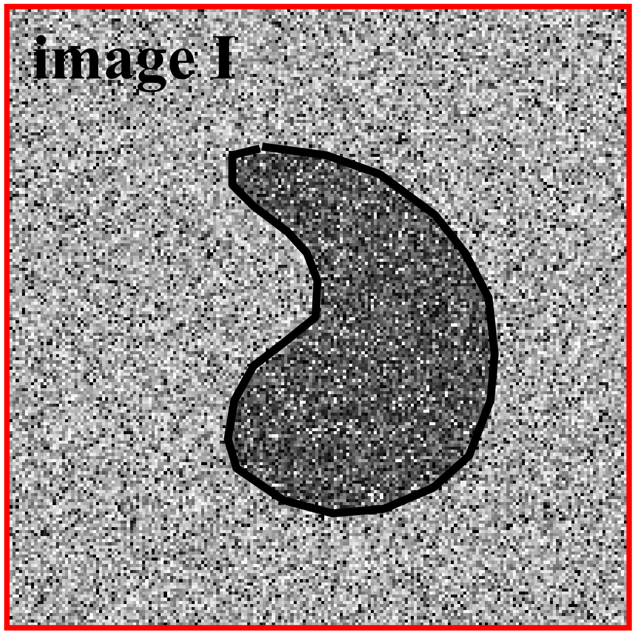
Two Probabilistic Approaches to Deformable Contours

Mário A. T. Figueiredo
"Instituto Superior Técnico"
Lisboa
PORTUGAL

Invited talk, *WDM2000 - Summer School and Workshop on Deformable Models*,
Gullmarstrand, Sweden, August 2000.

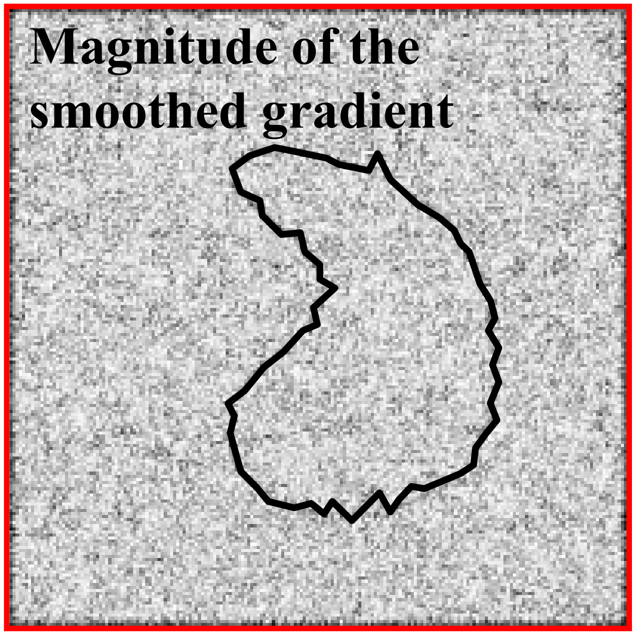
- **PART I - Snakes**
  - A brief review of standard snakes
  - A very brief review of Bayesian inference
  - The Bayesian interpretation of standard snakes
  - A Bayesian approach to (region-based) snakes
- **PART II – Parametrically Deformable Contours**
  - Introduction
  - A review of splines and B-splines
  - The model selection issue
  - A brief review of the MDL principle
  - An MDL-based approach and its implementation



Goal: deform snake (**v**) under the "image forces", to "find the contour".

Potential energy field $E_{ext}(\mathbf{v},\mathbf{I})$

For example, to attract **v** towards high-gradient regions (boundaries)

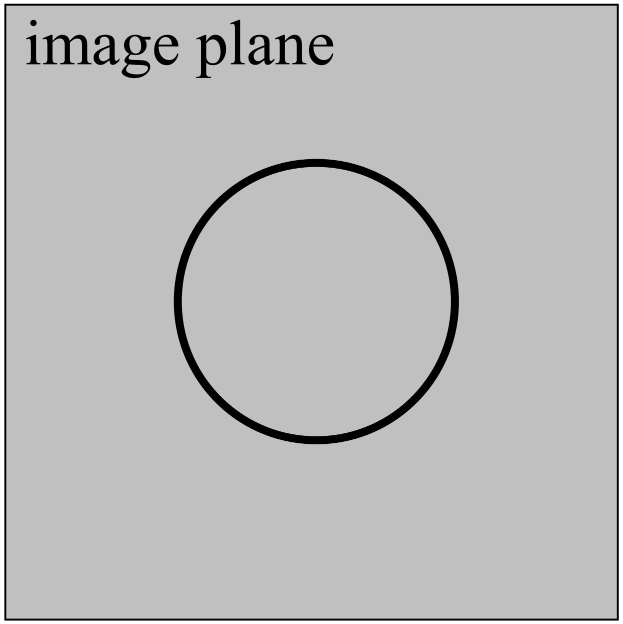


A configuration with low $E_{ext}(v,I)$

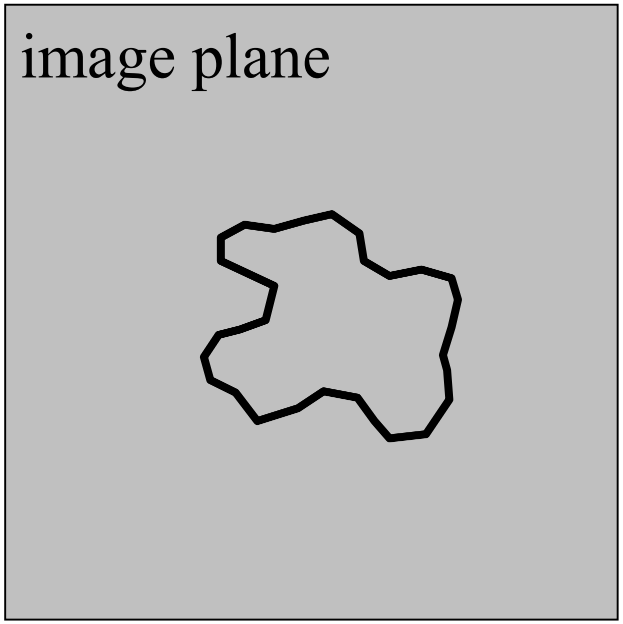
Obvious problem:
this field may be "noisy",
thus a curve with low $E_{ext}(\mathbf{v},\mathbf{I})$
is a "noisy" curve.

An elastically deformable line, **v**, on the image plane, …

$E_{int}(\mathbf{v})$ → elastic potential (internal) energy, under deformation



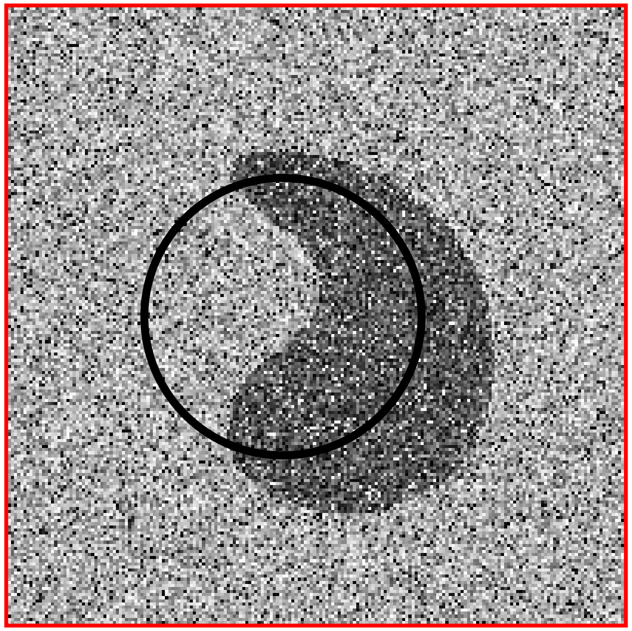
shape at rest (low E(**v**))



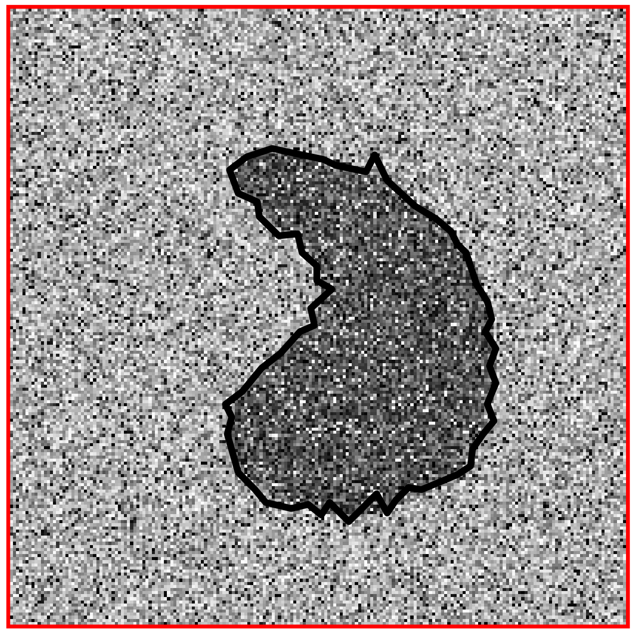
deformed shape (higher E(**v**))

Snakes

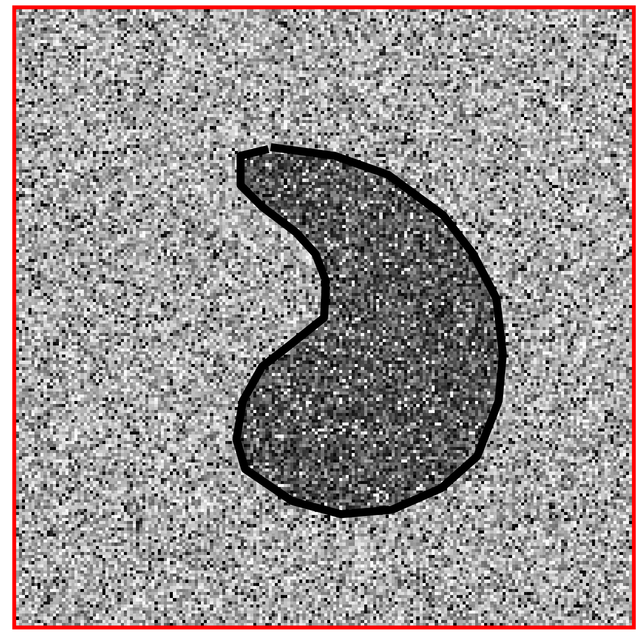
The “snake” approach: combine the two energies



Minimizer of $E_{int}(\mathbf{v})$



Minimizer of $E_{ext}(\mathbf{v},\mathbf{I})$



A good compromise: $\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \left\{ E_{ext}(\mathbf{v}, \mathbf{I}) + \alpha\, E_{int}(\mathbf{v}) \right\}$

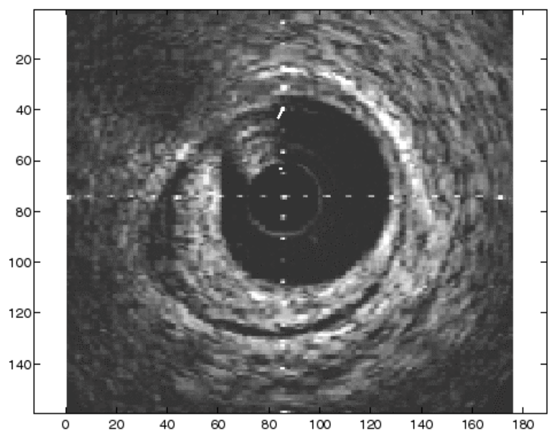
Most image analysis problems can/should be formulated as

Given observed data **g**, infer **f**

This is a trivial statement.

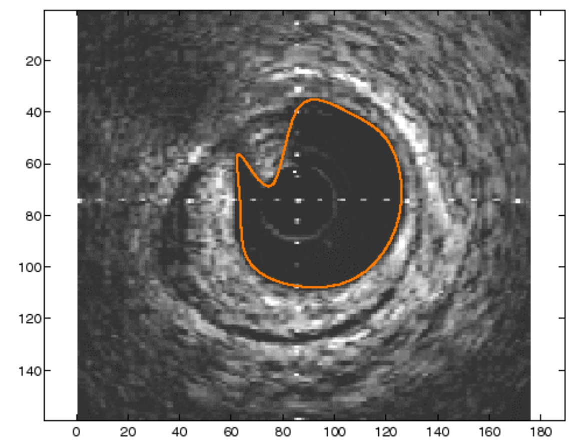
The message: “start by formalizing **f** and **g**”

- Example:

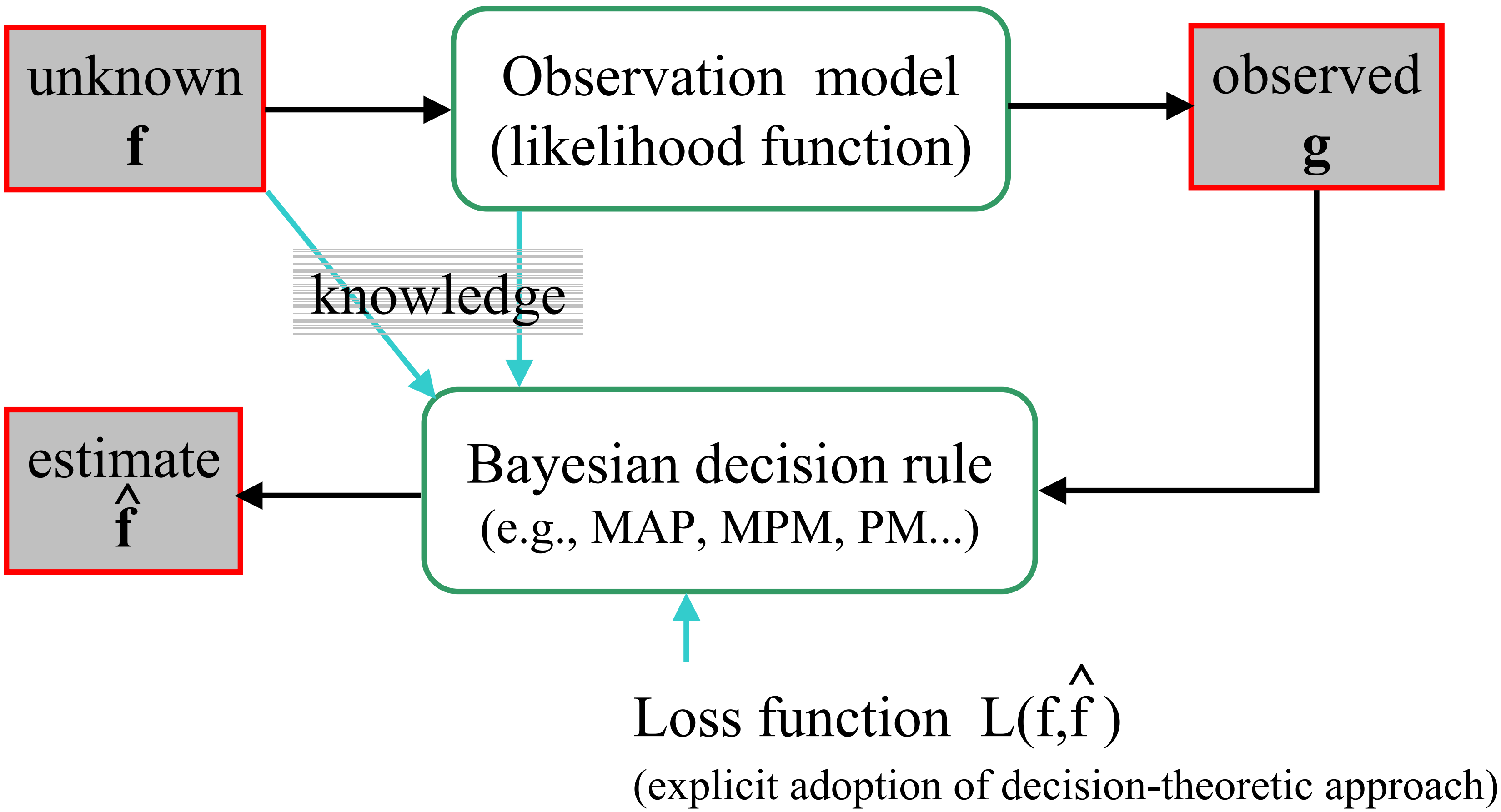


g, an observed image

inference



f, a contour, e.g., represented by a sequence of points



The Bayesian approach is explicitly model-based

- Observation model / likelihood function:

$$p(\mathbf{g} \,|\, \mathbf{f}, \phi)$$

$\mathbf{f}$ is the unknown
$\mathbf{g}$ is the observed data
ϕ are parameters

- Prior knowledge:

$$p(\mathbf{f} \,|\, \psi)$$

$\mathbf{f}$ is the unknown
ψ are parameters

- *A posteriori* knowledge, i.e., knowledge about **f** after observing **g**

Bayes law:

$$p(\mathbf{f} \,|\, \mathbf{g}, \phi, \psi) = \frac{p(\mathbf{g} \,|\, \mathbf{f}, \phi)\, p(\mathbf{f} \,|\, \psi)}{p(\mathbf{g} \,|\, \phi, \psi)}$$

Given $p(\mathbf{f} \mid \mathbf{g}, \phi, \psi)$ and a loss function $L(\mathbf{f}, \hat{\mathbf{f}})$

Optimal Bayes rule: minimizer of the *a posteriori* expected loss:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \int L(\mathbf{f}, \hat{\mathbf{f}})\, p(\mathbf{f} \mid \mathbf{g}, \phi, \psi)\, d\mathbf{f}$$

Particular case: the *maximum a posteriori* rule (0/1 loss)

$$L(\mathbf{f}, \hat{\mathbf{f}}) = \begin{cases} 1 \Leftarrow \mathbf{f} \neq \hat{\mathbf{f}} \\ 0 \Leftarrow \mathbf{f} \neq \hat{\mathbf{f}} \end{cases} \rightarrow \quad \hat{\mathbf{f}}_{\text{MAP}} = \arg\max_{\mathbf{f}} p(\mathbf{f} \mid \mathbf{g}, \phi, \psi)$$

$$= \arg\max_{\mathbf{f}} \left\{ p(\mathbf{g} \mid \mathbf{f}, \phi)\, p(\mathbf{f} \mid \psi) \right\}$$

$$\hat{\mathbf{f}}_{\text{MAP}} = \arg\max_{\mathbf{f}} \left\{ \log p(\mathbf{g} \mid \mathbf{f}, \phi) + \log p(\mathbf{f} \mid \psi) \right\}$$

Particular case of MAP: the *maximum likelihood* (ML) criterion

$$p(\mathbf{f} \mid \psi) \propto \text{const.} \quad \rightarrow \quad \hat{\mathbf{f}}_{\text{ML}} = \arg\max_{\mathbf{f}} \log p(\mathbf{g} \mid \mathbf{f}, \phi)$$

MAP rule: $$\hat{\mathbf{f}}_{MAP} = \arg\min_{\mathbf{f}} \left\{ -\log p(\mathbf{g} \mid \mathbf{f}, \phi) - \log p(\mathbf{f} \mid \psi) \right\}$$

Snake "rule": $$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \left\{ E_{ext}(\mathbf{v}, \mathbf{I}) + \alpha\, E_{int}(\mathbf{v}) \right\}$$

The similarity suggests: $$p(\mathbf{v}) = \frac{1}{Z_{int}} \exp\left\{ -\alpha\, E_{int}(\mathbf{v}) \right\}$$

$$p(\mathbf{I} \mid \mathbf{v}) = \frac{1}{Z_{ext}(\mathbf{v})} \exp\left\{ -E_{ext}(\mathbf{v}, \mathbf{I}) \right\}$$

Then, $$\hat{\mathbf{v}}_{MAP} = \arg\min_{\mathbf{v}} \left\{ E_{ext}(\mathbf{v}, \mathbf{I}) + \alpha\, E_{int}(\mathbf{v}) \right\}$$

if and only if $Z_{ext}(\mathbf{v}) = Z_{ext}$ …often not true.

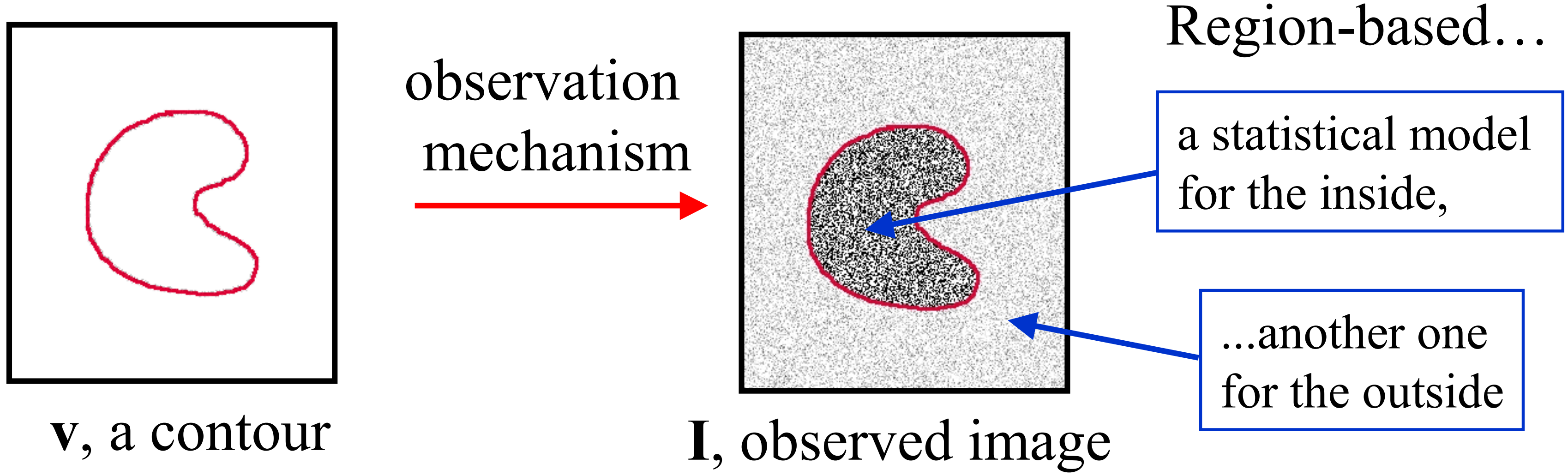
•Standard snakes (Witkin, Kass, Terzopoulos, 1987):
- Internal energy: squared first and second derivatives (Sobolev norm)
- External energy: $-|\nabla I|^2$
- Iterative energy minimization

•Drawbacks of standard snakes:
- myopia (only see data close to current position)
- unable to re-parameterize or change topology
- non-adaptive: parameters (e.g. α) have to be set *a priori*

•Many descendants of snakes have addressed some drawbacks:

Chakaraborty, Staib, & Duncan, 1994; Cohen & Cohen, 1993;
McInerney & Terzopoulos, 1995; Radeva, Serra, & Marti, 1995; Ronfard, 1994;
Xu & Prince, 1998; Zhu & Yuille, 1996, many others….

Region-based…



Under inside/outside independence assumption:

$$p(\mathbf{I} \mid \mathbf{v}, \phi_{in}, \phi_{out}) = p(\mathbf{I}_{inside(\mathbf{v})} \mid \phi_{in}) \; p(\mathbf{I}_{outside(\mathbf{v})} \mid \phi_{out})$$

Examples: - Gaussian of different mean and/or variance;

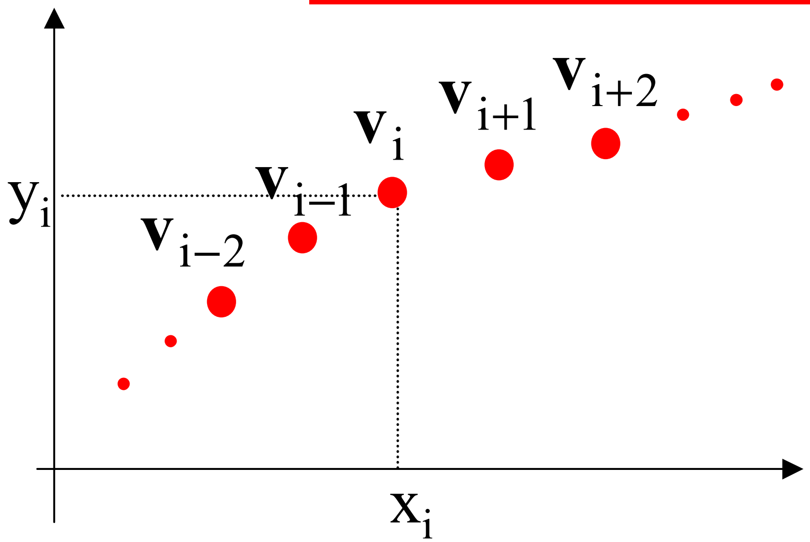
- Rayleigh of different variance (ultrasound images);
- Different textures, …

This type of region model also considered by:

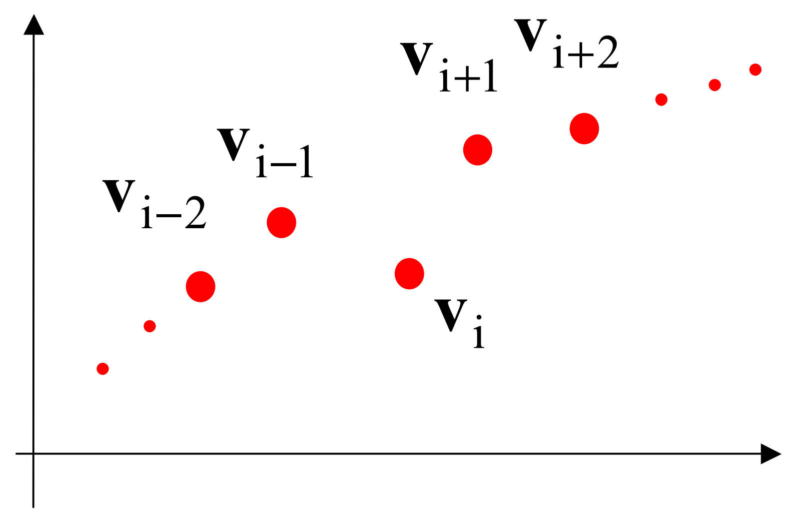
Ronfard, 1994; Chakaraborty, Staib, & Duncan, 1994;
Zhu & Yuille, 1996, Dias & Leitão, 1996; Figueiredo, Leitão & Jain, 1997,…

$$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n) \qquad \text{where} \qquad \mathbf{v}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

- Prior knowledge: **v** is "smooth"



More probable



Less probable

1-D Markov random field

$$p(\mathbf{v} \mid \psi) = \frac{1}{Z} \exp\left\{ -\frac{1}{\psi} \sum_i (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \right\}$$

Likelihood function:

$$p(\mathbf{I} \mid \mathbf{v}, \phi_{in}, \phi_{out}) = p(\mathbf{I}_{inside(\mathbf{v})} \mid \phi_{in}) \; p(\mathbf{I}_{outside(\mathbf{v})} \mid \phi_{out})$$

Example: assuming i.i.d. Gaussian pixels values:

$$\phi_{in} = \left(\mu_{in}, \sigma_{in}^2\right) \qquad \phi_{out} = \left(\mu_{out}, \sigma_{out}^2\right)$$

$$p(\mathbf{I} \mid \mathbf{v}, \phi_{in}, \phi_{out}) = \prod_{i \in inside(\mathbf{v})} N(I_i \mid \mu_{in}, \sigma_{in}^2) \prod_{i \in outside(\mathbf{v})} N(I_i \mid \mu_{out}, \sigma_{out}^2)$$

Note: $Z_{ext}(\mathbf{v})$ is not constant: $Z_{ext}(\mathbf{v}) \propto \left(\sigma_{in}^2\right)^{-N_{in}(\mathbf{v})} \left(\sigma_{out}^2\right)^{-N_{out}(\mathbf{v})}$

number of pixels inside/outside **v**

Likelihood function:

$$p(\mathbf{I} \mid \mathbf{v}, \phi_{in}, \phi_{out}) = \prod_{i \in inside(\mathbf{v})} p(I_i \mid \phi_{in}) \prod_{i \in outside(\mathbf{v})} N(I_i \mid \phi_{out})$$

Prior:

$$p(\mathbf{v} \mid \psi) = \frac{1}{Z} \exp\left\{ -\frac{1}{\psi} \sum_i (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \right\}$$

$$\hat{\mathbf{v}}_{MAP} = \arg\min_{\mathbf{v}} \{ -\log p(\mathbf{v} \mid \psi) - \log p(I \mid \mathbf{v}, \phi_{in}, \phi_{out}) \}$$

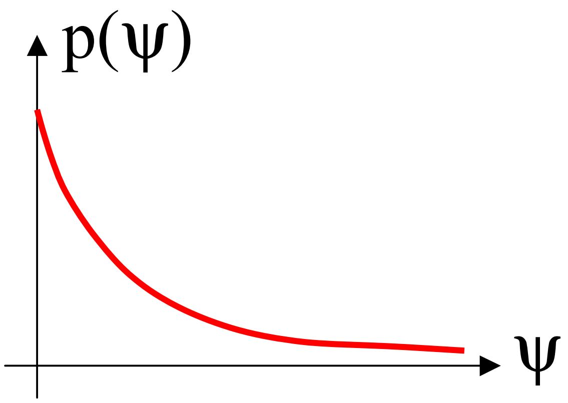
Questions: - How to find the maximum?

- What about the parameters? $(\psi, \phi_{in}, \phi_{out})$

Advantage of a probabilistic approach:
the parameters have meanings and can be estimated

• A (hyper)prior for ψ: $p(\psi) \propto \exp\{-a\psi\}, \quad \psi \geq 0$

…expressing preference for "smoother" contours



• A flat prior for the likelihood parameters

$$p(\phi_{in}, \phi_{out}) \propto \text{const.}$$

$$(\hat{\mathbf{v}}, \hat{\psi}, \hat{\phi}_{in}, \hat{\phi}_{out}) = \arg \min_{\mathbf{v}, \psi, \phi_{in}, \phi_{out}} \{-\log p(\mathbf{v} \mid \psi) - \log p(\psi) - \log p(\mathbf{I} \mid \mathbf{v}, \phi_{in}, \phi_{out})\}$$

Adaptive ICM, or component-wise iterative optimization

Iterated conditional modes (Besag, 1986)

Step 0 → Initialization: get initial contour $\hat{\mathbf{v}}^{(0)}$

set $t = 0$

Step 1 → Given $\hat{\mathbf{v}}^{(t)}$, update the parameter estimates:

$$\hat{\psi}^{(t+1)} = \arg\min_{\psi}\left\{-\log p(\psi) - \log p(\hat{\mathbf{v}}^{(t)} \mid \psi)\right\}$$

(MAP estimate)

$$(\hat{\phi}_{in}, \hat{\phi}_{out})^{(t+1)} = \arg\min_{\phi_{in}, \phi_{out}}\left\{-\log p(\mathbf{I} \mid \hat{\mathbf{v}}^{(t)}, \phi_{in}, \phi_{out})\right\}$$

(ML estimates)

Step 2 → Update contour by performing 1 ICM step: $\hat{\mathbf{v}}^{(t+1)}$

Convergence? Yes: stop; no: back to Step 1

Step 2 → Update contour by performing 1 ICM step: $\hat{\mathbf{v}}^{(t+1)}$

Given the current parameter estimates

$$\hat{\phi}^{(t+1)} \equiv (\hat{\phi}_{in}, \hat{\phi}_{out})^{(t+1)} \quad \text{and} \quad \hat{\psi}^{(t+1)},$$

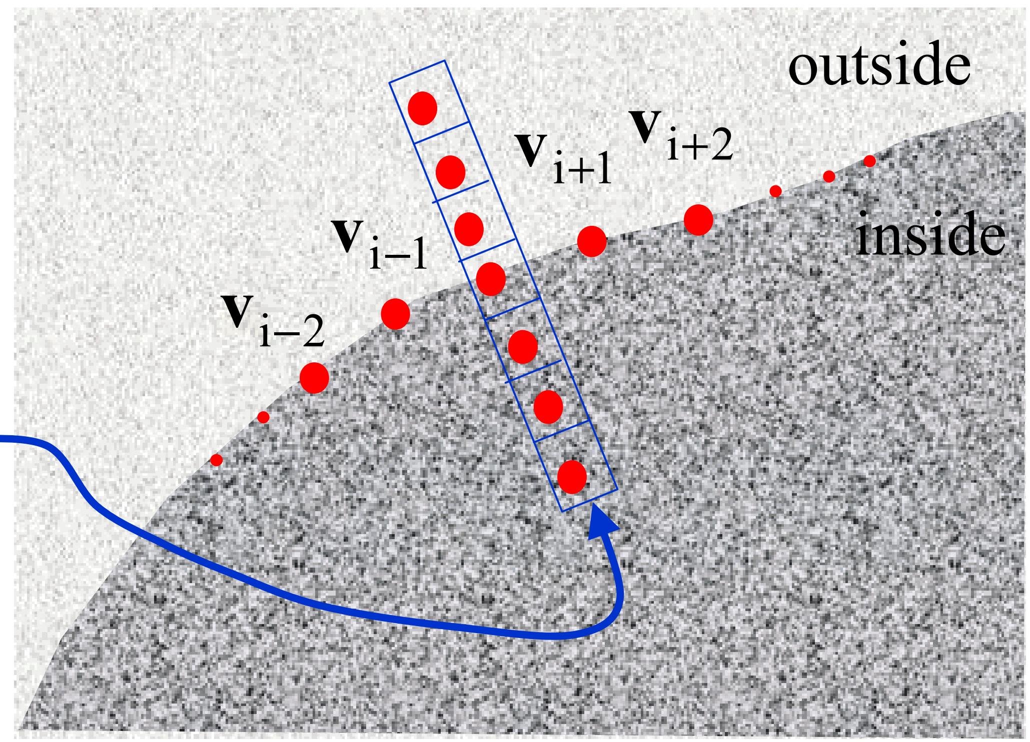
$$-\log p(\mathbf{v} \mid \hat{\psi}^{(t+1)}) - \log p(\mathbf{I} \mid \mathbf{v}, \hat{\phi}^{(t+1)}) \equiv E(\mathbf{v})$$ is non-convex in **v**

ICM

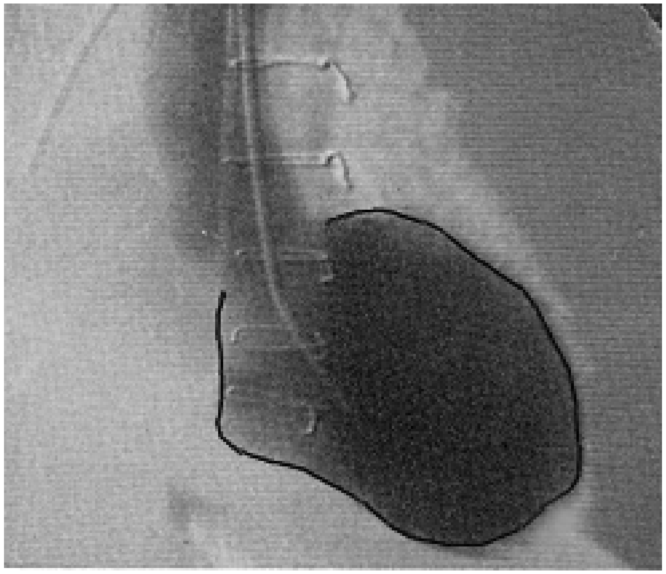
for each i=1,2,…,n

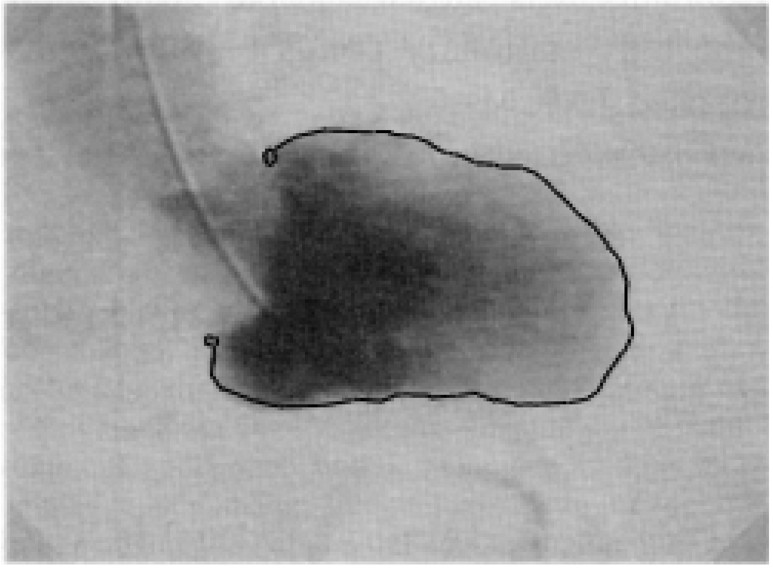
$$\hat{v}_i^{(t+1)} = \arg\min_{v_i} E(\mathbf{v} \mid \{v_{j \neq i}\} \text{ fixed})$$

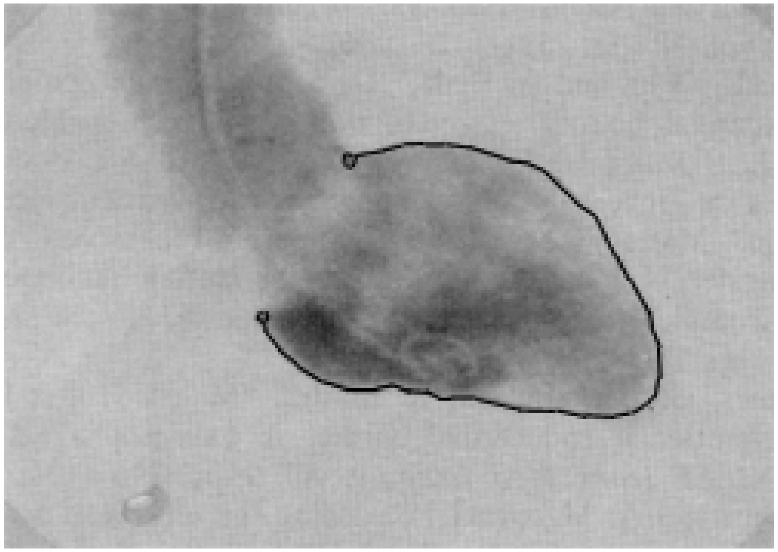
under the constraint $\hat{v}_i^{(t+1)} \in$



Alternatives: dynamic programming, simulated annealing,…







For more details, see:
M. Figueiredo and J. Leitão, “Bayesian estimation of ventricular contours”, in IEEE Trans. on Medical Imaging., vol. 11, pp. 416-429, 1992

PART II – Parametrically Deformable Contours

•Standard snakes: "explicit" contour description $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n)$

(nonparametric)

•Parametrically deformable contours:

- parametric, usually "short" description $\mathbf{v} = M(\boldsymbol{\theta})$

- Examples: Fourier descriptors (Staib & Duncan, 1992; Jain, Zhong, & Lakshmanan, 1996; Figueiredo, Leitão, & Jain, 1997)

  Splines (Menet, Saint-Marc, & Medioni, 1990; Rueckert & Burger, 1995; Amini, Curwen, and Gore, 1996; Dias, 1999; Cham & Cipolla, 1999)

  Wavelets (Chuang and Kuo, 1996)

  Polygons (Jolly, Lakshmanan, & Jain, 1996)

  Sinc functions (Dias, 1999).

  Application-specific models (many authors…)

Parametric description $\mathbf{v} = M(\boldsymbol{\theta})$

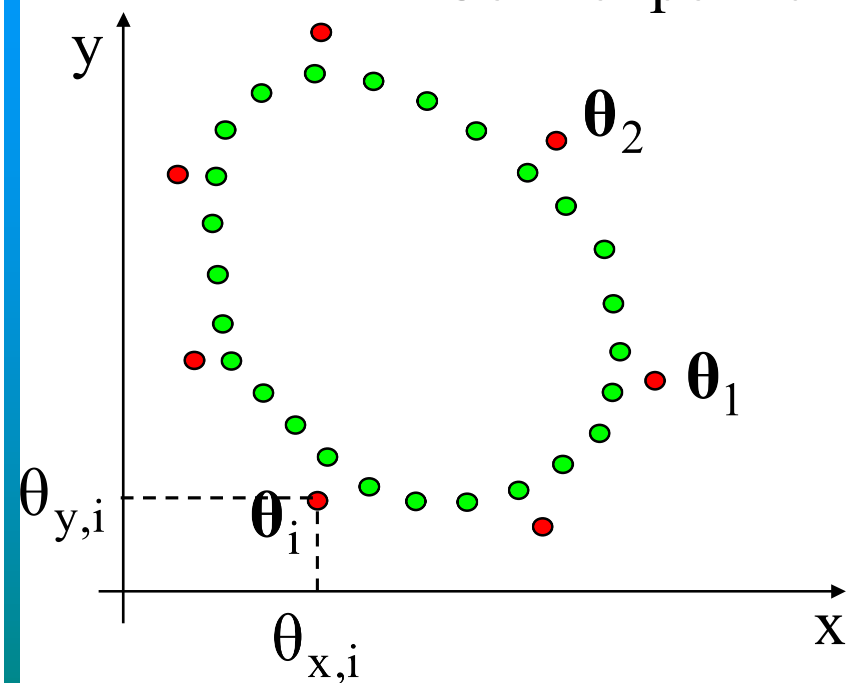
Usually:

Parameterization order ⟷ smoothness/simplicity of $\mathbf{v}$

Examples

- Fourier descriptors with few (low frequency) terms: smooth curves
- Polygon with few vertices: simple shapes
- Spline descriptors with few control points: smooth curves
- Small sinc-basis: low bandwidth (smooth) curves

Control points

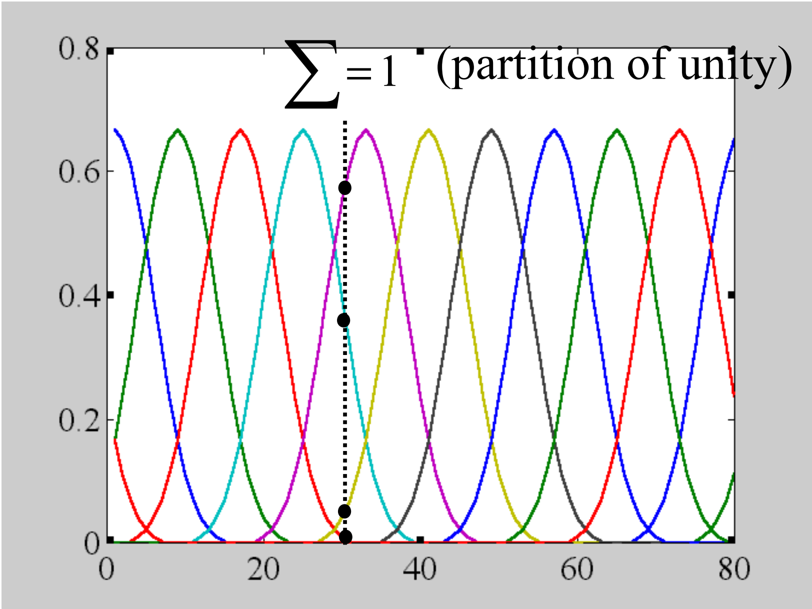


$$\begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} = \mathbf{B} \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_k \end{bmatrix} = \mathrm{B} \begin{bmatrix} \theta_{x,1} & \theta_{y,1} \\ \vdots & \vdots \\ \theta_{x,k} & \theta_{y,k} \end{bmatrix}$$

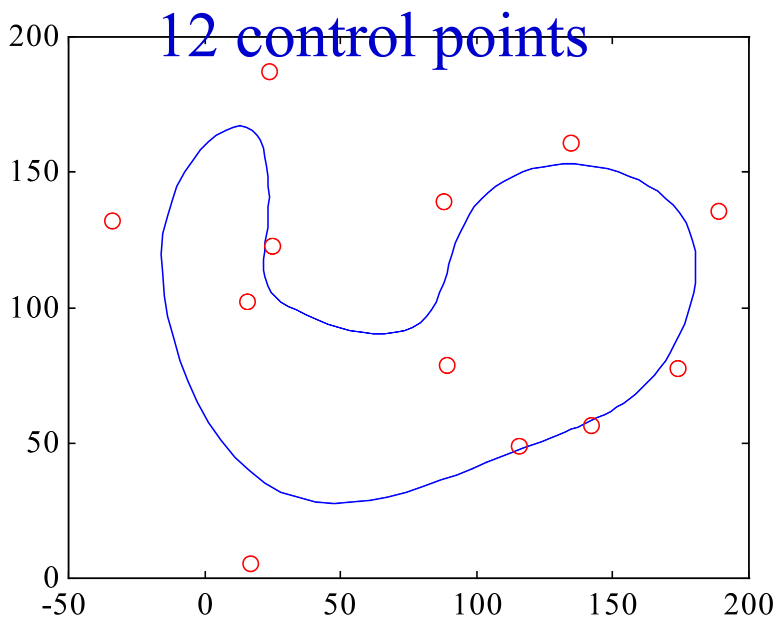
$$\mathbf{v} = [\mathbf{x} \quad \mathbf{y}] = \mathbf{B}\,[\boldsymbol{\theta}_x \quad \boldsymbol{\theta}_y] \Leftrightarrow \begin{cases} \mathrm{x} = \mathbf{B}\,\boldsymbol{\theta}_x \\ \mathrm{y} = \mathbf{B}\,\boldsymbol{\theta}_y \end{cases}$$

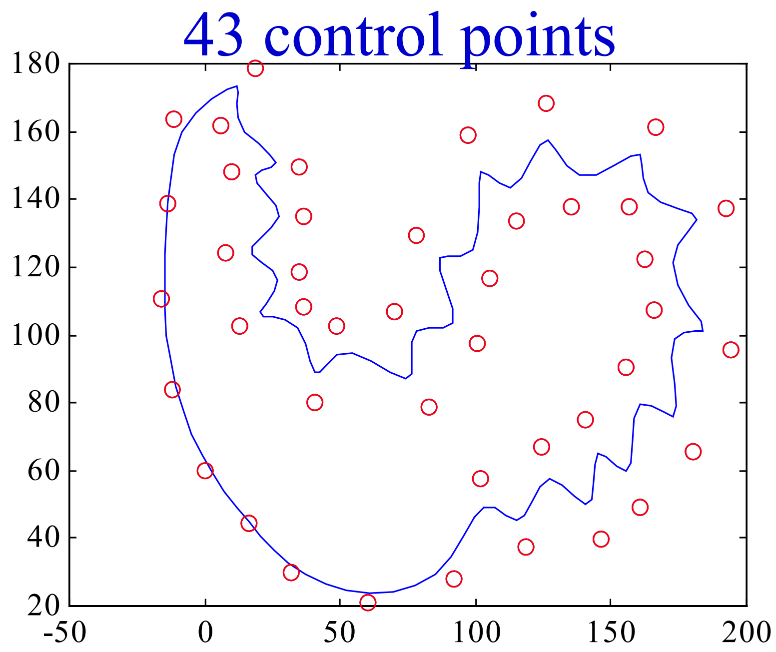
B $(n \times k)$ → Matrix of (discretized) periodic (cubic) B-spline basis

Columns of B (n=80, k=10) →



Number of control points: curve complexity





Given a set of points $\begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} = [\mathbf{x} \quad \mathbf{y}]$

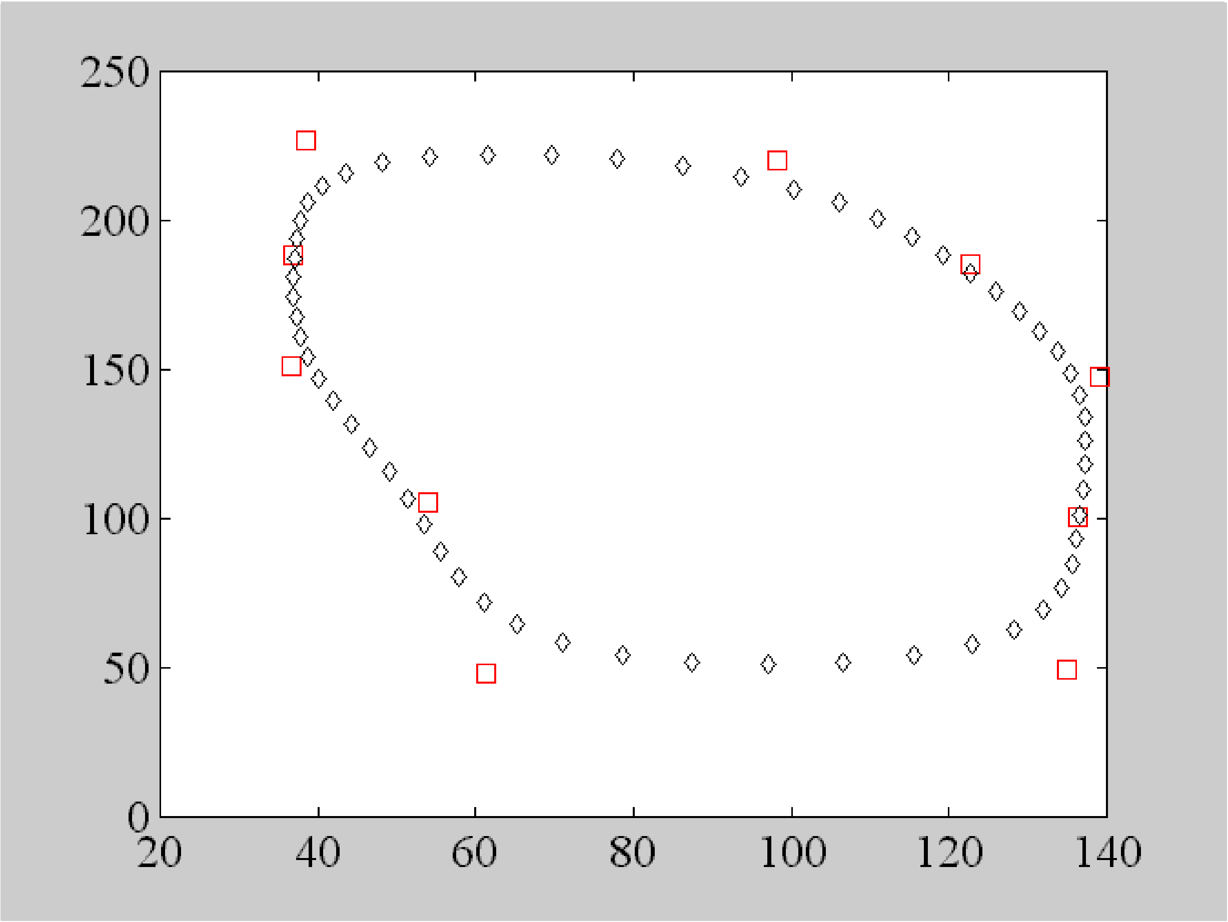
…and a B-spline matrix **B**, find the "best" control points.

in mean square sense

$$\hat{\boldsymbol{\theta}}_x = \arg\min_{\boldsymbol{\theta}_x} \left\| \mathbf{x} - \mathbf{B}\boldsymbol{\theta}_x \right\|^2 \qquad \hat{\boldsymbol{\theta}}_y = \arg\min_{\boldsymbol{\theta}_x} \left\| \mathbf{y} - \mathbf{B}\boldsymbol{\theta}_y \right\|^2$$

Solution: $\hat{\boldsymbol{\theta}}_x = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{x} = \mathbf{B}^{\#}\mathbf{x}$ $\qquad \hat{\boldsymbol{\theta}}_y = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{y} = \mathbf{B}^{\#}\mathbf{y}$

$B^{\#}$, pseudo-inverse of **B**



(Noisy) points
$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$

Control points
$\hat{\boldsymbol{\theta}} \equiv \mathbf{B}^{\#}\mathbf{v}$

Smoothed points
$$\mathbf{s} = \mathbf{B}\hat{\boldsymbol{\theta}} = \mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{v}$$

Key question:
how many control points ?

$$\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T \equiv \mathbf{B}^{\perp}$$

Orthogonal projection matrix

Projects v onto the span of the columns of B

Consider an i.i.d. Gaussian noise model:

$$p(\mathbf{x},\mathbf{y}\,|\,\boldsymbol{\theta}_x,\sigma_x^2,\boldsymbol{\theta}_y,\sigma_y^2) = p(\mathbf{y}\,|\,\boldsymbol{\theta}_y,\sigma_y^2)\,p(\mathbf{x}\,|\,\boldsymbol{\theta}_x,\sigma_x^2)$$

$$p(\mathbf{x}\,|\,\boldsymbol{\theta}_x) \propto \exp\left\{-\frac{\left\|\mathbf{x}-\mathbf{B}\,\boldsymbol{\theta}_x\right\|^2}{2\,\sigma_x^2}\right\} \qquad p(\mathbf{y}\,|\,\boldsymbol{\theta}_y) \propto \exp\left\{-\frac{\left\|\mathbf{y}-\mathbf{B}\,\boldsymbol{\theta}_y\right\|^2}{2\,\sigma_y^2}\right\}$$

Then, the ML estimate is the minimum mean square error estimate:

$$\hat{\boldsymbol{\theta}}_x = \arg\min_{\boldsymbol{\theta}_x}\left\|\mathbf{x}-\mathbf{B}\boldsymbol{\theta}_x\right\|^2 \qquad \hat{\boldsymbol{\theta}}_y = \arg\min_{\boldsymbol{\theta}_x}\left\|\mathbf{y}-\mathbf{B}\boldsymbol{\theta}_y\right\|^2$$

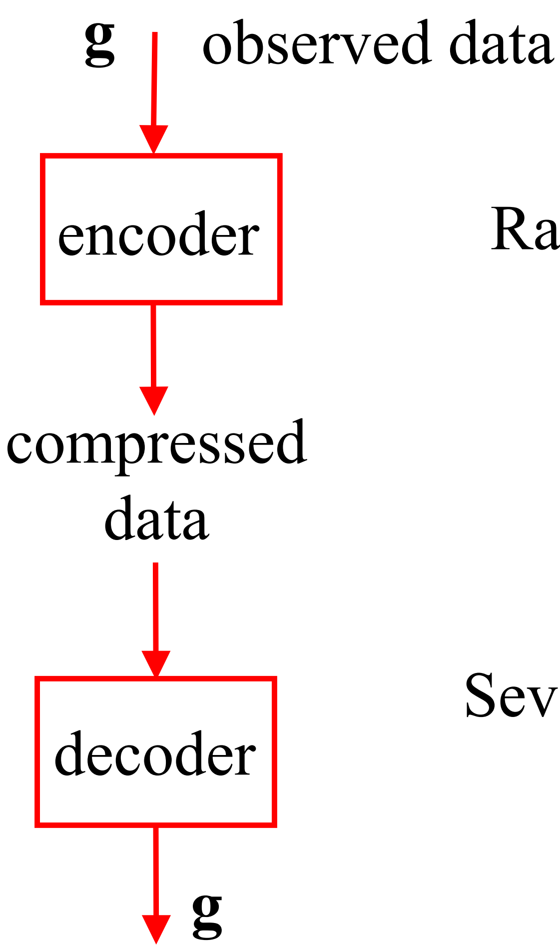
regardless of the values of σ_x^2 and σ_y^2

What about the dimension of $\boldsymbol{\theta}$?

(number of control points)

Proposed approach: MDL

Introduction to MDL



Rationale: short code $\Leftrightarrow$ good model

long code $\Leftrightarrow$ bad model

code length $\Leftrightarrow$ model adequacy

Several flavors: Rissanen 1978, 1987
Rissanen 1996,
Wallace and Freeman (MML), 1987

Scenario: A set of models (likelihoods) for the data

model m is characterized by (unknown) "parameters" $\mathbf{f}_{(m)}$

$$\{p(\mathbf{g}\,|\,\mathbf{f}_{(m)}, m)\,, m = m_1, m_2, \ldots, m_K\}$$

no prior information about $\mathbf{f}_{(m)}$

Goal: given data $\mathbf{g}$,

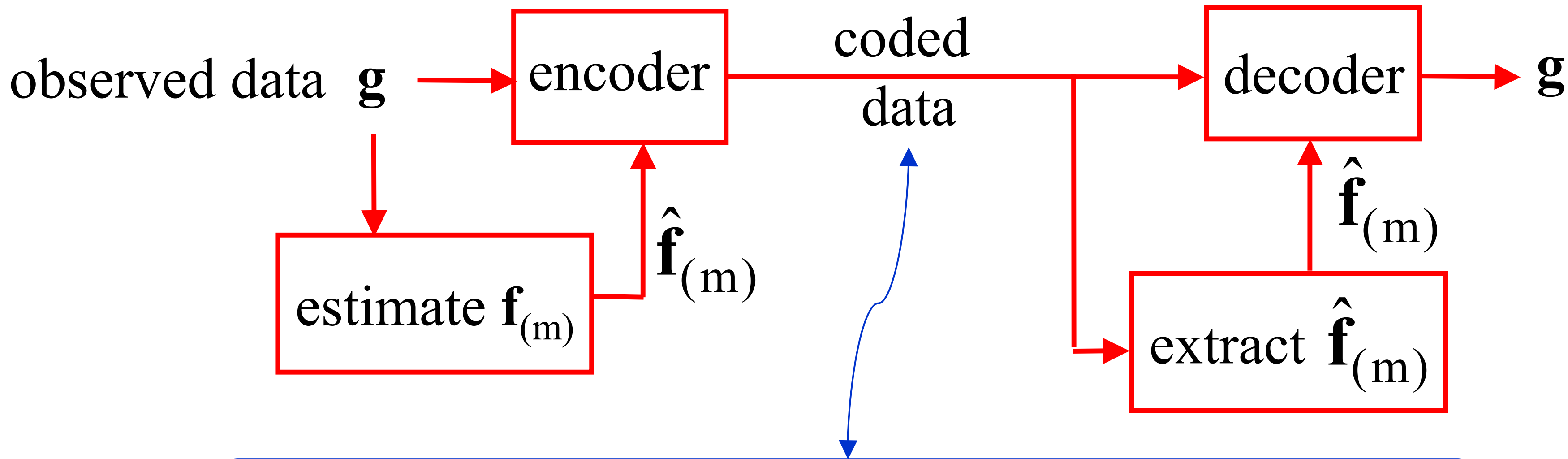
build the shortest possible code for $\mathbf{g}$

With $\mathbf{f}_{(m)}$ known, the shortest code-length for $\mathbf{g}$ is (Shannon's)

$$L(\mathbf{g}\,|\,\mathbf{f}_{(m)}) = -\log p(\mathbf{g}\,|\,\mathbf{f}_{(m)}, m)$$

However, $\mathbf{f}_{(m)}$ is, a priori, unknown; it has to be estimated.

Assumption: given $\mathbf{f}_{(m)}$, both encoder and decoder know how to build the same code



$$\text{coded data} = \text{code}(m) + \text{code}(\hat{\mathbf{f}}_{(m)}) + \text{code}(g \mid \hat{\mathbf{f}}_{(m)})$$

MDL principle:

choose m and $\hat{\mathbf{f}}_{(m)}$ so that length(coded data) is shortest

$$\text{coded data} = \text{code}(m) + \text{code}(\hat{\mathbf{f}}_{(m)}) + \text{code}(g \mid \hat{\mathbf{f}}_{(m)})$$

$$L(m, \mathbf{f}_{(m)}, \mathbf{g}) = \underbrace{L(m)}_{\text{Usually constant}} + L(\mathbf{f}_{(m)} \mid m) + L(\mathbf{g} \mid \mathbf{f}_{(m)})$$

MDL criterion

$$(\hat{m}, \hat{\mathbf{f}}_{(\hat{m})})_{MDL} = \arg\min_{m, \mathbf{f}_{(m)}} \{L(\mathbf{f}_{(m)}) + L(\mathbf{g} \mid \mathbf{f}_{(m)})\}$$

$$= \arg\min_{m, \mathbf{f}_{(m)}} \{L(\mathbf{f}_{(m)}) - \log p(\mathbf{g} \mid \mathbf{f}_{(m)})\}$$

$$(\hat{m}, \hat{\mathbf{f}}_{(\hat{m})})_{MDL} = \arg \min_{m, \mathbf{f}_{(m)}} \{ L(\mathbf{f}_{(m)}) - \log p(\mathbf{g} \mid \mathbf{f}_{(m)}) \}$$

$L(\mathbf{f}_{(m)})$? Finite $L(\mathbf{f}_{(m)}) \Rightarrow$ truncate to finite precision: $\tilde{\mathbf{f}}_{(m)}$

High precision

$$-\log f(\mathbf{g} \mid \tilde{\mathbf{f}}_{(m)}) \approx -\log f(\mathbf{g} \mid \hat{\mathbf{f}}^{ML}_{(m)}) \quad \text{but} \quad L(\tilde{\mathbf{f}}_{(m)}) \nearrow$$

Low precision

$$L(\tilde{\mathbf{f}}_{(m)}) \searrow \quad \text{but}$$

$$-\log f(g \mid \tilde{\mathbf{f}}_{(m)}) \quad \text{may be} \gg \quad -\log p(\mathbf{g} \mid \hat{\mathbf{f}}^{ML}_{(m)})$$

Optimal compromise (under regularity conditions, and asymptotic)

$$L(\text{each component of } \mathbf{f}_{(m)}) = \frac{1}{2}\log(n)$$

n, the sample size from which the parameter is estimated (growth rate of Fisher info.)

In our problem, $\mathbf{v} = [\mathbf{x} \quad \mathbf{y}] = \mathbf{B}\,\boldsymbol{\theta}$ is a "digital" curve

coordinates are quantized to pixel accuracy

What precision is required for $\boldsymbol{\theta}$, to guarantee pixel precision for $\mathbf{v}$?

Let $\Delta\boldsymbol{\theta}_x = \tilde{\boldsymbol{\theta}}_x - \boldsymbol{\theta}_x$ and $\Delta\boldsymbol{\theta}_y = \tilde{\boldsymbol{\theta}}_y - \boldsymbol{\theta}_y$

Finite precision versions

Goal: $\|\Delta\mathbf{x}\|_\infty \equiv \max_i |\Delta x_i| < 1$ and $\|\Delta\mathbf{y}\|_\infty \equiv \max_i |\Delta y_i| < 1$

By linearity, $\Delta\mathbf{x} = \mathbf{B}\,\Delta\boldsymbol{\theta}_x$ and $\Delta\mathbf{y} = \mathbf{B}\,\Delta\boldsymbol{\theta}_y$

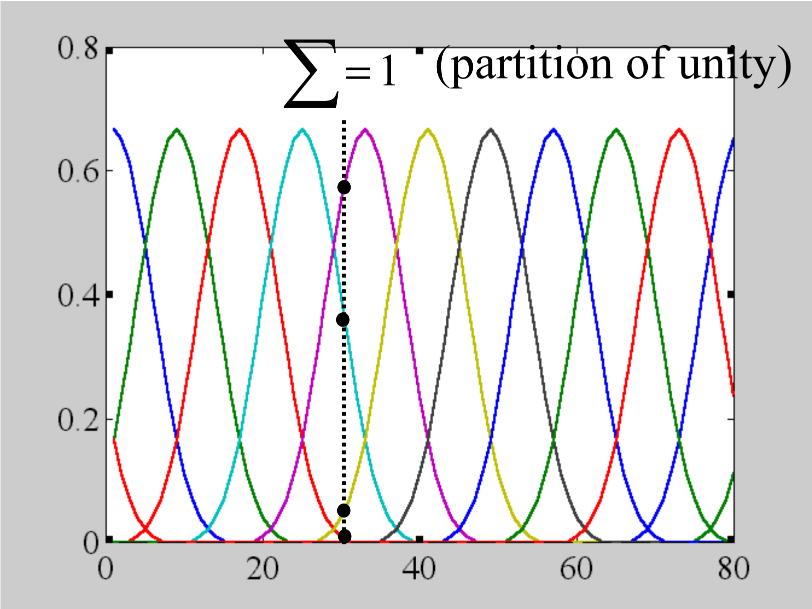
Key fact:

$$\|\mathbf{B}\|_\infty \equiv \max_i \sum_j |B_{ij}| = \max_i \sum_j B_{ij} = 1$$

Induced matrix norm

$$\|\mathbf{B}\,\mathbf{u}\|_\infty \leq \|\mathbf{B}\|_\infty \times \|\mathbf{u}\|_\infty$$

Columns of **B**



Recall our goal: $\|\Delta\mathbf{x}\|_\infty < 1, \quad \|\Delta\mathbf{y}\|_\infty < 1$

and $\quad \Delta\mathbf{x} = \mathbf{B}\,\Delta\boldsymbol{\theta}_x, \quad \Delta\mathbf{y} = \mathbf{B}\,\Delta\boldsymbol{\theta}_y$

then,

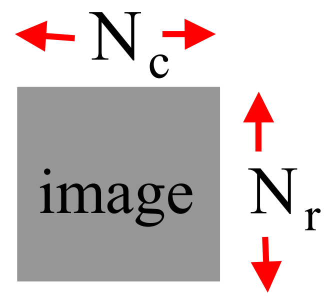
$\|\Delta\boldsymbol{\theta}_y\|_\infty < 1 \Rightarrow \|\Delta\mathbf{y}\|_\infty < 1$	$\|\Delta\boldsymbol{\theta}_x\|_\infty < 1 \Rightarrow \|\Delta\mathbf{x}\|_\infty < 1$

i.e., pixel precision is enough for the control points.

Natural code length for k control points

$$L(\boldsymbol{\theta}_{(k)}) = k\left(\log(N_r) + \log(N_c)\right) = L(k)$$

denotes k control points



MDL criterion:

$$\min_{k,\boldsymbol{\theta}_{(k)},\sigma_x^2,\sigma_y^2} \left\{ L(k) - \log p(\mathbf{x} \mid \theta_x, \sigma_x^2) - \log p(\mathbf{y} \mid \theta_y, \sigma_y^2) \right\}$$

$$\min_{\boldsymbol{\theta}_{(k)},\sigma_x^2,\sigma_y^2}$$

some simple manipulation leads to

$$\hat{k} = \arg\min_k \left\{ L(k) - n \log \sqrt{\hat{\sigma}_x^2(k) \;\; \hat{\sigma}_x^2(k)} \right\}$$

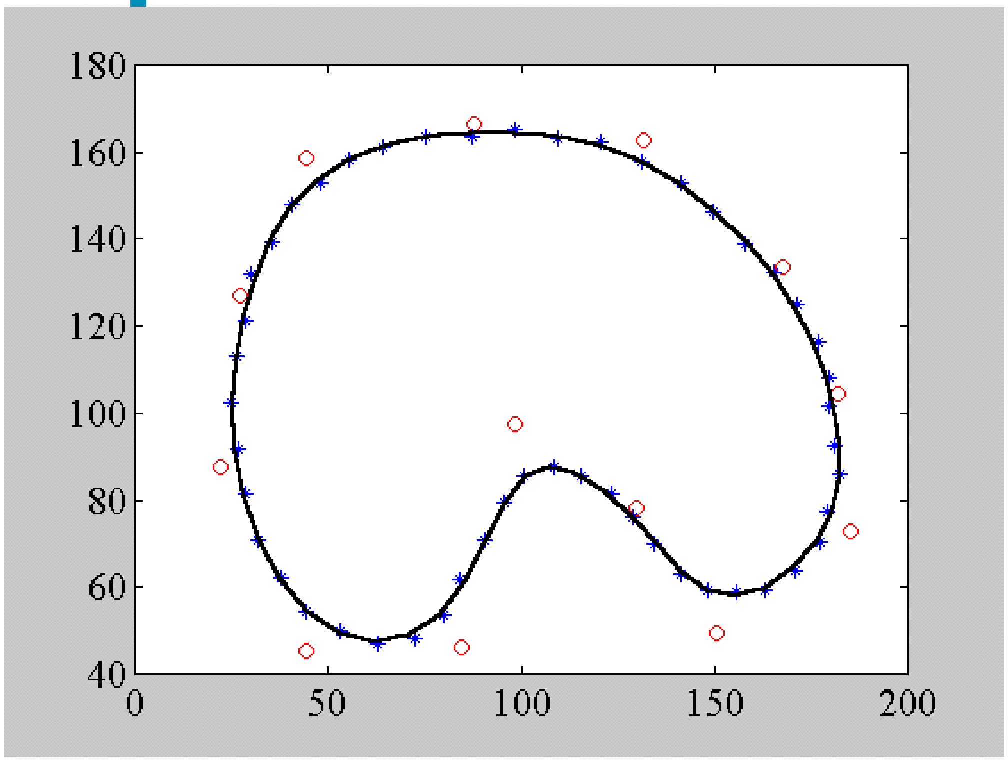
$$\hat{\sigma}_x^2(k) = \frac{1}{n}\left\| \mathbf{x} - \mathbf{B}(k)^{\perp}\mathbf{x} \right\|^2 \qquad \hat{\sigma}_y^2(k) = \frac{1}{n}\left\| \mathbf{y} - \mathbf{B}(k)^{\perp}\mathbf{y} \right\|^2$$

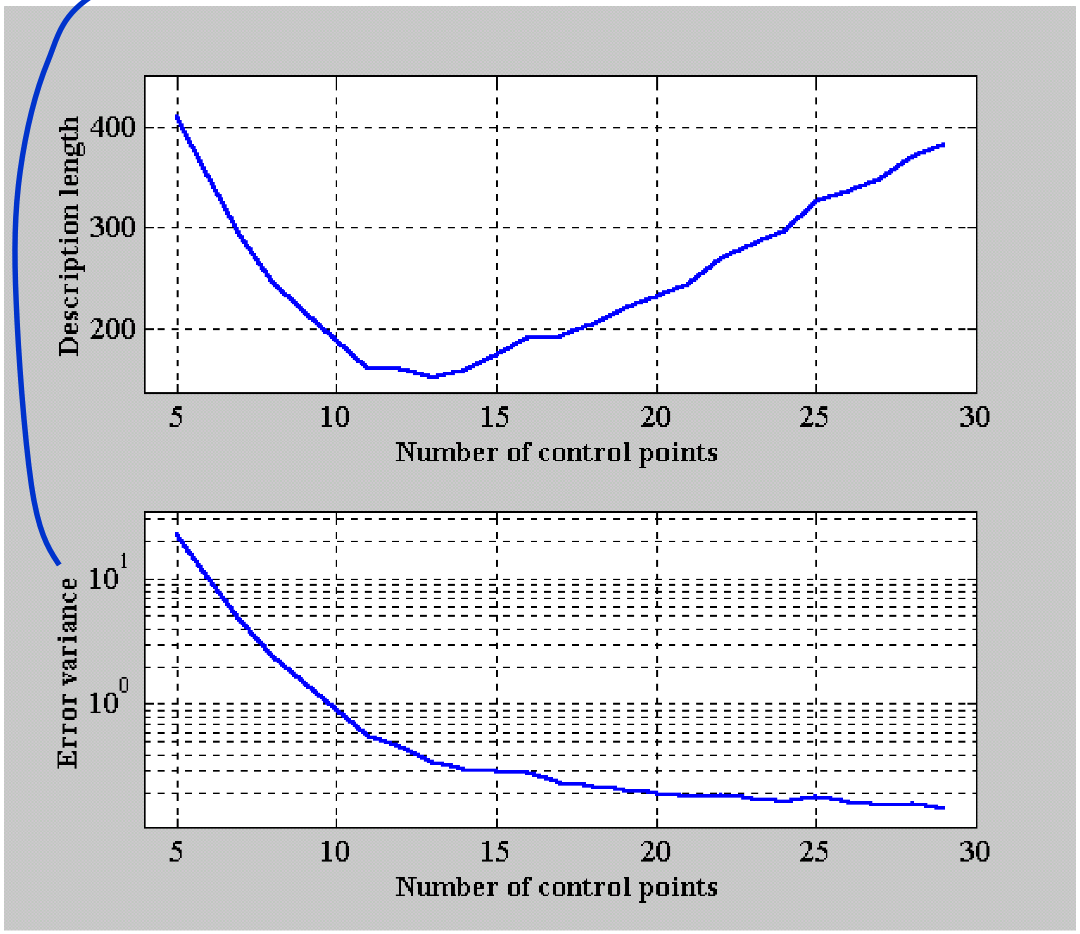
residual error variances

Model selection criterion:

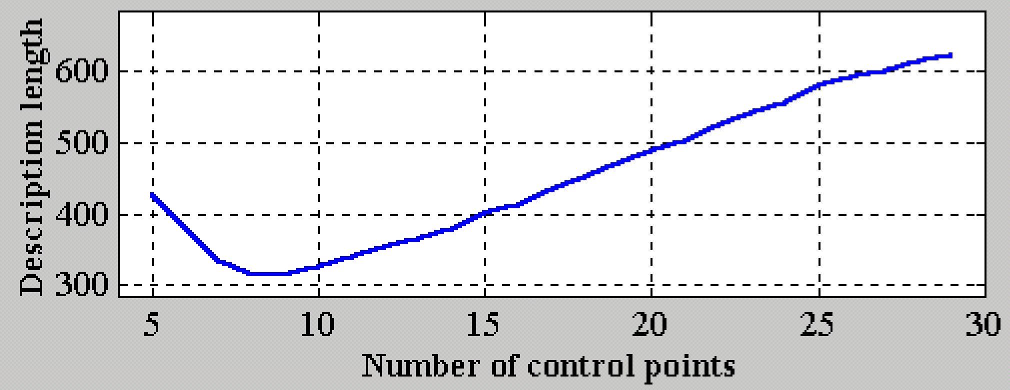
$$\hat{k} = \arg\min_{k}\left\{ k \log(N_r\ N_c) - n \log \sqrt{\hat{\sigma}_x^2(k)\ \ \hat{\sigma}_x^2(k)} \right\}$$

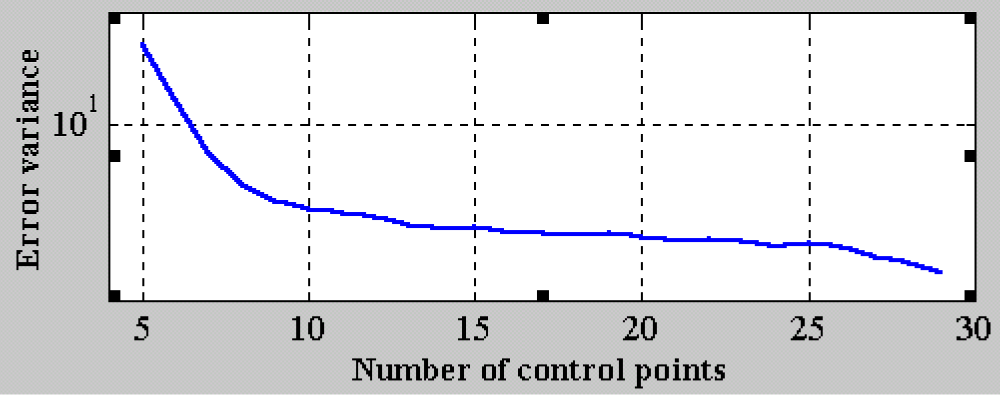
Example: hand drawn points

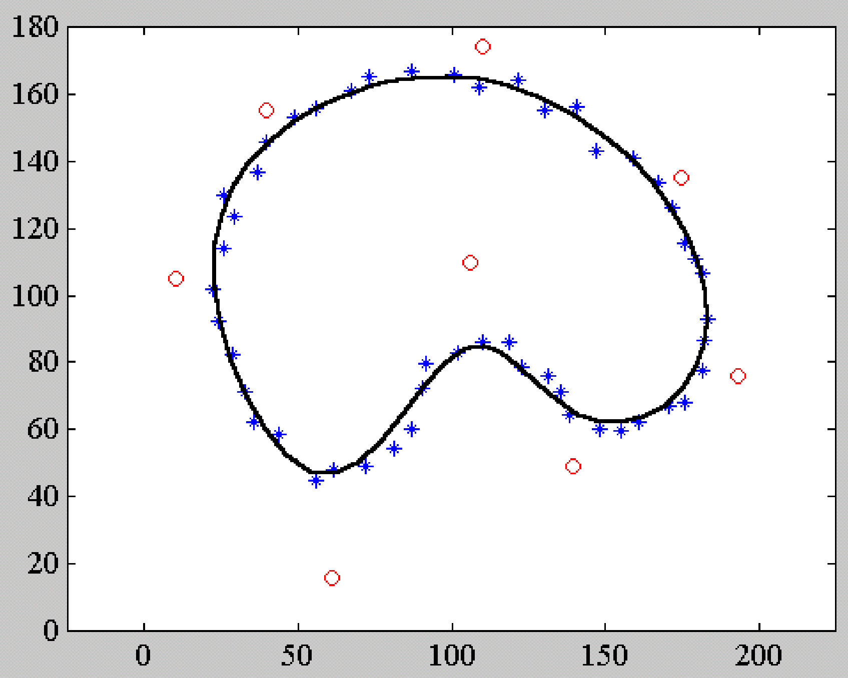




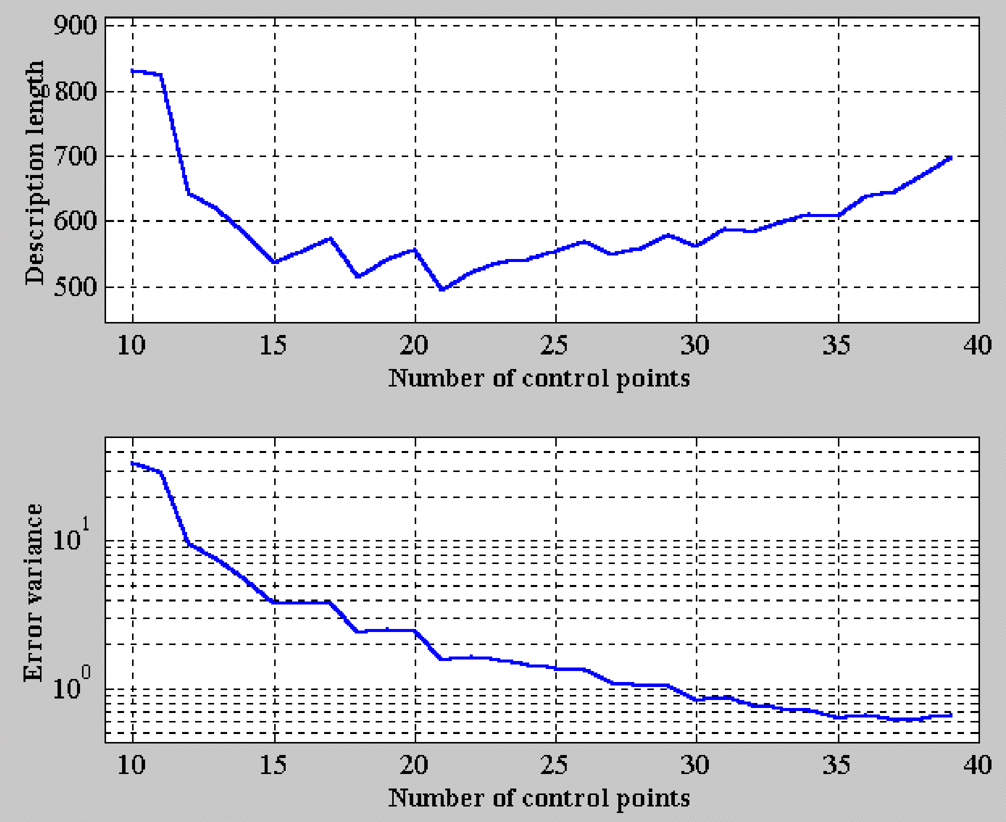
Example: hand drawn points, with added noise.

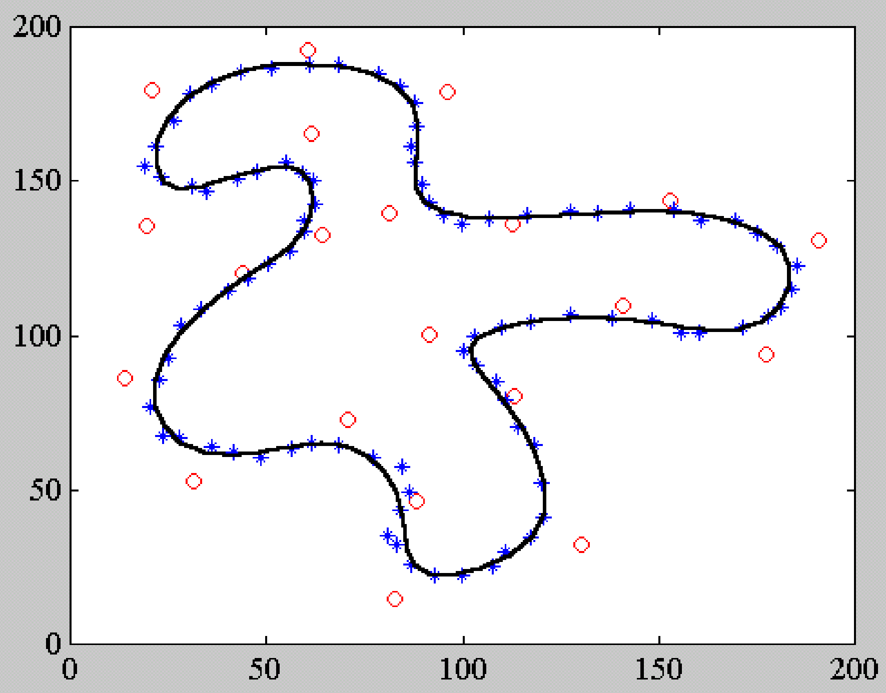




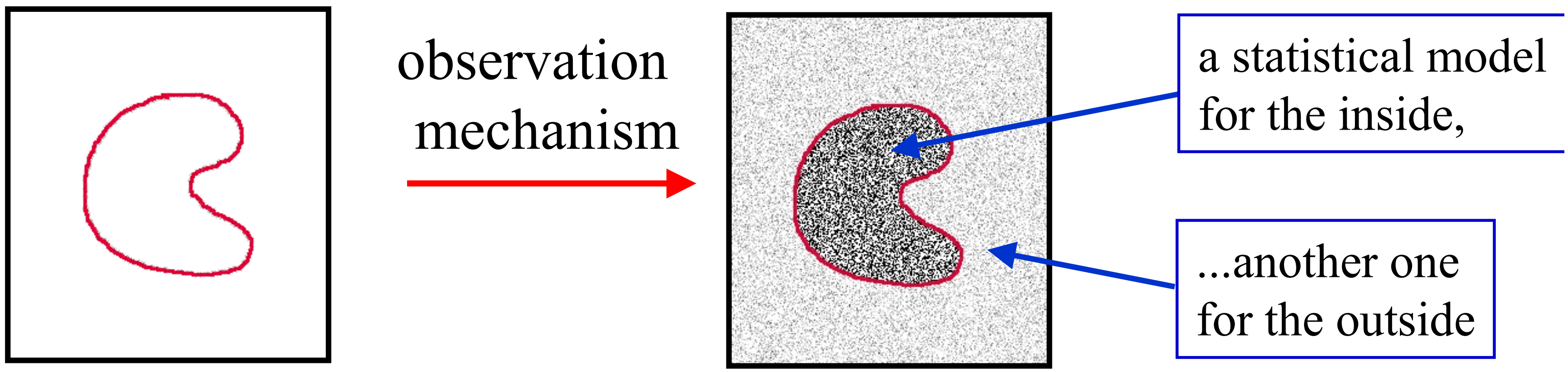


Example: a more complex shape





To use the MDL approach, we need the likelihood function:



$\boldsymbol{\theta}_{(k)} \rightarrow$ contour description (a 2-D spline curve with k control points)

$\mathbf{I} \rightarrow$ observed image

$$p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \phi_{in}, \phi_{out}) = \prod_{i \in inside(\mathbf{v}(\boldsymbol{\theta}_{(k)}))} p\,(I_i \mid \phi_{in}) \prod_{i \in outside(\mathbf{v}(\boldsymbol{\theta}_{(k)}))} p\,(I_i \mid \phi_{out})$$

where $\mathbf{v}(\boldsymbol{\theta}_{(k)}) = \mathbf{B}\,\boldsymbol{\theta}_{(k)}$

ϕ_{in}, ϕ_{out} are also considered unknown.

MDL criterion:

$$\min_{k,\boldsymbol{\theta}_{(k)},\phi_{in},\phi_{out}} \left\{ L(k) - \log p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \phi_{in}, \phi_{out}) \right\}$$

Now, it is not possible to solve analytically w.r.t. $\boldsymbol{\theta}_{(k)}, \phi_{in}, \phi_{out}$

Proposed approach: an iterative method.

$$\min_{k,\boldsymbol{\theta}_{(k)},\phi_{in},\phi_{out}} \left\{ L(k) - \log p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \phi_{in}, \phi_{out}) \right\}$$

can be rewritten as

$$\min_{k} \left\{ L(k) - \max_{\boldsymbol{\theta}_{(k)},\phi_{in},\phi_{out}} \left\{ \log p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \phi_{in}, \phi_{out}) \right\} \right\}$$

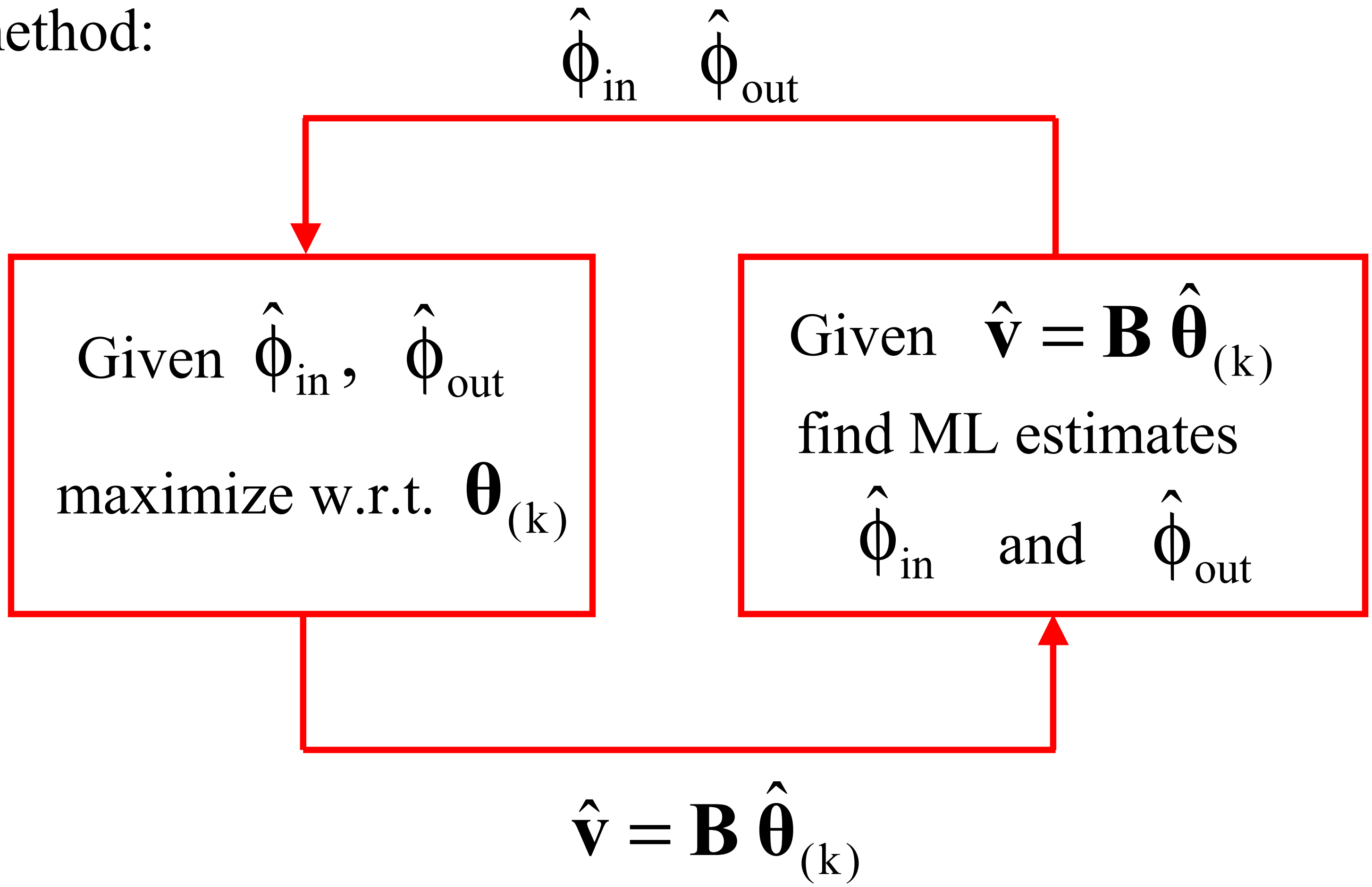
Solved by iterative method

$$\min_{k} \left\{ L(k) - G(\mathbf{I}, k) \right\}$$

Outer minimization: solved by exhaustive search

$$\max_{\boldsymbol{\theta}_{(k)}, \phi_{in}, \phi_{out}} \left\{ \log p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \phi_{in}, \phi_{out}) \right\}$$

Iterative method:



Given $\hat{\phi}_{in}, \quad \hat{\phi}_{out}$

$$\max_{\boldsymbol{\theta}_{(k)}} \left\{ \log p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \hat{\phi}_{in}, \hat{\phi}_{out}) \right\}$$

is equivalent to

$$\max_{\mathbf{v}} \left\{ \log p(\mathbf{I} \mid \mathbf{v}, \hat{\phi}_{in}, \hat{\phi}_{out}) \right\}$$

Subject to: $\quad \mathbf{v} \in \mathcal{R}(\mathbf{B}_{(k)})$

The range space of $\mathbf{B}_{(k)}$,
i.e., the span of its columns

$$\max_{\mathbf{v}} \left\{ \log p(\mathbf{I} \mid \mathbf{v}, \hat{\phi}_{in}, \hat{\phi}_{out}) \right\} \qquad \text{Subject to: } \mathbf{v} \in \mathcal{R}(\mathbf{B}_{(k)})$$

Grandient projection algorithm. Input: $\hat{\phi}_{in}, \hat{\phi}_{out}, \hat{\mathbf{v}}^{(0)} \in \mathcal{R}(\mathbf{B}_{(k)})$

Step 0: build $\mathbf{B}_{(k)}$ and compute $\mathbf{B}_{(k)}^{\perp} = \mathbf{B}_{(k)} \left(\mathbf{B}_{(k)}^{T} \mathbf{B}_{(k)} \right)^{-1} \mathbf{B}_{(k)}^{T}$

Step 1: compute the gradient $\delta\mathbf{v} = \nabla \log\, p(I \mid \mathbf{v})\big|_{\mathbf{v}=\hat{\mathbf{v}}^{(t)}}$

Step 2: project the gradient onto $\mathcal{R}(\mathbf{B}_{(k)})$: $(\delta\mathbf{v})^{\perp} = \mathbf{B}_{(k)}^{\perp}\, \delta\mathbf{v}$

Step 3: take a small step in the direction of the projected gradient:

$$\hat{\mathbf{v}}^{(t+1)} = \hat{\mathbf{v}}^{(t)} + \varepsilon (\delta\mathbf{v})^{\perp} = B_{(k)}^{\perp} \left(\hat{\mathbf{v}}^{(t)} + \varepsilon\ \delta\mathbf{v} \right)$$

No convergence: increment t, back to Step 1

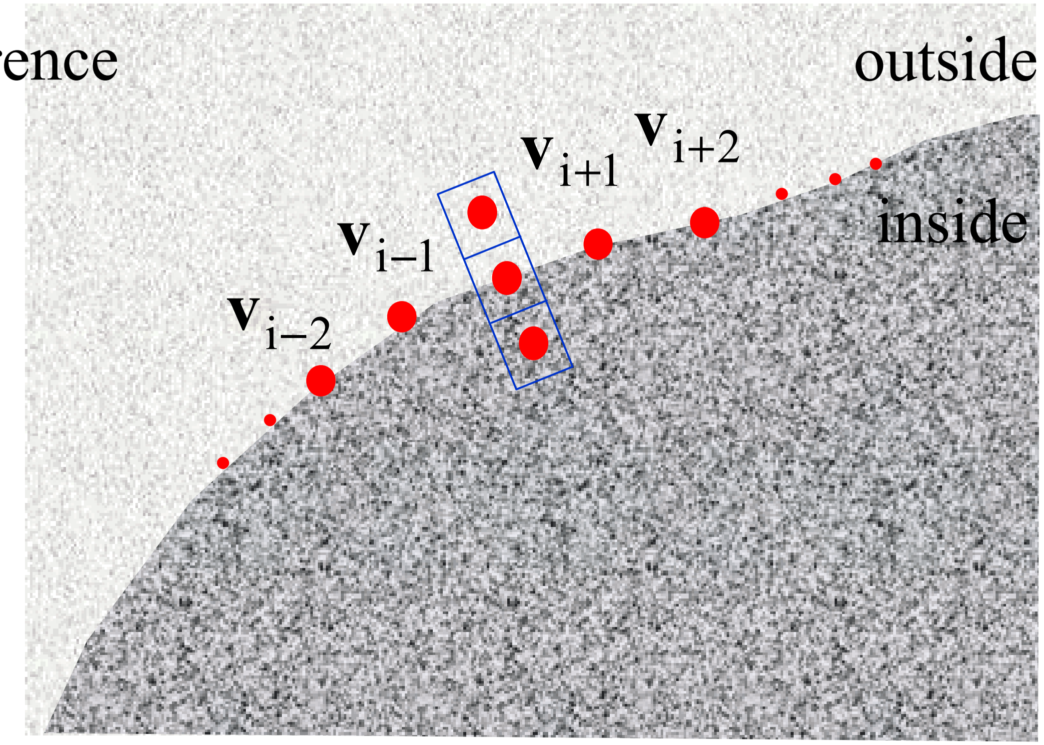
Computing the gradient

$$\delta \mathbf{v} = \nabla \log\ p(I \mid \mathbf{v})\Big|_{\mathbf{v}=\hat{\mathbf{v}}^{(t)}}$$

Gradient is perpendicular to the contour

Coordinate i of the gradient:
approximated with a finite difference

Only requires values on a small
perpendicular window



$$\min_{k}\left\{ L(k) - \max_{\boldsymbol{\theta}_{(k)},\phi_{in},\phi_{out}} \left\{ \log p(\mathbf{I} \mid \mathbf{v}(\boldsymbol{\theta}_{(k)}), \phi_{in}, \phi_{out}) \right\} \right\}$$

Solved by iterative method

$$\min_{k}\{ L(k) - G(\mathbf{I}, k) \}$$

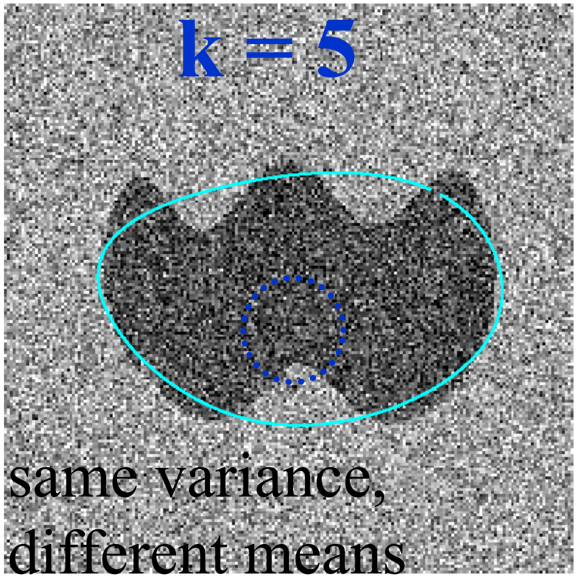
Outer minimization: solved by exhaustive search

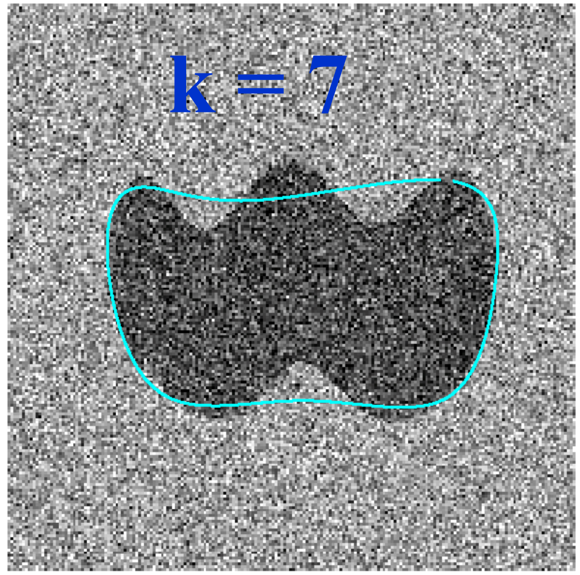
Sweep range of values $k \in \{k_{min}, k_{min}+1, \cdots, k_{max}\}$

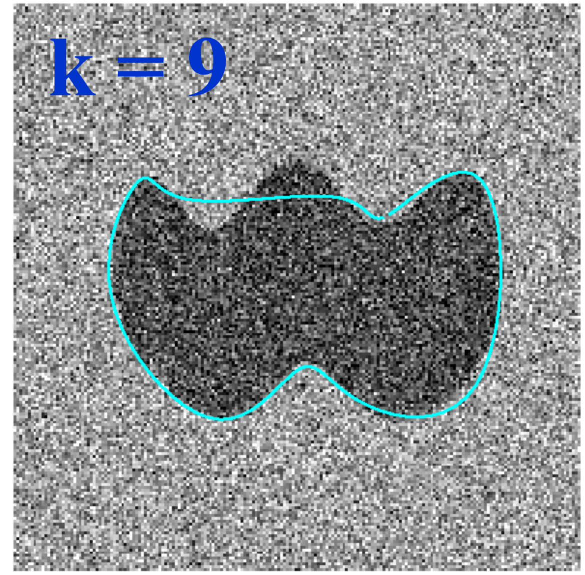
Start with $k = k_{min}$

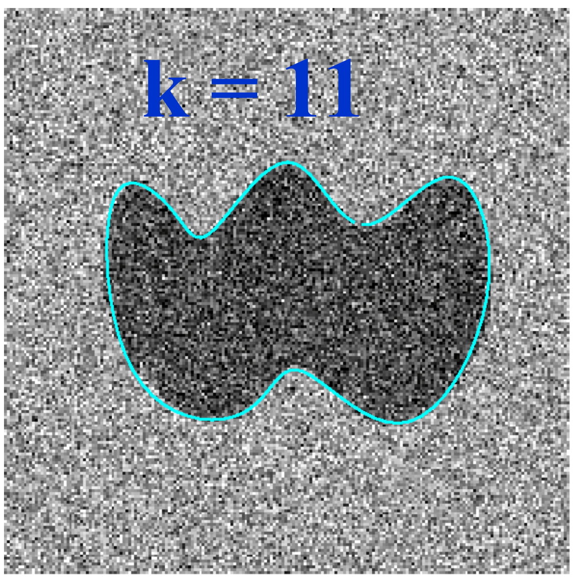
Use contour obtained at each k, to initialize the next iterative algorithm

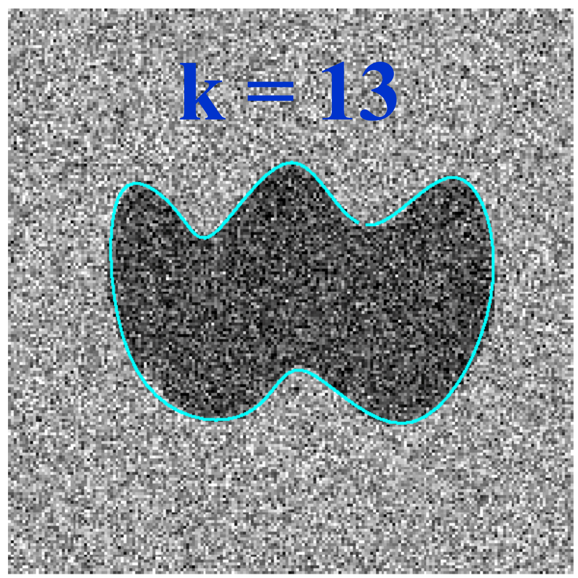
Contour estimation examples: synthetic data

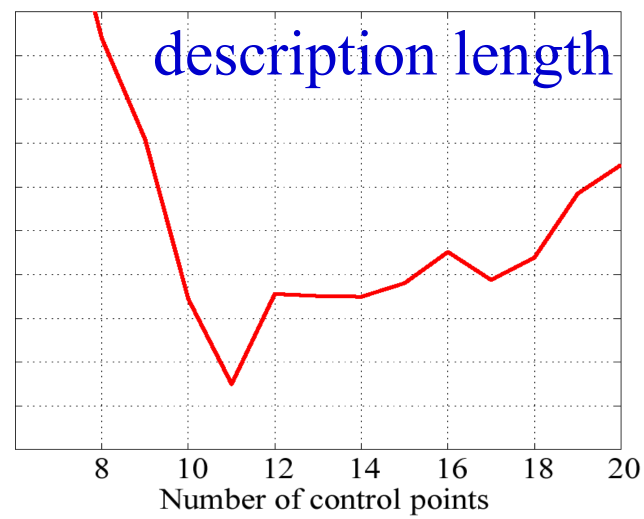






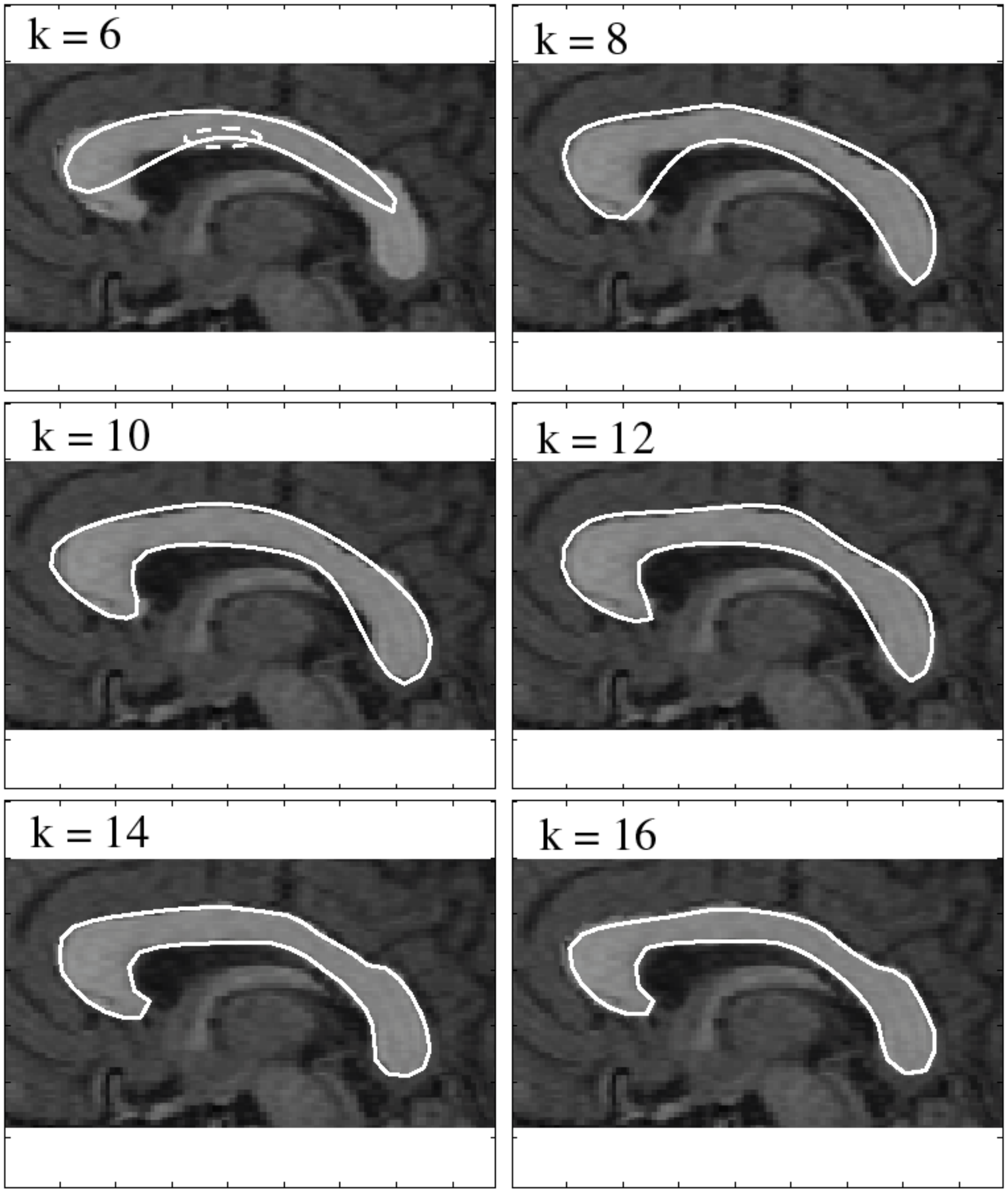


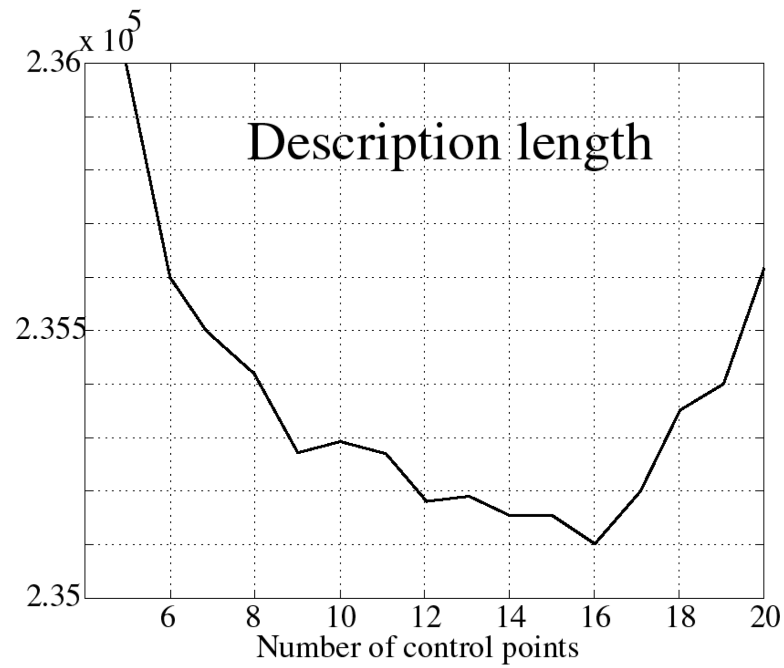




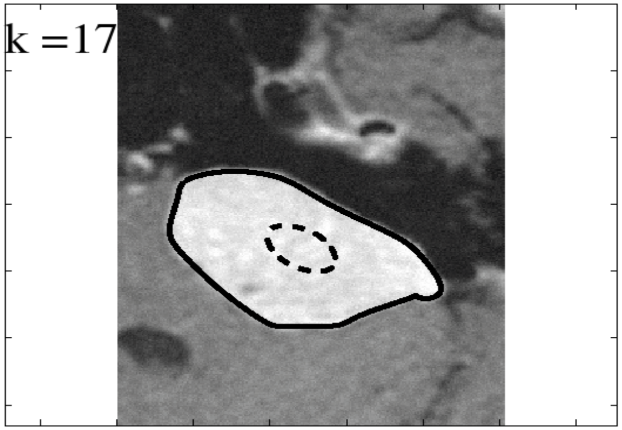
Dashed line = initial contour

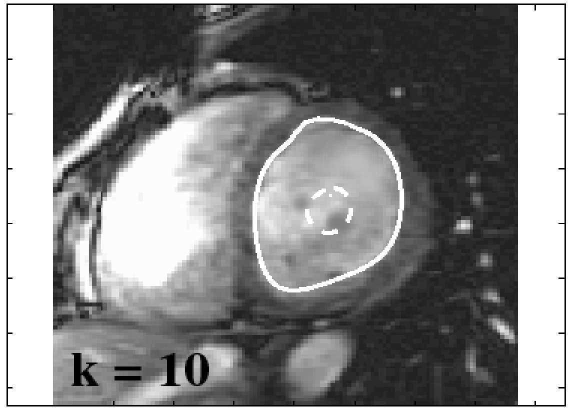
Contour estimation example: real medical image

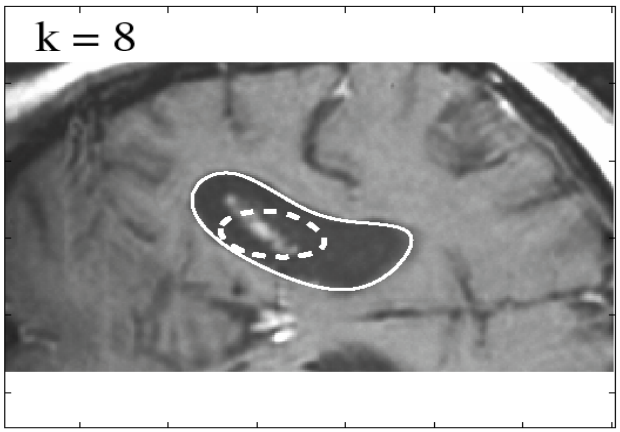


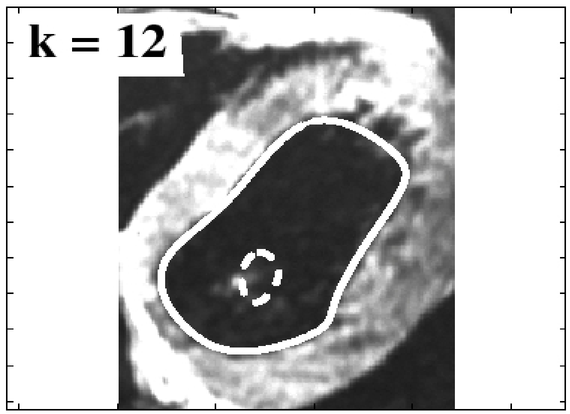


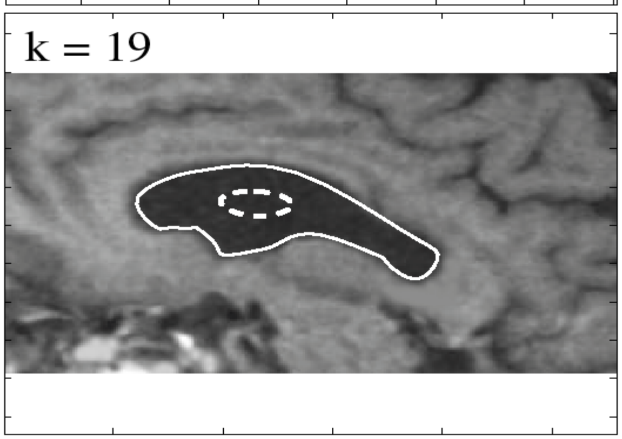
More examples on real medical images

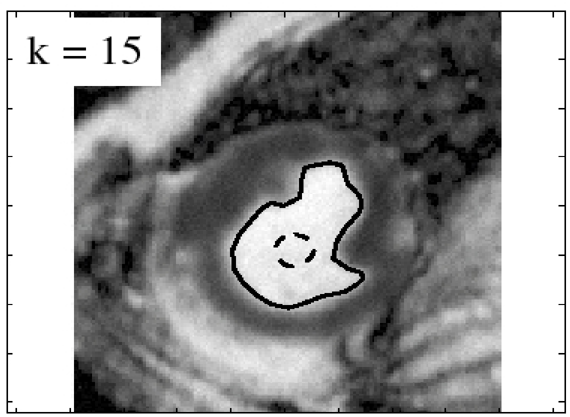












See:

M. Figueiredo, J. Leitão, and A.K.Jain, "Unsupervised contour representation and estimation using B-splines and a minimum description length criterion" in *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1075-1087, 2000.