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

Andreas Müller[1], Sven Behnke

Institute for Autonomous Intelligent Systems
Rheinische Friedrich-Wilhelms-Universität

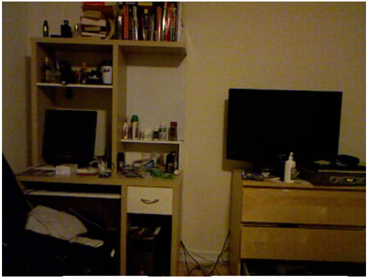
universitätbonn

Bonn, Germany

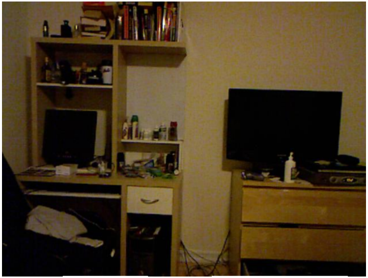
June 4, 2014

[1] now at Amazon.com

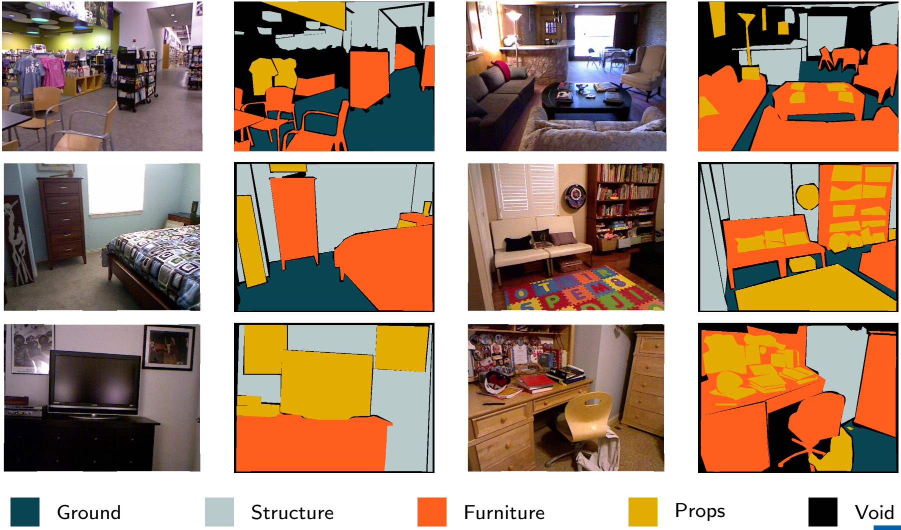
# Semantic Segmentation of Structure Classes



# Semantic Segmentation of Structure Classes

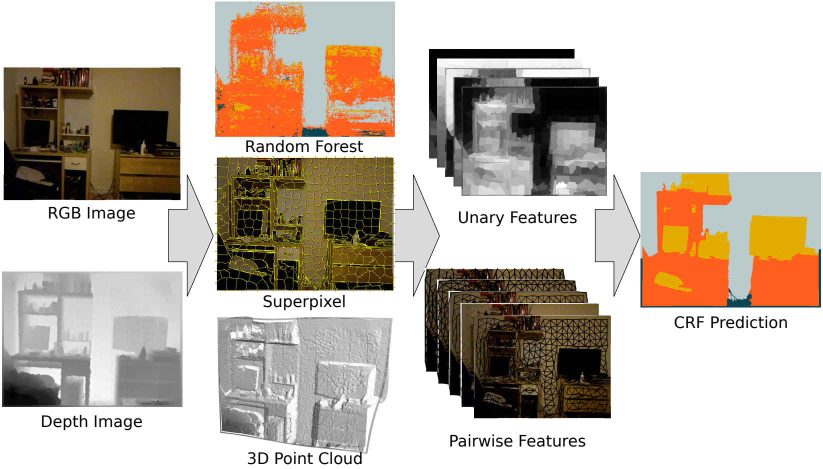


# Dataset: NYUv2



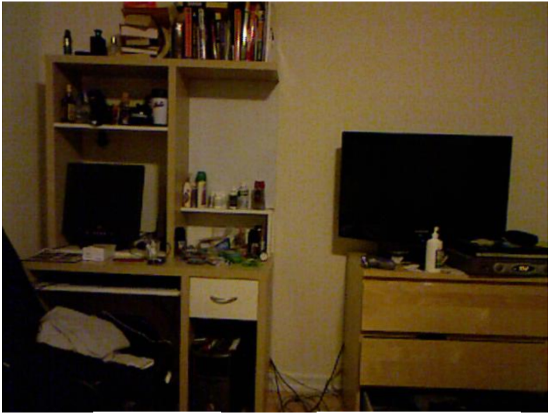

795 training images, 654 test images.

# Overview

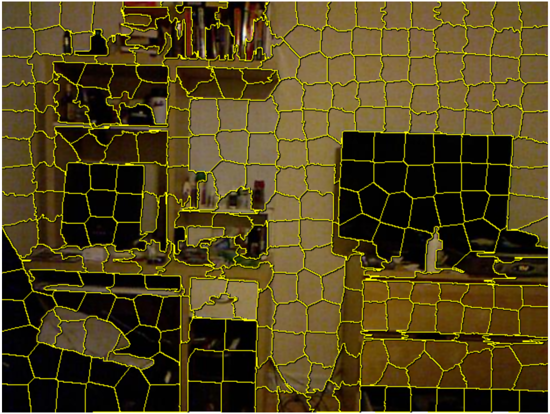

RGB Image
Depth Image
Random Forest
Superpixel
3D Point Cloud
Unary Features
Pairwise Features
CRF Prediction

# Outline

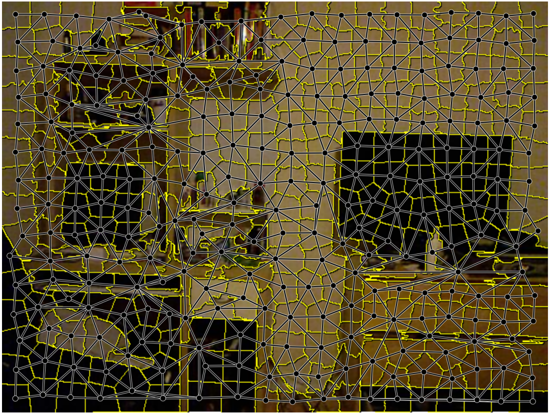
# Semantic Segmentation as Structured Prediction



# Semantic Segmentation as Structured Prediction



# Semantic Segmentation as Structured Prediction



# Structured Prediction

Learn prediction function of the form

$$g(x, w) := \arg\max_{y \in \mathcal{Y}} w^T \psi(x, y)$$

Objective

$$\min_w \frac{1}{2}\|w\|^2 + C \sum_i \ell(x^i, y^i, w)$$

$$\ell(x^i, y^i, w) = [\max_{y \in \mathcal{Y}} \Delta(y^i, y) + w^T \psi(x^i, y) - w^T \psi(x^i, y^i)]_+.$$

# Pairwise models

$$w^T \psi(x, y) = \sum_{(i,j)\in E} w_{i,j} \psi_{i,j}(x, y_i, y_j) + \sum_i w_i \psi_i(x, y_i)$$

# Pairwise models

$$w^T\psi(x,y) = \sum_{(i,j)\in E} w_{i,j}\psi_{i,j}(x,y_i,y_j) + \sum_i w_i\psi_i(x,y_i)$$

$$\psi_i(x,y_i) = f(x_i) \otimes 1_{y_i}$$
$$\psi_i(x,y_i,y_j) = f(x_i,x_j) \otimes 1_{y_i} \otimes 1_{y_j}$$

universitätbonn

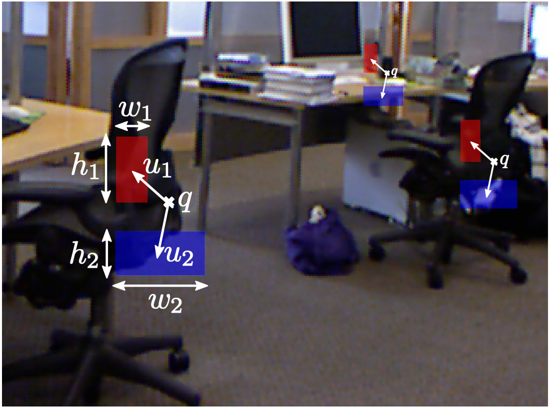
# Outline

Structured prediction formulation

Unary and Pairwise Features

Experiments

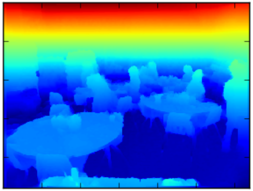
# Random Forest Unary Potentials




GPU implementation from Stückler et al. [2014].
Number of trees: 3, max-depth: 18

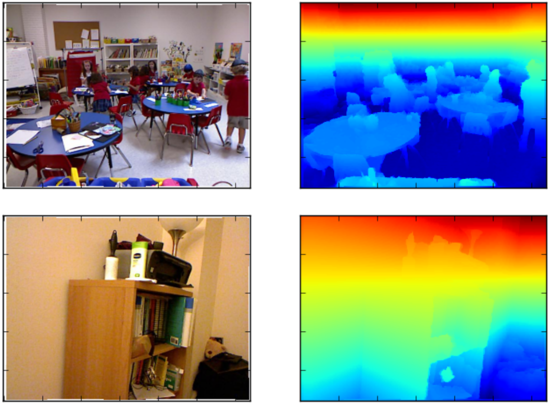
# Absolute height potentials

$$f_{\text{height}}(x_i) = \mathbf{n}_{\text{flat}}^T \mathbf{n}_{x_i}$$



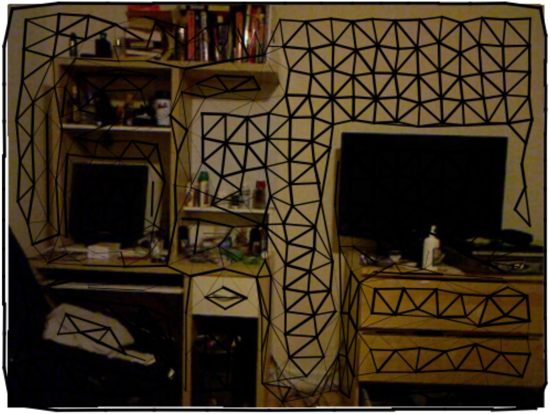
# Absolute height potentials

$$f_{\text{height}}(x_i) = \mathbf{n}_{\text{flat}}^T \mathbf{n}_{x_i}$$



# Color Contrast

$$f_{\text{color}}(x_i, x_j) = \exp\left(-\gamma \|c_i - c_j\|^2\right).$$

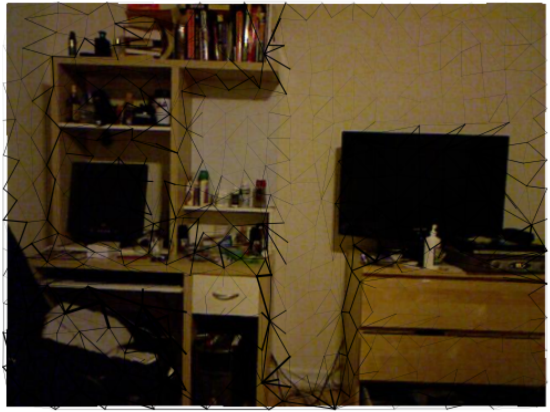


# Edge direction

$$f_{\text{direction}}(x_i, x_j) = \sphericalangle(\text{pos}_{x_i} - \text{pos}_{x_j}, [0,1]^T)$$

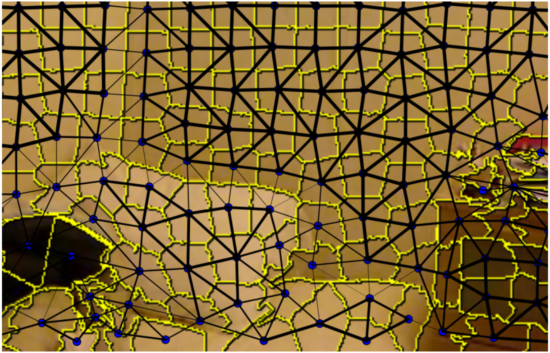
# Depth Difference

$$f_{\text{depth}}(x_i, x_j) = (d_i - d_j)/Z$$



# Pairwise Normal Orientation

$$f_{\text{normals}}(x_i, x_j) = 1 - \frac{1}{\pi} \sphericalangle(\mathbf{n}_{x_i}, \mathbf{n}_{x_j})$$



# Summary of Feature Functions

$$f(x_i) = f_{\text{RandomForest}}(x_i) \oplus f_{\text{height}}(x_i) \oplus 1$$

$$f(x_i, x_j) = f_{\text{color}}(x_i, x_j) \oplus f_{\text{direction}}(x_i, x_j) \oplus f_{\text{depth}}(x_i, x_j) \oplus f_{\text{normals}}(x_i, x_j) \oplus 1$$

# Learning and Optimization

- 1-slack SSVM

# Learning and Optimization

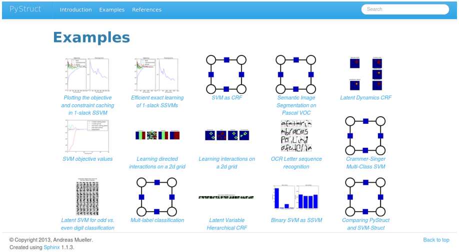
- 1-slack SSVM
- Inference using fusion moves and $AD^3$.

# Learning and Optimization

- 1-slack SSVM
- Inference using fusion moves and $AD^3$.
- Exact learning in loopy model [Müller and Behnke, 2014].

# Learning and Optimization

- 1-slack SSVM
- Inference using fusion moves and AD$^3$.
- Exact learning in loopy model [Müller and Behnke, 2014].




`http://pystruct.github.io`

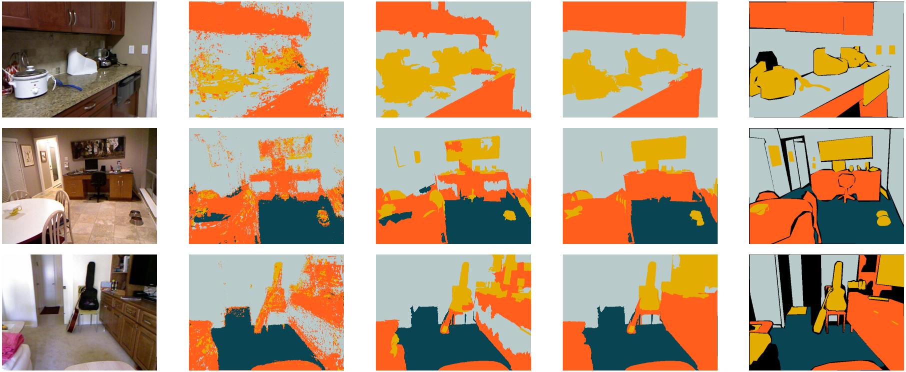
# Outline

# Results

| | ground | structure | furniture | props | class avg | pixel avg |
|---|---|---|---|---|---|---|
| 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. [2012] | 68 | 59 | 70 | 42 | 59.6 | 58.6 |
| Couprie et al. [2013] | 87.3 | **86.1** | 45.3 | 35.5 | 63.5 | 64.5 |
| Stückler et al. [2014]† | **95.6** | 83.0 | **75.1** | 14.2 | 67.0 | 70.9 |

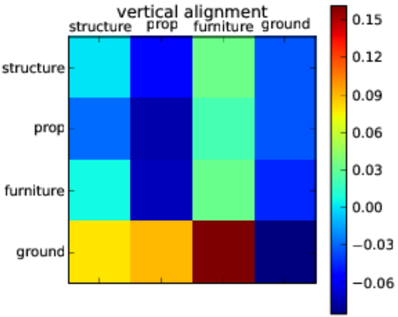
# Qualitative Results

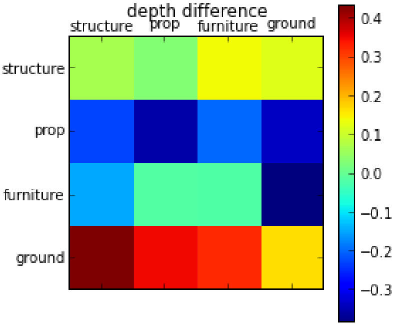
| Input | Random Forest | SVM on SP | CRF | Ground Truth |
| --- | --- | --- | --- | --- |



Ground · Structure · Furniture · Props · Void

# Learned Potentials



# Take Home

- Can incorporate geometric relations into CRF.
- Learn all potentials.
- Exact learning of CRF possible.
- Code available: `http://github.com/amueller/segmentation`

Thank you for your attention.

Questions?