Problem: Noisy Kinect Problem

Map background as foreground

Map foreground as background

\[(X, Y)_{r,c} = \text{row, column of the pixel}\]
\[X = 8\text{-bit color of pixel } r,c\]
\[Y = 8\text{-bit depth of pixel } r,c\]

Problem: Many pixels have missing or wrong depth

values

\[(X, ?)_{r,c} \Rightarrow \text{Find ?}\]

Solution 1:

\[
\begin{array}{ccc}
X & Y & P(X,Y) \\
0 & 0 & 0.001 \\
\vdots & \vdots & \vdots \\
255 & 255 & 0.001 \\
\end{array}
\]

If \((X=67, ?)\)

\[\hat{Y}_{\text{est}} = \max_{*} P(X=67, Y=*)\]

- inference
- probability inference \(\Rightarrow\)

Optimization
This solution does not work well because color and depth are not that related.

What's missing: neighborhood context

Soln 2:

<table>
<thead>
<tr>
<th>(Y, Y)_{-1}</th>
<th>(X, Y)_{-1, -1}</th>
<th>(X, Y)_{0, -1}</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X, Y)_{0}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(X, Y)_{0, 0}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table size = \( 256 \times 256 \times 256 \times 256 \times \ldots \times 256 \times 256 \)

\[ = 256^{18} \approx 10^{43} \]

⇒ will need \( 10^{30} \) TB hard drive

Big Problem: Not enough memory

Possible solution

1. Approximate optimization
   - Sampling (Monte Carlo simulation)
   - Functional approximation
   - Loopy belief propagation, variational, particle filtering

2. Compression
   - Geometrical or graph-theoretical
Geometrical

\[
\begin{array}{ccc}
10 & 10 & 10 \\
10 & 10 & 10 \\
10 & 10 & 10 \\
\end{array}
\sim
\begin{array}{ccc}
11 & 10 & 11 \\
9 & 10 & 9 \\
9 & 11 & 9 \\
\end{array}
\]

X: Color patch

\[\text{visual distance} \approx 0\]

Compression — clustering: group similar neighborhood contexts together — hard separation

Compression — dimension reduction

\[X \rightarrow f(X)\]

9-dim \quad \text{ex} \; 2-dim

ex/ PCA, Kernel PCA

Graph-theoretical

Factorization: let's say

\[P(\{(X, Y)_{0,0}, (X, Y)_{1,-1}, (X, Y)_{0,-1}, (X, Y)_{1,-1}\}, \ldots) = \frac{g_1((X, Y)_{-1,-1}, (X, Y)_{0,-1}) \cdot g_2((X, Y)_{0,-1}, (X, Y)_{1,-1}) \cdot g_3 \cdot g_4 \cdot g_5 \cdot g_6 \cdot g_7 \cdot g_8 \cdot g_9 \cdot g_{10} \cdot g_{11} \cdot g_{12}}{g} \]

\]
To store
\[ g_1 : 256^4 \]
\[ g_2 : \ldots \]
\[ g_{12} : 256^4 \]

Total memory needed \( = 12 \times 256^4 \approx 4.2 \text{ GB} \)

Path blockage: \( Z_{-1,1} \) is conditionally independent of \( Z_{1,-1} \) given \( Z_{0,1} \) and \( Z_{1,0} \)

\[ \text{(Reachability)} \]

\( Z_{1,0} \) is conditionally independent of everything else given \( Z_{1,-1}, Z_{0,0}, Z_{-1,1} \)

Non-trivial: Factorization \( \iff \) Conditional Independence statements

Another way to summarize:

Each pixel is conditionally independent of all other pixels given its immediate neighbors

\( \Rightarrow \) Encode our knowledge: Natural image is smooth!
Adding more variables

Depth value \( Y_{i,j} \) — true depth value

\( \hat{Y}_{i,j} \) — kinect depth value

Model: \( \hat{Y}_{i,j} = Y_{i,j} + N \)

independent of \( Y_{i,j} \)

\[
P(\hat{Y}_{i,j}) = \int P(\hat{Y}_{i,j}, Y_{i,j}) \, dY_{i,j}
\]

\[
= \int P(\hat{Y}_{i,j} | Y_{i,j}) \cdot P(Y_{i,j}) \, dY_{i,j}
\]

\[
= \int P(Y_{i,j} + N | Y_{i,j}) \cdot P(Y_{i,j}) \, dY_{i,j}
\]

\[
= \int P(N) \cdot P(Y_{i,j}) \, dY_{i,j}
\]

Just 2 parameter: mean, variance

Our solution: mean = 0

variance \( \propto / \text{distance of the pixel to Binary Background/Foground} \) the closest color edge
Without hidden variables: \( P(X_{ij}, Y_{ij}) \)

With hidden variables: \( f(M_{ij}, X_{ij}) \cdot g(M_{ij}, Y_{ij}) \cdot \frac{1}{\sigma \sqrt{2\pi}} \int_{\mathbb{R}^d} P(Y_{ij}) dx_{ij} \)

\( \sim O(|X_{ij}| \cdot |M_{ij}|) \)

\( \sim O(|Y_{ij}|) \)

Same complexity as \( P(Y_{ij}) \)