Introduction to Computer Vision

Michael J. Black

Nov 2009

Dense flow and Tracking

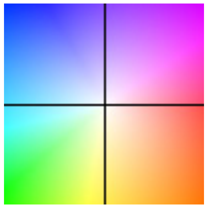
Goals

- Today
  - Finish dense flow
  - Start tracking (e.g. for homework)
- Monday
  - Particle filtering

Optical flow

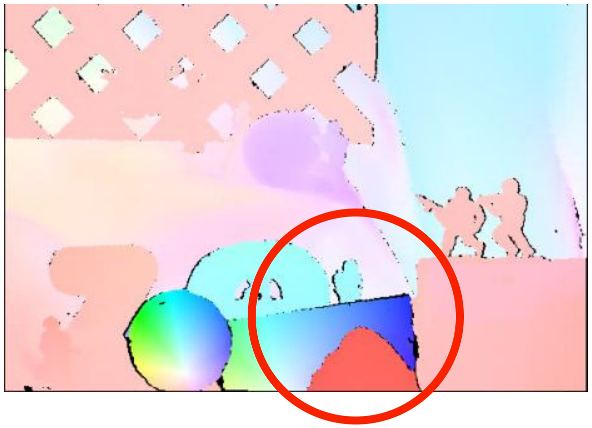
"Army"

Horn & Schunck 1981

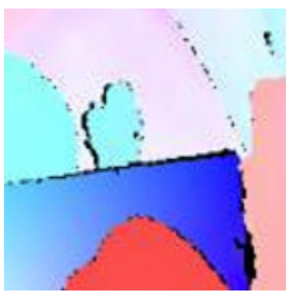


Key

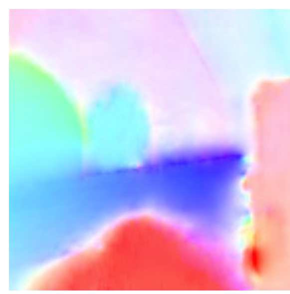
Two standard methods



"Army": Ground truth

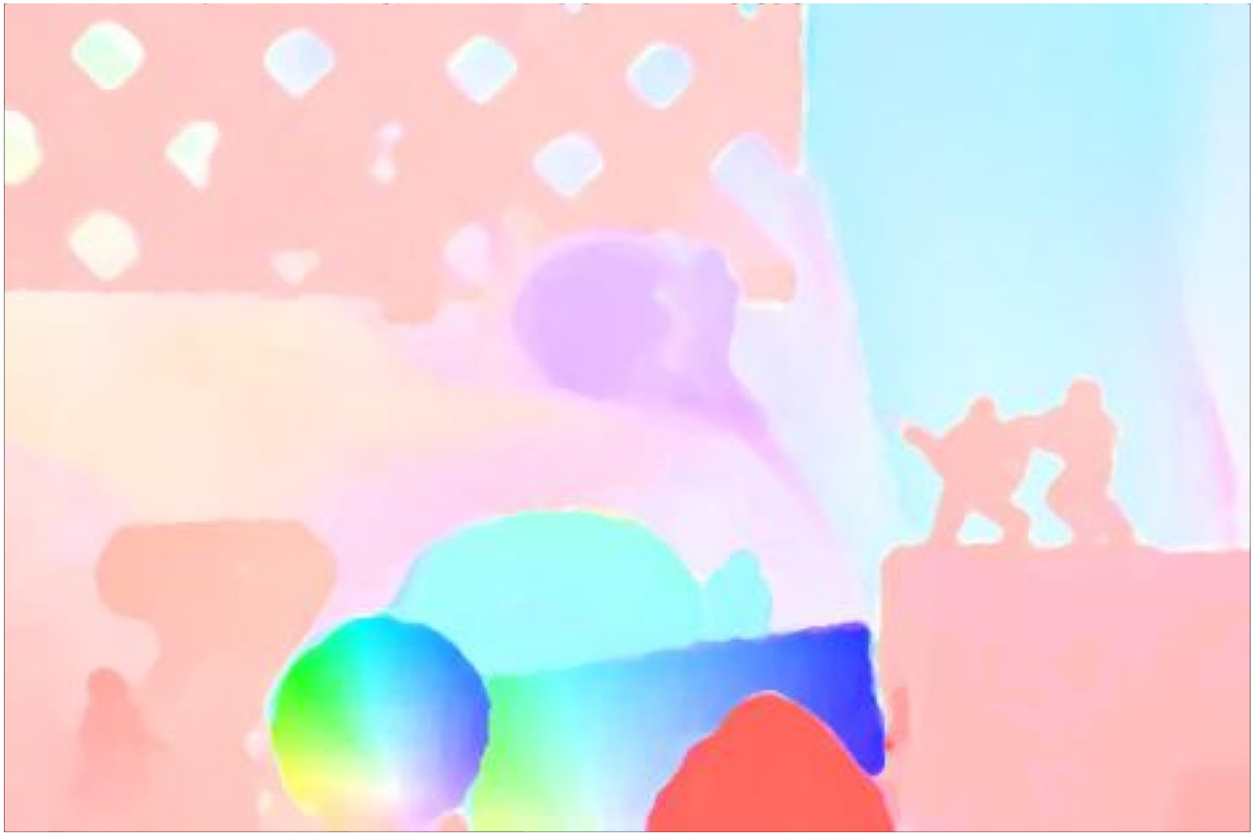


Horn & Schunck 1981 (HS)



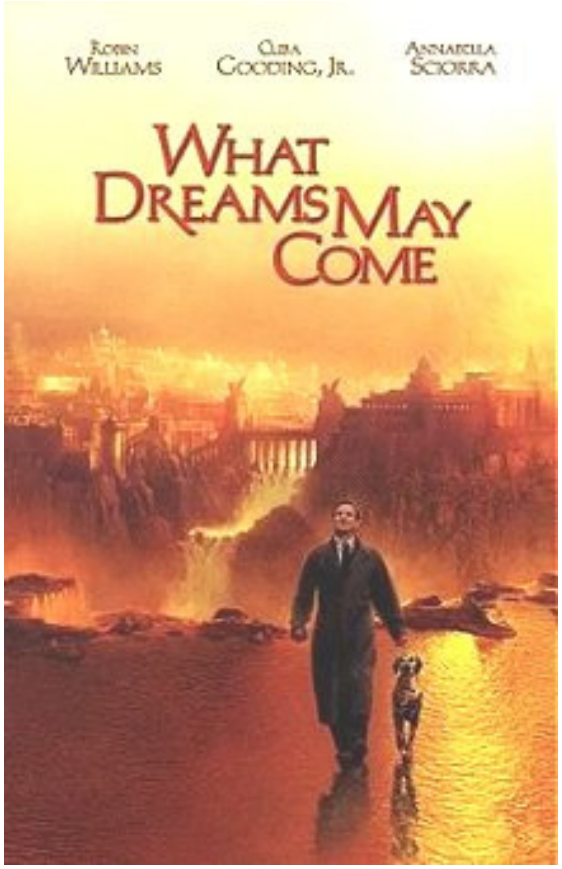
Black & Anandan 1996 (BA)

Today's best method



Improved derivatives, improved optimization, different robust function.

Applications of Optical Flow



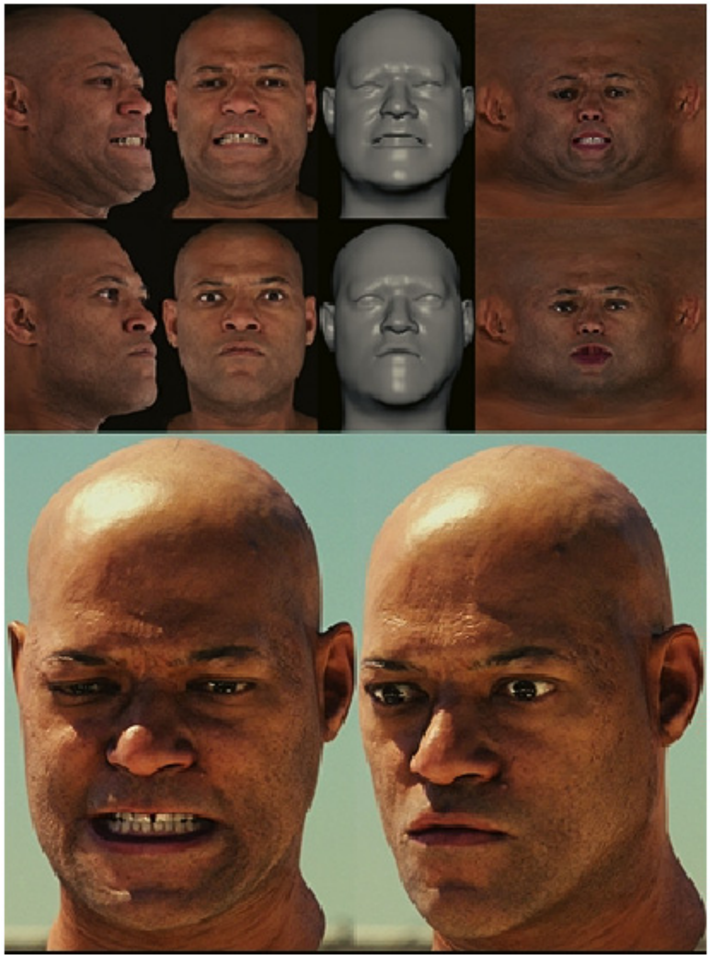
Impressionist effect.
Transfer motion of real world to a painting

Bullet Time

Use optical flow to compute correspondence between different camera views. Allows smooth interpolation between views.

Facial Animation

George Borshukov, Dan Piponi, Oystein Larsen, J.P.Lewis, Christina Tempelaar-Lietz
ESC Entertainment



Tracking in Images

How?

http://http.cs.berkeley.edu/~pm/RoadWatch/tracking.mpg

Tracking

Approach 1:

Detect an object (e.g. a face) in every frame independently.

Approach 2:

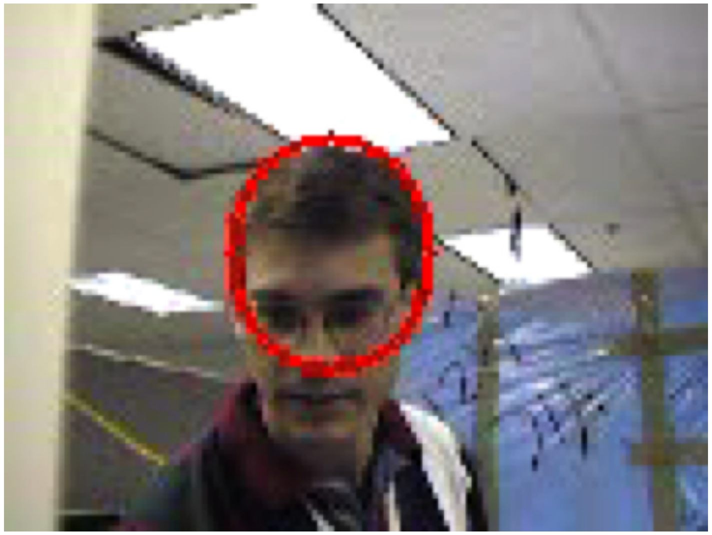
Use what you know about where the object was in the previous frame(s) to make predictions about the current frame and restrict the search.

Tracking vs Flow

Flow: track region from time t to time t+1, forget what you knew about the region at t and then track from t+1 to t+2. Updates the "model" completely at every time instant.

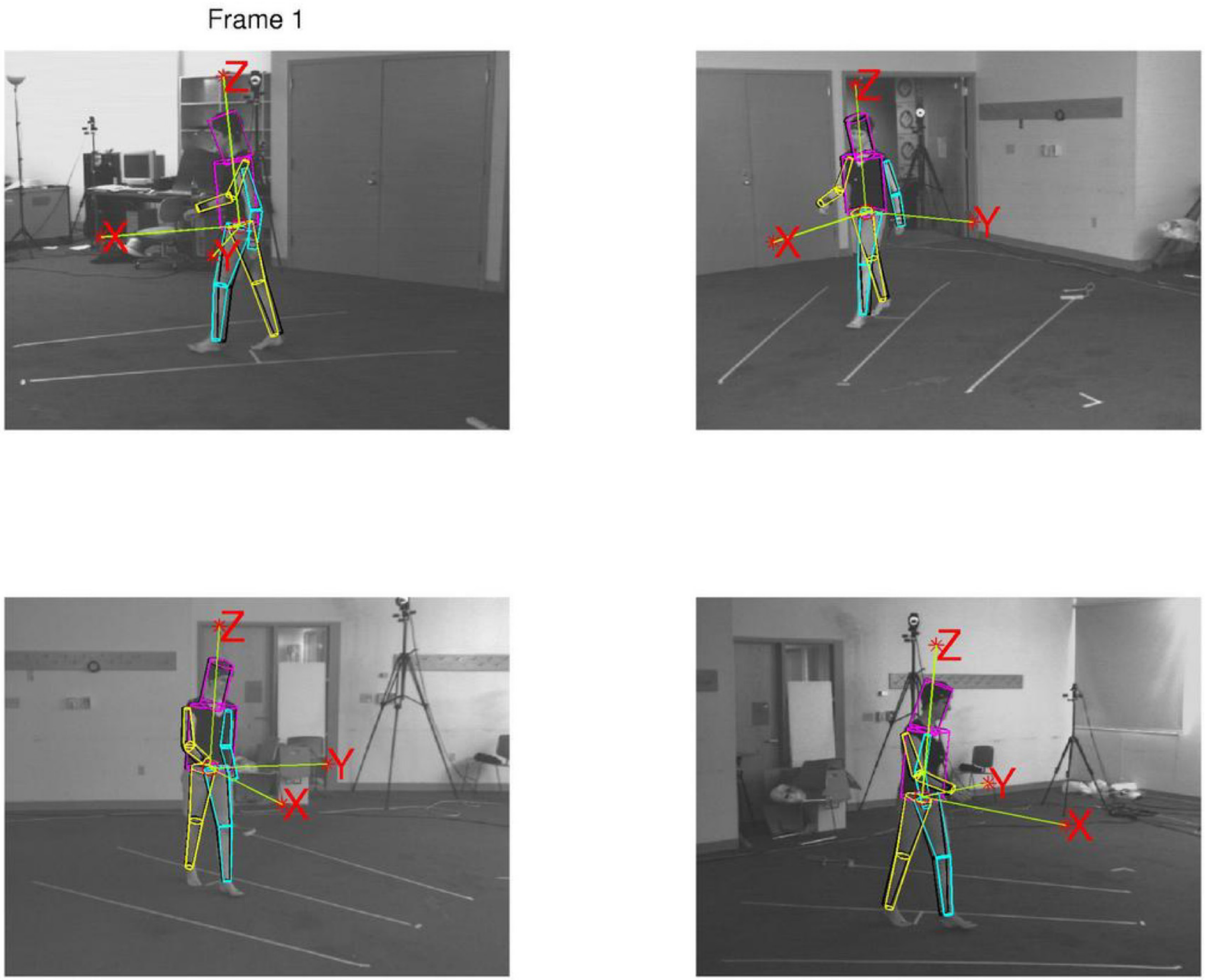
Tracking: build some model of what you want to track, if you know where it is at time t estimate its motion to t+1, repeat (possibly updating the model).

Face tracking



* Color histograms and image gradients along contour.

http://robotics.stanford.edu/~birch/headtracker/


Frame 1
Z
X
Y
Z
X
Y
Z
Y
X
Z
Y
X

What's Constant?

- Need something to be "constant" to track.
- Pixel brightness – optical flow, template tracking
  - Robustness extends this but only so far
- What else could we do?

WSL tracker

Jepson et al WSL tracker results

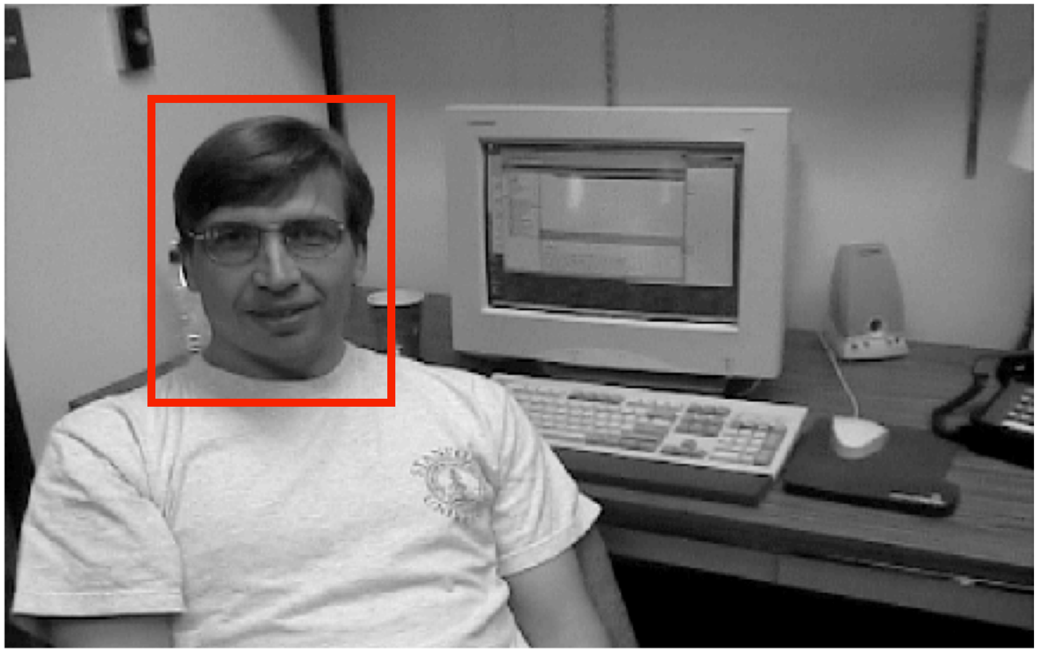
What's Constant?

- Need something to be "constant" to track.
- Pixel brightness – optical flow, template tracking
  - Robustness extends this but only so far
- What else could we do?

Subspace constancy – extend the notion of a template to a linear subspace (EigenTracking)

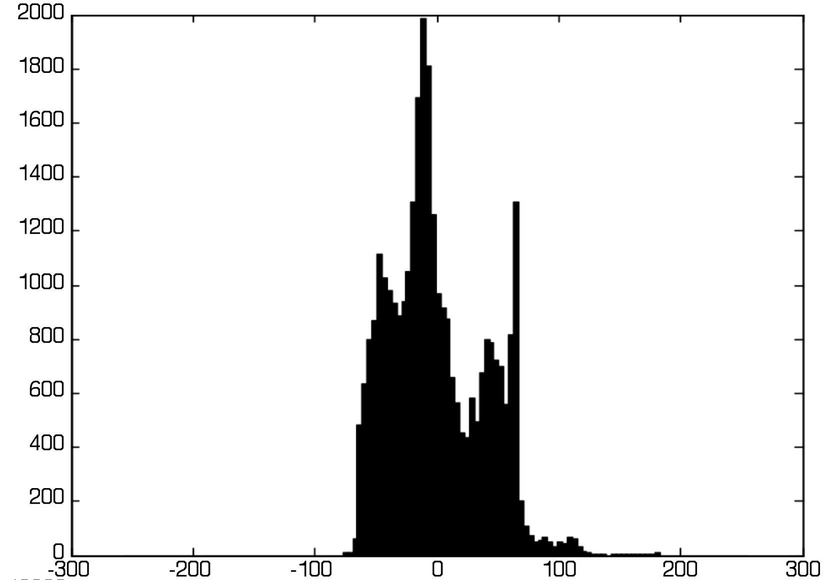
Statistical feature constancy – the distribution of filter responses remains constant.

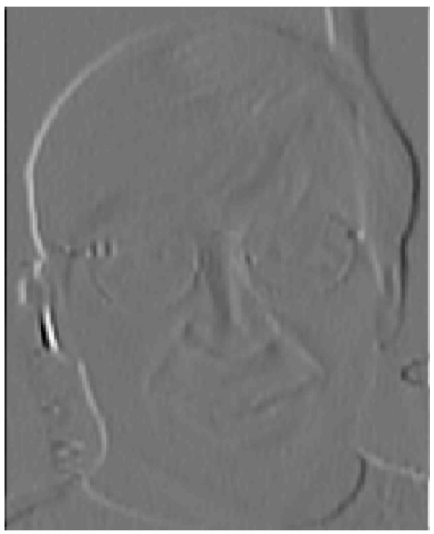
Homework 4

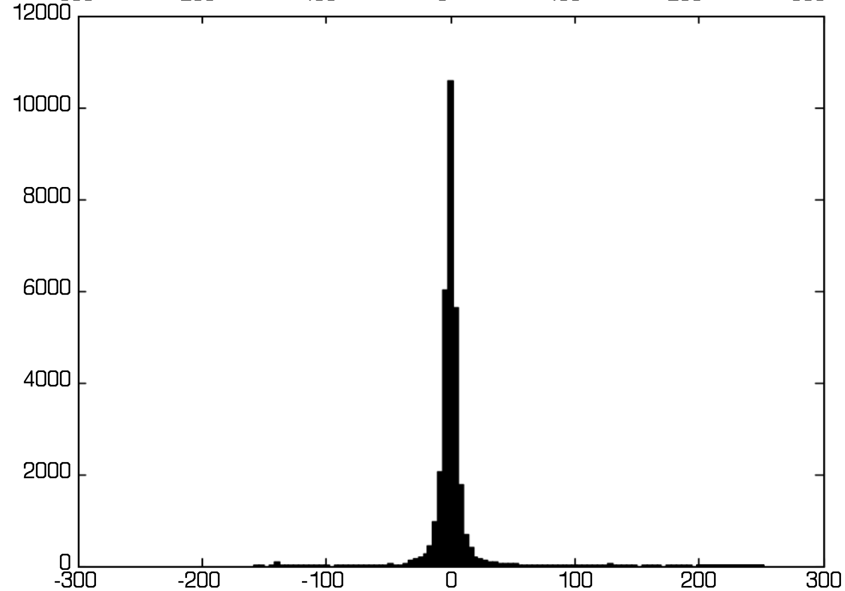


Characterize an image region by its statistics.
If the statistics differ from background, should enable tracking.

http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html


2000
1800
1600
1400
1200
1000
800
600
400
200
0
-300
-200
-100
0
100
200
300




12000
10000
8000
6000
4000
2000
0
-300
-200
-100
0
100
200
300

Histograms

Compute histograms of 1) pixel values; 2) x derivatives; 3) y derivatives.

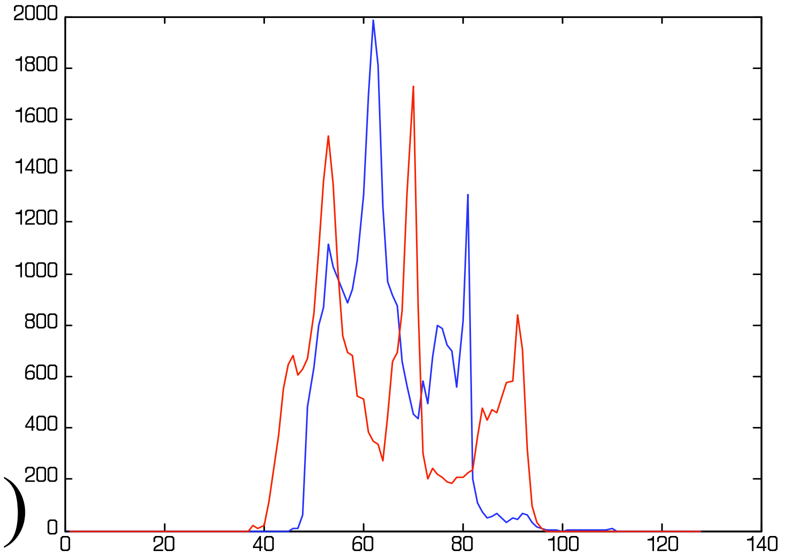
Be careful when using hist to define the range so that it is the same for the histograms you want to compare (ie hist(region(:),-x:y:x))

Comparing histograms

Bhattacharyya coefficient between two distributions:

$$bc(H1,H2)=\sum_{i=1}^{N}\sqrt{H1(i)H2(i)}$$

Histogram of the face pixels (blue) and another image region (red).

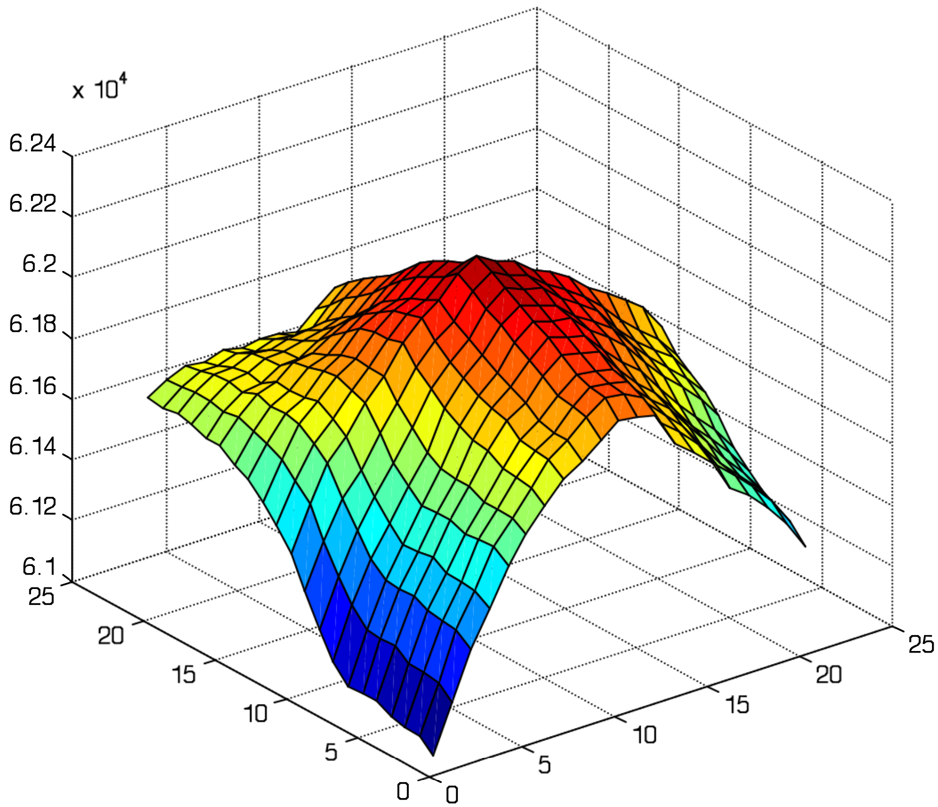


Distance measure:

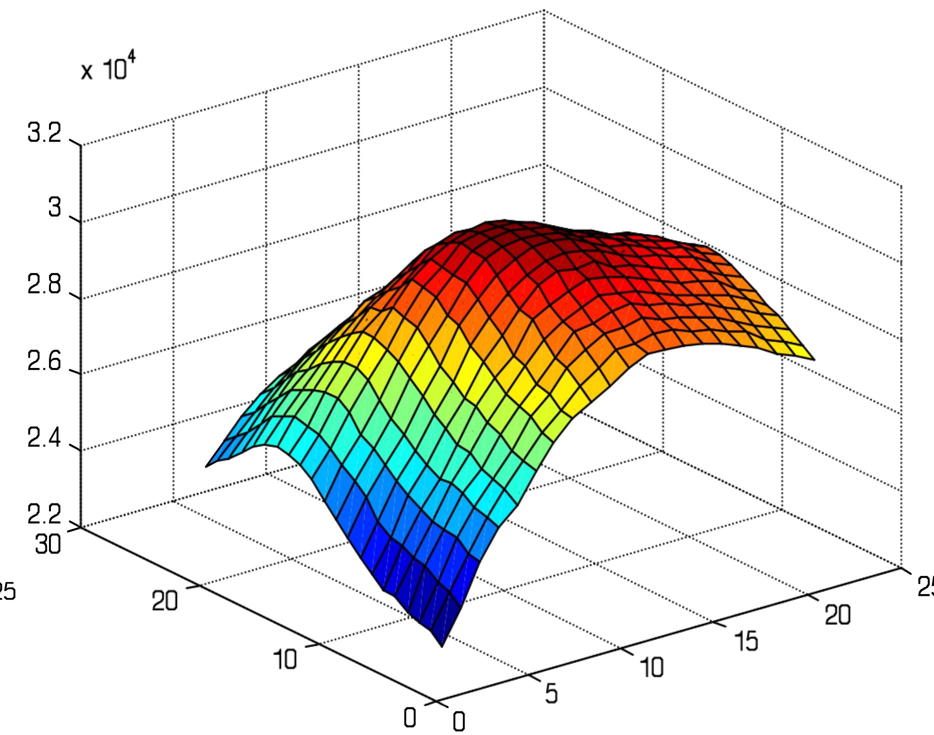
$$bd(H1,H2)=\sqrt{1-bc(H1,H2)}$$

$$or\ \ bd(H1,H2)=-\log(bc(H1,H2))$$

Bhattacharyya coefficient between image regions

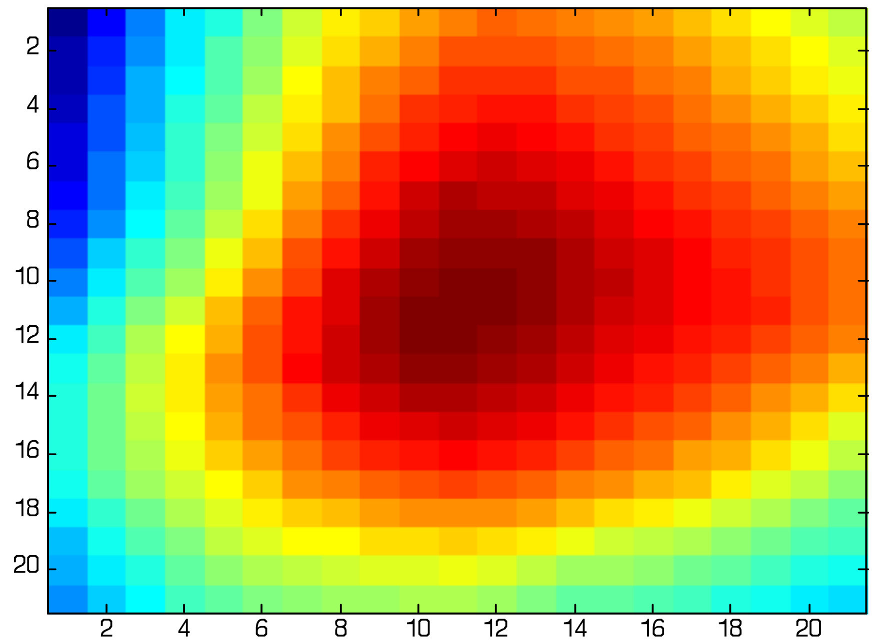


x derivatives



Pixel values

Bhattacharyya Coefficient



Combining pixel and derivative histograms.

Un-normalized Likelihood

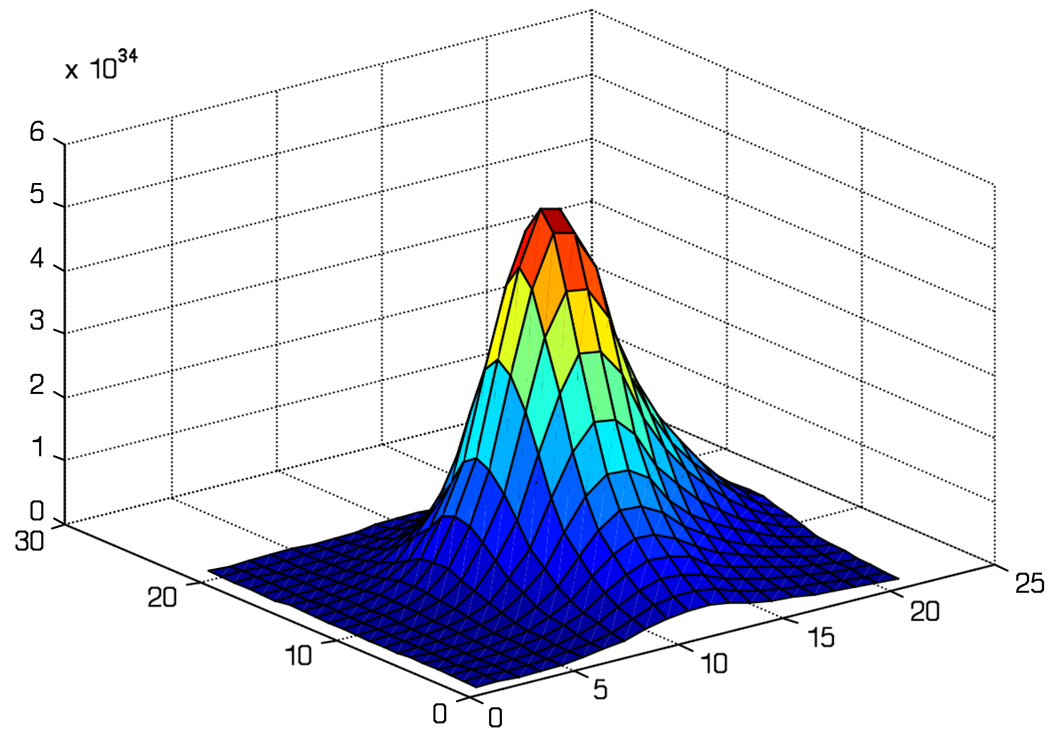
Exponentiate the Bhattacharyya coefficient.

surf(exp(bc))

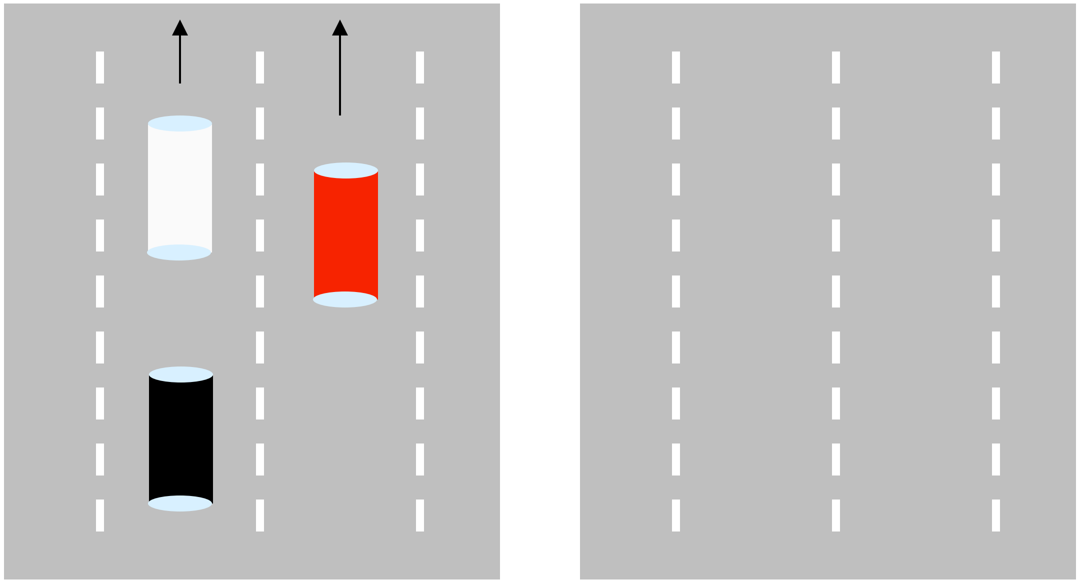
or

surf(exp(bc^k))

0<k<=2 (e.g.)



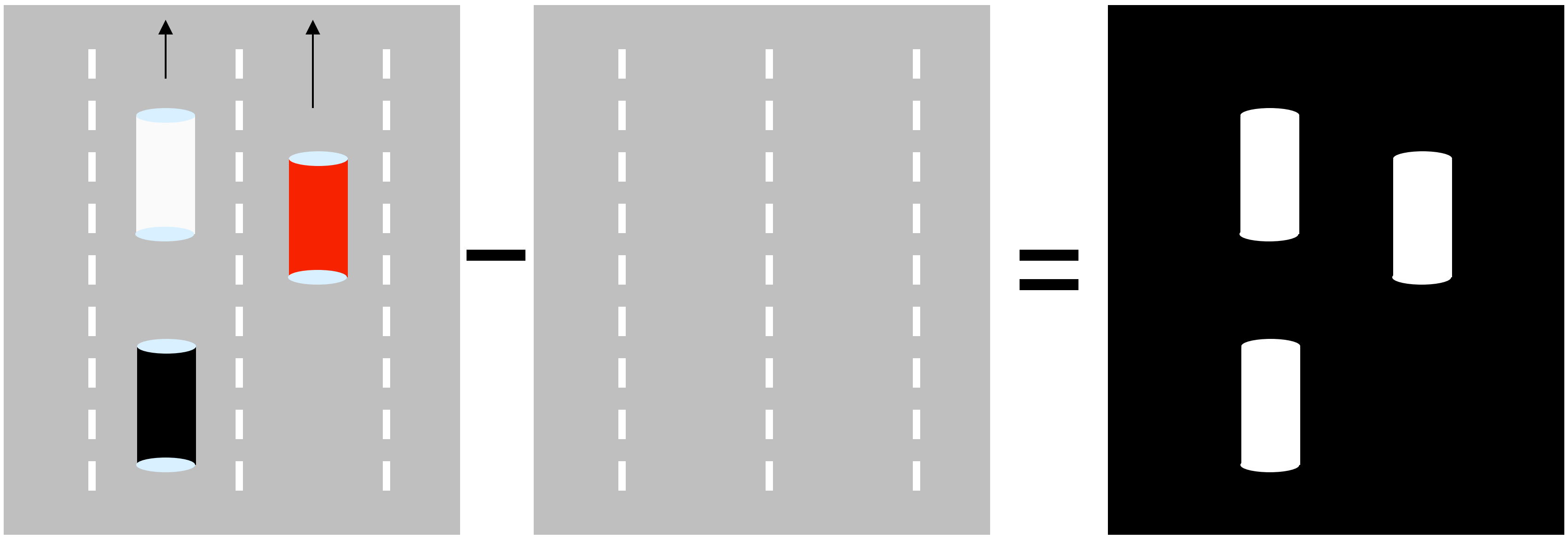
Mathematical Formulation



Goal: estimate car positions at each time instant

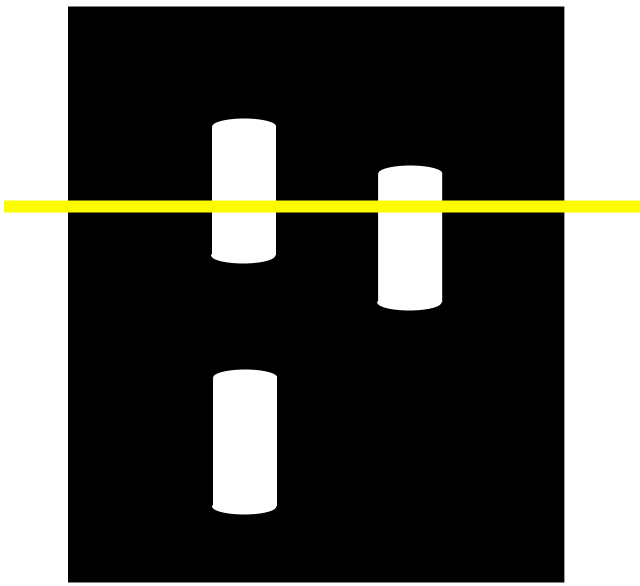
Observations: image sequences and known background

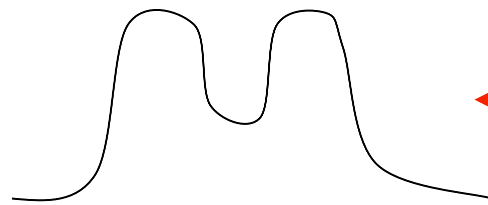
Mathematical Formulation



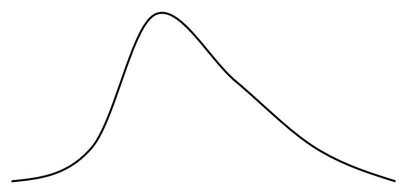
Define image likelihood: p(fg | car=(x,y))

Mathematical Formulation

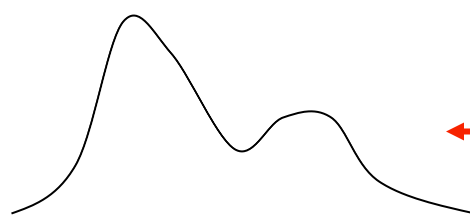




Likelihood:
noisy observation
p(fg | car=(x,y))



Prior:
p(car=(x,y))



Posterior:
Bayesian update
p(car=(x,y)|fg)

system states: car positions

observations: images

Notation

- $\mathbf{x}_k \in \mathbf{R}^d$: internal state at k^{th} frame (hidden random variable, e.g. position of the object in the image).

  $\mathbf{X}_k = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]^T$: history up to time step k

- $\mathbf{z}_k \in \mathbf{R}^c$: measurement at k^{th} frame (observable random variable, e.g. the given image).

  $\mathbf{Z}_k = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k]^T$:
  history up to time step k

Goal

Estimating the posterior probability $p(\mathbf{x}_k \mid \mathbf{Z}_k)$

How ???

One idea: recursion $p(\mathbf{x}_{k-1} \mid \mathbf{Z}_{k-1}) \Rightarrow p(\mathbf{x}_k \mid \mathbf{Z}_k)$

- How to realize the recursion ?
- What assumptions are necessary ?