Depth Sweep Regression Forests for Estimating 3D Human Pose from Images

Ilya Kostrikov
ilya.kostrikov@rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Juergen Gall
gall@iai.uni-bonn.de
University of Bonn
Bonn, Germany

Abstract

In this work we address the problem of estimating the 3D human pose from a single RGB image, which is a challenging problem since different 3D poses may have similar 2D projections. Following the success of regression forests for 3D pose estimation from depth data or 2D pose estimation from RGB images, we extend regression forests to infer missing depth data of image features and 3D pose simultaneously. Since we do not observe depth for inference or training directly, we hypothesize the depth of the features by sweeping with a plane through the 3D volume of potential joint locations. The regression forests are then combined with a pictorial structure framework, which is extended to 3D. The approach is evaluated on two challenging benchmarks where state-of-the-art performance is achieved.

1 Introduction

Over decades estimating the human pose from still images has been an intensive research topic [19]. In recent years, the majority of research has been focused on estimating the 2D pose, *e.g.* [9, 15, 23, 26, 36, 37], since this is already very challenging. However, many applications require the 3D pose. While some approaches estimate first the 2D pose and then reconstruct the 3D pose from the 2D pose estimate [20], estimating the 3D pose directly from the images is more practical since it directly solves the problem at hand. For this task, discriminative approaches that learn a mapping from image features to 3D pose, *e.g.* [1, 6, 18, 22, 25, 27], have been most successful. These methods perform a nearest-neighbor search or regression without taking the skeletal structure of a human into account. This is in contrast to state-of-the-art human pose estimation approaches that rely on discriminative parts and combine them within a pictorial structure model [11, 12] that represents the human skeleton. A prominent example of these approaches is [39].

In this paper we address the problem of estimating the 3D pose from still images. However, instead of learning a regression from image features to the full pose, we regress the positions of the joints in 3D space and then infer the pose using a 3D pictorial structure framework. For regression, we rely on regression forests that have been shown to efficiently predict 2D pose from images [9] or 3D pose from depth data [29]. These approaches, however, cannot be directly applied since each local image or depth feature estimates the relative positions of the joints from the feature location. While the relative position is well defined

if feature and joint locations are given either in 2D or in 3D, it is not defined if the features are sampled from 2D images without depth information and the joint locations need to be predicted in a 3D world coordinate system. We therefore hypothesize the depth of the image features using a depth sweep approach. To this end, we sweep with a plane through the 3D volume of potential joint locations and use a regression forest to predict the relative 3D position of a joint given the hypothesized depth of the feature. The final pose estimate is then obtained by a 3D pictorial model.

In our experiments, we show that our approach achieves state-of-the-art performance on the Human3.6m benchmark [16] and the HumanEva-I dataset [30].

2 Related work

Decision forests. Recently, decision forests [8] and their modifications have been intensively used for a large number of applications in computer vision including human pose estimation [9, 14, 29, 34, 35]. The works most similar to our approach are [14, 29] that independently estimate 3D human body joint locations from depth images and [9] that estimates 2D pose from images by learning two layers of decision forests. These approaches were inspired by Hough forests [13] and Implicit Shape Models applied for human pose estimation [21].

Pictorial structure models. Pictorial structure models were originally presented in the seventies [12] and rapidly gained popularity in tasks of object detection and human pose estimation since 2005 when an efficient inference algorithm was introduced in [11]. Initially, simple models for body part templates were used, however, later pictorial stucture models were combined with more advanced discriminative classifiers as SVM [17], AdaBoost [3] and decision forests [9]. The efficient inference algorithm proposed in [11] represents relative parts offsets by a single Gaussian distribution. In order to overcome these limitations, mixtures of pictorial structure models [17] and mixtures of parts [38] were proposed. Furthermore, the pictorial structure framework was adapted for 3D human pose estimation from multiple views [2, 7]. In contrast to these approaches, we use the pictorial structure framework to estimate 3D human pose from a single view.

3D pose from RGB images. There is a variety of approaches that learn a mapping from the space of the image features to the space of 3D poses. In [1] features extracted from a human silhouette are evaluated together with different regression techniques. An approach based on fast Bayesian Mixture of Expert (fBME) combined with more advanced image features was introduced in [6]. This was improved by a Twin Gaussian Process (TGP) regression approach [5] that estimates relative human pose from features extracted from an image (HOG or HMAX).

Recently, researchers proposed methods that estimate 2D body joint locations using state-of-the-art 2D detectors and disambiguate the 3D poses that could produce such projections in a second step. Recovering 3D pose from 2D pose estimates, however, requires very strong prior knowledge on 3D poses. Over the recent years, researcher proposed a number of ways to incorporate this information. For example, the authors of [4, 10] used the temporal information in order to recover 3D poses, while in [32] a latent generative model was proposed that treats parameters of 3D pose as latent variables.

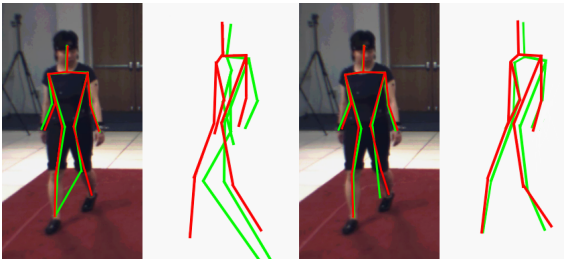

Figure 1: First, we estimate 3D joint location probabilities from a 2D image using the regression forests described in the paper. The two left images depict the front and side view of the inferred 3D pose (green) and compare it with the ground-truth (red). Second, we use a 3D pictorial structure framework in order to enforce kinematic constraints and improve the 3D pose estimates (right).

3 Method

Our approach consists of two parts as shown in Figure 1: first, we independently estimate joint 3D location probabilities; second, we use the estimated probabilities together with the pictorial structure framework in order to infer the full skeleton. For the first part, we propose depth sweep regression forests which are regression forests [13, 29] that hypothesize the missing depth information of image features are discussed in Section 5. For the second part, we extend the mixture of PSMs [17] for 3D inference. Since depth sweep regression forests belong to the family of random forests [8], we briefly describe a regression forest framework as used in [13, 29] and introduce relevant notation first.

4 Regression forests

In the context of pose estimation [9, 29], a regression tree represents a mapping from the space of image patches and patch locations $\mathcal{P} \times \Omega$ to the space of probabilities over joint locations $\mathcal{X}$. For a given patch P from location $\mathbf{y}$, which ends in leaf L of the tree, the probability of the location $\mathbf{x}$ of a joint j is then given by

$$p_j(\mathbf{x}|P,\mathbf{y}) = p(j|L(P,\mathbf{y}))p_j(\mathbf{d}(\mathbf{x},\mathbf{y})|L(P,\mathbf{y})), \tag{1}$$

where $p(j|L)$ denotes the class probability of joint j stored at leaf L and $p_j(\mathbf{d}|L)$ denotes the probability of relative locations of the joint j. In case of 2D pose estimation from images or 3D pose estimation from depth data or 3D point clouds, we have $\Omega \subset \mathbb{R}^d$, $\mathcal{X} \subset \mathbb{R}^d$ and $\mathbf{d}(\mathbf{x},\mathbf{y}) = \mathbf{x} - \mathbf{y}$.

For localizing a joint j, the probabilities of all trees of a forest are averaged and summed over all patches sampled from locations $\mathbf{y} \in \Omega$:

$$\phi_j(\mathbf{x}) = \sum_{\mathbf{y}\in\Omega} \frac{1}{|\mathcal{T}|} \sum_{T\in\mathcal{T}} p(j|L_T(P,\mathbf{y}))p_j(\mathbf{d}(\mathbf{x},\mathbf{y})|L_T(P,\mathbf{y})). \tag{2}$$

The functional $\phi_j(\mathbf{x})$ can be considered as a confidence measure of joint j being at location $\mathbf{x}$. The position can then estimated by $\hat{\mathbf{x}}_j = argmax_{\mathbf{x}\in\mathcal{X}}\, \phi_j(\mathbf{x})$. Since this can be efficiently implemented by a voting procedure combined with mean-shift, these forests are also called Hough forests [13].

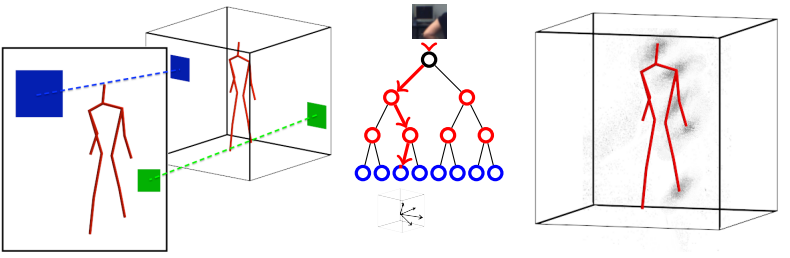

Figure 2: Illustration of a depth sweep regression forest for 3D pose estimation from a 2D image. **Left.** Patches sampled from different depths project onto the image with different scale. **Middle.** The projected patches traverse the tree evaluating splitting functions in the intermediate nodes (black and red) until they reach a leaf node (blue). A leaf node contains 3D offsets that point to locations of a joint with associated weights. **Right.** Based on the offsets, the patches sampled in the 3D volume cast 3D votes for several joint locations.

Each tree is trained with a set of patches sampled from a random subset of the training images annotated with the joint positions. In [9], patches are augmented by a class label where patches close to a joint are labeled by the joint and otherwise labeled as background. Each training sample is therefore an element of $\mathcal{P} \times \Omega \times J^- \times \mathcal{X}$, where J^- denotes the set of joints augmented by a background class. For the trees only the relative joint locations are used, *i.e.* a sample is defined by $(P, \mathbf{y}, j, \mathbf{d})$. The trees are trained recursively where a parametrized splitting function is selected at each node that optimizes the classification or regression performance. The training is continued until the maximum depth is reached or the number of training samples is below a threshold. At the leaves, the probabilities $p(j|L)$ and $p_j(\mathbf{d}|L)$ are computed given the class labels j and relative joint locations $\mathbf{d}$ of the training data arriving at the leaf L.

5 Depth sweep regression forests

In order to predict 3D joint locations from 2D images, the approach briefly described in Section 4 cannot be directly applied since $\Omega \subset \mathbb{R}^2$ and $\mathcal{X} \subset \mathbb{R}^3$. The relative location $\mathbf{d}$ of a 3D joint given the 2D location of a patch, and thus (2), are not defined. We therefore propose to perform the inference in $\Omega' \subset \mathbb{R}^3$ instead:

$$\phi_j^{ds}(\mathbf{x}) = \sum_{\mathbf{y}' \in \Omega'} \frac{1}{|\mathcal{T}|} \sum_{T \in \mathcal{T}} p\left(j|L_T(P,\mathbf{y}')\right) p_j\left(\mathbf{d}(\mathbf{x},\mathbf{y}')|L_T(P,\mathbf{y}')\right). \tag{3}$$

In this formulation $\mathbf{d}(\mathbf{x},\mathbf{y}') = \mathbf{x} - \mathbf{y}'$ is well defined, but the regression trees have to learn a mapping from $\mathcal{P} \times \Omega'$ to $\mathcal{X}$. This causes a problem since $\mathcal{P} \times \Omega'$ is not observed neither for training nor for testing, which is in contrast to other works that assume that depth is observed at least during training [33]. However, assuming that the camera projection π is known, which maps a point from Ω' to the image plane Ω, we can rephrase the problem as learning a mapping from $\mathcal{P} \times \Omega \times \mathcal{Z}$ to $\mathcal{X}$, where the appearance of a 2D patch P depends on the 2D image location and the depth z. Since we do not observe depth for training or testing, we hypothesize it by sweeping with a plane parallel to the image plane along the z-axis through a 3D volume. The patch P corresponding to the 3D point y' is then the patch centered at the projection $\pi(\mathbf{y}') \in \Omega$ and the leaf it ends depends on $z' \in \mathcal{Z}$:

$$\phi_j^{ds}(\mathbf{x}) = \sum_{\mathbf{y}' \in \Omega'} \frac{1}{|\mathcal{T}|} \sum_{T \in \mathcal{T}} p\left(j | L_T(P, \pi(\mathbf{y}'), z')\right) p_j\left(\mathbf{d}(\mathbf{x}, \mathbf{y}') | L_T(P, \pi(\mathbf{y}'), z')\right). \quad (4)$$

Since the appearance of patches changes for different depth values, the maximum of (4) corresponds to a set of patches that are associated to the correct hypothesized depth values and agree on the 3D joint location. The approach is illustrated in Figure 2. In the following, we describe our modifications to regression forests [9, 29] in order to perform inference as in (4).

5.1 Training samples

The training data consists of a set of 2D images with annotated 3D pose. For training the trees, we need samples $(P, \mathbf{y}, z, j, \mathbf{d})$ from $\mathcal{P} \times \Omega \times \mathcal{Z} \times J^- \times \mathcal{X}$, where J^- denotes the set of joints augmented by a background class and $\mathbf{x} \in \mathcal{X}$ are the 3D joint positions. For each sample $\mathbf{y}'$, we obtain z and $\mathbf{y} = \pi(\mathbf{y}')$ directly. For P we use the full training image represented by 13 feature channels I_k similar to [13]: (1) a normalized gray scale image; (2-4) Lab color space; (5-13) HOG-like features with 9 bins. The variation of a patch based on $\mathbf{y}'$ will be directly encoded by the splitting functions of the trees. For each joint j located at (x_j, y_j, z_j), we sample positive samples in the 3D neighbourhood of j. To this end, we use a 3D Gaussian with mean (x_j, y_j, z_j) and variance 0.1 of the upper body size computed as in [9]. The negative samples ($j = -1$) are sampled uniformly from the scene volume and selected if the distance to a joint is larger than 0.2 of the upper body size. In addition, we performed mining of hard negative samples [13], *i.e.*, after sampling negatives uniformly and training the first trees we select negative samples with a low background class probability. This step, however, resulted only in a minor improvement and can be omitted. While for positive samples $\mathbf{d} = (x_j, y_j, z_j) - \mathbf{y}'$, $\mathbf{d} = 0$ for negative samples.

For each joint, we sample 100 positive samples per training image. The number of negative samples per image is equivalent to all positive samples, *i.e.* $100|J|$.

5.2 Splitting functions

In contrast to 2D joint detection, we do not evaluate patches that correspond to different pixel locations of the image but rather a set of patches that corresponds to a discrete set of points inside a 3D volume. For this reason, we also need to take variations of the patches based on the depth into account. Given an image P with extracted feature channels I_k and a 3D point $\mathbf{y}' = (x, y, z)$ projected to $\mathbf{y} = \pi(\mathbf{y}')$, we define the family of splitting functions by

$$f_\theta(P, \mathbf{y}, z) = R_{I_k, \frac{w_1}{z}, \frac{h_1}{z}}\left(\mathbf{y} + \left(\frac{u_1}{z}, \frac{v_1}{z}\right)\right) - R_{I_k, \frac{w_2}{z}, \frac{h_2}{z}}\left(\mathbf{y} + \left(\frac{u_2}{z}, \frac{v_2}{z}\right)\right) > \tau, \quad (5)$$

$$\theta = (u_1, v_1, w_1, h_1, u_2, v_2, w_2, h_2, k, \tau), \quad (6)$$

where $R_{I,w,h}(\mathbf{y})$ is the average value of the feature channel I inside a $w \times h$ rectangular area centered at the point $\mathbf{y}$. (u_*, v_*) defines the offset for each rectangle from $\mathbf{y}$. Each splitting function is therefore parametrized by the 10D vector θ and depends on the image features, the 2D location of the patch, and the depth value z.

Given a set of training samples $(P, \mathbf{y}, z, j, \mathbf{d})$, 8 trees are trained as in Section 4 with maximum depth 20 and at least 20 samples per leaf.

5.3 Leaf probabilities

At the leaves, we store the probabilities $p(j|L_T)$ and $p_j(\mathbf{d}|L_T)$ based on the training samples $(P, \mathbf{y}, z, j, \mathbf{d})$ arriving at the leaf L_T of tree T. While $p(j|L_T)$ is computed based on the class labels j, $p_j(\mathbf{d}|L_T)$ is computed for each joint. As in [14], we cluster the relative displacement vectors $\mathbf{d}$ at the leaves for each joint j and use for each cluster k a Gaussian with the cluster centroid $\mathbf{d}_k^{L_T,j}$ as mean to model the probability:

$$p_j(\mathbf{d}|L_T) \propto \sum_k \exp\left(-\frac{\left\|\mathbf{d}-\mathbf{d}_k^{L_T,j}\right\|^2}{a_{T,j}}\right). \tag{7}$$

For each leaf and joint, we use 4 clusters. In contrast to [14], we learn the hyper-parameters $a_{T,j}$ for each tree T independently.

5.4 Inference

In order to sweep through the volume efficiently, we discretize the 3D space. Without loss of generalization, we assume that the center of the coordinate system is at the camera location and that the z-axis is perpendicular to the image plane. We use a uniform grid with voxel size h, *i.e.* for a given bounding volume $[x_a, x_b] \times [y_a, y_b] \times [z_a, z_b]$ the discretization is defined by

$$\Omega' = \{\mathbf{y}'_{ijk} = (x_a + i \cdot h, y_a + j \cdot h, z_a + k \cdot h) : (i,j,k) \in \mathbb{N}^3; (x_a, y_a, z_a) \leq \mathbf{y}'_{ijk} \leq (x_b, y_b, z_b)\}.$$

For inference, we estimate the joint probabilities only for points in Ω', *i.e.* $\mathcal{X} = \Omega'$. The impact of h is evaluated in the experimental section. It is worth to mention that the bounding volume is only needed for the discretization and does not need to be tight. We only assume that the joints are located inside of the volume.

Instead of taking the maximum of (4) as an estimate of the joint locations, we use a pictorial structure model as in [9] to infer the pose.

6 Pictorial Structure Models

Inferring 3D joint locations independently from 2D RGB images is prone to depth ambiguities. Many of the ambiguities, however, can be resolved by using a kinematic body model that provides information about constraints between joint locations. To this end, we use the well known pictorial structure framework [11] that provides accurate results while keeping the inference tractable. Similar to the work [9] for 2D pose estimation, we use a joint representation:

$$p(X_J|I,\vartheta) \propto \prod_{j \in J} \left(\phi_j^{ds}(\mathbf{x}_j)\right)^{\alpha} \prod_{(i,j) \in E} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j|\vartheta_{ij}), \tag{8}$$

where X_J are the 3D locations of all joints, I a given image, and ϑ_{ij} are the parameters for the binary terms. In our case, ψ_{ij} is modeled by 3D Gaussian distributions. In contrast to [9], we have an additional scaling term α. Since $\phi_j^{ds}(\mathbf{x}_j)$ is more spread in the 3D space than in the 2D space, we increase the peakiness of the unary terms by $\alpha = 5$. The impact of the additional scaling term is evaluated in the experimental section.

Contrary to [2, 7], we use a straightforward 3D extension of the original PSM framework. To this end, we perform a 3D distance transform in the discretized bounding volume and use

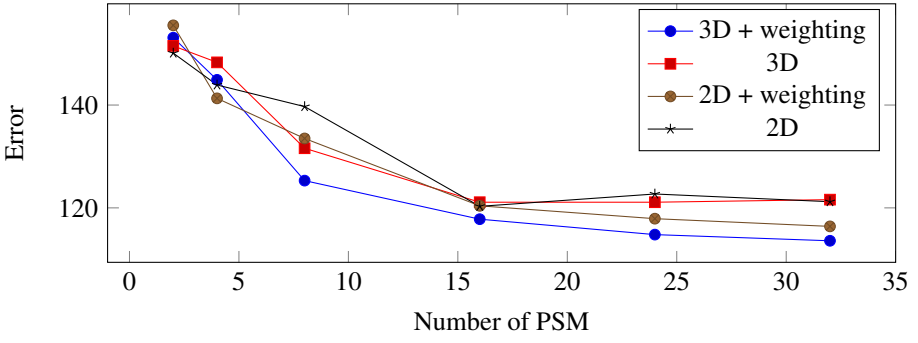

Figure 3: Impact of number of PS models, cluster weights (9) and clustering 3D poses or 2D poses. Average pose error is reported in mm.

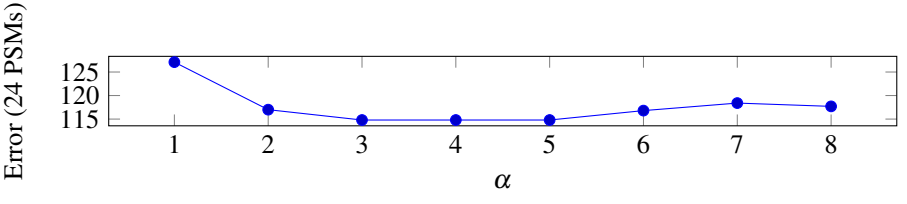

Figure 4: Impact of scaling parameter α (8). Average pose error is reported in mm.

dynamic programming to get the estimate $\hat{X}_J = argmax_{X \in \mathcal{X}^J} p(X|I,\vartheta)$. As proposed in [17], we use a mixture of PS models to overcome the limitations of a single tree model. To this end, we cluster the annotated poses in the training data.

Given a set of training poses M, we convert them to relative poses by subtracting the root node of the skeleton. We cluster the relative poses by k-means and estimate for each cluster k the parameters ϑ_k of a PS model. Inference with k PS models is then performed by:

$$\hat{X}_J = \hat{X}_{\hat{k}} \quad \text{where} \quad \hat{k} = \underset{k}{argmax} \left\{ \frac{|M_k|}{|M|} p(\hat{X}_k|I,\vartheta_k) \right\} \text{ and } \hat{X}_k = \underset{X \in \mathcal{X}^J}{argmax}\, p(X|I,\vartheta_k). \qquad (9)$$

This means that inference is first performed for each PS model independently and the solution of the model with highest confidence is taken. We weight the confidence of each model by the number of poses within each cluster, where $|M_k|$ denotes the number of poses used to train model k and $|M|$ is the overall number of poses. The impact of the weighting is evaluated in the experimental section.

7 Experiments

We evaluated our approach on the HumanEva-I [30] and Human3.6M [16] datasets. For both datasets we used the same 3D skeleton model shown in Figure 1. It consists of 14 joints (right/left ankle, right/left knee, right/left hip, right/left wrist, right/left elbow, right/left shoulder, neck, head top). In all experiments we assumed that the center (x_c, y_c, z_c) of the

bounding volume is given and define the volume generously by $[x_c - 1280, x_c + 1280] \times [y_c - 1280, y_c + 1280] \times [z_c - 1280, z_c + 1280]$, where 1280 is twice the image width. While we obtain the center from the annotations, modifications of the bounding volume like increasing the size do not change the accuracy since it is only needed to define a volume for voxelization. In our experiments we use two error measurements in order to compare with other methods: While *3D error* corresponds to mean average Euclidean distance of the estimated joints to the ground truth, *3D pose error* was introduced in [31] to compare with methods that do not estimate a global rigid transformation, *i.e.* only relative pose. For *3D pose error*, the inferred pose and ground truth are aligned by a rigid transformation using least squares before computing the 3D error.

HumanEva. We follow the protocol used in [31]. We evaluate our approach on the jogging and walking sequences using the data from all three RGB cameras for training and from the first camera for testing. The training sequences are used for training and the validation sequences for evaluation. For each action, DS regression forests and PS models are trained separately on the training images from all three RGB cameras. In order to train a forest, we randomly split the training data in proportion $\frac{2}{3}$ to train each tree and $\frac{1}{3}$ to learn the hyper-parameters.

Human3.6m. In contrast to the evaluation on the HumanEva-I dataset, we do not perform separate evaluations for different actions. We used six subjects (S1, S5, S6, S7, S8 and S9) for training and S11 for testing. In order to train a tree, we randomly selected four subjects to train the trees and two subjects to learn the hyper-parameters. As far as the dataset is redundant, we used only 1 out of 16 frames for training. It corresponds to approximately three frames per second. In order to select images for training, we clustered the data as in [28] using the distance $\max_j \|\mathbf{x}_{1j} - \mathbf{x}_{2j}\|$, where $\mathbf{x}_{1j}$ and $\mathbf{x}_{2j}$ are the relative poses, *i.e.* joint positions subtracted by pelvis joint. We selected 374 poses per subject, *i.e.* we used $374 \cdot 4$ images for training a tree and $374 \cdot 2$ for setting the hyper-parameters.

Learning hyper-parameters. To speed up the training, we optimize the parameters of the forest independently of the PS model. Since there are too many depth ambiguities without a PS model, we search for the hyper-parameters that minimize the 2D error measured between ground truth 2D joint locations and projected 3D locations inferred by the forest. As in [9], we use the fraction of joints with 2D error below 0.15 of the upper body size as measure.

Mixture of PSMs. We first evaluated the impact of having several PS models on the Human3.6m dataset. For this experiment, we evaluated our approach using only 1 out of 256 consecutive frames of the test sequences and used $h = 30mm$ as discretization step. With a single PS model, we obtain an average error of 164.92 mm. With an increasing number of PS models the error decreases as shown in Figure 3. The error is further reduced by weighting the PS models based on the cluster size (9). It is interesting to note that without weighting, the performance saturates with 16 models. In contrast, the error is further reduced with more models when they are weighted. Since we observe 2D images but estimate 3D pose, we investigated the difference between clustering the poses in 3D space or in the projected 2D space. This changes only the clusters M_k, but not the learning of the PSM parameters. The results show that the clustering of 3D poses gives better results. We also evaluated the effect of different values of the parameter α (8). The results in Figure 4 show that the error decreases by scaling the unaries with $\alpha > 1$.

Discretization and sampling. For inference, we discretize the bounding volume as described in Section 5.4. The parameter h is evaluated on the same subset of Human3.6m. For the experiments, we used a mixture of 32 PSMs. The results in Figure 5 show that the

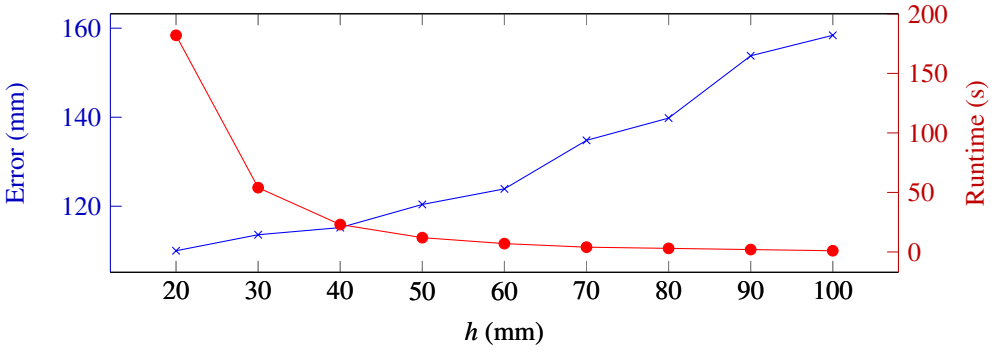

Figure 5: Impact of the discretization step h. We measured the runtime on Core i7-3770. Average pose error is reported in mm, runtime in seconds per image.

error decreases with a finer quantization of the volume at the cost of higher computational cost. We also evaluated the difference of sampling positive patches from 3D Gaussians or 2D Gaussians on a plane in the 3D volume as discussed in Section 5.1. The difference is minor. On HumanEva-I (S1,A1,C1) the 3D pose error for 2D Gaussians is 44.4 mm and for 3D Gaussians 44.0 mm.

Comparisons. We compare with other methods on HumanEva I and Human3.6m. We used 32 PSMs for both datasets. The 3D volume is discretized with $h = 20mm$. Table 1 compares our approach with other approaches. Since some approaches were evaluated using a relative error [5, 31, 32] and others use an absolute error [10, 24], we provide the results using both evaluation measures. Our approach outperforms most other approaches on the majority of subsets used for evaluation. Only [5], which estimates only relative pose, achieves a lower mean error. However, the standard deviation of the error is higher for [5]. We suppose that our approach, which aims at estimating absolute pose, has a constant error due to the discretization. Furthermore, the PS model allows more limb size variations, while the skeleton is stronger constrained in [5] when the training data contains only few subjects. Another reason could be the MoCap data that is not always well aligned in this dataset. Poorly aligned data has a bigger impact on learning absolute pose since the alignment error is partially removed when an approach is trained on relative poses.

We therefore compared to [5] on Human3.6m using the publicly available source code. For our approach, we use 32 PSMs and a uniform grid with step $h = 30mm$. In comparison to HumanEva-I, we use a sparser grid to speed up the computation since the dataset is larger. We evaluated both methods on 1 out of 64 frames of the testing dataset. The 3D pose error for TGP is 117.90 mm, while the error of our approach is 115.7 mm. Figure 6 compares the results for the two methods more in detail and shows that the accuracy of both methods is comparable. However, our approach estimates absolute pose, *i.e.* changes in the global position and orientation are obtained while relative pose does not provide such information.

8 Conclusion

We have presented an approach for estimating absolute 3D pose from RGB images. To this end, we extended regression forests that learn 2D-2D or 3D-3D mappings for estimating

	Walking (A1,C1)			Jogging (A2,C1)		
	S1	S2	S3	S1	S2	S3
	3D pose error (mm)					
DSRF	44.0 ± 15.9	30.9 ± 12.0	41.7 ± 14.9	57.2 ± 18.5	35.0 ± 9.9	33.3 ± 13.0
[32]	65.1 ± 17.4	48.6 ± 29.0	73.5 ± 21.4	74.2 ± 22.3	46.6 ± 24.7	32.2 ± 17.5
[31]	99.6 ± 42.6	108.3 ± 42.3	127.4 ± 24.0	109.2 ± 41.5	93.1 ± 41.1	115.8 ± 40.6
[5]	38.2 ± 21.4	32.8 ± 23.1	40.2 ± 23.2	42.0 ± 12.9	34.7 ± 16.6	46.4 ± 28.9
	3D error (mm)					
DSRF	89.3 ± 38.6	59.8 ± 24.1	78.5 ± 39.0	131.1 ± 115.4	70.2 ± 34.5	63.5 ± 38.2
[24]	75.1 ± 35.6	99.8 ± 32.6	93.8 ± 19.3	79.2 ± 26.4	89.8 ± 34.2	99.4 ± 35.1
[10]	89.3	108.7	113.5	-	-	-

Table 1: Comparison of our approach (DSRF) with state-of-the-art methods on HumanEva-I dataset. For comparison, we use two measures. While *3D error* is the error of estimated 3D joint positions, *3D pose error* is the 3D error up to a rigid transformation. Details are given in the text. The average error and standard deviation are reported.

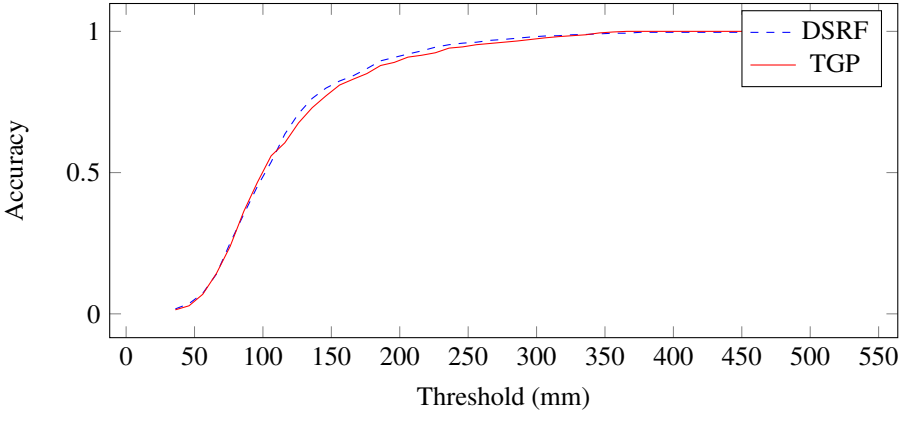

Figure 6: Comparison of our approach with Twin Gaussian Process regression [5] on Human3.6m dataset. The plot shows the number of estimated poses with an error below a certain threshold.

pose from relative feature locations to the 2D-3D case. The missing depth information in the 2D images is hypothesized by sweeping with a plane parallel to the image through the 3D volume and modeling patch distortions based on the depth by the splitting functions in the trees directly. The regression forests predict 3D confidence volumes of the joint positions, which are then used within a mixture of pictorial structure models extended to the 3D case to infer the pose. On two challenging datasets, our approach achieves state-of-the-art performance in terms of absolute and relative pose error.

Acknowledgement: Authors acknowledge financial support from the DFG Emmy Noether program (GA 1927/1-1).

References

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

[2] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multi-view pictorial structures for 3d human pose estimation. In *British Machine Vision Conference*, 2013.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[4] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[5] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2), 2010.

[6] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[7] M. Burenius, J. Sullivan, and S. Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] A. Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013.

[9] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[10] B. Daubney and X. Xie. Tracking 3d human pose with large root node uncertainty. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[11] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1), 2005.

[12] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1), 1973.

[13] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11), 2011.

[14] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *International Conference on Computer Vision*, 2011.

[15] G. Gkioxari, P. Arbeláez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[16] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.

[17] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference*, 2010.

[18] R. Memisevic, L. Sigal, and D.J. Fleet. Shared kernel information embedding for discriminative inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), 2012.

[19] T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, editors. *Visual Analysis of Humans - Looking at People*. Springer, 2011.

[20] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7), 2006.

[21] J. Müller and M. Arens. Human pose estimation with implicit shape models. In *International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, 2010.

[22] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *European Conference on Computer Vision*, 2008.

[23] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[24] I. Radwan, A. Dhall, and R. Goecke. Monocular image 3d human pose estimation under self-occlusion. In *International Conference on Computer Vision*, 2013.

[25] G. Rogez, J. Rihan, C.Orrite-Uruñuela, and P. H. S. Torr. Fast human pose detection using randomized hierarchical cascades of rejectors. *International Journal of Computer Vision*, 99(1), 2012.

[26] B. Rothrock, S. Park, and S.-C. Zhu. Integrating grammar and segmentation for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[27] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[28] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[29] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, 2013.

[30] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2), 2010.

[31] E. Simo-Serra, A. Ramisa, G. Alenya, C. Torras, and F. Moreno-Noguer. Single image 3d human pose estimation from noisy observations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[32] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A joint model for 2d and 3d pose estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[33] M. Sun, G. R. Bradski, B.-X. Xu, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *European Conference on Computer Vision*, 2010.

[34] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[35] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[36] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[37] F. Wang and Y. Li. Learning visual symbols for parsing human poses in images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[38] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[39] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures-of-parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013.