

A Sequential Approach to 3D Human Pose Estimation: Separation of Localization and Identification of Body Joints

Ho Yub Jung², Yumin Suh¹, Gyeongsik Moon¹, and Kyoung Mu Lee¹

¹ Department of ECE, ASRI, Seoul National University, Seoul, Korea
{n12345,mks0601,kyoungmu}@snu.ac.kr

² Division of CESE, Hankuk University of Foreign Studies, Yongin-si, Korea
jung.ho.yub@gmail.com

Abstract. In this paper, we propose a new approach to 3D human pose estimation from a single depth image. Conventionally, 3D human pose estimation is formulated as a detection problem of the desired list of body joints. Most of the previous methods attempted to simultaneously localize and identify body joints, with the expectation that the accomplishment of one task would facilitate the accomplishment of the other. However, we believe that identification hampers localization; therefore, the two tasks should be solved separately for enhanced pose estimation performance. We propose a two-stage framework that initially estimates all the locations of joints and subsequently identifies the estimated joints for a specific pose. The locations of joints are estimated by regressing K closest joints from every pixel with the use of a random tree. The identification of joints are realized by transferring labels from a retrieved nearest exemplar model. Once the 3D configuration of all the joints is derived, identification becomes much easier than when it is done simultaneously with localization, exploiting the reduced solution space. Our proposed method achieves significant performance gain on pose estimation accuracy, thereby improving both localization and identification. Experimental results show that the proposed method exhibits an accuracy significantly higher than those of previous approaches that simultaneously localize and identify the body parts.

Keywords: Depth camera · Human pose · Regression forest

1 Introduction

Real-time 3D human pose estimation is a core technique for activity recognition. It has diverse applications including human-computer-interface development, teleoperation, health monitoring, and learning by demonstration (LbD) [1]. Conventionally, 3D human pose estimation is formulated as a detection problem of the desired list of body joints. Most of the previous methods simultaneously localize and identify joints, such as localization of left shoulder, with the expectation that the accomplishment of one task would facilitate the accomplishment

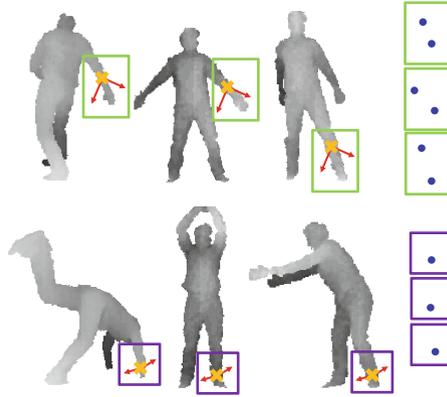


Fig. 1. Similar local depth features are exhibited by different body parts at various poses. Thus, a local depth feature may be insufficiently distinctive to distinguish the identity of parts

of other [2–5]. By contrast, we assert in this paper that identification hampers localization; therefore, the two tasks are treated separately and significant performance boost is obtained.

In depth images, similar local patch patterns are easily observed from different body parts at various poses. For example, in Fig. 1, right arm of the first person, left arm of the second person, and left leg of the third person in the first row shows similar depth map features. Similarly, in the second row, right hand of the first person and left foot of the second and third person also share similar depth features. Therefore, if one tries to localize a particular joint of interest by primarily relying on local depth features, then a confusion between the parts may arise [2, 3, 5]. On the contrary, for a given 3D configuration of joints and possible pose candidates, people can easily infer the corresponding pose and distinguish which joints are matched to body parts such as head, knees, wrists and etc., as shown in Fig. 2. Thus, the configuration of joints itself without labels already provides us a general idea of the body pose.

Based on the observation, we propose a new two-stage framework that first localize joints and then assign identity to the discovered joint positions. This strategy avoids the possible confusion between parts of similar depth maps in the localization step by postponing the body part label assignment to the identification step, wherein the global configuration of joints can be exploited.

In particular, a regression tree that predicts the relative displacements of the K nearest joints from a given input pixel position is trained in the localization step. During the test, all the joint locations are predicted by aggregating votes cast by the foreground pixels. Considering that each leaf node may contain similar offset vectors from different body parts as well as different poses, it can effectively pin-point joints from rare or unconventional poses, by sharing a prediction model across different parts. As a result, in the experiments, our

method shows enhanced accuracy compared to previous methods in particular for wrists, elbows, ankles, and knees.

In the identification step, each joint position is identified for a given 3D configuration of detected body joints. As shown in Fig. 2, a configuration of joints serves as a distinctive representation of a corresponding pose. Therefore, the nearest exemplar is retrieved from the training set based on a simple distance measure between point sets, and joint labels are transferred from this exemplar. This simple strategy works sufficiently well enough to achieve the upper-bound accuracy on the available dataset and effective enough to be applied to a real-time operation.

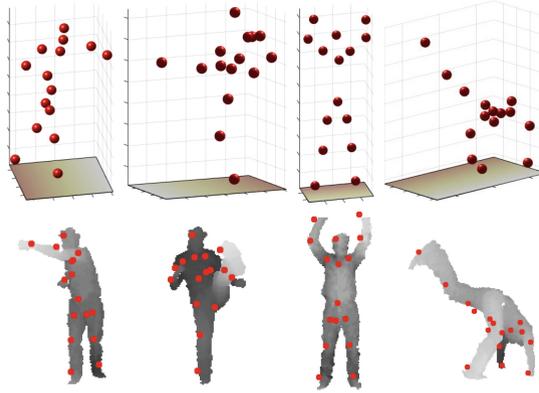


Fig. 2. 3D configuration of joints itself provides a general idea of a pose. People can easily find similar pose from possible candidates (below) and distinguish the joints, even when only a 3D joint configuration is given without body joint labels (above)

In summary, the contributions of this paper are as follows. The 3D body pose estimation problem is formulated into two subsequent problems, namely, localization and identification, and a new two-stage pose estimation algorithm is proposed to solve them. This approach makes both problem easier to solve, thus enabling improvement for both tasks.

2 Related Works

Existing 3D human pose estimation methods can be classified into generative, discriminative and combined approach according to the presence of a body model. Generative approaches [6–10] estimate pose by finding point correspondence between the input depth map and the known body model. These approaches usually require an accurate body model and good initialization to avoid being trapped in a local optimum. Recently, Ye et al. [11] proposed a fast and accurate generative method by representing a body as a Gaussian mixture

model. On the contrary, discriminative approaches do not assume a generative model and directly estimate the location of body joints. Random forest [12, 13] based approaches have achieved impressive performances in terms of speed and accuracy by formulating 3D body pose estimation as pixel-wise classification [2], offset regression [3], and random walk [5]. These approaches estimate the location of each individual joint independently while, ignoring the spatial dependence between them. To exploit global information, Sun et al. [4] proposed a conditional regression forest that considers dependence between joints through a global latent variable. All of these approaches focus on improving localization accuracy of a given target joint, with the use of various cues including local and global appearances.

Our method is also related to some works on 3D body/hand pose estimation in which the extracted geometric extreme points are used as rough but fast global configurations of body/hand poses. Baak et al. [14] used Dijkstra algorithm to compute geodesic distances and consequently extract extreme points from a point cloud to retrieve the nearest pose exemplar from the database. Liang et al. [15] adapted the same idea to hand pose estimation. Qian et al. [16] also initialized their tracking based hand pose estimation by detecting fingers without identification. The weakness of the approaches using extreme points, however, is apparent when desired parts, such as hands and feet, do not correspond to the extreme points in the depth maps, for example, folded hands. Therefore, all of the previous methods used detected extreme points to estimate coarse global configuration, whereas elaborate localization of joints are conducted simultaneously with identification

Works from Plagemann et al. [17] is related to our approach as they also first extracted interesting points and then classified them into specific parts or background. However, their interest point detector targeted on general interest points rather than body joints. Given that several interesting points could also be detected from the background, they used a classifier based on local features to classify the points, rather than using a global joint configuration. In addition, their method also exhibits the limitation of the aforementioned approaches using extreme points. Agarwal and Triggs [18] detected points by shape descriptor vectors instead of the extreme points, however detected points do not directly corresponds to the location of joints. By contrast, our approach targets directly the detection of body joints, thus enabling the use of the global configuration of joints for identification.

The body joint localization without identification can also be interpreted as semantic saliency detection [19, 20]. The idea of limiting the search space by finding the points of interest shares the same motivation behind saliency detection. In our problem, saliency regions or the points of interest are defined by semantically identifiable body joints.

3 Overview

We address the 3D human pose estimation problem from a single depth image. We formulate this problem as two separate sequential problems, namely,

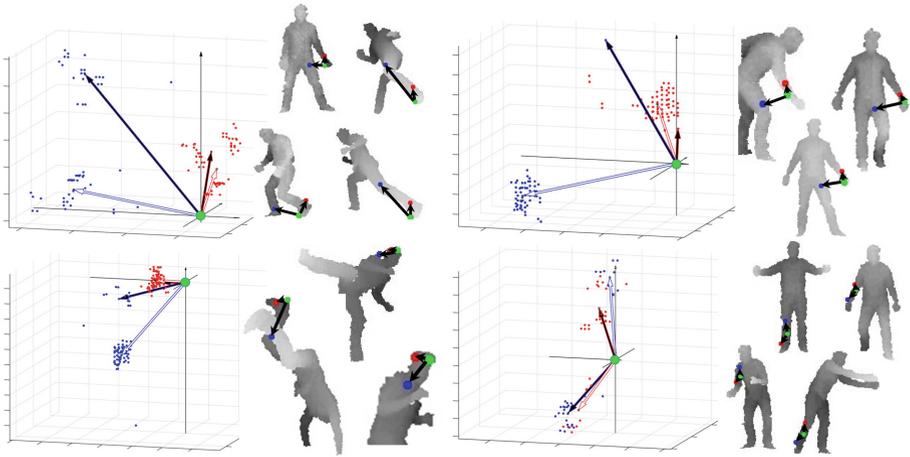


Fig. 3. The offset distributions in leaf nodes are shown for 2-nearest joints ($K = 2$). In each pair of images, the left image shows the distribution of offset vectors contained in a leaf node and the right image illustrates the corresponding training images in the leaf node. In both the left and right image, input pixel positions are marked as green circles, and the offsets to the nearest and the second nearest joints are respectively represented by red and blue dots. The average offset vectors for unidentified joint are drawn with arrows. Note that many of leaves contain pixels from different body parts as well as different poses. (Best viewed in color) (Color figure online)

localization and identification. The goal of the localization step is to discover every joint location in the 3D world coordinate system without identifying labels. Our method localizes joints based on a regression forest and k -means clustering. The details of training and inference are explained in Sect. 4. In the identification step, labels are assigned to each of the joint locations discovered in the previous step. Joints are identified by transferring labels from the nearest training sample, whose joint configuration is closest to the estimated configuration. The detailed procedures for exemplar set construction and nearest pose retrieval are described in Sect. 5. Section 6 presents the evaluation of our method on the publicly available dataset and provide the comparison with two existing methods. We also present in Sect. 6 the analyses of the effect of the proposed localization and identification decomposition and that of the system parameters.

4 Body Joint Localization Without Identification

We establish a variant of the Hough forest [21] to localize body joints, following the approach of Girshick et al. [3]. However, in our proposed method, a regression tree is trained to estimates offset vectors to the nearest K joints of any body parts, whereas in [3] the forest was trained to regress offset vectors to the specific body joints. In the test stage, every foreground pixel in the input depth image

traverses the regression tree and casts K votes in the 3D world coordinate system. The joints are localized by aggregating the votes cast by the regression tree.

4.1 Training Set Collection

Let us denote a set of body depth images by $\{I^1, I^2, \dots\}$ and the corresponding ground truth poses by $\{P^1, P^2, \dots\}$. Each pose P is represented by a set of 15 skeletal body joint positions in the world coordinate system: $P = (p_1, p_2, \dots, p_{15})$, where $p_j = (x, y, z)$, $j = 1, \dots, 15$. The corresponding projected pixel coordinate position of p in the depth image I and its depth are represented by $\tilde{p} = (\tilde{x}, \tilde{y})$ and $d_I(\tilde{p})$, respectively. For a given depth image, the corresponding points are reconstructed in the world coordinate system using the camera calibration parameters.

For a given position q in a depth image of pose P^n , the location of the closest body joints from q in the world coordinate system is expressed as

$$\alpha_1^n(q) = \arg \min_{p \in P^n} \|p - q\|^2 \quad (1)$$

and the k -th nearest joint from q is found recursively as $\alpha_k^n(q)$. The offset vector from q to the k -th nearest joint $\alpha_k^n(q)$ is denoted by $\Delta_q^k = \alpha_k^n(q) - q$. A training sample S consists of a body depth image I , a pixel position \tilde{q} in the pixel coordinate system, and offset vectors to K nearest joints from q in the world coordinate system.

$$S = (I^n, \tilde{q}, \Delta_q^1, \dots, \Delta_q^K), \quad (2)$$

where K is the number of nearest body joints to consider.

4.2 Training Regression Tree

The goal of training is to find a regression tree that minimizes the sum of the variances of offset vectors Δ_q^k in the leaf nodes. The objective function is formulated as follows,

$$E^{reg}(\mathbf{Q}) = \sum_{Q_s \subset \mathbf{Q}} \sum_{(I^n, q) \in Q_s} \sum_{k=1}^K \|\Delta_q^k - \bar{\Delta}_{\alpha_k, s}\|^2, \quad (3)$$

where Q_s is a set of training samples contained in a leaf node s , $\mathbf{Q} = \{Q_s\}$ is a partition of the training samples, and $\bar{\Delta}_{\alpha_k, s} = \frac{1}{|Q_s|} \sum_{(I^n, q) \in Q_s} \Delta_q^k$ is the mean offset vector of all the offsets in Q_s . For notational simplicity, we slightly abused the notation $(I^n, q) \in Q_s$ to represent $S = (I^n, \tilde{q}, \Delta_q^1, \dots, \Delta_q^K) \in Q_s$.

The following depth difference comparison features, similar to those in [2], are employed for splitting at the nodes:

$$f_\theta(I, \tilde{q}) = d_I\left(\tilde{q} + \frac{\tilde{t}_1}{d_I(\tilde{q})}\right) - d_I\left(\tilde{q} + \frac{\tilde{t}_2}{d_I(\tilde{q})}\right), \quad (4)$$

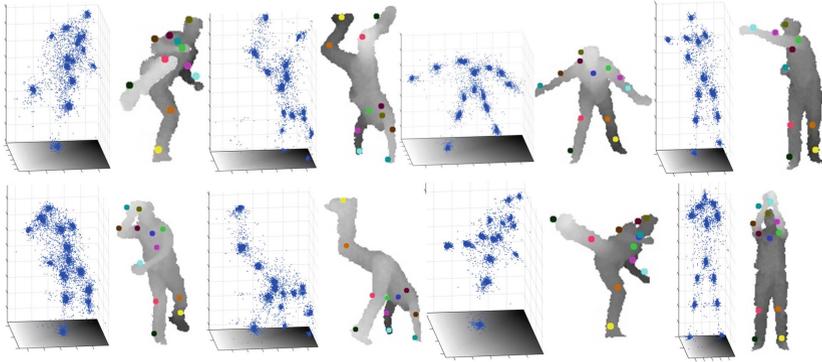


Fig. 4. Qualitative results of the proposed method. In each pair of images, the left side visualizes a voting result and the right side shows the estimated joint positions on the input depth image. Estimated locations of joints are indicated by circles

where parameters $\theta = (\tilde{t}_1, \tilde{t}_2)$ are offsets from the current pixel position \tilde{q} . Given that \tilde{t}_1 and \tilde{t}_2 are the values in the pixel coordinate system, they are divided by depth to provide the same relative position in the world coordinate system at different distances from the depth sensor.

The standard greedy decision tree algorithm is used to train a tree structure and to obtain the parameters for node splitting [12]. In each node, samples are iteratively separated into left and right children using a weak binary classifier until the termination conditions are satisfied. A pool of binary tests consisting of $\phi = (\theta, \tau) = (\tilde{t}_1, \tilde{t}_2, \tau)$ with random values of $\tilde{t}_1, \tilde{t}_2, \tau$, where τ is a threshold, is generated to train a weak learner. At each node, all the binary tests in the pool are evaluated and the ϕ^* that minimizes the following objective is selected:

$$\phi^* = \arg \min_{\phi} \sum_{s \in \{l,r\}} \sum_{(I^n, q) \in Q_s} \sum_{k=1}^K \|\Delta_q^k - \bar{\Delta}_{\alpha_k, s}\|^2, \tag{5}$$

where Q is a set of training samples reached at the current split node, $Q_l(\phi) = \{S = (I^n, q) \mid f_{\theta}(I, \tilde{q}) < \tau, S \in Q\}$ and $Q_r(\phi) = Q \setminus Q_l(\phi)$. The objective measures the uncertainty of leaf node prediction models by the sum of offset variations.

For a set of training samples reached at each leaf node, representative values are stored in the leaf node. Each set of offset vectors pointing to k -th nearest joint are cluster them into two clusters by $k = 2$ -means clustering. A leaf node s is represented by K pairs of two cluster centers and two corresponding relative cluster sizes as

$$\left(\left(\frac{|C_{1,1}|}{|Q_s|}, \bar{\Delta}_{\alpha_1, C_{1,1}}, \frac{|C_{1,2}|}{|Q_s|}, \bar{\Delta}_{\alpha_1, C_{1,2}} \right), \dots, \left(\frac{|C_{K,1}|}{|Q_s|}, \bar{\Delta}_{\alpha_K, C_{K,1}}, \frac{|C_{K,2}|}{|Q_s|}, \bar{\Delta}_{\alpha_K, C_{K,2}} \right) \right). \tag{6}$$

where $C_{k,1}$ and $C_{k,2}$ are sets of training samples clustered into the first and second cluster, respectively, based on the k -th nearest joints. $\bar{\Delta}_{\alpha_k, C_{k,i}}$ is the average of offset vectors from k -th nearest joints, included in the i -th cluster $C_{k,i}$. Figure 3 shows some examples of leaf nodes and the training samples they represent. The leaf nodes contain offsets for different body parts at different poses. See Fig. 3 for the detailed description.

4.3 Inference

During the test time, each foreground pixel of an input depth image passes through the trained regression tree (Sect. 4.2), until a leaf node s is reached. From each of K pairs in Eq. (6), an element is randomly sampled according to the probability proportional to the weight $|C_{K,i}|/|Q_s|$. Then, pixel casts votes to K positions in the 3D world coordinate where the positions are obtained by adding offset vectors to its own location. There are 15 different body joints, therefore, 15 regions with dense votes are expected to be generated, which correspond to the joint locations. Accordingly, k -means clustering [22] is used to find cluster centers which represent joint locations. Some example results of voting are shown in Fig. 4. To make the clustering more reliable, the outlier votes are excluded, which lies outside of the depth image image when projected to the pixel coordinate, before the clustering. Also, the computation time for k -means clustering is reduced by sub-sampling n_s pixels. The detailed performance for various values of n_s is tested in the Sect. 6.3. Note that there is still a room for further improvement of both speed and accuracy by using an advanced method to find a set of local clusters.

5 Body Joint Identification

Once the joint locations are discovered in the localization step, each joint is labeled in the identification step by retrieving the nearest training sample as an exemplar. Then, the labels from the exemplar is simply transferred. For the exemplar retrieval, the distance between a set of joints $P' = \{p'_1, p'_2, \dots, p'_{15}\}$ and pose $P = (p_1, p_2, \dots, p_{15})$ can be measured by rigidly aligning the exemplar pose to the localized points.

$$d(P, P') = \sum_{j=1}^{15} \|s\mathbf{R}(p_j - \Delta c) - (p'_{\sigma(j)} - \Delta c)\|^2. \quad (7)$$

$\sigma(\cdot)$ is the assignment function and Δc is the offset vector between the centers of foreground pixels. Respectively, s and \mathbf{R} are scale factor and rotation matrix that minimizes the sum of squared error between two set of points [23]. We use $\sigma(j) = \arg \min_{j' \in \{1, \dots, 15\}} \|s\mathbf{R}(p_j - \Delta c) - (p_{j'} - \Delta c)\|^2$ which assigns a nearest discovered joint to each of known joint in an exemplar. The subscript of $\{p'_1, p'_2, \dots, p'_{15}\}$ is used here only for distinction in the equation, while having

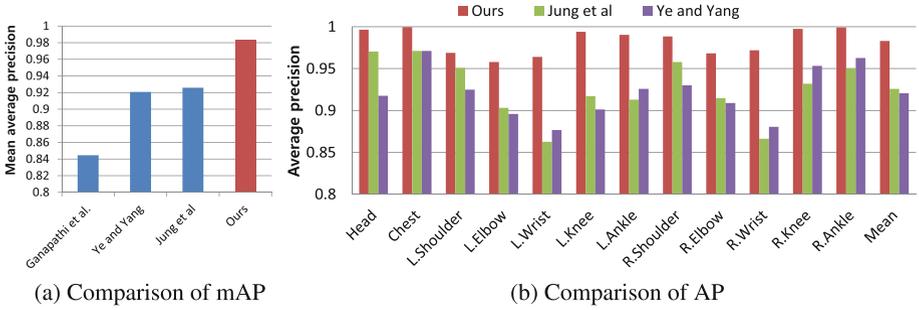


Fig. 5. Comparison of body joint detection accuracy on the EVAL dataset

no relation with joint labels. For EVAL [24] dataset, however, only a minor accuracy difference is found between the least squared solution and simple translation invariant point matching solution, and the simpler solution was implemented.

A set of exemplars from a subset of training set is constructed and a nearest exemplar is retrieve. In order to diversify poses contained in the exemplars, well scattered points [25] in the pose space are selected iteratively. In the first iteration, a random sample is selected from the training set. In each subsequent iteration, a sample whose pose is farthest from those of previously selected sample set is selected. For the distance between a set of poses \mathcal{P} and a pose P , the minimum distance is used, $d(\mathcal{P}, P) = \min_{R \in \mathcal{P}} \sum_{i=1}^{15} \|p_i - r_i\|^2$, where $P = (p_1, p_2, \dots, p_{15})$ and $R = (r_1, r_2, \dots, r_{15})$.

In our experiments, the simple retrieval strategy almost achieves the upper bound of average precision (AP) with 1000 exemplars on the EVAL dataset (Sect. 6). This performance implies that the intermediate representation of joint configuration is adequately discriminate to find the reliable nearest neighbor. We only used the configuration of joints for the retrieval, without any depth information. We believe incorporating depth information as a feature for each joint label would further increase the accuracy by helping discriminate confusing poses, which will have the same set of joint locations but with different labels. However, we leave this aspect as future extension.

6 Experimental Results

The evaluation of the proposed method consists of three parts. In the first part, we evaluate our method on the EVAL [24] dataset. We compare the performance of our method with two previous 3D pose estimation algorithms, namely, the method of Jung et al. [5] and Ye and Yang [11]. In the second part, we analyze each step of our algorithm; the localization step the and identification step. Finally, we investigate the effect of various system parameters on accuracy.

6.1 Comparison with Existing Methods

We evaluated our method on the publicly available EVAL dataset [24] using leave-one-out cross validation, as done in the previous works [5, 11]. The EVAL

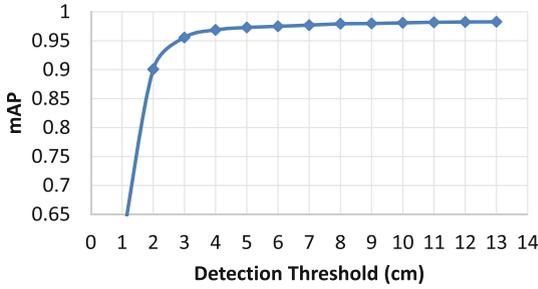


Fig. 6. mAP values for various detection thresholds

dataset consists of 24 sequences, which are obtained from 3 different people performing 8 actions. The sequences from two people are used in the training set, whereas the sequenced from excluded person is used as test set. Final accuracy is averaged over all three people. We used the conventional 10 cm rule as an evaluation measure [5, 11]. If the distance between the estimated and ground truth joint is less than 10 cm, then the estimation is considered to be correct. As pose estimation is considered as the detection problem of every joints, AP is used to quantify the accuracy for each joint, and the mean average precision (mAP) of joints is used as the overall accuracy measure. The runtime is measured on a PC with Intel Core i5-2500 3.3GHz CPU.

The accuracy comparison of our proposed method with those of Jung et al. [5] and Ye and Yang [11] is shown in Fig. 5(a). The proposed method achieved an mAP of 98.3%, which is 5.7% and 6.2% higher compared to 92.6% and 92.1% of [5] and [11], respectively. As shown in Fig. 5(b), our method outperforms the previous works in every joint. In particular, our method improves accuracy of second best methods by 8.7%, 7.1%, 6.3% and 5.4% for wrists, knees, ankles, and elbows, respectively. The accuracy boosts are prominent for the joints with large articulation, whereas AP of other joints, head, chest and shoulder, are 2.6%, 2.8%, and 2.4% higher than the second best method, respectively. Figure 4 shows some qualitative results of our method. Each circle indicates the estimated positions of joints.

We further evaluated our method on the EVAL dataset while varying the detection threshold from 1 cm to 13 cm. The mAP values are shown in Fig. 6. The curve steeply increases, showing 95.6% mAP at 3 cm threshold. Our approach was able to precisely estimate the joint locations within 3 cm radius for 95.6% of time. This is higher accuracy percentage even compared to the 10 cm rule of the previous methods.

6.2 Algorithm Analysis

In this section, we evaluate the accuracy of the localization step and that of the identification step separately to investigate the effect of problem decomposition.

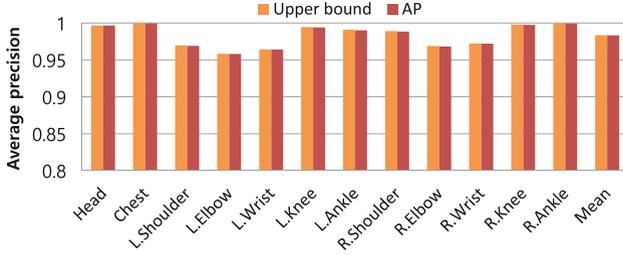


Fig. 7. Comparison of the upper bound and AP for each joint

When the joint localization is considered as a preprocessing step for the joint detection, the upper bound of AP serves as a meaningful indication of localization accuracy. On the other hand, the relative value of AP with respect to the upper bound indicates the accuracy of identification step. Each ground truth joint should be considered correctly detected if at least one of the estimated joints exists within the distance threshold(10 cm) to obtain the upper bound. Upper bound matching allows for multiple corresponding matches for a single joint. Ideally, the upper bound accuracy is achieved if every joint is properly labeled.

Figure 7 shows the upper bound and AP for each joint. In our method, the difference between AP and upper bound is smaller than 0.1 %. The simple exemplar matching method is sufficient to deal with the translation and scale variations in the EVAL set. To deal with drastic view point and pose variations, we should consider a pose tracking approach with temporal prior.

Table 1. Runtime speed comparison

Alg.	Ours	Jung et al. [5]	Ye and Yang [11]
Runtime speed (fps)	37	1262	>30

Table 2. Runtime analysis

Proces	Voting (Sect. 4.3)	Clustering (Sect. 4.3)	Labeling (Sect. 5)	Total
Runtime (ms)	13 (48 %)	10 (37 %)	4 (15 %)	27 (100 %)

The runtime speed comparison is shown in Table 1. Our method performs in real-time (37 fps) under a single-core CPU operation. Ye and Yang achieves real-time with GPU and Jung et al. is extremely fast with its random walk strategy. The computation time used in each process of our method is shown in Table 2. Most of the computation time is spent during the localization step, occupying 85% of the entire process, where the offset generation from regression tree and *k*-means clustering take about the same computational resources.

6.3 Effect of the Parameters

In this section, we analyze the effects of different parameters on the performance of our method. We performed the same experiment in Sect. 6.1 on the EVAL dataset while varying the number of training samples, the number of nearest joints, the number of minimum samples allowed at each leaf node, the number of subsampled offset votes, and the number of exemplars.

Figure 8(a) shows the AP values for each joint at varying the number of training samples during regression tree construction. Our method achieves 92.8% mAP using 500 training images, where each images contains about 8000 foreground pixels. Head, knees, and ankles require a smaller number of training samples while wrists, elbows, and shoulders need a larger number of samples to achieve the same accuracy.

We also evaluated our method at different numbers of K nearest joints used in the localization step. A high K increases the number of votes per joint as well as the computational cost and memory, while making the voting results more stable. However, with large K , the offsets reach farther away joints with higher uncertainty. Each leaf node is also less likely to contain samples from different parts and poses; thus, the effect of sharing different joints is reduced. The result is shown in Fig 8(b). When two-nearest joints are used, the accuracy is higher for elbows and wrists, but lower for shoulders with respect to the accuracy when three-nearest joints are used. We used $K = 2$ for all other experiments settings.

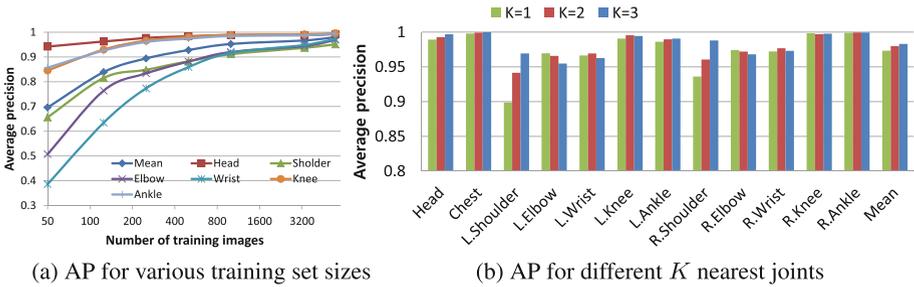


Fig. 8. Effect of various system parameters

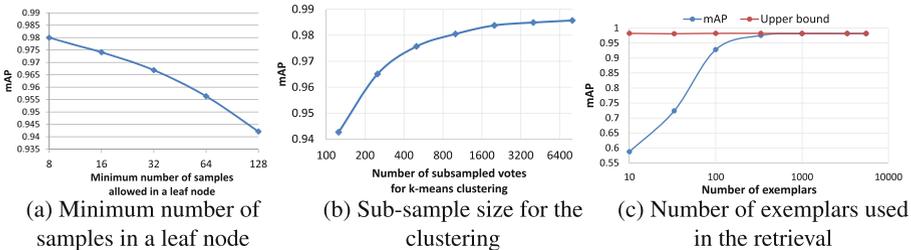


Fig. 9. Effect of various system parameters

Figure 9(a) shows the effect of the number of minimum samples n_{min} allowed in each leaf node with unlimited tree depth when $K = 2$. Allowing a smaller number of samples increases the memory required for the model while better fitting the training set. As the value of n_{min} varies from 128 to 8, mAP consistently increases. We used $n_{min} = 8$ for all the other experiments.

Figure 9(b) shows the mAP values at varying sub-sample sizes before k -means clustering. As the clustering step consumes approximately 40 % of the total runtime of our method, the sub-sample size directly affects both speed and accuracy. With only 125 votes, our algorithm achieves 94.3 % mAP, which is higher than those of the previous methods. We used sub-sampled 1000 votes for k -means clustering for all the other experiments.

Figure 9(c) shows the mAP values at varying numbers of exemplars used in the identification step. With only 100 templates, our method achieved 93.4 % mAP, which is comparable to those of the state-of-the-art methods. We used 1000 exemplars in all the other experiments.

7 Conclusion

In this paper, we propose a new two-stage framework for 3D human pose estimation from a single depth image. Unlike existing approaches that address localization and identification of body joints as a single task, our proposed method decomposes the problem into two subproblems, namely, localization without identification and identity assignment. These subproblems are then solved sequentially. Our method effectively overcomes the confusion between body parts and joints with large articulation by sharing the offset prediction model across different body parts and poses. The experimental results show that our method exhibits significantly improved performance compared with the previous methods. It achieved high accuracy particularly for wrists, elbows, ankles, and knees, which are usually considered difficult to localized.

There are a number of different implementation improvements we may consider in the future. First, a more advanced offset clustering scheme can be used. We currently use the k -means clustering algorithm which may require large number of iterations and has high dependency on the initial points. There are various advanced cluster and mode seeking methods available for replacing the k -mean algorithm. Second, the simple least squared error point matching algorithm many not be robust to large view point and pose variations. RANSAC like algorithm can be used to deal with large view changes, as well as considering time sequenced pose estimation which uses previous frame's pose estimation as the initial pose of current frame.

Acknowledgments. This work was supported by Hankuk University of Foreign Studies Research Fund of 2016.

References

1. Romero, J., Kjelstrom, H., Kragic, D.: Monocular real-time 3D articulated hand pose estimation. In: *Humanoids* (2009)
2. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from a single depth image. In: *CVPR* (2011)
3. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient regression of general-activity human poses from depth images. In: *ICCV* (2011)
4. Sun, M., Kohli, P., Shotton, J.: Conditional regression forests for human pose estimation. In: *CVPR* (2012)
5. Yub Jung, H., Lee, S., Seok Heo, Y., Dong Yun, I.: Random tree walk toward instantaneous 3d human pose estimation. In: *CVPR* (2015)
6. Wei, X., Zhang, P., Chai, J.: Accurate realtime full-body motion capture using a single depth camera. In: *SIGGRAPH ASIA* (2012)
7. Helten, T., Baak, A., Bharaj, G., Muller, M., Seidel, H., Theobalt, C.: Personalization and evaluation of a real-time depth-based full body tracker. In: *3DV* (2014)
8. Gall, J., Stoll, C., de Auiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: *CVPR* (2009)
9. Grest, D., Krüger, V., Koch, R.: Single view motion tracking by depth and Silhouette information. In: Ersbøll, B.K., Pedersen, K.S. (eds.) *SCIA 2007*. LNCS, vol. 4522, pp. 719–729. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-73040-8_73](https://doi.org/10.1007/978-3-540-73040-8_73)
10. Ionescu, C., Carreira, J., Sminchisescu, C.: Iterated second-order label sensitive pooling for 3d human pose estimation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1661–1668 (2014)
11. Ye, M., Yang, R.: Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In: *CVPR* (2014)
12. Criminisi, A., Shotton, J.: *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media (2013)
13. Breiman, L.: Random forest. *Mach. Learn.* **45**, 5–32 (2001)
14. Baak, A., Müller, M., Bharaj, G., Seidel, H.P., Theobalt, C.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In: *ICCV* (2011)
15. Liang, H., Yuan, J., Thalmann, D., Zhang, Z.: Model-based hand pose estimation via spatial-temporal hand parsing and 3d fingertip localization. In: *The Visual Computer* (2013)
16. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: *CVPR* (2014)
17. Plogemann, C., Ganapathi, V., Koller, D., Thrun, S.: Real-time identification and localization of body parts from depth images. In: *ICRA* (2010)
18. Agarwal, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, vol. 2, p. II-882. IEEE (2004)
19. Zhang, Z., Liu, Z., Zhang, Z., Zhao, Q.: Semantic saliency driven camera control for personal remote collaboration. In: *2008 IEEE 10th Workshop on Multimedia Signal Processing* (2008)
20. Chang, X., Yang, Y., Xing, E., Yu, Y.: Complex event detection using semantic saliency and nearly-isotonic SVM. In: *ICML* (2015)
21. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: *PAMI* (2009)

22. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. *Appl. Stat.* **28**(1), 100–108 (1979)
23. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. In: *Pattern Analysis and Machine Intelligence* (1991)
24. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real-time human pose tracking from range data. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7577, pp. 738–751. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-33783-3_53](https://doi.org/10.1007/978-3-642-33783-3_53)
25. Guha, S., Rastogi, R., Shim, K.: Cure: an efficient clustering algorithm for large databases. In: *ACM SIGMOD Record* (1998)