Module 4
Tracking Cells using the Particle Filter

Scott T. Acton

Virginia Image and Video Analysis (VIVA), Charles L. Brown Department of Electrical and Computer Engineering
Department of Biomedical Engineering
University of Virginia, Charlottesville, VA 22904

Virginia Image and Video Analysis

Developing Leaders of Innovation

"I find that the harder I work, the more luck I seem to have."

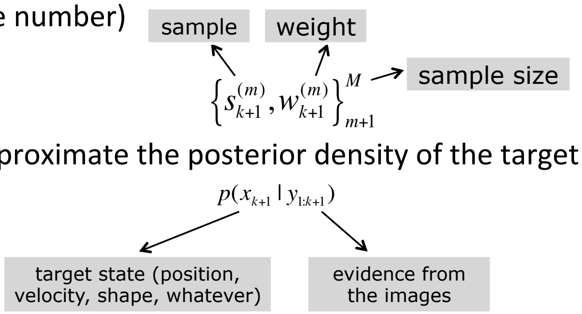
- Thomas Jefferson (1743-1826)

Goal

- Instead of one "best guess," pursue a multitude of "guesses" about the target state
- What's presented here
  - A simple particle filter implementation useful for cell tracking
- Not presented here
  - The "full breadth" of particle filter methods – this example is a Monte Carlo approach to tracking…

Particle Filter Tracking

- Use a sample set (these are the particles!) (*k* is the current frame number)



sample

weight

sample size

$$\left\{ s_{k+1}^{(m)}, w_{k+1}^{(m)} \right\}_{m+1}^{M}$$

to approximate the posterior density of the target state

$$p(x_{k+1} \mid y_{1:k+1})$$

target state (position, velocity, shape, whatever)

evidence from the images

- For each new frame in a video sequence, we:
  - Generate samples of the target state (e.g., position) using a motion model
  - Compute sample likelihood by the observation model

PF Tracker (an example for biological imaging)

- Position prediction
  - horizontal movement does not change dramatically (assumption: cell is moving horizontally)
  - vertical movement is negligible

predicted position

$$\bar{x}_{k+1} = \hat{x}_k + \alpha(\hat{x}_k - \hat{x}_{k-1}) + (1-\alpha)(\hat{x}_{k-1} - \hat{x}_{k-2})$$

$$\bar{y}_{k+1} = \hat{y}_k$$

estimated position

- Sample generation
  - Generate samples randomly with a 2-D Gaussian distribution (for example)

$$x_{k+1}^{(m)} = \bar{x}_{k+1} + R\cos\beta \qquad R \sim N(0, \sigma_R^2)$$

$$y_{k+1}^{(m)} = \bar{y}_{k+1} + R\sin\beta \qquad \beta \sim U(0, 2\pi)$$

PF Tracker

- Radial edge detection
  - construct N line segments extending radially from each sample
  - specify K+1 points on each line segment

$$x_{i,j} = x_{k+1}^{(m)} + [r_1 + (r_2 - r_1)\, j/K]\cos(2\pi i/N) \qquad i = 0, 1, \ldots, N-1$$

$$y_{i,j} = y_{k+1}^{(m)} + [r_1 + (r_2 - r_1)\, j/K]\sin(2\pi i/N) \qquad j = 0, 1, \ldots, K$$

  - apply edge detection operator on each line segment

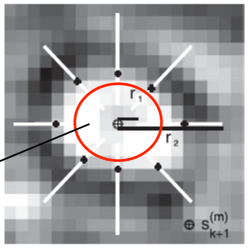
$$e_{i,j} = I_{i,j-1} + 2I_{i,j} - 2I_{i,j+1} - I_{i,j+2}$$

  - obtain local image intensity measurement

$$r_{\theta,k+1}^{(m)} \qquad \sigma_{I,k+1}^{(m)}$$

detected radius in each radial direction

intensity standard deviation of detected target interior



PF Tracker

- Sample weighting
  - assume target center position in the first frame is known
  - for the first frame, apply radial edge detection at the center position to obtain image intensity measurement as reference

standard deviation of radius measurement from 1st frame
↗
average radius from 1st frame ← $\bar{r}$ σ_r σ_I → intensity standard deviation of detected target interior from 1st frame

  - for each sample in a new frame, define difference measures

$$d_{1,k+1}^{(m)} = \left| \sum_{\theta} \left(r_{\theta,k+1}^{(m)} - \bar{r} \right)^2 - N\sigma_r^2 \right| \qquad d_{2,k+1}^{(m)} = \left| \sigma_{I,k+1}^{(m)} - \sigma_I \right|$$

  - assign a normalized weight to each sample

$$z_{k+1}^{(m)} = e^{-d_{1,k+1}^{(m)2} / 2\sigma_1^2} e^{-d_{2,k+1}^{(m)2} / 2\sigma_2^2} \qquad w_{k+1}^{(m)} = \frac{z_{k+1}^{(m)}}{\sum_{i=1}^{M} z_{k+1}^{(i)}}$$

A Monte Carlo Tracker

- Target position estimation
  - weighted sum of the sample set is the estimated target position

$$\left(\hat{x}_{k+1}, \hat{y}_{k+1} \right) = \sum_{m=1}^{M} w_{k+1}^{(m)} s_{k+1}^{(m)} \qquad s_{k+1}^{(m)} = \left(x_{k+1}^{(m)}, y_{k+1}^{(m)} \right)$$

END

- We showed one possibility of using the particle filter for tracking cells

*example in PFTracker