GraphCut-based Optimisation for Computer Vision

Ľubor Ladický

UNIVERSITY OF OXFORD

Overview

- Motivation
- Min-Cut / Max-Flow (Graph Cut) Algorithm
- Markov and Conditional Random Fields
- Random Field Optimisation using Graph Cuts
    - Submodular vs. Non-Submodular Problems
    - Pairwise vs. Higher Order Problems
    - 2-Label vs. Multi-Label Problems
- Recent Advances in Random Field Optimisation
- Conclusions

Overview

- Motivation
- Min-Cut / Max-Flow (Graph Cut) Algorithm
- Markov and Conditional Random Fields
- Random Field Optimisation using Graph Cuts
    - Submodular vs. Non-Submodular Problems
    - Pairwise vs. Higher Order Problems
    - 2-Label vs. Multi-Label Problems
- Recent Advances in Random Field Optimisation
- Conclusions

Image Labelling Problems

Assign a label to each image pixel

Geometry Estimation

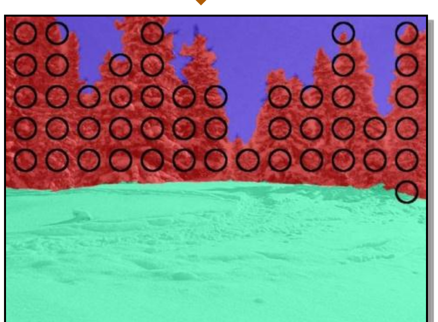
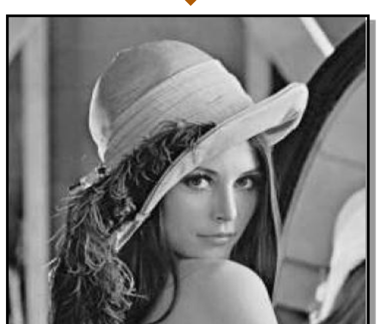
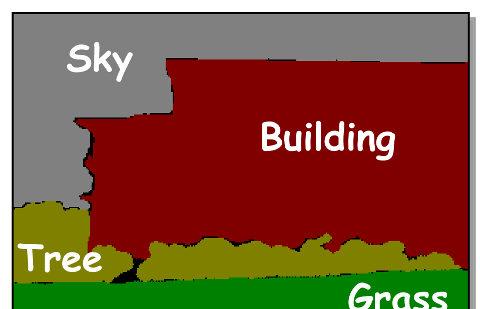


Image Denoising



Object Segmentation



Depth Estimation

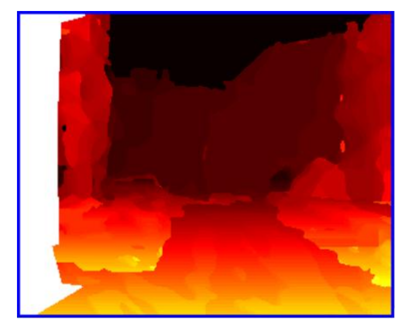
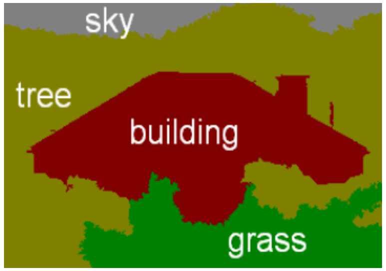
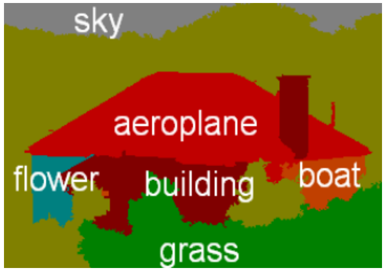


Image Labelling Problems

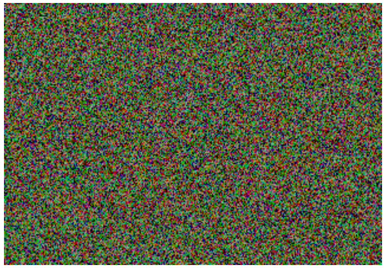
- Labellings highly structured



Possible labelling



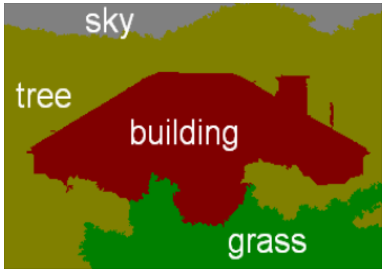
Unprobable labelling



Impossible labelling

Image Labelling Problems

- Labellings highly structured
- Labels highly correlated with very complex dependencies



- Neighbouring pixels tend to take the same label
- Low number of connected components
- Classes present may be seen in one image
- Geometric / Location consistency
- Planarity in depth estimation
- … many others (task dependent)

Image Labelling Problems

- Labelling highly structured
- Labels highly correlated with very complex dependencies
- Independent label estimation too hard

Image Labelling Problems

- Labelling highly structured
- Labels highly correlated with very complex dependencies
- Independent label estimation too hard
- Whole labelling should be formulated as one optimisation problem

Image Labelling Problems

- Labelling highly structured
- Labels highly correlated with very complex dependencies
- Independent label estimation too hard
- Whole labelling should be formulated as one optimisation problem
- Number of pixels up to millions
  - Hard to train complex dependencies
  - Optimisation problem is hard to infer

Image Labelling Problems

- Labelling highly structured
- Labels highly correlated with very complex dependencies
- Independent label estimation too hard
- Whole labelling should be formulated as one optimisation problem
- Number of pixels up to millions
  - Hard to train complex dependencies
  - Optimisation problem is hard to infer

Vision is hard !

Image Labelling Problems

- You can

  either

  - Change the subject from the Computer Vision to the History of Renaissance Art

Vision is hard !

Image Labelling Problems

- You can

  either

  - Change the subject from the Computer Vision to the History of Renaissance Art

  or

  - Learn everything about Random Fields and Graph Cuts

Vision is hard !

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**

Data term

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

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

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

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

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbf{L}} E(\mathbf{x})$$

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**

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbf{L}} E(\mathbf{x})$$

How to solve this optimisation problem?

Foreground / Background Estimation

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

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbf{L}} E(\mathbf{x})$$

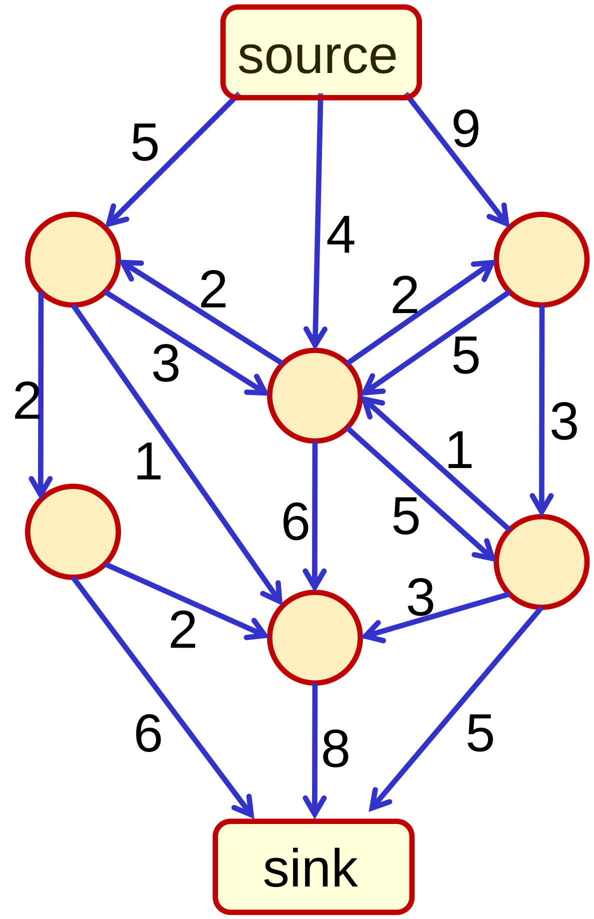
How to solve this optimisation problem?

- Transform into min-cut / max-flow problem
- Solve it using min-cut / max-flow algorithm

Overview

- Motivation
- Min-Cut / Max-Flow (Graph Cut) Algorithm
- Markov and Conditional Random Fields
- Random Field Optimisation using Graph Cuts
    - Submodular vs. Non-Submodular Problems
    - Pairwise vs. Higher Order Problems
    - 2-Label vs. Multi-Label Problems
- Applications
- Conclusions

Max-Flow Problem



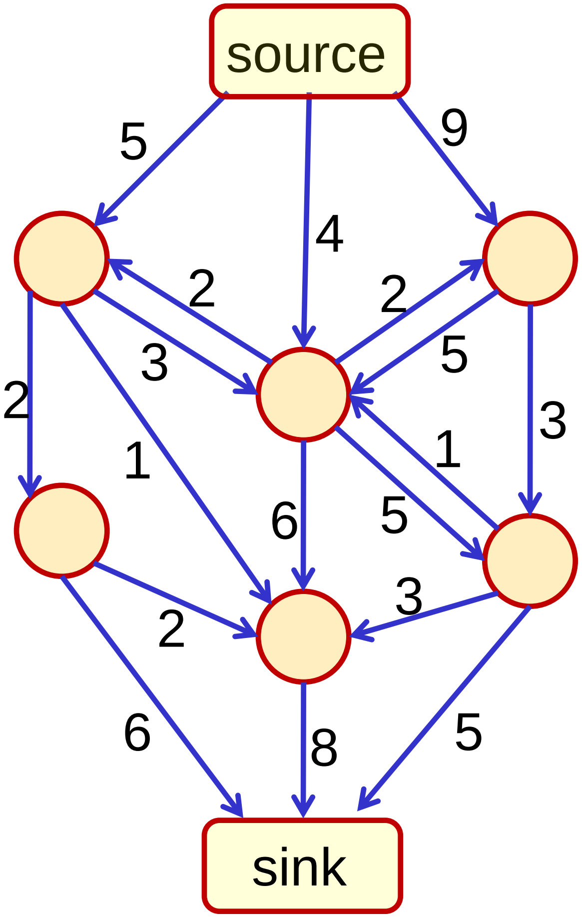
Task :

Maximize the flow from the sink to the source such that

1) The flow it conserved for each node

2) The flow for each pipe does not exceed the capacity

Max-Flow Problem

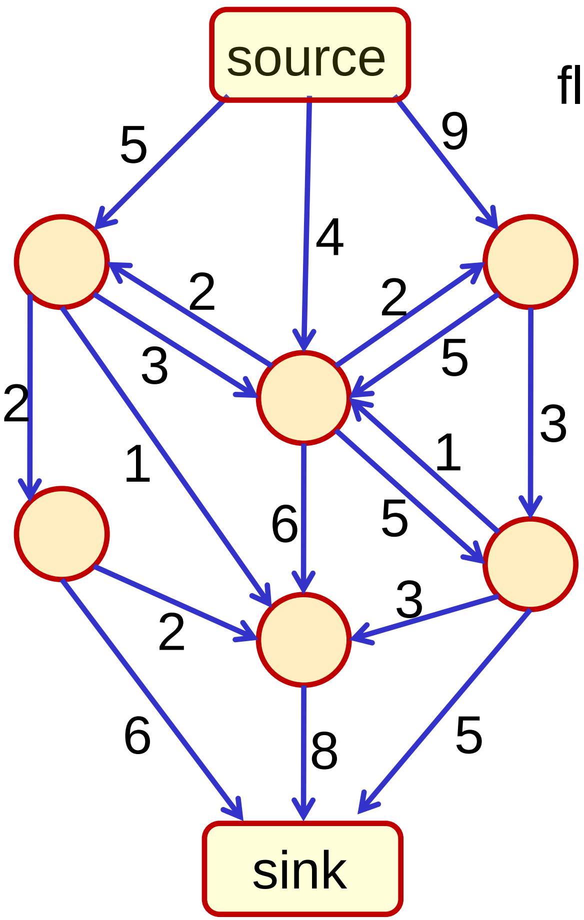


$$\max \textstyle\sum_{i \in V} f_{si}$$

$$\text{s.t.} \qquad 0 \leq f_{ij} \leq c_{ij}, \qquad \forall (i,j) \in E$$

$$\textstyle\sum_{j \in N(i)} f_{ji} - f_{ij} = 0, \qquad \forall i \in V \setminus \{s, t\}$$

Max-Flow Problem



flow from node i to node j

flow from the source

capacity

set of edges

$$\max \boxed{\sum_{i\in V} f_{si}}$$

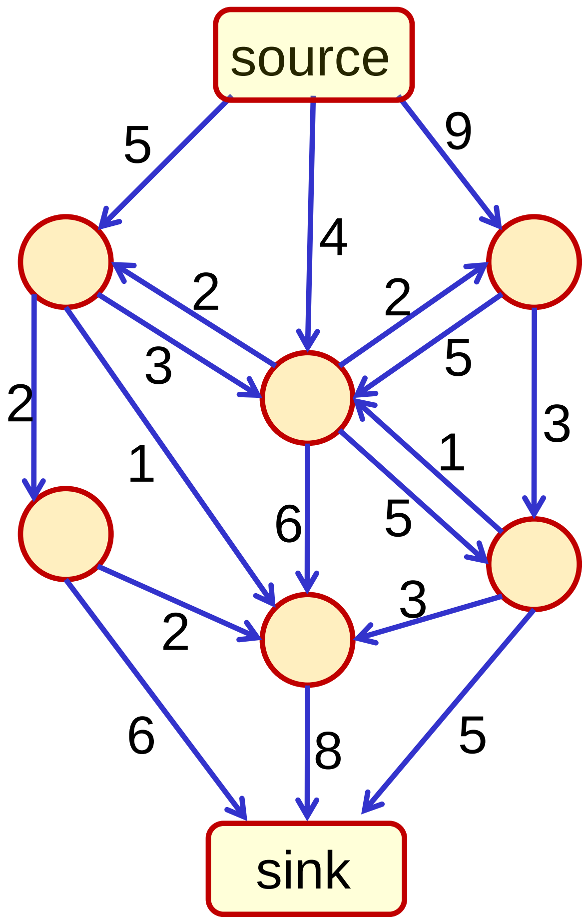
$$\text{s.t.} \qquad 0 \le \boxed{f_{ij}} \le \boxed{c_{ij},} \qquad \boxed{\forall (i,j) \in E}$$

$$\boxed{\sum_{j\in N(i)} f_{ji} - f_{ij} = 0,} \qquad \boxed{\forall i \in V \setminus \{s,t\}}$$

conservation of flow

set of nodes

Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

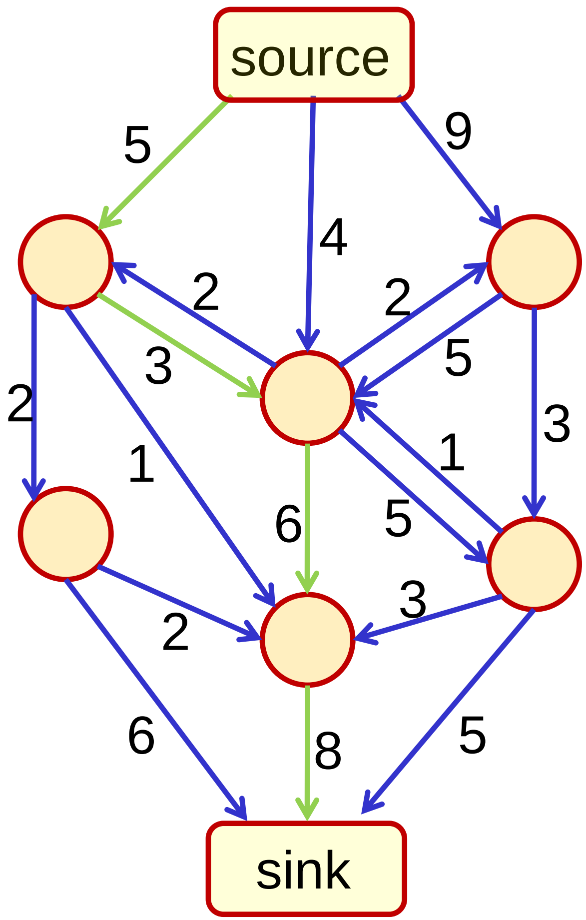
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

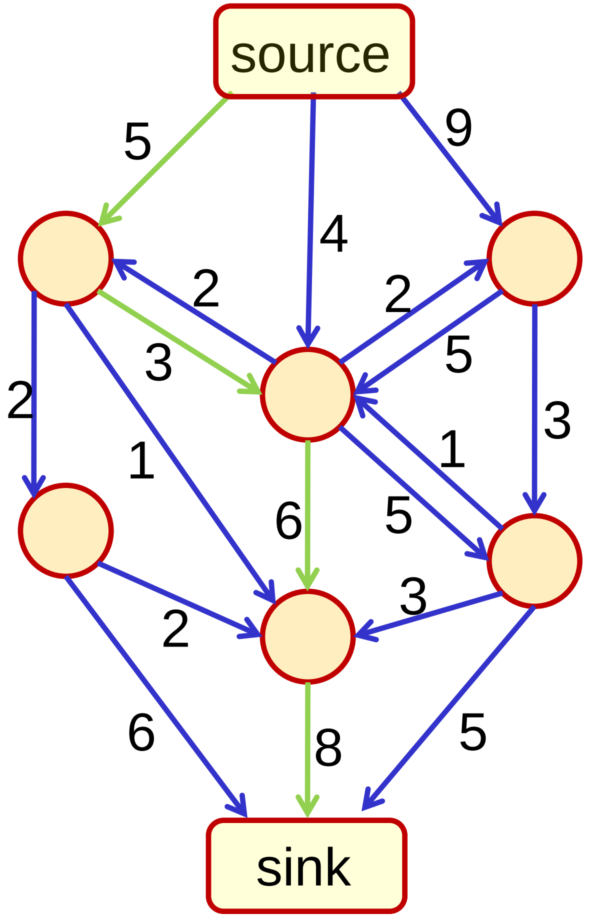
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 3

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

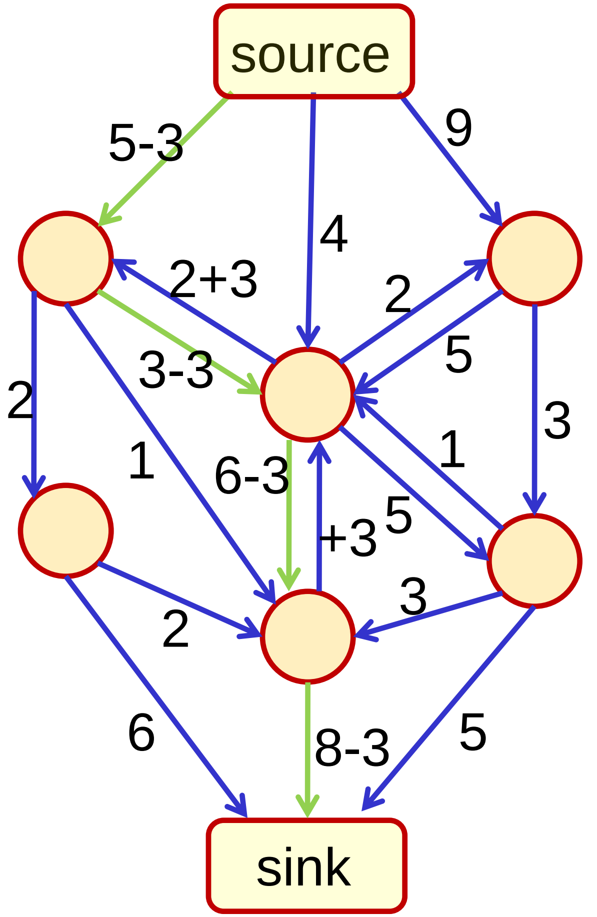
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 3

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

flow += maximum capacity in the path

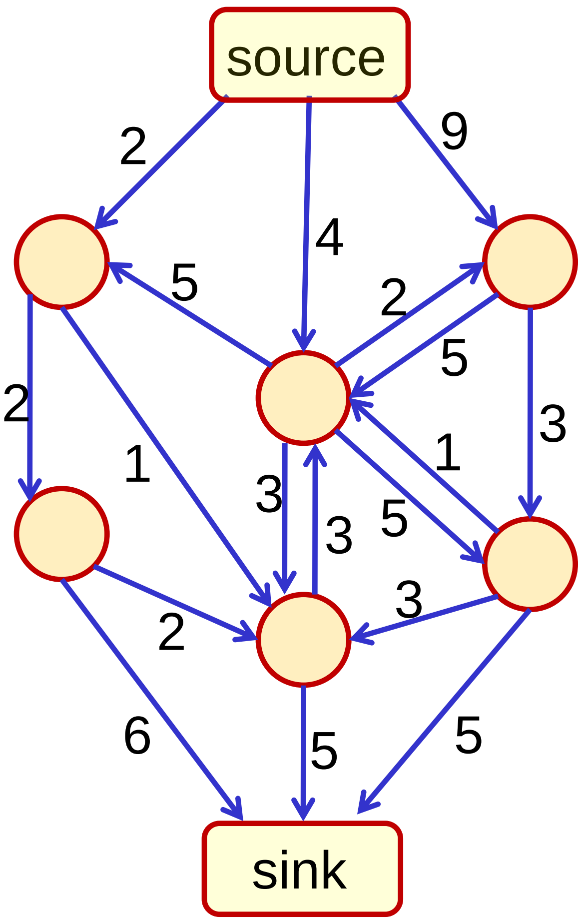
Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

$$r_{ij} = c_{ij} - f_{ij} + f_{ji}$$

Max-Flow Problem



flow = 3

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

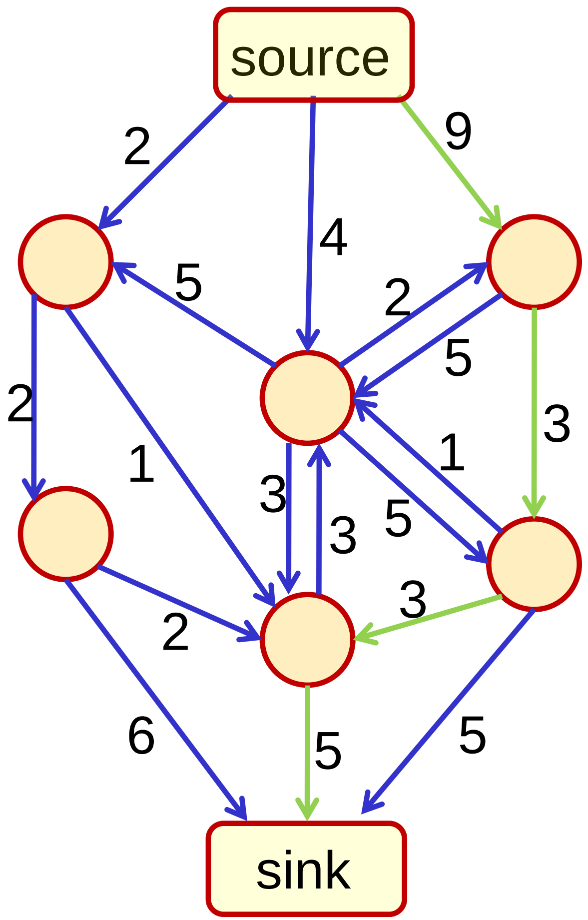
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 3

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

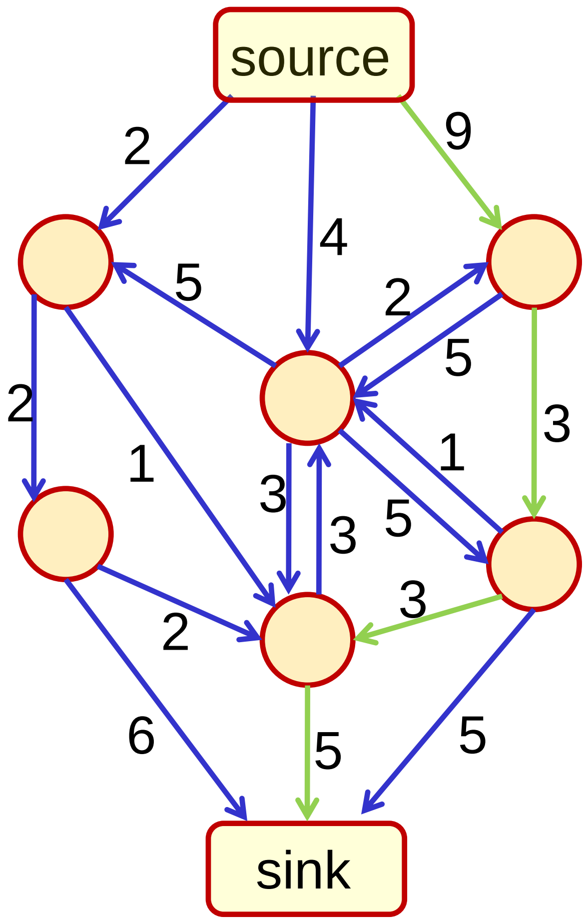
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 6

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

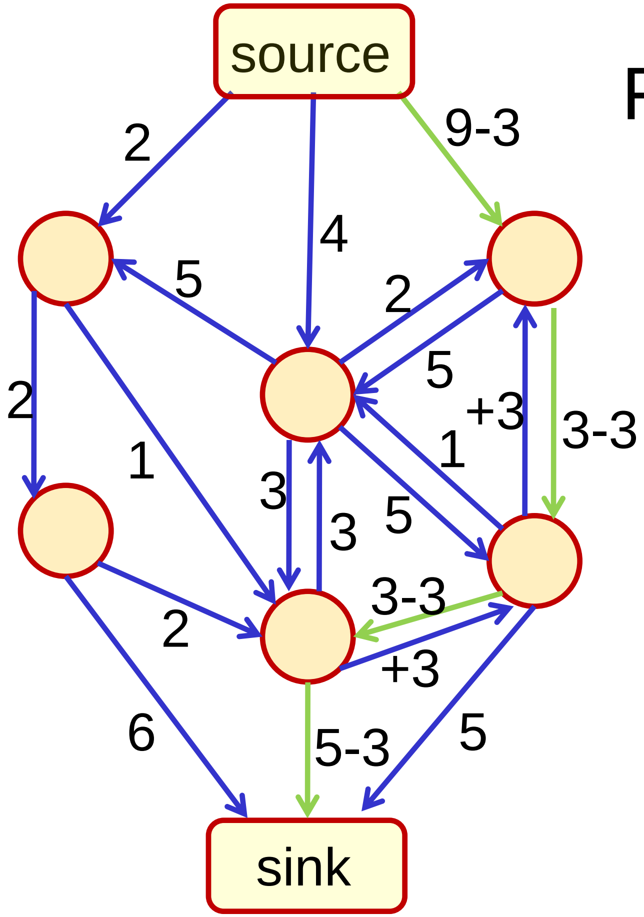
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 6

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

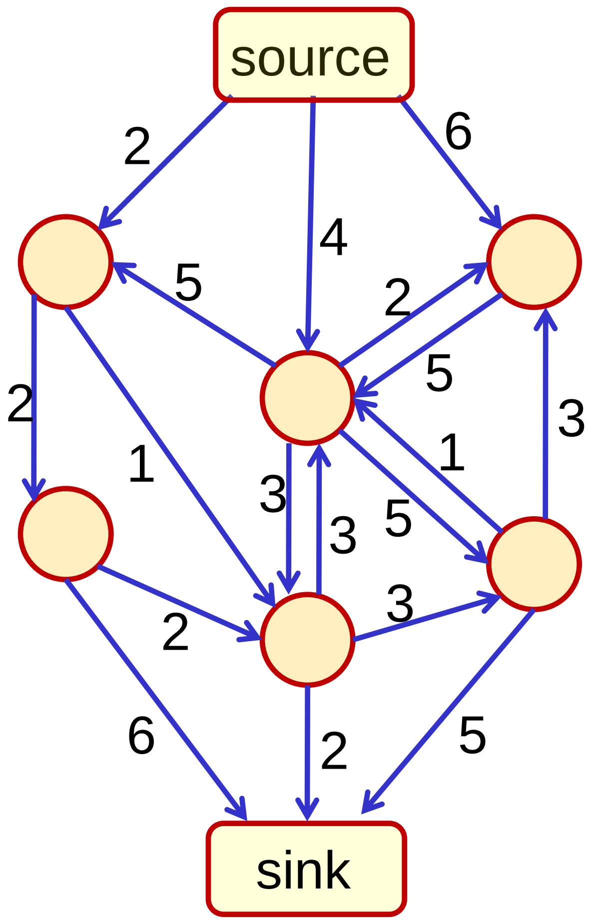
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 6

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

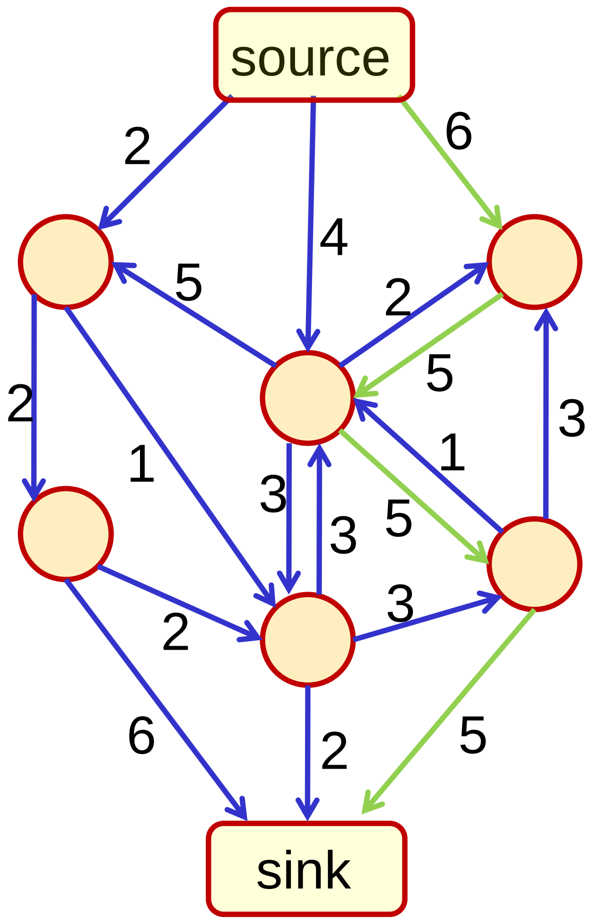
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 6

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

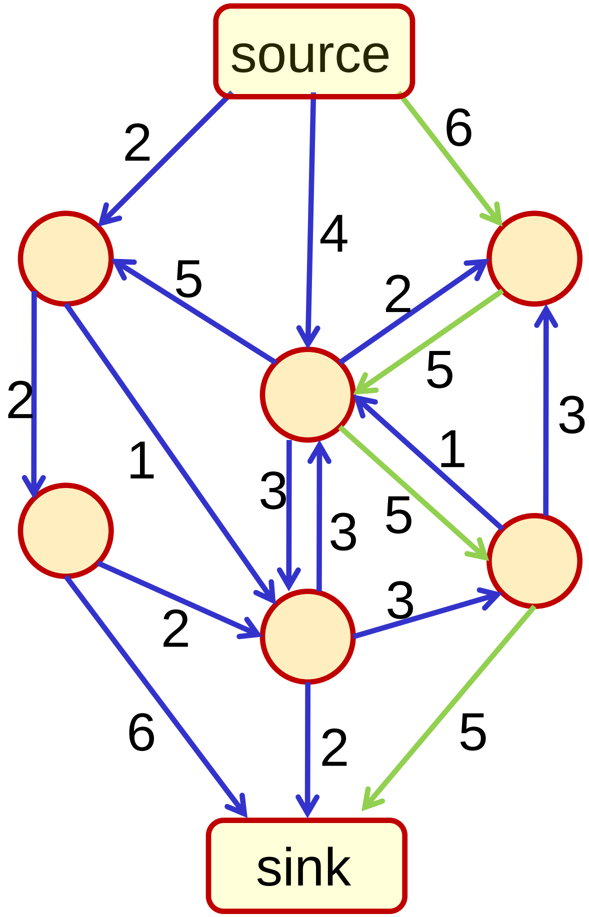
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 11

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

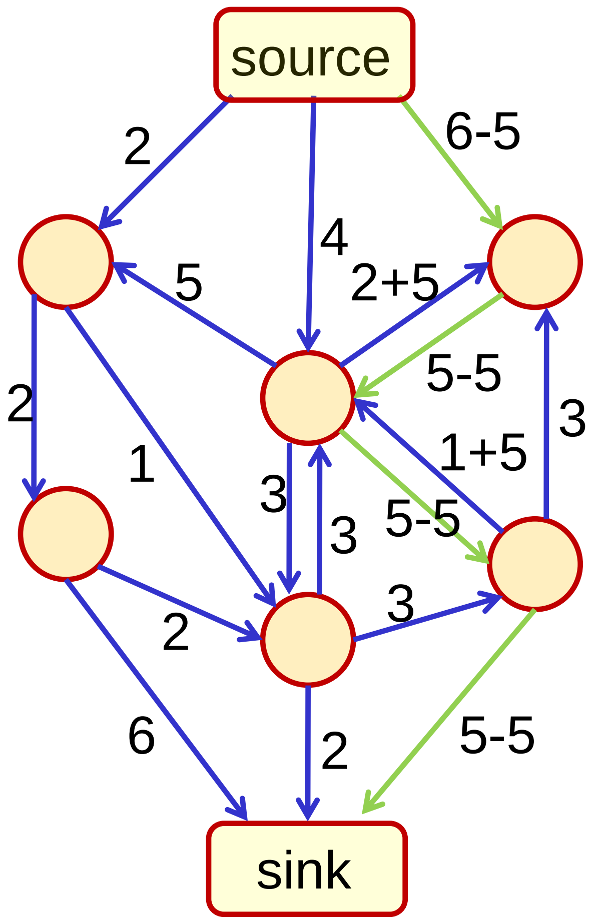
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 11

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

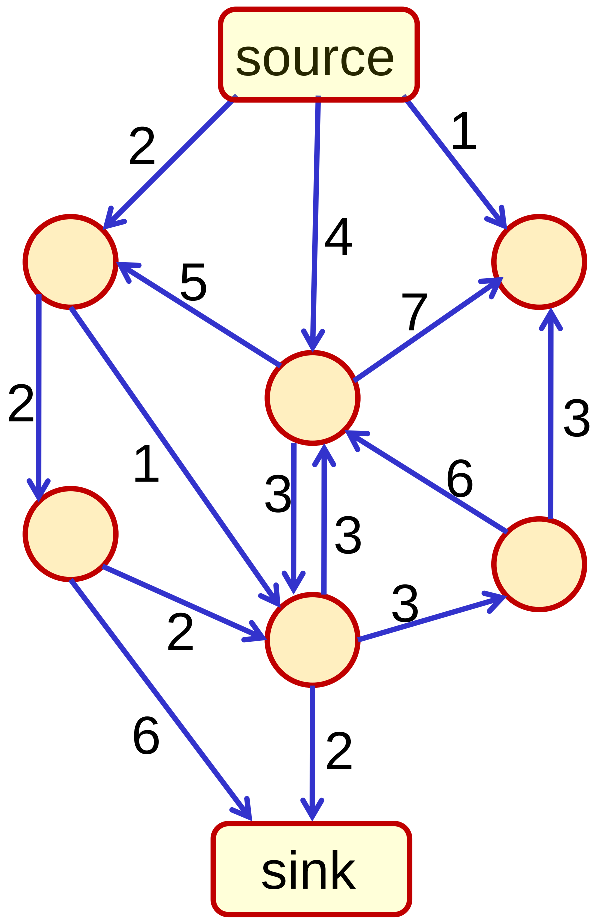
  flow += maximum capacity in the path

  Build the residual graph ("subtract" the flow)

  Find the path in the residual graph

End

Max-Flow Problem



flow = 11

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

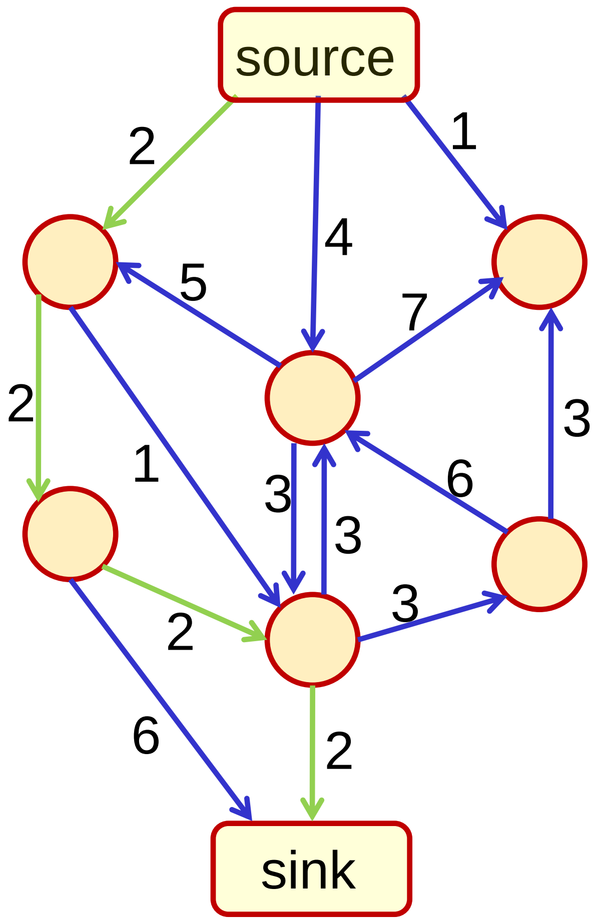
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 11

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

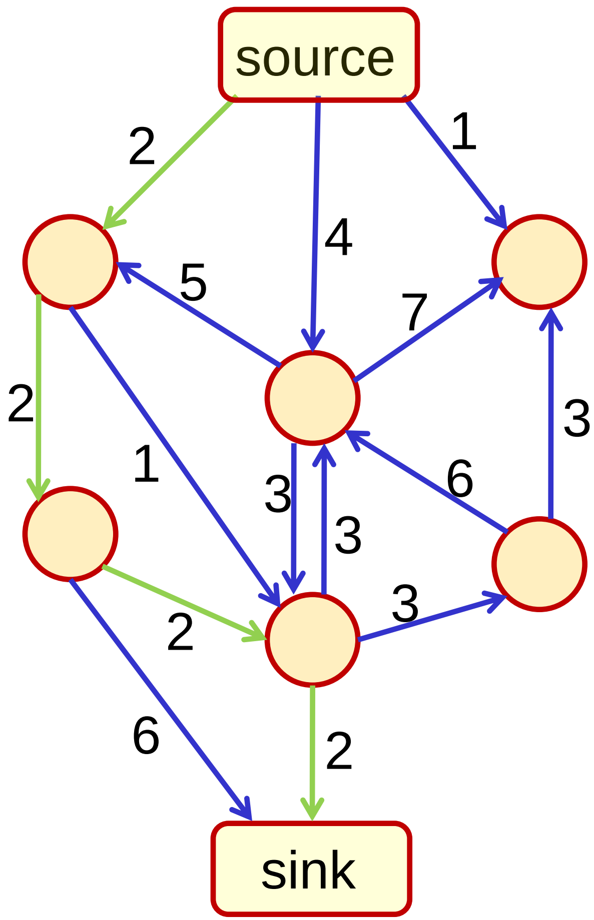
  flow += maximum capacity in the path

  Build the residual graph (“subtract” the flow)

  Find the path in the residual graph

End

Max-Flow Problem



flow = 13

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

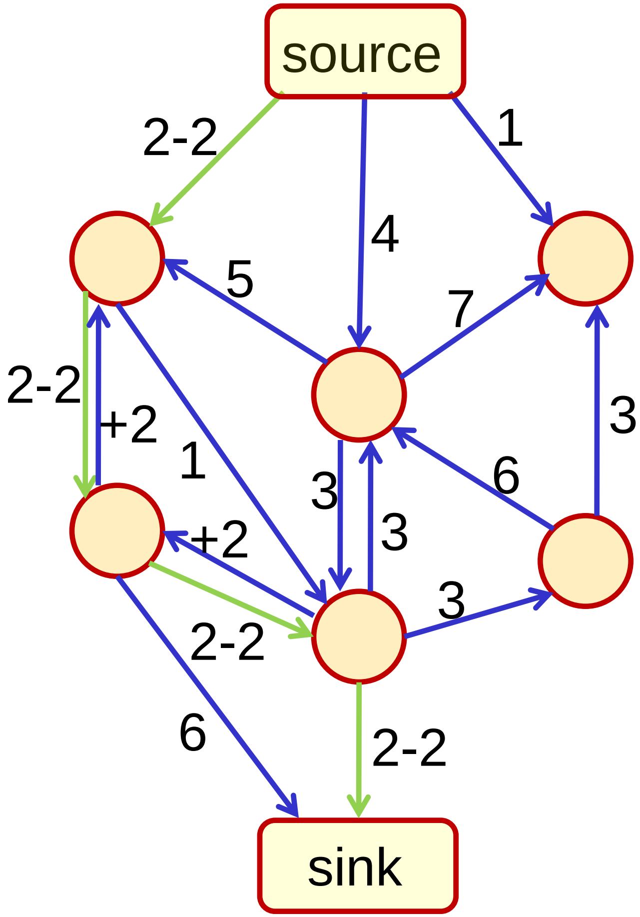
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 13

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

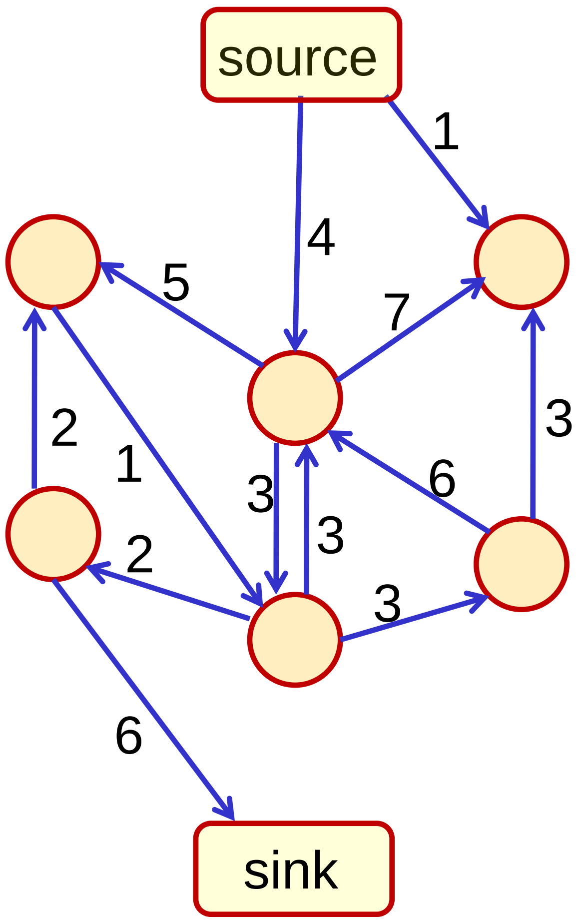
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 13

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

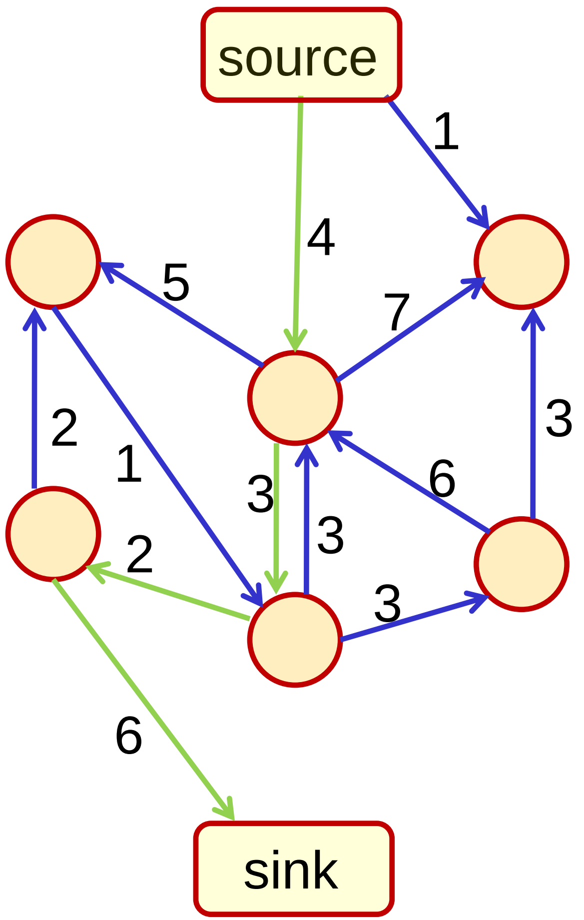
  flow += maximum capacity in the path

  Build the residual graph (“subtract” the flow)

  Find the path in the residual graph

End

Max-Flow Problem



flow = 13

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

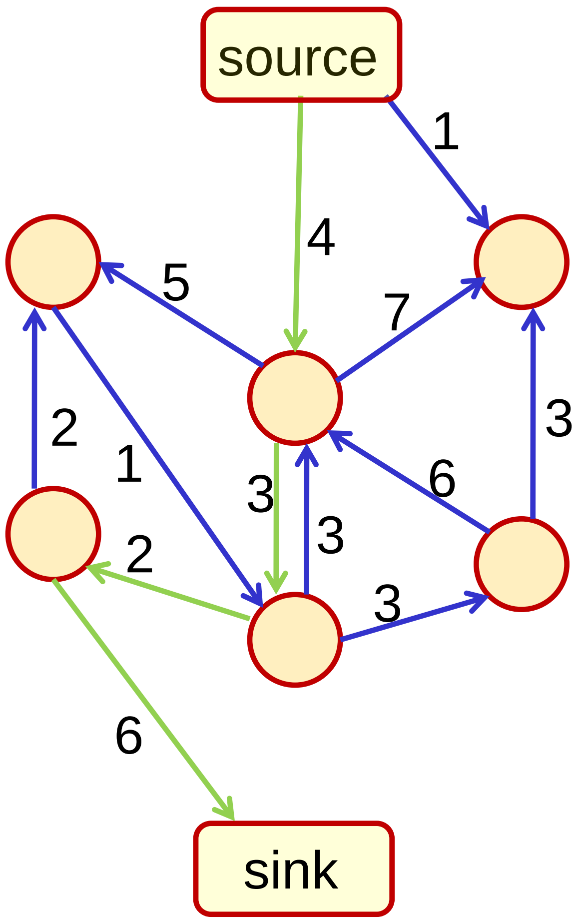
  flow += maximum capacity in the path

  Build the residual graph ("subtract" the flow)

  Find the path in the residual graph

End

Max-Flow Problem



flow = 15

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

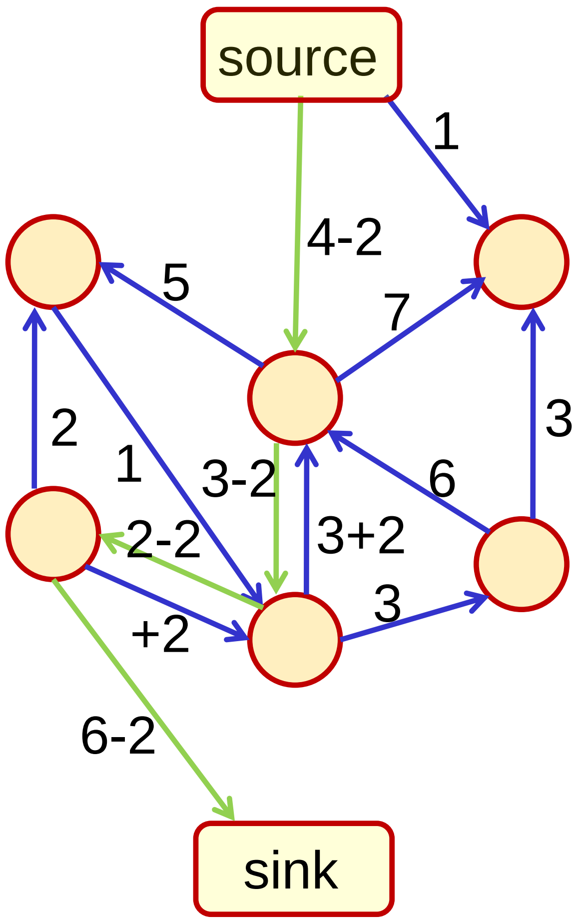
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 15

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

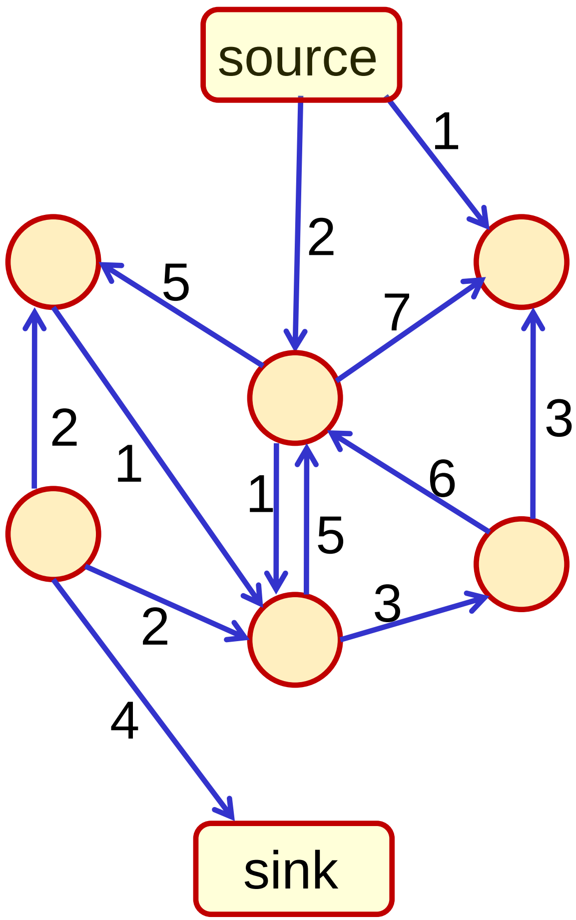
flow += maximum capacity in the path

Build the residual graph ("subtract" the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 15

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

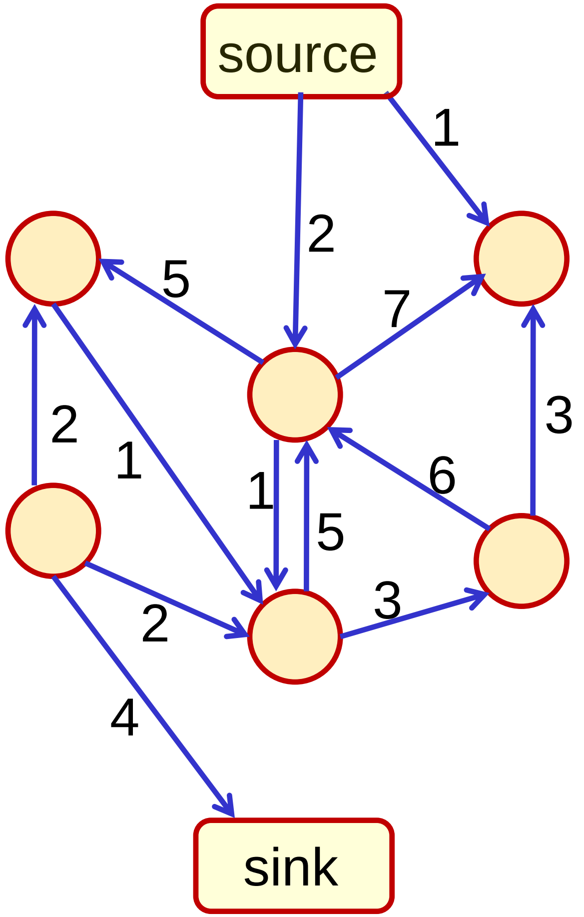
    flow += maximum capacity in the path

    Build the residual graph ("subtract" the flow)

    Find the path in the residual graph

End

Max-Flow Problem



flow = 15

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

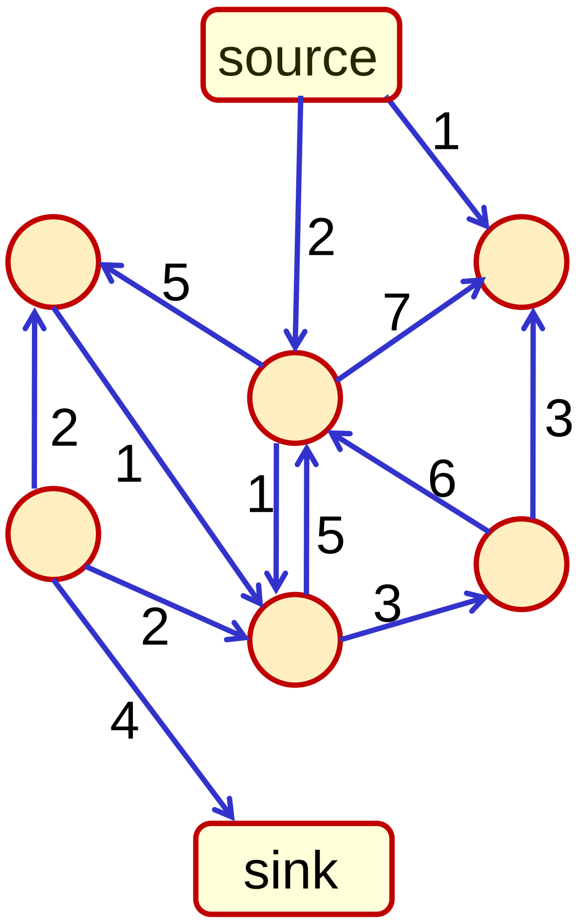
flow += maximum capacity in the path

Build the residual graph (“subtract” the flow)

Find the path in the residual graph

End

Max-Flow Problem



flow = 15

Ford & Fulkerson algorithm (1956)

Find the path from source to sink

While (path exists)

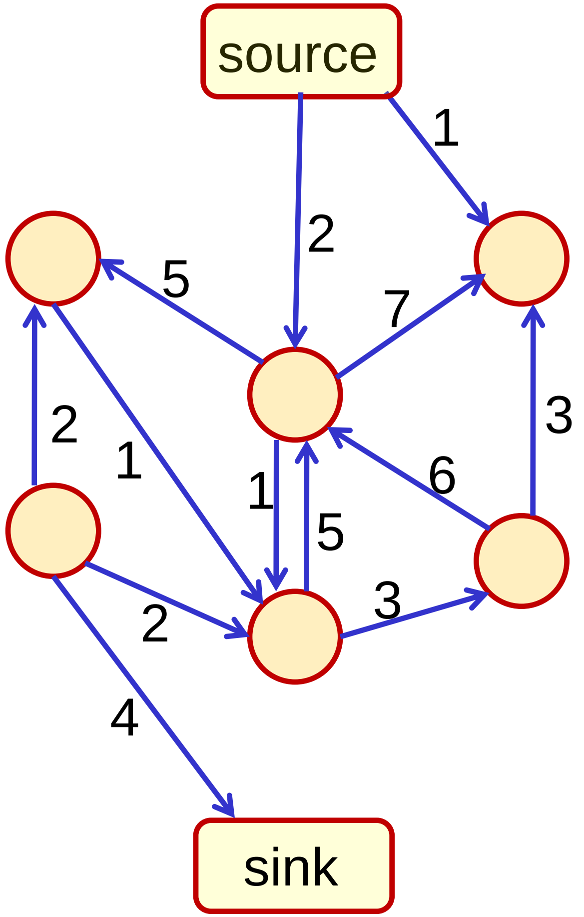
 flow += maximum capacity in the path

 Build the residual graph (“subtract” the flow)

 Find the path in the residual graph

End

Max-Flow Problem

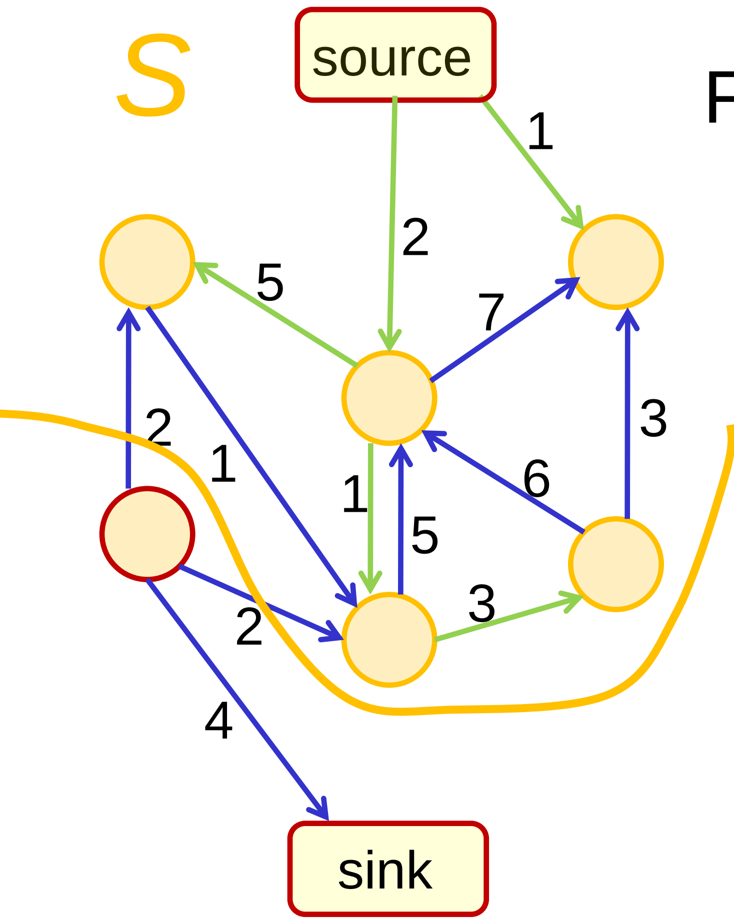


flow = 15

Ford & Fulkerson algorithm (1956)

Why is the solution globally optimal ?

Max-Flow Problem



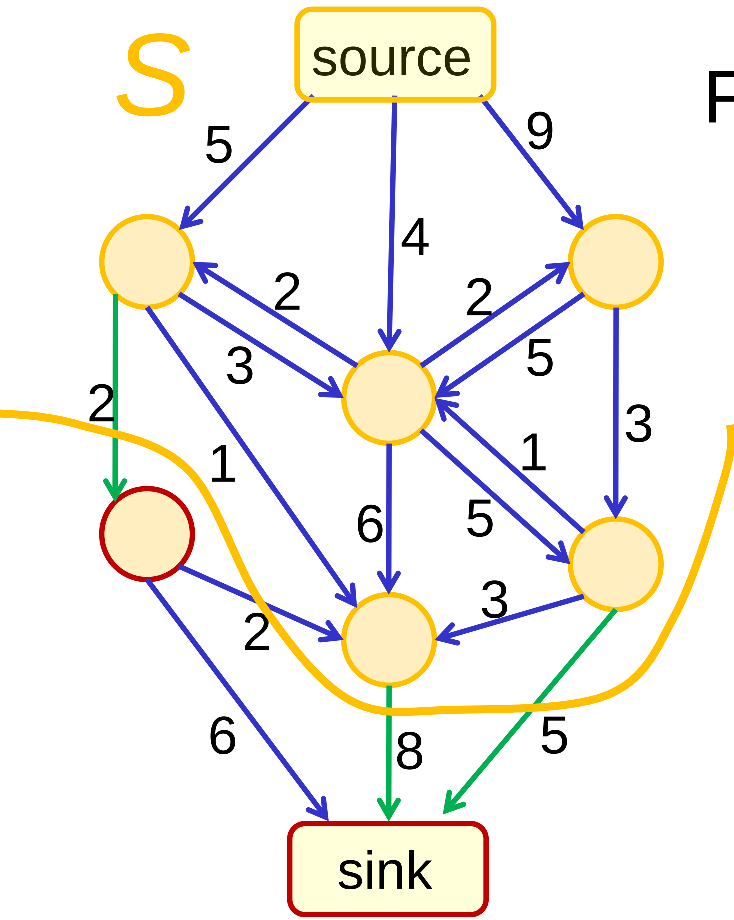
flow = 15

Ford & Fulkerson algorithm (1956)

Why is the solution globally optimal ?

1. Let S be the set of reachable nodes in the residual graph

Max-Flow Problem

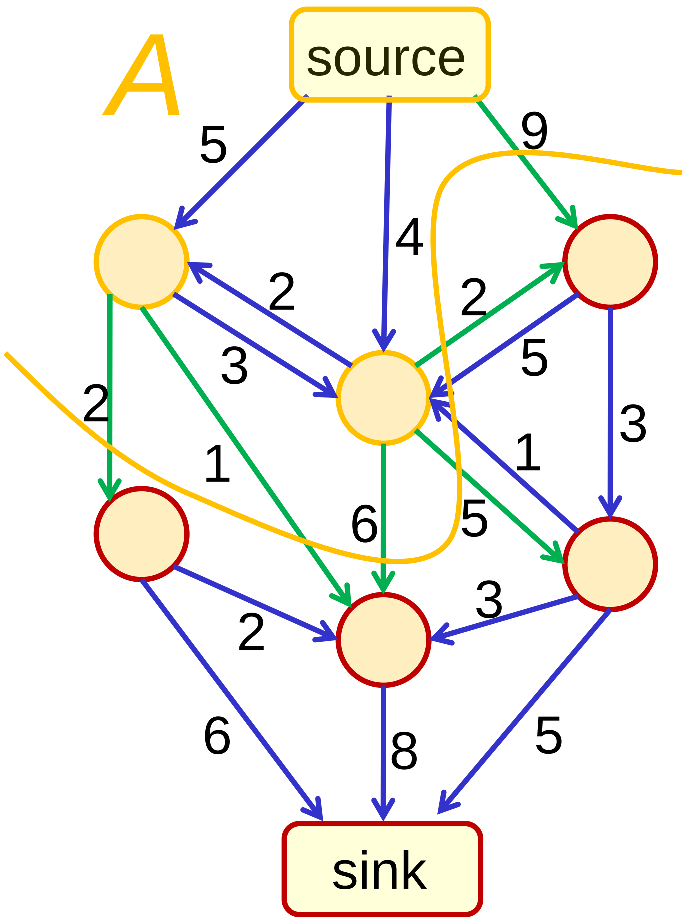


Ford & Fulkerson algorithm (1956)

Why is the solution globally optimal ?

1. Let S be the set of reachable nodes in the residual graph
2. The flow from S to V - S equals to the sum of capacities from S to V – S

Max-Flow Problem

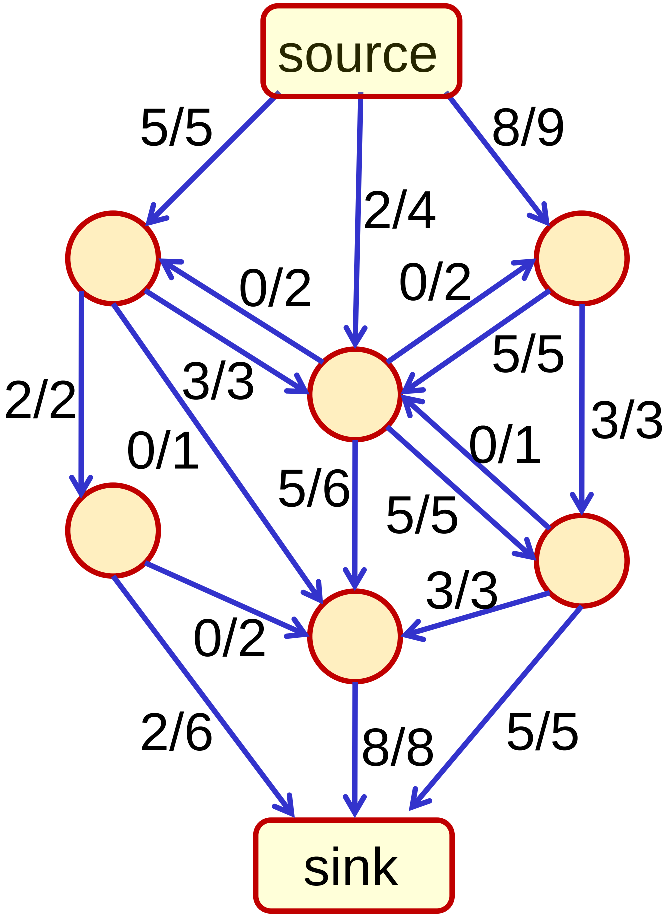


Ford & Fulkerson algorithm (1956)

Why is the solution globally optimal ?

1. Let S be the set of reachable nodes in the residual graph
2. The flow from S to V - S equals to the sum of capacities from S to V – S
3. The flow from any A to V - A is upper bounded by the sum of capacities from A to V – A

Max-Flow Problem



flow = 15

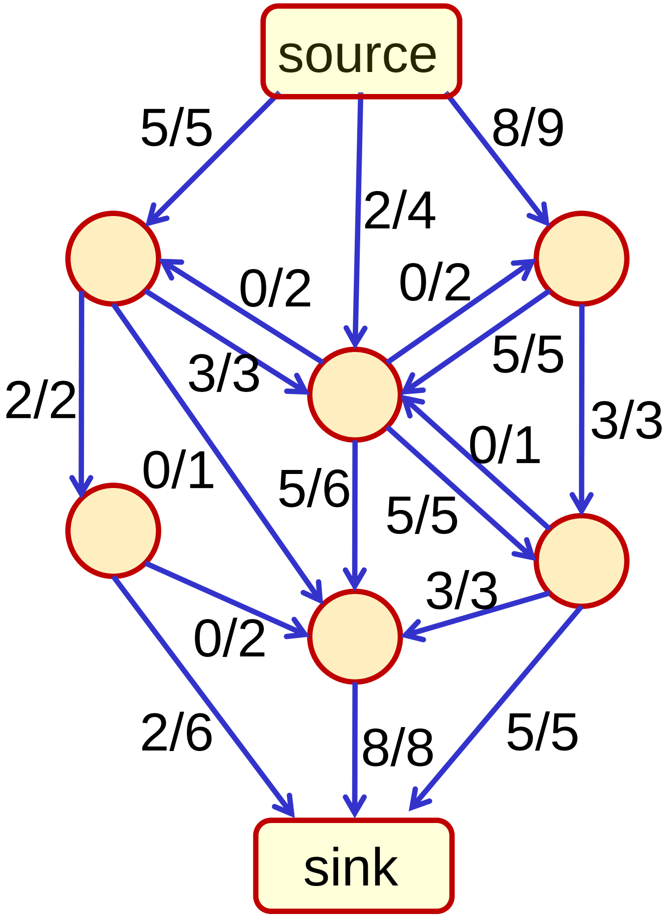
Individual flows obtained by summing up all paths

Ford & Fulkerson algorithm (1956)

Why is the solution globally optimal ?

1. Let S be the set of reachable nodes in the residual graph
2. The flow from S to V - S equals to the sum of capacities from S to V – S
3. The flow from any A to V - A is upper bounded by the sum of capacities from A to V – A
4. The solution is globally optimal

Max-Flow Problem



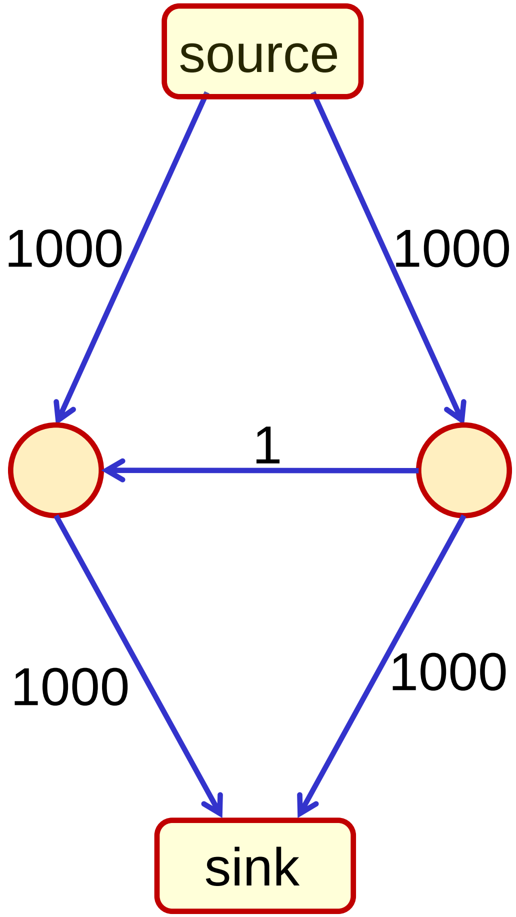
flow = 15

Ford & Fulkerson algorithm (1956)

Why is the solution globally optimal ?

1. Let S be the set of reachable nodes in the residual graph
2. The flow from S to V - S equals to the sum of capacities from S to V – S
3. The flow from any A to V - A is upper bounded by the sum of capacities from A to V – A
4. The solution is globally optimal

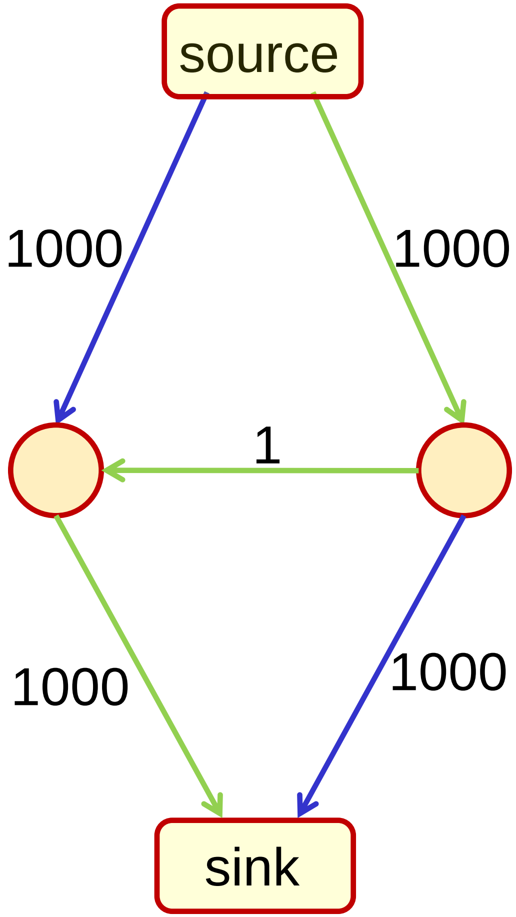
Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Order does matter

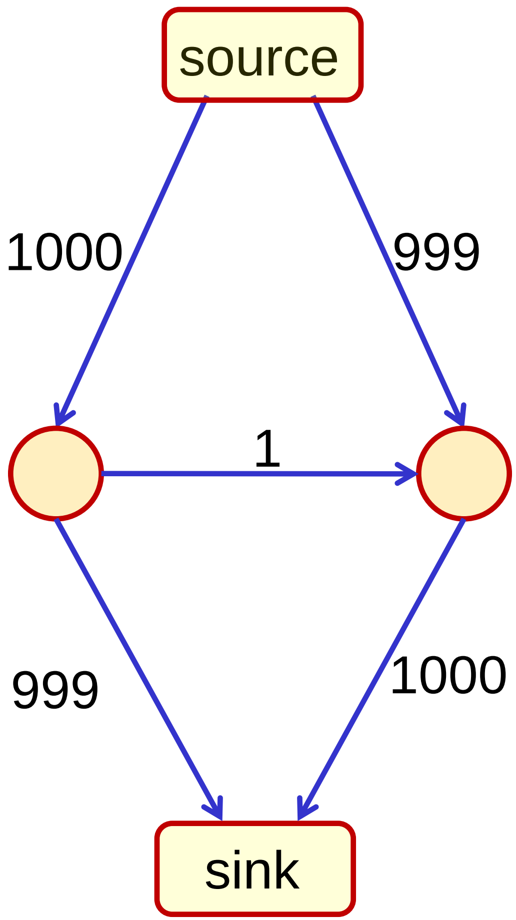
Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Order does matter

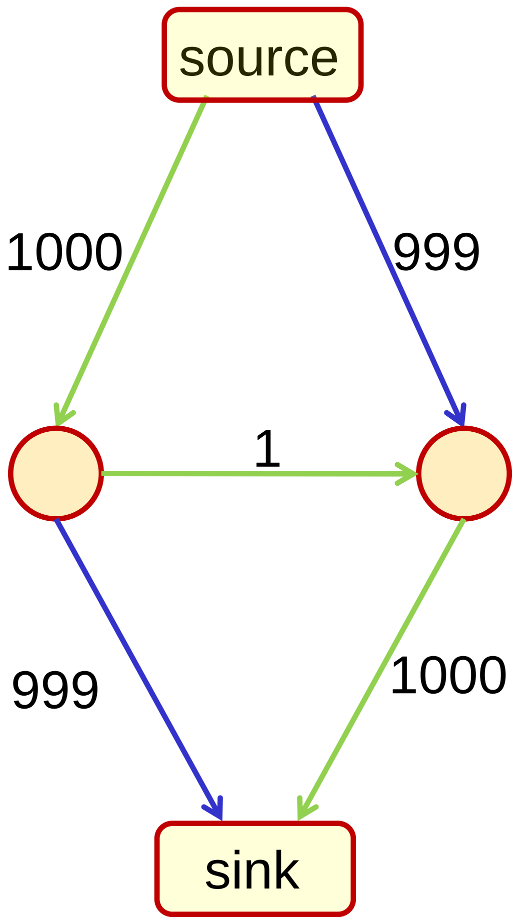
Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Order does matter

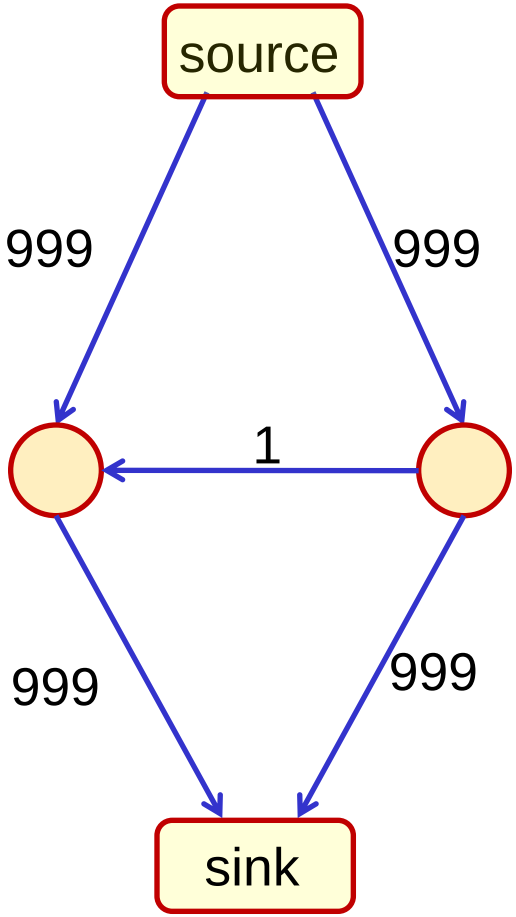
Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Order does matter

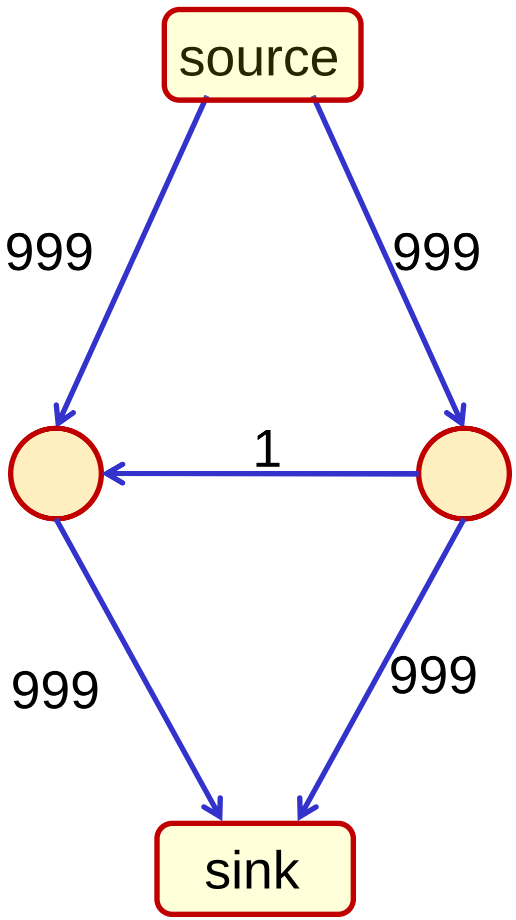
Max-Flow Problem



Ford & Fulkerson algorithm (1956)

Order does matter

Max-Flow Problem



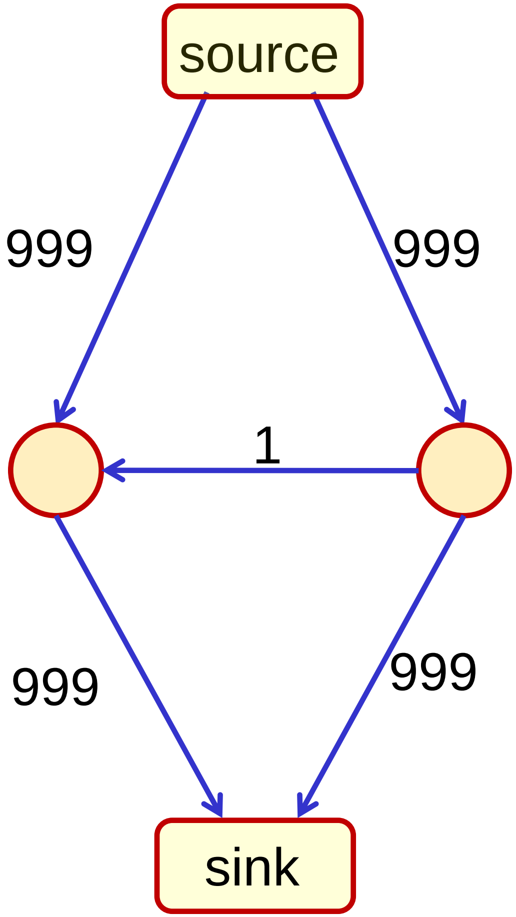
Ford & Fulkerson algorithm (1956)

Order does matter

- Standard algorithm not polynomial
- Breath first leads to O(VE^2)
  - Path found in O(E)
  - At least one edge gets saturated
  - The saturated edge distance to the source has to increase and is at most V leading to O(VE)

(Edmonds & Karp, Dinic)

Max-Flow Problem



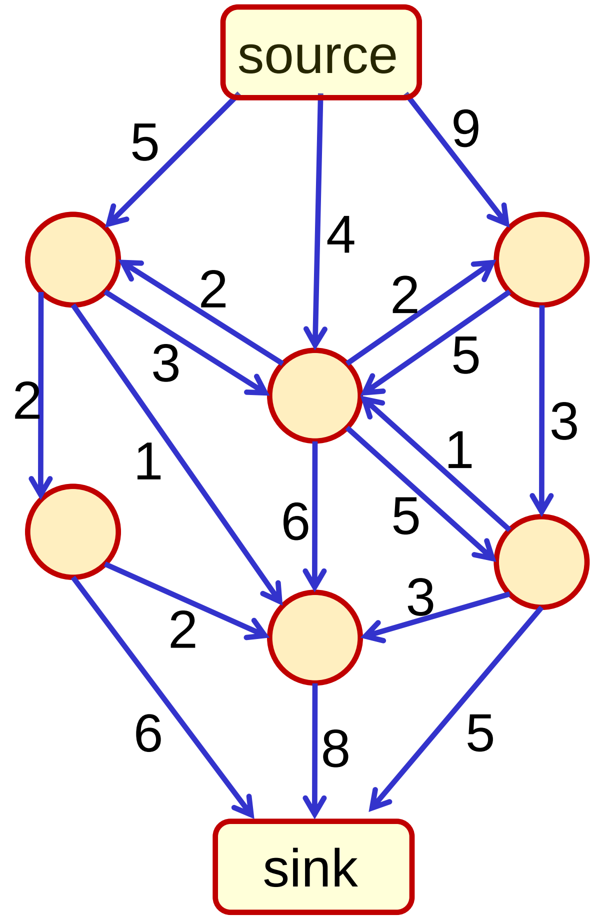
Ford & Fulkerson algorithm (1956)

Order does matter

- Standard algorithm not polynomial
- Breath first leads to $O(VE^2)$

  (Edmonds & Karp)
- Various methods use different algorithm to find the path and vary in complexity

Min-Cut Problem



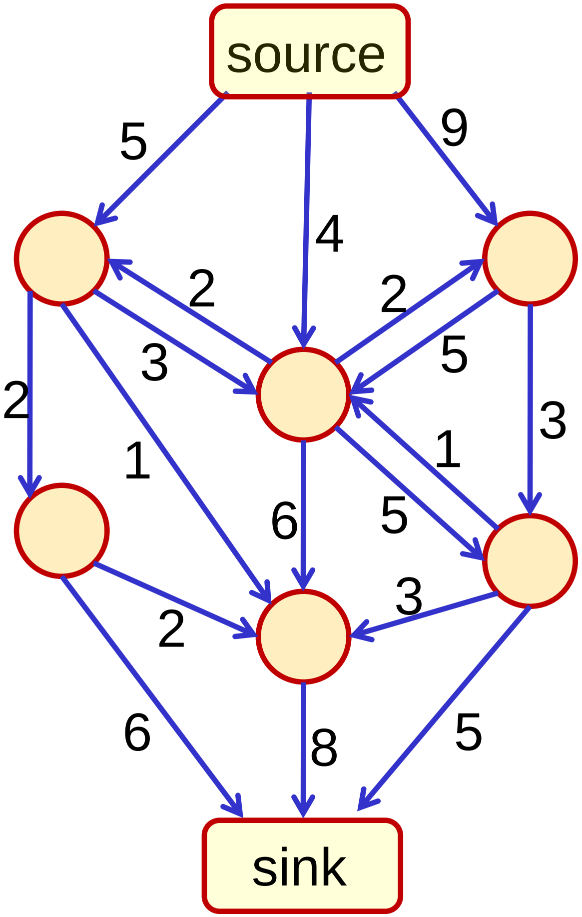
Task :

Minimize the cost of the cut

1) Each node is either assigned to the source S or sink T

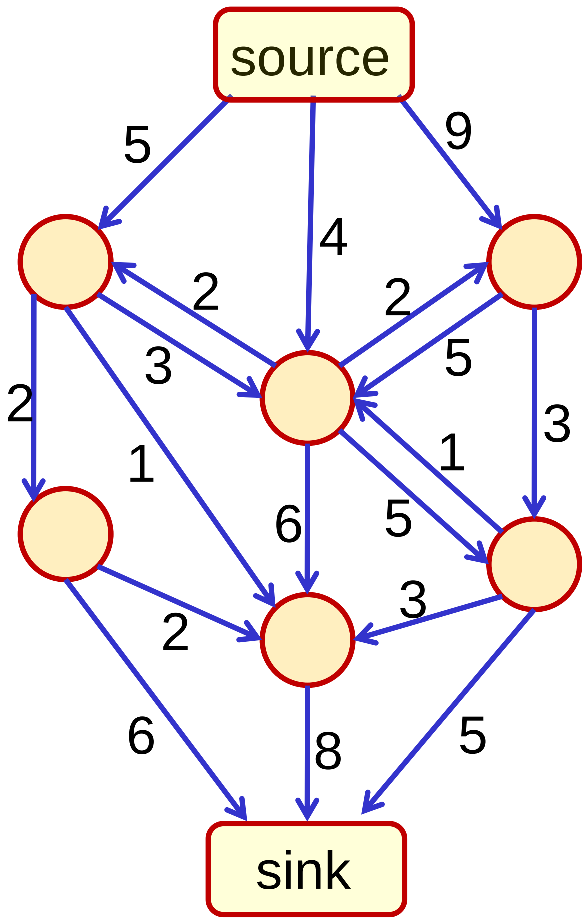
2) The cost of the edge (i, j) is taken if ($i \in S$) and ($j \in T$)

Min-Cut Problem



$$\min_{S,T} \sum_{i \in S, j \in T} c_{ij}$$

Min-Cut Problem



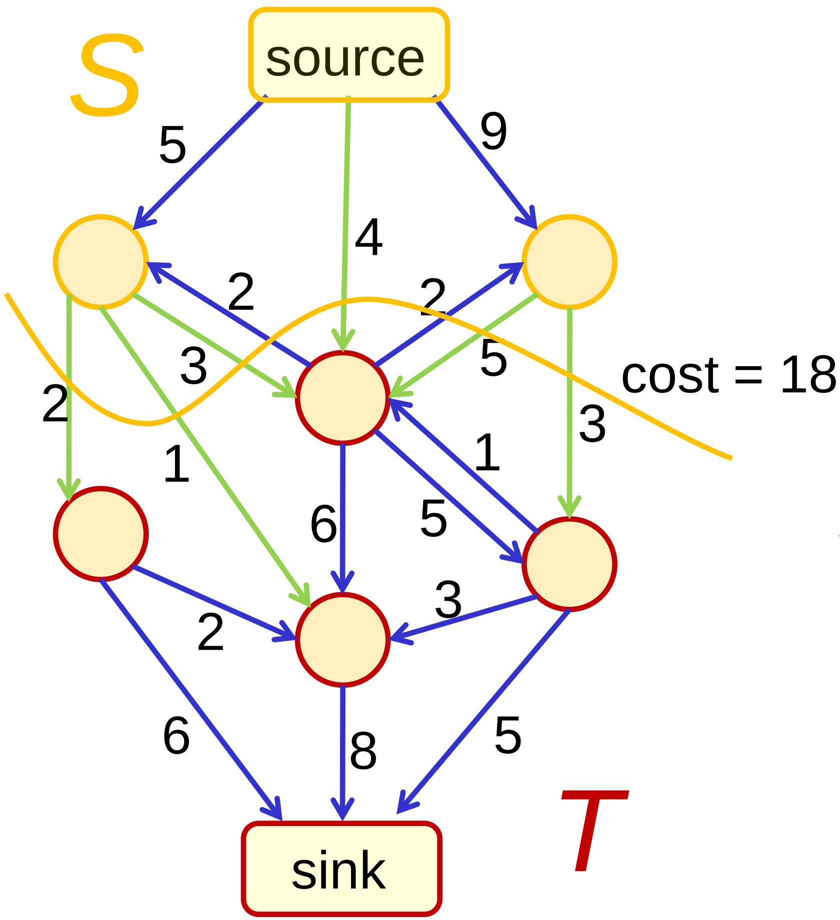
edge costs

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

source set

sink set

Min-Cut Problem



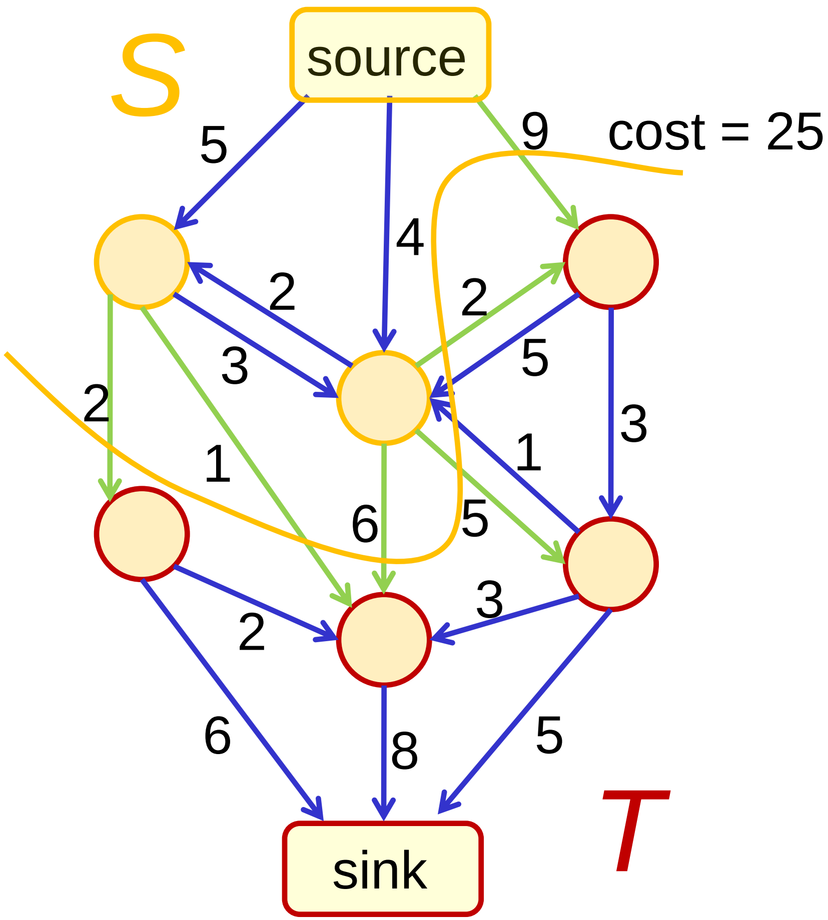
$$\min_{S,T} \sum_{\boxed{i\in S},\boxed{j\in T}} \boxed{c_{ij}} \qquad s.t. \quad s \in S, \quad t \in T$$

edge costs

source set

sink set

Min-Cut Problem



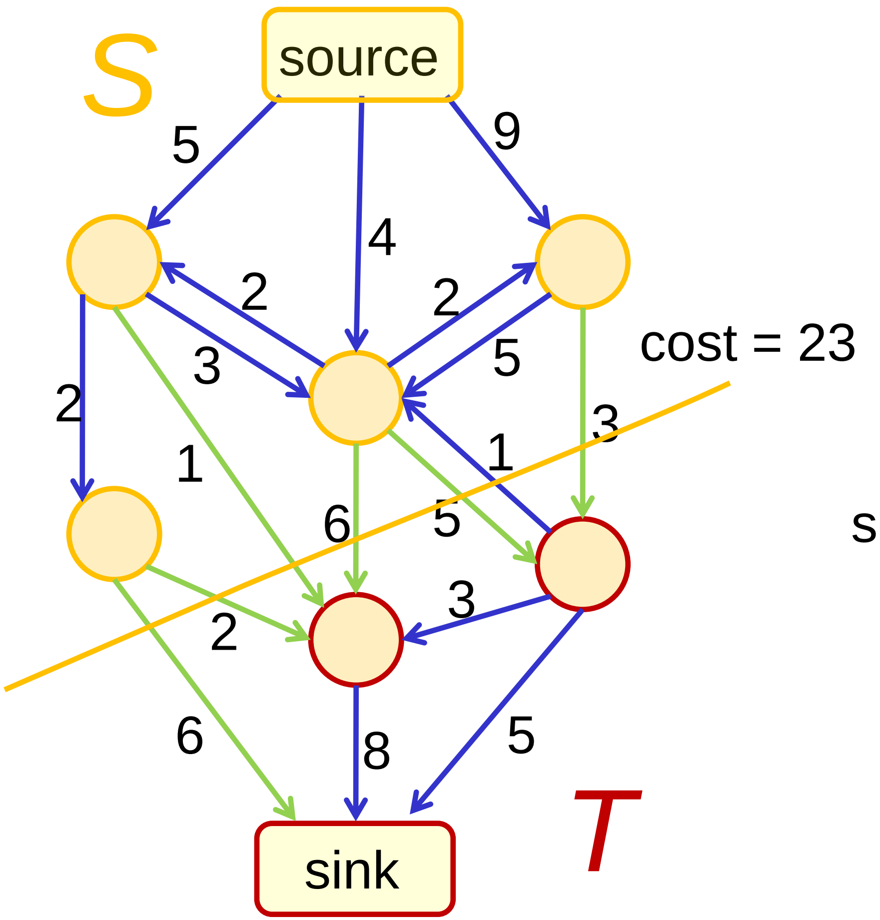
edge costs

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

source set

sink set

Min-Cut Problem



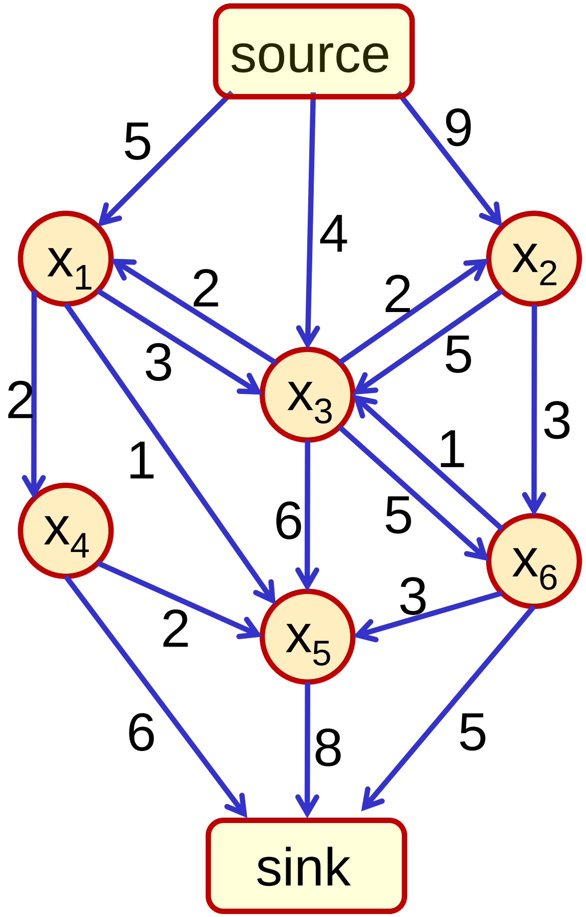
$$\min_{S,T} \sum_{\boxed{i\in S},\boxed{j\in T}} \boxed{c_{ij}} \qquad s.t. \quad s \in S, \quad t \in T$$

edge costs

source set

sink set

Min-Cut Problem

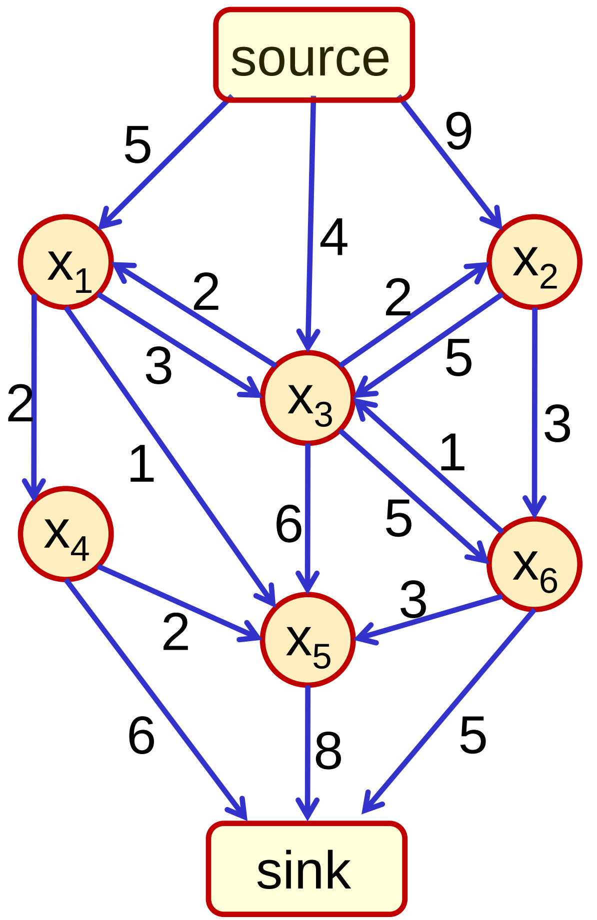


$$\min_{\mathbf{x}} \sum_{(i,j)\in E} c_{ij}(1 - x_i)x_j$$

$$s.t. \quad x_s = 0, \quad x_t = 1$$

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

Min-Cut Problem



$$\min_{\mathbf{x}} \sum_{(i,j)\in E} c_{ij}(1 - x_i)x_j$$

$$s.t. \quad x_s = 0, \quad x_t = 1$$

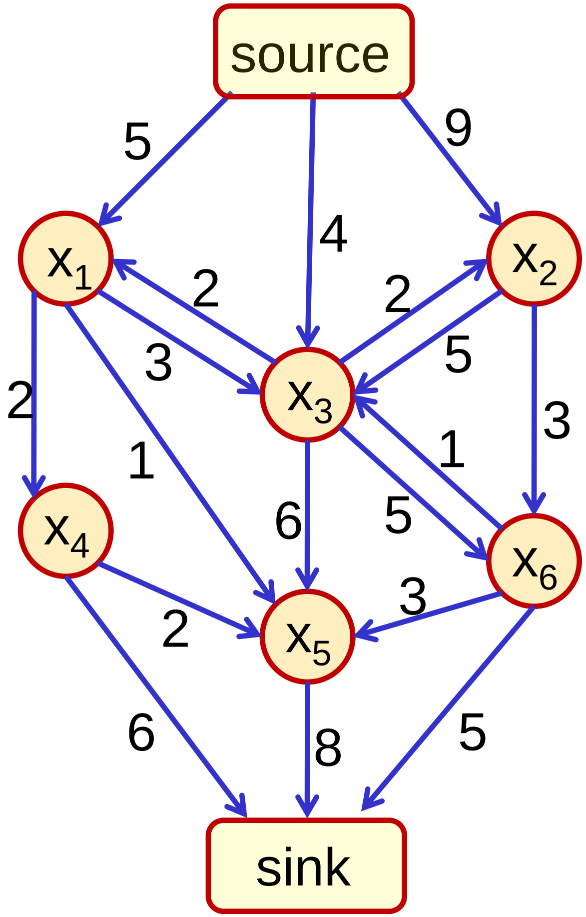
transformable into Linear program (LP)

$$\min_{\mathbf{x},\mathbf{d}} \quad c_{ij} d_{ij}$$

$$s.t. \quad d_{ij} \geq x_j - x_i \qquad d_{ij} \geq 0$$

$$x_s = 0 \qquad\qquad x_t = 1$$

Min-Cut Problem



$$\min_{\mathbf{x}} \sum_{(i,j)\in E} c_{ij}(1-x_i)x_j$$

$$s.t. \quad x_s = 0, \quad x_t = 1$$

transformable into Linear program (LP)

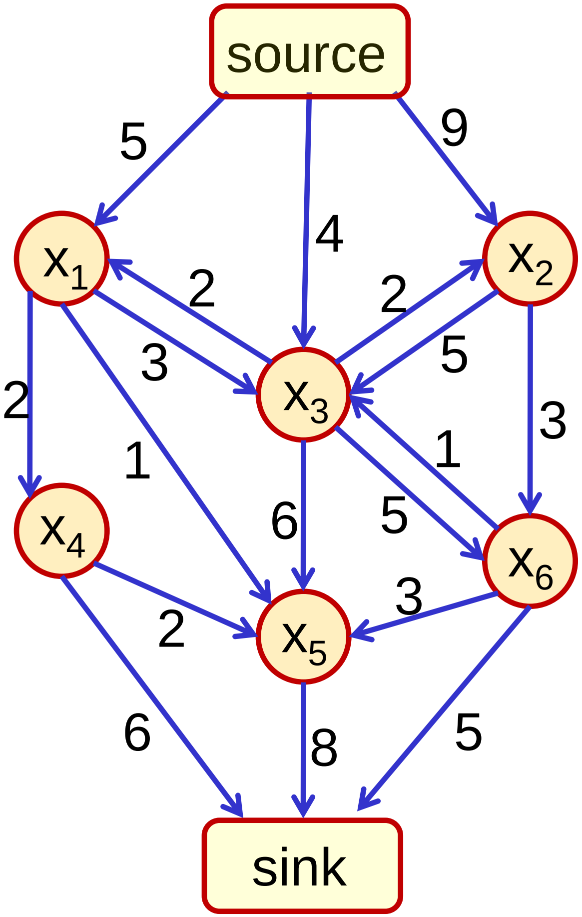
$$\min_{\mathbf{x},\mathbf{d}} \quad c_{ij}d_{ij}$$

$$s.t. \quad d_{ij} \geq x_j - x_i \qquad d_{ij} \geq 0$$

$$x_s = 0 \qquad x_t = 1$$

Dual to max-flow problem

Min-Cut Problem

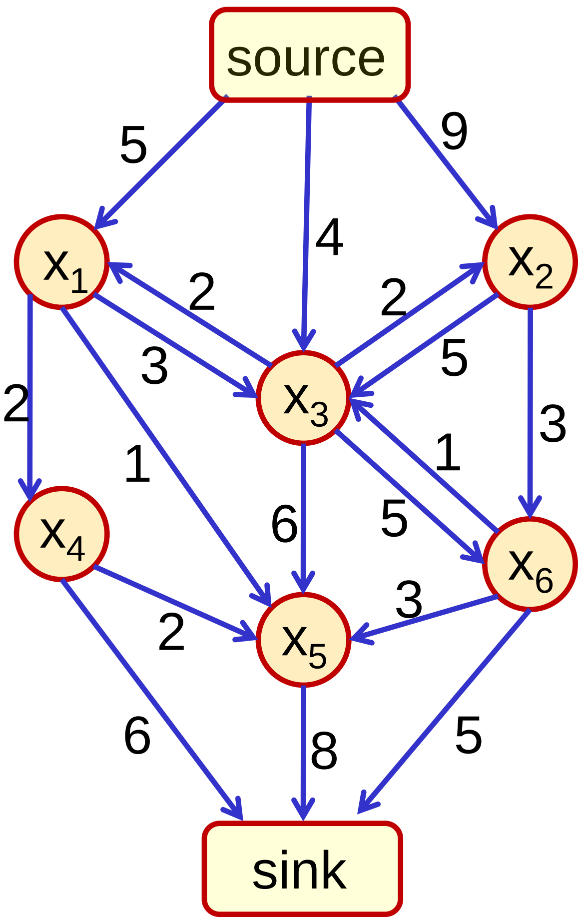


After the substitution of the constraints :

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

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

Min-Cut Problem

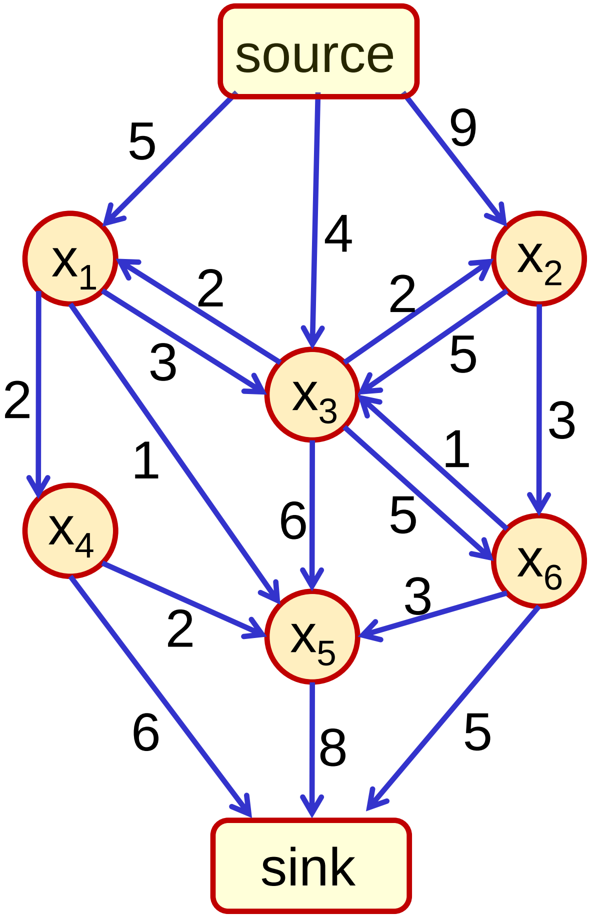


$$\min_{\mathbf{x}} \underbrace{\sum_{(s,i)\in E} c_{si} x_i}_{\text{source edges}} + \underbrace{\sum_{(i,t)\in E} c_{it}(1 - x_i)}_{\text{sink edges}}$$

$$\underbrace{+ \sum_{(i,j)\in E,\, i,j \notin \{s,t\}} c_{ij}(1 - x_i) x_j}_{\text{Edges between variables}}$$

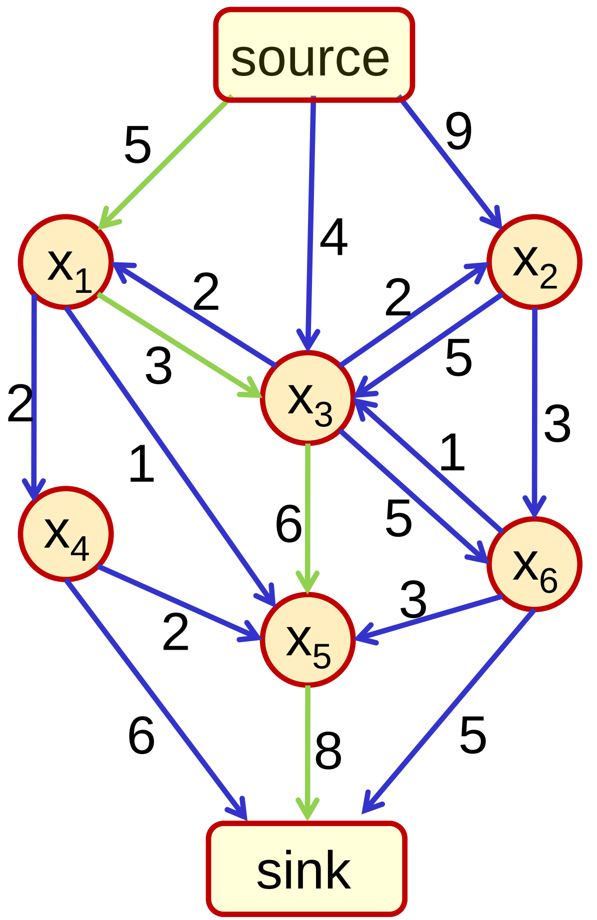
$x_i = 0 \;\Rightarrow\; x_i \in S$ $\qquad$ $x_i = 1 \;\Rightarrow\; x_i \in T$

Min-Cut Problem



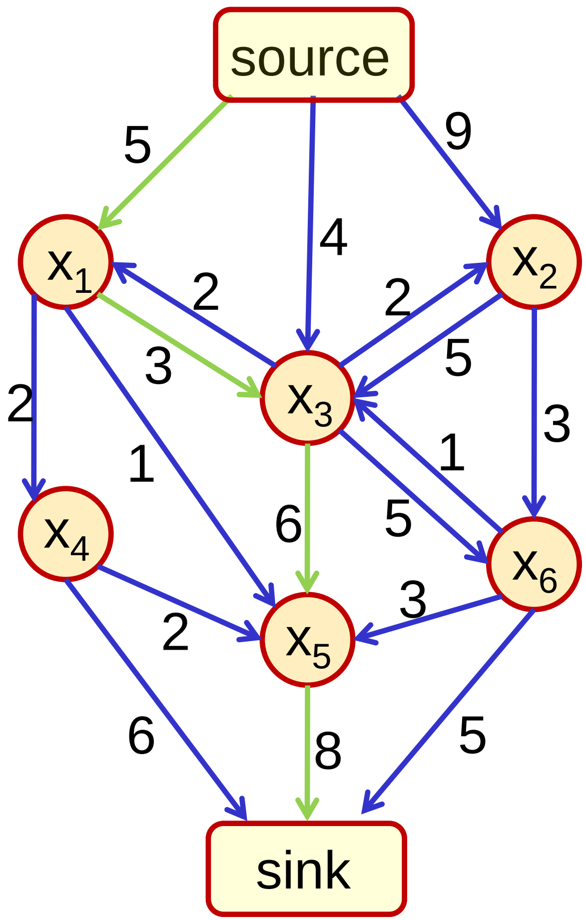
$$C(\boldsymbol{x}) = 5x_1 + 9x_2 + 4x_3 + 3x_3(1-x_1) + 2x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1)$$
$$+ 1x_5(1-x_1) + 6x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6)$$
$$+ 3x_6(1-x_2) + 2x_4(1-x_5) + 3x_6(1-x_5) + 6(1-x_4)$$
$$+ 8(1-x_5) + 5(1-x_6)$$

Min-Cut Problem



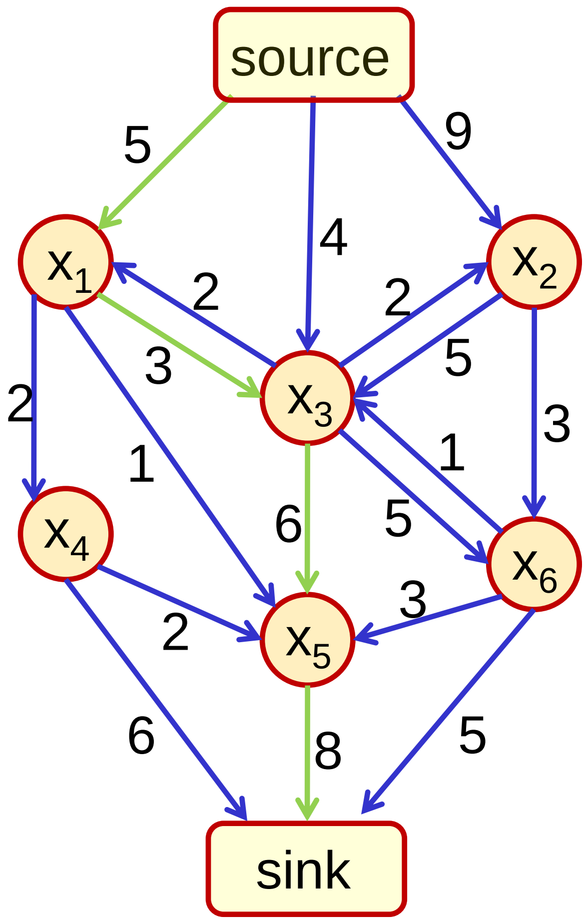
$$
\begin{aligned}
C(\boldsymbol{x}) = {} & 5x_1 + 9x_2 + 4x_3 + 3x_3(1-x_1) + 2x_1(1-x_3) \\
& + 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1) \\
& + 1x_5(1-x_1) + 6x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6) \\
& + 3x_6(1-x_2) + 2x_4(1-x_5) + 3x_6(1-x_5) + 6(1-x_4) \\
& + 8(1-x_5) + 5(1-x_6)
\end{aligned}
$$

Min-Cut Problem



$$C(\mathbf{x}) = 2x_1 + 9x_2 + 4x_3 + 2x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1)$$
$$+ 1x_5(1-x_1) + 3x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6)$$
$$+ 3x_6(1-x_2) + 2x_4(1-x_5) + 3x_6(1-x_5) + 6(1-x_4)$$
$$+ 5(1-x_5) + 5(1-x_6)$$
$$+ 3x_1 + 3x_3(1-x_1) + 3x_5(1-x_3) + 3(1-x_5)$$

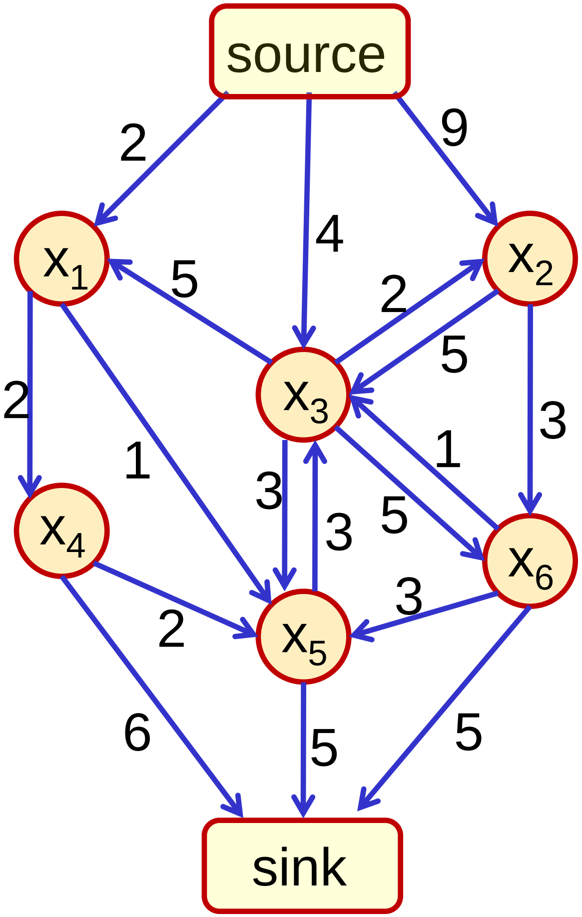
Min-Cut Problem



$$C(\mathbf{x}) = 2x_1 + 9x_2 + 4x_3 + 2x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1)$$
$$+ 1x_5(1-x_1) + 3x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6)$$
$$+ 3x_6(1-x_2) + 2x_4(1-x_5) + 3x_6(1-x_5) + 6(1-x_4)$$
$$+ 5(1-x_5) + 5(1-x_6)$$
$$+ 3x_1 + 3x_3(1-x_1) + 3x_5(1-x_3) + 3(1-x_5)$$

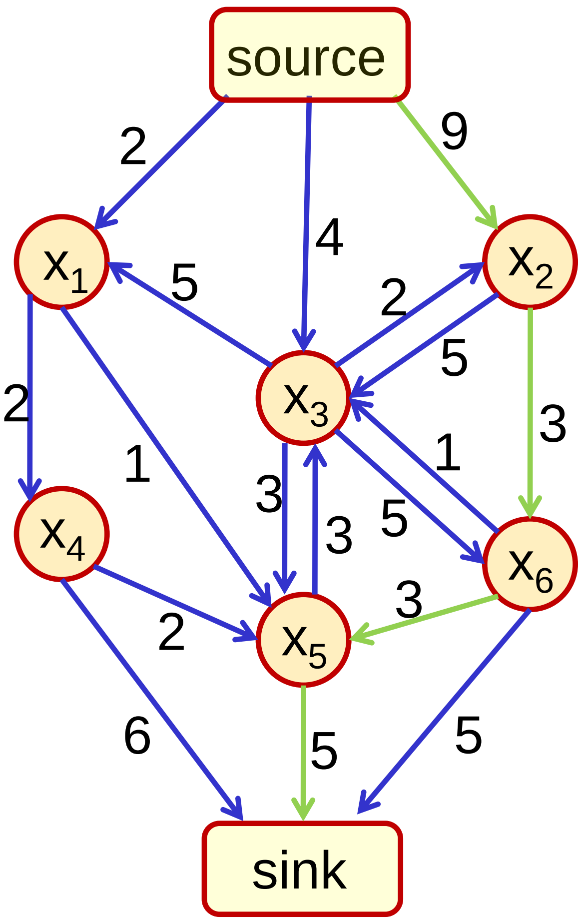
$$3x_1 + 3x_3(1-x_1) + 3x_5(1-x_3) + 3(1-x_5)$$
$$=$$
$$3 + 3x_1(1-x_3) + 3x_3(1-x_5)$$

Min-Cut Problem



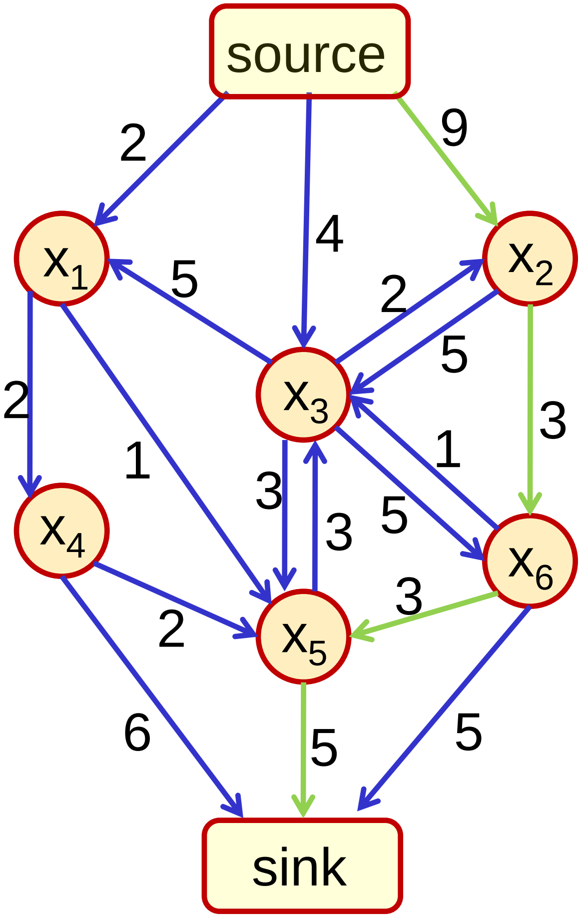
$$C(\boldsymbol{x}) = 3 + 2x_1 + 9x_2 + 4x_3 + 5x_1(1\text{-}x_3)$$
$$+ 3x_3(1\text{-}x_1) + 2x_2(1\text{-}x_3) + 5x_3(1\text{-}x_2) + 2x_4(1\text{-}x_1)$$
$$+ 1x_5(1\text{-}x_1) + 3x_5(1\text{-}x_3) + 5x_6(1\text{-}x_3) + 1x_3(1\text{-}x_6)$$
$$+ 3x_6(1\text{-}x_2) + 2x_4(1\text{-}x_5) + 3x_6(1\text{-}x_5) + 6(1\text{-}x_4)$$
$$+ 5(1\text{-}x_5) + 5(1\text{-}x_6) + 3x_3(1\text{-}x_5)$$

Min-Cut Problem



$$C(\boldsymbol{x}) = 3 + 2x_1 + 9x_2 + 4x_3 + 5x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1)$$
$$+ 1x_5(1-x_1) + 3x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6)$$
$$+ 3x_6(1-x_2) + 2x_4(1-x_5) + 3x_6(1-x_5) + 6(1-x_4)$$
$$+ 5(1-x_5) + 5(1-x_6) + 3x_3(1-x_5)$$

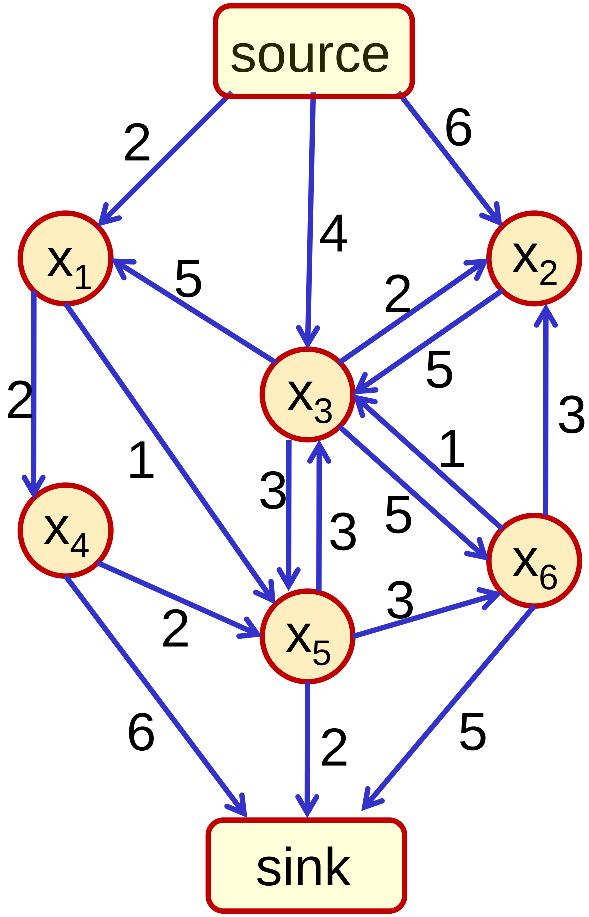
Min-Cut Problem



$$C(\boldsymbol{x}) = 3 + 2x_1 + 6x_2 + 4x_3 + 5x_1(1-x_3)$$
$$+\ 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1)$$
$$+\ 1x_5(1-x_1) + 3x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6)$$
$$+\ 2x_5(1-x_4) + 6(1-x_4)$$
$$+\ 2(1-x_5) + 5(1-x_6) + 3x_3(1-x_5)$$
$$+\ 3x_2 + 3x_6(1-x_2) + 3x_5(1-x_6) + 3(1-x_5)$$

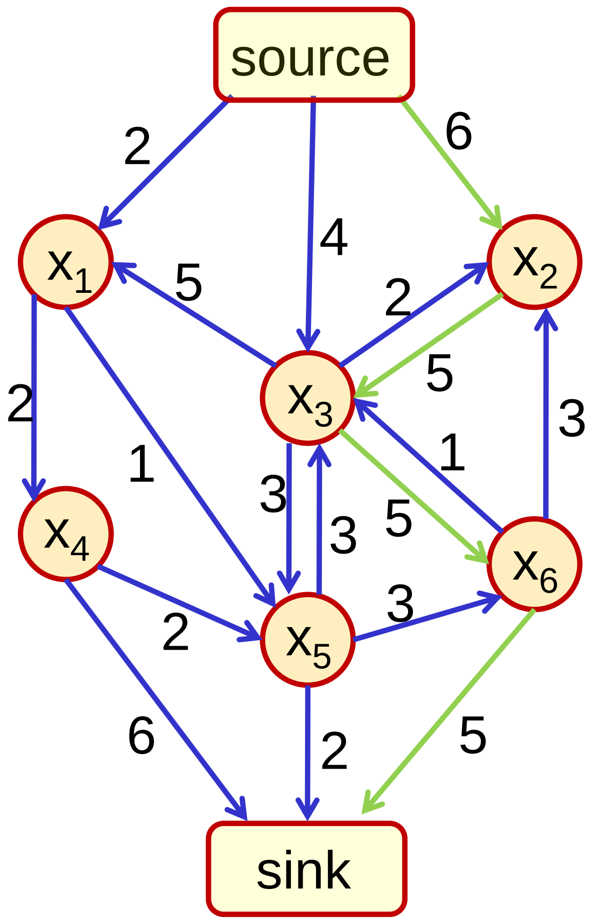
$$\boxed{\begin{gathered} 3x_2 + 3x_6(1-x_2) + 3x_5(1-x_6) + 3(1-x_5) \\ = \\ 3 + 3x_2(1-x_6) + 3x_6(1-x_5) \end{gathered}}$$

Min-Cut Problem



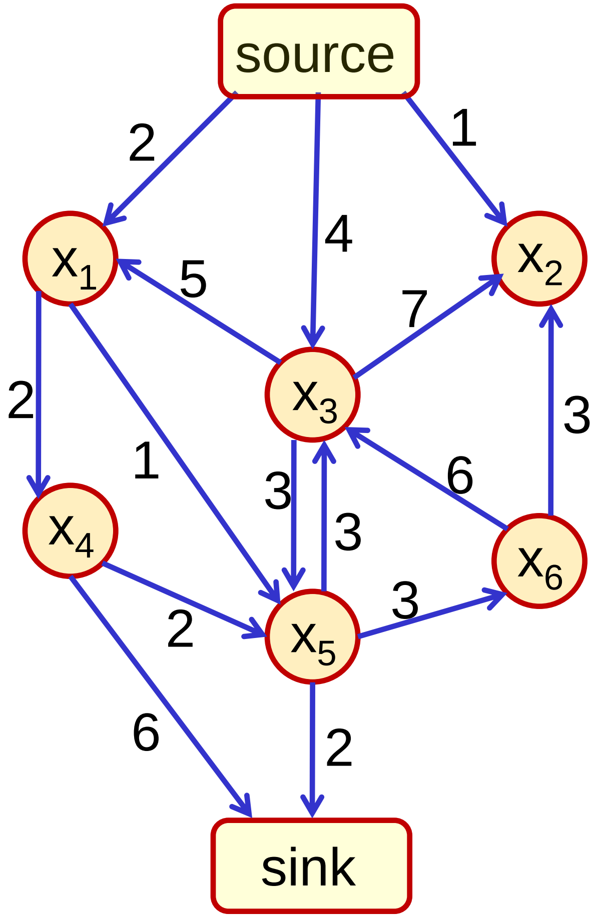
$$C(\boldsymbol{x}) = 6 + 2x_1 + 6x_2 + 4x_3 + 5x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1)$$
$$+ 1x_5(1-x_1) + 3x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6)$$
$$+ 2x_5(1-x_4) + 6(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5)$$
$$+ 2(1-x_5) + 5(1-x_6) + 3x_3(1-x_5)$$

Min-Cut Problem



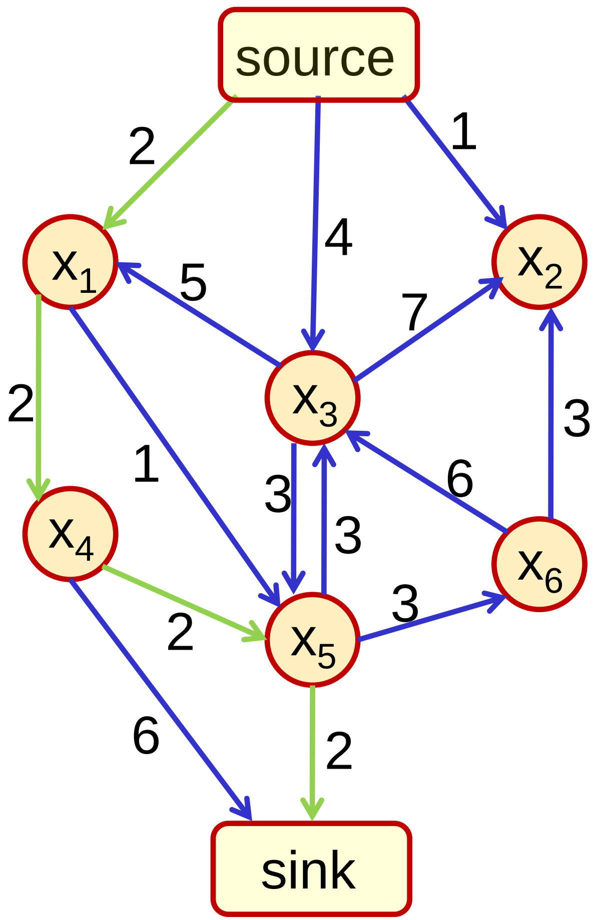
$$
\begin{aligned}
C(\mathbf{x}) = {} & 6 + 2x_1 + 6x_2 + 4x_3 + 5x_1(1-x_3) \\
& + 3x_3(1-x_1) + 2x_2(1-x_3) + 5x_3(1-x_2) + 2x_4(1-x_1) \\
& + 1x_5(1-x_1) + 3x_5(1-x_3) + 5x_6(1-x_3) + 1x_3(1-x_6) \\
& + 2x_5(1-x_4) + 6(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5) \\
& + 2(1-x_5) + 5(1-x_6) + 3x_3(1-x_5)
\end{aligned}
$$

Min-Cut Problem



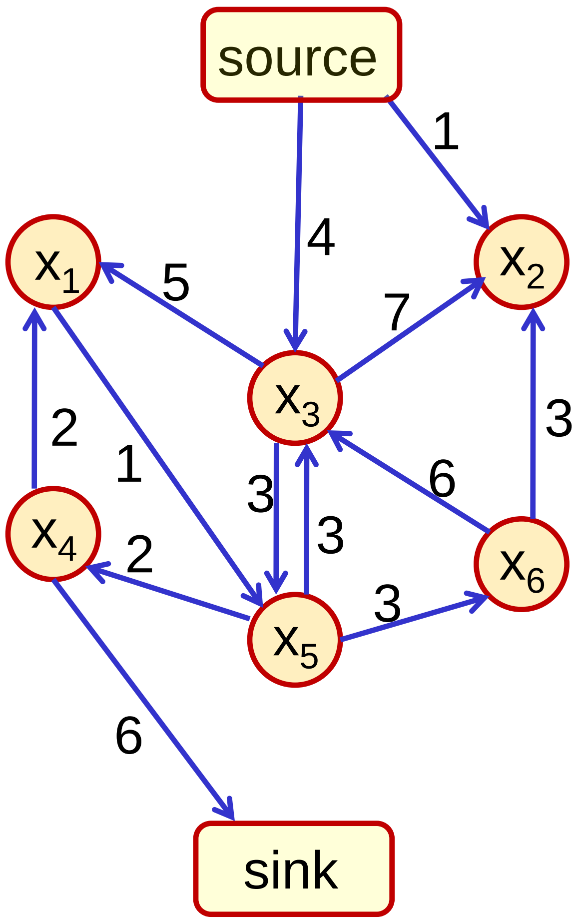
$$C(\mathbf{x}) = 11 + 2x_1 + 1x_2 + 4x_3 + 5x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_4(1-x_1)$$
$$+ 1x_5(1-x_1) + 3x_5(1-x_3) + 6x_3(1-x_6)$$
$$+ 2x_5(1-x_4) + 6(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5)$$
$$+ 2(1-x_5) + 3x_3(1-x_5)$$

Min-Cut Problem



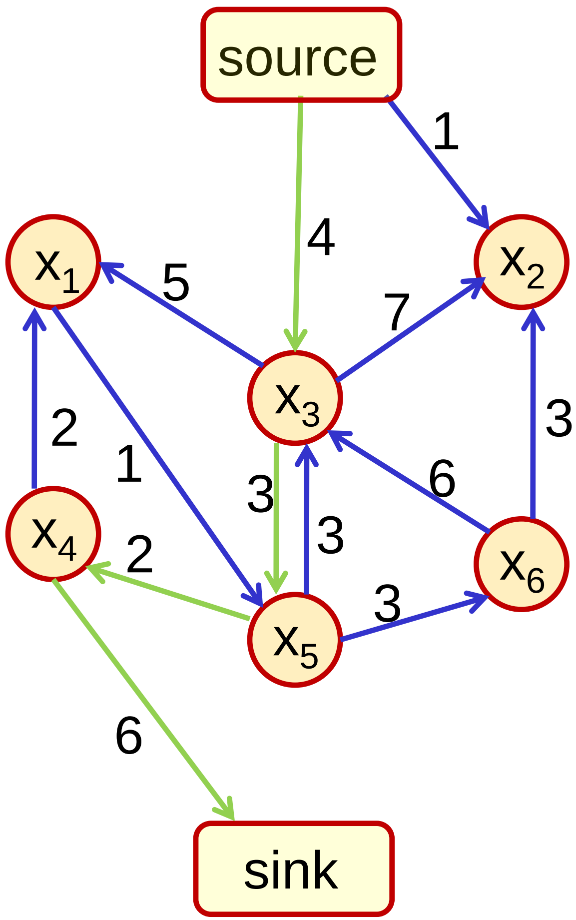
$$
\begin{aligned}
C(\mathbf{x}) = {} & 11 + 2x_1 + 1x_2 + 4x_3 + 5x_1(1-x_3) \\
& + 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_4(1-x_1) \\
& + 1x_5(1-x_1) + 3x_5(1-x_3) + 6x_3(1-x_6) \\
& + 2x_5(1-x_4) + 6(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5) \\
& + 2(1-x_5) + 3x_3(1-x_5)
\end{aligned}
$$

Min-Cut Problem



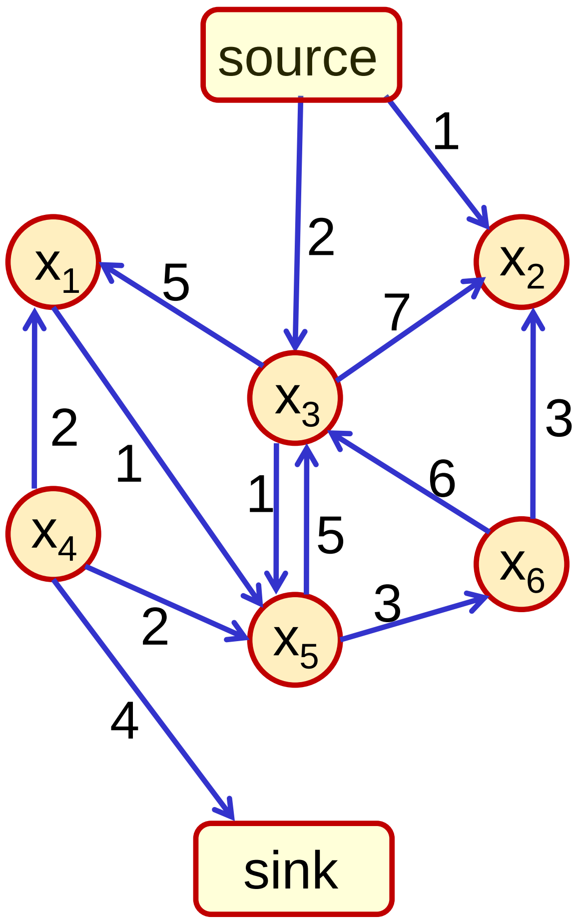
$$
\begin{aligned}
C(\boldsymbol{x}) = 13 &+ 1x_2 + 4x_3 + 5x_1(1-x_3) \\
&+ 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_1(1-x_4) \\
&+ 1x_5(1-x_1) + 3x_5(1-x_3) + 6x_3(1-x_6) \\
&+ 2x_4(1-x_5) + 6(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5) \\
&+ 3x_3(1-x_5)
\end{aligned}
$$

Min-Cut Problem



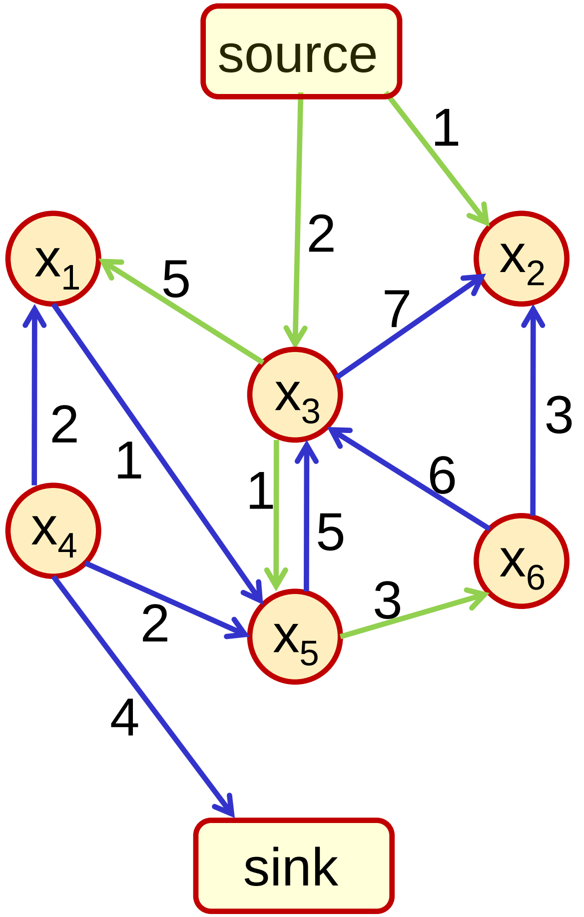
$$
\begin{aligned}
C(\boldsymbol{x}) = {} & 13 + 1x_2 + 2x_3 + 5x_1(1-x_3) \\
& + 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_1(1-x_4) \\
& + 1x_5(1-x_1) + 3x_5(1-x_3) + 6x_3(1-x_6) \\
& + 2x_4(1-x_5) + 6(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5) \\
& + 3x_3(1-x_5)
\end{aligned}
$$

Min-Cut Problem



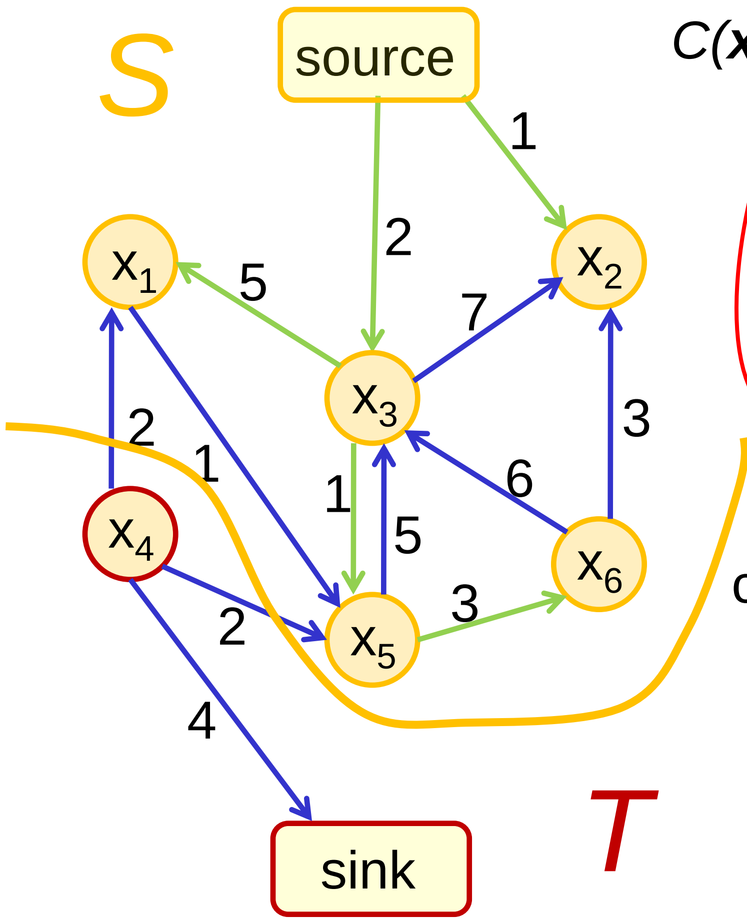
$$
\begin{aligned}
C(\boldsymbol{x}) = 15 &+ 1x_2 + 4x_3 + 5x_1(1-x_3) \\
&+ 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_1(1-x_4) \\
&+ 1x_5(1-x_1) + 6x_3(1-x_6) + 6x_3(1-x_5) \\
&+ 2x_5(1-x_4) + 4(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5)
\end{aligned}
$$

Min-Cut Problem



$$C(\boldsymbol{x}) = 15 + 1x_2 + 4x_3 + 5x_1(1-x_3)$$
$$+ 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_1(1-x_4)$$
$$+ 1x_5(1-x_1) + 6x_3(1-x_6) + 6x_3(1-x_5)$$
$$+ 2x_5(1-x_4) + 4(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5)$$

Min-Cut Problem



cost = 0

min cut

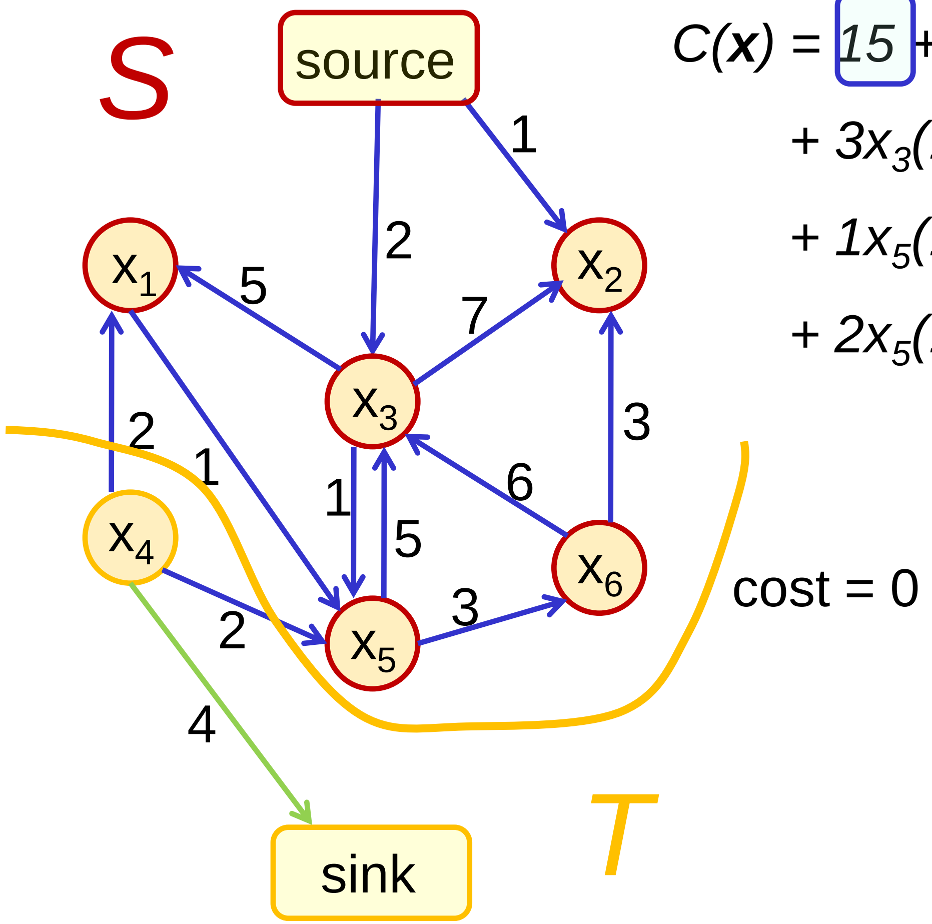
$$
\begin{aligned}
C(\mathbf{x}) = \boxed{15} &+ 1x_2 + 4x_3 + 5x_1(1-x_3) \\
&+ 3x_3(1-x_1) + 7x_2(1-x_3) + 2x_1(1-x_4) \\
&+ 1x_5(1-x_1) + 6x_3(1-x_6) + 6x_3(1-x_5) \\
&+ 2x_5(1-x_4) + 4(1-x_4) + 3x_2(1-x_6) + 3x_6(1-x_5)
\end{aligned}
$$

cost = 0

- All coefficients positive
- Must be global minimum

S – set of reachable nodes from s

Min-Cut Problem



min cut

$$C(\boldsymbol{x}) = \boxed{15} + 1x_2 + 4x_3 + 5x_1(1\text{-}x_3)$$
$$+ 3x_3(1\text{-}x_1) + 7x_2(1\text{-}x_3) + 2x_1(1\text{-}x_4)$$
$$+ 1x_5(1\text{-}x_1) + 6x_3(1\text{-}x_6) + 6x_3(1\text{-}x_5)$$
$$+ 2x_5(1\text{-}x_4) + 4(1\text{-}x_4) + 3x_2(1\text{-}x_6) + 3x_6(1\text{-}x_5)$$

- All coefficients positive
- Must be global minimum

T – set of nodes that can reach t

(not necesarly the same)

UNIVERSITY OF OXFORD

Overview

- Motivation
- Min-Cut / Max-Flow (Graph Cut) Algorithm
- Markov and Conditional Random Fields
- Random Field Optimisation using Graph Cuts
    - Submodular vs. Non-Submodular Problems
    - Pairwise vs. Higher Order Problems
    - 2-Label vs. Multi-Label Problems
- Recent Advances in Random Field Optimisation
- Conclusions

Markov and Conditional RF

- Markov / Conditional Random fields model probabilistic dependencies of the set of random variables

Markov and Conditional RF

- Markov / Conditional Random fields model conditional dependencies between random variables
- Each variable is conditionally independent of all other variables given its neighbours

Markov and Conditional RF

- Markov / Conditional Random fields model conditional dependencies between random variables
- Each variable is conditionally independent of all other variables given its neighbours
- Posterior probability of the labelling *x* given data *D* is :

$$\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{\boxed{Z}} \exp(-\sum_{\boxed{c \in \mathcal{C}}} \boxed{\psi_c(\mathbf{x}_c)})$$

partition function cliques potential functions

Markov and Conditional RF

- Markov / Conditional Random fields model conditional dependencies between random variables
- Each variable is conditionally independent of all other variables given its neighbours
- Posterior probability of the labelling *x* given data *D* is :

$$\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{\underbrace{\boxed{Z}}_{\text{partition function}}} \exp\Big(-\sum_{\underbrace{\boxed{c \in \mathcal{C}}}_{\text{cliques}}} \underbrace{\boxed{\psi_c(\mathbf{x}_c)}}_{\text{potential functions}}\Big)$$

- Energy of the labelling is defined as :

$$E(\mathbf{x}) = -\log \Pr(\mathbf{x}|\mathbf{D}) - \log Z = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c)$$

Markov and Conditional RF

- The most probable (Max a Posteriori (MAP)) labelling is defined as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{L}} \Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x}\in\mathbf{L}} E(\mathbf{x})$$

Markov and Conditional RF

- The most probable (Max a Posteriori (MAP)) labelling is defined as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{L}} \Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x}\in\mathbf{L}} E(\mathbf{x})$$

- The only distinction (MRF vs. CRF) is that in the CRF the conditional dependencies between variables depend also on the data

Markov and Conditional RF

- The most probable (Max a Posteriori (MAP)) labelling is defined as:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{L}} \Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x}\in\mathbf{L}} E(\mathbf{x})$$

- The only distinction (MRF vs. CRF) is that in the CRF the conditional dependencies between variables depend also on the data
- Typically we define an energy first and then pretend there is an underlying probabilistic distribution there, but there isn't really (Pssssst, don't tell anyone)

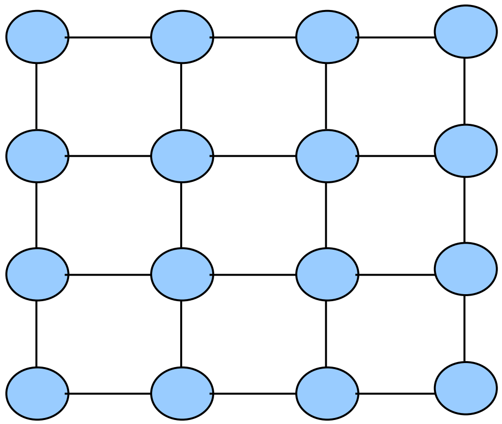
Overview

- Motivation
- Min-Cut / Max-Flow (Graph Cut) Algorithm
- Markov and Conditional Random Fields
- Random Field Optimisation using Graph Cuts
    - Submodular vs. Non-Submodular Problems
    - Pairwise vs. Higher Order Problems
    - 2-Label vs. Multi-Label Problems
- Recent Advances in Random Field Optimisation
- Conclusions

Pairwise CRF models

Standard CRF Energy

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



Graph Cut based Inference

The energy / potential is called submodular (only) if for every pair of variables :

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

Graph Cut based Inference

For 2-label problems x∈{0, 1} :

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

Graph Cut based Inference

For 2-label problems x∈{0, 1} :

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

Pairwise potential can be transformed into :

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

where

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

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

Graph Cut based Inference

For 2-label problems x∈{0, 1} :

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

Pairwise potential can be transformed into :

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

where

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

Graph Cut based Inference

For 2-label problems x∈{0, 1} :

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

Pairwise potential can be transformed into :

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

where

$$
\begin{aligned}
K_{ij} &= g_{ij}^{00} \\
g_i' &= \frac{g_{ij}^{10} + g_{ij}^{11} - g_{ij}^{01} - g_{ij}^{00}}{2}
\end{aligned}
\qquad
\begin{aligned}
g_j' &= \frac{g_{ij}^{01} + g_{ij}^{11} - g_{ij}^{10} - g_{ij}^{00}}{2} \\
c_{ij} &= \frac{g_{ij}^{01} + g_{ij}^{10} - g_{ij}^{00} - g_{ij}^{11}}{2} \geq 0
\end{aligned}
$$

Graph Cut based Inference

After summing up :

$$E(\mathbf{x}) = K + \sum_{i \in \mathcal{V}} c_i x_i + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} c_{ij}(1 - x_i)x_j + c_{ij}x_i(1 - x_j)$$

Graph Cut based Inference

After summing up :

$$E(\mathbf{x}) = K + \sum_{i \in \mathcal{V}} c_i x_i + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} c_{ij}(1 - x_i)x_j + c_{ij} x_i (1 - x_j)$$

Let :

$$\begin{aligned} c_{it} &= c_i \text{ and } c_{si} = 0 && \text{if } c_i \geq 0. \\ c_{it} &= 0 \text{ and } c_{si} = -c_i && \text{otherwise} \end{aligned}$$

Graph Cut based Inference

After summing up :

$$E(\mathbf{x}) = K + \sum_{i \in \mathcal{V}} c_i x_i + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} c_{ij}(1 - x_i)x_j + c_{ij}x_i(1 - x_j)$$

Let :

$$\begin{array}{ll} c_{it} = c_i \text{ and } c_{si} = 0 & \text{if } c_i \geq 0 \\ c_{it} = 0 \text{ and } c_{si} = -c_i & \text{otherwise} \end{array}$$

Then :

$$\begin{aligned} c_i x_i &= c_{it} x_i \\ c_i x_i &= -c_{si} + c_{si}(1 - x_i) \end{aligned}$$

Graph Cut based Inference

After summing up :

$$E(\mathbf{x}) = K + \sum_{i \in \mathcal{V}} c_i x_i + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} c_{ij}(1 - x_i)x_j + c_{ij}x_i(1 - x_j)$$

Let :

$$\begin{aligned} c_{it} &= c_i \text{ and } c_{si} = 0 && \text{if } c_i \geq 0. \\ c_{it} &= 0 \text{ and } c_{si} = -c_i && \text{otherwise} \end{aligned}$$

Then :

$$\begin{aligned} c_i x_i &= c_{it} x_i \\ c_i x_i &= -c_{si} + c_{si}(1 - x_i) \end{aligned}$$

Equivalent st-mincut problem is :

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i \in \mathcal{V}} c_{si} x_i + c_{it}(1 - x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} c_{ij}(1 - x_i)x_j + c_{ij}x_i(1 - x_j)$$

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**

Data term

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

Estimated using FG / BG colour models

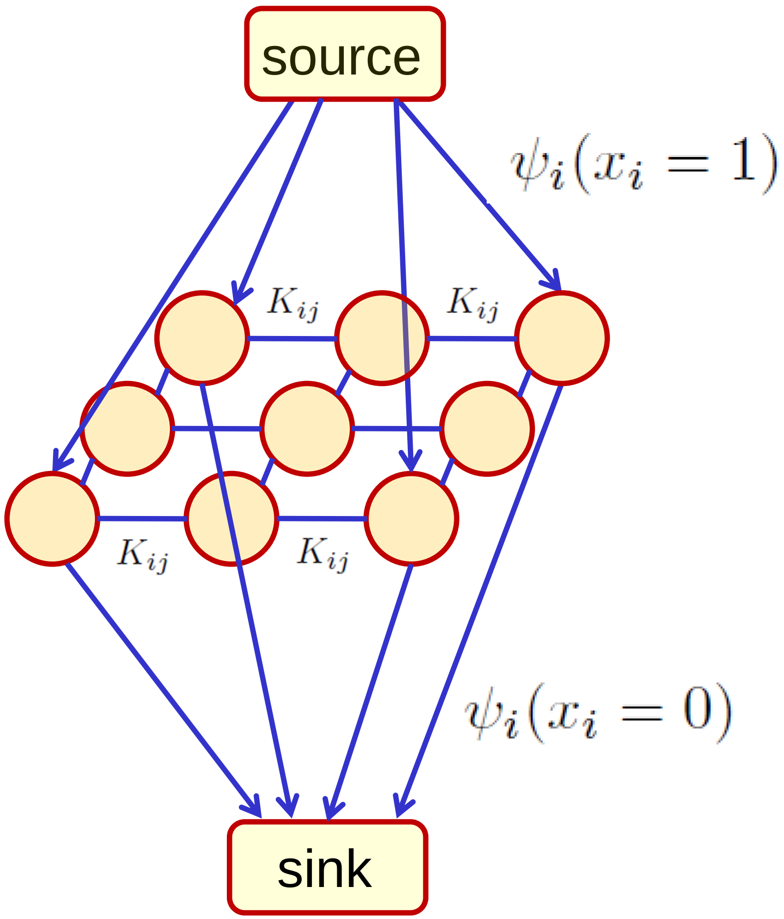
Smoothness term

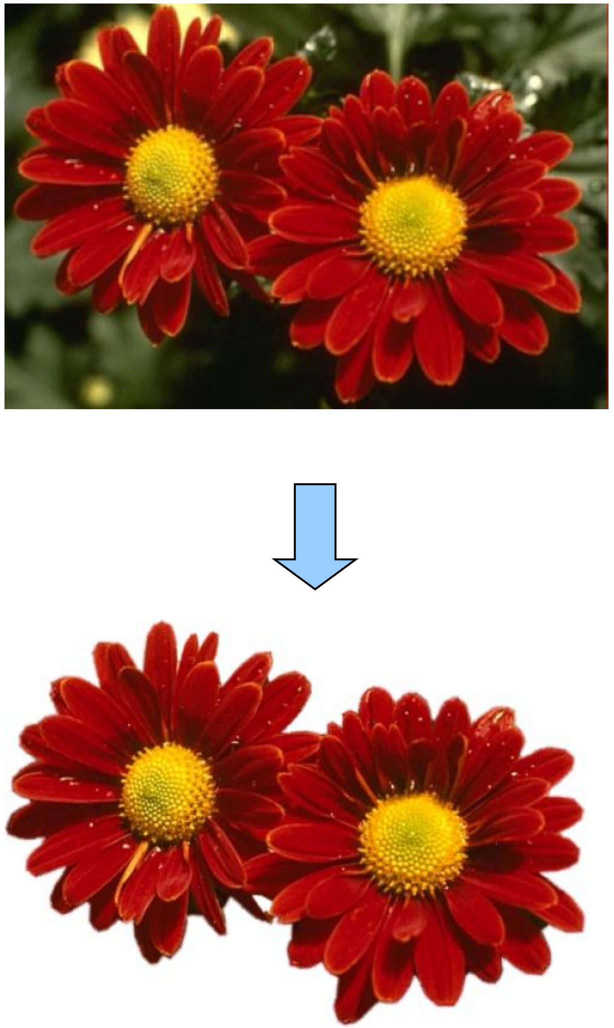
$$\psi_{ij}(x_i, x_j) = K_{ij}\delta(x_i \neq x_j)$$

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

Intensity dependent smoothness

Foreground / Background Estimation



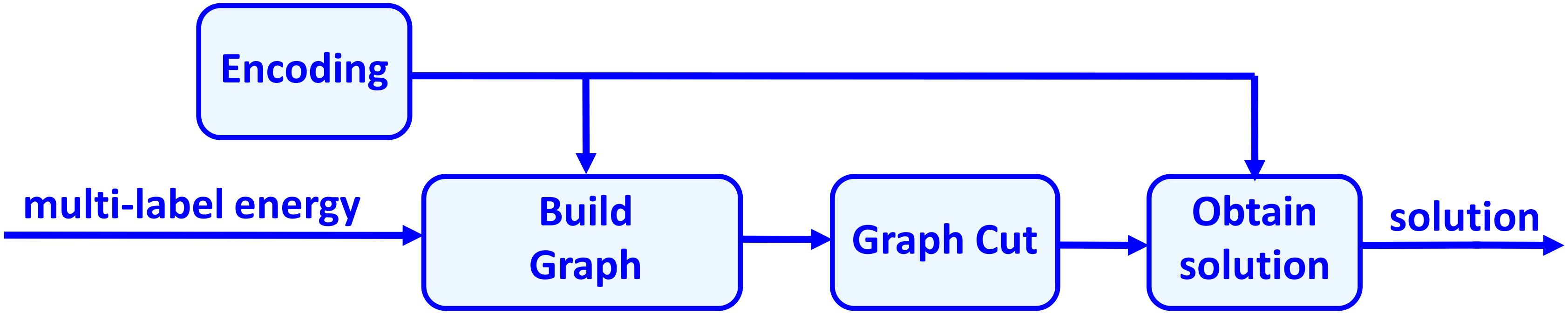


Rother et al. SIGGRAPH04

Graph Cut based Inference

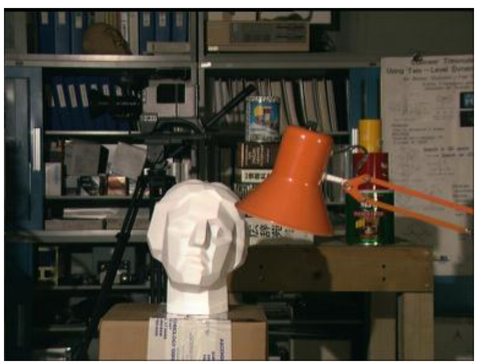
Extendable to (some) Multi-label CRFs

- each state of multi-label variable encoded using multiple binary variables
- the cost of every possible cut must be the same as the associated energy
- the solution obtained by inverting the encoding



Dense Stereo Estimation

- For each pixel assigns a disparity label : z
- Disparities from the discrete set {*0*, *1*, .. *D*}



Left Camera Image

Right Camera Image

Dense Stereo Result

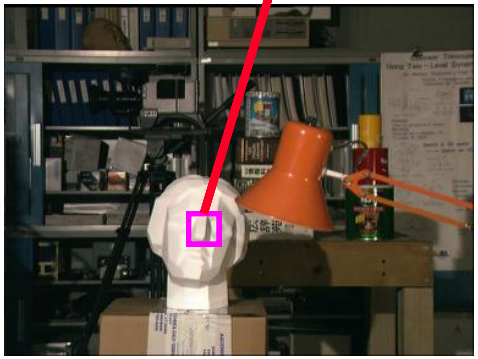
Dense Stereo Estimation

Data term

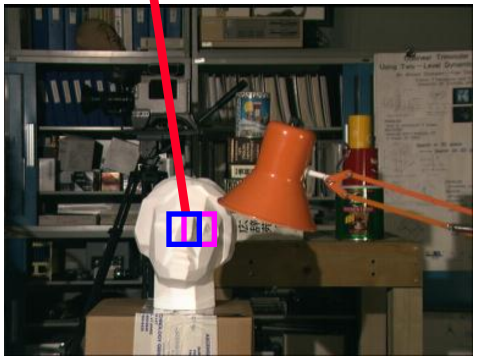
$$\psi_i(z_i) = d(\boxed{F^L(x_i, y_i)}, \boxed{F^R(x_i - z_i, y_i)})$$

Left image feature

Shifted right image feature



Left Camera Image



Right Camera Image

Dense Stereo Estimation

Data term

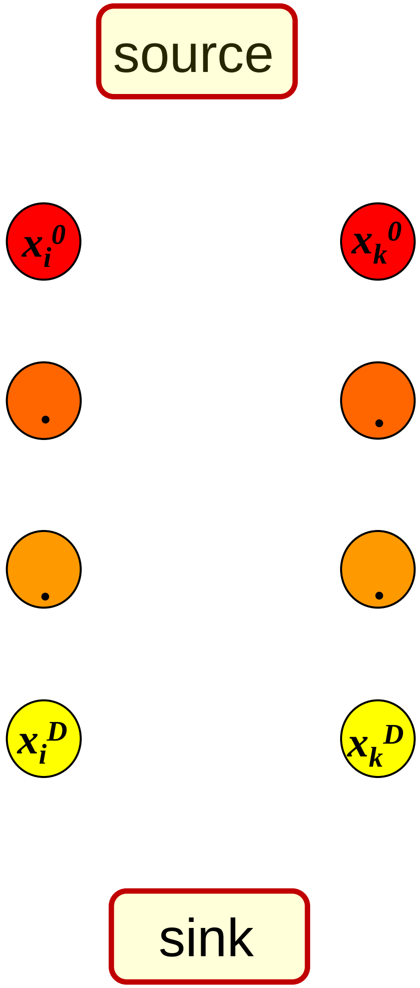
$$\psi_i(z_i) = d(\boxed{F^L(x_i, y_i)}, \boxed{F^R(x_i - z_i, y_i)})$$

Left image feature **Shifted right image feature**

Smoothness term

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

Dense Stereo Estimation

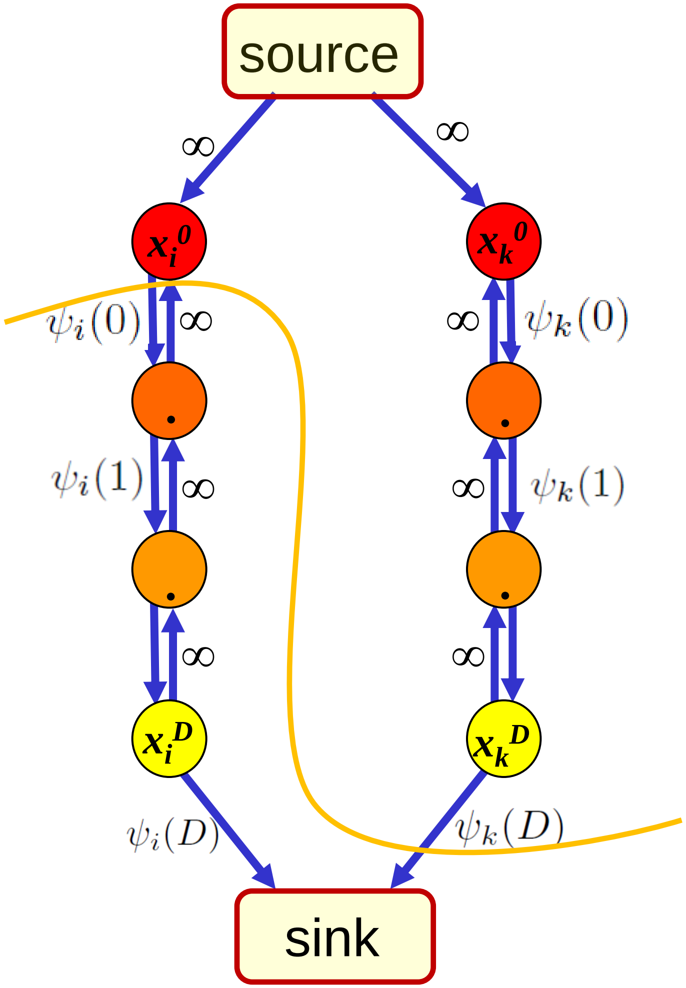


Encoding

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

Ishikawa PAMI03

Dense Stereo Estimation



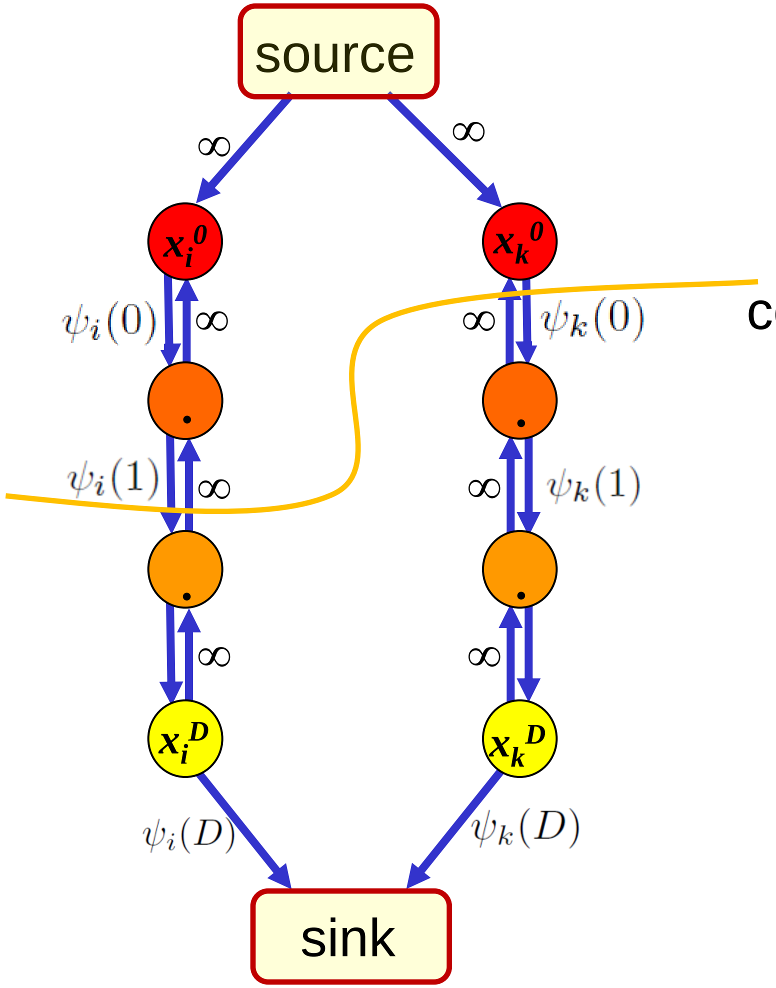
Unary Potential

The cost of every cut should be equal to the corresponding energy under the encoding

cost = $\psi_i(0) + \psi_k(D)$

Ishikawa PAMI03

Dense Stereo Estimation



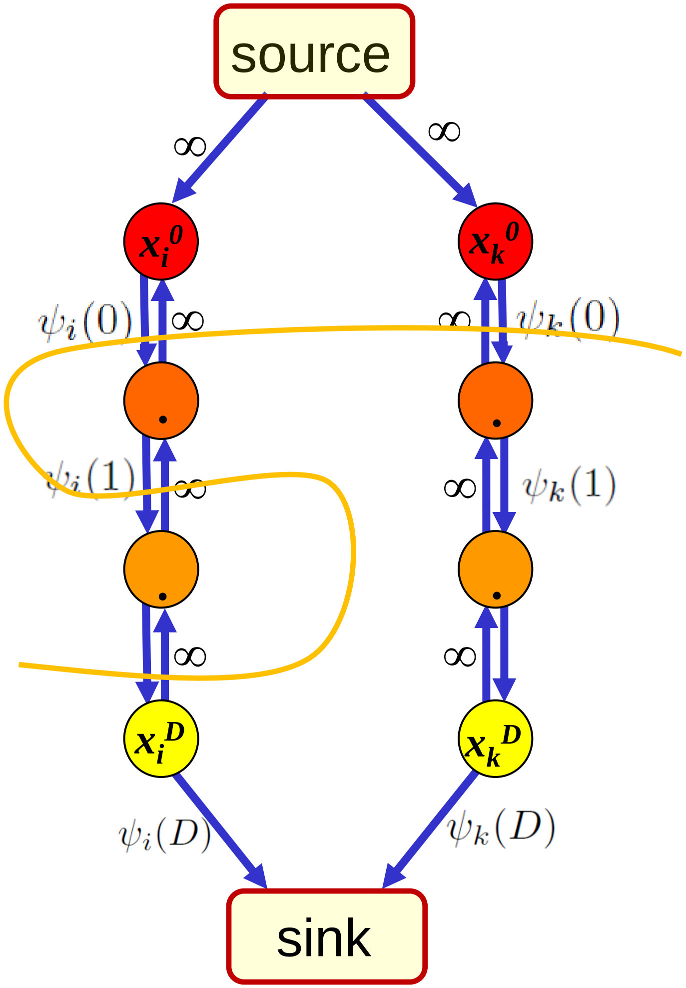
Unary Potential

cost = $\psi_i(1) + \psi_k(0)$

The cost of every cut should be equal to the corresponding energy under the encoding

Ishikawa PAMI03

Dense Stereo Estimation



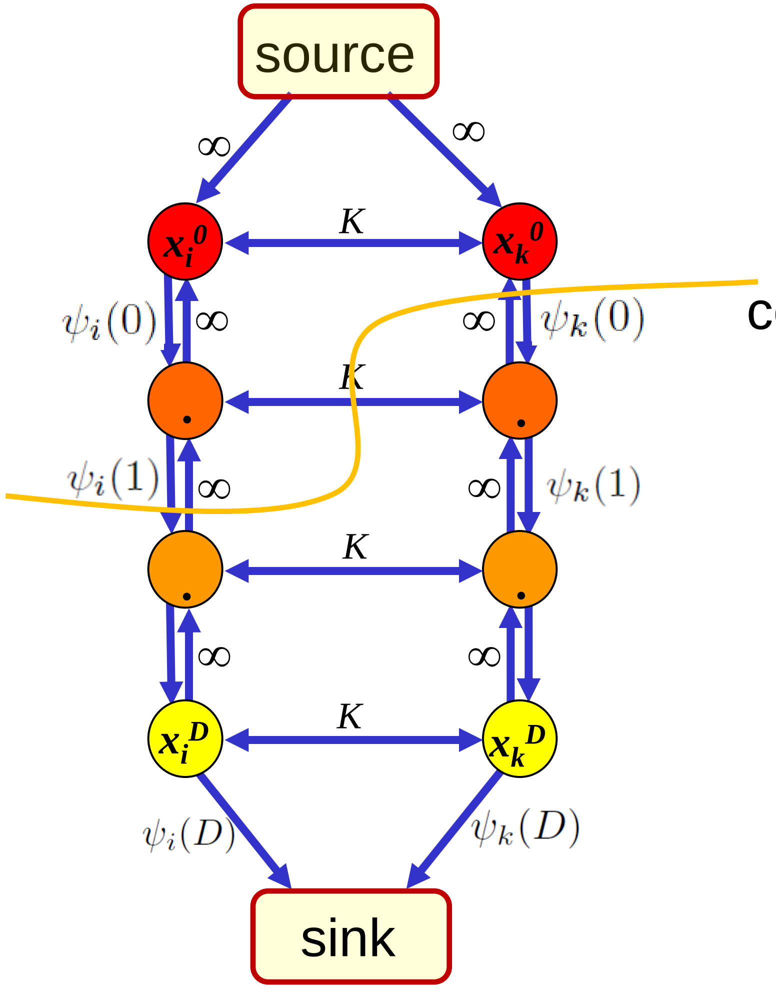
cost = ∞

Unary Potential

The cost of every cut should be equal to the corresponding energy under the encoding

Ishikawa PAMI03

Dense Stereo Estimation



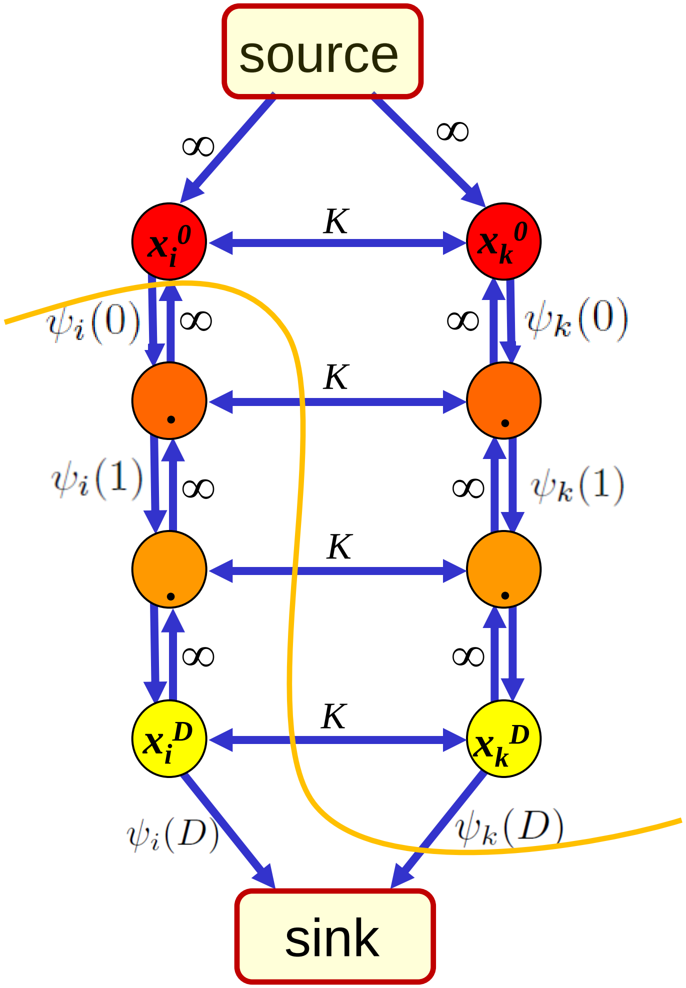
Pairwise Potential

cost = $\psi_i(1)$ + $\psi_k(0)$ + K

The cost of every cut should be equal to the corresponding energy under the encoding

Ishikawa PAMI03

Dense Stereo Estimation



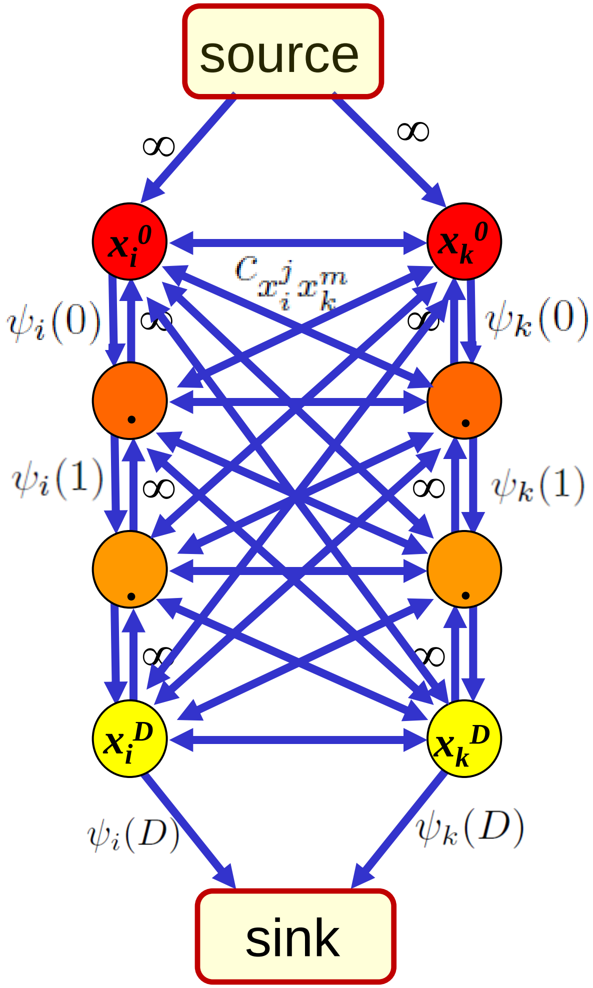
Pairwise Potential

The cost of every cut should be equal to the corresponding energy under the encoding

cost = $\psi_i(0)$ + $\psi_k(D)$ + D K

Ishikawa PAMI03

Dense Stereo Estimation



Extendable to any convex cost

$$\psi_{ij}(z_i, z_j) = \boxed{f(z_i - z_j)}$$

Convex function

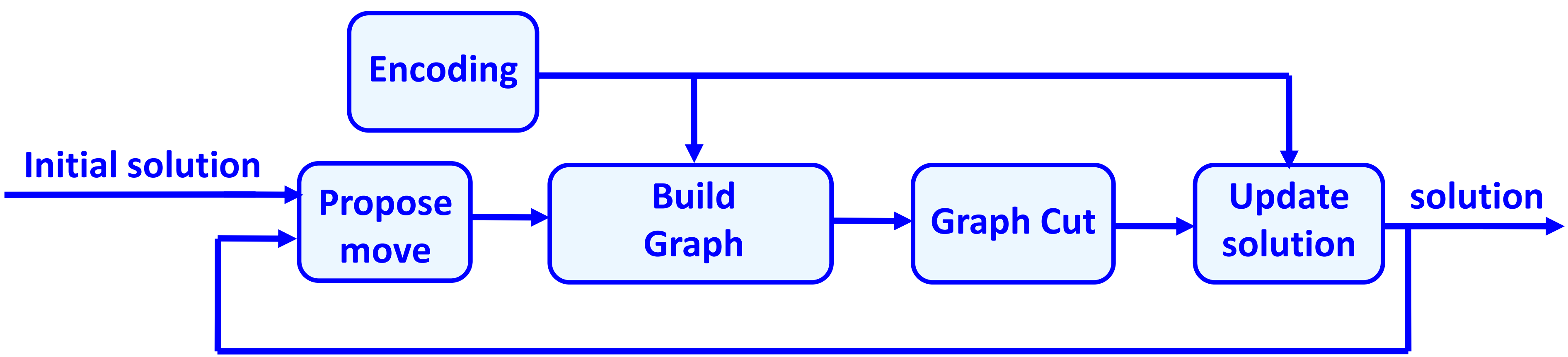
$$c_{x_i^j x_k^m} = c_{x_k^m x_i^j} = \frac{f(l_i - l_k + 1) - 2f(l_i - l_k) + f(l_i - l_k - 1)}{2}$$

See [*Ishikawa PAMI03]* for more details

Graph Cut based Inference

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



Graph Cut based Inference

Example : [Boykov et al. 01]

αβ-swap

- each variable taking label α or β can change its label to α or β
- all αβ-moves are iteratively performed till convergence

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

α-expansion

- each variable either keeps its old label or changes to α
- all α-moves are iteratively performed till convergence

Transformation function :

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

Graph Cut based Inference

Sufficient conditions for move making algorithms :
(all possible moves are submodular)

αβ-swap : semi-metricity

$$\forall\, l_a, l_b \in \mathcal{L}$$

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

Proof:

$$\psi^p(l_a, l_b) + \psi^p(l_b, l_a) - \psi^p(l_a, l_a) - \psi^p(l_b, l_b) = 2\psi^p(l_a, l_b) \geq 0$$

Graph Cut based Inference

Sufficient conditions for move making algorithms :
(all possible moves are submodular)

α-expansion : metricity

$$\forall\ l_a, l_b, l_c \in \mathcal{L}$$

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

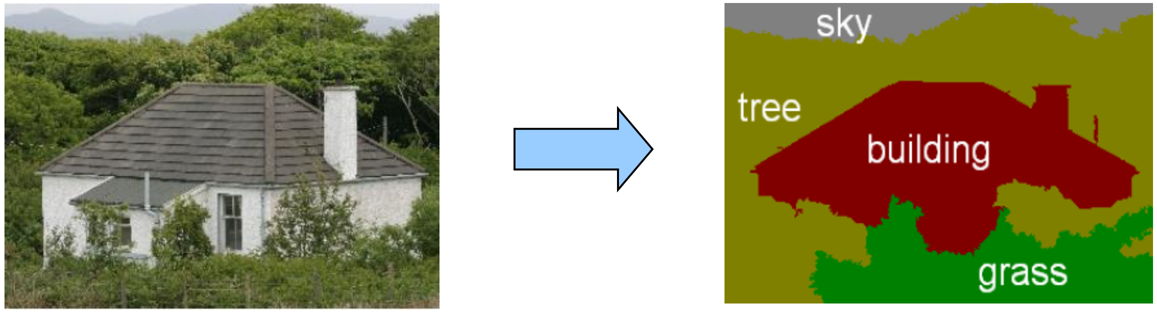
$$\psi^p(l_a, l_b) + \psi^p(l_b, l_c) \geq \psi^p(l_a, l_c)$$

Proof:

$$\psi^p(l_a, l_b) + \psi^p(l_c, l_a) - \psi^p(l_a, l_a) - \psi^p(l_b, l_c)$$

$$= \psi^p(l_a, l_b) + \psi^p(l_c, l_a) - \psi^p(l_b, l_c) \geq 0$$

Object-class Segmentation



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

Data term

Discriminatively trained classifier

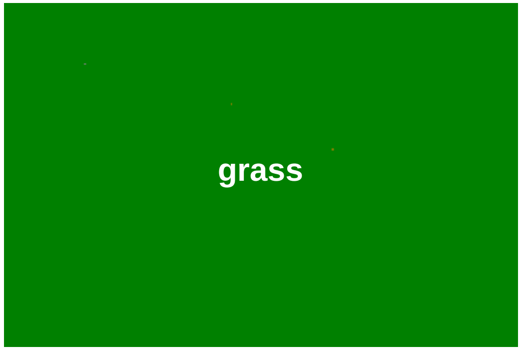
Smoothness term

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

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

Object-class Segmentation

Original Image

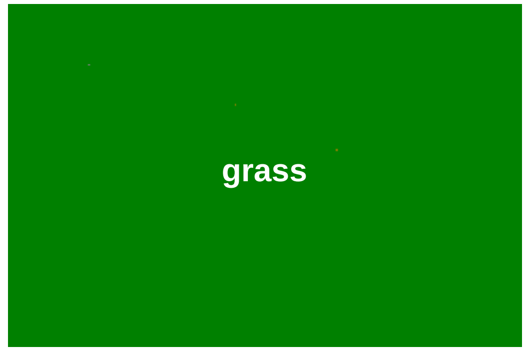


Initial solution

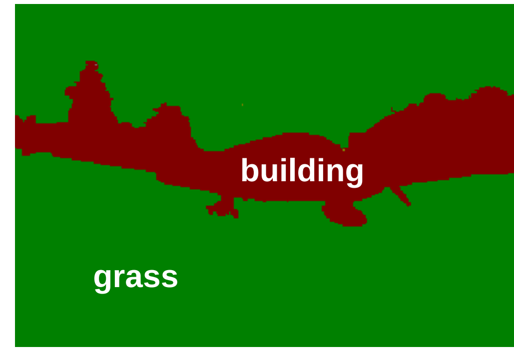
UNIVERSITY OF OXFORD

Object-class Segmentation

Original Image



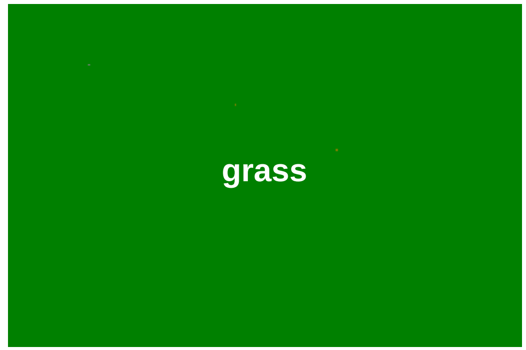
Initial solution



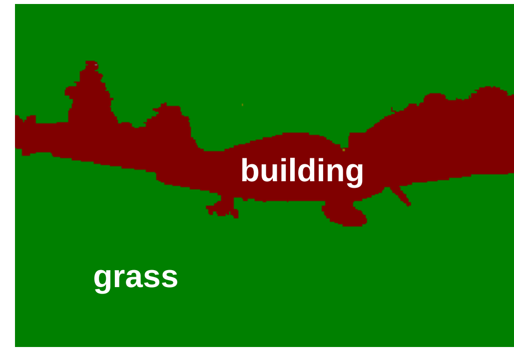
Building expansion

Object-class Segmentation

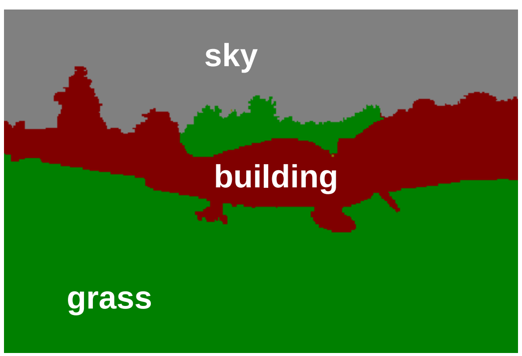
Original Image



Initial solution



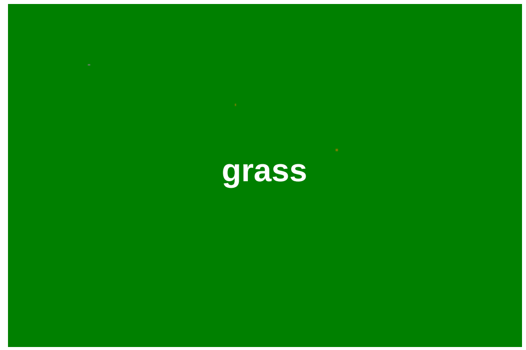
Building expansion



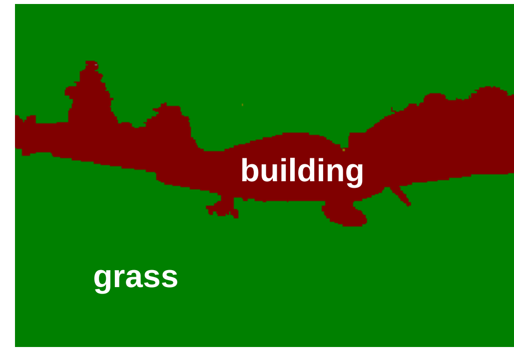
Sky expansion

Object-class Segmentation

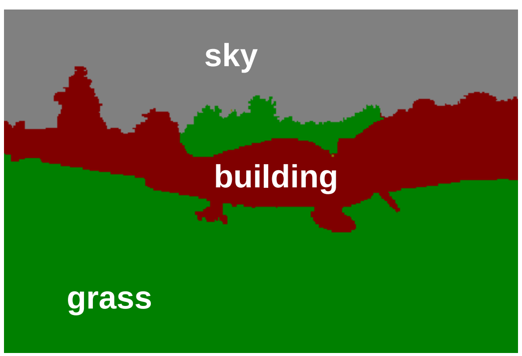
Original Image



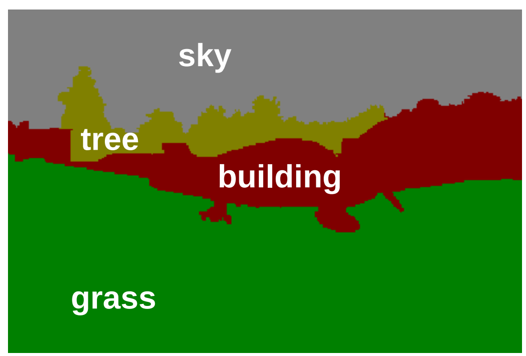
Initial solution



Building expansion



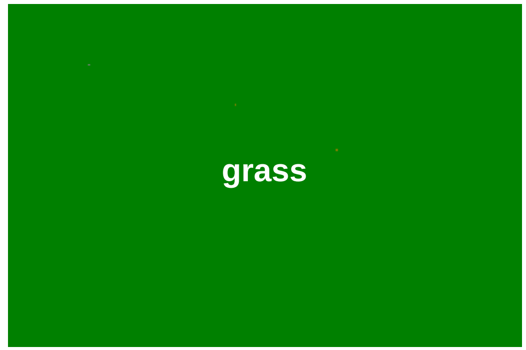
Sky expansion



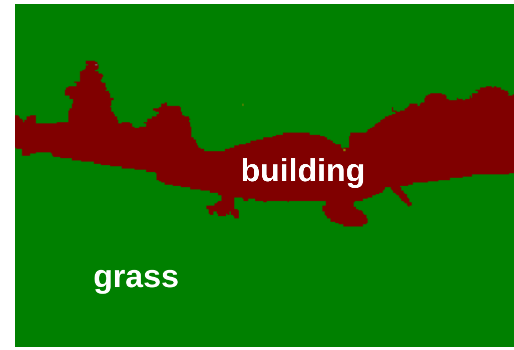
Tree expansion

Object-class Segmentation

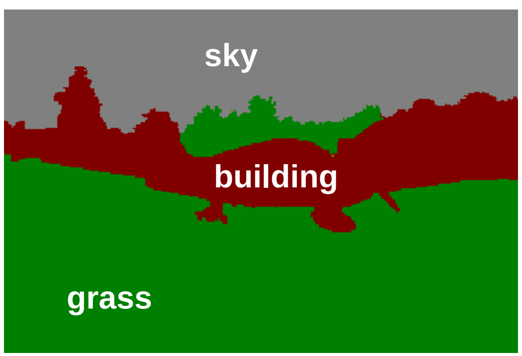
Original Image



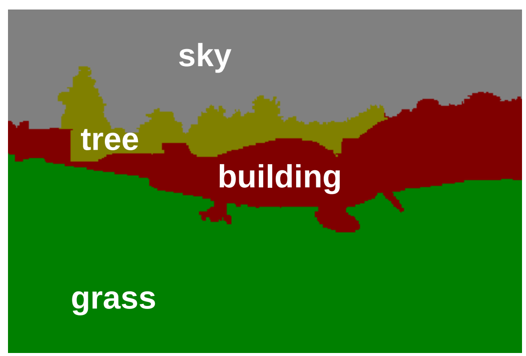
Initial solution



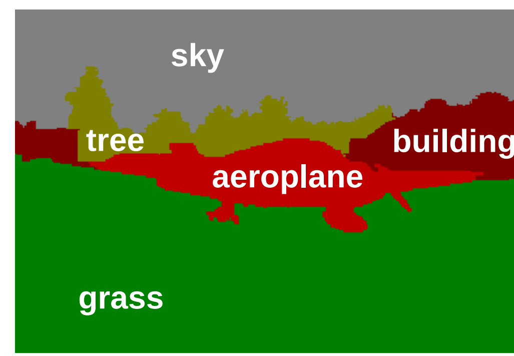
Building expansion



Sky expansion



Tree expansion



Final Solution

Graph Cut based Inference

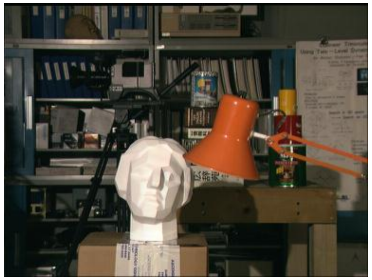
Range-(swap) moves [Veksler 07]

- Each variable in the convex range can change its label to any other label in the convex range
- all range-moves are iteratively performed till convergence

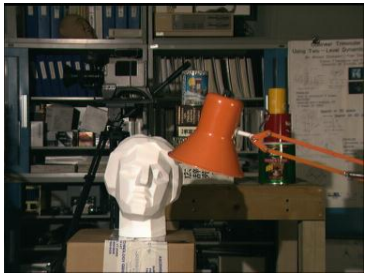
Range-expansion [Kumar, Veksler & Torr 1]

- Expansion version of range swap moves
- Each varibale can change its label to any label in a convex range or keep its old label

Dense Stereo Reconstruction



Left Camera Image



Right Camera Image

Dense Stereo Result

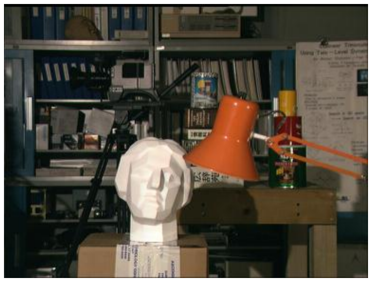
Data term

Same as before

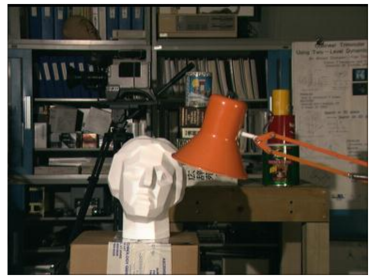
Smoothness term

$$\psi_{ij}(z_i, z_j) = \min(K|z_i - z_j|, T)$$

Dense Stereo Reconstruction



Left Camera Image



Right Camera Image

Dense Stereo Result

Data term

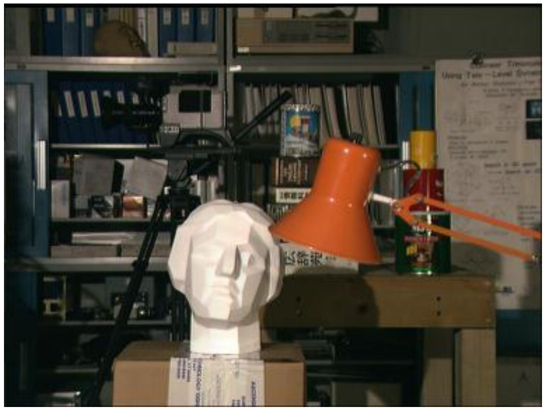
Same as before

Smoothness term

$$\psi_{ij}(z_i, z_j) = \min(\boxed{K|z_i - z_j|}, \boxed{T})$$

Convex range **Truncation**

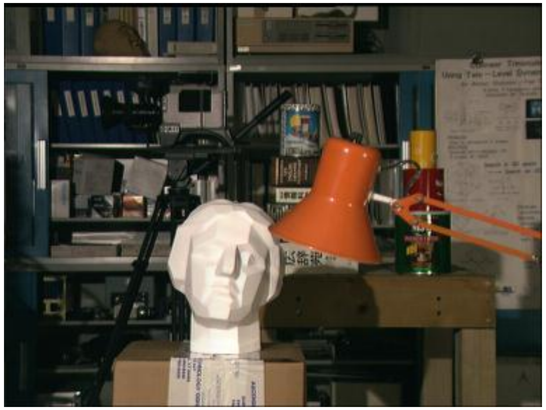
Graph Cut based Inference



Original Image

Initial Solution

Graph Cut based Inference

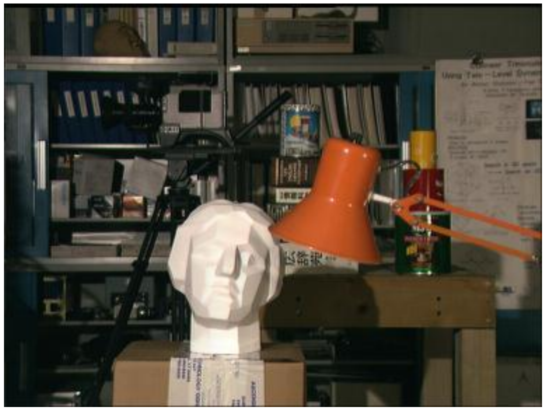


Original Image

Initial Solution

After 1st expansion

Graph Cut based Inference



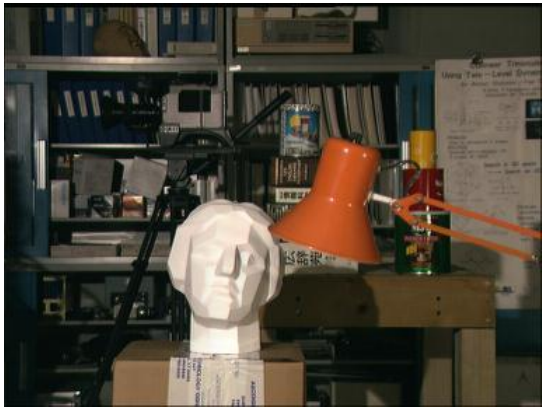
Original Image

Initial Solution

After 1st expansion

After 2nd expansion

Graph Cut based Inference


Original Image

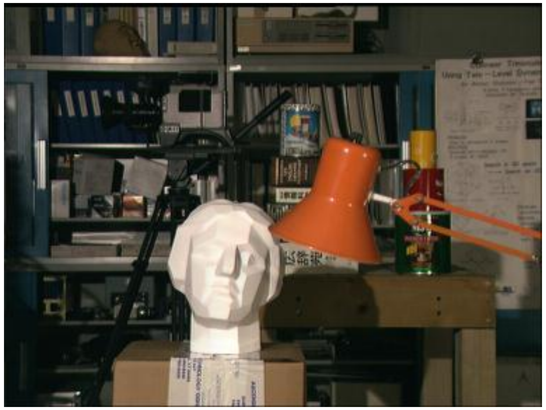

Initial Solution


After 1st expansion


After 2nd expansion


After 3rd expansion

Graph Cut based Inference



Original Image

Initial Solution

After 1st expansion

After 2nd expansion

After 3rd expansion

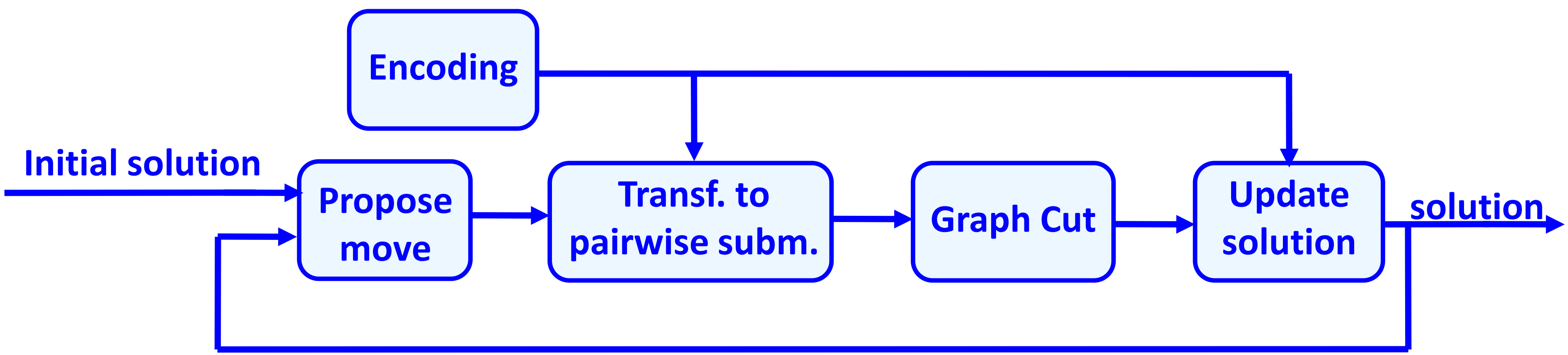
Final solution

Graph Cut based Inference

- Extendible to certain classes of binary higher order energies
- Higher order terms have to be transformable to the pairwise submodular energy with binary auxiliary variables $\mathbf{z}_C$

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

Higher order term **Pairwise term**



Graph Cut based Inference

- Extendible to certain classes of binary higher order energies
- Higher order terms have to be transformable to the pairwise submodular energy with binary auxiliary variables $\mathbf{z}_C$

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

Higher order term **Pairwise term**

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

			$-x_1x_2x_3$	$\min_z z(2 - x_1 - x_2 - x_3)$
$x_1 = 0$	$x_2 = 0$	$x_3 = 0$	0	$\min_z 2z = 0$
$x_1 = 0$	$x_2 = 0$	$x_3 = 1$	0	$\min_z z = 0$
$x_1 = 0$	$x_2 = 1$	$x_3 = 1$	0	$\min_z 0 = 0$
$x_1 = 1$	$x_2 = 1$	$x_3 = 1$	-1	$\min_z(-z) = -1$

Kolmogorov ECCV06, Ramalingam et al. DAM12

Graph Cut based Inference

$$\psi(\mathbf{x}_c) = -\prod_{i \in c} x_i = \min_z(|c| - 1 - \sum_{i \in c} x_i) z$$

$$\psi(\mathbf{x}_c) = -\prod_{i \in c}(1 - x_i) = \min_z(|c| - 1 - \sum_{i \in c}(1 - x_i))(1 - z)$$

Enforces label consistency of the clique as a weak constraint

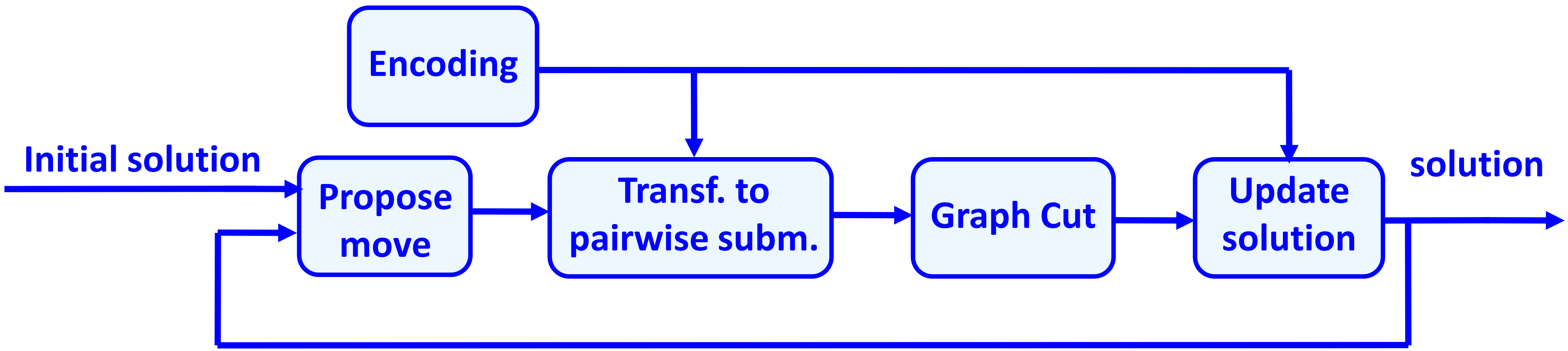
Input image Pairwise CRF Higher order CRF

Kohli et al. CVPR07

Graph Cut based Inference

If the energy is not submodular / graph-representable

- Overestimation by a submodular energy
- Quadratic Pseudo-Boolean Optimisation (QPBO)

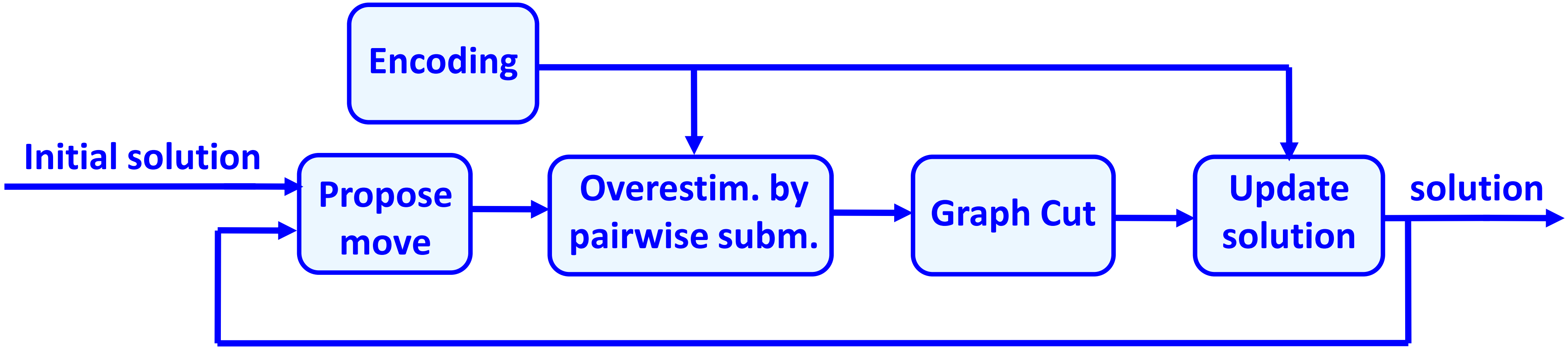


Graph Cut based Inference

Overestimation by a submodular energy (convex / concave)

- We find an energy E'(**t**) s.t.
  - it is tight in the current solution $\quad E'(\mathbf{t}_0) = E(\mathbf{t}_0)$
  - Overestimates E(**t**) for all **t** $\quad E'(\mathbf{t}) \geq E(\mathbf{t})$
- We replace E(**t**) by E'(**t**)
- The moves are not optimal, but quaranteed to converge
- The tighter over-estimation, the better

Example : $$\psi(x_1, x_2) = x_1 x_2 \leq x_1 + x_2$$

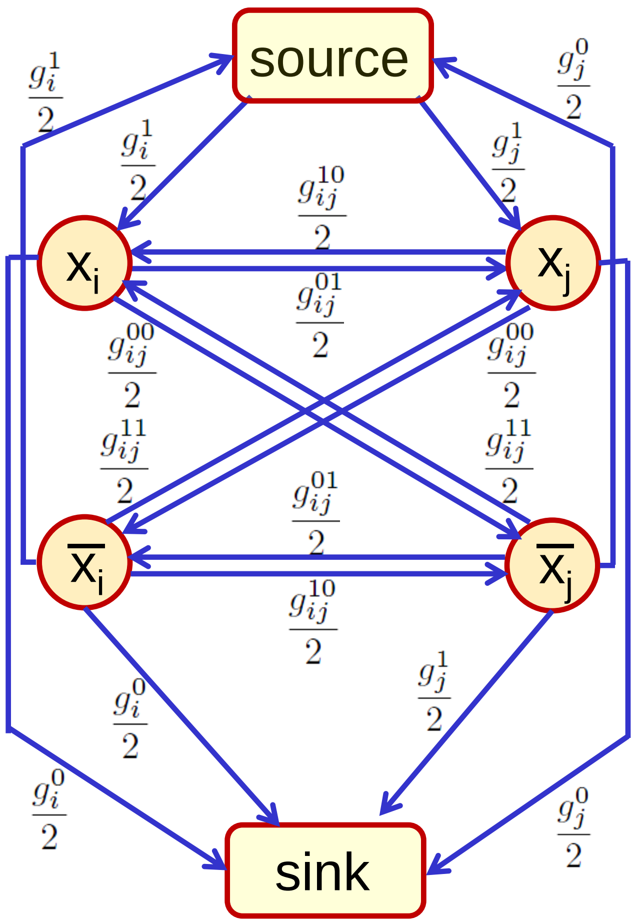


Quadratic Pseudo-Boolean Optimisation

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

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

Quadratic Pseudo-Boolean Optimisation



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

- Energy submodular in x_i and $\overline{x}_i$
- Solved by dropping the constraint $\overline{x}_i = 1 - x_i$
- If $\overline{x}_i = 1 - x_i$ the solution for $\mathrm{x_i}$ is optimal

Overview

- Motivation
- Min-Cut / Max-Flow (Graph Cut) Algorithm
- Markov and Conditional Random Fields
- Random Field Optimisation using Graph Cuts
    - Submodular vs. Non-Submodular Problems
    - Pairwise vs. Higher Order Problems
    - 2-Label vs. Multi-Label Problems
- Recent Advances in Random Field Optimisation
- Conclusions

Motivation

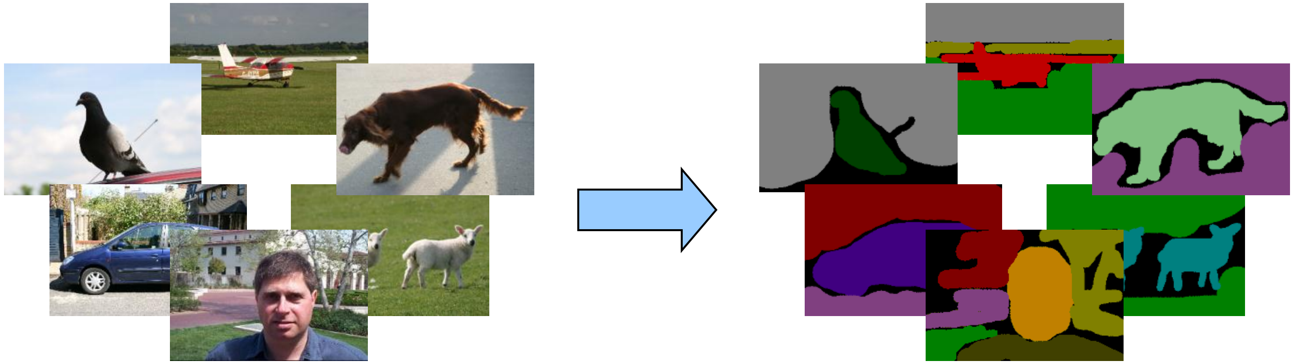
- To propose formulations enforcing certain structure properties of the output useful in computer vision
  - Label Consistency in segments as a weak constraint
  - Label agreement over different scales
  - Label-set consistency
  - Multiple domains consistency
- To propose feasible Graph-Cut based inference method

Structures in CRF

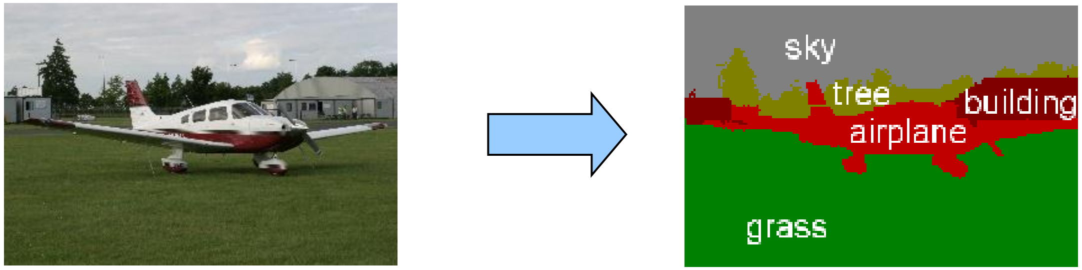
- **Taskar et al. 02 –** associative potentials
- **Woodford et al. 08 –** planarity constraint
- **Vicente et al. 08 –** connectivity constraint
- **Nowozin & Lampert 09 –** connectivity constraint
- **Roth & Black 09** – field of experts
- **Woodford et al. 09 –** marginal probability
- **Delong et al. 10 –** label occurrence costs

Object-class Segmentation Problem

- Aims to assign a class label for each pixel of an image
- Classifier trained on the training set



- Evaluated on never seen test images



UNIVERSITY OF OXFORD

Pairwise CRF models

Standard CRF Energy for Object Segmentation

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

Local context

Cannot encode global consistency of labels!!

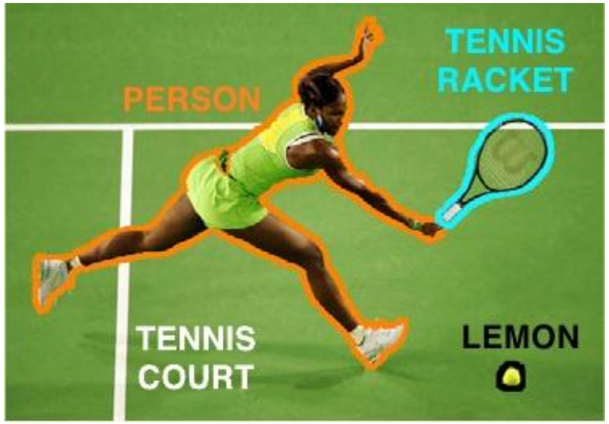
Ladický, Russell, Kohli, Torr ECCV10

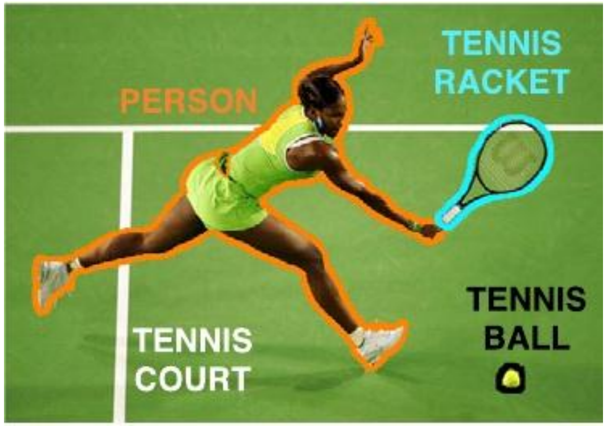
Encoding Co-occurrence

Co-occurrence is a powerful cue

[Heitz et al. '08]
[Rabinovich et al. ‘07]

- Thing – Thing
- Stuff - Stuff
- Stuff - Thing





[Images from Rabinovich et al. 07]

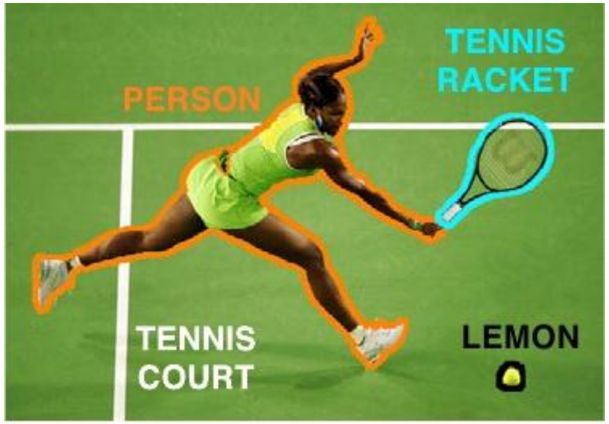
Ladický, Russell, Kohli, Torr ECCV10

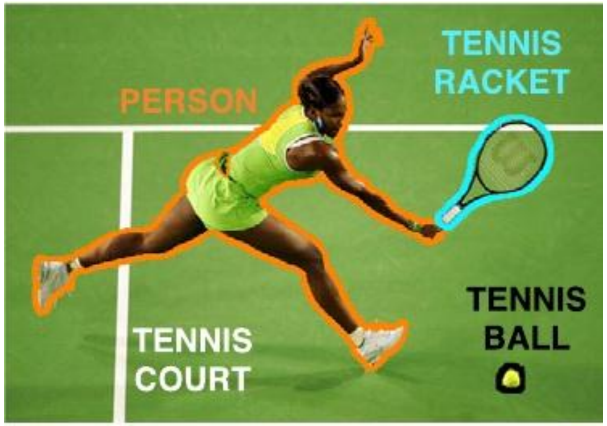
Encoding Co-occurrence

Co-occurrence is a powerful cue

[Heitz et al. '08]
[Rabinovich et al. '07]

- Thing – Thing
- Stuff - Stuff
- Stuff - Thing





Proposed solutions :

1. **Csurka et al. 08 - Hard decision for label estimation**
2. **Torralba et al. 03 - GIST based unary potential**
3. **Rabinovich et al. 07 - Full-connected CRF**

[Images from Rabinovich et al. 07]

So…

What properties should these global co-occurence potentials have ?

Ladický, Russell, Kohli, Torr ECCV10

Desired properties

1. No hard decisions

Ladický, Russell, Kohli, Torr ECCV10

Desired properties

1. No hard decisions

Incorporation in probabilistic framework

Unlikely possibilities are not completely ruled out

Ladický, Russell, Kohli, Torr ECCV10

Desired properties

1. No hard decisions
2. Invariance to region size

Ladický, Russell, Kohli, Torr ECCV10

Desired properties

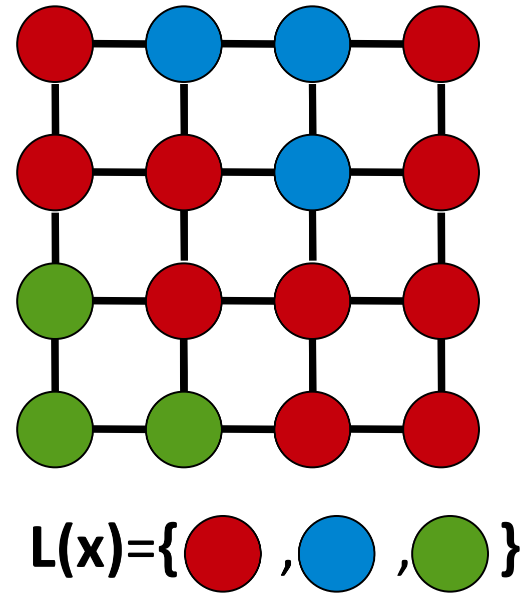
1. No hard decisions
2. Invariance to region size

Cost for occurrence of {people, house, road etc .. } invariant to image area

Ladický, Russell, Kohli, Torr ECCV10

Desired properties

1. No hard decisions
2. Invariance to region size



The only possible solution :

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

Local context

Global context

Cost defined over the assigned labels **L(x)**

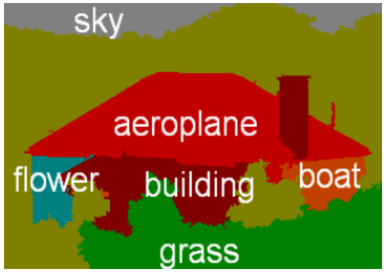
Ladický, Russell, Kohli, Torr ECCV10

Desired properties

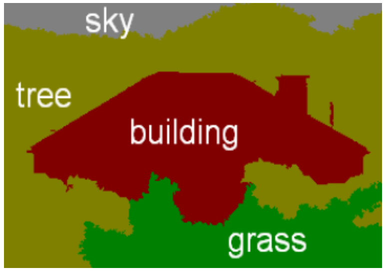
1. **No hard decisions**
2. **Invariance to region size**
3. **Parsimony – simple solutions preferred**

$$\boxed{L_1 \subset L_2 \implies C(L_1) \leq C(L_2).}$$

L(x)={ aeroplane, tree, flower, building, boat, grass, sky **}**



L(x)={ building, tree, grass, sky **}**



Ladický, Russell, Kohli, Torr ECCV10

Desired properties

1. **No hard decisions**
2. **Invariance to region size**
3. **Parsimony – simple solutions preferred**
4. **Efficiency**

Ladický, Russell, Kohli, Torr ECCV10

Desired properties

1. No hard decisions
2. Invariance to region size
3. Parsimony – simple solutions preferred
4. Efficiency

a) Memory requirements as O(n) with the image size and number or labels

b) Inference tractable

Previous work

- **Torralba et al.(2003)** – Gist-based unary potentials
- **Rabinovich et al.(2007)** - complete pairwise graphs
- **Csurka et al.(2008)** - hard estimation of labels present

Method	Global energy	Invariance	Parsimony	Efficiency
Torralba *et al.*	✓	✗	✗	✓
Rabinovich *et al.*	✓	✗	✓	✗
Csurka *et al.*	✗	—	—	✓
Our approach	✓	✓	✓	✓

Ladický, Russell, Kohli, Torr ECCV10

UNIVERSITY OF OXFORD

Related work

- **Zhu & Yuille 1996 –** MDL prior
- **Bleyer et al. 2010 –** Surface Stereo MDL prior
- **Hoiem et al. 2007** – 3D Layout CRF MDL Prior

$$C(x) = K\ |L(x)|$$

- **Delong et al. 2010 –** label occurence cost

$$C(x) = \Sigma_L K_L \delta_L(x)$$

Ladický, Russell, Kohli, Torr ECCV10

Related work

- **Zhu & Yuille 1996 –** MDL prior
- **Bleyer et al. 2010 –** Surface Stereo MDL prior
- **Hoiem et al. 2007** – 3D Layout CRF MDL Prior

$$C(x) = K\ |L(x)|$$

- **Delong et al. 2010 –** label occurence cost

$$C(x) = \Sigma_L K_L \delta_L(x)$$

All special cases of our model

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Co-occurence representation

$$C(L) = \sum_{B \subseteq L} k_B \qquad k_B = C(B) - \sum_{B' \subset B} k_{B'}$$

Label indicator functions

$$\delta_\alpha(\mathbf{t}) = \begin{cases} 1 & \text{if} \quad \exists i \in \mathcal{V} \text{ s.t. } t_i = 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$\forall l \in A \,,\, \delta_l(\mathbf{t}) = \begin{cases} 1 & \text{if} \quad \exists i \in \mathcal{V}_l \text{ s.t. } t_i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

Cost of current label set

$$E(\mathbf{t}) = E_{new}(\mathbf{t}) - E_{old} = \sum_{B \subseteq A \cup \{\alpha\}} k_B \prod_{l \in B} \delta_l(\mathbf{t}) - \boxed{C(A)}$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = E_{new}(\mathbf{t}) - E_{old} = \sum_{B \subseteq A \cup \{\alpha\}} k_B \prod_{l \in B} \delta_l(\mathbf{t}) - C(A)$$

Decomposition to α-dependent and α-independent part

$$E(\mathbf{t}) = \underbrace{\boxed{\sum_{B \subseteq A} k_B \prod_{l \in B} \delta_l(\mathbf{t})}}_{\alpha\text{-independent}} + \underbrace{\boxed{\sum_{B \subseteq A} k_{B \cup \{\alpha\}} \delta_\alpha(\mathbf{t}) \prod_{l \in B} \delta_l(\mathbf{t})}}_{\alpha\text{-dependent}}$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = E_{new}(\mathbf{t}) - E_{old} = \sum_{B \subseteq A \cup \{\alpha\}} k_B \prod_{l \in B} \delta_l(\mathbf{t}) - C(A)$$

Decomposition to α-dependent and α-independent part

$$E(\mathbf{t}) = \sum_{B \subseteq A} k_B \prod_{l \in B} \delta_l(\mathbf{t}) + \sum_{B \subseteq A} k_{B \cup \{\alpha\}} \delta_\alpha(\mathbf{t}) \prod_{l \in B} \delta_l(\mathbf{t})$$

Either α or all labels in the image after the move

$$\delta_\alpha(\mathbf{t}) \prod_{l \in B} \delta_l(\mathbf{t}) = \delta_\alpha(\mathbf{t}) + \prod_{l \in B} \delta_l(\mathbf{t}) - 1$$

$$E(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{B \subseteq A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = \underbrace{\boxed{k'_\alpha \delta_\alpha(\mathbf{t})}}_{\text{submodular}} + \underbrace{\boxed{\sum_{B \subset A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})}}_{\text{non-submodular}}$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \boxed{\sum_{B \subset A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})}$$

non-submodular

Non-submodular energy overestimated by E'(t)

- **E'(t) = E(t) for current solution**
- **E'(t) ≥ E(t) for any other labelling**

$$E(\mathbf{t}) \leq k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{l \in A} k''_l \delta_l(\mathbf{t})$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \boxed{\sum_{B \subseteq A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})}$$

non-submodular

Non-submodular energy overestimated by E'(t)

- E'(t) = E(t) for current solution
- E'(t) ≥ E(t) for any other labelling

$$E(\mathbf{t}) \leq k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{l \in A} k''_l \delta_l(\mathbf{t})$$

Occurrence - tight

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \boxed{\sum_{B \subset A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})}$$

non-submodular

Non-submodular energy overestimated by E'(t)

- E'(t) = E(t) for current solution
- E'(t) ≥ E(t) for any other labelling

$$E(\mathbf{t}) \leq k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{l \in A} k''_l \delta_l(\mathbf{t})$$

Co-occurrence overestimation

$$k_{ab}\delta_a\delta_b \leq \frac{1}{2} k_{ab}(\delta_a + \delta_b)$$

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \boxed{\sum_{B \subset A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})}$$

non-submodular

Non-submodular energy overestimated by E'(t)

- E'(t) = E(t) for current solution
- E'(t) ≥ E(t) for any other labelling

$$E(\mathbf{t}) \leq k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{l \in A} k''_l \delta_l(\mathbf{t})$$

General case

[See the paper]

Ladický, Russell, Kohli, Torr ECCV10

Inference for Co-occurence

Move Energy

$$E(\mathbf{t}) = k'_\alpha \delta_\alpha(\mathbf{t}) + \boxed{\sum_{B \subset A} k'_B \prod_{l \in B} \delta_l(\mathbf{t})}$$

non-submodular

Non-submodular energy overestimated by E'(t)

- **E'(t) = E(t) for current solution**
- **E'(t) ≥ E(t) for any other labelling**

$$E(\mathbf{t}) \leq k'_\alpha \delta_\alpha(\mathbf{t}) + \sum_{l \in A} k''_l \delta_l(\mathbf{t})$$

Quadratic representation

$$E'(\mathbf{t}) = \min_{\mathbf{z}} \left[k'_\alpha (1 - z_\alpha) + \sum_{l \in A} k''_l z_l + \sum_{i \in \mathcal{V}} k'_\alpha (1 - t_i) z_\alpha + \sum_{l \in A} \sum_{i \in \mathcal{V}_l} k''_l t_i (1 - z_l) \right]$$

Ladický, Russell, Kohli, Torr ECCV10

UNIVERSITY OF OXFORD

Potential Training for Co-occurence

Generatively trained Co-occurence potential

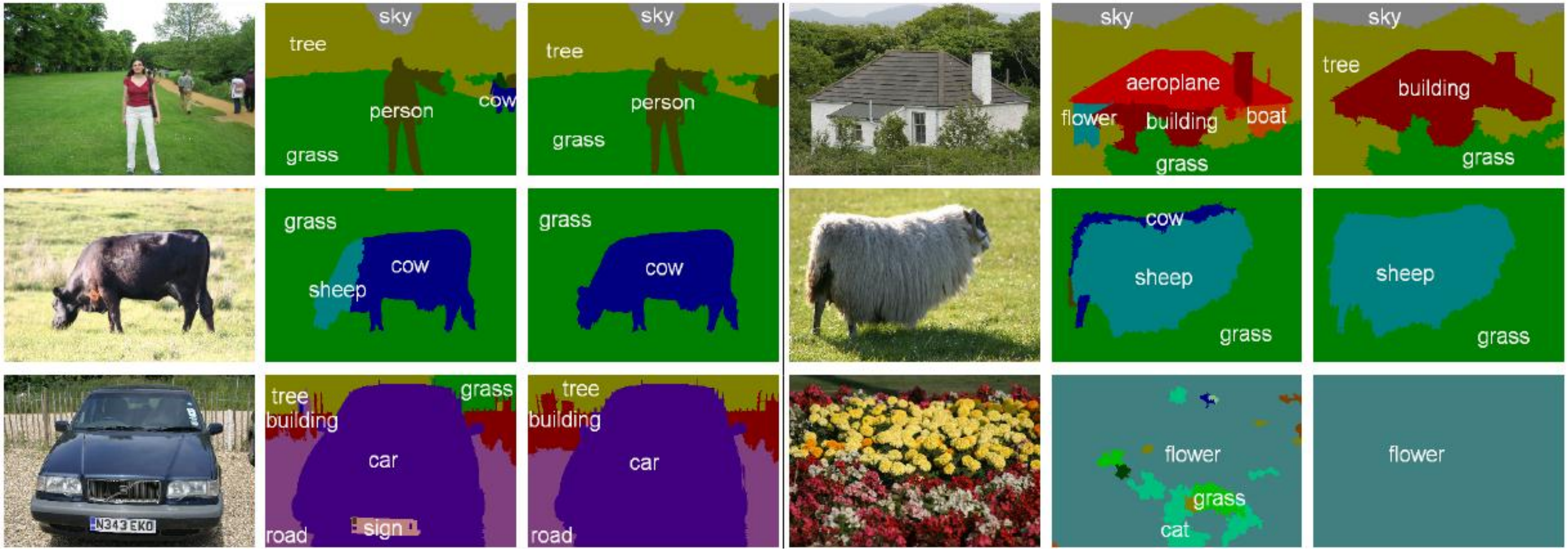
$$C(L) = -w \log \frac{1}{M+1}\left(1 + \sum_{m=1}^{M} \prod_{l \in L} z_l^{(m)}\right)$$

Approximated by 2nd order representation

Ladický, Russell, Kohli, Torr ICCV09

Results on MSRC data set

Comparisons of results with and without co-occurence



Input Image | Without Co-occurence | With Co-occurence | Input Image | Without Co-occurence | With Co-occurence

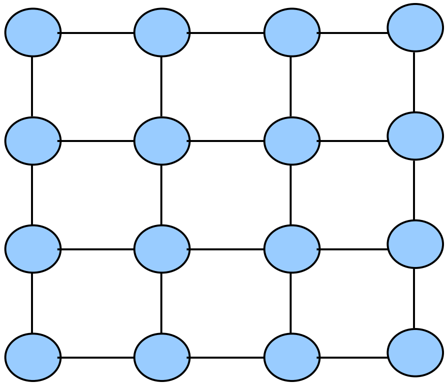
Ladický, Russell, Kohli, Torr ICCV09

Pairwise CRF models

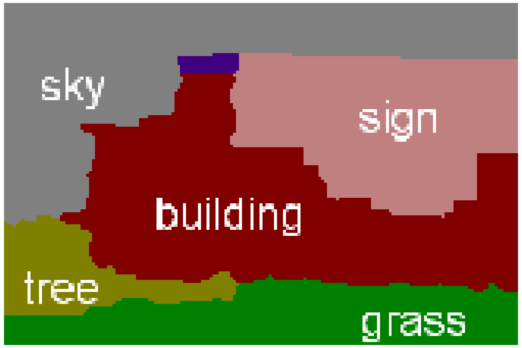
Pixel CRF Energy for Object-class Segmentation

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

Input Image



Pairwise CRF



Result

Shotton et al. ECCV06

Pairwise CRF models

Pixel CRF Energy for Object-class Segmentation

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

- Lacks long range interactions
- Results oversmoothed
- Data term information may be insufficient

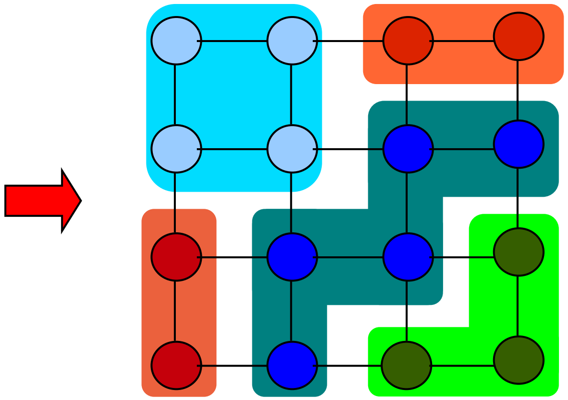
Pairwise CRF models

Segment CRF Energy for Object-class 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**

Input Image



Unsupervised Segmentation

Shi, Malik PAMI2000,
Comaniciu, Meer PAMI2002,
Felzenschwalb, Huttenlocher, IJCV2004,

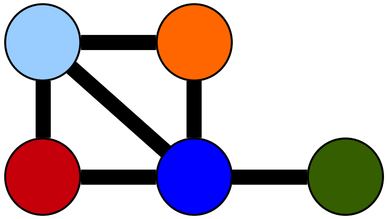
Pairwise CRF models

Segment CRF Energy for Object-class Segmentation

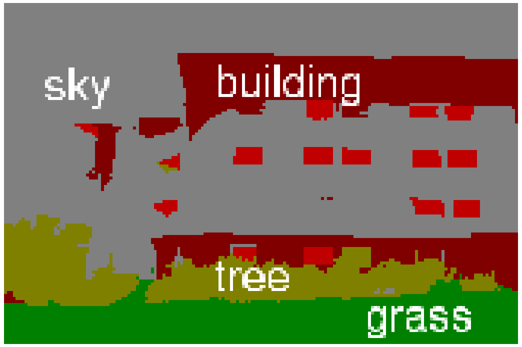
$$E(\mathbf{x}) = \boxed{\sum_{i \in \mathcal{V}} \psi_i(x_i)} + \boxed{\sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)}$$

Data term **Smoothness term**

Input Image



Pairwise CRF



Result

Batra et al. CVPR08, Yang et al. CVPR07, Zitnick et al. CVPR08, Rabinovich et al. ICCV07, Boix et al. CVPR10

Pairwise CRF models

Segment CRF Energy for Object-class 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**

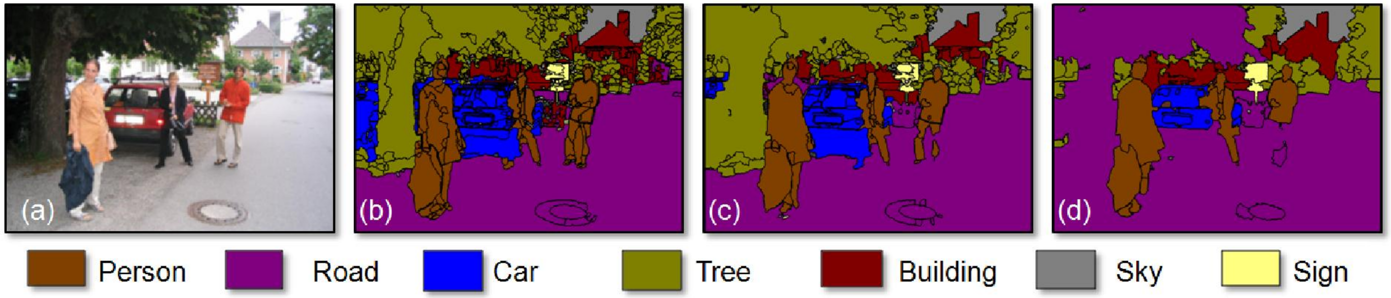
- Allows long range interactions
- Does not distinguish between instances of objects
- Cannot recover from incorrect segmentation

Pairwise CRF models

Segment CRF Energy for Object-class Segmentation

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

- Allows long range interactions
- Does not distinguish between instances of objects
- Cannot recover from incorrect segmentation



Robust P^N approach

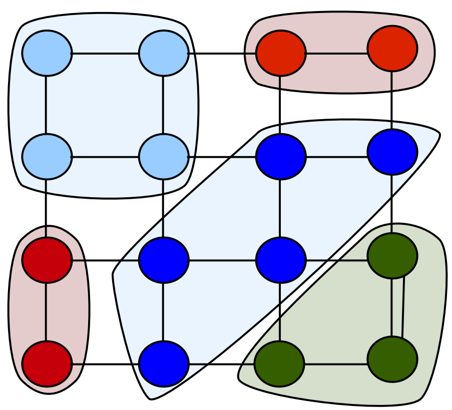
Segment CRF Energy for Object-class Segmentation

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

Pairwise CRF

Segment consistency term

Input Image



Higher order CRF

Kohli, Ladický, Torr CVPR08

Robust P^N approach

Segment CRF Energy for Object-class Segmentation

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

Pairwise CRF

Segment consistency term

$$\psi_c^h(\mathbf{x}_c) = \min_{l \in \mathcal{L}}(\gamma_c^{\max}, \sum_{i \in c} k_c \delta(x_i \neq l))$$

Kohli, Ladický, Torr CVPR08

Robust P^N approach

Segment CRF Energy for Object-class Segmentation

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

Pairwise CRF

Segment consistency term

$$\psi_c^h(\mathbf{x}_c) = \min_{l \in \mathcal{L}} \left(\boxed{\gamma_c^{\max}}, \sum_{i \in c} k_c \delta(x_i \neq l) \right)$$

No dominant label

Kohli, Ladický, Torr CVPR08

Robust P^N approach

Segment CRF Energy for Object-class Segmentation

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

Pairwise CRF

Segment consistency term

$$\psi_c^h(\mathbf{x}_c) = \min_{l \in \mathcal{L}}(\gamma_c^{\max}, \boxed{\sum_{i \in c} k_c \delta(x_i \neq l)})$$

Dominant label l

Kohli, Ladický, Torr CVPR08

Robust P^N approach

Segment CRF Energy for Object-class Segmentation

$$E(\mathbf{x}) = \underbrace{\boxed{\sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)}}_{\text{Pairwise CRF}} + \underbrace{\boxed{\sum_{c \in \mathcal{S}} \psi_c^h(\mathbf{x}_c)}}_{\text{Segment consistency term}}$$

Pairwise CRF

Segment consistency term

- Enforces consistency as a weak constraint
- Can recover from incorrect segmentation
- Can combine multiple segmentations
- Does not use features at segment level

Kohli, Ladický, Torr CVPR08

Robust P^N approach

Segment CRF Energy for Object-class Segmentation

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

Pairwise CRF

Segment consistency term

$$\psi_c^h(\mathbf{x}_c) = \min_{l \in \mathcal{L}} \left(\gamma_c^{\max}, \sum_{i \in c} k_c \delta(x_i \neq l) \right)$$

Transformable to pairwise graph with auxiliary variable

Kohli, Ladický, Torr CVPR08

Robust P^N approach

Segment CRF Energy for Object-class Segmentation

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

Pairwise CRF

Segment consistency

$$\psi_c^h(\mathbf{x}_c) = \min_{l \in \mathcal{L}} (\gamma_c^{\max}, \sum_{i \in c} k_c \delta(x_i \neq l)) = \min_{y_c} \phi_c(y_c) + \sum_{i \in c} \phi_c(y_c, x_i)$$

where

$$\phi_c(y_c) = \begin{cases} \gamma_c^{\max} & \text{if} \quad y_c = l_F \\ 0 & \text{if} \quad y_c \in \mathcal{L} \end{cases}$$

$$\phi_c(y_c, x_i) = \begin{cases} 0 & \text{if} \quad y_c = l_F \text{ or } y_c = x_i \\ k_c & \text{if} \quad y_c \in \mathcal{L} \text{ and } x_i \neq y_c \end{cases}$$

Ladický, Russell, Kohli, Torr ICCV09

Robust P^N approach

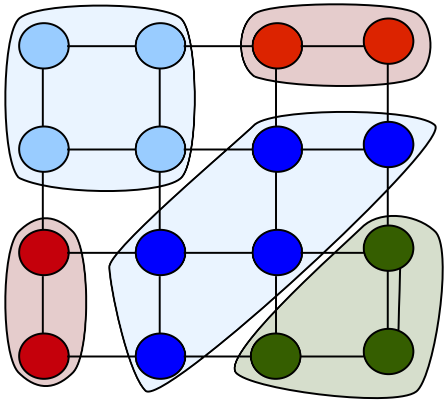
Segment CRF Energy for Object-class Segmentation

$$E(\mathbf{x}) = \underbrace{\sum_{i\in\mathcal{V}} \psi_i(x_i) + \sum_{i\in\mathcal{V}, j\in\mathcal{N}_i} \psi_{ij}(x_i, x_j)}_{\text{Pairwise CRF}} + \underbrace{\sum_{c\in\mathcal{S}} \psi_c^h(\mathbf{x}_c)}_{\text{Segment consistency term}}$$

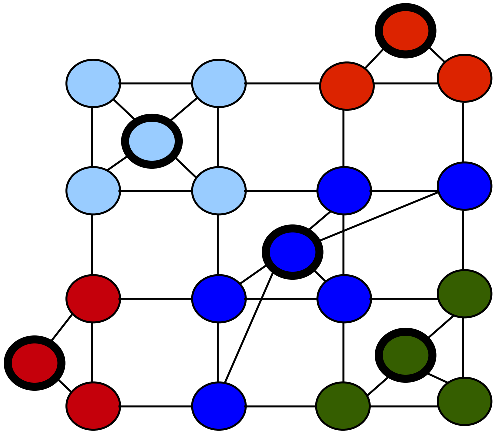
Pairwise CRF

Segment consistency term

Input Image



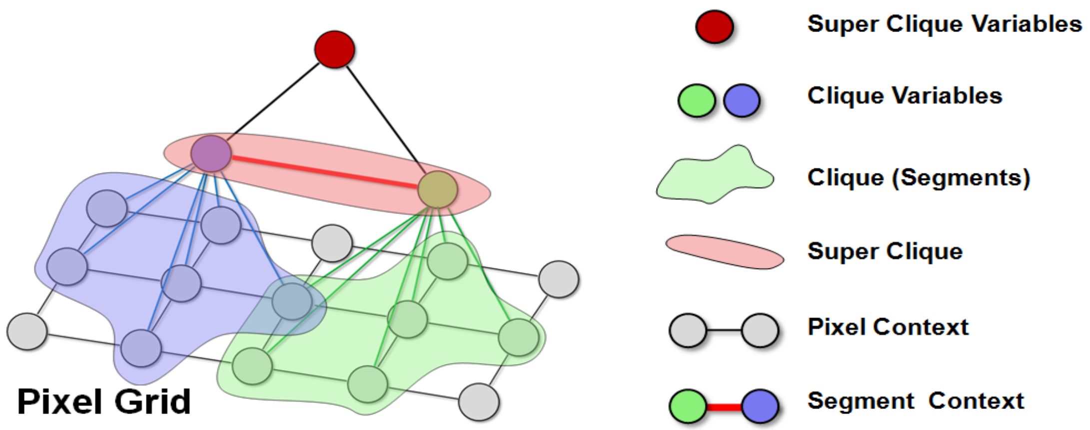
Higher order CRF



Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

Can be generalized to Hierarchical model



- Allows unary potentials for region variables
- Allows pairwise potentials for region variables
- Allows multiple layers and multiple hierarchies

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

AH CRF Energy for Object-class Segmentation

$$\boxed{E^{(0)}(\mathbf{x}) = \sum_{i \in \mathcal{S}^{(0)}} \psi_i^{(0)}(x_i^{(0)}) + \sum_{ij \in \mathcal{N}^{(0)}} \psi_{ij}^{(0)}(x_i^{(0)}, x_j^{(0)})} + \boxed{\min_{\mathbf{x}^{(1)}} E^{(1)}(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})}$$

Pairwise CRF **Higher order term**

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

AH CRF Energy for Object-class Segmentation

$$\boxed{E^{(0)}(\mathbf{x}) = \sum_{i \in \mathcal{S}^{(0)}} \psi_i^{(0)}(x_i^{(0)}) + \sum_{ij \in \mathcal{N}^{(0)}} \psi_{ij}^{(0)}(x_i^{(0)}, x_j^{(0)})} + \boxed{\min_{\mathbf{x}^{(1)}} E^{(1)}(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})}$$

Pairwise CRF **Higher order term**

Higher order term recursively defined as

$$E^{(n)}(\mathbf{x}^{(n-1)}, \mathbf{x}^{(n)}) = \boxed{\sum_{c \in \mathcal{S}^{(n)}} \psi_c^{(n)}(x_c^{(n-1)}, x_c^{(n)})} + \boxed{\sum_{cd \in \mathcal{N}^{(n)}} \psi_{cd}^{(n)}(x_c^{(n)}, x_d^{(n)})} + \boxed{\min_{\mathbf{x}^{(n+1)}} E^{(n+1)}(\mathbf{x}^{(n)}, \mathbf{x}^{(n+1)})}$$

Segment unary term **Segment pairwise term** **Segment higher order term**

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

AH CRF Energy for Object-class Segmentation

$$\boxed{E^{(0)}(\mathbf{x}) = \sum_{i \in \mathcal{S}^{(0)}} \psi_i^{(0)}(x_i^{(0)}) + \sum_{ij \in \mathcal{N}^{(0)}} \psi_{ij}^{(0)}(x_i^{(0)}, x_j^{(0)})} + \boxed{\min_{\mathbf{x}^{(1)}} E^{(1)}(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})}$$

Pairwise CRF

Higher order term

Higher order term recursively defined as

$$E^{(n)}(\mathbf{x}^{(n-1)}, \mathbf{x}^{(n)}) = \boxed{\sum_{c \in \mathcal{S}^{(n)}} \psi_c^{(n)}(x_c^{(n-1)}, x_c^{(n)})} + \boxed{\sum_{cd \in \mathcal{N}^{(n)}} \psi_{cd}^{(n)}(x_c^{(n)}, x_d^{(n)})} + \boxed{\min_{\mathbf{x}^{(n+1)}} E^{(n+1)}(\mathbf{x}^{(n)}, \mathbf{x}^{(n+1)})}$$

Segment unary term

Segment pairwise term

Segment higher order term

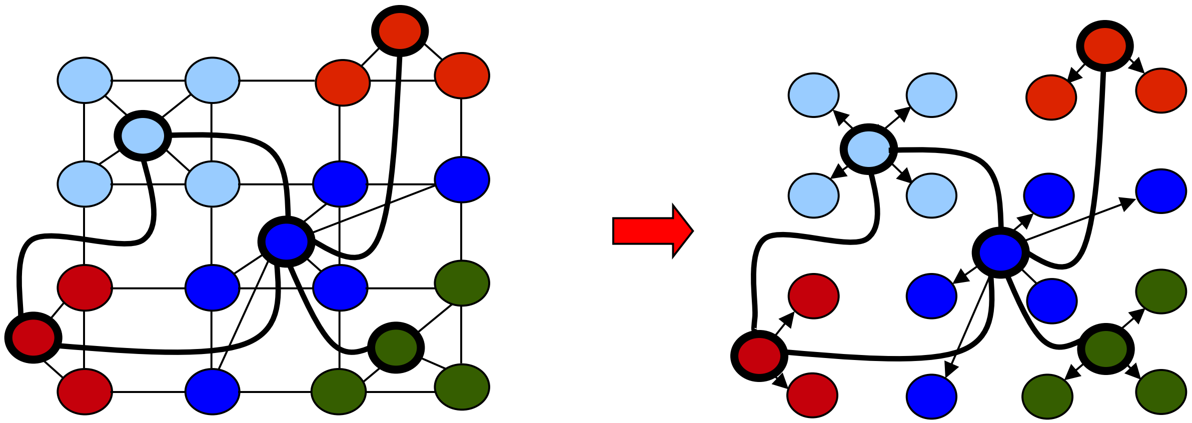
Why is this generalisation useful?

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

Let us analyze the case with

- one segmentation
- potentials only over segment level

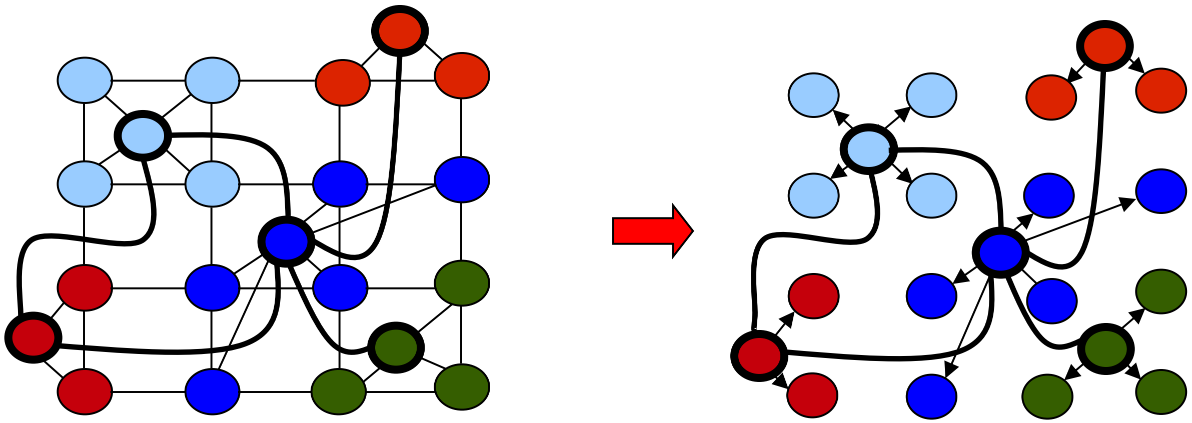


Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

Let us analyze the case with

- one segmentation
- potentials only over segment level



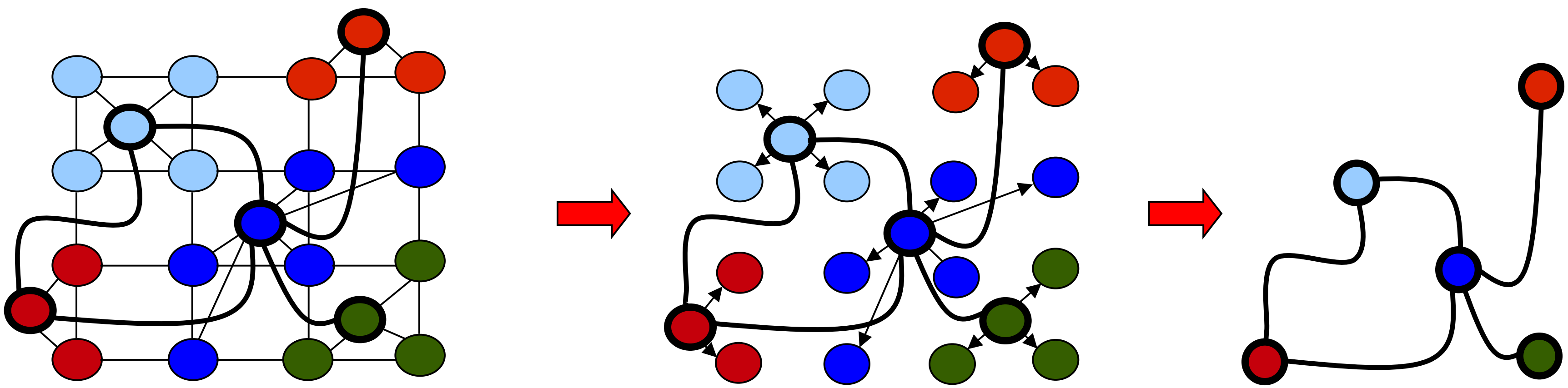
1) Minimum is segment-consistent

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

Let us analyze the case with

- one segmentation
- potentials only over segment level



1) Minimum is segment-consistent

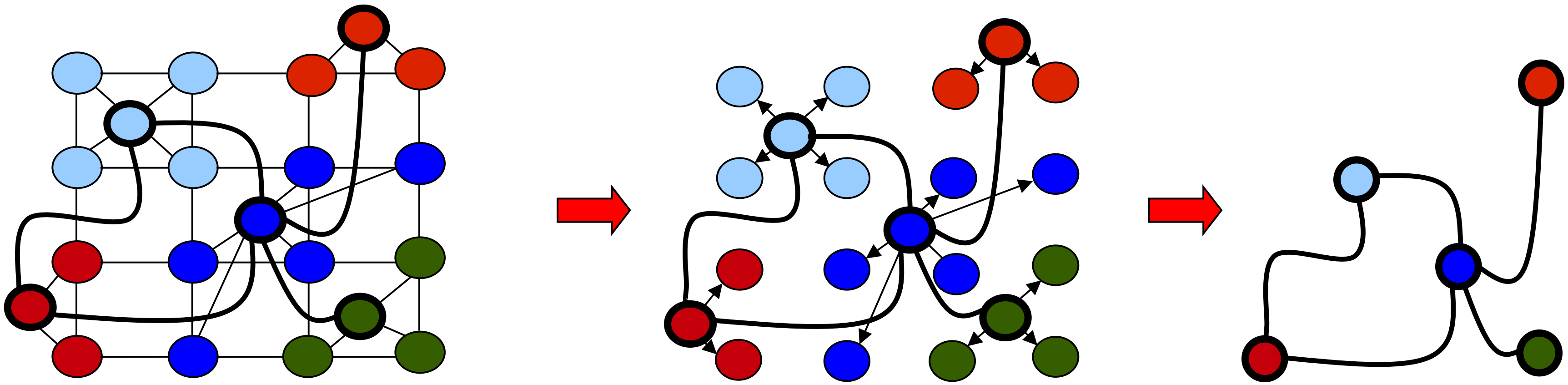
2) Cost of every segment-consistent CRF equal to the cost of pairwise CRF over segments

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs

Let us analyze the case with

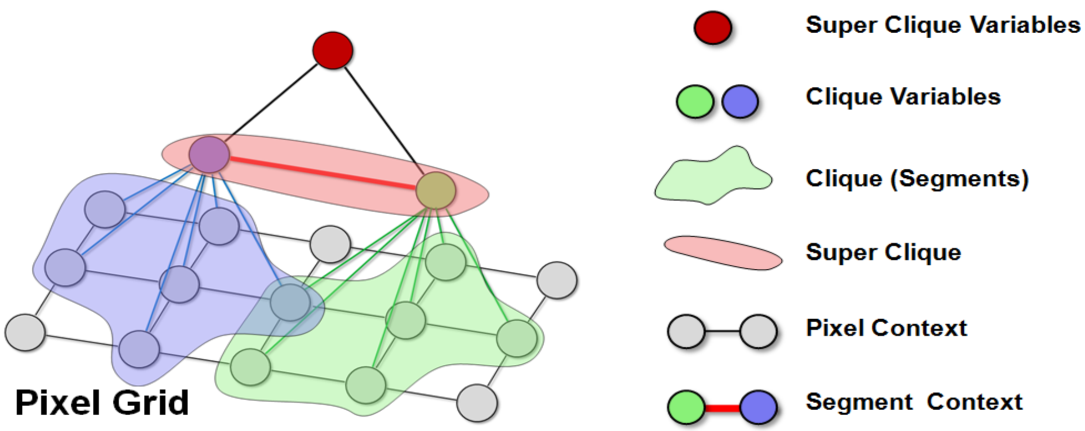
- one segmentation
- potentials only over segment level



Equivalent to pairwise CRF over segments!

Ladický, Russell, Kohli, Torr ICCV09

Associative Hierarchical CRFs



- Merges information over multiple scales
- Easy to train potentials and learn parameters
- Allows multiple segmentations and hierarchies
- Allows long range interactions
- Limited (?) to associative interlayer connections

Ladický, Russell, Kohli, Torr ICCV09

Inference for AH-CRF

Graph Cut based move making algorithms [Boykov et al. 01]

α-expansion transformation function for base layer

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

Ladický (thesis) 2011

Inference for AH-CRF

Graph Cut based move making algorithms [Boykov et al. 01]

α-expansion transformation function for base layer

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

α-expansion transformation function for auxiliary layer

(2 binary variables per node)

$$T_\alpha(\mathbf{x}_c^{(n)}, a_c^{(n)}, b_c^{(n)}) = \begin{cases} \alpha & \text{if} \quad a_c^{(n)} = 0 \text{ and } b_c^{(n)} = 0 \\ x_c^{(n)} & \text{if} \quad a_c^{(n)} = 1 \text{ and } b_c^{(n)} = 1 \\ l_F & \text{if} \quad a_c^{(n)} = 1 \text{ and } b_c^{(n)} = 0 \end{cases}$$

Ladický (thesis) 2011

Inference for AH-CRF

Interlayer connection between the base layer and the first auxiliary layer:

$$\psi_c^p(\mathbf{x}_c, x_c^{(1)}) = \phi_c(x_c^{(1)}) + \sum_{i \in c} \phi_c(x_c^{(1)}, x_i)$$

$$\phi_c(x_c^{(1)}, x_i) = \begin{cases} 0 & \text{if } x_c^{(1)} = L_F \text{ or } x_c^{(1)} = x_i \\ w_i k_c^{x_c^{(1)}} & \text{otherwise.} \end{cases}$$

Ladický (thesis) 2011

Inference for AH-CRF

Interlayer connection between the base layer and the first auxiliary layer:

$$\psi_c^p(\mathbf{x}_c, x_c^{(1)}) = \phi_c(x_c^{(1)}) + \sum_{i \in c} \phi_c(x_c^{(1)}, x_i)$$

$$\phi_c(x_c^{(1)}, x_i) = \begin{cases} 0 & \text{if } x_c^{(1)} = L_F \text{ or } x_c^{(1)} = x_i \\ w_i k_c^{x_c^{(1)}} & \text{otherwise.} \end{cases}$$

The move energy is:

$$\psi_c^p(\mathbf{t}_c, a_c^{(1)}, b_c^{(1)}) = \begin{cases} \phi_c(\alpha) + \sum_{i \in c} w_i k_c^{\alpha} t_i & \text{if } a_c^{(1)} = 0 \text{ and } b_c^{(1)} = 0 \\ \chi_c(x_c^{(1)}) + \sum_{i \in c} w_i k_c^{x_c^{(1)}} (1 - t_i) \delta(x_i = x_c^{(1)}) & \text{if } a_c^{(1)} = 1 \text{ and } b_c^{(1)} = 1 \\ \phi_c(l_F) & \text{if } a_c^{(1)} = 1 \text{ and } b_c^{(1)} = 0 \end{cases}$$

where $$\chi_c(x_c^{(1)}) = \phi(x_c^{(1)}) + \sum_{i \in c} w_i k_c^{x_c^{(1)}} \delta(x_i \neq x_c^{(1)})$$

Ladický (thesis) 2011

Inference for AH-CRF

The move energy for interlayer connection between the base and auxiliary layer is:

$$\psi^p_c(\mathbf{t}_c, a^{(1)}_c, b^{(1)}_c) = \begin{cases} \phi_c(\alpha) + \sum_{i \in c} w_i k^\alpha_c t_i & \text{if } a^{(1)}_c = 0 \text{ and } b^{(1)}_c = 0 \\ \chi_c(x^{(1)}_c) + \sum_{i \in c} w_i k^{x^{(1)}_c}_c (1 - t_i)\delta(x_i = x^{(1)}_c) & \text{if } a^{(1)}_c = 1 \text{ and } b^{(1)}_c = 1 \\ \phi_c(l_F) & \text{if } a^{(1)}_c = 1 \text{ and } b^{(1)}_c = 0 \end{cases}$$

Can be transformed to binary submodular pairwise function:

$$\begin{aligned} \psi^p_c(\mathbf{t}_c, a^{(1)}_c, b^{(1)}_c) &= \phi_c(\alpha) + \chi_c(x^{(1)}_c) - \phi_c(l_F) \\ &+ \sum_{i \in c} w_i k^\alpha_c t_i (1 - a^{(1)}_c) + (\phi_c(l_F) - \phi_c(\alpha)) a^{(1)}_c \\ &+ \sum_{i \in c} w_i k^{x^{(1)}_c}_c \delta(x_i = x^{(1)}_c)(1 - t_i) b^{(1)}_c + (\phi_c(l_F) - \chi_c(x^{(1)}_c))(1 - b^{(1)}_c) \end{aligned}$$

Ladický (thesis) 2011

Inference for AH-CRF

The move energy for interlayer connection between auxiliary layers is:

$$\psi_c^p(\mathbf{a}^{(n-1)}, \mathbf{b}^{(n-1)}, a_c^{(n)}, b_c^{(n)}) = \begin{cases} \phi_c(\alpha) & + \sum_{i \in c} w_i k_c^{\alpha} a_i^{(n-1)} \\ & \text{if } a_c^{(n)} = 0 \text{ and } b_c^{(n)} = 0 \\ \chi_c(x_c^{(n)}) & + \sum_{i \in c} w_i k_c^{x_c^{(n)}} (1 - b_i^{(n-1)}) \delta(x_i^{(n-1)} = x_c^{(n)}) \\ & \text{if } a_c^{(n)} = 1 \text{ and } b_c^{(n)} = 1 \\ \phi_c(l_F) & \\ & \text{if } a_c^{(n)} = 1 \text{ and } b_c^{(n)} = 0, \end{cases}$$

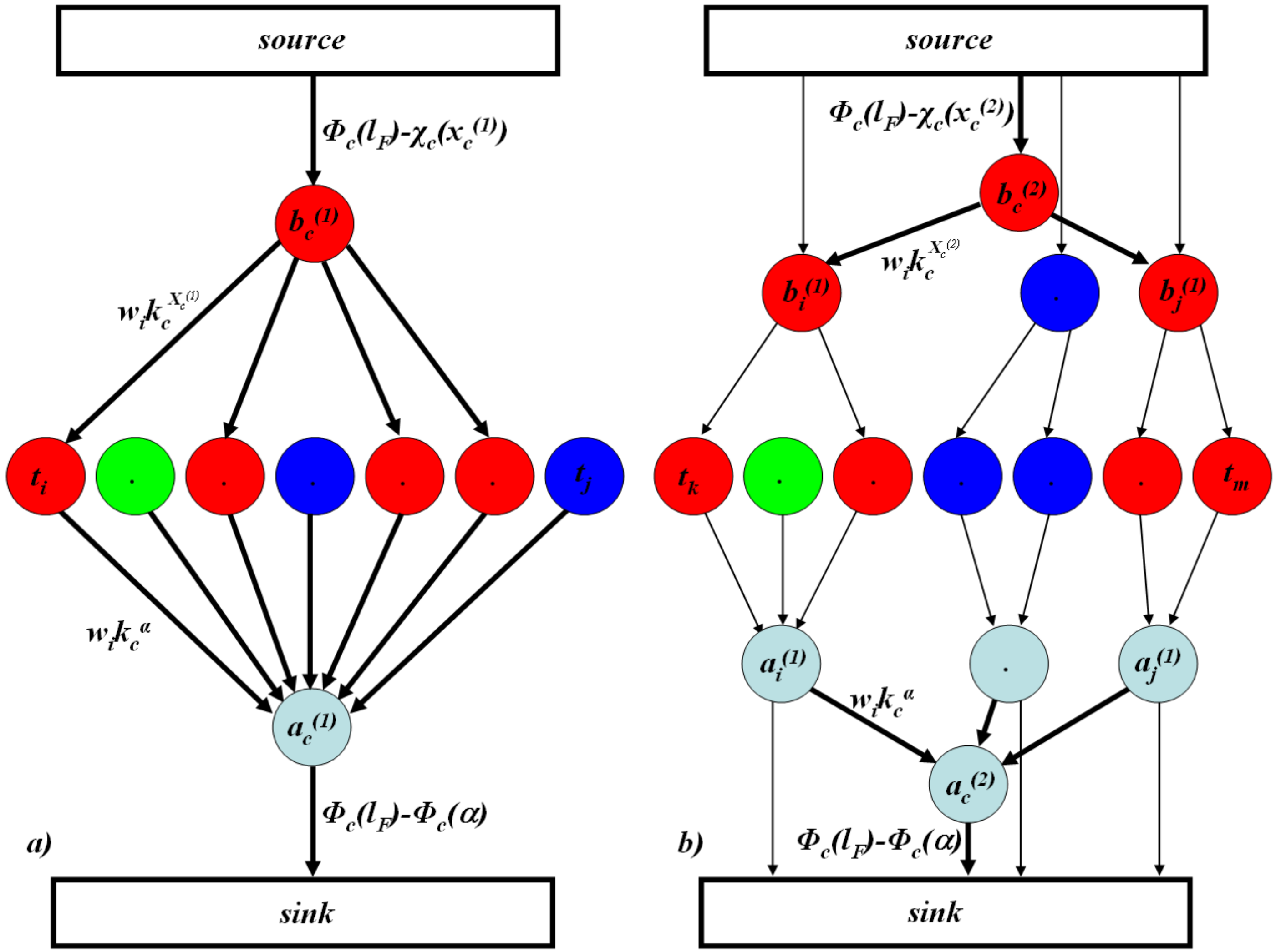
Can be transformed to binary submodular pairwise function:

$$\begin{aligned} \psi_c^p(\mathbf{a}^{(n-1)}, \mathbf{b}^{(n-1)}, a_c^{(n)}, b_c^{(n)}) &= \phi_c(\alpha) + \chi_c(x_c^{(n)}) - \phi_c(l_F) \\ &+ \sum_{i \in c} w_i k_c^{\alpha} a_i^{(n-1)} (1 - a_c^{(n)}) + (\phi_c(l_F) - \phi_c(\alpha)) a_c^{(n)} \\ &+ \sum_{i \in c} w_i k_c^{x_c^{(n)}} \delta(x_i^{(n-1)} = x_c^{(n)}) (1 - b_i^{(n-1)}) b_c^{(n)} \\ &+ (\phi_c(l_F) - \chi_c(x_c^{(n)})) (1 - b_c^{(n)}) \end{aligned}$$

Ladický (thesis) 2011

Inference for AH-CRF

Graph constructions



Ladický (thesis) 2011

UNIVERSITY OF OXFORD

Inference for AH-CRF

Pairwise potential between auxiliary variables :

$$\psi^p_{cd}(x_c^{(n)}, x_d^{(n)}) = \begin{cases} 0 & \text{if } x_c^{(n)} = x_d^{(n)} \\ \frac{K}{2} & \text{if } (x_c^{(n)} = l_F \text{ and } x_d^{(n)} \neq l_F) \text{ or } (x_c^{(n)} \neq l_F \text{ and } x_d^{(n)} = l_F) \\ K & \text{if } x_c^{(n)} \neq x_d^{(n)} \neq l_F. \end{cases}$$

Ladický (thesis) 2011

Inference for AH-CRF

Move energy if $x_c = x_d$:

$$
\psi^p_{cd}(a_c^{(n)}, b_c^{(n)}, a_d^{(n)}, b_d^{(n)}) = \begin{cases} 0 & \text{if } a_c^{(n)} = a_d^{(n)} \text{ and } b_c^{(n)} = b_d^{(n)} \\ \frac{K}{2} & \text{if } (a^{(n)} \neq a_d^{(n)} \text{ and } b_c^{(n)} = b_d^{(n)}) \\ & \text{or } (a^{(n)} = a_d^{(n)} \text{ and } b_c^{(n)} \neq b_d^{(n)}) \\ K & \text{if } a_c^{(n)} \neq a_d^{(n)} \text{ and } b_c^{(n)} \neq b_d^{(n)} \end{cases}
$$

Can be transformed to binary submodular pairwise function:

$$
\begin{aligned}
\psi^p_{cd}(a_c^{(n)}, b_c^{(n)}, a_d^{(n)}, b_d^{(n)}) \quad &= \quad \frac{K}{2} a_c^{(n)}(1 - a_d^{(n)}) + \frac{K}{2}(1 - a_c^{(n)}) a_d^{(n)} \\
&+ \quad \frac{K}{2} b_c^{(n)}(1 - b_d^{(n)}) + \frac{K}{2}(1 - b_c^{(n)}) b_d^{(n)}
\end{aligned}
$$

Ladický (thesis) 2011

Inference for AH-CRF

Move energy if $x_c \neq x_d$:

$$
\psi^p_{cd}(a_c^{(n)}, b_c^{(n)}, a_d^{(n)}, b_d^{(n)}) = \begin{cases} 0 & \text{if } a_c^{(n)} = a_c^{(n)} \text{ and } b_c^{(n)} = b_d^{(n)} = 0 \\ K & \text{if } a_c^{(n)} = a_d^{(n)} \text{ and } b_c^{(n)} = b_d^{(n)} = 1 \\ \frac{K}{2} & \text{if } (a^{(n)} \neq a_d^{(n)} \text{ and } b_c^{(n)} = b_d^{(n)}) \\ & \text{or } (a^{(n)} = a_d^{(n)} \text{ and } b_c^{(n)} \neq b_d^{(n)}) \\ K & \text{if } a_c^{(n)} \neq a_d^{(n)} \text{ and } b_c^{(n)} \neq b_d^{(n)} \end{cases}
$$

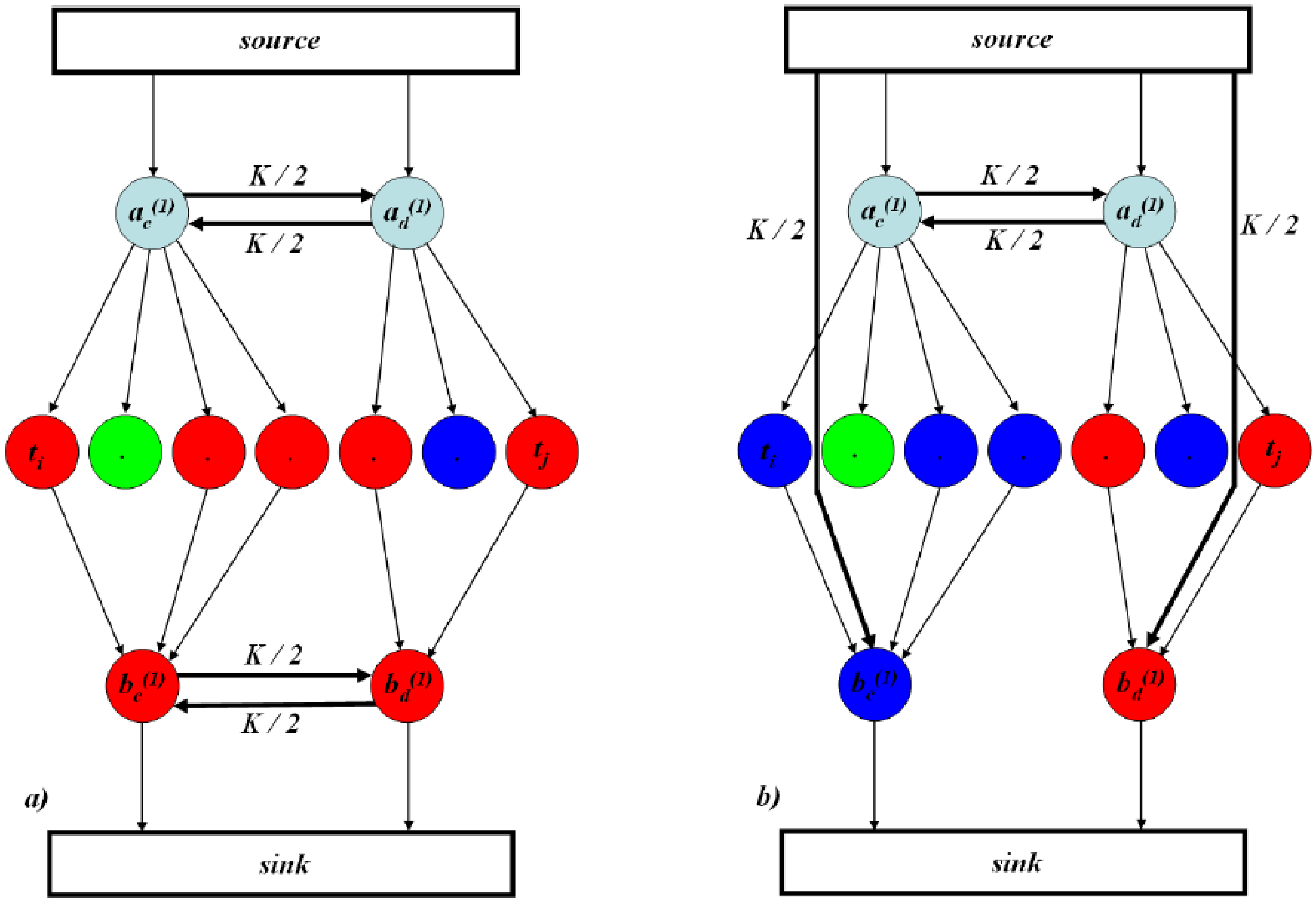
Can be transformed to binary submodular pairwise function:

$$
\psi^p_{cd}(a_c^{(n)}, b_c^{(n)}, a_d^{(n)}, b_d^{(n)}) = \frac{K}{2} a_c^{(n)}(1 - a_d^{(n)}) + \frac{K}{2}(1 - a_c^{(n)}) a_d^{(n)} + \frac{K}{2} b_c^{(n)} + \frac{K}{2} b_d^{(n)}
$$

Ladický (thesis) 2011

Inference for AH-CRF

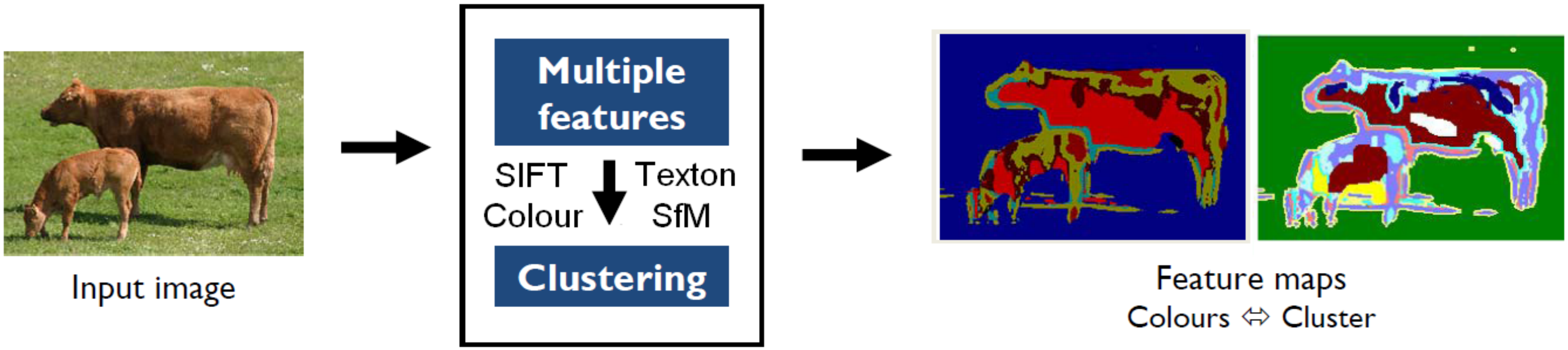
Graph constructions



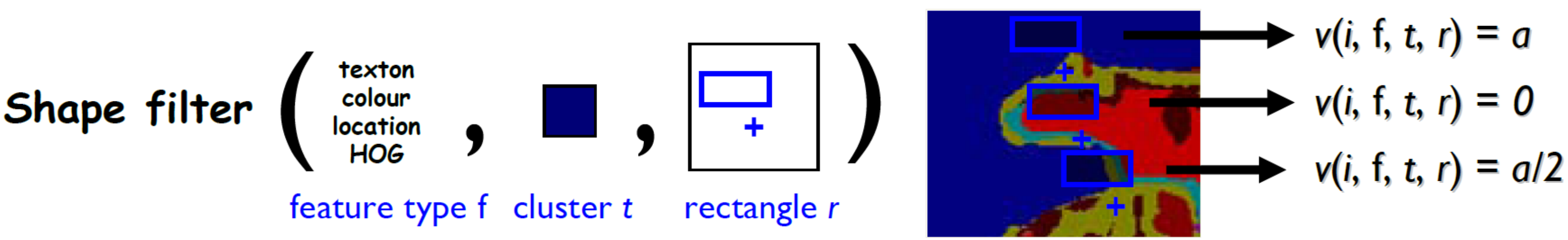
Ladický (thesis) 2011

Potential training for Object class Segmentation

Pixel unary potential



Unary likelihoods based on spatial configuration (*Shotton et al. ECCV06*)

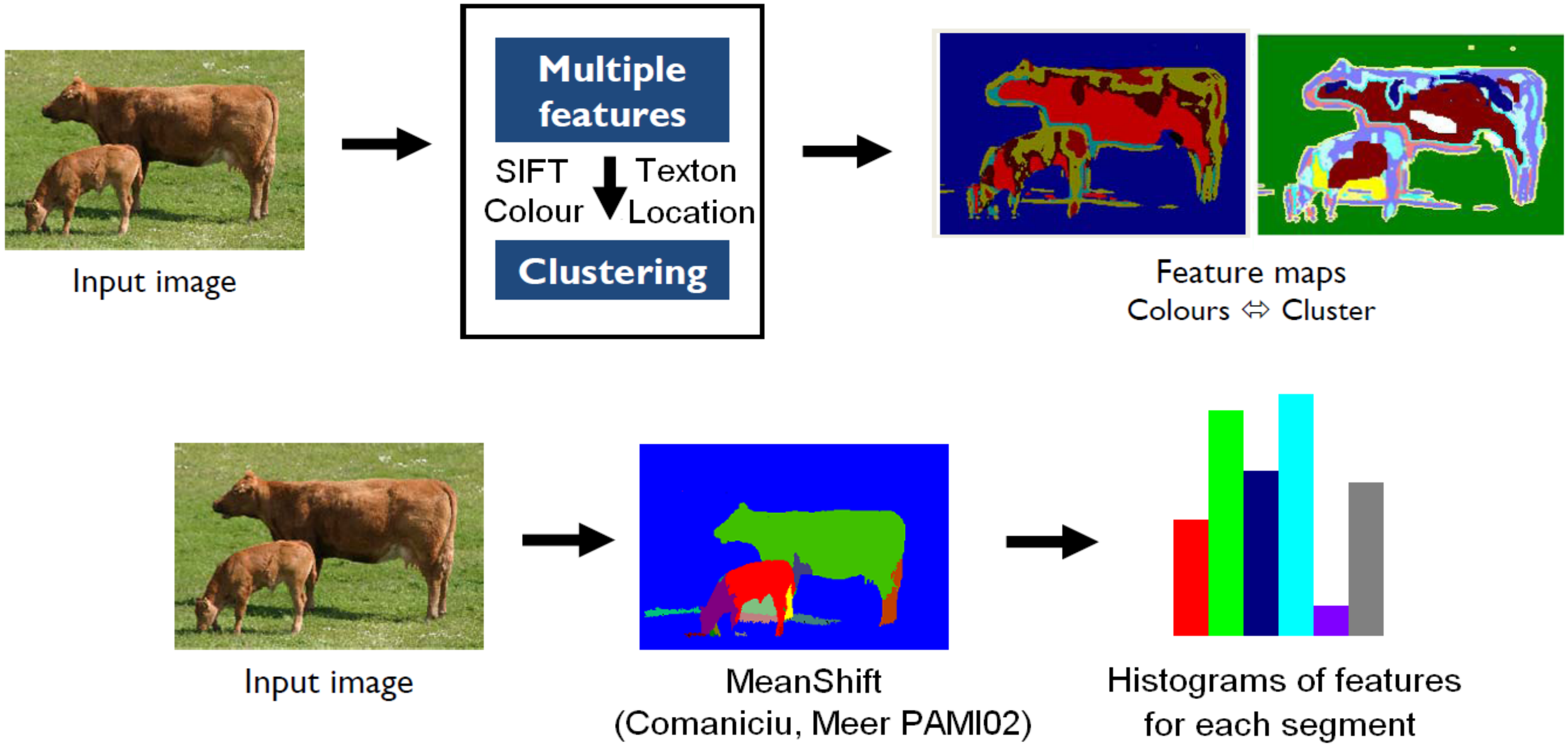


Classifier trained using boosting

Ladický, Russell, Kohli, Torr ICCV09

Potential training for Object class Segmentation

Segment unary potential

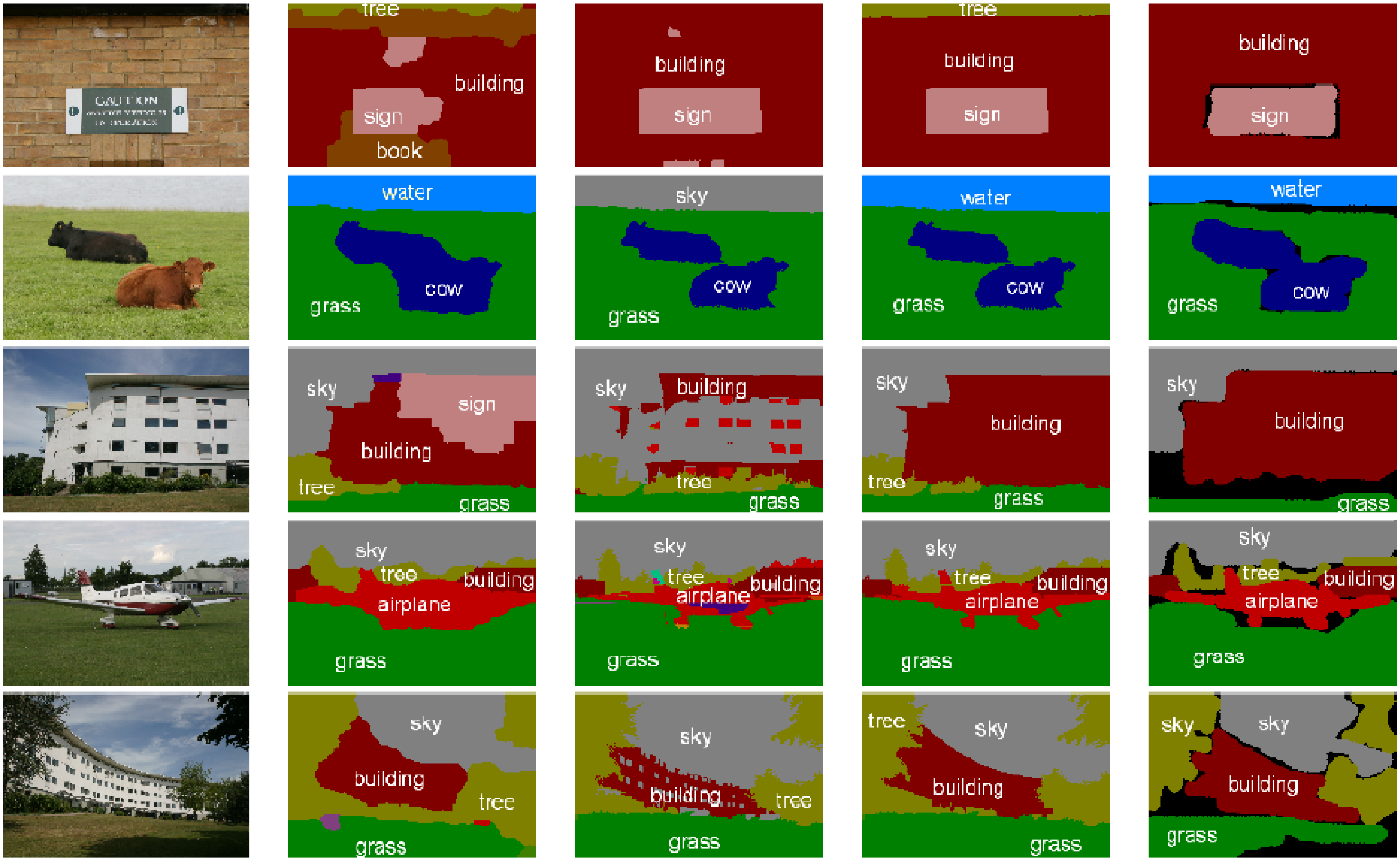


Classifier trained using boosting (resp. Kernel SVMs)

Ladický, Russell, Kohli, Torr ICCV09

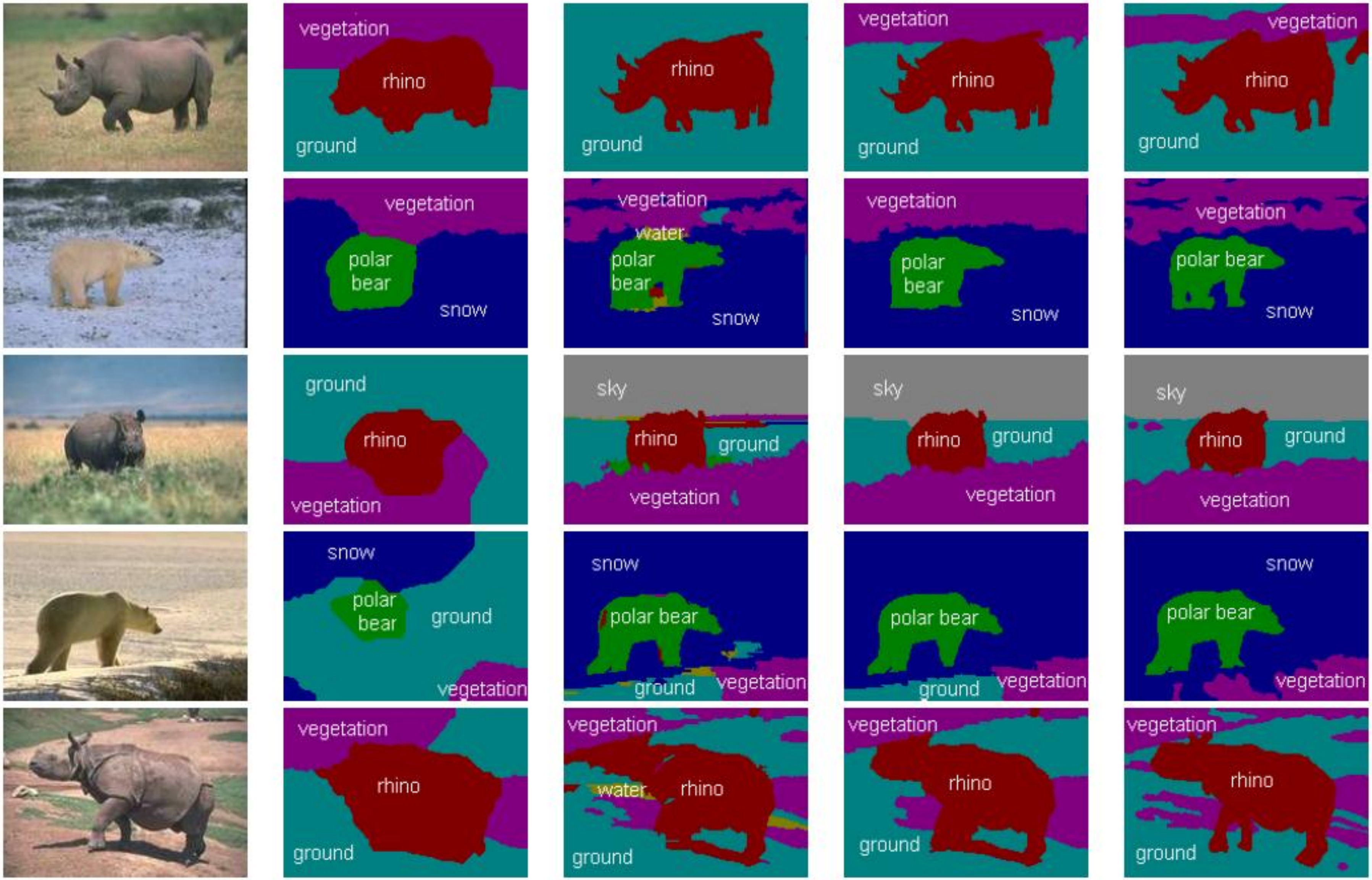
UNIVERSITY OF OXFORD

Results on MSRC dataset



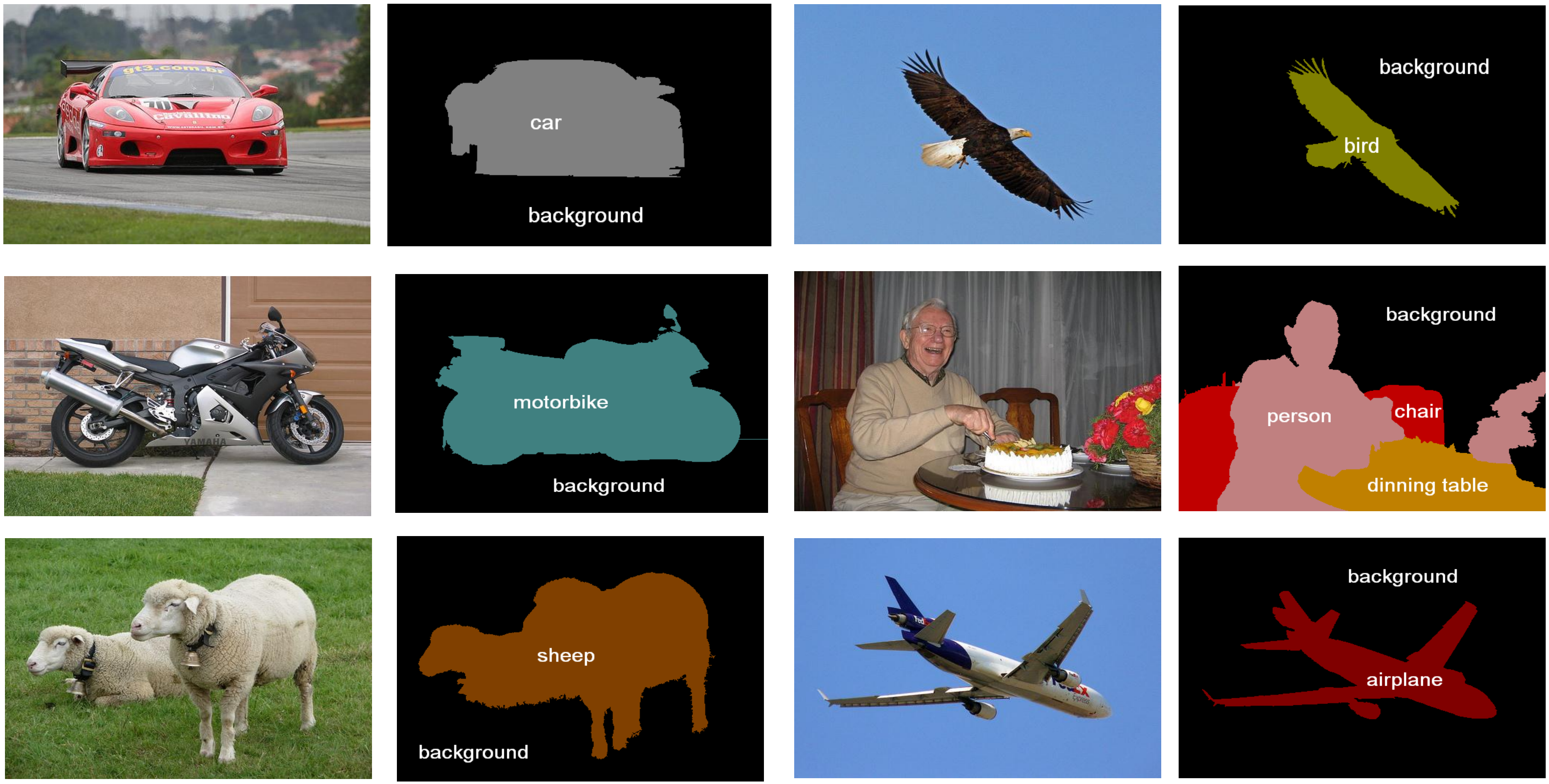
Original Image | Pixel-based CRF | Segment-based CRF | Hierarchical CRF | Ground Truth

Results on Corel dataset



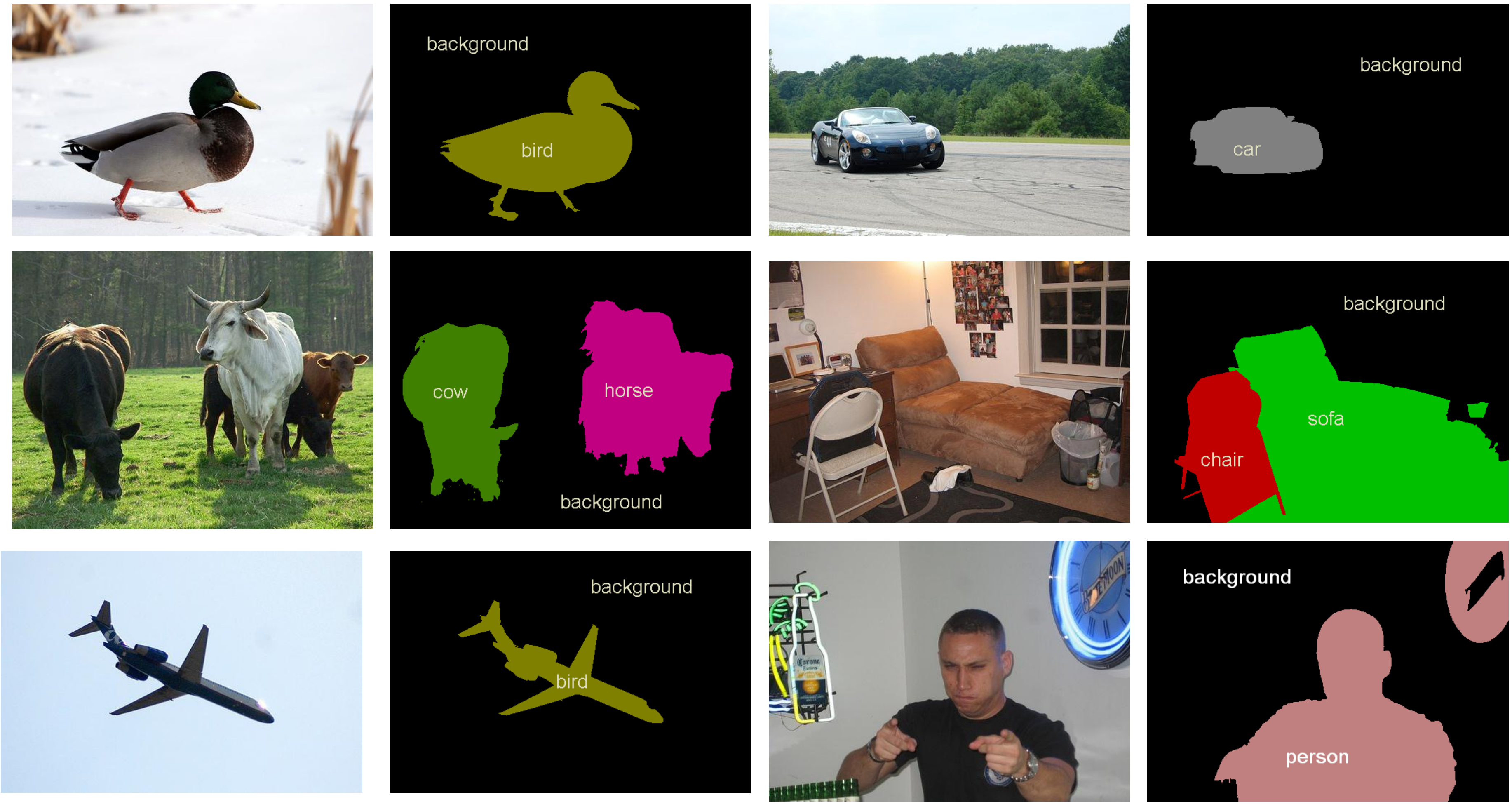
Original Image | Pixel-based CRF | Segment-based CRF | Hierarchical CRF | Ground Truth

Results on VOC 2010 dataset



Input Image Result Input Image Result

VOC 2010-Qualitative



Input Image | Result | Input Image | Result

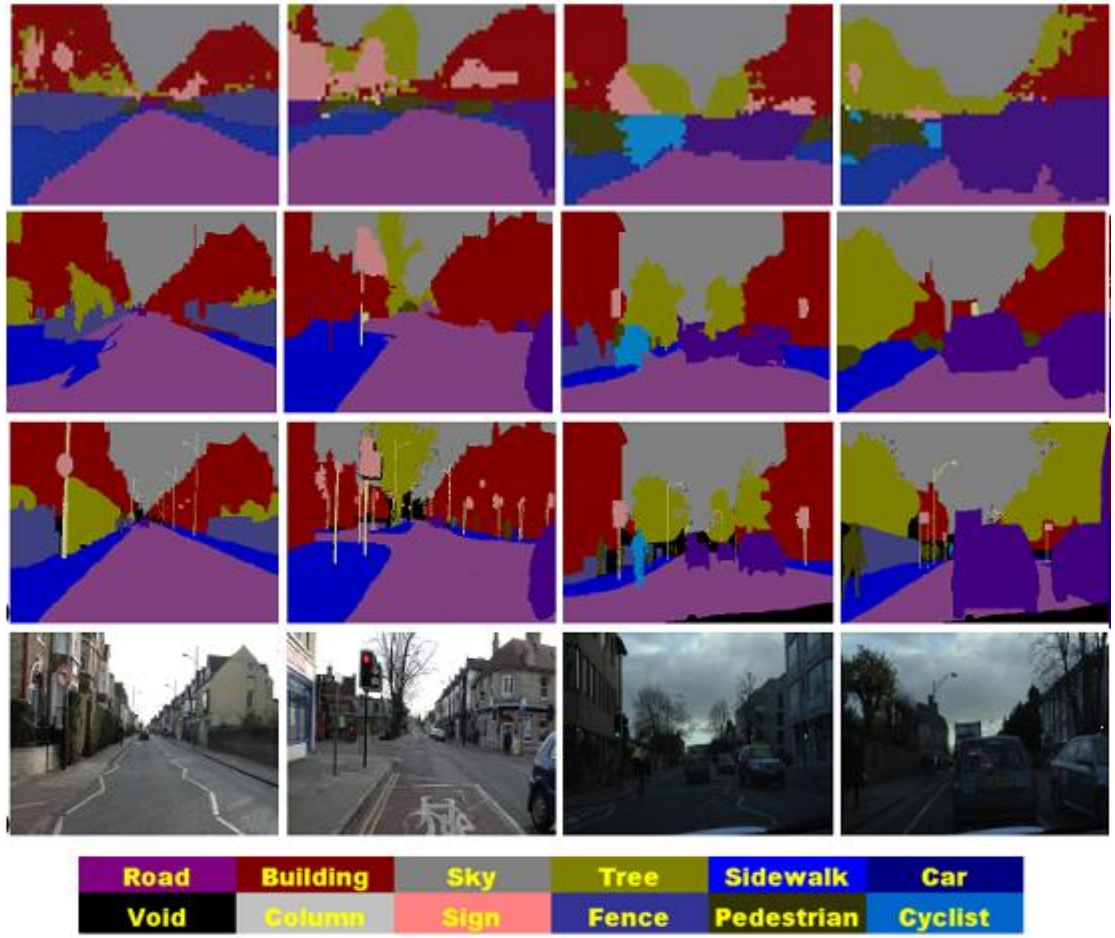
CamVid-Qualitative

Brostow et al.

Our Result

Ground Truth

Input Image



Quantitative comparison

MSRC dataset	Global	Average	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Shotton et al. [Boost]	72	67	49	88	79	97	97	78	82	54	87	74	72	74	36	24	93	51	78	75	35	66	18
Shotton et al. [Forest]	72	58	62	98	86	58	50	83	60	53	74	63	75	63	35	19	92	15	86	54	19	62	07
Batra et al.	70	55	68	94	84	37	55	68	52	71	47	52	85	69	54	05	85	21	66	16	49	44	32
Yang et al.	75	62	63	98	89	66	54	86	63	71	83	71	79	71	38	23	88	23	88	33	34	43	32
Pixel-based CRF	81	72	73	92	85	75	78	92	75	76	86	79	87	96	95	31	81	34	84	53	61	60	15
Robust P^N CRF	83	73	74	92	86	75	83	94	75	83	86	85	84	95	94	30	86	35	87	53	73	63	16
Segment-based CRF	75	60	64	95	78	53	86	99	71	75	70	71	52	72	81	20	58	20	89	26	42	40	05
Hierarchical CRF	86	75	80	96	86	74	87	99	74	87	86	87	82	97	95	30	86	31	95	51	69	66	09

VOC2009 dataset	Average	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV/monitor
BONN_SVM-SEGM	36.3	83.9	64.3	21.8	21.7	32.0	40.2	57.3	49.4	38.8	5.2	28.5	22.0	19.6	33.6	45.5	33.6	27.3	40.4	18.1	33.6	46.1
CVC_HOCRF	34.5	80.2	67.1	26.6	30.3	31.6	30.0	44.5	41.6	25.2	5.9	27.8	11.0	23.1	40.5	53.2	32.0	22.2	37.4	23.6	40.3	30.2
UoCTTI_LSVM-MDPM	29.0	78.9	35.3	22.5	19.1	23.5	36.2	41.2	50.1	11.7	8.9	28.5	1.4	5.9	24.0	35.3	33.4	35.1	27.7	14.2	34.1	41.8
NECUIUC_CLS-DTCT	29.7	81.8	41.9	23.1	22.4	22.0	27.8	43.2	51.8	25.9	4.5	18.5	18.0	23.5	26.9	36.6	34.8	8.8	28.3	14.0	35.5	34.7
NECUIUC_CLS-DTCT	25.7	79.1	44.6	15.5	20.5	13.3	28.8	29.3	35.8	25.4	4.4	20.3	1.3	16.4	28.2	30.0	24.5	12.2	31.5	18.3	28.8	31.9
BROOKESMSRC_AHCRF	24.8	79.6	48.3	6.7	19.1	10.0	16.6	32.7	38.1	25.3	5.5	9.4	25.1	13.3	12.3	35.5	20.7	13.4	17.1	18.4	37.5	36.4
Our method	32.1	81.2	46.1	15.4	24.6	20.9	36.9	50.0	43.9	28.4	11.5	18.2	25.4	14.7	25.1	37.7	34.1	27.7	29.6	18.4	43.8	40.8

Corel dataset	Global	Average	Rhino/Hippo	Polar Bear	Water	Snow	Grass	Ground	Sky
Batra et al.	83	**85**	87	**92**	82	**91**	66	**83**	**94**
Pixel-based CRF	76	72	80	85	88	83	75	57	35
Segment-based CRF	80	78	**92**	65	91	84	81	67	73
Hierarchical CRF	**84**	**85**	**92**	82	**94**	88	**83**	77	76

Quantitative comparison

MSRC dataset

	Global	Average	Building	Grass	Tree	Cow	Sheep	Sky	Aeroplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Segment-based CRF	81	66	80	**98**	83	64	81	**99**	59	89	85	68	68	**98**	76	26	85	39	84	30	49	50	07
Segment-based CRF with CO	82	68	81	**98**	83	65	81	**99**	59	91	85	69	68	**98**	76	27	85	39	85	29	49	51	07
Hierarchical CRF	87	78	81	96	**89**	74	**84**	**99**	**84**	92	**90**	86	**92**	**98**	91	**35**	**95**	53	90	62	**77**	**70**	**12**
Hierarchical CRF with CO	**89**	**80**	**83**	96	**89**	**75**	**84**	**99**	**84**	**94**	**90**	**87**	**92**	**98**	**92**	**35**	**95**	**55**	**91**	**64**	**77**	**70**	11

VOC2009 dataset

	Average	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV/monitor
Hierarchical CRF	27	78	38	10	**24**	36	**31**	59	37	21	08	**02**	**23**	**14**	17	27	21	15	**16**	15	48	33
Hierarchical CRF with CO	**31**	**82**	**49**	**12**	19	**38**	**31**	**63**	**46**	**24**	**10**	01	**23**	**14**	**22**	**34**	**36**	**18**	12	**23**	**53**	**37**

Ladický, Russell, Kohli, Torr ICCV09

Dense Stereo Reconstruction

$$E^D(\mathbf{y}) = \boxed{\sum_{i \in \mathcal{V}} \psi_i^D(y_i)} + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}^D(y_i, y_j)$$

Unary Potential

Disparity = 0

$$\boxed{\psi_i^D(0) = d\,(P_1(x_i\,,\, y_i),\, P_2(x_i + 0,\, y_i\,))}$$

Unary Cost dependent on the similarity of patches, e.g.cross correlation

Dense Stereo Reconstruction

$$E^D(\mathbf{y}) = \boxed{\sum_{i\in\mathcal{V}} \psi_i^D(y_i)} + \sum_{i\in\mathcal{V}, j\in\mathcal{N}_i} \psi_{ij}^D(y_i, y_j)$$

Unary Potential

Disparity = 5

$$\boxed{\psi_i^D(5) = d\,(P_1(x_i\,, y_i), P_2(x_i + 5,\, y_i\,))}$$

Unary Cost dependent on the similarity of patches, e.g.cross correlation

Dense Stereo Reconstruction

$$E^D(\mathbf{y}) = \boxed{\sum_{i \in \mathcal{V}} \psi_i^D(y_i)} + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}^D(y_i, y_j)$$

Unary Potential

Disparity = 10

$$\boxed{\psi_i^D(10) = d\,(P_1(x_i\,,\,y_i),\, P_2(x_i + 10,\, y_i))}$$

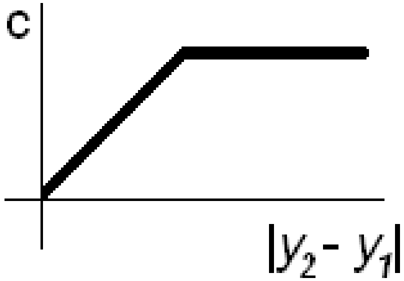
Unary Cost dependent on the similarity of patches, e.g.cross correlation

Dense Stereo Reconstruction

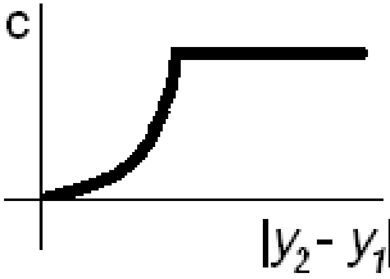
$$E^D(\mathbf{y}) = \sum_{i \in \mathcal{V}} \psi_i^D(y_i) + \boxed{\sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}^D(y_i, y_j)}$$

Pairwise Potential

- Encourages label consistency in adjacent pixels
- Cost based on the distance of labels



Linear Truncated

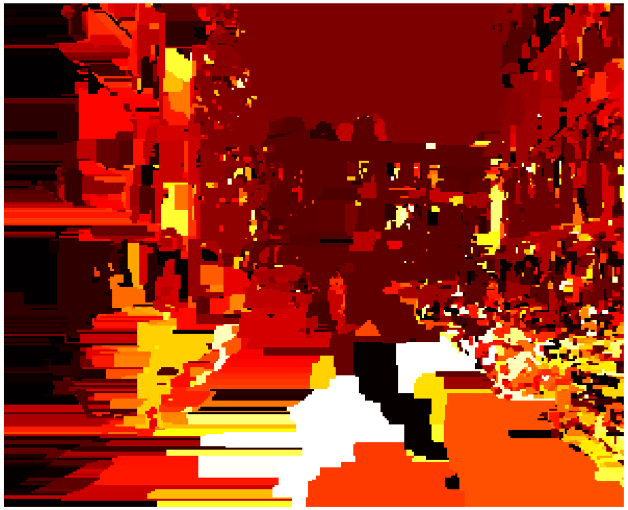


Quadratic Truncated

Dense Stereo Reconstruction

Does not work for Road Scenes !

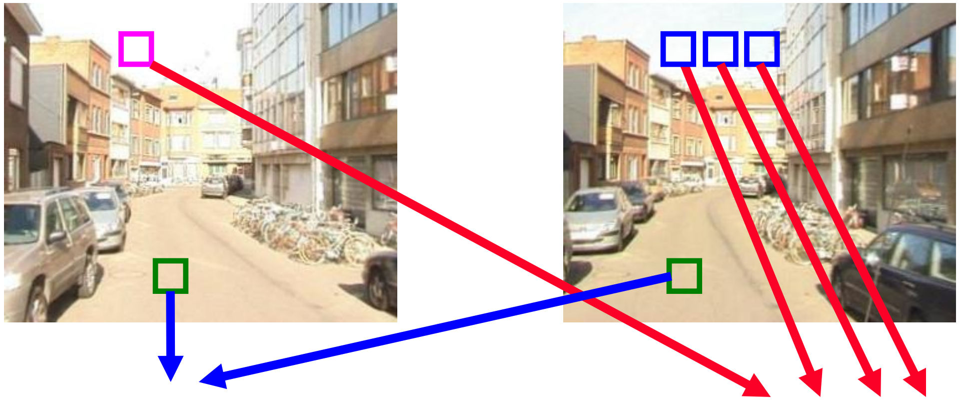
Original Image



Dense Stereo Reconstruction

Dense Stereo Reconstruction

Does not work for Road Scenes !

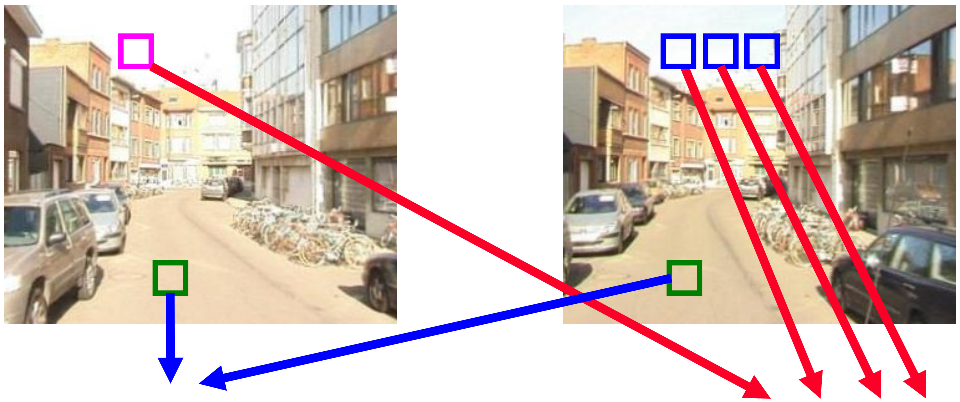


Different brightness in cameras

Patches can be matched to any other patch for flat surfices

Dense Stereo Reconstruction

Does not work for Road Scenes !



Different brightness in cameras

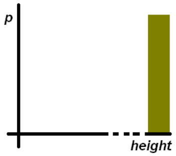
Patches can be matched to any other patch for flat surfices

Could object recognition for road scenes help?

Recognition of road scenes is relatively easy

Joint Dense Stereo Reconstruction and Object class Segmentation

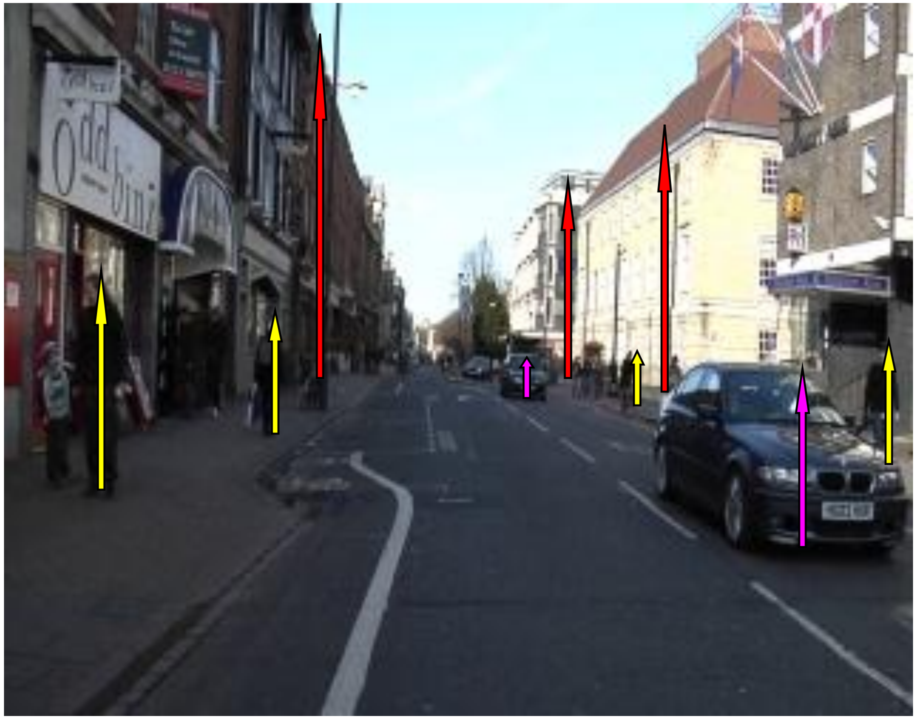
- Object class and 3D location are mutually informative
  - Sky always in infinity (disparity = 0)

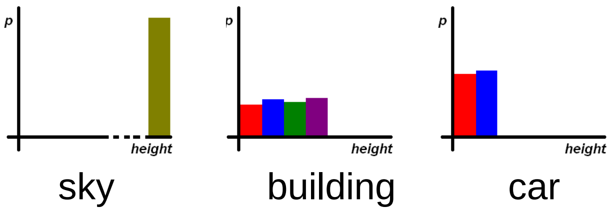


sky

Joint Dense Stereo Reconstruction and Object class Segmentation

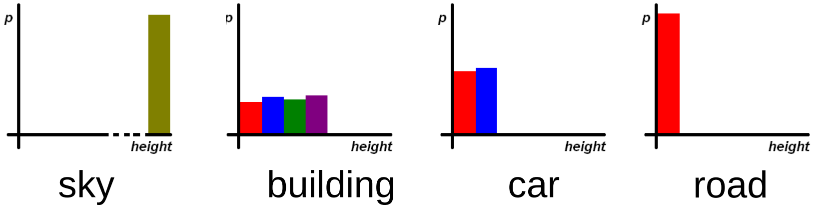
- Object class and 3D location are mutually informative
  - Sky always in infinity (disparity = 0)
  - Cars, buses & pedestrians have their typical height





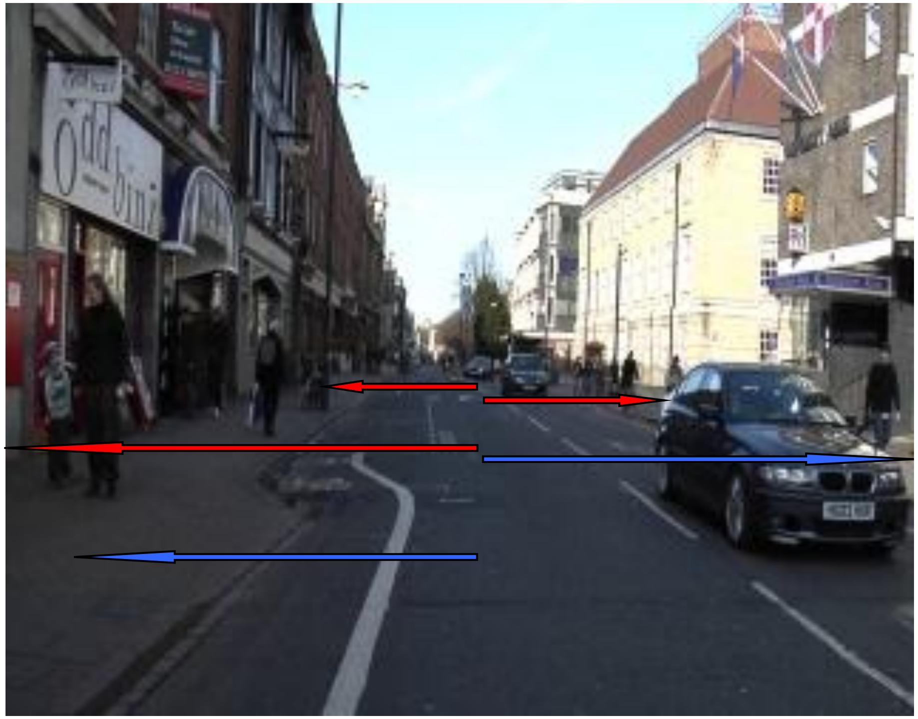
Joint Dense Stereo Reconstruction and Object class Segmentation

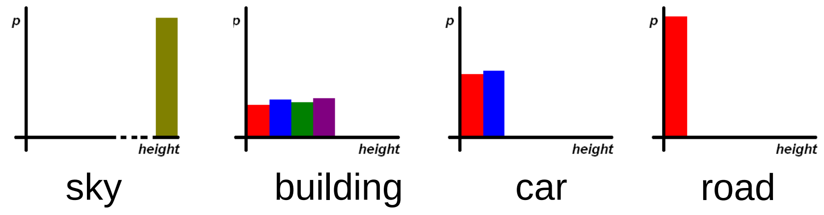
- Object class and 3D location are mutually informative
  - Sky always in infinity (disparity = 0)
  - Cars, buses & pedestrians have their typical height
  - Road and pavement on the ground plane



Joint Dense Stereo Reconstruction and Object class Segmentation

- Object class and 3D location are mutually informative
  - Sky always in infinity (disparity = 0)
  - Cars, buses & pedestrians have their typical height
  - Road and pavement on the ground plane
  - Buildings and pavement on the sides

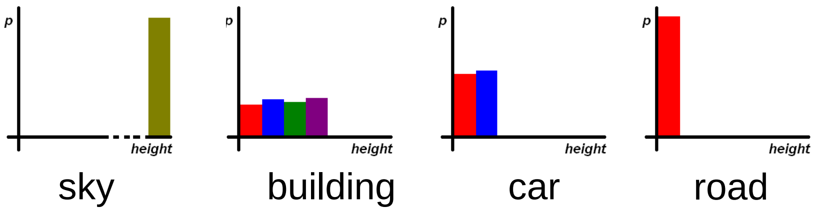




Joint Dense Stereo Reconstruction and Object class Segmentation

- Object class and 3D location are mutually informative
  - Sky always in infinity (disparity = 0)
  - Cars, buses & pedestrians have their typical height
  - Road and pavement on the ground plane
  - Buildings and pavement on the sides

- Both problems formulated as CRF
  - Joint approach possible?



Joint Formulation

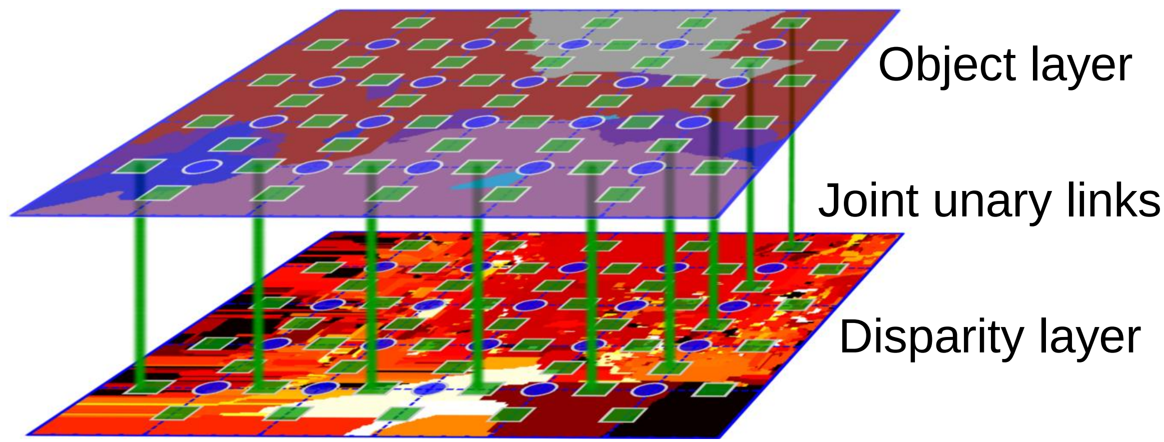
$$E(\mathbf{z}) = \sum_{i \in \mathcal{V}} \psi_i^J(z_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}^J(z_i, z_j) + \sum_{c \in \mathcal{C}} \psi_c^J(\mathbf{z}_c)$$

- Each pixels takes label $z_i = [\, x_i \;\; y_i \,] \in L_1 \times L_2$
- Dependency of x_i and y_i encoded as a unary and pairwise potential, *e.g.*
  - strong correlation between x = *road*, y = *near ground plane*
  - strong correlation between x = sky, y = *0*
  - Correlation of edge in object class and disparity domain

Joint Formulation

$$E(\mathbf{z}) = \boxed{\sum_{i\in\mathcal{V}} \psi_i^J(z_i)} + \sum_{i\in\mathcal{V},j\in\mathcal{N}_i} \psi_{ij}^J(z_i,z_j) + \sum_{c\in\mathcal{C}} \psi_c^J(\mathbf{z}_c)$$

Unary Potential



- Weighted sum of object class, depth and joint potential

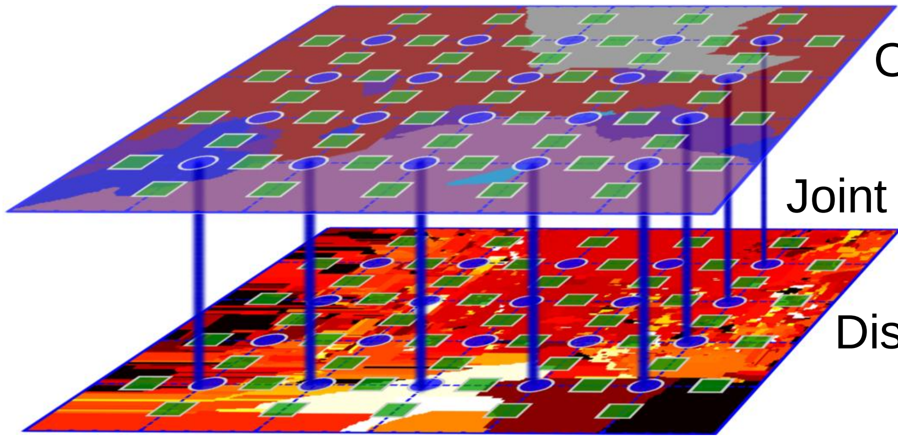
$$\psi_i^J([x_i,y_i]) = w_O^u \psi_i^O(x_i) + w_D^u \psi_i^D(y_i) + w_C^u \psi_i^C(x_i,y_i)$$

- Joint unary potential based on histograms of height

Joint Formulation

$$E(\mathbf{z}) = \sum_{i \in \mathcal{V}} \psi_i^J(z_i) + \boxed{\sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}^J(z_i, z_j)} + \sum_{c \in \mathcal{C}} \psi_c^J(\mathbf{z}_c)$$

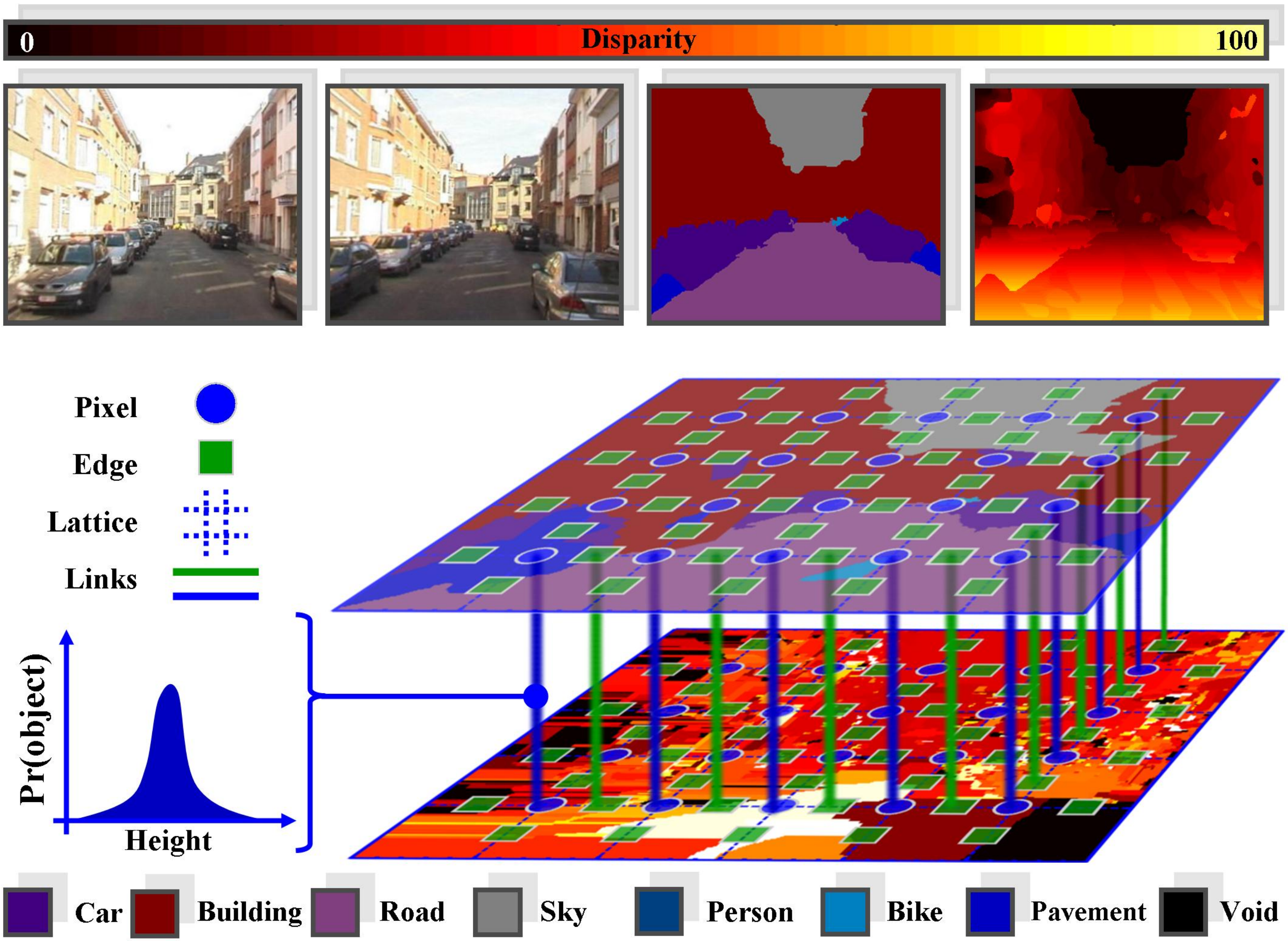
Pairwise Potential



- Object class and depth edges correlated
- Transitions in depth occur often at the object boundaries

$$\psi_{ij}^J([x_i, y_i], [x_j, y_j]) = w_O^p \psi_{ij}^O(x_i, x_j) + w_D^p \psi_{ij}^D(y_i, y_j) + w_C^p \psi_{ij}^O(x_i, x_j) \psi_{ij}^D(y_i, y_j)$$

Joint Formulation



Inference

- Standard α-expansion
  - Each node in each expansion move keeps its old label or takes a new label $[x_{L1}, y_{L2}]_{,}$
  - Possible in case of metric pairwise potentials

Inference

- Standard α-expansion
  - Each node in each expansion move keeps its old label or takes a new label [y_{L1}, y_{L2}]
  - Possible in case of metric pairwise potentials

Too many moves! (|*L1*| |*L2*|)

Impractical !

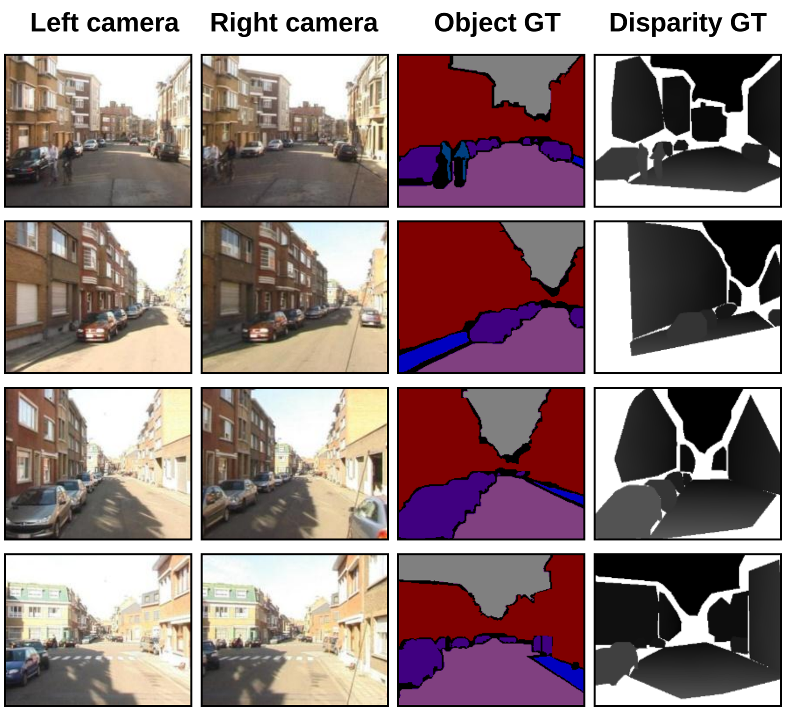
UNIVERSITY OF OXFORD

Inference

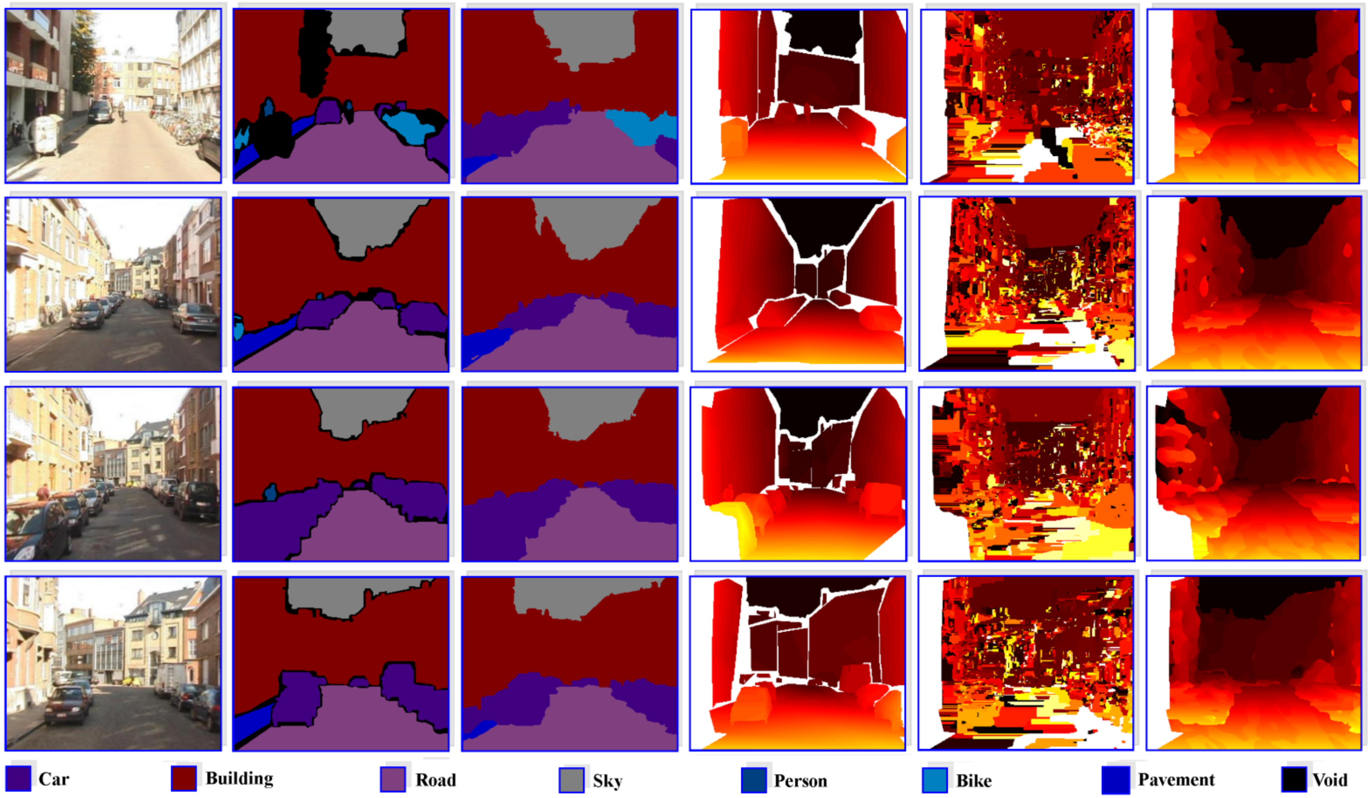
- Projected move for product label space
  - One / Some of the label components remain(s) constant after the move
- Set of projected moves
  - α-expansion in the object class projection
  - Range-expansion in the depth projection

Dataset

- Leuven Road Scene dataset
  - Contained
    - 3 sequences
    - 643 pairs of images
  - We labelled
    - 50 training + 20 test images
    - Object class (7 labels)
    - Disparity (100 labels)
  - Publicly available
    - *http://cms.brookes.ac.uk/research/visiongroup/files/Leuven.zip*

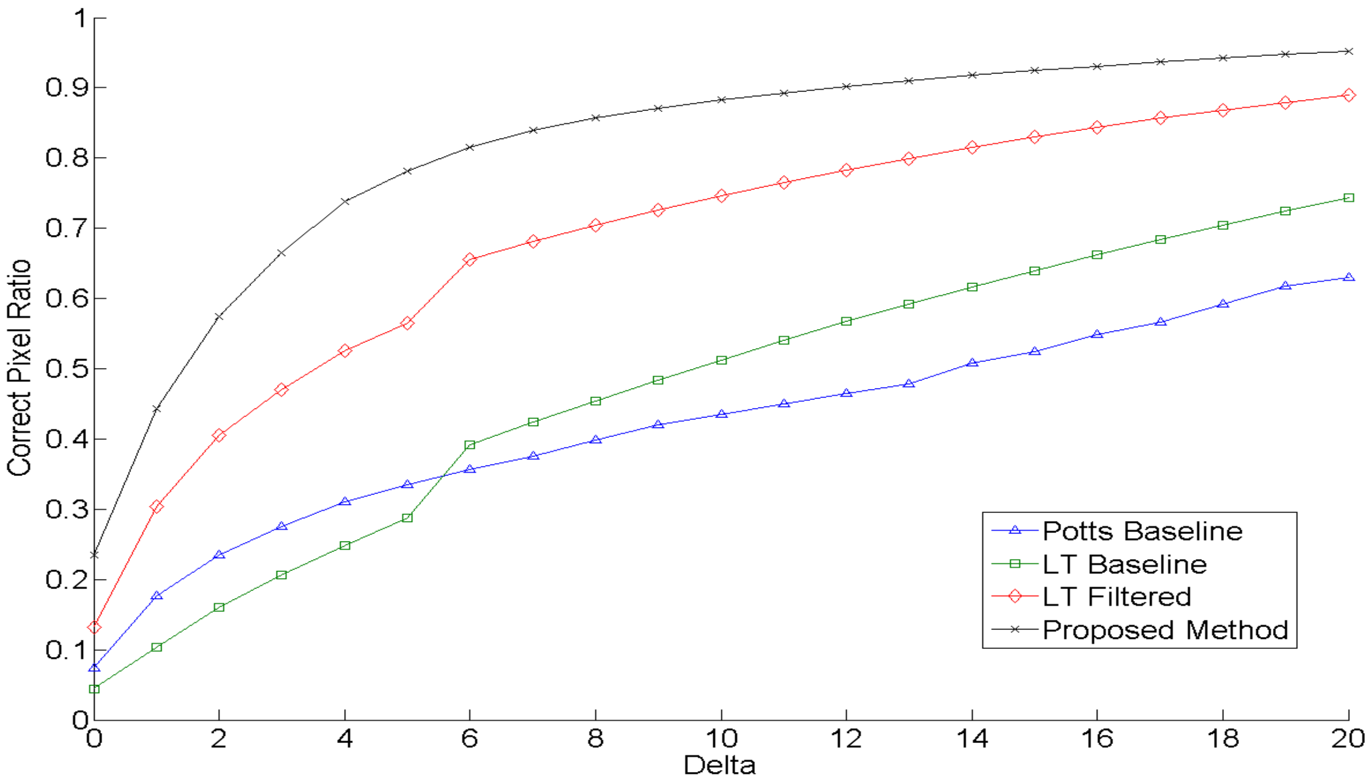


Qualitative Results



- Large improvement for dense stereo estimation
- Minor improvement in object class segmentation

Quantitative Results



Dependency of the ratio of correctly labelled pixels within the maximum allowed error delta

UNIVERSITY OF OXFORD

Summary

- We proposed :
  - New models enforcing higher order structure
    - Label-consistency in segments
    - Consistency between scale
    - Label-set consistency
    - Consistency between different domains
  - Graph-Cut based inference methods
  - Source code http://cms.brookes.ac.uk/staff/PhilipTorr/ale.htm

There is more to come !

Thank you

Questions?