UFo - Coupling Uncertain Active Constellation Models with Cascaded Forest Predictors for Sematic Segmentation

Anonymous CVPR submission

Paper ID ****

Abstract

We consider the task of model-based semantic segmentation. The model is described by a constellation model of parts that are represented by active shape- and appearance models. We term this an active constellation model. As a running example we utilize a 21-part, chain-based spine model of zebra-fish observed in microscopic images. The prevailing approach to solve such a task is to first generate pixel-independent features for each part, e.g. via a cascaded decision forest predictor, which are then fed into an MRF-based model-fitting objective to infer the optimal MAP solution of the constellation model. Our key contribution is to abandon this static, two-stage approach and mix feature generation and model-based inference in a new, more flexible, way. In particular we interleave the cascaded forest predictors with inference steps for the model-fitting. A key finding is that uncertain model-outputs at intermediate stages of the cascade, in the form of part-based marginals, are essential for best performance. This is because, as opposed to MAP inference, the soft marginals do not commit to a certain – potentially wrong – solution "at first sight". If unsure at first sight, soft marginals allow for "narrowing down" on the correct solution in later stages of the cascade. We validate our findings with an in-depth study of alternative inference steps, including popular geodesic smoothing as well as MAP inference. We believe that our findings are not only relevant for other types of constellation models, but, more generally, for the recent trend of combing deep learning models with physically-motivated structured models.

1. Introduction

Many tasks in computer vision have as input an image and as output a dense labeling, where each pixel is assigned one out of many pre-defined classes. An example is a sematic segmentation of a person in an image, where each pixel is assigned a label such as background, left leg, or head. So-called *structured models*, such as a Conditional Random Field (CRFs), are often used for semantic segmentation. Depending on the prior knowledge about the task at hand, the underlying graph may be a (super-)pixel grid, or a graphical constellation model that captures relative locations of multiple parts of an object.

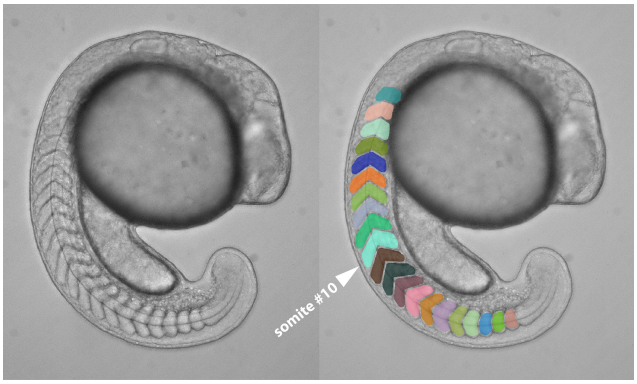

Figure 1. Exemplary application of semantic segmentation: Zebrafish larvae, where 21 parts of the spine, called *somites*, have to be segmented in microscopic images. Left: Exemplary image. Right: Ground truth segmentation of 21 somites overlaid onto image.

Such structured models commonly capture the task of semantic segmentation via an objective function that is composed of a data term and a prior, where the data term is derived from the image at hand, yielding pixel-wise distributions over class labels, and the prior is enforced subsequently. Current state-of-the-art approaches typically employ pixel-wise forest predictors combined with MAP inference on a (super-)pixel graph for semantic segmentation [22, 20] (see also e.g. [5]), or on a graphical constellation model for the localization of object parts (e.g. [6, 7, 4, 21]). A recent trend in computer vision replaces single-level forest predictors by deep, *cascaded* models for feature generation, such as CNNs [14] (see also e.g. [5]) and Auto-Context Models [23] (see also e.g. [18]). These models play the role of learning a complex non-linear mapping from images to features which are relevant for the task at

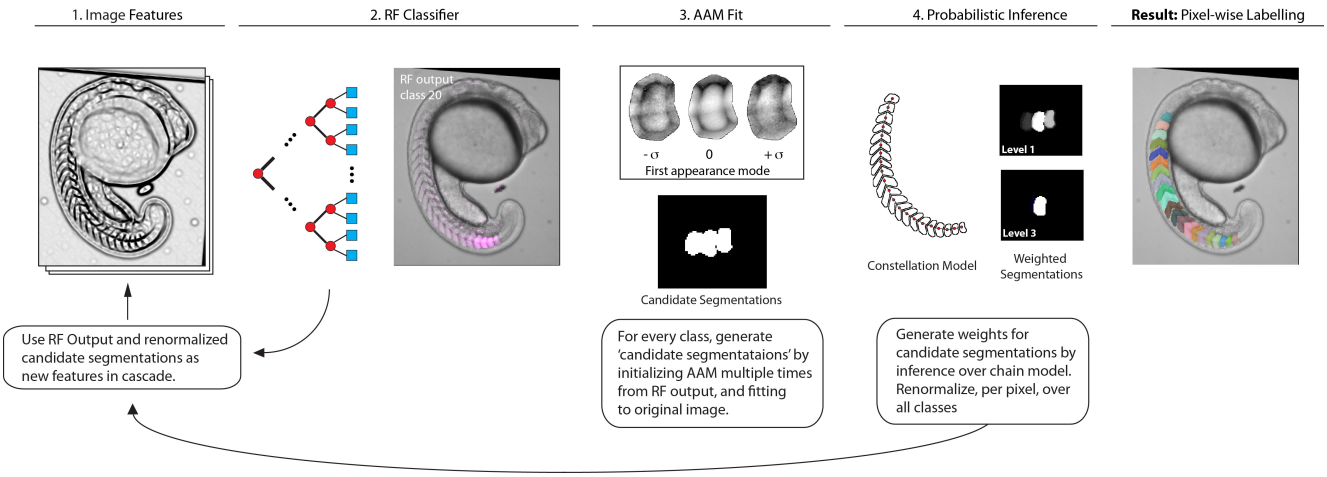

Figure 2. Our proposed pipeline for multi-class, semantic segmentation. First, a stack of feature images is created by a standard filter bank. These features are used to train a random forest classifier. The random forest output, i.e. a probability map for each class, is then used in combination with the original image to generate candidate segmentations for each class, with the help of active appearance models (AAM). These candidate segmentations are weighted by means of probabilistic inference in a constellation model that captures relative locations of classes. The weighted and fused candidate segmentations are then fed back as additional features into a next random forest classifier. This approach is iterated, forming a cascade: Random forests are interleaved with "smoothing" of the forest output by means of probabilistic inference in a constellation model. As opposed to the common MAP inference, probabilistic inference keeps up the uncertainty about intermediate solutions – this is why we term our approach *Uncertain Forests* (UFo).

hand.

This modelling framework is however very static, as it separates feature generation and inference (i.e. "'model fitting"'). It has been shown that better features can be generated by *interleaving* feature generation with MAP inference in pixel-grid structured models [17, 10, 19] or model agnostic smoothing smoothing [13]. Note that this is conceptually different from the classical "hierarchical" approach that, purely for the sake of pruning the search space to reduce run-time, performs feature generation and inference/model fitting multiple times on different scales.

In this work we take the idea of interleaving feature generation and inference a step further: Instead of interleaving feature generation with a pixel-level structured model or model-agnostic smoothing, we interleave with a global, generative *active constellation model*. By this term we refer to a graphical constellation model, where the individual object parts are captured by *active appearance models* [3]. We suggest a cascaded pipeline, as illustrated in Figure 2. The most important aspect of this cascade is the question of what to infer from the constellation model at intermediate stages of the cascade. Options are marginal distributions or the MAP solution. A model-agnostic, yet "image-aware" alternative to model-based inference is geodesic smoothing [13]. One of the main aspects of this work is to study the trade-offs that come with these options.

We show that marginal distributions are a clear winner for an exemplary application of semantic segmentation of many self-similar structures, namely vertebrae in spines of zebra-fish embryos. Figure 1 shows an exemplary image, and a ground truth segmentation. The reason that *uncertainty is beneficial* here is that individual somites are highly ambiguous with respect to shape and appearance, hence only the relative spatial arrangement can disambiguate this situation.

Related work has tackled semantic segmentation of vertebrae (in CT scans of humans) via the classical approach of feature generation followed by MAP inference in a constellation model [7], without cascading. We show that employing marginals instead of MAP in a cascaded feature generation pipeline helps to avoid committing to a wrong solution in the early stages of a cascade, and lead to a major increase in resulting segmentation accuracy.

Closely related to the presented work are (1) Auto Context [23], but they do not perform any smoothing in-between levels of the cascade. (2) Geodesic Forests [13], but they do not use a structured model for smoothing. (3) Cascaded classifiers interleaved with MAP inference [17, 10, 19], but they do not use a (global generative) constellation model and do not explore marginals for inference. (4) Constellation models for vertebrae and other self-similar object segmentation (like e.g. [6, 7, 12, 4, 11]) but none of these run a cascade, and do not exploit marginal distributions.

To summarize, we claim the following three **contributions**:

- In the field of semantic segmentation with constellation models, we are the first to interleave feature gen-

eration and model-based inference. We show that this boosts performance considerably, compared to not cascading.

- We show, for the first time, that probabilistic inference gives a major (6%) boost in performance in cascaded MRF-Forest-based models. This is compared to standard MAP inference (as e.g. in [7, 21, 4]) and model-agnostic geodesic smoothing [13, 16].

- We are the first to tackle spine detection in zebra fish, where we achieve an overall average Dice score of 82%.

2. Background

Random Forests and Cascading. We employ Random Forests (RF) [1] for feature generation in a cascaded fashion. We assume that the reader is familiar with the general concept of Random Forests. In the related Auto Context approach [23], the probability maps yielded by a random forest are fed as features into subsequent random forests, yielding a cascade of forests. Interleaved inference/smoothing [17, 10, 19, 13] operates on these probability maps, and feeds "smoothed" versions of them into the next forest.

Active Appearance Models. We employ active appearance models [3] for generating binary *candidate segmentations* for each class in the semantic segmentation task (cf. Sec. 3.1).

AAMs are linear, generative, parametric models of shape and appearance that are learned from training data, and are widely used e.g., for face landmark localization and medical image segmentation [9]. The shape model is defined as follows:

$$s = s_0 + \sum_{i=1}^{n} p_i s_i$$

where s is a vector of coordinates of the landmarks that define the shape. Principal component analysis (PCA) yields is the mean shape, s_0, and eigenvectors s_i, sorted by their respective eigenvalues. The scalars p_i are the *shape parameters* of the model.

The Appearance Model is defined on the base-mesh x, which is defined by the mean shape s_0, as follows:

$$A(x) = A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x)$$

The average appearance, i.e. an image A_0, as well as eigenvectors A_i are computed by PCA on a set of *shape normalized* training images, i.e. images which have been warped onto the base-mesh. The scalars λ_i are the *appearance parameters* of the model.

An even more compact representation can be realized by a subsequent step of PCA on the combined shape and appearance parameters, leading to a combined AAM; however, this limits the choice of efficient solvers, such as the Inverse Compositional Algorithm [15]. For the rest of this paper, we will restrict ourselves to independent shape and appearance models.

Fitting an AAM is a non-linear optimization problem that consists of finding the model instance that minimizes the error to the input image. Optimization is commonly done either by learning a linear mapping from the error image to parameter updates [3], or by iteratively computing incremental gradient descent updates to model parameters [15]. Since AAMs are generative models, they can be used to generate images which can be directly compared to the input image. Thus, fitting an AAM consists of finding the model parameters that minimize the sum-of-squared-distances between the image and the corresponding model instance, evaluated on the base mesh:

$$CostAAM = \sum_{x \epsilon s_0} [A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(N(W(x;p);q))]^2 \quad (1)$$

To create a model instance, first render an image $A(x)$ on the base-mesh. $A(x)$ is defined by appearance parameters λ. Then warp this image to a shape instance s, defined by shape parameters p_i. Warping can be done using e.g., a thin-plate spline, parametrized by the set of landmarks, s_0 and s. This defines the unique warp parametrized by p, called W(x;p). Finally, the model instance is transformed into the image by the global shape normalizing transform N(x;q), in our case a similarity transform, parameterized by q.

A central challenge of AAMs is their sensitivity during the fitting process. However, AAMs can be easily extended to include priors on the model parameters (see also [15]), with minimal additional cost to the fitting algorithm, and can be interpreted from a Bayesian viewpoint as Gaussian Regularization.

3. Method

Given an image as input, we seek a pixel-wise multi-class labeling as output, i.e. a *semantic segmentation*. We assume that a model of the spatial relation of classes can be learned, i.e. a *constellation model*. This is the case for many applications, as e.g. body part segmentation in natural [18] or medical images [21], vertebra segmentation [6, 7], to name just a few.

We propose the following pipeline for *model-based semantic segmentation*, as layed out in Figure 2: First, we

generate probability maps for each class with a random forest classifier. Second, we generate many *candidate segmentations* for each class, with the help of Active Appearance models (cf. Sec. 3.1). Each candidate segmentation is a binary segmentation of the respective class. It serves as a "segmentation hypothesis". Third, we perform probabilistic inference in a constellation model to weigh candidate segmentations (cf. Sec. 3.2), and effectively "smooth" the probability maps generated by the RF classifier. Fourth, we feed the resulting "smoothed" probability maps into a next RF classifier, together with the original probability maps as well as all image features used as input to the previous RF.

To generate a resulting labeling per pixel from the last RF output in the cascade, one can take the class with maximum probability according to either the RF probability maps, or the respective "smoothed" versions.

3.1. Generating Candidate Segmentations

Given an RF-generated probability map of a class, we first compute its centroid via the mean shift algorithm. Second, we fit an average constellation model (i.e. a static constellation of landmarks) to these centroids to yield an optimal global similarity transform w.r.t. the sum of squared landmark distances. In our application, this is sufficient to define an approximate orientation of the respective object part. Third, we sample a number of candidate locations around the centroids of the RF-generated probability maps to get sets of location initializations for the respective classes. Fourth, we fit a class specific active appearance model (AAM) to the image, multiple times, starting at the initial locations computed in the previous step. Each AAM fit results in a binary segmentation, together with a cost for the fit (cf. Eq. (1)). These binary segmentations serve as candidate segmentations for their respective classes.

3.2. Weighting and Fusing Candidate Segmentations

The above method generates a number of candidate segmentations per class. We assign weights to these candidate segmentations by means of a constellation model in the form of a second order CRF. The nodes of the CRF correspond to the classes, $c \in \{1..n_C\} =: C$, and the labels of each node correspond to the respective candidate segmentations, $l \in \{1..n_L\} =: L$. Note that here, for the sake of notation simplicity, we assume we have the same number of candidate segmentations for each class.

Unary factors, $\phi_c(l)$, reflect the cost of the respective AAM fit, $A_c(l)$. Furthermore they reflect the RF probability map $P_c : \Omega \rightarrow [0, 1]$, accumulated over the foreground of the respective binary segmentation, $H_{c,l} : \Omega \rightarrow \{0, 1\}$:

$$\phi_c(l) = \exp\left(-\alpha \cdot CostAAM_c(l)\right) \cdot \frac{\sum_{x,y} H_{c,l}(x,y) \cdot P_c(x,y)}{\sum_{x,y} H_{c,l}(x,y)} \quad (2)$$

A parameter α weights the relative influence of the two terms.

The pairwise factors, $\psi_{c,b}(l, k)$, reflect the probability of relative locations of neighboring proposals. To this end, we learn the average offset between part centroids, as well as respective covariances, and assume an acccording gaussian distribution.

We compute weights for each proposal and each class by means of probabilistic inference in this CRF. In our application, the respective graphical model is a chain, and hence probabilistic inference can be performed optimally and efficiently by means of dynamic programming. Given the resulting marginals $p_c(l)$, we compute a weighted average of candidate segmentations:

$$S_c(x,y) = \frac{1}{Z(x,y)} \cdot \sum_{l \in L} p_c(l) \cdot H_{c,l}(x,y) \quad (3)$$

Here, $Z(x, y)$ serves for pixel-wise re-normalization; I.e., $Z(x,y) = \sum_{c \in C} \sum_{l \in L} p_c(l) \cdot H_{c,l}(x,y)$. We call S_c a *smoothed probability map* for class c.

4. Results and Discussion

We applied our semantic segmentation approach to a data set of 32 images of developing zebrafish, where the goal is semantic segmentation of the 21 segments of the spine, called *somites*. In a pre-processing step, all images were aligned to a reference image by rigid registration. Experts in biology manually created ground truth segmentations of these images. A segmentation exhibits 22 classes, corresponding to 21 segments and the background.

This data set poses multiple challenges for automated segmentation, (1) due to the similar appearance of neighboring segments, and (2) due to the the small amount of training data.

We approach this problem by interleaving a cascaded random forest classifier with model fitting via probabilistic inference to generate new, smoothed, features for the next level of the cascade, as described in Section 3. We provide comparisons of our approach with a range of other cascaded random forest classifiers, including Auto-context [23], GeoF [13], and MAP instead of probabilistic inference, as well as with a state-of-the-art approach for spine segmentation [7]. See Figure 3 for an overview of the different types for inference/smoothing that we evaluate. We evaluate all algorithms in terms of the Dice score averaged over all 21 foreground classes of the 32 images. We employ two-fold cross-validation to obtain scores for all 32 images.

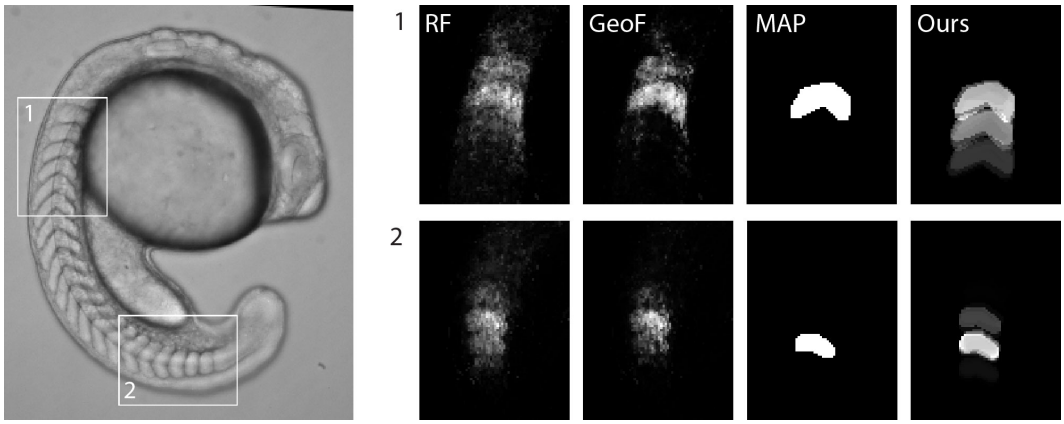

Figure 3. In our evaluation we interleave different types of inference/smoothing into our cascaded random forest pipeline for semantic segmentation of somites (i.e. vertebrae) in zebra-fish embryos. The Figure shows examples of the types of inference/smoothing we evaluate. Left: Exemplary zebra-fish embryo. Boxes (1,2): Exemplary somites. Right: Close-ups on probability maps of the respective class labels. Note that close-ups on (2) are rotated by 90 degrees. Four versions of smoothing/inference: (RF) random forest probability map; (GeoF) smoothed by geodesic smoothing [13]; (MAP) "smoothed" by MAP inference in our constellation model, yielding a binary "probability map"; (Ours) "smoothed" by our proposed approach, i.e. probabilistic inference in our constellation model.

Our Approach. We begin by training a random forest of 16 trees, each with maximum depth 12, on features stemming from a standard filter bank [8]. Additionally, we employ the use of contextual features. Contextual features are generated by evaluating the filter bank response at a random offset to the current pixel, and either thresholding on this value directly, or taking the difference between this value at the current pixel.

During training, we use different subsets of the training data at each level of the cascade, to avoid over-fitting the small data set. As additional "synthetic" training data we generate two random rotations (+/- 10 degrees) for each training image. The remaining training data at each level is then used to build a constellation model for the backbone, and an active appearance model for every individual segment. To initialize the AAMs for fitting, we find the mode of the RF output for each class using mean shift, and then fit the rigid constellation model to this. We additionally re-center each AAM on the mode of RF output for its corresponding class. From these two sets of initializations, we generate many more by sampling locally along the axis of the constellation model, and then run gradient descent to fit each AAM instance to the input image. Fitting is done over 6 parameters, one shape parameter, one appearance parameter, and an additional four parameters for the similarity transformation.

Each AAM fit yields a hypothetical segmentation of the corresponding somite; however, there is not yet any global consistency imposed. To introduce this feature, we do probabilistic inference over the chain using Belief Propagation. The resulting marginals are then used to scale the mask of the corresponding fit, and finally a posterior probability distribution is calculated for every pixel by re-normalizing the weighted fit masks over all classes (cf. Equation (3)). See Figure 3 (Ours) for examples of probability maps induced by the marginals of the constellation model.

Auto Context. The canonical example of stacked classifiers is Auto-context [23], where the RF output is sampled over a regular grid of offsets, and these features are used in the next level of a cascade. Since our random forests already sample the local features, we evaluate the performace of Auto-context by simply concatenating the RF output as additional features in the next layer. See Figure 3 (RF) for examples of how the RF output looks.

GeoF. An "image aware" smoothed RF Output can be generated in a model-agnostic fashion, by geodesic smoothing [13]. Given an initial probability map, the rationale behind geodesic smoothing is that pixels with a small geodesic distance to a pixel with high probability should be up-weighted, and otherwise down-weighted. Smoothing is accomplished indirectly, as a competition between different possible class labels for a given pixel, mediated by pixel-wise normalization. See [13] for details, and Figure 3 (GeoF) for examples.

MAP Inference. Finally, we consider another model-based method of smoothing: Instead of marginal distributions as new features, we use the MAP solution as features for the next layer. The map solution is represented as a stack of 21 binary, "hard" segmentations, one for each class. See Figure 3 (MAP) for examples.

	RF Output	Smoothed RF Output
Auto-context	0.60 (0.20)	-
Geodesic	0.63 (0.21)	0.66 (0.22)
MAP	0.71 (0.27)	0.76 (0.27)
Model Marginals	**0.82** (**0.16**)	**0.82** (0.18)

Figure 4. Evaluation on 32 datasets. Dice Scores on all 21 Somites: Average (over 32x21 values), and standard deviation (in brackets). Left column: Segmentations obtained from RF-generated probability maps, by assigning to each pixel the class with the highest probability. Right column: Segmentations obtained from repective "smoothed" RF probability maps.

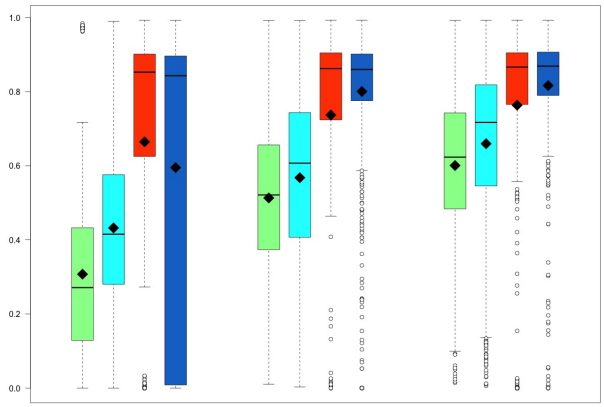

Figure 5. Three-level cascade. Segmentation accuracy of 4 methods after each level: RF output (green), GeoF (cyan), MAP (red), Ours (blue). For every method at every level, Dice scores of 21 somites in 32 images, i.e. 672 scores, are visualized as a *box plot* [2]. A colored box spans from lower to upper quartile, i.e. the inter-quartile range. I.e. 50% of the data points lie within the box. The horizontal bar within the box depicts the median. The black diamond depicts the mean. Whiskers depict the outlier-free data range. Circles depict outliers. Outliers are defined as data points beyond median $\pm$ 2 inter-quartile ranges.

4.1. Quantitative Results

Figure 4.1 lists Dice scores for all four methods described above, after three levels of the cascade. The first column states Dice scores obtained for segmentations generated by assigning to each pixel the class with highest probability in the RF output. The second column states Dice scores obtained analogously from the respective smoothed RF output. Figure 4.1 shows box plots of the Dice scores obtained from the smoothed RF output (cf. second column of Figure 4.1, if present; else first column), at all three levels of the cascade.

Auto-context returns a final average Dice score of 0.60 (sd=0.20) after three levels (cf. Figure 4.1, 1st row). Compared to Auto-context, GeoF generates considerably smoother posteriors, and performs better at every level of the cascade (see green vs. cyan box plots in Figure 4.1). The best average score obtained by GeoF was 0.66 (sd=0.22) after three levels (cf. Figure 4.1, 2nd row), when evaluated on its smoothed output. This increase of 6% w.r.t. Auto-context is comparable to the gains reported in [13] when applying geodesic smoothing without changing the training objective.

Solely after the first level of the cascade does MAP Inference perform best among all approaches (red box plots in Figure 4.1), with a mean Dice Score of 0.66 (sd=0.36). This approach also improves over the cascade, reaching a final Dice Score of 0.76 (sd=0.27) after three levels, when evaluated on the MAP output (cf. Figure 4.1, 3rd row). However, our proposed approach of replacing MAP by marginals yields the highest overall average Dice Score of 0.82 (sd=0.18) (cf. Figure 4.1, 4th row), outperforming MAP by 6%. With our approach, the accuracy increases considerably from level to level (blue box plots in Figure 4.1).

4.2. Discussion

Cascading Helps! Observe in Figure 4.1 that the accuracy of either approach increases over the levels of the cascade. This confirms the power of cascading. Note, however, that interestingly, all of the methods that we evaluate stopped improving accuracy after three levels of cascading, presumably due to the small size of our training data set.

Smoothing Helps! Approaches that employ any kind of smoothing between levels perform better than auto-context. This confirms the power of interleaved smoothing. Model-based smoothing performs considerably better than model-agnostic geodesic smoothing. We argue that this is due to the more specific prior knowledge induced by the model.

Uncertainty To the Rescue! A first observation makes a point for the conventional approach of feature generation and subsequent MAP inference without cascading: MAP inference does yield the best results after the first level of our cascade. However, the power of our approach (probabilistic inference) is revealed over the levels of the cascade: As opposed to MAP, our approach undergoes a dramatic increase in the mean Dice Score and concurrent reduction in the standard deviation over the 3 levels of the cascade. We observe that this is due to failure cases that are "rescued" by our approach, but not by MAP, as shown in Figure 6.

5. Conclusion

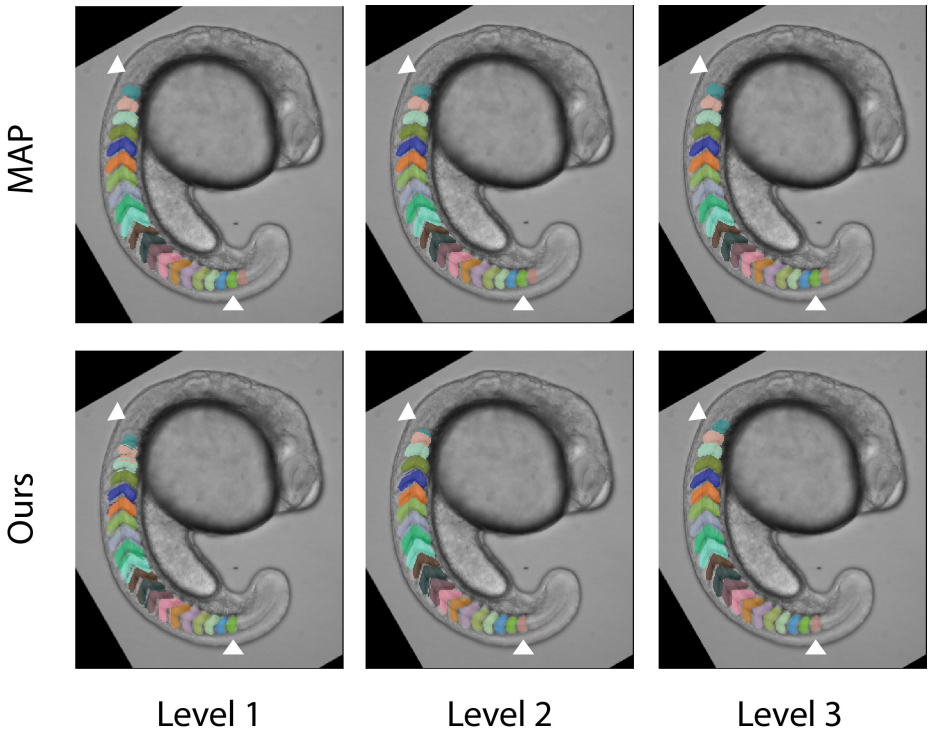

Figure 6. Exemplary failure case that is "rescued" by our approach (probabilistic inference) over the levels of the cascade, but not by MAP inference. Arrows point to ground truth start end end of spine. Columns: Three levels of the cascade: Top row: MAP inference. Bottom row: Ours (probabilistic inference). After the first level, segmentations are off by one somite in both approaches. This stays constant for MAP. Our approach, however, gradually rescues this case and obtains a correct segmentation after level three of the cascade.

References

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 3

[2] J. Chambers. *Graphical Methods for Data Analysis*. Chapman & Hall statistics series. Wadsworth International Group, 1983. 6

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 2, 3

[4] N. Duy, H. Lamecker, D. Kainmueller, and S. Zachow. Automatic detection and classification of teeth in ct data. In N. Ayache, H. Delingette, P. Golland, and K. Mori, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2012*, volume 7510 of *Lecture Notes in Computer Science*, pages 609–616. Springer Berlin Heidelberg, 2012. 1, 2, 3

[5] J. Funke, J. N. Martel, S. Gerhard, B. Andres, D. C. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, H. Pfister, A. Cardona, et al. Candidate sampling for neuron reconstruction from anisotropic electron microscopy volumes. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 17–24. Springer, 2014. 1

[6] B. Glocker, J. Feulner, A. Criminisi, D. Haynor, and E. Konukoglu. Automatic localization and identification of vertebrae in arbitrary field-of-view ct scans. In N. Ayache, H. Delingette, P. Golland, and K. Mori, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2012*, volume 7512 of *Lecture Notes in Computer Science*, pages 590–598. Springer Berlin Heidelberg, 2012. 1, 2, 3

[7] B. Glocker, D. Zikic, E. Konukoglu, D. Haynor, and A. Criminisi. Vertebrae localization in pathological spine ct via dense classification from sparse annotations. In K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2013*, volume 8150 of *Lecture Notes in Computer Science*, pages 262–270. Springer Berlin Heidelberg, 2013. 1, 2, 3, 4

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009. 5

[9] T. Heimann and H.-P. Meinzer. Statistical Shape Models for 3D Medical Image Segmentation: A Review. *Medical Image Analysis*, 13(4):543 – 563, 2009. 3

[10] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression tree fields – an efficient, non-parametric approach to image labeling problems. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2376–2383, June 2012. 2, 3

[11] D. Kainmueller, F. Jug, C. Rother, and G. Myers. Active graph matching for automatic joint segmentation and annotation of c. elegans. In P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2014*, volume 8673 of *Lecture Notes in Computer Science*, pages 81–88. Springer International Publishing, 2014. 2

[12] T. Klinder, J. Ostermann, M. Ehm, A. Franz, R. Kneser, and C. Lorenz. Automated model-based vertebra detection, identification, and segmentation in {CT} images. *Medical Image Analysis*, 13(3):471 – 482, 2009. 2

[13] P. Kontschieder, P. Kohli, J. Shotton, and A. Criminisi. Geof: Geodesic forests for learning coupled predictors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 65–72, June 2013. 2, 3, 4, 5, 6

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1

[15] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. 3

[16] A. Montillo, J. Shotton, J. Winn, J. Iglesias, D. Metaxas, and A. Criminisi. Entangled decision forests and their application for semantic segmentation of ct images. In G. Szkely and H. Hahn, editors, *Information Processing in Medical Imaging*, volume 6801 of *Lecture Notes in Computer Science*, pages 184–196. Springer Berlin Heidelberg, 2011. 3

[17] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1668–1675, Nov 2011. 2, 3

[18] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 33–47. Springer International Publishing, 2014. 1, 3

[19] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative non-blind deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 604–611, June 2013. 2, 3

[20] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. bmvc, 2008. 1

[21] S. Seifert, A. Barbu, S. K. Zhou, D. Liu, J. Feulner, M. Huber, M. Suehling, A. Cavallaro, and D. Comaniciu. Hierarchical parsing and semantic navigation of full body ct data. volume 7259, pages 725902–725902–8, 2009. 1, 3

[22] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg, 2006. 1

[23] Z. Tu. Auto-context and its application to high-level vision tasks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 1, 2, 3, 4, 5