Semantic 3D Modelling

Ľúbor Ladický

work with Christian Häne, Nikolay Savinov, Jianbo Shi, Bernhard Zeisl, Marc Pollefeys
Schedule

• Introduction
  • Discrete MRF Optimization using Graph Cuts
  • Classifiers for Semantic 3D Modelling

• Higher Order MRFs with Ray Potentials
  • Discrete Formulation
  • Continuous Relaxation
Schedule

- Introduction
  - Discrete MRF Optimization using Graph Cuts
  - Classifiers for Semantic 3D modelling

- Higher Order MRFs with Ray Potentials
  - Discrete formulation
  - Continuous relaxation
Graph-Cut (st-mincut)

Set formulation:
\[
\min_{S,T} \sum_{i \in S, j \in T} c_{ij} \quad \text{s.t.} \quad s \in S, \quad t \in T
\]
Graph-Cut (st-mincut)

Set formulation

\[
\min_{S,T} \sum_{i \in S, j \in T} c_{ij} \\
\text{s.t. } s \in S, \ t \in T
\]
Graph-Cut (st-mincut)

Set formulation

$$\min_{S,T} \sum_{i \in S, j \in T} c_{ij} \quad s.t. \quad s \in S, \quad t \in T$$

Algebraic formulation

$$\min_{x} \sum_{(i,j) \in E} c_{ij}(1 - x_i)x_j \quad s.t. \quad x_s = 0 \quad x_t = 1$$

$$x_i = 0 \implies x_i \in S \quad x_i = 1 \implies x_i \in T$$
Graph-Cut (st-mincut)

**Set formulation**

\[
\min_{S,T} \sum_{i \in S, j \in T} c_{ij} \quad \text{s.t.} \quad s \in S, \ t \in T
\]

**Algebraic formulation**

\[
\min_{x} \sum_{(i,j) \in E} c_{ij}(1 - x_i)x_j \quad \text{s.t.} \quad x_s = 0 \quad x_t = 1
\]

\[
x_i = 0 \implies x_i \in S \quad x_i = 1 \implies x_i \in T
\]

**After substitution**

\[
\min_{x} \sum_{(s,i) \in E} c_{si}x_i + \sum_{(i,t) \in E} c_{it}(1 - x_i) + \sum_{(i,j) \in E, i,j \notin \{s,t\}} c_{ij}(1 - x_i)x_j
\]

\[
x_i = 0 \implies x_i \in S \quad x_i = 1 \implies x_i \in T
\]
Graph-Cut (st-mincut)

Algorithms
Augmented path method
Push-relabel method

Cost = 18
Foreground / Background Estimation

Rother et al. SIGGRAPH04
Foreground / Background Estimation

\[ E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) \]

Data term  Smoothness term

\[ \psi_i(0) = -\log(p(x_i \notin FG)) \]
\[ \psi_i(1) = -\log(p(x_i \in FG)) \]

Estimated using FG / BG colour models

Smoothness term

\[ \psi_{ij}(x_i, x_j) = K_{ij} \delta(x_i \neq x_j) \]

Intensity dependent smoothness

where

\[ K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2) \]
Foreground / Background Estimation

\[ E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) \]

\[ x_i = 0 \implies i \in \text{Background} \]
\[ x_i = 1 \implies i \in \text{Foreground} \]

Data term

\[ \psi_i(x_i) = \psi_i(0)(1 - x_i) + \psi_i(1)x_i \]

Smoothness term

\[ \psi_{ij}(x_i, x_j) = K_{ij}\delta(x_i \neq x_j) = K_{ij}(1 - x_i)x_j + K_{ij}(1 - x_j)x_i \]

\[ x^* = \arg \min_x \sum_{i \in \mathcal{V}} \psi_i(0)(1 - x_i) + \psi_i(1)x_i + \sum_{i, j \in \mathcal{E}} (K_{ij}(1 - x_i)x_j + K_{ij}(1 - x_j)x_i) \]
Foreground / Background Estimation

\[ E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) \]

\[ x_i = 0 \implies i \in \text{Background} \]
\[ x_i = 1 \implies i \in \text{Foreground} \]

**Data term**

\[ \psi_i(x_i) = \psi_i(0)(1 - x_i) + \psi_i(1)x_i \]

**Smoothness term**

\[ \psi_{ij}(x_i, x_j) = K_{ij}\delta(x_i \neq x_j) = K_{ij}(1 - x_i)x_j + K_{ij}(1 - x_j)x_i \]

**Min-Cut problem**

\[ x^* = \arg \min_x \sum_{i \in \mathcal{V}} \psi_i(0)(1 - x_i) + \psi_i(1)x_i + \sum_{i, j \in \mathcal{E}} (K_{ij}(1 - x_i)x_j + K_{ij}(1 - x_j)x_i) \]
Foreground / Background Estimation

\[ E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) \]

\[ x_i = 0 \implies i \in \text{Background} \]
\[ x_i = 1 \implies i \in \text{Foreground} \]
Solvability using GraphCut

Submodularity

\[ E(0, 0, \bar{x}_{ij}) + E(1, 1, \bar{x}_{ij}) \leq E(0, 1, \bar{x}_{ij}) + E(1, 0, \bar{x}_{ij}) \]
Solvability using GraphCut

Submodularity

\[ E(0, 0, \bar{x}_{ij}) + E(1, 1, \bar{x}_{ij}) \leq E(0, 1, \bar{x}_{ij}) + E(1, 0, \bar{x}_{ij}) \]

\[
\arg\min_x \sum_{i \in V} c_{it}(1 - x_i) + \sum_{i \in V} c_{si}x_i + \sum_{i,j \in \mathcal{E}} c_{ij}(1 - x_i)x_j
\]

all terms submodular

submodularity = necessary condition
Solvability using GraphCut

**Submodularity**

\[ E(0, 0, \bar{x}_{ij}) + E(1, 1, \bar{x}_{ij}) \leq E(0, 1, \bar{x}_{ij}) + E(1, 0, \bar{x}_{ij}) \]

**General pairwise potential**

\[ \psi_{ij}(x_i, x_j) = g_{ij}^{00}(1 - x_i)(1 - x_j) + g_{ij}^{01}(1 - x_i)x_j + g_{ij}^{10}x_i(1 - x_j) + g_{ij}^{11}x_ix_j \]
Solvability using GraphCut

Submodularity

\[ E(0, 0, \bar{x}_{ij}) + E(1, 1, \bar{x}_{ij}) \leq E(0, 1, \bar{x}_{ij}) + E(1, 0, \bar{x}_{ij}) \]

General pairwise potential

\[ \psi_{ij}(x_i, x_j) = g_{ij}^{00}(1 - x_i)(1 - x_j) + g_{ij}^{01}(1 - x_i)x_j + g_{ij}^{10}x_i(1 - x_j) + g_{ij}^{11}x_ix_j \]

\[ = K_{ij} + g'_i x_i + g'_j x_j + c_{ij}(1 - x_i)x_j + c_{ij}x_i(1 - x_j) \]

where

\[ K_{ij} = g_{ij}^{00} \]

\[ g'_i = \frac{g_{ij}^{01} + g_{ij}^{11} - g_{ij}^{00}}{2} \]

\[ g'_j = \frac{g_{ij}^{01} + g_{ij}^{11} - g_{ij}^{10} - g_{ij}^{00}}{2} \]

\[ c_{ij} = \frac{g_{ij}^{01} + g_{ij}^{10} - g_{ij}^{00} - g_{ij}^{11}}{2} \]
Solvability using GraphCut

Submodularity

\[
E(0, 0, \bar{x}_{ij}) + E(1, 1, \bar{x}_{ij}) \leq E(0, 1, \bar{x}_{ij}) + E(1, 0, \bar{x}_{ij})
\]

General pairwise potential

\[
\psi_{ij}(x_i, x_j) = g_{ij}^{00}(1 - x_i)(1 - x_j) + g_{ij}^{01}(1 - x_i)x_j + g_{ij}^{10}x_i(1 - x_j) + g_{ij}^{11}x_ix_j
\]

\[
= K_{ij} + g'_{i}x_i + g'_{j}x_j + c_{ij}(1 - x_i)x_j + c_{ij}x_i(1 - x_j)
\]

could be arbitrary

where

\[
K_{ij} = g_{ij}^{00}
\]

\[
g'_{i} = \frac{g_{ij}^{10} + g_{ij}^{11} - g_{ij}^{01} - g_{ij}^{00}}{2}
\]

\[
g'_{j} = \frac{g_{ij}^{01} + g_{ij}^{11} - g_{ij}^{10} - g_{ij}^{00}}{2}
\]

\[
c_{ij} = \frac{g_{ij}^{01} + g_{ij}^{10} - g_{ij}^{00} - g_{ij}^{11}}{2} \geq 0
\]

Submodularity = sufficient condition
Energy minimization transformed into GraphCut:

- Each state of original variables encoded using binary variables
- Designed such that the energy under this encoding is pairwise submodular
- The solution obtained by solving st-mincut and inverting the encoding
Mulit-label energy with linear pairwise potentials

\[
\psi_i(z_i) \quad \psi_{ij}(z_i, z_j) = K|z_i - z_j|
\]

Encoding

\[
\begin{align*}
    z_i &= 0 \quad \iff \quad \{x_i^0 = 0, x_i^1 = 1, x_i^2 = 1, x_i^3 = 1, \ldots, x_i^D = 1\} \\
    z_i &= 1 \quad \iff \quad \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 1, x_i^3 = 1, \ldots, x_i^D = 1\} \\
    z_i &= 2 \quad \iff \quad \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 1, \ldots, x_i^D = 1\} \\
    \ldots \\
    z_i &= D \quad \iff \quad \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 0, \ldots, x_i^D = 0\}
\end{align*}
\]

Ishikawa PAMI03
Mulit-label energy with linear pairwise potentials

Data term
\[ \psi_i(z_i) \]

Smoothness term
\[ \psi_{ij}(z_i, z_j) = K |z_i - z_j| \]

Encoding
\[ z_i = 0 \iff \{ x_i^0 = 0, x_i^1 = 1, x_i^2 = 1, x_i^3 = 1, \ldots, x_i^D = 1 \} \]
\[ z_i = 1 \iff \{ x_i^0 = 0, x_i^1 = 0, x_i^2 = 1, x_i^3 = 1, \ldots, x_i^D = 1 \} \]
\[ z_i = 2 \iff \{ x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 1, \ldots, x_i^D = 1 \} \]
.. 
\[ z_i = D \iff \{ x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 0, \ldots, x_i^D = 0 \} \]

Ishikawa PAMI03
Multilabel energy with convex pairwise potentials

Data term

Smoothness term

\[
\psi_{ik}(z_i, z_k) = f(z_i - z_k)
\]

\[
\psi_i(z_i) = \left\{ \begin{array}{ll}
    c_{ik}^{d} \max(z_i - z_k + d, 0) \\
    + c_{ki}^{d} \max(z_k - z_i + d, 0)
\end{array} \right.
\]

where

\[
c_{ik}^{d} = \frac{f(d + 1) - 2f(d) + f(d - 1)}{2}
\]

Ishikawa PAMI03
Higher order minimization with GraphCut

\[ \psi_c(x_c) = \min_{z_c} \psi^p_c(x_c, z_c) \]

Higher order term Pairwise term

Flowchart:
- Encoding
- Transform into submodular E
- Invert Encoding
- Graph Cut
- Obtain solution
- Energy
- Solution
Higher order minimization with GraphCut

\[ \psi_c(x_c) = \min_{z_c} \psi^p_c(x_c, z_c) \]

Higher order term \hspace{2cm} Pairwise term

Example:

\[ \psi(x_1, x_2, x_3) = -x_1 x_2 x_3 \]
Higher order minimization with GraphCut

\[ \psi_c(x_c) = \min_{z_c} \psi^p_c(x_c, z_c) \]

**Higher order term**  **Pairwise term**

Example:

\[ \psi(x_1, x_2, x_3) = -x_1 x_2 x_3 = \min_z z(2 - x_1 - x_2 - x_3) \]

<table>
<thead>
<tr>
<th></th>
<th>(-x_1 x_2 x_3)</th>
<th>(\min_z z(2 - x_1 - x_2 - x_3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(\min_z 2z = 0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>(\min_z z = 0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>(\min_z 0 = 0)</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>(\min_z (-z) = -1)</td>
</tr>
</tbody>
</table>

Kolmogorov ECCV06, Ramalingam et al. DAM12
What if no encoding leads to pairwise submodular problem?
Move making algorithms

- Original problem decomposed into a series of subproblems solvable with graph cut
- In each subproblem we find the optimal move from the current solution in a restricted search space

Boykov et al., PAMI01
**Move making algorithms**

**αβ-swap**
- Each variable taking label $\alpha$ or $\beta$ can change its label to $\alpha$ or $\beta$
- Move space defined by the transformation function

### Transformation function

$$T_{\alpha\beta}(x_i, t_i) = \begin{cases} 
\alpha & \text{if } x_i \in \{\alpha, \beta\} \text{ and } t_i = 0 \\
\beta & \text{if } x_i \in \{\alpha, \beta\} \text{ and } t_i = 1 
\end{cases}$$
Move making algorithms

α-expansion
- Each variable may keep the old label or change to α
- Move space defined by the transformation function

Transformation function

\[ T_\alpha(x_i, t_i) = \begin{cases} \alpha & \text{if } t_i = 0 \\ x_i & \text{if } t_i = 1 \end{cases} \]
Move making algorithms

Sufficient condition for submodularity of each move:

\[ \forall l_a, l_b \in \mathcal{L} \]

\[ \psi^p(l_a, l_a) = 0 \]

\[ \psi^p(l_a, l_b) = \psi^p(l_b, l_a) \geq 0 \]

**\( \alpha \beta \)-swap**

**semi-metricity**

\[ \forall l_a, l_b \in \mathcal{L} \]

\[ \psi^p(l_a, l_a) = 0 \]

\[ \psi^p(l_a, l_b) = \psi^p(l_b, l_a) \geq 0 \]

**\( \alpha \)-expansion**

**metricity**

\[ \forall l_a, l_b \in \mathcal{L} \]

\[ \psi^p(l_a, l_a) = 0 \]

\[ \psi^p(l_a, l_b) = \psi^p(l_b, l_a) \geq 0 \]

\[ \psi^p(l_a, l_b) + \psi^p(l_b, l_c) \geq \psi^p(l_a, l_c) \]
Semantic Segmentation

\[ E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) \]

Data term

Smoothness term

Discriminatively trained classifier

Smoothness term

\[ \psi_{ij}(x_i, x_j) = K_{ij} \delta(x_i \neq x_j) \]

\[ K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2) \]
Semantic Segmentation

Original Image

Initial solution

grass
Semantic Segmentation

Original Image

Initial solution

Building expansion
Semantic Segmentation

Original Image

Initial solution

Building expansion

Sky expansion
Semantic Segmentation

Original Image

Initial solution

Building expansion

Sky expansion

Tree expansion
Semantic Segmentation

Original Image

Initial solution

Building expansion

Sky expansion

Tree expansion

Aeroplane expansion
Non-submodular energy minimization

What can we do?
Non-submodular energy minimization

What can we do?

Relax!
Non-submodular energy minimization

QPBO

• Each original variable is encoded using two binary variables $x_i$ and $\overline{x}_i$ s.t. $x_i = 1 - \overline{x}_i$

• Energy transformed into a submodular over $x_i$ and $\overline{x}_i$
Non-submodular energy minimization

QPBO

• Each original variable is encoded using two binary variables $x_i$ and $\bar{x}_i$ s.t. $x_i = 1 - \bar{x}_i$

• Energy transformed into a submodular over $x_i$ and $\bar{x}_i$

$$E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) = \sum_{i \in \mathcal{V}} (g^1_i x_i + g^0_i (1 - x_i))$$

$$+ \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} (g_{ij}^{00} (1 - x_i)(1 - x_j) + g_{ij}^{01} (1 - x_i)x_j + g_{ij}^{10} x_i(1 - x_j) + g_{ij}^{11} x_i x_j)$$

$$= \sum_{i \in \mathcal{V}} \left(\frac{g^1_i}{2} (x_i + (1 - \bar{x}_i)) + \frac{g^0_i}{2} (\bar{x}_i + (1 - x_i))\right)$$

$$+ \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \left(\frac{g_{ij}^{00}}{2} (\bar{x}_i (1 - x_j) + (1 - x_i)\bar{x}_j) + \frac{g_{ij}^{01}}{2} ((1 - x_i)x_j + \bar{x}_i (1 - \bar{x}_j))\right)$$

$$+ \frac{g_{ij}^{11}}{2} (x_i (1 - \bar{x}_j) + (1 - \bar{x}_i)x_j) + \frac{g_{ij}^{10}}{2} (x_i (1 - x_j) + (1 - x_i)\bar{x}_j)$$
Non-submodular energy minimization

QPBO

- Each original variable is encoded using two binary variables \( x_i \) and \( \overline{x_i} \) s.t. \( x_i = 1 - \overline{x_i} \)
- Energy transformed into a submodular over \( x_i \) and \( \overline{x_i} \)
- Solved by dropping the constraint \( x_i = 1 - \overline{x_i} \)
Non-submodular energy minimization

QPBO

• Each original variable is encoded using two binary variables $x_i$ and $\overline{x}_i$ s.t. $x_i = 1 - \overline{x}_i$

• Energy transformed into a submodular over $x_i$ and $\overline{x}_i$

• Solved by dropping the constraint $x_i = 1 - \overline{x}_i$

• All variables satisfying the constraint guaranteed to be part of globally optimal solution
Non-submodular energy minimization

QPBO

• Each original variable is encoded using two binary variables $x_i$ and $\bar{x}_i$ s.t. $x_i = 1 - \bar{x}_i$
• Energy transformed into a submodular over $x_i$ and $\bar{x}_i$
• Solved by dropping the constraint $x_i = 1 - \bar{x}_i$
• All variables satisfying the constraint guaranteed to be part of globally optimal solution
• Remaining variables assigned by iteratively estimated per node (ICM), or by keeping old labels for move algorithms
Other Structural Properties
Solvable with Graph-Cut

- Kohli et al. 07, 08 – label consistency over large cliques (super-pixels)
- Woodford et al. 08 – planarity constraint
- Vicente et al. 08 – connectivity constraint
- Woodford et al. 09 – marginal probability
- Nowozin & Lampert 09 – connectivity constraint
- Ladický et al. 09 – consistency over hierarchies (associative potentials)
- Delong et al. 10 – label occurrence costs
- Ladický et al. 10 – consistency between domains (semantic + depth)
- Ladický et al. 10 – detectors in CRF
- Ladický et al. 10 – co-occurrence potentials
- Savinov et al. 15 – ray potentials (semantic 3D visibility)
Schedule

• Introduction
  • Discrete MRF Optimization using Graph Cuts
  • Classifiers for Semantic 3D Modelling

• Higher Order MRFs with Ray Potentials
  • Discrete Formulation
  • Continuous Relaxation
Semantic classifier

Input image

Multiple features
- SIFT
- Colour
- Texton
- SfM

Clustering

Feature maps
- Colours ↔ Cluster

Shape filter
\[
\begin{pmatrix}
\text{texton} \\
\text{colour} \\
\text{location} \\
\text{HOG}
\end{pmatrix}, 
\begin{pmatrix}
\text{feature type } f \\
\text{cluster } t \\
\text{rectangle } r
\end{pmatrix}
\]

\[v(i, f, t, r) = a\]
\[v(i, f, t, r) = 0\]
\[v(i, f, t, r) = a/2\]

Shotton et al. ECCV06, Ladický et al. ICCV09
Data-driven Depth Estimation

- No common structure of the scene
- Ground plane not always visible
- Large variation of viewpoints and of objects in the scene
- Both *things* and *stuff* in the scene
Data-driven Depth Estimation

Desired properties:
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier

Super-pixels not necessarily planar
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier
2. Translation invariant

\[
H_d(x) := H_d(W_{w,h}(I, x))
\]

Classifier response for \(x\) and at a depth \(d\)

Window \(w \times h\) around the point \(x \in I\)
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier
2. Translation invariant
3. Depth transforms with inverse scaling

\[ H_d(W^{w,h}(I, x)) = H_{d/\alpha}(W^{w,h}(\alpha I, \alpha x)) \]
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier
2. Translation invariant
3. Depth transforms with inverse scaling

\[ H_d(W_w^h(I, x)) = H_{d/\alpha}(W_{w}^{\alpha I, \alpha x}) \]

Sufficient to train a binary classifier predicting a single \( d_C \)
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier
2. Translation invariant
3. Depth transforms with inverse scaling

\[ H_d(W^{w,h}(I, x)) = H_{d/\alpha}(W^{w,h}(\alpha I, \alpha x)) \]

Sufficient to train a binary classifier predicting a single \( d_C \)

For other depths \( d \):

\[ H_d(W^{w,h}(I, x)) = H_{d/d_C}(W^{w,h}(\frac{d}{d_C} I, \frac{d}{d_C} x)) \]
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier
2. Translation invariant
3. Depth transforms with inverse scaling

\[ H_d(W^{w,h}(I, x)) = H_{d/\alpha}(W^{w,h}(\alpha I, \alpha x)) \]
Data-driven Depth Estimation

Desired properties:

1. Pixel-wise classifier
2. Translation invariant
3. Depth transforms with inverse scaling

Generalized to multiple semantic classes

\[ H_{(l,d)}(W^{w,h}(I, x)) = H_{(l,d_c)}(W^{w,h}(\frac{d}{d_c} I, \frac{d}{d_c} x)) \]

semantic label
Training the classifier

1. Image pyramid is built
Training the classifier

1. Image pyramid is built
2. Training data randomly sampled
Training the classifier

1. Image pyramid is built
2. Training data randomly sampled
3. Samples of each class at $d_C$ used as positives
Training the classifier

1. Image pyramid is built
2. Training data randomly sampled
3. Samples of each class at $d_C$ used as positives
4. Samples of other classes or at $d \neq d_C$ used as negatives
Training the classifier

1. Image pyramid is built
2. Training data randomly sampled
3. Samples of each class at $d_C$ used as positives
4. Samples of other classes or at $d \neq d_C$ used as negatives
5. Multi-class classifier trained
Classifying the patch

Dense Features: SIFT, LBP, Self Similarity, Texton
Classifying the patch

Dense Features  SIFT, LBP, Self Similarity, Texton
Representation  Soft BOW representations in the set of random rectangles
Classifying the patch

Dense Features  
SIFT, LBP, Self Similarity, Texton

Representation  
Soft BOW representations in the set of random rectangles

Classifier  
AdaBoost
Experiments

KITTI dataset

- 30 training & 30 test images (1382 x 512)
- 12 semantic labels, depth 2-50m (except sky)
- ratio of neighbouring depths $d_{i+1} / d_i = 1.25$

NYU2 dataset

- 725 training & 724 test images (640 x 480)
- 40 semantic labels, depth in the range 1-10 m
- ratio of neighbouring depths $d_{i+1} / d_i = 1.25$
KITTI results
NYU2 results
Surface Normal Estimation

Not explored much in the literature... so how to approach it?
Surface Normal Estimation

Not explored much in the literature… so how to approach it?

Pixels or Super-pixels?
Pixel-based Classifiers

• Context-based (context pixels or rectangles) feature representations

[Shotton06, Shotton08]
Pixel-based Classifiers

- Context-based (context pixels or rectangles) feature representations [Shotton06, Shotton08]
- Classifier typically noisy and does not follow object boundaries
Segment-based Classifiers

- Based on feature statistics in segments

Input image

Feature representation
Segment-based Classifiers

- Based on feature statistics in segments
- Segments expected to be label-consistent
Segment-based Classifiers

- Based on feature statistics in segments
- Segments expected to be label-consistent
- One particular segmentation has to be chosen
Joint Regularization

- Existing optimization methods (Ladicky09) designed for discrete labels
Existing optimization methods (Ladicky09) designed for discrete labels

Not obvious how to generalize for continuous problems
Joint Regularization

- Existing optimization methods (Ladicky09) designed for discrete labels
- Not obvious how to generalize for continuous problems
- Maybe we can directly learn joint classifier
Joint Learning

How to convert segment representation into pixel representation?
Joint Learning

Input image

Segment representation

How to convert segment representation into pixel representation?

• Representation of a pixel the same as of the segment it belongs to
Joint Learning

How to convert segment representation into pixel representation?

- Representation of a pixel the same as of the segment it belongs to
- Equivalent to weighted segment based approach
How to convert segment representation into pixel representation?

- Representation of a pixel the same as of the segment it belongs to
- Equivalent to weighted segment based approach
- Concatenation to combine pixel and multiple segment representations
Joint Learning

To simplify regression problem

- Normals clustered using K-means clustering
- Each represented as weighted sums of cluster centres using local coding
Joint Learning

To simplify regression problem

- Normals clustered using K-means clustering
- Each represented as weighted sums of cluster centres using local coding
- Learning formulated as a regression into local coding coordinates
Pipeline of our Method
RMRC Challenge Results

<table>
<thead>
<tr>
<th>Input image</th>
<th>err = 40.366</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input image</td>
<td>err = 32.446</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 33.636</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 35.109</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 35.429</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 37.066</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 38.043</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 35.109</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 28.379</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 35.429</td>
</tr>
</tbody>
</table>
RMRC Challenge Results

<table>
<thead>
<tr>
<th>Input image</th>
<th>err = 37.688</th>
<th>Input image</th>
<th>err = 40.784</th>
<th>Input image</th>
<th>err = 51.897</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input image</td>
<td>err = 28.216</td>
<td>Input image</td>
<td>err = 32.034</td>
<td>Input image</td>
<td>err = 68.038</td>
</tr>
<tr>
<td>Input image</td>
<td>err = 33.174</td>
<td>Input image</td>
<td>err = 41.131</td>
<td>Input image</td>
<td>err = 38.873</td>
</tr>
</tbody>
</table>
Schedule

• Introduction
  • Discrete MRF Optimization using Graph Cuts
  • Classifiers for Semantic 3D Modelling

• Higher Order MRFs with Ray Potentials
  • Discrete Formulation
  • Continuous Relaxation
Semantic 3D Reconstruction

Input images ➔ Semantic estimates ➔ Depth estimates ➔ Semantic 3D model
Semantic 3D Reconstruction

Pixel predictions - prediction of the first occupied voxel along the ray

Predictions of the semantic label of the first occupied voxel

Predictions of the depth of the first occupied voxel
Semantic 3D Reconstruction

Volumetric formulation

$$E(x) = \sum_{r \in \mathcal{R}} \psi_r(x^r) + \sum_{(i,j) \in \mathcal{E}} \psi_p(x_i, x_j)$$
Semantic 3D Reconstruction

Volumetric formulation

\[ E(x) = \sum_{r \in R} \psi_r(x^r) + \sum_{(i,j) \in E} \psi_p(x_i, x_j) \]

Ray potentials \hspace{1cm} \text{Pairwise regularizer}
Semantic 3D Reconstruction

Volumetric formulation

\[ E(x) = \sum_{r \in \mathcal{R}} \psi_r(x^r) + \sum_{(i,j) \in \mathcal{E}} \psi_p(x_i, x_j) \]

Ray potentials \hspace{1cm} Pairwise regularizer

Ray potentials typically approximated by unary potentials

- voxels behind the depth estimate should be occupied
- voxels just in front of the depth estimate should be free space

( Zach 3DPVT08, Häne CVPR13, Kundu ECCV14, ..)
Semantic 3D Reconstruction

Volumetric formulation

\[ E(x) = \sum_{r \in R} \psi_r(x^r) + \sum_{(i,j) \in E} \psi_p(x_i, x_j) \]

Ray potentials\hspace{2cm}Pairwise regularizer

We try to solve the right problem!
Semantic 3D Reconstruction

Volumetric formulation

\[
E(x) = \sum_{r \in R} \psi_r(x^r) + \sum_{(i,j) \in E} \psi_p(x_i, x_j)
\]

Ray potentials \hspace{1cm} Pairwise regularizer

Cost based on the first occupied voxel along the ray

\[
\psi_r(x^r) = \phi_r(K^r, x^r_{K^r})
\]

depth \hspace{1cm} label

\[
K^r = \begin{cases} 
\min(i | x^r_i \neq l_f) & \text{if } \exists x^r_i \neq l_f \\
N_r & \text{otherwise}
\end{cases}
\]
Two-label problem

Discrete formulation using QPBO relaxation

\[ x_0 \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \]

\[ \overline{x}_0 \quad \overline{x}_1 \quad \overline{x}_2 \quad \overline{x}_3 \quad \overline{x}_4 \quad \overline{x}_5 \quad \overline{x}_6 \]
Two-label problem

Discrete formulation using QPBO relaxation

\[
\begin{align*}
\text{x}_0 & \quad \text{x}_1 & \quad \text{x}_2 & \quad \text{x}_3 & \quad \text{x}_4 & \quad \text{x}_5 & \quad \text{x}_6 \\
\overline{\text{x}}_0 & \quad \overline{\text{x}}_1 & \quad \overline{\text{x}}_2 & \quad \overline{\text{x}}_3 & \quad \overline{\text{x}}_4 & \quad \overline{\text{x}}_5 & \quad \overline{\text{x}}_6
\end{align*}
\]

Our goal is to find :

\[
\psi_r(\text{x}^r) = \min_{\text{z}} \psi_q(\text{x}^r, \overline{\text{x}}^r, \text{z})
\]
Two-label problem

Discrete formulation using QPBO relaxation

Our goal is to find:

\[ \psi_r(x^r) = \min_{z} \psi_q(x^r, \bar{x}^r, z) \]

such that \( \psi_q(x^r, \bar{x}^r, z) \) is:

1) A pairwise function
2) Number of edges grows linearly with the length for a ray
3) Symmetric to inherit QPBO properties
Two-label problem

To find $\psi_q(x^r, \overline{x}^r, z)$ we do these steps:

1) Polynomial representation of the ray potential
2) Transformation into submodular function over $x$ and $\overline{x}$
3) Pairwise construction using auxiliary variables $z$
4) Merging variables (Ramalingam12) for linear complexity
5) Symmetrization of the graph
Polynomial representation of the ray potential

Two-label ray potential takes the form:

\[ \psi_r(x^r) := \begin{cases} 
\phi_r(\min(i | x_i^r = 0)) & \text{if } \exists x_i^r = 0 \\
\phi_r(N_r) & \text{otherwise} 
\end{cases} \]

where \( x_i = 0 \) for occupied voxel \( x_i = 1 \) for free-space
Polynomial representation of the ray potential

Two-label ray potential takes the form:

$$\psi_r(x^r) := \begin{cases} 
\phi_r(\min(i | x^r_i = 0)) & \text{if } \exists x^r_i = 0 \\
\phi_r(N_r) & \text{otherwise}
\end{cases}$$

where \( x_i = 0 \) for occupied voxel \( x_i = 1 \) for free-space

We want to transform the potential into:

$$\psi_r(x^r) = k^r + \sum_{i=0}^{N_r-1} c_i^r \prod_{j=0}^i x^r_j$$
Polynomial representation of the ray potential

Two-label ray potential takes the form:

$$\psi_r(x^r) := \begin{cases} 
\phi_r(\min(i|x_i^r = 0)) & \text{if } \exists x_i^r = 0 \\
\phi_r(N_r) & \text{otherwise}
\end{cases}$$

where $x_i = 0$ for occupied voxel $\quad x_i = 1$ for free-space

We want to transform the potential into:

$$\psi_r(x^r) = k^r + \sum_{i=0}^{N_r-1} c_i^r \prod_{j=0}^{i} x_j^r$$

Plugging it in:

$$\phi_r(K) = k^r + \sum_{i=0}^{K-1} c_i^r \quad \text{thus} \quad k^r = \phi_r(0)$$

$$c_i^r = \phi_r(i + 1) - \phi_r(i)$$
Transformation into a submodular function

\[ c_i^R \prod_{j=0}^{i} x_j^R \quad \text{submodular for} \quad c_i^R \leq 0 \]
Transformation into a submodular function

\[ c_i^r \prod_{j=0}^{i} x_j^r \quad \text{submodular for} \quad c_i^r \leq 0 \]

For \( c_i^r > 0 \):

\[ c_i^r \prod_{j=0}^{i} x_j^r = c_i^r (1 - x_i^r) \prod_{j=0}^{i-1} x_j^r = -c_i^r x_i^r \prod_{j=0}^{i-1} x_j^r + c_i^r \prod_{j=0}^{i-1} x_j^r \]
Transformation into a submodular function

\[ c_i^r \prod_{j=0}^{i} x_j^r \text{ - submodular for } c_i^r \leq 0 \]

For \( c_i^r > 0 \):

\[ c_i^r \prod_{j=0}^{i} x_j^r = c_i^r (1 - x_i^r) \prod_{j=0}^{i-1} x_j^r = -c_i^r x_i^r \prod_{j=0}^{i-1} x_j^r + c_i^r \prod_{j=0}^{i-1} x_j^r \]

Starting from the last term, we can iteratively transform:

\[ \sum_{i=0}^{N_r-1} c_i^r \prod_{j=0}^{i} x_j^r = \sum_{i=0}^{N_r-1} \left( -a_i^r \prod_{j=0}^{i} x_j^r - b_i^r x_i^r \prod_{j=0}^{i-1} x_j^r \right) \]
Pairwise graph construction

Standard graph constructions (Freedman CVPR05) for negative products:

\[-a_i^r \prod_{j=0}^{i} x_j^r = a_i^r \min_{z_i} \left( -z_i + \sum_{j=0}^{i} z_i (1 - x_j^r) \right)\]
Pairwise graph construction

Standard graph constructions (Freedman CVPR05) for negative products:

\[-b_i^r \overline{x}_i^r \prod_{j=0}^{i-1} x_j^r = b_i^r \min_{z_i'} \left( -z_i' + z_i'(1 - \overline{x}_i^r) + \sum_{j=0}^{i-1} z_i'(1 - x_j^r) \right)\]
Pairwise graph construction

Standard graph constructions (Freedman CVPR05) for negative products:

\[-a_i^r \prod_{j=0}^{i} x_j^r = a_i^r \min_{z_i} \left( -z_i + \sum_{j=0}^{i} z_i (1 - x_j^r) \right)\]

\[-b_i^r \prod_{j=0}^{i-1} x_j^r = b_i^r \min_{z_i'} \left( -z_i' + z_i' (1 - \overline{x}_i^r) + \sum_{j=0}^{i-1} z_i' (1 - x_j^r) \right)\]

Leads to a quadratic growth of the number of edges!
Pairwise graph construction

Non-standard graph constructions for negative products:

\[-a_i^r \prod_{j=0}^{i} x_j^r = a_i^r \min_{z_i} \left( -z_i^i + z_i^i (1 - x_i^r) + \sum_{j=0}^{i-1} (z_j^j (1 - z_j^j) + z_j^j (1 - x_j^r)) \right)\]

Optimal \( z \):

\[z_j^i = \prod_{k=0}^{j} x_k^r\]
Pairwise graph construction

Non-standard graph constructions for negative products:

$$-b_i^r \overline{x_i^r} \prod_{j=0}^{i-1} x_j^r = b_i^r \min_{z_i'} \left( -z_i'^r + z_i'^r (1 - \overline{x_i^r}) + \sum_{j=0}^{i-1} (z_{j+1}'^r (1 - z_i'^r) + z_j'^r (1 - x_j^r)) \right)$$

Optimal $z$:

$$z_i'^r = \overline{x_i^r} \prod_{k=0}^{i-1} x_k^r$$
$$z_j'^r = \prod_{k=0}^{j} x_k^r$$
Merging theorem

(Ramalingam12)

\[
\psi(x) = \min_z \psi_q(x, z_i, \ldots z_j)
\]

\[\forall x \quad z_i = z_j \quad \text{(optimal)}\]
Merging theorem

(Ramalingam12)

If

$$\psi(x) = \min_z \psi_q(x, z_i, ..z_j)$$

$$\forall x \ z_i = z_j \quad \text{(optimal)}$$

then

$$\psi(x) = \min_z \psi_q(x, z_i, ..z_i)$$
Pairwise graph construction

Non-standard graph constructions for negative products:

\[-a_i^r \prod_{j=0}^{i} x_j^r = a_i^r \min_{z_i} \left(-z_i^i + z_i^i (1 - x_i^r) + \sum_{j=0}^{i-1} (z_j^i + z_j^i (1 - x_j^r))\right)\]

Optimal \(z\):

\[z_j^i = \prod_{k=0}^{j} x_k^r\]

\[-b_i^r \overline{x_i^r} \prod_{j=0}^{i-1} x_j^r = b_i^r \min_{z_i'} \left(-z_i'^i + z_i'^i (1 - \overline{x_i^r}) + \sum_{j=0}^{i-1} (z_{j+1}^i + z_{j+1}^i (1 - x_{j+1}^r))\right)\]

Optimal \(z\):

\[z_i'^i = \overline{x_i^r} \prod_{k=0}^{i-1} x_k^r\]

\[z_j'^i = \prod_{k=0}^{j} x_k^r\]

\[\forall j, k \in \{0, ..., i\} : z_j^i = z_k^i\]

\[\forall j, k \in \{0, ..., i - 1\} : z_j'^i = z_k^i\]
Pairwise graph construction
Pairwise graph construction

$$\psi_r(x^r) = \min_{z,z'} \left( \sum_{i=0}^{N_r-1} \left( -a_i^r z_i - b_i^r z_i' + f_i^r(1 - z_i)x_i^r + b_i^r(1 - z_i)x_i^r \right) + \sum_{i=1}^{N_r-2} \left( f_{i+1}^r(1 - z_{i+1})z_i + b_i^r(1 - z_{i+1})z_i \right) \right)$$

$$f_i^r = \sum_{j=0}^{N_r-1} a_j^r + \sum_{j=i+1}^{N_r-1} b_j^r$$
Symmetrization of the graph

\[ \psi_r(x^r) = \frac{1}{2} \left( \min_z \psi^r(x, \overline{x}, z) + \min_z \psi^r(1 - \overline{x}, 1 - x, 1 - z) \right) \]
Multi-label problem

• Standard alpha-expansion
• Multi-label ray potential projects into 2-label ray potential
• Variables not labelled by QPBO labelled using ICM
Implementation details

\[ \phi_r(i, l) = \left( \lambda_{sem} C(l) + \lambda_{dep} C(d(i)) \right) d(i)^2 \]

Semantic cost \quad Depth cost

- Semantic classifier [Ladický ICCV09]
- Multi-view stereo depth matches using zero-mean NCC

For the top n matches:

\[ C(d(i)) = \begin{cases} w_n (-1 + \frac{|d(i) - d_{top}^n|}{\Delta}) & \text{if } |d(i) - d_{top}^n| \leq \Delta \\ 0 & \text{otherwise.} \end{cases} \]
Results

Input                    Depth              Semantics                3D model

[Images of input, depth, semantics, and 3D models for different scenes]
Results

Input                  Depth              Semantics                  3D model
Results

South Building Data Set
Results

Castle Data Set
Conclusions

- Volumetric optimization over rays is feasible
- Solvable using QPBO relaxation and suitable graph
- Results do not suffer from artifacts of ray approximations
  - objects are not fattened
  - holes are not closed
Continuous Formulation

\[ E(x) = \sum_{r \in R} \psi_r(x^r) + \sum_{(i, j) \in E} \psi_p(x_i, x_j) \]

Continuous approach possible?
Continuous Formulation

\[
\psi_r(x_r) = \left( \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^\ell \left( \min_{j \leq i-1} x_{srj}^f \right) x_{sri}^\ell \right) + c_r^f \min_{j \leq N_r} x_{srj}^f
\]
Continuous Formulation

$$\psi_r(x_r) = \left( \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c^\ell_{ri} \left( \min_{j \leq i-1} x^f_{srj} \right) x^\ell_{sri} \right) + c^f_r \min_{j \leq N_r} x^f_{srj}$$

The last term can be dropped by:

$$c^\ell_{ri} \leftarrow c^\ell_{ri} - c^f_r \quad c^f_r \leftarrow 0$$
Continuous Formulation

\[ \psi_r(x_r) = \sum_{\ell \in \mathcal{L}\setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^\ell \left( \min_{j \leq i-1} x_{s_rj}^f \right) x_{s_{ri}}^\ell \]

Introducing visibility variables: \[ y_{ri}^\ell = \min(y_{r,i-1}^f, x_{s_{ri}}^\ell) \]
Continuous Formulation

\[
\psi_r(x_r) = \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^\ell \left( \min_{j \leq i-1} x_{srj}^f \right) x_{sri}^\ell
\]

Introducing visibility variables: \( y_{ri}^\ell = \min(y_{r,i-1}^f, x_{sri}^\ell) \)

\[
\psi_r(x_r) = \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_{i}^\ell y_{i}^\ell \quad \text{where} \quad c_{i}^f = 0
\]
Continuous Formulation

\[ \psi_r(x_r) = \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^\ell \left( \min_{j \leq i-1} x_{srj}^f \right) x_{sri}^\ell \]

Introducing visibility variables: \[ y_{ri}^\ell = \min(y_{r,i-1}^f, x_{sri}^\ell) \]

\[ \psi_r(x_r) = \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_{i}^\ell y_{i}^\ell \]

where \[ c_{i}^f = 0 \]

convex for \[ c_{i}^\ell \leq 0 \]
Continuous Formulation

\[ \psi_r(x_r) = \sum_{\ell \in \mathcal{L} \backslash \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \left( \min_{j \leq i-1} x_{srj}^f \right) x_{sri}^\ell \]

Introducing visibility variables:

\[ y_{ri}^\ell = \min(y_{r,i-1}^f, x_{sri}^\ell) \]

\[ \psi_r(x_r) = \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_i^\ell y_i^\ell \]

convex for \( c_i^\ell \leq 0 \)

\[ y_i^\ell \leq y_{i-1}^f \]
\[ y_i^\ell \leq x_{si}^\ell \]
\[ y_i^\ell \geq 0 \]

\[ \forall \ell \in \mathcal{L}, \forall i \]
Continuous Formulation

Can we make $c_i^\ell \leq 0$?
Continuous Formulation

Can we make $c_i^\ell \leq 0$? Yes!
Continuous Formulation

Can we make $c_i^\ell \leq 0$?

First, we notice:

$$y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell = 0$$
Can we make $c_i^\ell \leq 0$ ?

First, we notice:

$$y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell = 0$$

The cost function does not change by adding:

$$\left( \max_{\ell' \in \mathcal{L}} c_i^{\ell'} \right) \left( y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell \right) = 0$$
Continuous Formulation

Can we make \( c_i^f \leq 0 \)?

First, we notice:

\[
y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell = 0
\]

The cost function does not change by adding:

\[
\left( \max_{\ell' \in \mathcal{L}} c_i^{\ell'} \right) \left( y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell \right) = 0
\]

for all \( i \in \{N, \ldots, 0\} \)

\[
c_{i-1}^f \leftarrow c_{i-1}^f + \max_{\ell' \in \mathcal{L}} c_i^{\ell'}
\]

\[
c_i^\ell \leftarrow c_i^\ell - \max_{\ell' \in \mathcal{L}} c_i^{\ell'}
\]

\( \forall \ell \in \mathcal{L} \)
Continuous Formulation

Can we make $c_i^\ell \leq 0$?

First, we notice:

$$y_i^{f} - \sum_{\ell \in \mathcal{L}} y_i^\ell = 0$$

The cost function does not change by adding:

$$\left( \max_{\ell' \in \mathcal{L}} c_i^{\ell'} \right) \left( y_i^{f} - \sum_{\ell \in \mathcal{L}} y_i^\ell \right) = 0$$

for all $i \in \{N, \ldots, 0\}$

$$c_{i-1}^f \leftarrow c_{i-1}^f + \max_{\ell' \in \mathcal{L}} c_i^{\ell'}$$

$$c_i^\ell \leftarrow c_i^\ell - \max_{\ell' \in \mathcal{L}} c_i^{\ell'}$$

$\forall \ell \in \mathcal{L}$

$c_i^\ell \geq 0$
Integer Formulation

\[
\min \sum_{r \in R} \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c^\ell_i y^\ell_i + \psi_S(x)
\]

s.t. \( y^\ell_i \leq y^f_{i-1} \quad y^\ell_i \leq x^\ell_{s_i} \)

\[
\sum_{\ell \in \mathcal{L}} x^\ell_s = 1 \quad y^\ell_i \geq 0 \quad \forall \ell \in \mathcal{L}, \forall i
\]

\[
x^\ell_s \in \{0, 1\} \quad \forall s \in \Omega
\]
Convex relaxation

\[
\min \sum_{r \in \mathcal{R}} \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_{i}^{\ell} y_{i}^{\ell} + \psi_{S}(x)
\]

s.t. \( y_{i}^{\ell} \leq y_{i-1}^{f} \), \( y_{i}^{\ell} \leq x_{s_{i}}^{\ell} \)

\[
\sum_{\ell \in \mathcal{L}} x_{s}^{\ell} = 1, \quad y_{i}^{\ell} \geq 0, \quad \forall \ell \in \mathcal{L}, \forall i
\]

\[
0 \leq x_{s}^{\ell} \leq 1, \quad \forall s \in \Omega
\]
Convex relaxation

\[
\min \sum_{r \in \mathcal{R}} \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_{i}^{\ell} y_{i}^{\ell} + \psi_{S}(x)
\]

s.t. \[y_{i}^{\ell} \leq y_{i-1}^{\ell} \quad y_{i}^{\ell} \leq x_{s_{i}}^{\ell}\]
\[\sum_{\ell \in \mathcal{L}} x_{s}^{\ell} = 1 \quad y_{i}^{\ell} \geq 0 \quad \forall \ell \in \mathcal{L}, \forall i\]
\[0 \leq x_{s}^{\ell} \leq 1 \quad \forall s \in \Omega\]

Will it work ?
Convex relaxation

$$\min \sum_{r \in \mathcal{R}} \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_{i}^{\ell} y_{i}^{\ell} + \psi_{S}(x)$$

s.t. 
$$y_{i}^{\ell} \leq y_{i-1}^{\ell} \quad y_{i}^{\ell} \leq x_{s_{i}}^{\ell}$$
$$\sum_{\ell \in \mathcal{L}} x_{s_{i}}^{\ell} = 1 \quad y_{i}^{\ell} \geq 0$$
$$0 \leq x_{s}^{\ell} \leq 1$$

Will it work؟

Unfortunately not
\[(x^*, y^*) = \arg \min_{(x, y)} (-2y_0^o - 3y_1^o - 2y_2^o)\]

\text{s.t.} \quad y_i^o \leq y_{i-1}^f, \quad y_i^f \leq y_{i-1}^f, \\
\quad y_i^o \leq x_i^o, \quad y_i^f \leq 1 - x_i^o, \\
\quad x_i^o \in [0, 1], \quad \forall i.
Convex relaxation

\[(x^*, y^*) = \arg \min_{(x, y)} (-2y_0^o - 3y_1^o - 2y_2^o)\]

s.t. \[y_i^o \leq y_{i-1}^f, \quad y_i^f \leq y_{i-1}^f,\]

\[y_i^o \leq x_i^o, \quad y_i^f \leq 1 - x_i^o,\]

\[x_i^o \in [0, 1], \quad \forall i.\]

Desired solution

\[x_0^o = 0, \quad x_1^o = 1, \quad x_2^o = 0\]

\[y_0^o = 0, \quad y_1^o = 1, \quad y_2^o = 0\]

\[y_0^f = 1, \quad y_1^f = 0, \quad y_2^f = 0\]
Convex relaxation

\[(x^*, y^*) = \arg \min_{(x,y)} (-2y_0^0 - 3y_1^0 - 2y_2^0)\]

s.t. \[y_i^0 \leq y_{i-1}^f, \quad y_i^f \leq y_{i-1}^f,\]

\[y_i^0 \leq x_i^0, \quad y_i^f \leq 1 - x_i^0,\]

\[x_i^0 \in [0, 1], \quad \forall i.\]

Desired solution

\[x_0^0 = 0, \quad x_1^0 = 1, \quad x_2^0 = 0\]

\[y_0^0 = 0, \quad y_1^0 = 1, \quad y_2^0 = 0\]

\[y_0^f = 1, \quad y_1^f = 0, \quad y_2^f = 0\]

Global optimum

\[x_0^0 = x_1^0 = x_2^0 = 0.5\]

\[y_0^0 = y_1^0 = y_2^0 = 0.5\]

\[y_0^f = y_1^f = y_2^f = 0.5\]
Convex relaxation

\[(x^*, y^*) = \arg \min_{(x, y)} (-2y_0^o - 3y_1^o - 2y_2^o)\]

s.t. \[y_i^o \leq y_{i-1}^f, \quad y_i^f \leq y_{i-1}^f,\]
    \[y_i^o \leq x_i^o, \quad y_i^f \leq 1 - x_i^o,\]
    \[x_i^o \in [0, 1], \quad \forall i.\]

Problem solved, if cost is taken, when there is a visibility drop:

\[
\sum_{\ell \in \mathcal{L} \setminus \{f\}} y_i^\ell = y_{i-1}^f - y_i^f = y_{i-1}^f - \min(y_{i-1}^f, x_{si}^f)
\]

\[
\sum_{\ell \in \mathcal{L} \setminus \{f\}} y_i^\ell = \max(0, y_{i-1}^f - x_{si}^f)
\]
Non-convex Formulation

\[
\min \sum_{r \in \mathcal{R}} \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_i^\ell y_i^\ell + \psi_S(x)
\]

s.t. \( y_i^\ell \leq y_{i-1}^f \) \( y_i^\ell \leq x_{s_i}^\ell \)

\[
\sum_{\ell \in \mathcal{L}} x_{s_i}^\ell = 1 \quad y_i^\ell \geq 0 \quad \forall \ell \in \mathcal{L}, \forall i
\]

\[
0 \leq x_{s_i}^\ell \leq 1
\]

\[
\sum_{\ell \in \mathcal{L}\setminus\{f\}} y_i^\ell = \max(0, y_{i-1}^f - x_{s_i}^f)
\]
Non-convex Formulation

Solved using majorize-minimize strategy:
Non-convex Formulation

Solved using majorize-minimize strategy:

We replace constraint

\[ \sum_{\ell \in \mathcal{L} \setminus \{f\}} y_{i\ell} \leq \max\{0, y_{i-1}^f - x_{s_i}^f\} \]

by

\[ \sum_{\ell \in \mathcal{L} \setminus \{f\}} y_{i\ell} \leq g(x_{s_i}^f, y_{i-1}^f | x_{s_i}^{f,(n)}, y_{i-1}^{f,(n)}) \]
Non-convex Formulation

Solved using majorize-minimize strategy:

We replace constraint

$$\sum_{\ell \in \mathcal{L} \setminus \{f\}} y_i^\ell \leq \max\{0, y_{i-1}^f - x_{s_i}^f\}$$

by

$$\sum_{\ell \in \mathcal{L} \setminus \{f\}} y_i^\ell \leq g(x_{s_i}^f, y_{i-1}^f | x_{s_i}^{f,(n)}, y_{i-1}^{f,(n)})$$

where

$$g(x_{s_i}^f, y_{i-1}^f | x_{s_i}^{f,(n)}, y_{i-1}^{f,(n)}) = \begin{cases} 0 & \text{if } y_{i-1}^{f,(n)} \leq x_{s_i}^{f,(n)} \\ y_{i-1}^f - x_{s_i}^f & \text{if } y_{i-1}^{f,(n)} > x_{s_i}^{f,(n)} \end{cases}$$
Results on Middlebury dataset
## Results

<table>
<thead>
<tr>
<th>Sort By</th>
<th>Temple Full 312 views</th>
<th>Temple Ring 47 views</th>
<th>Temple Sparse 16 views</th>
<th>Dino Full 363 views</th>
<th>Dino Ring 48 views</th>
<th>Dino Sparse 16 views</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc [mm]</td>
<td>Comp [%]</td>
<td>Acc [mm]</td>
<td>Comp [%]</td>
<td>Acc [mm]</td>
<td>Comp [%]</td>
</tr>
<tr>
<td>Savinov</td>
<td>0.41</td>
<td>99.7</td>
<td>0.5</td>
<td>99.5</td>
<td>0.69</td>
<td>97.8</td>
</tr>
<tr>
<td>Furukawa 3</td>
<td>0.49</td>
<td>99.6</td>
<td>0.47</td>
<td>99.6</td>
<td>0.63</td>
<td>99.3</td>
</tr>
<tr>
<td>DCV</td>
<td>0.73</td>
<td>98.2</td>
<td>0.66</td>
<td>97.3</td>
<td>0.28</td>
<td>100</td>
</tr>
<tr>
<td>Galliani</td>
<td>0.39</td>
<td>99.2</td>
<td>0.48</td>
<td>99.1</td>
<td>0.53</td>
<td>97.0</td>
</tr>
<tr>
<td>ECCV2016_624</td>
<td>0.37</td>
<td>98.9</td>
<td>0.48</td>
<td>97.6</td>
<td>1.27</td>
<td>39.2</td>
</tr>
<tr>
<td>3DV2014_25</td>
<td>0.51</td>
<td>96.4</td>
<td>1.23</td>
<td>90.2</td>
<td>0.32</td>
<td>97.3</td>
</tr>
<tr>
<td>Furukawa 2</td>
<td>0.54</td>
<td>99.3</td>
<td>0.55</td>
<td>99.1</td>
<td>0.62</td>
<td>99.2</td>
</tr>
<tr>
<td>Schroers</td>
<td>0.57</td>
<td>99.1</td>
<td>0.64</td>
<td>96.4</td>
<td>2.12</td>
<td>62.9</td>
</tr>
<tr>
<td>Kostrikov</td>
<td>0.57</td>
<td>99.1</td>
<td>0.79</td>
<td>95.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yichao Li</td>
<td>0.46</td>
<td>96.4</td>
<td>0.56</td>
<td>89.6</td>
<td>0.4</td>
<td>94.9</td>
</tr>
<tr>
<td>Song</td>
<td>0.61</td>
<td>98.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khuboni</td>
<td>0.67</td>
<td>98.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhu</td>
<td>0.4</td>
<td>99.2</td>
<td>0.45</td>
<td>95.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results on Thin Structures

Data

TV Flux (high reg)

TV Flux (medium reg)

TV Flux (low reg)

Our method
Multi-class results

Input Data     Häne et al. CVPR13          Discrete result         Continuous result
Questions ?