MVA/IMA – 3D Vision

Graph Cuts and Application to Disparity Map Estimation

Renaud Marlet
LIGM/École des Ponts ParisTech (IMAGINE) & Valeo.ai
Pascal Monasse
LIGM/École des Ponts ParisTech (IMAGINE)

(with many borrowings from Boykov & Veksler 2006)

Introduction

3D reconstruction

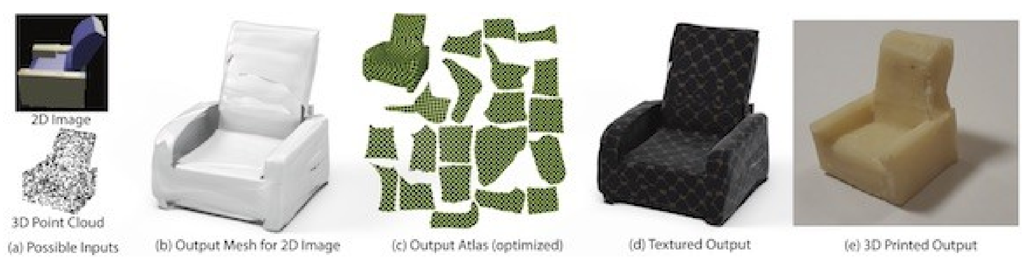
- capturing reality
  - for diagnosis, simulation, movies, video games, interaction in virtual/augmented reality, ...

This course:

- camera calibration
  - relevance of accuracy: 1° error, at 10m → 17cm error
- low-level 3D (disparity/depth map, mesh)
  - as opposed to high-level geometric primitives, semantics...

Mathematical tools for 3D reconstruction

- Deep learning:
  - very good for matching image regions
    → subcomponent of 3D reconstruction algorithm
  - a few methods for <u>direct</u> disparity/depth map estimation
  - fair results on 3D reconstruction from <u>single view</u>



Groueix et al. 2017

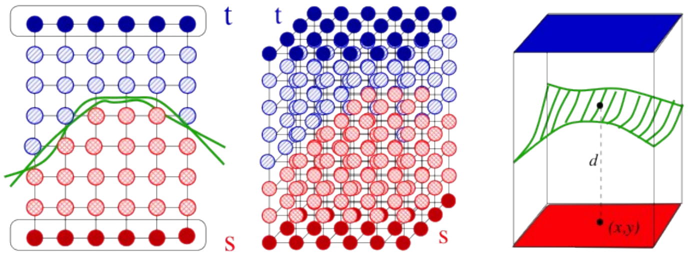
- Graph cut
  - practical, well-founded, general (→ maps, meshes…)

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

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

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



Many links to other domains

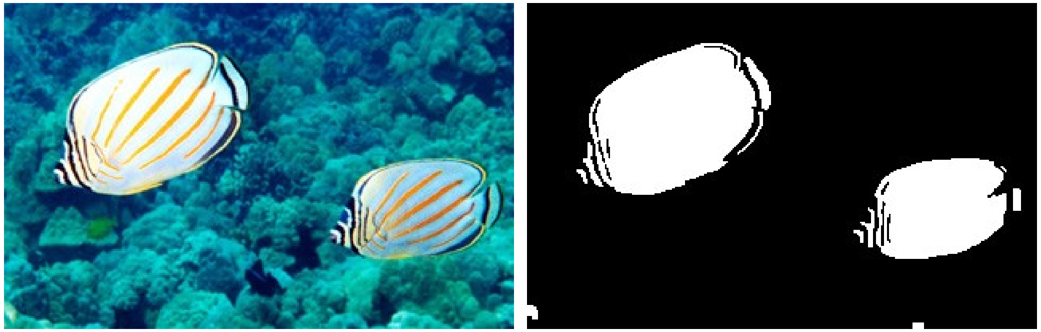
(cf. Boykov & Veksler 2006)

- 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.

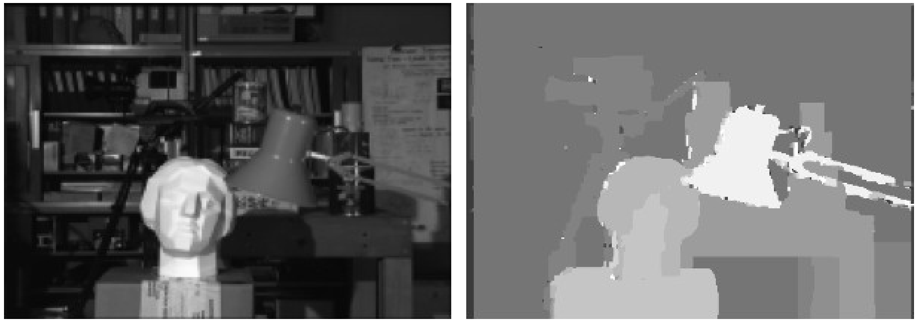
dynamic programming = programmation dynamique
simulated annealing = recuit simulé
Markov random field = champ (aléatoire) de Markov
random walk = marche aléatoire
Bayesian network = réseaux bayésien
level set = ligne de niveau
submodular function = fonction sous-modulaire

Overview of the course

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



segmentation

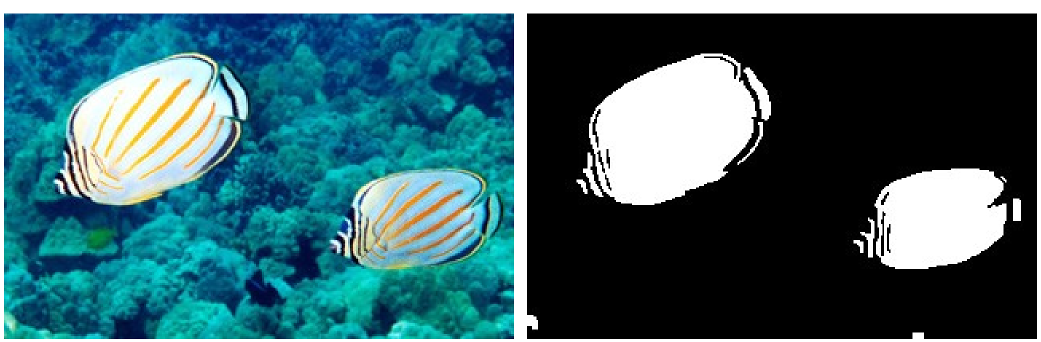


disparity map estimation

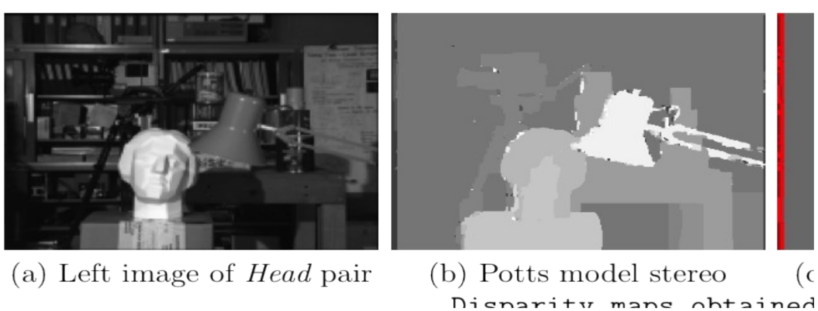
Overview of the course

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

No time to go deep into every topic → general ideas, read the references



segmentation



disparity map estimation

Part 1

Graph cuts basics

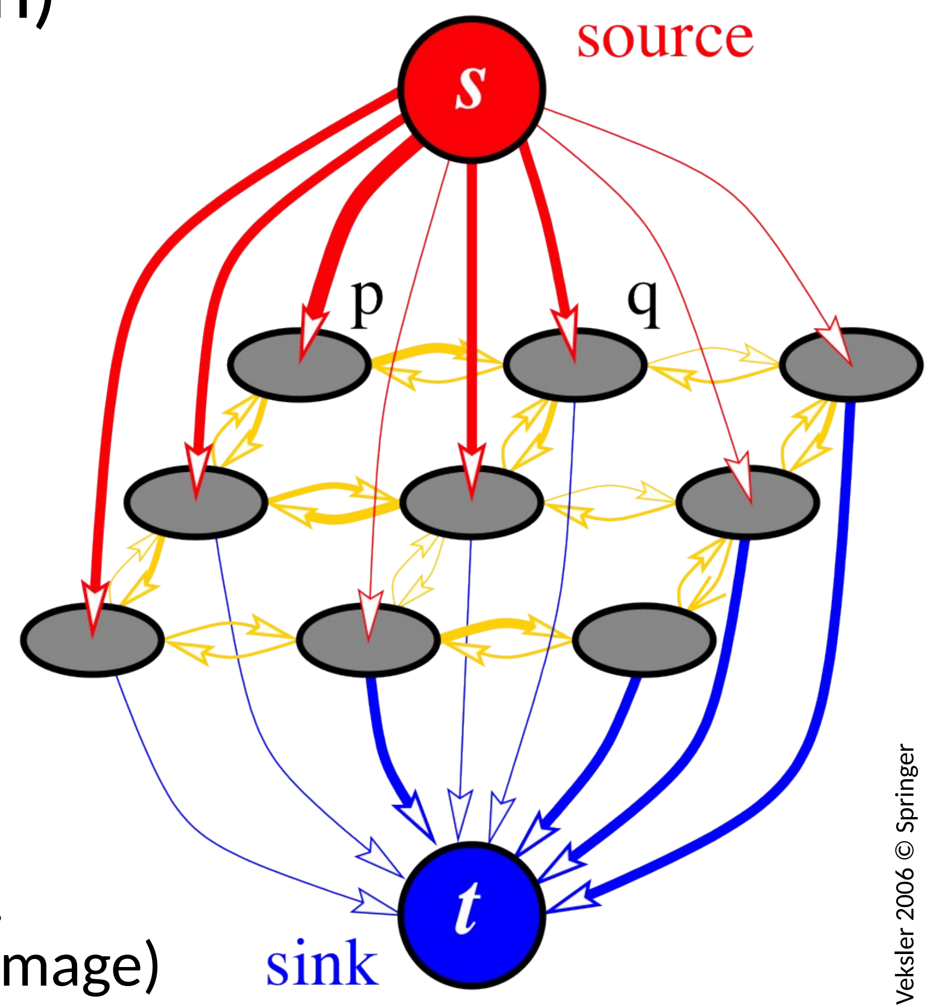
Max-flow min-cut theorem

Application to image restoration
and image segmentation

node = nœud
vertex (vertices) = sommet(s)
edge = arête
directed = orienté
digraph (directed graph) = graphe orienté
sink = puits

Graph cut basics

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



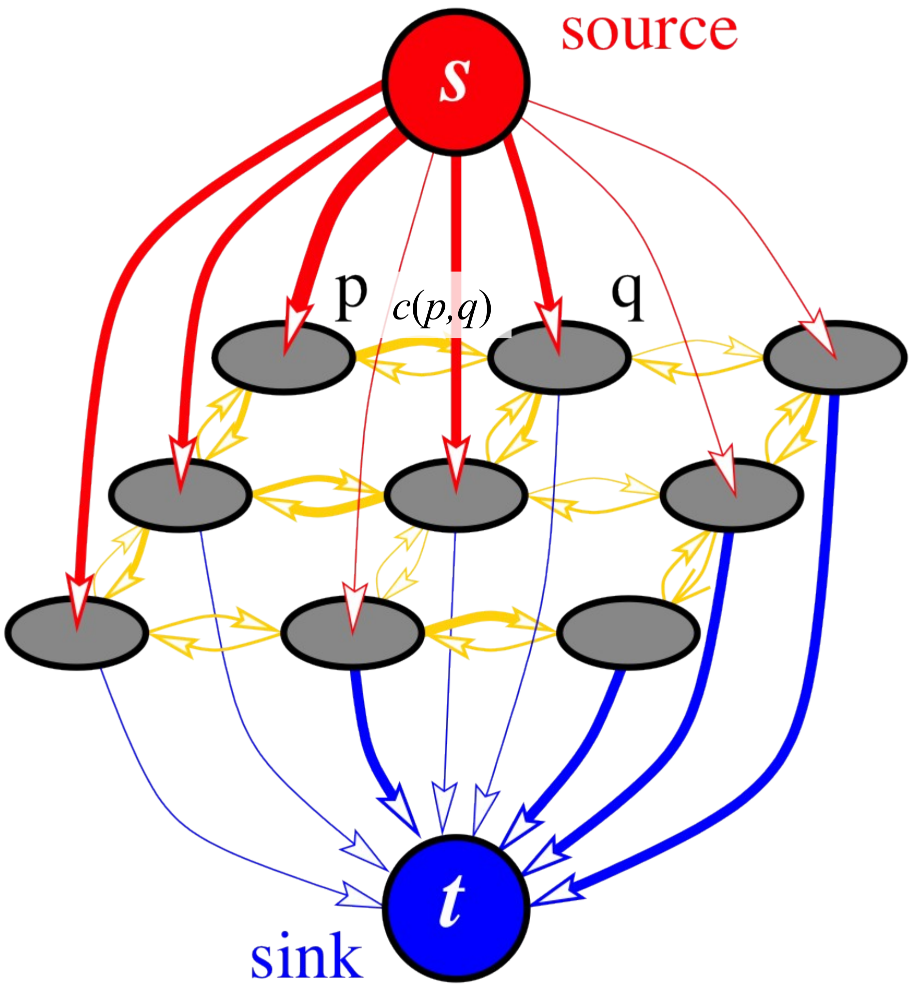
Example of connectivity

Graph cut basics

label = étiquette
weight = poids
link = lien

- Edge labels, for $p \to q \in$ 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.
    - $\{s \to p \mid p \neq t\}, \{q \to t \mid q \neq s\}$
  - n-link: non-term. $\to$ non-term.
    - N $= \{p \to q \mid p,q \neq s,t\}$



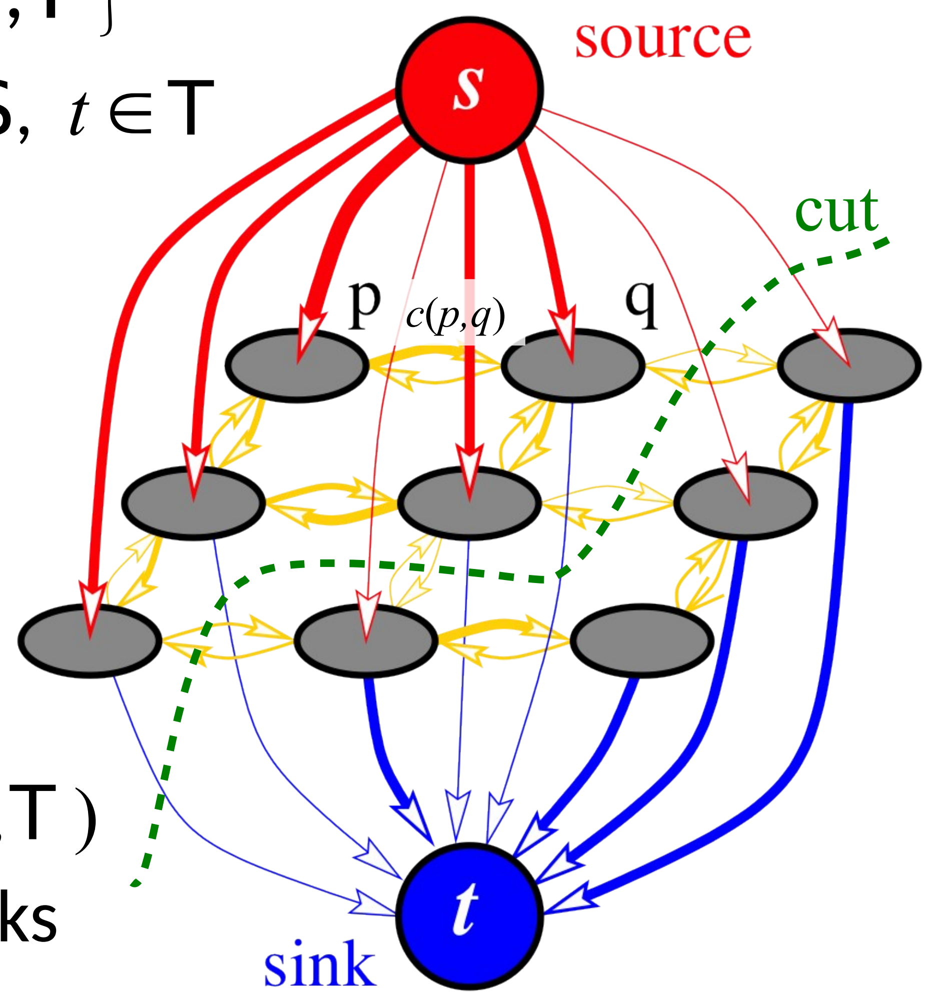
Cut and minimum cut

cut = coupe
severed = coupé, sectionné

- ***s-t* cut** (or just "cut"): $C = \{\mathrm{S},\mathrm{T}\}$

  node partition such that $s \in \mathrm{S},\ t \in \mathrm{T}$

- **Cost of a cut $\{\mathrm{S},\mathrm{T}\}$:**
  - $c(\mathrm{S},\mathrm{T}) = \sum_{p\in \mathrm{S},\ q\in \mathrm{T}} c(p,q)$
  - N.B. cost of severed edges: **only from *S* to *T***
- **Minimum cut:**
  - i.e., with min cost: $\min_{\mathrm{S},\mathrm{T}}\ c(\mathrm{S},\mathrm{T})$
  - intuition: cuts only "weak" links



Boykov & Veksler 2006 © Springer

Different view: flow network

(or transportation network)

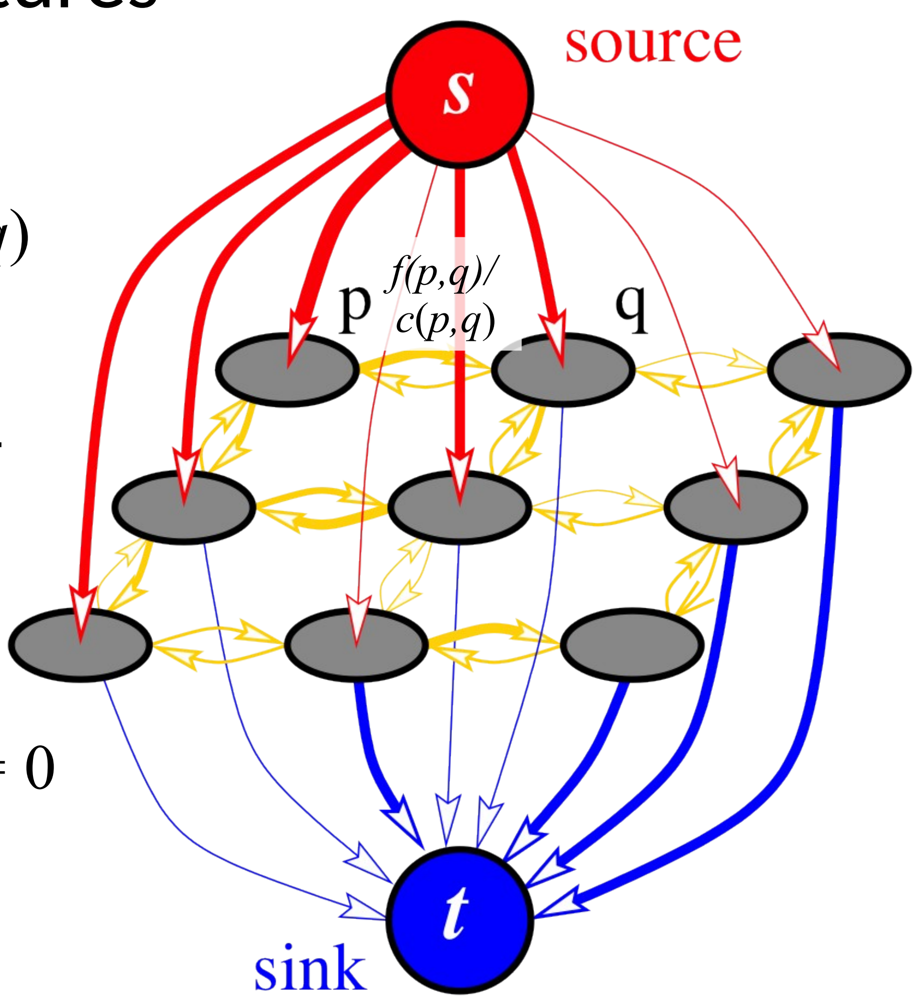
flow = flot
network = réseau
transportation = transport
vertex = sommet
node = nœud
edge = arête

- 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 source<u>s</u> & sink<u>s</u>
- Flow $f : V \times V \rightarrow \mathbb{R}$
  - $f(p,q)$: amount of flow $p \rightarrow q$
    - $(p,q) \notin E \Leftrightarrow c(p,q) = 0, f(p,q) = 0$
  - e.g. road traffic, fluid in pipes, current in electrical circuit, ...

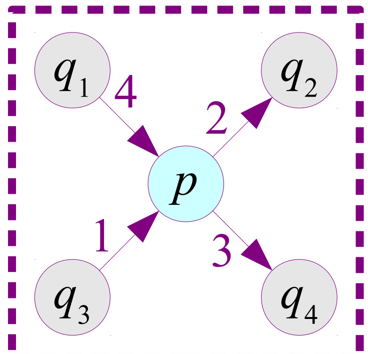


Boykov & Veksler 2006 © Springer

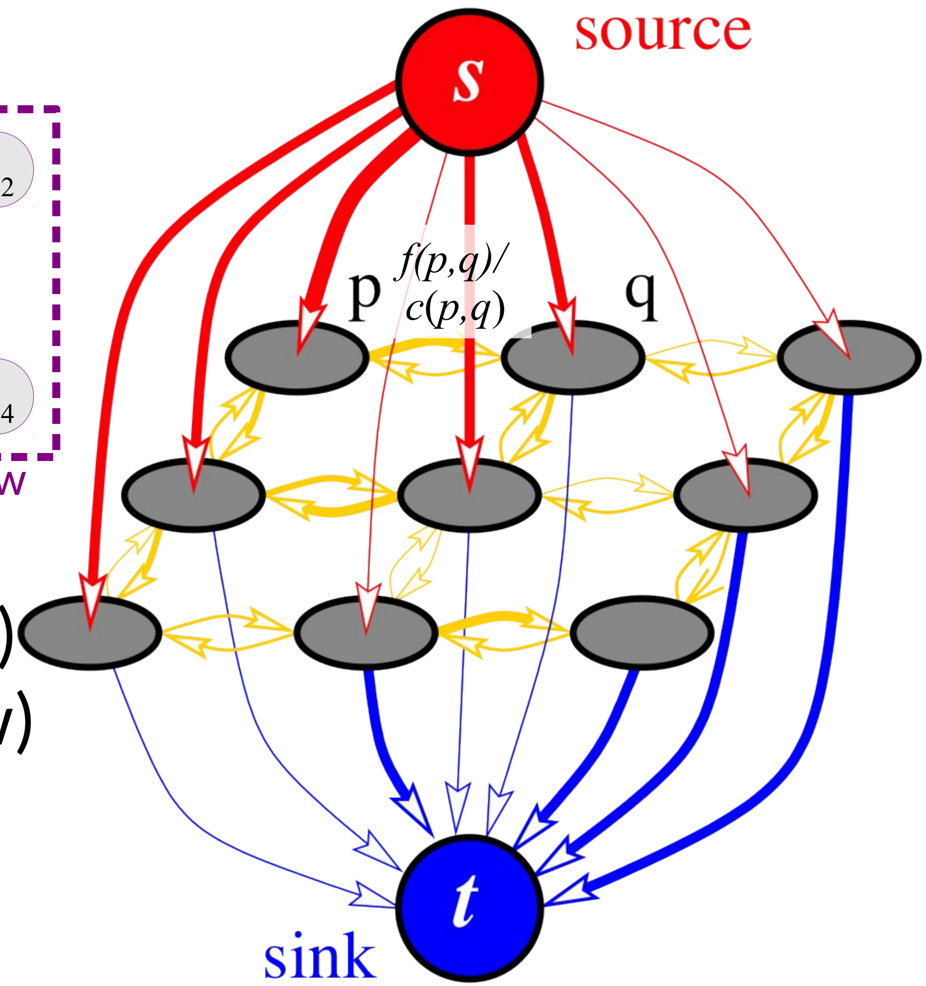
Flow network constraints

skew symmetry = antisymétrie

- Capacity constraint
  - $f(p,q) \leq c(p,q)$
- Skew symmetry
  - $f(p,q) = -f(q,p)$
- Flow conservation
  - $\forall p$, 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 $\sum_{(p,q) \in E} f(p,q)$



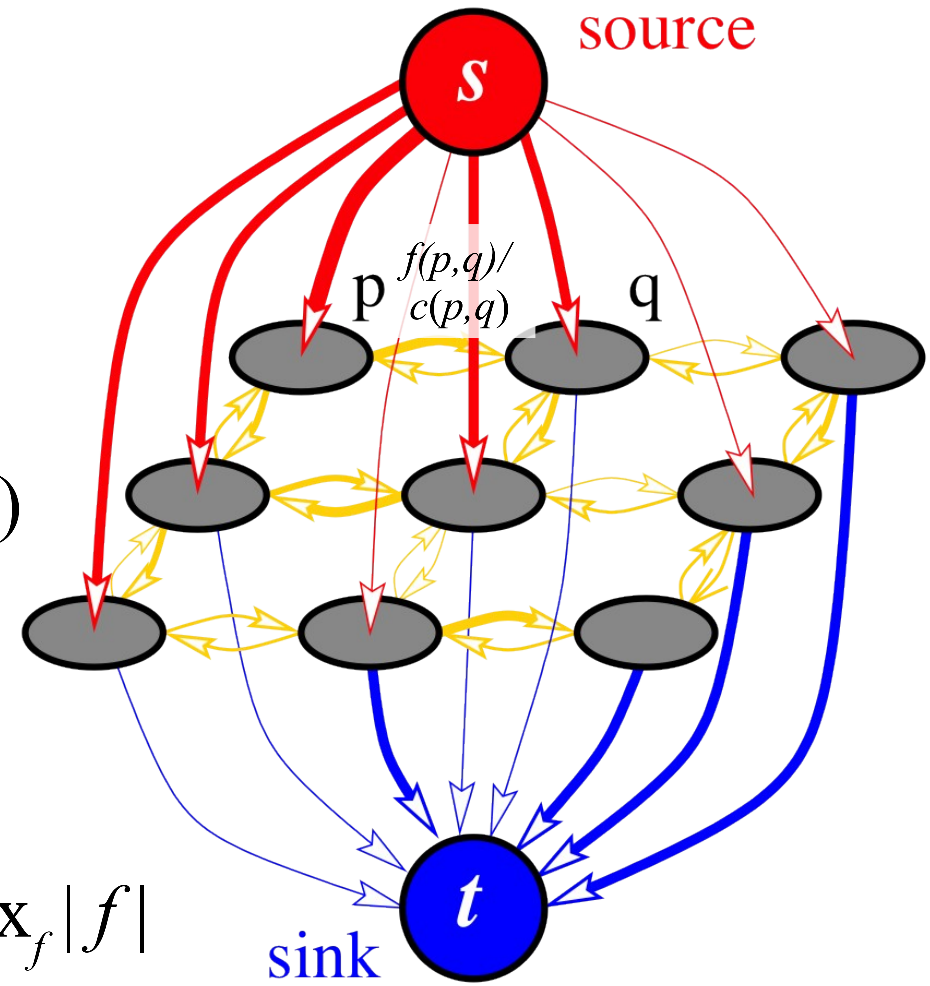
Kirchhoff's law



Flow network constraints

skew symmetry = antisymétrie

- ***s-t* flow** (or just "flow") f
  - $f : V \times V \to \mathbb{R}$
    satisfying flow constraints
- **Value of *s-t* flow**
  - $|f| = \sum_{q \in V} f(s,q) = \sum_{p \in V} f(p,t)$
    - 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

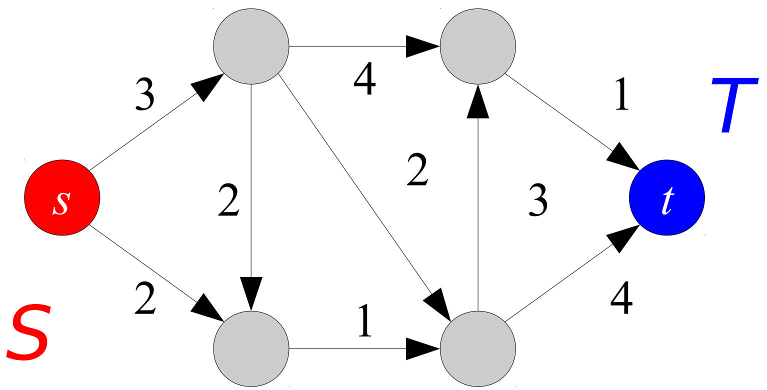


Max-flow min-cut theorem

- 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
  - $|f| = c(\mathrm{S},\mathrm{T}) = ?$



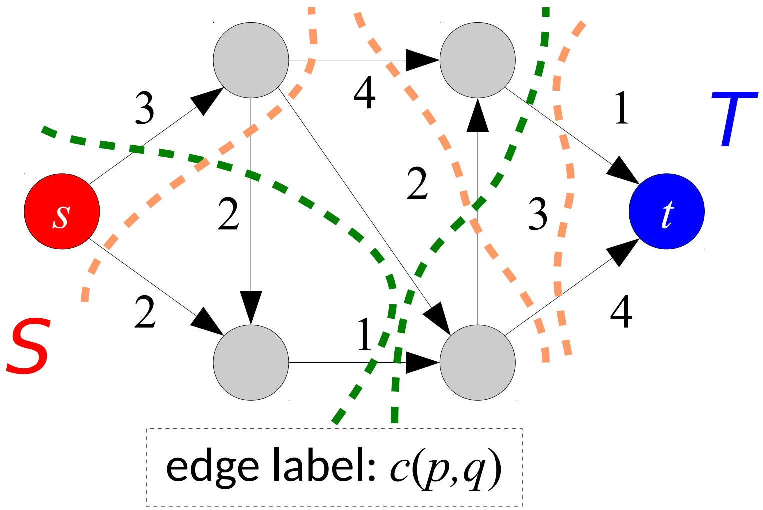
edge label: $c(p,q)$

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
  - $|f| = c(\mathrm{S},\mathrm{T}) = 4$
  - min: enumerate partitions…

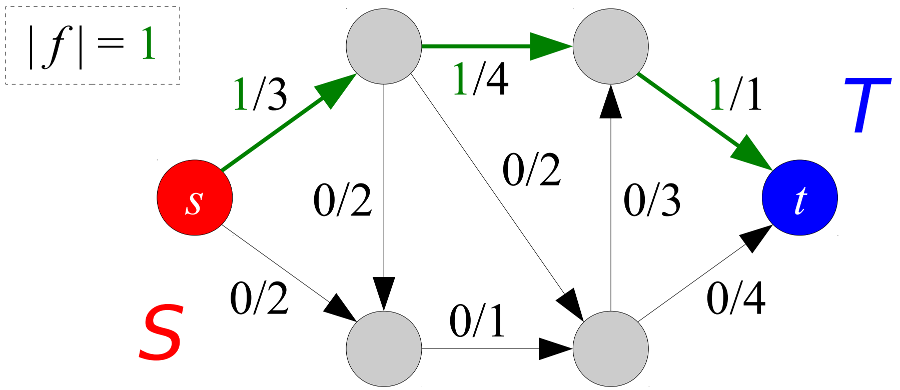


Max-flow min-cut theorem

- 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
  - $|f| = c(\mathrm{S},\mathrm{T}) = 4$
  - min: enumerate partitions...
  - max: try increasing $f(p,q)$...



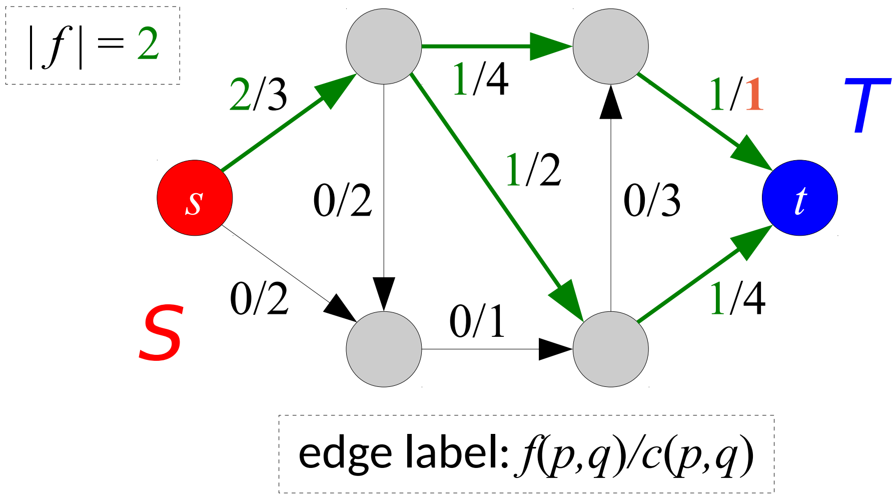
edge label: $f(p,q)/c(p,q)$

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
  - $|f| = c(\mathrm{S,T}) = 4$
  - min: enumerate partitions…
  - max: try increasing $f(p,q)$…

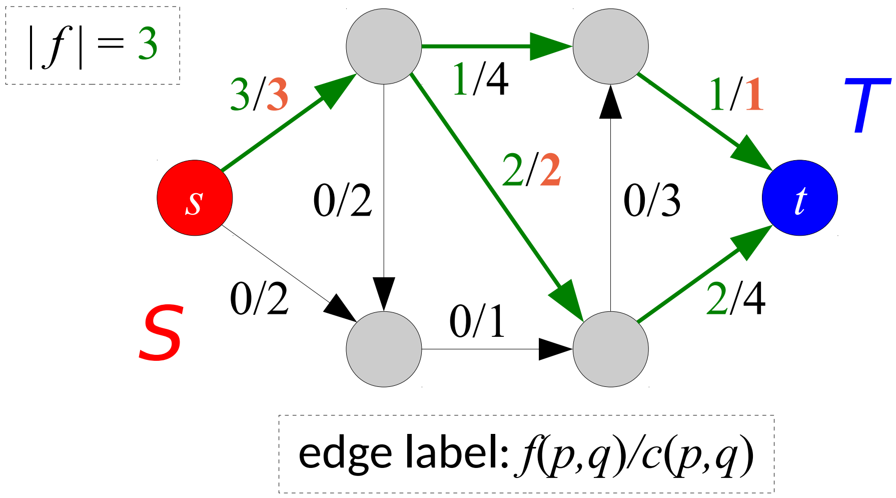


Max-flow min-cut theorem

- 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
  - $|f| = c(\mathrm{S},\mathrm{T}) = 4$
  - min: enumerate partitions…
  - max: try increasing $f(p,q)$…

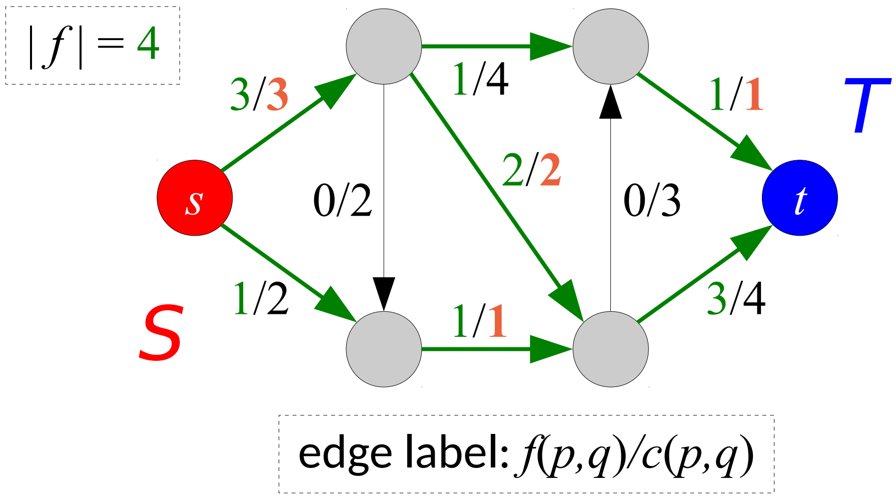


Max-flow min-cut theorem

- 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
  - $|f| = c(\mathrm{S},\mathrm{T}) = 4$
  - min: enumerate partitions…
  - max: try increasing $f(p,q)$…

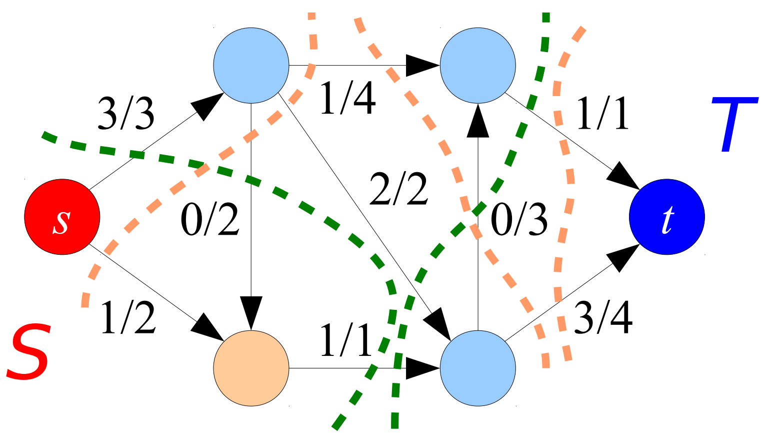


Max-flow min-cut theorem

- 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
  - $|f| = c(\mathrm{S},\mathrm{T}) = 4$
  - min: enumerate partitions…
  - max: try increasing $f(p,q)$…



edge label: $f(p,q)/c(p,q)$

- 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

linear programmaing = programmationlinéaire

- Theorem

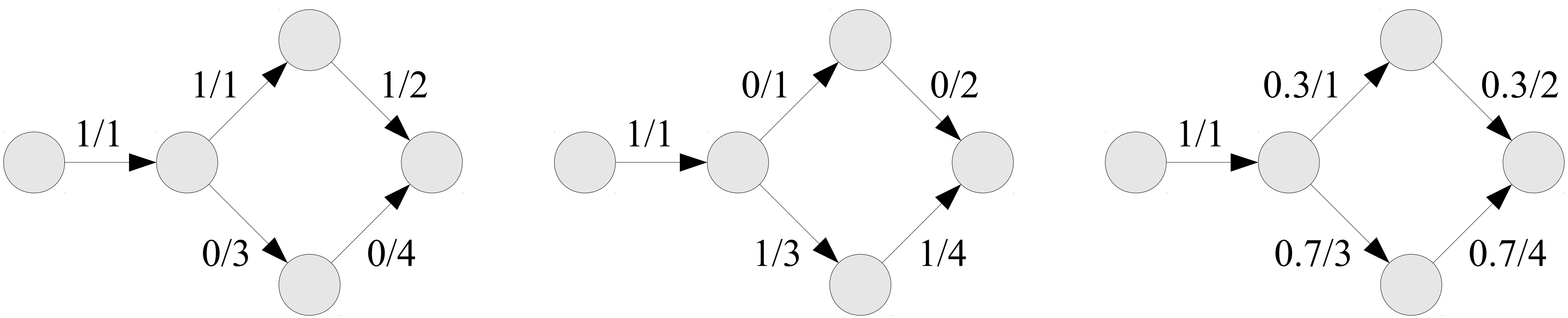
  The maximum value of an s-t flow is equal to the minimum capacity (i.e., min cost) of an s-t cut.

  - 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

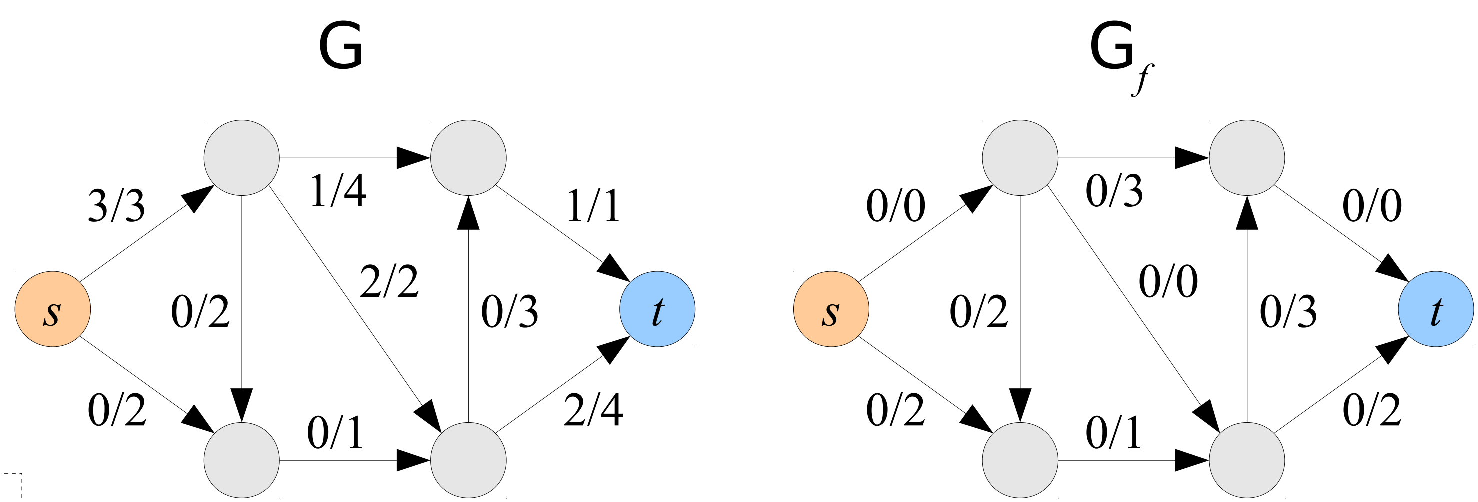
- Polynomial time
- Push-relabel methods
  - better performance for general graphs
  - e.g. Goldberg and Tarjan 1988: $O(V\,E\log(V^2/E))$
    - where V: number of vertices, E: number of edges
- Augmenting paths methods
  - iteratively push flow from source to sink along some path
  - better performance on specific graphs
  - e.g. Ford-Fulkerson 1956: $O(E \max|f|)$ for integer capacity c

To go further on this subject

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:



edge label: $f(p,q)/c(p,q)$

To go further on this subject

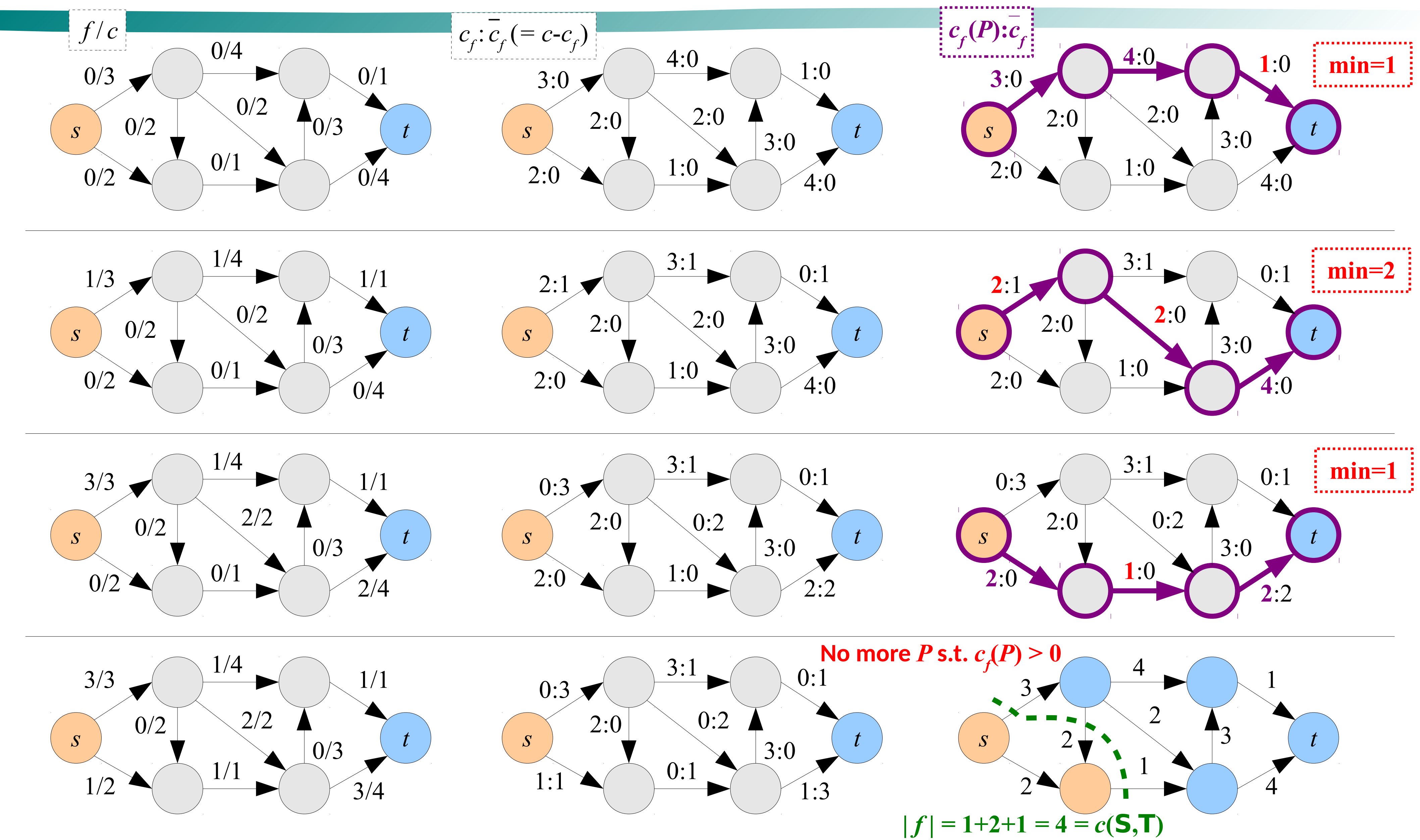
Ford-Fulkerson algorithm (1956)

termination = terminaison
semi-algorithm: termination not guaranteed for all inputs

$f(p,q) \leftarrow 0$ for all edges [P: augmenting path]
while $\exists$ path P from s to t such that $\forall (p,q) \in P \; c_f(p,q) > 0$
    $c_f(P) \leftarrow \min\{c_f(p,q) \mid (p,q) \in P\}$ [min residual capacity]
    for each edge $(p,q) \in P$
        $f(p,q) \leftarrow f(p,q) + c_f(P)$ [push flow along path]
        $f(q,p) \leftarrow f(q,p) - c_f(P)$ [keep skew symmetry]

- 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)

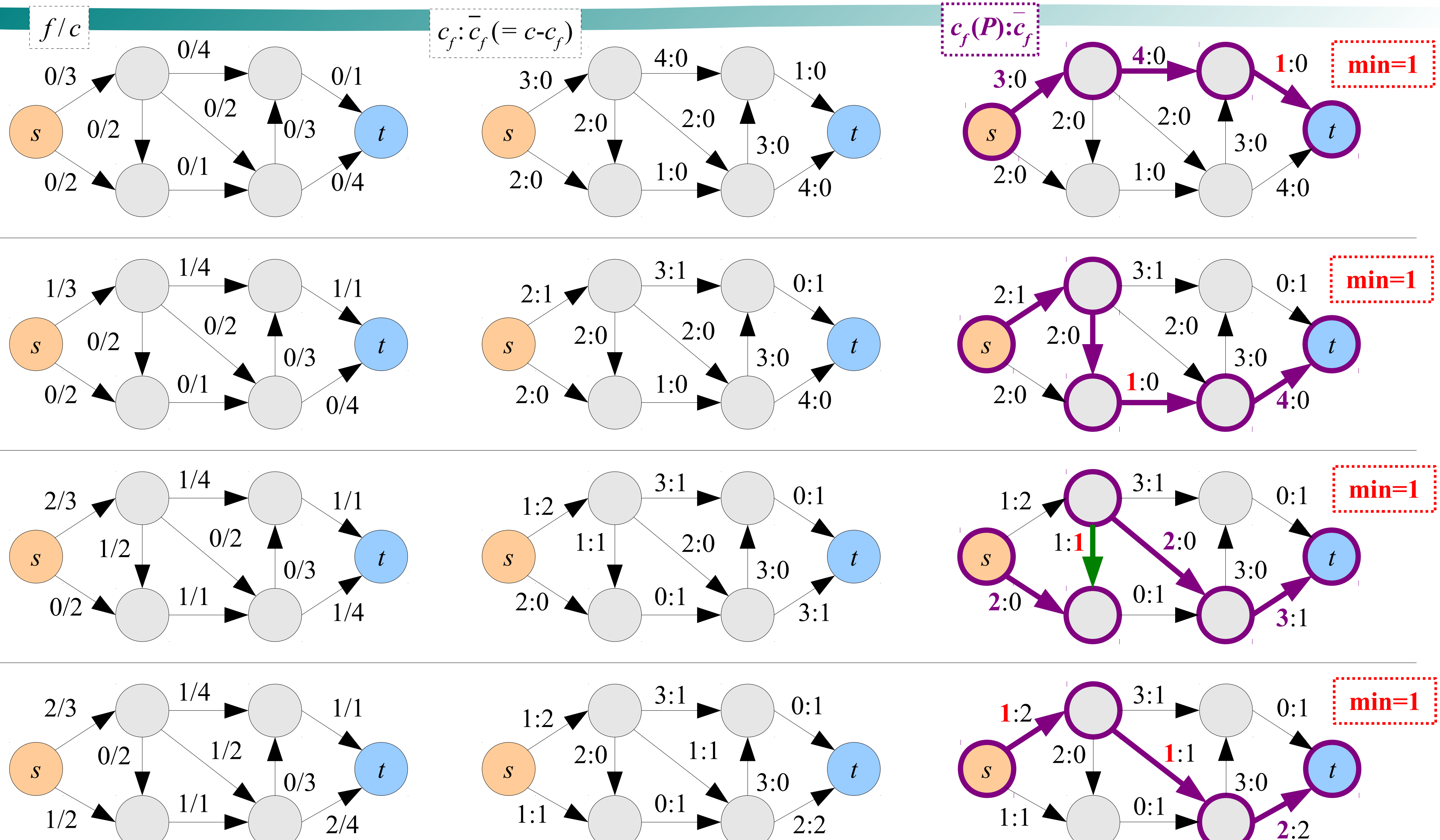
Ford-Fulkerson algorithm: an example



To go further on this subject

Ford-Fulkerson algorithm: an example

Taking edges backwards = OK (and sometimes needed)



$|f| = 1+1+1+1 = 4 = c(\mathbf{S},\mathbf{T})$

Edmonds-Karp algorithm (1972)

breadth-first = en largeur d'abord
sparse = épars, peu dense

- As Ford-Fulkerson but **shortest path** with >0 capacity
  - breadth-first search for augmenting path (cf. example above)
- Termination: now guaranteed
- Complexity: $O(V\,E^2)$
  - slower than push-relabel methods for general graphs
  - faster in practice for sparse graphs
- Other variant (Dinic 1970), complexity: $O(V^2\,E)$
  - other flow selection (blocking flows)
  - $O(V\,E\log V)$ 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 “best” algorithm depends on the context

To go further on this subject

Variant: Multiway cut problem

planar = planaire

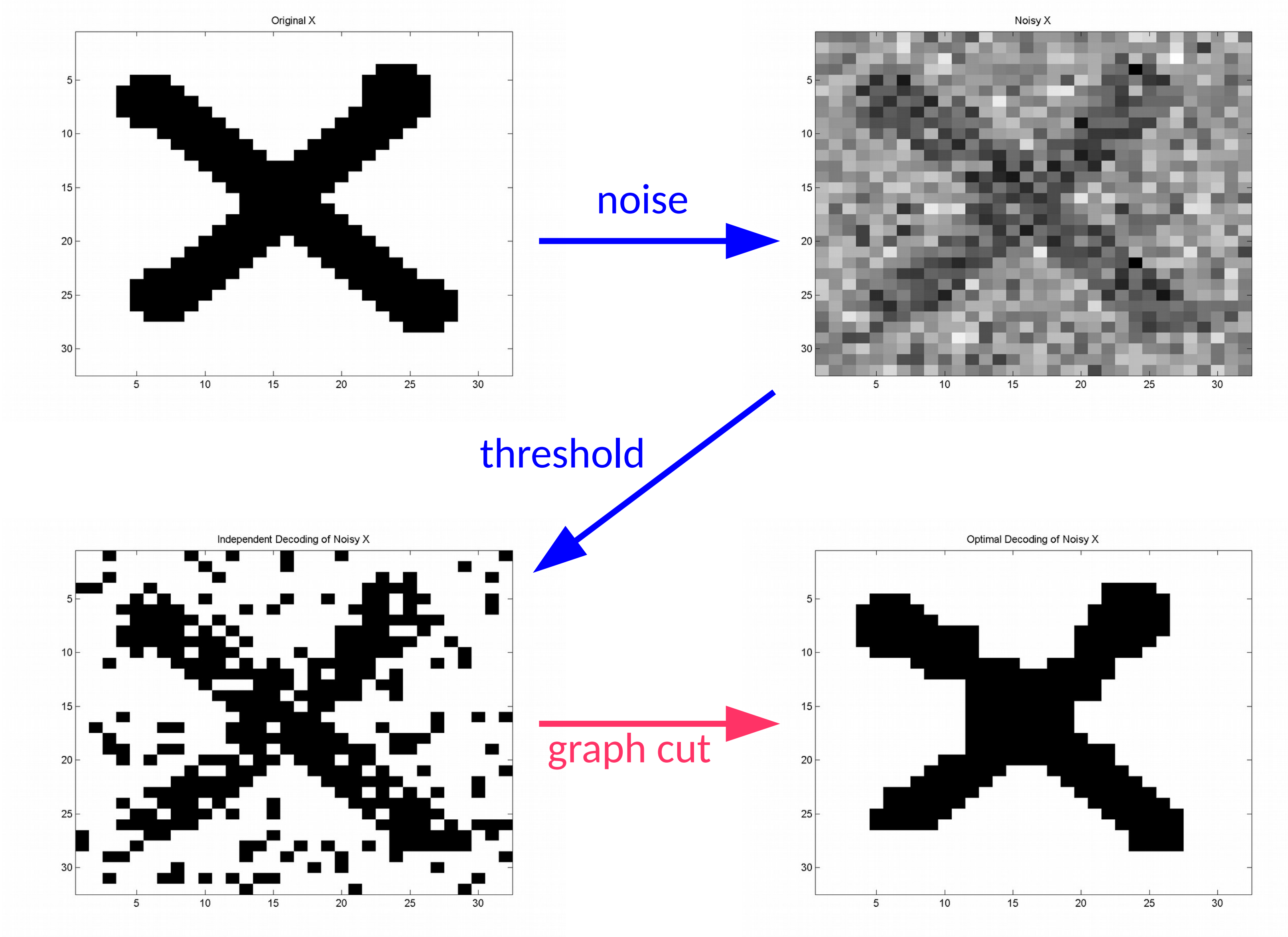
- More than two terminals: $\{s_1,\ldots,s_k\}$
- Multiway cut:
  - set of edges leaving each terminal in a separate component
- Multiway cut problem
  - find cut with minimum weight
  - same as min cut when $k=2$
  - NP-hard if $k \geq 3$ (in fact APX-hard, i.e., NP-hard to approx.)
  - but can be solved 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

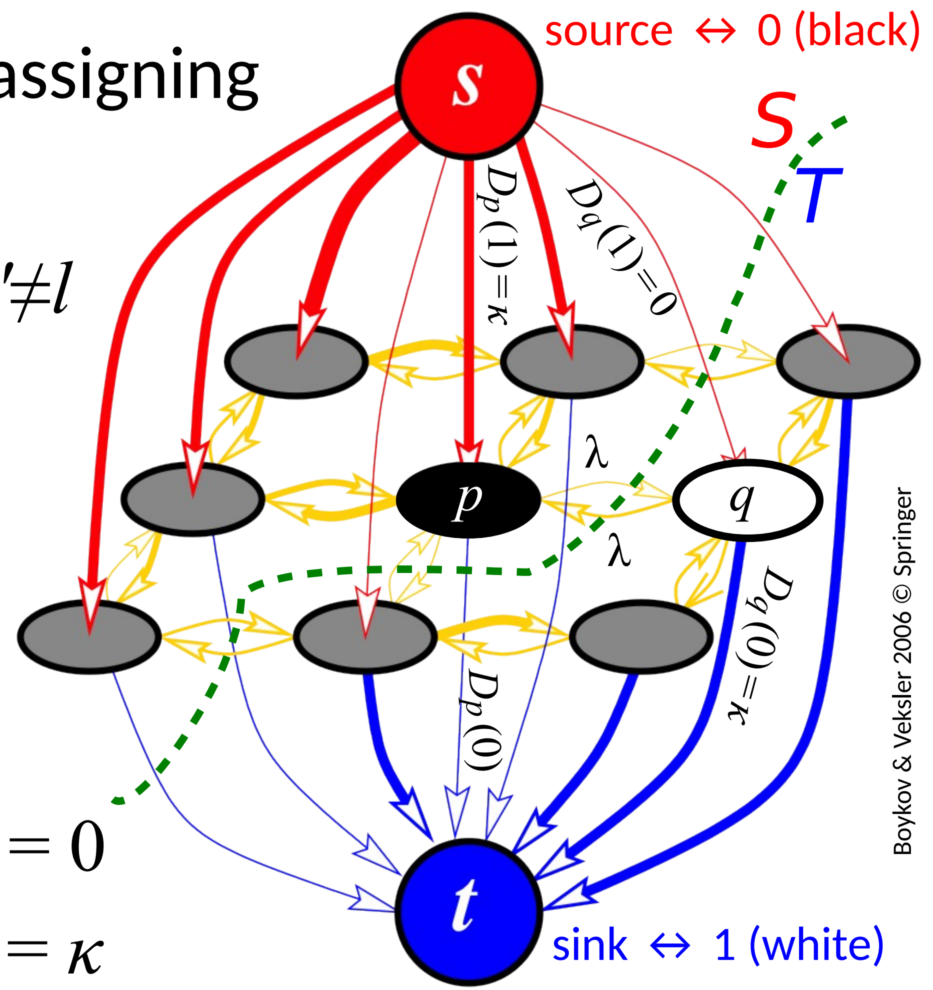


Binary image restoration: The graph cut view

penalty = pénalité, coût
reward = récompense

I_p : intensity of image I at pixel p

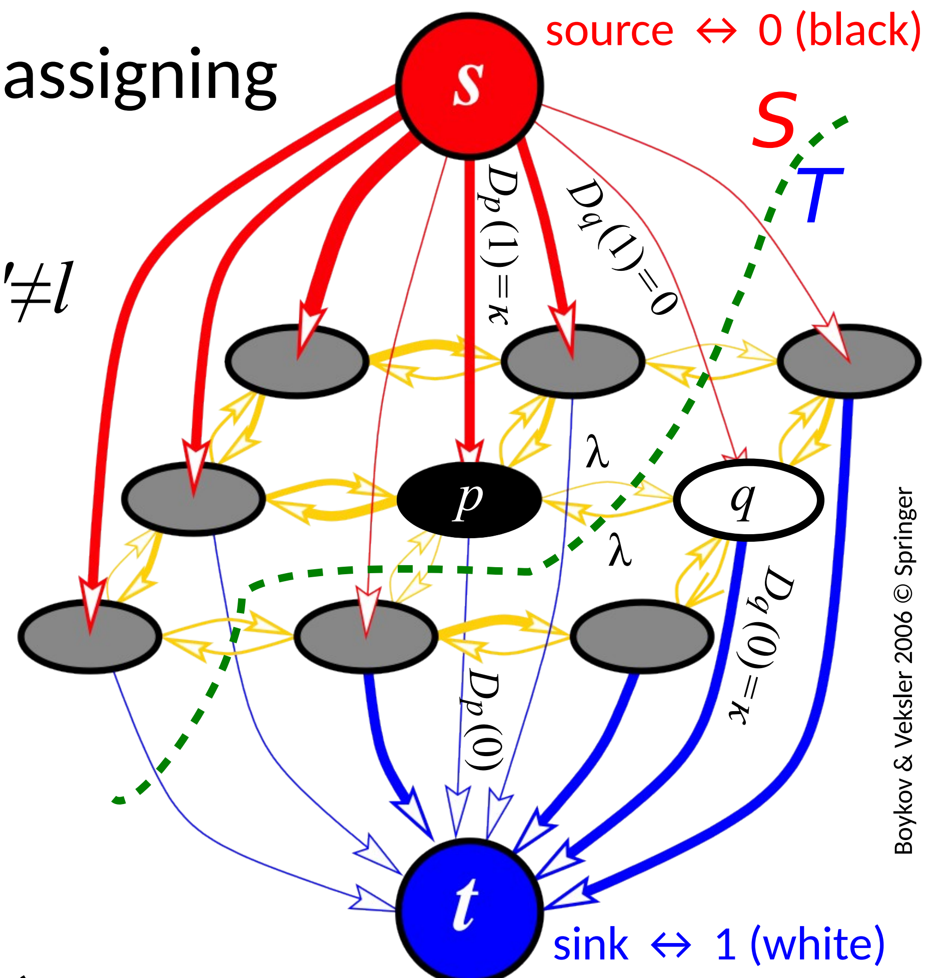
- Agreement with observed data
  - $D_p(l)$: penalty (= -reward) for assigning label $l \in \{0,1\}$ to pixel $p \in$ P
  - if $I_p = l$ then $D_p(l) < D_p(l')$ for $l' \neq l$
  - $w(s,p) = D_p(1)$, $w(p,t) = D_p(0)$
- Example:
  - if $I_p = 0$, $D_p(0) = 0$, $D_p(1) = \kappa$
    if $I_p = 1$, $D_p(0)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



Binary image restoration: The graph cut view

penalty = pénalité, coût
reward = récompense
regularizing constraint = contrainte de régularisation
smoothing = lissage

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

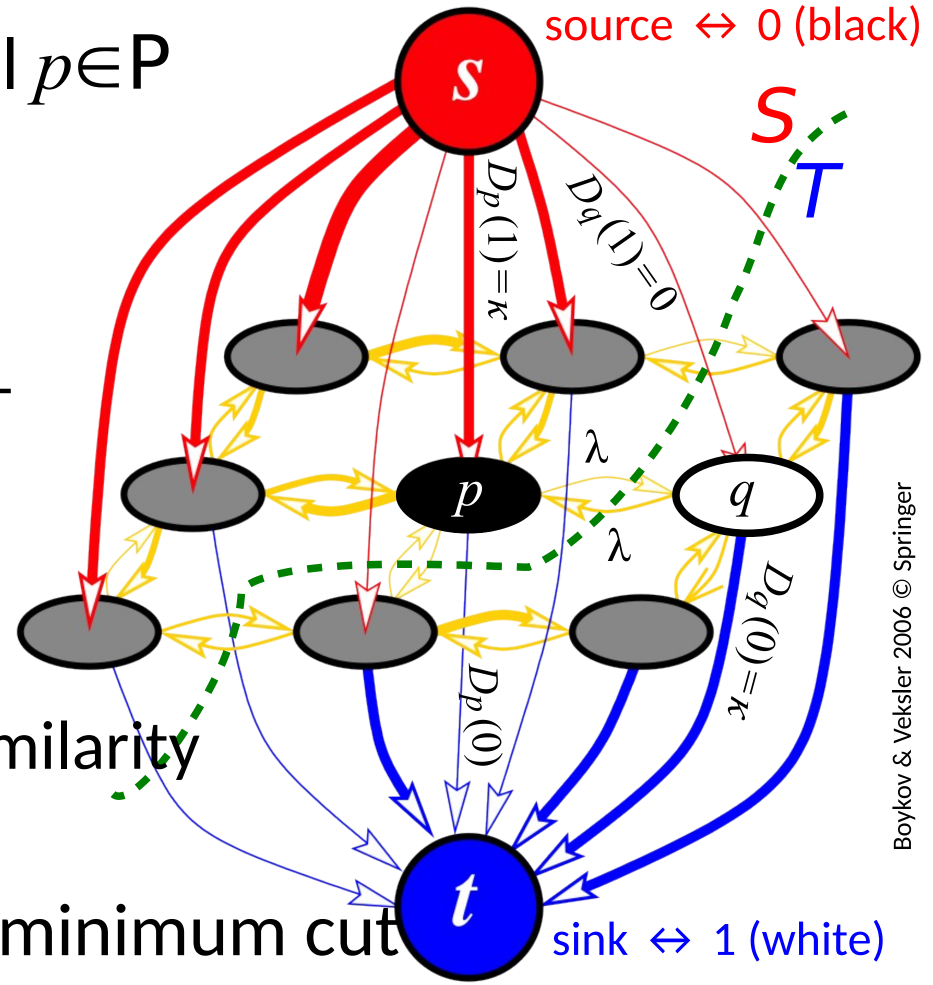
labeling = étiquetage

- Binary labeling f [N.B. different from "flow f"]
  - assigns label $f_p \in \{0,1\}$ to pixel $p \in$ P
    - $f : P \rightarrow \{0,1\}$ $\quad f(p) = f_p$
- Cut $C = \{S,T\} \leftrightarrow$ labeling f
  - 1-to-1 correspondence: $f = \mathbf{1}_{|T}$
- Cost of a cut: $|C| =$

  $\sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in S \times T} w(p,q)$

  = cost of flip + cost of local dissimilarity
- Restored image:

  = labeling corresponding to a minimum cut



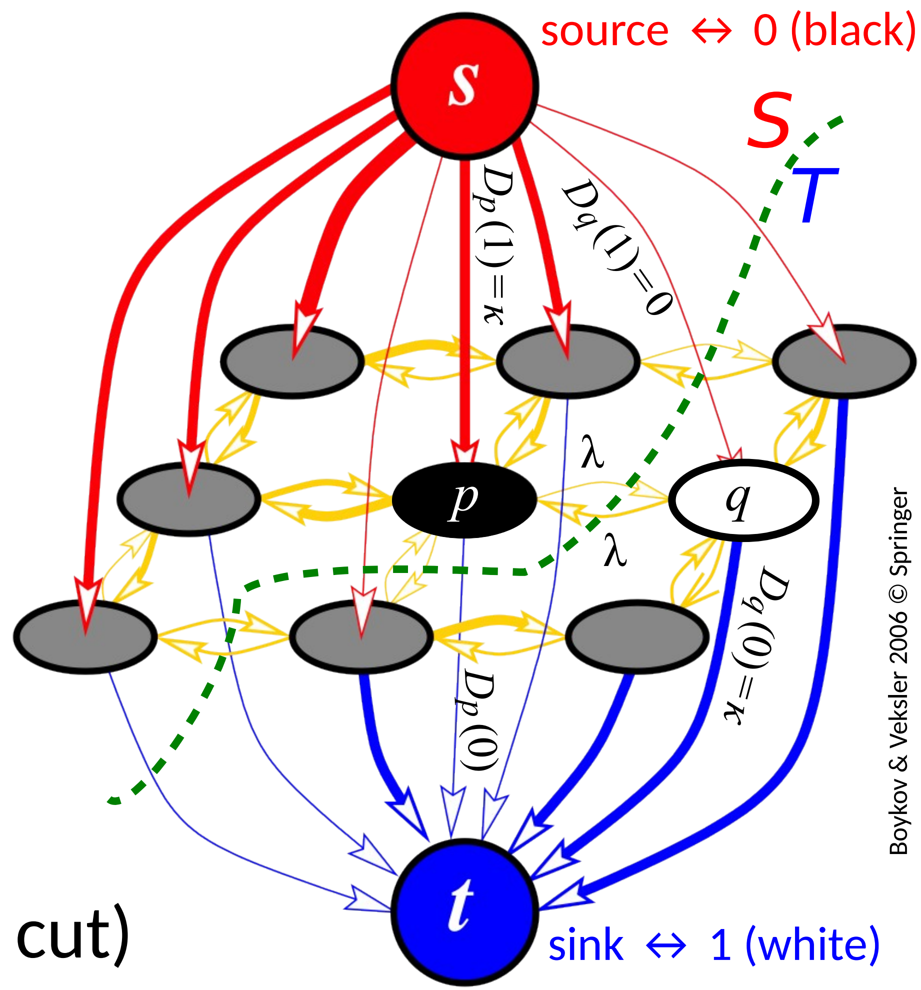
Binary image restoration: The energy view

- Energy of labeling f
  - $E(f) \overset{\text{def}}{=} |C| =$
    $\sum_{p \in \mathrm{P}} D_p(f_p) +$
    $\lambda \sum_{(p,q) \in \mathrm{N}} \mathbf{1}(f_p = 0 \wedge f_q = 1)$

    where

    $\mathbf{1}(\text{false}) = 0 \;\mid\; \mathbf{1}(\text{true}) = 1$

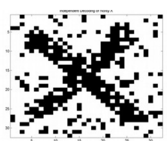
    [or: $\frac{1}{2} \lambda \sum_{(p,q) \in \mathrm{N}} \mathbf{1}(f_p \neq f_q)$]
- Restored image:
  - labeling corresponding to minimum energy (= minimum cut)



Binary image restoration: The smoothing factor

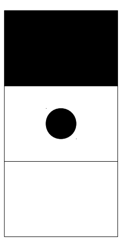
cluster = amas
outlier = point aberrant

- Small λ (actually λ/κ):



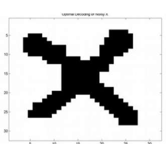
  - pixels choose their label independently of their neighbors

- Large λ:

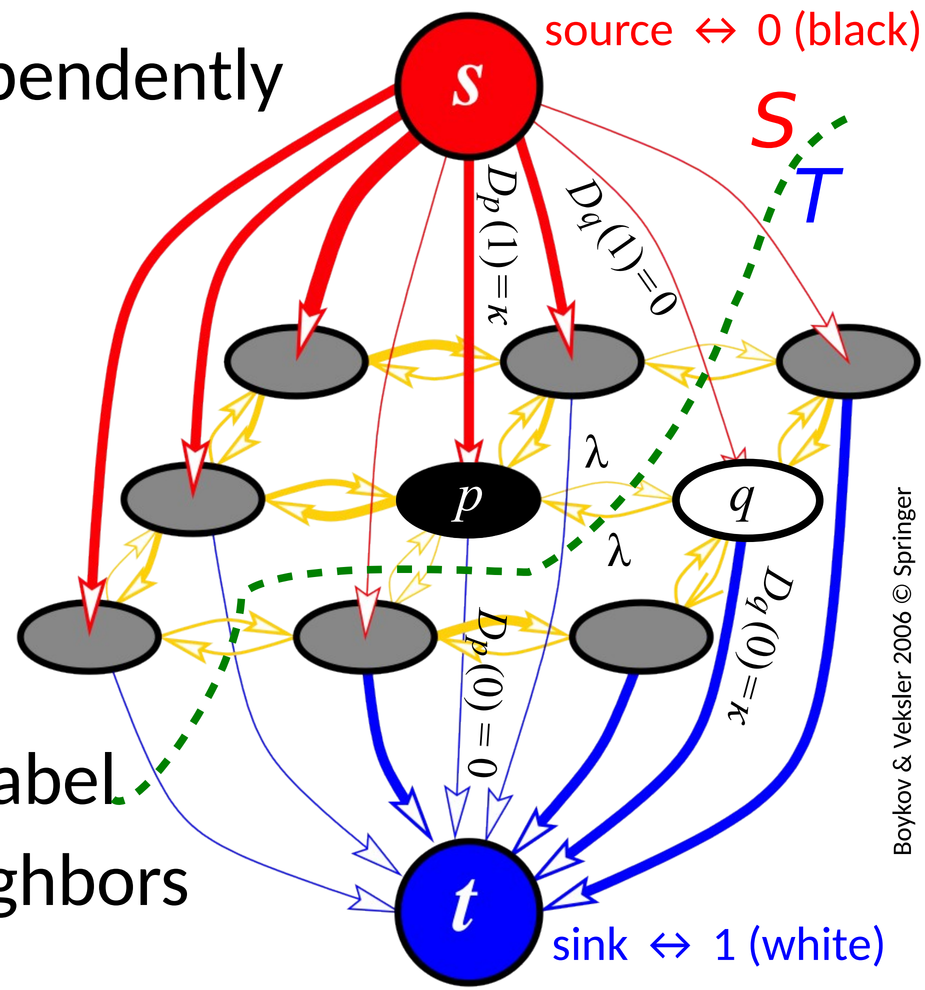


  - pixels choose the label with smaller average cost

- Balanced λ value:



  - pixels form compact, spatially coherent clusters with same label
  - noise/outliers conform to neighbors



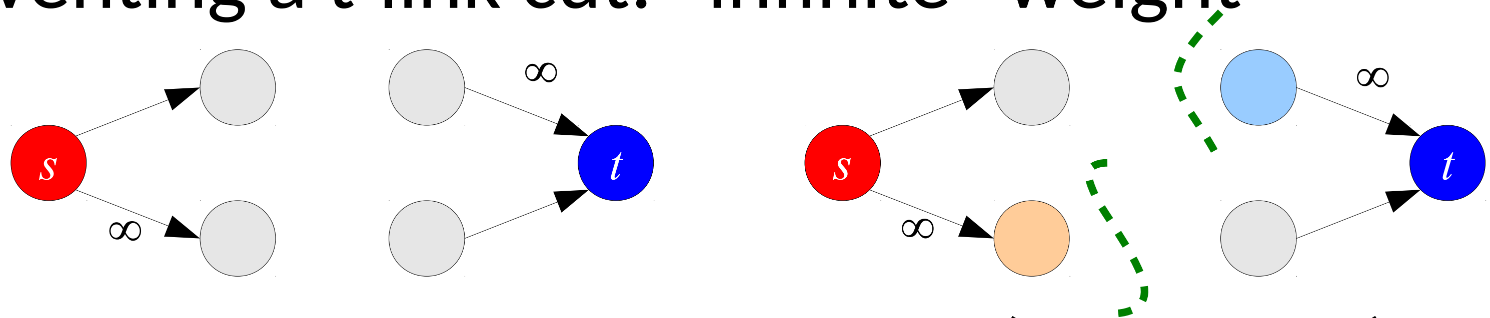
Graph cuts for energy minimization

- **Given some energy** $E(f)$ such that
  - $f : \mathsf{P} \rightarrow \mathsf{L} = \{0,1\}$ binary labeling
  - $E(f) = \underbrace{\sum_{p\in\mathsf{P}} D_p(f_p)}_{E_{\text{data}}(f)} + \underbrace{\sum_{(p,q)\in\mathsf{N}} V_{p,q}(f_p\,, f_q)}_{E_{\text{regul}}(f)}$
  - regularity condition (see below)
    - $V_{p,q}(0,0) + V_{p,q}(1,1) \leq V_{p,q}(0,1) + V_{p,q}(1,0)$
- Theorem: **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]

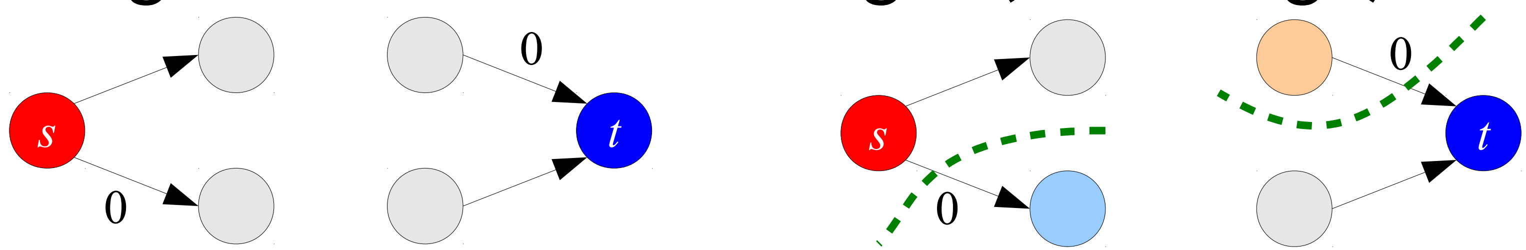
  [structure of graph somehow similar to above form]

Graph construction

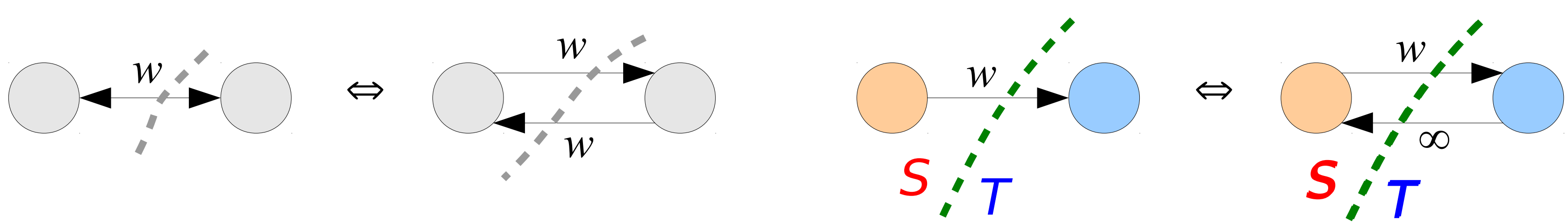
- Preventing a t-link cut: “infinite” weight



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



- Bidirectional edge vs monodirectional & back edges



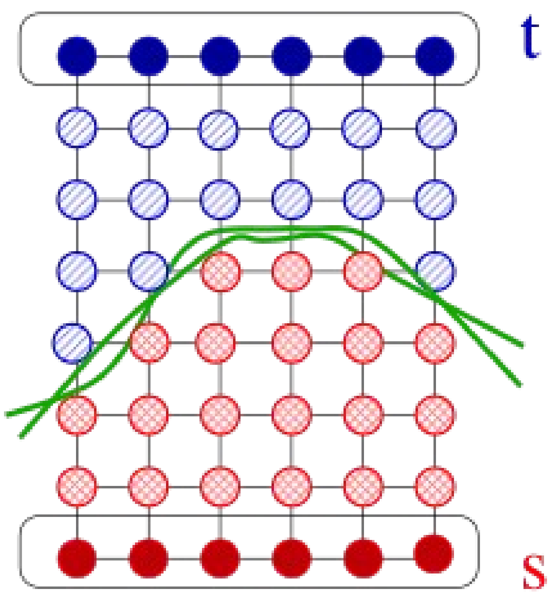
To go further on this subject

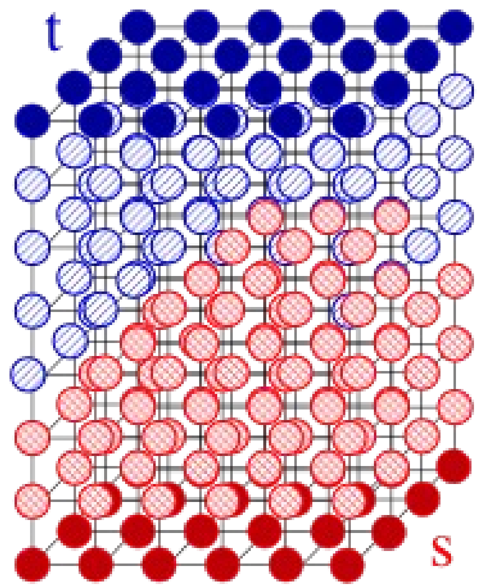
Graph cuts as hypersurfaces

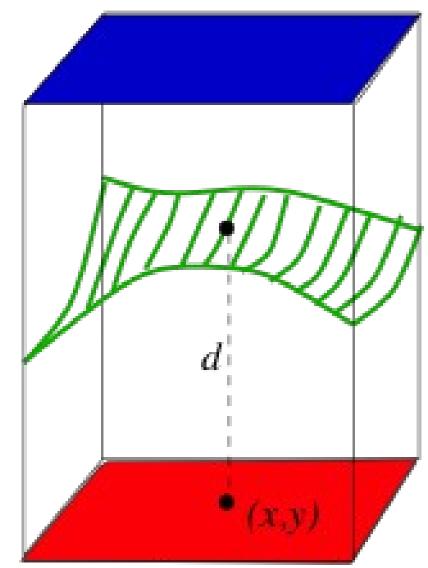
(cf. Boykov & Veksler 2006)

seed = graine

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







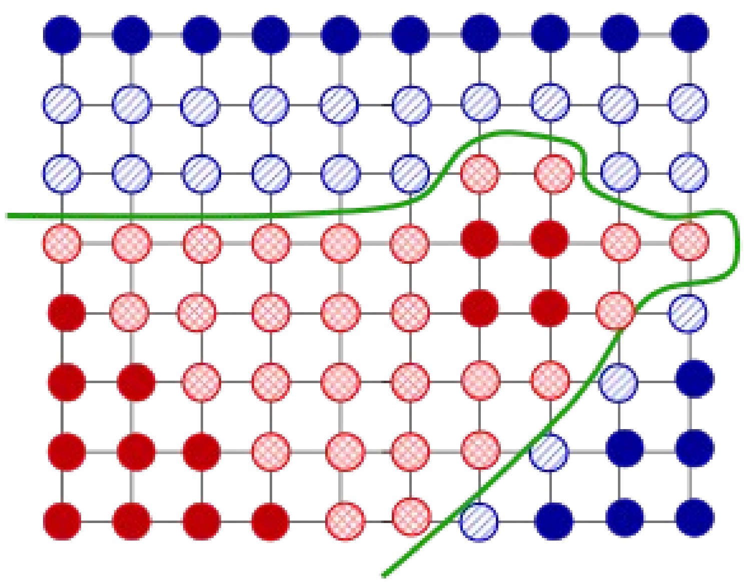
N.B. Several “seeds” (sources and sinks)

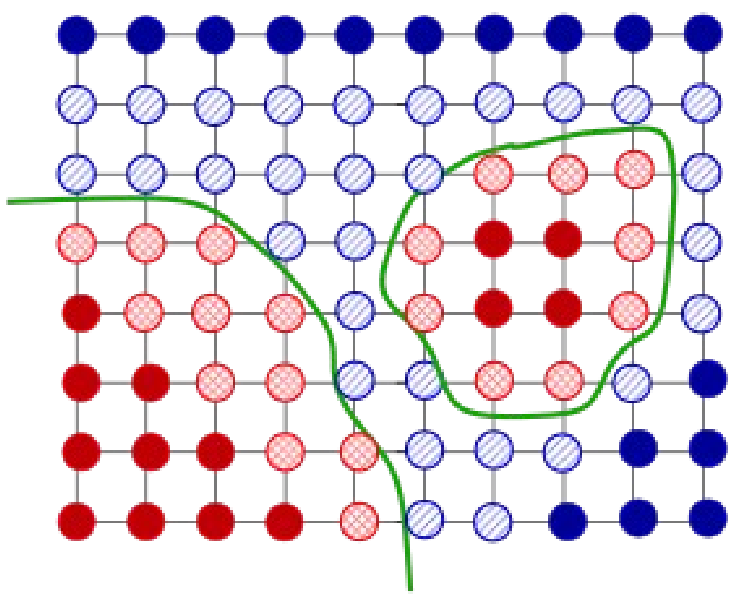
To go further on this subject

Example of topological issue

seed = graine

- Connected seeds
- Disconnected seeds



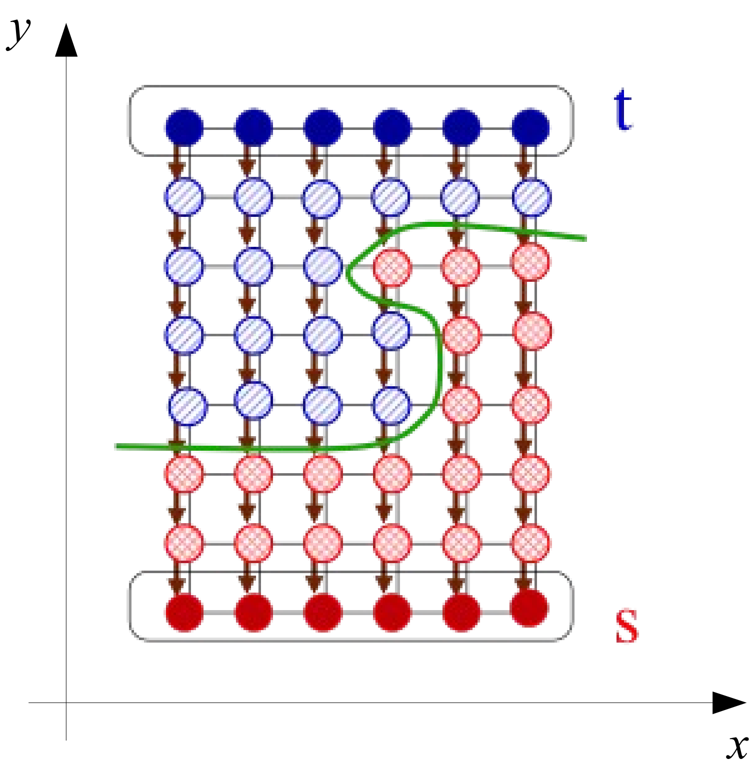


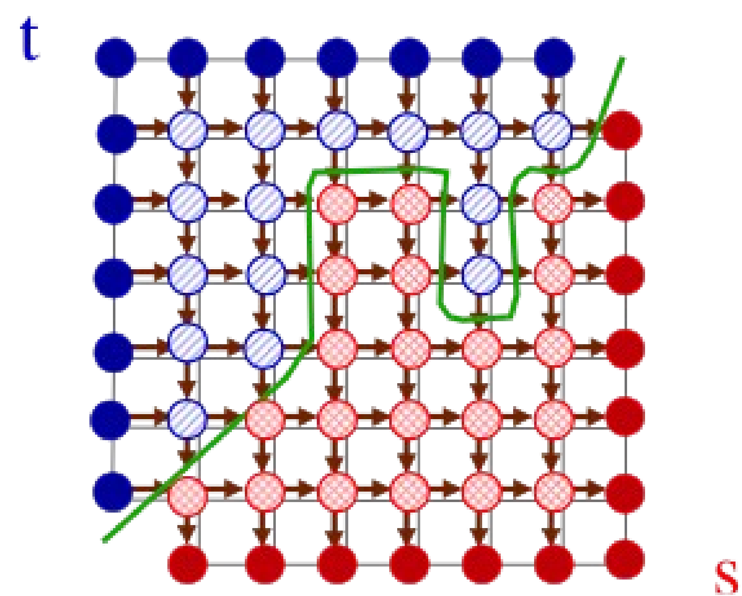
To go further on this subject

Example of topological constraint: fold prevention

disparity map = carte de disparité

- 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

simulated annealing = recuit simulé

- 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 ≠ 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., λ)

☛ Global optimum of **abstracted problem**, not necessarily best solution **in real life**

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 (contrary to some aspects of deep learning 😉)

Graph cut vs deep learning

- Graph cut
  - works well, with proven optimality bounds
- Deep learning
  - works extremely well, but mainly empirical
- Somewhat complementary
  - graph cut sometimes used to regularize network output

Application to image segmentation

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

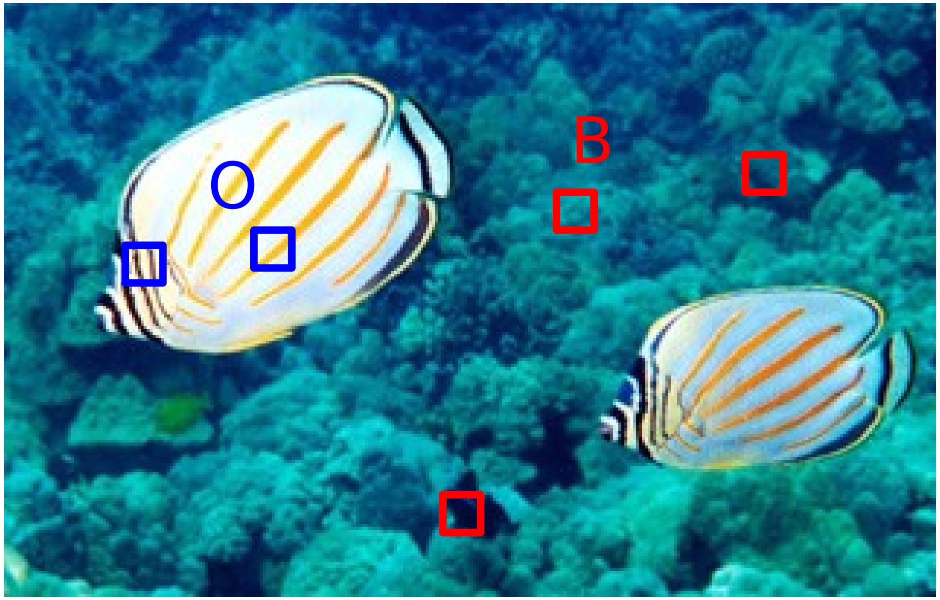
background = arrière-plan
sample = échantillon
area = zone

[Duchenne & al. 2008]

Application to image segmentation

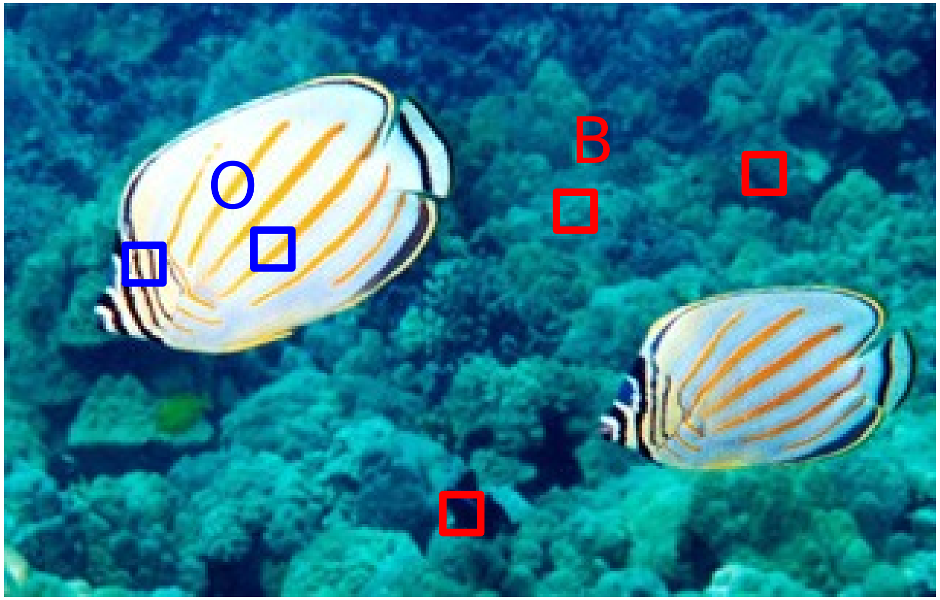
background = arrière-plan
sample = échantillon
area = zone

- 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 ?

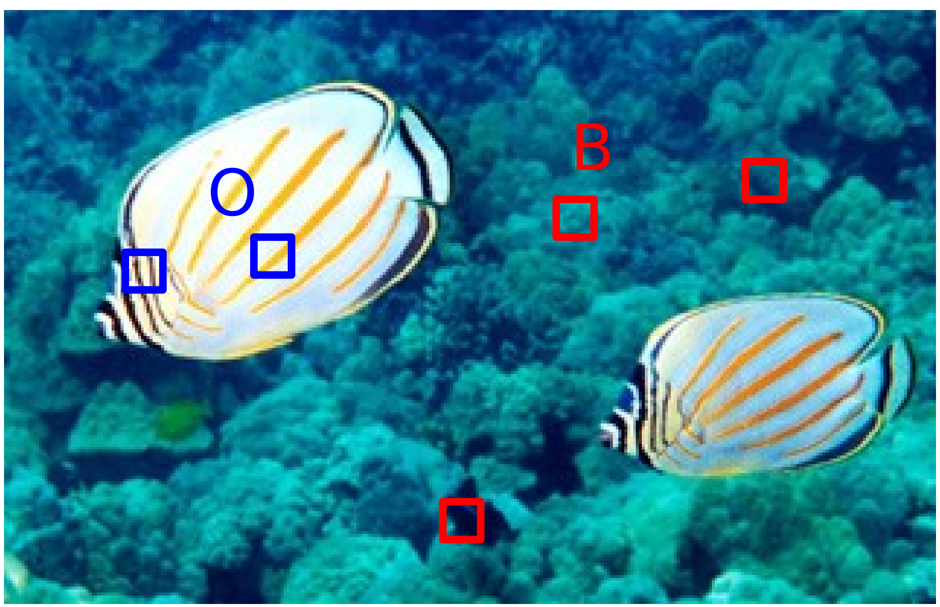


Intuition

background = arrière-plan
sample = échantillon
area = zone

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

[Boykov & Jolly 2001]

- Pixel labeling with binary decision $f_p \in L = \{0,1\}$
  - $1 =$ object, $0 =$ 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
  - λ : relative importance of regularization term vs data term

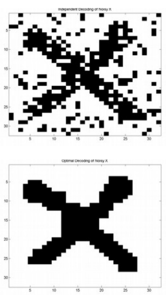
data term = **terme d'attache aux données**
regularization term = **terme de régularisation**
a.k.a. = also known as
penalty = pénalité, coût
to assign = affecter (une valeur à qq chose)
sample = échantillon
background =
boundary = frontière
neighboring pixel = pixel voisin

To go further on this subject

Probabilistic justification/framework

posterior probability = probabilité a posteriori
likelihood = vraisemblance
(log-)likelihood = (log-)vraisemblance

- Minimize $E(f) \leftrightarrow$ maximize posterior proba. $\Pr(f|I)$
- Bayes theorem:



$$\boxed{\Pr(f|I)}\ \boxed{\Pr(I)} = \boxed{\Pr(I|f)}\ \boxed{\Pr(f)}$$

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

A constant (independent of f)

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

$\leftrightarrow$ 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 ($\rightarrow$ sums)

$$E(f) = D(f) + \lambda R(f) \;\Leftrightarrow\; -\log \Pr(f|I) + c = -\log \Pr(I|f) - \log \Pr(f)$$

To go further on this subject

Data term: linking estimated labels to observed pixels

penalty = pénalité, coût
to assign = affecter
sample = échantillon
likelihood = vraisemblance
random variable = variable aléatoire

- $D(f)$ and likelihood
  - penalty for assigning labels f in I given sample assignments
    ↔ (log-)likelihood that f is consistent with image samples
  - $D(f) = -\log L(f \mid I) = -\log \Pr(I \mid f)$
- Pixel independence hypothesis (common approximation)
  - $\Pr(I \mid f) = \prod_{p \in P} \Pr(I_p \mid f_p)$ if pixels iid (independent and identically distributed random variables)
  - $D(f) = \sum_{p \in P} D_p(f_p)$ where $D_p(f_p) = -\log \Pr(I_p \mid f_p)$
    - $D_p(f_p)$: penalty for observing I_p for a pixel of type f_p

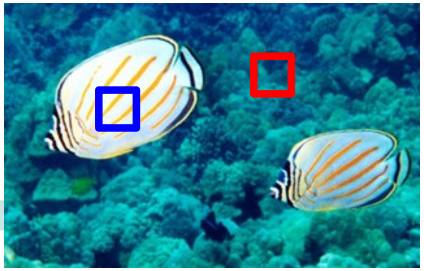
wrong strictly speaking, but "true enough" to be often assumed

☛ Find an estimate of $\Pr(I_p \mid f_p)$

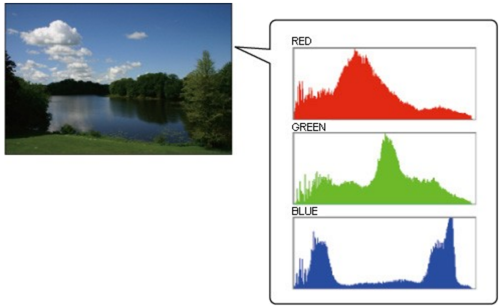
To go further on this subject

Data term: likelihood/color model

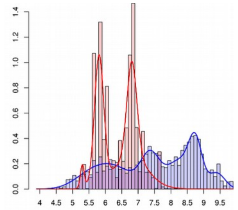
empirical probability = probabilité empirique (fréquence relative)
Gaussian mixture = mélange de gaussiennes

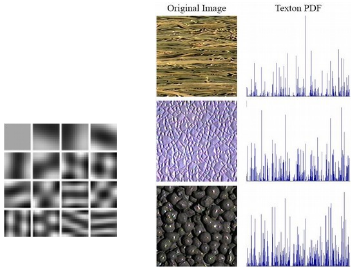


- Approaches to find an estimate of $\Pr(I_p | f_p)$
  - histograms
    - build an empirical distribution of the color of object/background pixels, based on pixels marked as object/background
    - estimate $\Pr(I_p \,|\, f_p)$ based on histograms: $\Pr_{\text{emp}}(rgb|O), \Pr_{\text{emp}}(rgb|B)$
  - 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…



© Canon 2012

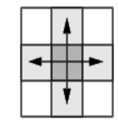
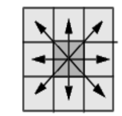




Blunsden 2006 @ U. of Edinburgh

Regularization term: locality hypotheses

- Markov random field (MRF), or Markov network
  - neighborhood system: $N = \{N_p \mid p \in P\}$

    - N_p : set neighbors of p such that $p \notin N_p$ and $p \in N_q \Leftrightarrow q \in N_p$
  - $X = (X_p)_{p \in P}$: field (set) of random variables such that each random variable X_p depends on other random variables only through its neighbors N_p

    ☛ **locality hypothesis:** $\Pr(X_p = x \mid X_{P \setminus \{p\}}) = \Pr(X_p = x \mid X_{N_p})$
  - $N \approx$ undirected graph: (p,q) edge iff $p \in N_q$ $(\Leftrightarrow q \in N_p)$

    (MRF also called undirected graphical model)

Markov random field = champ de Markov
random variable = variable aléatoire
neighborhood = voisinage
undirected graph = graph non orienté
graphical model = modèle graphique

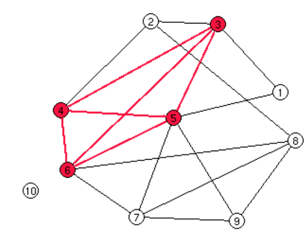
To go further on this subject

Regularization term: locality hypotheses

Gibbs random field = champ de Gibbs
undirected graph = graph non orienté
clique = clique (!)
clique potential = potentiel de clique
prior probability = probabilité a posteriori

- Gibbs random field (GRF)
  - *G* undirected graph, $X = (X_p)_{p \in P}$ random variables such that

$$\Pr(X = x) \propto \exp(-\textstyle\sum_{C \text{ clique of } G} V_C(x))$$



    - 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 MRF w.r.t. graph *N* iff X GRF w.r.t. graph *N*

  ☛ provides a characterization of MRFs as GRFs

To go further on this subject

Regularization term: locality hypotheses

[Boykov, Veksler & Zabih 1998]

- Hypothesis 1: only 2nd-order cliques (i.e., edges)

$$R(f) = -\log \Pr(f) = -\log \exp(-\textstyle\sum_{(p,q) \text{ edge of } G} V_{(p,q)}(f)) \quad \text{[GRF]}$$

$$= \textstyle\sum_{(p,q)\in N} V_{p,q}(f_p, f_q) \quad \text{[MRF \textbf{pairwise potentials}]}$$

- Hypothesis 2: (generalized) Potts model

$$V_{p,q}(f_p, f_q) = B_{p,q}\, \mathbf{1}(f_p \neq f_q)$$

i.e., $V_{p,q}(f_p, f_q) = 0$ if $f_p = f_q$

$V_{p,q}(f_p, f_q) = B_{p,q}$ if $f_p \neq f_q$

(Origin: statistical mechanics
- spin interaction in crystalline lattice
- link with “energy” terminology)

pairwise = par paire
pairwise potential = potentiel d'ordre 2
Potts model = modèle de Potts
statistical mechanics = physique statistique

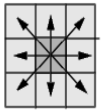
To go further on this subject

Examples of boundary penalties (ad hoc)

- Penalize label discontinuity at intensity continuity

  [Boykov & Jolly 2001]

  – $B_{p,q} = \exp(-(I_p - I_q)^2/2\sigma^2) \,/\, \mathrm{dist}(p,q)$
    - 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 $\mathrm{dist}(p,q)$ [here: 1 or $\sqrt{2}$]
    - ≈ distribution of noise among neighboring pixels



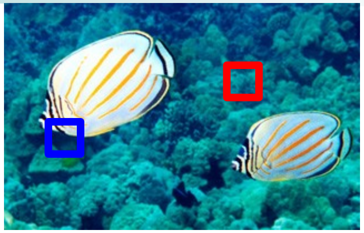
- Penalize label discontinuity at low gradient

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

Wrapping up

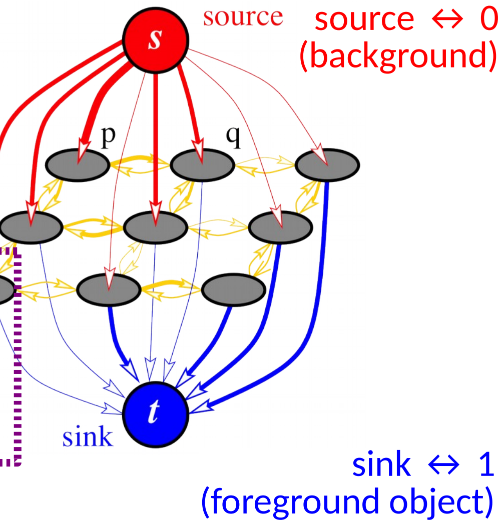
- Pixel labeling with binary decision $f_p \in \{0,1\}$
  - 0 = background, 1 = object
- Energy formulation
  - minimize $E(f) = D(f) + \lambda\, R(f)$
  - data term: $D(f) = \sum_{p\in \mathrm{P}} D_p(f_p)$
    - $D_p(f_p)$: penalty for assigning label f_p to pixel p given its color/texture
  - regularization term: $R(f) = \sum_{(p,q)\in \mathrm{N}} B_{p,q}\, \mathbf{1}(f_p \neq f_q)$
    - $B_{p,q}$: penalty for label discontinuity between neighbor pixels p, q
  - λ : relative importance of regularization term vs data term

Graph-cut formulation (version 1)



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

Edge	Weight	Sites
(p,q)	$\lambda\, B_{p,q}$	$(p,q) \in$ N
(s,p)	$D_p(1)$	$p \in$ P
(p,t)	$D_p(0)$	$p \in$ P

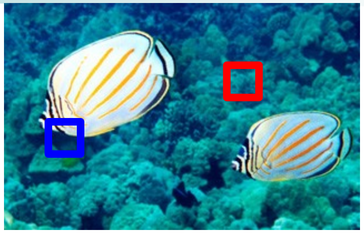


Warning:
the connexity N
is specific to
the problem !

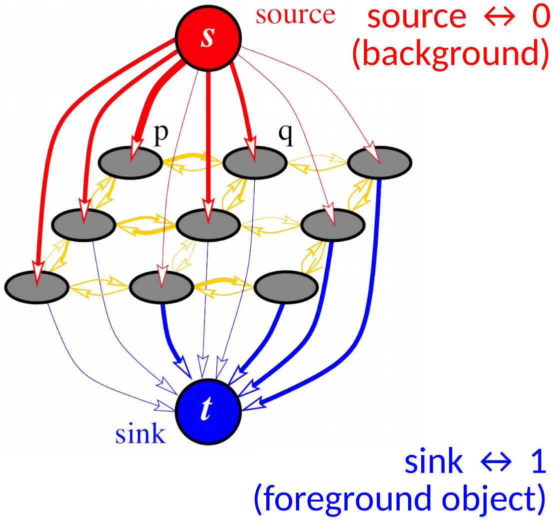
  - $E(f) = \sum_{p\in \mathrm{P}} D_p(f_p) + \lambda \sum_{(p,q)\in \mathrm{N}} B_{p,q}\, \mathbf{1}(f_p \neq f_q)$
    - ex. $D_p(l) = -\log \Pr_{\mathrm{emp}}(I_p \mid f_p = l)$ [empirical probability for O et B]
    - ex. $B_{p,q} = \exp(-(I_p - I_q)^2/2\sigma^2) \,/\, \mathrm{dist}(p,q)$

Graph-cut formulation (version 1)

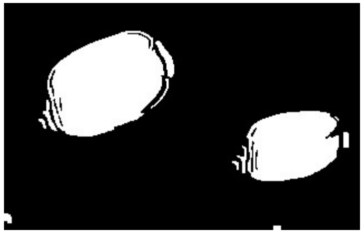
- Direct expression as graph-cut problem:
  - $V = \{s,t\} \cup P$
  - $E = \{(s,p) \mid p \in P\} \cup \{(p,q) \mid p,q \in N\} \cup \{(p,t) \mid p \in P\}$



Edge	Weight	Sites
(p,q)	$\lambda B_{p,q}$	$(p,q) \in N$
(s,p)	$D_p(1)$	$p \in P$
(p,t)	$D_p(0)$	$p \in P$

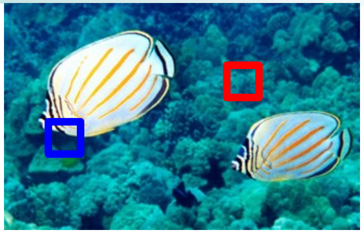


  - $E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{(p,q) \in N} B_{p,q} \mathbf{1}(f_p \neq f_q)$



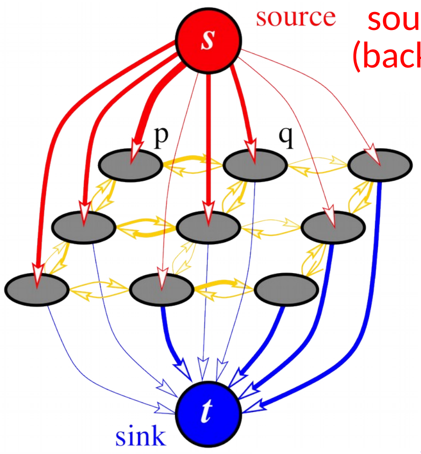
- Any problem/risk with this formulation ?

Graph-cut formulation (version 1)



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

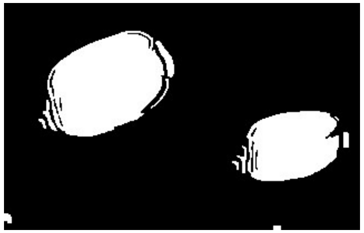
Edge	Weight	Sites
(p,q)	$\lambda\, B_{p,q}$	$(p,q) \in \mathsf{N}$
(s,p)	$D_p(1)$	$p \in \mathsf{P}$
(p,t)	$D_p(0)$	$p \in \mathsf{P}$



source ↔ 0 (background)

sink ↔ 1 (foreground object)

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



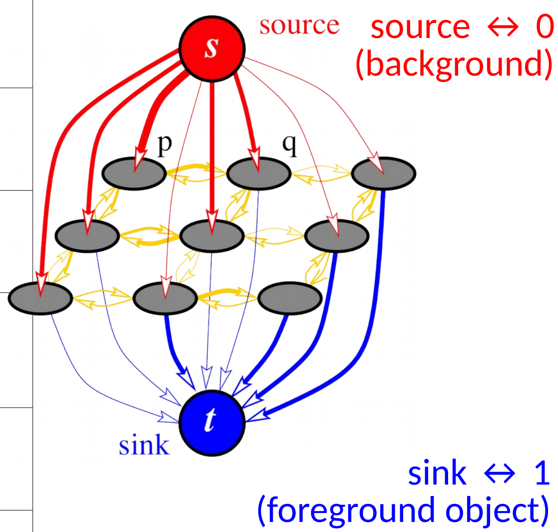
- Pb: pixels of object/background samples not necessarily assigned with good label !

Graph-cut formulation (version 2)

[Boykov & Jolly 2001]

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

Edge	Weight	Sites
(p,q)	$\lambda\, B_{p,q}$	$(p,q) \in$ N
(s,p)	$D_p(1)$	$p \in$ P, $p \notin$ (O ∪ B)
	K	$p \in$ B
	0	$p \in$ O
(p,t)	$D_p(0)$	$p \in$ P, $p \notin$ (O ∪ B)
	0	$p \in$ B
	K	$p \in$ O

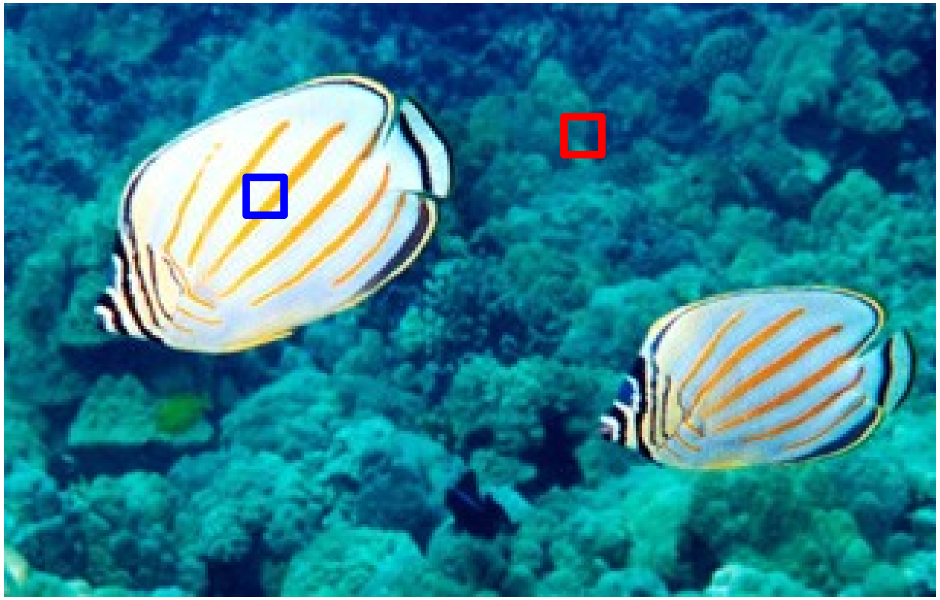


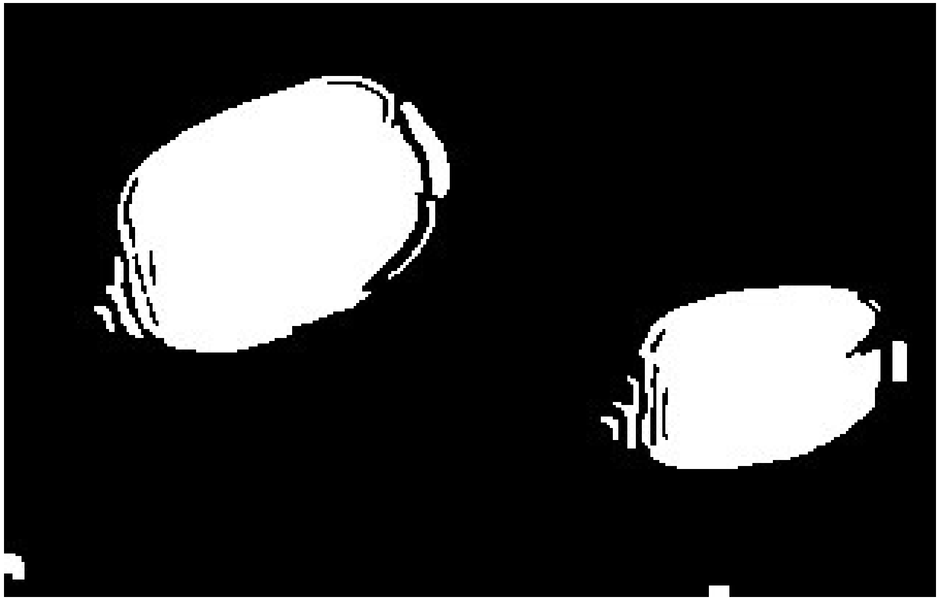
– where $K = 1 + \max_{p\in \mathrm{P}} \lambda \sum_{(p,q)\in \mathrm{N}} B_{p,q}$
$K \approx +\infty$, i.e., too expensive to pay ⇒ label never assigned

To go further on this subject

Some limitations (here with simple color model)

- Is the segmentation OK ?

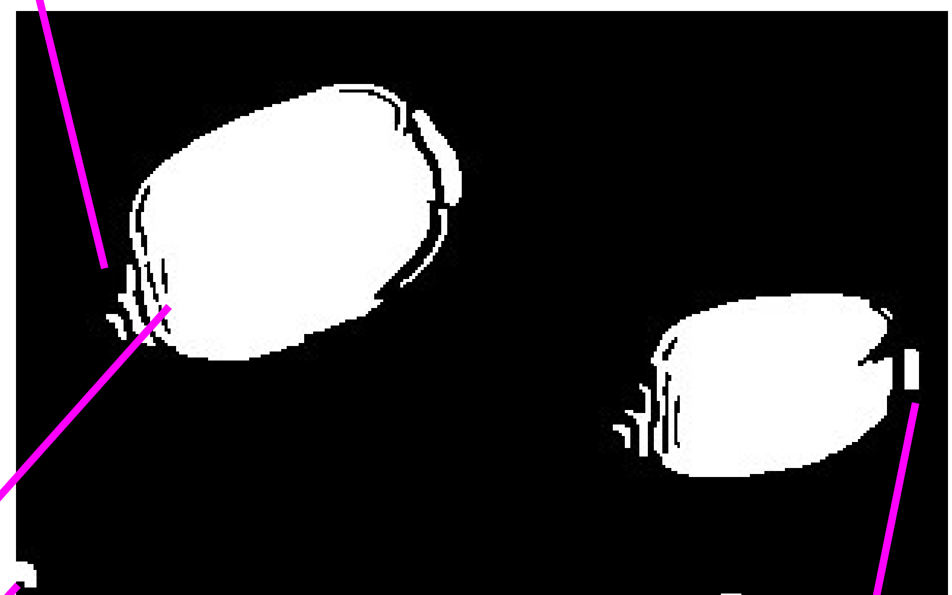




To go further on this subject

Some limitations (here with simple color model)

color model not complex enough

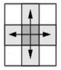
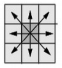


sensitivity to regularization parameter

neighboring model not complex enough

To go further on this subject

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 $B_{p,q}$ expressions defined in course
  - model color of object/background with 1 single gaussian centered around mean color
    - write $\Pr(I_p \,|\, f_p)$ w.r.t. mean color of obj/bg: what is $D_p(f_p)$ then ?
- Compute max flow, extract cut, display image segments
- Experiment with various parameters (λ, σ, c, ...)
- And **comment what you observe**

Part 2

Multi-label problems

Exact vs approximate solutions

Application to stereovision
(disparity/depth map estimation):
disparity/depth ↔ label

Two-label (binary) problem

- P : set of sites (pixels, voxels…)
- N : set of neighboring site pairs
- L $= \{0,1\}$: binary labels
- f : P $\rightarrow$ L binary labeling [notation: $f_p = f(p) = l$]
- $E : (P \rightarrow L) \rightarrow \mathbb{R}$: energy
  - $E(f) = \underbrace{\sum_{p \in \mathsf{P}} D_p(f_p)}_{E_{\text{data}}(f)} + \underbrace{\sum_{(p,q) \in \mathsf{N}} V_{p,q}(f_p, f_q)}_{E_{\text{regul}}(f)}$
    - $D_p(l)$: 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

- $E(f) = \sum_{p \in \mathsf{P}} D_p(f_p) + \sum_{(p,q) \in \mathsf{N}} V_{p,q}(f_p, f_q)$
- $D_p(l)$: label penalty for site p
  - small/null for preferred label, large for undesired label
  - assumption $D_p(l) \geq 0$ (else add constant ➔ same optimum)
- $V_{p,q}(l,l')$: prior knowledge on optimal pairwise labeling
  - in general, smoothness: non-decreasing function of $\mathbf{1}(l \neq l')$
    - e.g., $V_{p,q}(l,l') = u_{p,q}\, \mathbf{1}(l \neq l')$ [Potts model]
- 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

disparity = disparité

- P : set of sites (pixels, voxels...)
- N : set of neighboring site pairs
- L : finite set of labels (→ can model scalar or even vector)
  - e.g., **discretization** of intensity, stereo disparity, motion vector...
- f : P → L labeling
- $E : (P \rightarrow L) \rightarrow \mathbb{R}$: energy
  - $E(f) = \sum_{p \in \mathrm{P}} D_p(f_p) + \sum_{(p,q) \in \mathrm{N}} V_{p,q}(f_p, f_q) = E_{\mathrm{data}}(f) + E_{\mathrm{regul}}(f)$
    - $D_p(l)$: 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

smoothness = lissage

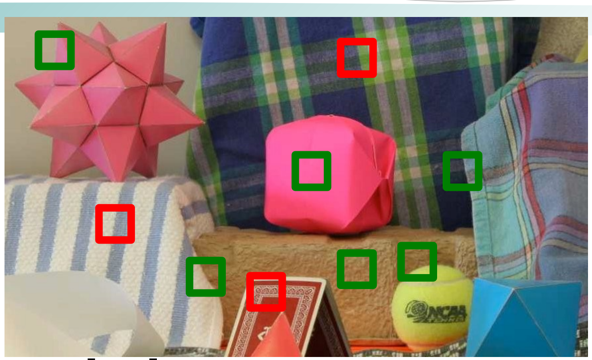
- $E(f) = \sum_{p \in \mathsf{P}} D_p(f_p) + \sum_{(p,q) \in \mathsf{N}} V_{p,q}(f_p, f_q)$
- $D_p(l)$: label penalty for site p
  - small for preferred label, large for undesired label
  - assumption $D_p(l) \geq 0$ (else add constant ➔ same optimum)
- $V_{p,q}(l_p, l_q)$: prior knowledge on optimal 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

Graph cuts for "general" energy minimization

- Problem: find labeling $f^*: \mathsf{P} \rightarrow \mathsf{L}$ minimizing energy
  $E(f) = \sum_{p \in \mathsf{P}} D_p(f_p) + \sum_{(p,q) \in \mathsf{N}} V_{p,q}(f_p, f_q)$
- Question: can a **globally optimal** labeling f^* be found using some graph-cut construction?
- Answer:
  - binary labeling: yes <u>iff</u> $V_{p,q}$ is regular (Kolmogorov & Zabih 2004)
    $V_{p,q}(0,0) + V_{p,q}(1,1) \leq V_{p,q}(0,1) + V_{p,q}(1,0)$ [otherwise NP-hard]
  - multi-labeling: yes if $V_{p,q}$ convex (Ishikawa 2003)
    and if L linearly ordered (⇒ 1D only ⇒ not 2D motion vector)
  - otherwise: approximate solutions (but some very good)

Piecewise-smooth vs everywhere-smooth

piecewise = par morceaux



- Observation: object properties often smooth everywhere except on boundaries

☛ Consequence: <u>piecewise-smooth</u> models more appropriate than <u>everywhere-smooth</u> models


original

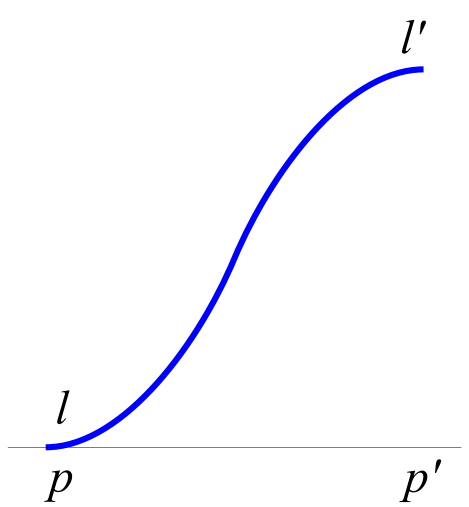

uniform smoothing


piecewise smoothing

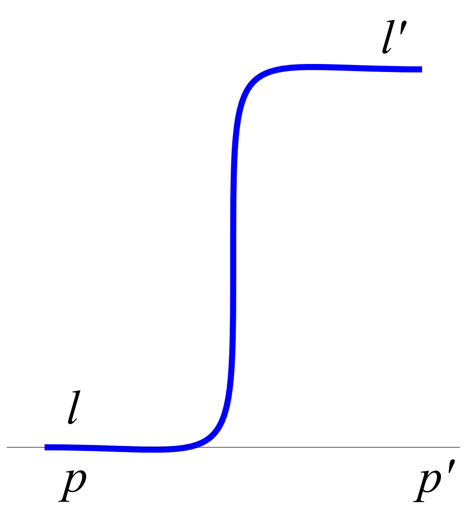
Piecewise-smooth models vs everywhere-smooth models

steep = raide, très pentu

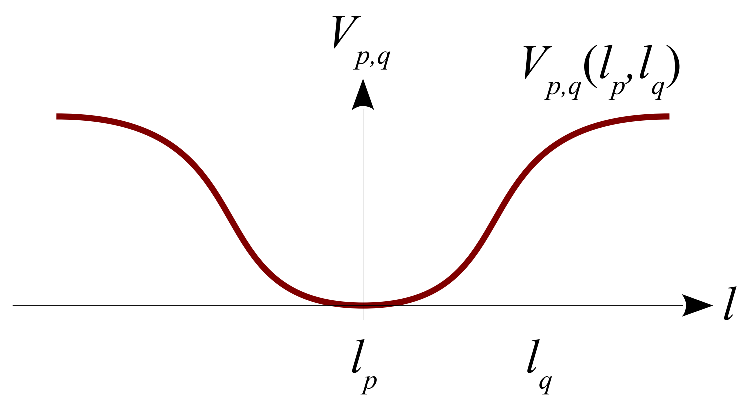
- Local variation of potentials $V_{p,q}$ depending on label variation



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



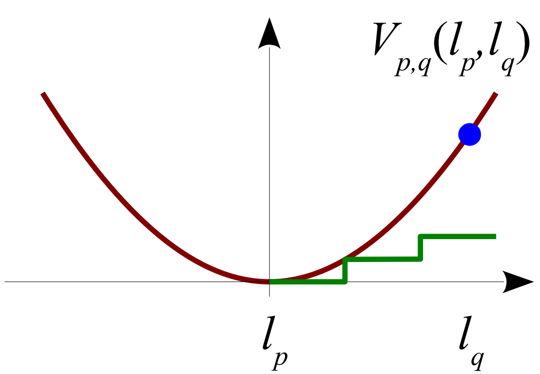
locally steep from l to l' when going from p to p'



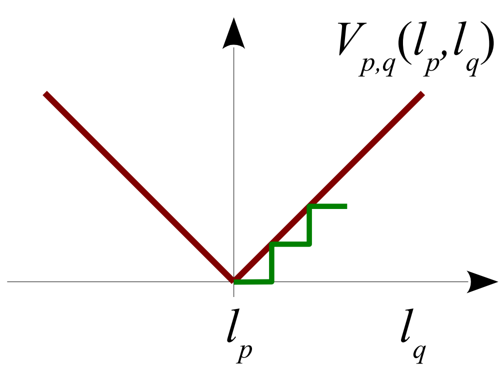
piecewise-smooth potential

Piecewise-smooth potentials vs everywhere-smooth potentials

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

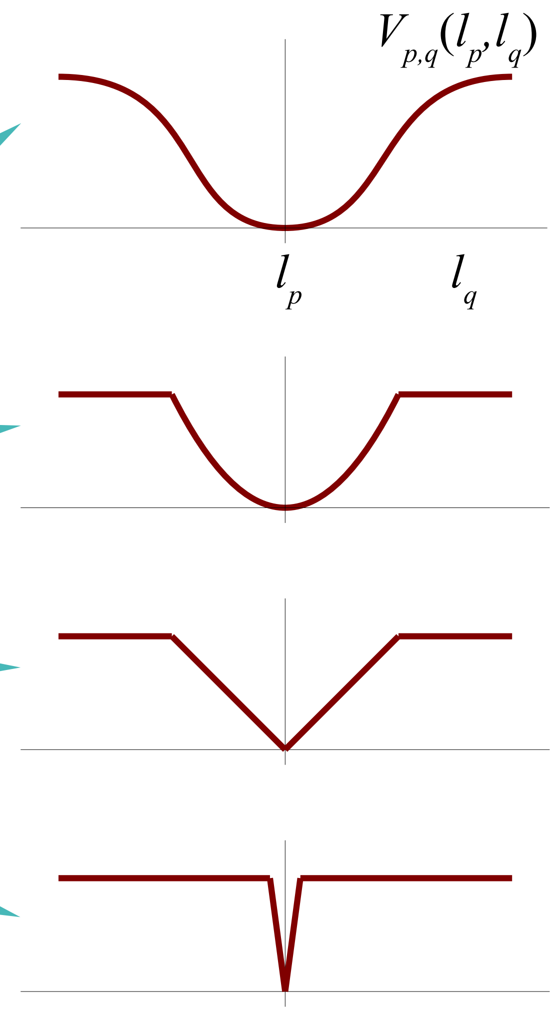


In practice, best results with “least convex” function, e.g., $V_{p,q}(l_p,l_q) = \lambda_{p,q} \,||\, l_p - l_q \,||$



Discontinuity-preserving energy

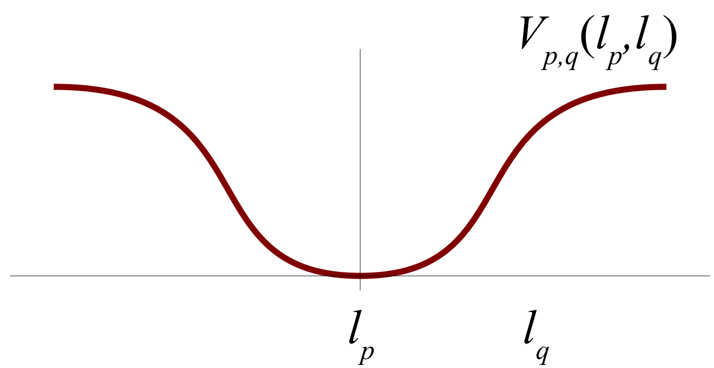
- At edges, very different labels for adjacent pixels are OK
- To not overpenalize in E adjacent but very different labels:
  - $V_{p,q}$ non-convex function of $\| l_p - l_q \|$
  - for instance (cap max):
    - $V_{p,q} = \min(K, \| l_p - l_q \|^2)$
    - $V_{p,q} = \min(K, \| l_p - l_q \|)$
    -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



Difficulty of minimization

simulated annealing = recuit simulé

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



- $\min_f E(f)$: minimization of non-convex function in large-dimension space (dimension = $|\mathrm{P}|$)
  - NP-hard even in simple cases
    - e.g. $V_{pq}(f_p, f_q) = \mathbf{1}(f_p \neq f_q)$ (Potts model) with $|\mathrm{L}| > 2$
  - general case: simulated annealing...

Exact binary optimization (<u>reminder</u>)

- Pb: find labeling $f^*: \mathsf{P} \to \mathsf{L} = \{0,1\}$ minimizing energy
  $E(f) = \sum_{p\in\mathsf{P}} D_p(f_p) + \sum_{(p,q)\in\mathsf{N}} V_{p,q}(f_p, f_q)$
- Question:
  - can a **globally optimal** labeling f^* be found using some graph-cut construction?
- Answer (Kolmogorov & Zabih 2004):
  - yes <u>iff</u> V_{pq} is regular
    - $V_{p,q}(0,0) + V_{p,q}(1,1) \le V_{p,q}(0,1) + V_{p,q}(1,0)$
  - otherwise it's NP-hard
- But what about **general energies** on **binary** variables ?

Exact binary optimization

[Kolmogorov & Zabih 2004]

- Question:
  - what functions can be minimized using graph cuts?
- Classes of functions on binary variables:
  - F^2: $E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j)$
  - F^3: $E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j) + \sum_{i<j<k} E^{i,j,k}(x_i, x_j, x_k)$
  - F^m: $E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \ldots + \sum_{u_1<\ldots<u_m} E^{u_1,\ldots,u_m}(x_{u_1}, \ldots, x_{u_m})$

***m*-th order potentials**

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

To go further on this subject

Exact binary optimization

[Kolmogorov & Zabih 2004]

- E regular iff
  - F ^{2}: $\forall i,j \;\; E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0)$
  - 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):
  - F ^{2}, F ^{3}: E graph-representable $\Leftrightarrow$ E regular
  - any binary E: E not regular $\Rightarrow$ E not graph-representable

To go further on this subject

Link with submodularity

submodular = sous-modulaire

- $g : 2^{\mathrm{P}} \to \mathbb{R}$ submodular
  - iff $g(X) + g(Y) \geq g(X \cup Y) + g(X \cap Y)$

    for any $X, Y \subset \mathrm{P}$
  - iff $g(X \cup \{j\}) - g(X) \geq g(X \cup \{i,j\}) - g(X \cup \{j\})$

    for any $X \subset \mathrm{P}$ and $i, j \in \mathrm{P} \setminus X$
- g submodular $\Leftrightarrow$ E regular, with $E(\mathbf{x}) = g(\{p \in \mathrm{P} \mid x_p = 1\})$
  - $E^{i,j}(0,1) + E^{i,j}(1,0) \geq E^{i,j}(0,0) + E^{i,j}(1,1)$
- $\exists$ independent results on submodular functions
  - minimization in polynomial time but slow, best known $O(n^6)$

Exact multi-label optimization (for 2nd-order potentials)

- Problem: find labeling f^* : P → L minimizing energy

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

- Assumption: L linearly ordered — w.l.o.g. 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)

linearly ordered = linéairement ordonné
w.l.o.g. = without loss of generality

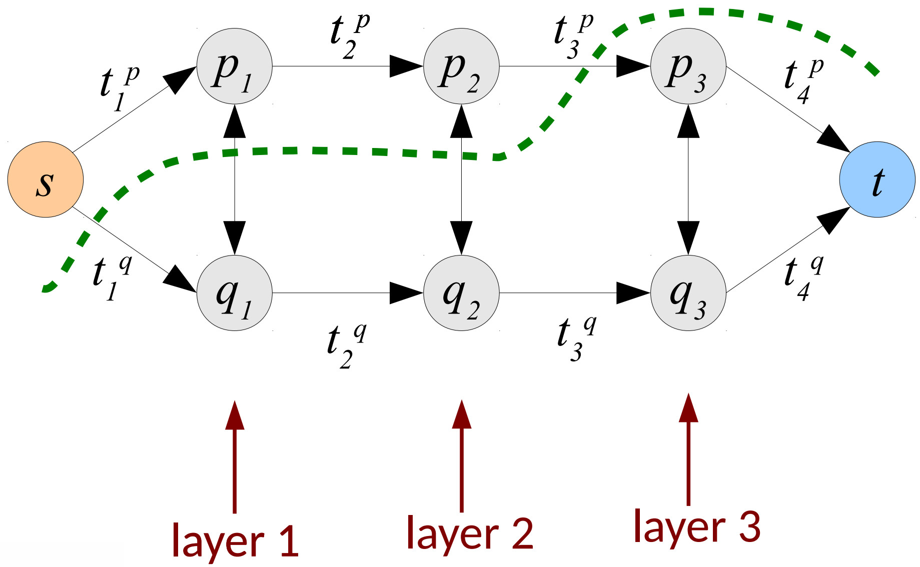
Linear multi-label graph construction

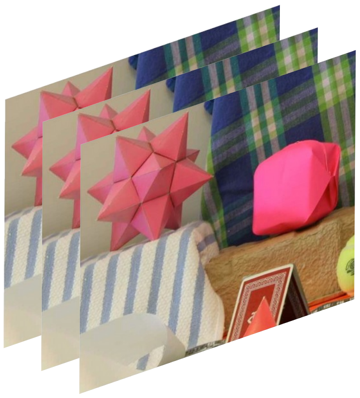
(cf. Boykov et al. 1998)

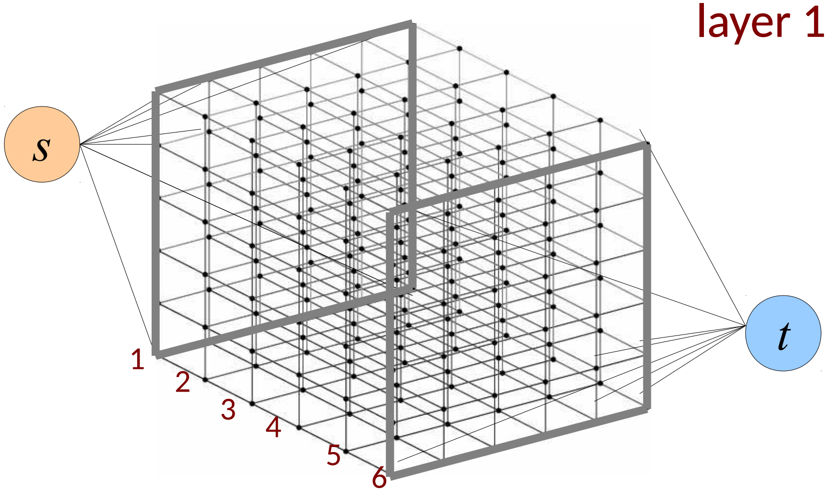
- Given L $= \{1,\ldots,k\}$
- General idea:
  - construct one layer per label value
  - read label value from cut location

e.g., $k = 4$

cut: $f_p\!=\!3, f_q\!=\!1$







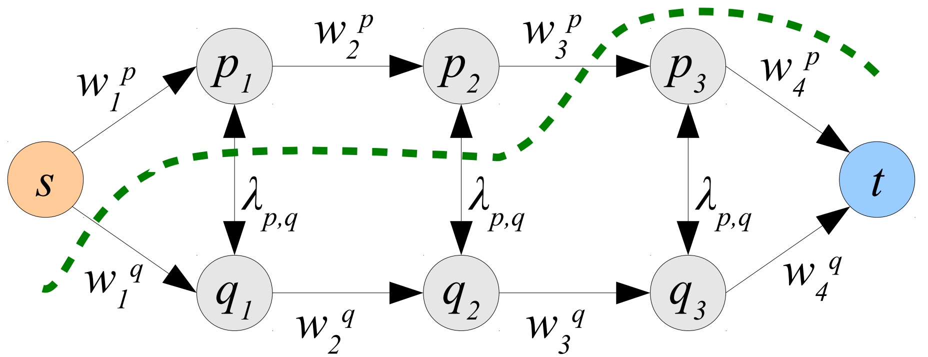
layer = couche

Linear multi-label graph construction

(cf. Boykov et al. 1998)

- $E(f) = \sum_{p\in\mathsf{P}} D_p(f_p) + \sum_{(p,q)\in\mathsf{N}} \lambda_{p,q} |f_p - f_q|$ with $f_p \in \mathsf{L} = \{1,\dots,k\}$

cut: $f_p = 3, f_q = 1$



Attempt 1:

- For each site p
  - create nodes $p_1,\dots,p_{k-1}$
  - 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 $(p_j,q_j)_{j\in\{1,\dots,k-1\}}$ with weight $\lambda_{p,q}$
- Read label value from cut location, e.g., $p_2 \in \mathsf{S}$, $p_3 \in \mathsf{T} \Rightarrow f_p = 3$

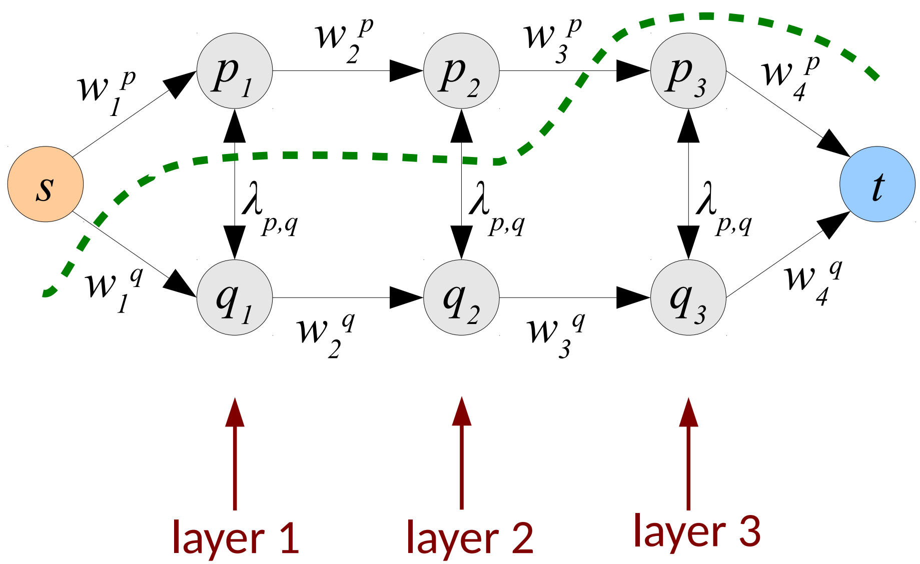
Linear multi-label graph construction

(cf. Boykov et al. 1998)

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

e.g., $k = 4$

cut: $f_p\!=\!3, f_q\!=\!1$



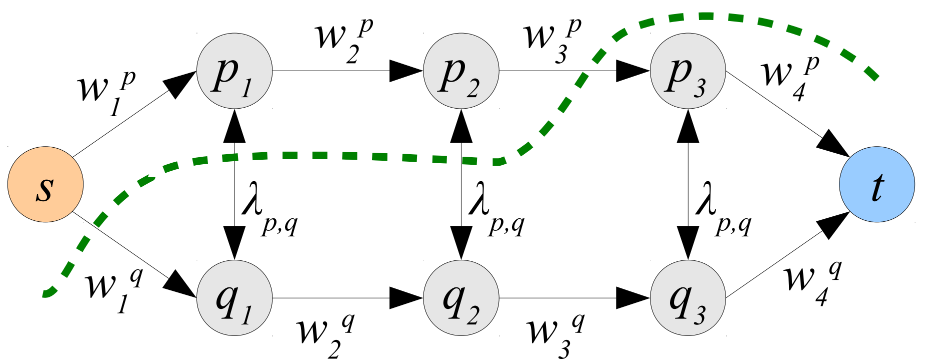
- Any problem ?

Linear multi-label graph construction

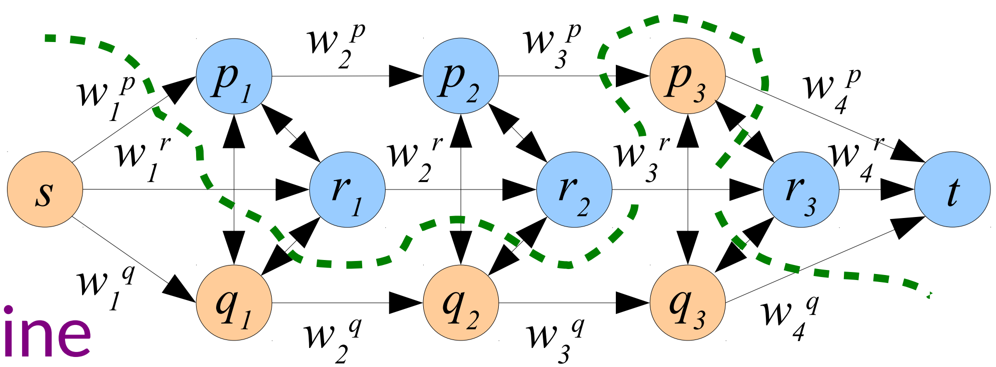
(cf. Boykov et al. 1998)

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

e.g., $k = 4$ cut: $f_p=3, f_q=1$



- Any problem ?
  - there could be several cut locations on the same line

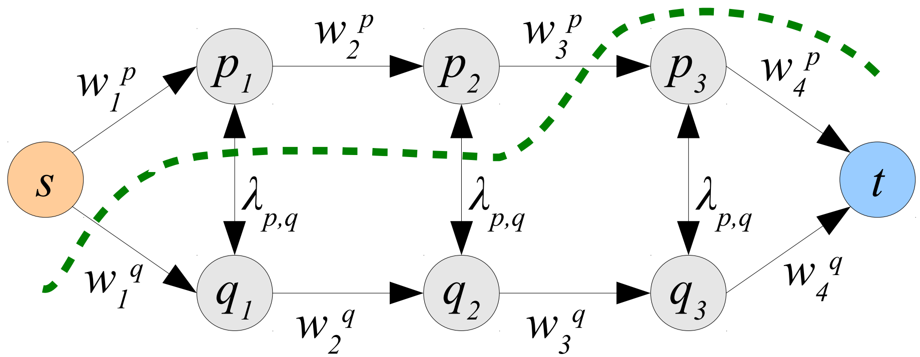


Linear multi-label graph construction

(cf. Boykov et al. 1998)

- $E(f) = \sum_{p \in \mathrm{P}} D_p(f_p) + \sum_{(p,q) \in \mathrm{N}} \lambda_{p,q} \, |f_p - f_q|$ cut: $f_p = 3, f_q = 1$
  with $f_p \in \mathrm{L} = \{1,\ldots,k\}$

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 $K_p = 1 + (k-1) \sum_{q \in \mathrm{N}_p} \lambda_{p,q}$ (where 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 $\lambda_{p,q}$

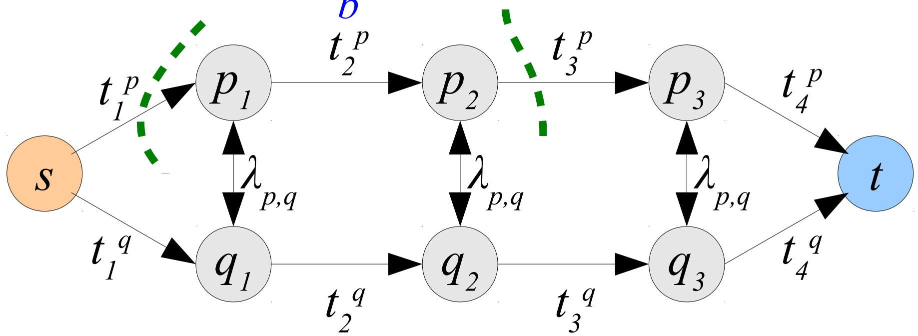
Linear multi-label graph properties

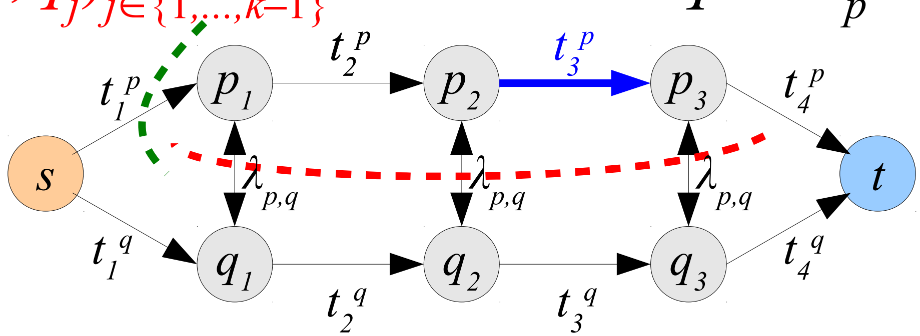
(cf. Boykov et al. 1998)

- Lemma: for each site p, a minimum cut severs exactly one t_j^p
  - [≥ 1] Any cut severs at least one t_j^p

severed = coupé, sectionné

  - [≤ 1] Suppose t_a^p, t_b^p are cut (same line p), then build new cut with t_b^p restored and links $(p_j,q_j)_{j\in\{1,\ldots,k-1\}}$ broken for $q \in \mathrm{N}_p$





Impact on (minimum) cost: $-\,w(t_b^p) + (k-1)\sum_{q\in\mathrm{N}_p} \lambda_{p,q}$

$= -\,D_p(j) - 1 < 0$ → strictly smaller cost → contradiction

- Theorem (Boykov et al. 1998): a minimum cut minimizes $E(f\,)$

Application to stereovision: disparity map estimation

rectified images ↔ aligned cameras

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

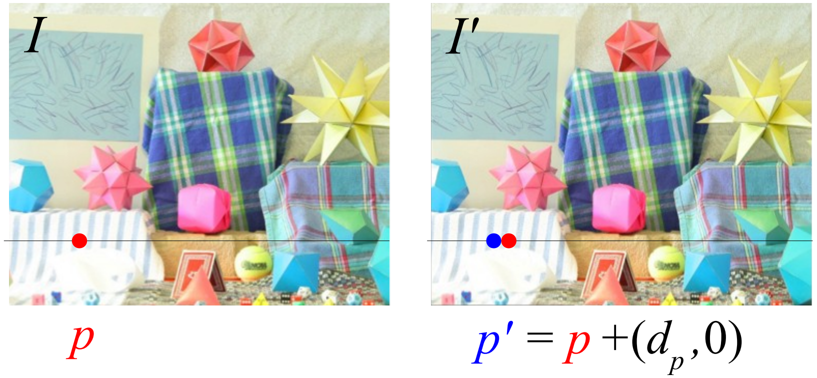
p

$p' = p + (d_p, 0)$

- Graph-cut setting
  - discrete disparities: $d_p \in \mathrm{L} = \{d_{\min}, \ldots, d_{\max}\}$
  - 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: $d_p \in \mathrm{L} = \{d_{\min}, \ldots, d_{\max}\}$
  - data term: $D_p(d_p)$
    - small when pixel p in I similar to pixel $p' = p + (d_p, 0)$ in I'

e.g., what definition?

  - smoothness term: $V_{p,q}(d_p\ ,\ d_q)$
    - small when disparities d_p and d_q are similar

e.g., what definition?

Application to stereovision: disparity map estimation

$\bar{I}_P = 1/|P| \sum_{q \in P} I_q$ $\sigma = [1/|P| \sum_{q \in P} (I_q - \bar{I}_P)^2]^{1/2}$

$E_{ZNSSD}(P\,;\boldsymbol{u}) = 1/|P| \sum_{q \in P} [(I'_{q+\boldsymbol{u}} - \bar{I}'_P)/\sigma' - (I_q - \bar{I}_P)/\sigma]^2$

- 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: $d_p \in \mathrm{L} = \{d_{min}, \ldots, d_{max}\}$
  - data term: $D_p(d_p)$
    - e.g., $D_p(d_p) = E_{ZNSSD}(P_p\,; (d_p, 0))$ where P_p patch around pixel p
  - smoothness term: $V_{p,q}(d_p\,,\, d_q)$
    - e.g., $V_{p,q}(d_p\,,\, d_q) = \lambda\, |d_p - d_q|$

SSD = sum of square differences
NSSD = normalized ...
ZNSSD = zero-normalized ...

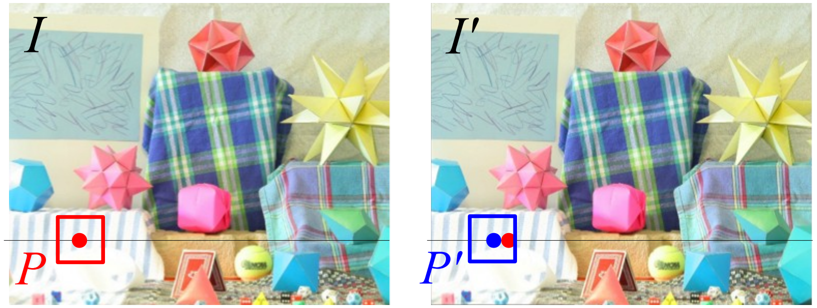
e.g., what definition?

e.g., what definition?

[Boykov et al. → 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 $p = (u,v)$



p $\quad p' = p + (d_p, 0)$

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

Is it the "optimal" solution to the disparity map estimation problem ?

[Boykov et al. → 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 $p = (u,v)$

p

$p' = p + (d_p, 0)$

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

- Meaningful but arbitrary choices: patch size, similarity, smoothness…
- Optimal solution for energy $\not\Rightarrow$ optimal solution for problem

[Boykov et al. → optimal disparities]

Application to stereovision: disparity map estimation

CC = cross-correlation
NCC = normalized ...
ZNCC = zero-normalized ...

$\bar{I}_P = 1/|P| \sum_{q \in P} I_q$ $\quad \sigma = [1/|P| \sum_{q \in P} (I_q - \bar{I}_P)^2]^{1/2}$

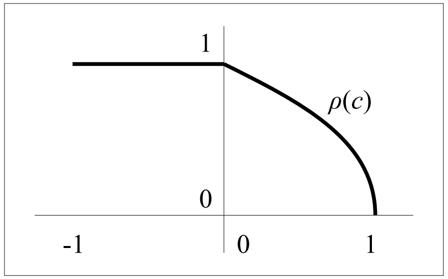
$E_{ZNCC}(P\,;\boldsymbol{u}) = 1/|P| \sum_{q \in P} [(I'_{q+\boldsymbol{u}} - \bar{I}'_P)/\sigma' . (I_q - \bar{I}_P)/\sigma]$ $\quad E_{ZNSSD}(P\,;\boldsymbol{u}) = 2 - 2\,E_{ZNCC}(P\,;\boldsymbol{u})$

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

p

$p' = p + (d_p, 0)$

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



dissimilar similar equal

N.B. only w_{cc} / λ matters

Approximate optimization

bound = borne

- 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)

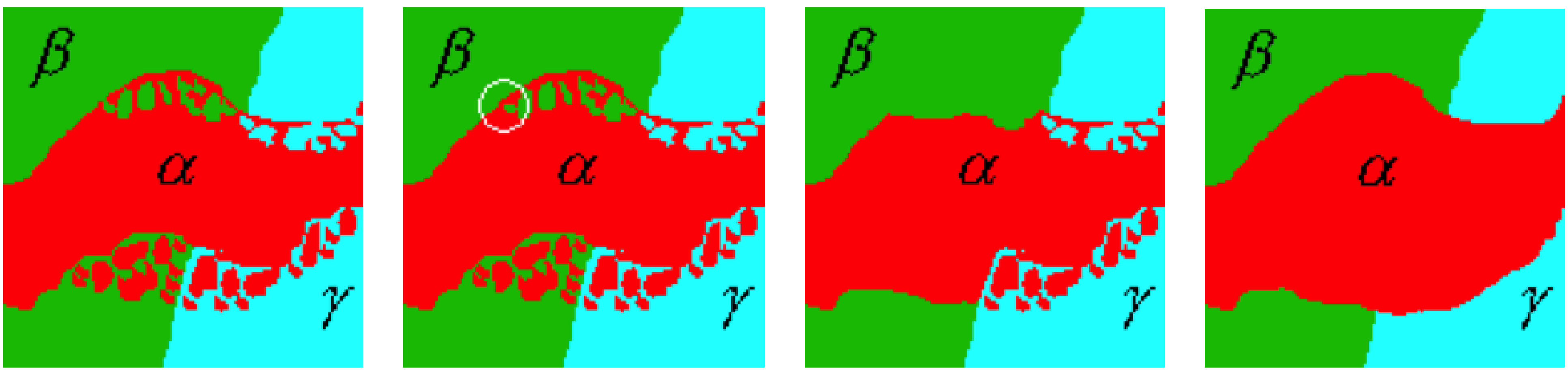
- → local minimum ☹
- but within known bounds of global minimum ☺

Notion of move — Examples

at once = à la fois
move = déplacement
(≈ modification)
de la solution

Move: maps a labeling $f : \mathrm{P} \to \mathrm{L}$ to a labeling $f' : \mathrm{P} \to \mathrm{L}$

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



(a) initial labeling

(b) standard move
$\alpha \to \beta$
at <u>one site</u>
only

(c) α-β-swap
$\alpha \leftrightarrow \beta$
at <u>many sites</u>
at once

(d) α-expansion
any $l \to \alpha$
at <u>many sites</u>
at once

Moves

move = déplacement
(≈ modification)
de la solution
α-β-swap =
permutation α-β

Given a labeling $f : \mathrm{P} \rightarrow \mathrm{L}$ and labels α, β

- f' is a **standard move** from f iff

  f and f' differ at most on one site p

- f' is an **expansion move** (or ***α*-expansion**) from f iff

  $\forall p \in \mathrm{P},\ \ f'_p = f_p$ or α

  → in f', compared to f, extra sites p can now be labeled α

- f' is a **swap move** (or ***α*-β-swap**) from f iff

  $\forall p \in \mathrm{P},\ \ f_p \neq \alpha, \beta \ \Rightarrow\ f_p = f'_p$

  → some sites that were labeled α are now β and vice versa

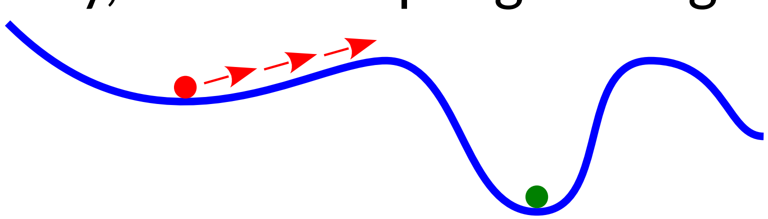
N.B. Other kinds of moves can be defined…

Optimization w.r.t. moves

simulated annealing = recuit simulé
sampling = échantionnage

(cf. Boykov et. al 2001)

- Iterative optimization over moves
  - random cycle over all labels until convergence → 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
      ⇒ 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

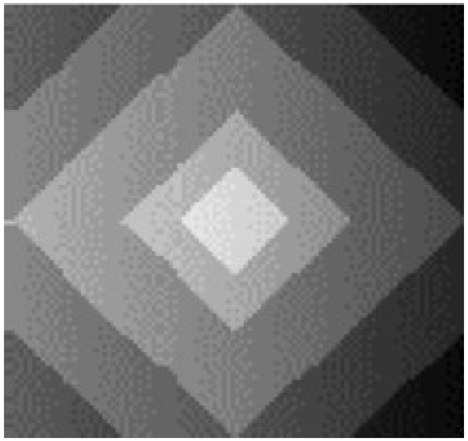
(cf. Boykov et. al 2001)

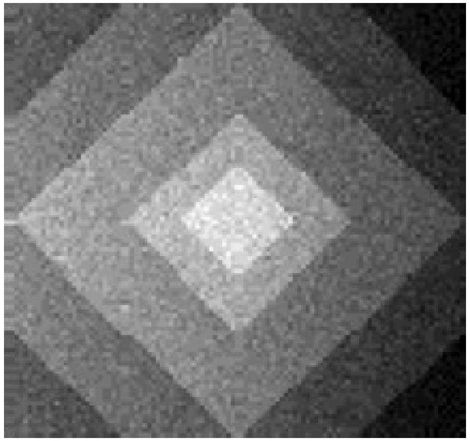
- Iterative optimization over moves
  - random cycle over all labels until convergence → local min
- **Iterating expansion/swap moves** (<u>strong</u> moves)
  - number of possible moves exponential in number of sites
  - compute optimal move using graph cut **= binary problem!**
    - see Boykov et. 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 → optimum within factor 2

Image restoration with moves

- Restoration with standard moves vs α-expansions

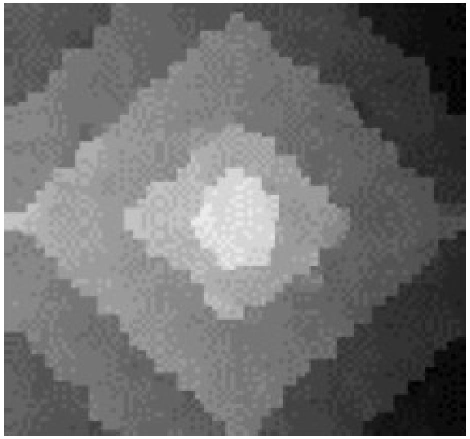
original image

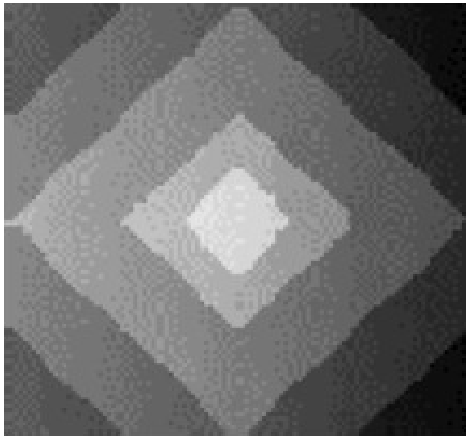




noisy image

restoration with standard moves





restoration with α-expansions

Boykov et al. 2001 © IEEE

Constraints on interaction potential

(see details in Boykov et. al 2001)

- Expansion move: V metric, → expansion inequality:

metric = métrique
(= fonct distance)
d(x,y) = 0 ⇔ x = y
d(x,y) = d(y,x) ≥ 0
d(x,z) ≤ d(x,y)+d(y,z)

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

- Swap move: V semi-metric, → swap inequality:

semi-metric =
semimétrique
d(x,y) = 0 ⇔ x = y
d(x,y) = d(y,x) ≥ 0

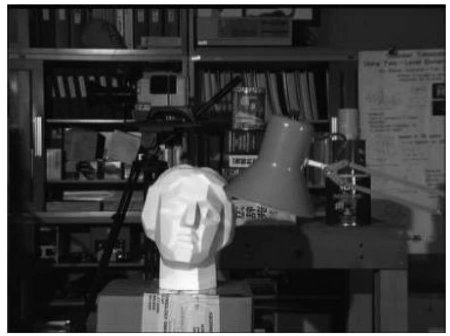
$$V_{p,q}(\alpha,\alpha) + V_{p,q}(\beta,\beta) \leq V_{p,q}(\alpha,\beta) + V_{p,q}(\beta,\alpha) \quad \text{for all } \alpha,\beta \in \mathrm{L}$$

[= as metric but triangle inequality not required: $V_{p,q}(\alpha,\gamma) \leq V_{p,q}(\alpha,\beta) + V_{p,q}(\beta,\gamma)$]
[weaker condition than for expansion move]

discontinuity-preserving!

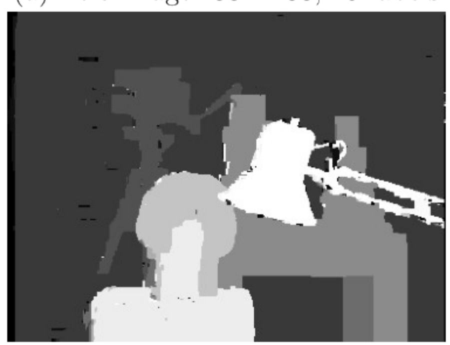
- Examples
  – Potts model: $V_{p,q}(\alpha,\beta) = \lambda_{p,q}\ \mathbf{1}(\alpha \neq \beta)$
  – truncated L_2 distance: $V_{p,q}(\alpha,\beta) = \min(K, \|\alpha-\beta\|)$

Disparity map estimation with moves

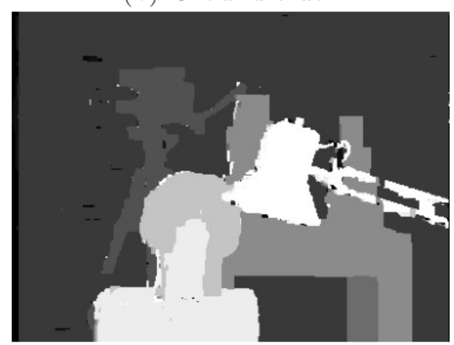


(a) Left image: 384x288, 15 labels

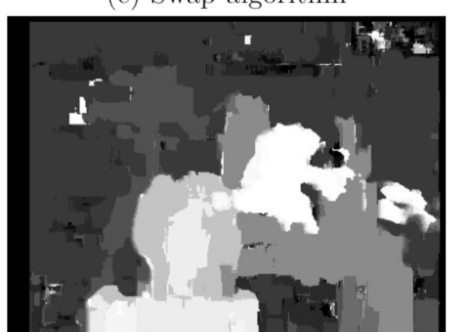
(b) Ground truth



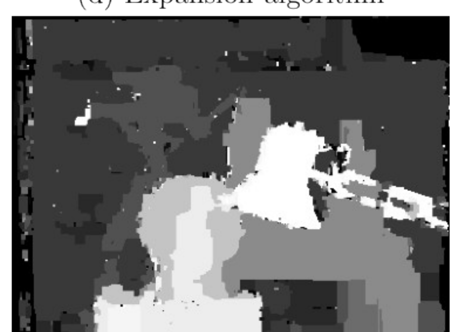
(c) Swap algorithm



(d) Expansion algorithm



(e) Normalized correlation



(f) Simulated annealing

Tsukuba images from famous Middlebury benchmark (also contains Moebius images)

Boykov et al. 2001 © IEEE

To go further on this subject

Disparity map estimation: alternative data term

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

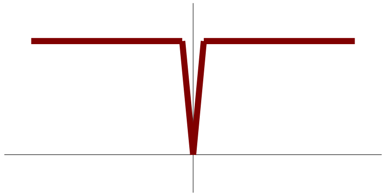
- Idea: direct intensity comparison, but sensitive to sampling
  - $D_p(d_p) = \min(K\,,\ |I_p - I'_{p+d_p}|^2)$
- With image sampling insensitivity:
  - disparity range discretized to 1 pixel accuracy
    → sensitivity to high gradients
  - (sub)pixel dissimilarity measure for greater accuracy, e.g.,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

No patch similarity here: the local consistency is given by the smoothness term

Disparity map estimation: smoothness term

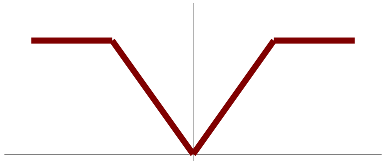
- Scene with fronto-parallel objects
  - **piecewise-constant** model = OK
  - e.g., Potts model:

$$V_{p,q}(d_p,d_q) = u_{p,q}\ \mathbf{1}(d_p \neq d_q)$$



- Scene with slanted surfaces (e.g., ground)
  - **piecewise-smooth** model = better
  - e.g., smooth cap max value:

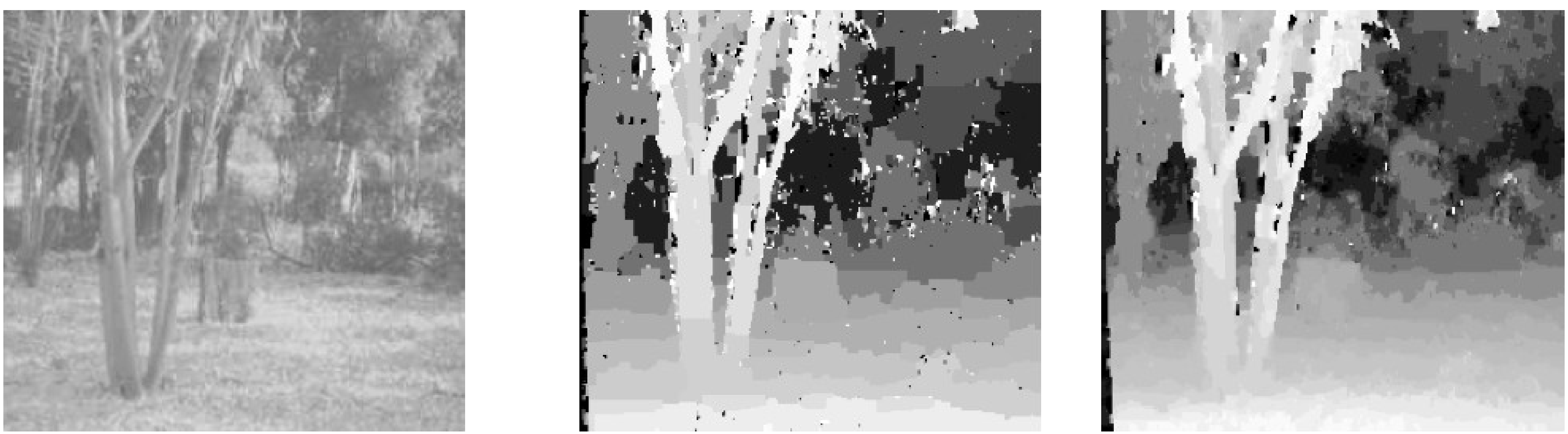
$$V_{p,q} = \lambda \min(K, |d_p - d_q|)$$



- Metric ⇒ both swap and expansion algorithms usable

Potts model vs smooth cap max value

- Potts model : **piecewise-constant**
  - suited for uniform areas (⇒ 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

To go further on this subject

Disparity map estimation: smoothness term

(cf. Boykov et al. 1998, Boykov et al. 2001)

- Contextual information
  - neighbors p,q more likely to have same disparity if $I_p \approx I_q$
    → 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.,

$$U(|I_p - I_q|) = \begin{cases} 2K & \text{if } |I_p - I_q| \leq 5 \\ K & \text{if } |I_p - I_q| > 5 \end{cases}$$

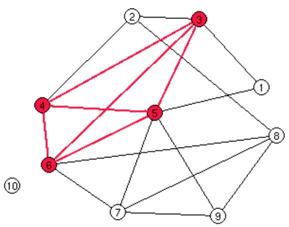
Many extensions to more complex energies

(cf. Pansari & Kumar 2017)

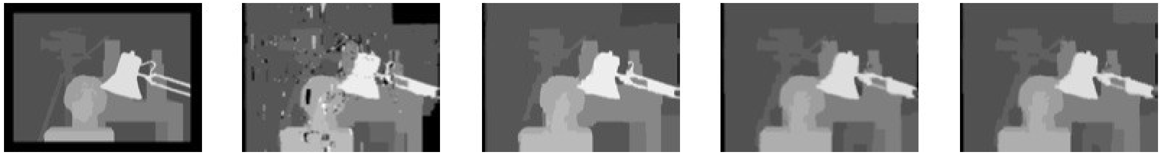
- Truncated Convex Models (TCM)
  - several other approximate algorithms to minimize

$$E(\mathbf{x}) = \sum_{a \in \mathcal{V}} \theta_a(x_a) + \sum_{(a,b) \in \mathcal{E}} \omega_{ab} \min\{d(x_a - x_b), M\}$$

- Truncated Max of Convex Models (TMCM)
  - no clique size restriction (high-order > pairwise)



$$\theta_{\mathbf{c}}(\mathbf{x_c}) = \omega_{\mathbf{c}} \sum_{i=1}^{m} \min\{d(p_i(\mathbf{x_c}) - p_{c-i+1}(\mathbf{x_c})), M\}$$

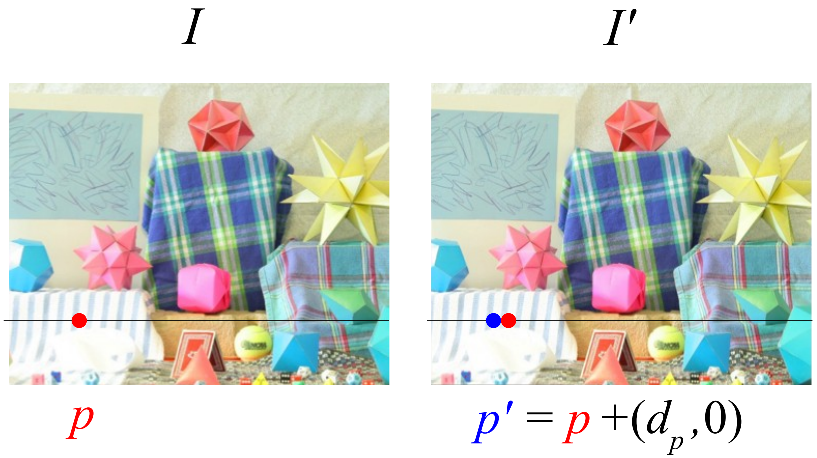


(a) Ground truth (Energy, Time (s)) | (b) Cooccurrence (2098800, 101) | (c) Parsimonious (1364200, 225) | (d) $m = 1, h' = 4$ **(1257249, 256)** | (e) $m = 3, h' = 4$ (1267449*, 335)

$\mathbf{c}$: clique
$\mathbf{x_c}$: labeling of a clique
$\omega_{\mathbf{c}}$: clique weight
d : convex function
M : truncation factor
$p_i(\mathbf{x_c})$: i-th largest label in $\mathbf{x_c}$
$c = |\mathbf{c}|$

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)

occlusion = occultation

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

I

p

I'

$p' = p + (d_p, 0)$

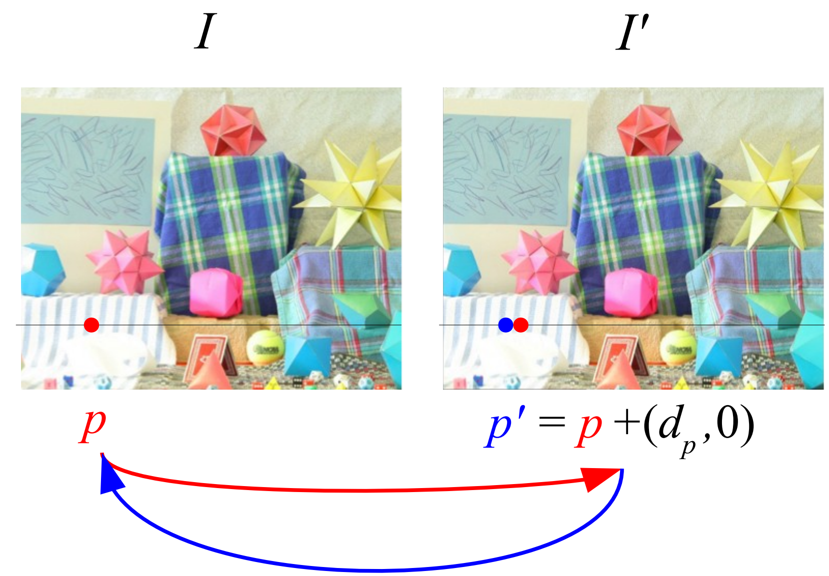
- Are the preceding formulations OK?
  - no treatment of occlusion
  - no symmetry: one center image, one auxiliary image
    - treatment of second image relative to the first (main) one
    - difficulty to incorporate occlusion naturally

Cross-checking

(Bolles & Woodfill, 1993)

occlusion = occultation

- 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

(cf. Kolmogorov & Zabih 2001)

occluded = occulté

- Occlusion
  - pixel visible in one image only
  - occurs usually at discontinuities
- Uniqueness model hypothesis
  - pixel in one image → 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 <u>pairs</u>), not pixel disparity

Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

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

Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

- $N_p(f)$: set of correspondences for pixel p
  - $N_p(f) = \{a \in A(f) \mid \exists p' \in \mathrm{P},\ a = (p,p')\}$
  - configuration f unique iff $\forall p \in \mathrm{P}\ \ |N_p(f)| \leq 1$
  - occluded pixels defined as pixels such that $|N_p(f)| = 0$
- $\mathcal{N}$: a neighborhood system on assignments (used for smoothness term)
  - $\mathcal{N} \subset \{\ \{a_1, a_2\} \subset \mathrm{A}\ \}$
  - for efficient energy minimization via graph cuts:
    - neighbors having the same disparity
    - $\mathcal{N} = \{\ \{(p,p'),(q,q')\} \subset \mathrm{A} \mid p,p'$ are neighbors and $d(p,p') = d(q,q')\}$

      (→ then q,q' are also neighbors)

Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

- $E(f) = E_{\text{data}}(f) + E_{\text{smooth}}(f) + E_{\text{occ}}(f)$
  - $E_{\text{data}}(f) = \sum_{a=(p,p')\in A(f)} (I_p - I'_p)^2$
    - single pixel similarity
  - $E_{\text{smooth}}(f) = \sum_{\{a_1,a_2\}\in \mathrm{N}} V_{a_1,a_2} \mathbf{1}(f_{a_1} \neq f_{a_2})$
    - $N = \{\{(p,p'),(q,q')\} \subset \mathrm{A} \mid p,p'$ 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**
  - $E_{\text{occ}}(f) = \sum_{p\in \mathrm{P}} C_p . \mathbf{1}(|N_p(f)| = 0)$ [occlusion penalty]
    - 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^{α} : set of all assignments with disparity α
    - $A^{\alpha,\beta} = A^{\alpha} \cup A^{\beta}$
  - expansion move:
    - f' within single α-expansion move of f iff $A(f') \subset A(f) \cup A^{\alpha}$
      - currently active assignments can be deleted
      - new assignments with disparity α 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 α or β

Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

f unique $\Leftrightarrow$ $\forall p \in \mathrm{P}\ \ |N_p(f)| \leq 1$

- Expansion-move algorithm:
  1. start with arbitrary, unique configuration f_0
  2. set success ← false
  3. for each disparity α
     1. find $f^\alpha = \operatorname{argmin}_f E(f)$
        subject to f unique and within single α-move of f_0
     2. if $E(f^\alpha) < E(f_0)$, then set f_0 ← f^α, success ← true
  4. if success go to 2
  5. return f_0
- Critical step: efficient computation of α-move with smallest energy

Stereovision with occlusion

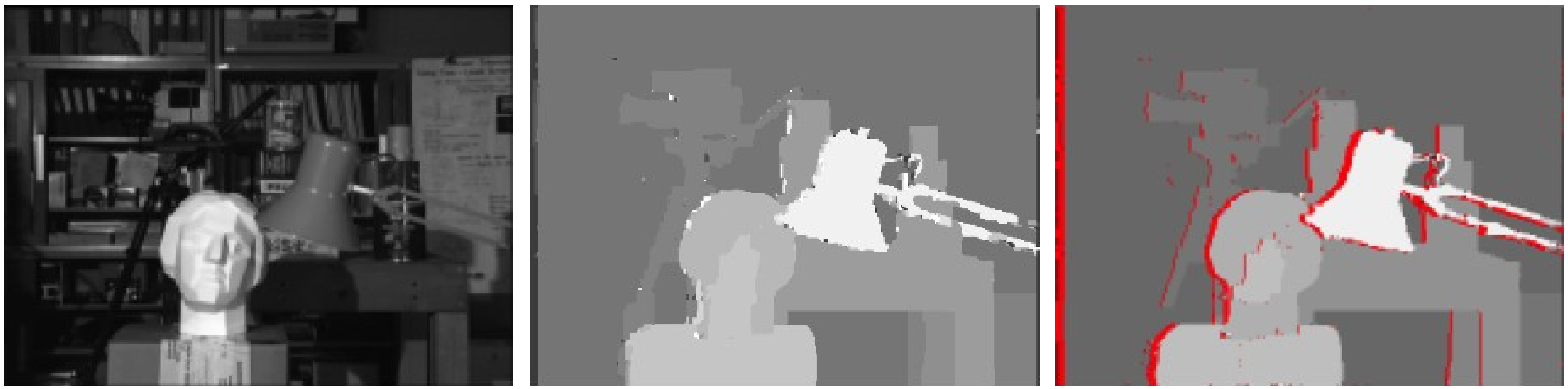
(cf. Kolmogorov & Zabih 2001)

f unique $\Leftrightarrow$
$\forall p \in \mathrm{P} \ \ |N_p(f)| \leq 1$

- Swap-move algorithm:
  1. start with arbitrary, unique configuration f_0
  2. set success ← false
  3. for each pair of disparities $\alpha, \beta \ \ (\alpha \neq \beta)$
     - 3.1. find $f^{\alpha\beta} = \text{argmin}_f E(f)$
       subject to f unique and within single αβ-swap of f_0
     - 3.2. if $E(f^{\alpha\beta}) < E(f_0)$, then set f_0 ← $f^{\alpha\beta}$, success ← true
  4. if success go to 2
  5. return f_0
- Critical step: efficient computation of αβ-swap with smallest energy

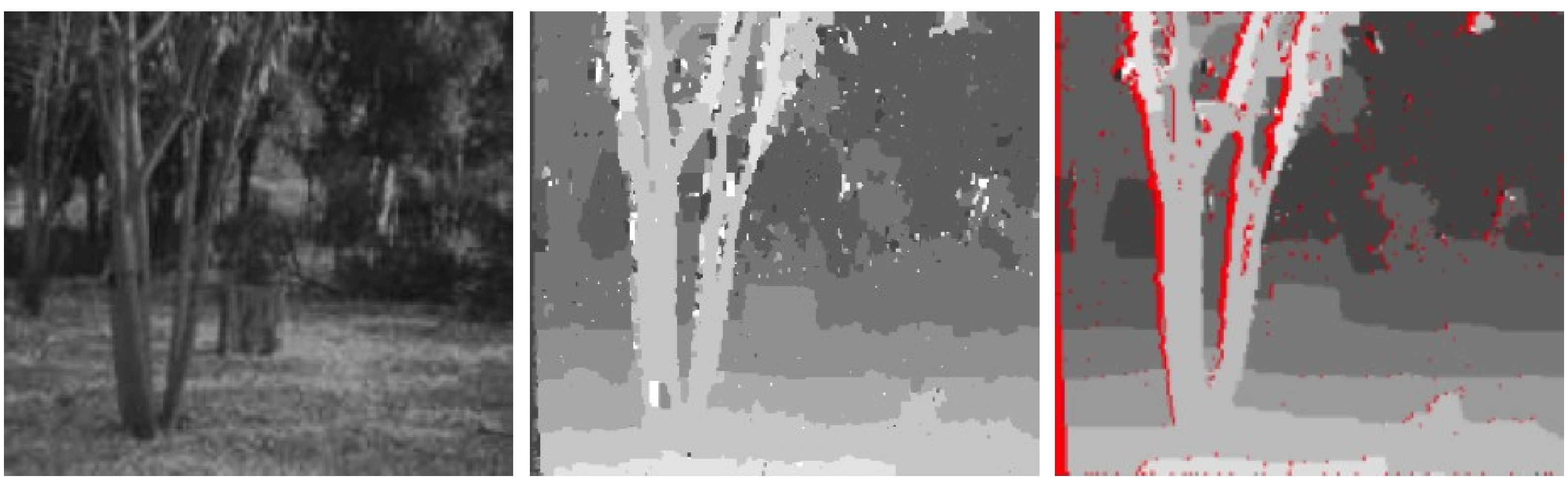
Stereovision with occlusion

(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



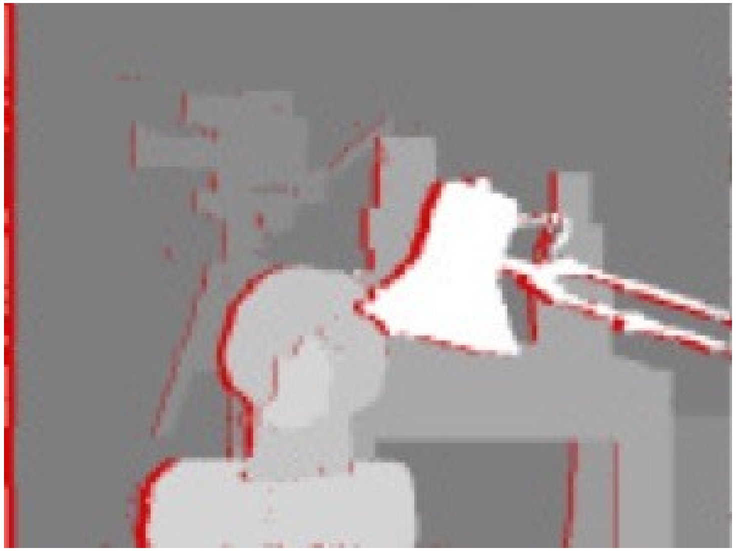
(d) Left image of *Tree* pair (e) Potts model stereo (f) Stereo with occlusions

Disparity maps obtained for the *Tree* pair

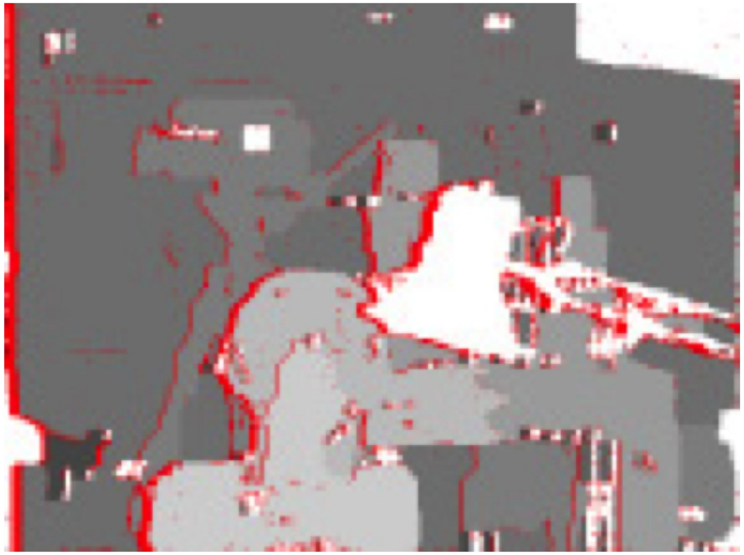
Stereovision with occlusion

(cf. Kolmogorov & Zabih 2001)

- Expansion moves vs swap moves



with α-expansions



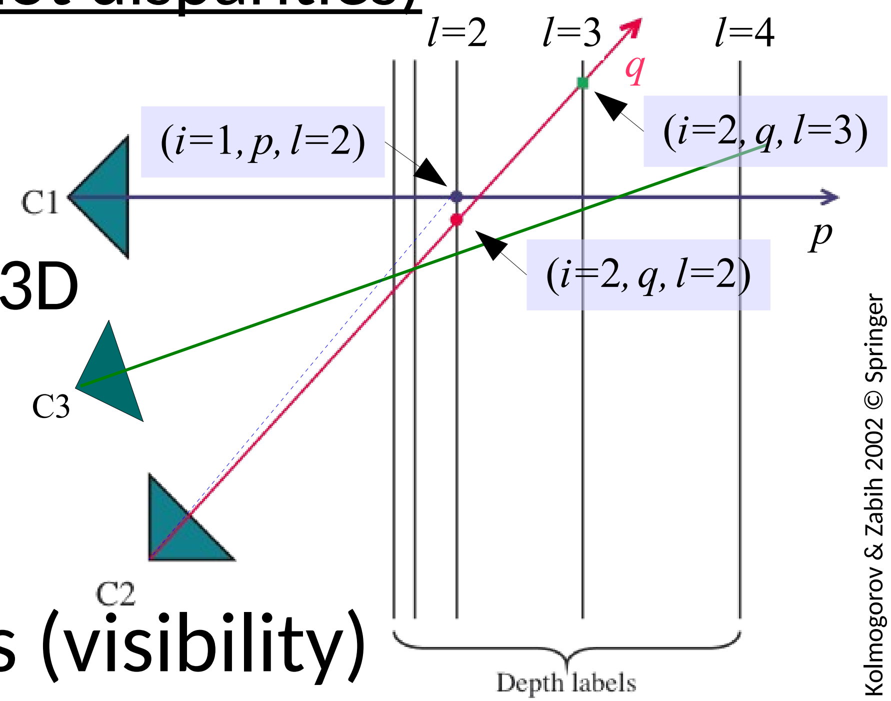
with $\alpha\beta$-swaps

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

Multi-view reconstruction

(cf. Kolmogorov & Zabih 2002)

- Given n calibrated images on the “same side” of scene
- Global model
  - 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 → too many interactions → ☹
  - sol.: def q closest pixel of projection of (i,p,l) on j → ☺



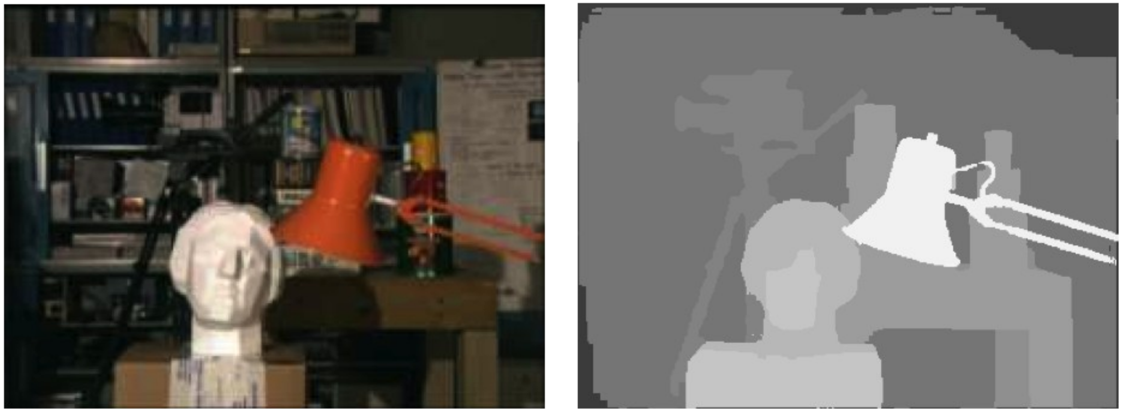
- Photo-consistency constraints (visibility)
  - red point, at depth l=2, blocks C2's view of green point, at depth l=3

Multi-view reconstruction

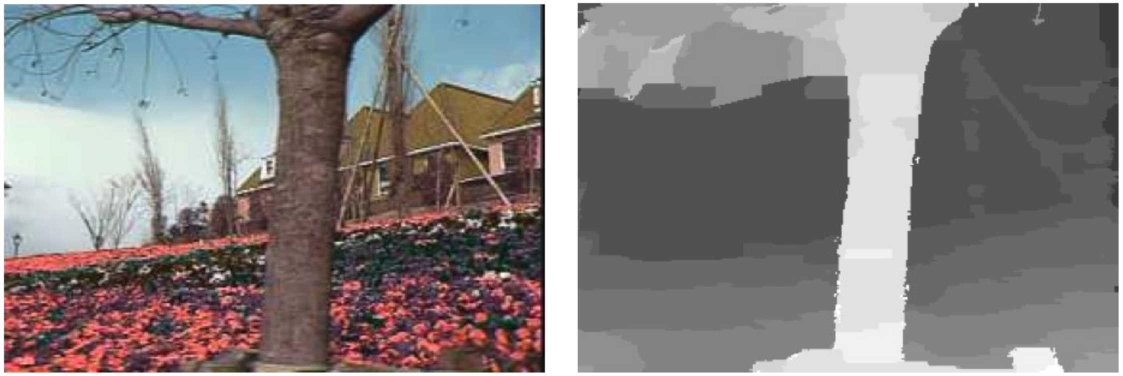
(cf. Kolmogorov & Zabih 2002)

- Terms in the energy: data, smoothness, visibility
- Optimization by α-expansion

See paper for details



(a) Middle image of *Head* dataset (b) Scene reconstruction for *Head* dataset



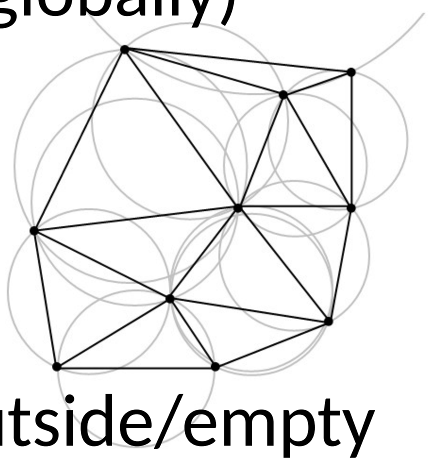
(c) Middle image of *Garden* sequence (d) Scene reconstruction for *Garden* sequence

point cloud = nuage de points
sweep = balayage
outliers = donnée (ici points) aberrantes
tetrahedralization = tétraédrisation

Beyond disparity maps: 3D mesh reconstruction

(cf. Vu et al. 2012)

- Merging of depth maps into single point cloud
  - possibly sparse depth maps, e.g., obtained by plane sweep
- Problems:
  - multi-view visibility (to be taken into account globally)
  - outliers
- Solution:
  - Delaunay tetrahedralization of point cloud
  - binary labelling of tetrahedra: inside/full or outside/empty
  - 3D surface = interface inside/outside



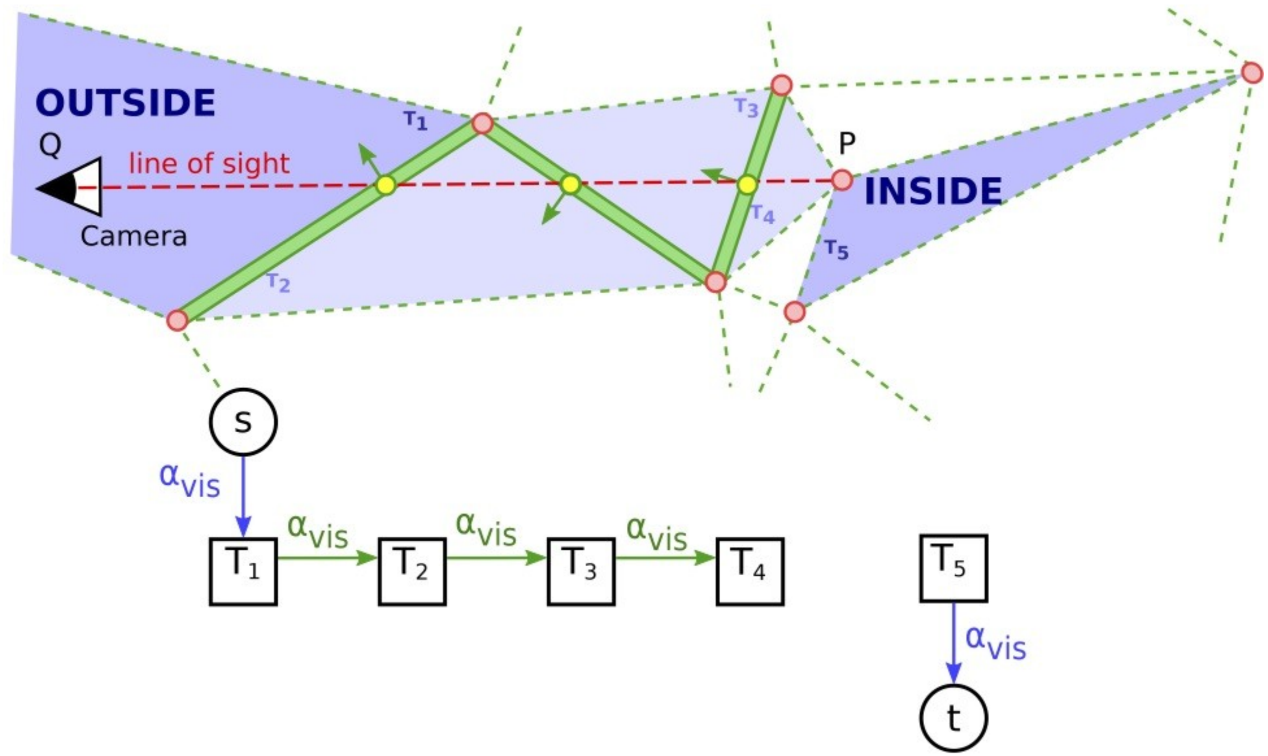
Visibility consistency via graph cut

sight = vue, vision

(cf. Vu et al. 2012)

- Lines of sight from cameras to visible points ⇒ outside

Q,P : points
T : tetrahedron
S : surface
$\boldsymbol{P}$: point cloud
v : line of sight
$l_T = 0$: T outside (empty space)
$l_T = 1$: T inside (occupied space)

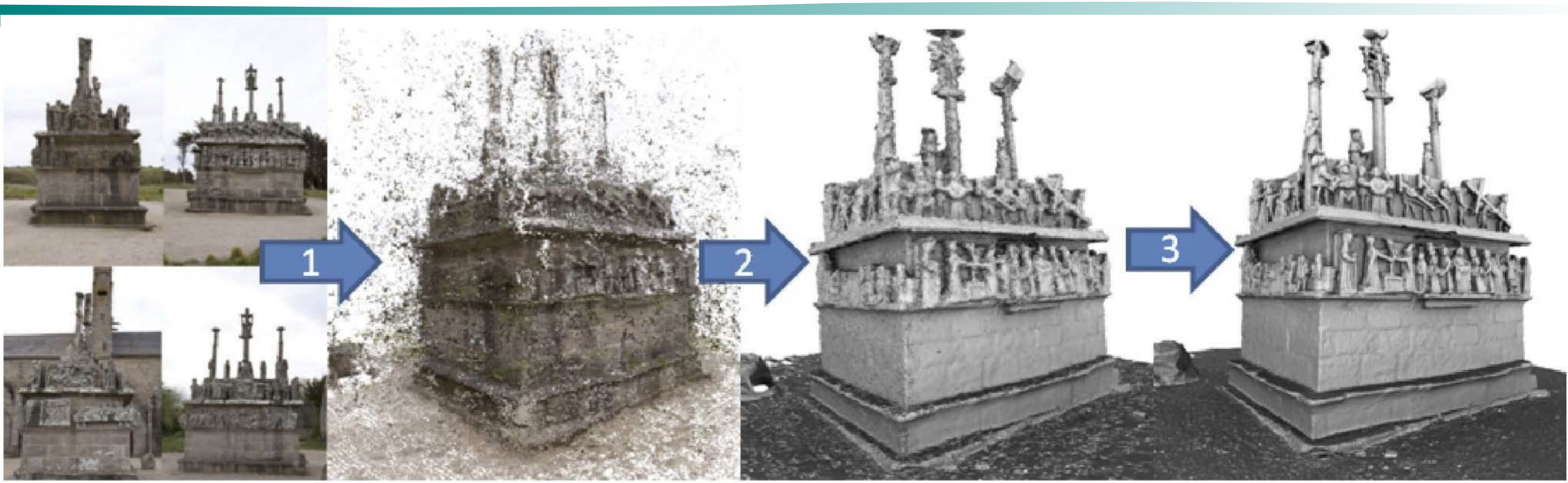


$$D_{\text{out}}(l_T) = \alpha_{\text{vis}} \mathbf{1}[l_T = 1]$$
$$D_{\text{in}}(l_T) = \alpha_{\text{vis}} \mathbf{1}[l_T = 0]$$
$$V_{\text{align}}(l_{T_i}, l_{T_j}) = \alpha_{\text{vis}} \mathbf{1}[l_{T_i} = 0 \wedge l_{T_j} = 1]$$

$$E_{\text{vis}}(S, \boldsymbol{P}, v) = \sum_{P \in \boldsymbol{P}} \left(\sum_{Q \in v_P} D_{\text{out}}(l_{T_1^{Q\to P}}) + \sum_{i=1}^{N_{[PQ]}-1} V_{\text{align}}(l_{T_i^{Q\to P}}, l_{T_{i+1}^{Q\to P}}) + D_{\text{in}}(l_{T_{N_{[PQ]+1}}^{Q\to P}}) \right)$$

Beyond disparity maps: 3D mesh reconstruction

(cf. Vu et al. 2012)



Input images Point cloud Visibility-consistent mesh Refined mesh

- Best reconstruction results on international benchmarks
- Startup company with IMAGINE members (2011)
  - 15 employees, 90% revenue = international
  - bought by Bentley Systems (2015), still success

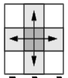
acute3D capturingreality

Exercise 2: simple disparity map estimation (without moves nor occlusion)

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

p

$p' = p + (d_p, 0)$

- Setting: linear multi-label graph construction (cf. pp. 85-96)
  - discrete disparities: $d_p \in \mathrm{L} = \{d_{\min}, \ldots, d_{\max}\}$
  - N_p: 4 neighbors of pixel p 
  - $D_p(d_p) = w_{cc}\, \rho(E_{ZNCC}(P; (d_p, 0)))$ with $\rho(c) = \begin{cases} 1 & \text{if } c < 0 \\ \sqrt{1-c} & \text{if } c \geq 0 \end{cases}$
  - $V_{p,q}(d_p, d_q) = \lambda \, | d_p - d_q |$

☛ See material provided for the exercise on web site (template code and detailed exercise description)

N.B. only w_{cc} / λ matters

Advertisement

Internship/PhD positions
related to 3D
in **IMAGINE research group**
(École des Ponts ParisTech)
and in **Valeo.ai**

Bibliography

- Birchfield, S., and Tomasi, C. (1998). A Pixel Dissimilarity Measure that Is Insensitive to Image Sampling. In IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol. 20, no. 4, pp. 401-406, Apr. 1998.
- Bolles, R. and Woodfill, J. (1993). Spatiotemporal consistency checking of passive range data. In International Symposium on Robotics Research.
- Boykov, Y., and Veksler, O., and Zabih, R. (1998). Markov random fields with efficient approximations. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 648–655, 1998.
- Boykov, Y., Veksler, O., and Zabih,R. (2001). Fast Approximate Energy Minimization via Graph Cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 23:1222–1239, 2001. (Also ICCV 1999.)
- Boykov, Y. and Jolly, M.-P. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In International Conference on Computer Vision (ICCV), vol. I, pp. 105-112.
- Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 26(9):1124–1137, September 2004.
- Boykov, Y. and Veksler, O. (2006). Graph Cuts in Vision and Graphics: Theories and Applications. In Handbook of Mathematical Models in Computer Vision, edited by Nikos Paragios, Yunmei Chen and Olivier Faugeras. Springer.
- Darbon, J. (2009). Global optimization for first order Markov Random Fields with submodular priors. Discrete Applied Mathematics, vol. 157, no. 16, pp. 3412-3423, August 2009.
- Dinic, E. A. (1970). Algorithm for solution of a problem of maximum flow in a network with power estimation. Soviet Math. Doklady (Doklady) 11: 1277–1280.
- Edmonds, J., and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM 19 (2): 248–264.
- Ford, L. R., Jr. and Fulkerson, D. R. (1956). Maximal Flow through a Network. Canadian Journal of Mathematics, 1956, pp.399-404.

Bibliography

- Ford, L. R., Jr. and Fulkerson, D. R. (1962). Flows in Networks, Princeton University Press.
- Goldberg, A., and Tarjan, R. (1988). A new approach to the maximum-flow problem. Journal of the Association for Computing Machinery, 35(4):921–940, October 1988. (Also proc. ACM STOC 1986.)
- Greig, D., Porteous, B., and Seheult, A. (1989). Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society, Series B, 51(2):271–279.
- Ishikawa, H. (2003). Exact optimization for Markov Random Fields with convex priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 25(10):1333– 1336, 2003.
- Ishikawa, H., and Geiger, D. (1998). Occlusions, discontinuities, and epipolar lines in stereo. In 5th European Conference on Computer Vision (ECCV), pages 232–248, 1998.
- Kolmogorov, V., and Zabih, R. (2001). Computing visual correspondence with occlusions via graph cuts. In International Conference on Computer Vision (ICCV), July 2001.
- Kolmogorov, V., and Zabih, R. (2002). Multi-camera scene reconstruction via graph cuts. In 7th European Conference on Computer Vision (ECCV), volume III of LNCS 2352, pages 82–96, Copenhagen, Denmark, May 2002. Springer-Verlag.
- Kolmogorov, V., and Zabih, R. (2004). What energy functions can be minimized via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 26(2):147– 159, February 2004.
- Pansari, P. and Kumar, P. (2017). Truncated Max-of-Convex Models. International Conference on Computer Vision and Pattern Recognition (CVPR).
- 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.
- Sleator, D. D., and Tarjan, R. E. (1981). A Data Structure for Dynamic Trees. Proceedings of the thirteenth annual ACM symposium on Theory of computing (STOC), pp. 114-122. (Also Journal of Computer and System Sciences, 26(3), 1983.)
- Vu, H., Labatut, P., Pons, J.Ph., and Keriven, R. (2012). High Accuracy and Visibility-Consistent Dense Multi-view Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012.