose or retrieved from the human pose database. The feature-based human pose estimation performs four functional subtasks sequentially: joint prediction, candidates collection, invalid-data pruning, and joint estimation. The subtasks are performed to track the human joints in a hierarchical order of head, torso, and limbs.

The convolutional neural network(CNN)-based method consists of two methods: (1) shallow CNN-based regression method and (2) multi-channel CNN-based regression method. The shallow CNN-based regression method consists of three convolutional layers and three fully-connected layers and directly regress the 3D human pose from the input depth image using three types of loss functions. We utilize the ridge data as one additional channel whose pixel represents the ridgeness at the position. The multi-channel CNN-based regression method projects the depth image and ridge data onto three orthogonal planes and generates 2D heatmaps to estimate the keypoints on each plane. The estimated keypoints of each plane are concatenated and fed to three fully-connected layers to regress the 3D human pose.

We consider the K-Pop dance teacher to evaluate the learner's dance performance automatically in order to show the accuracy of the proposed human pose estimation method. The K-Pop dance teacher evaluates the learners' dance performances concerning timing and pose accuracy using dance features that encode the learner's pose as the relative angles between adjusted joints.

To validate our proposed methods, we conduct several experiments on the

benchmark dataset SMMC-10 and EVAL, and a sizeable K-Pop dance dataset. To validate the effectiveness of the proposed ridge data, we compare the ridge data with the existing skeletonization techniques such as medial axis transform (MAT) and dilated medial axis transform (DMAT). The proposed feature-based human pose estimation method achieves the pose estimation accuracy of 0.9735 and 0.9358 mAP, and average pose error 3.88 and 4.72 cm on the SMMC-10 and EVAL dataset, respectively, and the average computation time of 3.45 ms (290 fps). The proposed four- and six-channel CNN-based human pose estimation method achieved the pose estimation accuracy of 0.9667 and 0.9801 mAP, respectively, on the EVAL dataset. The proposed K-Pop dance teacher achieves 98% concordance with the experts' evaluation of dance performance.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Human pose estimation is a challenging problem that is trying to find human model parameters such as the length and orientation of body parts (e.g., head, torso, limbs) that fit an input image, and to find the trajectories of specified skeletal joints. Interactive human pose estimation can be used in several applications that include entertainment, kinesiology, security, health-care systems, and marker-less motion capture [1, 2]. Recently, with an introduction of high-speed depth imaging devices, rich information can be obtained from depth silhouettes, so human pose estimation is simplified.

There are two approaches to identifying human poses from depth silhouettes: discriminative and generative. The discriminative approach is to find human joints from an observed input image using the pre-trained body part detectors. Plagemann *et al.* [3] proposed an interest-point detector to localize body parts in a single depth image. However, this method failed when body parts occluded each other. Jain *et al.* [4] proposed a head-torso detector based on Haar candidates and a template matching algorithm for each limb, but it required that

upper-body and face be visible without any occlusion. Shotton *et al.* [5] proposed a human pose recognition approach that predicted an intermediate body-part representation to estimate the human pose but the prediction usually required expensive training steps and a large number of training samples to cover the wide human pose-space. Girshick *et al.* [6] used a regression forest to localize the joint positions directly from votes of each pixel, but it required increasingly complex training because their voting was modeled from the result of [5]. Wang *et al.* [7] used semi-local features extracted from randomly-sampled 4D sub-volumes of depth sequences and it required highly-complicated training for large processing time. Buys *et al.* [8] described a method to transform the depth image into the intermediate representation without background subtraction. Most of these discriminative approaches do not miss the body parts totally but are suffered from the occlusion problem because they can detect only visible parts, which degrades the human pose estimation accuracy seriously.

The generative approach is to find human joints by fitting the pre-defined human body model to an observed input image. Grest *et al.* [9] and Knoop *et al.* [10] proposed an iterative closest points (ICP) method to estimate human poses and to track human body parts, but they were computationally complicated, so they were not applicable in real-time systems. Rosenhahn *et al.* [11] presented a marker-less motion capture system that took the lower-dimensional human pose manifold into account by using soft constraints to model the motion restrictions during human pose optimization, but it could not be used for challenging outdoor scenes that included shadows and strong illumination changes. Straka *et al.* [12] addressed the occlusion problems by using graph-based inference, and Ye *et al.* [13] did so by using joint energy minimization, but they have limited applicabilities because they required multiple cameras. Zhang *et al.* [14] used a parameterized human shape model that estimated the human pose from all

available data retrieved from several cameras. Although it improved human pose estimation, it still could not differentiate occluded body parts. Ganapathi *et al.* [15] proposed an efficient filtering algorithm for tracking human pose that combined an accurate generative model with a discriminative model that provided data-driven evidence about body part locations. Most of these generative approaches can solve the occlusion problem in some degree by exploiting prior knowledge of human body model and requires no training steps but fails to track the body parts entirely and takes long computation time because they fit a complicated human model to an observed input image iteratively.

Recently, convolutional neural networks (CNNs) have been successfully applied to the task of pose estimation. Toshev and Szegedy [16] proposed to directly regress the 2D Cartesian coordinate of joints in a holistic manner. Cao *et al.* [17] used the heatmaps that are intermediate representations for each joint to refine the joint position in the 2D pose estimation. Huang and Altamar [18] took depth images to regress joint position for 3D pose estimation using a simple convolutional architecture [19] and a large synthetic dataset.

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 [20, 21]. Several researchers considered ways to recognize human behaviors by using a hidden Markov model (HMM) [22] and dynamic time warping (DTW) [23]. Moreover, various activity recognition has been explored such as off-line activity recognition [24], online activity recognition [25], and human-to-human interaction recognition [26].

Most applications so far have focused on classifying human activities and evaluating the precision of human actions. Raptis *et al.* [27] proposed a distance metric to examine dancing gestures. Schramm *et al.* [28] used dynamic time

warping to recognize and evaluate music-conducting gestures. Ofli *et al.* [29] applied a performance-evaluation method to a health-care application.

The main contributions of this thesis are; First, we propose the ridge data that is a novel representation of the human body. The ridge data is more plentiful and scale-invariant representation than the existing skeletonization techniques [30, 31]. The ridge data help both feature-based and deep learning based methods. Second, we achieve higher pose estimation accuracy (0.9801 mAP) than the current state-of-the-art methods on EVAL dataset. Third, the proposed feature-based method estimates each human joint efficiently (290 fps) by using ridge data and data pruning. Forth, we constructed a sizeable K-Pop dance database, which contains 100 dance experts' sequences and 400 dance learners' sequences for the dance teacher program. Fifth, the proposed dance teacher program achieved 98% agreement with experts' dance evaluation.

The rest of this thesis is organized as follows. Chapter 2 describes the proposed human depth silhouette segmentation. Chapter 3 describes the motivation, definition and extraction of ridge data, which is a novel feature for human pose estimation. Chapter 4 describes the proposed initial human model parameter acquisition and hierarchical human pose estimation. Chapter 5 describes the CNN-based methods: shallow and multi-channel CNN-based regression methods. Chapter 6 shows a practical application of the proposed method that automatically evaluates the learner's dance performance, Chapter 7 validates the proposed feature and methods using the experiments on the SMMC-10 and EVAL datasets. Finally, chapter 8 presents conclusions.

CHAPTER 2

Human Segmentation

The process of human segmentation consists of four steps: floor removal, object segmentation, human detection, and human identification. (see Fig 2.1.)

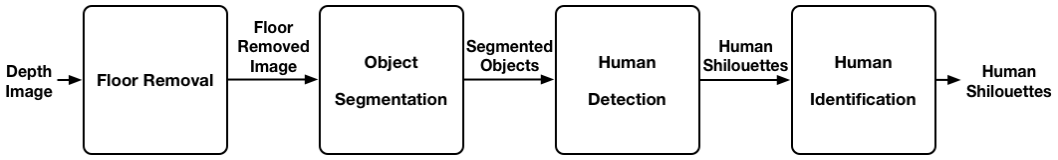


Figure 2.1. Overall process of human segmentation.

2.1 Floor Removal

We use the floor-removal technique to disconnect every object that is connected with the floor. We know that the y values of the floor data are smaller than those of the other objects because the floor is located at the bottom of depth image (see Fig. 2.2(a)). Therefore, we collect the raw depth data with the smallest y value as the initial floor data candidates. Then, we model the initial floor data

candidates as a planar equation

$$\begin{aligned} ax_1 + by_1 + c &= z_1, \\ \vdots &= \vdots \\ ax_n + by_n + c &= z_n, \end{aligned} \tag{2.1}$$

where $(a, b, 1)$ denote the floor normal vector, c denotes the plane distance from the origin of the 3D coordinate system and n is the number of the candidates. These equations can be solved by the least squares method in matrix form as

$$\sum_{i=1}^{n} \left(ax_i + by_i + c - z_i\right)^2 = \left|\mathbf{X}A - \mathbf{z}\right|^2, \tag{2.2}$$

where $\mathbf{X}$ is an $n \times 3$ matrix of $(x_1, y_1, 1; \ldots; x_n, y_n, 1)$ and A denotes a vector of unknown variables (a, b, c). To minimize the sum of squared errors, we derive the partial derivative of the Eq. (2.2) with respect to A as

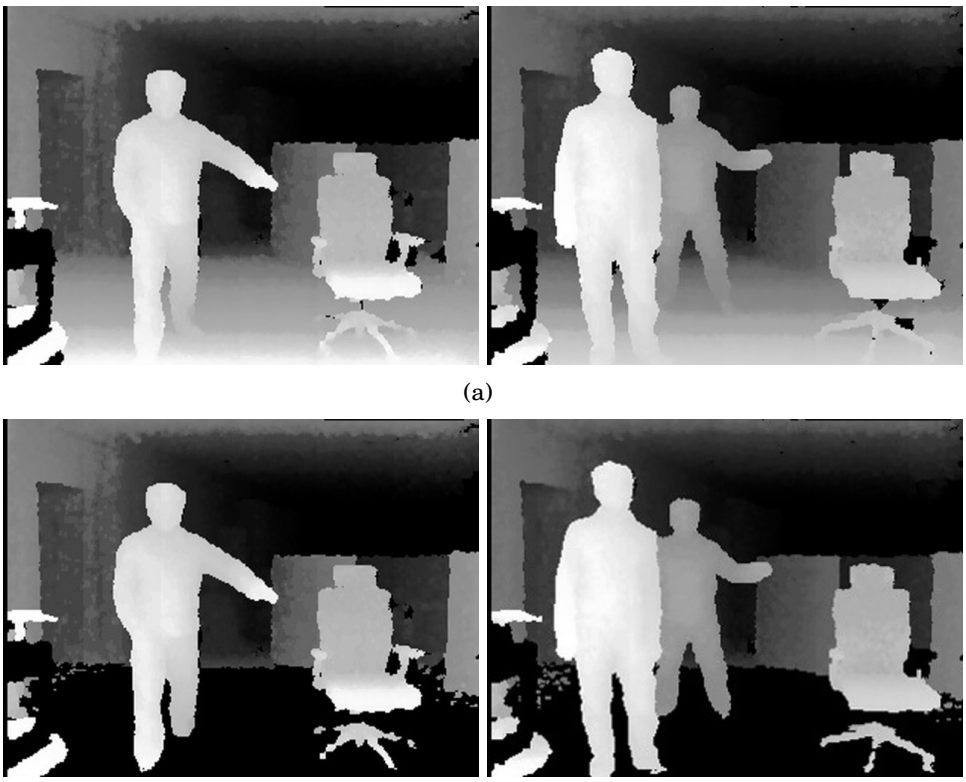
$$\begin{aligned} \frac{\partial \left|\mathbf{X}A - \mathbf{z}\right|^2}{\partial A} &= -2\mathbf{X}^T\mathbf{X}A + 2\mathbf{X}^T\mathbf{z} = 0, \\ \mathbf{X}^T\mathbf{X}A &= \mathbf{X}^T\mathbf{z}, \\ A &= \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{z}, \end{aligned} \tag{2.3}$$

where we obtain the floor normal vector $(a, b, 1)^T$ and the plane distance c from $A = (a, b, c)^T$.

We use new floor data candidates that satisfy the current floor planar equationto update the floor planar equation, then repeat the update procedure until the floor planar equation stops changing; the result (Fig. 2.2(b)) is the depth image.

2.2 Object Segmentation

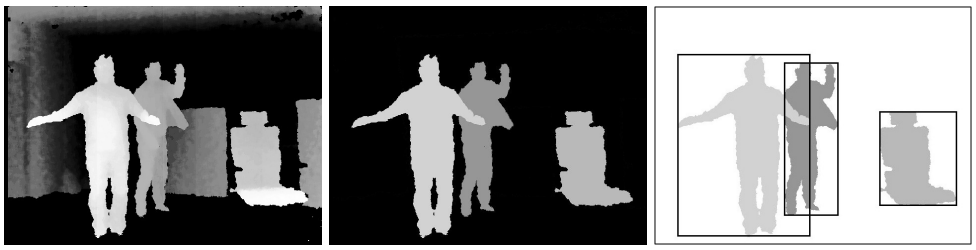
We use the 3D connected component labeling (CCL) technique [32] to segment the objects in the original depth image (Fig. 2.3(a)). The CCL algorithm uses a



(a)

(b)

Figure 2.2. Some depth images (a) before floor removal and (b) after floor removal.



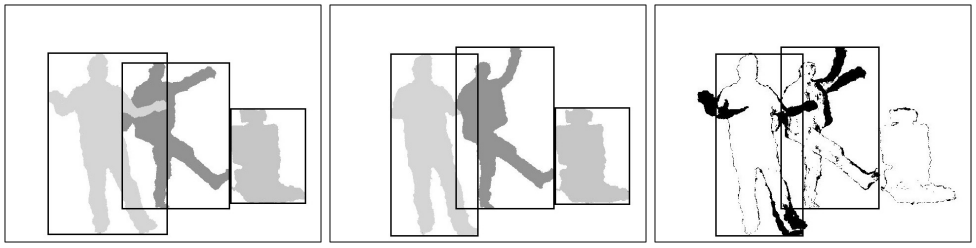
(a) (b) (c)

Figure 2.3. Object segmentation using 3D CCL technique (see the text).

binary connectivity criterion to assign a unique label to each connected component. To process the depth image, we modify the binary connectivity criterion that considers the difference in the depth values of neighboring pixels as follows. When $|d(X_i) - d(X_i + \mathbf{i})| \leq \epsilon_z$, two neighboring pixels are contained into the same object (Fig. 2.3(b)), where $\mathbf{i} = \{(0,-1),(-1,0),(+1,0),(0,+1)\}$, $d(X_i)$, $d(X_i)$ denotes the depth value at the position $X_i = (u_i, v_i)$ and ϵ_z is a threshold depth value. In this work, ϵ_z is chosen appropriately such that the 3D CCL works irrespective of the size of the connected components, the frequent changes of background and the distance (i.e., between 2 m and 3.5 m) from object to the camera. The final result (Fig. 2.3(c)) segments objects including humans, chair, and the background.

2.3 Human Detection

We identify human objects among the segmented objects but assuming that only they move. The motion information can be obtained by computing the depth difference between two consecutive frames $t-1$ (Fig. 2.4(a)) and t (Fig. 2.4(b)) 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 ($\Delta d(B_i)$) as a measure to discriminate moving objects from



(a) (b) (c)

Figure 2.4. Human detection using the sum of depth differences ($\Delta z(B_i)$). In, the first and second segmented objects are identified as the humans because their sums of depth differences exceed ϵ_m and they are enclosed within their own bounding boxes. (procedure is described in the text.)

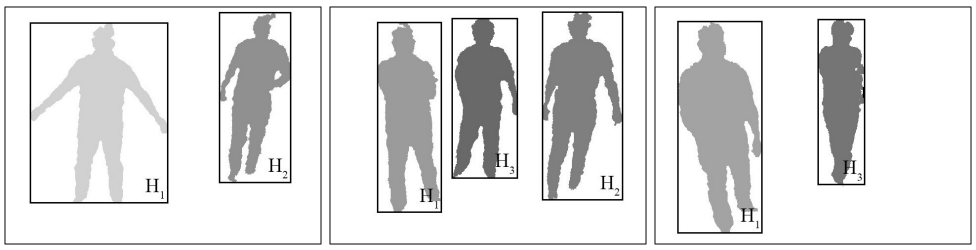
stationary objects as

$$\Delta d(B_i) = \sum_{X_i \in B_i} \delta d(X_i), \tag{2.4}$$

where B_i is the bounding box of the ith segmented object. If $\Delta d(B_i) \geq$ threshold motion value ϵ_m, the segmented object is identified as a human (Fig. 2.4(c)).

2.4 Human Identification

We assign each detected human a unique ID that does not change over a sequence of depth images. We use the simple association rule that we compute the distances between a specific human in the current frame and all humans in the previous frame, and assume that the specific human in the current frame is the human for whom this distance is minimal. Let the number of humans be N^{t-1} in the $t-1$th frame and N^t in the tth frame. Then three situations are possible: (1) no change in the number of humans ($N^{t-1} = N^t$), (2) one or more new humans appear ($N^{t-1} < N^t$), and (3) one or more existing humans disappear ($N^{t-1} > N^t$) (Fig. 2.5).



(a) (b) (c)

Figure 2.5. Human identification using the simple association rule (process is described in the text).

CHAPTER 3

Ridge Data

3.1 Motivation

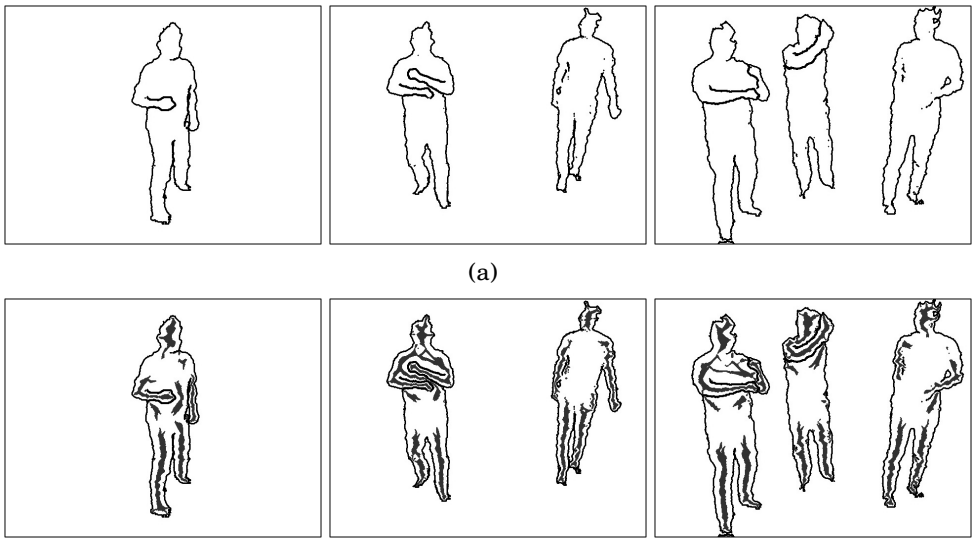
The motivation to address a novel feature called ridge data is to solve the joint positional drift problem. When we use the raw depth data directly in the human pose estimation, the estimation errors of the overlapping body part are likely to be increased because it is uncertain to distinguish the raw depth data among the overlapped body parts. For example, when a forearm overlaps the torso part, raw depth data from the torso part can be included as the joint candidates for the hand position. This uncertainty makes the joint positions more likely to be drifted. To solve this uncertainty, we collect the proper candidates inside the body part from the depth edge image, which is easily obtained from the segmented human silhouette. Unlike the 2D edge image, the depth edge image can distinguish the boundary of overlapped body parts.

3.2 Definition of Ridge Data

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

$$R(I) = \left\{ X_i \in I \;\middle|\; \frac{\sum_{X' \in C(X_i)} D_T(X')}{\sum_{X' \in C(X_i)} I(X')} \leq \epsilon_r \times D_T(X_i) \right\}, \tag{3.1}$$

where $D_T(X')$ is the pixel distance in the distance transform map (DTM) at the pixel position X', $I(X')$ 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 ridge parameter that controls the amount of ridge data by having richer ridge data as ϵ_r increases. In this work, we choose 0.8 experimentally as an optimal value of ϵ_r. The process converts depth edge images (Fig. 3.1(a)) to corresponding ridge data images (Fig. 3.1(b)).



(a)

(b)

Figure 3.1. Conversion of (a) depth edge images to (b) ridge data.

The intuitive interpretation of the ridge data can be explained as follows. In

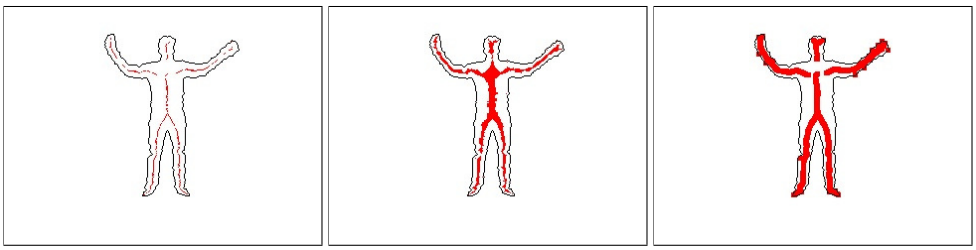
the ridge region of DTM, the distances $D_T(X')$ to the closest edge from the circle $C(X_i)$ centered on X_i are smaller than the distance $D_T(X_i)$ between the center X_i and the closest edge. Therefore, the distance value of the center is greater than the average value of the distances on the circle. 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.

3.3 Difference with Medial Axis Transform

Existing skeletonization techniques such as shock graph and medial axis transform [30, 31] use the medial axis information that is obtained from the zero gradients of the DTM that is obtained from the silhouette image of the 2D image (see Fig. 3.2(a)). Since the skeletonization techniques try to make the features compact as much as possible, the extracted medial axis features are useful for feature matching in the object recognition and posture recognition. However, the proposed human joint tracking is based on the pruning of invalid data, and it is necessary to have valid skeleton data as much as possible. To meet this requirement, we propose a novel 3D feature (ridge data) that represents the skeletal shape of a human silhouette plentifully [33], which is obtained from a set of depth pixels that are located in the local maximal region of a distance transform map (DTM) [34] (see Fig. 3.2(b)).

To make the medial axis feature plentiful, we apply the dilation operation (see Fig. 3.2(c)). The dilated medial axis feature looks similar to the proposed ridge data but has some serious defects. First of all, the dilation is controlled by a predefined kernel which governs the amount of dilation. Therefore, the

dilated medial axis feature in the arms covers most of arm parts. This means that the proposed ridge data can capture the skeleton of the body part, but the dilated medial axis feature include most of the body parts. Secondly, the dilated medial axis feature may include the outside region of body parts. Even if we can eliminate the feature points outside body parts using the human segmentation results, the feature points near the boundary of body parts make the dilated medial axis feature to lose the accurate localization capability for joint positions.



(a) (b) (c)

Figure 3.2. Comparison of (a) medial axis, (b) ridge data, and (c) dilated medial axis.

3.4 Ridge Data Extraction

The process of extracting ridge data from the depth map image involves four steps (see Fig. 3.3). (1) Build the depth edge map that is obtained by thresholding the depth difference between two neighboring pixels (body contour line in Fig. 3.3(a)). (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. 3.3(b)). (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_{DT} of the average D_T over the pixel's $D_T(X_i)$ (Fig. 3.3(c)).

(4) Take pixels for which R_{DT} is smaller than a given threshold value as the local maximal pixels, i.e., ridge data (Fig. 3.3(d)).

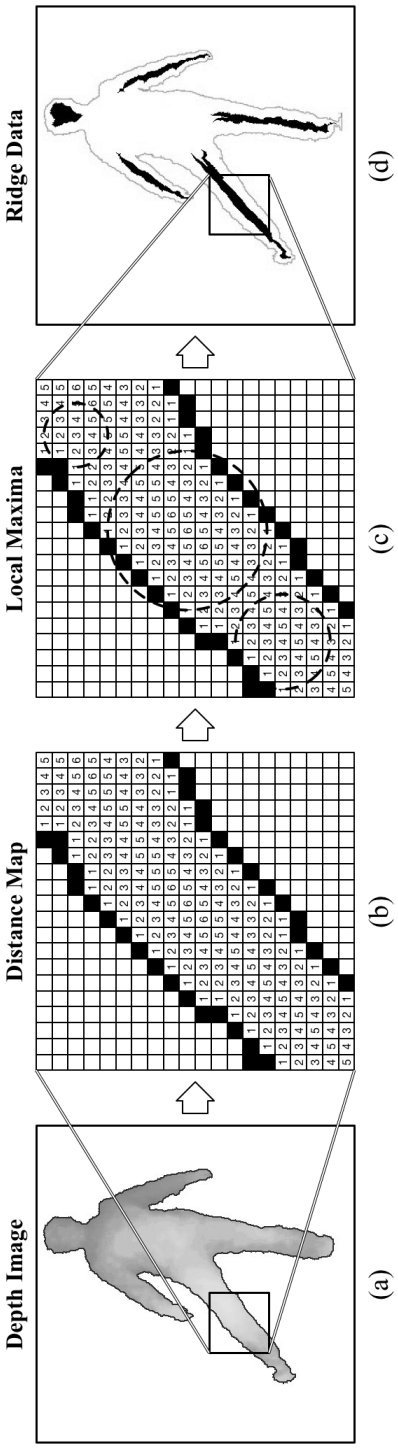


Figure 3.3 Overall process of extracting ridge data from the depth map image.

CHAPTER 4

Feature-Based Hierarchical Human Pose Estimation

In this chapter, we propose hierarchical human joint tracking using depth-silhouette-based ridge data. The proposed method generates or retrieves a set of initial human model parameters for body parts with a tree-structured kinematic model that considers 15 joint points of body parts, and a specially-proposed shoulder structure. Then, the model estimates the positions of human body parts by tracking the human body joints in a hierarchical, top-down manner by eliminating invalid ridge data by consulting predefined model parameters and by constraining the local search area of each body part. To the best of our knowledge, this is the first approach that uses ridge data along with depth values for 3D human pose estimation and tracking from depth silhouettes.

4.1 Initial Human Model Parameter Acquisition

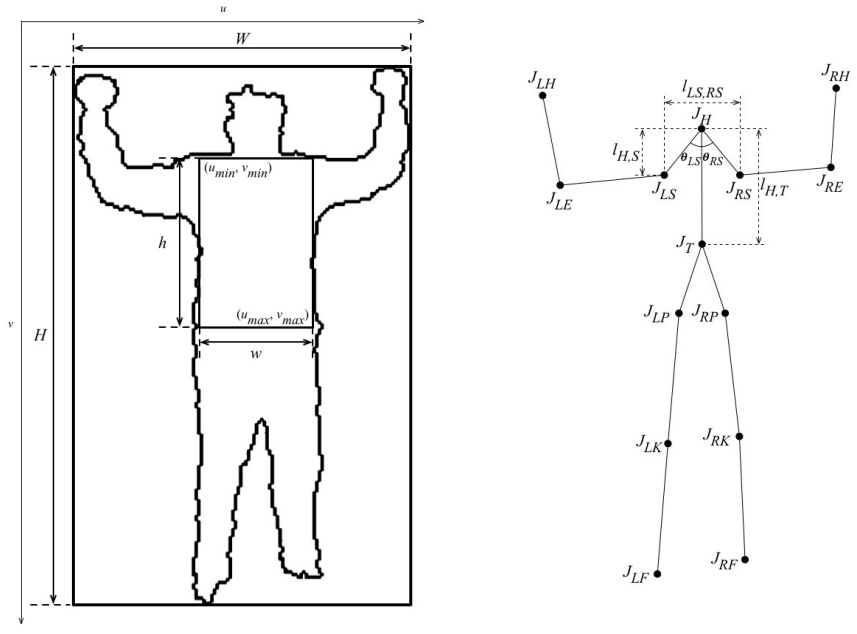
For the proposed feature-based human pose estimation, we must acquire the initial human model parameters such as the lengths and angles between adjacent joints. The human body model is characterized by 14 distinct joints: J_H (center of head), J_{LH} (left hand), J_{LE} (left elbow), J_{LS} (left shoulder), J_{RH} (right hand), J_{RE} (right elbow), J_{RS} (right shoulder), J_T (center of torso), J_{LP} (left pelvis), J_{LK} (left knee), J_{LF} (left foot), J_{RP} (right pelvis), J_{RK} (right knee), and J_{RF} (right foot), and two distinct angles θ_{LS} (angle between J_H and J_{LS}) and θ_{RS} (angle between J_H and J_{RS}) (see Fig. 4.1(b)). In this work, we propose two methods to acquire human model parameters: (1) generation of human model parameters from a specific initial pose and (2) retrieval of human model parameters from an initial-pose database.

4.1.1 Method I: Estimation of human model parameters

Estimation of the human model parameters is a semi-automatic method that acquires the initial human model parameters hierarchically. The overall procedure of estimating human model parameters from a specific initial pose consists of six steps.

Step 1: Initial Positions of The Center of Torso J_T

The proposed method needs each human to hold an initial Y-shaped pose. To locate J_T, we model the region of the torso as a maximal rectangle $(u_{min}, v_{min}, u_{max}, v_{max})^T$ that contains the upper human body, where (u_{min}, v_{min}) and (u_{max}, v_{max}) are the upper left corner position and lower right corner position of the maximal rectangle. Consider a bounding box of width W and height H that contains the human silhouette. Then (u_{min}, u_{max}) and (v_{min}, v_{max}) can be determined by column-wise and row-wise scanning, respectively (see Fig. 4.1(a)). The column-



(a) (b)

Figure 4.1. (a) Initial Y-shaped pose and maximal rectangle of torso and (b) 14 joint of human body model and HST structural parameters.

wise and row-wise scanning are a maximal length of connected depth pixels in the column and row, respectively, as

$$\begin{bmatrix} u_{min} \\ v_{min} \\ u_{max} \\ v_{max} \end{bmatrix} = \begin{bmatrix} \underset{u}{\operatorname{argmin}} \{length_c(u) > 0.5 \times W\} \\ \underset{v}{\operatorname{argmin}} \{length_r(v) > 0.5 \times H\} \\ \underset{u}{\operatorname{argmax}} \{length_c(u) > 0.5 \times W\} \\ \underset{v}{\operatorname{argmax}} \{length_r(v) > 0.5 \times H\} \end{bmatrix}, \tag{4.1}$$

where $length_c(i)$ and $length_r(j)$ are a maximal length of connected depth pixels in the ith column and the jth row, respectively. Then, the torso width w and torso height h are computed as $w = x_{max} - x_{min}$ and $h = y_{max} - y_{min}$, where x_{max} and x_{min} are the maximum and the minimum value of the x axis among 3D points within the maximal rectangle, respectively, and y_{max} and y_{min} are the maximum and the minimum value of the y axis among 3D points within the maximal rectangle, respectively.

Step 2: Initial Positions of Head Center

Then, we estimate the position of the center of the head. We guess that the head is located within a bounding box that has upper left corner $(u_{min}, 0)$ and lower right corner (u_{max}, v_{min}) from the maximal rectangle of torso. We use a simple head center search method that collects the ridge data within the bounding box and use a mean shift to find a local maximum of the collected ridge data. We place the initial position of J_H at the position of the local maximum.

Step 3: Initial Positions of Torso Joints

Then, we estimate the positions of torso joints (J_{LS}, J_{RS}), and (J_{LP}, J_{RP}), and J_T sequentially. We divide the maximal rectangle into top, middle, and bottom parts with a ratio of $(0.3, 0.4, 0.3)$ then divide the top part into the left, middle, and right parts with a ratio of $(0.3, 0.4, 0.3)$ and divide the bottom part into the

left and right parts with a ratio of $(0.5, 0.5)$ to yield a total of six sub-parts. Then we estimate the positions of J_{LS} and J_{RS} by averaging the depth pixels within the top left and the top right sub-parts, respectively. We estimate the positions of J_{LP} and J_{RP} by averaging the depth pixels within the bottom left and the bottom right sub- part, respectively. We estimate the position of J_T by averaging the positions of shoulders and pelvises.

Step 4: HST Structural Parameters

Then, we construct a head-shoulder-torso (HST) structure which consists of ten distinct lengths and two angles. The lengths are $l_{H\perp S}$ perpendicular from the head center to the left and right shoulders, $l_{LS,RS}$ between the left 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 left and right pelvis, $l_{LS,LP}$ between the left shoulder and the left pelvis, $l_{LS,RP}$ between the left shoulder and the right pelvis, $l_{RS,LP}$ between the right shoulder and the left pelvis, and $l_{RS,RP}$ between the right shoulder and the right pelvis. The angles are θ_{LS} between the head center and the left shoulder, and θ_{RS} between the head center and the right shoulder (see Fig. 4.1(b)). The length parameters in

the HST structure can be computed as

$$\begin{bmatrix} l_{H\perp S} \\ l_{LS,RS} \\ l_{H,T} \\ l_{LS,T} \\ l_{RS,T} \\ l_{T\perp P} \\ l_{LS,LP} \\ l_{LS,RP} \\ l_{RS,LP} \\ l_{RS,RP} \end{bmatrix} = \begin{bmatrix} \frac{\left\| \overrightarrow{J_{LS}J_{RS}} \times \overrightarrow{J_H J_{LS}} \right\|}{\left\| \overrightarrow{J_{LS}J_{RS}} \right\|} \\ \|J_{LS} - J_{RS}\| \\ \|J_H - J_T\| \\ \|J_{LS} - J_T\| \\ \|J_{RS} - J_T\| \\ \frac{\left\| \overrightarrow{J_{LP}J_{RP}} \times \overrightarrow{J_T J_{LP}} \right\|}{\left\| \overrightarrow{J_{LP}J_{RP}} \right\|} \\ \|J_{LS} - J_{LP}\| \\ \|J_{LS} - J_{RP}\| \\ \|J_{RS} - J_{LP}\| \\ \|J_{RS} - J_{RP}\| \end{bmatrix}, \tag{4.2}$$

where $\overrightarrow{J_{LS}J_{RS}} = J_{RS} - J_{LS}$ denotes a vector from the left shoulder to the right shoulder, $\overrightarrow{J_H J_{LS}} = J_{LS} - J_H$ denotes a vector from the head center to the left shoulder, $\overrightarrow{J_{LP}J_{RP}} = J_{RP} - J_{LP}$ denotes a vector from the left pelvis to the right pelvis, and $\overrightarrow{J_T J_{LP}} = J_{LP} - J_T$ denotes a vector from the torso center to the left pelvis. The angle parameters in the HST structure can be computed as

$$\begin{bmatrix} \theta_{LS} \\ \theta_{RS} \end{bmatrix} = \begin{bmatrix} cos^{-1}\left(\frac{\overrightarrow{J_H J_{LS}} \cdot \overrightarrow{J_H J_T}}{\left\| \overrightarrow{J_H J_{LS}} \right\| \left\| \overrightarrow{J_H J_T} \right\|} \right) \\ cos^{-1}\left(\frac{\overrightarrow{J_H J_{RS}} \cdot \overrightarrow{J_H J_T}}{\left\| \overrightarrow{J_H J_{RS}} \right\| \left\| \overrightarrow{J_H J_T} \right\|} \right) \end{bmatrix}, \tag{4.3}$$

where $\overrightarrow{J_H J_{RS}} = J_{RS} - J_H$ denotes a vector from the head center to the right shoulder and $\overrightarrow{J_H J_T} = J_T - J_H$ denotes a vector from the head center and the torso center.

Step 5: Initial Positions of Limb Joints

Then, we estimate the positions of limbs that contain two arms and two legs, where each arm includes the hand and elbow; and each leg includes the foot and knee. We propose to use Hough transform over the ridge data (see Fig. 4.2(a)) within a left arm bounding box that has the upper left corner $(0, 0)$ and the lower right corner $(u_{min}, \frac{H}{2})$. Similarly we process a right arm bounding box that has the upper left corner $(u_{max}, 0)$ and lower right corner $(W, \frac{H}{2})$. The Hough transform algorithm provide several straight lines within each arm bounding box, which are parameterized as (r, θ) (see Fig. 4.2(b)). Among many lines, we take two dominant straight lines that correspond to the upper and lower arm by applying the k-means algorithm with $k = 2$ over parameter space. We estimate the position of the left elbow J_{LE} (see Fig. 4.2(c)) to be at the intersection point of two straight lines within the left arm bounding box and estimate the position of the left hand J_{LH} (see Fig. 4.2(d)) by averaging the ridge data near the highest end within the left arm bounding box. We similarly process the right arm bounding box to estimate the positions of the right elbow J_{RE} and hand J_{RH}.

We estimate the position J_{LF} of the left foot by averaging the depth pixels of the ridge data near the lowest ends within the left leg bounding box of which the left upper corner is (u_{min}, v_{max}) and the right lower corner is $(\frac{u_{min}+u_{max}}{2}, H)$. Similarly, we estimate the position J_{RF} of the right foot by processing the right leg bounding box in which the left upper corner is $(\frac{u_{min}+u_{max}}{2}, v_{max})$ and the right lower corner is (u_{max}, H). We estimate the position J_{LK} of the left knee as the center position between the left pelvis and the left foot. We estimate the position J_{RK} of the right knee as the center position between the right pelvis and the right foot.

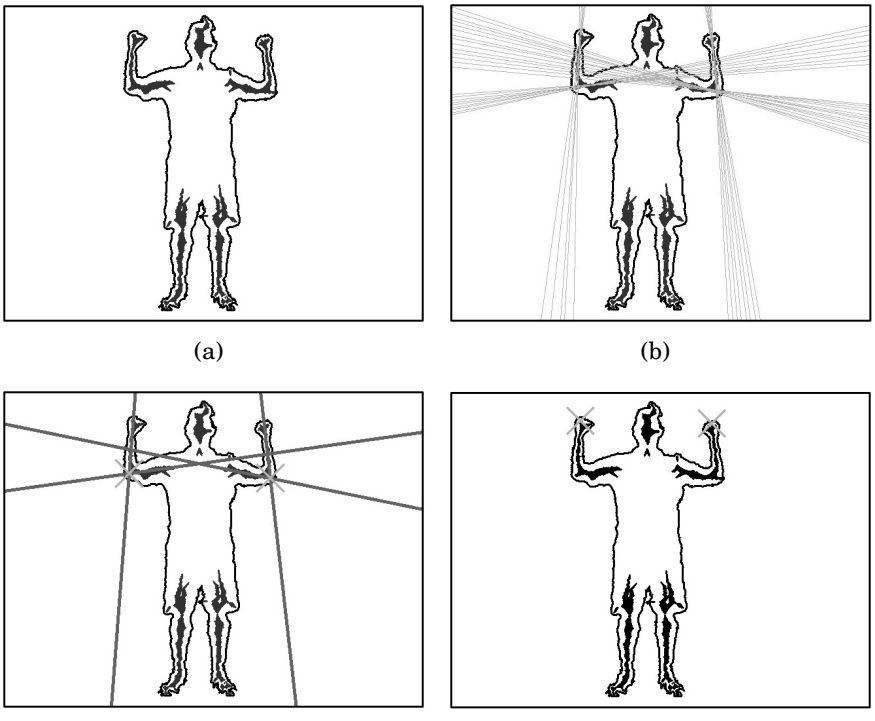


Figure 4.2. Overall procedure of estimating the elbows and the hands (see the text).

Step 6: Limb Parameters

After estimating the hand position, we compute the lengths of the upper and lower arms:

$$\begin{bmatrix} l_{LUA} \\ l_{RUA} \\ l_{LLA} \\ l_{RLA} \end{bmatrix} = \begin{bmatrix} \|J_{LS} - J_{LE}\| \\ \|J_{RS} - J_{RE}\| \\ \|J_{LE} - J_{LH}\| \\ \|J_{RE} - J_{RH}\| \end{bmatrix}, \tag{4.4}$$

where l_{LUA}, l_{RUA}, l_{LLA} and l_{RLA} denote the lengths of the left upper arm, the right upper arm, the left lower arm, and the right lower arm, respectively.

After estimating the foot position, we compute the lengths of the upper and lower legs:

$$\begin{bmatrix} l_{LUL} \\ l_{RUL} \\ l_{LLL} \\ l_{RLL} \end{bmatrix} = \begin{bmatrix} \|J_{LP} - J_{LK}\| \\ \|J_{RP} - J_{RK}\| \\ \|J_{LK} - J_{LF}\| \\ \|J_{RK} - J_{RF}\| \end{bmatrix}, \tag{4.5}$$

where l_{LUL}, l_{RUL}, l_{LLL} and l_{RLL} denote the lengths of the left upper leg, the right upper leg, the left lower leg, and the right lower leg, respectively.

4.1.2 Method II: Retrieval of Human Model Parameters

Retrieval of human model parameters is an exemplar-based method that retrieves a human pose hypothesis that resembles the pose of the input depth image from the human pose database, then obtains the human model parameters of the input depth image from the predefined human model parameters of the retrieved human pose. Overall, the retrieval of human model parameters from the human pose database consists of five steps (Fig. 4.3).

Step 1: Preparation of Human Pose Dataset

We built a human pose database of 30,000 different human poses that covers a sufficient space of human poses and kinematic models. The images were collected using a commercial depth camera (Microsoft Kinect) to capture the

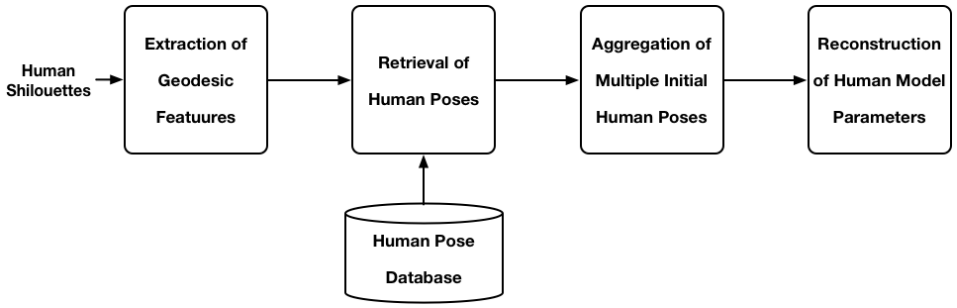


Figure 4.3. Overall process of the human model parameters retrieval.

silhouettes of seven subjects while they performed a variety of daily activities such as walking and stretching arms. We manually label 15 joints in each silhouette, then normalize the human pose as follow. First, we move the 15 joints such that the center position of the joints is located at $(0, 0, 0)$. Second, we apply the principal component analysis (PCA) to the collection of 15 joints and compute the rotation matrix such that the principal components are aligned to the global axes, i.e. $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$. To build a compact human pose database, we eliminated redundant poses by considering the joint differences in two human poses as $d(P^1, P^2) = \sum_{k=1}^{15} \left\| J_k^1 - J_k^2 \right\|$, where J_k^1 and J_k^2 denote the predefined position of the kth joint of the human pose 1 and 2, respectively, and eliminating two poses whenever their joint difference was smaller than 5 cm.

Step 2: Extraction of Geodesic Features

We consider a geodesic extremal feature [35] of each human pose to retrieve similar poses from the database. The feature is a rough representation of a certain human pose. The feature encodes the positions of hands, feet, and head as $F = (f_1, \ldots, f_5) \in (X)^5$, where f_k denotes one of the components, and X denotes the 3D depth coordinate of the human silhouette. We model the human

silhouette as a weighted graph (V, E), where the vertices (V) are the depth pixels and edges (E) are the distance weights $\|V_p - V_q\|$, and the vertex q is one of eight neighboring vertices of the vertex p.

The overall procedure of extracting geodesic features entails four steps. (1) Rotate the silhouette into an up-right pose by orienting the first principal axis obtained using PCA. (2) Apply a modified Dijkstra's algorithm to compute the number of geodesic extremal features. In contrast with the typical Dijkstra's algorithm, the modified Dijkstra's algorithm keeps the distance after finding an extremal feature, then uses this distance when restarting from the found extremal feature in the next iteration. (3) Normalize the maximum size of the geodesic extremal features into 1. (4) Use a kd-tree with 15 dimensions (=5 geodesic extremal features $times$ 3-dimensional positions) to represent each human pose efficiently.

Step 3: Retrieval of Human Poses

After obtaining the geodesic extremal features of the input human silhouette from step 2, we retrieve the human pose dataset and identify the five human poses that are most similar to the input human silhouette by using the pose similarity as

$$S(F^{in}, F^k) = exp(-\|F^{in} - F^k\|), \tag{4.6}$$

where F^{in} and F^k denote the 15-dimensional geodesic extremal features of the input human silhouette and the kth similar human silhouette in the human pose dataset, respectively.

Algorithm 1: Geodesic Extreme Features

Input: V: Vertices as depth pixels, E: Edges between neighboring pixels, S: Starting position

Output: Geodesic extremal features F

dist $\leftarrow$ inf;
dist[S] $\leftarrow$ 0;
$F \leftarrow \{\}$;
for $i = 1$ *to* M **do**
  $Q \leftarrow Q \cup \{S\}$;
  while Q *is not empty* **do**
    $u \leftarrow \underset{u}{\text{argmin}}$ dist[u];
    $Q \leftarrow Q \setminus \{u\}$;
    foreach *neighbor* v *of* u **do**
      $t \leftarrow$ dist[u] + length(u, v);
      if $t <$ *dist[*v **then**
        dist[v] $\leftarrow t$;
        $Q \leftarrow Q \cup \{v\}$;
      end
    end
  end
  $e \leftarrow \underset{u}{\text{argmax}}$ dist[u];
  $F \leftarrow F \cup \{e\}$;
  $S \leftarrow e$;
end

Step 4: Aggregation of Multiple Initial Human Poses

After obtaining the five most similar human poses, we aggregate them using a weighted average as

$$P_j^* = \sum_{k=1}^{5} S^k \cdot J_j^k, \tag{4.7}$$

where P_j^* denotes the jth joint of the aggregated human pose, S^k is the pose similarity between F^{in} and F^k, and J_j^k is the jth joint of the kth similar human pose.

Step 5: Reconstruction of Human Model Parameters

Because the aggregated human pose does not include the human model parameters of the input human silhouette, we must reconstruct them by deforming an articulated mesh model [36], which is optimized by alternating between (1) searching the points of the mesh model of the aggregated human pose that corresponds to the input human silhouette and (2) adjusting the mesh vertices and the joint positions such that the mesh model of the aggregated human pose fits the input human silhouette. The overall procedure of reconstructing human model parameters of the input human silhouette involves three steps. (1) Use linear blend skinning [37] to construct a mesh model of the aggregated human pose. (2) Use an iterative closest point (ICP) scheme to optimize the mesh model to fit the input human silhouette. The fitting is done to minimize the energy function as

$$E\left(\mathbf{V}, \mathbf{J}\right) = \lambda_{skin} E_{skin}(\mathbf{V}) + \lambda_{bone} E_{bone}(\mathbf{V}, \mathbf{J}) + \lambda_{deform} E_{deform}(\mathbf{V}), \tag{4.8}$$

where $\mathbf{V} = \{v_i \in \mathbb{R}^3\}$ denotes the vertices of the mesh M and $\mathbf{J} = \{J_j \in \mathbb{R}^3\}$ denotes the joint positions and E_{skin}, E_{bone}, and E_{deform} denote the energy functions of surface smoothness, vertex-bone attachment, and mesh deformation,

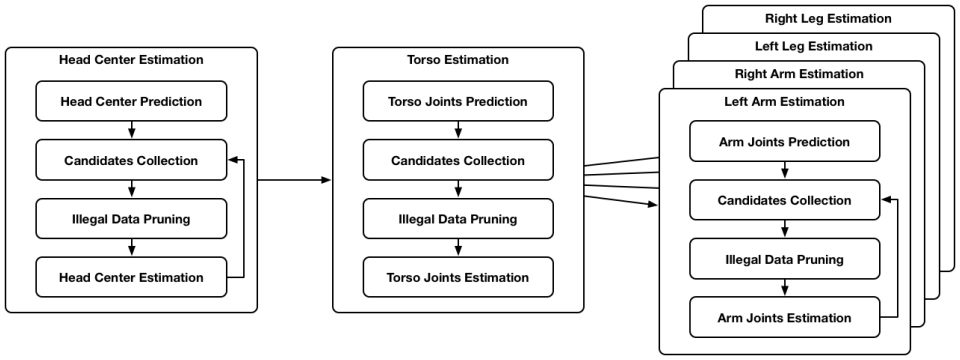


Figure 4.4. Overall process of the proposed hierarchical human pose estimation.

respectively. The surface smoothness energy function E_{skin} controls the deformation of surface meshes using Laplacian mesh editing. The vertex-bone attachment energy function E_{bone} controls the attachment of the mesh surfaces to the skeleton that consists of bones, which is a virtual line between joints. Finally, the mesh deformation energy function E_{deform} controls the displacement between mesh vertices and target depth data using the ICP method. Following the setting in [36], we use sparse Cholesky matrix decomposition to solve the energy minimization function. (3) Reconstruct the initial human model parameters from the optimized mesh model.

4.2 Hierarchical Human Pose Estimation

We propose a hierarchical human pose estimation method (Fig. 4.4) that determines the human joints in a top-down manner from the head through the torso to limbs. Estimation of the location of each joint involves four steps: joint prediction, candidate collection, invalid-data pruning, and joint estimation.

4.2.1 Head Center Estimation

The head center position is tracked using either the ridge data or raw depth data. When the ridge data are rich and sufficient, we use the ridge-data-based method to localize the exact position of head center. When the ridge data are insufficient due to the self-occlusion, we use the raw-depth-data-based method. The procedure of the ridge-data-based method involves four steps.

(1) We predict the head center position at the current frame from the head center position in the previous frame as

$$\widetilde{J}_H^t \leftarrow J_H^{t-1} + \Delta J_H^{t-1}, \tag{4.9}$$

where $\widetilde{J}_H^t$ is the predicted head center position in the tth frame and $\Delta J_H^{t-1} = J_H^{t-1} - J_H^{t-2}$ is the increment of the velocity model of the head center. This prediction of the joint position may reduce the computation time by limiting the search space for the joint tracking.

(2) We collect the candidates for the head center from the ridge data $R(I)$ as

$$C_H(I) = \left\{ X_i \in R(I) \;\middle|\; D_E(X_i, \widetilde{J}_H^t) \leq r_H \right\}, \tag{4.10}$$

where $D_E(X_i, \widetilde{J}_H^t)$ is the Euclidean distance between the pixel position $X_i = (x_i, y_i, z_i)$ and the predicted head center $\widetilde{J}_H^t$, and r_H is the radius of head as determined by the initial human model.

(3) We eliminate implausible candidates to leave only the valid set of head data according to the constraints of the initial human model. To apply the head to torso length constraint such as $l_{H,T}$ in Eq. (4.2), we need the torso center position. Because we do not know the exact torso center position at this moment, we predict it as

$$\widetilde{J}_T^t \leftarrow J_T^{t-1} + \Delta J_T^{t-1}, \tag{4.11}$$

where $\widetilde{J}_T^t$ is the predicted torso center position at the tth frame and $\Delta J_T^{t-1} = J_T^{t-1} - J_T^{t-2}$ is the increment of the velocity model of torso center. We use the

length constraint $l_{H,T}$ to reject invalid data, then collect the valid data as

$$V_H(I) = \left\{ X_i \in C_H(I) \;\middle|\; \left| D_E(X_i, \widetilde{J}_T^t) - l_{H,T} \right| \leq \epsilon_H \right\}, \tag{4.12}$$

where $l_{H,T}$ is the length parameter in the HST structure (Eq. (4.2)).

(4) When sufficient valid ridge data remain after eliminating the invalid data, we aggregate the collected valid data into the estimated position for the head center as

$$J_H^t = \frac{1}{|V_H(I)|} \sum_{X_i \in V_H(I)} X_i, \tag{4.13}$$

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

If valid ridge data remains insufficiently after eliminating invalid data, we use the raw-depth-data-based method to track the head center as

$$\begin{aligned} C'_H(I) &= \left\{ X_i \in I \;\middle|\; D_E(X_i, \widetilde{J}_H^t) \leq r_H \right\}, \\ V'_H(I) &= \left\{ X_i \in C'_H \;\middle|\; \left| D_E(X_i, \widetilde{J}_T^t) - l_{H,T} \right| \leq \epsilon_H \right\}, \\ J_H^t &= \frac{1}{|V'_H(I)|} \sum_{X_i \in V'_H(I)} X_i, \end{aligned} \tag{4.14}$$

where $C'_H(I)$ is the candidates for the head center from the raw depth data and $V'_H(I)$ is the valid raw depth data for the head center.

4.2.2 Torso Estimation

The human torso consists of the left and right shoulders J_{LS}, J_{RS}, the torso center J_T, and the left and right pelvises J_{LP}, J_{RP}. We use only raw depth data to estimate these torso joints because the human torso can be easily occluded by another body part such as arms and legs. To estimate the position of occluded torso joints, we apply the HST structure (Sec. 4.1.1) as follows.

(1) We predict the torso joint positions in the current frame from those in the previous position as

$$\widetilde{J}_k^t \leftarrow J_k^{t-1} + \Delta J_k^{t-1}, \tag{4.15}$$

where $\widetilde{J}_j^t$ is the predicted torso joint position $k \in \{LS, RS, T, LP, RP\}$ in the tth frame and $\Delta J_k^{t-1} = J_k^{t-1} - J_k^{t-2}$ is the increment of the velocity model for the kth torso joint position.

(2) We collect the candidates for the torso joints from the raw depth data image I as

$$C_k(I) = \left\{ X_i \in I \;\middle|\; D_E(X_i, \widetilde{J}_j^t) \leq r_k \right\}. \tag{4.16}$$

(3) We select the valid torso joints using different sets of torso joint constraints to eliminate candidates. The valid data for the shoulders is collected as

$$V_s(I) = \left\{ X_i \in C_s(I) \;\middle|\; \begin{array}{c} \left| \dfrac{\left\| \overrightarrow{J_{s'}X_i} \times \overrightarrow{J_H J_{s'}} \right\|}{\left\| \overrightarrow{J_{s'}X_i} \right\|} - l_{H\perp S} \right| \leq \epsilon_s, \\ |D_E(J_{s'}, X_i) - l_{LS,RS}| \leq \epsilon_s \end{array} \right\}, \tag{4.17}$$

where $C_s(I)$ is the set of candidates of shoulder joint $s = \{LS, RS\}$, $\overrightarrow{J_{s'}X_i} = X_i - J_{s'}$ is a vector from the opposite shoulder $s' = \{RS, LS\}$ to the candidate data X_i, $\overrightarrow{J_H J_{s'}} = J_{s'} - J_H$ 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 (Eq. (4.2)).

The valid data for the torso center are collected as

$$V_T(I) = \left\{ X_i \in C_T(I) \;\middle|\; \begin{array}{l} |D_E(J_H, X_i) - l_{H,T}| \leq \epsilon_T, \\ |D_E(J_{LS}, X_i) - l_{LS,T}| \leq \epsilon_T, \\ |D_E(J_{RS}, X_i) - l_{RS,T}| \leq \epsilon_T \end{array} \right\}, \tag{4.18}$$

where $C_T(I)$ is the set of candidates of torso center and $l_{H,T}$, $l_{LS,T}$, and $l_{RS,T}$ are the length parameters in the HST structure (Eq. (4.2)).

The valid set for the pelvis is collected as

$$V_p(I) = \left\{ X_i \in C_p(I) \;\middle|\; \begin{array}{c} \left| \dfrac{\left\| \overrightarrow{J_{p'}X_i} \times \overrightarrow{J_T J_{p'}} \right\|}{\left\| \overrightarrow{J_{p'}X_i} \right\|} - l_{T\perp P} \right| \le \epsilon_p, \\ |D_E(J_{LS}, X_i) - l_{LS,p}| \le \epsilon_p, \\ |D_E(J_{RS}, X_i) - l_{RS,p}| \le \epsilon_p \end{array} \right\}, \tag{4.19}$$

where $C_p(I)$ is the set of candidates of the pelvis joint $p \in \{LP, RP\}$, $\overrightarrow{J_{p'}X_i} = X_i - J_{p'}$ is a vector from the opposite pelvis joint $p' = \{RP, LP\}$ to the candidate data X_i, $\overrightarrow{J_T J_{p'}} = J_{p'} - J_T$ 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 (Eq. (4.2)).

(4) When valid ridge data remain sufficiently after eliminating the invalid data, we aggregate the collected valid data into the estimated torso joint position as

$$J_k^t = \frac{1}{|V_k(I)|} \sum_{X_i \in V_k(I)} X_i, \tag{4.20}$$

where $|V_k(I)|$ is the number of valid torso joint candidates.

When other body parts occlude some torso joints, the number of valid torso joint candidates may be too small to estimate the torso joints. To overcome this situation, we propose a method that exploits the assumption that all torso joints lie in one plane. The proposed method consists of two steps: torso rotation estimation and tracking of the occluded joints.

(1) We estimate the torso rotation matrix R_T that minimizes the difference between the valid joint positions in the current frame and their corresponding positions in the previous frame:

$$\sum_{k=1}^{N} \left(R_T^t \cdot \left(J_k^{t-1} - J_H^{t-1} \right) - \left(J_k^t - J_H^t \right) \right)^2, \tag{4.21}$$

where N is the number of valid torso joint candidates , J_k^t and J_k^{t-1} are the valid torso joint positions in the tth frame and the corresponding valid torso joint positions in the $t-1$th frame, respectively. The torso rotation matrix can be easily obtained by the least square method.

(2) We track the positions of the occluded joints as

$$J_o^t = R_T^t \cdot \left(J_o^{t-1} - J_H^{t-1}\right) \cdot J_H^t, \tag{4.22}$$

where J_o^t are the positions of the occluded joints in the tth frame and J_o^{t-1} are the positions of the corresponding joints in the $t-1$th frame. Because we use the planarity assumption and at least two torso joints are available, this method can estimate the positions of the occluded torso joints.

4.2.3 Limb Estimation

The positions of limb joints, such as elbows, hands, knees, and feet are tracked using either the ridge data or raw depth data. The procedure of the ridge-data-based method involves four steps. To track the positions of occluded joints, we adopt an additional constraint: the degree of straightness.

(1) We use forward kinematics to predict the positions of limb joints from the previous positions as

$$\widetilde{J}_k^t = R_k^{t-1} \cdot (J_k^{t-1} - J_l^{t-1}) + J_l^t, \tag{4.23}$$

where $k \in \{LE, RE, LH, RH, LK, RK, LF, RF\}$ is the index of the limb joint, $l \in \{LS, RS, LE, RE, LP, RP, LK, RK\}$ is the index of the parent limb joint of the kth limb joint, and R_k^{t-1} is the rotation matrix that is defined by the rotation axis $\overrightarrow{J_l^{t-1}J_k^{t-1}} \times \overrightarrow{J_l^{t-2}J_k^{t-2}}$ and the rotation angle $sin^{-1}(\overrightarrow{J_l^{t-1}J_k^{t-1}} \cdot \overrightarrow{J_l^{t-2}J_k^{t-2}})$.

(2) Collect the candidates for the limb joints as

$$C_k(I) = \left\{ X_i \in R(I) \;\middle|\; D_E(X_i, J_i^t) \leq r_k \right\}, \tag{4.24}$$

where the search area is set to have a radius of 0.2 m to cover the fast movement of the limb joints.

(3) We eliminate the invalid data for the limb joints as

$$V_k(I) = \left\{ X_i \in C_k(I) \;\middle|\; \begin{array}{c} \left| D_E(X_i, J_l^t) - l_k \right| \leq \epsilon_k, \\ P_L\left(X_i, J_l^t\right) > \epsilon_{Line} \end{array} \right\}, \tag{4.25}$$

where l_k are the length constraints in Eq. (4.4) and Eq. (4.5), ϵ_L is a straightness threshold that controls the tolerance of straight lines. The degree of straightness $P_L(X_i, J_l^t)$ is defined by the ratio of the Euclidean distance over the geodesic distance between X_i and J_l^t as

$$P_L\left(X_i, J_l^t\right) = \frac{D_E(X_i, J_l^t)}{D_G(X_i, J_l^t)}, \tag{4.26}$$

where J_l^t is the position of the parent joint of the kth joint at the tth frame.

We consider the degree of straightness to avoid data ambiguity when arms are bent. For example, we eliminate invalid ridge data around the elbow by using the length constraint between the shoulder and the elbow. When we bend the arm, the lower arm's ridge data can be identified as part of the valid elbow's ridge data because two ridge data are close together due to the arm bending. We must remove the lower arm's ridge data from the valid elbow's ridge data when we apply the length constraint between the shoulder and the elbow. The degree of straightness separates the lower arm effectively from the elbow because their values of the lower arm and the upper arm have strong discrimination in that the upper arm has large value ≈ 1 (Fig. 4.5(a)) but the lower arm has small value < 1 (Fig. 4.5(b)).

(4) Finally, we aggregate the valid ridge data to estimate the positions of limb joints as

$$J_k = \frac{1}{|V_k(I)|} \sum_{X_i \in V_k} X_i, \tag{4.27}$$

when the number of valid ridge data $|V_k(I)|$ is sufficiently large.

(a) (b)

Figure 4.5. Examples of the degree of straightness; (a) strong straightness and (b) weak straightness, green line: Euclidean distance; and red line: geodesic distance.

If valid ridge data remains insufficiently after eliminating invalid data, we perform the same procedure with the raw depth data as

$$
\begin{aligned}
C'_k &= \left\{ X_i \in I \;\middle|\; D_E(X_i, \widetilde{J}^t_k) \leq r_k \right\}, \\
V'_k &= \left\{ X_i \in C'_k \;\middle|\; \begin{array}{c} \left| D_E(X_i, J^t_l) - l_k \right| \leq \epsilon_k, \\ P\left(X_i, J^t_l\right) > \epsilon_{Line} \end{array} \right\}, \\
J^t_k &= \frac{1}{\left|V'_k(I)\right|} \sum_{X_i \in V'_k(I)} X_i.
\end{aligned}
\tag{4.28}
$$

The parameter r_j (Eq. 4.10, 4.14, 4.16, 4.24, 4.28) 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. The parameter ϵ_j (Eq. 4.12, 4.14, 4.17, 4.18, 4.19, 4.25, 4.28) controls the tolerance of difference between the estimated length and the length constraint of the human model. We experimentally chose these parameters and Table 4.1 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

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

2 $C_j = \left\{ X_i \in R(I) \middle| D_E(X_i, \widetilde{J}_j) \leq r_j \right\}$; /* Candidate Collection */

3 **switch** j **do**

4 **case** H **do** /* Invalid-Data Pruning of Head */

5 $V_j = \left\{ X_i \in C_j \;\middle|\; \left| D_E(X_i, \widetilde{J}_T) - l_{H,T} \right| \leq \epsilon_j \right\}$

6 **case** LS *or* RS **do** /* Invalid-Data Pruning of Shoulders */

7 $V_j = \left\{ X_i \in C_j \;\middle|\; \begin{array}{c} \left| D_E(J_{j'}, X_i) - l_{LS,RS} \right| \leq \epsilon_j \\ \left| \frac{\| \overrightarrow{J_{j'} X_i} \times \overrightarrow{J_H J_{j'}} \|}{\| \overrightarrow{J_{j'} X_i} \|} - l_{H \perp S} \right| \leq \epsilon_j \end{array} \right\}$

8 **case** T **do** /* Invalid-Data Pruning of Torso Center */

9 $V_j = \left\{ X_i \in C_j \;\middle|\; \begin{array}{c} |D_E(J_H, X_i) - l_{H,T}| \leq \epsilon_j \\ |D_E(J_{LS}, X_i) - l_{LS,T}| \leq \epsilon_j \\ |D_E(J_{RS}, X_i) - l_{RS,T}| \leq \epsilon_j \end{array} \right\}$

10 **case** LP *or* RP **do** /* Invalid-Data Pruning of Hips */

11 $V_j = \left\{ X_i \in C_j \;\middle|\; \begin{array}{c} |D_E(J_{LS}, X_i) - l_{LS,j}| \leq \epsilon_j \\ |D_E(J_{RS}, X_i) - l_{RS,j}| \leq \epsilon_j \\ \left| \frac{\left\| \overrightarrow{J_{j'} X_i} \times \overrightarrow{J_T J_{j'}} \right\|}{\left\| \overrightarrow{J_{j'} X_i} \right\|} - l_{T \perp P} \right| \leq \epsilon_j \end{array} \right\}$

12 **otherwise do** /* Invalid-Data Pruning of Limb */

13 $V_j = \left\{ X_i \in C_j \;\middle|\; \begin{array}{c} |D_E(X_i, J_p) - l_j| \leq \epsilon_j \\ P_L(X_i, J_p) > \epsilon_L \end{array} \right\}$

14 **end**

15 **end**

16 **if** $|V_j(I)|$ *is sufficient* **then**

17 $J_j = \frac{1}{|V_j(I)|} \sum_{X_i \in V_j(I)} X_i$; /* Joint Estimation using $R(I)$ */

18 **else**

19 $C_j = \left\{ X_i \in I \middle| D_E(X_i, \widetilde{J}_j) \leq r_j \right\}$

20 repeat from line 3; /* Joint Estimation using I */

21 **end**

Table 4.1. 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	
	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
45	0.910	**227.2**	0.934	220.7	0.934	198.3	0.930	180.7
50	0.906	220.8	0.931	221.5	**0.936**	199.6	0.935	182.3
55	0.904	221.6	0.930	218.9	0.935	200.0	0.935	182.0
60	0.900	220.5	0.927	218.7	0.935	199.5	0.935	181.8

CHAPTER 5

Deep Learning-Based Human Pose Estimation

In this chapter, we consider two CNN-based human pose estimation methods: shallow and multi-channel CNN-based human pose estimation. The shallow CNN-based method directly regresses the 3D positions of human joints with an additional input channel of the ridge image. To further utilize the characteristics of the depth image, the multi-channel CNN-based method projects the depth points and ridge data onto three orthogonal planes, it generates 2D heatmaps to estimate the keypoints on each plane, and the estimated keypoints are concatenated and fed to two fully-connected layers to regress the 3D human pose.

5.1 Shallow CNN-based Human Pose Estimation

Motivated by the Li *et al.* [19] that used CNN for 3D pose estimation from a monocular color image, we propose the shallow CNN (Fig. 5.1) that is similar to Li *et al.* [19]. Alternatively, we take two input channels of the depth and ridge image instead of three RGB channels, where the ridge image represents

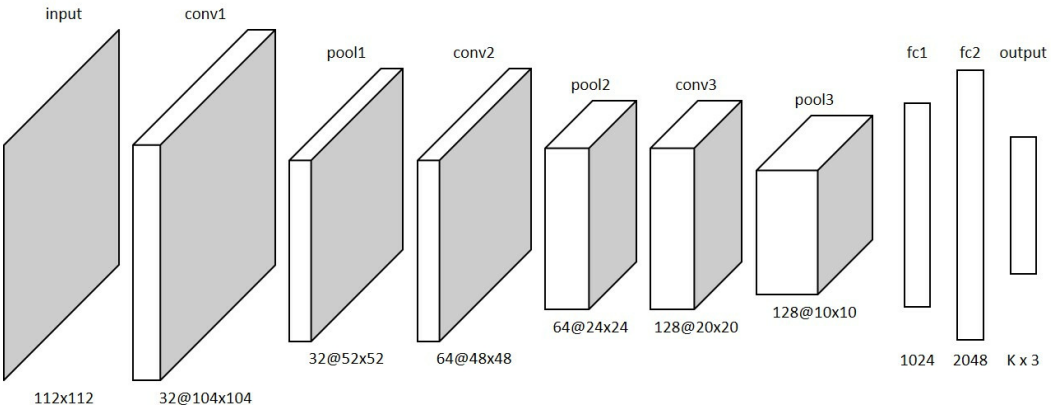


Figure 5.1. Architecture of the shallow CNN.

a probability map of ridge data. Each proposed CNN consists of nine trainable layers; three convolutional layers, three pooling layer, and three fully connected layers. We use the rectified linear units (ReLu) for conv1, conv2, and the first two fully connected layers and use tanh as the activation function for the last fully connected layer.

The direct mapping from the input image to the 3D positions of human joints has high non-linearity and complicated learning. To reduce the complexity, we use three different types of loss function; distance loss, length loss, and rotation loss. The most commonly used distance loss function L_d is defined as the sum L2-distance of joint position errors in world coordinate,

$$L_d = \sum_i \left\| J_i - \hat{J}_i \right\|^2, \tag{5.1}$$

where $J_i = (x_i, y_i, z_i)$ is the 3D position of the ith joint and $\hat{J}_i$ is its ground truth.

Li *et al.* proposed a relative loss function L_p that is the sum of L2-distance of relative joint position errors as

$$L_p = \sum_i \left\| P_i - \hat{P}_i \right\|^2, \tag{5.2}$$

where $P_i = J_i - J_{p(i)}$ is the relative position of the ith joint with respect to its parent joint $p(i)$. This loss is better for measuring the difference of the predicted position relative to its parent. However, they have a strong assumption that the root of the joint is always at the origin, which limits their loss to handle absolute 3D joint locations. Thus, the relative loss itself may not be good enough to train the CNN.

The hierarchical information of a human pose can be denoted as the limb lengths and their rotations relative to its parent. Intuitively, the wellness of prediction for the relative joint conditions depends on the loss of the hierarchy information. We define the length loss as the sum of L2-distances of the limb length prediction (Eq. 5.3)

$$L_s = \sum_i \left(\|P_i\| - \|\hat{P}_i\| \right)^2 \tag{5.3}$$

To overcome the problem that has the root of joint as the origin of the world coordinate, we use the torso center as the root of joint and define it's parent as the centroid of human depth silhouette. Therefore, the problem space can be moved from the world coordinate to the personal depth coordinate.

For rotation loss, we use a cosine distance metric as shown Eq. (5.4).

$$L_r = \sum_i \left(1 - \frac{P_i \cdot \hat{P}_i}{\|P_i\| \cdot \|\hat{P}_i\|} \right) \tag{5.4}$$

We combine these loss functions of joint locations and relative joint conditions by taking the linear combination of L_d, L_s and L_r as

$$L_p = L_d + \alpha L_s + \beta L_r, \tag{5.5}$$

where α and β are the hyper-parameters and we use 30 and 5 in our experiment, respectively.

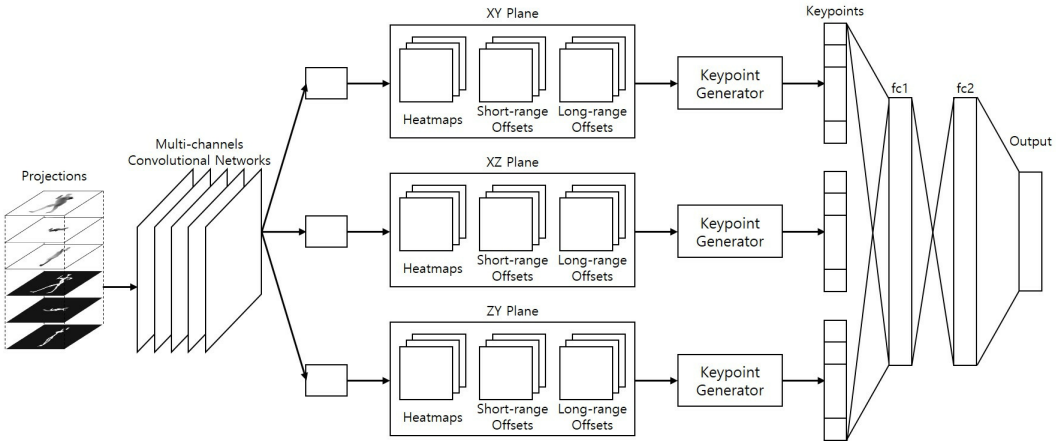


Figure 5.2. Overall process of multi-channel CNN.

5.2 Multi-Channel CNN-Based Human Pose Estimation

Since the direct mapping from the input image to the 3D position of joints is highly non-linear and complicated learning, the recent trend of human pose estimation tries to map the input image to a set of heatmaps which represent the probability distributions of joint positions. However, the heatmap only provides 2D information of the joint position [38, 39] and the few depth information is utilized.

5.2.1 Depth Points and Ridge Data Projection

Inspired by Ge *et al.* [40], we consider the multi-channel CNN (Fig. 5.2) that use the projected images. It fully utilizes the 3D information of depth points by projecting them onto three orthogonal planes. As the ridge data is defined as a set of 3D local maximal points in the distance transform map, we can still project the ridge data as the depth points.

To generate the projected images, we first transform the depth image as a

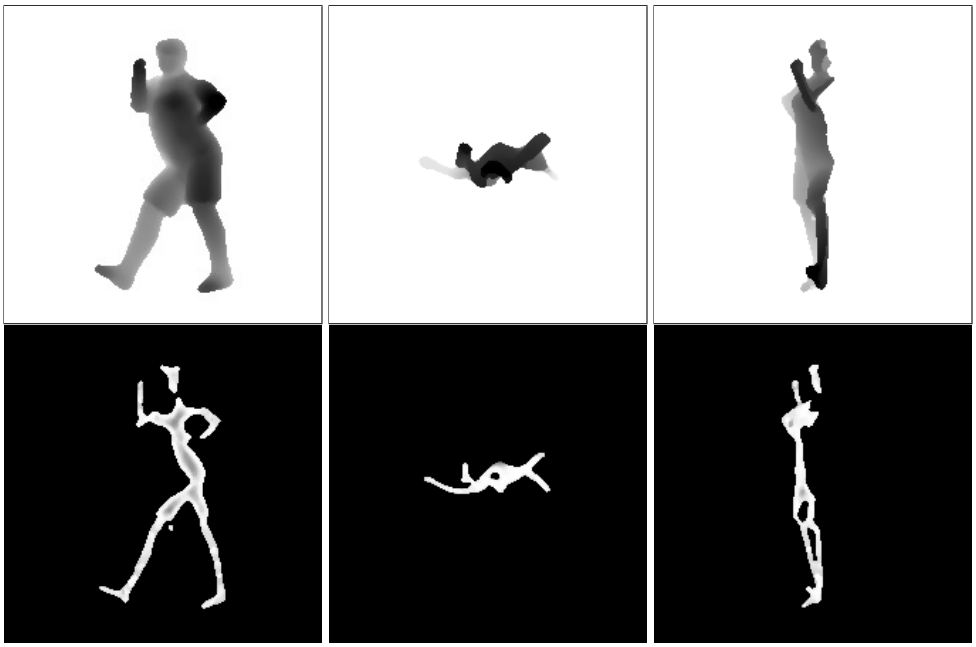


Figure 5.3. The upper and lower row represent the projected images of depth points and ridge data. From left to right, projection planes are XY, XZ, and ZY, respectively.

set of depth points using the camera's intrinsic parameter. We project the depth points and ridge data onto three orthogonal planes, XY, XZ, and ZY.

The depth points project onto a plane, and the distances from depth points to the projection plane are normalized between 0 and 1 (with the nearest points set to 0, the farthest points set to 1), and the normalized distances are stored as the pixel values of the projected images. The ridge data project on to a plan and each pixel value is set to the corresponding probability of ridge data. If multiple depth points or ridge data are projected onto the same pixel in their projected images, the smallest normalized distance or largest probability will be stored as the pixel value. Fig. 5.3 shows the examples of the projected images of depth points and ridge data.

Inspired by the work of Chen *et al.* [38], we employ the feature pyramid structure based on the ResNet [41] backbone. The projected input images are resized as 224×224 and then stacked as one input image, which enables us to design a single network for the heatmap generation instead of using multiple networks for three projected images. The output feature maps of the last residual blocks of different convolution features have the size of 112×112, 56×56, 28×28, and 14×14 and are denoted as C_2, C_3, C_4, and C_5, respectively. The 3×3 convolution filters are applied on the C_2, C_3, C_4, and C_5 to generate the 45 (3 projections $\times 15$ joints) heatmaps.

In this work, we generate the disk-shaped probability of the heatmap instead of Gaussian shape. For each position x and each joint k, we compute the probability $p_k(x) = 1$ if $\|x - l_k\| \leq R$ that the position x is within a disk of the radius R from the joint position l_k of the kth joint.

5.2.2 3D Joint Prediction

In addition to generating the heatmap for each joint, the multi-channel CNN also generates two-types of offsets vectors: short and long-range offset. At each position x within the joint disks and for each joint type k, the short-range 2D offset vector $S_k(x) = l_k - x$ points from the image position x to the kth joint. By dividing the regression problem into the classification problem and regression problem, we can predict the more accurate position of each joint.

Although the joint probability of the heatmap and the short-range offset can improve the accuracy, their receptive field tends to be very localized. To overcome the problem, we propose to generate the long-range offset vector that $L_k(x) = l_k - x$ points from the image position x to the kth joint at each position x within the human silhouette and for each joint type k. By predicting the joint positions from the long-range offset, we can predict the joint position even if the target joint is occluded.

To make the multi-channel CNN end-to-end trainable, we add the non-trainable keypoint generator module that generates a set of 2D keypoint vectors for a specific plane. For each plane, we generate two types of keypoints: short- and long-range keypoints, where they are generated by the bilinear interpolation kernel of the short-range and long-range offset, respectively. The generated 2D keypoint vectors are concatenated as the 1D vector, then fed to the fully-connected layers, where they regress the 3D positions from three 2D positions.

CHAPTER 6

Dance Performance Evaluation

To apply the proposed human pose estimation to a practical field, we developed a K-Pop dance teacher program (Fig. 6.1) that can help people to learn dances from examples in six steps. (1) The program shows the K-Pop dance performed by a professional, and the learner tries to follow the K-Pop dance as closely as possible. (2) The human pose estimator extracts the joint positions of the learner. (3) The dance feature generator for the learner makes the dance features from the extracted joint positions. (4) The dance feature generator for the dance teacher also makes the dance features from the joint positions in the teacher's dance dataset. (5) The program performs dynamic time warping of the learner's dance features to the teacher's dance features. (6) The program evaluates the learner's dance performance by comparing the dance features of the learner and teacher.

The learner can select one K-Pop dance among 100 popular K-Pop dances, which are listed in a menu window (Fig. 6.2, left). To show the dance performed by a dance expert, we use standard computer graphics techniques to render the

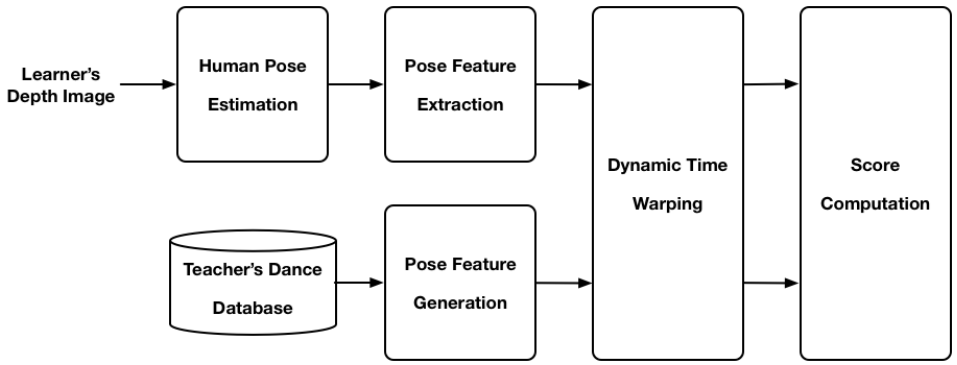


Figure 6.1. Overall process of the K-pop dance teacher.

realistic virtual teacher (Fig. 6.2, 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. 6.2, lower right).

6.1 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 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 [27] 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.

6.1.1 Torso Feature

In [27], 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

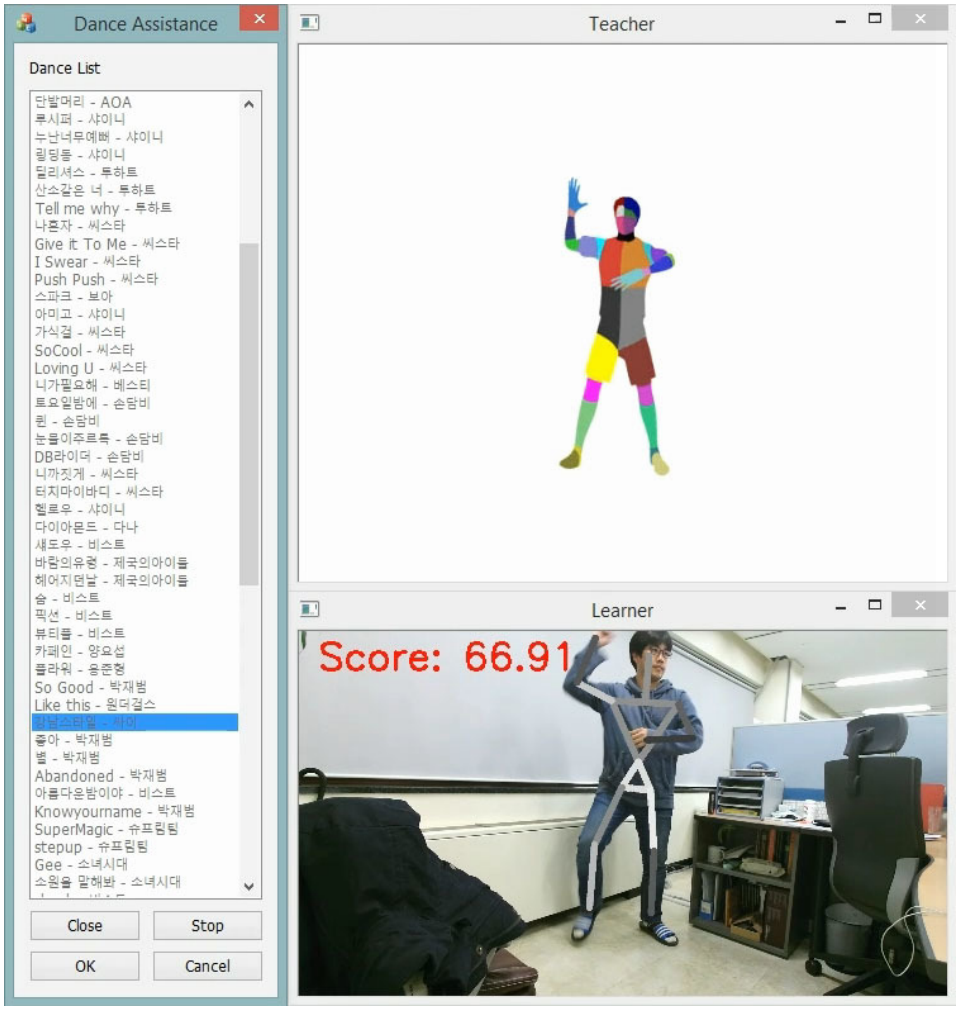


Figure 6.2. Screen shot of the K-pop dance teacher program. Left window: list of 100 K-Pop dance sequences; upper-right window: dance teacher; lower-right window: learner with an instant dance similarity score and pose similarity of each body part represented by a intensity value (bright intensity = high similarity).

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 includes 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 calculating 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 the torso and full-body rotation.

6.1.2 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 [27]. 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 [27].

6.2 Dance Similarity

The time accuracy of each subsequence is measured by the timing similarity of the learner's and teacher's dance sequences 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

$$S_t = 1 - \min\left(1, \exp\left(\frac{|T_{teacher} - T_{learner}| - \tau}{\alpha\tau}\right)\right), \tag{6.1}$$

where $T_{teacher} = \frac{T^s_{teacher}+T^e_{teacher}}{2}$ is the middle time of the teacher's dance sequence, $T_{learner} = \frac{T^s_{learner}+T^e_{learner}}{2}$ is the middle time of the learner's dance sequence, and α is a tolerance parameter that governs the slope of Eq. (6.1), where the superscripts s and e denote the start and end point of dance sequence, respectively.

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

$$S_p = \frac{1}{T^e_{teacher} - T^s_{teacher} + 1} \sum_{i=T^s_{teacher}}^{T^e_{teacher}} \exp\left(-\frac{\left\|\mathbf{f}^i_{learner} - \mathbf{f}^i_{teacher}\right\|}{\beta}\right), \tag{6.2}$$

where vectors $\mathbf{f}^i_{learner}$ and $\mathbf{f}^i_{teacher}$ denote the dance features of the learner and teacher at the ith frame, respectively, 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 time and pose accuracies as

$$S = \frac{1}{N}\sum_{j=1}^{N} S^j_t S^j_p, \tag{6.3}$$

where N is the number of subsequences of the learner's dance sequence and S^j_t and S^j_p denote the timing accuracy and the posture accuracy at the jth subsequence, respectively.

Algorithm 3: Dance Performance Evaluation

Input: Learner's dance feature $\mathbf{f}^i_{learner}, i = [1, n]$, teacher's dance feature $\mathbf{f}^j_{teacher}, j = [1, m]$

Parameters: Timing tolerant value τ, slope control parameters α, β

Output: Timing accuracy $S_{teacher}$, 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**

    $DTW[i, j] = \|\mathbf{f}^i_{learner} - \mathbf{f}^j_{teacher}\| + \min(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1])$;

  end

end

/* Timing accuracy */

$T^s_{teacher} = \arg\min_j DTW[1, j]$;

$T^e_{teacher} = \arg\min_j DTW[n+1, j]$;

$T_{teacher} = (T^e_{teacher} - T^s_{teacher} + 1)/2$;

$S_{teacher} = 1 - \min\left(1, \exp\left(\frac{(|T_{teacher} - n/2| - \tau)}{\alpha\tau}\right)\right)$;

/* Pose accuracy */

$S_p = \exp\left(-\frac{\min_j DTW[n+1, j]}{\beta T_{teacher}}\right)$;

CHAPTER 7

Experimental Results and Discussion

7.1 Datasets and Evaluation Protocol

We evaluated the human pose estimation methods using the benchmark dataset SMMC-10 [15] and EVAL [42]. The SMMC-10 was recorded using a Mesa SwissRanger time-of-flight camera at a speed of 25 fps and a resolution of 176 $\times$ 144 pixels, and consists of 28 real-world depth image sequences, each of which had a variety of sizes and complexities from 100 frames to 400 frames and from single-limb motions to fast kicks, swings, self-occlusions, and full-body rotations. The EVAL was recorded using a Microsoft Kinect depth camera at a speed of 30 fps and a resolution of 320 $\times$ 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 and from simple punch motions to handstands, kicks, and sitting down on floor. The true position of each joint was labeled using a commercial markers-based motion capture system.

We compared the proposed human pose estimation method quantitatively with state-of-the-art methods such as Shotton *et al.* [5], Baak *et al.* [35], Gana-

pathi *et al.* [42], Demirdjian *et al.* [43], Girshick *et al.* [6], Ye and Yang [44] and Jung *et al.* [45]. 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. In subsequent subsections, the experimental results are described using the following evaluation protocols.

For fair comparison, we used two measurements: mean average precision (mAP) and average pose error. 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 $\delta = 0.1m$ of the true joint position. We define average pose error as

$$\epsilon_{avg} = \frac{1}{\sum_i^N K_i} \sum_{i=1}^{N} \sum_{k=1}^{K_i} \|J_k^i - \tilde{J}_k^i\|, \tag{7.1}$$

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 kth joint in ith frame, and $\tilde{J}_k^i$ is the corresponding 3D position of the estimated kth joint in the ith frame.

7.2 Comparison of Ridge Data and Medial Axis

To validate the effectiveness of the proposed ridge data, we compare the pose estimation accuracy of the feature-based human pose estimation method using the proposed ridge data with that of the medial axis transform (MAT) [31] and the dilated medial axis transform (DMAT) on the SMMC-10 and EVAL dataset (see Table 7.1 and 7.2).

Since the MAT feature shows weak representation of the proposed ridge data, its accuracy is similar to that of pose estimation without the ridge data. The

Table 7.1. Comparison of pose estimation accuracy on SMMC-10

Feature	Head	Neck	L. Shoulder	R. Shoulder	L. Elbow	R. Elbow	L. Hand	R. Hand	mAP
No Ridge	0.9870	0.9811	0.9991	0.9996	0.5579	0.7493	0.5507	0.7746	0.8517
MAT	0.9818	0.9809	0.9996	0.9989	0.5531	0.7409	0.5437	0.7758	0.8496
DMAT	0.9843	0.9792	0.9933	0.9953	0.7398	0.8551	0.7341	0.8578	0.9114
Ridge	0.9963	0.9770	0.9881	0.9925	0.9280	0.9642	0.9248	0.9577	0.9735

Table 7.2. Comparison of pose estimation accuracy on EVAL

Feature	Head	Chest	L. Shoulder	R. Shoulder	L. Elbow	R. Elbow	L. Hand	R. Hand	mAP
No Ridge	0.9661	0.9838	0.9414	0.9470	0.5003	0.7000	0.4927	0.7225	0.8091
MAT	0.9658	0.9834	0.9416	0.9472	0.5001	0.7062	0.4898	0.7286	0.8107
DMAT	0.9822	0.9858	0.9380	0.9449	0.6858	0.8024	0.6813	0.8179	0.8740
Ridge	0.9787	0.9874	0.9325	0.9433	0.8779	0.9176	0.8705	0.9090	0.9358

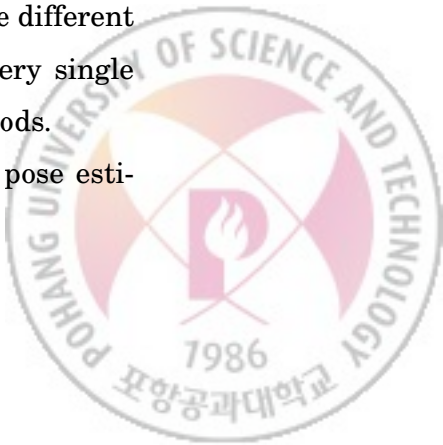
pose estimation using the DMAT feature, which is plentiful representation of the MAT feature, shows higher accuracy than the pose estimation without the ridge data but lower than the pose estimation with the ridge data. There is a big difference in the pose estimation accuracy of elbows and hands between DMAT feature and ridge data, which implies that that the ridge data is more effective to localize the peripheral joint positions.

7.3 Feature-Based Human Pose Estimation

7.3.1 Pose Estimation Accuracy

We compared the pose estimation accuracy of the proposed method with those of the state-of-the-art methods such as Shotton *et al.* [5], Ganapathi *et al.* [42], Demirdjian *et al.* [43], Girshick *et al.* [6], Ye and Yang [44], on SMMC-10 [15] dataset and Ye and Yang [44] and Jung *et al.* [45] on EVAL dataset [42], where Shotton *et al.* [5], Girshick *et al.* [6], and Jung *et al.* [45] methods were different from other remaining methods in that they estimated poses in every single frame but other methods estimated poses using tracking based methods.

The proposed human pose estimation method achieved the best pose esti-



mation accuracies 0.9735 mAP and 0.9358 mAP on the SMMC-10 dataset (see Fig. 7.1(a)) and the EVAL dataset (see Fig. 7.1(b)), respectively. The SMMC-10 dataset contains relatively easy poses such as punch, kicking, bending, and so on, and the EVAL dataset contains relatively complicated poses such as crossed punch, hand standing, sitting down on the floor, and so on.

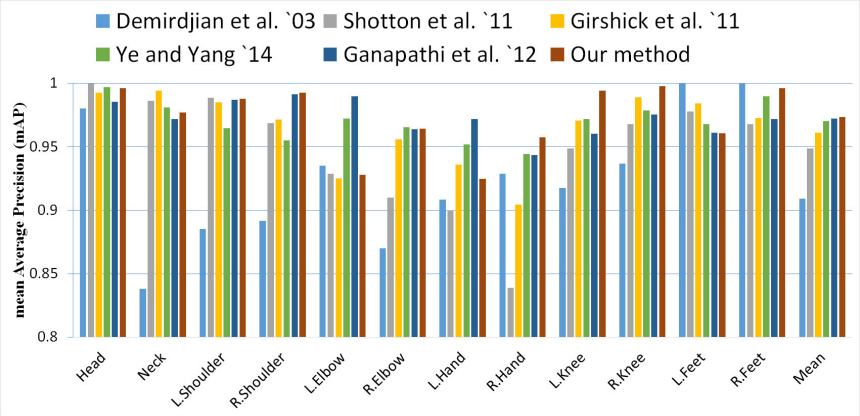
The proposed human pose estimation method shows the higher mAP than other methods because we use the ridge data instead of raw depth data to prune the invalid data. This improvement comes from the fact that (1) ridge data remains within the body parts, but the raw depth data drifts over the overlapped body parts and (2) the valid data that meets the requirement for data pruning comes from the wrong data points in the case of using raw depth data.

Specifically, the proposed human pose estimation method tracks the human joints robustly when the human body rotates. When the human body rotates, the head, chest, and shoulders are prune to be self-occluded by other body parts, which reduces the accuracy of the human pose estimation. The proposed HST structure successfully handles the full body rotation (see the 24th sequence of Fig. 7.2) in that the mAPs of the left and right shoulders are higher than those of other methods on the SMMC-10 dataset (see Fig. 7.1(a)).

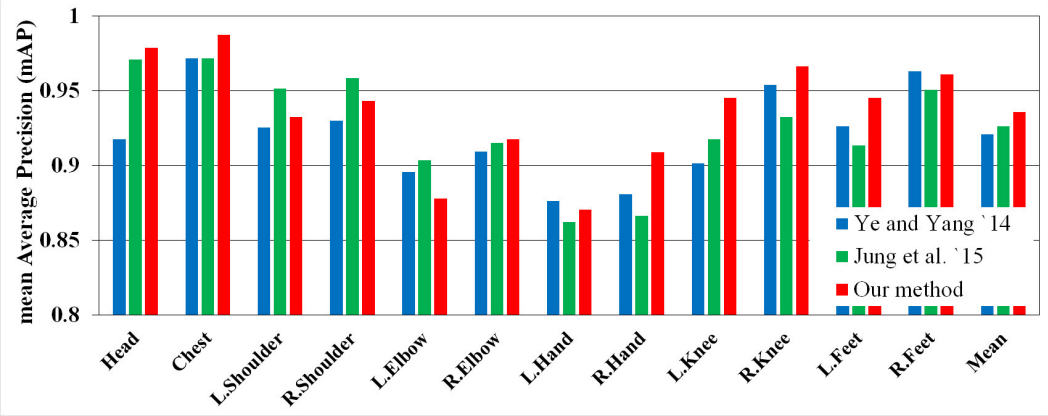
7.3.2 Pose Estimation Error

We compared the pose estimation error of the proposed method with those of the state-of-the-art methods (Ganapathi *et al.* [15] and Baak *et al.* [35]), because the existing methods for body part detection did not report pose estimation error.

The proposed human pose estimation method achieved the lowest pose errors in most of the benchmark sequences (see Fig. 7.2). In some cases such as sequence 5 and 7, other methods reported lower mean pose errors than ours. However, our method shows smaller standard deviations. This result means that estimates obtained using our method are more consistent than the state-



(a)



(b)

Figure 7.1. Comparison of mean average precision (mAP) using (a) the SMMC-10 dataset and (b) the EVAL dataset.

of-the-art methods.

Especially, the proposed method achieved the lowest pose errors in sequences 24 to 27. These sequences contain very complex motions such as fast kicks and swings, self-occlusions, and full-body rotations. This result shows that the ridge data and HST structure capture the fast movements and full-body rotations effectively.

The proposed method using the ridge data achieved the smallest average pose error (3.88 cm) and reduced the average pose error by 1.11 cm compared to the proposed method that did not use ridge data (4.99 cm). This result shows that the proposed method with the ridge data is more effective to localize the joint positions inside the human body.

7.3.3 Computation Speed

We compared the computation speed of the proposed method with those of state-of-the-art methods (Shotton *et al.* [5], Baak *et al.* [35], Ganapathi *et al.* [42], Girshick *et al.* [6], and Jung *et al.* [45]). Note that Shotton *et al.* [5], Girshick *et al.* [6], and Jung *et al.* [45] are body part detectors and we use SMMC-10 dataset to compare the computation speed, since the computation time on EVAL dataset has not been reported.

The proposed method ran at 290 frames per second (fps) on a single core (Intel i7) and was faster than all but the method of Jung *et al.*, which is a body part detector (Table 7.3). Our method ran faster when it did not use the ridge data, because the result considers the entire process including human detection, feature extraction, and human joint tracking. When we consider only the human joint tracking step, the computation time with the ridge data is 1.4 ms and the time without the ridge data is 2.0 ms. This result shows that the ridge data reduces the search area effectively for tracking human joint.

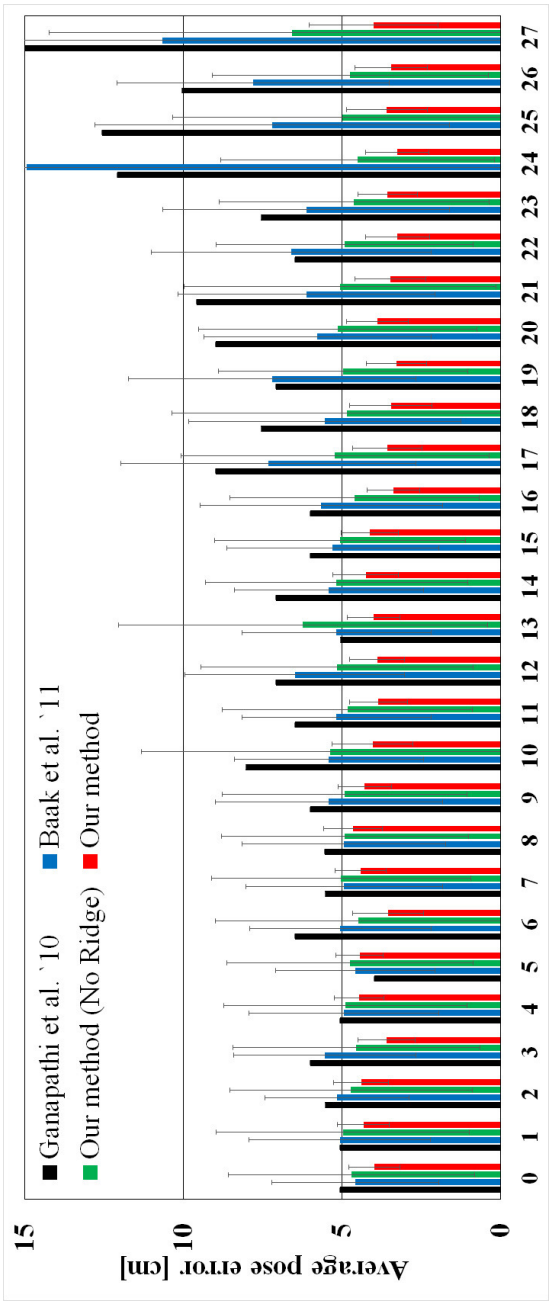


Figure 7.2 Comparison of average pose error and standard deviation using the SMMC-10 dataset.

Table 7.3. Comparison of computation speed on the SMMC-10 dataset.

Method	Computation speed (fps)
Shotton *et al.* [5]	50 (8-core)
Baak *et al.* [35]	60
Ganapathi *et al.* [15]	4
Girshick *et al.* [6]	200
Jung *et al.* [45]	1000
Our method (No ridge)	327.1
Our method	289.9

7.4 Deep Learning-Based Human Pose Estimation

We compared the pose estimation accuracy of the proposed methods with Huang and Altamar [18] on the EVAL dataset [42], which is the only deep learning-based method. The proposed human pose estimation method achieved the best pose estimation accuracy 0.9801 mAP on the EVAL dataset (see Fig. 7.3). In Fig. 7.3, *Depth* and *Depth + Ridge* represent the pose estimation accuracies of the shallow CNN-based method whose input is only depth image and both depth and ridge images, respectively. The *3, 4*, and *6* channels represent the pose estimation accuracies of the multi-Channel CNN with three projected depth images, three projected depth images + one projected ridge image, and three projected depth images + three projected ridge image, respectively.

As shown in Fig. 7.3, the proposed ridge data improves the mean average precision in both deep learning methods. Mainly, we can observe the significant improvement on each peripheral, such as elbows and hands. Fig. 7.4 shows some representative results of the proposed method using six channels and multi-channel CNN estimates successively in case of the full-body rotation and the occlusion.

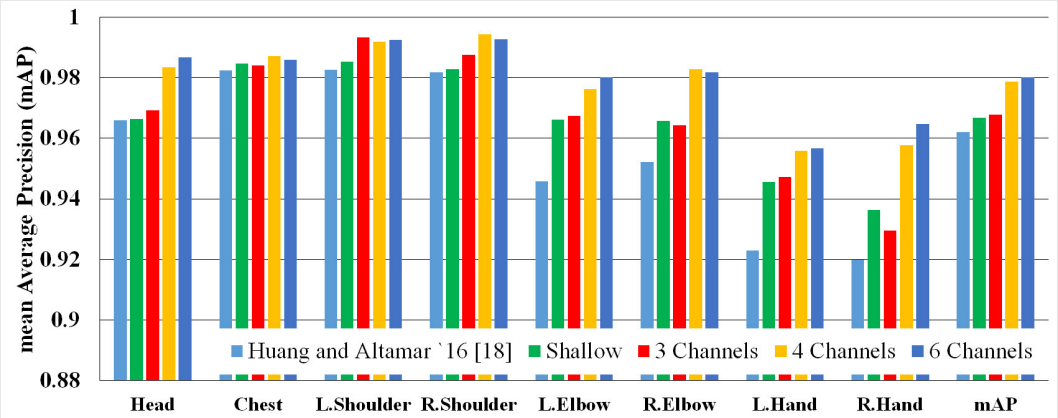


Figure 7.3. Comparison of the mean average precision (mAP) using the EVAL dataset.

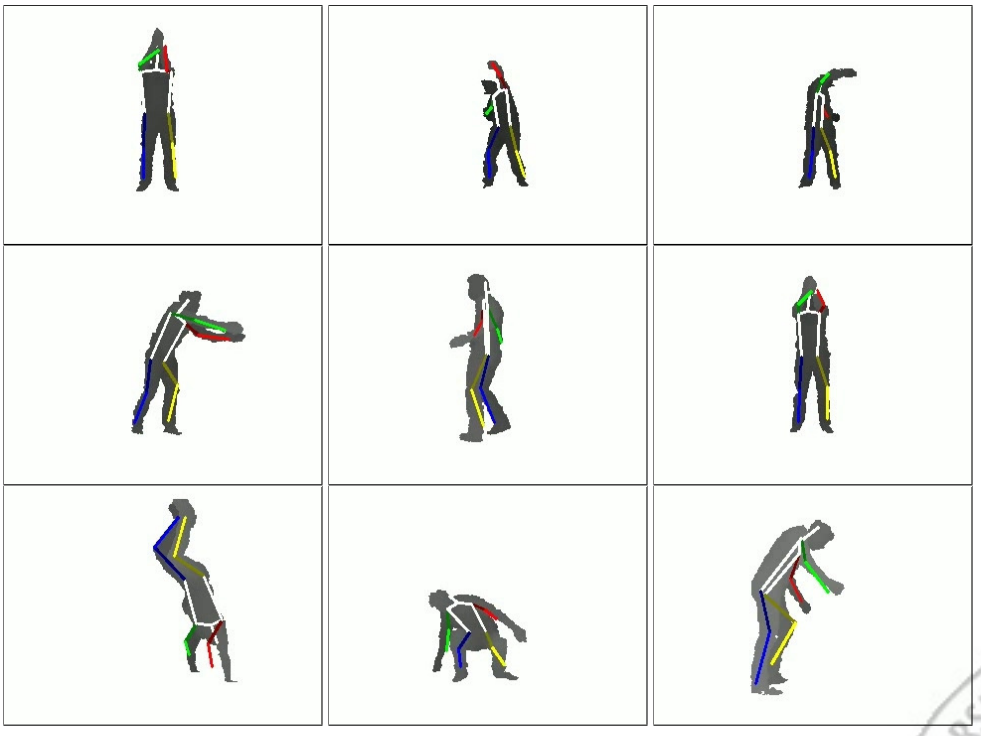


Figure 7.4. Some successful human pose estimation results using six-channels.

7.5 Dance Performance Evaluation

To validate the dance performance evaluation, we constructed a large K-Pop dance dataset 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 build a learners' K-Pop dance database that was recorded by four learners with various dance skill levels. Learner's dance sequences were labeled subjectively by a group of the dance experts as *best*, *good*, *bad*, and *worst*.

Note that the proposed human pose estimation method differentiated frontal and rear poses successfully (Fig. 7.5).

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 by the program. Evaluation scores were calculated by a conventional feature in [27], by the proposed dance feature using the feature-based pose estimation method, and by the proposed dance feature using the multi-channel CNN-based pose estimation method. Table. 7.4 shows the evaluation scores of the proposed method with multi-channel CNN-based pose estimation.

Dance evaluation scores from the proposed dance feature using the multi-channel CNN-based pose estimation agreed with 98 times out of 100 with the evaluations of dance experts; the exceptions were the 15th and 23rd sequences, whereas dance evaluation from [27] agreed 86 times and the proposed dance feature using the feature-based pose estimation agreed 97 times.

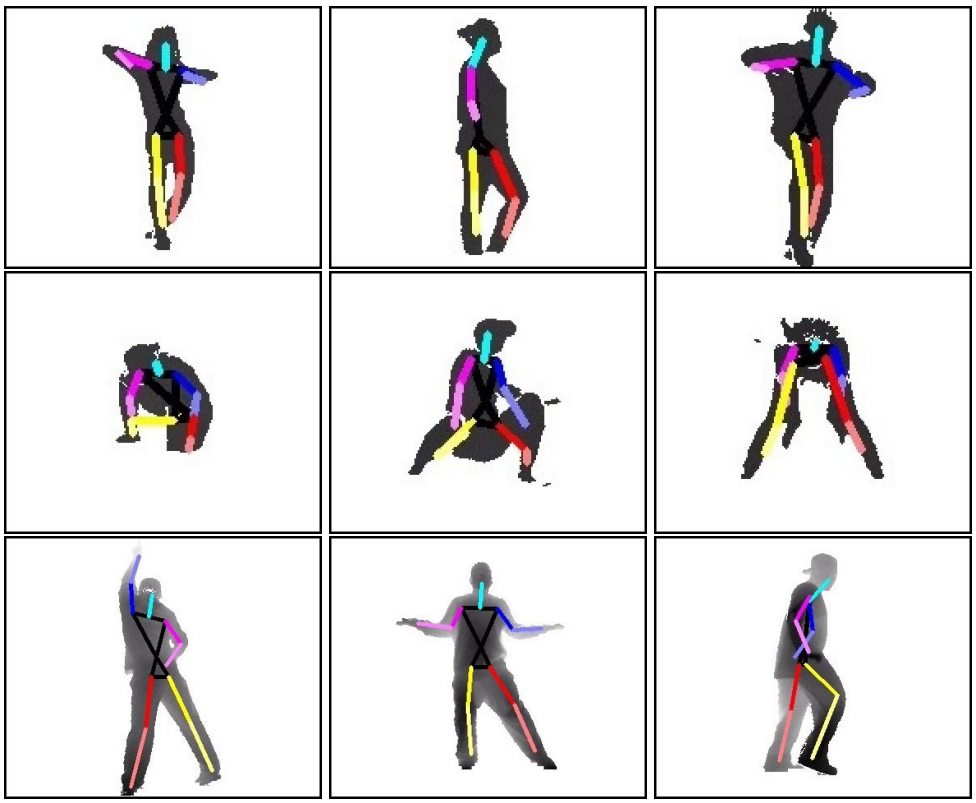
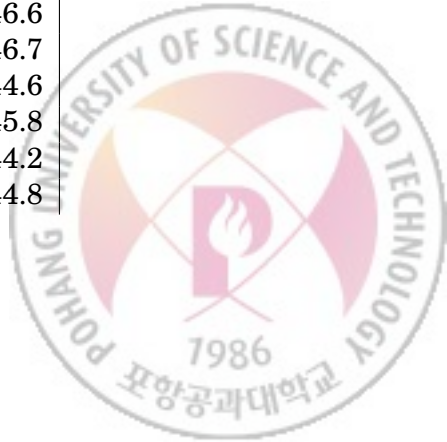


Figure 7.5. Human pose estimation results with different learners.

Table 7.4 Four learners' examination scores that are measured by the K-Pop dance teacher program. **Bold scores** within row and section disagree with expert evaluation.

Seq	S1	S2	S3	S4	Seq	S1	S2	S3	S4
1	62.4	56.2	51.9	47.6	51	64.0	55.2	52.3	48.5
2	58.2	57.8	49.5	46.8	52	61.7	56.6	53.7	47.3
3	61.9	55.9	51.8	46.6	53	63.0	57.6	52.5	47.7
4	62.1	58.0	53.6	49.2	54	62.4	57.6	51.9	46.9
5	62.0	56.6	52.4	48.4	55	63.3	52.8	52.3	49.6
6	64.1	57.9	53.7	49.3	56	60.4	56.1	50.9	45.9
7	58.0	52.7	48.1	45.1	57	62.5	57.1	53.7	47.2
8	60.1	56.0	52.4	46.7	58	59.7	56.2	53.7	47.9
9	61.3	58.2	53.4	45.7	59	59.4	54.8	52.3	46.6
10	60.7	54.7	52.9	46.5	60	57.7	55.8	51.7	44.3
11	61.7	56.4	54.0	46.4	61	59.2	55.5	52.7	45.0
12	62.8	58.7	54.5	47.2	62	59.7	57.4	51.9	47.7
13	60.2	58.3	53.0	47.7	63	60.3	52.7	52.3	46.2
14	63.0	59.8	52.9	49.2	64	57.1	54.8	50.7	47.1
15	**40.5**	**37.9**	**35.9**	**48.0**	65	61.7	58.3	51.0	48.6
16	62.4	56.3	51.9	46.5	66	53.3	52.2	51.2	45.1
17	62.1	57.4	53.8	46.5	67	62.0	57.0	52.5	48.8
18	60.3	55.2	50.4	46.2	68	60.5	58.1	52.0	47.5
19	58.4	52.9	52.6	45.7	69	60.2	52.6	51.9	46.5
20	62.2	53.1	53.1	47.8	70	62.2	57.7	52.6	48.0
21	63.2	56.7	52.0	47.0	71	62.1	56.6	51.7	46.5
22	62.3	55.9	51.9	47.9	72	61.9	56.3	52.2	47.4
23	58.7	51.5	**45.6**	**46.2**	73	62.2	55.2	49.5	45.3
24	61.2	55.0	52.1	46.6	74	60.8	54.9	51.5	45.8
25	60.2	56.8	50.1	46.1	75	62.5	57.3	53.1	47.0
26	63.0	56.9	50.5	45.6	76	60.8	56.9	48.0	45.4
27	60.1	58.6	50.3	46.8	77	62.5	56.7	49.9	48.0
28	61.5	57.6	51.0	48.1	78	61.5	55.3	50.7	47.5
29	57.9	53.5	50.8	47.4	79	61.0	57.2	51.6	45.6
30	59.9	54.1	50.1	46.4	80	61.3	58.6	51.4	46.6
31	60.8	56.7	52.8	46.3	81	62.0	50.7	49.7	46.7
32	61.9	57.4	52.5	47.8	82	60.0	54.4	51.5	44.6
33	62.2	56.8	50.8	47.1	83	62.8	56.0	52.2	45.8
34	57.7	56.0	52.6	46.7	84	60.1	54.0	49.0	44.2
35	60.3	57.1	50.2	48.4	85	63.2	55.0	50.7	44.8



36	54.4	49.3	44.9	42.6	86	59.3	58.4	53.5	48.3
37	61.6	57.2	51.0	46.6	87	59.5	57.2	52.3	49.4
38	49.2	46.9	40.7	38.4	88	62.3	57.5	50.5	49.2
39	61.9	56.7	51.7	45.7	89	62.0	57.2	51.1	48.3
40	63.0	57.4	53.1	47.5	90	60.3	57.0	52.9	47.7
41	59.9	56.1	53.8	47.9	91	56.1	52.8	46.9	43.6
42	62.1	56.3	51.6	47.0	92	60.1	58.3	53.0	47.8
43	60.1	57.1	51.5	48.1	93	61.8	57.2	50.8	48.4
44	60.3	55.9	51.1	47.5	94	58.8	52.7	50.2	45.2
45	62.7	59.2	51.8	48.4	95	62.6	56.8	51.5	48.0
46	56.4	54.3	48.6	43.3	96	58.6	56.4	49.7	47.7
47	61.7	56.6	52.4	48.4	97	59.2	57.2	50.2	48.0
48	61.7	58.2	51.6	47.9	98	61.5	57.4	49.9	46.5
49	63.5	55.6	50.5	46.0	99	60.0	58.7	50.4	47.2
50	59.7	56.4	52.1	48.1	100	61.3	56.4	51.3	46.9

CHAPTER 8

Conclusions

We proposed a novel representation of the human body using ridge data that defined as the local maxima in the distance transform map in a selective representation of the skeleton. The proposed ridge data can be extracted although there are occlusion, full-body rotation, and fast movement.

The process of segmenting the human silhouette consists of four steps; (1) To disconnect every object, we removed the floor data using iterative planar equation fitting, (2) Using the modified connected component labeling, we segmented every object in the image, (3) We uniquely detected human among the segmented object by using motion information, (4) To maintain the human information, we needed to identify each human by data association,

The process of extracting the ridge data from the human silhouette consists of three steps; (1) From the segmented human silhouette, we computed the depth edge, (2) We transform the edge image to the distance transform map, (3) We extracted the ridge data by using the sum of distance values on the predefined circle. We compare the proposed ridge data with existing medial

axis feature and dilated medial axis feature and show that the ridge data is the plentiful and scale-invariant representation of body skeletons.

We validated that the ridge data can be used in both of the feature-based and the CNN-based human pose estimation methods through the various experiments. For the feature-based human pose estimation, we utilized the ridge data as essential evidence for the joint position since the positions of most joint lie on the ridge data. When we generated the initial human model, we tried to find the geodesic extremal points to localize the peripheral joints. After finding the joints, we could compute the intermediate joints, such as elbows, and knees. In the hierarchical human joint estimation, we tried to collect the ridge data that meet the constraints for each joint. We think that the search space could be reduced and joint drift problem was solved in this step. Finally, the feature-based method achieves high pose estimation accuracy (0.9435 mAP on SMMC-10, 0.9358 mAP on EVAL) and low pose estimation error (3.88 cm on the SMMC-10 dataset and 4.72 cm on the EVAL dataset) with 3.45 ms of average response time.

For the CNN-based methods, we utilized the ridge data in either the ridge image or the projected ridge image. We defined the ridge image and the projected ridge image to represents the probability of ridgeness at each position on the XY plane and XY, YZ, and ZX plane, respectively. As the multi-channel CNN-based method used the projected ridge images with the projected depth images, we could achieve the significant improvement of the pose estimation accuracy. The proposed keypoint generation modules in the multi-channel CNN-based human pose estimation improved the accuracy without the ridge data, and they achieved the highest pose estimation accuracy (0.9801 mAP) on EVAL dataset.

By using the proposed human pose estimation methods and the K-pop dance

dataset, we evaluated the dance performances of the K-Pop dance learners with high confidence, and the proposed K-Pop dance teacher coincided with the dance experts' evaluations of dance performance at 98% accuracy.

REFERENCES

[1] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele, "Articulated people detection and pose estimation: Reshaping the future," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3178–3185.

[2] H. Lu, X. Shao, and Y. Xiao, "Pose estimation with segmentation consistency," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4040–4048, 2013.

[3] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2010, pp. 3108–3113.

[4] H. P. Jain, A. Subramanian, S. Das, and A. Mittal, "Real-time upper-body human pose estimation using a depth camera," in *Computer Vision and Computer Graphics Collaboration Techniques*. Springer, 2011, pp. 227–238.

[5] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1297–1304.

[6] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proceedings of IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 415–422.

[7] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *Proceedings of European conference on Computer Vision*. Springer-Verlag, 2012, pp. 872–885.

[8] K. Buys, C. Cagniart, A. Baksheev, D. T. Laet, J. D. Schutter, and C. Pantofaru, "An adaptable system for rgb-d based human body detection and pose estimation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 39–52, 2014.

[9] D. Grest, J. Woetzel, and R. Koch, "Nonlinear body pose estimation from depth images," in *Pattern Recognition*. Springer, 2005, pp. 285–292.

[10] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3d human body tracking with an articulated 3d body model," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2006, pp. 1686–1691.

[11] B. Rosenhahn, C. Schmaltz, T. Brox, J. Weickert, D. Cremers, and H. Seidel, "Markerless motion capture of man-machine interaction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[12] M. Straka, S. Hauswiesner, M. R"uther, and H. Bischof, "Skeletal graph based human pose estimation in real-time." in *Proceedings of British Machine Vision Conference*, 2011, pp. 1–12.

[13] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt, "Performance capture of interacting characters with handheld kinects," in *Proceedings of European Conference on Computer Vision*. Springer, 2012, pp. 828–841.

[14] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *Proceedings of IEEERSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2389–2395.

[15] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 755–762.

[16] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[17] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[18] J. Huang and D. Altamar, "Pose estimation on depth images with convolutional neural network," 2016.

[19] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 332–347.

[20] X. Yang and Y. Tian, "Effective 3d action recognition using eigenjoints," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 2–11, 2014.

[21] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *IEEE computer society conference on Computer vision and pattern recognition workshops (CVPRW)*, 2012, pp. 20–27.

[22] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human activity detection from rgbd images." *plan, activity, and intent recognition*, vol. 64, 2011.

[23] M. Reyes, G. Dominguez, and S. Escalera, "Featureweighting in dynamic timewarping for gesture recognition in depth data," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1182–1188.

[24] A. Jalal, S. Kamal, and D. Kim, "A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments," *Sensors*, vol. 14, no. 7, pp. 11 735–11 759, 2014.

[25] A. Jalal, S. Kamal, and D. Kim, "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 (URAI)*. IEEE, 2015, pp. 294–299.

[26] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, pp. 28–35.

[27] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*. ACM, 2011, pp. 147–156.

[28] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.

[29] F. Ofli, G. Kurillo, Š. Obdržálek, R. Bajcsy, H. Jimison, and M. Pavel, "Design and evaluation of an interactive exercise coaching system for older adults: lessons learned," *IEEE journal of biomedical and health informatics*, vol. 20, no. 1, p. 201, 2016.

[30] N. Thome, D. Merad, and S. Miguet, "Human body part labeling and tracking using graph matching theory," in *International Conference on Video and Signal Based Surveillance*. IEEE, 2006, pp. 38–38.

[31] N. M. Tahir, A. Hussain, S. A. Samad, H. Husain, M. Demiralp, N. Baykara, and N. Mastorakis, "A machine learning approach for posture recognition based on simplified shock graph," in *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, no. 8. World Scientific and Engineering Academy and Society, 2009.

[32] M. Dillencourt, H. Samet, and M. Tamminen, "A general approach to connected-component labeling for arbitrary image representations," *Journal of the ACM*, vol. 39, no. 2, pp. 253–280, 1992.

[33] Y. Kim and D. Kim, "Efficient body part tracking using ridge data and data pruning," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 114–120.

[34] H. Jain, A. Subramanian, S. Das, and A. Mittal, "Real-time upper-body pose estimation using a depth camera," in *Lecture Notes in Computer Science on Pattern Recognition*. Springer, 2011, vol. 6930, pp. 227–238.

[35] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Processing of IEEE International Conference on Computer Vision*, 2011, pp. 1092–1099.

[36] M. Straka, S. Hauswiesner, M. Rüther, and H. Bischof, "Simultaneous shape and pose adaption of articulated models using linear optimization," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 724–737.

[37] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, 2007, p. 72.

[38] Y. Chen, Z. Wang, Y. Peng, and Z. Zhang, "Cascaded pyramid network for multi-person pose estimation," in *IEEE conference on computer vision and pattern recognition*, 2018.

[39] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *Proceedings of European Conference on Computer Vision*, 2018.

[40] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns,"

in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 3593–3601.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real-time human pose tracking from range data," *Proceedings of European Conference on Computer Vision*, pp. 738–751, 2012.

[43] D. Demirdjian, T. Ko, and T. Darrell, "Constraining human body tracking," in *Proceedings of Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1071–1078.

[44] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2345–2352.

[45] H. Jung, S. Lee, Y. Heo, and I. Yun, "Random tree walk toward instantaneous 3d human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2467–2474.

한글요약문

깊이 영상내 산등성이 데이터를 이용한 3차원 사람 자세 추정

사람 자세 추정은 동작 인식, 컴퓨터 상호 작용, 오락 등 다양한 실용 분야에서 인간의 행동을 이해하는 가장 근본적인 기술이다. 본 학위 논문은 단일 깊이 영상의 거리 변환 지도에서 국지적인 최대치인 산등성이 데이터를 제안하며, 가려짐, 전신 회전 및 빠른 움직임이 발생해도 사람 뼈대에 대한 선택적 표현을 보여준다.

깊이 영상에서 사람의 실루엣을 분리하여 산등성이 데이터를 추출한다. 사람 실루엣 분리 프로세스는 바닥 제거, 객체 분할, 사람 검출 및 사람 식별의 네 단계로 구성된다. 그런 다음, 분리된 사람 실루엣의 가장자리 이미지로부터 거리 변환 지도를 계산한다. 산등성이 데이터는 거리 변환 지도에서 국부적 최대치를 찾아 추출한다. 산등성이 데이터의 효과를 나타내기 위해 산등성이 데이터를 사용하는 두 가지 유형의 사람 자세 추정 방법을 제안한다. (1) 특징 기반 계층적 사람 자세 추정, (2) 합성곱 신경망 네트워크 기반 사람 자세 추정.

특징 기반의 방법은 초기 사람 모델에 따라 잘못된 데이터를 제거하여 사람의 관절을 계층적으로 추적한다. 초기 사람 모델의 매개변수는 신체 부위의 길이와 각도로서, 초기 자세에서 측정되거나 사람 자세 데이터베이스에서 검색된다. 특징 기반 사람 자세 추정은 관절 예측, 후보 수집, 데이터 정리 및 관절 추정의 네 가지 순차적 하위 작업으로 구성된다. 하위 작업은 머리, 몸통 및 팔다리의 계층적 순서로 사람 관절을 추적한다.

CNN(Convolutional Neural Network)기반 방법은 다음과 같은 두 가지 방법으로 구성된다. (1) 얕은 CNN 기반 회귀분석 방법 (2) 다채널 CNN 기반 회귀분석 방법 얕은

CNN에 기반한 회귀분석 방법은 3개의 합성곱 레이어와 3개의 완전히 연결된 레이어로 구성되어 있으며, 3가지 유형의 손실 함수를 사용하여 입력 깊이 영상에서 3차원 사람 자세를 직접적으로 분석한다. 본 학위 논문에서는 산등성이 데이터를 개별 화소에서 산등성이 정도을 나타내는 하나의 추가 채널로써 사용한다. 다채널 CNN 기반 회귀분석 방법은 깊이 영상과 산등성이 데이터를 3개의 직교 평면에 투영하고 2차원 히트맵을 생성하여 각 평면에서 관절의 위치를 추정한다. 각 평면의 추정 관절 위치는 3차원 사람 자세를 분석하기 위해 결합되어 완전히 연결된 세 개의 레이어에 공급된다.

본 학위 논문에서는 사람 자세 추정 방법의 정확성을 보여주기 위해 K-Pop 댄스 선생님을 제안한다. K-Pop 댄스 선생님은 학습자의 춤 실력을 타이밍과 자세 정확도 측면에서 자동으로 평가하며, 이를 위해 학습자의 자세를 인접한 관절 간의 각도로 표현한 댄스 특징으로 변환한다.

본 학위 논문에서 제안한 방법의 유효성을 확인하기 위해 벤치마크 Dataset인 SMMC-10 및 EVAL과, 대용량 K-Pop 댄스 Dataset에 대해 여러 가지 실험을 수행하였다. 또한 제안한 산등성이 데이터의 효과를 검증하기 위해, 산등성이 데이터를 중축 변환(Medial Axis Transform) 및 확장 중축 변환(Dilated Medial Axis Transform)과 같은 기존의 골격화 기술과 비교하였다. 제안한 특징 기반 사람 자세 추정 방법은 SMMC-10과 EVAL Dataset에서 각각 0.7735와 0.9358의 자세 추정 정확도(mAP)와 3.88cm와 4.72의 평균 자세 오차(cm)를 달성하였다. 특징 기반 방법의 평균 계산 시간은 3.45ms(290fps)이다. 제안한 다채널 CNN 기반 사람 자세 추정 방법은 EVAL Dataset에서 자세 추정 정확도가 0.9801 mAP이다. K-Pop 댄스 선생님은 전문가 평가와 98%의 일관성을 달성했다.

Acknowledgements

감사의 글

돌아보면 참 길기만 한 포항에서의 삶이 한 순간으로 압축이 되는 신기한 경험을 합니다. 이전 연구실에서의 짧은 생활에서 IM 연구실로의 변화, 너무나 소중한 나의 벗과의 헤어짐과 재회, 원하는 결과가 안 나와 안 돌아가는 머리를 싸매며 밤을 낮과 같이 해 아리던 날들. 처음부터 가지고 있던 졸업 후의 이미지가 흐려질 때 쯤 안 올 것 같던 그 날이 왔습니다. 포기하자는 말을 수없이 되뇌였지만 그 미약한 부정의 단어를 고이 접어 날리며 오늘까지 올 수 있었습니다.

지금 이 자리도 저의 힘만으로 올 수 없음을 알고 있기에 하늘 아래 감사를 드려야 할 분이 너무나 많습니다. 우선 부족한 저를 항상 참고 기다려주시고 많은 가르침을 주신 김대진 교수님에게 정말 큰 감사를 드립니다. 연구실을 옮기려고 하는 저를 기꺼이 받아주신 교수님이 안 계셨다면 학위 심사 조차도 가질 수 없었다는 점 깊이 명심하겠습니다. 교수님의 100세 인생을 위해 조금만 더 건강에 신경쓰시는 모습을 보여주시면 좋겠습니다. 갑작스러운 학위 심사 요청에도 소중하고 날카로운 지적을 아끼지 않아 주신 조민수 교수님, 곽수하 교수님, 박순용 교수님, 최희열 교수님께도 감사의 인사를 드립니다. 엄중한 학위 심사를 통해 박사 학위의 무거움과 책임감을 더욱 크게 느낄 수 있었습니다. POSTECH 출신으로서 부끄럽지 않도록 더욱 더 노력하겠습니다.

언제든 곁에서 도와주고 격려해준 봉남이형, 형이 있어서 졸업을 할 수 있네요. 항상 도움만 얻어와 미안하고 감사합니다. 항상 시니컬하지만 그래서 핵심을 찝어내는 현성이, 논문, 실험, 결혼 산적한 일이 많지만 너에게도 좋은 날이 반드시 올 거야. 존재로서 의지가 됐던 용현이, 넌 별로 걱정이 안 돼. 누구보다 잘 해낼거란 걸 알기 때문이지. 조금만 먼저 졸업할게. 덩치에서든 태도에서든 항상 든든한 인한이, 연구실에서 누구도

걸어가지 않은 길을 가느라 힘들겠지만 누구보다 잘 해낼거라 믿어. 밝은 웃음이 보기 좋은 혜민이, 연구실 생활이 힘들겠지만 너의 장점인 큰 목소리와 같이 힘찬 믿음으로 이겨나갈 수 있을거야. 연구실 풋살 스트라이커 은섭이, 랩장과 과제, 연구 할 일도 많고 힘들겠지만 운동 열심히해서 더욱 체력을 키워 버티다 보면 너에게도 좋은 날이 올거야. 항상 병약해 보이는 정현이, 어느덧 4년차가 되었구나. 좋은 연구 성과를 얻어 어서 논문 통과도 되야하지만, 너야 말로 운동이 필요해. 말수는 적지만 가끔씩 하는 말이 촌철살인인 동민이, 조금만 더 너의 고민이나 생각을 다른 사람과 공유를 하길 바래. 다들 너의 동지거든. 같은 동네 산다는 것도 1년이 가까이 되서 알게된 준영이, 1년차 답지 않게 날카롭고 정확한 분석이 돋보이는 구나. 거제도에서 만나면 족발 사줄게. 항상 멋있고 좋아보이는 물건들로 가득한 얼리 어답터 태훈이, 너의 미래에도 항상 멋있고 좋은 일만 있길 바랄게. 주말 밤마다 다음 주의 시작을 준비하듯 연구실을 찾던 태욱이, 너의 성실함과 진지함에서 밝은 미래가 보이는 것 같다. 내 마지막 부사수 명준이, 나이 차이도 많고 아직까진 이야기도 많이 못 해봤지만 내가 나가기 전까지 알고 있는 모든 걸 알려줄게. 기대해. 이야기 한 번 제대로 못 해본 Debi, I wish you get every success and let your future shine. 밤낮가리지 않고 도와달라 연락해도 흔쾌히 받아준 지은씨, 더욱 더 연구실의 역사를 지켜보고 연구실을 지키는 수호신이 되시길... 상훈이, 남편과의 행복한 시간도 함께 빌어줄게요.

제 첫번째이자 마지막 사수인 대환이형, 함께한 시간이 너무나 짧아 아쉽기만 하지만 그 때 알려주신 것들이 저에겐 너무 소중한 지식으로 남았습니다. 먼저 연구실을 떠난 동기 현진이와 종민이형, 동기이지만 다른 사람보다 더 이야기를 못 한것 같아 아쉽네요. 지금까지 내 부사수 중에 유일하게 졸업한 은지, 최근 소식은 전해 듣지 못하지만 항상 너의 이야기가 궁금하단다. 포항에서 외로운 몸과 마음을 함께 달래준 동갑내기 진욱이, 앞으로 더 자주 연락하고 소식 전하자.

5년만 더 기다려달라며 포항에 내려올 때, 묵묵히 저를 믿어주신 부모님에게 이제야 감사의 말씀을 올립니다. 아들이 힘들까봐 한번도 아쉬운 소리 안 하시며 모든 걸 참고

기다려주시며 너무 고생 많으셨습니다. 부족한 아들이 효도 한 번 못 해드리는 사이, 곁에서 내 몫까지 다 해준 은경이도 너무 고맙다. 앞으로 지금까지 못한 효도 몰아서 해드릴게요. 건강하게 오래 오래 함께 하세요. 소중한 따님을 먼 타지로 보내주신 것도 모자라 매일 매주 기도로써 저에게 힘을 주신 장모님, 믿음으로 감사의 마음을 전하겠습니다.

못나고, 무뚝뚝하고, 상처도 많이 줬지만, 언제나 어디서나 함께 할 내 소중한 벗 유정아. 지금까지 내 모든 즐거움과 어려움을 함께해줘서 너무 고맙고, 앞으로도 모든 일 같이 머리 맞대고 생각하고 고민하고 즐깁시다. 늘 연구실에서 늦게 돌아와 홀로 집을 지키느라 외롭고 힘들었겠지만 잘 참고 견뎌줘서 너무 고마워. 항상 행복하게 살기위해 노력하고, 서로의 있는 그대로의 모습을 존중하고 아끼는 모습 보여줄게요. 원하는 모든 것을 얻을 수는 없겠지만 지금 우리에게 주어진 모든 것에 감사하고 사랑해보자.

Curriculum Vitae

Name : Yeonho Kim

Education

2001.3-2008.8 : B.S. in Department of Computer Science and Information Engineering, Catholic University of Korea

2009.3-2016. : Ph.D. in Department of Computer Science and Engineering, POSTECH
Thesis Title :
깊이 영상내 산등성이 데이터를 이용한 3차원 사람 자세 추정(3D Human Pose Estimation Using Ridge Data in Depth Image)
Advisor: Prof. 김대진(Daijin Kim)

PUBLICATIONS

International Journal

1. **Yeonho Kim** and Daijin Kim, "Real-Time Dance Evaluation by Markerless Human Pose Estimation," *Multimedia Tools and Applications, ISSN: 1573-7721,* pp. 1-22, 2018.

2. Heeseung Kwon, **Yeonho Kim**, Jin S. Lee, and Minsu Cho, "First Person Action Recognition via Two-stream ConvNet with Long-term Fusion Pooling," *Pattern Recognition Letters, ISSN: 0167-8665, vol. 112,* pp. 161-167, 2018.

3. Ahmad Jalal, **Yeonho Kim**, Yongjoong Kim, and Daijin Kim, "Robust Human Activity Recognition from Depth Video using Spatiotemporal Multi-Fused Features," *Pattern Recognition, ISSN 0031-3203, vol. 61,* pp. 295-308, 2017.

International Conference

1. **Yeonho Kim**, Daijin Kim, "Interactive Dance Performance Evaluation using Timing and Accuracy Similarity," *ACM SIGGRAPH 2018 Poster ACM,* 2018.

2. **Yeonho Kim**, Daijin Kim, "Efficient Body Part Tracking using Ridge Data and Data Pruning," *IEEE-RAS International Conference on Humanoid*

Robots, 2015.

3. Eunji Cho, **Yeonho Kim**, and Daijin Kim, "Accurate Human Pose Estimation by Aggregating Multiple Pose Hypotheses using Modified Kernel Density Approximation," *IEEE International Conference on Image Processing 2015,* 2015.

4. Ahmad Jalal, **Yeonho Kim**, Shaharyar Kamal, Adnan Farooq, and Daijin Kim, "Human Daily Activity Recognition with Joints Plus Body Features Representation using Kinect Sensor," *4th International Conference on Informatics, Electronics and Vision,* 2015.

5. **Yeonho Kim**, Myoungcheol Sung, and Daijin Kim, "A Virtual Keyboard Interface for Head-Mounted Devices," *9th International Conference on Interfaces and Human Computer Interaction,* 2015.

6. Ahmad Jalal, **Yeonho Kim**, and Daijin Kim, "Ridge Body Parts Features for Human Pose Estimation and Recognition from RGB-D Video Data," *2014 5th International Conference on Computing Communication and Networking Technologies (ICCCNT),* 2014.

7. Ahmad Jalal, **Yeonho Kim**, and Daijin Kim, "Dense Depth Maps-Based Human Pose Tracking and Recognition in Dynamic Scenes using Ridge Data," *11th IEEE International Conference on Advanced Video and Signal-Based Surveillance,* 2014.

8. Daehwan Kim, **Yeonho Kim**, and Daijin Kim, "Separating Occluded Humans by Bayesian Pixel Classifer with Re-Weighted Posterior Probability," *Advanced Concepts for Intelligent Vision systems,* 2011.

International Workshop

1. **Yeonho Kim**, Hyunjin An, Daehwan Kim, Daijin Kim, "Illumination and Pose Robust Hand Detection Using Volumetric Feature Vector and 3D Haar-like Filters," *POSTECH-KYUTECH Joint Workshop On Neuroinformatics*, 2011.

2. Daehwan Kim, **Yeonho Kim**, Daijin Kim, "Pixel Classification Method for Separating Occluded Humans," *POSTECH-KYUTECH Joint Workshop On Neuroinformatics*, 2010.

Domestic Conference

1. **김연호**, 성명철, 김대진, "HMD를 위한 가상 키보드 인터페이스," *한국컴퓨터종합학술대회 (KCC 2015)*, 2015.

2. **김연호**, 김대진, "혼합 부위 모델을 이용한 실시간 사람 자세 추정," *영상처리 및 이해에 관한 워크샵 (IPIU 2014)*, 2014.

3. 성명철, **김연호**, 김대진, "Touchless Interface for Head-Mounted Devices," *영상처리 및 이해에 관한 워크샵 (IPIU 2014)*, 2014.

4. **김연호**, 김대진, "산등선이 데이터와 데이터 소거법을 이용한 사람 포즈 인식 방법," *정보과학회 추계 학술대회 (KIISE 2013)*, 2013.

5. 김대환, 안현진, **김연호**, 김대진, "TOF 영상에서의 볼륨 특징 기반 3차원 손 검출," *HCI 2012*, 2012.

PATENTS

Patents in US

1. Dai-Jin Kim, **Yeon-Ho Kim**,
   "Method of Extracting Ridge Data and Apparatus and Method for Tracking Joint Motion of Object" US, 14/562,848 (2016-10-04).

2. Daijin Kim, Hyunjin An, DaeHwan Kim, **Yeonho Kim**,
   "Method and Apparatus for Hand Detection using Volumetric Feature Vector and 3D Haar-Like Filters" US, 13/407,487 (2012-02-28).

Patents in Korea

1. **Yeonho Kim**, Daijin Kim,
   "안무 평가 장치 및 방법", 출원번호 10-2017-0184573, 2017

2. **Yeonho Kim**, Daijin Kim,
   "산등성이 데이터를 이용한 객체의 움직임 추적 장치 및 방법", 등록번호 10-0152128, 2015

3. Daijin Kim, Hyunjin An, DaeHwan Kim, **Yeonho Kim**,
   "부피 특징 백터와 3차원 하르-유사 필터를 이용한 물체 검출 방법 및 장치", 등록번호 10-1233843, 2013

PROJECTS

1. Development of 3D Human Pose Estimation, "Core Technology Development for Breakthrough of Robot Vision Research," Silver Robot, 2009.05.01-2014.04.30
2. System Integration on Unity 3D, "Research Laboratory for Natural Language-based Immersive English Tutoring System," National Research Foundation (NRF), 2009.09.01-2015.08.31
3. Development of 3D Human Pose Estimation, "Development of Robot Vision SoC/Module for Acquiring 3D Depth Information and Recognizing Objects/Faces," Korea Electronics Technology Institute,
4. Development of 3D Human Pose Estimation, "Human Detection, Face and Emotion Recognition based 2D/3D Images," Korea Institute of Science and Technology (KIST), 2010.04.01-2013.03.31
5. Development of 3D Human Pose Estimation, "Development of 3D Vision Human Modeling," LG Electronics, 2010.07.01-2010.12.31
6. Development of Hand Gesture Recognition, "Deveopment of Efficient Hand Detection and Stereo Image based Pose Detection," Electronics and Telecommunications Research Institute (ETRI), 2010.08.01-2011.01.31
7. Development of Hand Gesture Recognition, "Development of Hand Gesture Recognition for Head-Mounted Device (HMD)," Future IT Innovation Laboratory, 2010.11.01-2011.04.29

8. Development of Hand Gesture Recognition, "Development of Hand Gesture-based Multi-Spatial Touch Interface for 3D Smart TV," LG Electronics, 2011.10.01-2012.06.30

9. Development of 3D Human Pose Estimation, "Development of Next Generation IT Convergence Project," Future IT Innovation Laboratory, 2011.08.01-2014.10.23

10. Depth Camera Calibration, "Research and Development of Test System Control based on Mixed Reality Interaction," Electronics and Telecommunications Research Institute (ETRI), 2011.11.07-2012.01.31

11. Development of 3D Human Pose Estimation, "Development of Real-time Body Part Detector using 3D Image," LG Electronics, 2012.04.15-2013.02.28

12. Development of 3D Human Pose Estimation, "Implementation of Technologies for Identification, Behavior, and Location of Human based on Sensor Network Fusion," Korea Institute of Science and Technology (KIST), 2012.06.01-2017.05.31

13. Development of 2D Human Pose Estimation, "Development of Elementary Technology for 2D Image based Human Body Joint Detection," Samsung Electronics, 2012.10.16-2013.05.10

14. Development of 2D Human Pose Estimation, "Research on 2D Image based Human Body Joint Detection," Samsung Electronics, 2013.07.15-2013.12.13

15. Development of 3D Human Pose Estimation, "SW Developer Platform for Smart Appliance and Industrial Motion Recognition using TOF and Dual Camera," National IT Industry Promotion Agency (NIPA), 2014.06.01-2014.11.30

16. Development of 3D Human Pose Estimation, "Development of Real-time Human Skeleton Detection and Tracking based on 3D Depth Map," Electronics and Telecommunications Research Institute (ETRI), 2014.09.01-2015.02.28

17. Development of 3D Human Pose Estimation, "Development of IT Convergence Leading Technology for Next-generation Commercialization," Future IT Innovation Laboratory, 2014.11.01-2017.10.31

18. Development of 3D Human Pose Estimation, "Development of 3D Position Estimation for Overlapped and Occluded Joint," Electronics and Telecommunications Research Institute (ETRI), 2015.06.01-2016.02.28

19. Development of 2D Human Pose Estimation and Action Recognition, "Development of Robotics for Mood Stabilization and Cognitive Improvement in Patients with Mild Cognitive Impairment and Dementia," Ministry of Trade, Industry & Energy, 2016.05.01-2018.12.31

20. Development of Highlight Detector, "The Development of Multi-view Multi-channel Broadcasting System for Sports Digital Contents," Ministry of Culture, Sports and Tourism, 2016.06.01-2017.12.31

21. Development of Baggage Detector in Depth Image, "Development of Intelligent Video Analysis for Object Detection and Recognition in

Classification System," POSCO ICT, 2017.01.01-2017.12.31

22. Project Management, "Development of Object Detection and Recognition for Intelligent Vehicles," Ministry of Science and ICT, 2017.04.01-2018.12.31

23. Development of Baggage Tracker in Depth Image, "Development of Intelligent Video Analysis for Efficient Baggage Transfer in Logistics System," POSCO ICT, 2018.01.01-2018.12.31

24. Development of 3D Human Pose Estimation, "Research on Deep Learning Networks for Extracting 3D Human Skeleton," Electronics and Telecommunications Research Institute (ETRI), 2018.09.15-2018.11.30