Real-Time Human Pose Recognition in Parts from Single Depth Images

Jamie Shotton Andrew Fitzgibbon Mat Cook Toby Sharp Mark Finocchio
Richard Moore Alex Kipman Andrew Blake
Microsoft Research Cambridge & Xbox Incubation

Abstract

We propose a new method to quickly and accurately predict 3D positions of body joints from a single depth image, using no temporal information. We take an object recognition approach, designing an intermediate body parts representation that maps the difficult pose estimation problem into a simpler per-pixel classification problem. Our large and highly varied training dataset allows the classifier to estimate body parts invariant to pose, body shape, clothing, etc. *Finally we generate confidence-scored 3D proposals of several body joints by reprojecting the classification result and finding local modes.*

The system runs at 200 frames per second on consumer hardware. Our evaluation shows high accuracy on both synthetic and real test sets, and investigates the effect of several training parameters. We achieve state of the art accuracy in our comparison with related work and demonstrate improved generalization over exact whole-skeleton nearest neighbor matching.

1. Introduction

Robust interactive human body tracking has applications including gaming, human-computer interaction, security, telepresence, and even health-care. The task has recently been greatly simplified by the introduction of real-time depth cameras [16, 19, 44, 37, 28, 13]. However, even the best existing systems still exhibit limitations. In particular, until the launch of Kinect [21], none ran at interactive rates on consumer hardware while handling a full range of human body shapes and sizes undergoing general body motions. Some systems achieve high speeds by tracking from frame to frame but struggle to re-initialize quickly and so are not robust. In this paper, we focus on pose recognition in parts: detecting from a single depth image a small set of 3D position candidates for each skeletal joint. Our focus on per-frame initialization and recovery is designed to complement any appropriate tracking algorithm [7, 39, 16, 42, 13] that might further incorporate temporal and kinematic coherence. The algorithm presented here forms a core component of the Kinect gaming platform [21].

Illustrated in Fig. 1 and inspired by recent object recognition work that divides objects into parts (*e.g.* [12, 43]), our approach is driven by two key design goals: computational efficiency and robustness. A single input depth image is segmented into a dense probabilistic body part labeling, with the parts defined to be spatially localized near skeletal joints of interest. Reprojecting the inferred parts into world space, we localize spatial modes of each part distribution and thus generate (possibly several) confidence-weighted proposals for the 3D locations of each skeletal joint.

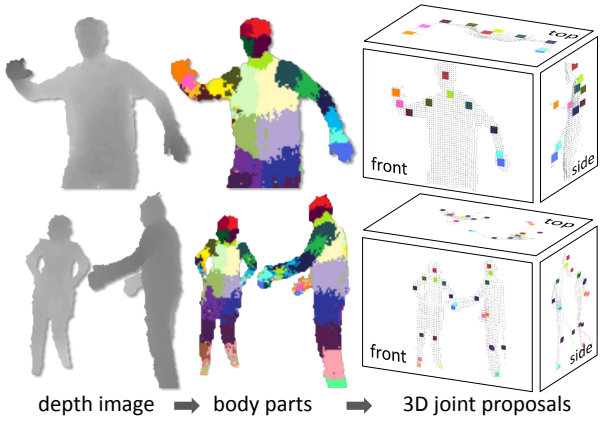

Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

We treat the segmentation into body parts as a per-pixel classification task (no pairwise terms or CRF have proved necessary). Evaluating each pixel separately avoids a combinatorial search over the different body joints, although within a single part there are of course still dramatic differences in the contextual appearance. For training data, we generate realistic synthetic depth images of humans of many shapes and sizes in highly varied poses sampled from a large motion capture database. We train a deep randomized decision forest classifier which avoids overfitting by using hundreds of thousands of training images. Simple, discriminative depth comparison image features yield 3D translation invariance while maintaining high computational efficiency. For further speed, the classifier can be run in parallel on each pixel on a GPU [34]. Finally, spatial modes of the inferred per-pixel distributions are computed using mean shift [10] resulting in the 3D joint proposals.

An optimized implementation of our algorithm runs in under 5ms per frame (200 frames per second) on the Xbox 360 GPU, at least one order of magnitude faster than existing approaches. It works frame-by-frame across dramatically differing body shapes and sizes, and the learned discriminative approach naturally handles self-occlusions and

poses cropped by the image frame. We evaluate on both real and synthetic depth images, containing challenging poses of a varied set of subjects. Even without exploiting temporal or kinematic constraints, the 3D joint proposals are both accurate and stable. We investigate the effect of several training parameters and show how very deep trees can still avoid overfitting due to the large training set. We demonstrate that our part proposals generalize at least as well as exact nearest-neighbor in both an idealized and realistic setting, and show a substantial improvement over the state of the art. Further, results on silhouette images suggest more general applicability of our approach.

Our main contribution is to treat pose estimation as object recognition using a novel intermediate body parts representation designed to spatially localize joints of interest at low computational cost and high accuracy. Our experiments also carry several insights: (i) synthetic depth training data is an excellent proxy for real data; (ii) scaling up the learning problem with varied synthetic data is important for high accuracy; and (iii) our parts-based approach generalizes better than even an oracular exact nearest neighbor.

Related Work. Human pose estimation has generated a vast literature (surveyed in [22, 29]). The recent availability of depth cameras has spurred further progress [16, 19, 28]. Grest *et al*. [16] use Iterated Closest Point to track a skeleton of a known size and starting position. Anguelov *et al*. [3] segment puppets in 3D range scan data into head, limbs, torso, and background using spin images and a MRF. In [44], Zhu & Fujimura build heuristic detectors for coarse upper body parts (head, torso, arms) using a linear programming relaxation, but require a T-pose initialization to size the model. Siddiqui & Medioni [37] hand craft head, hand, and forearm detectors, and show data-driven MCMC model fitting outperforms ICP. Kalogerakis *et al*. [18] classify and segment vertices in a full closed 3D mesh into different parts, but do not deal with occlusions and are sensitive to mesh topology. Most similar to our approach, Plagemann *et al*. [28] build a 3D mesh to find geodesic extrema interest points which are classified into 3 parts: head, hand, and foot. Their method provides both a location and orientation estimate of these parts, but does not distinguish left from right and the use of interest points limits the choice of parts.

Advances have also been made using conventional intensity cameras, though typically at much higher computational cost. Bregler & Malik [7] track humans using twists and exponential maps from a known initial pose. Ioffe & Forsyth [17] group parallel edges as candidate body segments and prune combinations of segments using a projected classifier. Mori & Malik [24] use the shape context descriptor to match exemplars. Ramanan & Forsyth [31] find candidate body segments as pairs of parallel lines, clustering appearances across frames. Shakhnarovich *et al*. [33] estimate upper body pose, interpolating k-NN poses matched by parameter sensitive hashing. Agarwal & Triggs [1] learn a regression from kernelized image silhouettes features to pose. Sigal *et al*. [39] use eigen-appearance template detectors for head, upper arms and lower legs proposals. Felzenszwalb & Huttenlocher [11] apply pictorial structures to estimate pose efficiently. Navaratnam *et al*. [25] use the marginal statistics of unlabeled data to improve pose estimation. Urtasun & Darrel [41] proposed a local mixture of Gaussian Processes to regress human pose. Auto-context was used in [40] to obtain a coarse body part labeling but this was not defined to localize joints and classifying each frame took about 40 seconds. Rogez *et al*. [32] train randomized decision forests on a hierarchy of classes defined on a torus of cyclic human motion patterns and camera angles. Wang & Popović [42] track a hand clothed in a colored glove. Our system could be seen as automatically inferring the colors of an virtual colored suit from a depth image. Bourdev & Malik [6] present 'poselets' that form tight clusters in both 3D pose and 2D image appearance, detectable using SVMs.

2. Data

Pose estimation research has often focused on techniques to overcome lack of training data [25], because of two problems. First, generating realistic intensity images using computer graphics techniques [33, 27, 26] is hampered by the huge color and texture variability induced by clothing, hair, and skin, often meaning that the data are reduced to 2D silhouettes [1]. Although depth cameras significantly reduce this difficulty, considerable variation in body and clothing *shape* remains. The second limitation is that synthetic body pose images are of necessity fed by motion-capture (mocap) data. Although techniques exist to simulate human motion (*e.g*. [38]) they do not yet produce the range of volitional motions of a human subject.

In this section we review depth imaging and show how we use real mocap data, retargetted to a variety of base character models, to synthesize a large, varied dataset. We believe this dataset to considerably advance the state of the art in both scale and variety, and demonstrate the importance of such a large dataset in our evaluation.

2.1. Depth imaging

Depth imaging technology has advanced dramatically over the last few years, finally reaching a consumer price point with the launch of Kinect [21]. Pixels in a depth image indicate calibrated depth in the scene, rather than a measure of intensity or color. We employ the Kinect camera which gives a 640x480 image at 30 frames per second with depth resolution of a few centimeters.

Depth cameras offer several advantages over traditional intensity sensors, working in low light levels, giving a calibrated scale estimate, being color and texture invariant, and resolving silhouette ambiguities in pose. They also greatly

Figure 2. **Synthetic and real data**. Pairs of depth image and ground truth body parts. Note wide variety in pose, shape, clothing, and crop.

simplify the task of background subtraction which we assume in this work. But most importantly for our approach, it is straightforward to synthesize realistic depth images of people and thus build a large training dataset cheaply.

2.2. Motion capture data

The human body is capable of an enormous range of poses which are difficult to simulate. Instead, we capture a large database of motion capture (mocap) of human actions. Our aim was to span the wide variety of poses people would make in an entertainment scenario. The database consists of approximately 500k frames in a few hundred sequences of driving, dancing, kicking, running, navigating menus, *etc*.

We expect our semi-local body part classifier to *generalize* somewhat to unseen poses. In particular, we need not record all possible combinations of the different limbs; in practice, a wide range of poses proves sufficient. Further, we need not record mocap with variation in rotation about the vertical axis, mirroring left-right, scene position, body shape and size, or camera pose, all of which can be added in (semi-)automatically.

Since the classifier uses no temporal information, we are interested only in static *poses* and not motion. Often, changes in pose from one mocap frame to the next are so small as to be insignificant. We thus discard many similar, redundant poses from the initial mocap data using 'furthest neighbor' clustering [15] where the distance between poses p_1 and p_2 is defined as $\max_j \|p_1^j - p_2^j\|_2$, the maximum Euclidean distance over body joints j. We use a subset of 100k poses such that no two poses are closer than 5cm.

We have found it necessary to iterate the process of motion capture, sampling from our model, training the classifier, and testing joint prediction accuracy in order to refine the mocap database with regions of pose space that had been previously missed out. Our early experiments employed the CMU mocap database [9] which gave acceptable results though covered far less of pose space.

2.3. Generating synthetic data

We build a randomized rendering pipeline from which we can sample fully labeled training images. Our goals in building this pipeline were twofold: realism and variety. For the learned model to work well, the samples must closely resemble real camera images, and contain good coverage of the appearance variations we hope to recognize at test time. While depth/scale and translation variations are handled explicitly in our features (see below), other invariances cannot be encoded efficiently. Instead we learn invariance from the data to camera pose, body pose, and body size and shape.

The synthesis pipeline first randomly samples a set of parameters, and then uses standard computer graphics techniques to render depth and (see below) body part images from texture mapped 3D meshes. The mocap is retargetting to each of 15 base meshes spanning the range of body shapes and sizes, using [4]. Further slight random variation in height and weight give extra coverage of body shapes. Other randomized parameters include the mocap frame, camera pose, camera noise, clothing and hairstyle. We provide more details of these variations in the supplementary material. Fig. 2 compares the varied output of the pipeline to hand-labeled real camera images.

3. Body Part Inference and Joint Proposals

In this section we describe our intermediate body parts representation, detail the discriminative depth image features, review decision forests and their application to body part recognition, and finally discuss how a mode finding algorithm is used to generate joint position proposals.

3.1. Body part labeling

A key contribution of this work is our intermediate body part representation. We define several localized body part labels that densely cover the body, as color-coded in Fig. 2. Some of these parts are defined to directly localize particular skeletal joints of interest, while others fill the gaps or could be used in combination to predict other joints. Our intermediate representation transforms the problem into one that can readily be solved by efficient classification algorithms; we show in Sec. 4.3 that the penalty paid for this transformation is small.

The parts are specified in a texture map that is retargetted to skin the various characters during rendering. The pairs of depth and body part images are used as fully labeled data for learning the classifier (see below). For the experiments in this paper, we use 31 body parts: LU/RU/LW/RW head, neck, L/R shoulder, LU/RU/LW/RW arm, L/R elbow, L/R wrist, L/R hand, LU/RU/LW/RW torso, LU/RU/LW/RW leg, L/R knee, L/R ankle, L/R foot (Left, Right, Upper, loWer). Distinct

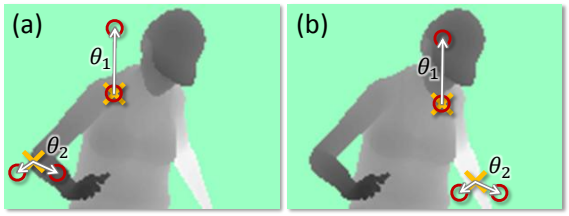

Figure 3. **Depth image features.** The yellow crosses indicates the pixel $\mathbf{x}$ being classified. The red circles indicate the offset pixels as defined in Eq. 1. In (a), the two example features give a large depth difference response. In (b), the same two features at new image locations give a much smaller response.

parts for left and right allow the classifier to disambiguate the left and right sides of the body.

Of course, the precise definition of these parts could be changed to suit a particular application. For example, in an upper body tracking scenario, all the lower body parts could be merged. Parts should be sufficiently small to accurately localize body joints, but not too numerous as to waste capacity of the classifier.

3.2. Depth image features

We employ simple depth comparison features, inspired by those in [20]. At a given pixel $\mathbf{x}$, the features compute

$$f_\theta(I,\mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right), \quad (1)$$

where $d_I(\mathbf{x})$ is the depth at pixel $\mathbf{x}$ in image I, and parameters $\theta = (\mathbf{u}, \mathbf{v})$ describe offsets $\mathbf{u}$ and $\mathbf{v}$. The normalization of the offsets by $\frac{1}{d_I(\mathbf{x})}$ ensures the features are depth invariant: at a given point on the body, a fixed *world space* offset will result whether the pixel is close or far from the camera. The features are thus 3D translation invariant (modulo perspective effects). If an offset pixel lies on the background or outside the bounds of the image, the depth probe $d_I(\mathbf{x}')$ is given a large positive constant value.

Fig. 3 illustrates two features at different pixel locations $\mathbf{x}$. Feature f_{θ_1} looks upwards: Eq. 1 will give a large positive response for pixels $\mathbf{x}$ near the top of the body, but a value close to zero for pixels $\mathbf{x}$ lower down the body. Feature f_{θ_2} may instead help find thin vertical structures such as the arm.

Individually these features provide only a weak signal about which part of the body the pixel belongs to, but in combination in a decision forest they are sufficient to accurately disambiguate all trained parts. The design of these features was strongly motivated by their computational efficiency: no preprocessing is needed; each feature need only read at most 3 image pixels and perform at most 5 arithmetic operations; and the features can be straightforwardly implemented on the GPU. Given a larger computational budget, one could employ potentially more powerful features based on, for example, depth integrals over regions, curvature, or local descriptors *e.g.* [5].

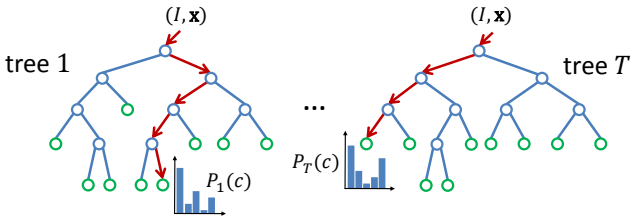

Figure 4. **Randomized Decision Forests.** A forest is an ensemble of trees. Each tree consists of split nodes (blue) and leaf nodes (green). The red arrows indicate the different paths that might be taken by different trees for a particular input.

3.3. Randomized decision forests

Randomized decision trees and forests [35, 30, 2, 8] have proven fast and effective multi-class classifiers for many tasks [20, 23, 36], and can be implemented efficiently on the GPU [34]. As illustrated in Fig. 4, a forest is an ensemble of T decision trees, each consisting of split and leaf nodes. Each split node consists of a feature f_θ and a threshold τ. To classify pixel $\mathbf{x}$ in image I, one starts at the root and repeatedly evaluates Eq. 1, branching left or right according to the comparison to threshold τ. At the leaf node reached in tree t, a learned distribution $P_t(c|I,\mathbf{x})$ over body part labels c is stored. The distributions are averaged together for all trees in the forest to give the final classification

$$P(c|I,\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,\mathbf{x}) . \quad (2)$$

Training. Each tree is trained on a different set of randomly synthesized images. A random subset of 2000 example pixels from each image is chosen to ensure a roughly even distribution across body parts. Each tree is trained using the following algorithm [20]:

1. Randomly propose a set of splitting candidates $\phi = (\theta, \tau)$ (feature parameters θ and thresholds τ).
2. Partition the set of examples $Q = \{(I,\mathbf{x})\}$ into left and right subsets by each ϕ:
$$Q_\text{l}(\phi) = \{\, (I,\mathbf{x}) \mid f_\theta(I,\mathbf{x}) < \tau \,\} \quad (3)$$
$$Q_\text{r}(\phi) = Q \setminus Q_\text{l}(\phi) \quad (4)$$
3. Compute the ϕ giving the largest gain in information:
$$\phi^\star = \operatorname*{argmax}_\phi G(\phi) \quad (5)$$
$$G(\phi) = H(Q) - \sum_{s\in\{\text{l,r}\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)) \quad (6)$$
where Shannon entropy $H(Q)$ is computed on the normalized histogram of body part labels $l_I(\mathbf{x})$ for all $(I,\mathbf{x}) \in Q$.
4. If the largest gain $G(\phi^\star)$ is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets $Q_\text{l}(\phi^\star)$ and $Q_\text{r}(\phi^\star)$.

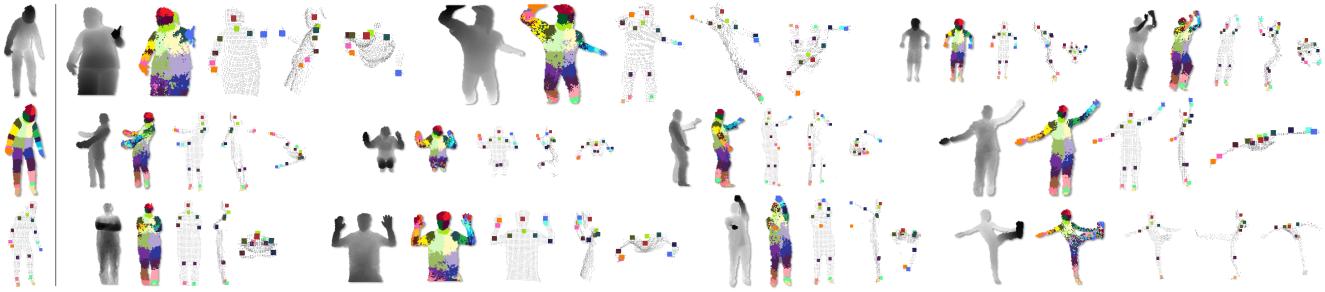

Figure 5. **Example inferences.** Synthetic (top row); real (middle); failure modes (bottom). Left column: ground truth for a neutral pose as a reference. In each example we see the depth image, the inferred most likely body part labels, and the joint proposals show as front, right, and top views (overlaid on a depth point cloud). Only the most confident proposal for each joint above a fixed, shared threshold is shown.

To keep the training times down we employ a distributed implementation. Training 3 trees to depth 20 from 1 million images takes about a day on a 1000 core cluster.

3.4. Joint position proposals

Body part recognition as described above infers per-pixel information. This information must now be pooled across pixels to generate reliable proposals for the positions of 3D skeletal joints. These proposals are the final output of our algorithm, and could be used by a tracking algorithm to self-initialize and recover from failure.

A simple option is to accumulate the global 3D centers of probability mass for each part, using the known calibrated depth. However, outlying pixels severely degrade the quality of such a global estimate. Instead we employ a local mode-finding approach based on mean shift [10] with a weighted Gaussian kernel.

We define a density estimator per body part as

$$f_c(\hat{\mathbf{x}}) \propto \sum_{i=1}^{N} w_{ic} \exp\left(- \left\| \frac{\hat{\mathbf{x}} - \hat{\mathbf{x}}_i}{b_c} \right\|^2 \right) , \qquad (7)$$

where $\hat{\mathbf{x}}$ is a coordinate in 3D world space, N is the number of image pixels, w_{ic} is a pixel weighting, $\hat{\mathbf{x}}_i$ is the reprojection of image pixel $\mathbf{x}_i$ into world space given depth $d_I(\mathbf{x}_i)$, and b_c is a learned per-part bandwidth. The pixel weighting w_{ic} considers both the inferred body part probability at the pixel and the world surface area of the pixel:

$$w_{ic} = P(c|I, \mathbf{x}_i) \cdot d_I(\mathbf{x}_i)^2 . \qquad (8)$$

This ensures density estimates are depth invariant and gave a small but significant improvement in joint prediction accuracy. Depending on the definition of body parts, the posterior $P(c|I, \mathbf{x})$ can be pre-accumulated over a small set of parts. For example, in our experiments the four body parts covering the head are merged to localize the head joint.

Mean shift is used to find modes in this density efficiently. All pixels above a learned probability threshold λ_c are used as starting points for part c. A final confidence estimate is given as a sum of the pixel weights reaching each mode. This proved more reliable than taking the modal density estimate.

The detected modes lie on the *surface* of the body. Each mode is therefore pushed back into the scene by a learned z offset ζ_c to produce a final joint position proposal. This simple, efficient approach works well in practice. The bandwidths b_c, probability threshold λ_c, and surface-to-interior z offset ζ_c are optimized per-part on a hold-out validation set of 5000 images by grid search. (As an indication, this resulted in mean bandwidth 0.065m, probability threshold 0.14, and z offset 0.039m).

4. Experiments

In this section we describe the experiments performed to evaluate our method. We show both qualitative and quantitative results on several challenging datasets, and compare with both nearest-neighbor approaches and the state of the art [13]. We provide further results in the supplementary material. Unless otherwise specified, parameters below were set as: 3 trees, 20 deep, 300k training images per tree, 2000 training example pixels per image, 2000 candidate features θ, and 50 candidate thresholds τ per feature.

Test data. We use challenging synthetic and real depth images to evaluate our approach. For our synthetic test set, we synthesize 5000 depth images, together with the ground truth body part labels and joint positions. The original mocap *poses* used to generate these images are held out from the training data. Our real test set consists of 8808 frames of real depth images over 15 different subjects, hand-labeled with dense body parts and 7 upper body joint positions. We also evaluate on the real depth data from [13]. The results suggest that effects seen on synthetic data are mirrored in the real data, and further that our synthetic test set is by far the 'hardest' due to the extreme variability in pose and body shape. For most experiments we limit the rotation of the user to $\pm 120^\circ$ in both training and synthetic test data since the user is facing the camera (0°) in our main entertainment scenario, though we also evaluate the full 360° scenario.

Error metrics. We quantify both classification and joint prediction accuracy. For classification, we report the average per-class accuracy, *i.e.* the average of the diagonal of the confusion matrix between the ground truth part label and the most likely inferred part label. This metric weights each

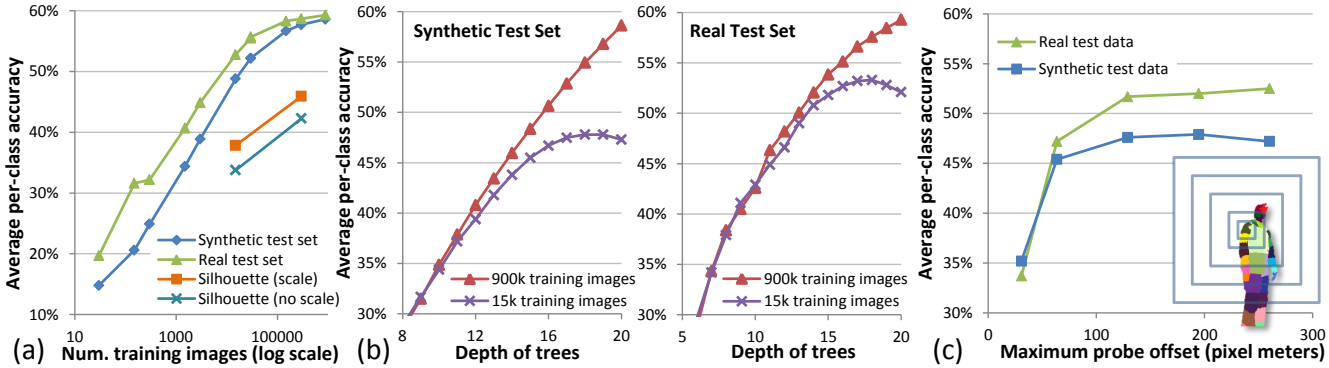

Figure 6. **Training parameters *vs*. classification accuracy.** (a) Number of training images. (b) Depth of trees. (c) Maximum probe offset.

body part equally despite their varying sizes, though mislabelings on the part boundaries reduce the absolute numbers.

For joint proposals, we generate recall-precision curves as a function of confidence threshold. We quantify accuracy as average precision per joint, or mean average precision (mAP) over all joints.The first joint proposal within D meters of the ground truth position is taken as a true positive, while other proposals also within D meters count as false positives. This penalizes multiple spurious detections near the correct position which might slow a downstream tracking algorithm. Any joint proposals outside D meters also count as false positives. Note that *all* proposals (not just the most confident) are counted in this metric. Joints invisible in the image are not penalized as false negatives. We set $D = 0.1\text{m}$ below, approximately the accuracy of the hand-labeled real test data ground truth. The strong correlation of classification and joint prediction accuracy (*c.f*. the blue curves in Figs. 6(a) and 8(a)) suggests the trends observed below for one also apply for the other.

4.1. Qualitative results

Fig. 5 shows example inferences of our algorithm. Note high accuracy of both classification and joint prediction across large variations in body and camera pose, depth in scene, cropping, and body size and shape (*e.g*. small child *vs*. heavy adult). The bottom row shows some failure modes of the body part classification. The first example shows a failure to distinguish subtle changes in the depth image such as the crossed arms. Often (as with the second and third failure examples) the most likely body part is incorrect, but there is still sufficient correct probability mass in distribution $P(c|I, \mathbf{x})$ that an accurate proposal can still be generated. The fourth example shows a failure to generalize well to an unseen pose, but the confidence gates bad proposals, maintaining high precision at the expense of recall.

Note that no temporal or kinematic constraints (other than those implicit in the training data) are used for any of our results. Despite this, per-frame results on video sequences in the supplementary material show almost every joint accurately predicted with remarkably little jitter.

4.2. Classification accuracy

We investigate the effect of several training parameters on classification accuracy. The trends are highly correlated between the synthetic and real test sets, and the real test set appears consistently 'easier' than the synthetic test set, probably due to the less varied poses present.

Number of training images. In Fig. 6(a) we show how test accuracy increases approximately logarithmically with the number of randomly generated training images, though starts to tail off around 100k images. As shown below, this saturation is likely due to the limited model capacity of a 3 tree, 20 deep decision forest.

Silhouette images. We also show in Fig. 6(a) the quality of our approach on synthetic silhouette images, where the features in Eq. 1 are either given scale (as the mean depth) or not (a fixed constant depth). For the corresponding joint prediction using a 2D metric with a 10 pixel true positive threshold, we got 0.539 mAP with scale and 0.465 mAP without. While clearly a harder task due to depth ambiguities, these results suggest the applicability of our approach to other imaging modalities.

Depth of trees. Fig. 6(b) shows how the depth of trees affects test accuracy using either 15k or 900k images. Of all the training parameters, depth appears to have the most significant effect as it directly impacts the model capacity of the classifier. Using only 15k images we observe overfitting beginning around depth 17, but the enlarged 900k training set avoids this. The high accuracy gradient at depth 20 suggests even better results can be achieved by training still deeper trees, at a small extra run-time computational cost and a large extra memory penalty. Of practical interest is that, until about depth 10, the training set size matters little, suggesting an efficient training strategy.

Maximum probe offset. The range of depth probe offsets allowed during training has a large effect on accuracy. We show this in Fig. 6(c) for 5k training images, where 'maximum probe offset' means the max. absolute value proposed for both x and y coordinates of $\mathbf{u}$ and $\mathbf{v}$ in Eq. 1. The concentric boxes on the right show the 5 tested maximum off-

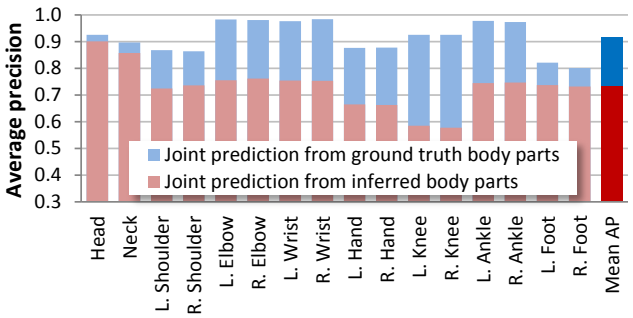

Figure 7. **Joint prediction accuracy.** We compare the actual performance of our system (red) with the best achievable result (blue) given the ground truth body part labels.

sets calibrated for a left shoulder pixel in that image; the largest offset covers almost all the body. (Recall that this maximum offset scales with world depth of the pixel). As the maximum probe offset is increased, the classifier is able to use more spatial context to make its decisions, though without enough data would eventually risk overfitting to this context. Accuracy increases with the maximum probe offset, though levels off around 129 pixel meters.

4.3. Joint prediction accuracy

In Fig. 7 we show average precision results on the synthetic test set, achieving 0.731 mAP. We compare an idealized setup that is given the *ground truth* body part labels to the real setup using inferred body parts. While we do pay a small penalty for using our intermediate body parts representation, for many joints the inferred results are both highly accurate and close to this upper bound. On the real test set, we have ground truth labels for head, shoulders, elbows, and hands. An mAP of 0.984 is achieved on those parts given the ground truth body part labels, while 0.914 mAP is achieved using the inferred body parts. As expected, these numbers are considerably higher on this easier test set.

Comparison with nearest neighbor. To highlight the need to treat pose recognition in *parts*, and to calibrate the difficulty of our test set for the reader, we compare with two variants of exact nearest-neighbor whole-body matching in Fig. 8(a). The first, idealized, variant matches the ground truth *test skeleton* to a set of training exemplar skeletons with optimal rigid translational alignment in 3D world space. Of course, in practice one has no access to the test skeleton. As an example of a realizable system, the second variant uses chamfer matching [14] to compare the test image to the training exemplars. This is computed using depth edges and 12 orientation bins. To make the chamfer task easier, we throw out any cropped training or test images. We align images using the 3D center of mass, and found that further local rigid translation only reduced accuracy.

Our algorithm, recognizing in parts, generalizes better than even the idealized skeleton matching until about 150k training images are reached. As noted above, our results may get even better with deeper trees, but already we robustly infer 3D body joint positions and cope naturally with cropping and translation. The speed of nearest neighbor chamfer matching is also drastically slower (2 fps) than our algorithm. While hierarchical matching [14] is faster, one would still need a massive exemplar set to achieve comparable accuracy.

Comparison with [13]. The authors of [13] provided their test data and results for direct comparison. Their algorithm uses body part proposals from [28] and further tracks the skeleton with kinematic and temporal information. Their data comes from a time-of-flight depth camera with very different noise characteristics to our structured light sensor. Without any changes to our training data or algorithm, Fig. 8(b) shows considerably improved joint prediction average precision. Our algorithm also runs at least 10x faster.

Full rotations and multiple people. To evaluate the full 360° rotation scenario, we trained a forest on 900k images containing full rotations and tested on 5k synthetic full rotation images (with held out poses). Despite the massive increase in left-right ambiguity, our system was still able to achieve an mAP of 0.655, indicating that our classifier can accurately learn the subtle visual cues that distinguish front and back facing poses. Residual left-right uncertainty after classification can naturally be propagated to a tracking algorithm through multiple hypotheses. Our approach can propose joint positions for multiple people in the image, since the per-pixel classifier generalizes well even without explicit training for this scenario. Results are given in Fig. 1 and the supplementary material.

Faster proposals. We also implemented a faster alternative approach to generating the proposals based on simple bottom-up clustering. Combined with body part classification, this runs at $\sim$ 200 fps on the Xbox GPU, *vs.* $\sim$ 50 fps using mean shift on a modern 8 core desktop CPU. Given the computational savings, the 0.677 mAP achieved on the synthetic test set compares favorably to the 0.731 mAP of the mean shift approach.

5. Discussion

We have seen how accurate proposals for the 3D locations of body joints can be estimated in super real-time from single depth images. We introduced body part recognition as an intermediate representation for human pose estimation. Using a highly varied synthetic training set allowed us to train very deep decision forests using simple depth-invariant features without overfitting, learning invariance to both pose and shape. Detecting modes in a density function gives the final set of confidence-weighted 3D joint proposals. Our results show high correlation between real and synthetic data, and between the intermediate classification and the final joint proposal accuracy. We have highlighted the importance of breaking the whole skeleton into parts, and show state of the art accuracy on a competitive test set.

As future work, we plan further study of the variability

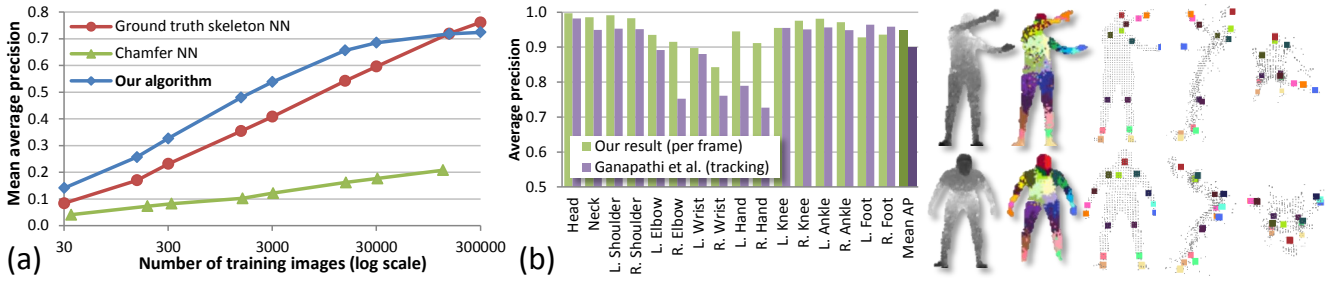

Figure 8. **Comparisons.** (a) Comparison with nearest neighbor matching. (b) Comparison with [13]. Even without the kinematic and temporal constraints exploited by [13], our algorithm is able to more accurately localize body joints.

in the source mocap data, the properties of the generative model underlying the synthesis pipeline, and the particular part definitions. Whether a similarly efficient approach that can directly regress joint positions is also an open question. Perhaps a global estimate of latent variables such as coarse person orientation could be used to condition the body part inference and remove ambiguities in local pose estimates.

Acknowledgements. We thank the many skilled engineers in Xbox, particularly Robert Craig, Matt Bronder, Craig Peeper, Momin Al-Ghosien, and Ryan Geiss, who built the Kinect tracking system on top of this research. We also thank John Winn, Duncan Robertson, Antonio Criminisi, Shahram Izadi, Ollie Williams, and Mihai Budiu for help and valuable discussions, and Varun Ganapathi and Christian Plagemann for providing their test data.

References

[1] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *Proc. CVPR*, 2004. 1298

[2] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997. 1300

[3] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, and A. Ng. Discriminative learning of markov random fields for segmentation of 3D scan data. In *Proc. CVPR*, 2005. 1298

[4] Autodesk MotionBuilder. 1299

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24, 2002. 1300

[6] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *Proc. ICCV*, 2009. 1298

[7] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. CVPR*, 1998. 1297, 1298

[8] L. Breiman. Random forests. *Mach. Learning*, 45(1):5–32, 2001. 1300

[9] CMU Mocap Database. http://mocap.cs.cmu.edu/. 1299

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5), 2002. 1297, 1301

[11] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, Jan. 2005. 1298

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003. 1297

[13] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Proc. CVPR*, 2010. 1297, 1301, 1303, 1304

[14] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. ECCV*, June 2000. 1303

[15] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comp. Sci.*, 38, 1985. 1299

[16] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. In *In Proc. DAGM*, 2005. 1297, 1298

[17] S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *IJCV*, 43(1):45–68, 2001. 1298

[18] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *ACM Trans. Graphics*, 29(3), 2010. 1298

[19] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3D human body tracking with an articulated 3D body model. In *Proc. ICRA*, 2006. 1297, 1298

[20] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. CVPR*, pages 2:775–781, 2005. 1300

[21] Microsoft Corp. Redmond WA. Kinect for Xbox 360. 1297, 1298

[22] T. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 2006. 1298

[23] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2006. 1300

[24] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. ICCV*, 2003. 1298

[25] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *Proc. ICCV*, 2007. 1298

[26] H. Ning, W. Xu, Y. Gong, and T. S. Huang. Discriminative learning of visual words for 3D human pose estimation. In *Proc. CVPR*, 2008. 1298

[27] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *Proc. ECCV*, 2008. 1298

[28] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Proc. ICRA*, 2010. 1297, 1298, 1303

[29] R. Poppe. Vision-based human motion analysis: An overview. *CVIU*, 108, 2007. 1298

[30] J. R. Quinlan. Induction of decision trees. *Mach. Learn*, 1986. 1300

[31] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *Proc. CVPR*, 2003. 1298

[32] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. Torr. Randomized trees for human pose detection. In *Proc. CVPR*, 2008. 1298

[33] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. ICCV*, 2003. 1298

[34] T. Sharp. Implementing decision trees and forests on a GPU. In *Proc. ECCV*, 2008. 1297, 1300

[35] B. Shepherd. An appraisal of a decision tree approach to image classification. In *IJCAI*, 1983. 1300

[36] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. CVPR*, 2008. 1300

[37] M. Siddiqui and G. Medioni. Human pose estimation from a single view point, real-time range sensor. In *CVCG at CVPR*, 2010. 1297, 1298

[38] H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV*, 2002. 1298

[39] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *Proc. CVPR*, 2004. 1297, 1298

[40] Z. Tu. Auto-context and its application to high-level vision tasks. In *Proc. CVPR*, 2008. 1298

[41] R. Urtasun and T. Darrell. Local probabilistic regression for activity-independent human pose inference. In *Proc. CVPR*, 2008. 1298

[42] R. Wang and J. Popović. Real-time hand-tracking with a color glove. In *Proc. ACM SIGGRAPH*, 2009. 1297, 1298

[43] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proc. CVPR*, 2006. 1297

[44] Y. Zhu and K. Fujimura. Constrained optimization for human pose estimation from depth sequences. In *Proc. ACCV*, 2007. 1297, 1298