ECS289 VISUAL RECOGNITION

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network – NIPS 2014
- D. Eigen, C. Puhrsch, and R. Fergus

Presenter Wei-Chih Chen(Michael)
2015/11/12
Introduction

- Estimate depth from a single image:
  - Line angles
  - Texture variations
  - Object sizes
  - Haze color (far)

Cite from Ashutosh Saxena
Model architecture

- Global Coarse-Scale Network – learn spatial information
- Local Fine-Scale Network – local refine
Global Coarse-Scale Network (1/2)

- Red: Positive (farther)
- Blue: Negative (closer)
- Record spatial features
Global Coarse-Scale Network (2/2)

- All hidden layers: rectified linear units
- Layer 7 is linear (Softmax)
- Dropout: fully connected hidden layer 6
Local Fine-Scale Network (1/2)
Local Fine-Scale Network (2/2)

- Convolutional layers only
- All hidden layers: rectified linear units
- Fine 4 is linear
- Keep coarse-scale output fixed
Previous 2D->3D work: Make3D(1/8)

- Using superpixels to find absolute depth.
- Group into small homogenous regions by filters
- Find 3D location and orientation of these patch

[Image of superpixels and filters]

Cite from Ashutosh Saxena
Markov Random Field (MRF) (2/8)

- $X_i, X_j$ are superpixels
- MRF models the relations (shown by the edge $X_{ij}$) between neighboring superpixels. This defines whether $X_i$ and $X_j$ are on the same plane.
Plane Parameters (3/8)

- View each superpixel is a plane
- Plane parameters ($\alpha \in \mathbb{R}^3$) (location, depth and orientation)
- Rays $R_i$, depth $d_i = 1/R_i^T\alpha$
- Want to model $P(\alpha \mid x; \theta)$; $\theta$ is MRF model

Cite from Ashutosh Saxena
Coplanarity and Connectivity (4/8)

$\alpha_C$ is different plane

Not Coplanar

Coplanar

$\alpha_B$ and $\alpha_A$ are same plane
Occlusion Boundary / Fold (5/8)

- Learn occlusion boundary/folds using features
- As 3D-model with true edges for MRF
- Learn $X_{ij}$ by $y_{ij}$

$y$: occlusion boundary or fold.
$y_{ij}=0$ indicates an boundary (black)
MRF Model (6/8)
MRF Model (7/8)
MRF Model (8/8)

\[
P(\alpha | X, \nu, y, R; \theta) = \frac{1}{Z} \prod_i f_1(\alpha_i | X_i, \nu_i, R_i; \theta) \prod_{i,j} f_2(\alpha_i, \alpha_j | y_{ij}, R_i, R_j)
\]

\[\begin{align*}
\alpha & : \text{Plane parameters} \\
X & : \text{Image features} \\
y & : \text{Occlusion boundary/fold} \\
R & : \text{Rays from the camera} \\
v & : \text{Confidence in features}
\end{align*}\]

\[\alpha^* = \arg\max_\alpha \log P(\alpha | X, \nu, y, R; \theta_r)\]
Scale-Invariant Error

- Make3D uses elementwise to get each plane but average error is still high (0.41 and 0.33 error on RMSE).
- Not only minimize distance between $y$ and $y^*$ in each pixels, but also minimize average error at the same time.

$$D(y, y^*) = \frac{1}{2n} \sum_{i=1}^{n} \left( \log y_i - \log y_i^* + \alpha(y, y^*) \right)^2,$$

where 
$$\alpha(y, y^*) = \frac{1}{n} \sum_i (\log y_i^* - \log y_i)$$
Scale-Invariant Error

\[ D(y, y^*) = \frac{1}{2n} \sum_{i=1}^{n} (\log y_i - \log y_i^* + \alpha(y, y^*))^2, \quad (1) \]

where \( \alpha(y, y^*) = \frac{1}{n} \sum_i (\log y_i^* - \log y_i) \)

\[ D(y, y^*) = \frac{1}{2n^2} \sum_{i,j} \left( (\log y_i - \log y_j) - (\log y_i^* - \log y_j^*) \right)^2 \]

\[ = \frac{1}{n} \sum_i d_i^2 - \frac{1}{n^2} \sum_i d_i d_j = \frac{1}{n} \sum_i d_i^2 - \frac{1}{n^2} \left( \sum_i d_i \right)^2 \quad (3) \]

where \( d_i = \log y_i - \log y_i^* \)
Scale-Invariant Error

- Not only decrease error pixelwise but also error between other pixels.
- Every pixels share the same error. For any prediction $y$, $e$ is the scale that best aligns it to the ground truth. All scalar multiples of $y$ have the same error, hence the scale invariance.

$$D(y, y^*) = \frac{1}{n} \sum_i d_i^2 - \frac{1}{n^2} \sum_{i,j} d_i d_j = \frac{1}{n} \sum_i d_i^2 - \frac{1}{n^2} \left( \sum_i d_i \right)^2$$

L2 error  Credits mistake if they are predict wrong in the same direction.
Experiments

- **Loss function:**
  \[ L(y, y^*) = \frac{1}{n} \sum d_i^2 - \frac{\lambda}{n^2} \left( \sum d_i \right)^2 \]

- **Data Augmentation:** scale, rotation, translation, color, flips

- **NYU Depth v2** (By Microsoft Kinect camera)
  - 249 scenes for training, 215 for testing,
  - Coarse network 2M using SGD with batches of size 32,
  - Fine network for 1.5M samples.

- **KITTI** (By extra LIDAR scanner)
  - 28 scenes for training, 28 scenes for testing,
  - Coarse network 1.5M, fine network for 1M samples.
Experiments

Threshold: % of $y_i$ s.t. $\max\left(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}\right) = \delta < thr$

Abs Relative difference: $\frac{1}{|T|} \sum_{y \in T} |y - y^*| / y^*$

Squared Relative difference: $\frac{1}{|T|} \sum_{y \in T} (y - y^*)^2 / y^*$

RMSE (linear): $\sqrt{\frac{1}{|T|} \sum_{y \in T} ||y_i - y_i^*||^2}$

RMSE (log): $\sqrt{\frac{1}{|T|} \sum_{y \in T} ||\log y_i - \log y_i^*||^2}$

RMSE (log, scale-invariant): The error Eqn. 1

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Make3D</th>
<th>Ladicky &amp; al</th>
<th>Karch &amp; al</th>
<th>Coarse</th>
<th>Coarse + Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold $\delta &lt; 1.25$</td>
<td>0.418</td>
<td>0.447</td>
<td>0.542</td>
<td>–</td>
<td>0.618</td>
<td>0.611</td>
</tr>
<tr>
<td>threshold $\delta &lt; 1.25^2$</td>
<td>0.711</td>
<td>0.745</td>
<td>0.829</td>
<td>–</td>
<td>0.891</td>
<td>0.887</td>
</tr>
<tr>
<td>threshold $\delta &lt; 1.25^3$</td>
<td>0.874</td>
<td>0.897</td>
<td>0.940</td>
<td>–</td>
<td>0.969</td>
<td>0.971</td>
</tr>
<tr>
<td>abs relative difference</td>
<td>0.408</td>
<td>0.349</td>
<td>–</td>
<td>0.350</td>
<td>0.228</td>
<td>0.215</td>
</tr>
<tr>
<td>sqr relative difference</td>
<td>0.581</td>
<td>0.492</td>
<td>–</td>
<td>–</td>
<td>0.223</td>
<td>0.212</td>
</tr>
<tr>
<td>RMSE (linear)</td>
<td>1.244</td>
<td>1.214</td>
<td>–</td>
<td>1.2</td>
<td>0.871</td>
<td>0.907</td>
</tr>
<tr>
<td>RMSE (log)</td>
<td>0.430</td>
<td>0.409</td>
<td>–</td>
<td>–</td>
<td>0.283</td>
<td>0.285</td>
</tr>
<tr>
<td>RMSE (log, scale inv.)</td>
<td>0.304</td>
<td>0.325</td>
<td>–</td>
<td>–</td>
<td>0.221</td>
<td>0.219</td>
</tr>
</tbody>
</table>

Table 1: Comparison on the NYUDepth dataset
Experiments

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Make3D</th>
<th>Coarse</th>
<th>Coarse + Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold $\delta &lt; 1.25$</td>
<td>0.556</td>
<td>0.601</td>
<td>0.679</td>
<td>0.692</td>
</tr>
<tr>
<td>threshold $\delta &lt; 1.25^2$</td>
<td>0.752</td>
<td>0.820</td>
<td>0.897</td>
<td>0.899</td>
</tr>
<tr>
<td>threshold $\delta &lt; 1.25^3$</td>
<td>0.870</td>
<td>0.926</td>
<td>0.967</td>
<td>0.967</td>
</tr>
<tr>
<td>abs relative difference</td>
<td>0.412</td>
<td>0.280</td>
<td>0.194</td>
<td>0.190</td>
</tr>
<tr>
<td>sqr relative difference</td>
<td>5.712</td>
<td>3.012</td>
<td>1.531</td>
<td>1.515</td>
</tr>
<tr>
<td>RMSE (linear)</td>
<td>9.635</td>
<td>8.734</td>
<td>7.216</td>
<td>7.156</td>
</tr>
<tr>
<td>RMSE (log)</td>
<td>0.444</td>
<td>0.361</td>
<td>0.273</td>
<td>0.270</td>
</tr>
<tr>
<td>RMSE (log, scale inv.)</td>
<td>0.359</td>
<td>0.327</td>
<td>0.248</td>
<td>0.246</td>
</tr>
</tbody>
</table>

higher is better

Table 2: Comparison on the KITTI dataset.

Figure 3: Qualitative comparison of Make3D, our method trained with $l_2$ loss ($\lambda = 0$), and our method trained with both $l_2$ and scale-invariant loss ($\lambda = 0.5$).
Experiments

(a) input, (b) output of coarse network, (c) refined output of fine network, (d) ground truth.
Experiments

- uncorrected alignment issues between the depth map and input in the training data
Conclusion

• Coarse network -> global depth
• Fine network -> local depth
• Scale-invariant error -> reduce relevant error