Learning Depth-Sensitive Conditional Random Fields for Semantic Segmentation of RGB-D Images

Andreas C. Müller and Sven Behnke

***Abstract*— We present a structured learning approach to semantic annotation of RGB-D images. Our method learns to reason about spatial relations of objects and fuses low-level class predictions to a consistent interpretation of a scene. Our model incorporates color, depth and 3D scene features, on which an energy function is learned to directly optimize object class prediction using the loss-based maximum-margin principle of structural support vector machines. We evaluate our approach on the NYU V2 dataset of indoor scenes, a challenging dataset covering a wide variety of scene layouts and object classes. We hard-code much less information about the scene layout into our model then previous approaches, and instead learn object relations directly from the data. We find that our conditional random field approach improves upon previous work, setting a new state-of-the-art for the dataset.**

I. INTRODUCTION

For robots to perform varied tasks in unstructured environments, understanding their surroundings is essential. We formulate the problem of semantic annotation of maps as a dense labeling of RGB-D images into semantic classes. Dense labeling of measured surfaces allows for a detailed reasoning about the scene.

In this work, we propose the use of random forests combined with conditional random fields (CRF) to perform robust estimation of structure classes in RGB-D images. The CRF is learned using a structural support vector machine, allowing it to integrate the noisy categorization produced by a pixel-based random forest to a consistent interpretation of the scene.

We thereby extend the success of learned CRF models for semantic segmentation in RGB images to the domain of 3D scenes. Our emphasis lies on exploiting the additional depth and 3D information in all processing steps, while relying on *learning* to create a model that is adjusted to the properties of the sensor input and environment.

Our approach starts with a random forest, providing a noisy local estimate of semantic classes based on color and depth information. These estimates are grouped together using a superpixel approach, for which we extend previous superpixel algorithms from the RGB to the RGB-D domain. We then build a geometric model of the scene, based on the neighborhood graph of superpixels. We use this graph not only to capture spatial relations in the 2D plane of the image, but also to model object distances and surface angles in 3D, using a point cloud generated from the RGB-D image. The process is illustrated in Figure 1.

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany. Email: amueller@ais.uni-bonn.de

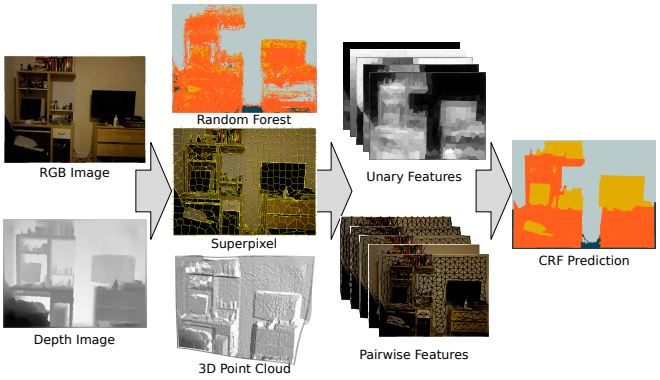

Fig. 1. Overview of the proposed semantic segmentation method.

We assess the accuracy of our model on the challenging NYU Segmentation V2 dataset [1], where our model outperforms previous approaches. Our analysis shows that while our random forest model already has competitive performance, the superpixel-based grouping and in particular the loss-based learning are integral ingredients of the success of our method.

II. RELATED WORK

The task of dense semantic annotation of 3D maps has seen an increased interest in recent years. Early work includes the approach of Nüchter and Hertzberg [2], who combine 6D SLAM, surface annotation, and object recognition to build semantically annotated maps. The approach was demonstrated on a mobile robot in an indoor environment. More recently, Sengupta *et al.* [3] introduced a dataset of semantically annotated street-scenes in a closed track, captured as pairs of stereo images. They approach the task by jointly reasoning about 3D layout and semantics of the scenes and produce a dense labeling on image level. Sengupta *et al.* [4] extended their approach to produce a volumetric reconstruction of the scene, together with a dense semantic labeling of the volumetric representation. This image segmentation method builds on the hierarchical CRF approach of Ladicky *et al.* [5], which is similar in spirit to our approach, but uses Potts potentials together with cross-validation to set potentials.

Recent work on indoor semantic annotation of maps mostly focused on RGB-D images, which are now easy to obtain using structured light sensors. Stückler *et al.* [6], for example, used a Random Forest to obtain a dense semantic labeling of images and integrated predictions over multiple

views in 3D. They evaluated their approach on table-top and simple indoors scenes. Silberman and Fergus [7] introduced the NYU Depth Dataset V1 dataset, which consisted of a large variety of densely annotated indoor scenes, captured as RGB-D images. Their work also introduced a baseline method for semantic segmentation of RGB-D image, which is based on a CRF over superpixels, with unary potentials given by interest point descriptors. While pairwise potentials for the CRF were carefully designed for the dataset, potentials were either directly set by hand or estimated using empirical frequencies. This is in contrast to our work, which applies structured prediction techniques to learn potentials that optimize predictive performance. Ren *et al.* [8] evaluated the design of features for semantic labeling of RGB-D data, and used a hierarchical segmentation to provide context. While they also defined a CRF on superpixels, their model is again not learned, but a weighted Potts model, using only a probability of boundary map, and not taking spatial layout into account at all. Silberman *et al.* [1] extended the NYU Depth Dataset V1 to the NYU Depth Dataset V2 that we are using in this work. Their focus is on inferring support relations in indoor scenes, such as objects resting on tables or shelfs, which in turn rest on the floor. Their approach is based on robust estimation of 3D plane hypotheses, which are then jointly optimized with support relations and structure classes. Silberman *et al.* [1] use a complex pipeline, employing significant domain knowledge. In this work, on the other hand, we try to learn all relevant domain specific features directly from the data, which allows us to out-perform the work of Silberman *et al.* [1] with respect to structure class segmentation.

Couprie *et al.* [9] approach the task of semantic segmentation of structure classes in RGB-D using the paradigm of convolutional neural networks, extending previous work of Farabet *et al.* [10] and Schulz and Behnke [11]. Similar to our approach, Couprie *et al.* [9] combine the output of a pixel-based, low-level learning algorithm with an independent unsupervised segmentation step. In contrast to their work, we improve our results by not only averaging predictions within superpixels, but also explicitly learning interactions between neighboring superpixels, favoring a consistent interpretation of the whole image.

Stückler *et al.* [12] extend the approach of Stückler *et al.* [6] to a real-time system for online learning and prediction of semantic classes. They use a GPU implementation of random forests, and integrate 3D scene information in an online fashion. They evaluate their approach on the dataset of Silberman *et al.* [1] with good results. We use the implementation of random forest provided by Stückler *et al.* [12], but instead of integrating predictions over time, we focus on exploiting the structure within a single frame.

While many of the works mentioned in this chapter make use of a CRF approach, we are not aware of any prior work on semantic annotation of 3D maps that fully learns their potentials.

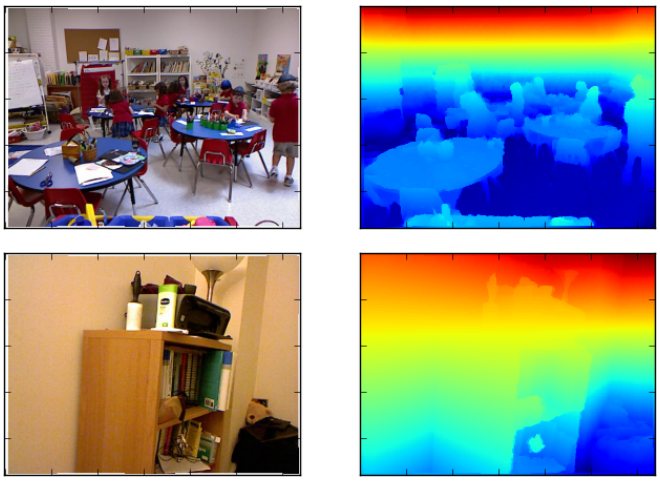

Fig. 2. Visualization of the height computed using the method described in Section III-B. Input images are shown on the left (depth not shown), the computed height is depicted on the right. The top row exemplifies a typical scene, while the bottom row shows a scene without horizontal surfaces, where our method fails.

III. LEARNING DEPTH-SENSITVE CONDITIONAL RANDOM FIELDS

We take a CRF approach, whose nodes represent a labeling of superpixels. We use an energy consisting of first and second order factors (also called unary and pairwise potentials), with learned potential functions. Let us denote the representation of an input image by x and a labeling of superpixels into semantic classes by y. Then the general form of the energy is

$$g(x,y) = \sum_{v \in V} \psi_v(x, y_v) + \sum_{(v,w) \in E} \psi_{v,w}(x, y_v, y_w). \quad (1)$$

Here V enumerates the superpixels, and $E \subset V \times V$ is a set of edges, encoding adjacence between superpixels.

We learn the unary and pairwise energy functions ψ_v and $\psi_{v,w}$ from the training data using a structural support vector machine (SSVM) [13]. The concept of SSVMs allows for a principled, maximum-margin based, loss-sensitive training of CRFs. Learning the potential yields much more complex interactions than the simple Potts potentials that are often used in the literature.

In general, structural SSVMs learn the parameters θ of a predictor of the form

$$f(x) = \arg\max_{y \in \mathcal{Y}} \theta^T \Phi(x, y). \quad (2)$$

We choose ψ in Equation (1) to be linear in the learnable parameters and the data-depended features, resulting in a form equivalent to Equation (2). Our features are described in detail below. We use the 1-slack formulation of the structural SVM [13] and solve the maximization in Equation (2) using a combination of fusion moves [14] and the AD^3 algorithm of Martins *et al.* [15]. In contrast to graph-cut inference, fusion moves and AD^3 can work with arbitrary potential functions, and allow precise learning using the SSVM approach.

A. Low Level Segmentation

We take a super-pixel based approach to semantic segmentation. Our superpixel generation is based on the SLIC algorithm [16]. We extend the standard SLIC algorithm, which works on the Lab space, to also include depth information. The resulting algorithm is a localized k-Means in Lab-D-XY space. Our implementation is publicly available through the scikit-image library[1]. Similar to Silberman *et al.* [1], we found little visual improvement over the RGB segmentation when using additional depth information. On the other hand, estimation of per-superpixel features based on the 3D point cloud was more robust when including depth information into the superpixel procedure. The resulting superpixels are compact in the 2D image. As the density of the corresponding point cloud is dependent on depth, we did not succeed in creating superpixels that are compact in 3D while maintaining a meaningful minimum size.

B. Unary Image Features

Our method builds on the probability output of a random forest, trained for pixel-wise classification of the structure classes. We use the GPU implementation provided by Stückler *et al.* [12][2]. The input for training are the full RGB-D images, transformed to Lab color space. Each tree in the forest uses training pixels only from a subset of training images. For each training image, an equal number of pixels for each occurring class is sampled. Split features are given by difference of regions on color or depth channels. Region size and offsets are normalized using the depth at the target pixel. We accumulate the probabilistic output for all pixels within a superpixel, and use the resulting distribution as a feature for the unary node potentials in our CRF model. We augment these prediction with another feature, based on the height of a superpixel in 3D. This is a very informative feature, in particular to determine the floor. To compute the height of a (super) pixel, we first find the "up" direction. We use a very simple approach that we found effective: we cluster normal directions of all pixels into 10 clusters using k-means, and use the one that is most parallel to the Y direction, which roughly corresponds to height in the dataset. We then project the 3D point cloud given by the depth along this direction, and normalize the result between 0 and 1. This procedure works robustly given there is some horizontal surface in the image, such as the ground or a table. A few scenes contain only walls and furniture, and the approach fails for these. Figure 2 illustrates a typical case and one of the much rarer failure cases. While we could use a more elaborate scheme, such as the one from Silberman *et al.* [1], we suspect that the feature is of little use in scenes without horizontal surfaces.

C. Pairwise Depth-Sensitive Features

There are five different features used to build pairwise potentials in our model:

[1]http://scikit-image.org

[2]https://github.com/deeplearningais/curfil

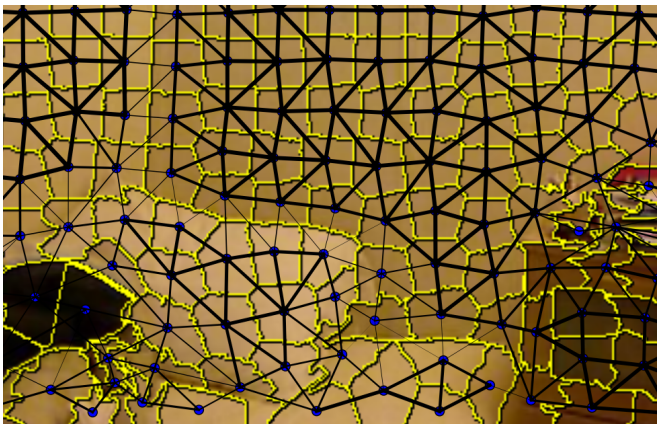

Fig. 3. Visualization of one of the pairwise features, the similarity between superpixel normals. The image shows the zoom-in of a bedroom scene, together with the superpixel over-segmentation. Lines connect adjacent superpixels, and line-strength gives the magnitude of the orientation similarity.

- *Constant.* A constant feature allows to model general neighborhood relations.
- *Color Contrast.* We employ a non-linear color contrast, as is common in the computer vision literature, between the superpixel mean colors c_i and c_j: $\exp\left(-\gamma\|c_i - c_j\|^2\right)$.
- *Vertical Alignment.* We model the directed angle between superpixel centers in the 2D image plane. This allows the model to learn that "structure" is above "floor", but not the other way around.
- *Depth Difference.* We include the signed depth difference between superpixels, which allows the model to detect depth discontinuities that are not represented in the 2D neighborhood graph of the superpixels.
- *Normal Orientations.* Differences in normal vector orientation are a strong clue on whether two superpixels belong to the same surface, and therefore the same structural class. We compute the 3D orientation of normals using the method of Holz *et al.* [17], as implemented in the point cloud library (pcl)[3]. All normals within a superpixel are then averaged, to get a single orientation for each superpixel. The feature is computed as the difference of $\frac{\pi}{4}$ and the (undirected) angle between the normals belonging to two adjacent superpixels. An example is shown in Figure 3. The change in normal orientation highlights that pillow and wall are distinct objects, even though there is no strong distinction in color or depth.

IV. EXPERIMENTS

We evaluate our approach on the public NYU depth V2 segmentation dataset of indoor scenes. The dataset comes with a detailed annotation of 1449 RGB-D images belonging to a wide variety of indoor scenes, categorized into 26 scene classes. The annotation contains four semantic structural classes: structure, floor, furniture and prop. There is an additional "void" class, which is used for object boundaries and hard-to-annotate regions. We follow the literature in excluding these pixels completely from the evaluation. We optimize our model for *average class accuracy* (the mean of

[3]http://pointclouds.org

TABLE I

QUANTITATIVE COMPARISON OF THE PROPOSED METHOD WITH THE LITERATURE.

	ground	structure	furniture	props	class average	pixel average
RF	90.8	81.6	67.9	19.9	65.0	68.3
RF + SP	92.5	83.3	**73.8**	13.9	65.7	70.1
RF + SP + SVM	94.4	79.1	64.2	**44.0**	70.4	70.3
RF + SP + CRF	**94.9**	78.9	71.1	42.7	**71.9**	**72.3**
Silberman *et al.* [1]	68	59	70	42	59.6	58.6
Couprie *et al.* [9]	87.3	**86.1**	45.3	35.5	63.5	64.5
Stückler *et al.* [12]†	**95.6**	83.0	**75.1**	14.2	67.0	70.9

The best value in each column is printed in bold†. The upper part of the table shows contributions by different parts of our pipeline. RF stands for random forest prediction, RF + SP for aggregated random forests prediction within superpixels, RF + SP + SVM for an SVM trained on the unary potentials, and RF + SP + CRF is our proposed pipeline. We optimized our approach for class average accuracy.
† Note that the work of Stückler *et al.* [12] is not directly comparable, as they integrated information over multiple frames, and did not measure accuracy for pixels without valid depth measurement.

the diagonal of the confusion matrix), putting more emphasis on the harder classes of props and furniture, which have smaller area than structure and floor. The dataset is split into 795 images for training and 654 images for testing. Our approach is implemented using our PYSTRUCT library[4]. We use SCIKIT-LEARN [18] for k-means clustering and the SVM baseline.

All hyper-parameters were adjusted using 5-fold cross-validation on the training set. The hyper parameters of the random forests were found using the `hyperopt` framework [19]. For the CRF model, the only hyper-parameters are related to the superpixel segmentation, and the single hyper-parameter C of the structural SVM formulation. These were adjusted using grid search. We found 500 superpixels per image to work best, which allow for a maximum possible performance of 95% average class accuracy on the validation set. Training of the random forests took about 15 minutes on a NVIDIA GeForce GTX Titan. Training the structural SVM took 45 minutes on a Xeon X5650 CPU. Prediction using only the random forest takes 33 ms on average, while segmentation and prediction using the structural SVM approximately take an additional 500 ms.

A. Results

Table I compares different components of our approach with the literature. Please note that we *first* designed our final model, using only the validation data. We now report accuracies of simpler models for reference, but these results were not used for model selection. To separate the influence of loss-based training and the spatial reasoning of the CRF, we also train a usual support vector machine (SVM) on the unary potentials for comparison.

The random forest prediction, as reported in Stückler *et al.* [12] is already quite competitive. Grouping into superpixels slightly improves performance, by removing high-frequency noise and snapping to object boundaries. Somewhat surprisingly, using a standard unstructured SVM with rescaled loss already advances the mean accuracy above the previous state-of-the art. We attribute this mostly to the ability of the SVM to exploit correlation between classes and uncertainty within the superpixels. Additionally, the SVM has access to the "height" feature, that was not included in the random forest. This performance is still improved upon, both in class average and pixel average performance by the learned CRF approach, yielding the best published result so far for both measures. The increase over the standard SVM is 1.5% for class average accuracy and 2.0% for pixel average accuracy.

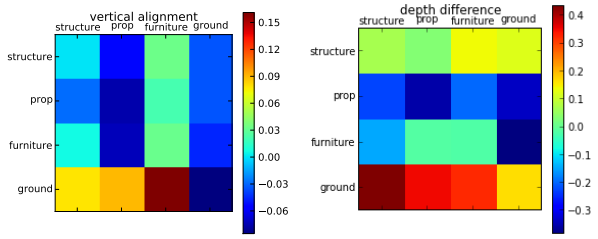

Fig. 4. Visualization of some of the learned potentials. The left potential is on the feature encoding whether one superpixel is above the other in the image. The right potential is applied to the relative depth between superpixels. See section IV-A for details.

A visualization of the impact of each processing step can be found in Figure 5, which shows prediction results on the test set. The four prediction methods correspond to the rows of Table I. The difference between the SVM and CRF approaches are clearly visible, with the CRF producing results that are very close to the ground truth in several complex scenes. We found that our approach improves results most for scenes with a clear geometric structure, which is not surprising. We see that evidence from the random forest is often very noisy, and biased away from the "props" class. While the unstructured SVM can correct somewhat for the class imbalance, it has no way to make larger areas consistent, which the CRF can. On the other hand, performance of the CRF deteriorates slightly on very crowded scenes with a mixture of small furniture and prop objects, as can be seen in the two right-most images. In these scenes, depth information is often noisy, and it is hard to make geometric statements on the superpixel level. As the input from the random forest is also often of low quality for crowded scenes, the CRF has little chance to recover.

Figure 4 visualizes two of the learned potential functions. Higher values correspond to favored configurations. We did not force potentials to be symmetric or anti-symmetric, which makes interpretation of the figures a bit harder, but increases performance. Edges are constructed to go from top

[4] `http://pystruct.github.io`

left to bottom right. Potentials below the diagonal are those for an edge going from a label given by the column to the one given by the row, while the ones above the diagonal are for the opposite direction. For vertical alignment, one would therfore expect to find anti-symmetric potentials. However the left-right direction seems to also contain useful information, breaking the symmetry. One can see that the vertical alignment potential expresses that the floor is much more likely to be below other classes. It also encodes the fact that props rest on furniture, but not the other way around. The potential of the depth feature encodes, for example, that the ground is usually behind the other classes, while furniture is in front of structures, such as the wall.

V. SUMMARY AND DISCUSSION

We introduce a CRF formulation for semantic segmentation of structure classes in RGB-D images. We base our model on the output of an efficient GPU implementation of random forest, and model spatial neighborhood using a superpixel-based approach. We combine color, depth and 3D orientation features into an energy function that is learned using the SSVM approach. By explicitly modeling 3D relations in a fully learned framework, we improve the state-of-the-art on the NYU V2 dataset for semantic annotation of structure classes. While our approach allows modeling of spatial relations, these are limited to local interactions. In future work, these interactions could be extended to larger areas using latent variable models or higher order potentials [5]. Another possible line of future work is to combine our approach with a more task-specific one, directly including support plane assumptions into the model, as done by Silberman *et al.* [1]. Finally, we could also combine our single-frame approach with the approach of Stückler *et al.* [12], which fuses individual views in 3D to exploit temporal coherence. While this work did not explicitly address real time application, the random forest implementation that we build upon allows for real-time processing [12]. The SLIC superpixel algorithm can also be implemented on GPU in real-time, as was demonstrated by Ren and Reid [20], and similarly the normal features we use also have real-time capabilities [17]. Finally, fusion move inference for our model is very efficient for our model, opening up the possibility to implement our approach entirely in real time.

REFERENCES

[1] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in *European Conference on Computer Vision*, 2012.

[2] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, 2008.

[3] S. Sengupta, P. Sturgess, P. H. Torr, *et al.*, "Automatic dense visual semantic mapping from street-level imagery," in *Intelligent Robots and Systems*, 2012.

[4] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr, "Urban 3d semantic modelling using stereo vision," in *International Conference on Robotics and Automation*, 2013.

[5] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Associative hierarchical CRFs for object class image segmentation," in *International Conference on Computer Vision*, 2009.

[6] J. Stückler, N. Biresev, and S. Behnke, "Semantic mapping using object-class segmentation of RGB-D images," in *Intelligent Robots and Systems*, 2012.

[7] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Computer Vision Workshops (ICCV Workshops)*, 2011.

[8] X. Ren, L. Bo, and D. Fox, "RGB-(D) Scene Labeling: Features and Algorithms," in *Computer Vision and Pattern Recognition*, 2012.

[9] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," in *International Conference on Learning Representations*, 2013.

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence*, 2013.

[11] H. Schulz and S. Behnke, "Learning object-class segmentation with convolutional neural networks," in *11th European Symposium on Artificial Neural Networks (ESANN)*, vol. 3, 2012.

[12] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense Real-Time Mapping of Object-Class Semantics from RGB-D Video," *Journal of Real-Time Image Processing*, 2014.

[13] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, 2009.

[14] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for markov random field optimization," *Pattern Analysis and Machine Intelligence*, 2010.

[15] A. F. Martins, M. A. Figueiredo, P. M. Aguiar, N. A. Smith, and E. P. Xing, "An augmented lagrangian approach to constrained map inference," in *International Conference on Machine Learning*, 2011.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *Pattern Analysis and Machine Intelligence*, 2012.

[17] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-Time Plane Segmentation using RGB-D Cameras," in *RoboCup International Symposium*, 2011.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, *et al.*, "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, 2011.

[19] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, *et al.*, "Algorithms for hyper-parameter optimization," in *Neural Information Processing Systems*, 2011.

[20] C. Y. Ren and I. Reid, "gSLIC: a real-time implementation of SLIC superpixel segmentation," *University of Oxford, Technical Report*, 2011.

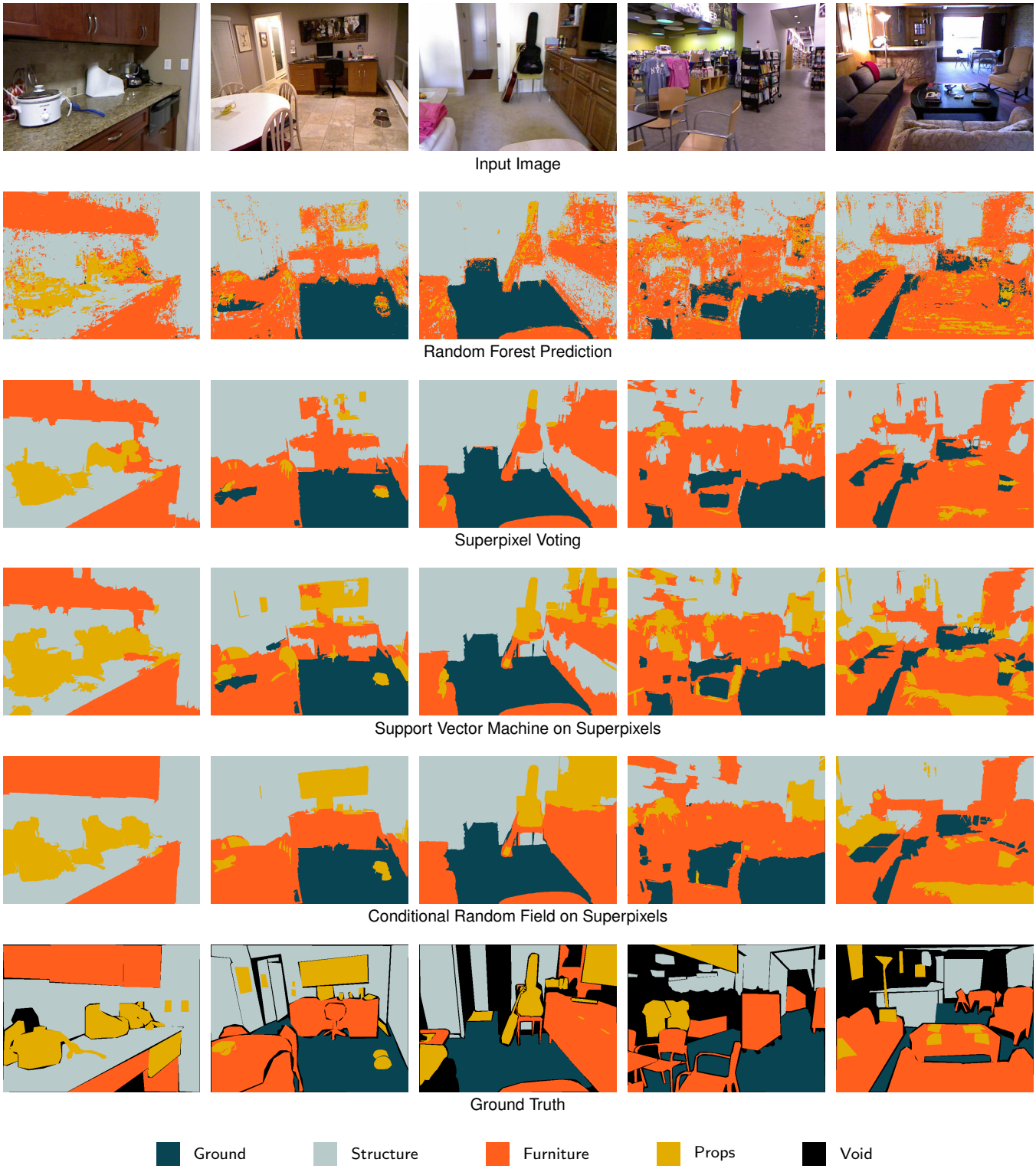

Fig. 5. Qualitative evaluation of the CRF. The first three images illustrate errors in the original prediction that can be corrected, while the second two images illustrate failure modes. Pixels marked as void are excluded from the evaluation. See the section IV-A for details.