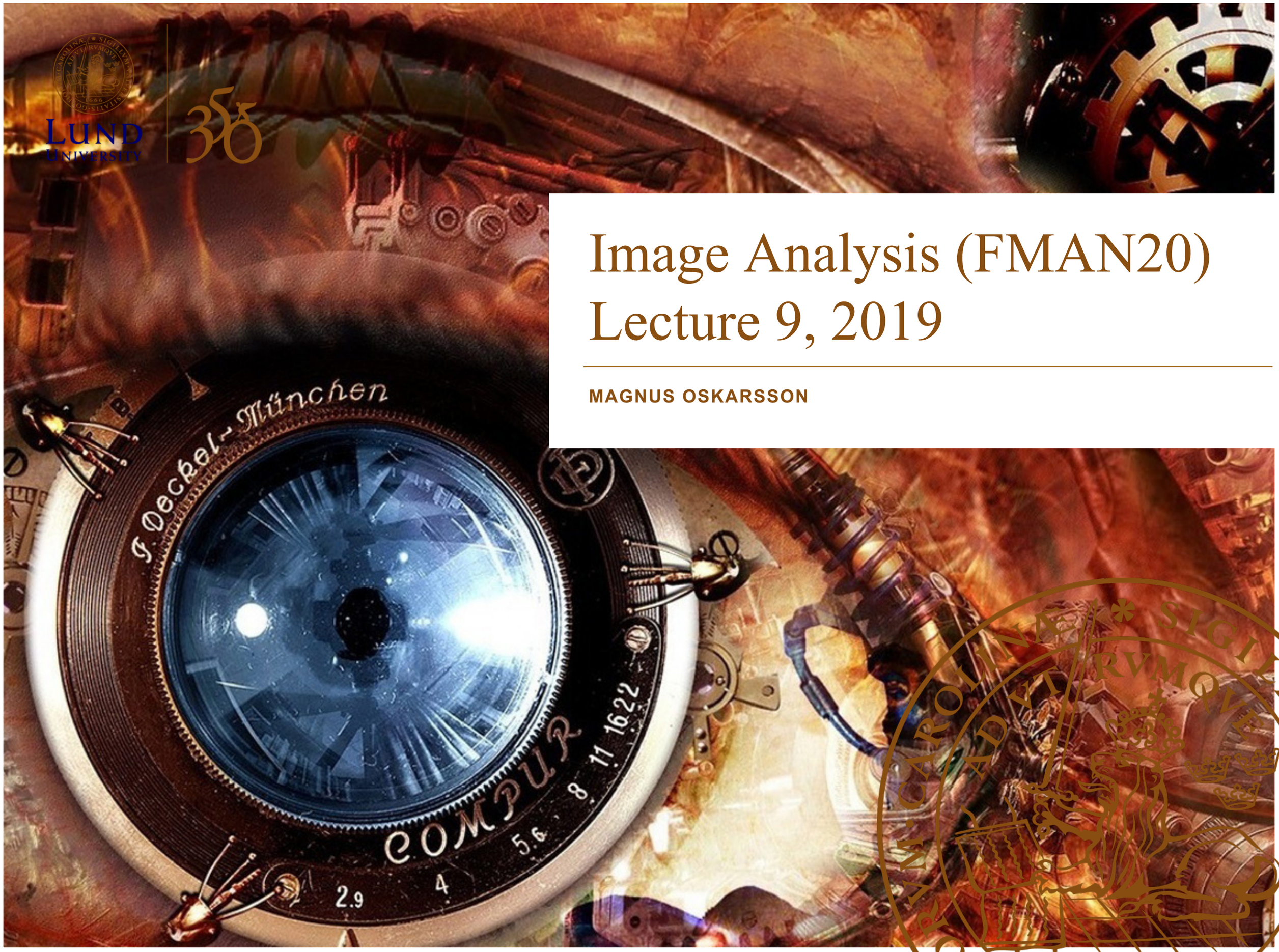


Image Analysis (FMAN20) Lecture 9, 2019

MAGNUS OSKARSSON

How?

Segmentation using Graphs

- Graphs
- A simple graph based clustering method
- The Mumford-Shah functional
- Graph cuts

Graph theory

A graph $G = (V, E)$ consists of vertices(nodes) V and edges E.
Every edge connects two vertices.
In a directed graph, every edge has an orientation.
In a weighted graph, every edge has a weight (a number).
A graph is connected if one can 'walk' between all pairs of vertices through one or several edges.
Every graph can be split into a disjoint set of connected components.

Graph theory

Weighted graphs can be represented as a matrix. A weighted edge between vertex i and vertex j with v is represented by matrix element (i, j).
For un-directed graphs, half the weight is put at position (i, j) and half in (j, i).
Connected components - blocks in block diagonal matrices.

Graph theoretic clustering

- Represent tokens using a weighted graph.
  - affinity matrix
- Cut up this graph to get subgraphs with strong interior links

Graph theoretic clustering

When solving clustering problems with graph theoretical methods one need a closeness measure $v_{i,j}$, for every pair of nodes (i, j). A large number means that they are close. A small number means that they are different.
The affinity measure depends on which problem one has.
Usual ingredients are

- Distance - e.g. $aff(x, y) = e^{-(x-y)^T(x-y)/(2\sigma_d^2)}$
- Intensity - e.g. $aff(x, y) = e^{-(I(x)-I(y))^T(I(x)-I(y))/(2\sigma_I^2)}$
- Color - e.g. $aff(x, y) = e^{-dist(c(x),c(y))^2/(2\sigma_c^2)}$
- Texture - e.g. $aff(x, y) = e^{-(f(x)-f(y))^T(f(x)-f(y))/(2\sigma_f^2)}$

Graph theoretic clustering

Assume that w_n is a vector of ones for those elements that belong to a particular cluster and zeros otherwise. Then the sum of all weights for edges within a cluster is

$$w_n^T A w_n.$$

By maximizing $w_n^T A w_n$ with the constraint $w_n^T w_n = 1$ one might argue that we maximize clustering.
Maxima with this problem corresponds to stationary points of the Lagrange function.

Graph theoretic clustering

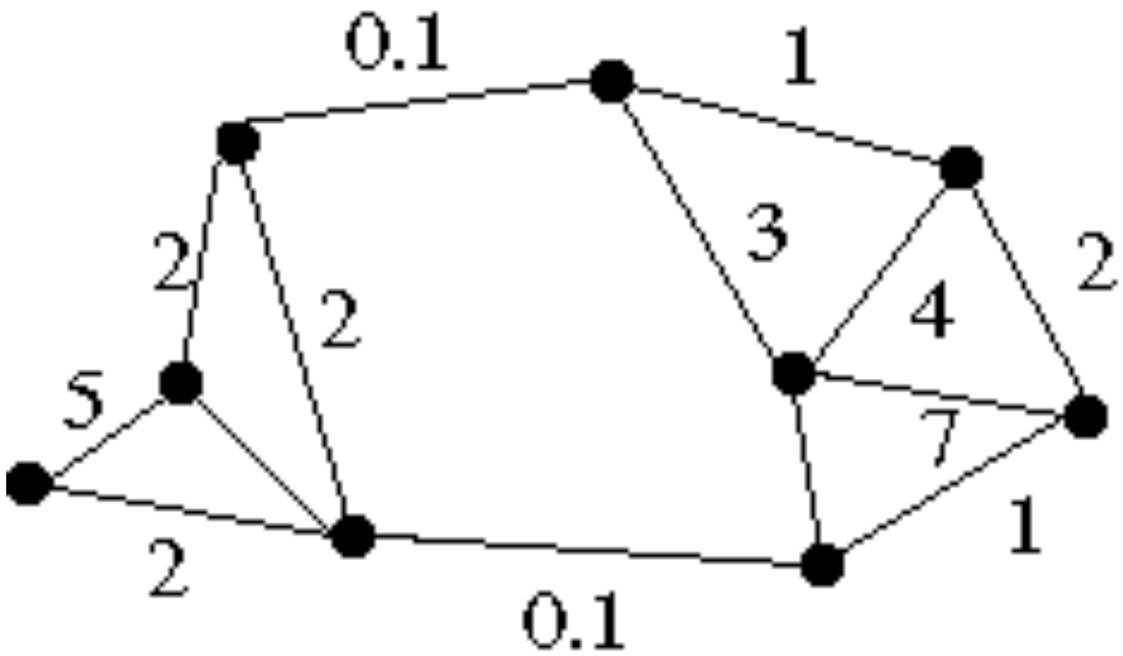
Maximize $w_n^T A w_n$ with constraint $w_n^T w_n = 1$.
Study the Lagrange function

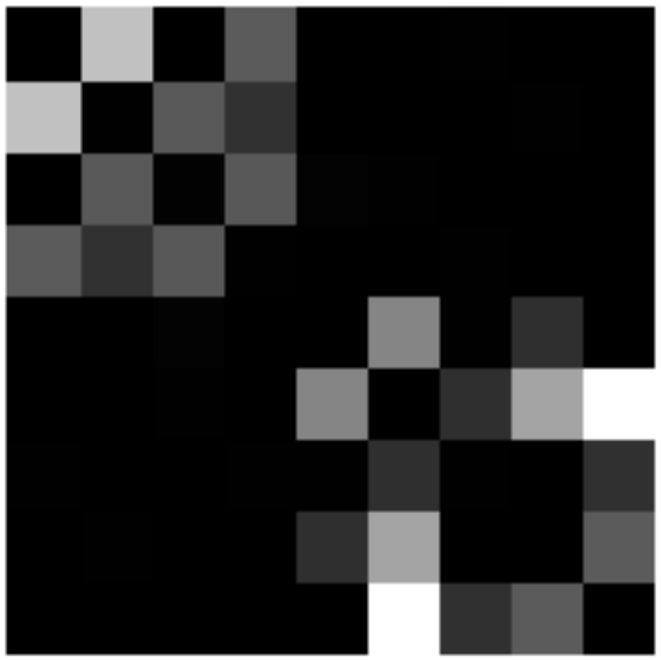
$$L(w_n, \lambda) = w_n^T A w_n + \lambda(w_n^T w_n - 1)$$

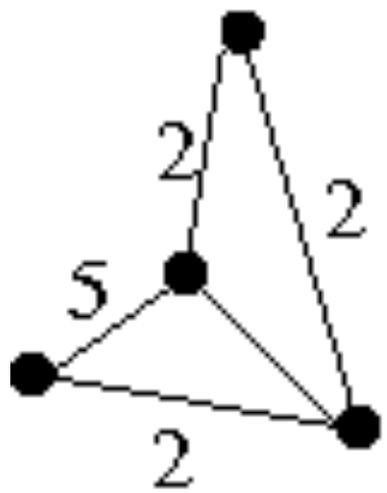
Differentiate and divide with two gives

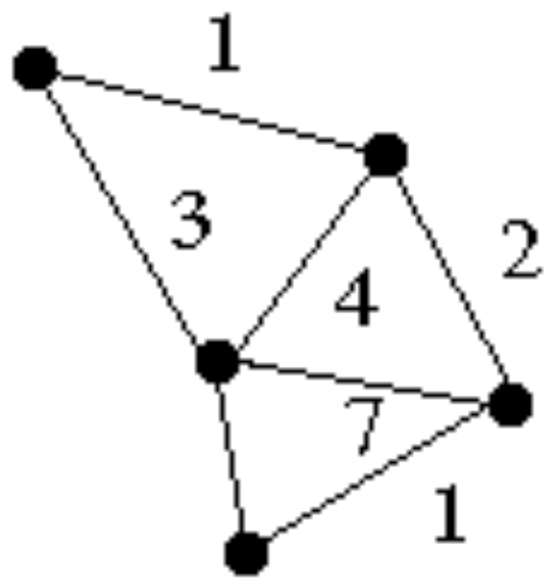
$$A w_n = -\lambda(w_n)$$

This is an eigenvalue problem.

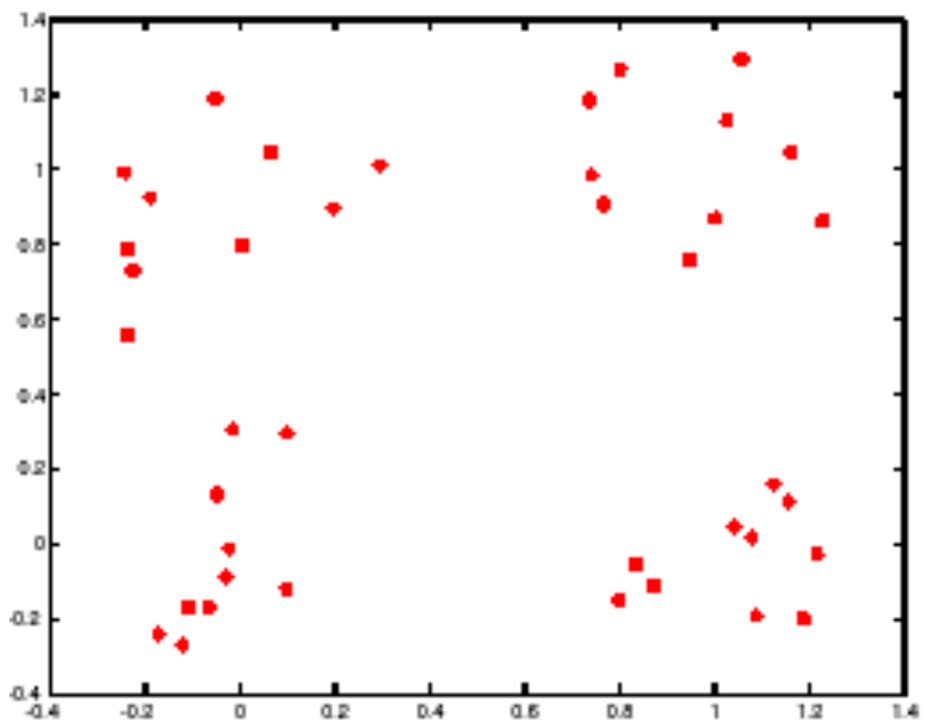

0.1
1
2
2
3
4
2
5
7
2
1
0.1



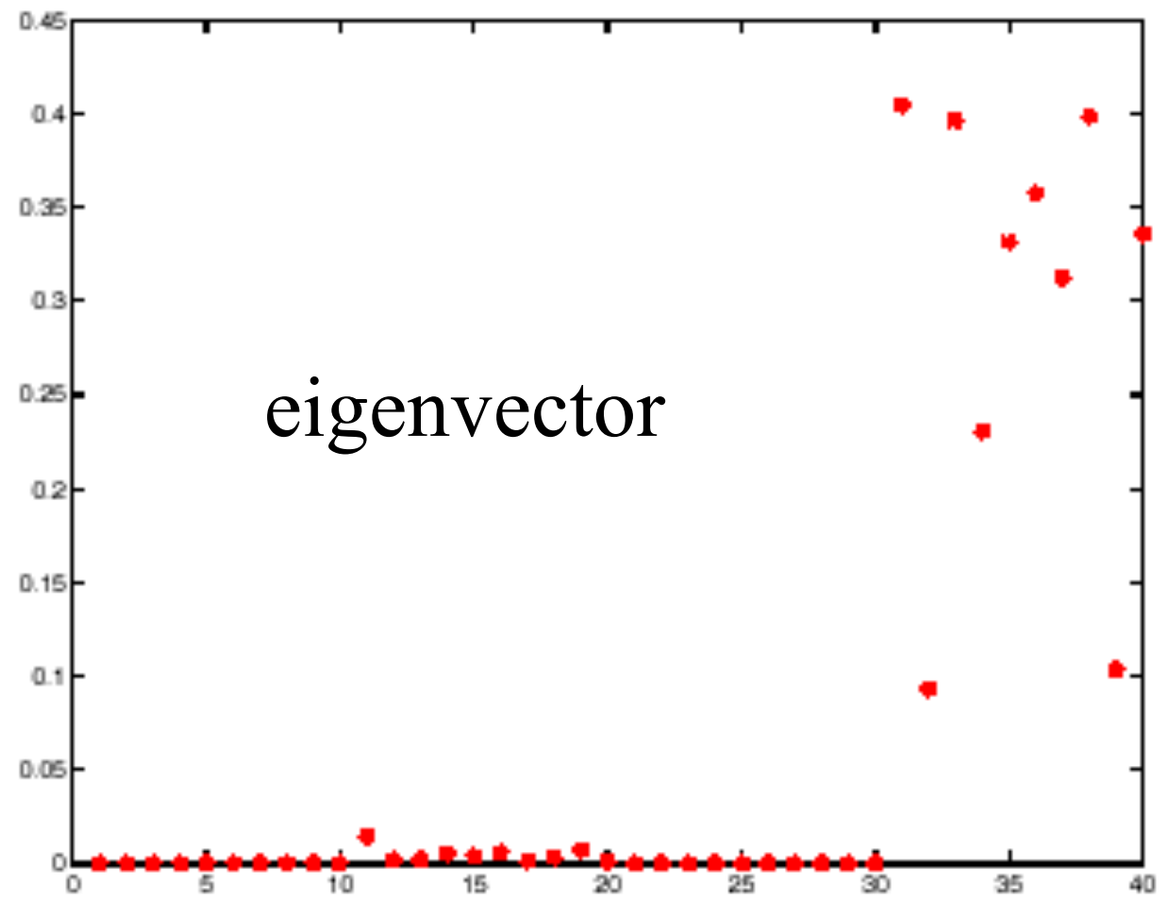

2
2
5
2


1
3
4
2
7
1

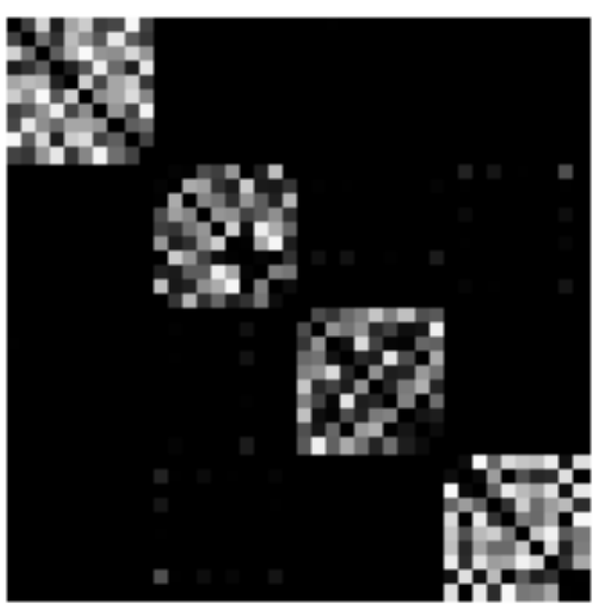
Example eigenvector



points



eigenvector



matrix

More than two segments

- Two options
  - Recursively split each side to get a tree, continuing till the eigenvalues are too small
  - Use the other eigenvectors

3. Segmentation problem
The Mumford-Shah functional

- Consists in computing a decomposition of the domain of the image ***f(x,y)***

$$R = \bigcup_{i=1}^{n} R_i$$

1. ***f*** varies smootly and/or slowly within $\boldsymbol{R_i}$
2. ***f*** varies discontinuously and/or rapidly across most of the boundary Γ between regions $\boldsymbol{R_i}$

Segmentation problem

- Segmentation problem may be restated as
  - finding optimal approximations of a general function $\boldsymbol{f}$ by piece-wise smooth functions $\boldsymbol{g}$, whose restrictions $\boldsymbol{g_i}$ to the regions $\boldsymbol{R_i}$ are differentiable
- Many other applications:
  - Speech recognition
  - Sonar, radar or laser range data
  - MR images and CT scans
  - etc…

Optimal Segmentation

- Mumford and Shah studied 3 functionals which measure the degree of match between an image ***f(x,y)*** and a segmentation.
- First, they defined a general functional ***E*** (the famous Mumford-Shah functional):
  - $\boldsymbol{R_i}$ will be disjoint connected open subsets of the planar domain ***R***, each one with a piece-wise smooth boundary
  - *Γ* will be the union of the boundaries.

$$R = \bigcup_{i=1}^{n} R_i \bigcup \Gamma$$

Mumford-Shah functional

- Let ***g*** differentiable on $\cup R_i$ and allowed to be discontinuous across Γ.

$$E(g,\Gamma) = \mu^2 \int_R (g-f)^2 dxdy + \int_{R-\Gamma} ||\nabla g||^2 dxdy + \nu|\Gamma|$$

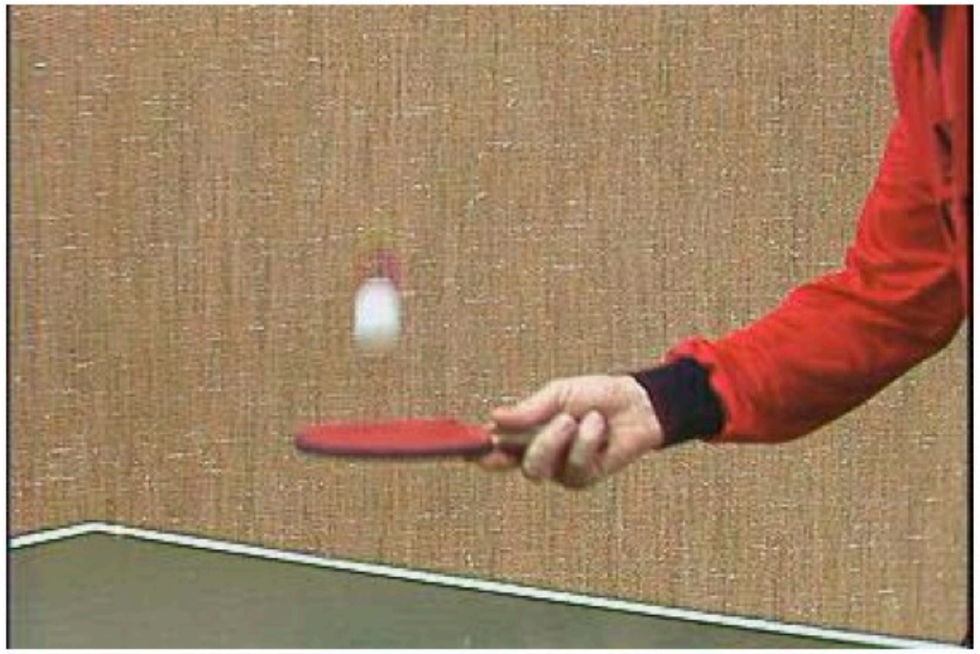
- The smaller ***E***, the better ***(g, Γ)*** segments ***f***
  1. ***g*** approximates ***f***
  2. ***g*** (hence ***f)*** does not vary much on R_is
  3. The boundary *Γ* be as short as possible

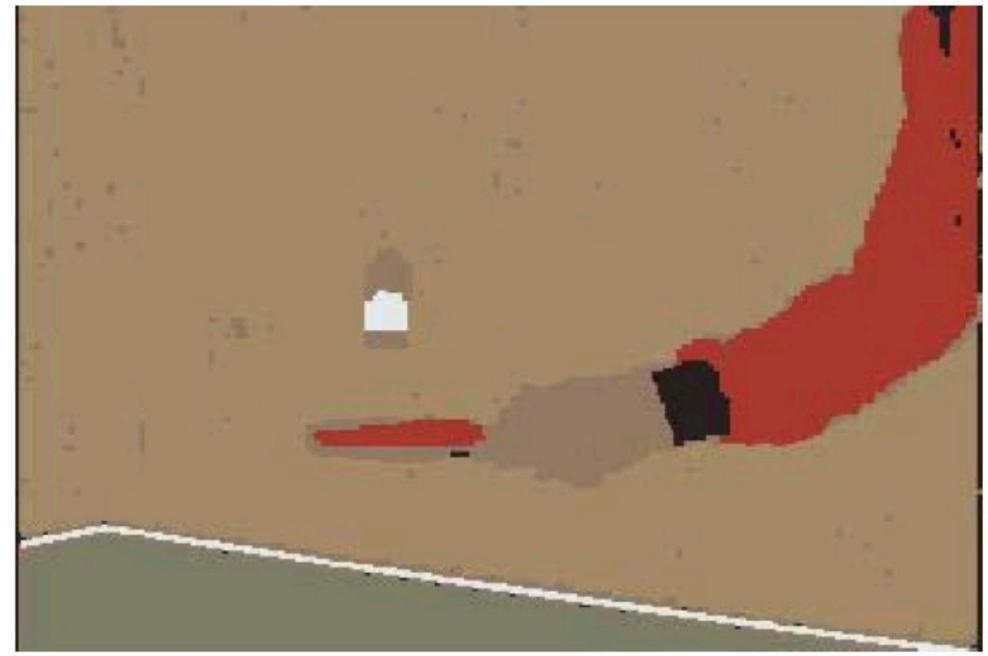
- Dropping any term would cause ***inf E=0.***

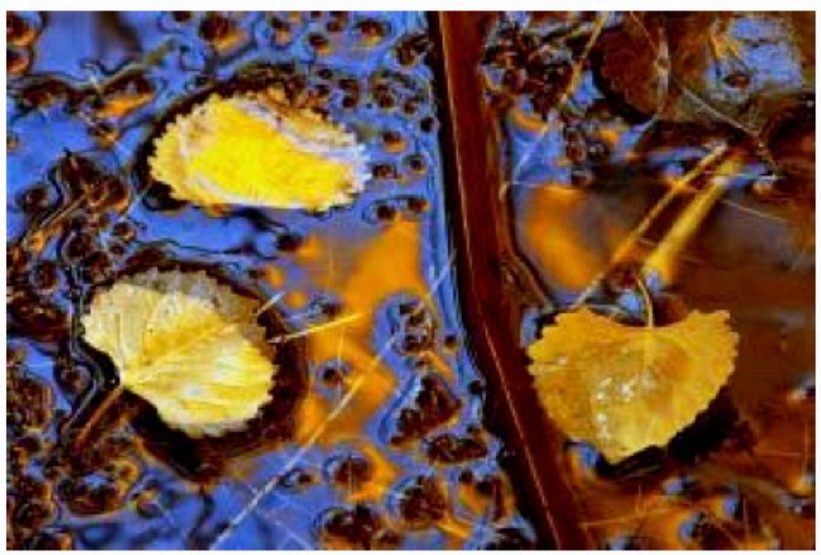
Cartoon Image

- ***(g, Γ)*** is simply a cartoon of the original image ***f.***
  - ✓ Basically ***g*** is a new image with edges drawn sharply.
  - ✓ The objects are drawn smootly without texture.
  - ✓ ***(g, Γ)*** is essentially an idealization of ***f*** by the sort of image created by an artist.
  - ✓ Such cartoons are perceived correctly as representing the same scene as ***f->g*** is a simplification of the scene containing most of its essential features.

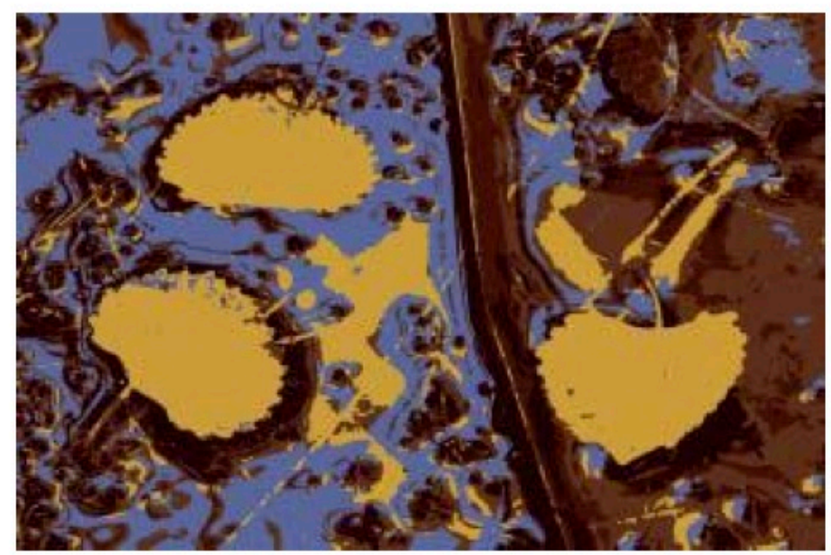
Cartoon Image example







f



g

Piecewise constant approximation

- A special case of ***E*** where ***g=a***$_i$ is constant on each open set ***R***$_i$.

$$E_0(a,\Gamma) = \sum_{i=1}^{n} \int_{R_i} (a_i - f)^2 dxdy + \nu|\Gamma|$$

- It is minimized in ***a***$_i$ by setting ***a***$_i$ to the mean of ***f*** in ***R***$_i$.

$$a_i = \int_{R_i} f\,dxdy/\mathrm{area}(R_i)$$

Piecewise constant approximation

- A special case of ***E*** where ***g=a$_i$*** is constant on each open set ***R$_i$***.

$$E_0(a,\Gamma) = \sum_{i=1}^{n} \int_{R_i} (a_i - f)^2 dxdy + \nu|\Gamma|$$

- It can be proven that minimizing ***E$_0$*** is well posed:
  - ✓ For any continuous ***f***, there exists a ***Γ*** made up of finite number of singular points joined by a finite number of arcs on which ***E$_0$*** attains a minimum.

Two phase Mumford-Shah functional

$$E_0(a_1, a_2, \Gamma) = \int_{R_1} (a_1 - f)^2 dxdy + \int_{R_2} (a_2 - f)^2 dxdy + \nu|\Gamma|$$

- Energy based on two segments R_1 and R_2
- Assume a_1 and a_2 known
- Regularization based on boundary length

4. Segmentation - Graph Cuts

- Idea:

1. See the segmentation problem as a classification problem
2. Finding the highest a posteriori classification (segmentation) is an optimization problem
3. Construct a graph so that the min-cut problem is equivalent to the optimization problem in step 2.
4. Compute a minimum cut that gives the optimal solution.

Note: Min-cut of a graph can be efficiently computed (polynomial time) via max flow algorithms.

A priori probabilities of segmentations

Idea:

We are segmenting some pixels as foreground (1) and some as background (0).

It might be more probable with foreground pixels or the inverse, e.g. $P(g_i=1)=p_1$

Note: Min-cut of a graph can be efficiently computed (polynomial time) via max flow algorithms.

Statistical interpretation

Notation:

f – observed image

g – sought, unknown image

$P(g|f)$ - posterior distribution

Using the *Maximum A Posteriori* (*MAP*) principle, we should maximize the posterior distribution.

Bayes rule: $$P(g|f) = \frac{P(f|g)P(g)}{P(f)}$$

Negative logs give:

$$-\log(P(g|f)) = -\log(P(f|g)) - \log(P(g)) + \text{const}$$

Energy:

$$E(f,g) = E_{data}(f,g) + E_{prior}(g)$$

Statistical two-phase Mumford-Shah

Energy: $E(f,g) = E_{data}(f,g) + E_{prior}(g)$

Recall:

$$E_0(a_1, a_2, \Gamma) = \int_{R_1} (a_1 - f)^2 dxdy + \int_{R_2} (a_2 - f)^2 dxdy + \nu|\Gamma|$$

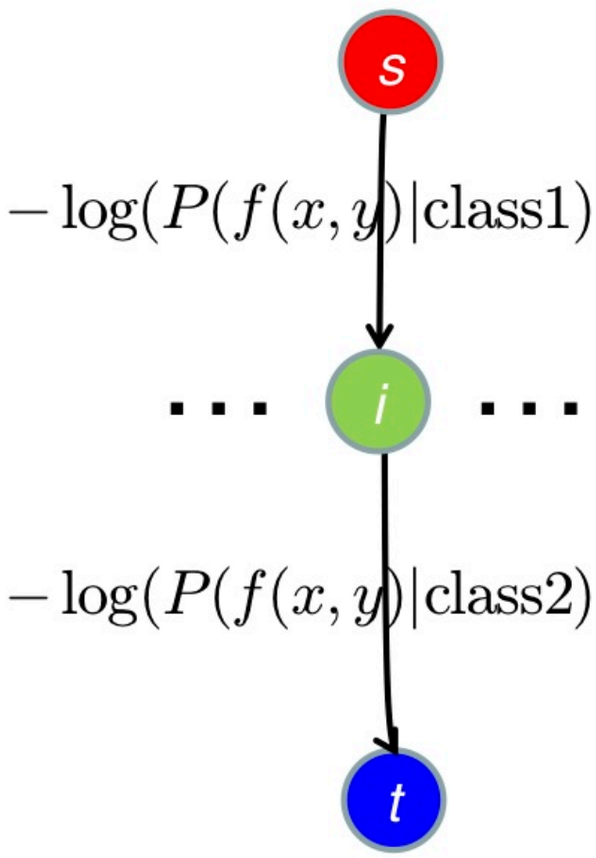
First two data terms: ”reconstructed *g* should be close to data (image) *f*”.
Third term: ”prior knowledge says that shorter curves *g* are preferred”.

More general formulation:

$$E_0(\Gamma) = \int_{R_1} -\log(P(f(x,y)|\text{class1})dxdy + \int_{R_2} -\log(P(f(x,y)|\text{class2})dxdy + \nu|\Gamma|$$

Edge weights for statistical model

Set edge weights such that a cut corresponds to a solution of the optimization problem



Consider pixel *i*. A cut must contain either:

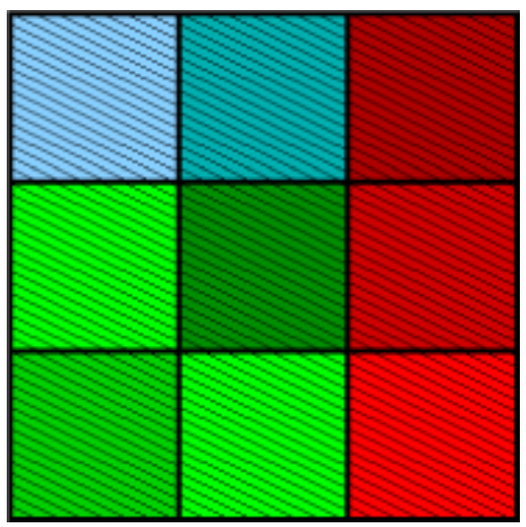
1. the edge (*s,i*), or
2. the edge (*i,t*)

Set edge weights accordingly:

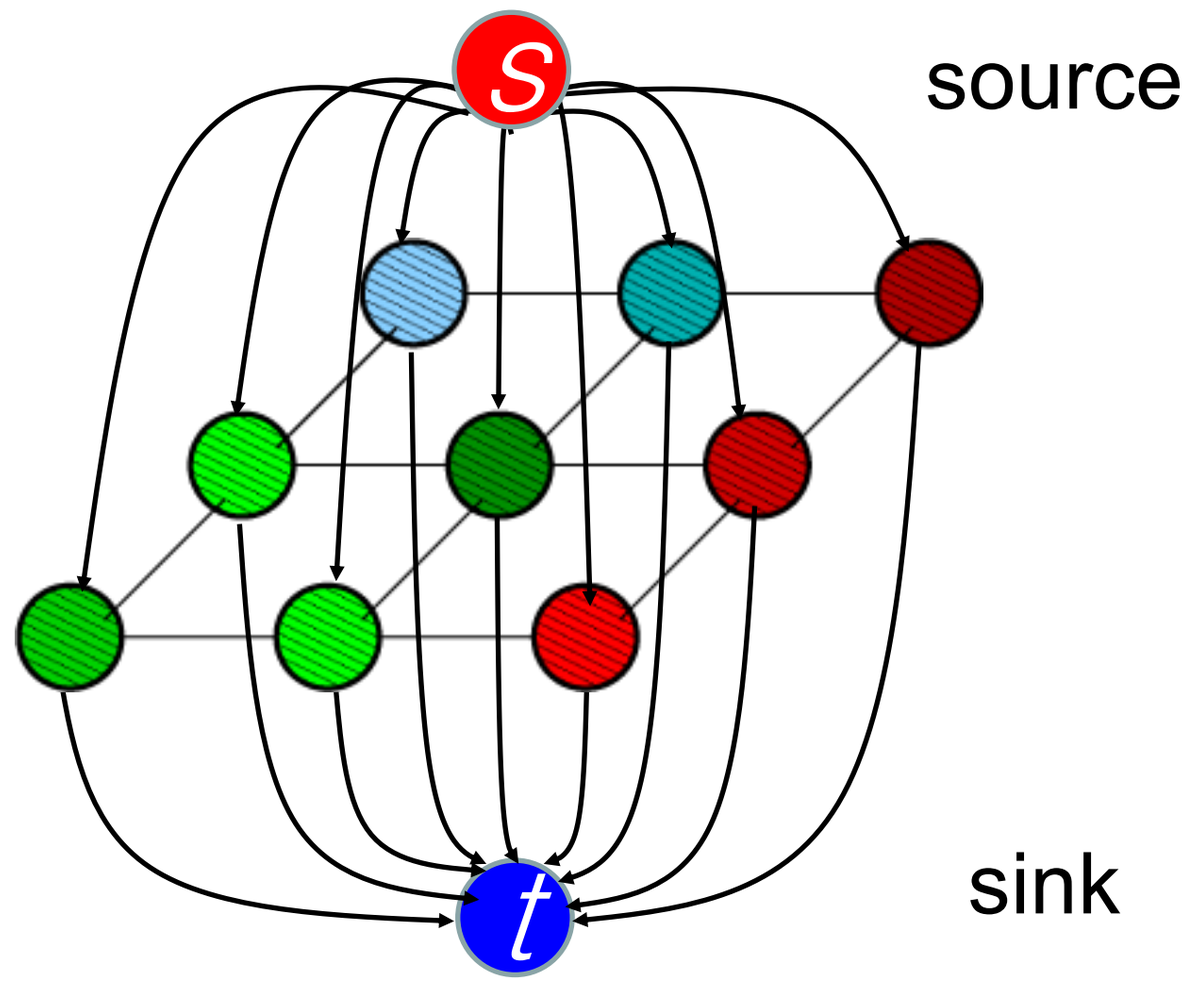
1. $-\log(P(f(x,y)|\text{class1})$ for edge (*s,i*),
2. $-\log(P(f(x,y)|\text{class2})$ for edge (*i,t*)

$$E_0(\Gamma) = \int_{R_1} -\log(P(f(x,y)|\text{class1})dxdy + \int_{R_2} -\log(P(f(x,y)|\text{class2})dxdy + \nu|\Gamma|$$

Graph representation of images

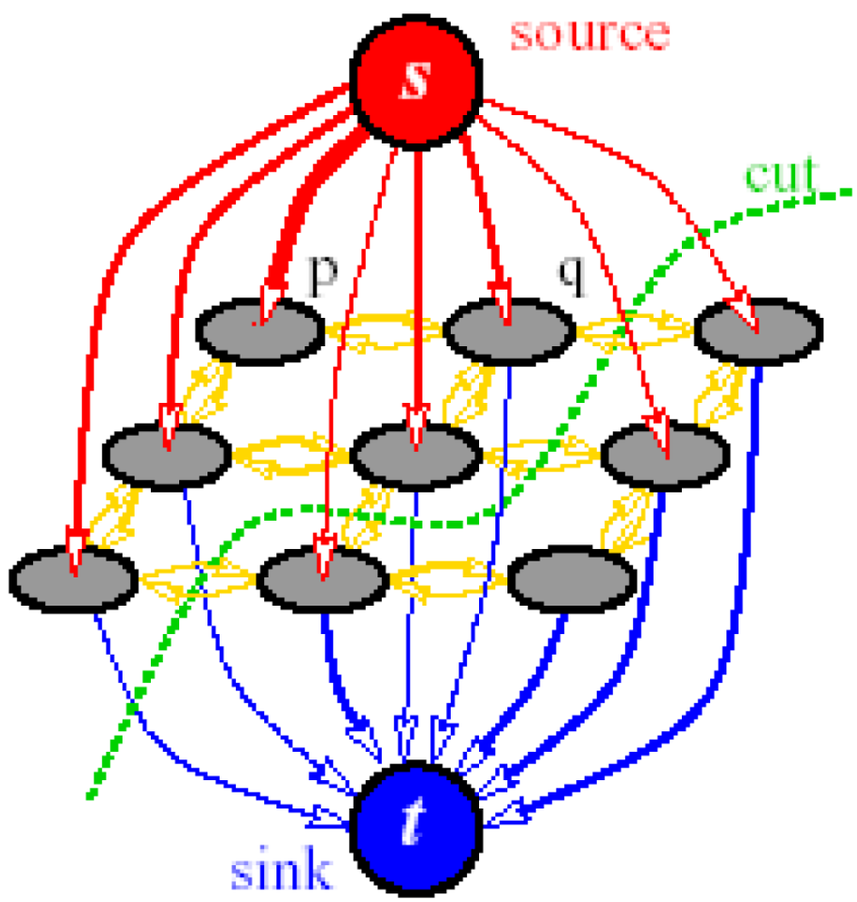


3x3 image



Directed, weighted graph, one node for every pixel + source and sink nodes

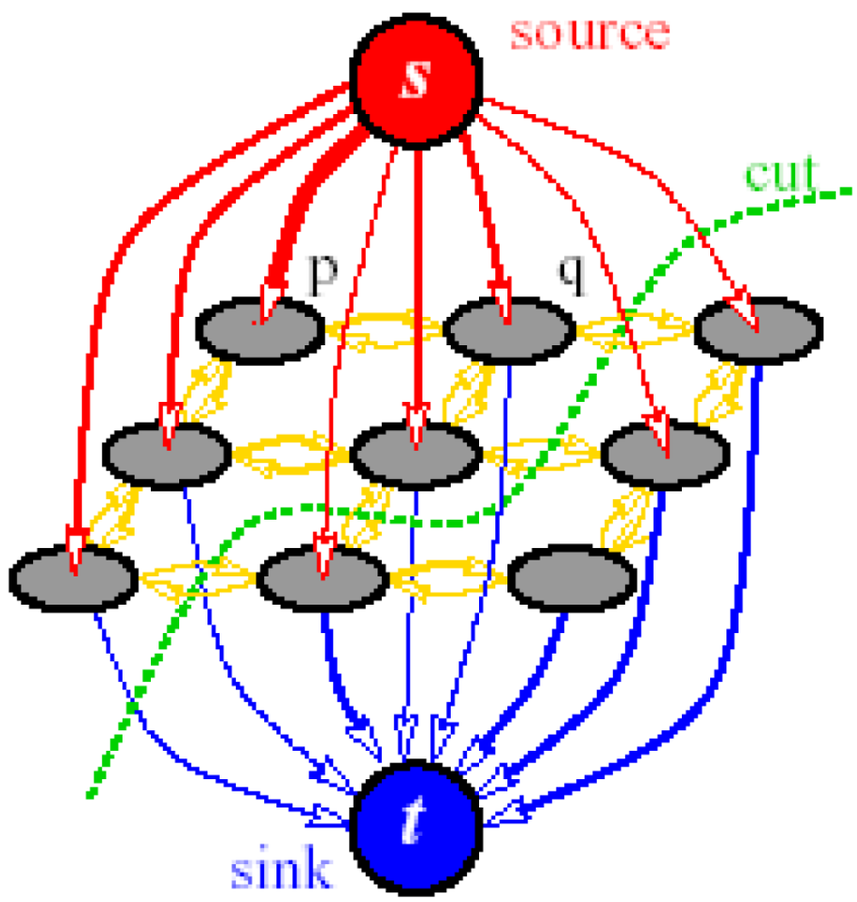
Graph Cuts



Definition: A *cut* (or *s-t cut*) in a graph $G=(V,E)$ is a subset of edges E_c such that there is no path from s to t when E_c is removed.

Definition: The *cost* of a cut is the sum of all edge weights for the edges in the cut.

Minimum Cuts

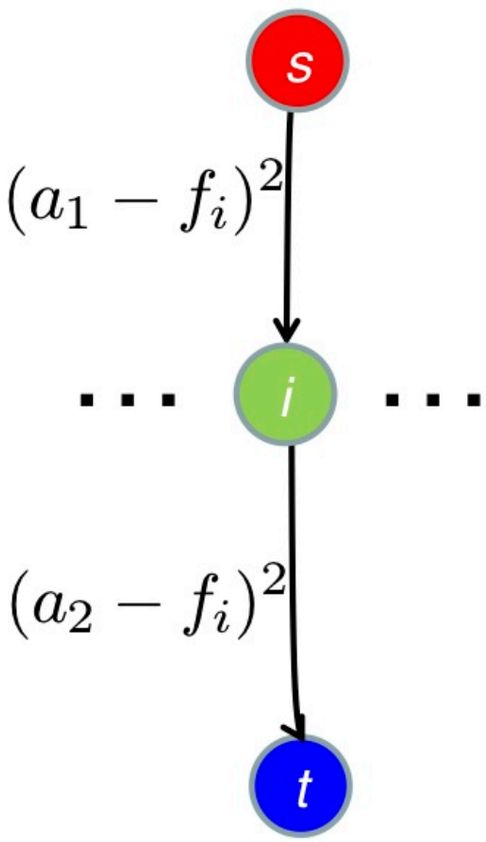


Definition: A *minimum cut* is a cut with minimum cost.

Note: A cut separates all nodes in two sets: (i) nodes that are connected to the source nodes, and (ii) those that are not.

Edge weights – data term

Set edge weights such that a cut corresponds to a solution of the optimization problem



Consider pixel *i*. A cut must contain either:
1. the edge (*s,i*), or
2. the edge (*i,t*)

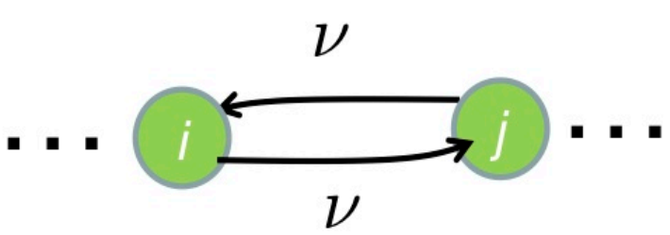
Set edge weights accordingly:
1. $(a_1 - f_i)^2$ for edge (*s,i*),
2. $(a_2 - f_i)^2$ for edge (*i,t*)

data terms

$$E_0(a_1, a_2, \Gamma) = \int_{R_1} (a_1 - f)^2 dxdy + \int_{R_2} (a_2 - f)^2 dxdy + \nu|\Gamma|$$

Edge weights – regularization term

Set edge weights such that a cut corresponds to a solution of the optimization problem



Consider two neighbouring pixels *i* and *j*. If they are in different classes and hence a boundary is passing between them, then a cut must contain either:

1. the edge (*i,j*), or
2. the edge (*j,i*)

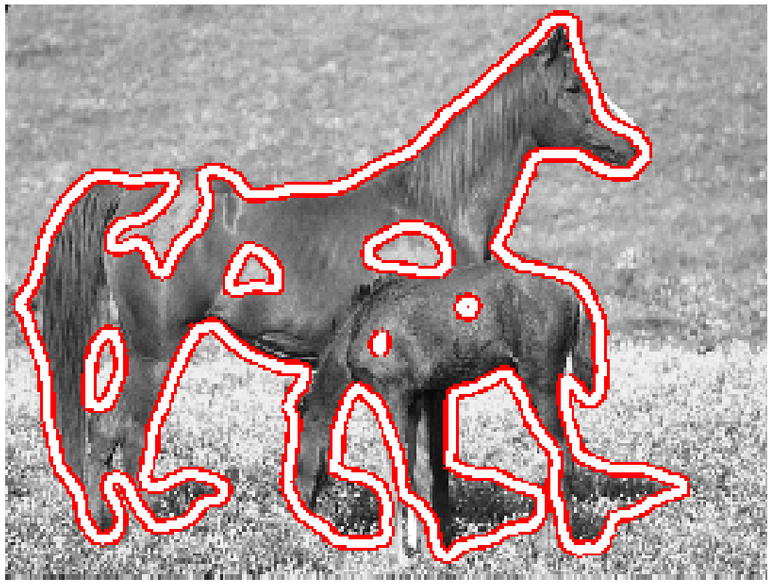
Set edge weights accordingly:

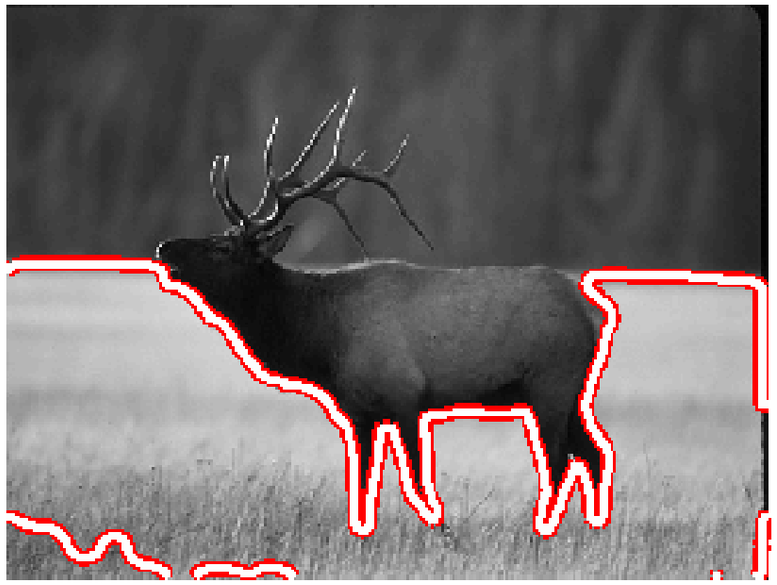
1. ν for edge (*i,j*),
2. ν for edge (*j,i*)

reg. term

$$E_0(a_1, a_2, \Gamma) = \int_{R_1} (a_1 - f)^2 dxdy + \int_{R_2} (a_2 - f)^2 dxdy + \nu|\Gamma|$$

Results of Two-Class Segmentation





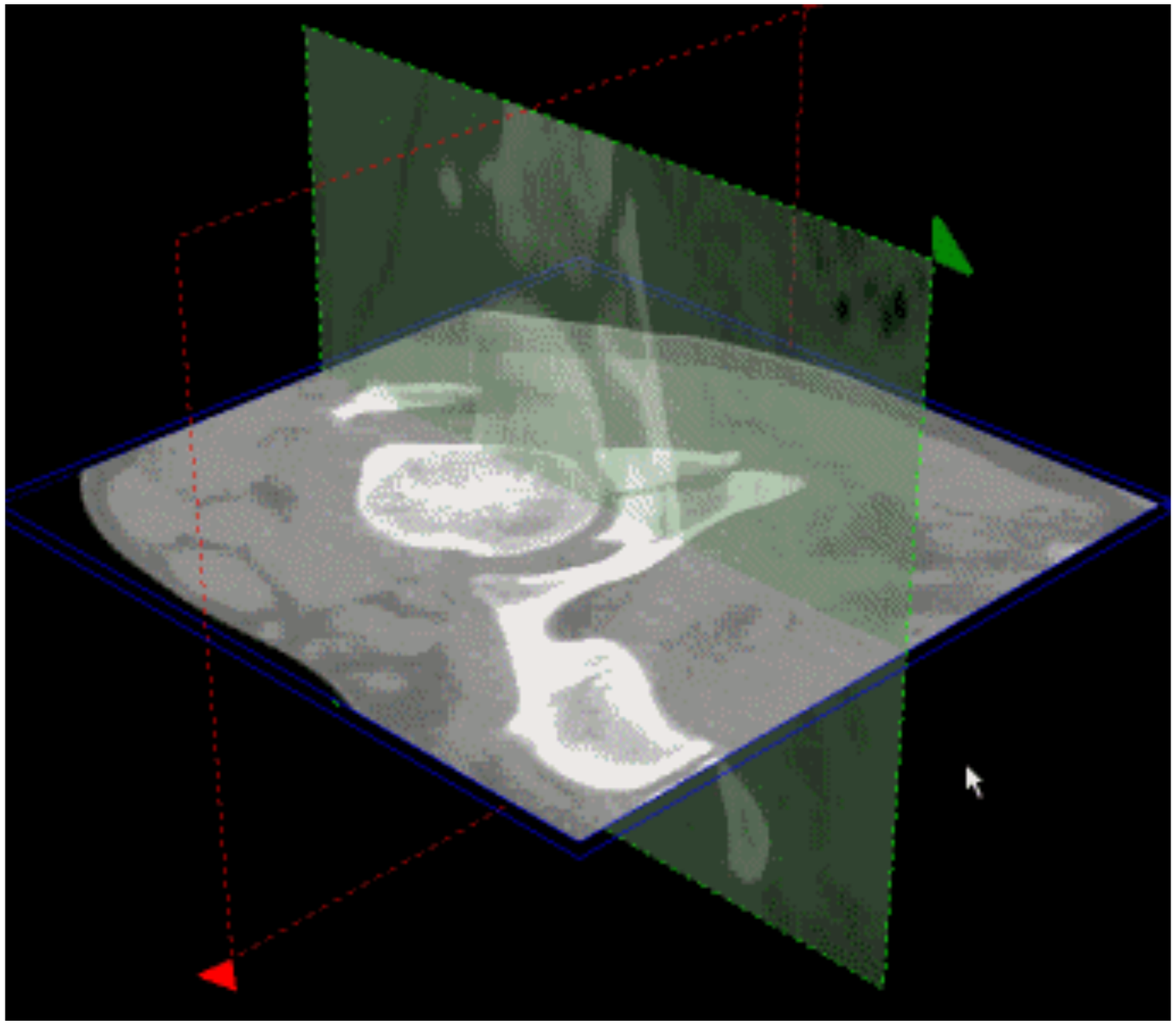
P. Strandmark, F. Kahl, Optimizing Parametric Total Variation Models, *International Conference on Computer Vision*, Sep., Kyoto, Japan 2009.

Example of graph-cut application:
Multi-view volumetric reconstruction

Calibrated images of Lambertian scene

3D model of scene

CVPR'05 slides from Vogiatzis, Torr, Cippola



There are many, many other applications with Graph Cuts in recent years
(recognition, stereo, motion estimation…)

Graph cuts homepage

http://www.cs.cornell.edu/~rdz/graphcuts.html

Wikipedia
http://en.wikipedia.org/wiki/Graph_cuts_in_computer_vision

Review

- Graphs
- Simple graph based segmentation -> eigenvalue problem
- Mumford-Shah functional
- Graph cuts

- Finish assignment 2

LUND UNIVERSITY