DESIGNING DEEP NETWORKS FOR SURFACE NORMAL ESTIMATION

Xiaolong Wang, David F. Fouhey, Abhinav Gupta

Presented by Yu-Cheng Lin
The Problem

Given a single image (RGBD), estimates the surface normal at each pixel.
Dataset NYU Depth v2

Left:
Samples of the RGB image

Center:
The raw depth image

Right:
The class labels from the dataset

http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html
Dataset NYU Depth v2

Left:
Output from the RGB camera

Center:
Preprocessed depth

Right:
A set of labels for the image

http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html
Heart of the Problem: Two Questions

1. What are the right primitives for understanding?
   Use the data to derive a representation right from the pixels.

2. Given the local evidence, how can one obtain a global 3D scene understanding?
   Fusing global and local evidence with several constraints (man-made, Manhattan world) and meaningful intermediate representations (room layout, edge labels)
Overview

1. Separately learn global and local processes
2. Use a fusion network to fuse the contradictory beliefs into a final interpretation
Making Regression as Classification

Surface Normal:
1. Use k-means to learn a codebook
2. Delaunay triangulation cover is constructed over the words
3. Normals rewritten as weighted codeword combinations of the triangles

Room Layout:
Use k-medoids clustering over 6000 room layouts

Edge Label: convex, concave, occluding, no-edge
Global Network

Goal: capture the coarse structure, clarifying ambiguous image portions

Input: Whole image rescaled to 55 x 55 x 3

Output: (Mt = 20, Kt = 20)

surface normal estimation: Mt x Mt x Kt, Kt is # (classes in codebook)

room layout: simple classification over 300 categories
Local Network

**Goal:** capture local evidence at a higher resolution

**Input:** 55 x 55 sliding window on an image (size = 195 x 260) with stride = 13

**Output:** (Mb = 13, Kb = 40, to capture finer details)

- surface normal for M x M pixels at center of window: Mb x Mb x Kb
- edge label: one for 13 x 13 pixels
Loss Functions

room layout and edge label: softmax regression

surface normal estimation:

\[ L(I, Y) = - \sum_{i=1}^{M \times M} \sum_{k=1}^{K} (1(y_i = k) \log F_{i,k}(I)), \]  

\( F_{i,k}(I) \): probability that \( i^{\text{th}} \) pixel has the normal defined by the \( k^{\text{th}} \) codeword

\( 1(y_i = k) \): indicator function, \( Y = \{y_i\} \) are ground truth labels for normals
Visualization (Global Network)

Top regions for 4\textsuperscript{th} convolutional layer units

Receptive field: 31 x 31

Tendency: Capture structure information
Visualization (Local Network)

Top regions for 4\textsuperscript{th} convolutional layer units

Receptive field:
31 x 31

Tendency:
Respond to local texture and edges
Visualization (Local Network)

After sliding window, we obtain the surface normals and edge labels for the whole image.

Blue: convex  Green: concave  Red: occlusion

Edge label plotting: output of Structured Edges [6].

Fusion Network

**Input:** 55 x 55 sliding window on an image (size = 195 x 260) with stride = 13

- **Global Coarse Output:** 20 x 20 with 20 classes. Decode the output to a 3-d continuous surface normal map and upscale it to 195 x 260 x 3
- **Layout:** 3-channel normals in the layout. Resize it to 195 x 260 x 3
- **Vanishing Point-Aligned Coarse Output:** vanishing points estimated by [14], yielding another feature representation with the same size.

Fusion Network

**Input:** 55 x 55 sliding window on an image (size = 195 x 260) with stride = 13

- **Local Surface Normals:** 195 x 260 x 3
- **Edge Labels:** Upsample this 3-d vector to size 13 x 13 x 3 for each window and obtain 195 x 260 x 3 inputs. (no-edge excluded)

**Final Input:** 195 x 260 x 18 = (15 channels described above, original image)
Fusion Network

**Output:** (Mb = 13, Kb = 40, to capture finer details)

Surface normal for M x M pixels at center of window: Mb x Mb x Kb

**Testing:**
Apply the fusion network on the feature maps with the stride of Mb
Training Details

Fine-tune the network with stochastic gradient descent with learning rate \( \sigma = 1.0 \times 10^{-6} \)

During joint tuning with the layouts and edges, the learning rate set as \( 50 \times \sigma \)

For training the local and fusion network, rescale the training images to 195 x 260 and randomly sample 400K patches with size 55 x 55 from them
Qualitative Results

1. Captures the coarse layout of the room
2. Preserves the fine details. Notice that fine details like the top of couches and the legs of table are captured.
Qualitative Results
### Quantitative Results

Table 1: Results on NYU v2 for per-pixel surface normal estimation, evaluated over valid pixels.

<table>
<thead>
<tr>
<th></th>
<th>(Lower Better Mean)</th>
<th>Median</th>
<th>Higher Better 11.25°</th>
<th>22.5°</th>
<th>30°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Network</td>
<td>26.9</td>
<td>14.8</td>
<td>42.0</td>
<td>61.2</td>
<td>68.2</td>
</tr>
<tr>
<td>Stacked CNN [7]</td>
<td>23.7</td>
<td>15.5</td>
<td>39.2</td>
<td><strong>62.0</strong></td>
<td><strong>71.1</strong></td>
</tr>
<tr>
<td>UNFOLD [10]</td>
<td>35.2</td>
<td>17.9</td>
<td>40.5</td>
<td>54.1</td>
<td>58.9</td>
</tr>
<tr>
<td>Discr. [22]</td>
<td>33.5</td>
<td>23.1</td>
<td>27.7</td>
<td>49.0</td>
<td>58.7</td>
</tr>
<tr>
<td>3DP (MW) [9]</td>
<td>36.3</td>
<td>19.2</td>
<td>39.2</td>
<td>52.9</td>
<td>57.8</td>
</tr>
<tr>
<td>3DP [9]</td>
<td>35.3</td>
<td>31.2</td>
<td>16.4</td>
<td>36.6</td>
<td>48.2</td>
</tr>
</tbody>
</table>
Table 2: Ablative Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>11.25°</th>
<th>22.5°</th>
<th>30°</th>
<th>42.0</th>
<th>61.2</th>
<th>68.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>26.9</td>
<td>14.8</td>
<td>42.0</td>
<td>61.2</td>
<td>68.2</td>
<td></td>
</tr>
<tr>
<td>Full w/o Global</td>
<td>28.8</td>
<td>17.7</td>
<td>34.6</td>
<td>57.8</td>
<td>66.0</td>
<td></td>
</tr>
<tr>
<td>Fusion (+VP)</td>
<td>27.3</td>
<td>15.6</td>
<td>40.2</td>
<td>60.1</td>
<td>67.5</td>
<td></td>
</tr>
<tr>
<td>Fusion (+Edge)</td>
<td>27.8</td>
<td>16.4</td>
<td>37.5</td>
<td>59.4</td>
<td>67.4</td>
<td></td>
</tr>
<tr>
<td>Fusion (+Layout)</td>
<td>27.7</td>
<td>16.0</td>
<td>38.8</td>
<td>59.9</td>
<td>67.4</td>
<td></td>
</tr>
<tr>
<td>Fusion</td>
<td>27.9</td>
<td>16.6</td>
<td>37.4</td>
<td>59.4</td>
<td>67.1</td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td>34.0</td>
<td>25.1</td>
<td>25.6</td>
<td>46.4</td>
<td>56.2</td>
<td></td>
</tr>
<tr>
<td>Global</td>
<td>30.9</td>
<td>20.8</td>
<td>31.4</td>
<td>52.3</td>
<td>60.5</td>
<td></td>
</tr>
<tr>
<td>Coarse CNN [8]</td>
<td>30.1</td>
<td>24.7</td>
<td>24.1</td>
<td>46.4</td>
<td>57.9</td>
<td></td>
</tr>
</tbody>
</table>
Berkeley B3DO Dataset

Mismatch in dataset bias:
NYU: contains almost exclusively full scenes
B3DO: contains many close-ups

http://kinectdata.com/
Generalization Results to B3DO

Table 3: B3DO

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>11.25°</th>
<th>22.5°</th>
<th>30°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>34.5</td>
<td>20.1</td>
<td>36.7</td>
<td>52.4</td>
<td>59.2</td>
</tr>
<tr>
<td>3DP(MW) [9]</td>
<td>38.0</td>
<td>24.5</td>
<td>33.6</td>
<td>48.5</td>
<td>54.5</td>
</tr>
<tr>
<td>Hedau et al. [14]</td>
<td>43.5</td>
<td>30.0</td>
<td>32.8</td>
<td>45.0</td>
<td>50.0</td>
</tr>
<tr>
<td>Lee et al. [25]</td>
<td>41.9</td>
<td>28.4</td>
<td>32.7</td>
<td>45.7</td>
<td>50.8</td>
</tr>
</tbody>
</table>
Possible Extensions

[7] introduced a stacked CNN model for surface normal estimation. The contributions of this paper are complementary to [7] and combining both should provide further improvement.

Conclusion

1. A novel CNN-based approach for surface normal estimation
2. Designing networks based on meaningful intermediate representations and constraints can help improve the performance
3. Making Regression as Classification and adopt CNN architectures