Estimating Body Pose of Infants in Depth Images using Random Ferns

Nikolas Hesse Gregor Stachowiak Timo Breuer
Michael Arens
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Gutleuthausstr. 1, 76275 Ettlingen, Germany
nikolas.hesse@iosb.fraunhofer.de

Abstract

In recent years, many systems for motion analysis of infants have been developed which either use markers or lack 3D information. We propose a system that can be trained fast and flexibly to fit the requirements of markerless 3D movement analysis of infants. Random Ferns are used as an efficient and robust alternative to Random Forests to find the 3D positions of body joints in single depth images. The training time is reduced by several orders of magnitude compared to the Kinect approach using a similar amount of data. Our system is trained in 9 hours on a 32 core workstation opposed to 24 hours on a 1000 core cluster, achieving comparable accuracy to the Kinect SDK on a publicly available pose estimation benchmark dataset containing adults. On manually annotated recordings of an infant, we obtain an average distance error over all joints of 41 mm. Building on the proposed approach, we aim to develop an automated, unintrusive, cheap and objective system for the early detection of infantile movement disorders like cerebral palsy using 3D motion analysis techniques.

1. Introduction

When Microsoft released the low-cost commodity depth sensor Kinect in conjunction with its body tracking functionality [17], many researchers started developing medical applications, e.g. for gait analysis [8], rehabilitation monitoring [5, 14], postural control assessment [6], or monitoring of musculo-skeletal disorders [26]. One application that can not utilize the tracking, although it involves the analysis of human motion, is the automated examination of infant movements, e.g. for early detection of cerebral palsy. The body tracking was originally developed for the game console XBox and aims at handling a wide range of potential gamers from children to adults. It therefore will not work reliably for humans smaller than one meter, which makes it unfeasible for tracking infants [18].

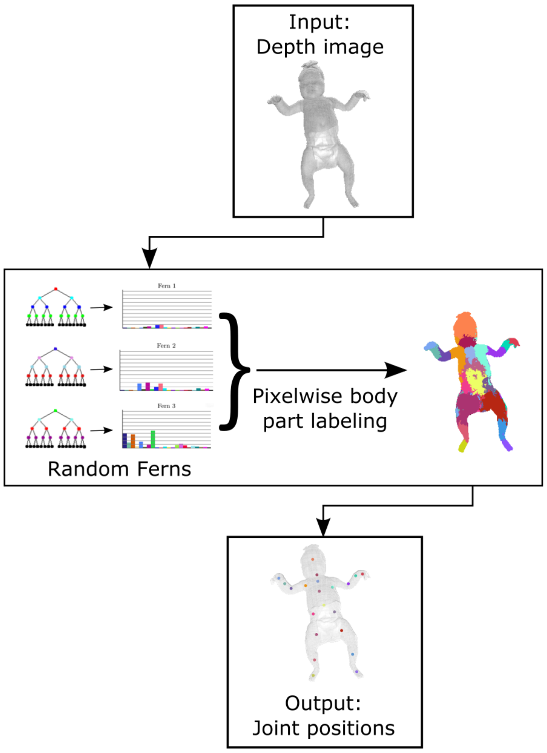


Figure 1: Pose estimation pipeline. For an input pixel, each Random Fern outputs a probability distribution over body part classes. After combination of distributions, the body part class with highest probability is assigned to the input pixel. Joint positions are inferred from estimated body parts. Best viewed in color.

Cerebral palsy (CP) is a movement disorder that is caused by abnormal development or damage to parts of the brain, often during pregnancy. It leads to abnormal muscle tone, reflexes, or motor development and coordination [22]. For the diagnosis of cerebral palsy at the age of three months, the *General Movement Assessment* (GMA) [7] is a well-established instrument. This time-consuming examination must be performed by trained and experienced experts (often doctors), who assess the spontaneous movements of the infants. However, the outcome measures are not standardized and represent the subjective opinion of the examiner. Therefore, it is desirable to automate the process of evaluating the quality of infant movements.

Several systems have been proposed to tackle this issue. In order to assess the motions of the infants, the bodily movements have to be captured. [16] use a marker-based Vicon system, while [21] and [24] use a 2D optical flow-based approach. Other systems propose the attachment of sensors directly to the infant's limbs using accelerometers [11] or electromagnetical sensors [13]. One approach, utilizing the Kinect, fits a body model consisting of basic shapes to the depth image [19]. Based on the measured movements, different methods are applied to predict whether or not a child is afflicted by CP.

An ideal system for analyzing the motions of infants in a clinical environment offers the following properties:

- It is cheap
- It is easy to set up
- It is usable by non-experts
- It is non-intrusive
- It is accurate and reliable
- It provides objective measures

We aim to provide a 3D motion analysis system that fulfills all of these requirements. A first step towards this goal is the development of a markerless body pose estimator in single depth images for infants.

The contributions of our work are threefold. We implement a procedure for generating labeled depth images synthetically for arbitrary body models. We develop a training procedure for our pose estimation system that is several orders of magnitude faster than [23] while maintaining comparable accuracy. We introduce a markerless and unintrusive system for infant body pose estimation.

2. Related work

Human body pose estimation is a very active field of research in which many systems have been developed in recent years. We divide the approaches into model-based approaches which fit a detailed human body model to the given data ([9], [27], [1], [12], [25]), and into discriminative approaches which aim to directly find body parts from which joint positions are inferred ([23], [28], [4]).

Although there is a large variety of systems, we focus our comparison on the Kinect approach due to the similarity of characteristics to our work. The body tracking system of the Kinect is based on an approach by Shotton et al. [23], who combine binary depth comparison features using Random Decison Forests to assign a body part label to each input depth pixel. Based on the detected body parts, the joint positions are estimated. Unfortunately, the training of the forests is computationally very intense, requiring one day on a 1000 core cluster using 1 million training images to train 3 trees of depth 20. Besides the costs of huge computation units, many problems arise when distributively processing such a large amount of data [3]. Therefore, it is desirable to speed up and to simplify the training procedure, not only to avoid these problems, but to be able to flexibly adapt the system to different application requirements.

For this reason, we propose the use of *random ferns*, which were introduced by [20] as an efficient and robust alternative to random forests.

3. Methods

We use random ferns for fusing binary depth comparison features to create a pixelwise body part classifier. Based on the body parts, we estimate the positions of the body joints.

3.1. Binary Features

The task at hand is to correctly estimate the body part class of each pixel of a depth image displaying a human. We do this by applying depth comparisons between the current input pixel and several pixels within a predefined neighborhood radius. Each of these comparisons together with a threshold is considered a feature. The outcome of the feature is either 1 or 0, depending on the result of the depth comparison being greater than the threshold or not.

We use depth comparisons similar to [23], defined as

$$z_\phi(I,x) = d_I(x) - d_I(x + \phi \cdot r(x)), \tag{1}$$

where ϕ is the relative offset to given pixel x in image I, $d_I(x)$ returns the depth of x and $r(x) = \frac{foc}{d_I(x)}$ is a normalization factor for depth invariance, where foc is the focal length of the depth sensor in pixels. A binary feature consists of a depth comparison in combination with a threshold τ, and is evaluated as

$$f_\phi(I,x) = \begin{cases} 1, & \text{if } z_\phi(I,x) > \tau, \\ 0, & \text{else.} \end{cases} \tag{2}$$

Each binary feature for itself is not very meaningful, therefore we combine many features using so called *ferns* to obtain an accurate estimation.

3.2. Ferns

We follow the formulation of [20].

Let $c_i, i = 1, \ldots, H$ be the set of classes and let $f_j, j = 1, \ldots, N$ be the binary features. We want to find

$$\hat{c} = \arg\max_{c_i} P(C = c_i | f_1, f_2, \ldots, f_N), \tag{3}$$

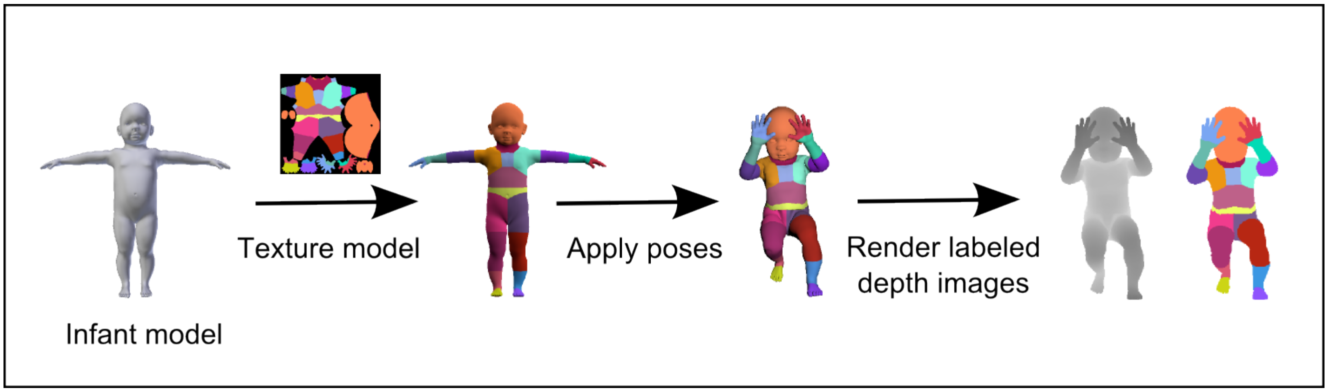


Figure 2: Generation of synthetic training data. Infant model is textured with colors correspondig to body parts. After different poses are applied, labeled depth images are generated that serve as input for the fern training. Best viewed in color.

where C is a random variable representing the body part class. Applying Bayes' formula leads to

$$P(C = c_i|f_1, f_2, \dots, f_N) = \frac{P(f_1,f_2,\dots,f_N|C=c_i)P(C=c_i)}{P(f_1,f_2,\dots,f_N)}. \quad (4)$$

In contrast to [20], who assume a uniform prior $P(C)$, we use prior probabilities depending on the number of pixels representing each body part in the training data. Being independent of the class, the denominator is regarded as a scaling factor and therefore omitted.

A complete representation of the joint probability of all features requires storing and estimating 2^N entries for each class, which makes it computationally intractable for more than a few classes. If we assume complete independence of the features, the problem becomes trivial, but the correlation between features is ignored. Therefore, a compromise is chosen by partitioning the set of all features into M groups of size $S = \frac{N}{M}$, where each group is called *fern*. In each fern the joint probability is computed. This leads to

$$P(f_1, f_2, \dots, f_N|C = c_i) = \prod_{k=1}^{M} P(F_m|C = c_i), \quad (5)$$

where $F_m = \{f_{\sigma(m,1)}, f_{\sigma(m,2)}, \dots, f_{\sigma(m,S)}\}, m = 1, \dots, M$ represents the m^{th} fern and $\sigma(m, j)$ is a random permutation function with range $1, \dots, N$.

This semi-naive bayesian approach models only some of the dependencies between features, but can be handled easily, as we need to store and estimate $M \times 2^S$ parameters. In all our experiments we use M = 15, S = 12. Each fern can be seen as a special kind of decision tree, with binary features as split nodes, where within each level of the tree the same features are evaluated. The depth of the tree corresponds to the group size S. In each leaf node the probability distribution of all classes is stored. The complete set of M ferns is called an *ensemble of ferns*.

3.3. Training data generation

We synthetically create a large amount of labeled data for training the ferns. We use a 3D body model of an infant from MakeHuman [15], an open source tool for making 3D characters. A subset of the available body joints is selected in our experiments. The joints do not necessarily correspond to real joints of the human body, but serve as a tool for dividing the model into different regions. We choose 21 joints for infants: head, neck, shoulders, elbows, hands, fingers, upper body center, body center, stomach, hips, knees, feet and toes. A texture is applied to the model that maps each skin pixel to a color according to the closest joint. The open source software Blender [2] is used for animating the model in many different poses, using the CMU motion capture dataset [10]. The dataset contains a variety of poses that were captured by a Vicon system at 120 Hz. Consecutive poses are very similar due to the high capture rate and are omitted if the summed joint distances lie below a predefined threshold. We generate depth images from the posed model and assign a body part label to each pixel according to the model texture. The virtual camera viewpoint from which the depth images are generated can be chosen arbitrarily. We use mainly frontal views of the body as we assume the infants to be lying on their back in real data. The data generation procedure is illustrated in Figure 2.

3.4. Fern Training

The algorithm for the training procedure is outlined in Algorithm 1. The goal is to build an ensemble E consisting of M ferns. We start by creating a fern, given its depth S and the neighborhood radius for pixel offsets. The randomness is introduced in the sampling of the binary features, i.e. pixel offsets within the specified neighborhood and thresholds (as used in Eq. 1 and 2). The outcome of the features in the fern accumulates to the descriptor $F_m = (f_{\phi 1}, f_{\phi 2}, \dots, f_{\phi S})$ and is indexed by a binary code that indicates which leaf node is reached by the input pixel. In each leaf node, the probability distribution over all body part classes is stored which is given by

$$P(F_m = k|C = c_i) = \frac{n_{k,i} + u}{\sum_k (n_{k,i} + u)}, \tag{6}$$

where we consider F_m to be equal to k if the binary code of the feature descriptor equals k. Furthermore, $n_{k,i}$ is the entry in the histogram of descriptor F_m and is equal to the number of pixels belonging to class i that evaluate to fern value k. The u can be seen as a Dirichlet prior to avoid probabilities of zero, in case a leaf is not reached by any input pixel, which makes the multiplicative combination of all ferns zero. Choosing $u = 1$ leads to

$$P(F_m = k|C = c_i) = \frac{n_{k,i} + 1}{N_i + K}, \tag{7}$$

with N_i being the total number of pixels in the training data that belong to class i.

We evaluate the error that results from classifying all input pixels using the ferns in the ensemble together with the current fern: as before, every pixel of every input image is input to each of the ferns and the resulting probability distributions of all ferns are multiplied. From the resulting distribution, the class with highest probability is the estimated class for the current input pixel. The estimate is compared to the ground truth and the average error over all pixels in all images is computed.

The described steps are repeated i_F times, after which the fern generating the lowest error rate together with E is added to E. After M ferns are added to the ensemble E, the algorithm terminates.

3.5. From body parts to joint positions

After passing the ensemble of ferns, each pixel of the body is assigned to a body part class. We filter out assumingly incorrectly labeled pixels by creating connected clusters in the depth image from equally labeled pixels and ignoring all pixels not belonging to the largest cluster of their respective body part class. We infer the joint positions by calculating the mean of the remaining cluster of each body part. A drawback of this method is that the joints lie close to the body surface, which does not resemble a real human skeleton. We need to incorporate a procedure for finding the joint positions more accurately and in a way that they conform to a human body with respect to bone sizes and locations.

4. Evaluation

Since no publicly available depth image datasets of infants exist, we evaluate our method on a public pose estimation dataset containing data from adults in order to compare it to the Kinect SDK. To show the usefulness of the system for the proposed application we also evaluate it on manually annotated recordings of an infant consisting of more than 1000 depth images. As a measure of accuracy we use the average distance between estimated joints and ground truth in all experiments. We experimentally determined the used parameters in order to find a good tradeoff between speed and accuracy.

Algorithm 1 Fern training procedure

```
Input: $M$: predefined size of fern ensemble
       $i_F$: number of iterations per fern
       $S$: depth of fern
       $R$: size of neighborhood radius
       $I$: labeled depth images
Output: ensemble of ferns $E$

Initialize:
2-D Array $Histogram$ of size (#leaf nodes ($2^S$)) $\times$ (#classes) with all zeros
$F_{best}$ = NULL
$Err_{min} = \infty$
for $i = 0$ to $M$ do
  for $j = 0$ to $i_F$ do
    $F$ := CreateRandomFern($S, R$)
    for all images $im \in I$ do
      for all pixels $p$ in $im$ do
        $k$ := GetLeafNodeIndex($F, p$)
        Set $Histogram[k]$[GetLabel($p$)] += 1
      end for
    end for
    $Err$ := EvaluateTrainingError($E \bigcup F$)
    if $Err < Err_{min}$ then
      $F_{best} = F$
      $Err_{min} = Err$
    end if
    $j$ += 1
  end for
  $E = E \bigcup F_{best}$
  $i$ += 1
end for
```

4.1. Results - PDT13 dataset

We evaluate our system on the publicly available Personalized Depth Tracker Dataset (PDT13) [12]. It offers Kinect depth recordings of 5 different adults, each performing 4 movement sequences of increasing difficulty. Ground truth joint positions were generated based on measurements of a full-body laser scanner.

We have trained our system using 735 K labeled depth images. We set the number of ferns M to 15, the depth

Table 1: Average error in mm per joint / body part over all sequences of PDT13 dataset.

Joint / body part	Stomach	HipC	HipL	HipR	KneeL	KneeR	Neck	FootL	FootR
Avg. error	77	80	95	101	114	116	70	113	125
Joint / body part	Head	ToesL	ToesR	ShoulderL	ShoulderR	ElbowL	ElbowR	HandL	HandR
Avg. error	96	154	172	97	90	166	146	248	332

Table 2: Average joint positions error in mm per sequence for PDT13 dataset. The column containing M and F + number specifies the subject, the row containing D1-D4 the sequence.

	Kinect SDK				Our approach			
	D1	D2	D3	D4	D1	D2	D3	D4
M1	59	86	138	160	92	129	153	228
M2	38	67	81	183	70	119	117	220
M3	43	85	87	133	69	134	124	154
F1	54	98	112	187	88	142	178	178
F2	36	58	79	130	53	99	105	154
Mean	96				130			

of each fern S to 12, the number of iterations per fern i_F to 32, and the neighborhood radius for pixel offsets R to 80 cm. The complete training takes about 9 hours on a 32 core workstation and the body part classification runs in real-time on a cpu.

As the underlying skeleton differs from ours, we apply a calibration step to remove the constant offset from our estimation to the ground truth. We take our estimate of an ”easy” pose (e.g. T-pose) once for each subject and calculate the offset to the ground truth. We add that offset to all our estimates for all sequences of that subject.

Table 2 shows a comparison with the Kinect SDK, indicating the average joint position error per sequence. We compare our results with those of the Kinect SDK, as both approaches are constructed in a similar manner opposed to the approach of [12], who uses a more complex model fitting process. The reader is referred to the original work for their results. Our approach achieves comparable performance, although the average joint error over all sequences is a bit higher than that of the Kinect SDK, with 130 mm compared to 96 mm.

In table 1 the average error over all sequences is given for each joint / body part. Our system lacks accuracy when it is confronted with challenging limb poses and body rotations leading to occlusions, which results in an increased distance error.

4.2. Results - infant recording

Infants at the age of 3 months are sized around 60 cm in average. We generate 180 K synthetic depth images using a body model of appropriate size. We train an ensemble of ferns containing 15 ferns of depth 13 and set the pixel offset neighborhood radius to 20 cm.

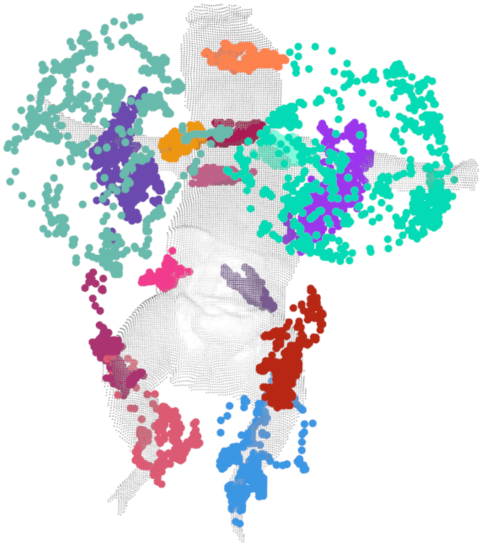
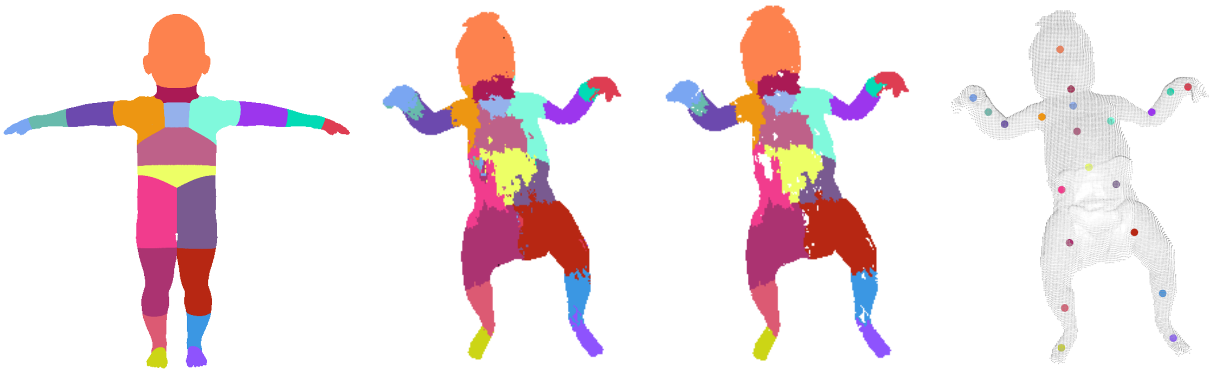


Figure 3: Annotations of joints illustrating infant movements in the recorded sequence. Best viewed in color.

We annotated a recording of an infant of size 60 cm containing 1082 frames. The background of the depth image is removed prior to the pose estimation, so that it only contains the infant. It is lying on the back at a distance of 90 cm to the camera. Figure 3 displays the annotations to illustrate the bodily movements observed in the recording. While the body center remains in a steady position, movements of both arms and legs are observed. The infant in the recording is wearing a diaper, which makes the hips and thighs look much wider than those of the ground truth model. Figure 4 depicts an illustrative sample of the pose estimation. Figure 4a displays the ground truth labels for comparison with the estimated body part labels. In Figure 4b and 4c the estimate and the filtered estimate of body part labels are shown. We can see that the body regions are found in the correct positions and the filtering removes wrong classifications. The size of some regions differs from the ground truth, e.g. the left shoulder, the hips and the knees. The estimated joint positions are displayed in Figure 4d. Table 3 lists the average distance error per joint. The left hand shows the largest average error of 149 mm, followed by the left shoulder with



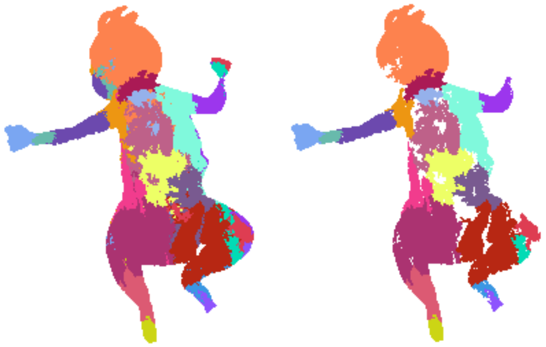
(a) Ground truth labels (b) Estimated body parts (c) Filtered classification (d) Estimated joint positions

Figure 4: Body part classification on infant data. Filtering removes wrong classifications. Estimated left shoulder and knee regions are too big, possibly due to the diaper not being present in the training data. Best viewed in color.

Table 3: Average error in mm per joint / body part in infant recording.

Joint / body part	Head	Neck	ShoulderR	ShoulderL	ElbowR	ElbowL	HandR	HandL
Avg. error	37	20	27	73	24	20	44	149
Joint / body part	Body center	HipR	HipL	KneeR	KneeL	FootR	FootL	**Mean**
Avg. error	30	33	12	45	49	28	30	**41**

73 mm. Figure 6 displays the joint distance error per frame for all joints. The error for the left hand jumps between up to 40 cm and less than 5 cm distance to ground truth. There are two main reasons for the failure cases of the left hand. If the infant pulls up the knee sideways, the filtering sometimes removes the correctly detected hand because there is a bigger region labeled 'hand' on the knee (see Figure 5). This problem occurs in frames 0 to 100 and frames 830 to 1082. As soon as the size of the wrongly detected body part exceeds the size of the correctly detected part, the joint is positioned at the wrong location, resulting in a jump of the distance error. The training data does not contain such poses which may be a reason for the misclassification. The second problem we encounter occurs if the infant puts the hand very close to the body, which is the case in frames 190 to 350. The hand merges with the body, and the predicted hand is positioned on the knee. The system shows a very steady and accurate performance with the distance error of the vast majority of joints hardly exceeding 5 cm. The overall average joint position error is 41 mm.



(a) Before filtering (b) After filtering

Figure 5: Sample of body part estimation with high error for hand. Filtering removes correctly labeled hand as there is a bigger region at the knee with the same label. Left hand is colored in light green, left fingers in red. Best viewed in color.

4.3. Multi-view ferns

To show the flexibility and the potential of our approach we train three ensembles of ferns using data from different views: front (-45 to +45 degrees body rotation), left (-135 to -45 degrees) and right (45 to 135 degrees). The training data is produced using a human body model of size 130 cm. We chose the following parameters for this experiment: the depth of the ferns is 12, we use 15 ferns per ensemble and run 64 iterations per fern. The neighborhood radius is set to 40 cm. Each fern is trained with 60 K synthetic depth images. For evaluation, we manually annotated a short sequence of a child walking from left to right and back. The recording is manually divided into different sections (frontal, left, right, back) according to which side of the child is turned towards the camera. Not every frame

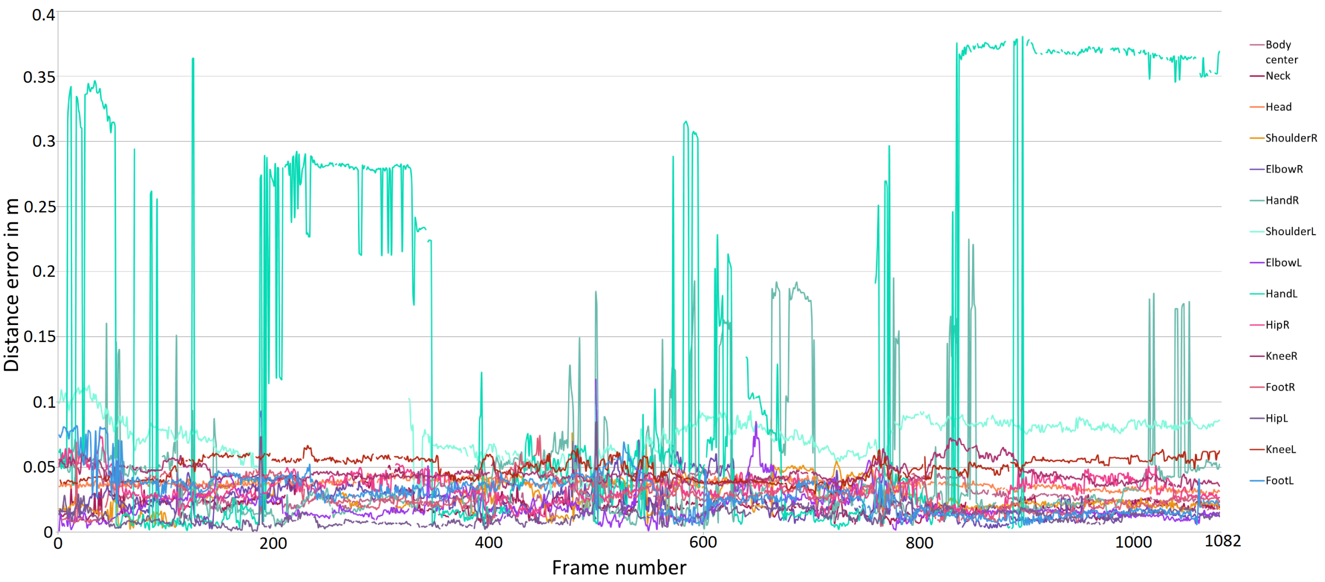


Figure 6: Joint distance error per frame for all joints in infant sequence. Best viewed in color.

of the sequence is annotated, which is why the front / back view is skipped sometimes when turning from left to right (and back). Figure 7 shows how the specialized ferns outperform the others in frames in which the person is seen from the corresponding view. For each section, the respective specialized fern ensemble shows an average joint distance error of 5 to 10 cm, compared to distances of 15 to 25 cm for the others. These preliminary results support the assumption that many specialized classifiers can improve the overall accuracy over one very general classifier. Our system allows fast training and therefore can generate different classifiers quickly to suit varying requirements. We plan to further explore the specialization of fern ensembles to specific tasks.

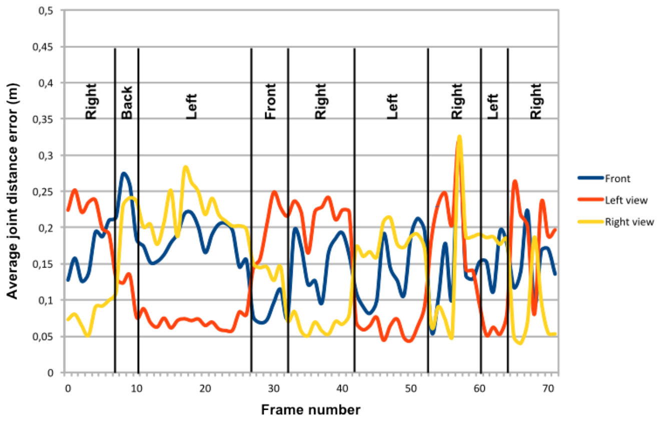


Figure 7: Average joint distance error for three ferns, each trained for a different view (front, left, right). Division of the plot in sections Front, Left, Right, Back was done manually according to what is observed in the recording. Best viewed in color.

5. Conclusion & Future work

We proposed a flexible, fast to train body pose estimation system which can be adapted to varying application requirements. We implemented a data generation pipeline, which lets us produce large amounts of labeled synthetic depth images to feed the training method of the pose estimator. To overcome the computational burden of recently proposed approaches for training pose estimators, we present a system based on random ferns. The training time is reduced by several orders of magnitude compared to existing approaches. It requires 9 hours on a 32 core workstation, opposed to 24 hours on a 1000 core cluster for the Kinect approach using a similar amount of training data. Evaluation on the PDT13 dataset shows comparable results, with an average distance error of 130 mm for our system and 90 mm for the Kinect SDK. Evaluation on a manually annotated recording of an infant containing 1082 frames shows an overall average distance error of 41 mm. The vast majority of joints is steadily estimated with distance errors below 50 mm, only the left hand shows high distance error in certain cases. By applying three view-dependent ferns to a manually annotated sequence of a child, we have shown how using specialized ferns for different views can improve overall accuracy.

This system enables a reliable estimation of the body pose of infants in 3D without restrictions like markers or sensors attached to the body. It is a first step towards markerless 3D motion analysis of infants and will be the core component in an automatic system for early-detection of movement disorder cerebral palsy in 3 month old infants. We demonstrated the flexibility of our system by applying it to three different scenarios: adults, infants and children. There is a wide range of applications in clinical motion assessment that can profit from our system, e.g. analyzing movements of toddlers or persons with missing limbs or other rare bodily conditions for which a general body tracking system like the Kinect fails.

To eliminate the failure cases that appeared in the infant recording, we will apply a scheme that weights the pixelwise estimates depending on their conformity to the kinematic chain of the body model. We intend to further improve the accuracy of the estimation by integrating a tracking component and we plan to combine our discriminative approach with a model-based approach in the manner of [25]. Currently, more extensive studies with more infants are in process. Based on the measured movements, we plan to develop scores to resemble GMA as an aid for doctors in diagnosing cerebral palsy. By building on the proposed system, we aim to achieve an accuracy of pose estimation that suits clinical requirements. We intend to build a tool to be commonly used in clinics and pediatrists offices to increase the number of early detections of movement disorders, so that treatment can be started as soon as possible to minimize the influences of the medical condition.

References

[1] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Consumer Depth Cameras for Computer Vision*, pages 71–98. Springer, 2013. 2

[2] Blender. Free and open 3d creation software. www.blender.org, Sept. 2015. 3

[3] M. Budiu, J. Shotton, D. G. Murray, and M. Finocchio. Parallelizing the training of the kinect body parts labeling algorithm. *Big Learning: Algorithms, Systems and Tools for Learning at Scale*, pages 1–6, 2011. 2

[4] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru. An adaptable system for rgb-d based human body detection and pose estimation. *Journal of Visual Communication and Image Representation*, 25(1):39–52, 2014. 2

[5] C.-Y. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, A. Sawchuk, A. Rizzo, et al. Towards pervasive physical rehabilitation using microsoft kinect. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, pages 159–162. IEEE, 2012. 1

[6] R. A. Clark, Y.-H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & posture*, 36(3):372–377, 2012. 1

[7] C. Einspieler, H. F. Prechtl, F. Ferrari, G. Cioni, and A. F. Bos. The qualitative assessment of general movements in preterm, term and young infantsreview of the methodology. *Early human development*, 50(1):47–60, 1997. 1

[8] M. Gabel, R. Gilad-Bachrach, E. Renshaw, and A. Schuster. Full body gait analysis with kinect. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1964–1967. IEEE, 2012. 1

[9] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 755–762. IEEE, 2010. 2

[10] R. Gross and J. Shi. The cmu motion of body (mobo) database. 2001. 3

[11] F. Heinze, K. Hesels, N. Breitbach-Faller, T. Schmitz-Rode, and C. Disselhorst-Klug. Movement analysis by accelerometry of newborns and infants for the early detection of movement disorders due to infantile cerebral palsy. *Medical & biological engineering & computing*, 48(8):765–772, 2010. 2

[12] T. Helten, A. Baak, G. Bharaj, M. Muller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 279–286. IEEE, 2013. 2, 4, 5

[13] D. Karch, K.-S. Kim, K. Wochner, J. Pietz, H. Dickhaus, and H. Philippi. Quantification of the segmental kinematics of spontaneous infant movements. *Journal of biomechanics*, 41(13):2860–2867, 2008. 2

[14] B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. S. Rizzo, and M. Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1831–1834. IEEE, 2011. 1

[15] MakeHuman. Open source tool for making 3d characters. www.makehuman.org, Sept. 2015. 3

[16] L. Meinecke, N. Breitbach-Faller, C. Bartz, R. Damen, G. Rau, and C. Disselhorst-Klug. Movement analysis in the early detection of newborns at risk for developing spasticity due to infantile cerebral palsy. *Human movement science*, 25(2):125–144, 2006. 2

[17] Microsoft. Kinect for windows. https://dev.windows.com/en-us/kinect, Sept. 2015. 1

[18] Microsoft. Xbox 360 kinect sensor. http://support.xbox.com/en-GB/xbox-360/kinect/body-tracking-troubleshoot, Sept. 2015. 1

[19] M. D. Olsen, A. Herskind, J. B. Nielsen, and R. R. Paulsen. Model-based motion tracking of infants. In *Computer Vision-ECCV 2014 Workshops*, pages 673–685. Springer, 2014. 2

[20] M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. Ieee, 2007. 2, 3

[21] H. Rahmati, O. M. Aamo, O. Stavdahl, R. Dragon, and L. Adde. Video-based early cerebral palsy prediction using motion segmentation. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 3779–3783. IEEE, 2014. 2

[22] P. Rosenbaum, N. Paneth, A. Leviton, M. Goldstein, M. Bax, D. Damiano, B. Dan, B. Jacobsson, et al. A report: the definition and classification of cerebral palsy april 2006. *Dev Med Child Neurol Suppl*, 109(suppl 109):8–14, 2007. 1

[23] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304. IEEE Computer Society, 2011. 2

[24] A. Stahl, C. Schellewald, Ø. Stavdahl, O. M. Aamo, L. Adde, and H. Kirkerød. An optical flow-based method to predict infantile cerebral palsy. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(4):605–614, 2012. 2

[25] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE, 2012. 2, 8

[26] R. Wang, G. Medioni, C. J. Winstein, and C. Blanco. Home monitoring musculo-skeletal disorders with a single 3d sensor. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 521–528. IEEE, 2013. 1

[27] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2353–2360. IEEE, 2014. 2

[28] H. Yub Jung, S. Lee, Y. Seok Heo, and I. Dong Yun. Random tree walk toward instantaneous 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2015. 2