Latent variable pictorial structure for human pose estimation on depth images

Li He[a], Guijin Wang[a,*], Qingmin Liao[b], Jing-Hao Xue[c]

[a] *Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*
[b] *Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua Campus, Xili university town, Shenzhen 518055, China*
[c] *Department of Statistical Science, University College London, London WC1E 6BT, UK*

Abstract

Prior models of human pose play a key role in state-of-the-art techniques for monocular pose estimation. However, a simple Gaussian model cannot represent well the prior knowledge of the pose diversity on depth images. In this paper, we develop a latent variable-based prior model by introducing a latent variable into the general pictorial structure. Two key characteristics of our model (we call Latent Variable Pictorial Structure) are as follows: (1) it adaptively adopts prior pose models based on the estimated value of the latent variable; and (2) it enables the learning of a more accurate part classifier. Experimental results demonstrate that the proposed method outperforms other state-of-the-art methods in recognition rate on the public datasets.

Keywords: Pose estimation, Pictorial structure, Latent variable, Body silhouette, Regression forest, Depth images.

*Corresponding author. Tel.: +86-18911389502; Fax: +86-62770317
Email addresses: `l-he10@mails.tsinghua.edu.cn` (Li He), `wangguijin@tsinghua.edu.cn` (Guijin Wang), `liaoqm@tsinghua.edu.cn` (Qingmin Liao), `jinghao.xue@ucl.ac.uk` (Jing-Hao Xue)

1. Introduction

Human pose estimation [1, 1, 2] is widely applied in human-computer interaction, smart video surveillance, health care, etc. Although a lot of efforts have been devoted to the research of pose estimation, it remains a very challenging problem in computer vision because of occlusion, high dimensionality of the search space and high variability in people's appearance.

The depth image obtained by the depth sensor [3, 4, 5] can provide 2.5D scene geometry, which facilitates both the segmentation of human body from background and the disambiguation of similar poses. Recently, the focus of pose estimation [6, 7, 8, 9] has been shifted toward pose estimation on depth images. Most of these works can be divided into two categories: generative methods and discriminative methods.

Typical generative methods include the proposals in [10, 9, 11, 12], in which a kinematic chain and a 3D surface mesh are built as the human body model. They treat the depth image as a point cloud over 3D space and apply a model-fitting algorithm, such as the iterative closest point (ICP), to the human body model to fit the 3D point cloud. Ye et al. [11], Ganapathi et al. [12] and Baak et al. [9] combine dataset searching and model fitting to approach the problem of 3D pose estimation. Ganapathi et al. [10] extend the ICP to an articulated model by enforcing constraints over the pose space. Although such methods do not need a training step, they suffer many drawbacks. For example, the accuracy depends on the surface mesh level [13] and the fitting usually needs long processing and inconvenient setups.

Compared with the generative methods, the discriminative methods do not iteratively fit models to the observed data. Rather they directly estimate the parameters about pose. Thus they can estimate the pose quickly and adapt to various conditions. They regard the human pose as a collection of different parts/joints and learn discriminative classifiers for the part/joint detection [6, 8, 7, 14]. The most famous works on depth images are those based on random forest [6, 8, 7]. Shotton et al. [6] formulate the pose estimation as a classification task and use the random forests to learn the classifiers. Girshick et al. [8] convert the classification task to the regression problem for the estimation of the occluded parts. In [7], Sun et al. incorporate temporary states of the object, such as person's height and facing direction, to boost the performance of the classifiers. However, these methods infer locations of body joints either independently [6, 8] or relying on some global information [7], neglecting the dependence between body joints.

It is natural to boost the pose estimation performance by adding constraints among joints. One of the most widely used approach in this direction is to use graph model-based prior structure, which was first proposed in [15] for general computer vision problems and later applied to the pose estimation problem in [16]. It assumes that the relationships among joints are state-constrained among the body parts. Two important components are defined in the model: one is the appearance model which represents the probability of a body part at a particular location in the given image; the other is the prior model which represents the probability distribution over pose space. To make a trade-off between computational efficiency and estimation accuracy, tree-structured models with a single Gaussian prior are commonly used [15, 16, 17, 18]. However, as the diversity of human pose increases, a simple Gaussian prior usually leads to a poor model of human articulation, which cannot be applied well to the tasks on the depth images. This is mainly due to two reasons. One is that it is not an easy work to find a proper kernel number for the Gaussian model in a large dataset. A small number may cause a poor fitting of the prior, while a large number will cost extra computation and is prone to over-fitting. The other is that the method always applies the same prior model to test samples, even when they are of distinct poses. This limits the adaptability of the method. The works in [19, 20] cluster poses into sub-clusters and learn a GMM for each sub-cluster to enhance the adaptability of prior model. However, at the inference stage, they need to infer all possible poses and select one as the final output. This makes the inference complex.

In this paper, we propose a novel framework called Latent Variable Pictorial Structure (LVPS) for pose estimation on depth images. We construct and estimate a latent variable based on the human silhouette. At the inference stage, our model rebuilds the appearance model and the prior model based on the values of the latent variable and then infers human poses. We shall show its effectiveness through experiments on public datasets. Compared with the state-of-the-art methods, our proposal can significantly increase the accuracy of pose estimation.

The rest of the paper is organized as follows. We overview the proposal in Section 2. Our LVPS model is introduced in Section 3 and its application to the pose estimation in Section 4. We present experiments and discussions in Section 5 and draw conclusions in Section 6.

2. Overview of the proposed method

Fig. 1 shows the framework of our LVPS. It consists of two main processes: the training stage indicated by green arrows and the inference stage indicated by blue arrows.

The training stage. The keys of the training stage involve generation and selection of the latent variable and the training of models. In our work, we extract silhouette features of poses, obtain their distributions, quantize the distributions into a set of states C, and use the state label as the latent variable. According to the value of the latent variable, all the training samples are partitioned into subsets. After that, we attach the value of the latent variable to each sample and treat each sample as a two-labels object: a body part label and a latent variable state. Samples with labels are then input into classifiers to learn appearance models and prior models. As a result, the diversity of the appearance and prior in each cluster would be reduced and the prior model can be better learned and the discrimination ability of the appearance model can be largely enhanced.

The inference stage. As the blue arrows indicate, to estimate one body pose on depth image I, we shall first evaluate its latent state. This is, the likelihood $p(c_i|I)$ is estimated. After that we rebuild our prior model and appearance model by assembling the learned models of individual clusters according to the likelihoods. As a result, our proposal adapts the models based on the specific test image.

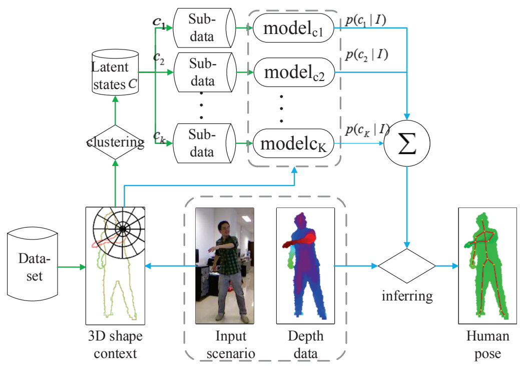

Figure 1: The flowchart of the proposed method: the process with green arrows is the training stage and that with blue arrows is the inference stage.

3. Latent variable pictorial structure

A classical pictorial structure model of the human body was proposed in [15]. It assumes that the dependences between body joints can be expressed by a predefined graph, $G = (V, E)$, as shown in Fig 2, where V and E denote the sets of nodes and edges in the graph G, respectively. We use $X = \{x_1, x_2, ...\}$ to denote the pose, in which x_i denotes the position of joint i. For the detection of an articular object, the objective function to be

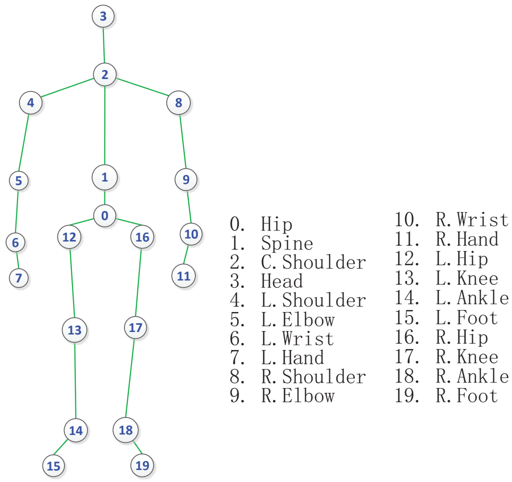

Figure 2: The graph model on human pose. The circle with a number is a vertex in V, which presents a joint/part of the body; the line between two joints is an edge in E, which indicates that the connected joints/parts are dependent.

maximized when given image I can be written as

$$p^{PS}(X|I) \propto \left\{ \prod_{i \in V} \phi(x_i|I) \right\} \left\{ \prod_{(i,j) \in E} \phi(x_i, x_j) \right\}, \tag{1}$$

where $\phi(x_i|I)$ denotes the appearance likelihood, which models the probability of a part at a particular location and orientation given the input image I, and the factor $\phi(x_i, x_j)$ denotes a prior, which models the probability distribution over pose space. In this paper, the factor $\phi(x_i, x_j)$ describes the distribution of relative position between joint i and joint j.

In most existing methods based on the general pictorial structure model, only one tree-structured Gaussian prior is used to speed up the inference, and the appearance models of individual parts are learned independently. This leads to a prior of low descriptive ability and an appearance model which cannot capture the multi-modal appearance of body parts, e.g. the different appearances of a body part in different views.

To overcome these issues, we incorporate a latent variable into the general pictorial structure to propose a latent variable pictorial structure model (LVPS). Specifically, we utilize the discrete state of the latent variable to partition samples and the pose space. Hence the diversity of the appearance and prior in each cluster would be reduced, which results in more effective and reliable appearance and prior models at the cluster level than the global models. Besides, clustering over the latent variable feature space leads to a simple classifier. We use c to denote the discrete latent variable, $C = (c_1, \ldots, c_K)$ to denote the set of the K states of the latent variable, and $p(c_k|I)$ to denote the probability of the state c_k given image I.

Then based on the latent structure we obtain the posterior probability of pose X as

$$p^{LVPS}(X|I) \propto \sum_{c_k \in C} \left\{ p^{PS}(X|c_k, I) p(c_k|I) \right\}, \tag{2}$$

where $p^{PS}(X|c_k, I)$ denotes the posterior probability conditional on the specific cluster corresponding to c_k. The latent variable c may encode any desirable properties of the target objects. In this paper, we propose to utilize it to encode the whole human pose through body silhouette.

The inference stage is show in Fig 3. To the given image I, we first extract its latent variable value $Hist(I)$, which has a form of histogram of silhouette

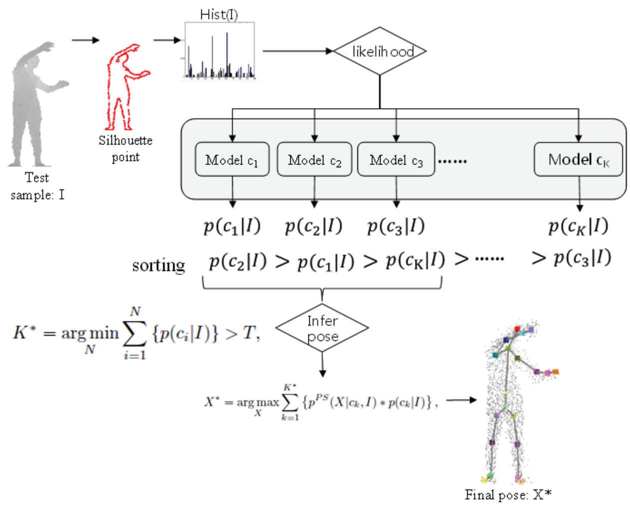

Figure 3: The flowchart of inferring a human pose X from the given image I.

features in this paper. Then the likelihood is evaluated between $Hist(I)$ and those of sub-models. We sort the sub-models in descending order of the likelihood $p(c|I)$ and the first K^* sub-models that have their total likelihood beyond threshold T are selected. At last, a linear strategy is used to build the final detection model for the pose inference using (3):

$$\begin{aligned} X^* &= \arg\max_X \sum_{k=1}^{K^*} \{p^{PS}(X|c_k, I) * p(c_k|I)\}, \\ K^* &= \arg\min_N \sum_{i=1}^{N} \{p(c_i|I)\} > T, \end{aligned} \tag{3}$$

where K^* is the number of selected sub-model, T is the threshold and N is a variable for counting sub-models. In this way we can adjust the number of the models used for the test sample, and its effect can be shown in the experiments in Section 5.4.

4. Details of LVPS

This section describes how the LVPS models are implemented for human pose estimation. Since the samples are partitioned into subsets based on the value of the latent variable, the variation of the pose space is decomposed and a pose subspace can be better modeled even with a simple model. As a result, two main parts will be discussed in the following: the generation and selection of the latent variable and the learning of the appearance models.

4.1. The latent variable

A simple way to model the variation of the pose space is to cluster poses directly in the pose space as in [20]. However, they have to learn a multinomial logistic regression to classify each cluster. Another way is to use some properties of the object [7, 21], such as torso orientation, person's height or facing direction. These features are natural, but they are not much associated with the pose as a whole.

In our proposal, we extract a kind of silhouette feature to represent the pose and use such feature to build our latent variable and to cluster our samples. The most commonly used silhouette feature to represent a pose is the shape context feature, which was first proposed in [22] for shape matching and then used for human pose estimation [23, 24]. However, the silhouette features extracted from RGB/grey images cannot represent the 3D structure of pose. So He et al. [25] extend the 2D shape context [23] to 3D space. To the best of our knowledge, existing pose estimation methods only use silhouette feature to learn maps from the feature space to the pose space, rather than to build latent variables to boost the prior model.

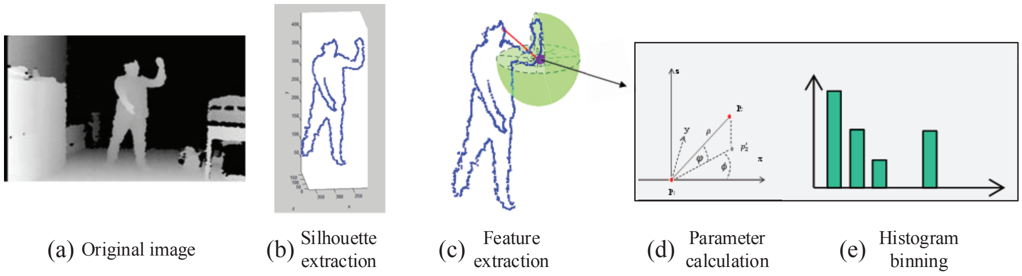

Figure 4: Extraction of shape context feature in [25]: (a) is the original depth image; (b) is the result after silhouette extraction; (c) shows how to extract the shape context feature on point p_1; (d) calculates the offset parameters between p_1 and any other points on the silhouette; (e) shows the building of histogram of the shape context feature.

In the following, we discuss how to generate and select the latent variable using the feature proposed in [25].

One brief flowchart of feature extraction [25] is shown in Fig 4. First, a sequence of edge points are extracted on each depth image. Then, for each edge point, the offsets between it and other points are calculated and voted into a histogram. This histogram encodes local pose information by collecting offsets on the edge points and is called shape context feature. At last, one pose is encoded by a bag of shape context features. More details about the feature extraction can be found in [25]. However, with such a bag of features, it is computationally consuming to compare two images.

To tackle this issue, we use the method in [23] to align these shape context features, which will be then used to form a feature vector for the construction of our latent variable. Specifically, we run k-means on the shape context features from all the training samples to obtain B quantized centers. To represent one pose, we softly vote the shape context features of one image onto these learned centers with Gaussian weights. Finally, each pose on a depth image can be represented by a B-dimensional feature vector f. In the experiments, we set B to 100 as with [23]. Feature vectors of some samples are shown in Fig. 5.

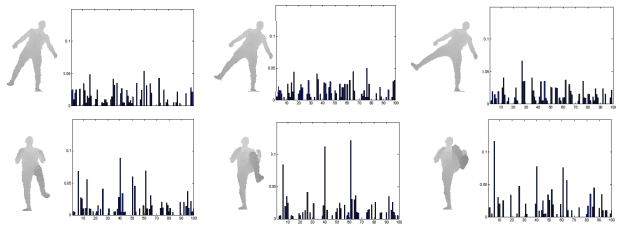

Figure 5: Some samples and their feature vectors f.

The feature vector f encodes the silhouette of body and can capture richer pose information than some straight properties, such as torso orientation and persons height. To quantize a feature vector further, we perform another k-means algorithm to obtain K discrete states (i.e. cluster labels) as the values of the feature vectors. We adopt the cluster label as the latent variable. After that, we can partition the training data into K subsets based on the value of the latent variable and can estimate the likelihood that image I belongs to the kth cluster c_k by using a simple histogram distance as

$$p(c_k|I) \propto 1/dst(Hist(I), Hist(c_k)), \tag{4}$$

where $dst(Hist(I), Hist(c_k))$ indicates the distance between two histograms

$Hist(I)$ and $Hist(c_k)$.

We show some average poses of individual clusters in Fig. 6. From Fig. 6, we can find that by clustering poses through the mid-level representation we can encode pose states and reduce the pose diversity in each cluster. For example, in Fig. 6, clusters (1), (4) and (6) show the hands changes while (1), (2) and (5) focus on the facing direction. In Section 5, more samples are shown in the experiments and the value of sub-model number K and its influence on the performance will be discussed.

(1) (2) (3) (4) (5) (6)

Figure 6: Six average poses of individual clusters: our latent variable encodes pose states and reduce the pose diversity in each cluster. Average poses (1), (4) and (6) show the hands changes while (1), (2) and (5) focus on the facing direction.

4.2. Learning of the model

Random forests [26] have been proved as an effect and efficient algorithm for human pose estimation on depth images. This section introduces how to learn the structure of random forest and the corresponding parameters of appearance models. We learn the structure of random forest based on the method in [8], but different from [8], we treat each sample as a two-label structure.

Overview of random forest. Random forest $\Gamma = \{T_t\}$ is a collection of randomized decision trees T_t. Each tree T_t is built on a randomly selected subset of training samples and learns a mapping from a sampled point to parameter space Θ. For the classification task, the parameter space is the label set, indicating the body part, and for the regression task, it may be R^3 in our case. To learn the structure of tree T_t, the selected samples corresponding to tree T_t will be iteratively divided into two separated subsets by a binary splitting function ζ. The splitting function ζ could be simple comparison of feature values and its threshold is generated randomly. The best one of the splitting functions will be chosen by maximizing the information gain. We use $S = \{s_i\}$ to denote the set of the training samples and S_L, S_R for the two split subsets. As a result, the destination function can be written as

$$\zeta^* = \arg\min_{\zeta} g(\zeta), \tag{5}$$

$$g(\zeta) = H(S) - \sum_{i \in \{L,R\}} \frac{|S_i|}{|S|} H(S_i), \tag{6}$$

where $H(\cdot)$ is the entropy or the sum-of-squared-differences depending on the specific task. This splitting continues recursively until the stop criteria are met, e.g. the tree reaches the maximal depth or there are less than a minimum number of samples in set S.

Learning tree structures. We treat each pixel labeled by a body part on the depth image as a sample and use random forest for the multi-label classification task. If each sample subset is used to train each sub-model independently, the complexity of the final model will increase at least linearly in number of states of the latent variable. To address this issue, we employ a shared-structure model to train the random forest. We see each sample (pixel) as a multi-tag object $s_i = (f_i, l_i, c_i)$, where f_i refers to features, l_i refers to the body part label and c_i refers to the latent state. To fit the multi-tag samples, we adjust the expression of entropy $H(S)$ to be

$$H(S) = \sum_{c \in C} H(S_c), \tag{7}$$

$$H(S_c) = -\sum_{l_i} p_{l_i,c} \log(p_{l_i,c}), \tag{8}$$

where $H(S_c)$ is the entropy from the sample subset under the same latent state c, and $p_{l_i,c}$ is the probability of the sample with the label l_i in the subset. We adopt the depth comparison features proposed in [6], then the splitting function ζ for sample s could be:

$$\zeta(s; k, \eta) = \begin{cases} 0, & \text{if } f_s(k) < \eta \ , \\ 1, & \text{otherwise} \ . \end{cases} \tag{9}$$

where $f_s(k)$ is the kth value in the depth comparsion features and η is the random threshold.

Parameters of appearance model. At each leaf ι of a tree we learn a compact expression $p(x_i|\iota, c_k)$ of votes for the position x_i conditional on the value of latent variable c_k. Specifically, for each sample set with latent variable c_k, a mean-shift algorithm with a Gaussian kernel is applied to cluster the relative votes which present the offsets from the sampled position to the body part. The largest M centers $\{\Delta_{\iota m c_k}\}$ are stored at leaf node ι with a confidence weight $w_{\iota m c_k}$ which is equal to the size of the cluster. As a result, the conditional distribution $p(x_i|\iota, c_k)$ can be expressed by using the Gaussian Parzen density estimators as:

$$p(x_i|c_k, \iota) \propto \sum_{m \in M} \mathrm{w}_{\iota m c_k} \exp(-\frac{\|x_i - (\Delta_{\iota m c_k} + x_s)\|^2}{b^2}), \tag{10}$$

where x_s is the 3D location of sampled point s, b is the kernel bandwidth and we set an empirical value 0.05m in the experiments. While (10) models the probability for a voting element arriving at the leaf ι of a single tree, the probability over the forest is calculated by averaging over all trees,

$$\phi(x_i|c_k) \propto \frac{1}{|T|} \sum_{T_t \in T} p(x|c_k, \iota_t), \tag{11}$$

where ι_t is the corresponding leaf of tree T_t in the forest.

Parameters of prior model. Besides learning parameters of the appearance model at each leaf ι, we also learn a compact expression $p(\Delta_{ij}|\iota, c_k)$ of the relative position between joints i and j conditional on the value of latent variable c_k using the similar method as that in the learning of appearance parameters. We use $\{\Delta_{ij,\iota m c_k}\}$ to denote the learned centers of the relative position between joints i and j by mean-shift algorithm and $w_{ij,\iota m c_k}$

to denote its weight. So, the relative position distribution between joints i and j conditional on the leaf ι and latent variable c_k can be expressed as

$$p(x_i, x_j|c_k, \iota) \propto \sum_{m \in M} \mathrm{w}_{\mathrm{ij},\iota m c_k} \exp\left(-\frac{\left\|x_i - x_j - \Delta_{ij,mc_k}\right\|^2}{{b_{ij}}^2}\right), \tag{12}$$

where x_i and x_j are the estimated positions of joints i and j, and b_{ij} is the kernel bandwidth, which we set to the average limb length in the training data. As a result, the probability of the forest is calculated by averaging over all trees,

$$\phi(x_i, x_j|c_k) \propto \frac{1}{|T|} \sum_{T_t \in T} p(x_i, x_j|c_k, \iota). \tag{13}$$

Compared with the Gaussian prior model, our prior model builds its expression using specific sampling points on each test image. This would enhance adaptability of a prior model.

5. Experiments and Discussion

5.1. Datasets

In this section, we evaluate our algorithm for human pose estimation on two depth datasets, the Stanford dataset [12] and our THU pose dataset.

The Stanford dataset. It consists of 28 action sequences of one person, which includes 7891 images in total with a resolution of 176×144. The images were captured by using a ToF camera in a lab environment and joint positions are obtained by motion sensors. Among the images, 6000 are selected for training and the rest, less than 2000, are for testing.

The THU dataset2. To further evaluate our method, we collect a new dataset for experiments. Our THU dataset2 contains 15000 depth images captured by a Kinect camera, which consists of 5 persons performing general actions (including upper/lower limbs movements, turning, jumping, etc.). Some samples are shown in Fig. 7. We use motion detection method, such as [27, 28], to get the foreground manually labeled landmarks as the ground truth. Among the images, 10000 are randomly selected for training and the rest are for testing.

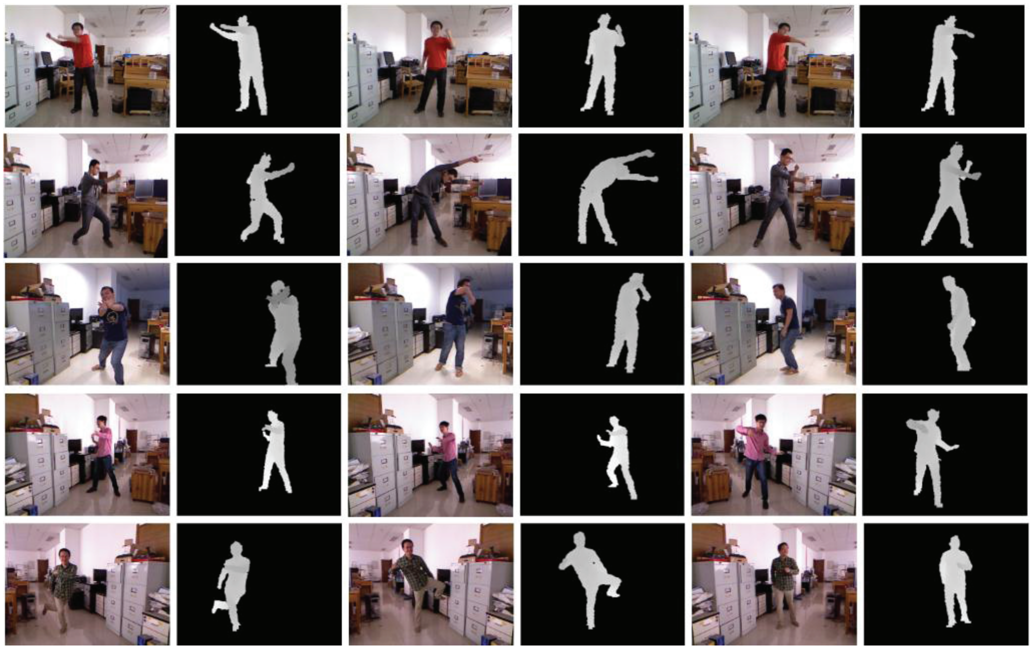

Figure 7: Samples from the THU dataset2: RGB and depth images.

5.2. Preprocessing of the training data

We assume the foreground is clear in our model. So to ensure this, some preprocessing should be done before training. We perform a motion-based method [29] to segment the foreground from background. Some segmentation results in the Stanford dataset are shown in Fig. 8. Besides, the baseline method [8] used in this paper needs to label the pixels for each part. It involves a great deal of work. To facilitate this, we label each pixel as the nearest body part.

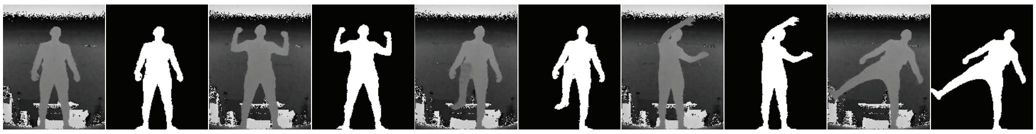

Figure 8: Results of foreground segmentation of [25] in the Stanford dataset: pairs of original and foreground images.

5.3. Performance evaluation

To evaluate our algorithm, we compare our proposed method with some state-of-the-art methods in [8, 30, 12, 11]. Two measures are used to demonstrate the performance: the average error and the mean of average precision (mAP). The average error for each joint evaluates the average difference between the estimated position and its ground truth under the Euclidean space and the mAP presents the ratio of the most confident joint hypothesis within the distance tolerance $\tau = 0.1$m, as with [8]. For the specific joint i, its mAP can be calculated by

$$mAP_i = \frac{1}{M}\sum_{m=1}^{M} 1(|\hat{x}_i(m) - x_i(m)| < \tau), \tag{14}$$

where M is the number of testing samples, $\hat{x}_i(m)$ is the estimated position of joint i, $x_i(m)$ is the ground-truth and $1(\cdot)$ is an indicator function.

Experiments on the Stanford Dataset. Considering the sample size and pose variation in this dataset, we set K to 4 ($\|C\| = 4$), the centers of the four clusters are shown in Fig. 9, and we set $T = 0.2$ for the inference stage. The influence of the cluster number K and the threshold T will be discussed in Section 5.4.

On this dataset, we compare our method with some state-of-the-art methods [8, 30, 12, 11]. The experimental results are shown in the second column of Table 1. We can observe that compared with the published results, our method obtains a better result, the mAP of 98.2%. Some of the estimated results are illustrated in Fig. 10. From Fig. 10, it can be found that our method can get good results for the most samples with a front-facing angle and with a small side-facing angle. We note that it fails under a large side-facing angle, the results are shown in the black box in Fig. 10. It is a challenging task to estimate human pose within a side-standing body. The first result in the black box shows that our method fails to estimate the part on the right body due to a large area occlusion. The second result in the black box shows that our method makes a symmetric error because it cannot recognize a back-facing body on this depth image. To overcome this issue, methods in [31, 32] on sequence of images or some tracking methods in [33] may help. Additionally, we test the speed of our algorithm in processing one image on the Stanford dataset. With our non-optimized code, it runs the processing at about 36fps on our 4-cores computer. This would be fast enough for many visual interaction tasks.

Table 1: Comparison of mAP ($\tau = 0.1$m) with some state-of-the-art methods.

Method	On Stanford dataset	On THU dataset2
Ganapathi et al. [12]	0.898	–
Ye et al. [11]	0.950	–
Shotton et al. [6]	0.947	–
Girshick et al. [8]	0.957	0.89
He15 [25]	0.98	0.88
ours	**0.982**	**0.971**


(1) (2) (3) (4)

Figure 9: The centers of clusters on the Stanford dataset.

Experiments on the THU Dataset2. For this dataset, we set K to 16 and the average poses are shown in Fig. 11. For each cluster, we use the method in [25] to train the random forest. The rest of the settings are the same as that on the Stanford dataset.

We compare our approach to a state-of-the-art method proposed by Girshick et al. [8] and the method in [25]. They both estimate the joint locations

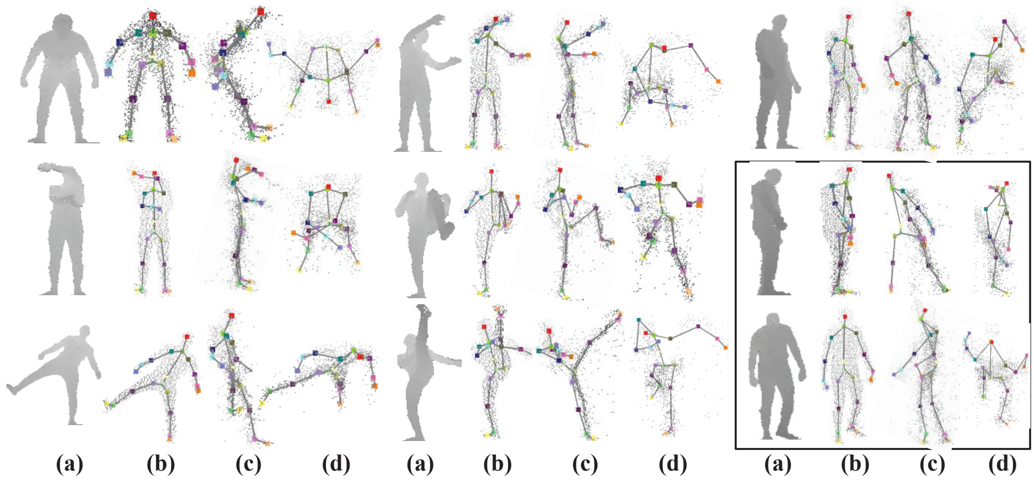

Figure 10: Nine estimated results on the Stanford dataset: (a) the original depth image; (b), (c) and (d) our results from the front view, the left-side view and the top view, respectively. Results in the box are those that our method fails.

by regression forest. Experimental results are shown in the third column of Table 1. The detailed comparison of the approach [8], denoted by 'Girshick et al.', and our LVPS, denoted by 'ours', is shown in Fig. 12. From the Fig. 12, we can find that our algorithm achieves better results than that of [8]. More specifically, our algorithm obtains 3.6cm in the average error and 97.1% in mAP. Besides, the superior results can be remarkably observed at limb ends, such as elbow, wrist and hand, which we think benefits from the use of latent models and the graphical models. Compared with the method [25], our method yields a better result. By the way, the method [25] can be seen as the case that $K = 1$ the proposed algorithm. Some samples are illustrated in Fig. 13.

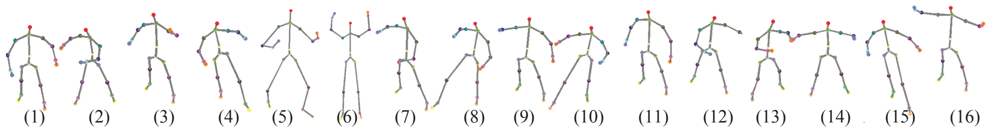

Figure 11: The centers of clusters on the THU dataset2.

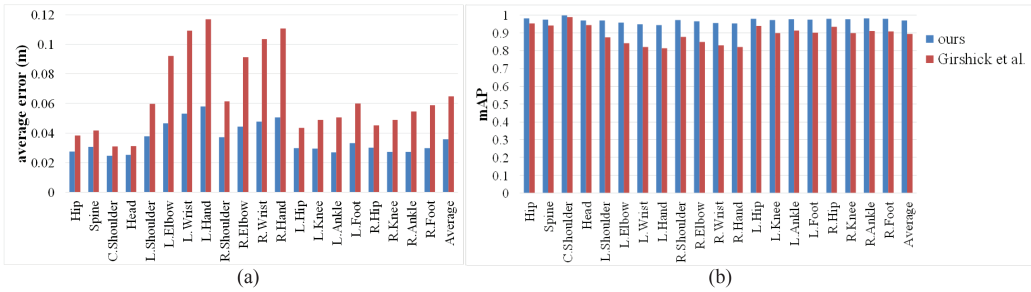

Figure 12: Performance on the THU dataset2: (a) average estimation error vs. body joint; (b) mAP vs. body joint.

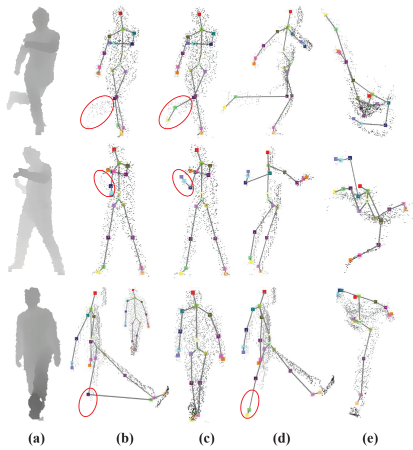

Figure 13: Three estimated sample images from the method in [8] and ours: (a) the original depth image; (b) results from the method [8]; (c), (d) and (e) our results from the front view, the left-side view and the top view, respectively.

5.4. Discussion

In this section we investigate the effects of three main factors that may affect the pose estimation accuracy of our method. These factors are the cluster number $K = \|C\|$, the construction of the inference model and the threshold T.

Cluster Number K. We retrain our models with different cluster numbers K from 1 to 32 on both datasets. The results are shown in Fig. 14. On the THU dataset2, when K is increased from 1 to 16, the value of mAP is enhanced from about 0.88 to 0.97 and after that it drops. It illustrates that the larger the cluster number K is, the better the models are learned, but if K is too large, it causes over-fitting. On the Stanford dataset, splitting the pose space does not boost the performance. We think the small diversity of the pose on the Stanford dataset causes this. Nevertheless, when K is equal to 1, the method can be seen as the method [25]. Compared with it, we can observe the superiority of our method.

Construction of Inference Model. In the inference stage, we use a linear strategy (3) to construct the detection model. We compare our strategy in (3) with another usual strategy: using a fixed value of K^* (K^*=1 and 2) for inference, denoted by '$K^* = 1$' and '$K^* = 2$'. K^*=1 means the most plausible sub-model is used for inference while K^*=2 means that two sub-models with the largest likelihoods are linearly combined for inference. The results are shown in Table 2. We can observe that our proposal obtains the best result among these combining methods, which indicates the effectiveness

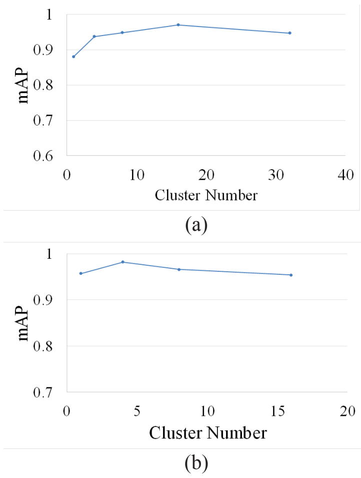

Figure 14: mAP vs. cluster number: (a) results on the THU dataset2, (b) results on the Stanford dataset.

of our method. Moreover, it can be observed that the results of '$K^* = 1$' and '$K^* = 2$' are very close. It implies that our latent variable is discriminative to cluster the pose.

Table 2: Performance of various combining strategies.

Method	mAP ($\tau = 0.1$m)
$K^* = 1$	0.956
$K^* = 2$	0.962
ours	**0.97**

Threshold T. The threshold T controls the number of sub-models used for inference. We investigate the pose estimation performance under different thresholds T and show the results in Fig. 15. Although it yields the best results at $T = 0.2$, it still maintains an mAP of higher than 0.9 for other values of T, which indicates the robustness of our model. Additionally, in order to further show how the threshold T works, we calculate the proportions of cluster numbers used for inference in Fig. 16. It demonstrates that as the threshold T goes up, there are more clusters used for inference. Before T reaches 0.2, only the most probable model is used to estimate the human pose. After that, more and more models are involved in the inference. Considering both the results in Fig. 15 and Fig. 16, we find that merging the proper number of models can improve the performance.

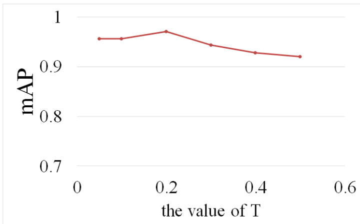

Figure 15: mAP vs. the threshold T.

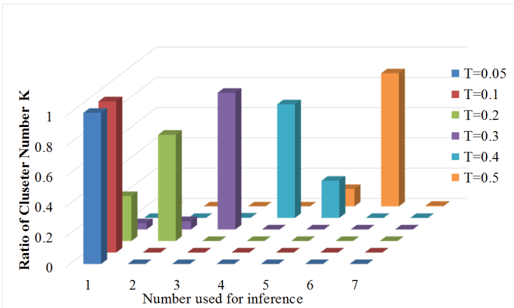

Figure 16: The proportions of cluster numbers used for inference under different thresholds T: different colors indicate the values of T.

6. Conclusion and Future Work

In this paper, we have proposed a novel approach to pose estimation on depth images. In the approach, we have proposed the latent variable pictorial structure (LVPS) to adapt the prior model and enhance the discrimination of the appearance model by incorporating a latent variable. We have also modified the silhouette features to encode the human pose, clustered the pose space and established a new pose dataset to evaluate the performance of the proposed method. Through these enhancements, our LVPS model can learn better appearance and prior models. Experiments have verified that the proposed method could achieve higher accuracy on the published datasets, compared with other state-of-the-art methods. It would be interesting to further our work by combining this method with object tracking.

7. Acknowledgments

We were very grateful to the two reviewers for their professional comments to our manuscript and their constructive suggestions for its revision. This work was partially sponsored by NSFC 61271390 and by the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (No. ZDSYS201405091729599 & No. YJ20130402145002441).

References

[1] T. B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Computer vision and image understanding 81 (3) (2001)

231–268.

[2] Z. Hu, G. Wang, X. Lin, H. Yan, Recovery of upper body poses in static images based on joints detection, Pattern Recognition Letters 30 (5) (2009) 503–512.

[3] A. Kolb, E. Barth, R. Koch, R. Larsen, Time-of-flight sensors in computer graphics, in: Proc. Eurographics (State-of-the-Art Report), 2009, pp. 119–134.

[4] G. Wang, X. Yin, X. Pei, C. Shi, Depth estimation for speckle projection system using progressive reliable points growing matching, Applied Optics 52 (3) (2013) 516–524.

[5] C. Shi, G. Wang, X. Yin, X. Pei, B. He, X. Lin, High-accuracy stereo matching based on adaptive ground control points, Image Processing, IEEE Transactions on 24 (4) (2015) 1412–1423.

[6] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Communications of the ACM 56 (1) (2013) 116–124.

[7] M. Sun, P. Kohli, J. Shotton, Conditional regression forests for human pose estimation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3394–3401.

[8] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, A. Fitzgibbon, Efficient regression of general-activity human poses from depth images, in: Computer Vision (ICCV), IEEE International Conference on, IEEE, 2011, pp. 415–422.

[9] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, C. Theobalt, A data-driven approach for real-time full body pose reconstruction from a depth camera, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 71–98.

[10] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun, Real-time human pose tracking from range data, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 738–751.

[11] M. Ye, X. Wang, R. Yang, L. Ren, M. Pollefeys, Accurate 3d pose estimation from a single depth image, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 731–738.

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

[13] E. De Aguiar, C. Theobalt, C. Stoll, H.-P. Seidel, Marker-less deformable mesh tracking for human shape and motion capture, in: Computer Vi-

sion and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[14] S.-Z. Su, Z.-H. Liu, S.-P. Xu, S.-Z. Li, R. Ji, Sparse auto-encoder based feature learning for human body detection in depth image, Signal Processing 112 (2015) 43–52.

[15] P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.

[16] M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: People detection and articulated pose estimation, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1014–1021.

[17] M. Sun, S. Savarese, Articulated part-based model for joint object detection and pose estimation, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 723–730.

[18] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1385–1392.

[19] S. Johnson, M. Everingham, Learning effective human pose estimation from inaccurate annotation, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1465–1472.

[20] S. Johnson, M. Everingham, Clustered pose and nonlinear appearance models for human pose estimation., in: BMVC, Vol. 2, 2010, p. 5.

[21] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9) (2010) 1627–1645.

[22] S. G. Salve, K. Jondhale, Shape matching and object recognition using shape contexts, in: Computer Science and Information Technology (ICCSIT), IEEE International Conference on, Vol. 9, IEEE, 2010, pp. 471–474.

[23] A. Agarwal, B. Triggs, Recovering 3D human pose from monocular images, Pattern Analysis and Machine Intelligence, IEEE Transactions on 28 (1) (2006) 44–58.

[24] C. Ek, P. Torr, N. Lawrence, Gaussian process latent variable models for human pose estimation, in: MLMI'07, 2007, pp. 132–143.

[25] L. He, G. Wang, Q. Liao, J.-H. Xue, Depth-images-based pose estimation using regression forests and graphical models, Neurocomputing 164 (2015) 210–219.

[26] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[27] W. Lin, M.-T. Sun, H. Li, Z. Chen, W. Li, B. Zhou, Macroblock classi-

fication method for video applications involving motions, Broadcasting, IEEE Transactions on 58 (1) (2012) 34–46.

[28] X. Han, G. Li, W. Lin, X. Su, H. Li, H. Yang, H. Wei, Periodic motion detection with roi-based similarity measure and extrema-based reference-frame selection, in: Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, IEEE, 2012, pp. 1–4.

[29] Y. Li, G. Wang, X. Lin, G. Cheng, Real-time depth-based segmentation and tracking of multiple objects, in: Technology in Automation, Control, and Intelligent Systems (CYBER), IEEE Conference on, IEEE, 2012, pp. 429–433.

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

[31] T. Pfister, J. Charles, A. Zisserman, Flowing convnets for human pose estimation in videos, In Proc. ICCV, 2015.

[32] B. Tekin, X. Sun, X. Wang, V. Lepetit, P. Fua, Predicting people's 3D poses from short sequences, arXiv preprint arXiv:1504.08200, 2015.

[33] A. Yao, X. Lin, G. Wang, S. Yu, A compact association of particle

filtering and kernel based object tracking, Pattern Recognition 45 (7) (2012) 2584–2597.