Graph Cuts and Application to Disparity Map Estimation

Renaud Marlet
LIGM/École des Ponts ParisTech - IMAGINE

http://imagine.enpc.fr/~marletr

(with many borrowings from Boykov & Veksler 2006)
Motivating graph cuts

- **Powerful multidimensional energy minimization tool**
  - wide class of binary and non-binary energies
  - in some cases, globally optimal solutions
  - some provably good approximations (and good in practice)
  - allowing regularizers with contrast preservation
    - enforcement of piecewise smoothness while preserving relevant sharp discontinuities

- **Geometric interpretation**
  - hypersurface in $n$-D space

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V_{pq}(f_p, f_q)$$
Many links to other domains

- Combinatorial algorithms (e.g., dynamic programming)
- Simulated annealing
- Markov random fields (MRFs)
- Random walks and electric circuit theory
- Bayesian networks and belief propagation
- Level sets and other variational methods
- Anisotropic diffusion
- Statistical physics
- Submodular functions
- Integral/differential geometry, etc.

(cf. Boykov & Veksler 2006)
Overview of the course

• Notions
  – graph cut, minimum cut
  – flow network, maximum flow
  – optimization: exact (global), approximate (local)

• Illustration with emblematic applications

segmentation

disparity map estimation
Overview of the course

- Notions
  - graph cut, minimum cut
  - flow network, maximum flow
  - optimization: exact (global), approximate (local)

- Illustration with emblematic applications

segmentation

disparity map estimation

No time to go deep into every topic → read the references
Part 1

Graph cuts basics

Max-flow min-cut theorem

Application to image restoration and image segmentation
Graph cut basics

- Graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ (digraph)
  - set of nodes (vertices) $\mathcal{V}$
  - set of directed edges $\mathcal{E}$
    - $p \rightarrow q$
- $\mathcal{V} = \{ s, t \} \cup \mathcal{P}$
  - terminal nodes: $\{ s, t \}$
    - $s$: source node
    - $t$: target node (= sink)
  - non-terminal nodes: $\mathcal{P}$
    - ex. $\mathcal{P}$ = set of pixels, voxels, etc.
      (can be very different from an image)
Graph cut basics

- **Edge labels**, for $p \rightarrow q \in \mathcal{E}$
  - $c(p,q) \geq 0$: **nonnegative** costs
    also called weights $w(p,q)$
  - $c(p,q)$ and $c(q,p)$, **if any**, may differ

- **Links**
  - **t-link**: term. $\leftrightarrow$ non-term.
    - \{s $\rightarrow$ p $|$ p $\neq$ t\}, \{q $\rightarrow$ t $|$ q $\neq$ s\}
  - **n-link**: non-term. $\rightarrow$ non-term.
    - $\mathcal{N} = \{p \rightarrow q \mid p,q \neq s,t\}$
Cut and minimum cut

- **s-t cut** (or just “cut”): \( C = \{ S, \mathcal{T} \} \)
  node partition such that \( s \in S \), \( t \in \mathcal{T} \)

- **Cost of a cut** \( \{ S, \mathcal{T} \} \):
  \[
  c(S, \mathcal{T}) = \sum_{p \in S, q \in \mathcal{T}} c(p, q)
  \]
  - N.B. cost of severed edges: **only from** \( S \) **to** \( \mathcal{T} \)

- **Minimum cut**:
  - i.e., with min cost: \( \min_{S, \mathcal{T}} c(S, \mathcal{T}) \)
  - intuition: cuts only “weak” links

\( c(p,q) \) = coupé, sectionné
Different view: flow network
(or transportation network)

- Different vocabulary and features
  - graph ↔ network
    - vertex = node \( p, q, \ldots \)
    - edge = arc \( p \rightarrow q \) or \( (p,q) \)
    - cost = capacity \( c(p,q) \)
      - possibly many sources & sinks
  - Flow \( f: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R} \)
    - \( f(p,q) \): amount of flow \( p \rightarrow q \)
      - \( (p,q) \notin \mathcal{E} \Leftrightarrow c(p,q) = 0, f(p,q) = 0 \)
    - e.g. road traffic, fluid in pipes, current in electrical circuit, ...
Flow network constraints

- **Capacity constraint**
  - \( f(p,q) \leq c(p,q) \)

- **Skew symmetry**
  - \( f(p,q) = -f(q,p) \)

- **Flow conservation**
  - \( \forall p, \text{ net flow } \sum_{q \in V} f(p,q) = 0 \)
  - unless \( p = s \) (\( s \) produces flow)
  - or \( p = t \) (\( t \) consumes flow)
  - i.e., incoming \( \sum_{(q,p) \in E} f(q,p) \)
  = outgoing \( \sum_{(p,q) \in E} f(p,q) \)
Flow network constraints

- **s-t flow** (or just “flow) $f$
  - $f: \mathcal{V} \times \mathcal{V} \to \mathbb{R}$
  - satisfying flow constraints

- **Value of s-t flow**
  - $|f| = \sum_{q \in \mathcal{V}} f(s,q) = \sum_{p \in \mathcal{V}} f(p,t)$
    - amount of flow from source = amount of flow to sink

- **Maximum flow**:
  - i.e., with maximum value: $\max_f |f|$
  - intuition: arcs saturated as much as possible
Max-flow min-cut theorem

- **Theorem**
  
  The maximum value of an \( s-t \) flow is equal to the minimum capacity (i.e., min cost) of an \( s-t \) cut.

- **Example**
  
  \[- |f| = c(S, T) = ? \]
Max-flow min-cut theorem

- **Theorem**
  The maximum value of an $s$-$t$ flow is equal to the minimum capacity (i.e., min cost) of an $s$-$t$ cut.

- **Example**
  - $|f| = c(S, \overline{T}) = 4$
  - min: enumerate partitions...
Max-flow min-cut theorem

- **Theorem**
  The maximum value of an $s$-$t$ flow is equal to the minimum capacity (i.e., min cost) of an $s$-$t$ cut.

- **Example**
  - $|f| = c(S,T) = 4$
  - min: enumerate partitions...
  - max: try increasing $f(p,q)$...

![Diagram](image_url)
Max-flow min-cut theorem

• Theorem

The maximum value of an $s$-$t$ flow is equal to the minimum capacity (i.e., min cost) of an $s$-$t$ cut.

• Example

- $|f| = c(S, T) = 4$
- min: enumerate partitions...
- max: try increasing $f(p,q)$...
Max-flow min-cut theorem

• Theorem
  The maximum value of an $s$-$t$ flow is equal to the minimum capacity (i.e., min cost) of an $s$-$t$ cut.

• Example
  - $|f| = c(S, T) = 4$
  - min: enumerate partitions...
  - max: try increasing $f(p, q)$...

\[
\begin{align*}
|f| &= 3 \\
S &\quad T \\
edge label: f(p,q)/c(p,q)
\end{align*}
\]
Max-flow min-cut theorem

- **Theorem**
  The maximum value of an \( s-t \) flow is equal to the minimum capacity (i.e., min cost) of an \( s-t \) cut.

- **Example**
  - \(|f| = c(S, T) = 4\)
  - min: enumerate partitions...
  - max: try increasing \( f(p,q) \)...

\[ |f| = 4 \]
Max-flow min-cut theorem

• Theorem
The maximum value of an $s$-$t$ flow is equal to the minimum capacity (i.e., min cost) of an $s$-$t$ cut.

• Example
- $|f| = c(S, T) = 4$
- min: enumerate partitions...
- max: try increasing $f(p,q)$...

• Intuition
- pull $s$ and $t$ apart: the graph tears where it is weak
- min cut: cut corresponding to a small number of weak links
- max flow: flow bounded by low-capacity links in a cut
Max-flow min-cut theorem

- Theorem
  The maximum value of an $s-t$ flow is equal to the minimum capacity (i.e., min cost) of an $s-t$ cut.

- proved independently by Elias, Feinstein & Shannon, and Ford & Fulkerson (1956)

- special case of strong duality theorem in linear programming

- can be used to derive other theorems
Max flows and min cuts configurations are not unique

- Different configurations with same maximum flow
  
  edge label: $f(p,q)/c(p,q)$

- Different configurations with same min-cut cost
Algorithms for computing max flow

• Polynomial time

• Push-relabel methods
  – better performance for general graphs
  – e.g. Goldberg and Tarjan 1988: $O(VE \log(V^2/E))$
    • where $V$: number of vertices, $E$: number of edges

• Augmenting paths methods
  – iteratively push flow from source to sink along some path
  – better performance on specific graphs
  – e.g. Ford-Fulkerson 1956: $O(E \max|f|)$ for integer capacity $c$
Residual network/graph

- Given flow network $G = \langle V, E, c, f \rangle$

Define residual network $G_f = \langle V, E, c_f, 0 \rangle$ with
  - residual capacity $c_f(p,q) = c(p,q) - f(p,q)$
  - no flow, i.e., value 0 for all edges

- Example:
Ford-Fulkerson algorithm (1956)

\[ f(p,q) \leftarrow 0 \text{ for all edges} \]

while \( \exists \text{ path } P \text{ from } s \text{ to } t \text{ such that } \forall (p,q) \in P \ c_f(p,q) > 0 \)

\[ c_f(P) \leftarrow \min \{c_f(p,q) \mid (p,q) \in P\} \]

for each edge \((p,q) \in P\)

\[ f(p,q) \leftarrow f(p,q) + c_f(P) \]

\[ f(q,p) \leftarrow f(q,p) - c_f(P) \]

• N.B. termination not guaranteed
  - maximum flow reached if (semi-)algorithm terminates
    (but may “converge” to less than maximum flow if it does not terminate)
  - always terminates for integer values (or rational values)
Ford-Fulkerson algorithm: an example

$f/c$

$c_f: c_f = c - c_f$

$c_f(P): c_f$

$|f| = 1 + 2 + 1 = 4 = c(S, T)$
Ford-Fulkerson algorithm: an example

Taking edges backwards = OK (and sometimes needed)

\[ f = c \]

\[ c_f(P) = c - c_f \]

\[ |f| = 1 + 1 + 1 + 1 = 4 = c(S, T) \]
Edmonds-Karp algorithm (1972)

• As Ford-Fulkerson but **shortest path** with >0 capacity
  - breadth-first search for augmenting path (cf. example above)

• Termination: now guaranteed

• Complexity: $O(VE^2)$
  - slower than push-relabel methods for general graphs
  - faster in practice for sparse graphs

• Other variant (Dinic 1970), complexity: $O(V^2E)$
  - other flow selection (blocking flows)
  - $O(VE \log V)$ with dynamic trees (Sleator & Tarjan 1981)
Maximum flow for grid graphs

- Fast augmenting path algorithm (Boykov & Kolmogorov 2004)
  - often significantly outperforms push-relabel methods
  - observed running time is linear
  - many variants since then

- But push-relabel algorithm can be run in parallel
  - good setting for GPU acceleration

The battle continues... (amenable to real time for real pb ?)
Variant: Multiway cut problem

• More than two terminals: \( \{s_1, \ldots, s_k\} \)

• Multiway cut:
  - set of edges leaving each terminal in a separate component

• Multiway cut problem
  - find cut with minimum weight
  - same as min cut when \( k = 2 \)
  - NP-hard if \( k \geq 3 \) (in fact APX-hard, i.e., NP-hard to approx.)
  - but can be solved exactly for planar graphs
Graph cuts for binary optimization

• Inherently a binary technique
  - splitting in two

• 1st use in image processing:
  binary image restoration (Greig et al. 1989)
  - black&white image with noise → image with no noise

• Can be generalized to large classes of binary energy
  - regular functions
Binary image restoration

Noise = bruit
Threshold = seuil

Graph cut
Binary image restoration: the graph cut view

- Agreement with observed data
  - $D_p(l)$: penalty (= –reward) for assigning label $l \in \{0,1\}$ to pixel $p \in P$
  - if $I_p = l$ and $l' \neq l$ then $D_p(l') > D_p(l)$
  - $w(s,p) = D_p(1)$, $w(p,t) = D_p(0)$

- Example:
  - if $I_p = 0$, $D_p(0) = 0$, $D_p(1) = \kappa$
  - if $I_p = 1$, $D_p(0) = \kappa$, $D_p(1) = 0$
  - if $I_p = 0$ and $p \in S$, cost = $D_p(0) = 0$
  - if $I_p = 0$ and $p \in T$, cost = $D_p(1) = \kappa$

$I_p$: intensity of image $I$ at pixel $p$
Binary image restoration: the graph cut view

- Agreement with observed data
  - $D_p(l)$: penalty (reward) for assigning label $l \in \{0,1\}$ to pixel $p \in P$
  - if $I_p = l$ and $l' \neq l$ then $D_p(l') > D_p(l)$
  - $w(s,p) = D_p(1)$, $w(p,t) = D_p(0)$

- Minimize discontinuities
  - penalty for (long) contours
    - $w(p,q) = w(q,p) = \lambda > 0$
  - spatial coherence, regularizing constraint, smoothing factor... (see below)
Binary image restoration: the graph cut view

- **Binary labeling** $f$  [N.B. different from flow $f$]
  - assigns label $f_p \in \{0,1\}$ to pixel $p \in \mathcal{P}$
    - $f : \mathcal{P} \rightarrow \{0,1\}$  \hspace{1cm} \hspace{1cm} f(p) = f_p$
  - **Cut** $C = \{S, T\} \leftrightarrow$ labeling $f$
    - 1-to-1 correspondence: $f = 1|_T$
- **Cost of a cut:** $|C| = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in S \times T} w(p,q)$
  - \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} = \text{cost of flip} + \text{cost of local dissimilarity}$
- **Restored image:**
  - labeling corresponding to a minimum cut
Binary image restoration: the energy view

- **Energy of labeling** \( f \)

\[
E(f) \overset{\text{def}}{=} |C| = \sum_{p \in P} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} 1(f_p = 0 \land f_q = 1)
\]

where

\[
1(\text{false}) = 0 \quad | \quad 1(\text{true}) = 1
\]

[\text{ou: } \frac{1}{2} \lambda \sum_{(p,q) \in \mathcal{N}} 1(f_p \neq f_q)]

- **Restored image:**
  - labeling corresponding to minimum energy (= minimum cut)
Binary image restoration: the smoothing factor

- **Small $\lambda$**: pixels chose their label independently of their neighbors
- **Large $\lambda$**: pixels chose the label with smaller average cost
- **Balanced $\lambda$ value**: pixels form compact, spatially coherent clusters with same labels
  - noise/outliers conform to neighbors
Graph cuts for energy minimization

- Given some energy $E(f)$ such that
  
  $f: \mathcal{P} \rightarrow \mathcal{L} = \{0,1\}$ binary labeling
  
  $E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q)$

  $E_{\text{data}}(f)$ $E_{\text{regul}}(f)$

  - regularity condition (see below)
    
    $V_{p,q}(0,0) + V_{p,q}(1,1) \leq V_{p,q}(0,1) + V_{p,q}(1,0)$

- Then there is a graph whose minimum cut defines a labeling $f$ that reaches the minimum energy

(Kolmogorov & Zabih 2004) [N.B. Vladimir Kolmogorov, not Andrey Kolmogorov] [structure of graph somehow similar to above form]
Graph construction

- Preventing a t-link cut: infinite weight

\[ w = \infty \]

- Favoring a t-link cut: null weight (≈ no edge)

\[ w = 0 \]

- Bidirectional edge vs monodirectional & back edges
Graph cuts as hypersurfaces

- Cut on a 2D grid
- Cut on a 3D grid

N.B. Several “seeds” (sources and sinks)

(seed = graine)
Example of topological issue

- Connected seeds
- Disconnected seeds

seed = graine
Example of topological constraint: fold prevention

- Ex. in disparity map estimation: \( d = f(x,y) \)
- In 2D: \( y = f(x) \), only one value for \( y \) given one \( x \)
A “revolution” in optimization

• Previously (before Greig et al. 1989)
  – exact optimization like this was not possible
  – used approaches:
    • iterative algorithms such as simulated annealing
    • very far from global optimum, even in binary case like this
    • work of Greig et al. was (primarily) meant to show this fact

• Remained unnoticed for almost 10 years in the computer vision community...
  • maybe binary image restoration was viewed as too restrictive?
    (Boykov & Veksler 2006)
Graph cut techniques: now very popular in computer vision

• Extensive work since 1998
  – Boykov, Geiger, Ishikawa, Kolmogorov, Veksler, Zabih and others...

• Almost linear in practice (in nb nodes/edges)
  – but beware of the graph size: it can be exponential in the size of the problem

• Many applications
  – regularization, smoothing, restoration
  – segmentation
  – stereovision: disparity map estimation, ...
Warning:

global optimum \neq \text{best real-life solution}

- Graph cuts provide exact, global optimum
  - to binary labeling problems (under regularity condition)
- But the problem remains a model
  - approximation of reality
  - limited number of factors
  - parameters (e.g., $\lambda$)

\begin{itemize}
  \item Global optimum of \textit{abstracted problem,}
  \item not necessarily best solution \textit{in real life}
\end{itemize}
Not for free

- Many papers construct
  - their own graph
  - for their own specific energy function
- The construction can be fairly complex

Powerful tool but does not exempt from thinking
Application to image segmentation

● Problem:
  - given an image with foreground objects and background
  - given sample areas of both kind
  - separate objects from background

[Duchenne & al. 2008]
Application to image segmentation

• Problem:
  - given an image with foreground objects and background
  - given sample areas of both kinds ($O$, $B$)
  - separate objects from background
Intuition

What characterizes an object/background segmentation?
Intuition

What characterizes an object/background segmentation?

- pixels of segmented object and background look like corresponding sample pixels $O$ and $B$
- segment contours have high gradient, and are not too long
General formulation

- Pixel labeling with binary decision $f_p \in L = \{0, 1\}$
  - $1 = \text{object}$, $0 = \text{background}$
- Energy formulation
  - minimize $E(f) = D(f) + \lambda R(f)$
  - $D(f)$: data term (a.k.a. data fidelity term) = regional term
    - penalty for assigning labels $f$ in image $I$ given pixel sample assignments in $L$ : $O$ (object pixels), $B$ (background pixels)
  - $R(f)$: regularization term = boundary term
    - penalty for label discontinuity of neighboring pixels
  - $\lambda$: relative importance of regularization term vs data term

[Boykov & Jolly 2001]
Probabilistic justification/framework

• Minimize $E(f) \iff$ maximize posterior proba. $Pr(f|I)$

• Bayes theorem:

$$Pr(f|I) \cdot Pr(I) = Pr(I|f) \cdot Pr(f)$$

The term want to maximize w.r.t. $f$

A constant (independent of $f$)

$\iff$ data term, probability to observe image $I$ knowing labeling $f$

$\iff$ regularization term, depending on type of labeling and with various hypotheses (e.g., locality, cf. MRF below)

• Consider likelihoods $L(f|I) = Pr(I|f)$

• Actually consider log-likelihoods (→ sums)

$$E(f) = D(f) + \lambda R(f) \iff -\log Pr(f|I) + c = -\log Pr(I|f) - \log Pr(f)$$
Data term: linking estimated labels to observed pixels

- $D(f)$ and likelihood
  - penalty for assigning labels $f$ in $I$ given sample assignments $\leftrightarrow$ (log-)likelihood that $f$ is consistent with image samples
  - $D(f) = -\log L(f | I) = -\log \Pr(I | f)$

- Pixel independence hypothesis (common approximation)
  - $\Pr(I | f) = \prod_{p \in P} \Pr(I_p | f_p)$ if pixels iid (independent and identically distributed random variables)
  - $D(f) = \sum_{p \in P} D_p(f_p)$ where $D_p(f_p) = -\log \Pr(I_p | f_p)$
    - $D_p(f_p)$: penalty for observing $I_p$ for a pixel of type $f_p$

Find an estimate of $\Pr(I_p | f_p)$
Data term: likelihood/color model

- Approaches to find an estimate of $\Pr(I_p|f_p)$
  - histograms
    - build an empirical distribution of the color of object/background pixels, based on pixels marked as object/background
    - estimate $\Pr(I_p|f_p)$ based on histograms: $\Pr_{\text{emp}}(rgb|O), \Pr_{\text{emp}}(rgb|B)$
  - Gaussian Mixture Model (GMM)
    - model the color of object (resp. background) pixels with a distribution defined as a mixture of Gaussians
  - texon (or texton): texture patch (possibly abstracted)
    - compare with expected texture property: response to filters (spectral analysis), moments...
Regularization term: locality hypotheses

- Markov random field (MRF), or Markov network
  - neighborhood system: $\mathcal{N} = \{\mathcal{N}_p \mid p \in \mathcal{P}\}$
    - $\mathcal{N}_p$: set neighbors of $p$ such that $p \notin \mathcal{N}_p$ and $p \in \mathcal{N}_q \iff q \in \mathcal{N}_p$
  - $X = (X_p)_{p \in \mathcal{P}}$: field (set) of random variables such that each random variable $X_p$ depends on other random variables only through its neighbors $\mathcal{N}_p$

- locality hypothesis: $\Pr(X_p = x \mid X_{\mathcal{P}\setminus\{p\}}) = \Pr(X_p = x \mid X_{\mathcal{N}_p})$
  - $\mathcal{N}$ ≈ undirected graph: $(p,q)$ edge iff $p \in \mathcal{N}_q$ (⇔ $q \in \mathcal{N}_p$)

(MRF also called undirected graphical model)
Regularization term: locality hypotheses

- **Gibbs random field (GRF)**
  - $G$ undirected graph, $X = \{X_p\}_{p \in P}$ random variables such that
    \[
    \Pr(X = x) \propto \exp(-\sum_{C \text{ clique of } G} V_C(x))
    \]
    - clique = complete subgraph: $\forall p \neq q \in C \ (p,q) \in G$
    - $V_C$: clique potential = prior probability of the given realization of the elements of the clique $C$ (fully connected subgraph)

- **Hammersley-Clifford theorem (1971)**
  - If probability distribution has positive mass/density, i.e., if $\Pr(X = x) > 0$ for all $x$, then:
    \[
    X \text{ MRF w.r.t. graph } \mathcal{N} \iff X \text{ GRF w.r.t. graph } \mathcal{N}
    \]
    ❄️ provides a characterization of MRFs as GRFs
Regularization term: locality hypotheses

- Hypothesis 1: only $2^{nd}$-order cliques (i.e., edges)
  \[
  R(f) = -\log \Pr(f) = -\log \exp(-\sum_{(p,q) \text{ edge of } G} V_{(p,q)}(f)) \quad \text{[GRF]}
  \]
  \[
  = \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) \quad \text{[MRF pairwise potentials]}
  \]

- Hypothesis 2: (generalized) Potts model
  \[
  V_{p,q}(f_p, f_q) = B_{p,q} \mathbf{1}(f_p \neq f_q)
  \]
  i.e.,
  \[
  V_{p,q}(f_p, f_q) = 0 \quad \text{si } f_p = f_q
  \]
  \[
  V_{p,q}(f_p, f_q) = B_{p,q} \quad \text{si } f_p \neq f_q
  \]

(Origin: statistical mechanics
  - spin interaction in crystalline lattice
  - link with “energy” terminology)
Examples of boundary penalties (ad hoc)

- Penalize label discontinuity at intensity continuity
  \[ B_{p,q} = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) / \text{dist}(p,q) \]
  - large between pixels of similar intensities, i.e., when \( |I_p - I_q| < \sigma \)
  - small between pixels of dissimilar intensities, i.e., when \( |I_p - I_q| > \sigma \)
  - decrease with pixel distance \( \text{dist}(p,q) \) [here: 1 or \( \sqrt{2} \)]
  - \( \approx \) distribution of noise among neighboring pixels

- Penalize label discontinuity at low gradient
  \[ B_{p,q} = g(\| \nabla I_p \|) \] with \( g \) positive decreasing
  - e.g., \( g(x) = 1/(1 + cx^2) \)
  - penalization for label discontinuity at low gradient

[Boykov & Jolly 2001]
Wrapping up

- Pixel labeling with binary decision $f_p \in \{0,1\}$
  - $1 = \text{object}, \ 0 = \text{background}$

- Energy formulation
  - minimize $E(f) = D(f) + \lambda R(f)$
  - data term: $D(f) = \sum_{p \in \mathcal{P}} D_p(f_p)$
    - $D_p(f_p)$: penalty for assigning label $f_p$ to pixel $p$ given its color/texture
  - regularization term: $R(f) = \sum_{(p,q) \in \mathcal{N}} B_{p,q} \ 1(f_p \neq f_q)$
    - $B_{p,q}$: penalty for label discontinuity between neighbor pixels $p, q$
  - $\lambda$: relative importance of regularization term vs data term
Graph-cut formulation (version 1)

- Direct expression as graph-cut problem:
  
  \[ \mathcal{V} = \{s,t\} \cup \mathcal{P} \]
  
  \[ \mathcal{E} = \{(s,p) \mid p \in \mathcal{P}\} \cup \{(p,q) \mid p,q \in \mathcal{N}\} \cup \{(p,t) \mid p \in \mathcal{P}\} \]

<table>
<thead>
<tr>
<th>Edge</th>
<th>Weight</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>((p,q))</td>
<td>(\lambda B_{p,q})</td>
<td>((p,q) \in \mathcal{N})</td>
</tr>
<tr>
<td>((s,p))</td>
<td>(D_p(0))</td>
<td>(p \in \mathcal{P})</td>
</tr>
<tr>
<td>((p,t))</td>
<td>(D_p(1))</td>
<td>(p \in \mathcal{P})</td>
</tr>
</tbody>
</table>

- \(E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} B_{p,q} 1(f_p \neq f_q)\)

  - ex. \(D_p(l) = -\log \Pr_{\text{emp}}(I_p | f_p = l)\) [empirical probability for \(O\) et \(B\)]
  
  - ex. \(B_{p,q} = \exp(-(I_p - I_q)^2/2\sigma^2) / \text{dist}(p,q)\)
Graph-cut formulation (version 1)

- Direct expression as graph-cut problem:
  - $\mathcal{V} = \{s,t\} \cup \mathcal{P}$
  - $\mathcal{E} = \{(s,p) \mid p \in \mathcal{P}\} \cup \{(p,q) \mid p,q \in \mathcal{N}\} \cup \{(p,t) \mid p \in \mathcal{P}\}$

<table>
<thead>
<tr>
<th>Edge</th>
<th>Weight</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p,q)$</td>
<td>$\lambda B_{p,q}$</td>
<td>$(p,q) \in \mathcal{N}$</td>
</tr>
<tr>
<td>$(s,p)$</td>
<td>$D_p(0)$</td>
<td>$p \in \mathcal{P}$</td>
</tr>
<tr>
<td>$(p,t)$</td>
<td>$D_p(1)$</td>
<td>$p \in \mathcal{P}$</td>
</tr>
</tbody>
</table>

- $E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} B_{p,q} 1(f_p \neq f_q)$

- Any problem/risk with this formulation?
Graph-cut formulation (version 1)

- Direct expression as graph-cut problem:
  \[ \mathcal{V} = \{s,t\} \cup \mathcal{P} \]
  \[ \mathcal{E} = \{(s,p) \mid p \in \mathcal{P}\} \cup \{(p,q) \mid p,q \in \mathcal{N}\} \cup \{(p,t) \mid p \in \mathcal{P}\} \]

<table>
<thead>
<tr>
<th>Edge</th>
<th>Weight</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>((p,q))</td>
<td>(\lambda B_{p,q})</td>
<td>((p,q) \in \mathcal{N})</td>
</tr>
<tr>
<td>((s,p))</td>
<td>(D_p(0))</td>
<td>(p \in \mathcal{P})</td>
</tr>
<tr>
<td>((p,t))</td>
<td>(D_p(1))</td>
<td>(p \in \mathcal{P})</td>
</tr>
</tbody>
</table>

- \(E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} B_{p,q} 1(f_p \neq f_q)\)

- \(\mathcal{P}_b\): pixels of object/background samples not necessarily assigned with good label!
Graph-cut formulation (version 2)

- Obj/Bg samples now always labeled OK in minimal $f^*$

<table>
<thead>
<tr>
<th>Edge</th>
<th>Weight</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p,q)$</td>
<td>$\lambda B_{p,q}$</td>
<td>$(p,q) \in \mathcal{N}$</td>
</tr>
<tr>
<td>$(s,p)$</td>
<td>$D_p(0)$</td>
<td>$p \in \mathcal{P}, p \notin (O \cup B)$</td>
</tr>
<tr>
<td></td>
<td>$K$</td>
<td>$p \in O$</td>
</tr>
<tr>
<td></td>
<td>$0$</td>
<td>$p \in B$</td>
</tr>
<tr>
<td>$(p,t)$</td>
<td>$D_p(1)$</td>
<td>$p \in \mathcal{P}, p \notin (O \cup B)$</td>
</tr>
<tr>
<td></td>
<td>$0$</td>
<td>$p \in O$</td>
</tr>
<tr>
<td></td>
<td>$K$</td>
<td>$p \in B$</td>
</tr>
</tbody>
</table>

- where $K = 1 + \max_{p \in \mathcal{P}} \lambda \sum_{(p,q) \in \mathcal{N}} B_{p,q}$

$K \approx +\infty$, too expensive to pay $\Rightarrow$ never assigned
Some limitations
(here with simple color model)

• Is the segmentation OK?
Some limitations (here with simple color model)

color model not complex enough

sensitivity to regularization parameter

neighboring model not complex enough
Exercise 1: implement simple image segmentation using graph cuts

• Load image and display it
• Get object/background samples with mouse clicks (e.g., small square area around mouse position) and draw them
• Define graph as in course (version 2)
  - choose neighbor connectivity [4: or 8: ]
  - choose one of expressions defined in course
  - model color of object/background with 1 single gaussian centered around mean color
    - write w.r.t. mean color of obj/bg: what is then ?
• Compute max flow, extract cut, display image segments
• Experiment with various parameters (, , , ...)
• And comment what you observe
Part 2

Multi-label problems

Exact vs approximate solutions

Application to stereovision
(disparity/depth map estimation):
\[
\text{disparity/depth} \leftrightarrow \text{label}
\]
Two-label (binary) problem

- $\mathcal{P}$: set of sites (pixels, voxels...)
- $\mathcal{N}$: set of neighboring site pairs
- $\mathcal{L} = \{0,1\}$: binary labels
- $f: \mathcal{P} \rightarrow \mathcal{L}$: binary labeling \[ \text{[notation: } f_p = f(p) = l \text{]} \]
- $E: (\mathcal{P} \rightarrow \mathcal{L}) \rightarrow \mathbb{R}$: energy
  \[ E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p,f_q) \]
  - $D_p(l)$: label penalty for site $p$
  - $V_{p,q}(l,l')$: prior knowledge about optimal pairwise labeling

- Pb: find $f^*$ that reaches the minimum energy $E(f^*)$
Two-label problem assumptions

- \( E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) \)

- \( D_p(l) \): label penalty for site \( p \)
  - small for preferred label, large for undesired label
  - assumption \( D_p(l) \geq 0 \) (else add constant \( \rightarrow \) same optimum)

- \( V_{p,q}(l, l') \): prior knowledge on optimal pairwise labeling
  - in general, smoothness: non-decreasing function of \( 1(l \neq l') \)
  - e.g., \( V_{p,q}(l, l') = u_{p,q} \ 1(l \neq l') \)

- Regularity condition, required for min-cut (\( \Rightarrow c(p,q) \geq 0 \))
  - \( V_{p,q}(0,0) + V_{p,q}(1,1) \leq V_{p,q}(0,1) + V_{p,q}(1,0) \) [see below]
Multi-label problem

- $\mathcal{P}$: set of sites (pixels, voxels...)
- $\mathcal{N}$: set of neighboring site pairs
- $\mathcal{L}$: finite set of labels \((\mapsto \text{can model scalar or even vector})\)
  - e.g., \textbf{discretization} of intensity, stereo disparity, motion vector...
- $f: \mathcal{P} \to \mathcal{L}$ labeling
- $E: (\mathcal{P} \to \mathcal{L}) \to \mathbb{R}$: energy
  - $E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) = E_{\text{data}}(f) + E_{\text{regul}}(f)$
    - $D_p(l)$: label penalty for site $p$
    - $V_{p,q}(l_p, l_q)$: prior knowledge about optimal pairwise labeling

- Pb: find $f^*$ that reaches the minimum energy $E(f^*)$
Multi-label problem assumptions

- \( E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) \)

- \( D_p(l) \): label penalty for site \( p \)
  - small for preferred label, large for undesired label
  - assumption \( D_p(l) \geq 0 \) (else add constant \( \rightarrow \) same optimum)

- \( V_{p,q}(l_p, l_q) \): prior knowledge on optimal pairwise labeling
  - in general, smoothness prior: non-decreasing function of \( \|l_p - l_q\| \) [norm used if vector]
    - e.g., \( V_{p,q}(l_p, l_q) = \lambda_{p,q} \|l_p - l_q\| \)
    - smaller penalty for closer labels

smoothness = lissage
Graph cuts for “general” energy minimization

- Problem: find labeling $f^* : \mathcal{P} \rightarrow \mathcal{L}$ minimizing energy
  \[ E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) \]

- Question: can a **globally optimal** labeling $f^*$ be found using some graph-cut construction?

- Answer:
  - binary labeling: yes if $f$ is regular \( V_{p,q} \) is regular \( V_{p,q} \) \( V_{p,q} \) \( V_{p,q} \)
    \[
    V_{p,q}(0,0) + V_{p,q}(1,1) \leq V_{p,q}(0,1) + V_{p,q}(1,0)
    \]
    \( \text{otherwise it's NP-hard} \)
  - multi-labeling: yes if $V_{p,q}$ convex \( V_{p,q} \) \( V_{p,q} \)
    \[
    \text{and if } \mathcal{L} \text{ linearly ordered (⇒ 1D only ⇒ not 2D motion vector)}
    \]
  - otherwise: approximate solutions (but some very good)
Piecewise-smooth vs everywhere-smooth

- Observation: object properties often smooth everywhere except on boundaries

- Consequence: piecewise-smooth models more appropriate than everywhere-smooth models

original  uniform smoothing  piecewise smoothing
Piecewise-smooth models vs everywhere-smooth models

- Local variation of potentials $V_{p,q}$ depending on label variation

locally smooth from $l$ to $l'$ when going from $p$ to $p'$

locally steep from $l$ to $l'$ when going from $p$ to $p'$

piecewise-smooth potential
Piecewise-smooth potentials vs everywhere-smooth potentials

- General graph construction for any convex $V_{p,q}$ (Ishikawa 2003)
  - convex $\Rightarrow$ large penalty for sharp jump
  - a few small jumps cheaper than one large jump
  - discontinuities smoothed with “ramp” $\Rightarrow$ oversmoothing

In practice, best results with “least convex” function, e.g.,

$$V_{p,q}(l_p, l_q) = \lambda_{p,q} \| l_p - l_q \|$$
Discontinuity-preserving energy

- At boundaries, very different labels for adjacent pixels are OK
- To not overpenalize in $E$ adjacent but very different labels:
  - $V_{p,q}$ non-convex function of $||l_p - l_q||$
  - for instance:
    - $V_{p,q} = \min(K, ||l_p - l_q||^2)$
    - $V_{p,q} = \min(K, ||l_p - l_q||)$
    - $V_{p,q} = u_{p,q} \mathbf{1}(l_p \neq l_q)$ (Potts model)
Difficulty of minimization

- $E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V_{p,q}(f_p, f_q)$ with
  
  - $f : P \to L$
  
  - $V_{p,q}(f_p, f_q)$ non convex

- $\min_f E(f) :$ minimization of non-convex function in large-dimension space (dimension = $|P|$)

  - NP-hard even in simple cases
    - e.g. $V_{pq}(f_p, f_q) = 1(f_p \neq f_q)$ (Potts model) with $|L| > 2$

  - general case: simulated annealing...
Exact binary optimization (reminder)

- **Pb:** find labeling $f^* : \mathcal{P} \to \mathcal{L} = \{0,1\}$ minimizing energy
  \[ E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q) \]

- **Question:**
  - can a globally optimal labeling $f^*$ be found using some graph-cut construction?

- **Answer (Kolmogorov & Zabih 2004):**
  - yes iff $V_{pq}$ is regular
    - $V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(0,1) + V_{pq}(1,0)$
  - otherwise it's NP-hard

- **But what about general energies on binary variables?**
Exact binary optimization

- **Question:**
  - what functions can be minimized using graph cuts?

- **Classes of functions on binary variables:**
  - $F^2: E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{ij}(x_i,x_j)$
  - $F^3: E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{ij}(x_i,x_j) + \sum_{i<j<k} E^{ijk}(x_i,x_j,x_k)$
  - $F^m: E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \ldots + \sum_{u_1<\ldots<u_m} E^{u_1,\ldots,u_m}(x_{u_1},\ldots,x_{u_m})$

- **“Using graph cuts”:** $E$ graph-representable iff

  $\exists$ graph $G = \langle V,E \rangle$ with $V = \{v_1,\ldots,v_n, s, t\}$ such that

  $\forall$ configuration $x=x_1,\ldots,x_n$, $E(x_1,\ldots,x_n) = \text{cost}(\min s-t\text{-cut in which } v_i \in S \text{ if } x_i = 0 \text{ and } v_i \in T \text{ if } x_i = 1) + k \text{ constant } \in \mathbb{R}$

[Kolmogorov & Zabih 2004]
Exact binary optimization

- $E$ regular iff
  - $\mathcal{F}^2$: $\forall i, j \ E_{i,j}(0,0) + E_{i,j}(1,1) \leq E_{i,j}(0,1) + E_{i,j}(1,0)$
  - $\mathcal{F}^m$: for all terms $E_{u_1,\ldots,u_m}$ in $E$, all projections (specializations) of $E_{u_1,\ldots,u_m}$ to a two-variable function (i.e., all variables fixed but two) are regular

- Question:
  - what functions can be minimized using graph cuts?

- Answer (Kolmogorov & Zabih 2004):
  - $\mathcal{F}^2, \mathcal{F}^3$: $E$ graph-representable $\Leftrightarrow$ $E$ regular
  - any binary $E$: $E$ not regular $\Rightarrow$ $E$ not graph-representable

[Kolmogorov & Zabih 2004]
Link with submodularity

- \( g : 2^p \rightarrow \mathbb{R} \) submodular
  - iff \( g(X) + g(Y) \geq g(X \cup Y) + g(X \cap Y) \)
    for any \( X, Y \subset P \)
  - iff \( g(X \cup \{j\}) - g(X) \geq g(X \cup \{i,j\}) - g(X \cup \{j\}) \)
    for any \( X \subset P \) and \( i, j \in P \setminus X \)

- \( g \) submodular \( \iff \) \( E \) regular, with \( E(x) = g(\{ p \in P \mid x_p = 1 \}) \)
  - \( E_{i,j}(0,1) + E_{i,j}(1,0) \geq E_{i,j}(0,0) + E_{i,j}(1,1) \)

- \( \exists \) independent results on submodular functions
  - minimization in polynomial time but slow, best known \( O(n^6) \)
Exact multi-label optimization

- Problem: find labeling $f^* : \mathcal{P} \rightarrow \mathcal{L}$ minimizing energy

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q)$$

- Assumption: $\mathcal{L}$ linearly ordered — w.l.o.g. $\mathcal{L} = \{1,\ldots,k\}$

  (1D only ⇒ not suited, e.g., for 2D motion vector estimation)

- Solution: reduction/encoding to binary label case

  - for $V_{p,q}(l_p, l_q) = \lambda_{p,q} |l_p - l_q|$ (Boykov et al. 1998, Ishikawa & Geiger 1998)
  - for any convex $V_{p,q}$ (Ishikawa 2003)

  - See also

    - MinSum pbs (Schlesinger & Flach 2006)
    - submodular $V_{p,q}$ (Darbon 2009)
Linear multi-label graph construction

Given $\mathcal{L} = \{1, ..., k\}$

- General idea:
  - construct one layer per label value
  - read label value from cut location

(e.g., $k = 4$

\[\text{cut: } f_p = 3, f_q = 1\]
Linear multi-label graph construction

\[ E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} \lambda_{p,q} |f_p - f_q| \]

with \( f_p \in \mathcal{L} = \{1,\ldots,k\} \)

**Attempt 1:**

- For each site \( p \)
  - create nodes \( p_1, \ldots, p_{k-1} \)
  - create edges \( t_1^p = (s, p_1), t_j^p = (p_{j-1}, p_j), t_k^p = (p_{k-1}, t) \)
  - assign weights \( w_j^p = w(t_j^p) = D_p(j) \)

- For each pair of neighboring sites \( p \) and \( q \)
  - create edges \( (p_j, q_j), j \in \{1, \ldots, k-1\} \) with weight \( \lambda_{p,q} \)

**Read label value from cut location,** e.g., \( p_2 \in S, p_3 \in T \Rightarrow f_p = 3 \)
Linear multi-label graph construction

(cf. Boykov et al. 1998)

- Given $\mathcal{L} = \{1, \ldots, k\}$
- General idea:
  - construct one layer per label value
  - read label value from cut location
- Any problem?

\[
\begin{align*}
\text{cut: } f_p = 3, f_q = 1 \\
e.g., k = 4
\end{align*}
\]
Linear multi-label graph construction

- Given $\mathcal{L} = \{1, \ldots, k\}$
- General idea:
  - construct one layer per label value
  - read label value from cut location
- Any problem?
  - there could be several cut locations on the same line

(cf. Boykov et al. 1998)
Linear multi-label graph construction

(cf. Boykov et al. 1998)

\[ E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} \lambda_{p,q} |f_p - f_q| \]

with \( f_p \in \mathcal{L} = \{1, \ldots, k\} \)

Attempt 2:

- For each site \( p \)
  - create nodes \( p_1, \ldots, p_{k-1} \)
  - create edges \( t_1^p = (s, p_1), t_j^p = (p_{j-1}, p_j), t_k^p = (p_{k-1}, t) \)
  - assign weights \( w_j^p = w(t_j^p) = D_p(j) + K_p \) [penalize more cutting \( t_j^p \)]
    with \( K_p = 1 + (k-1) \sum_{q \in \mathcal{N}_p} \lambda_{p,q} \) (where \( \mathcal{N}_p \) set of neighbors of \( p \))

- For each pair of neighboring sites \( p \) and \( q \)
  - create edges \( (p_j, q_j), j \in \{1, \ldots, k-1\} \) with weight \( \lambda_{p,q} \)
Linear multi-label graph properties

(cf. Boykov et al. 1998)

- **Lemma**: for each site $p$, a minimum cut severs exactly one $t^p_j$
  - $[\geq 1]$ Any cut severs at least one $t^p_j$
  - $[\leq 1]$ Suppose $t^p_a$, $t^p_b$ are cut (same line $p$) then build new cut with $t^p_b$ restored and links $(p_j, q_j)_{j \in \{1, \ldots, k-1\}}$ broken for $q \in \mathcal{N}_p$

Impact on (minimum) cost: 

\[-w(t^p_b) + (k-1) \sum_{q \in \mathcal{N}_p} \lambda_{p,q} = -D_p(j) - 1 < 0 \rightarrow \text{strictly smaller cost} \rightarrow \text{contradiction}\]

- **Theorem** (Boykov et al. 1998): a minimum cut minimizes $E(f)$
Application to stereovision: disparity map estimation

- Problem
  - given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u, v)$

- Graph-cut setting
  - discrete disparities: $d_p \in \mathcal{L} = \{d_{\text{min}}, ..., d_{\text{max}}\}$
  - data term: $D_p(d_p)$
    - small when pixel $p$ in $I$ similar to pixel $p' = p + (d_p, 0)$ in $I'$
  - smoothness term: $V_{p,q}(d_p, d_q)$
    - small when disparities $d_p$ and $d_q$ are similar
Application to stereovision: disparity map estimation

• Problem
  – given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u,v)$

• Graph-cut setting
  – discrete disparities: $d_p \in \mathcal{L} = \{d_{\text{min}},...,d_{\text{max}}\}$
  – data term: $D_p(d_p)$
    - small when pixel $p$ in $I$ similar to pixel $p' = p + (d_p,0)$ in $I'$
  – smoothness term: $V_{p,q}(d_p, d_q)$
    - small when disparities $d_p$ and $d_q$ are similar
Application to stereovision: disparity map estimation

- **Problem**
  - given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u,v)$

- **Graph-cut setting**
  - discrete disparities: $d_p \in \mathcal{L} = \{d_{\text{min}},...,d_{\text{max}}\}$
  - data term: $D_p(d_p)$
    - e.g., $D_p(d_p) = E_{\text{ZNSSD}}(P_p ; (d_p,0))$ where $P_p$ patch around pixel $p$
  - smoothness term: $V_{p,q}(d_p, d_q)$
    - e.g., $V_{p,q}(d_p, d_q) = \lambda |d_p - d_q|$ [Boykov et al. → optimal disparities]

$$
\begin{align*}
\tilde{I}_p &= \frac{1}{|P|} \sum_{q \in P} I_q \\
\sigma &= \left[ \frac{1}{|P|} \sum_{q \in P} (I_q - \tilde{I}_p)^2 \right]^{1/2} \\
E_{\text{ZNSSD}}(P ; u) &= \frac{1}{|P|} \sum_{q \in P} \left( \frac{(I_q + u - \tilde{I}_p) / \sigma - (I_q - \tilde{I}_p) / \sigma}{\sigma} \right)^2
\end{align*}
$$
Application to stereo vision: disparity map estimation

- **Problem**
  - given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u,v)$

- **Graph-cut setting**
  - discrete disparities: $d_p \in \mathcal{L} = \{d_{\text{min}},...,d_{\text{max}}\}$
  - data term: $D_p(d_p)$
    - e.g., $D_p(d_p) = E_{\text{ZNSSD}}(P_p ; (d_p,0))$
  - smoothness term: $V_{p,q}(d_p, d_q)$
    - e.g., $V_{p,q}(d_p, d_q) = \lambda |d_p - d_q|$ [Boykov et al. → optimal disparities]

Is it the “optimal” solution to the disparity map estimation problem?
Application to stereovision: disparity map estimation

• Problem
  - given 2 rectified images \( I, I' \), estimate optimal disparity 
    \( d(p) = d_p \) for each pixel \( p = (u,v) \)

• Graph-cut setting
  - discrete disparities: \( d_p \in \mathcal{L} = \{d_{\text{min}},...,d_{\text{max}}\} \)
  - data term: \( D_p (d_p) \)
    - e.g., \( D_p (d_p) = E_{\text{ZNSSD}}(P_p ; (d_p,0)) \)
  - smoothness term: \( V_{p,q} (d_p, d_q) \)
    - e.g., \( V_{p,q} (d_p, d_q) = \lambda \left| d_p - d_q \right| \)

- meaningful but arbitrary choices: patch size, similarity, smoothness...
- optimal solution for energy \( \neq \) optimal solution for problem

[Boykov et al. \( \rightarrow \) optimal disparities]
Application to stereovision: disparity map estimation

- **Problem**
  - given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u,v)$

- **Graph-cut setting (alternative)**
  - discrete disparities: $d_p \in \mathcal{L} = \{d_{\text{min}},...,d_{\text{max}}\}$
  - $D_p(d_p) = \omega_{\text{cc}} \rho(E_{\text{ZNCC}}(P; (d_p,0)))$ with $\rho(c) \in [0,1]$ \(\nabla\)
    - e.g. $\rho(c) = \begin{cases} 1 & \text{if } c < 0 \\ \sqrt{1-c} & \text{if } c \geq 0 \end{cases}$
  - $V_{p,q}(d_p, d_q) = \lambda |d_p - d_q|$  

N.B. only $\omega_{\text{cc}} / \lambda$ matters
Approximate optimization

- Exact multi-label optimization:
  - only limited cases
  - in practice, may require large number of nodes
- How to go beyond exact optimization constraints?

Iterate exact optimizations on subproblems (Boykov et al. 2001)
  - local minimum 😞
  - but within known bounds of global minimum 😊
Notion of move — Examples

Move: maps a labeling $f : P \rightarrow L$ to a labeling $f' : P \rightarrow L$

Idea: iteratively apply moves to get closer to optimum $f^*$

(a) initial labeling
(b) standard move
\[ \alpha \rightarrow \beta \]
at one site only

(c) $\alpha$-$\beta$-swap
\[ \alpha \leftrightarrow \beta \]
at many sites at once

(d) $\alpha$-expansion
\[ \text{any } l \rightarrow \alpha \]
at many sites at once

Boykov et al. 2001 © IEEE
Moves

Given a labeling $f : P \rightarrow L$ and labels $\alpha, \beta$

- $f'$ is a **standard move** from $f$ iff $f$ and $f'$ differ at most on one site $p$
- $f'$ is an **expansion move** (or $\alpha$-expansion) from $f$ iff
  \[ \forall p \in P, \quad f'_p = f_p \text{ or } \alpha \]
  \[ \rightarrow \quad \text{in } f', \text{ compared to } f, \text{ extra sites } p \text{ can now be labeled } \alpha \]
- $f'$ is a **swap move** (or $\alpha$-$\beta$-swap) from $f$ iff
  \[ \forall p \in P, \quad f_p \neq \alpha, \beta \Rightarrow f'_p = f'_p \]
  \[ \rightarrow \quad \text{some sites that were labeled } \alpha \text{ are now } \beta \text{ and vice versa} \]

N.B. Other kinds of moves can be defined...
Optimization w.r.t. moves

- Iterative optimization over moves
  - random cycle over all labels until convergence $\rightarrow$ local min

- Iterating standard moves
  = usual discrete optimization method
  - iterated conditional modes (ICM) = iterative maximization of the probability of each variable conditioned on the rest
    ▪ local minimum w.r.t. standard move, i.e., energy cannot decrease with a single pixel label difference
      $\Rightarrow$ weak condition, low quality
  - simulated annealing, ...
    ▪ slow convergence (optimal properties “at infinity”), modest quality, some sampling strategies but mostly random
Optimization w.r.t. moves

• Iterative optimization over moves
  - random cycle over all labels until convergence $\rightarrow$ local min

• Iterating expansion/swap moves (**strong** moves)
  - number of possible moves exponential in number of sites
  - compute optimal move using graph cut = binary problem!
    - see Boykov et. al 2001 for graph construction and details
  - significantly fewer local minima than with standard moves
  - sometimes within constant factor of global minimum
    - e.g., expansion moves & Potts model $\rightarrow$ optimum within factor 2
Image restoration with moves

- Restoration with standard moves vs $\alpha$-expansions

original image

restoration with standard moves

noisy image

restoration with $\alpha$-expansions
Constraints on interaction potential

(see details in Boykov et. al 2001)

- **Expansion move**: $V$ metric, $\rightarrow$ expansion inequality:

  $$V_{p,q}(\alpha,\alpha) + V_{p,q}(\beta,\gamma) \leq V_{p,q}(\alpha,\gamma) + V_{p,q}(\beta,\alpha) \quad \text{for all } \alpha,\beta,\gamma \in \mathcal{L}$$

- **Swap move**: $V$ semi-metric, $\rightarrow$ swap inequality:

  $$V_{p,q}(\alpha,\alpha) + V_{p,q}(\beta,\beta) \leq V_{p,q}(\alpha,\beta) + V_{p,q}(\beta,\alpha) \quad \text{for all } \alpha,\beta \in \mathcal{L}$$

  [= as metric but triangle inequality not required: $V_{p,q}(\alpha,\gamma) \neq V_{p,q}(\alpha,\beta) + V_{p,q}(\beta,\gamma)$]

  [weaker condition than for expansion move]

- **Examples**

  - Potts model: $V_{p,q}(\alpha,\beta) = \lambda_{p,q} \mathbf{1}(\alpha \neq \beta)$

  - cap max value: $V_{p,q}(\alpha,\beta) = \min(K,||\alpha - \beta||)$ or $\min(K,||\alpha - \beta||^2)$

  ![discontinuity-preserving!](discontinuity-preserving)
Disparity map estimation with moves

Tsukuba images from famous Middlebury benchmark (also contains Moebius images)
Disparity map estimation: data term

- Idea: direct intensity comparison, but sensitive to sampling
  \[ D_p(d_p) = \min(K, |I_p - I'_{p+d_p}|^2) \]

- With image sampling insensitivity:
  - disparity range discretized to 1 pixel accuracy
    \[ \rightarrow \text{sensitivity to high gradients} \]
  - (sub)pixel dissimilarity measure for greater accuracy, e.g., by linear interpolation (Birchfield & Tomasi 1998)
    \[ C_{fwd}(p,d) = \min_{d-1/2 \leq u \leq d+1/2} |I_p - I'_{p+u}| \]
    \[ C_{rev}(p,d) = \min_{p-1/2 \leq x \leq p+1/2} |I_x - I'_{p+d}| \]
    \[ \text{[for symmetry]} \]

- \[ D_p(d_p) = C(p,d_p) = \min(K, C_{fwd}(p,d_p), C_{rev}(p,d_p))^2 \]

(cf. Boykov et al. 1999, Boykov et al. 2001)

No patch similarity here: the local consistency is given by the smoothness term.
Disparity map estimation: smoothness term

- Scene with fronto-parallel objects
  - use **piecewise-constant** model
  - e.g., Potts model:
    \[ V_{p,q}(d_p, d_q) = u_{p,q} \ 1(d_p \neq d_q) \]

- Scene with slanted surfaces (e.g., ground)
  - use **piecewise-smooth** model
  - e.g., smooth cap max value:
    \[ V_{p,q} = \lambda \min(K, |d_p - d_q|) \]

- Metric ⇒ both swap and expansion algorithms usable
Potts model vs smooth cap max value

- **Potts model**: piecewise-constant
  - suited for uniform areas ($\Rightarrow$ fewer disparities on large areas)

- **Smooth cap max value**: piecewise-smooth model
  - suited for slowly varying areas (e.g., slope)

(a) Left image: 256x233, 29 labels  
(b) Piecewise constant model  
(c) Piecewise smooth model
Disparity map estimation: smoothness term

- Contextual information
  - neighbors $p, q$ more likely to have same disparity if $I_p \approx I_q$
    $\rightarrow$ make $V_{p,q}(d_p, d_q)$ also depend on $|I_p - I_q|$
  - meaningful in low texture areas (where $|I_p - I_q|$ meaningful)

- E.g., with Potts model: $V_{p,q}(d_p, d_q) = u_{p,q} \mathbf{1}(d_p \neq d_q)$
  - $u_{p,q}$: penalty for assigning different disparities to $p$ and $q$
  - textured regions: $u_{p,q} = K$
  - textureless regions: $u_{p,q} = U(|I_p - I_q|)$
    - $u_{p,q}$ smaller for pixels $p, q$ with large intensity difference $|I_p - I_q|$
    - e.g., $U(|I_p - I_q|) = \begin{cases} 2K & \text{if } |I_p - I_q| \leq 5 \\ K & \text{if } |I_p - I_q| > 5 \end{cases}$

(cf. Boykov et al. 1998, Boykov et al. 2001)
Disparity map estimation

- **Problem**
  - given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u,v)$

- **Are the preceding formulations OK?**
  - anything not modeled?
  - any bias?
Disparity map estimation

(Boykov et. al 2001)

- Problem
  - given 2 rectified images $I, I'$, estimate optimal disparity $d(p) = d_p$ for each pixel $p = (u,v)$

- Are the preceding formulations OK?
  - no treatment of occlusion
  - no symmetry: one image at the center
    - treatment of second image relative to the first (main) one
    - difficulty to incorporate occlusion naturally
Cross-checking

(Bolles & Woodfill, 1993)

• Problem
  - given 2 rectified images \( I, I' \), estimate optimal disparity
    \( d(p) = d_p \) for each pixel \( p = (u,v) \)

• Cross-checking method:
  - compute left-to-right disparity
  - compute right-to-left disparity
  - mark as occlusion pixels in one image mapping to pixels in the other image but which do not map back to them

• Common and easy to implement
Stereovision with occlusion handling

(c.f. Kolmogorov & Zabih 2001)

- **Occlusion**
  - pixel visible in one image only
  - occurs usually at discontinuities

- **Uniqueness model hypothesis**
  - pixel in one image $\rightarrow$ at most one pixel in other image
  [sometimes too restrictive]
  - pixel with no correspondence: labeled as occluded

- **Main idea:**
  - use labels representing corresponding pixels (= pixel pairs), not pixel disparity
Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

- $\mathcal{A}$: correspondence candidates (pixel pairs in $I \times I'$) = pixel assignments
  - $\mathcal{A} = \{ (p,p') \mid p_y = p'_y \text{ and } 0 \leq p'_x - p_x < k \}$ (same line, different position)
  - disparity: for $a = (p,p') \in \mathcal{A}$, $d(a) = p'_x - p_x$
  - hypothesis: disparities lie in limited range $[0,k]$
  - goal: find subset of $\mathcal{A}$ containing only corresponding pixels
  - use: subsets defined as labelings $f : \mathcal{A} \rightarrow \mathcal{L} = \{0,1\}$ such that
    $\forall a = (p,p') \in \mathcal{A}$, $f_a = 1$ if $p$ and $p'$ correspond, otherwise $f_a = 0$
  - symmetric treatment of images (& applicable to non aligned cameras)

- $A(f)$: active assignments, i.e., pixel pairs considered as corresponding
  - $A(f) = \{ a \in \mathcal{A} \mid f_a = 1 \}$
Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

- $N_p(f)$: set of correspondences for pixel $p$
  - $N_p(f) = \{ a \in A(f) \mid \exists p' \in P, a = (p,p') \}$
  - configuration $f$ unique iff $\forall p \in P \ |N_p(f)| \leq 1$
  - occluded pixels defined as pixels such that $|N_p(f)| = 0$

- $\mathcal{N}$: a neighborhood system on assignments (used for smoothness term)
  - $\mathcal{N} \subset \{ \{ a_1, a_2 \} \subset A \}$
  - for efficient energy minimization via graph cuts:
    - neighbors having the same disparity
    - $\mathcal{N} = \{ \{(p,p'),(q,q')\} \subset A \mid p,p' \text{ are neighbors and } d(p,p') = d(q,q') \}$
      ($\Rightarrow$ then $q,q'$ are also neighbors)
Stereovision with occlusion

\[ E(f) = E_{\text{data}}(f) + E_{\text{smooth}}(f) + E_{\text{occ}}(f) \]

- \( E_{\text{data}}(f) = \sum_{a=(p,p')\in A(f)} (I_p - I'_p)^2 \)
  - single pixel similarity

- \( E_{\text{smooth}}(f) = \sum_{\{a_1,a_2\} \in \mathcal{N}} V_{a_1,a_2} \mathbf{1}(f_{a_1} \neq f_{a_2}) \)
  - \( \mathcal{N} = \{(p,p'),(q,q')\} \subset A \) | \( p,p' \) are neighbors and \( d(p,p')=d(q,q') \)
  - penalty if \( f_{a_1} = 1, \ a_2 \) close to \( a_1, \ d(a_2)=d(a_1), \) but \( f_{a_2} = 0 \)
  - Potts model on assignments (pixel pairs), not on pixel disparity

- \( E_{\text{occ}}(f) = \sum_{p\in P} C_p \cdot \mathbf{1}(|N_p(f)| = 0) \)  
  - penalty \( C_p \) if \( p \) occluded
Stereovision with occlusion

( cf. Kolmogorov & Zabih 2001)

- $E(f) = E_{\text{data}}(f) + E_{\text{smooth}}(f) + E_{\text{occ}}(f)$

- Optimizable by graph cuts as multi-label problem (cf. paper)
  - graph construction on assignments (pixel pairs), not pixels
    - $A^\alpha$: set of all assignments with disparity $\alpha$
    - $A^{\alpha,\beta} = A^\alpha \cup A^\beta$
  - expansion move:
    - $f'$ within single $\alpha$-expansion move of $f$ iff $A(f') \subset A(f) \cup A^\alpha$
      - currently active assignments can be deleted
      - new assignments with disparity $\alpha$ can be added
  - swap move:
    - $f'$ within single swap move of $f$ iff $A(f') \cup A^{\alpha,\beta} = A(f) \cup A^{\alpha,\beta}$
      - only changes: adding or deleting assignments having disparities $\alpha$ or $\beta$
Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

Expansion-move algorithm:

1. start with arbitrary, unique configuration $f_0$

2. set success $\leftarrow$ false

3. for each disparity $\alpha$
   
   3.1. find $f^{\alpha} = \arg\min_f E(f)$
       subject to $f$ unique, within single $\alpha$-move of $f_0$

   3.2. if $E(f^{\alpha}) < E(f_0)$, then set $f_0 \leftarrow f^{\alpha}$, success $\leftarrow$ true

4. if success go to 2

5. return $f_0$

- Critical step: efficient computation of $\alpha$-move with smallest energy
Stereovision with occlusion

- **Swap-move algorithm:**

  1. start with arbitrary, unique configuration $f_0$
  2. set success $\leftarrow$ false
  3. for each pair of disparities $\alpha, \beta$ ($\alpha \neq \beta$)

    3.1. find $f^{\alpha\beta} = \arg\min_f E(f)$
        subject to $f$ unique, within single $\alpha\beta$-swap of $f_0$

    3.2. if $E(f^{\alpha\beta}) < E(f_0)$, then set $f_0 \leftarrow f^{\alpha\beta}$, success $\leftarrow$ true

  4. if success go to 2
  5. return $f_0$

- Critical step: efficient computation of $\alpha\beta$-swap with smallest energy (cf. Kolmogorov & Zabih 2001)
Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

(a) Left image of Head pair
(b) Potts model stereo
(c) Stereo with occlusions
   Disparity maps obtained for the Head pair

(d) Left image of Tree pair
(e) Potts model stereo
(f) Stereo with occlusions
   Disparity maps obtained for the Tree pair
Stereovision with occlusion

(c.f. Kolmogorov & Zabih 2001)

- Expansion moves vs swap moves

- Swap moves not powerful enough to escape local minima for this class of energy function
Multi-view reconstruction


- Given \( n \) calibrated images on the “same side” of scene
- Global model
  - \( \mathcal{L} = \) discretized set of **depths** (not disparities)
  - image \( i \), pixel \( p \), depth \( l \)
- Difficulty = point interaction
  - pb: def \( (i,p,l), (j,q,l) \) “close” in 3D
    \( \rightarrow \) too many interactions \( \rightarrow \) ☹
  - sol: def \( q \) closest pixel of projection of \( (i,p,l) \) on \( j \) \( \rightarrow \) ☺
- Photo-consistency constraints (visibility)
  - red point, at depth \( l=2 \), blocks C2's view of green point, at depth \( l=3 \)
Multi-view reconstruction


- Terms in the energy: data, smoothness, visibility
- Optimization by $\alpha$-expansion

See paper for details
Beyond disparity maps: 3D mesh reconstruction

- Merging of depth maps into single point cloud
  - possibly sparse depth maps, e.g., obtained by plane sweep

- Problems:
  - outliers
  - multi-view visibility (not taken into account globally)

- Solution:
  - Delaunay tetrahedralization of point cloud
  - binary labelling: inside or outside
  - 3D surface = interface inside/outside

(cf. Vu et al. 2012)
Visibility consistency with graph cut

- Lines of sight from cameras to visible points ⇒ outside

\[ D_{\text{out}}(l_T) = \alpha_{\text{vis}} [l_T = 1] \]
\[ D_{\text{in}}(l_T) = \alpha_{\text{vis}} [l_T = 0] \]
\[ V_{\text{align}}(l_{T_i}, l_{T_j}) = \alpha_{\text{vis}} [l_{T_i} = 0 \land l_{T_j} = 1] \]

\[ E_{\text{vis}}(S, P, v) = \sum_{P \in P} \left( \sum_{Q \in v_P} D_{\text{out}}(l_{T_{iQ}}) + \sum_{i=1}^{N_{[PQ]}-1} V_{\text{align}}(l_{T_{iQ}^P}, l_{T_{i+1Q}^P}) + D_{\text{in}}(l_{T_{N_{[PQ]}P}}) \right) \]
Beyond disparity maps: 3D mesh reconstruction

- Best reconstruction results on international benchmarks
- Startup company with IMAGINE members (2011)
  - 15 employees, 90% revenue = international
  - bought by Bentley Systems (2015)
Some PhD theses in IMAGINE next year:
Des selfies d’oreilles pour du son 3D

- Son stéréo : perception sur une simple ligne gauche/droite
- Son 3D : perception de provenance de n’importe quelle direction 3D
- HRTF : filtre directionnel de fréquences propre à chaque oreille/tête

• Objectif : personnaliser l’écoute 3D au casque

• Approche :
  - selfie d’oreille avec smartphone
  - reconstruction 3D partielle d’oreille (difficile, manque de texture)
  - reconnaissance d’un modèle d’oreille paramétré
  - calcul des HRTF : direct ou par deep learning
  - personnalisation d’un casque binaural => cinéma 5.1, gaming...
Some PhD theses in IMAGINE next year: Reconstruction de maquettes 3D bâtiments

• Besoins :
  - rénovation, performance énergétique...
  - créer des modèles de bâtiments existants :
    - connaître la géométrie : surfaces, volumes
    - connaître la sémantique : murs, portes, fenêtres..., reconnaissance d’objets (mobilier)
  - entreprises intéressées : Bouygues Construction, Autodesk...

• Outils :
  - 3D, photogrammétrie, géométrie algorithmique (1 scan = millions de pts), machine learning drones en autonomie...
Exercise 2: simple disparity map estimation (without moves nor occlusion)

- Given 2 rectified images $I, I'$, estimate optimal disparity $d(p)$ for pixels $p = (u,v)$

- Setting: linear multi-label graph construction (cf. pp. 82-93)
  - discrete disparities: $d_p \in L = \{d_{\text{min}}, ... , d_{\text{max}}\}$
  - $\mathcal{N}_p$: 4 neighbors of pixel $p$
  - $D_p(d_p) = w_{cc} \rho(E_{\text{ZNCC}}(P; (d_p, 0)))$ with $\rho(c) = \begin{cases} 1 & \text{if } c < 0 \\ \sqrt{1-c} & \text{if } c \geq 0 \end{cases}$
  - $V_{p,q}(d_p, d_q) = \lambda |d_p - d_q|$
Bibliography


Bibliography

- Schlesinger, D., and Flach, B., Transforming an arbitrary MinSum problem into a binary one, Technical report TUD-FI06-01, Dresden University of Technology, April 2006.