Two Probabilistic Approaches to Deformable Contours

Mário A. T. Figueiredo
“Instituto Superior Técnico”
Lisboa
PORTUGAL

• **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 \((\mathbf{v})\) under the “image forces”, to “find the contour”.

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

For example, to attract \(\mathbf{v}\) towards high-gradient regions (boundaries)

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

A configuration with low \(E_{\text{ext}}(\mathbf{v}, I)\)
An elastically deformable line, $v$, on the image plane, ...

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

shape at rest (low $E(v)$) \hspace{1cm} deformed shape (higher $E(v)$)
Snakes

The “snake” approach: combine the two energies

Minimizer of \( E_{\text{int}}(v) \)

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

A good compromise:

\[
\hat{v} = \arg \min_v \left\{ E_{\text{ext}}(v, I) + \alpha E_{\text{int}}(v) \right\}
\]
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

\( f \), a contour, e.g., represented by a sequence of points
Bayesian decision theory

unknown \( f \) → Observation model (likelihood function) → observed \( g \)

knowledge

estimate \( \hat{f} \) → Bayesian decision rule (e.g., MAP, MPM, PM...)

Loss function \( L(\hat{f}, \hat{f}) \) (explicit adoption of decision-theoretic approach)

The Bayesian approach is explicitly model-based
- Observation model / likelihood function:

\[ p(g \mid f, \phi) \]

\( f \) is the unknown
\( g \) is the observed data
\( \phi \) are parameters

- Prior knowledge:

\[ p(f \mid \psi) \]

\( f \) is the unknown
\( \psi \) are parameters

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

Bayes law:

\[
p(f \mid g, \phi, \psi) = \frac{p(g \mid f, \phi) p(f \mid \psi)}{p(g \mid \phi, \psi)}
\]
Given \( p(f | g, \phi, \psi) \) and a loss function \( L(f, \hat{f}) \)

Optimal Bayes rule: minimizer of the \textit{a posteriori} expected loss:

\[
\hat{f} = \arg \min_f \int L(f, \hat{f}) \, p(f | g, \phi, \psi) \, df
\]

Particular case: the \textit{maximum a posteriori} rule (0/1 loss)

\[
L(f, \hat{f}) = \begin{cases} 
1 & \text{if } f \neq \hat{f} \\
0 & \text{if } f = \hat{f}
\end{cases}
\]

\[
\hat{f}_{\text{MAP}} = \arg \max_f p(f | g, \phi, \psi) = \arg \max_f \left\{ p(g | f, \phi) \, p(f | \psi) \right\}
\]

\[\hat{f}_{\text{MAP}} = \arg \max_f \left\{ \log p(g | f, \phi) + \log p(f | \psi) \right\}\]

Particular case of MAP: the \textit{maximum likelihood} (ML) criterion

\[p(f | \psi) \propto \text{const.} \quad \Rightarrow \quad \hat{f}_{\text{ML}} = \arg \max_f \log p(g | f, \phi)\]
MAP rule: \[
\hat{f}_{\text{MAP}} = \arg \min_{f} \{- \log p(g \mid f, \phi) - \log p(f \mid \psi)\}
\]

Snake “rule”: \[
\hat{v} = \arg \min_{v} \{E_{\text{ext}}(v, I) + \alpha E_{\text{int}}(v)\}
\]

The similarity suggests: \[
p(v) = \frac{1}{Z_{\text{int}}} \exp\{-\alpha E_{\text{int}}(v)\}
\]

\[
p(I \mid v) = \frac{1}{Z_{\text{ext}}(v)} \exp\{-E_{\text{ext}}(v, I)\}
\]

Then, \[
\hat{v}_{\text{MAP}} = \arg \min_{v} \{E_{\text{ext}}(v, I) + \alpha E_{\text{int}}(v)\}
\]

if and only if \[
Z_{\text{ext}}(v) = Z_{\text{ext}} \quad \text{...often not true.}
\]
• **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. $\alpha$) 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…. 
A Bayesian region-based snake (Figueiredo and Leitão, IEEE-TMI, 1992)

Under inside/outside independence assumption:

$$p(I \mid \phi_{\text{in}}, \phi_{\text{out}}) = p(I_{\text{inside}}(v) \mid \phi_{\text{in}}) \ p(I_{\text{outside}}(v) \mid \phi_{\text{out}})$$

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;
\[ \mathbf{v} = (v_1, v_2, \ldots, v_n) \quad \text{where} \quad v_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \]

- Prior knowledge: \( \mathbf{v} \) is “smooth”

1-D Markov random field

\[
p(\mathbf{v} | \psi) = \frac{1}{Z} \exp \left\{ -\frac{1}{\psi} \sum_i (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \right\}
\]
Likelihood function:

\[ p(I | v, \phi_{in}, \phi_{out}) = p(I_{\text{inside}(v)} | \phi_{in}) \ p(I_{\text{outside}(v)} | \phi_{out}) \]

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

\[ \phi_{in} = (\mu_{in}, \sigma_{in}^2) \quad \phi_{out} = (\mu_{out}, \sigma_{out}^2) \]

\[ p(I | v, \phi_{in}, \phi_{out}) = \prod_{i \in \text{inside}(v)} N(I_i | \mu_{in}, \sigma_{in}^2) \ \prod_{i \in \text{outside}(v)} N(I_i | \mu_{out}, \sigma_{out}^2) \]

Note: \( Z_{\text{ext}}(v) \) is not constant: 

\[ Z_{\text{ext}}(v) \propto (\sigma_{in}^2)^{-N_{\text{in}}(v)} (\sigma_{out}^2)^{-N_{\text{out}}(v)} \]

number of pixels inside/outside \( v \)
Likelihood function:
\[
p(I \mid v, \phi_{\text{in}}, \phi_{\text{out}}) = \prod_{i \in \text{inside}(v)} p(I_i \mid \phi_{\text{in}}) \prod_{i \in \text{outside}(v)} N(I_i \mid \phi_{\text{out}})
\]

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

\[
\hat{v}_{\text{MAP}} = \arg \min_v \{-\log p(v \mid \psi) - \log p(I \mid v, \phi_{\text{in}}, \phi_{\text{out}})\}
\]

Questions: - How to find the maximum?
- What about the parameters? \((\psi, \phi_{\text{in}}, \phi_{\text{out}})\)
Advantage of a probabilistic approach: the parameters have meanings and can be estimated

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

...expressing preference for "smoother" contours

• A flat prior for the likelihood parameters

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

$$(\hat{v}, \hat{\psi}, \hat{\phi}_{in}, \hat{\phi}_{out}) = \arg\min_{v, \psi, \phi_{in}, \phi_{out}} \{ -\log p(v \mid \psi) - \log p(\psi) - \log p(I \mid v, \phi_{in}, \phi_{out}) \}$$
Adaptive ICM, or component-wise iterative optimization

Iterated conditional modes (Besag, 1986)

Step 0

Initialization: get initial contour \( \hat{v}^{(0)} \)
set \( t = 0 \)

Step 1

Given \( \hat{v}^{(t)} \), update the parameter estimates:

\[
\hat{\psi}^{(t+1)} = \arg \min_{\psi} \left\{ -\log p(\psi) - \log p(\hat{v}^{(t)} | \psi) \right\}
\]

(MAP estimate)

\[
(\hat{\phi}_{\text{in}}, \hat{\phi}_{\text{out}})^{(t+1)} = \arg \min_{\phi_{\text{in}}, \phi_{\text{out}}} \left\{ -\log p(I | \hat{v}^{(t)}, \phi_{\text{in}}, \phi_{\text{out}}) \right\}
\]

(ML estimates)

Step 2

Update contour by performing 1 ICM step: \( \hat{v}^{(t+1)} \)
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

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

\[- \log p(\mathbf{v} | \hat{\psi}^{(t+1)}) - \log p(\mathbf{I} | \mathbf{v}, \hat{\phi}^{(t+1)}) \equiv E(\mathbf{v}) \]

is non-convex in \( \mathbf{v} \).

ICM

for each \( i=1,2,\ldots,n \)

\[ \hat{\mathbf{v}}_i^{(t+1)} = \arg \min_{\mathbf{v}_i} E(\mathbf{v} | \{\mathbf{v}_j \neq i\} \text{ fixed}) \]

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

Alternatives: dynamic programming, simulated annealing,…
For more details, see:
PART II – Parametrically Deformable Contours
• Standard snakes: “explicit” contour description \( v = (v_1, v_2, \ldots, v_n) \)
  (nonparametric)

• Parametrically deformable contours:
  - parametric, usually “short” description \( v = M(\theta) \)
  - 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: $v = M(\theta)$

Usually:

Parameterization order $\leftrightarrow$ smoothness/simplicity of $v$

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
Brief review of B-spline curves

Matrix of (discretized) periodic (cubic) B-spline basis

Control points

\[
\begin{bmatrix}
\mathbf{v}_1 \\
\vdots \\
\mathbf{v}_n
\end{bmatrix} = \mathbf{B} \begin{bmatrix}
\theta_1 \\
\vdots \\
\theta_k
\end{bmatrix} = \mathbf{B} \begin{bmatrix}
\theta_{x,1} & \theta_{y,1} \\
\vdots & \vdots \\
\theta_{x,k} & \theta_{y,k}
\end{bmatrix}
\]

\[
\mathbf{v} = [x \ y] = \mathbf{B} [\theta_x \ \theta_y] \Leftrightarrow \begin{cases}
x = \mathbf{B} \theta_x \\
y = \mathbf{B} \theta_y
\end{cases}
\]

\[
\sum_{i=1}^{n} = 1 \quad \text{(partition of unity)}
\]

Columns of B (n=80, k=10)
Number of control points: curve complexity

12 control points

43 control points
Given a set of points
\[
\begin{bmatrix}
v_1 \\ \\
\vdots \\ \\
v_n
\end{bmatrix} = \begin{bmatrix}
x_1 & y_1 \\ \\
\vdots & \vdots \\ \\
x_n & y_n
\end{bmatrix} = [x \ y]
\]

…and a B-spline matrix \( B \), find the “best” control points.

in mean square sense

\[
\hat{\theta}_x = \arg \min_{\theta_x} \| x - B\theta_x \|^2 \quad \hat{\theta}_y = \arg \min_{\theta_x} \| y - B\theta_y \|^2
\]

Solution: 
\[
\hat{\theta}_x = \left( B^T B \right)^{-1} B^T x = B^# x \quad \hat{\theta}_y = \left( B^T B \right)^{-1} B^T y = B^# y
\]

\( B^# \), pseudo-inverse of \( B \)
(Noisy) points \( \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \)

Control points \( \hat{\mathbf{\theta}} = \mathbf{B}^\# \mathbf{v} \)

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

Orthogonal projection matrix
\[
\mathbf{B} \left( \mathbf{B}^T \mathbf{B} \right)^{-1} \mathbf{B}^T \equiv \mathbf{B}^\perp
\]

Projects \( \mathbf{v} \) onto the span of the columns of \( \mathbf{B} \)

Key question: how many control points?
Consider an i.i.d. Gaussian noise model:

\[ p(x, y \mid \theta_x, \sigma_x^2, \theta_y, \sigma_y^2) = p(y \mid \theta_y, \sigma_y^2) \, p(x \mid \theta_x, \sigma_x^2) \]

\[ p(x \mid \theta_x) \propto \exp\left\{-\frac{\|x - B \theta_x\|^2}{2 \sigma_x^2}\right\} \quad p(y \mid \theta_y) \propto \exp\left\{-\frac{\|y - B \theta_y\|^2}{2 \sigma_y^2}\right\} \]

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

\[ \hat{\theta}_x = \arg \min_{\theta_x} \|x - B \theta_x\|^2 \quad \hat{\theta}_y = \arg \min_{\theta_y} \|y - B \theta_y\|^2 \]

regardless of the values of \( \sigma_x^2 \) and \( \sigma_y^2 \)

What about the dimension of \( \theta \)?

(number of control points)

Proposed approach: MDL
Introduction to MDL

Rationale: short code ⇔ good model
long code ⇔ bad model
code length ⇔ 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” \( f_{(m)} \)
\[
\{ p(g | f_{(m)}, m), m = m_1, m_2, ..., m_K \}
\]
no prior information about \( f_{(m)} \)

Goal: given data \( g \),
   build the shortest possible code for \( g \)

With \( f_{(m)} \) known, the shortest code-length for \( g \) is (Shannon’s)
\[
L(g | f_{(m)}) = -\log p(g | f_{(m)}, m)
\]

However, \( f_{(m)} \) is, a priori, unknown; it has to be estimated.
Formalizing the MDL criterion: two-part codes

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

Observed data $g$ → **encoder** → coded data $\hat{f}(m)$ → **decoder** → $g$

- **estimate** $f(m)$
- **extract** $\hat{f}(m)$

Coded data = code(m) + code(\(\hat{f}(m)\)) + code(g | \(\hat{f}(m)\))

MDL principle:
choose $m$ and $\hat{f}(m)$ so that length(coded data) is shortest
Formalizing the MDL criterion

coded data = code(m) + code(\hat{f}(m)) + code(g | \hat{f}(m))

\[ L(m, f(m), g) = L(m) + L(f(m) | m) + L(g | f(m)) \]

Usually constant

MDL criterion

\[ (\hat{m}, \hat{f}(\hat{m}))_{MDL} = \arg \min_{m, f(m)} \{ L(f(m)) + L(g | f(m)) \} \]

\[ = \arg \min_{m, f(m)} \{ L(f(m)) - \log p(g | f(m)) \} \]
\[(\hat{m}, \hat{f}(\hat{m}))_{\text{MDL}} = \arg \min_{m, f(m)} \{L(f(m)) - \log p(g \mid f(m))\}\]

\[L(f(m)) \quad \text{Finite} \quad L(f(m)) \Rightarrow \text{truncate to finite precision: } \tilde{f}(m)\]

High precision

\[-\log f(g \mid \tilde{f}(m)) \approx -\log f(g \mid \hat{f}(m)^{\text{ML}}) \quad \text{but} \quad L(\tilde{f}(m)) \uparrow\]

Low precision

\[L(\tilde{f}(m)) \downarrow \quad \text{but} \quad -\log f(g \mid \tilde{f}(m)) \quad \text{may be } \gg \quad -\log p(g \mid \hat{f}(m)^{\text{ML}})\]

Optimal compromise (under regularity conditions, and asymptotic)

\[L(\text{each component of } f(m)) = \frac{1}{2} \log(n)\]

n, the sample size from which the parameter is estimated (growth rate of Fisher info.)
In our problem, \( \mathbf{v} = [x \ y] = \mathbf{B} \theta \) is a “digital” curve. Coordinates are quantized to pixel accuracy.

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

Let \( \Delta \theta_x = \tilde{\theta}_x - \theta_x \) and \( \Delta \theta_y = \tilde{\theta}_y - \theta_y \).

Finite precision versions

Goal: \( \| \Delta x \|_\infty \equiv \max_i |\Delta x_i| < 1 \) and \( \| \Delta y \|_\infty \equiv \max_i |\Delta y_i| < 1 \)

By linearity, \( \Delta x = \mathbf{B} \Delta \theta_x \) and \( \Delta y = \mathbf{B} \Delta \theta_y \).
Key fact:
\[ \|B\|_\infty = \max_i \sum_j |B_{ij}| = \max_i \sum_j B_{ij} = 1 \]

Induced matrix norm
\[ \|Bu\|_\infty \leq \|B\|_\infty \times \|u\|_\infty \]

Recall our goal: \( \|\Delta x\|_\infty < 1, \|\Delta y\|_\infty < 1 \)

and \( \Delta x = B \Delta \theta_x, \Delta y = B \Delta \theta_y \)

then,
\[ \|\Delta \theta_y\|_\infty < 1 \Rightarrow \|\Delta y\|_\infty < 1 \]
\[ \|\Delta \theta_x\|_\infty < 1 \Rightarrow \|\Delta x\|_\infty < 1 \]

i.e., pixel precision is enough for the control points.
Natural code length for \( k \) control points

\[
L(\theta_{(k)}) = k \left( \log(N_r) + \log(N_c) \right) = L(k)
\]

denotes \( k \) control points

MDL criterion:

\[
\min_{k, \theta_{(k)}, \sigma_x^2, \sigma_y^2} \left\{ L(k) - \log p(x | \theta_x, \sigma_x^2) - \log p(y | \theta_y, \sigma_y^2) \right\}
\]

some simple manipulation leads to

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

\[
\hat{\sigma}_x^2(k) = \frac{1}{n} \left\| x - B(k)^\perp x \right\|^2 \\
\hat{\sigma}_y^2(k) = \frac{1}{n} \left\| y - B(k)^\perp y \right\|^2
\]
Model selection criterion:

\[ \hat{k} = \arg \min_k \left\{ k \log(N_r, N_c) - n \log \sqrt{\hat{\sigma}^2_x(k) / \hat{\sigma}^2_x(k)} \right\} \]

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:

\[
p(I | v(\theta_{(k)}), \phi_{in}, \phi_{out}) = \prod_{i \in \text{inside}(v(\theta_{(k)}))} p(I_i | \phi_{in}) \prod_{i \in \text{outside}(v(\theta_{(k)}))} p(I_i | \phi_{out})
\]

where \( v(\theta_{(k)}) = B \theta_{(k)} \)
\( \phi_{\text{in}}, \phi_{\text{out}} \) are also considered unknown.

**MDL criterion:**

\[
\min_{k, \theta^{(k)}, \phi_{\text{in}}, \phi_{\text{out}}} \left\{ L(k) - \log p(I | v(\theta^{(k)}), \phi_{\text{in}}, \phi_{\text{out}}) \right\}
\]

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

**Proposed approach:** an iterative method.
\[
\min_{k, \theta_{(k)}, \phi_{\text{in}}, \phi_{\text{out}}} \left\{ L(k) - \log p(I \mid v(\theta_{(k)}), \phi_{\text{in}}, \phi_{\text{out}}) \right\}
\]

can be rewritten as

\[
\min_k \left\{ L(k) - \max_{\theta_{(k)}, \phi_{\text{in}}, \phi_{\text{out}}} \left\{ \log p(I \mid v(\theta_{(k)}), \phi_{\text{in}}, \phi_{\text{out}}) \right\} \right\}
\]

Solved by iterative method

\[
\min_k \left\{ L(k) - G(I, k) \right\}
\]

Outer minimization: solved by exhaustive search
\[
\max_{\theta^{(k)}, \phi_{\text{in}}, \phi_{\text{out}}} \left\{ \log p(I \mid v(\theta^{(k)}), \phi_{\text{in}}, \phi_{\text{out}}) \right\}
\]

Iterative method:

- Given \( \hat{\phi}_{\text{in}}, \hat{\phi}_{\text{out}} \)
  - maximize w.r.t. \( \theta^{(k)} \)
- Given \( \hat{v} = B \hat{\theta}^{(k)} \)
  - find ML estimates
    - \( \hat{\phi}_{\text{in}} \) and \( \hat{\phi}_{\text{out}} \)

\[
\hat{v} = B \hat{\theta}^{(k)}
\]
Given $\hat{\phi}_{\text{in}}$, $\hat{\phi}_{\text{out}}$

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

is equivalent to

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

Subject to: $v \in \mathcal{R} (B^{(k)})$

The range space of $B^{(k)}$, i.e., the span of its columns
\[
\max_v \left\{ \log p(I \mid v, \hat{\phi}_{in}, \hat{\phi}_{out}) \right\} \quad \text{Subject to: } v \in \mathcal{R}(B_{(k)})
\]

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

Step 0: build \( B_{(k)} \) and compute
\[
B_{(k)}^\perp = B_{(k)} \left( B_{(k)}^T B_{(k)} \right)^{-1} B_{(k)}^T
\]

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

Step 2: project the gradient onto \( \mathcal{R}(B_{(k)}) \):
\[
(\delta v)^\perp = B_{(k)}^\perp \delta v
\]

Step 3: take a small step in the direction of the projected gradient:
\[
\hat{\mathbf{v}}^{(t+1)} = \hat{\mathbf{v}}^{(t)} + \epsilon (\delta v)^\perp = B_{(k)}^\perp \left( \hat{\mathbf{v}}^{(t)} + \epsilon \delta v \right)
\]

No convergence: increment \( t \), back to Step 1
Computing the gradient

\[ \delta \mathbf{v} = \nabla \log p(I \mid \mathbf{v}) \bigg|_{\mathbf{v}=\hat{\mathbf{v}}^{(t)}} \]

Gradient is perpendicular to the contour

Coordinate i of the gradient:
approximated with a finite difference

Only requires values on a small perpendicular window
\[
\min_k \left\{ L(k) - \max_{\theta(k), \phi_{\text{in}}, \phi_{\text{out}}} \left\{ \log p(I | v(\theta(k)), \phi_{\text{in}}, \phi_{\text{out}}) \right\} \right\}
\]

Solved by iterative method

\[
\min_k \left\{ L(k) - G(I, k) \right\}
\]

Outer minimization: solved by exhaustive search

Sweep range of values \( k \in \{k_{\text{min}}, k_{\text{min}} + 1, \ldots, k_{\text{max}}\} \)

Start with \( k = k_{\text{min}} \)

Use contour obtained at each \( k \), to initialize the next iterative algorithm
Contour estimation examples: synthetic data

$k = 5$

$k = 7$

$k = 9$

$k = 11$

$k = 13$

Dashed line = initial contour

game variance, different means

description length

Number of control points
Contour estimation example: real medical image

- $k = 6$
- $k = 8$
- $k = 10$
- $k = 12$
- $k = 14$
- $k = 16$

Description length

Number of control points
More examples on real medical images

See: