Easy Minimax Estimation with Random Forests for Human Pose Estimation

P. Daphne Tsatsoulis and David Forsyth

Department of Computer Science
University of Illinois at Urbana-Champaign
{tsatsou2, daf}@illinois.edu

Abstract. We describe a method for human parsing that is straightforward and competes with state-of-the-art performance on standard datasets. Unlike the state-of-the-art, our method does not search for individual body parts or poselets. Instead, a regression forest is used to predict a body configuration in body-space. The output of this regression forest is then combined in a novel way. Instead of averaging the output of each tree in the forest we use minimax to calculate optimal weights for the trees. This optimal weighting improves performance on rare poses and improves the generalization of our method to different datasets. Our paper demonstrates the unique advantage of random forest representations: minimax estimation is straightforward with no significant retraining burden.

Keywords: Human pose estimation, regression, regression forests, minimax

1 Introduction

In this paper, we address the problem of human pose estimation from a single RGB image. Pose estimation (or parsing) takes an image that is known to contain a person and reports locations of some body parts (typically head, torso, upper arms, lower arms, upper and lower legs). Parsing is a core vision problem with a long history. Despite much recent activity[1–6], it remains very difficult to produce fast, accurate parses.

All current parsers use local part models of some form to identify body layout. Unlike existing methods that require complex inference or search, we train a regression forest to predict multiple poses for a test image. This set of proposals, depicted in Figure 1, defines the problems of finding the best prediction for an example or of combining the predictions in an optimal way.

In our paper we apply a minimax optimization that improves our method's accuracy and generalizability. We explain how regression forests have a unique advantage over other methods since minimax estimation is easy to apply to their outputs without re-training. Our experiments show the sharp improvements made by the minimax optimization on rare poses and on different datasets.

Contributions: Our approach to pose estimation is simple and compares to state-of-the-art results. By applying the minimax optimization method to the output of regression forests we have presented a new approach to pose estimation that finds an optimal combination of pose predictions. This lets us significantly improve the accuracy and the generalization of our results on hard datasets.

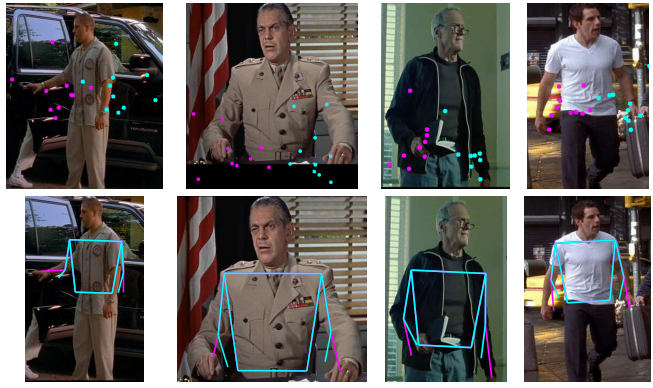

Table 1. (top) By utilizing regression forests our system is able to make multiple predictions for a joint location in an image. Images in the top row are from the FLIC dataset and show the original predictions made by each tree in a 10-tree forest for right (magenta) and left (cyan) wrists. (bottom) The advantage to using regression forests is that their output is quickly and easily recombined to improve predictions. We present an optimization over the forests that predicts an alternative weighting for the trees. The uniformly-weighted prediction over the tree outputs have been marked in cyan and the optimally re-weighted predictions in magenta.

2 Background

For single images, it is usual to model body segments with rectangles. One then parses by solving an optimization problem to place these segments on the image.

Felzenszwalb and Huttenlocher showed how to produce highly efficient matchers for tree-structured models [7] (see also [8]). Most work since then assumes a tree-structured model (e.g. [6, 9]). There have been efforts to build models beyond trees, e.g. loopy graphs [10–12], mixtures of trees [8, 13], and fully connected graphs [3], and there is some evidence that the advantages due to such a model outweigh the disadvantages of approximate inference.

The key challenge of using full relational models is how to perform learning and inference efficiently. This has led to intense activity exploring pruning strategies [14–16, 3, 5]. All part-based parsers must manage a tradeoff: limb appearance models that are specialized to a particular image tend to yield better

parses (for example, one can exploit the color of clothing), but obtaining such a specialized appearance model from a single image is hard to do. Various strategies have evolved, including re-estimation of the appearance model [9], strong priors on appearance [17], and using small, generic models [1].

One alternative is to look at the body on a longer spatial scale. Bourdev et al. [18] suggest that one should directly detect *poselets* — stylized structures on the body — then infer body configuration (respectively segmentation of the body [19]; body attributes [20]) from those poselets. One could produce a parse by directly decoding poselet response vectors [21], or by insisting on consistency between body segment detectors, local poselets, and long-scale poselets [2]. Another alternative is to look at small parts on a shorter spatial scale. This allows for the foreshortening of body segments, for example. Yang et al. show high accuracy parses derived from small parts [1]. There is some evidence that each method benefits from slightly different image information, so that a fused method could offer improved accuracy. Jammalamadaka et al. demonstrate that image based estimators can tell reliably whether a parser has succeeded or not [22].

Shotton et al. use regression forests in [23]. They construct 3 trees in order to predict pose from RGB-D data. In [24] regression trees were used as part of a tree-structured model to represent the unary potential. In neither of these cases was the output of a forest optimized to improve the generalization of the method as we have done. How well a method generalizes is a very important aspect of evaluation, as shown by the discussion in [25].

While image parsing itself is viewed as an important problem, there has been work demonstrating its applications. Ramanan et al. demonstrate a recipe for turning a parser into a tracker [26]. Ferrari et al. demonstrate search for human figures in video using pose as a search term [27]. Jammalamadaka et al. demonstrate improvements on this recipe, showing a variety of ways to query including a Kinect parse [22]. The parsing community has shown little interest in recovering 3D representations, though 3D estimates are known to be available from joint positions [28, 29]. Ikizler and Forsyth show an application of such 3D representations in activity recognition [30]. When depth data is available, parsing is largely solved [23].

3 Method

We approach parsing with a unique construct. Our method has three steps. First, we register all images into a normalized coordinate system and extract image features. Second, we run a regression forest over the features to produce multiple joint-location predictions. Third, we propose a new way to combine the predictions that optimizes performance and generalizability.

3.1 Body-Space

The datasets we use provide an annotation of joint locations with each image. We consider the right and left shoulder, elbow, wrist, and hip joints in our method.

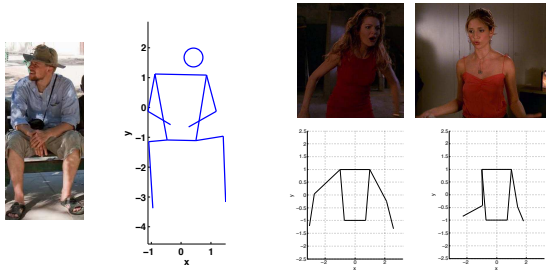

Fig. 1. (left) Example of joints projected into body-space. The torso is projected onto the xy-plane and aligned so that the shoulders are parallel to the x-axis. One unit in the x-dimension is equal to the distance between the neck and shoulders of the person. One unit in the y-dimension is equal to the distance between the shoulders and hips. (right) These two images have the right wrists in the same locations in the image, but in very different locations with respect to the body.

We want to be able to predict human pose in body-space rather than in an image. Body space is a normalized coordinate system that is centered at the torso and that is scale- and viewpoint-invarient.

We want to be able to predict pose in body-space because it is more interesting and informative to know what someone is doing relative to their body instead of at a specific pixel. With the number of monitoring/surveillance applications that want to know how people are interacting with scenes and with each other, it is more important to know where people's limbs are relative to objects (and themselves) than where in a picture their hands are located. Being able to do this in 3-dimensions is even more important but many current hardware systems only provide 2-dimensional images. By predicting joints in body-space we can construct a method gives real body poses in 2- or 3-dimensions.

We register the original joint locations (given as pixel coordinates) so that the origin lies at the center of the torso. One unit in the body-space's x-axis corresponds to the distance between the neck and a shoulder and one unit on the body-space's y-axis corresponds to the distance between the neck and center of the torso. An example of the effects of this projection can be seen in Figure 1. Images are registered by computing the rotation, translation, and scale needed to project shoulders and hips in pixel-coordinates to a target torso in body-space.

When preprocessing 3-d annotated datasets, such as the H3D dataset, joints were projected into a 3-dimensional body-space by rotation, scale, and translation. In this space, the z-axis corresponds to the dimension going in front of and behind the body. Examples of these projections can be seen in Figure 6.

3.2 Regression Forests

Previous work in pose estimation has run inference over a tree-structured model or searched images for local patches. Our method does neither of these. It instead builds a regression forest to predict joint locations for an example image.

Unlike template-based methods that have to learn a unique representation for every joint configuration a tree can build-in unique visual representations automatically. For example, a hand can be turned in many different directions whilst being in the same location. Discriminative methods need to learn a template for each orientation of the hand in order to make a prediction. Since trees can have multiple leaves with the same output value they are able to capture the fact that visually different examples can map to the same output with no additional effort.

Regression forests are an ensemble of trees. Each of t trees is constructed using a random sampling, with replacement, of the training data. Trees split the training data on a feature dimension f_i until a leaf contains a minimum number of elements and is not split again. At each split, a subset of $\sqrt{d}$ features (where d is the dimensionality of the feature vector) is considered.

Let $\mathbf{y}_i \in \mathbb{R}^d$ be the vector representing the joint being predicted for example i. Note that $\mathbf{y}_i$ can take on any dimensionality so it can represent a 2- or 3-joint location or it can represent a number of joints such as the right wrist and elbow. Let $\mathbf{y}_{i,t}$ be the prediction for example i made by tree t. Our method makes a final prediction, $\mathbf{y}_i^*$ by calculating the summed weight of all tree predictions:

$$\mathbf{y}_i^* = \sum_{t=1}^{T} \frac{1}{w_t} \mathbf{y}_{i,t} \tag{1}$$

w_t is the weight of each tree. Initially we weight the trees uniformly, as is typically done in regression forests.

The regression forest predicts poses in body-space, we project (using rotation, translation, and scale) the images back into pixel-space for comparison. We do this by projecting our body-space torso onto a predicted torso in pixel-space. For each dataset we used torso predictions that were provided by the authors, please see the Experiments and Results section for more detail.

3.3 Minimax

There is good evidence that in most vision tasks training and test sets are not sampled from the same distribution [25]. This is a particular nuisance in human parsing, where performance falls off when a method is trained on one dataset and tested on another (eg [3]). One standard method for obtaining a well-behaved estimator in a situation like this is minimax. The recipe is as follows: Assume we have a model of the family of possible test distributions, parametrized by a parameter vector $\mathbf{l}$. Assume we have a parametric family of estimators, parametrized by a vector $\mathbf{w}$. We cannot know the true value of $\mathbf{l}$, because we do not know the true test distribution. However, we can assume that the true test distribution is "not too far" from the training distribution. We can then search the collection of distributions that are "not too far" and the estimator parameters for a pair $(\mathbf{l}, \mathbf{w})$ such that the error on the worst test distribution is best. One can visualize this as a problem with two players: the

Table 2. These are example images of the FLIC training set. Images on the left are images that were up-weighted during an iteration of minimax when optimizing for the right wrist (right-most wrist in the image). These are images that were difficult to solve, and therefore given more weight in the re-weighting. The set of images on the right are those that the optimization function found easy and down-weighted.

first chooses a test distribution $\mathbf{l}$ to make the performance as poor as possible, and the second then chooses a $\mathbf{w}$ to produce as good as possible performance on the worst test distribution. Write $\mathcal{E}_{\mathbf{l}}$ for an expectation with respect to $\mathbf{l}$. The search becomes $\min_{\mathbf{w}} (\max_{\mathbf{l}} (\mathcal{E}_{\mathbf{l}} [\text{Loss}(\mathbf{w})]))$ subject to an appropriate notion of "not too far" on the test distribution. This approach has not been used in detection or parsing to our knowledge. Although building a representation of a family of test distributions is straightforward (one reweights the training samples to get a different empirical training distribution), estimating $\mathbf{w}$ is hard because it creates a massive re-training burden. At each stage of the search, we would need to retrain the estimator to get the best loss on the current worst distribution. For most representations, this burden is unmanageable.

A unique feature of regression forests, not to our knowledge previously remarked, is that they make minimax estimation easy. Assume that the test distribution is"not too far" from the training distribution. Then the structure of the trees in the forest (which variables are split, etc.) is unlikely to be affected by the difference, because it is strongly random. We can then define a reasonable parametric family of estimators by simply reweighting the trees in the forest, so $\mathbf{w}$ is a non-negative weight vector. The appropriate notion of"not too far" is to bound above the k-l divergence of $\mathbf{l}$ from the uniform distribution; alternatively, one could bound the entropy of $\mathbf{l}$ below. Each generates unpleasant non-linear constraints. We relax these constraints by requiring $\frac{1}{kN} \leq l_i \leq \frac{k}{N}$ and $\sum_i l_i = 1$ for each component l_i of $\mathbf{l}$ and a parameter k. This prevents

any example being over or under weighted. We define the loss of a particular weighting of a forest to be the squared error, so the objective function becomes $\min_{\mathbf{w}} \left(\max_{\mathbf{l}} \left(\sum_i l_i \left[(\mathbf{y}_i - \sum_j w_j T_j(\mathbf{x}))^2\right]\right)\right)$ where $w_j \geq 0$. No retraining of trees is required, and the resulting optimization problem is straightforward to deal with (below). In practice, we show that minimax tends to weight up difficult examples and to weight down common ones (the dataset player is trying to make it difficult to predict good solutions; Figure 2). Minimax is known to be severely pessimistic for some problems; in this case, it offers real improvements in performance. We believe this is the result of constraining $\mathbf{l}$. The constraints on $\mathbf{l}$ allow minimax to emphasize hard examples, but prevent it from putting all weight on the single most difficult example. As a result, the estimator is more willing to make risky predictions. This improves performance for lower arms, which move around a lot and where the distribution of configurations is poorly represented in training data.

Optimization Problem: Let $\mathbf{w} \in \mathbb{R}^t$ be the weighting over the t trees, $\mathbf{l} \in \mathbb{R}^n$ be the weighting over n training examples, $\mathbf{y} \in \mathbb{R}^{n \times d}$ be the correct location, and $\mathbf{M} \in \mathbb{R}^{t \times n \times d}$ be the predicted location made by each tree for every example.

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \ \max_{\mathbf{l}} \sum_{i=1}^{n} l_i ||\mathbf{y}_i - \mathbf{w}^T \mathbf{M}_i||_2^2 \tag{2}$$

$$\text{s. t.} \quad \sum_{i=1}^{n} l_i = 1 \tag{3}$$

$$\frac{1}{kn} \leq l_i \leq \frac{k}{n} \quad k \in \mathbb{R} \tag{4}$$

$$\mathbf{w} > 0$$

Where $\mathbf{M}_i \in \mathbb{R}^{t \times d}$ is the t tree predictions made for example i, $\mathbf{y}_i \in \mathbb{R}^{1 \times d}$ is the correct location for example i, and $l_i \in \mathbb{R}$ is the weight given to example i.

Solving the Optimization: Equation 2 is tractable in $\mathbf{w}$ and it is tractable in $\mathbf{l}$. To solve the optimization we iteratively solved for each variable separately. We began by fixing $\mathbf{l}$ and $\mathbf{w}$ to be uniform weights and solved for an optimal $\mathbf{l}^*$. We then set $\mathbf{l} = \mathbf{l}^*$ and solved for $\mathbf{w}^*$. We iterate until $\mathbf{w}^*$ converges or until a maximum number of iterations has been reached.

This new weighting $\mathbf{w}^*$ is then applied in Equation 1.

4 Experiments and Results

Datasets We evaluate our method on three datasets: Buffy Stickmen V3.01 [4], FLIC dataset (the smaller version of the dataset with approximately 5000 images) [31], and PASCAL Stickmen V1.11 [4]. The Buffy Stickmen and FLIC

datasets were split into train and test sets as specified by the datasets and the PASCAL Stickmen was used in its entirety as a test set. For the PASCAL dataset we trained on the FLIC dataset. For all datasets, we evaluated our method on cropped images that contained a person.

Image Features As image features we extract globalPb features [32] from the cropped images. We resized images uniformly and extracted the 8 per-pixel gPb features. We used all features when running our regressions. These features capture the overall contour of the individual in the image. This provides the global structure of a person's layout.

Joints Predicted We built a separate forest for each of the four joints (right/left elbow and wrist). To grow these trees we regressed image features against joint positions that were projected into body-space. For evaluation, we project our predictions back into pixels for fair comparison. In order to do this we rotate, translate, and scale the body-space torso onto a predicted torso in pixels. For the FLIC dataset we use the torsos predicted by [31] for BUFFY and PASCAL we used the torsos provided with the datasets that were provided by [33].

4.1 Evaluation Metrics

We report our results in two formats. The first is the PCP evaluation metric used to evaluate the Buffy and Pascal Stickmen datasets. In this metric a prediction is correct if the average error (between ground truth and prediction) of the two endpoints of a body part is below a threshold. The second metric we use is a radial precision curve (Figure 3) as used in [31]. The radial precision curve is calculated based upon a predicted point's euclidean distance from the ground truth point. The prediction is normalized by the distance between the left hip and the right shoulder so that that distance is 100 pixels.

$$acc_i(r) = \frac{100}{n} \sum_{i=1}^{n} 1 \left(\frac{||\mathbf{y}_i^* - \mathbf{y}_i||_2}{(\text{torsoheight}_i)/100} \geq r \right)$$

n is the number of examples, $\mathbf{y}_i^*$ is the prediction for the joint of example i, and $\mathbf{y}_i$ is the ground truth location of the joint.

Evaluation is done after images have been projected into the pixel-coordinate system. We take our body-space predictions and project the torso into the image by using the shoulders and hips of a predicted pixel torso as a target.

4.2 Regression Forests: Average Prediction

For each training dataset we built four forests of 100 trees that were constrained to stop growing once a leaf had a minimum of 10 elements. Since a regression tree can have output of any dimension, we experimented with regressing to predict joints separately and jointly. In Figure 2 we compare performance on the FLIC dataset's elbows and wrist when we predict the four joints separately or

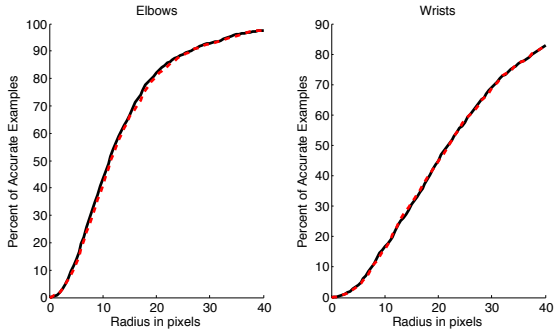

Fig. 2. The output of regression trees can be multidimensional. In this experiment we tested the performance of the regression forests when predicting each of the four joints (right/left elbows and wrists) separately (black curve) and when predicting the joints of each arm (wrist and elbow) jointly (red curve). Performance is not hurt by considering the full arm jointly, the advantage to predicting full arms is that you are insuring a real arm pose. The results shown here are the average prediction of the FLIC regression forest on the FLIC test dataset.

the arms (one forest for the right elbow and wrist, one forest for the left elbow and wrist) as a whole.

We evaluate our method on three standard datasets. We present results for each dataset in the metrics used by other authors for those datasets. We present three different types of results for the datasets. First, we show the performance of the dataset when the test set is evaluated by forests grown by the same dataset, see Figure 3 and Table 4. Second, we show the performance of the dataset when the test set is evaluated by forests grown by another dataset, see Figure 3. Third, we show the results of regressing the images against 3-dimensional skeletons to predict a 3-dimensional results, see Figure 6. Since the PASCAL dataset does not have a training set it was evaluated using the regression forests learned from the FLIC dataset.

In order to regress images against 3-dimensional skeletons we trained the forests on the H3D dataset. We projected the 3-dimensional annotations of the H3D dataset into body-space by rotation, translation and scale. We then built regression forests in the same way as before but with each leaf output being a 3-dimensional joint location prediction. Examples of these predictions for the FLIC dataset can be seen in Figure 6.

The simple average of regression tree outputs produces results that are competitive with state-of-the-art results on elbows and comparable to state-of-the-art results on wrists. This is encouraging because state-of-the-art methods require complex inference or search. Our method provides the added benefit of being able to produce multiple predictions for the same data allowing us to recombine predictions in useful ways.

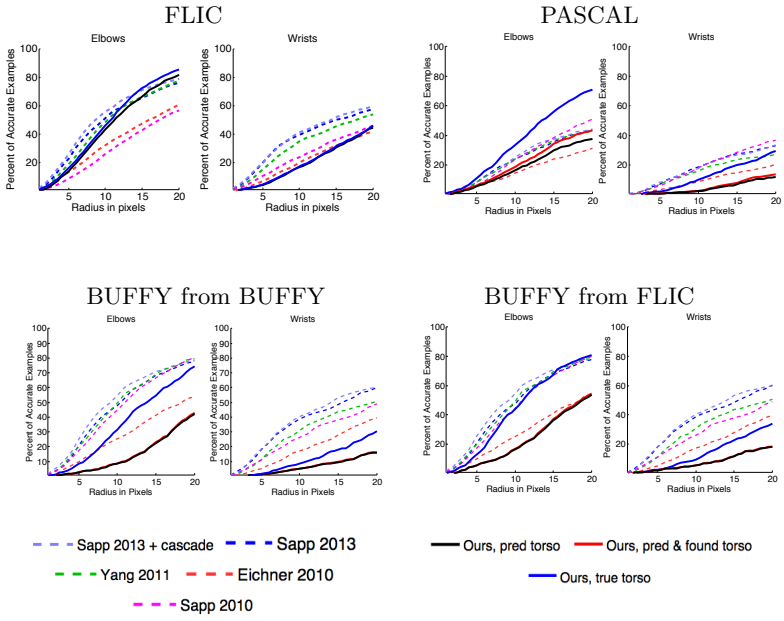

Table 3. Radial precision plots of elbow and wrist joint predictions. Radius (r) is reported for a normalized torso diagonal of 100 pixels. For all datasets we report our regression's prediction in black when projected onto a predicted torso. In 70 PASCAL images and 6 BUFFY images no torso was predicted, the result of the method when those rejected images are excluded from the test set are in red. In order to see how the method performs when the torso is known, we have projected our results onto the ground truth torso and reported results in blue. This image is best viewed in color. All test sets were evaluated on trees trained on the FLIC dataset, BUFFY was also evaluated on trees trained on the BUFFY dataset.

4.3 Regression Forests: Minimax Optimization Predictions

Training: We trained our minimax optimization method on the FLIC dataset. We used a 100-tree forest for each wrist and applied the mimimax optimization to each wrist separately. We constrained the weighting of the training examples by $k = 10$ and number of maximum iterations of the optimization to 100. We then applied the learned weighting to the predictions made by the 100 trees (learned on FLIC) on the FLIC dataset and on the BUFFY and PASCAL datasets.

Figure 7 shows examples of the output of both the uniformly-weighted regression forests and the minimax-weighted regression forests. The minimax-weighting pulls original predictions to be longer or to change the direction of the limb.

Performance on same dataset: For this experiment we define two types of pose, common and rare. In order to define these we gridded the space around the

Method	Torso	U. Arm	L. Arm
Andr. [6]	90.7	79.3	41.2
Eich. [4]	98.7	82.8	59.8
Sapp [34]	100	91.1	65.7
Sapp [5]	100	95.3	63.0
Yang [35]	98.8	97.8	68.6
Ours from BUFFY, predicted torso		80.6	18.5
Ours, found predicted torso		82.4	18.9
Ours, ground truth torso		96.1	45.3

Table 4. PCP is the common evaluation metric for the Buffy Stickmen dataset, we report results with the standard threshold of 0.5. We report the performance of the BUFFY test set when evaluated on forests built using BUFFY data. These results were all projected onto predicted torsos in pixels for evaluation. In 6 BUFFY images no torso was predicted, the result of the method when those rejected images are excluded are labeled 'found predicted torso'. In order to see how the method performs when the torso is known, we have projected our results onto the ground truth torso and reported results as well.

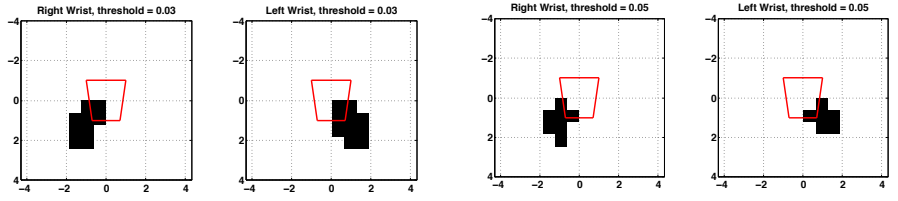

Fig. 3. This is a visualization of 'rare' and 'common' positions for wrists in the FLIC dataset relative to a torso in red. In black are marked locations that have more than 0.03 (left) and 0.05 (right) of the total data in them, these are considered common. For the right wrist, 65% of the data is common by a 0.05 threshold. For the left wrist, 62% of the data is common by a 0.05 threshold.

torso into bins as shown in Figure 3. We then classified a bin as common if more than a certain percentage of the data fell into this bin. As can be seen in Figure 4 the minimax method improves performance on rare (and often more difficult for this method) poses and loses accuracy on the common examples. The trade-off between the accuracies is made clear in the rare and common curves.

Performance on different datasets: In this experiment we test how the mimimax optimization can improve performance when training on one dataset and testing on another. We trained two regression forests on the FLIC dataset, one for each wrist. We then evaluated the BUFFY and FLIC datasets using these forests and plotted their accuracies in Figure 5 when projected onto predicted torsos. When we used the weights calculated by the minimax optimization the accuracies improved on both wrists and on both datasets. The improvement can

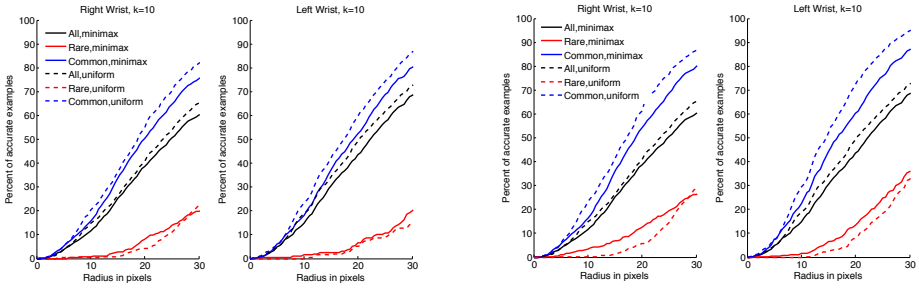

Fig. 4. For this experiment we separated the FLIC dataset into common and rare body-space locations (see Figure 3). Locations that were common by a threshold of 0.03 are evaluated on the left and results that were common by a threshold of 0.05 were evaluated on the right. The black curve represents the full FLIC test-set, the red curve represents rare pose locations, and the blue curve represents common pose locations. The dotted line is the result of uniformly weighting the tree outputs and the solid line is the result of using the minimax weights to predict a final output. Results were then projected onto predicted torsos for comparison. As expected, the minimax method improves performance on rare (and therefore often more difficult for this method) poses and loses accuracy on the common examples. The trade-off between the accuracies is made clear in the rare and common curves.

also be seen in Table 5. Using the minimax method has made our method more generalizable when evaluating on a different dataset than that which we trained on.

5 Conclusions

We present a radically different approach to the problem of human pose estimation from a single image. We have achieved accuracies comparable to state-of-the-art methods with a system that uses no inference or search techniques. Unlike current methods we use regression forests that provide multiple predictions for every example image. We have taken advantage of these many proposals by using

Method	U. Arm	L. Arm
Trained on BUFFY, found predicted torso	82.4	18.9
Trained on FLIC, found predicted torso	79.7	26.0
Trained on FLIC with minimax, found predicted torso	79.7	34.1

Table 5. We report the performance of the BUFFY test set when evaluated on forests built using BUFFY or FLIC data. To report the performance of the test data when evaluated on FLIC forests with and without minimax optimization. We believe that the method performed better when trained on FLIC than when trained on BUFFY because of the size of the datasets. It is clear that applying the minimax optimization helps the performance of the wrists significantly.

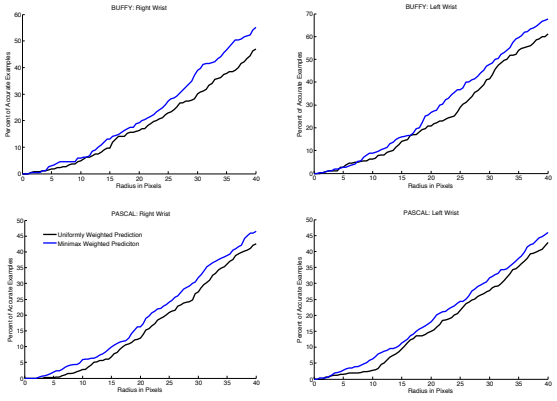

Fig. 5. The results of evaluating the BUFFY (top row) and PASCAL (bottom row) on the forests trained on FLIC. The black line is the average over the trees for the right and left wrists. The blue line is the result of the minimax re-weighting. The results were projected onto predicted torsos in pixel-space for evaluation (images in which no torso was found have been excluded from this evaluation). The minimax re-weighting has made our process more robust on datasets not trained on.

the minimax optimization that quickly and easily improves the generalization of our method. We further utilize the regression forests to make predictions in 3-dimensions with no additional efforts.

We have presented the parsing problem in a new way. By producing multiple outputs we can now apply re-weighting techniques (such as minimax) or consider the problem of selecting the best of all proposals.

The method can be easily extended to lower body predictions. Because we do not search for localized parts we expect our method could be robust to occlusions, baggy clothing and motion blur.

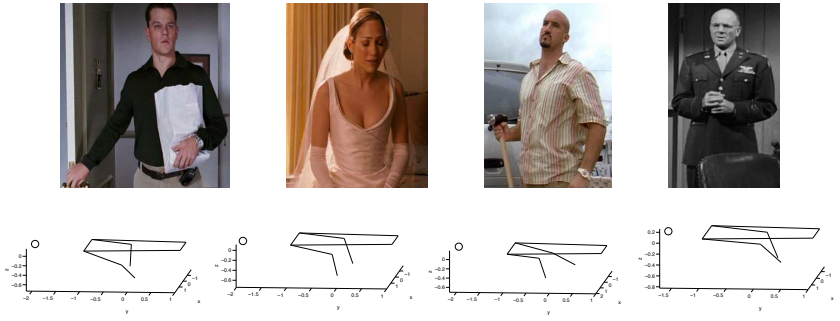

Fig. 6. Regression forests can predict multi-dimensional output. For this reason, when trained on the H3D dataset (that has 3-dimensional annotations) producing 3-dimensional output is easy. These are examples of the results obtained for the FLIC dataset when evaluated on the regression forests built by the H3D dataset. Skeletons are shown facing down and the x (across body), y (down body), and z (in-front of body) axes are marked.

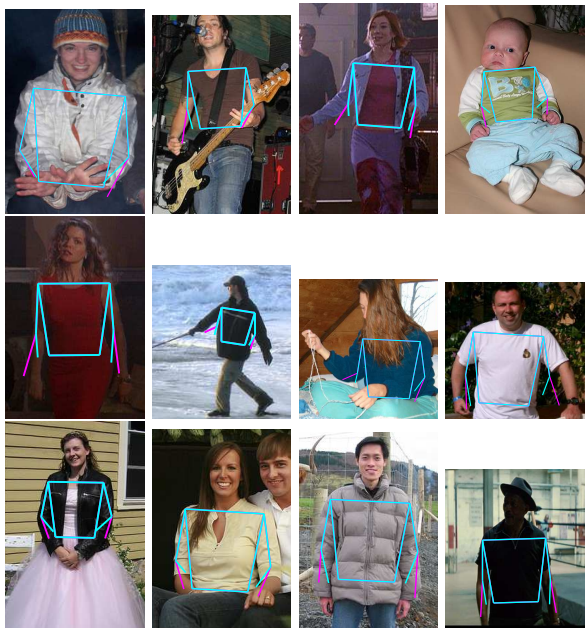

Fig. 7. Examples of original uniformly-weighted predictions in cyan and the minimax-reweighted predictions in magenta. The minimax often produces results that are similar to the original prediction but that shift it in length or direction in attempt to correct the original prediction.

References

1. Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: CVPR. (2011)
2. Wang, Y., Tran, D., Liao, Z.: Learning hierarchical poselets for human parsing. In: CVPR. (2011)
3. Tran, D., Forsyth, D.: Improved human parsing with a full relational model. In: ECCV. (2010)
4. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: IICCV. (2009)
5. Sapp, B., Toshev, A., Taskar, B.: Cascaded models for articulated pose estimation. In: ECCV. (2010)
6. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR. (2009)
7. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV **61**(1) (January 2005) 55–79
8. Ioffe, S., Forsyth, D.: Human tracking with mixtures of trees. In: ICCV. (2001) 690–695
9. Ramanan, D.: Learning to parse images of articulated bodies. In: ANIPS. Volume 19. (2006) 1129–1136
10. Jiang, H., Martin, D.R.: Globel pose estimation using non-tree models. In: CVPR. (2008)
11. Ren, X., Berg, A., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: IICCV. Volume 1. (2005) 824–831
12. Tian, T.P., Sclaroff, S.: Fast globally optimal 2d human detection with loopy graph models. In: CVPR. (2010)
13. Wang, Y., Mori, G.: Multiple tree models for occlusion and spatial constraints in human pose estimation. In: ECCV. (2008)
14. Mori, G., Ren, X., Efros, A., Malik, J.: Recovering human body configuration: Combining segmentation and recognition. In: CVPR. Volume 2. (2004) 326–333
15. Ferrari, V., Marín-Jiménez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR. (2008)
16. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: CVPR. (2010)
17. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: BMVC. (2009)
18. Bourdev, L., Malik, J.: Poselets: Body part detectors training using 3d human pose annotations. In: IICCV. (2009)
19. Brox, T., Bourdev, L., Maji, S., Malik, J.: Object segmentation by alignment of poselet activations to image contours. In: CVPR (CVPR). (2011)
20. Bourdev, L., Maji, S., Malik, J.: Describing people: Poselet-based attribute classification. In: ICCV (ICCV). (2011)
21. Gkioxari, G., Arbelaez, P., Bourdev, L.D., Malik, J.: Articulated pose estimation using discriminative armlet classifiers. In: CVPR. (2013) 3342–3349
22. Jammalamadaka, N., Zisserman, A., Eichner, M., Ferrari, V., Jawahar, C.V.: Video retrieval by mimicking poses. In: ACM ICMR. (2012)
23. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR. (2011)

24. Dantone, M., Gall, J., Leistner, C., van Gool, L.: Human pose estimation from still images using body parts dependent joint regressors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2013) to appear.
25. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR'11. (June 2011)
26. Ramanan, D., Forsyth, D.: Finding and tracking people from the bottom up. In: Proc CVPR. (2003)
27. Ferrari, V., Marin, M., Zisserman, A.: Pose search: retrieving people using their pose. In: CVPR. (2009)
28. Taylor, C.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In: CVPR. (2000) 677–84
29. Kakadiaris, I., Metaxas, D.: Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In: CVPR. (1996) 81–87
30. Ikizler, N., Forsyth, D.: Searching video for complex activities with finite state models. In: CVPR. (2007)
31. Sapp, B., Taskar, B.: Modec: Multimodal decomposable models for human pose estimation. In: In Proc. CVPR. (2013)
32. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5) (May 2011) 898–916
33. Eichner, M., Marin-Jimenez, M., Zisserman, A., Ferrari, V.: 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. In: ETH Zurich, D-ITET, BIWI, Technical Report No.272. (2010)
34. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose priors for pictorial structures. In: CVPR. (2010)
35. Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: IEEE PAMI. (To appear)