Semantic 3D Modelling

Ľubor Ladický

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

Computer Vision and Geometry Lab

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

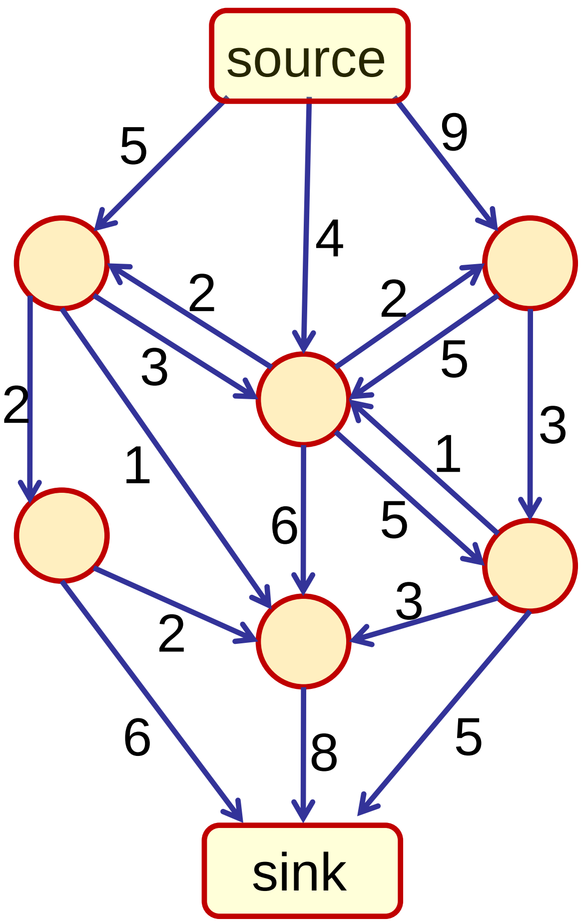
Schedule

- 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

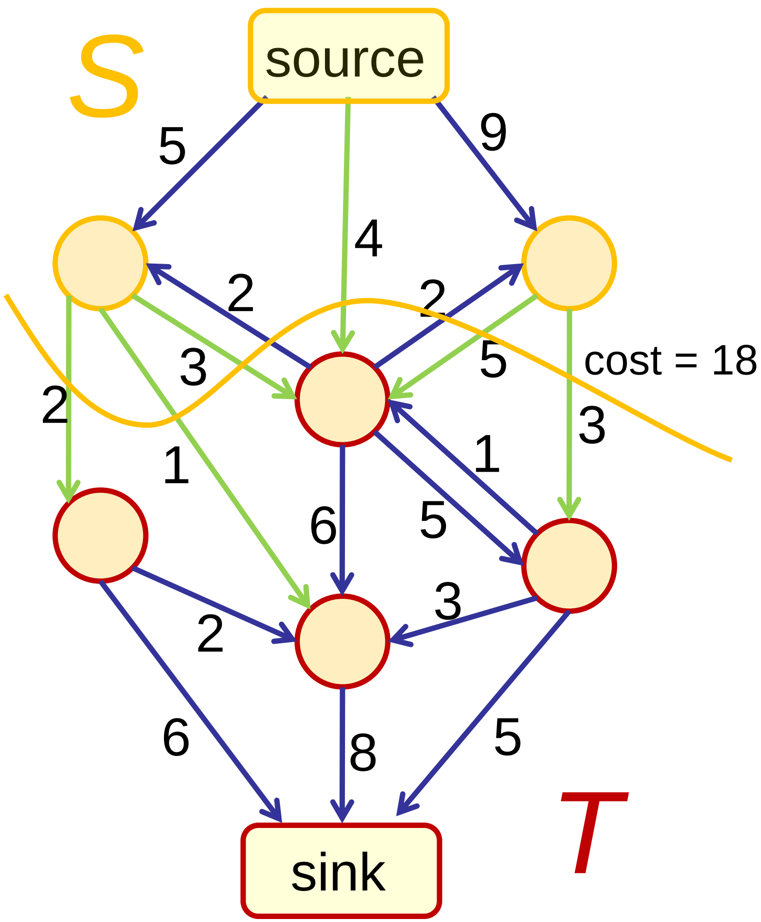
edge costs

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

source set

sink set

Graph-Cut (st-mincut)



Set formulation

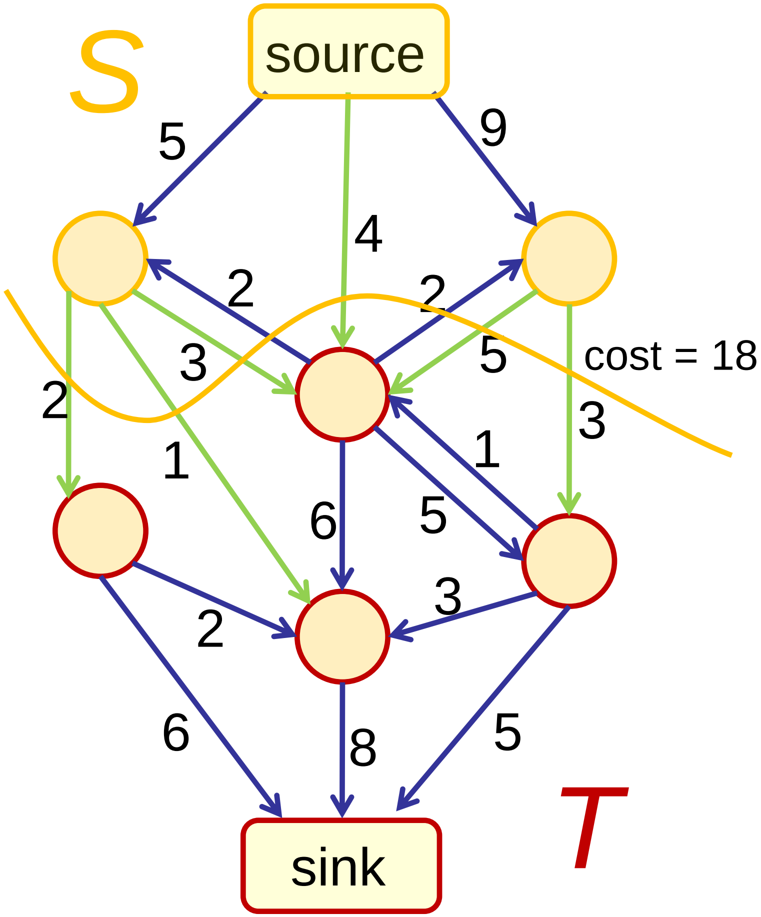
edge costs

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

source set

sink set

Graph-Cut (st-mincut)



Set formulation

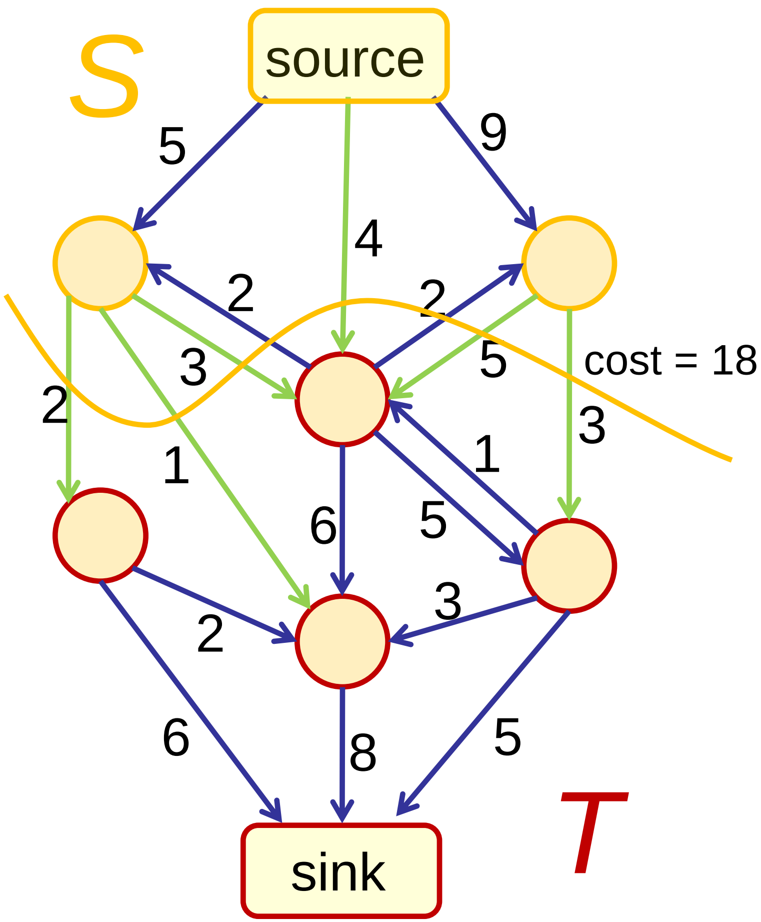
$$\min_{S,T} \sum_{i \in S, j \in T} c_{ij} \qquad s.t. \quad s \in S, \quad t \in T$$

Algebraic formulation

$$\min_{\mathbf{x}} \sum_{(i,j) \in E} c_{ij}(1 - x_i)x_j \qquad s.t. \begin{array}{l} x_s = 0 \\ x_t = 1 \end{array}$$

$$x_i = 0 \implies x_i \in S \qquad x_i = 1 \implies x_i \in T$$

Graph-Cut (st-mincut)



Set formulation

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

Algebraic formulation

$$\min_{\mathbf{x}} \sum_{(i,j) \in E} c_{ij}(1 - x_i)x_j \qquad s.t. \begin{array}{l} x_s = 0 \\ x_t = 1 \end{array}$$

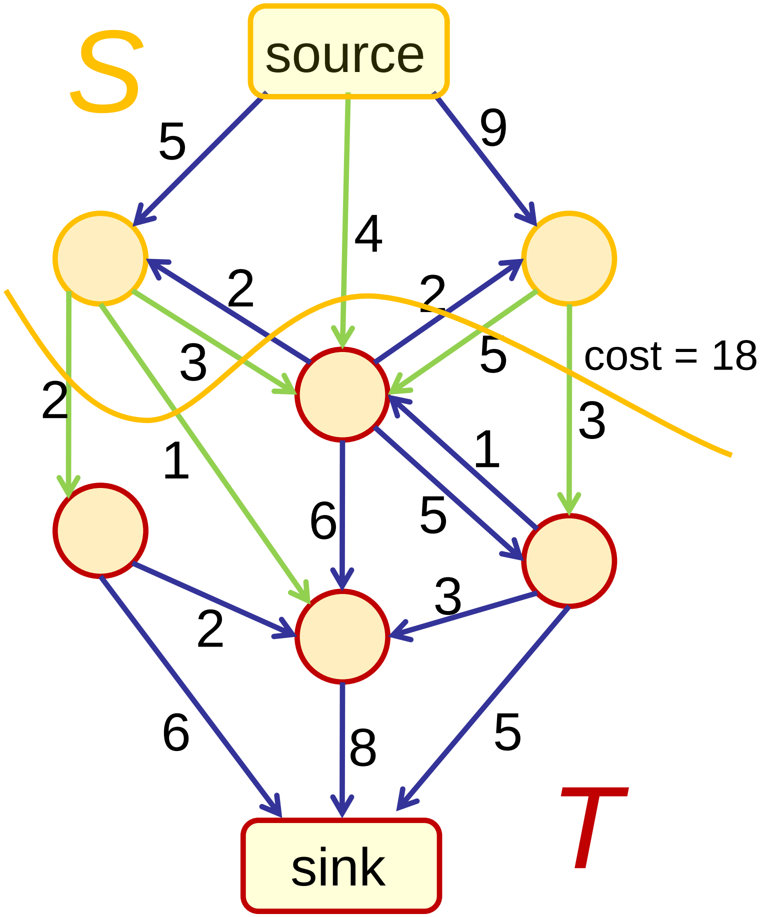
$$x_i = 0 \implies x_i \in S \qquad x_i = 1 \implies x_i \in T$$

After substitution

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

$$x_i = 0 \implies x_i \in S \qquad x_i = 1 \implies x_i \in T$$

Graph-Cut (st-mincut)



Algorithms

- Augmented path method
- Push-relabel method

Foreground / Background Estimation

Rother et al. SIGGRAPH04

Foreground / Background Estimation

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

Data term **Smoothness term**

$$x_i = 0 \implies i \in Background$$
$$x_i = 1 \implies i \in Foreground$$

Data term

$$\psi_i(0) = -\log(\boxed{p(x_i \notin FG)})$$
$$\psi_i(1) = -\log(\boxed{p(x_i \in FG)})$$

Estimated using FG / BG colour models

Smoothness term

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

where $K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2)$

Intensity dependent smoothness

Foreground / Background Estimation

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

$$x_i = 0 \implies i \in Background$$
$$x_i = 1 \implies i \in Foreground$$

Data term

$$\psi_i(x_i) = \psi_i(0)(1 - x_i) + \psi_i(1)x_i$$

Smoothness term

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

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i \in \mathcal{V}} \psi_i(0)(1 - x_i) + \psi_i(1)x_i + \sum_{i,j \in \mathcal{E}} (K_{ij}(1 - x_i)x_j + K_{ij}(1 - x_j)x_i)$$

Foreground / Background Estimation

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

$$x_i = 0 \implies i \in Background$$
$$x_i = 1 \implies i \in Foreground$$

Data term

$$\psi_i(x_i) = \psi_i(0)(1 - x_i) + \psi_i(1)x_i$$

Smoothness term

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

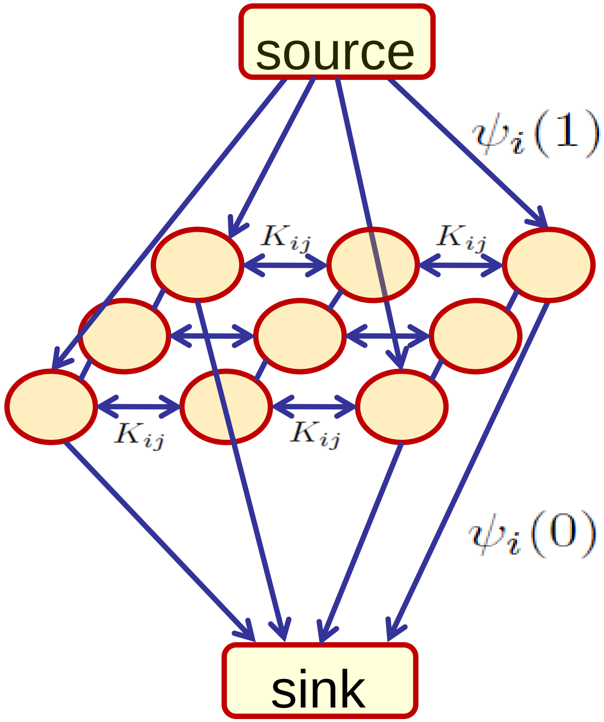
Min-Cut problem

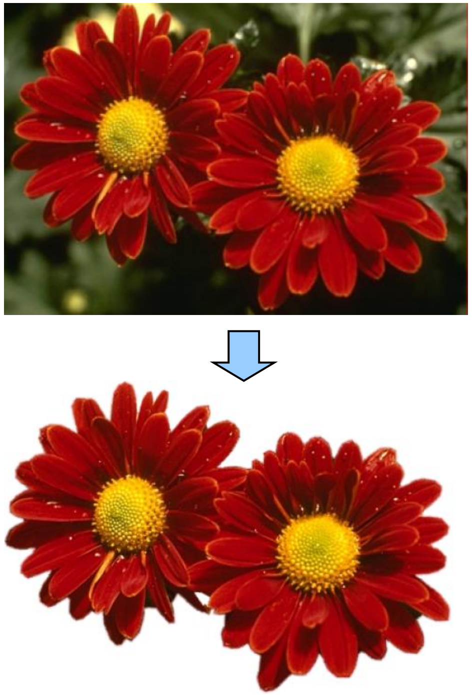
$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i \in \mathcal{V}} \psi_i(0)(1 - x_i) + \psi_i(1)x_i + \sum_{i,j \in \mathcal{E}} (K_{ij}(1 - x_i)x_j + K_{ij}(1 - x_j)x_i)$$

Foreground / Background Estimation

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

$$x_i = 0 \implies i \in Background$$
$$x_i = 1 \implies i \in Foreground$$





Solvability using GraphCut

Submodularity

$$E(0,0,\bar{\mathrm{x}}_{ij}) + E(1,1,\bar{\mathrm{x}}_{ij}) \leq E(0,1,\bar{\mathrm{x}}_{ij}) + E(1,0,\bar{\mathrm{x}}_{ij})$$

Solvability using GraphCut

Submodularity

$$E(0,0,\bar{\mathrm{x}}_{ij}) + E(1,1,\bar{\mathrm{x}}_{ij}) \leq E(0,1,\bar{\mathrm{x}}_{ij}) + E(1,0,\bar{\mathrm{x}}_{ij})$$

$$\arg\min_{\mathbf{x}} \sum_{i\in\mathcal{V}} \boxed{c_{it}(1-x_i)} + \sum_{i\in\mathcal{V}} \boxed{c_{si}x_i} + \sum_{i,j\in\mathcal{E}} \boxed{c_{ij}(1-x_i)x_j}$$

all terms submodular

submodularity = necessary condition

Solvability using GraphCut

Submodularity

$$E(0,0,\bar{\mathrm{x}}_{ij}) + E(1,1,\bar{\mathrm{x}}_{ij}) \leq E(0,1,\bar{\mathrm{x}}_{ij}) + E(1,0,\bar{\mathrm{x}}_{ij})$$

General pairwise potential

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

Solvability using GraphCut

Submodularity

$$E(0,0,\bar{\mathbf{x}}_{ij}) + E(1,1,\bar{\mathbf{x}}_{ij}) \leq E(0,1,\bar{\mathbf{x}}_{ij}) + E(1,0,\bar{\mathbf{x}}_{ij})$$

General pairwise potential

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

$$= K_{ij} + g_i' x_i + g_j' x_j + c_{ij}(1-x_i)x_j + c_{ij}x_i(1-x_j)$$

where

$$K_{ij} = g_{ij}^{00} \qquad g_j' = \frac{g_{ij}^{01} + g_{ij}^{11} - g_{ij}^{10} - g_{ij}^{00}}{2}$$

$$g_i' = \frac{g_{ij}^{10} + g_{ij}^{11} - g_{ij}^{01} - g_{ij}^{00}}{2} \qquad c_{ij} = \frac{g_{ij}^{01} + g_{ij}^{10} - g_{ij}^{00} - g_{ij}^{11}}{2}$$

Solvability using GraphCut

Submodularity

$$E(0,0,\bar{\mathbf{x}}_{ij}) + E(1,1,\bar{\mathbf{x}}_{ij}) \leq E(0,1,\bar{\mathbf{x}}_{ij}) + E(1,0,\bar{\mathbf{x}}_{ij})$$

General pairwise potential

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

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

could be arbitrary

where

$$K_{ij} = g_{ij}^{00} \qquad g'_j = \frac{g_{ij}^{01} + g_{ij}^{11} - g_{ij}^{10} - g_{ij}^{00}}{2}$$

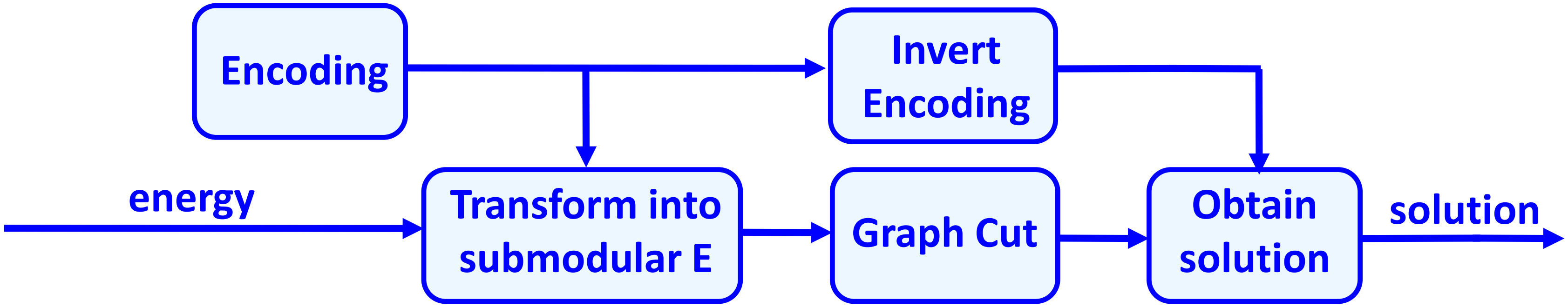
$$g'_i = \frac{g_{ij}^{10} + g_{ij}^{11} - g_{ij}^{01} - g_{ij}^{00}}{2} \qquad \boxed{c_{ij} = \frac{g_{ij}^{01} + g_{ij}^{10} - g_{ij}^{00} - g_{ij}^{11}}{2} \geq 0}$$

Submodularity = sufficient condition

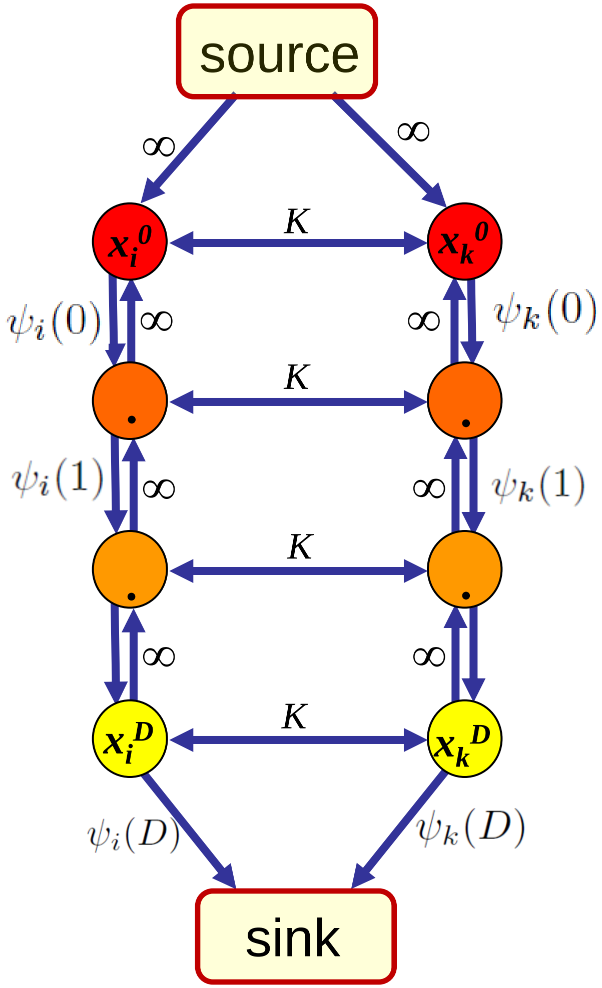
General GraphCut pipeline

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



Data term $\psi_i(z_i)$

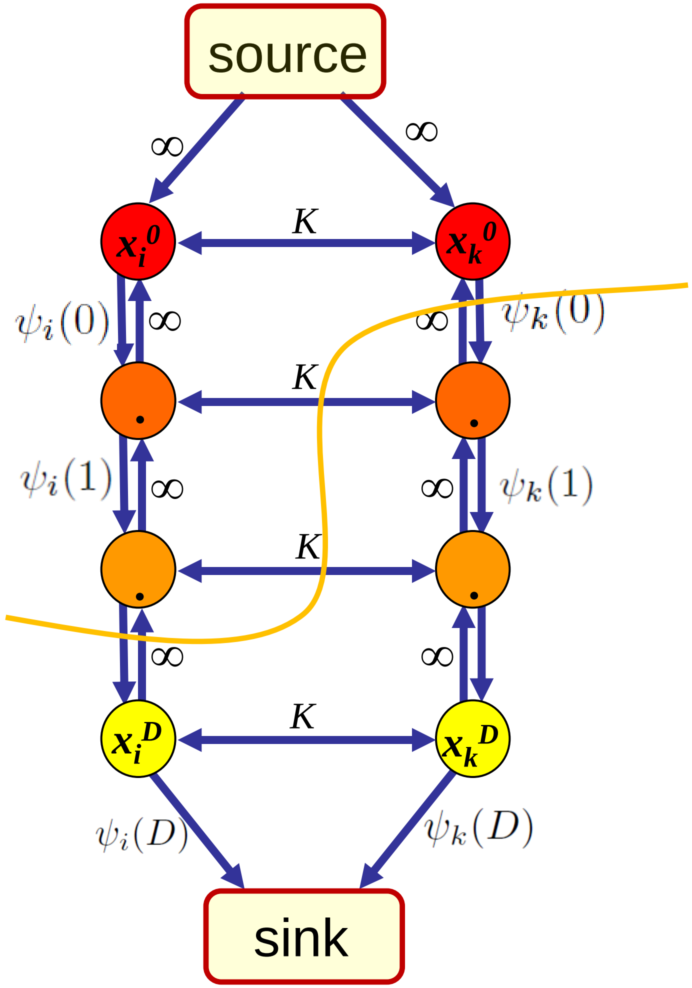
Smoothness term $\psi_{ij}(z_i, z_j) = K|z_i - z_j|$

Encoding

$$z_i = 0 \iff \{x_i^0 = 0, x_i^1 = 1, x_i^2 = 1, x_i^3 = 1, .., x_i^D = 1\}$$
$$z_i = 1 \iff \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 1, x_i^3 = 1, .., x_i^D = 1\}$$
$$z_i = 2 \iff \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 1, .., x_i^D = 1\}$$
$$..$$
$$z_i = D \iff \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 0, .., x_i^D = 0\}$$

Ishikawa PAMI03

Mulit-label energy with linear pairwise potentials



Data term $\psi_i(z_i)$

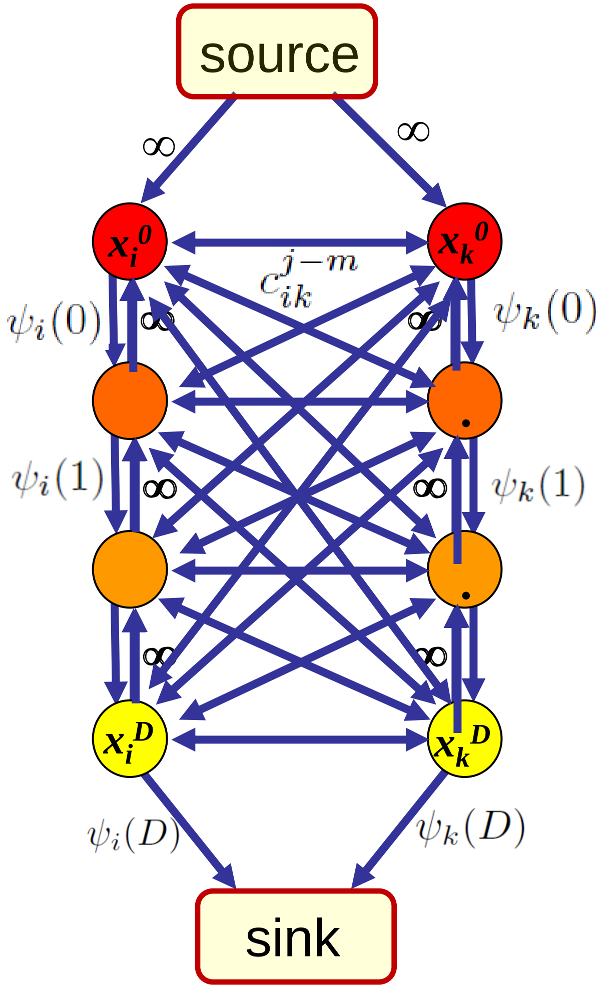
Smoothness term $\psi_{ij}(z_i, z_j) = K|z_i - z_j|$

Encoding

$$
\begin{aligned}
z_i = 0 &\iff \{x_i^0 = 0, x_i^1 = 1, x_i^2 = 1, x_i^3 = 1, .., x_i^D = 1\} \\
z_i = 1 &\iff \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 1, x_i^3 = 1, .., x_i^D = 1\} \\
z_i = 2 &\iff \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 1, .., x_i^D = 1\} \\
&\;\;..\\
z_i = D &\iff \{x_i^0 = 0, x_i^1 = 0, x_i^2 = 0, x_i^3 = 0, .., x_i^D = 0\}
\end{aligned}
$$

Ishikawa PAMI03

Mulit-label energy with convex pairwise potentials



Data term $\psi_i(z_i)$

Smoothness term $\psi_{ik}(z_i, z_k) = \boxed{f(z_i - z_k)}$

Any convex function

$$\psi_{ik}(z_i, z_k) = \sum_{d=-D}^{D} (c_{ik}^d \max(z_i - z_k + d, 0) + c_{ki}^d \max(z_k - z_i + d, 0))$$

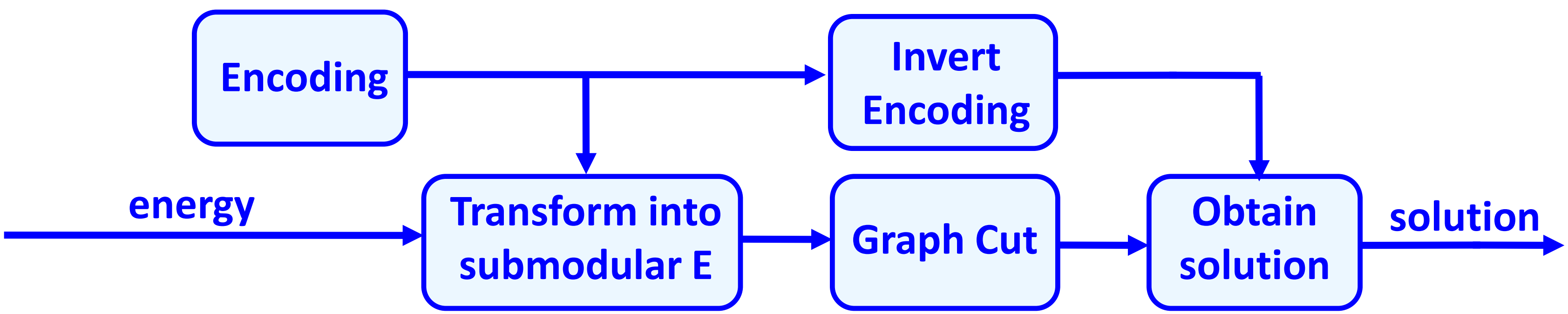
where $$c_{ik}^d = \frac{f(d+1) - 2f(d) + f(d-1)}{2}$$

Ishikawa PAMI03

Higher order minimization with GraphCut

$$\boxed{\psi_c(\mathbf{x}_c)} = \min_{\mathbf{z}_c} \boxed{\psi_c^p(\mathbf{x}_c, \mathbf{z}_c)}$$

Higher order term **Pairwise term**



Higher order minimization with GraphCut

$$\boxed{\psi_c(\mathbf{x}_c)} = \min_{\mathbf{z}_c} \boxed{\psi_c^p(\mathbf{x}_c, \mathbf{z}_c)}$$

Higher order term **Pairwise term**

Example : $\psi(x_1, x_2, x_3) = -x_1 x_2 x_3$

Higher order minimization with GraphCut

$$\boxed{\psi_c(\mathbf{x}_c)} = \min_{\mathbf{z}_c} \boxed{\psi_c^p(\mathbf{x}_c, \mathbf{z}_c)}$$

Higher order term **Pairwise term**

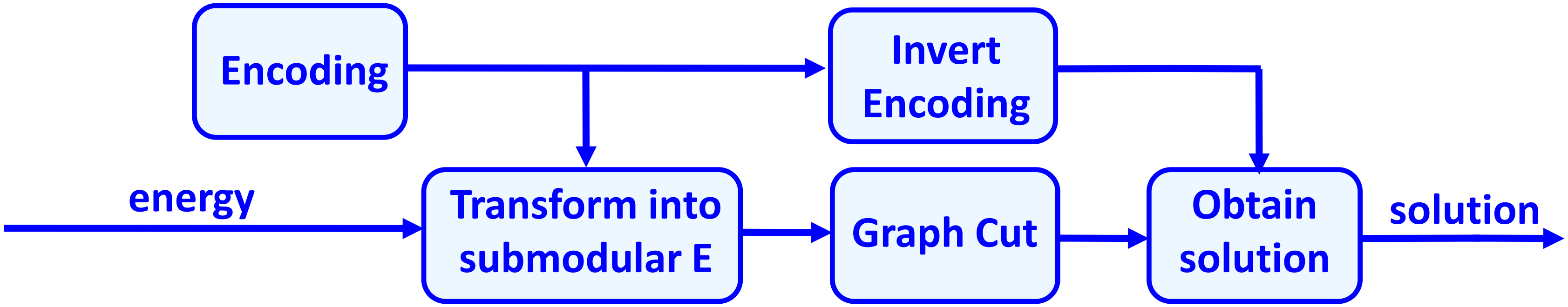
Example : $\psi(x_1, x_2, x_3) = -x_1 x_2 x_3 = \min_z z(2 - x_1 - x_2 - x_3)$

	$-x_1 x_2 x_3$	$\min_z z(2 - x_1 - x_2 - x_3)$
$x_1 = 0 \quad x_2 = 0 \quad x_3 = 0$	0	$\min_z 2z = 0$
$x_1 = 0 \quad x_2 = 0 \quad x_3 = 1$	0	$\min_z z = 0$
$x_1 = 0 \quad x_2 = 1 \quad x_3 = 1$	0	$\min_z 0 = 0$
$x_1 = 1 \quad x_2 = 1 \quad x_3 = 1$	-1	$\min_z (-z) = -1$

Kolmogorov ECCV06, Ramalingam et al. DAM12

General GraphCut pipeline

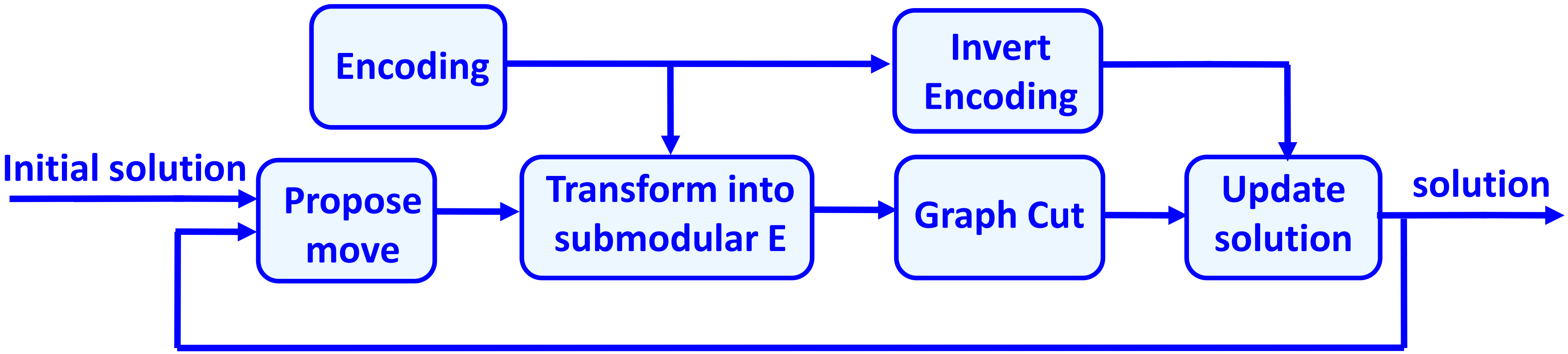
What if no encoding leads to pairwise submodular problem ?



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



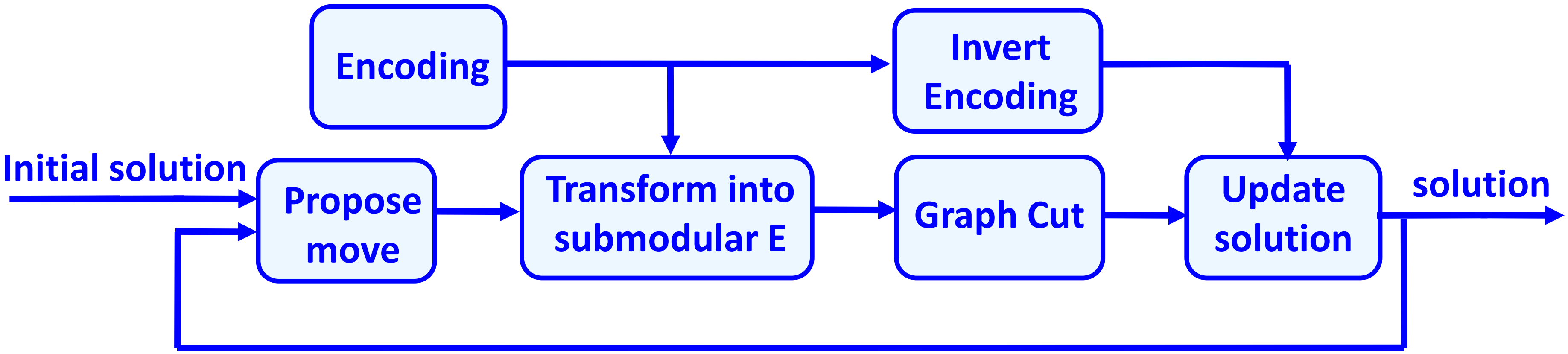
Move making algorithms

αβ-swap

- Each variable taking label α or β can change its label to α or β
- Move space defined by the transformation function

Transformation function

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



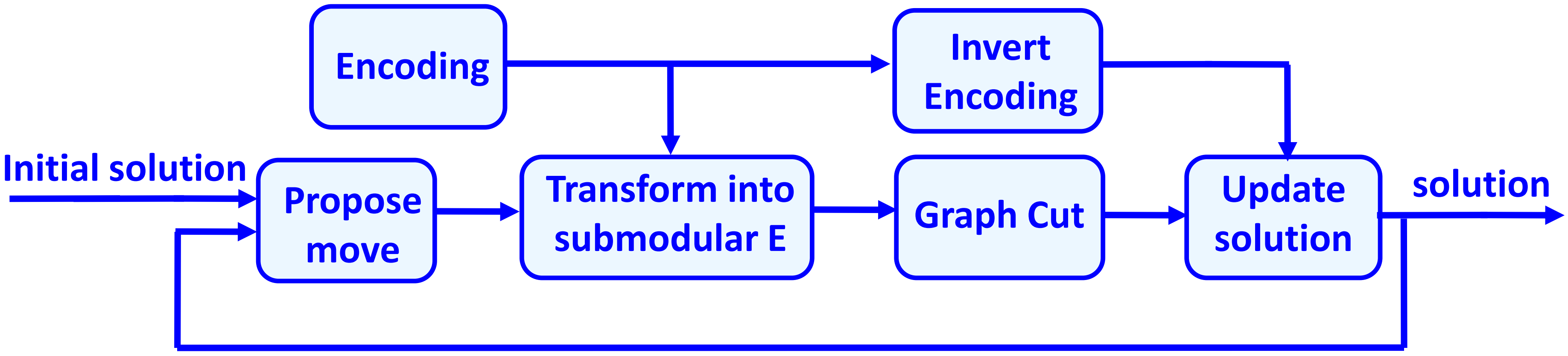
Move making algorithms

α-expansion

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

Transformation function

$$T_\alpha(x_i, t_i) = \begin{cases} \alpha & \text{if} \quad t_i = 0 \\ x_i & \text{if} \quad t_i = 1 \end{cases}$$



Move making algorithms

Sufficient condition for submodularity of each move :

αβ-swap

semi-metricity

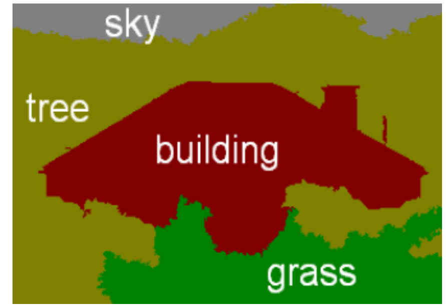
$$\forall\, l_a, l_b \in \mathcal{L}$$
$$\begin{aligned}\psi^p(l_a, l_a) &= 0 \\ \psi^p(l_a, l_b) &= \psi^p(l_b, l_a) \geq 0\end{aligned}$$

α-expansion

metricity

$$\forall\, l_a, l_b \in \mathcal{L}$$
$$\begin{aligned}\psi^p(l_a, l_a) &= 0 \\ \psi^p(l_a, l_b) &= \psi^p(l_b, l_a) \geq 0\end{aligned}$$
$$\psi^p(l_a, l_b) + \psi^p(l_b, l_c) \geq \psi^p(l_a, l_c)$$

Semantic Segmentation



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

Data term **Smoothness term**

Data term

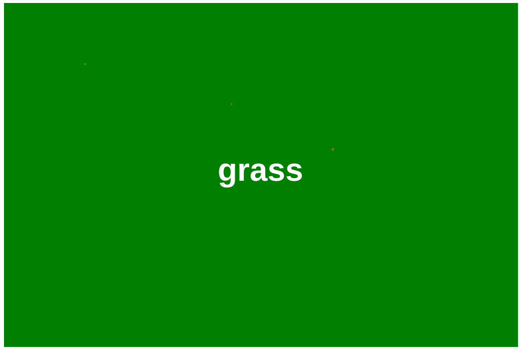
Discriminatively trained classifier

Smoothness term

$$\psi_{ij}(x_i, x_j) = K_{ij}\delta(x_i \neq x_j) \qquad K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2)$$

Semantic Segmentation

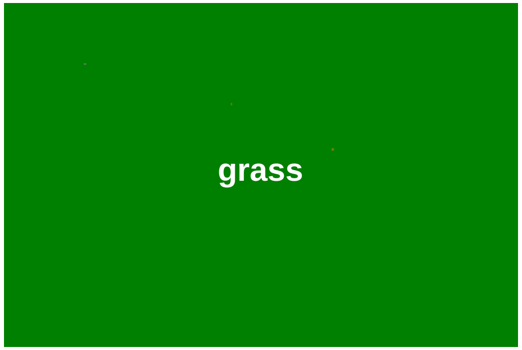
Original Image



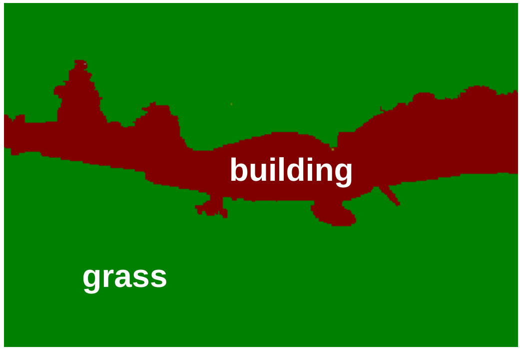
Initial solution

Semantic Segmentation

Original Image



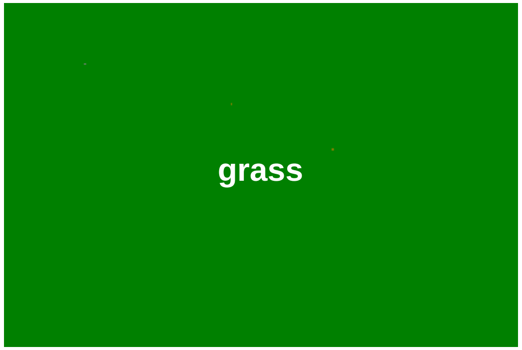
Initial solution



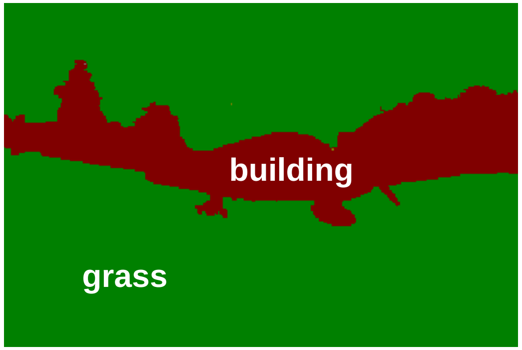
Building expansion

Semantic Segmentation

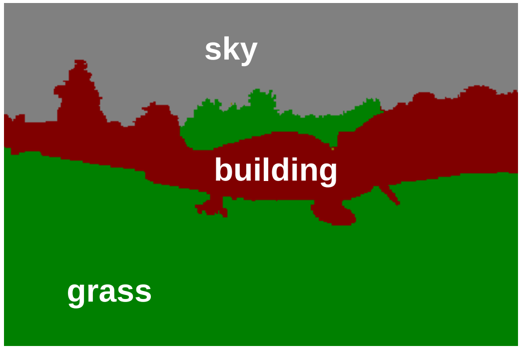
Original Image



Initial solution



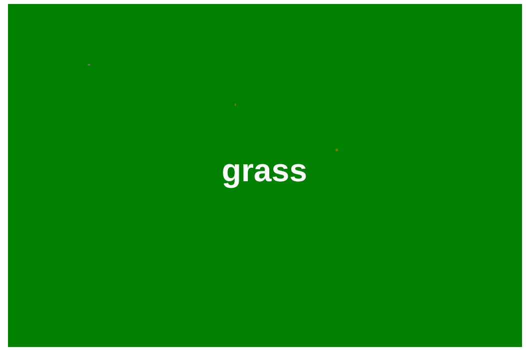
Building expansion



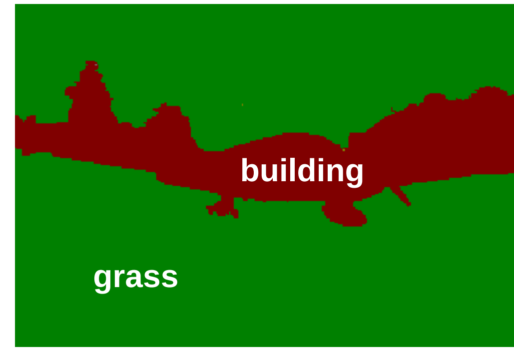
Sky expansion

Semantic Segmentation

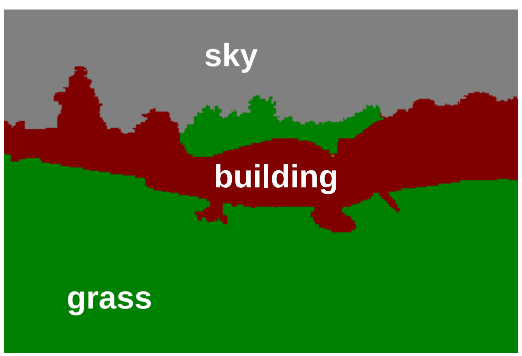
Original Image



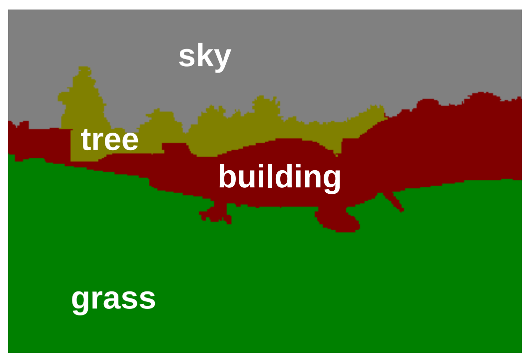
Initial solution



Building expansion



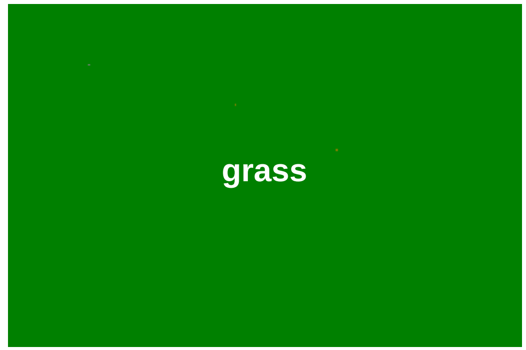
Sky expansion



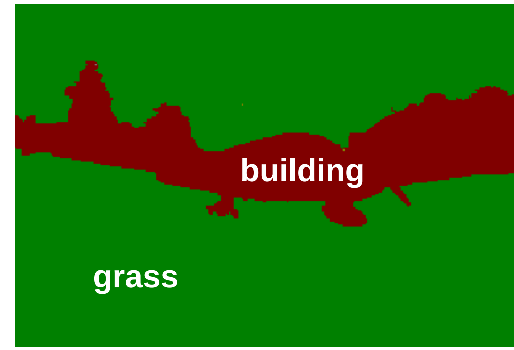
Tree expansion

Semantic Segmentation

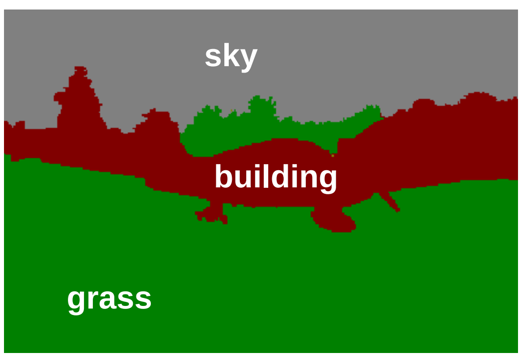
Original Image



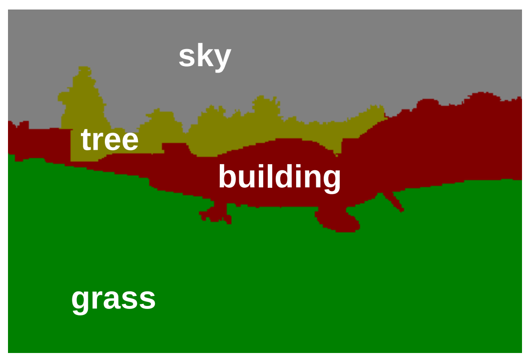
Initial solution



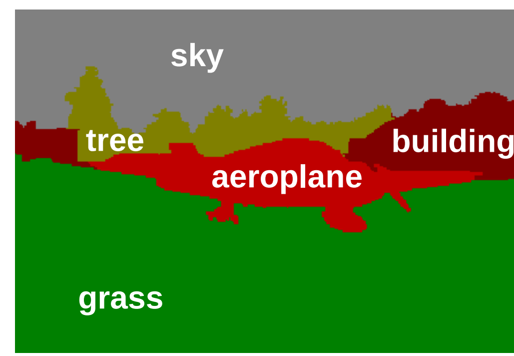
Building expansion



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 $\overline{x}_i$ s.t. $x_i = 1 - \overline{x}_i$
- Energy transformed into a submodular over x_i and $\overline{x}_i$

$$
\begin{aligned}
E(\mathbf{x}) &= \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j) = \sum_{i \in \mathcal{V}} (g_i^1 x_i + g_i^0 (1 - x_i)) \\
&+ \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} (g_{ij}^{00}(1 - x_i)(1 - x_j) + g_{ij}^{01}(1 - x_i)x_j + g_{ij}^{10} x_i (1 - x_j) + g_{ij}^{11} x_i x_j) \\
&= \sum_{i \in \mathcal{V}} (\frac{g_i^1}{2}(x_i + (1 - \bar{x}_i)) + \frac{g_i^0}{2}(\bar{x}_i + (1 - x_i))) \\
&+ \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} (\frac{g_{ij}^{00}}{2}(\bar{x}_i(1 - x_j) + (1 - x_i)\bar{x}_j) + \frac{g_{ij}^{01}}{2}((1 - x_i)x_j + \bar{x}_i(1 - \bar{x}_j)) \\
&+ \frac{g_{ij}^{11}}{2}(x_i(1 - \bar{x}_j) + (1 - \bar{x}_i)x_j)) + \frac{g_{ij}^{10}}{2}(x_i(1 - x_j) + (1 - \bar{x}_i)\bar{x}_j)
\end{aligned}
$$

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 $\overline{x}_i$ 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
- 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-occurence 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

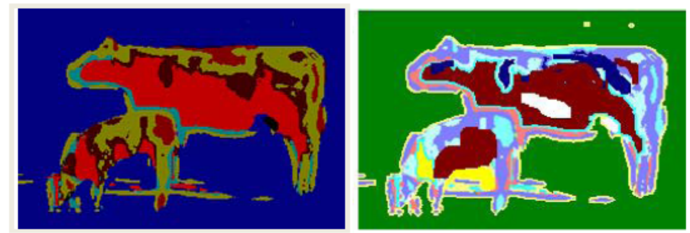
→

Multiple features

SIFT ↓ Texton
Colour SfM

Clustering

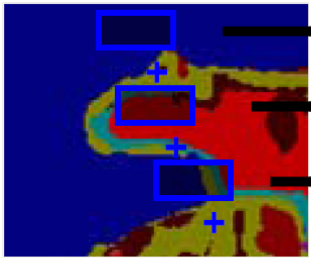
→



Feature maps
Colours ⇔ Cluster

Shape filter (texton / colour / location / HOG , ■ , ▭)

feature type f — cluster t — rectangle r



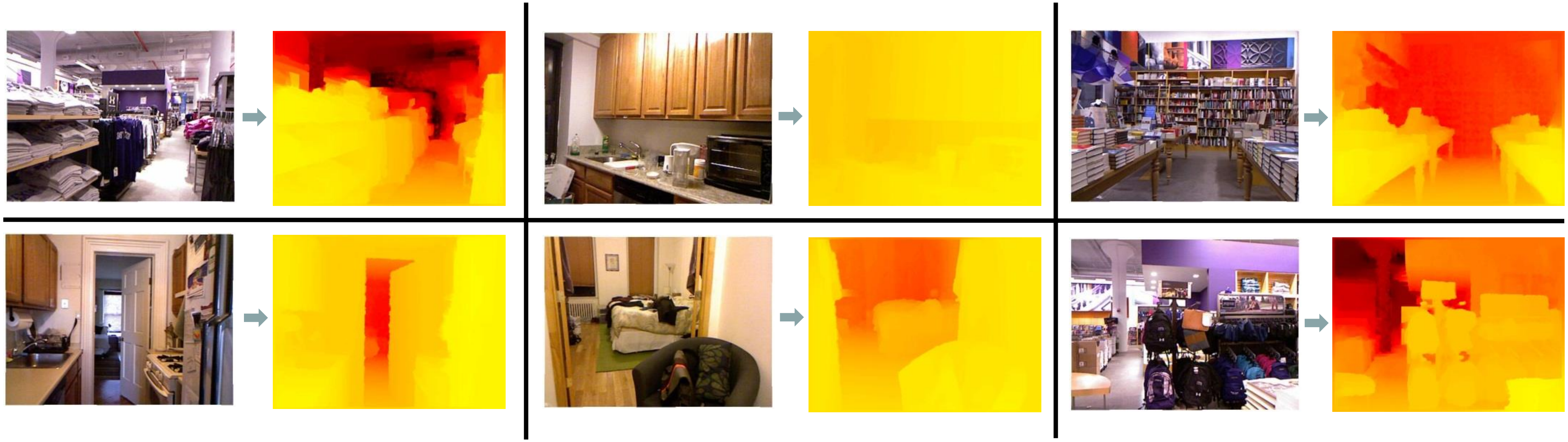
$v(i, f, t, r) = a$

$v(i, f, t, r) = 0$

$v(i, f, t, r) = a/2$

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

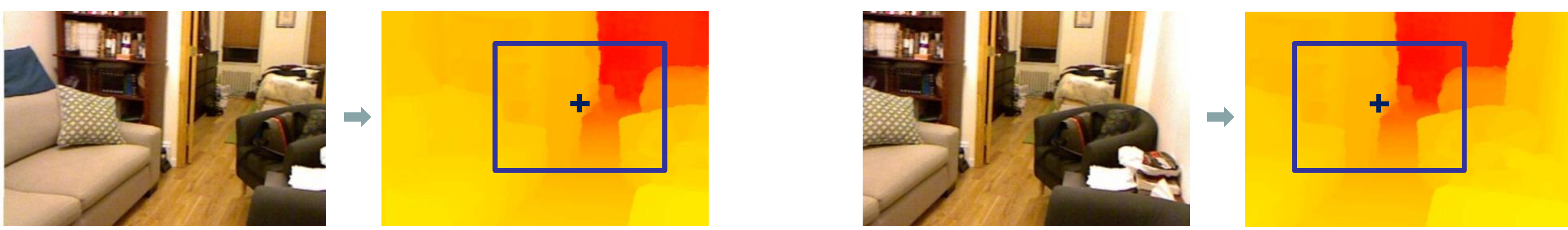
Desired properties :

1. Pixel-wise classifier
2. Translation invariant

$$\boxed{H_d(x)} := H_d(\boxed{W^{w,h}(I,x)})$$

Classifier response for *x* and at a depth *d*

window $w \times h$ around the point $x \in I$

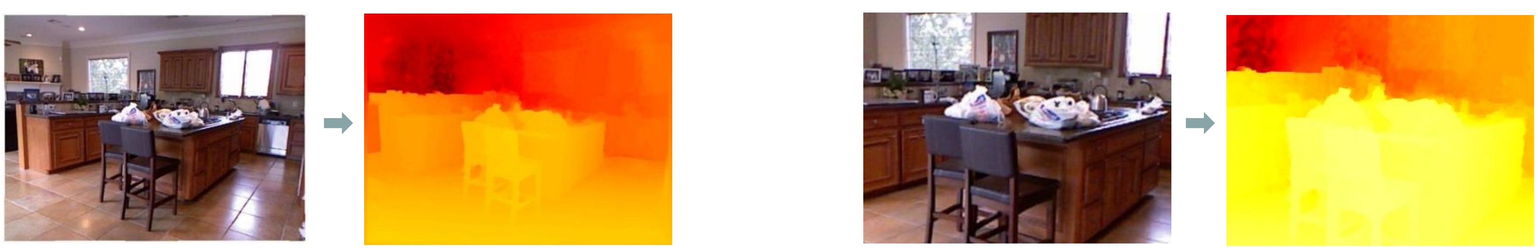


Data-driven Depth Estimation

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

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

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 :

$$H_d(W^{w,h}(I,x)) = H_{d_c}(W^{w,h}(\frac{d}{d_c}I, \frac{d}{d_c}x))$$

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

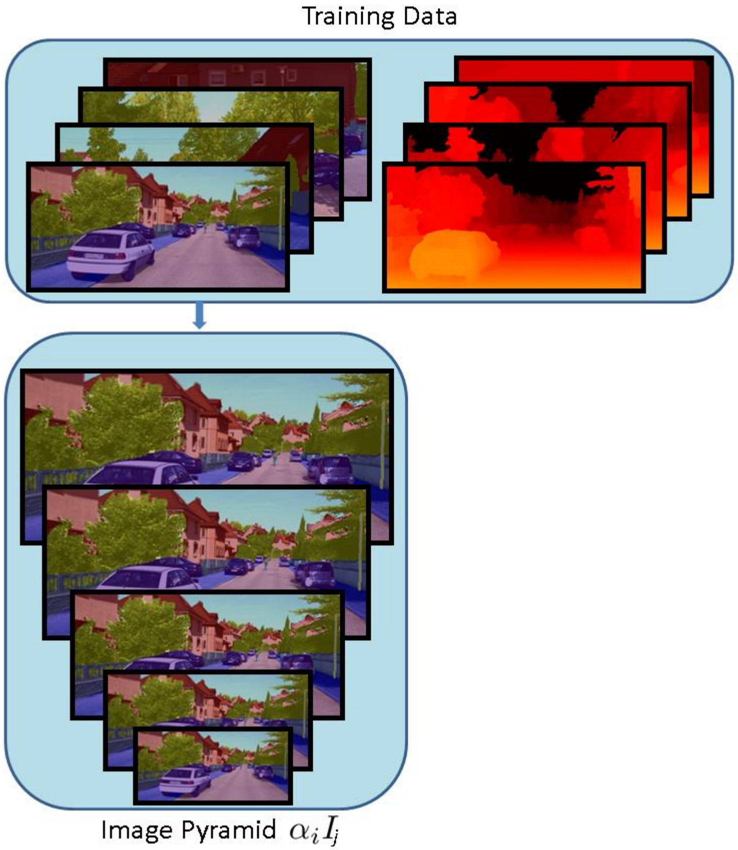
Generalized to multiple semantic classes

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

semantic label

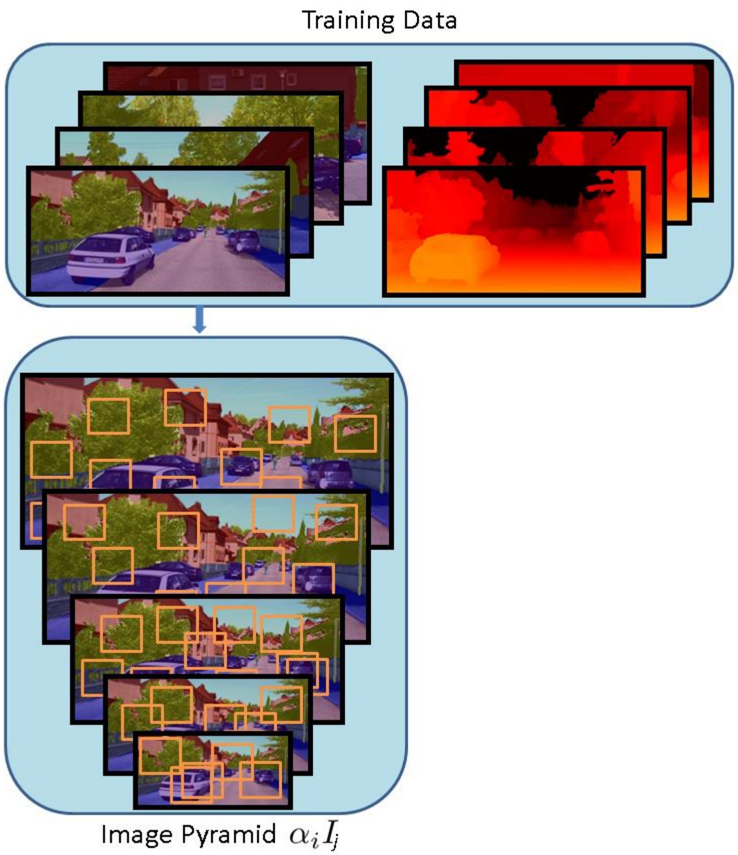
Training the classifier

1. Image pyramid is built



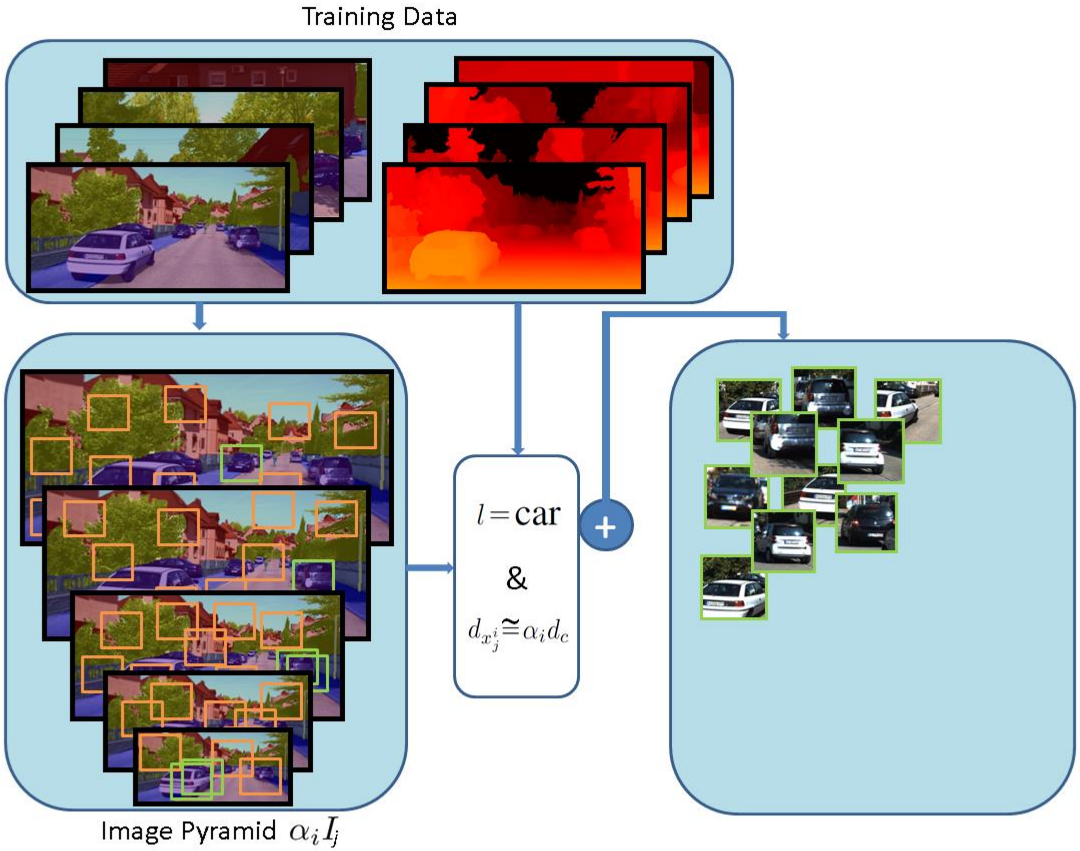
Training the classifier

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



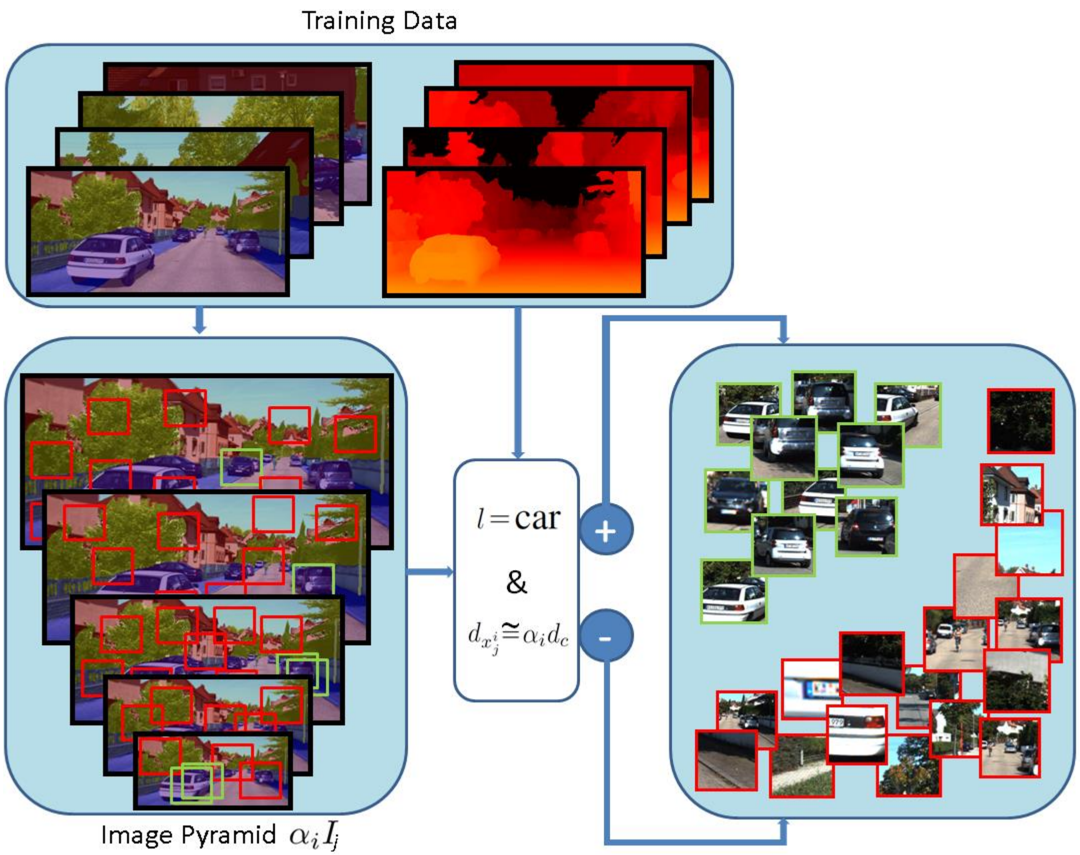
Training the classifier

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



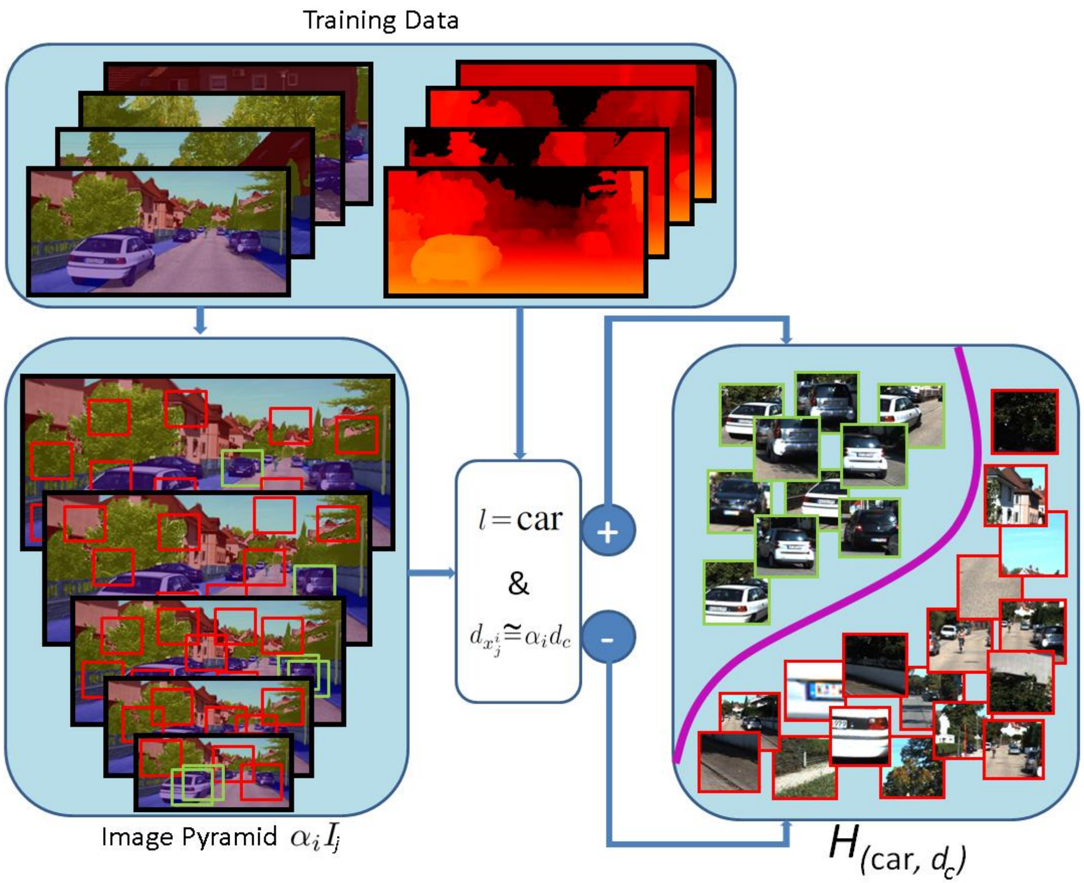
Training the classifier

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

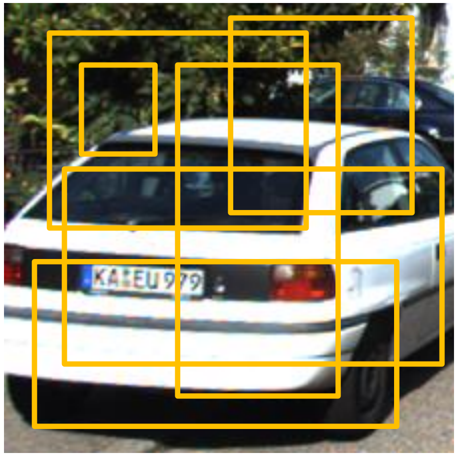
1. Image pyramid is built
2. Training data randomly sampled
3. Samples of each class at d_C used as positives
4. Samples of other classes or at $d \neq d_C$ used as negatives
5. Multi-class classifier trained



Classifying the patch

Dense Features   SIFT, LBP, Self Similarity, Texton

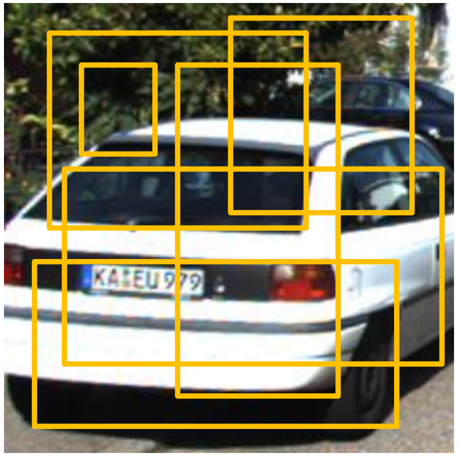
Classifying the patch



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

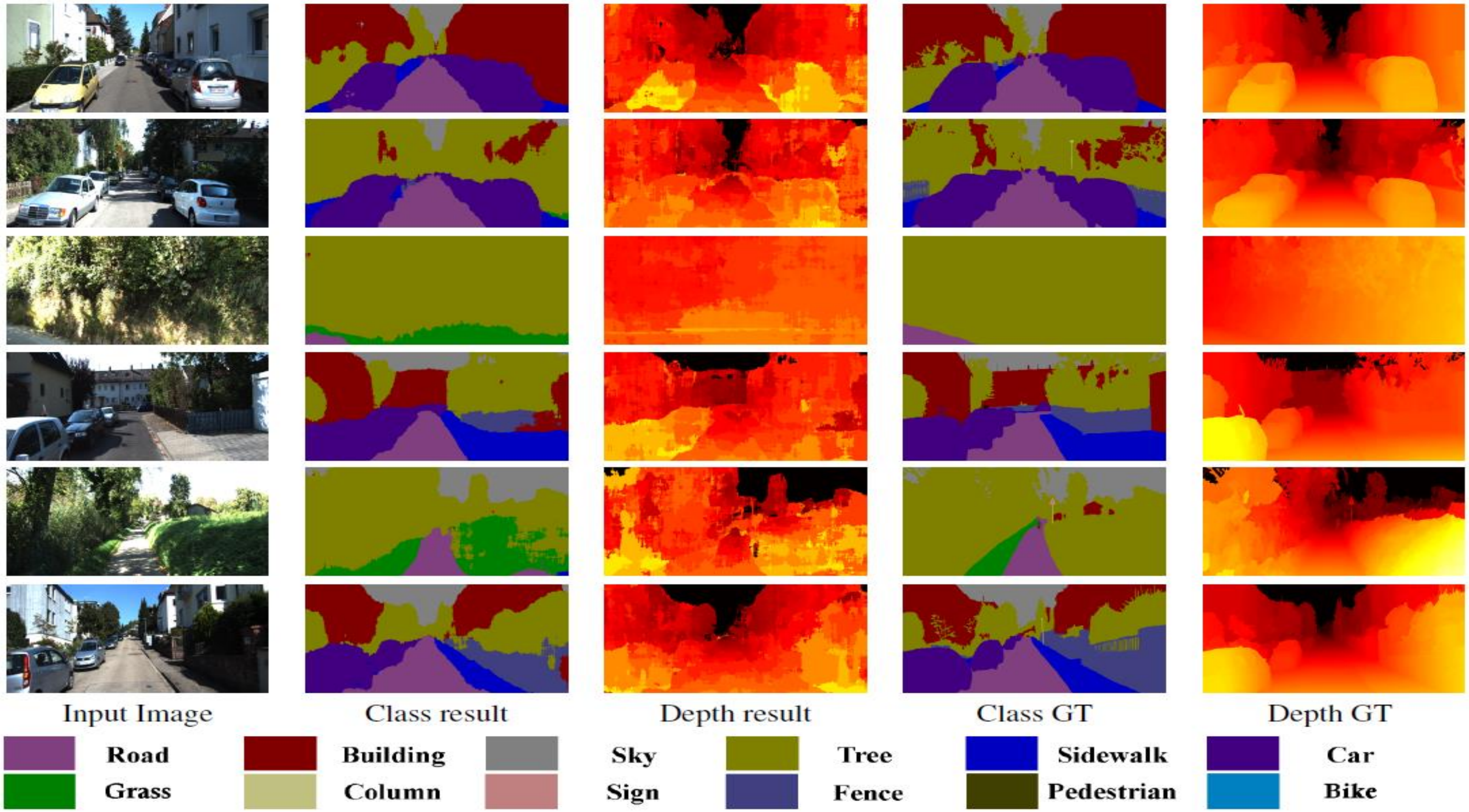
KITTI dataset

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

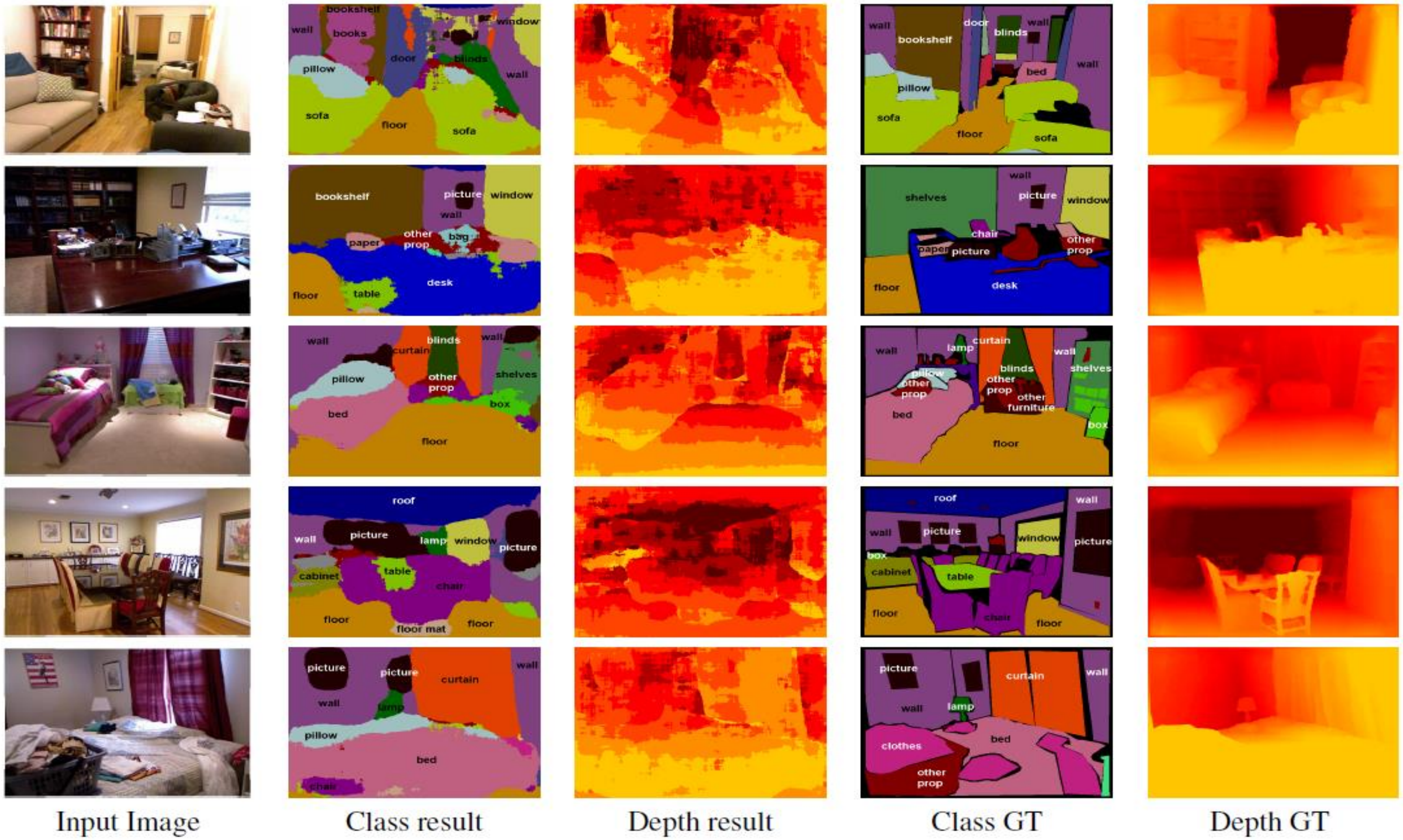
NYU2 dataset

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

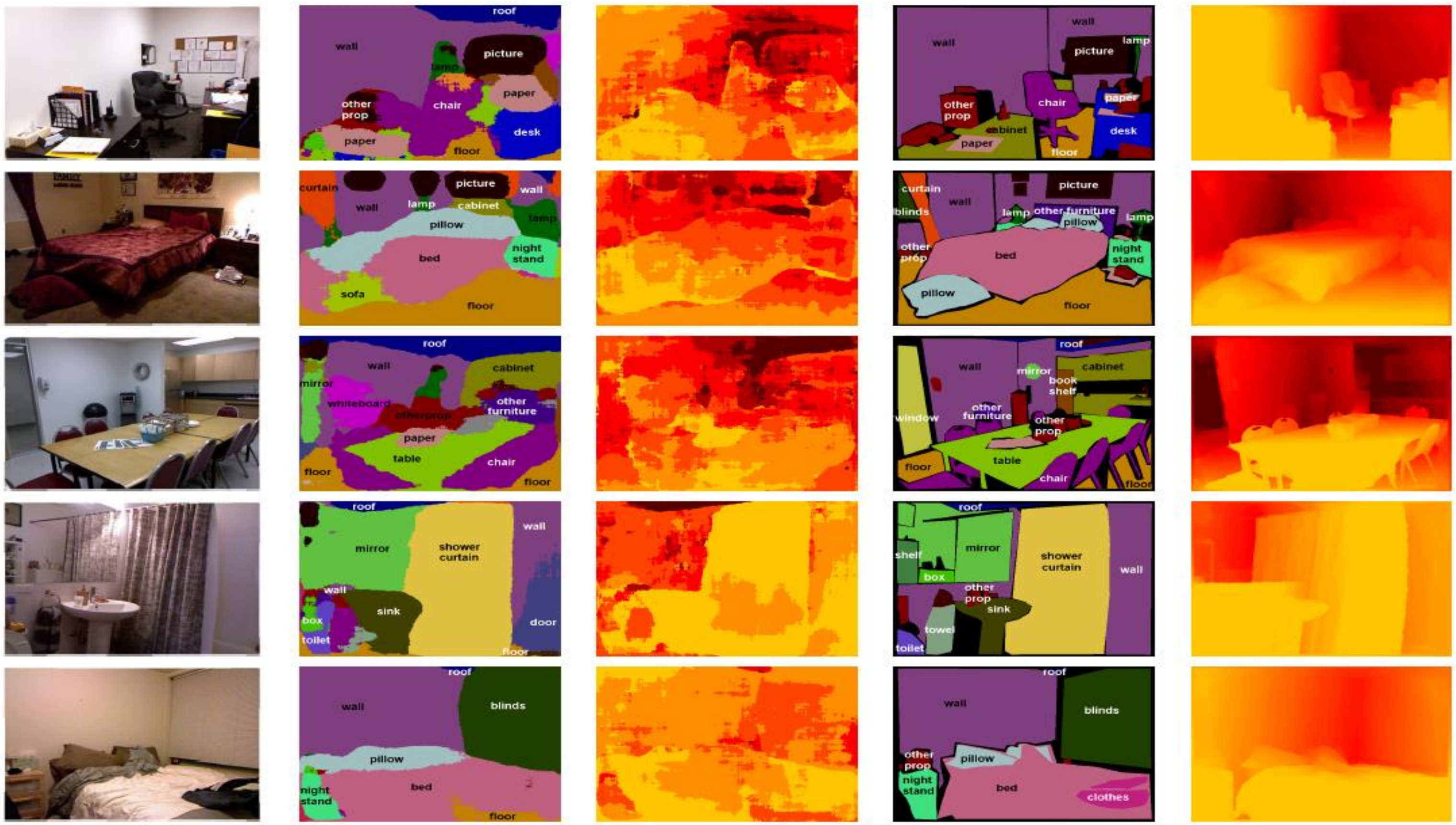
KITTI results



NYU2 results

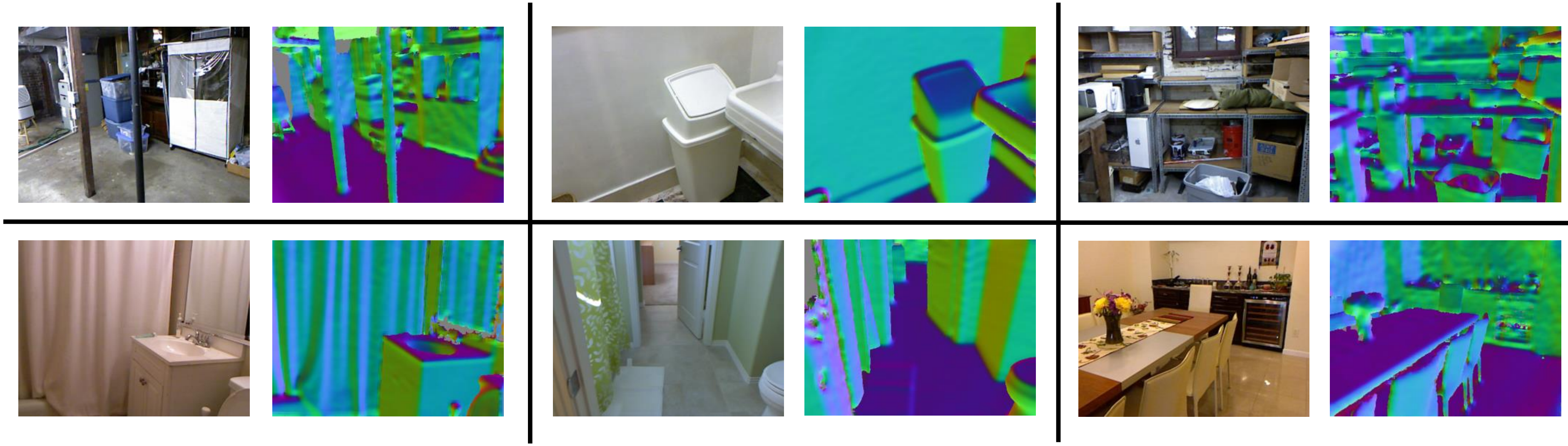


NYU2 results



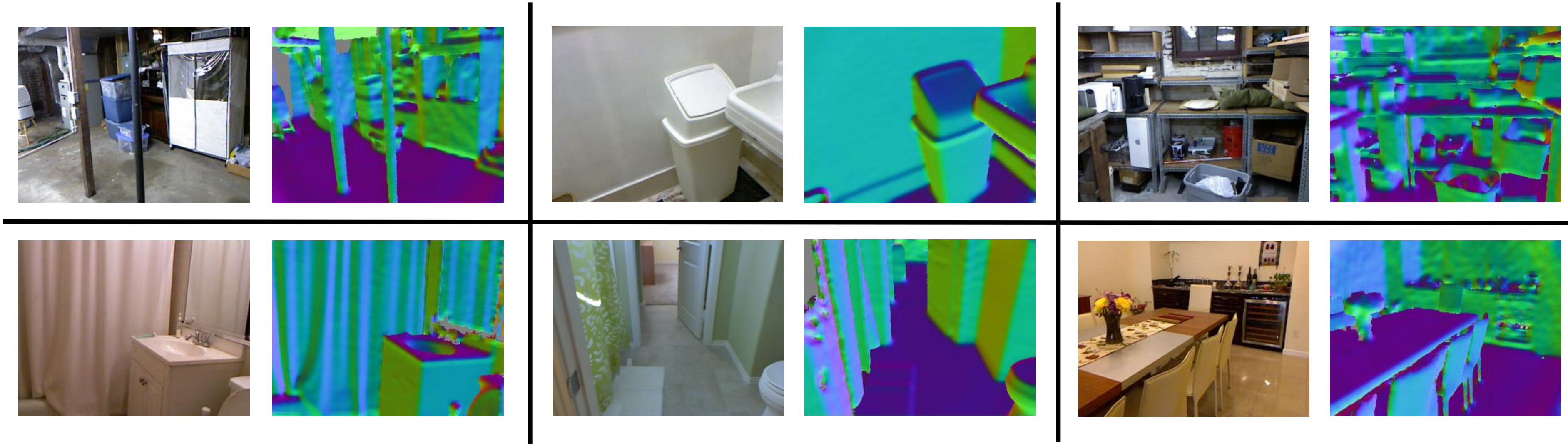
Input Image | Class result | Depth result | Class GT | Depth GT

Surface Normal Estimation



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

Surface Normal Estimation

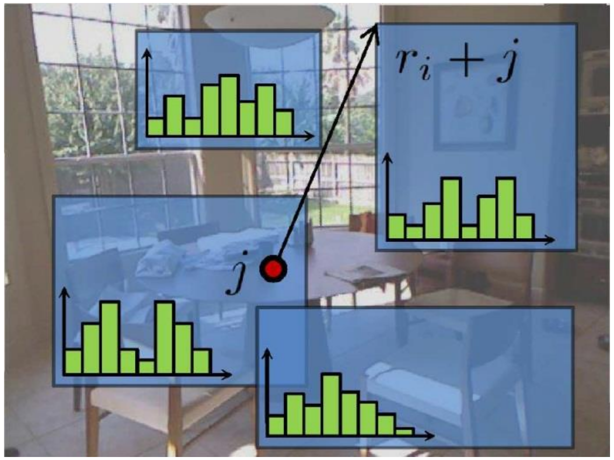


Not explored much in the literature… so how to approach it?

Pixels or Super-pixels?

Pixel-based Classifiers

Input image

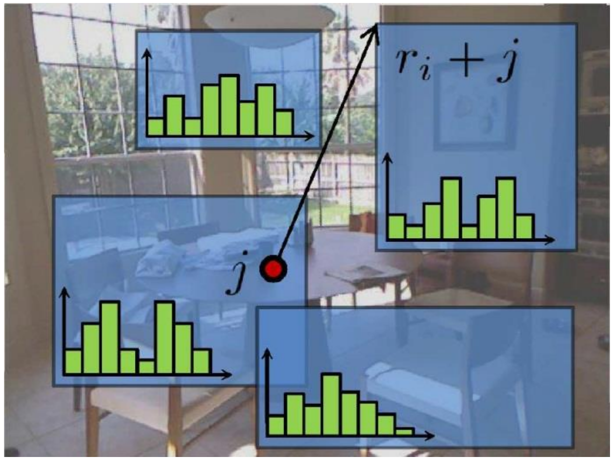


Feature representation

- Context-based (context pixels or rectangles) feature representations [Shotton06, Shotton08]

Pixel-based Classifiers

Input image

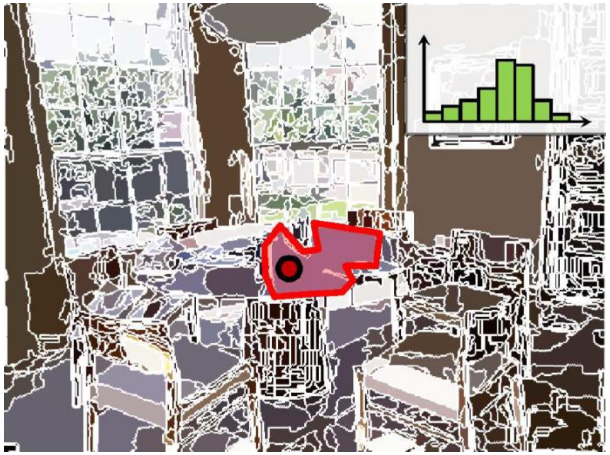


Feature representation

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

Segment-based Classifiers

Input image

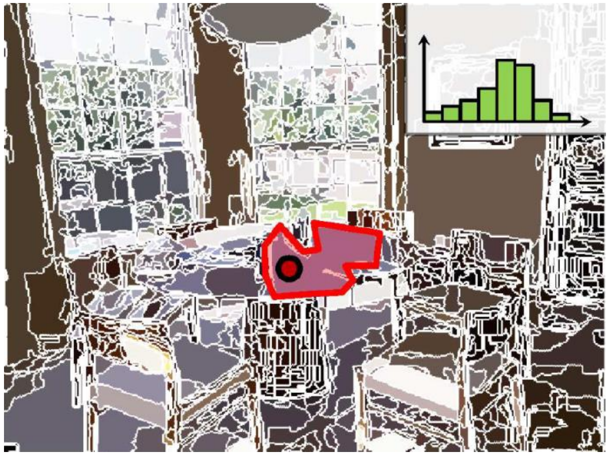


Feature representation

- Based on feature statistics in segments

Segment-based Classifiers

Input image

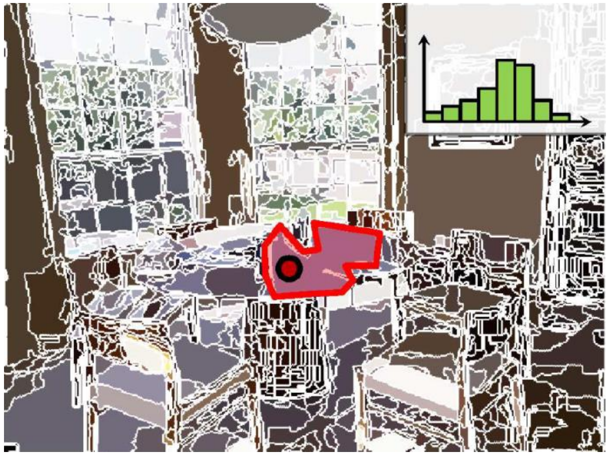


Feature representation

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

Segment-based Classifiers

Input image

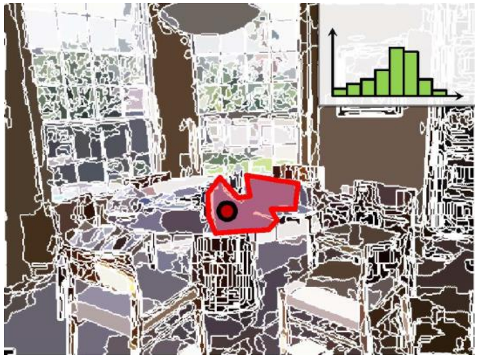


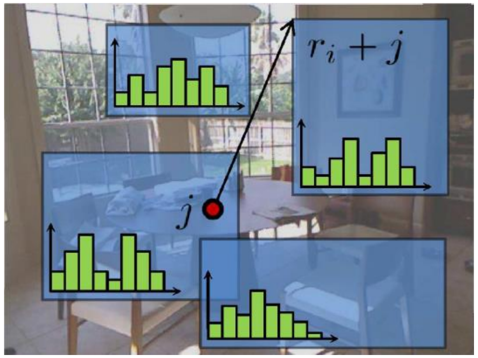
Feature representation

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

Joint Regularization

Input image



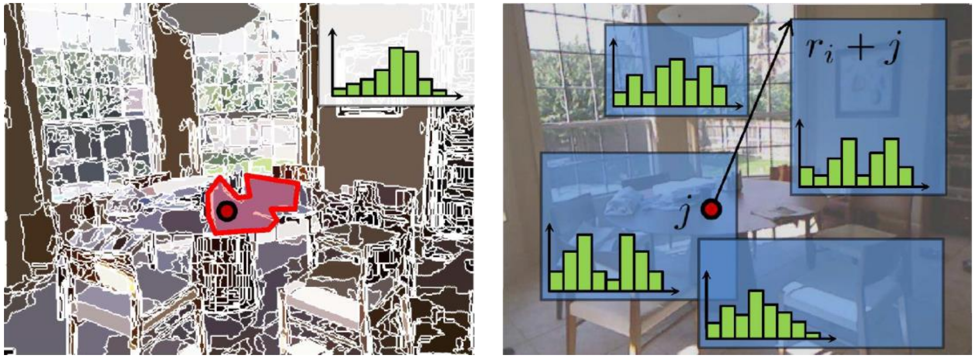


Independent classifiers

- Existing optimization methods (Ladicky09) designed for discrete labels

Joint Regularization

Input image



Independent classifiers

- Existing optimization methods (Ladicky09) designed for discrete labels
- Not obvious how to generalize for continuous problems

Joint Regularization

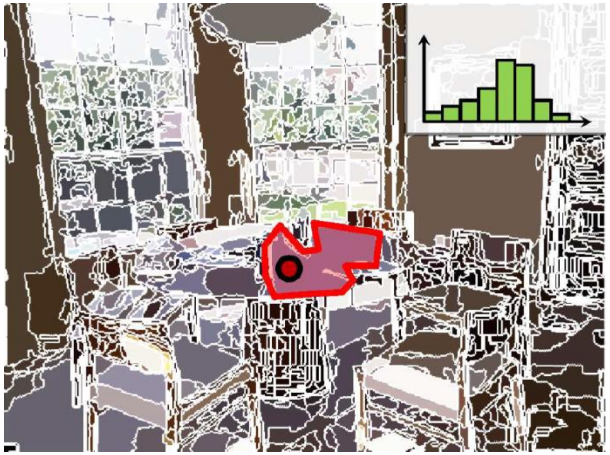
Input image

Independent classifiers

- 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

Input image

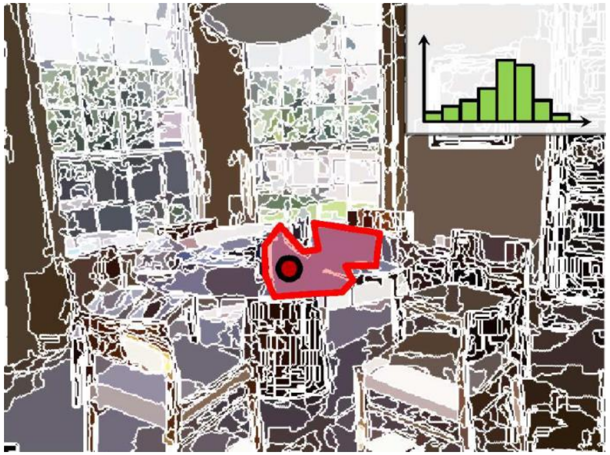


Segment representation

How to convert segment representation into pixel representation?

Joint Learning

Input image



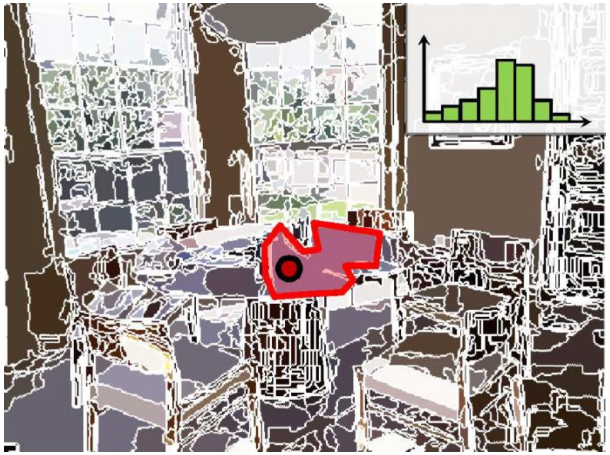
Segment representation

How to convert segment representation into pixel representation?

- Representation of a pixel the same as of the segment it belongs to

Joint Learning

Input image

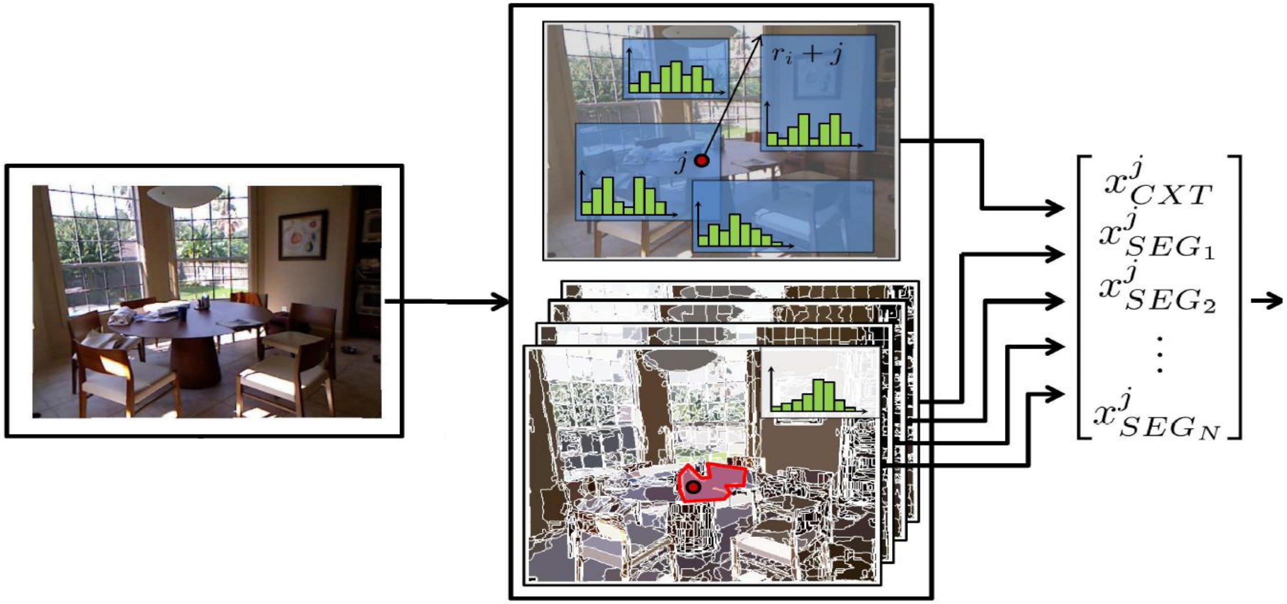


Segment representation

How to convert segment representation into pixel representation?

- Representation of a pixel the same as of the segment it belongs to
- Equivalent to weighted segment based approach

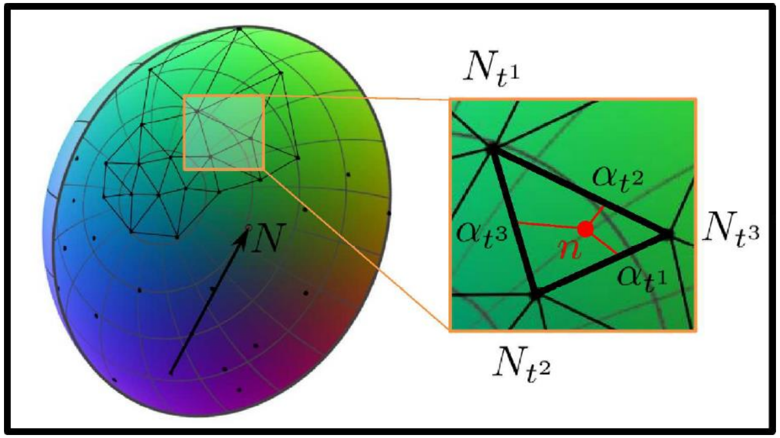
Joint Learning



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
- Concatenation to combine pixel and multiple segment representations

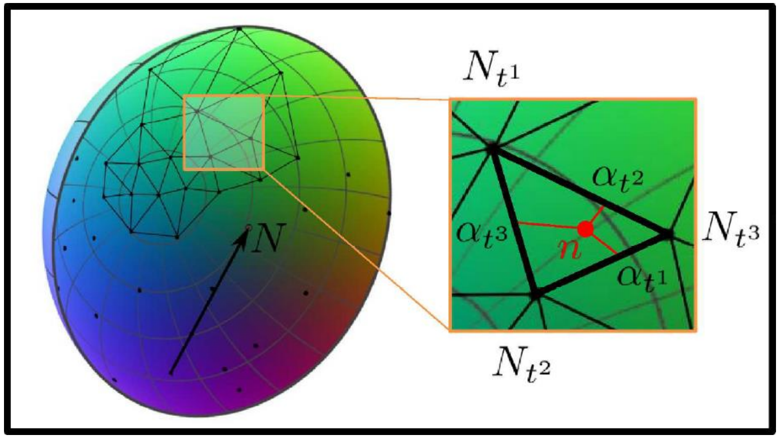
Joint Learning



To simplify regression problem

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

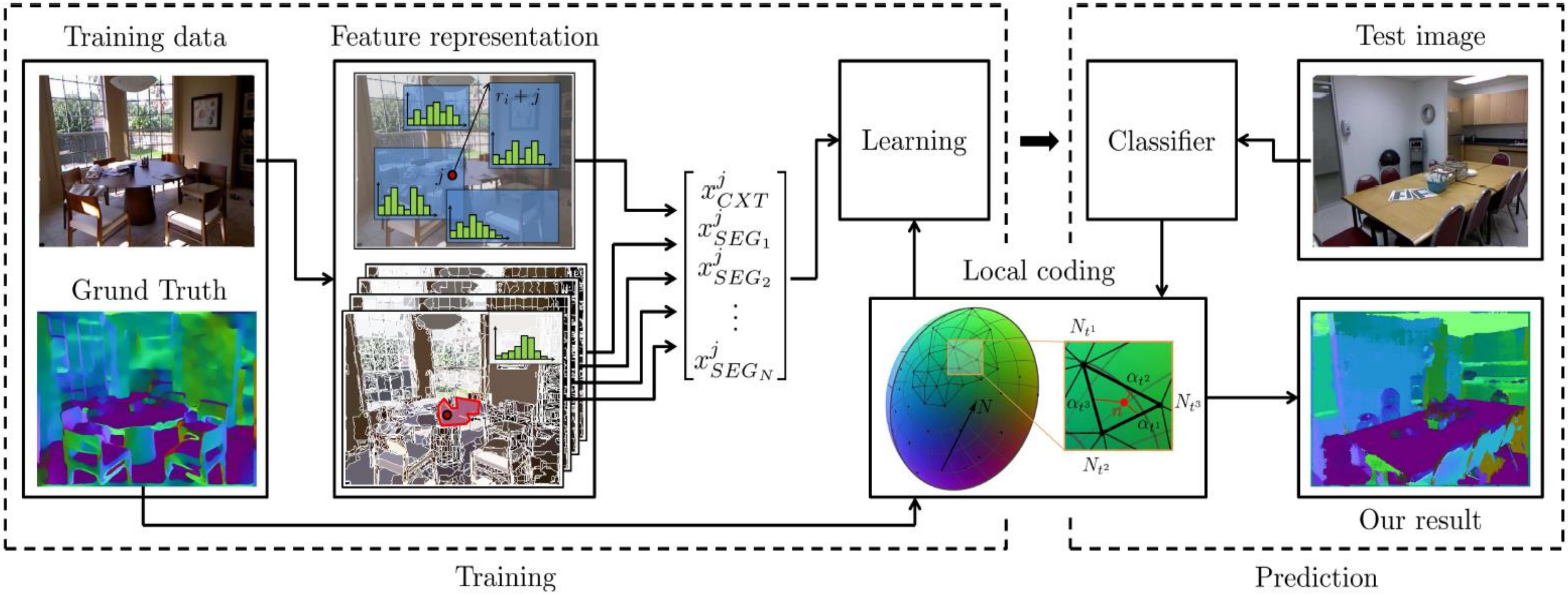
Joint Learning



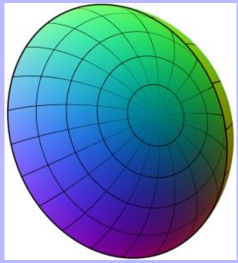
To simplify regression problem

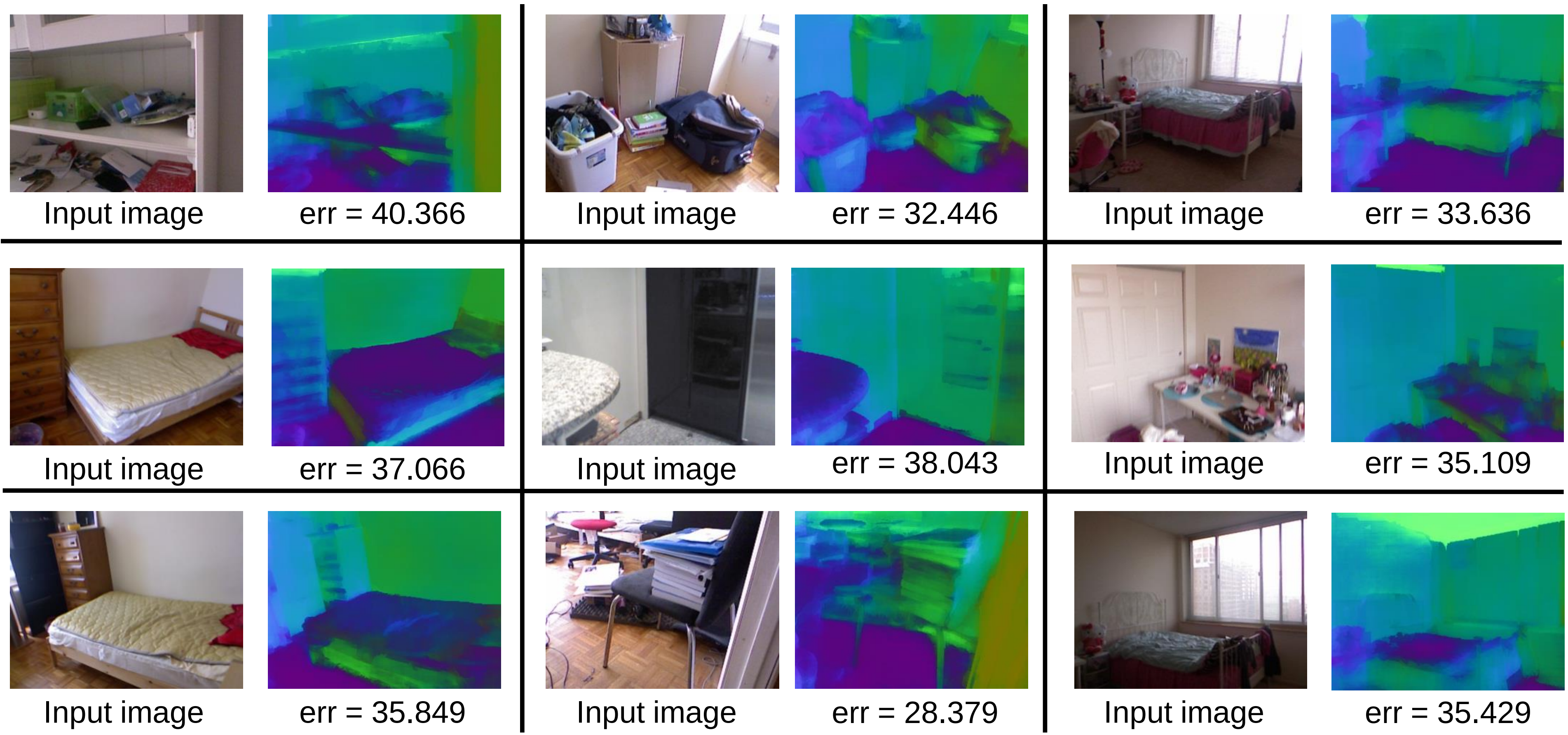
- Normals clustered using K-means clustering
- Each represented as weighted sums of cluster centres using local coding
- Learning formulated as a regression into local coding coordinates

Pipeline of our Method

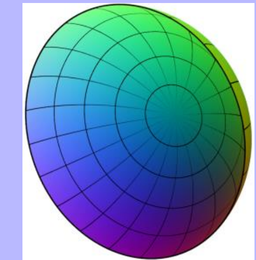


RMRC Challenge Results





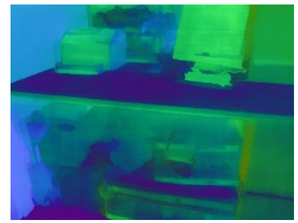
RMRC Challenge Results



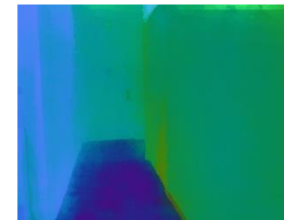

Input image

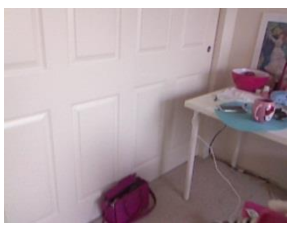

err = 37.688


Input image

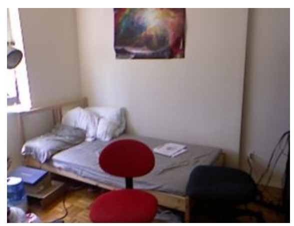

err = 40.784


Input image

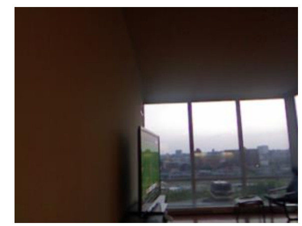

err = 51.897


Input image

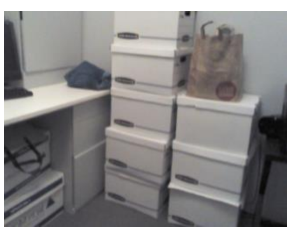

err = 28.216


Input image


err = 32.034


Input image


err = 68.038

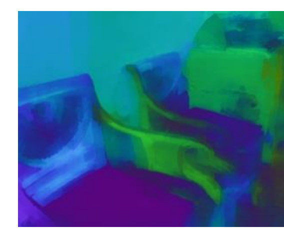

Input image


err = 33.174


Input image


err = 41.131

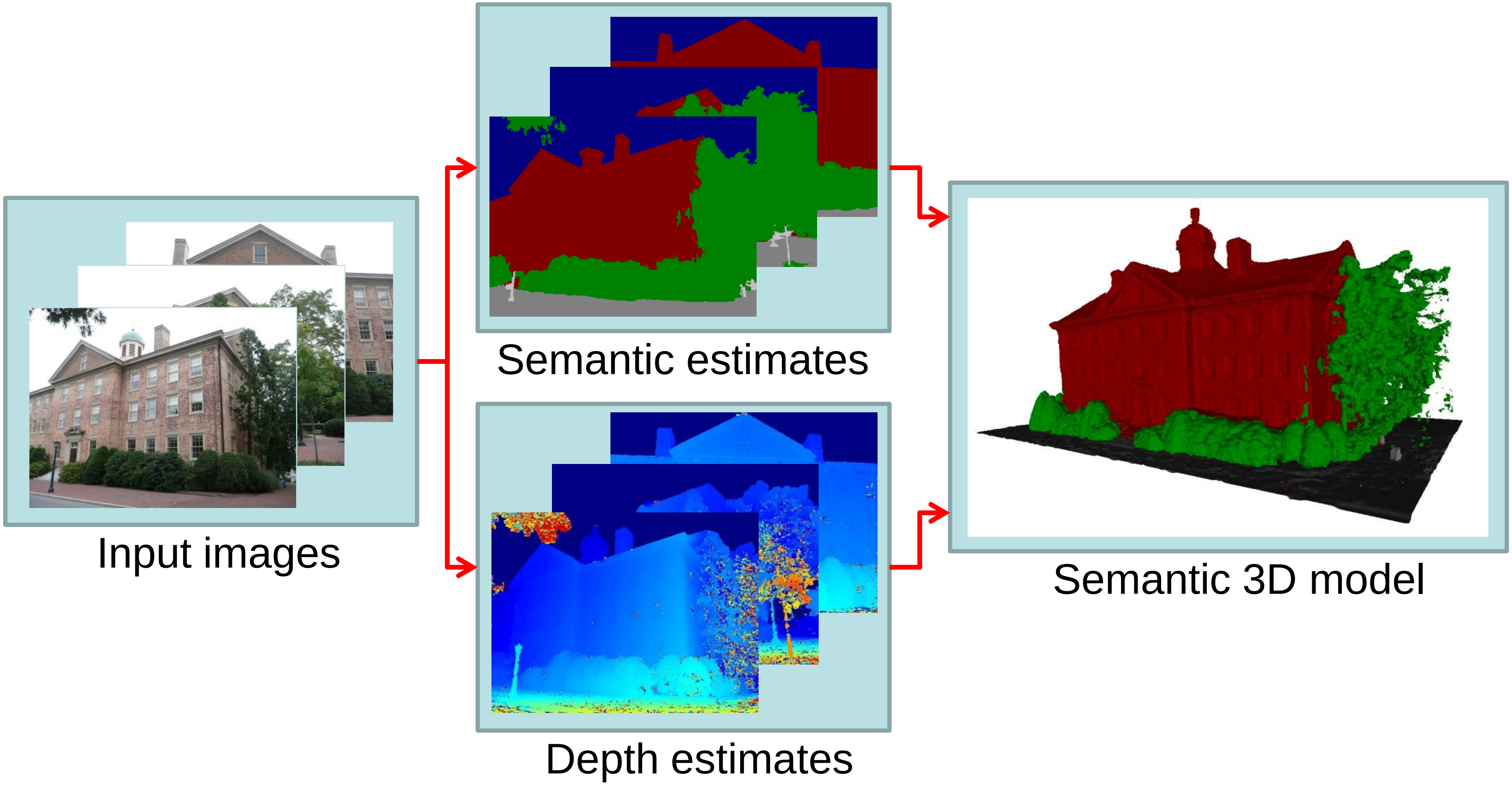

Input image


err = 38.873

Schedule

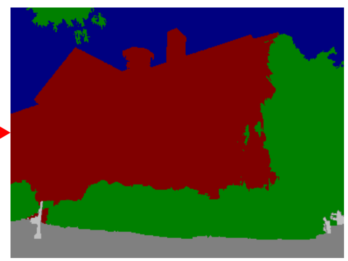
- Introduction
    - Discrete MRF Optimization using Graph Cuts
    - Classifiers for Semantic 3D Modelling
- Higher Order MRFs with Ray Potentials
    - Discrete Formulation
    - Continuous Relaxation

Semantic 3D Reconstruction

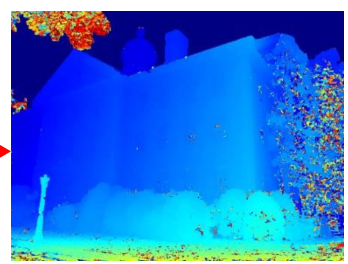


Semantic 3D Reconstruction

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

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

Semantic 3D Reconstruction

Volumetric formulation

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

Ray potentials   Pairwise regularizer

Semantic 3D Reconstruction

Volumetric formulation

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

Ray potentials   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

$$E(\mathbf{x}) = \underbrace{\boxed{\sum_{r\in\mathcal{R}} \psi_r(\mathbf{x}^r)}}_{\text{Ray potentials}} + \underbrace{\boxed{\sum_{(i,j)\in\mathcal{E}} \psi_p(x_i, x_j)}}_{\text{Pairwise regularizer}}$$

We try to solve the right problem!

Semantic 3D Reconstruction

Volumetric formulation

$$E(\mathbf{x}) = \underbrace{\boxed{\sum_{r \in \mathcal{R}} \psi_r(\mathbf{x}^r)}}_{\text{Ray potentials}} + \underbrace{\boxed{\sum_{(i,j) \in \mathcal{E}} \psi_p(x_i, x_j)}}_{\text{Pairwise regularizer}}$$

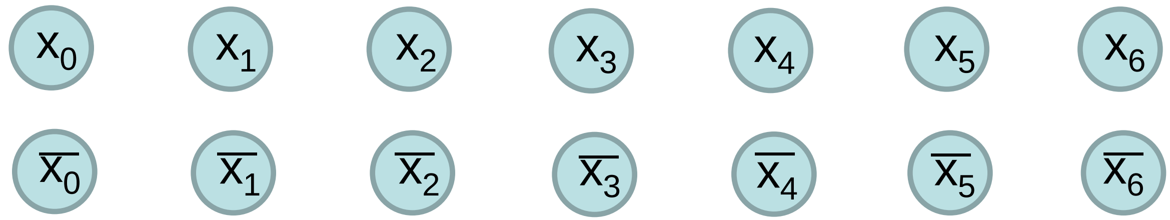
Cost based on the first occupied voxel along the ray

$$\psi_r(\mathbf{x}^r) = \phi_r(\underbrace{\boxed{K^r}}_{\text{depth}}, \underbrace{\boxed{x^r_{K^r}}}_{\text{label}})$$

$$K^r = \begin{cases} \min(i \mid x_i^r \neq l_f) & \text{if } \exists x_i^r \neq \overset{\text{freespace}}{\boxed{l_f}} \\ N_r & \text{otherwise} \end{cases}$$

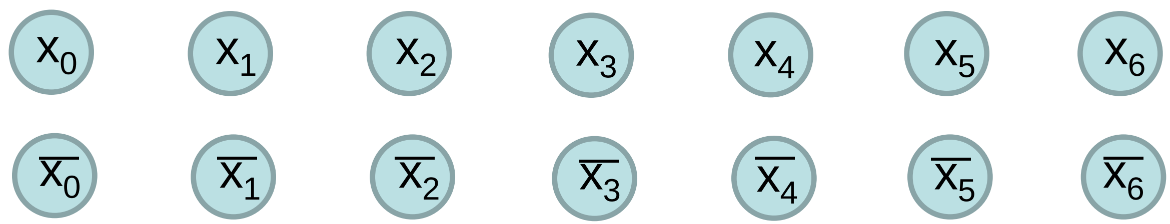
Two-label problem

Discrete formulation using QPBO relaxation



Two-label problem

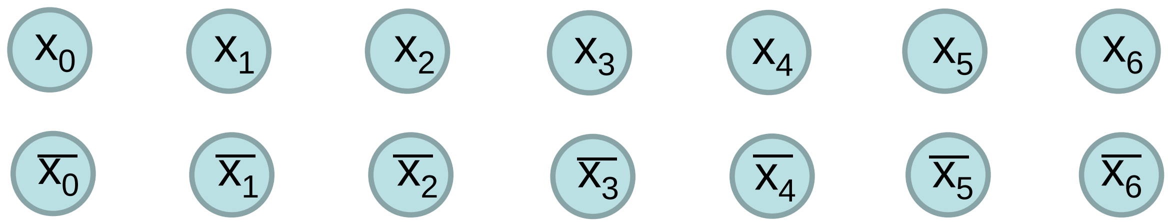
Discrete formulation using QPBO relaxation



Our goal is to find : $\psi_r(\mathbf{x}^r) = \min_{\mathbf{z}} \psi_q(\mathbf{x}^r, \overline{\mathbf{x}}^r, \mathbf{z})$

Two-label problem

Discrete formulation using QPBO relaxation



Our goal is to find : $\psi_r(\mathbf{x}^r) = \min_{\mathbf{z}} \psi_q(\mathbf{x}^r, \overline{\mathbf{x}}^r, \mathbf{z})$

such that $\psi_q(\mathbf{x}^r, \overline{\mathbf{x}}^r, \mathbf{z})$ 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 $\psi_q(\mathbf{x}^r, \overline{\mathbf{x}}^r, \mathbf{z})$ we do these steps:

1) Polynomial representation of the ray potential
2) Transformation into submodular function over **x** and $\overline{\mathbf{x}}$
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:

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

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:

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

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

We want to transform the potential into:

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

Polynomial representation of the ray potential

Two-label ray potential takes the form:

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

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

We want to transform the potential into:

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

Plugging it in:

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

thus

$$\begin{aligned} k^r &= \phi_r(0) \\ c_i^r &= \phi_r(i+1) - \phi_r(i) \end{aligned}$$

Transformation into a submodular function

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

Transformation into a submodular function

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

For $c_i^r > 0$:

$$c_i^r \prod_{j=0}^{i} x_j^r = c_i^r (1 - \overline{x}_i^r) \prod_{j=0}^{i-1} x_j^r = -c_i^r \overline{x}_i^r \prod_{j=0}^{i-1} x_j^r + c_i^r \prod_{j=0}^{i-1} x_j^r$$

Transformation into a submodular function

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

For $c_i^r > 0$:

$$c_i^r \prod_{j=0}^{i} x_j^r = c_i^r (1 - \overline{x}_i^r) \prod_{j=0}^{i-1} x_j^r = -c_i^r \overline{x}_i^r \prod_{j=0}^{i-1} x_j^r + c_i^r \prod_{j=0}^{i-1} x_j^r$$

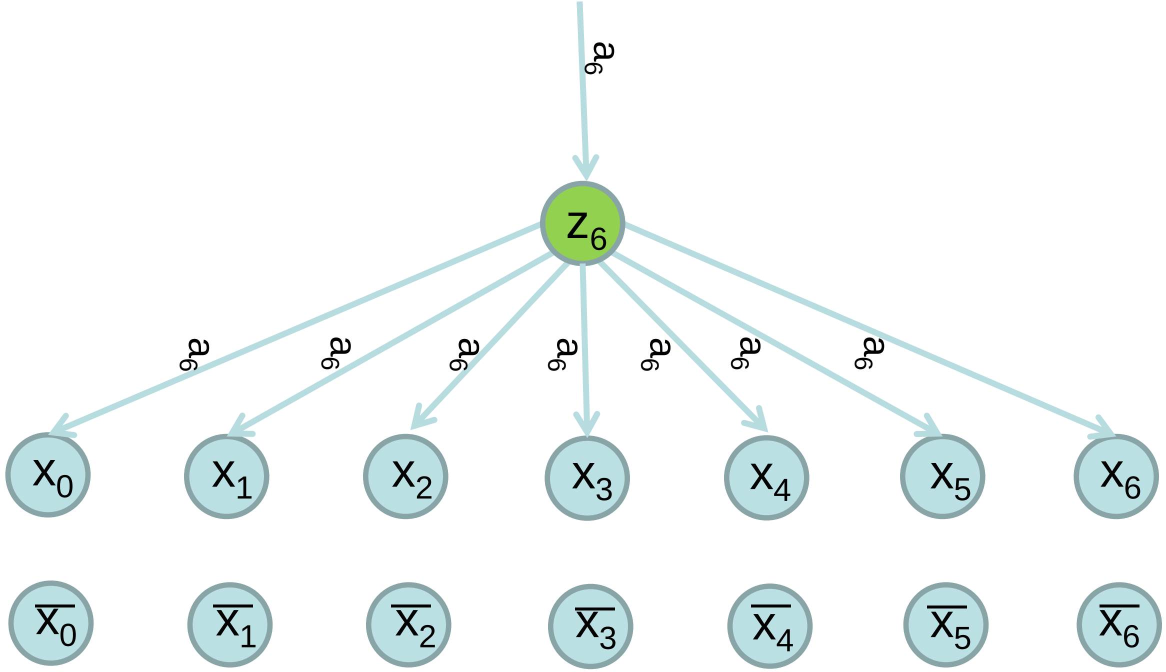
Starting from the last term, we can iteratively transform:

$$\sum_{i=0}^{N_r-1} c_i^r \prod_{j=0}^{i} x_j^r = \sum_{i=0}^{N_r-1} \left(- a_i^r \prod_{j=0}^{i} x_j^r - b_i^r \overline{x}_i^r \prod_{j=0}^{i-1} x_j^r \right)$$

Pairwise graph construction

Standard graph constructions (Freedman CVPR05) for negative products:

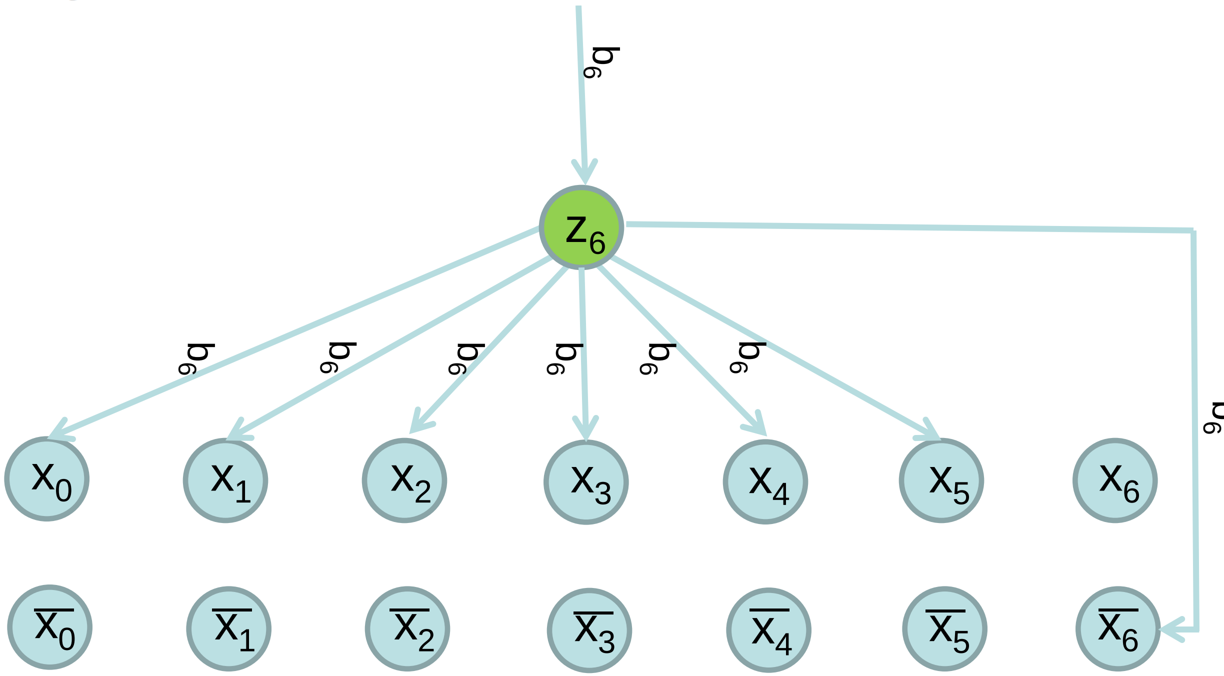
$$-a_i^r \prod_{j=0}^{i} x_j^r = a_i^r \min_{z_i} \left(-z_i + \sum_{j=0}^{i} z_i (1 - x_j^r) \right)$$



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:

$$-a_i^r \prod_{j=0}^{i} x_j^r = a_i^r \min_{z_i} \Big(-z_i + \sum_{j=0}^{i} z_i (1 - x_j^r) \Big)$$

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

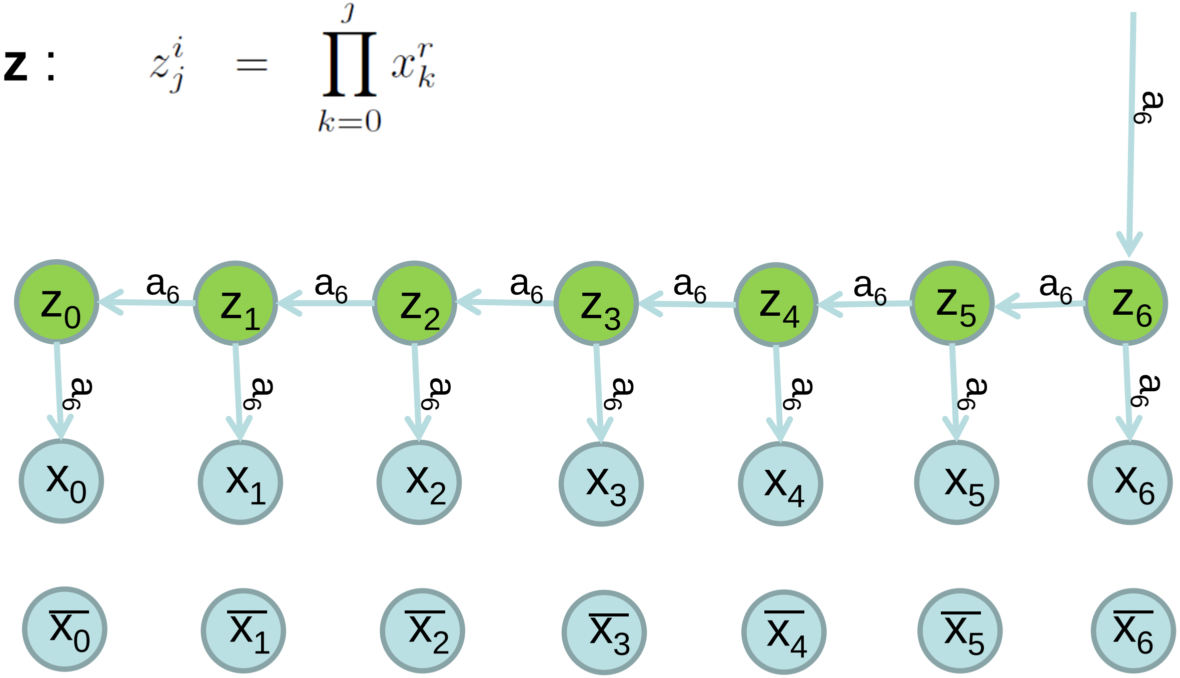
Leads to a quadratic growth of the number of edges!

Pairwise graph construction

Non-standard graph constructions for negative products:

$$-a_i^r \prod_{j=0}^{i} x_j^r \quad = \quad a_i^r \min_{\mathbf{z}_i} \left(- z_i^i + z_i^i(1 - x_i^r) \; + \; \sum_{j=0}^{i-1} (z_{j+1}^i (1 - z_j^i) + z_j^i (1 - x_j^r)) \right)$$

Optimal **z** : $\quad z_j^i \quad = \quad \prod_{k=0}^{j} x_k^r$

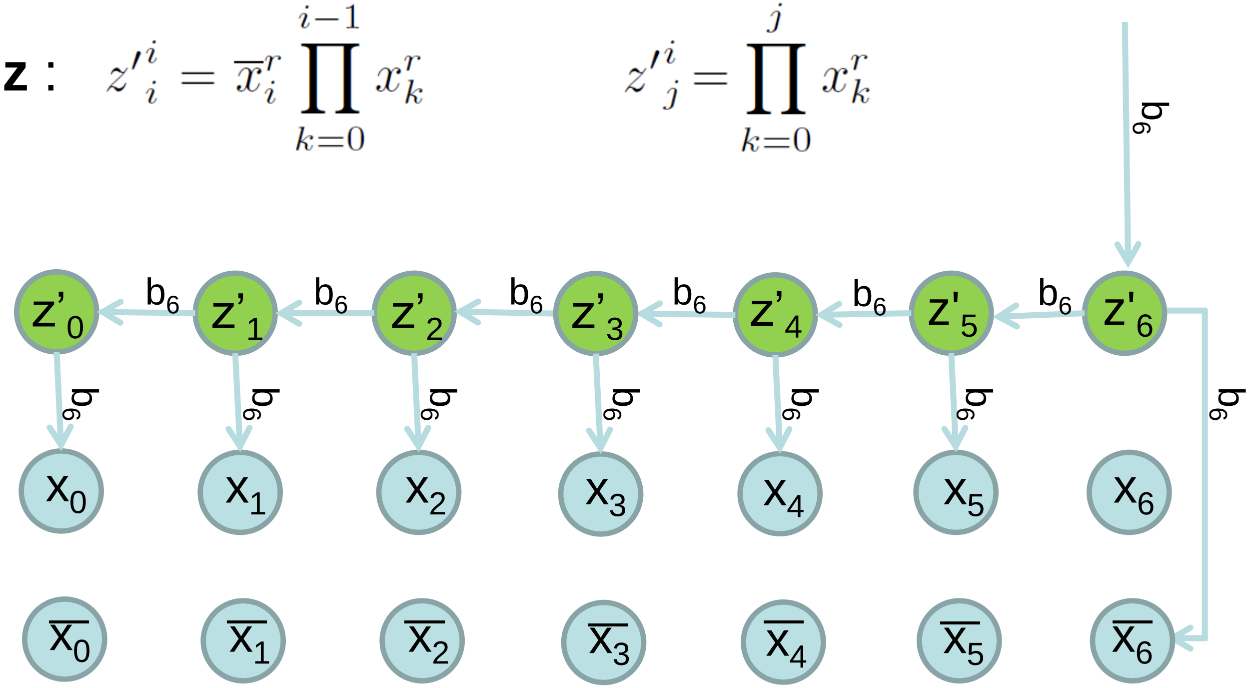


Pairwise graph construction

Non-standard graph constructions for negative products:

$$-b_i^r \overline{x}_i^r \prod_{j=0}^{i-1} x_j^r = \quad b_i^r \min_{\mathbf{z'}_i} \left(- {z'}_i^i + {z'}_i^i (1 - \overline{x}_i^r) + \sum_{j=0}^{i-1} ({z'}_{j+1}^i (1 - {z'}_i^j) + {z'}_j^i (1 - x_j^r)) \right)$$

Optimal **z** : $\quad {z'}_i^i = \overline{x}_i^r \prod_{k=0}^{i-1} x_k^r \qquad {z'}_j^i = \prod_{k=0}^{j} x_k^r$



Merging theorem

(Ramalingam12)

If $$\psi(\mathbf{x}) = \min_{\mathbf{z}} \psi_q(\mathbf{x}, z_i, ..z_j)$$

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

Merging theorem

(Ramalingam12)

If $\psi(\mathbf{x}) = \min_{\mathbf{z}} \psi_q(\mathbf{x}, z_i, ..z_j)$

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

then

$$\psi(\mathbf{x}) = \min_{\mathbf{z}} \psi_q(\mathbf{x}, z_i, ..z_i)$$

Pairwise graph construction

Non-standard graph constructions for negative products:

$$-a_i^r \prod_{j=0}^{i} x_j^r \quad = \quad a_i^r \min_{\mathbf{z}_i} \left(- z_i^i + z_i^i(1 - x_i^r) \; + \; \sum_{j=0}^{i-1} (z_{j+1}^i (1 - z_j^i) + z_j^i (1 - x_j^r)) \right)$$

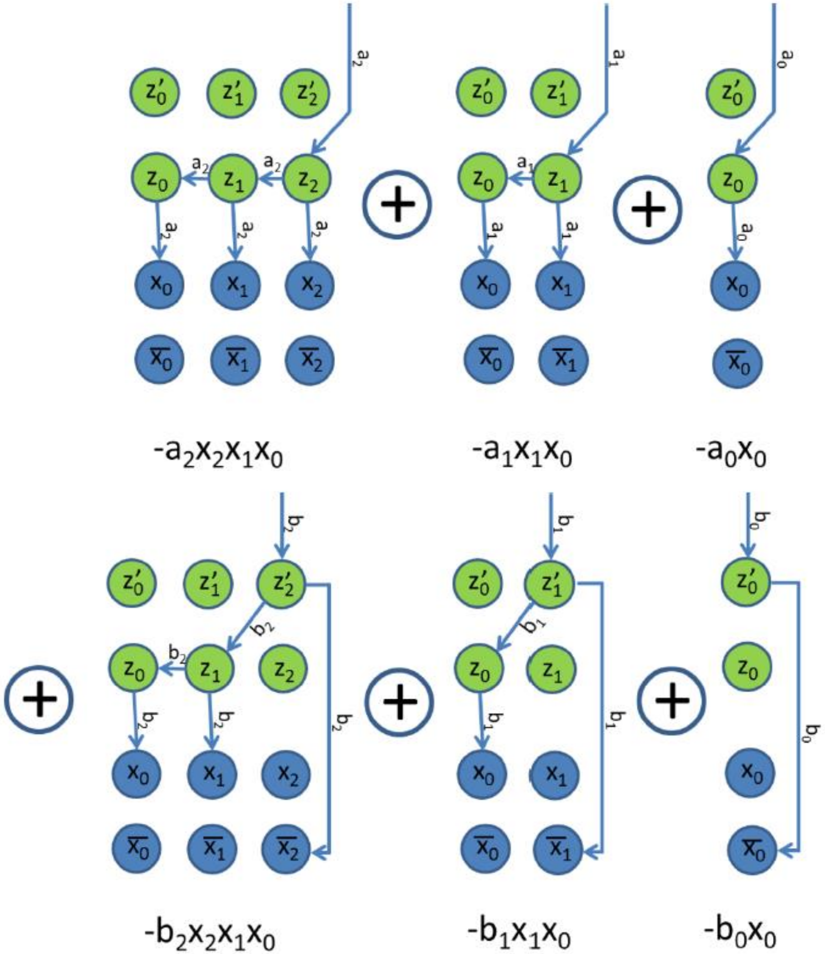
Optimal **z** : $\quad z_j^i \quad = \quad \prod_{k=0}^{j} x_k^r$

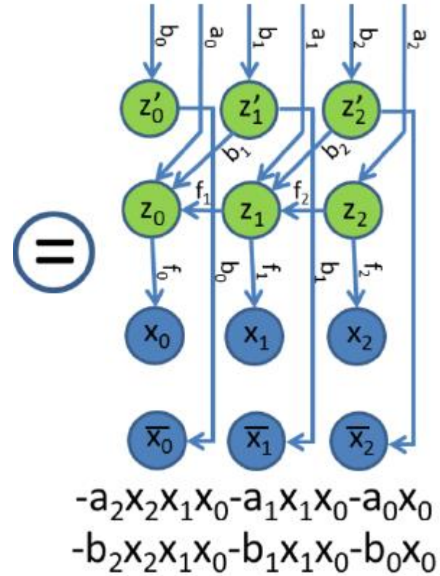
$$-b_i^r \overline{x}_i^r \prod_{j=0}^{i-1} x_j^r = \quad b_i^r \min_{\mathbf{z'}_i} \left(- z'^i_i + z'^i_i (1 - \overline{x}_i^r) \; + \sum_{j=0}^{i-1} (z'^i_{j+1} (1 - z'^j_i) + z'^i_j (1 - x_j^r)) \right)$$

Optimal **z** : $\quad z'^i_i = \overline{x}_i^r \prod_{k=0}^{i-1} x_k^r \qquad\qquad z'^i_j = \prod_{k=0}^{j} x_k^r$

$$\forall j, k \in \{0, .., i\} : z_i^j = z_i^k \qquad\qquad \forall j, k \in \{0, .., i-1\} : z'^j_i = z_i^k$$

Pairwise graph construction





$$f_0 = a_2 + a_1 + a_0 + b_2 + b_1$$

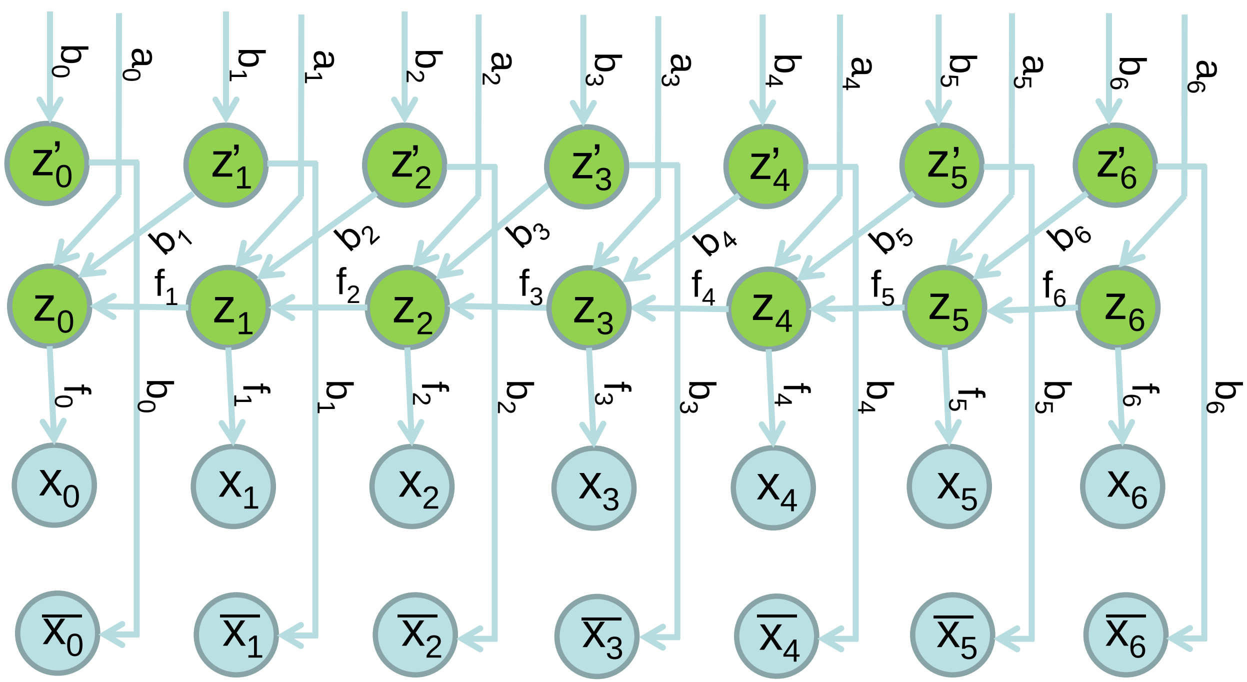
$$f_1 = a_2 + a_1 + b_2$$

$$f_2 = a_2$$

Pairwise graph construction

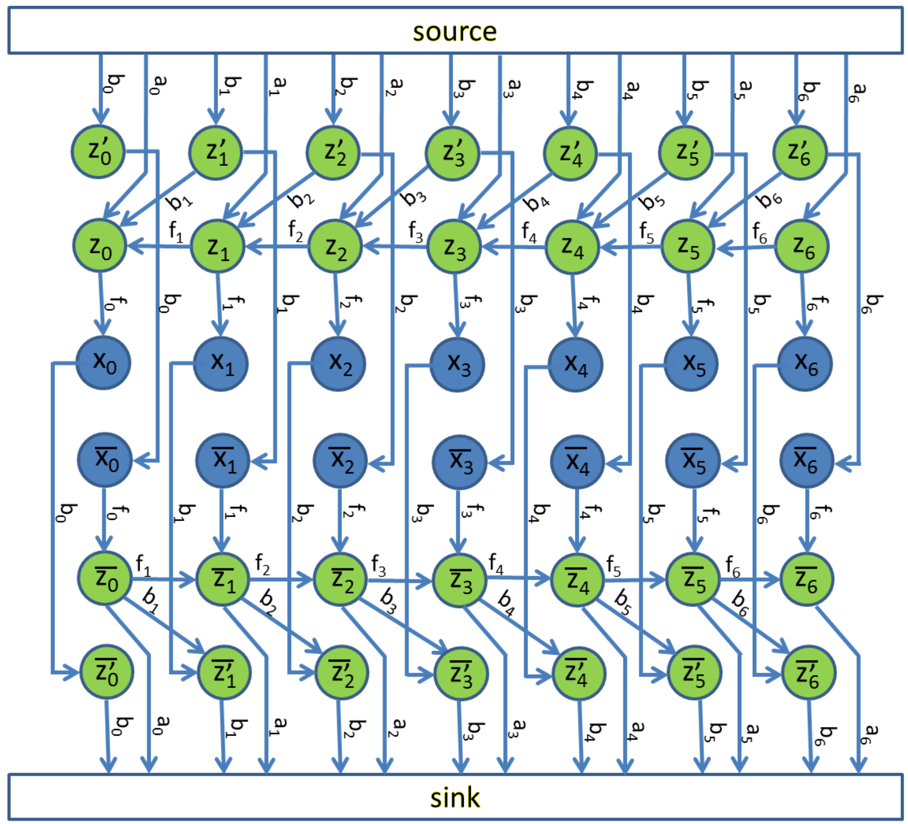
$$\psi_r(\mathbf{x}^r) \;=\; \min_{\mathbf{z},\mathbf{z}'} \left(\sum_{i=0}^{N_r-1} \left(-a_i^r z_i - b_i^r z_i' + f_i^r (1-z_i) x_i^r + b_i^r (1-z_i)\overline{x}_i^r \right) + \sum_{i=1}^{N_r-2} \Big((f_{i+1}^r (1-z_{i+1}) z_i + b_i^r (1 - z_{i+1}') z_i \Big) \right)$$

$$f_i^r = \sum_i^{N_r-1} a_j^r + \sum_{i+1}^{N_r-1} b_j^r$$



Symmetrization of the graph

$$\psi_r(\mathbf{x}^r) \quad = \quad \frac{1}{2}\left(\min_{\mathbf{z}} \psi^r(\mathbf{x}, \overline{\mathbf{x}}, \mathbf{z}) \;+\; \min_{\overline{\mathbf{z}}} \psi^r(\mathbf{1}-\overline{\mathbf{x}}, \mathbf{1}-\mathbf{x}, \mathbf{1}-\overline{\mathbf{z}})\right)$$



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

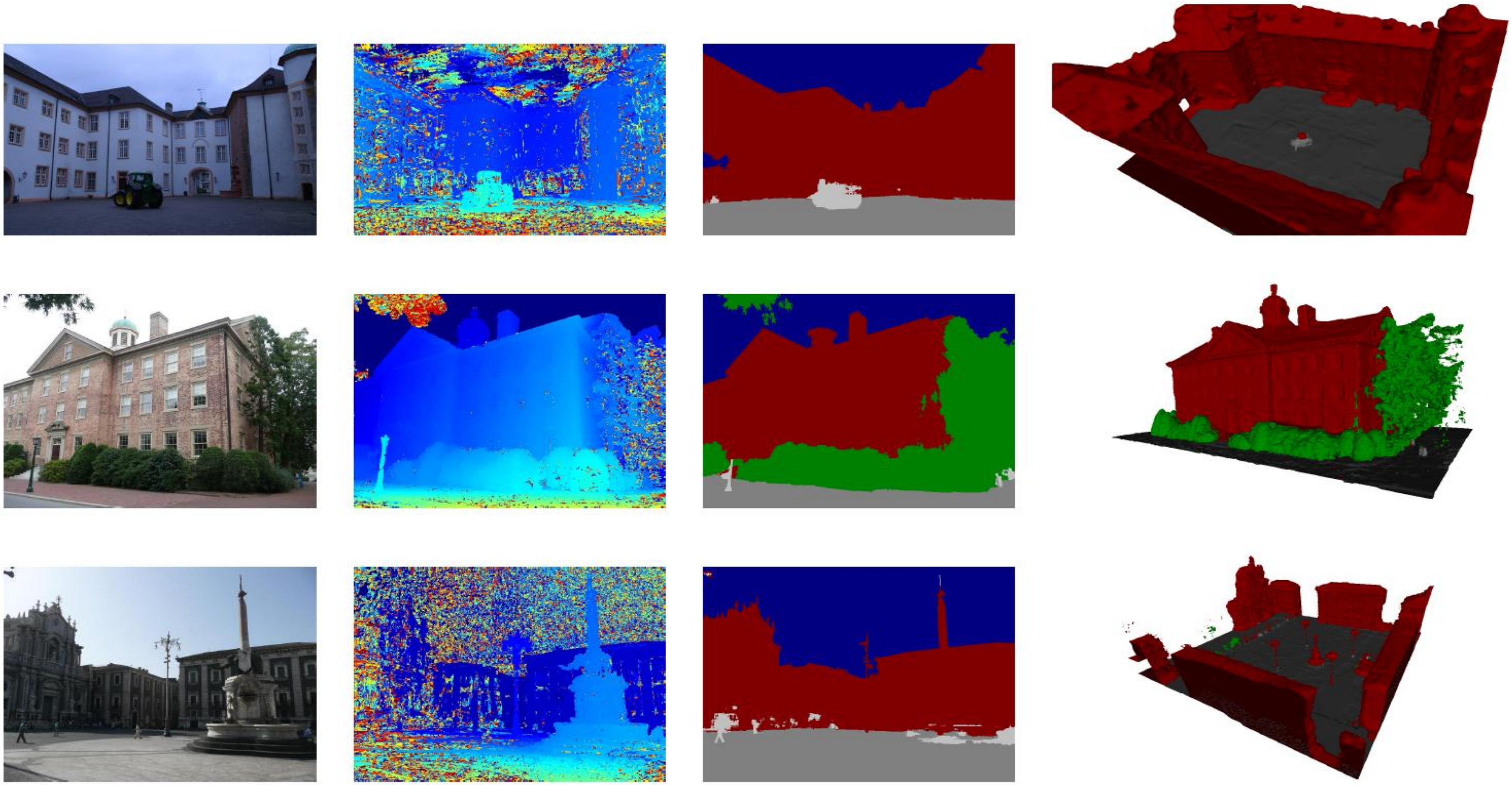
$$\phi_r(i,l) = \Big(\big(\underbrace{\boxed{\lambda_{sem} C(l)}}_{\text{Semantic cost}} + \underbrace{\boxed{\lambda_{dep} C(d(i))}}_{\text{Depth cost}} \Big) d(i)^2$$

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

For the top n matches :

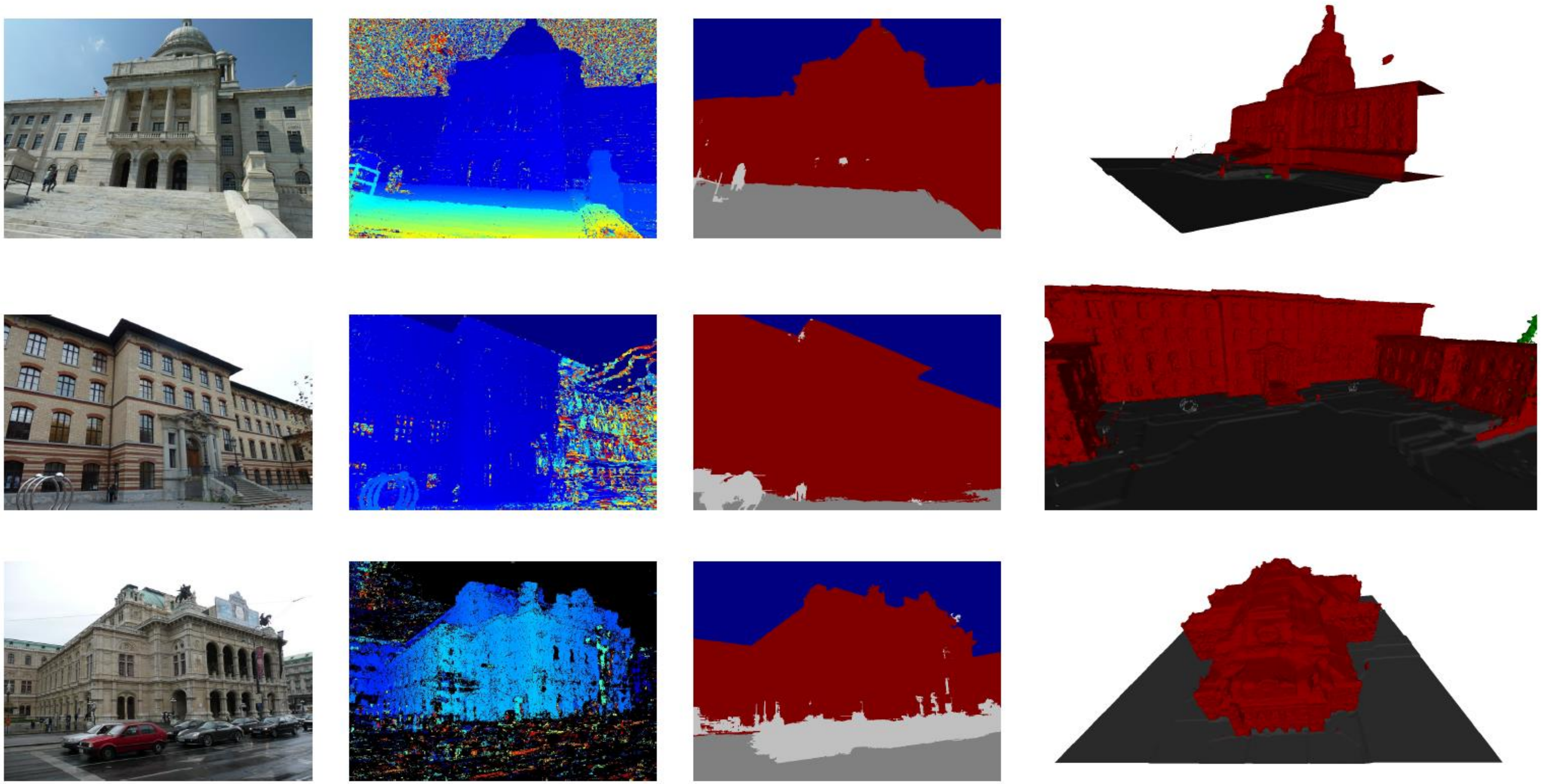
$$C(d(i)) = \begin{cases} w_n(-1 + \frac{|d(i) - d^n_{top}|}{\Delta}) & \text{if } |d(i) - d^n_{top}| \leq \Delta \\ 0 & \text{otherwise.} \end{cases}$$

Results



Input Depth Semantics 3D model

Results



Input Depth Semantics 3D model

Results

South Building Data Set

Results

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

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

Continuous approach possible ?

Continuous Formulation

$$\psi_r(\mathbf{x}_r) = \left(\sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \left(\min_{j \leq i-1} x_{s_{rj}}^{f} \right) x_{s_{ri}}^{\ell} \right) + c_r^f \min_{j \leq N_r} x_{s_{rj}}^{f}$$

Continuous Formulation

$$\psi_r(\mathbf{x}_r) = \left(\sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \left(\min_{j \leq i-1} x_{s_{rj}}^{f} \right) x_{s_{ri}}^{\ell} \right) + c_r^f \min_{j \leq N_r} x_{s_{rj}}^{f}$$

The last term can be dropped by:

$$c_{ri}^{\ell} \leftarrow c_{ri}^{\ell} - c_r^f \qquad\qquad c_r^f \leftarrow 0$$

Continuous Formulation

$$\psi_r(\mathbf{x}_r) = \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \Big(\min_{j \le i-1} x_{s_{rj}}^{f} \Big) x_{s_{ri}}^{\ell}$$

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

Continuous Formulation

$$\psi_r(\mathbf{x}_r) = \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \Big(\min_{j \le i-1} x_{s_{rj}}^{f} \Big) x_{s_{ri}}^{\ell}$$

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

$$\psi_r(\mathbf{x}_r) = \sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_i^{\ell} y_i^{\ell}$$

where $c_i^f = 0$

Continuous Formulation

$$\psi_r(\mathbf{x}_r) = \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \Big(\min_{j \leq i-1} x_{s_{rj}}^{f} \Big) x_{s_{ri}}^{\ell}$$

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

$$\psi_r(\mathbf{x}_r) = \boxed{\sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_i^{\ell} y_i^{\ell}}$$

where $c_i^{f} = 0$

convex for $c_i^{\ell} \leq 0$

Continuous Formulation

$$\psi_r(\mathbf{x}_r) = \sum_{\ell \in \mathcal{L} \setminus \{f\}} \sum_{i=0}^{N_r} c_{ri}^{\ell} \left(\min_{j \le i-1} x_{s_{rj}}^{f} \right) x_{s_{ri}}^{\ell}$$

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

$$\psi_r(\mathbf{x}_r) = \boxed{\sum_{\ell \in \mathcal{L}} \sum_{i=0}^{N} c_i^{\ell} y_i^{\ell}}$$

$$y_i^{\ell} \le y_{i-1}^{f} \quad y_i^{\ell} \le x_{s_i}^{\ell} \quad y_i^{\ell} \ge 0$$

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

convex for $c_i^{\ell} \le 0$

Continuous Formulation

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

Continuous Formulation

Can we make $\boldsymbol{c_i^\ell} \leq \mathbf{0}$ **?** Yes!

Continuous Formulation

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

First, we notice : $$y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell = 0$$

Continuous Formulation

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

First, we notice : $$y_{i-1}^{f} - \sum_{\ell \in \mathcal{L}} y_i^{\ell} = 0$$

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 $c_i^{\ell} \leq 0$?

First, we notice : $y_{i-1}^{f} - \sum_{\ell \in \mathcal{L}} y_i^{\ell} = 0$

The cost function does not change by adding:

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

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

$$\left.\begin{aligned} c_{i-1}^{f} &\leftarrow c_{i-1}^{f} + \max_{\ell' \in \mathcal{L}} c_i^{\ell'} \\ c_i^{\ell} &\leftarrow c_i^{\ell} - \max_{\ell' \in \mathcal{L}} c_i^{\ell'} \end{aligned}\right. \quad \forall \ell \in \mathcal{L}$$

Continuous Formulation

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

First, we notice : $y_{i-1}^f - \sum_{\ell \in \mathcal{L}} y_i^\ell = 0$

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

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

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

$c_i^\ell \geq 0$

Integer Formulation

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

$$\text{s.t.} \quad y_i^\ell \leq y_{i-1}^f \quad y_i^\ell \leq x_{s_i}^\ell \qquad \forall \ell \in \mathcal{L}, \forall i$$

$$\sum_{\ell \in \mathcal{L}} x_s^\ell = 1 \quad y_i^\ell \geq 0 \qquad \forall s \in \Omega$$

$$x_s^\ell \in \{0, 1\}$$

Convex relaxation

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

$$\text{s.t.} \quad y_i^\ell \leq y_{i-1}^f \quad y_i^\ell \leq x_{s_i}^\ell \qquad \forall \ell \in \mathcal{L}, \forall i$$

$$\sum_{\ell\in\mathcal{L}} x_s^\ell = 1 \quad y_i^\ell \geq 0 \qquad \forall s \in \Omega$$

$$0 \leq x_s^\ell \leq 1$$

Convex relaxation

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

$$\text{s.t.} \quad y_i^\ell \le y_{i-1}^f \quad y_i^\ell \le x_{s_i}^\ell \qquad \forall \ell \in \mathcal{L}, \forall i$$

$$\sum_{\ell\in\mathcal{L}} x_s^\ell = 1 \quad y_i^\ell \ge 0 \qquad \forall s \in \Omega$$

$$0 \le x_s^\ell \le 1$$

Will it work **?**

Convex relaxation

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

$$\text{s.t.} \quad y_i^\ell \leq y_{i-1}^f \quad y_i^\ell \leq x_{s_i}^\ell \qquad \forall \ell \in \mathcal{L}, \forall i$$

$$\sum_{\ell\in\mathcal{L}} x_s^\ell = 1 \quad y_i^\ell \geq 0 \qquad \forall s \in \Omega$$

$$0 \leq x_s^\ell \leq 1$$

Will it work ? Unfortunately not

Convex relaxation

$$(\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y})} \left(-2y_0^o - 3y_1^o - 2y_2^o\right)$$

$$\begin{aligned} \text{s.t.}\quad & y_i^o \leq y_{i-1}^f,\ y_i^f \leq y_{i-1}^f, \\ & y_i^o \leq x_i^o,\ y_i^f \leq 1 - x_i^o, \\ & x_i^o \in [0, 1],\quad \forall i. \end{aligned}$$

Convex relaxation

$$(\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y})} \left(-2y_0^o - 3y_1^o - 2y_2^o\right)$$

$$\begin{aligned} \text{s.t.} \quad & y_i^o \leq y_{i-1}^f,\ y_i^f \leq y_{i-1}^f, \\ & y_i^o \leq x_i^o,\ y_i^f \leq 1 - x_i^o, \\ & x_i^o \in [0,1], \quad \forall i. \end{aligned}$$

Desired solution

$$\begin{aligned} & x_0^o = 0,\ x_1^o = 1,\ x_2^o = 0 \\ & y_0^o = 0,\ y_1^o = 1,\ y_2^o = 0 \\ & y_0^f = 1,\ y_1^f = 0,\ y_2^f = 0 \end{aligned}$$

Convex relaxation

$$(\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y})} \left(-2y_0^o - 3y_1^o - 2y_2^o\right)$$

$$\begin{aligned} \text{s.t.} \quad & y_i^o \le y_{i-1}^f, \; y_i^f \le y_{i-1}^f, \\ & y_i^o \le x_i^o, \; y_i^f \le 1 - x_i^o, \\ & x_i^o \in [0,1], \quad \forall i. \end{aligned}$$

Desired solution

$$x_0^o = 0, \; x_1^o = 1, \; x_2^o = 0$$
$$y_0^o = 0, \; y_1^o = 1, \; y_2^o = 0$$
$$y_0^f = 1, \; y_1^f = 0, \; y_2^f = 0$$

Global optimum

$$x_0^o = x_1^o = x_2^o = \mathbf{0.5}$$
$$y_0^o = y_1^o = y_2^o = \mathbf{0.5}$$
$$y_0^f = y_1^f = y_2^f = \mathbf{0.5}$$

Convex relaxation

$$(\mathbf{x}^*, \mathbf{y}^*) = \arg\min_{(\mathbf{x},\mathbf{y})} \left(-2y_0^o - 3y_1^o - 2y_2^o\right)$$

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

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

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

Non-convex Formulation

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

$$\text{s.t.} \quad y_i^\ell \le y_{i-1}^f \quad y_i^\ell \le x_{s_i}^\ell \qquad \forall \ell \in \mathcal{L}, \forall i$$

$$\sum_{\ell\in\mathcal{L}} x_s^\ell = 1 \quad y_i^\ell \ge 0 \qquad \forall s \in \Omega$$

$$0 \le x_s^\ell \le 1$$

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

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

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

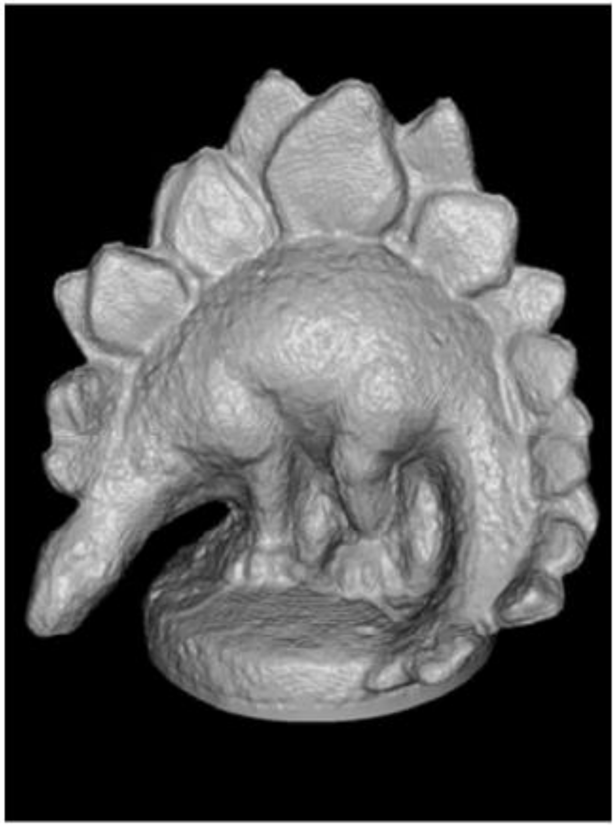
by

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

where

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

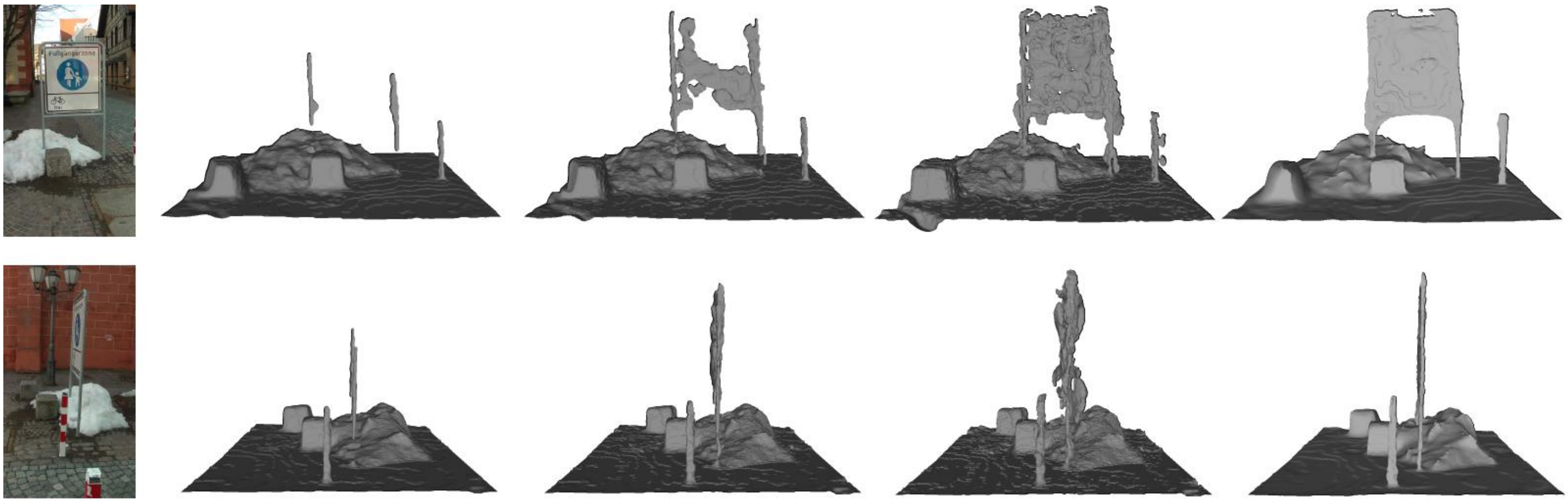
Results on Middlebury dataset



Results

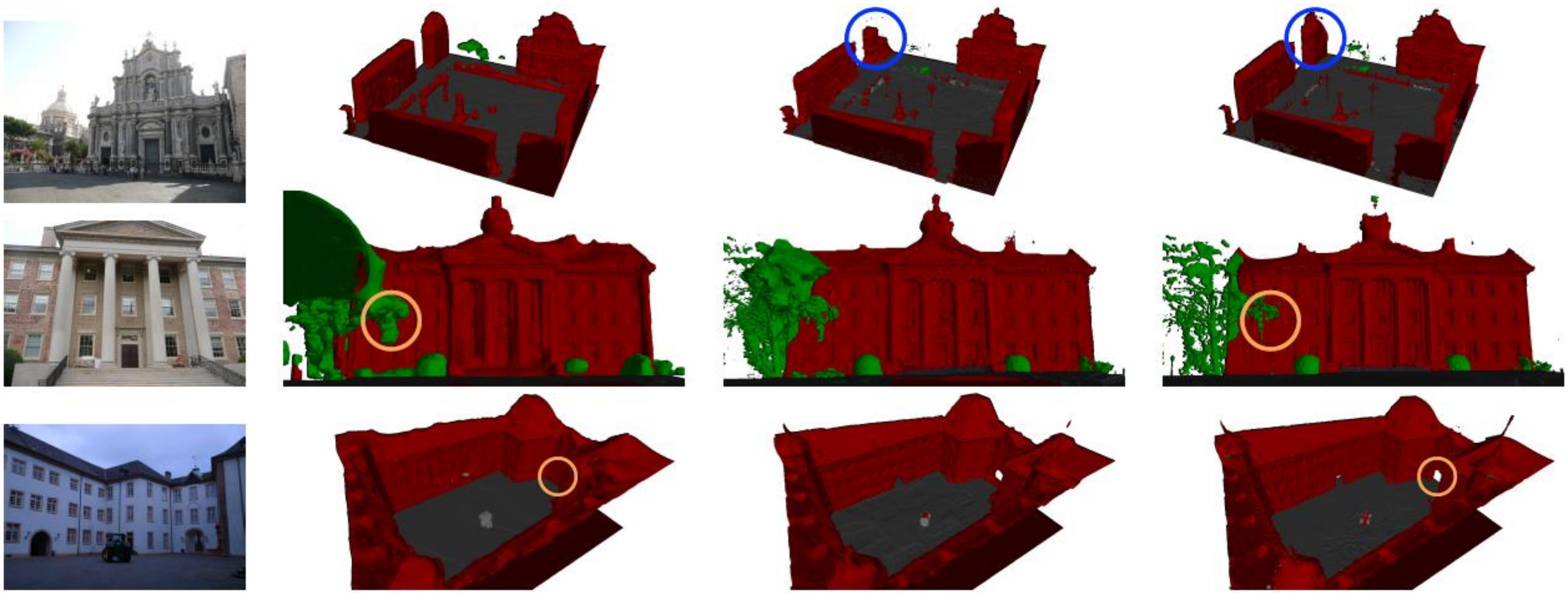
Sort By	Temple Full 312 views Acc [mm]	Temple Full 312 views Comp [%]	Temple Ring 47 views Acc [mm]	Temple Ring 47 views Comp [%]	Temple Sparse 16 views Acc [mm]	Temple Sparse 16 views Comp [%]	Dino Full 363 views Acc [mm]	Dino Full 363 views Comp [%]	Dino Ring 48 views Acc [mm]	Dino Ring 48 views Comp [%]	Dino Sparse 16 views Acc [mm]	Dino Sparse 16 views Comp [%]
Savinov	0.41	99.7	0.5	99.5	0.69	97.8	0.26	99.8	0.25	99.9	0.34	99.7
Furukawa 3	0.49	99.6	0.47	99.6	0.63	99.3	0.33	99.8	0.28	99.8	0.37	99.2
DCV			0.73	98.2	0.66	97.3			0.28	100	0.3	100
Galliani	0.39	99.2	0.48	99.1	0.53	97.0	0.31	99.9	0.3	99.4	0.38	98.6
ECCV2016_624	0.37	98.9	0.49	97.6	1.27	39.2	0.26	97.8	0.31	99.5	0.28	98.1
3DV2014_25			0.51	96.4	1.23	90.2			0.32	97.3	0.42	96.7
Furukawa 2	0.54	99.3	0.55	99.1	0.62	99.2	0.32	99.9	0.33	99.6	0.42	99.2
Schroers	0.57	99.1	0.64	96.4	2.12	62.9	0.33	99.7	0.33	99.7	0.54	98.6
Kostrikov			0.57	99.1	0.79	95.8			0.35	99.6	0.37	99.3
Yichao Li	0.46	96.4	0.56	89.6			0.4	94.9	0.37	80.6		
Song			0.61	98.3					0.38	99.4	0.54	95.5
Khuboni			0.67	98.3					0.38	99.5		
Zhu			0.4	99.2	0.45	95.7			0.38	98.3	0.48	95.4

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 ?