

Computer Aided Medical Procedures II
Lecture Notes
Advanced Segmentation Techniques: Graph-Cuts

Darko Zikic

15 May 2006

1 Disclaimer

Dear Reader. These notes are a supplementary to the lecture *Computer Aided Medical Procedures II*, offered at the TU München by Prof. Dr. Nassir Navab. The notes are *neither* complete *nor* self contained and are meaningful only combined with the visit of the lecture. Comments are gladly accepted at zikic@in.tum.de.

Contents

1 Disclaimer	1
2 Graph Basics	2
2.1 Definition of Graphs (for Graph-Cuts)	2
2.2 Graph-Cuts	2
2.3 Min-Cut = Max-Flow	3
3 Binary Minimization with Graph-Cuts	3
3.1 Example: Binary Image Restoration	3
3.2 General Formulation	5
4 Graph-Cuts as Hypersurfaces	5
5 Segmentation using Graph-Cuts	5
5.1 Method	5
5.2 Properties	6

2 Graph Basics

2.1 Definition of Graphs (for Graph-Cuts)

$G = \langle V, E \rangle$	Graph G with a set of vertices/nodes V and a set of edges E .
$e = (p, q)$	We denote one edge $e \in E$ connecting the two nodes p and q ($p, q \in V$) also as $e = (p, q)$. The edges might be directed or undirected. Here we assume directed graphs for which in general it holds that $(p, q) \neq (q, p)$ for $p, q \in V$.
s, t	The so called <i>Terminal Nodes/Terminals</i> s, t are special nodes, i.e. $s, t \in V$. The node s is called <i>source</i> and the node t is referred to as <i>sink</i> .
P	Set of all <i>non-terminal</i> nodes, i.e. $P = V \setminus \{s, t\}$
$w(p, q)$	Weight of the edge from p to q , $w(p, q) \geq 0$ (real function $w : V \times V \rightarrow \mathbb{R}$). For directed graphs it holds that in general $w(p, q) \neq w(q, p)$. Also denoted as $w(e)$ for $e = (p, q)$. Also known as <i>capacity</i> of an edge.
n-link, N	Edge between two non-terminal nodes, i.e. $N = \{(p, q) \in E : p, q \in P\}$
t-link	Edge between a terminal and a non-terminal node, i.e. $\{(s, p) \in E : p \in P\} \cup \{(p, t) \in E : p \in P\}$

2.2 Graph-Cuts

Cut, C	An s/t cut $C = (S, T)$ of a given graph $G = \langle V, E \rangle$ is a partition of nodes into two disjoint subsets S and T , such that $s \in S$ and $t \in T$, i.e. $V = S \cup T$ and $S \cap T = \emptyset$
Cut Border, B_C	A set of edges that get severed by the cut $C = (S, T)$, i.e. $B_C = \{(p, q) : p \in S, q \in T\}$, $B_C \subset E$.
Cost of Cut, $ C $	Weight of all boundary edges, i.e. $ C = \sum_{e \in B_C} w(e)$. Also known as <i>capacity</i> of the cut.
Minimum Cut	Cut with minimum cost: $\arg \min_C \sum_{e \in B_C} w(e)$
Labeling l	A cut can be described using a binary labeling function $l : V \rightarrow \{0, 1\}$, such that for $p \in S$ it holds that $l(p) = 0$ and for $p \in T$ it holds that $l(p) = 1$

The Min-Cut Problem is defined as the problem of finding a minimum cut, given a graph G .

2.3 Min-Cut = Max-Flow

A graph $G = \langle V, E \rangle$ with terminals source s and sink t and the weight/capacity function w can be thought of as a pipe-system transporting water. $w(p, q)$ gives the amount of water that can flow from p to q (capacity of the pipe). The *Flow* $f(p, q)$ gives the *actual* amount of water flowing from p to q .

Flow (f)	Real function $f : V \times V \rightarrow \mathbb{R}$ with following properties: $f(p, q) = -f(q, p)$, $f(p, q) \leq w(p, q)$, $\sum_{p, q \in P} f(p, q) = 0$.
Residual Capacity	Denotes the available capacity at edge (p, q) given flow f , i.e. $w_f(p, q) = w(p, q) - f(p, q)$.
Residual Graph	$G_f = \langle V, E_f \rangle$
Augmenting Path	A path along which there is still available capacity, i.e. (p_1, \dots, p_n) with $p_1 = s$ and $p_n = t$ with $W_f(p_i, p_{i+1}) > 0$.

The Max-Flow Problem is defined as the problem of finding a legal and maximal flow $\arg \min_f \sum_{p, q \in V} f(p, q)$

Theorem: Ford and Fulkerson Max-Flow $s \rightarrow t$ saturates the set of edges which divide the graph in two disjoint parts which correspond to the Min-Cut. For proof and further details compare, e.g. Papadimitriou et al. [2].¹

Bottom Line We can solve the Min-Cut Problem by solving the Max-Flow Problem.

Algorithms The Min-Cut/Max-Flow problem can be solved in polynomial time. There are two groups of algorithms (see Boykov and Veksler in [3]):

1. Push-Relabel Algorithms: better for general graphs, good parallelization properties
2. Augmenting Paths: for grids there exist fast methods with observed linear run times

3 Binary Minimization with Graph-Cuts

3.1 Example: Binary Image Restoration

Given a damaged binary image (intensities 0/1) where the values for certain points are corrupted by noise.

¹The Min-Cut and Max-Flow problem can both be seen as *Linear Programming Problems* and their relation can be seen as a specialization of the *Duality Theorem* for Linear Programming.

A Min-Cut of a graph as constructed below yields a reasonable solution to the problem.

Basic Definitions

- Interpret image as grid graph: $I(p)$ is the intensity at node p .
- Add terminal nodes with t-links to all non-terminal nodes.
- “ $s = 0$ ” and “ $t = 1$ ” - the source represents the intensity 0 and the sink the intensity 1.
- Assign the intensity 0 to a pixel p if $p \in S$. If $p \in T$ then it is assigned the intensity 1.
- Use the labeling l , such that for $p \in S$ it holds $l(p) = 0$ and for $p \in T$ it holds that $l(p) = 1$.

The solution will always sever exactly one t-link. Thus the remaining link will present the intensity to be assigned to the image point. Thus the Min-Cut will yield an admissible solution.

Penalty Term

- $D_p(L)$ fixed penalty for assigning node p to label $L = \{0, 1\}$.
- Set $w(s, p) = D_p(1)$ and $w(p, t) = D_p(0)$

Regularizing Component is used in order to remove image noise and ensure coherence between neighboring nodes.

- Create *n-links* between neighboring nodes, such that a cut severs as little links as possible.
- Set weight of the n-links to a parameter $\lambda > 0$

Cost of Cut

$$|C| = \sum_{p \in P} D_p(l(p)) + \sum_{e \in B \cap N} w(e) \quad (1)$$

$$= \sum_{p \in P} D_p(l(p)) + \lambda \sum_{(p,q) \in N} I(l(p) = 0 \wedge l(q) = 1) \quad (2)$$

with $I(\cdot)$ the identity function, i.e. $I(0) = 0$ and $I(1) = 1$.

3.2 General Formulation

The upper term for the cost of the cut can be seen as a general *Energy Functional* E which judges a given labeling l (and thus of course the given cut C).

$$E(l) = |C| = \sum_{p \in P} D_p(l(p)) + \sum_{(p,q) \in N} V_{pq}(l(p), l(q))$$

Here, the function V_{pq} presents a general function $V_{pq} : V \times V \rightarrow \mathbb{R}$ which assigns a weight to the edge (p, q) .

It can be shown, that an optimal solution minimizing the energy functional can be found via graph cuts if and only if the following condition is satisfied (compare Boykov and Veksler in [3]).

Regularity Condition

$$V_{pq}(0, 0) + V_{pq}(1, 1) \leq V_{pq}(0, 1) + V_{pq}(1, 0)$$

Multilabel Minimization For multi-label problems, the upper minimization is in general NP-complete [3].

4 Graph-Cuts as Hypersurfaces

Graph-Cuts can be interpreted as hypersurfaces in n-dimensional manifolds. This is important since many problems can be formulated s.t. the solution is represented by a hypersurface.

Modeling of the graph for hypersurface interpretation:

- Not all non-terminals are connected to the sink and the source
- All existing t-links have infinite weights (no “soft” t-links).

Further details can be found in Boykov and Veksler in [3].

5 Segmentation using Graph-Cuts

Segmentation of structures in images follows directly from the hypersurface interpretation.

5.1 Method

The image is interpreted as a graph with grid structure. The questions arising are the choice of suitable t-links and appropriate n-link weights.

Choice of Suitable t-Links For segmentation, the terminal nodes are interpreted as foreground and background. The user must select foreground and background points in the image (*seeds*). Between these selected non-terminal nodes and the terminals, t-links with infinite costs are created.

Choice of n-Link Weights The weights of the n-links are selected such that improbable segmentations are penalized. This is mostly done by the assumption that high gradients are presenting boundaries between objects. The weight function is chosen s.t. the edges between nodes with high intensity gradients are assigned high costs. A common choice for the weight function is

$$w(p, q) = f(|I(p) - I(q)|)$$

with the non-negative decreasing function $f : \mathbb{R} \rightarrow \mathbb{R}$

$$f(x) = K \cdot \exp\left(\frac{-x^2}{\sigma^2}\right)$$

with $K \in \mathbb{R}$, $K \geq 0$, and $\sigma \in \mathbb{R}$, $[1]$.

5.2 Properties

Positive

- Good and fast results for 2D and 3D images
- Interactive: possible to add or remove seeds without recomputing the whole solution

Negative

- High memory consumption: can be helped by using prior knowledge of anatomy, regions of interest and multi-resolution approaches.
- Multi-label problem in general case intractable (\rightarrow only hard segmentation possible)
- Small-Cut problem: algorithm gives solution with minimum boundary in certain cases of seed placement

References

- [1] Yuri Boykov and Marie-Pierre Jolly. Interactive organ segmentation using graph cuts. In Scott L. Delp, Anthony M. DiGioia, and Branislav Jaramaz, editors, *MICCAI*, volume 1935 of *Lecture Notes in Computer Science*, pages 276–286. Springer, 2000.
- [2] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization*. Dover Publ., corr., unabridged republ. of 1982 edition, 1998.
- [3] Nikos Paragios, editor. *Handbook of mathematical models in computer vision*. Springer, 2006.