IV. Curve detection: semi-global methods

- Instantiation parameters θ directly in terms of constraints the locations of the n points along the curve.

  $\theta Z = \theta = (\theta_1, \ldots, \theta_n)$.

- Prior formulated directly in terms of the locations - favoring small curvature.

- Assume data between pairs of consecutive points on curve independent.

  Model data along curve and estimate relevant parameters.

- Initialization through two points on the curve.

- Global optimization using dynamic programming.

IV.1 The Bayesian model

- Transform data into collection of point processes.

- Curve can either traverse a pixel x at one of A different angles: $\alpha(x) = a, a = 1, \ldots, A$

  or no curve traverses the pixel: $\alpha(x) = 0$.

- $X_a(x), a = 1, \ldots, A$ collection of local binary features (see below).

  $X_a(x)$ is more likely to be 1 if $\alpha(x)$ close to a.

$P(X_a(x) = 1|\alpha(x) = \alpha) = p_{\alpha,a}$,

$P(X_a(x) = 1|\alpha(x) = 0) = p_{0,a}$.

Data transform: $FI(x) = (X_1(x), \ldots, X_A(x))$.

- Conditional independence of processes at a pixel:

$$P(FI(x)|\alpha(x) = \alpha) = \prod_{a=1}^{A} p_{\alpha,a}^{X_a(x)}(1 - p_{\alpha,a})^{(1-X_a(x))} \tag{IV.1}$$

and given no curve passes through the pixel $\alpha(x) = 0$

$$P_0(FI(x)) = P(FI(x)|\alpha(x) = 0) = \prod_{a=1}^{A} p_{0,a}^{X_a(x)}(1 - p_{0,a})^{(1-X_a(x))}. \tag{IV.2}$$

- For instantiation $\theta Z = \theta$: L_i - set of pixels on segment θ_i, θ_{i+1}. $L(\theta) = \cup_i L_i$. α_i angle of segment L_i.

$x \in L(\theta) \implies x \in L_i$, for some $i \implies \alpha(x) = \alpha_i$.

- Assume conditional independence of $X_a(x), x \in L, a = 1, \ldots, A$, given θ.

Conditional distribution of FI given the curve is at instantiation θ:

$$\begin{aligned} P(FI|\theta) &= \prod_{x \in L(\theta)} P(FI(x)|\alpha(\theta, x)) \cdot \prod_{x \notin L(\theta)} P_0(FI(x)) \\ &= \prod_{i=1}^{n} \prod_{x \in L_i} P(FI(x)|\alpha_i) \prod_{x \notin L(\theta)} P_0(FI(x)) \end{aligned} \tag{IV.3}$$

Divide by probability of FI given *no* curve:

$$\begin{aligned} \frac{P(FI|\theta)}{P_0(FI)} &= \prod_{x \in L(\theta)} \frac{P(FI(x)|\alpha(\theta, x))}{P_0(FI(x))} \\ &= \prod_{i=1}^{n} \prod_{a=1}^{A} \left[\frac{p_{\alpha_i,a}}{p_{0,a}}\right]^{n_{ia}} \left[\frac{1 - p_{\alpha_i,a}}{1 - p_{0,a}}\right]^{(n_i - n_{ia})} \end{aligned} \tag{IV.4}$$

$$\tag{IV.5}$$

n_i - number of pixels on L_i,

$n_{ia} = \sum_{x \in L_i} X_a(x)$ - number of hits of feature a on segment i.

- The log- likelihood is a sum of terms depending on pairs of consecutive points θ_i, θ_{i+1} along the curve:

$$l(FI|\theta) = C + \sum_{i=1}^{n} \psi_i(FI_{L_i}, \theta_i, \theta_{i+1}),$$

where C does not depend on θ,

$FI_{L_i} = \{X_1(x), \ldots, X_a(x); x \in L_i\}$.

$\psi_i(FI_{L_i}, \theta_i, \theta_{i+1}) = \sum_{a=1}^{A} \left[n_{ia} \log \frac{p_{\alpha_i,a}}{p_{0,a}} + (n_i - n_{ia}) \log \frac{1-p_{\alpha_i,a}}{1-p_{0,a}} \right]$.

- The independence assumptions yield a simple linear form to ψ_i in terms of counts. Other forms of these functions are possible.

IV.1.1 The point processes

- Photometric invariance: significant variations in the range of intensities in the neighborhood of the curve - define X_a invariant to linear and other smooth gray level transformations.

- If the curve is expected to be brighter than its surroundings let

$$X_a^1(x) = 1, \text{ if } I(x) > I(x + \nu_a) \text{ and } I(x) > I(x - \nu_a),$$

ν_a vector of fixed length in direction orthogonal to a.

- If curve brighter or darker than its surroundings but relatively constant in intensity:

$$X_a^2(x) = 1, \text{ if } |I(x + \delta_a) - I(x)| < \min \Big(|I(x) - I(x + \nu_a)|, |I(x) - I(x - \nu_a)| \Big) \tag{IV.6}$$

δ_a vector of fixed length in direction of a.

- One can also use a conjunction of both filters i.e.

$$X_a(x) = X_a^1(x) \cdot X_a^2(x) \tag{IV.7}$$

- $p_{\alpha,a}, a = 1, \ldots, A$ estimated by maximum likelihood using samples of small subimages extracted around pixels from each of the categories: $\alpha = 0, \ldots, A$.

- Simplifying the model:

  $p_{\alpha,a} = p_b$ if $\alpha \neq a$,

  $p_{\alpha,\alpha} = p$ for all $\alpha = 1, \ldots, A$.

  Estimate p using all subimage samples containing a curve.

  Estimate p_b for subimages without a curve.

  Then equation IV.4 becomes:

$$\prod_{i=1}^{n} \left[\frac{p}{p_b}\right]^{n_{i\alpha_i}} \left[\frac{1-p}{1-p_b}\right]^{(n_i - n_{i\alpha_i})} \tag{IV.8}$$

- Non-parametric approach: Assume n_i constant. Use subimage samples containing curve segments. For each segment with angle α count the number of pixels for which $X_a(x) = 1$.

  Use histogram for each pair α, a to approximate $P(\alpha, a, n_{ia})$.

  Simplification possible with only two histograms as above.

IV.1.2 The prior and the posterior

- Prior assumptions on smoothness - low curvature:

$$P(\theta_1, \ldots, \theta_n) \propto \exp\left[-\mu\left(\sum_{i=1}^{n-2} \phi_i(\theta_i, \theta_{i+1}, \theta_{i+2})\right)\right],$$

  where $\phi_i(\theta_i, \theta_{i+1}, \theta_{i+2})$ is the modulus of the difference between the angle of the segment θ_i, θ_{i+1}, and that of the segment $\theta_{i+1}, \theta_{i+2}$.

  If θ_i in complex plane - $\phi_i(\theta_i, \theta_{i+1}, \theta_{i+2}) = \left| arg\left(\frac{\theta_{i+2}-\theta_{i+1}}{\theta_{i+1}-\theta_i}\right)\right|$. Each such term depends on a triple of consecutive points.

- Assume a template shape for the curve $z_1, \ldots, z_n$. ϕ could include a term penalizing deviation of angles:
  $\left| arg\left(\frac{\theta_{i+2}-\theta_{i+1}}{\theta_{i+1}-\theta_i} / \frac{z_{i+2}-z_{i+1}}{z_{i+1}-z_i}\right)\right|$.

- Posterior:
$$\log P(\theta_1, \ldots, \theta_n | FI) = C + \sum_{i=1}^{n-1} \Phi_i(\theta_i, \theta_{i+1}, \theta_{i+2}), \tag{IV.9}$$
  where
$$\Phi_i(\theta_i, \theta_{i+1}, \theta_{i+2}) = \phi_i(\theta_i, \theta_{i+1}, \theta_{i+2}) + \psi_i(FI_{L_i}, \theta_i, \theta_{i+1})$$
  for $i = 1, \ldots, n-1$ and setting $\phi_{n-1} = 0$.

IV.2 Computation - Dynamic programming

- Structure of log-posterior: a sum of terms depending on three consecutive variables. Can be efficiently optimized using dynamic programming.

- Impractical if state space for each variable consists of all locations in the image.

  Introduce hard constraints on the locations of the points. Assume the first and last points are given within neighborhoods S_1, S_n in the image. Determine regions $S_2, \ldots, S_{n-1}$: for any reasonable configuration $\theta_1, \ldots, \theta_n$: $\theta_i \in S_i, i = 1, \ldots, n$.

- **Algorithm:** Assume the points in S_i and $S_i \times S_{i+1}$ are ordered.

  H_i array with two columns indexed by points in $S_i \times S_{i+1}$ - initialized to 0.

  1. Set $i = n - 2$.
  2. While $i > 0$, do

     For every pair $p = (x, y) \in S_i \times S_{i+1}$ find the point $z \in S_{i+2}$ which optimizes $\Phi_i(x, y, z) + H_{i+1}(\bar{p}, 2)$, where $\bar{p} = (y, z)$. Store the index of the optimal point $z = \theta_{i+2}(p) \in S_{i+2}$ in $H_i(p, 1)$. In $H_i(p, 2)$ store the value of $\Phi_i(x, y, \theta_{i+2}(x, y)) + H_{i+1}(\bar{p}, 2)$, where $\bar{p} = (y, \theta_{i+2}(p))$.

Set $i = i - 1$.

3. Loop over the array H_1 and find the row with highest value Φ^* in the second column. Let (θ_1^*, θ_2^*) be the pair corresponding to that row and θ_3^* be the index in the first column.

   Set $i = 2$

4. While $i < n - 1$, do

   Set θ_{i+2}^* to be $H_i(p, 1)$, where $p = (\theta_i^*, \theta_{i+1}^*)$.

5. The configuration $\theta_i^* \in S_i, i = 1, \ldots, n$ is the optimal configuration and the associated cost is Φ^*.

IV.3 Example

- Data term ψ_i of the functions Φ_i depends only on two consecutive points.

  Possible to simplify prior functions ϕ_i only to measure deviations of $\theta_{i+1} - \theta_i$ from $z_{i+1} - z_i$.

- log-posterior then has form:

$$\log P(\theta_1, \ldots, \theta_n | FI) = C + \sum_{i=1}^{n-1} \Phi_i(\theta_i, \theta_{i+1}), \tag{IV.10}$$

  where

$$\Phi_i(\theta_i, \theta_{i+1}) = \phi_i(\theta_i, \theta_{i+1}) + \psi_i(FI_{L_i}, \theta_i, \theta_{i+1})$$

- The rows of H_i indexed by single elements of S_i.

- Figure IV.3 - outcome of simplified version.

  Initial two points - end points of the interior curve.

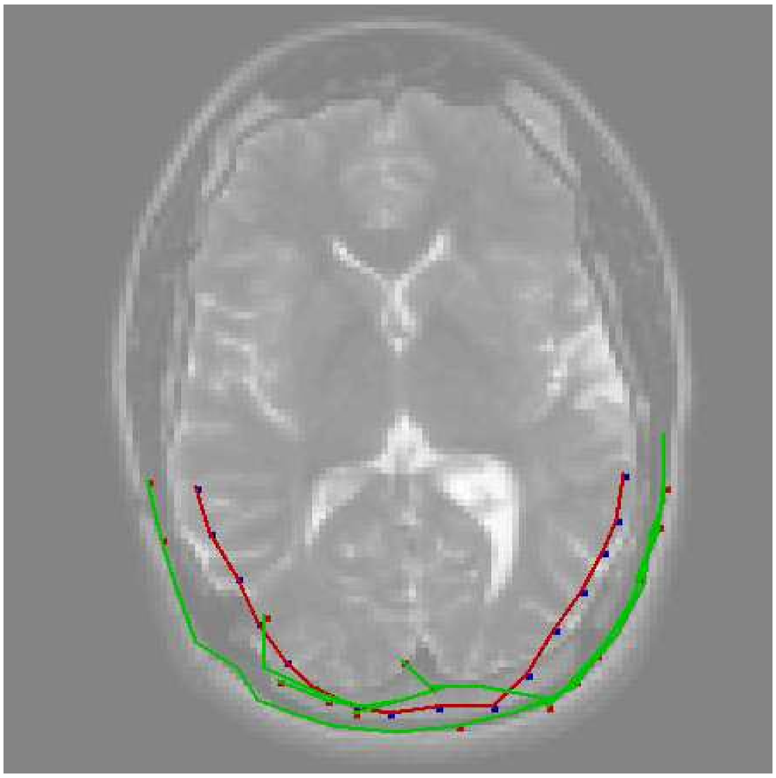

Figure IV.1: Dynamic programming to detect scalp in axial MRI image. Top: Points along red curve determine regions S_i - radius 10 pixels. Green curves: steps of dynamic programming.

Regions S_i - radius 10 pixels around each intermediate point of the red curve.

Point process - as in equation IV.7: model the data on the outline of the scalp.

Green curves show the stages of dynamic programming. At each step $i = n, n-1, \ldots, 1$ - the best match from point $\theta_i, \ldots, \theta_n$.

Note that at times the best match wanders off in wrong directions.

The final match does indeed lie along the outer scalp.

IV.4 Discussion

- A relaxation algorithm as in section II starting at the initial interior curve of figure IV.3 would have had a hard time overcoming the obstacles on the way to the exterior scalp. In particular the region in between the scalp and the brain.

- Similar types of algorithms are described in Geiger et al. (1995) and Petrocelli et al. (1992).

- The optimization is semi-global because there are the initial constraints in terms of the sets $S_1, \ldots, S_n$ which depend on the two initial points provided by the user.

  If S_i get large dynamic programming slows down significantly.

- Independence assumptions are not valid at the level of such simple local features - the point processes are very dense and local correlations are strong.

- Move to point processes with lower density: independence assumptions more realistic, full global optimization possible: section V.4.