Computer Vision
for HCI

Mean-Shift Tracking

1

Mean Shift Tracking

Based on: "Kernel-Based Object Tracking", D. Comaniciu, V. Ramesh, P. Meer
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, 564-575, 2003

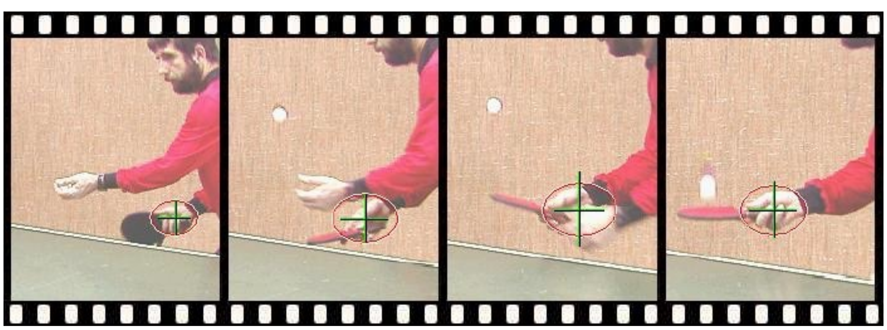
2

Overview

- Real-time non-rigid object tracking
- Mean shift tracking
  - Represent objects using color histograms
  - Maximization of similarity function using mean shift

3

Non-Rigid Object Tracking



… → …

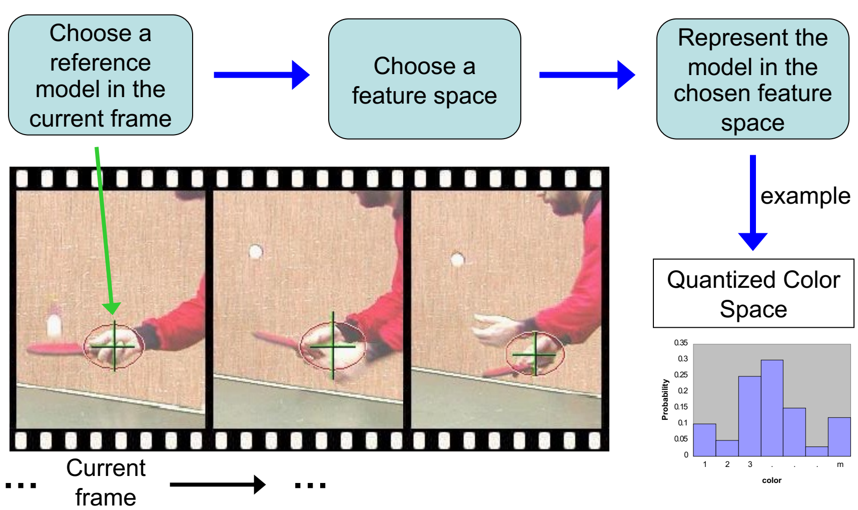
4

Algorithm

- Obtain a statistical distribution q for <u>current</u> appearance of the object of interest
- Get <u>next</u> video image
- For a candidate at location **y** = (x,y) in the <u>new</u> image, obtain a statistical distribution p(**y**)
- <u>Search</u> the neighborhood of **y** in the new image
  - Find new "better matching" location $\mathbf{y}_{new}$ using *mean shift* where the distribution $p(\mathbf{y}_{new})$ is the most similar to q

5

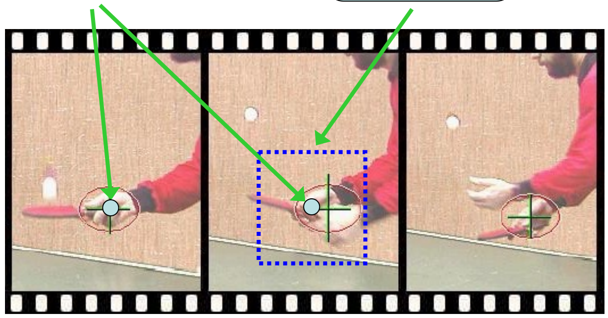
General Framework: Target Representation



6

General Framework: Target Localization

Start from the position of the model in the current frame → Search in the neighborhood in next frame → Choose best candidate location by maximizing a similarity func.



Repeat the same process in the next pair of frames

Model

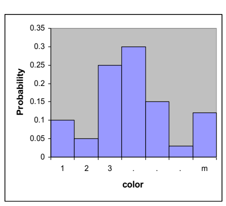
Candidate

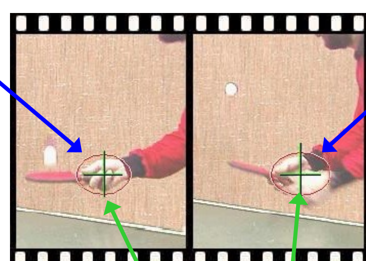
… Current frame → …

7

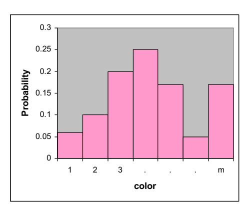
PDF Representation as m-bin Histograms

Target Model
(centered at **0**)





Candidate
(centered at **y**)



$\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\ldots m}$

$\sum_{u=1}^{m} \hat{q}_u = 1$

$\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m}$

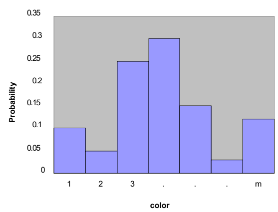
$\sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1$

Similarity Function: $\hat{\rho}(\mathbf{y}) = \rho[\hat{\mathbf{q}}, \hat{\mathbf{p}}(\mathbf{y})]$

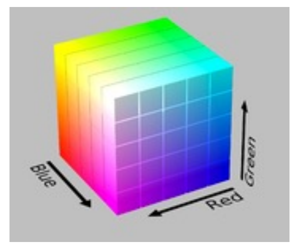
8

Color Histogram/Distribution

1-D quantized color
(grayscale)



3-D quantized color
(RGB)



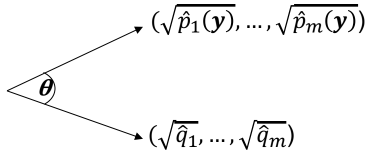
9

Similarity Function

Similarity between two discrete distributions estimated using Bhattacharyya Coefficient

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} = \cos\theta$$

Interpretation: Cosine of angle or (normalized) correlation between *m*-dimensional unit vectors



$$\sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1 \qquad \sum_{u=1}^{m} \hat{q}_u = 1$$

10

Similarity Function and Associated Problems

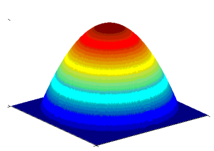
- Local maxima of similarity function gives the location of best match in the next frame
- If only color information is used, spatial information is lost (histograms), and it is not always a smooth surface
  - e.g., noisy pixels on periphery can cause drastic changes in the similarity function in adjacent locations
- Smooth the similarity function by using weighted histograms
  - Weight pixel contributions based on their spatial location (how close to center of region)

11

"Model"

- Use a fixed circular region of radius h centered at $\boldsymbol{x_0}$
  - Could squash/scale/normalize elliptical regions instead
- Let $\{\mathbf{x}_i\}_{i=1\ldots n}$ be the "centered" pixel locations
- Let $b(\mathbf{x})$ be the color bin index (1…m) of a pixel at location $\mathbf{x}$

Now need a differentiable, isotropic, monotonically decreasing kernel to assign smaller weights to pixels at periphery of the circular region of radius h



12

Choice of Smoothing Kernel

Use Epanechnikov profile:

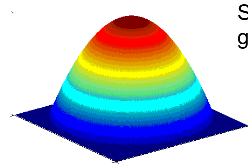
$$k(r) = \begin{cases} 1-r & \text{if } r<1 \\ 0 & \text{otherwise} \end{cases}$$

Distance from center (0,0)

"bandwidth" size

$$r = [\mathrm{sqrt}(x*x + y*y) / h]^2$$

Squared input gives this shape



13

Weighted "Model"

Probability of feature (color) u=1…*m* in the target model is computed as:

$$\hat{q}_u = C\sum_{i=1}^{n} \underbrace{k\left(\left\|\frac{\mathbf{x}_0-\mathbf{x}_i}{h}\right\|^2\right)}_{\text{pixel weight}} \underbrace{\delta\left[b(\mathbf{x}_i)-u\right]}_{\text{pixel bin color index is } u?}$$

Kronecker delta function: d(0)=1, d(t)=0 otherwise

C is just a normalization constant to make **q** a pdf

14

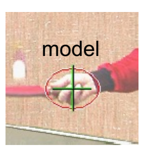
"Candidate" Region Model

- In the new image
- Let $\{\mathbf{x}_i\}_{i=1\ldots}$ be the pixel locations in valid bandwidth area h centered at $\mathbf{y}$
  - Need to center pixels around $\mathbf{y}$
- Hence, probability of color feature u=1…m in the candidate is given by

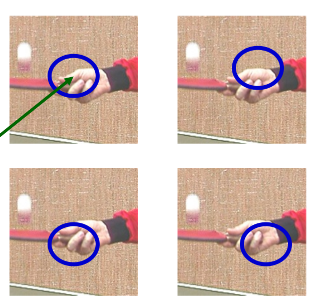
$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \delta[b(\mathbf{x}_i) - u]$$

15

Target Localization



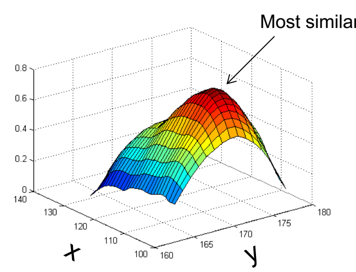
candidates



y is 2D Position (x,y)

"Search Surface"

similarity function evaluated at different locations



16

Maximizing Similarity Function

$$\hat{\rho}(\mathbf{y}) = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u}$$

Model location: $\mathbf{y}_0$

Candidate location: $\mathbf{y}$ (near $\mathbf{y}_0$)

$$\hat{\rho}(\mathbf{y}) \approx \underbrace{\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathbf{y}_0)\hat{q}_u}}_{\text{Independent of } \mathbf{y}} + \underbrace{\frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(\mathbf{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}}}_{\text{Maximize}}$$

Linear approximation around $\mathbf{y}_0$ (Taylor expansion)

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)\delta[b(\mathbf{x}_i)-u]$$

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}}\delta[b(\mathbf{x}_i)-u]$$

$$\frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)$$

17

Maximization by "Mean Shift"

Maximize the Bhattacharyya Coefficient by finding the mode (peak) of this density in the local neighborhood

$$\frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)$$

Use mean shift algorithm recursively to keep moving to a newer/better location given by (see paper ref):

NOTE: need to be able to "climb up" the mode (solution needs to "be nearby")

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\|\frac{\mathbf{y}_0-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\mathbf{y}_0-\mathbf{x}_i}{h}\right\|^2\right)}$$

$$g(r) = -k'(r)$$

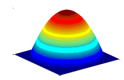
18

Choice of Smoothing Kernel

If use a radially symmetric kernel such as a one with an Epanechnikov profile

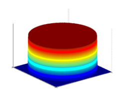
“Profile”
$r = [\text{sqrt}([x*x + y*y])/h]^2$

$$k(r) = \begin{cases} 1 - r & \text{if } r < 1 \\ 0 & \text{otherwise} \end{cases}$$



the derivative of whose profile is constant (uniform)…

$$g(r) = -k'(r) = \begin{cases} 1 & \text{if } r < 1 \\ 0 & \text{otherwise} \end{cases}$$



19

Simplified Mean Shift Vector

…then the mean shift vector (with an Epanechnikov kernel) becomes:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{h} \right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{h} \right\|^2\right)}$$

simplifies to

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i}{\sum_{i=1}^{n_h} w_i}$$

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(\mathbf{y}_0)}} \delta[b(\mathbf{x}_i) - u]$$

(just a weighted average!)

20

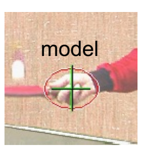
Algorithm

- STEP 0: Generate model $\hat{q}_u$
- STEP 1: Generate target $\hat{p}_u$ in current frame
  (start at previous frame location $\boldsymbol{y}_0$)
- STEP 2: Compute weights w_i
- STEP 3: Find <u>next best location</u> of the target candidate using $\mathbf{y}_1 = \dfrac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i}{\sum_{i=1}^{n_h} w_i}$
- STEP 4: If $\|\mathbf{y}_1 - \mathbf{y}_0\| < \varepsilon$ then stop, otherwise set $\mathbf{y}_0 \leftarrow \mathbf{y}_1$ and go to STEP 1

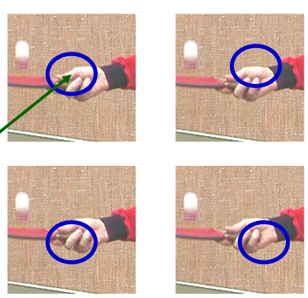
NOTE #1: Do **NOT** round <u>any</u> values for locations during the iterations!
NOTE #2: Could do for multiple bandwidth scales *h* and retain best (if object size/scale changes).

21

"It climbs up the search surface"



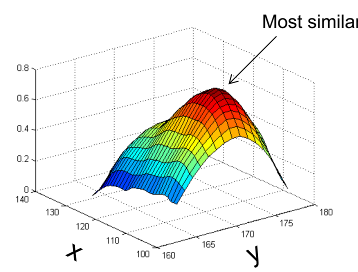
candidates



y is 2D Position (x,y)

"Search Surface"

similarity function evaluated at different locations



22

Mean-Shift Object Tracking

Results

Feature space: 16x16x16 quantized RGB
Target: manually selected on 1st frame
Average mean-shift iterations: 4.19

23

Mean-Shift Object Tracking

Results


Partial occlusion


Distraction


Motion blur

24

Summary

- Algorithm for non-rigid object tracking
- Mean Shift tracking
  - Weighted histograms using spatial kernels
  - Evaluating similarity between distributions using Bhattacharyya coefficient
  - Object tracking by target localization (in each frame) by maximizing the similarity function using mean shift

25