Computer Vision II

Other Segmentation Approaches

Carsten Rother, Dmitrij Schlesinger

SS2016, 8.07.2016

COMPUTER VISION LAB DRESDEN

TECHNISCHE UNIVERSITÄT DRESDEN

Outlines

Previously – lots of stuff. CNN-s, graphical models, energy minimization, algorithms, modelling issues ...

Today:

- Clustering and superpixels
- Normalized cuts
- Continuous approaches

Next class:

statistical inference and learning for structured models

Clustering

The pixel inside a segment should be "similar"

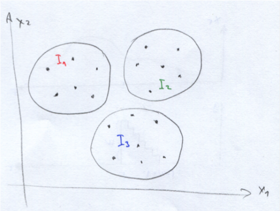
$\rightarrow$ partition the set of pixels into subsets so that the pixels inside one subset are similar $\rightarrow$ clustering

Example: colour segmentation. Dissimilarity: colour difference

Clustering

The task: partition a set of objects into "meaningful" subsets (clusters). The objects in a subset should be "similar"

Notations:



Set of clusters K

Set of indices $i = \{1, 2, \ldots, |I|\}$

Feature vectors $x^i, i \in I$

Partitioning:

$$C = (I_1, I_2, \ldots, I_{|K|}), I_k \cap I_{k'} = \emptyset \text{ for } k \neq k', \ \bigcup_k I_k = I$$

Clustering

Let $x^i \in \mathbb{R}^n$ and each cluster has a "representative" $y^k \in \mathbb{R}^n$

The task reads

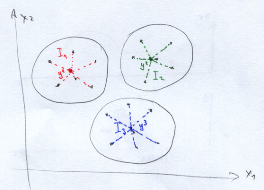
$$\sum_k \sum_{i \in I_k} ||x^i - y^k||^2 \to \min_{C,y}$$

An alternative: the clustering C is a mapping $C : I \to K$ that assigns a cluster number to each $i \in I$

$$\sum_i ||x^i - y^{C(i)}||^2 \to \min_{y,C}$$

$$\sum_i \min_k ||x^i - y^k||^2 \to \min_y$$

Clustering, K-Means algorithm



Initialize centers randomly,

Repeat until convergence:

1. Classify:

$$C(i) = k = \arg\min_{k'} ||x^i - y^{k'}||^2 \quad \Rightarrow \quad i \in I_k$$

2. Update centers:

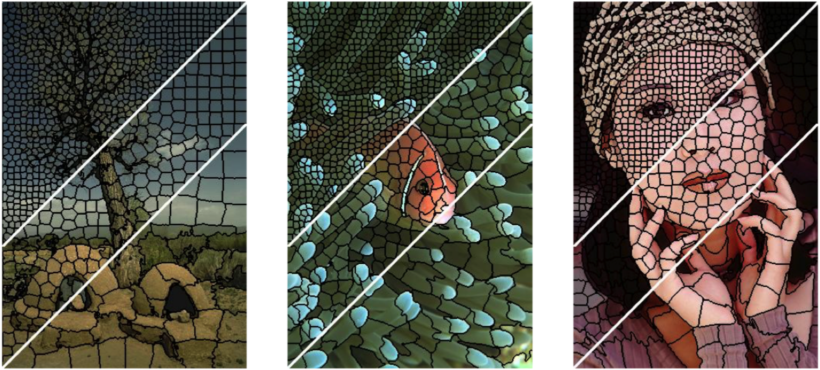
$$y^k = \arg\min_y \sum_{i \in I_k} ||x^i - y||^2 = \frac{1}{|I_k|} \sum_{i \in I_k} x^i$$

– The task is NP

– converges to a local optimum depending on initialization

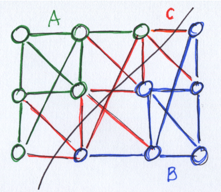
An application – superpixel segmentation

Object are pixels. Features are RGBXY-values.
$\rightarrow$ Those pixels belong to the same cluster that are close to each other both spatially and in in the RGB-space



SLIC: `http://ivrg.epfl.ch/research/superpixels`

Cuts



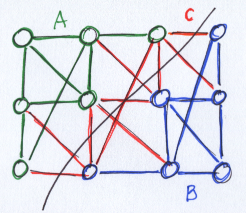
The Idea: the image is a graph $G = (V, E)$, pixels are nodes V, edges E link neighbouring nodes.

Edges are **weighted**, the weights w_{uv} depend on:

- how similar are the corresponding patches (coloring, features etc.);
- how far are the corresponding pixels from each other.

A cut is a (minimal) subset C of edges so, that the "remaining graph" is not connected.

Cuts



Alternatively, it can be seen as a **partitioning** of the node set V into two subsets A and B, i.e. $A \cup B = V$, $A \cap B = \emptyset$

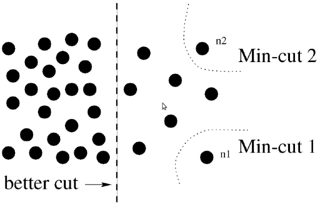
The quality of a cut is the sum of all its edge weights:

$$cut(A, B) = \sum_{u \in A, v \in B} w_{uv}$$

The task is to fing the cut of optimal quality.

Normalized Cut [Shi, Malik, 2000]

Cuts do not account for the cardinality of the partitions – they are "not ballanced"



Association:

$$assoc(A, V) = \sum_{u \in A, t \in V} w_{ut}$$

1) Normalized Cut:

$$NCut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

2) Total normalized association:

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)}$$

Normalized Cut

Interesting – it is the same:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} =$$

$$= \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)} =$$

$$= 2 - Nassoc(A,B) \to \min_{(A,B)}$$

$$\Rightarrow Nassoc(A,B) \to \max_{(A,B)}$$

Normalized Cut – Solution

Let $x_i \in \{+1, -1\}$ be an indicator variable:
$+1$ means "belongs to A", -1 – "belongs to B"

$$Ncut(A,B) = \frac{\sum_{x_i<0, x_j>0} -w_{ij}x_ix_j}{\sum_{x_i>0} d(i)} + \frac{\sum_{x_i>0, x_j<0} -w_{ij}x_ix_j}{\sum_{x_i<0} d(i)}$$

with $d_i = \sum_j w_{ij}$.

$\ldots$

$$\min_x Ncut(x) = \min_y \frac{y^T(D-W)y}{y^TDy}$$

with $y_i \in \{1, -b\}$, $y^TD1 = 0$

D is a diagonal matrix of d_i-s,
W is the matrix of w_{ij}-s,
b is also defined by w_{ij}-s

Normalized Cut – Solution

$$\min_y \frac{y^T(D-W)y}{y^T Dy}, \text{ s.t. } y_i \in \{1, -b\}, y^T D1 = 0$$

This expression is known as so-called **Rayleigh quotient**. For the **relaxed** problem, i.e. $y_i \in [1, -b]$, the solution is obtained by the one of **generalized eigenvalue system**

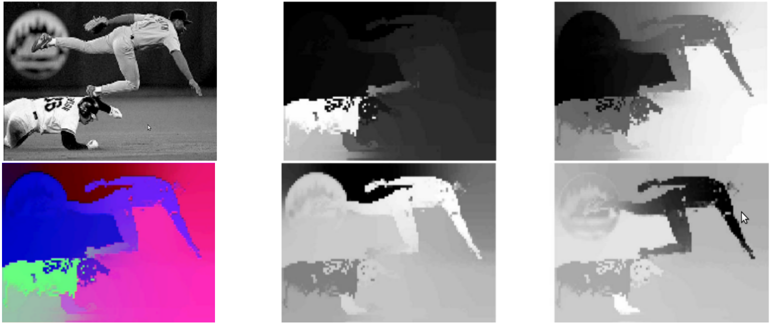
$$(D-W)y = \lambda Dy$$

"... Thus, the second smallest eigenvector of the generalized eigensystem is the real valued solution to our normalized cut problem."

λ-s – **Graph Spectrum** – spectral analysis

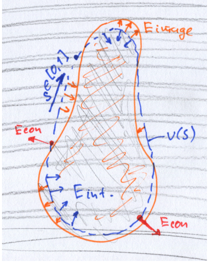
Normalized Cut

"... the eigenvector with the third smallest eigenvalue is the real valued solution that optimally subpartitions the first two parts. In fact, this line of argument can be extended to show that one can subdivide the existing graphs, each time using the eigenvector with the next smallest eigenvalue."



"Soft Segmentation"

Active Contours (Snakes) [Kass, Witkin, Terzopoulos, 1988]



There is a **parametric** definition of a contour:

$$v(s) = \big(x(s), y(s)\big), \quad s \in [0,1]$$

(may be closed ot not)

Its **energy** is:

$$E(v) = \int_0^1 \Big[E_{int}\big(v(s)\big) + E_{image}\big(v(s)\big) + E_{con}\big(v(s)\big)\Big] ds \to \min_v$$

Active Contours

Intrinsic energie:

$$E_{int}\Big(v(s)\Big) = \Big(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2\Big)/2$$

v_s und v_{ss} are the first and the second derivatives respectively

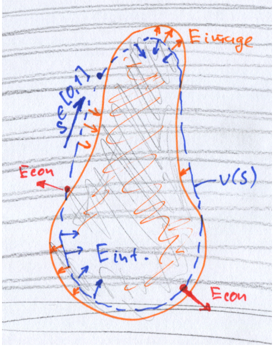
$\alpha(s)$ and $\beta(s)$ are position-specific weights – form model

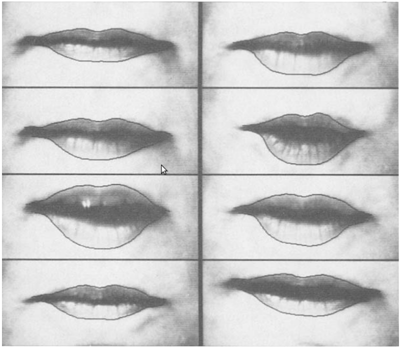
For example, $\beta(s^*) = 0$ means "A corner at s^* is possible"

$E_{image}\Big(v(s)\Big)$ is a data term
– how the image at $\Big(x(s), y(s)\Big)$ should look like
i.e. intensity $I\Big(x(s), y(s)\Big)$, edge $\Big|\nabla I\Big(x(s), y(s)\Big)\Big|^2$

$E_{con}\Big(v(s)\Big)$ are "additional forces" (e.g. user inputs)

Active Contours

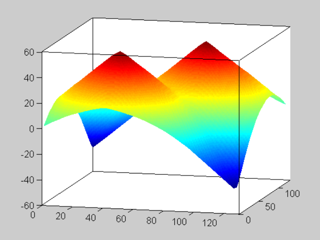




Andrew Blake and Michael Isard: Active Contours
http://www.robots.ox.ac.uk/~contours/

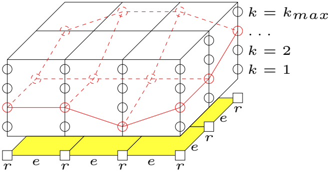
Continuous vs. discrete

Continuous



+ Low-level properties – length, curvature etc.
+ Shape modeling
– Statistical interpretation, learning

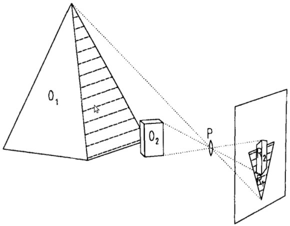
Discrete



– Metrication artifacts
+ Spatial relations
+ Multi-label problems
– Locality
+ Sound statistical interpretation

Mumford-Shah Funktional [Mumford, Shah, 1989]

Motivation:



The observed 2D-Image is a projection of a 3D-scene

Light rays meet objects
$\rightarrow$ image function inside the projection of an object is smooth

Object borders lead to discontinuities of the image function

$\Rightarrow$ image function is piece-wise smooth

Mumford-Shah Funktional

$R \subset \mathbb{R}^2$ – image domain, Γ – discontinuities
$g : R \to \mathbb{R}$ – image function to be approximated
$f : R \to \mathbb{R}$ – piece-wise smooth approximation

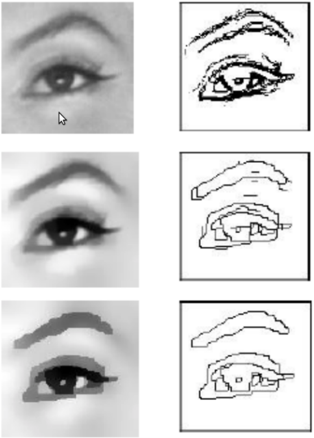
$$E(f,\Gamma) = \mu^2 \int_R (f-g)^2 dxdy + \int_{R-\Gamma} \|\nabla f\|^2 dxdy + \nu|\Gamma| \to \min_{f,\Gamma}$$

The approximation f should

- coincide with the image function as good as possible
- be smooth
- have as less discontinuities as possible

Note: without any of the requirements (energy terms) the problem is trivial

Mumford-Shah Funktional

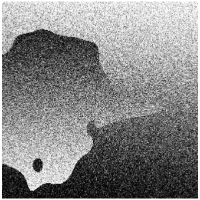


$\leftarrow$ examples for different μ and ν

Topology problem !!!

This is not a segmentation in a common sense – i.e. not a partitioning of the image into disjunct areas

The found edges are not necessarily segment borders



Mumford-Shah Funktional

Special case: piece-wise **constant** approximation – "Cartoon-Limit"

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

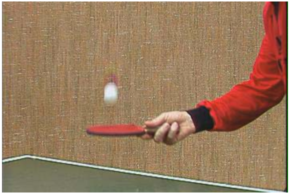
for $\mu,\nu\rightarrow 0$ converges to

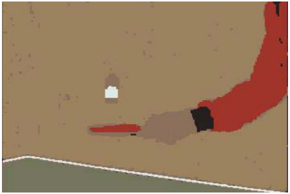
$$E(a,\Gamma)=\sum_i\int_{R_i}(g-a_i)^2 dxdy+\nu|\Gamma|\rightarrow\min_{a,\Gamma}$$

The solution wrt. a_i for fixed partitioning Γ is:

$$a_i=\frac{1}{|R_i|}\int_{R_i} g\ dxdy$$

– the image is partitioned into regions i,
– the colouring inside each region is a constant
Now the topology is correct: edges can not arbitrary end since it is obviously not optimal (due to $|\Gamma|$)





The number of segments in the Cartoon-Limit is not pre-defined – it is estimated as well

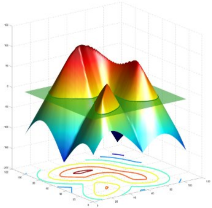
---

Even more specialized model: two regions only:

$$E(c_1, c_2, C) = \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(C) \\ + \lambda_1 \int_{inside(C)} |u_0(x,y) - c_1|^2 dxdy \\ + \lambda_2 \int_{outside(C)} |u_0(x,y) - c_2|^2 dxdy$$

Can be efficiently solved by **Level Set** method

Level Set Method [Osher, Sethian, 1988 ?]



A contour is represented as the zero-level of a **Levelset** function $\phi : \mathbb{R}^2 \to \mathbb{R}$

$$C = \{(x,y) \in \Omega : \phi(x,y) = 0\},$$
$$inside(C) = \{(x,y) \in \Omega : \phi(x,y) > 0\},$$
$$outside(C) = \{(x,y) \in \Omega : \phi(x,y) < 0\}$$

Consider two auxiliary function:
Heaviside function ("step")

$$H(z) = \begin{cases} 1 & \text{if } \; z > 0 \\ 0 & \text{otherwise} \end{cases}$$

and Dirac function ("delta")

$$\delta(z) = \nabla H(z)$$

Level Set Method

The constituents of the Chan-Vese Modell can be written as

$$\text{Length}(C) = \int_\Omega \delta\Big(\phi(x,y)\Big)dxdy,$$

$$\text{Area}(C) = \int_\Omega H\Big(\phi(x,y)\Big)dxdy$$

$$\int_{inside(C)} bla(x,y)dxdy = \int_\Omega H\Big(\phi(x,y)\Big)bla(x,y)dxdy$$

etc.

In short: integrals over areas are replaced by the corresponding integrals over the whole image domain Ω

Other contour characteristics like e.g. curvature can also be expressed relatively easy using these notations

Putting all together:

$$E(\phi) = \int_{\Omega} \Big[\lambda_1 \cdot H(\phi(x,y)) \cdot ||u_0(x,y) - c_1||^2 + \\ + \lambda_2 \cdot (1 - H(\phi(x,y))) \cdot ||u_0(x,y) - c_2||^2 + \\ + \mu \cdot \delta(\phi(x,y)) \Big] dxdy \rightarrow \min_{\phi}$$

Note: E is a mapping, argument of which is a function !!!

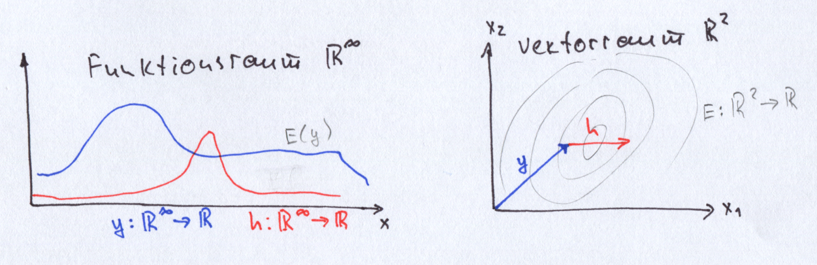
What is

$$\frac{\partial E(\phi)}{\partial \phi} \quad ?$$

Calculus of Variations

A function $y : \Omega \to \mathbb{R}$ can be understood as a "vector" in a space of infinite dimension, i.e. $y \in \mathbb{R}^\infty$.

Correspondingly, an energy functional $E(y)$ is $E : \mathbb{R}^\infty \to \mathbb{R}$.



Gâteaux derivative along a "direction" $h : \Omega \to \mathbb{R}$:

$$\left.\frac{\partial E(y)}{\partial y}\right|_h = \lim_{\varepsilon \to 0} \frac{E(y + \varepsilon h) - E(y)}{\varepsilon} = \left.\frac{dE(y + \varepsilon h)}{d\varepsilon}\right|_{\varepsilon=0}$$

Calculus of Variations, "standard" derivation chain

1. We have an expression for $E(f,\ldots) = \int_\Omega \ldots$
2. Write down the Gâteaux derivatives $\left.\frac{dE(f+\varepsilon h)}{d\varepsilon}\right|_{\varepsilon=0}$
3. Require $\left.\frac{dE(f+\varepsilon h)}{d\varepsilon}\right|_{\varepsilon=0} = 0$ for all $h \in \mathbb{R}^\infty$ – also known as **Euler-Lagrange equations**
4. Consider them not for all $h \in \mathbb{R}^\infty$ but only for the "basis" (delta-functions)

$$h(x,y) = \begin{cases} 1 & \text{if } (x,y) = (x^*,y^*) \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } (x^*,y^*) \in \Omega$$

5. Consider them only for integer positions $(x^*,y^*) \in \mathbb{Z}^2$
6. Replace the necessary stuff (gradients, divergences, Laplacians etc.) by **finite differences**

   $\rightarrow$ obtain "usual" gradient (finite number of variables)

Level Set Method

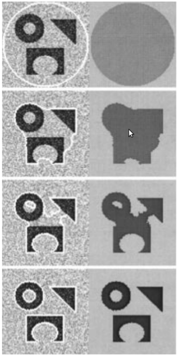
Approach:

Note: H and δ are not differentiable
$\rightarrow$ replace them by something differentiable (sigmoid)

Then:

- either
  - set **Gateau derivatives** to zero
    $\rightarrow$ obtain **Euler-Lagrange equations**
  - discretize $\rightarrow$ obtain a system of equations
  - resolve it by some iterative methods
- or
  - Gateau derivatives $\rightarrow$ gragient
  - gradient descent

Level Set Method, Summary



Topology is correct,
it may change during the contour evaluation

Less sensitive to the initialization as compared to the Active Contours

Many things can be expressed in a convinient and unified manner

---

Generalization to the more than two segments needs effort

Statistic interpretation is in question
$\Rightarrow$ problems to learn unknown parameters
(typical for almost all continuous approaches)

Take home messages

There is a **lot** of very different segmentation approaches over the (Computer Vision) world.

Sometimes, simple ones are preferable (like e.g. clustering or active contours).

Each one has its application area (remember on the comparison “Continuous vs. discrete”).

Next class:

statistical inference and learning for structured models