A Particle Filter Framework for Contour Detection

Nicolas Widynski and Max Mignotte – *surname*@iro.umontreal.ca

Aim

We propose a particle filter framework to automatically track and detect contours in natural images. The hidden state vector, $(\mathbf{e}_t, \mathbf{c}_t)$, combines:

- $\mathbf{e}_t$, an edgelet, i.e. a small set of 4-connected pixels
- $\mathbf{c}_t$, a jump variable, indicates if the tracking of a current contour is over or not

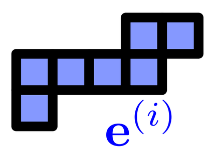
Contributions include:

- Edgelet *offline learning* stage for the *prior* and *transition* distributions
- Edgelet *online learning* stage for the *likelihood* and *initialization* distributions
- Edgelet *tracking* stage for the contour detection

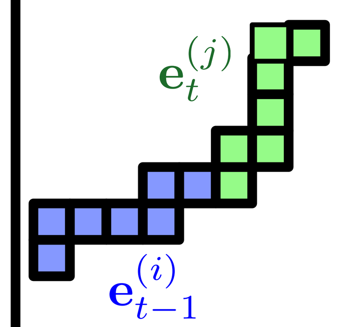
Learning

Offline

$\mathbf{e} = (\mathbf{e}^1, \ldots, \mathbf{e}^7)$



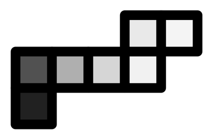
Prior
$p(\mathbf{e}^{(i)})$



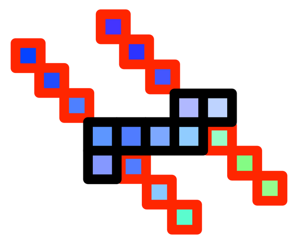
Transition
$p(\mathbf{e}_t^{(j)} | \mathbf{e}_{t-1}^{(i)})$

Online

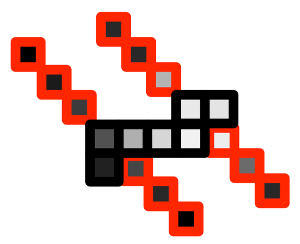
☐ Feature distributions:



Local gradient
$P(\mu_1 > f_1(\mathbf{e}_0^{(k)}, \mathbf{y}_0^1))$



Textural gradient
$P(\mu_2 > f_2(\mathbf{e}_0^{(k)}, \mathbf{y}_0^2))$



Profile gradient
$P(\mu_3 > f_3(\mathbf{e}_0^{(k)}, \mathbf{y}_0^3))$

☐ Initialization:

$p(\mathbf{e}_0 | \mathbf{y}_0)$
$\propto$ Likelihood $\times$ Prior
$= p(\mathbf{y}_0 | \mathbf{e}_0) \times p(\mathbf{e}_0)$

with

$$p(\mathbf{y}_0 | \mathbf{e}_0) = \prod_{j=1}^{J} p(\mathbf{y}_0^j | \mathbf{e}_0)$$

$$p(\mathbf{y}_0^j | \mathbf{e}_0) \propto \exp\left(-\lambda_j P(\mu_j > f_j(\mathbf{e}_0, \mathbf{y}_0^j)\right)$$

Tracking

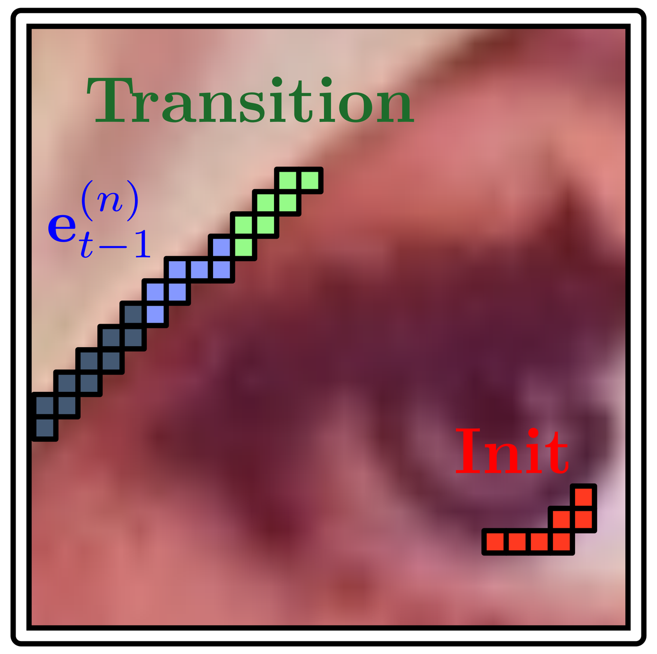
$\forall t \in \{1, \ldots, T\}$, $\forall n \in \{1, \ldots, N\}$,

Proposition

$$\mathbf{c}_t^{(n)} \sim q(\mathbf{c}_t | \ldots) = \begin{cases} \alpha = \sum_j \tilde{\lambda}_j P(\mu_j > f_j(\mathbf{e}_{t-1}, \mathbf{y}_{t-1}^j)) & \text{if } \mathbf{c}_t = \text{jump}, \\ 1 - \alpha & \text{otherwise.} \end{cases}$$

$$\mathbf{e}_t^{(n)} \sim q(\mathbf{e}_t | \ldots) = \begin{cases} \text{Init } p(\mathbf{e}_t | \mathbf{y}_0) & \text{if } \mathbf{c}_t^{(n)} = \text{jump}, \\ \text{Transition } p(\mathbf{e}_t | \mathbf{e}_{t-1}^{(n)}) & \text{otherwise.} \end{cases}$$

$$\times \ f_{\text{check}}\left(\mathbf{e}_t; \mathbf{e}_{1:t-1}^{(n)}\right) \text{ consistency check of the trajectory}$$



Weighting

$$w_t^{(n)} = \text{PastWeight} \times \frac{\text{Likelihood} \times \text{Prediction}}{\text{Proposition}}$$

$$= w_{t-1}^{(n)} \frac{p(\mathbf{y}_t | \mathbf{e}_t^{(n)})\, p(\mathbf{e}_t^{(n)} | \mathbf{e}_{t-1}^{(n)}, \mathbf{c}_t^{(n)})\, p(\mathbf{c}_t^{(n)})}{q(\mathbf{e}_t^{(n)} | \ldots)\, q(\mathbf{c}_t^{(n)} | \ldots)}$$

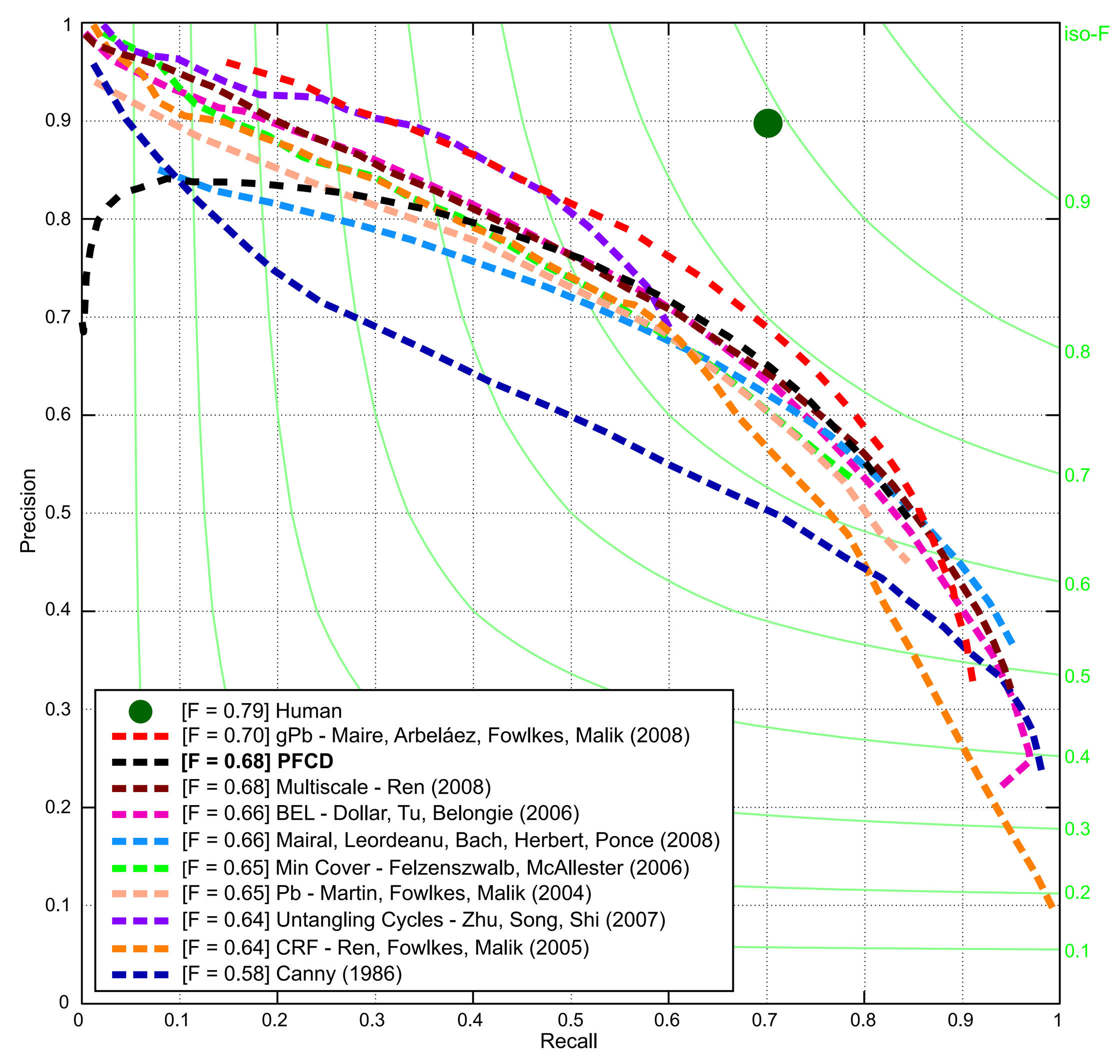
Output

Approximation of the posterior distribution

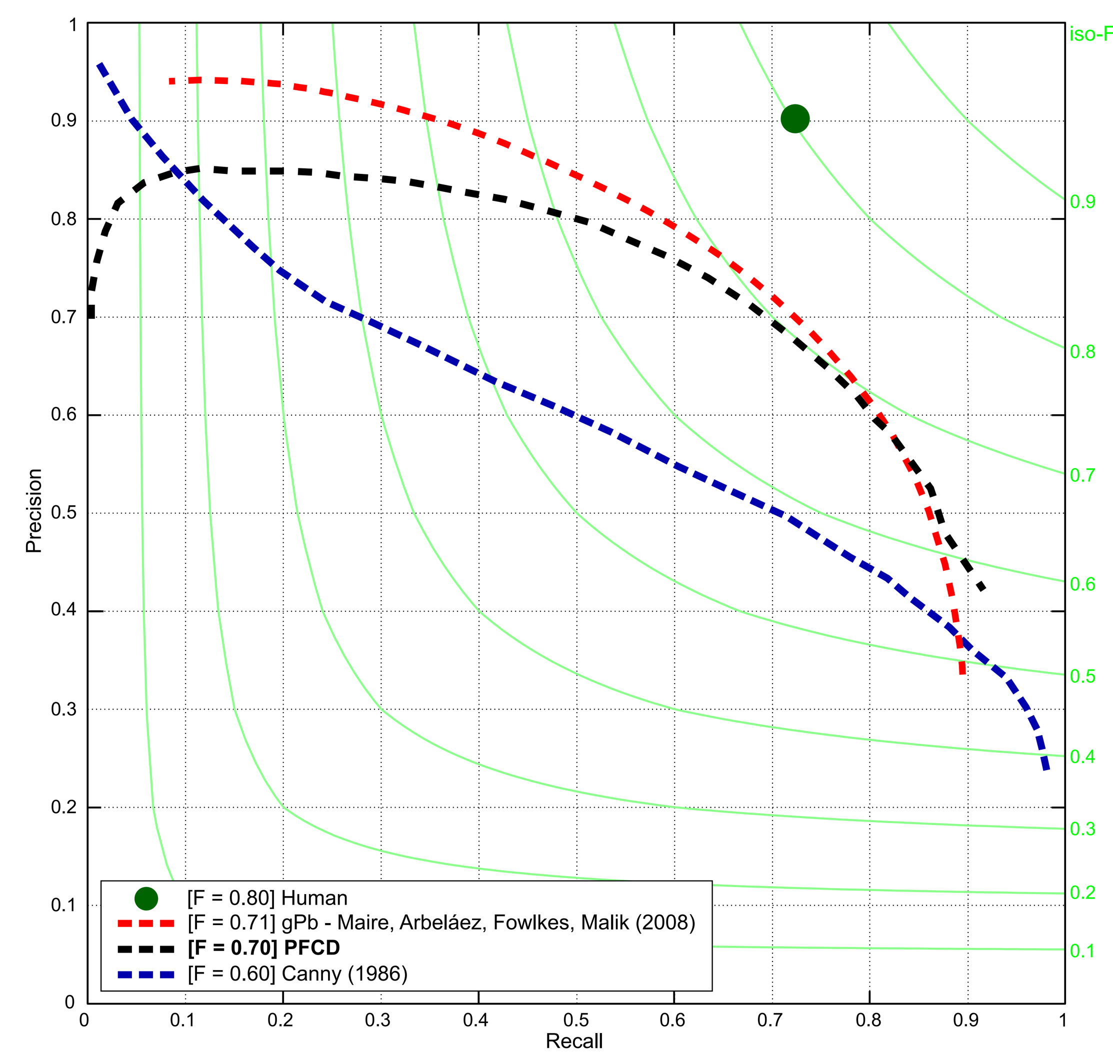
$$p(\mathbf{e}_{0:t}, \mathbf{c}_{0:t} | \mathbf{y}_{0:t}) \approx \sum_{n=1}^{N} \tilde{w}_t^{(n)} \, \delta_{\mathbf{e}_{0:t}}^{\mathbf{e}_{0:t}^{(n)}} \, \delta_{\mathbf{c}_{0:t}}^{\mathbf{c}_{0:t}^{(n)}}$$

Results

BSDS300



BSDS500



Conclusion

- Integration of a semi-local structure in a recursive Bayesian filter
- Mixed offline and online procedures
- BSDS300: 0.70 (r: 0.71, p: 0.65)
- BSDS500: 0.71 (r: 0.72, p: 0.68)
- Runs in 4 minutes and can be parallelized

Perspectives

- Multiscale extension
- Semi-interactive detection for cut-out application

University of Montreal
widynski@iro.umontreal.ca
www-etud.iro.umontreal.ca/~widynski/