Supplementary Material: Decision Tree Fields

Note, the supplementary material is not needed to understand the main paper.

Sebastian Nowozin
Microsoft Research
Cambridge, UK
Sebastian.Nowozin@microsoft.com

Carsten Rother
Microsoft Research
Cambridge, UK
carrot@microsoft.com

Shai Bagon
Weizmann Institute
shai.bagon@weizmann.ac.il

Toby Sharp
Microsoft Research
Cambridge, UK
toby.sharp@microsoft.com

Bangpeng Yao
Stanford University
Stanford, CA, USA
bangpeng@cs.stanford.edu

Pushmeet Kohli
Microsoft Research
Cambridge, UK
pkohli@microsoft.com

Abstract

As mentioned in the paper, in this supplementary document we describe experimental details that were omitted from the main paper for reasons of clarity and space. Additionally, we remind the reader of the the Gibbs sampler and the minimization via Simulated annealing (SA) that we use for inference.

1. Additional Experiment: Generic Object Class Recognition

As mentioned in the first paragraph of Section 5 in the main paper, we evaluated our DTF model on one other application, which was generic Object Class Recognition. We did not include these results in the main paper since our initial conclusion is that for this application DTF does not outperform a standard CRF approach, as we explain next.

We used the DAGS scene understanding data set [2]. In this data set, there are 715 images with each pixel labeled as one of eight classes (sky, tree, road, etc.); a labeling decomposes the image into semantic parts. The data set is partitioned to five folds of training and test images. For each fold there are 572 training images and 143 test images, and the benchmark measure is the cross-validated multiclass per-pixel accuracy.

For a simple 4-neighborhood CRF with contrast-sensitive potential we obtain results of 67.5%. Because we use only simple pixel-difference features and no region-based features, our results are below state-of the art of 79.42% reported in [5]. Moreover, we obtain the same performance with a DTF using the same 4-neighborhood structure but learned conditionally. Upon closer inspection we see that we learned contrast sensitive term. Initial tests on using a densely connected factor graph did not show improvements. We conjecture that in contrast to our other applications, the reasons may be: a) that the DAGS data set does not contain enough structure in the label set, or, b) there may exist such structure but given the variability in the task, the amount of training images is too small to be able to discover this structure.

We plan to investigate other challenging semantic segmentation data sets, such as the PASCAL VOC segmentation set, in the near future.

2. Experimental Details, Additions

2.1. Snakes Experiment

The unary model consists of 10 decision trees of depth 25 with tests that check whether a pixel at a fixed relative position is of a certain color (here 5 possible colors). Note, more trees or deeper trees lead to over-fitting, hence we assume that this is the best possible setting given the training database. For the MRF, we learn a total of two 10-by-10 tables of energy values, one for the horizontal and vertical factor, respectively. In our model, the MRF corresponds to a DTF with decision trees of depth one. For the DTF, we use four decision trees of depth 15.

The pairwise decision trees perform tests on the colors of two pixels within a 3-by-3 window. During training of the decision trees (both unary and pairwise) we evaluate all possible decision tree tests, which is possible since images are small. After decision tree training, the model weights are optimized using (4) over a range of different σ_t values for the pairwise factors.

We use TRW to obtain the MAP labelling for the test images, and Gibbs sampler to obtain samples for unary model.

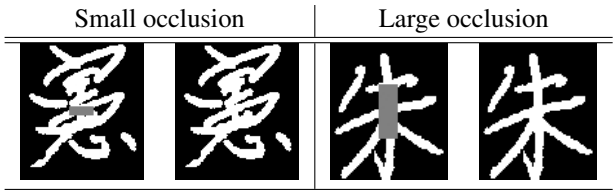

Figure 1. Completing small and large occluded boxes of Chinese characters: test (left) and ground truth (right).

2.2. Chinese Characters Experiment

We use the KAIST Hanja2 database, available at http://ai.kaist.ac.kr/Resource/dbase/Hanja/HanjaDB2.htm.

We have two data sets, containing small and large occlusions, respectively. We draw the width and height of the occluding box from a uniform distribution over $\{a_{\min}, \ldots, a_{\max}\}$. In the small occlusion setting we have a "small occlusion" box with $(a_{\min} = 5, a_{\max} = 20)$, and one with a "large occlusion" box with $(a_{\min} = 10, a_{\max} = 40)$. Figure 1 shows examples of large and small occlusions on test characters. The typical image in this data set is 80×100 pixels in size. We use a training set of 300 images and a disjoint test set of 100 images. The charcters in the training set are shown in figures 8.

The features for the unary and pairwise decision trees are simple: they can test whether a pixel at a relative offset is black, grey, or white. The pairwise features can test all pairs of combinations, nine in total, for two image pixels located at relative offsets of the potential variables. For each decision tree node we propose 2000 tests from a uniform random distribution, where each test is allows to look up to 80 pixels away, incorporating global context cues in both the unary and pairwise interactions. The maximum decision tree depth is 15 for the unaries, and 6 for the pairwise trees. We use a random subsampling with ratio of 0.5, meaning that on average every second pixel in the training image is used during training. For the prior parameters, we fix $\sigma_u = 1$ for the unary interactions, and select the pairwise prior parameter $\sigma_{pw} \in \{0.1, 0.01, 0.001\}$.

Training is very efficient; for the 300 training images and the most complex model (DTF, pairwise tree depth six) we have 11150 parameters and the entire training, including learning the decision trees and pseudolikelihood optimization, takes less than one hour.

For inference we run a Gibbs sampler for 50 burn-in sweeps and 200 sampling sweeps to obtain the posterior and MPM predictions. For MAP inference we use simulated annealing for a total of 200 sweeps, starting from a temperature of 20.0 down to 0.05, see Section 3.2.

2.3. Body-part Recognition

The task of body-part recognition reported by [6] takes depth images as input and assign each foreground pixel to one of 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).

We used the same experimental set as [6], here with 30 or 1500 training images, and 150 test images. Hence we assume that the mask of the person was already extracted in a pre-processing stage. Also, we randomly sample ≈ 2000 pixels from each training image. Figure 9 shows the 30 depth images used in the smaller training set.

The features for the body parts recognition performs simple depth comparisons (in the same fashion as the features reported in [6]). For a unary feature (acting on a single location x):

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

where $d_I(x)$ is the depth at pixel x in image I, and the feature parameters $\theta = (u, v)$ are the offsets for the depth difference test. In the same fashion we compute a pair-wise feature (acting on a pair of pixels x_1, and x_2):

$$f_\theta(I, x_1, x_2) = d_I\left(x_1 + \frac{u}{d_I(x_1)}\right) - d_I\left(x_2 + \frac{v}{d_I(x_2)}\right)$$

Note that the relative displacement between x_1 and x_2 is determined by the factor structure.

During tree learning we randomly sample offsets $\theta = (u, v)$ from a normal distribution. For each sample θ we explore 20 possible thresholds. For the prior parameters, we fix $\sigma_u = 1$ for the unary interactions, and varied the pair-wise interaction prior $\sigma_p \in \{0.1, 0.2, 0.5\}$ The depth of the decision trees for the unary interactions is fixed to 16 for the small training data set (30 images), and to 20 for the large training data set (1500 images).

For inference we used TRW [4] to compute MAP assignment for the different body parts. For our full connectivity model (+1,5,20) inference takes an average of 30 seconds for an input image of size $\approx 150 \times 150$ pixels. We believe that we can obtain additional speedups by avoiding the explicit unrolling of the factor graph; however, in this work we address the training problem and leave this extension to future work.

Additional qualitative examples of our body-part recognition results can be visualized in Figures 3 and 4. Note how varied the different poses of the input images are. The data set contains a large variety of different poses and body sizes, making this task challenging.

Varying tree depth. To demonstrate the improvement in recognition performance gained by using conditional pair-wise information we varied the pair-wise tree depth for

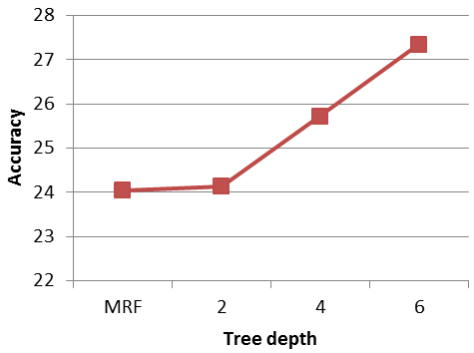

Figure 2. Body parts recognition results (30 training images): For neighborhood size +1,5,20 we increase the depth of the pair-wise trees from 1 (MRF) to 6, and record the resulting accuracy.

our most complex configuration (+1, 5, 20) from depth 1 (MRF) to depth 6. Results are shown in Figure 2: the deeper the trees, the better they are able to capture conditional pair-wise interactions, and recognition performance, accordingly, increase.

Comparison to Random Forests. As in previous experiments, we compared DTF with Random Forests. In this case we have also used Random Forests to heuristically set conditional pairwise terms. This is done by simply using the empirical histogram of the training data as probability distribution for the pairwise terms. The weight is then the negative log probability. Furthermore, the global weighting of the different pairwise factors is done optimally using the test data.

The results are summarized in table 1. We see that Random Forests consistently perform worse than DTFs. This is not surprising since random forests ignore the problem of "over-counting", i.e. ignore the fact that the same random variable is present in different pairwise terms.

A comment concerning the comparison with [6]. For unaries only we achieve 19.79% accuracy using a random forest, while [6] achieves 14.8%. The difference stems from the fact that [6] uses for each tree a subset of the training data, which is sub-optimal for small training data sets.

Model	unary	+1	+1,20	+1,5,20
Random Forest	19.79	20.91	20.79	23.39
DTF	21.36	23.71	25.72	27.35

Table 1. Comparison of Random Forests with DTFs for Body-part recognition (30 training images).

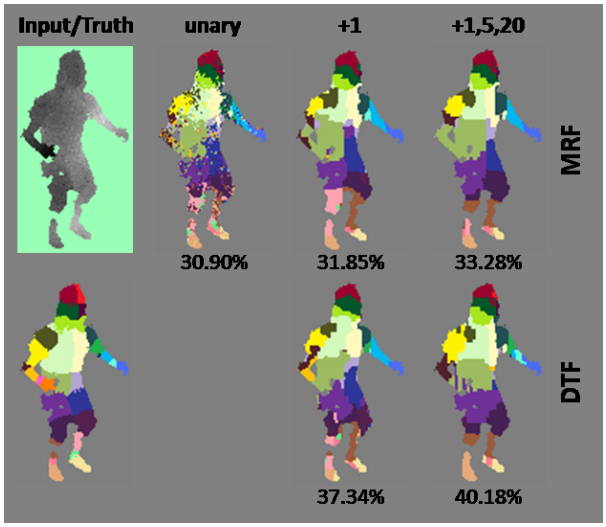

Figure 3. Body-parts recognition results (30 training images). MRF (top), DTF (bottom). Accuracy is shown below each result. Note the improvement in labeling the arm and elbow parts. (Notice that DTF unary is the same configuration as MRF unary).

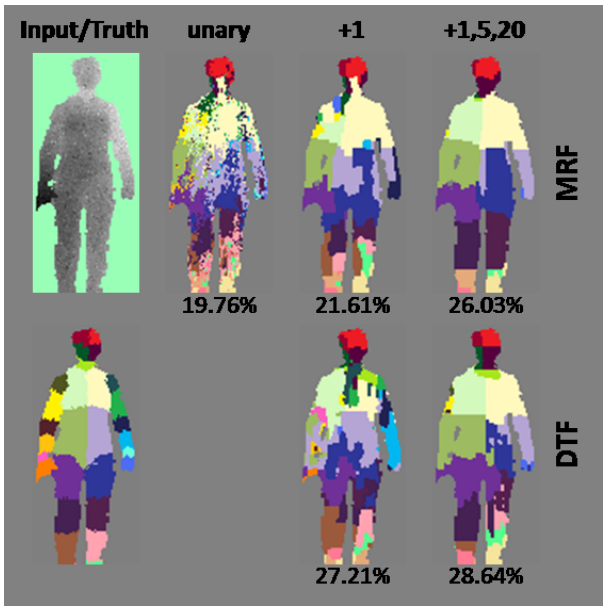

Figure 4. Body-parts recognition results (30 training images). MRF (top), DTF (bottom). Accuracy is shown below each result. Note the improvement in labeling of the shoulder and upper torso.

3. Inference Methods

3.1. Gibbs Sampler

We use Gibbs sampling as introduced by Geman and Geman [1] to obtain approximate samples from our model.

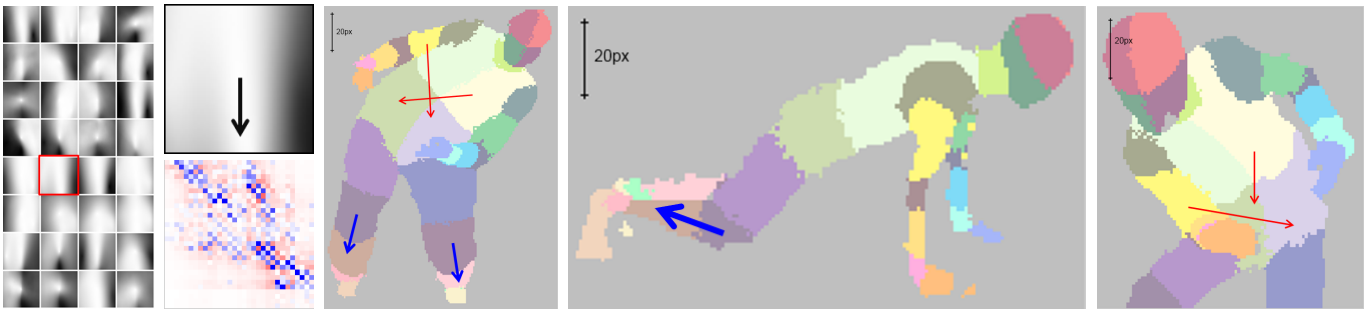

Figure 5. **Illustrating one learned vertical interaction (20 pixels apart):** The average depth-normalized silhouette reaching one of the 32 leaf nodes in the learned decision tree for the vertical pairwise interaction is shown on the left. Note how these patterns differ from those of Figure 10 in the main paper. For one specific leaf node (marked in red) the corresponding pattern and learned weight matrix is shown in the second column. The top two attractive terms (blue) and repulsive terms (red) are illustrated as arrows on poses taken from the test set (right). The first pose shows how the knee on top of the lower leg (both left and right) are plausible (vertical) configurations where the learned patch is matched. However, for the second pose the leaf pattern is not matched and indeed this (not-vertical) configuration is no longer valid. The first and third pose show that the left-arm-over-right-torso, and right-upper-torso-over-left-lower-torso are plausible (vertical) configurations, but only when the leaf pattern is not matched. Therefore these (vertical) configuration are inhibited for this specific leaf.

Algorithm 1 Gibbs Sampler

GIBBSSAMPLER($\tilde{p}$)
Input:
  $\tilde{p}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) \propto p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$, unnormalized target distribution,
  B, number of burn-in sweeps,
  T, number of sample sweeps.
Output:
  $y^{(t)}$, sample sequence with $y^{(t)} \sim p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$
Algorithm:
$y^{(0)} \leftarrow$ arbitrary in $\mathcal{Y}$
$\alpha \leftarrow \exp(-\log(20)/B)$
for $b = 0,\ldots,B$ **do**
  $\tau \leftarrow \alpha^b 20$ {Reduce temperature towards 1}
  for $i \in V$ **do**
    Sample $y_i^{(0)} \sim \tilde{p}_\tau(y_i|y^{(0)}_{V\setminus\{i\}}, x, w)$ using (1)
  end for
end for
for $t = 1,\ldots,T$ **do**
  $y^{(t)} \leftarrow y^{(t-1)}$
  for $i \in V$ **do**
    Sample $y_i^{(t)} \sim \tilde{p}_1(y_i|y^{(t)}_{V\setminus\{i\}}, x, w)$ using (1)
  end for
  output $y^{(t)}$
end for

Let us use (2) from the main paper to define an unnormalized temperized distribution,

$$\tilde{p}_\tau(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) := \exp(-\frac{1}{\tau}E(\boldsymbol{y},\boldsymbol{x},\boldsymbol{w})),$$

where $\tau > 0$ is a *temperature* parameter. For $\tau = 1$ we have $\tilde{p}_\tau(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) \propto p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$.

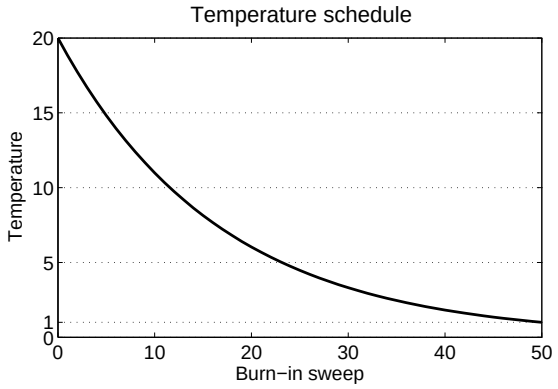

Figure 7. Temperature schedule used for the burn-in phase of the Gibbs sampler and during simulated annealing.

Algorithm 1 is the basic Gibbs sampler, consisting of a burn-in phase, and a sampling phase. Sampling each variable once is called a *sweep*.

In the algorithm, sampling from the conditional distribution is feasible because it only requires the unnormalized distribution $\tilde{p}$ and normalization over the domain of a single variable. This is shown in Figure 6.

In the algorithm, we use $B + 1$ sweeps during the so called "burn-in phase", aimed at diminishing the influence of the starting point. The temperature is initially set to a high value (20.0) and gradually decreased towards 1, so as to recover the true distribution. This heuristic is very effective at approximately placing $y^{(0)}$ at a high-mass region of the label space. Figure 7 shows the resulting annealing schedule for 50 burn-in sweeps.

Although we have not implemented this, the Gibbs sampler can be parallelized as well by partitioning the variable

$$\tilde{p}_\tau(y_i|y^{(t)}_{V\setminus\{i\}},x,w)=\frac{\tilde{p}_\tau(y_i,y^{(t)}_{V\setminus\{i\}}|x,w)}{\sum_{y_i\in\mathcal{Y}_i}\tilde{p}_\tau(y_i,y^{(t)}_{V\setminus\{i\}}|x,w)}=\frac{\sum_{F\in M(i)}\exp(-\frac{1}{\tau}E_F(y_i,y^{(t)}_{F\setminus\{i\}},x_F,w_{t_F}))}{\sum_{y_i\in\mathcal{Y}_i}\sum_{F\in M(i)}\exp(-\frac{1}{\tau}E_F(y_i,y^{(t)}_{F\setminus\{i\}},x_F,w_{t_F}))} \quad (1)$$

Figure 6. Gibbs sampling updates around a variable $i \in \mathcal{V}$. All factors not in $M(i)$ appear in both the numerator and denominator and therefore do not influence the ratio. Note the similarity between (1) here and (5) from the main paper. Due to this similarity, we can use the same code to compute both equations efficiently.

set $\mathcal{V}$ into disjoint subsets such that no pairwise or higher-order interaction is present between any two variables contained in the same set. This can be achieved by approximately solving a graph coloring problem on a small auxiliary graph. We are currently investigating this further and will report results in a future report.

3.2. Simulated Annealing

For very small temperatures the probability mass in $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$ will become concentrated at the MAP state $\boldsymbol{y}^*$. If we reduce the temperature from a high value, say $\tau_{\text{start}} = 20$ to a very small one, say $\tau_{\text{end}} = 0.05$ while running the Gibbs sampler, then we will obtain an approximate MAP state. This is the idea of simulated annealing [3, 1]; one implementation of simulated annealing is shown in Algorithm 2.

Algorithm 2 Simulated Annealing MAP Inference

SIMULATEDANNEALINGINFERENCE($\tilde{p}$)
Input:
  $\tilde{p}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) \propto p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$, unnormalized target distribution,
  B, number of sweeps,
  $(\tau_{\text{start}}, \tau_{\text{end}})$, initial and final temperatures.
Output:
  y^*, approximate MAP state
Algorithm:
$y \leftarrow$ arbitrary in $\mathcal{Y}$
$\alpha \leftarrow \exp(\log(\tau_{\text{end}}/\tau_{\text{start}})/B)$
for $b = 0, \ldots, B$ **do**
  $\tau \leftarrow \alpha^b \tau_{\text{start}}$ {Reduce temperature towards τ_{end}}
  for $i \in V$ **do**
    Sample $y_i \sim \tilde{p}_\tau(y_i|y_{V\setminus\{i\}},x,w)$ using (1)
  end for
end for
$y^* \leftarrow y$

3.3. A Comment on the Inference Problem

The difficulty of the test-time inference problem in the DTF model depends on the task.

In the *body parts experiment* we can solve the MAP inference problem well by tree-reweighted message passing (TRW) [4], as indicated by a small primal-dual gap.

For the *Chinese character task*, the learned model has repulsive potentials, complicating MAP inference. In that case, for one 100-by-100 degree-27 graph it takes $< 30s$ to carry out 200 simulated annealing sweeps, and $< 10s$ to approximately minimize the energy using TRW. For the quality of the approximate minimizers obtained, in 59% of the cases E(SA) $<$ E(TRW), in 15% of the cases E(GT) $<$ E(TRW), and in 2% of the cases E(GT) $<$ E(SA), where GT is the ground truth labeling. This indicates that the energy minimization is not optimal. We plan to release a set of these learned inference problems as an energy minimization benchmark.

References

[1] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6:721–741, 1984. 3, 5

[2] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 1

[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983. 5

[4] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell*, 28(10):1568–1583, 2006. 2, 5

[5] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, 2010. 1

[6] 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 a single depth image. In *CVPR*, 2011. 2, 3

Figure 8. Chinese characters: training set with 300 characters.

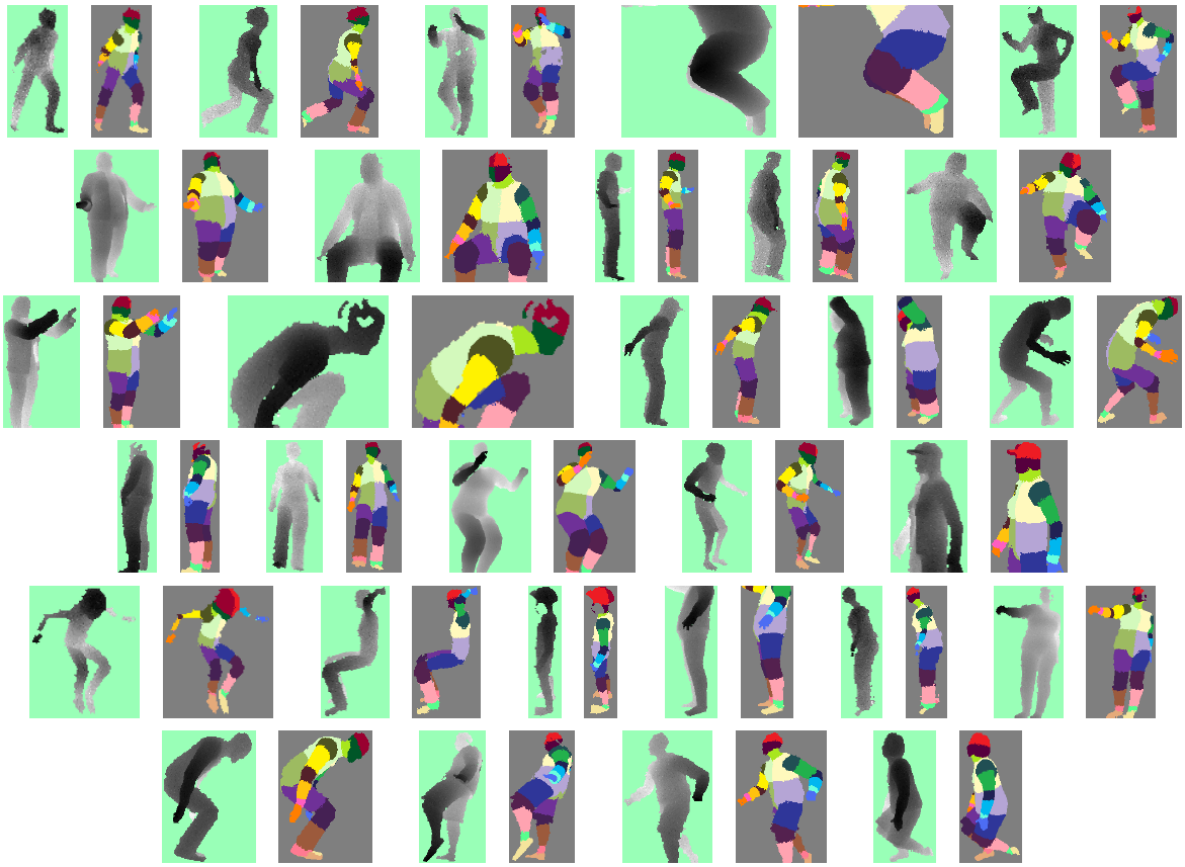

Figure 9. Train set used for body-part recognition. Depth map next to ground-truth labeling.