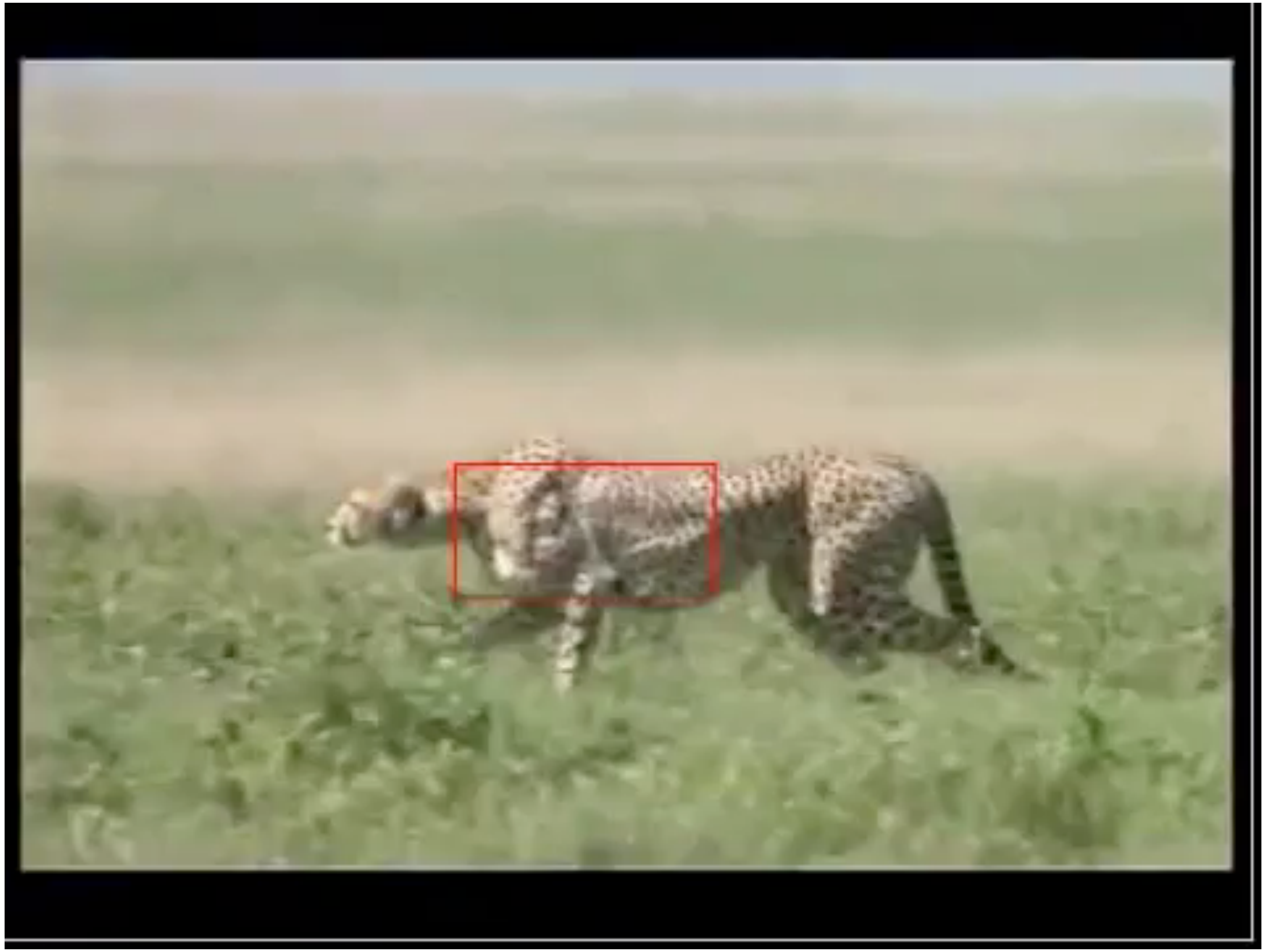
Motion, Tracking & Optical Flow II

Gary Overett (Slides adapted from CMU 16-720 2014)
Szeliski Chapter 8

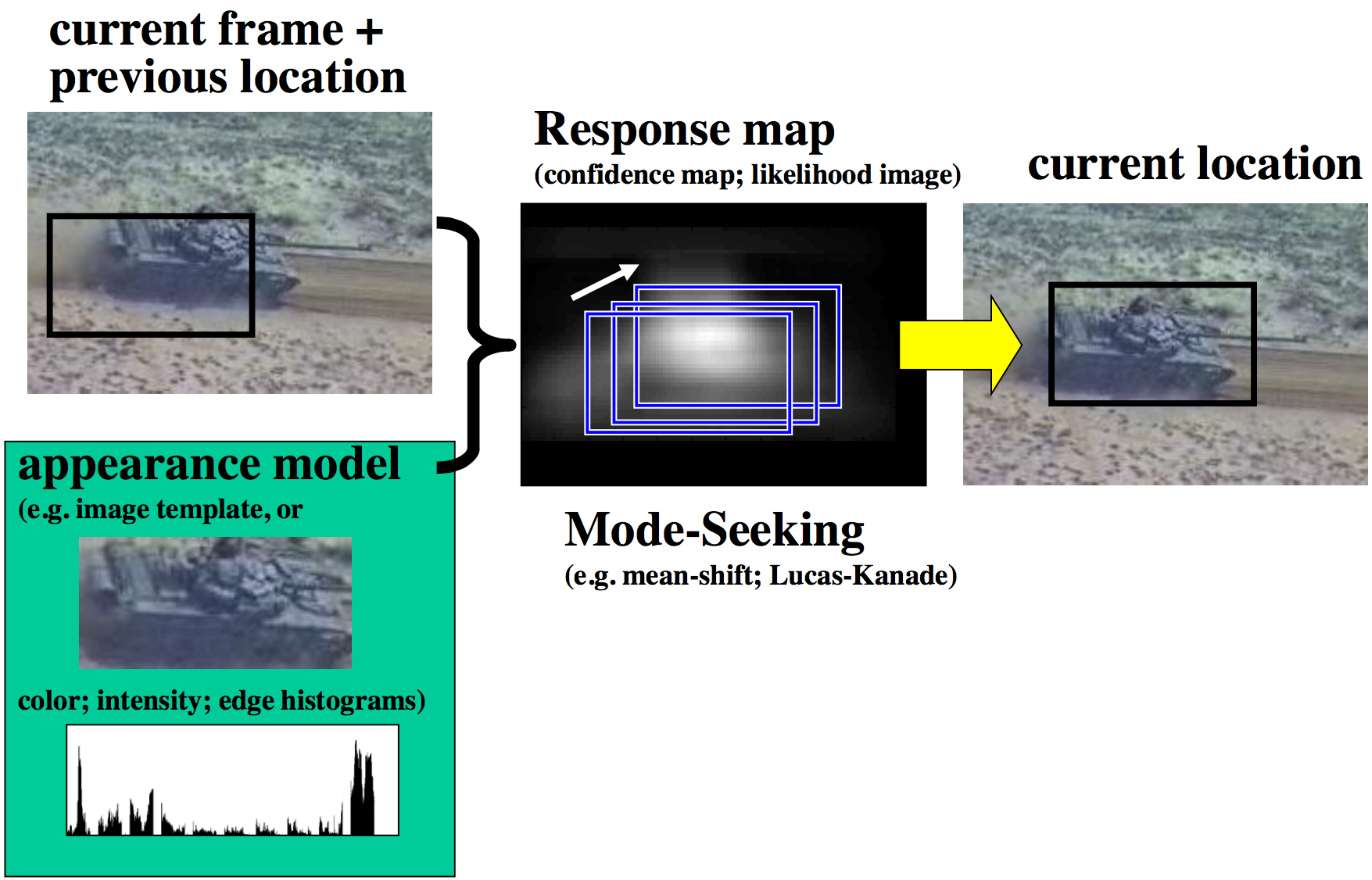
Summary

- Using Gradients to Estimate Motion
- Ambiguities
- Constant Motion Assumption
- Affine Flow Assumption
- Planar Assumption
- Tracking Using Mean Shift
- Motion Segmentation
- Background Detection

Tracking Using Mean Shift



Tracking as "Hill Climbing"

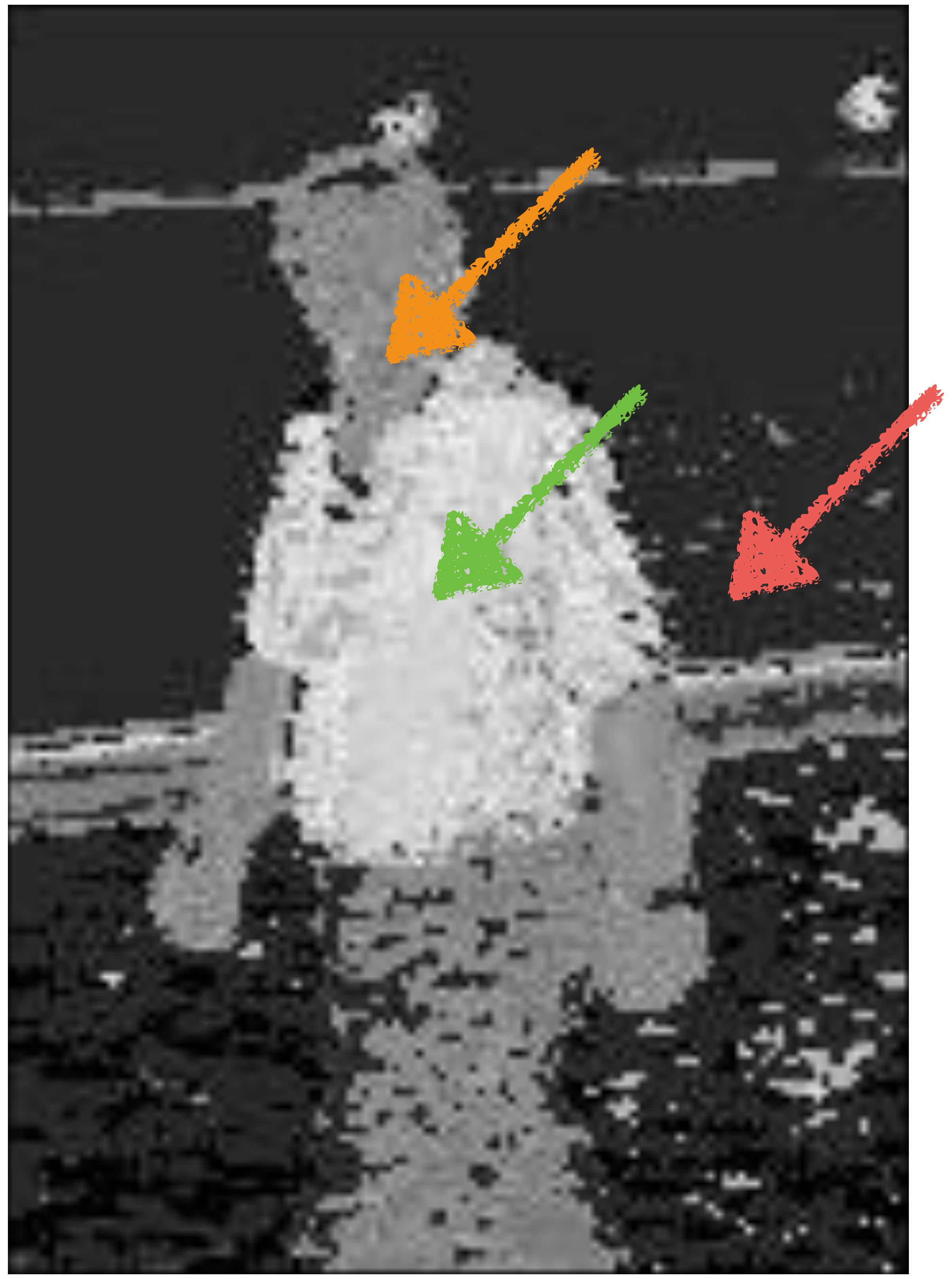


Mean-Shift Tracking

Create a response map with pixels weighted by "likelihood" that they belong to the object being tracked. Perform mean- shift algorithm on pixels (weighted points) in that image.

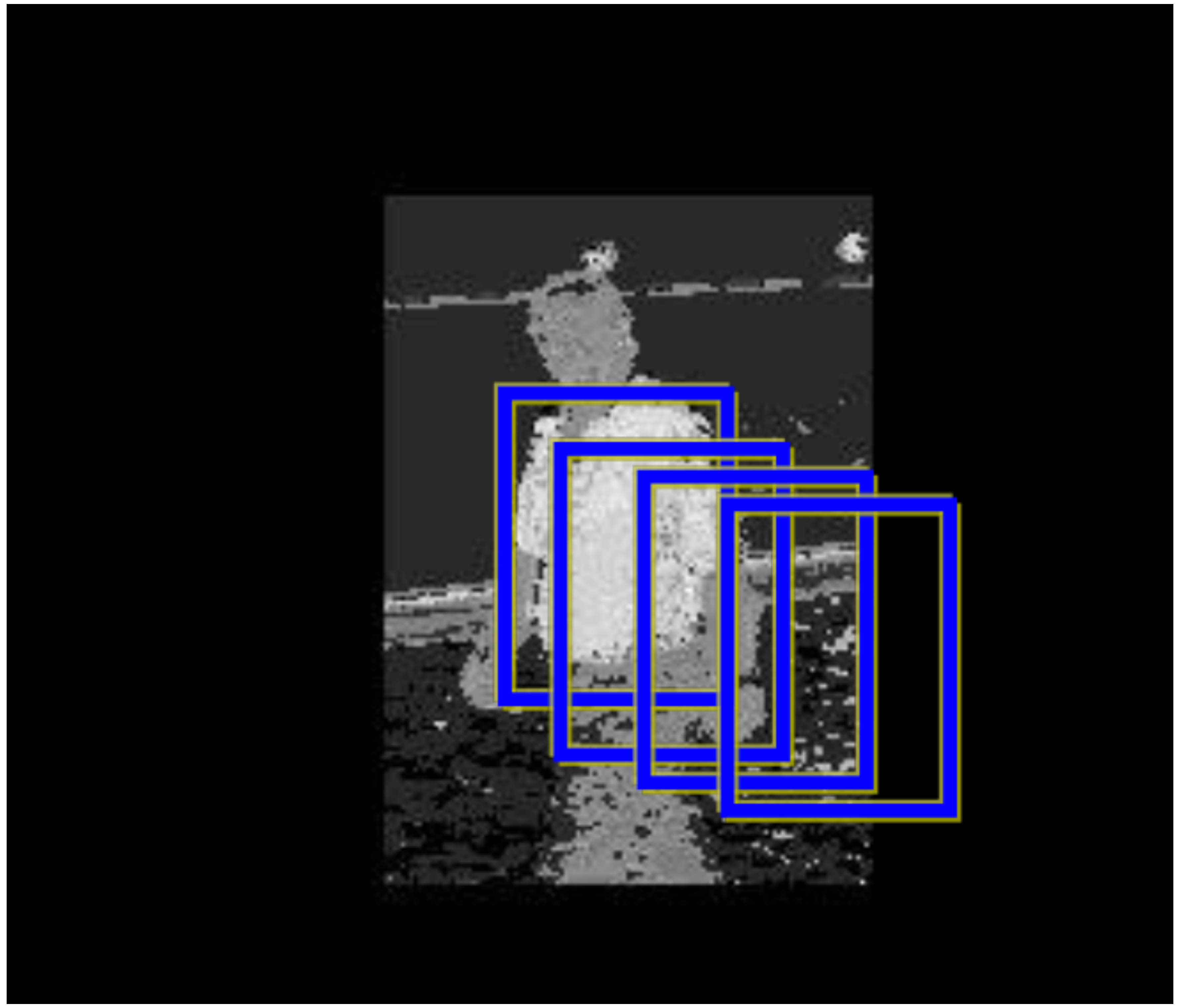
Computation of likelihood can be based on

- color
- texture
- shape (boundary)
- predicted location
- classifier outputs



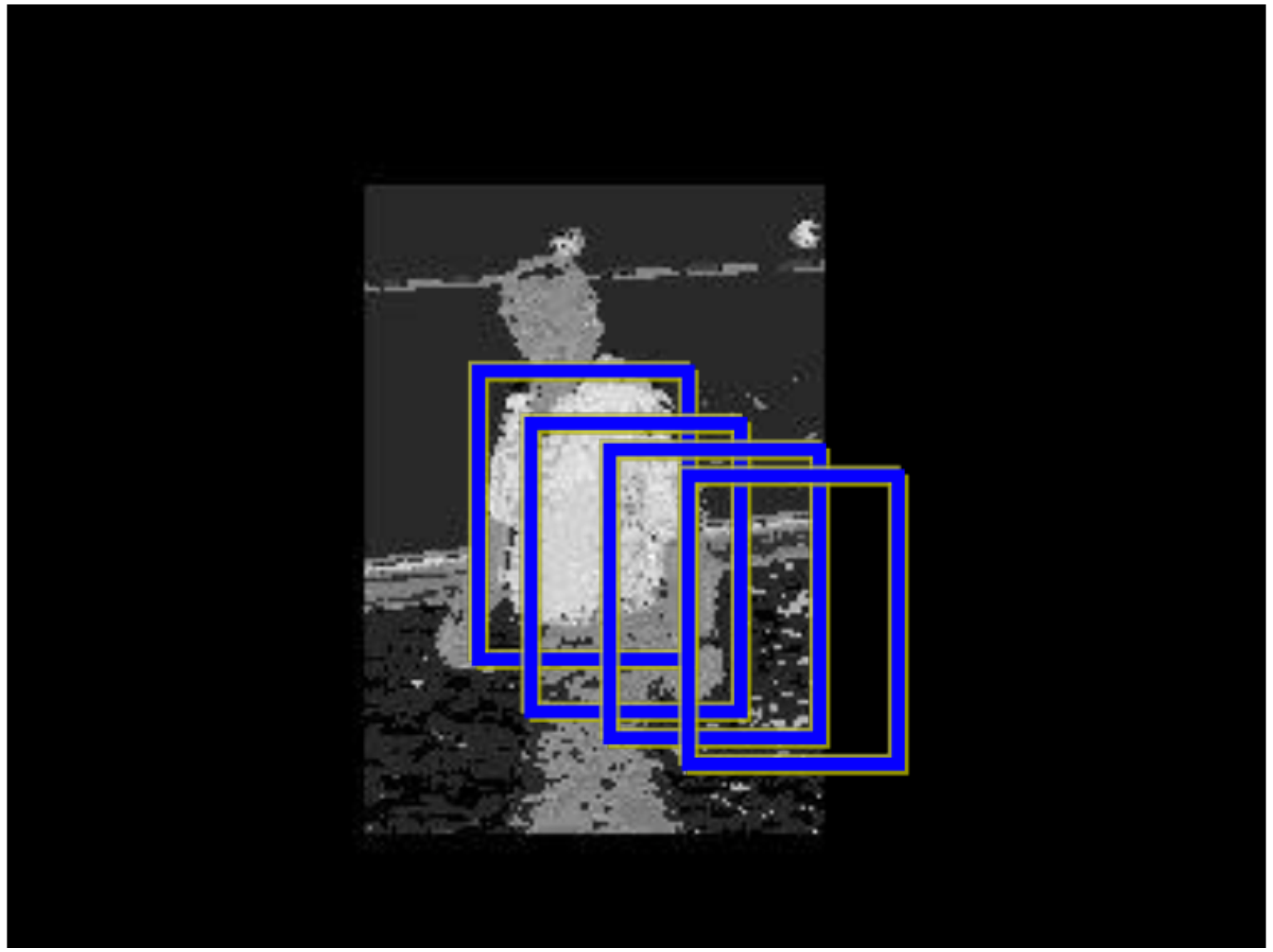
Mean-Shift on Weight Images

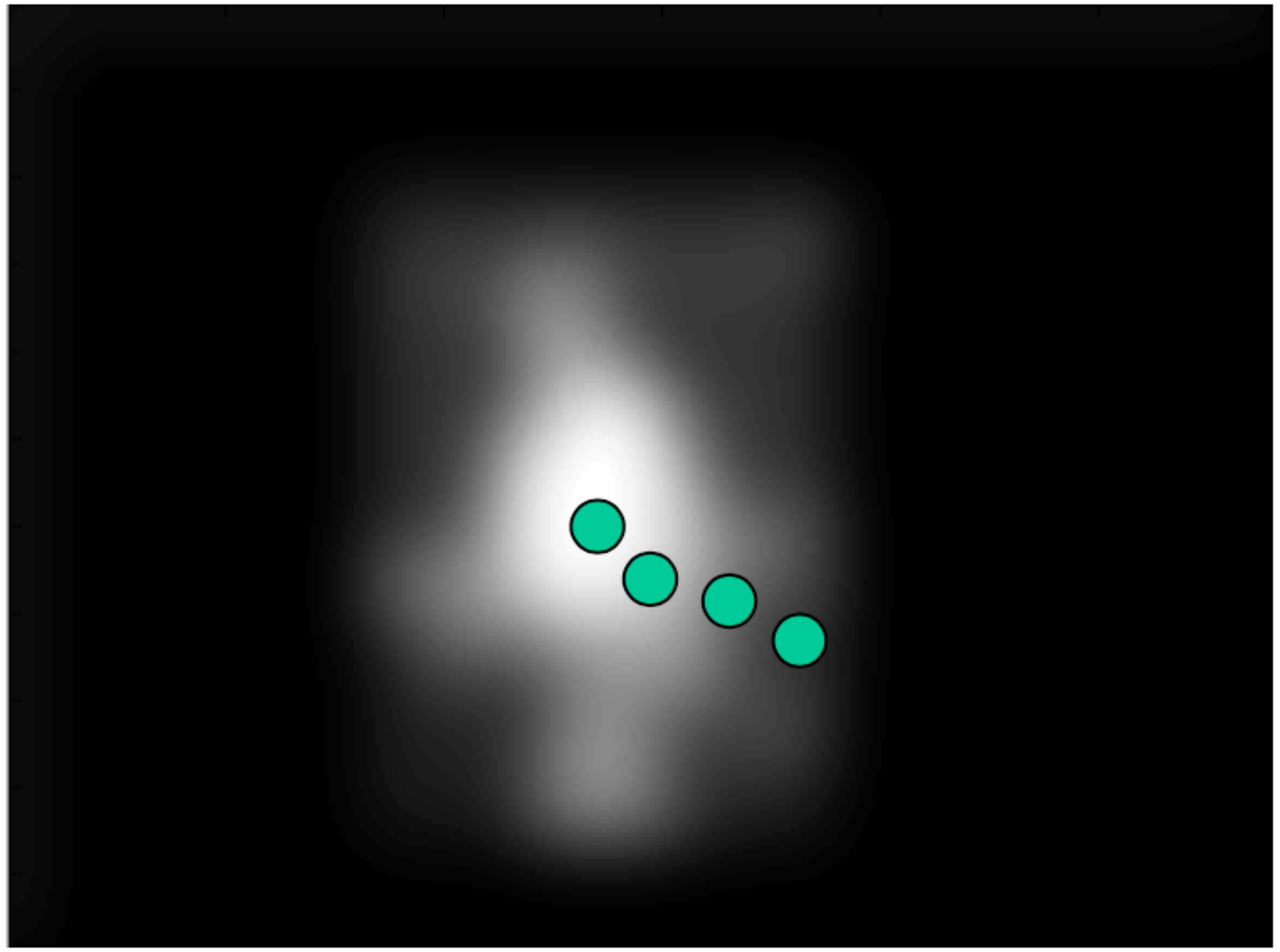
- The pixels form a uniform grid of data points, each with a weight (pixel value). Perform standard mean-shift algorithm using this weighted set of points.



Nice Property

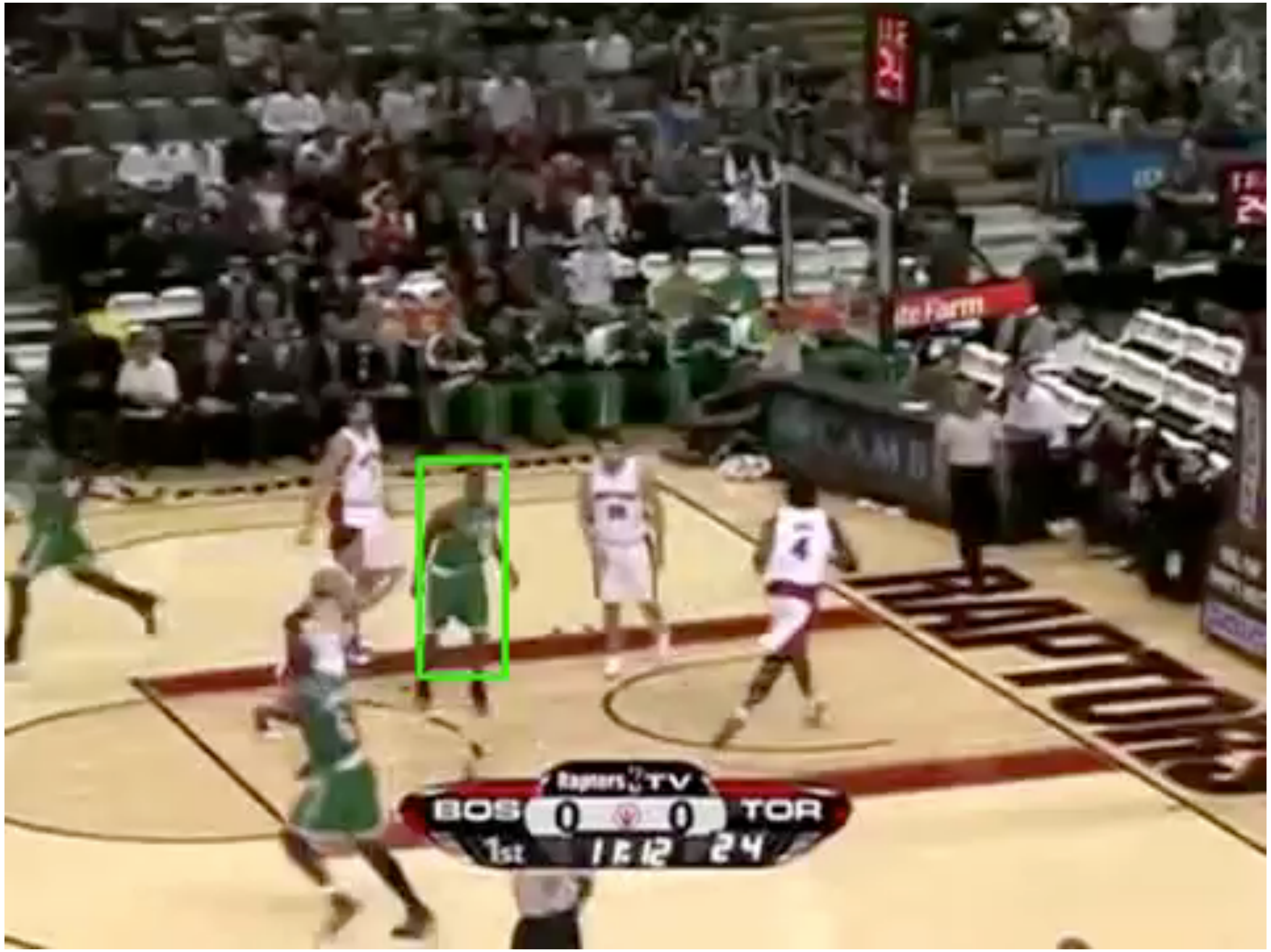
- Running mean-shift with kernel K on weight image w is equivalent to performing gradient ascent in a (virtual) image formed by convolving w.



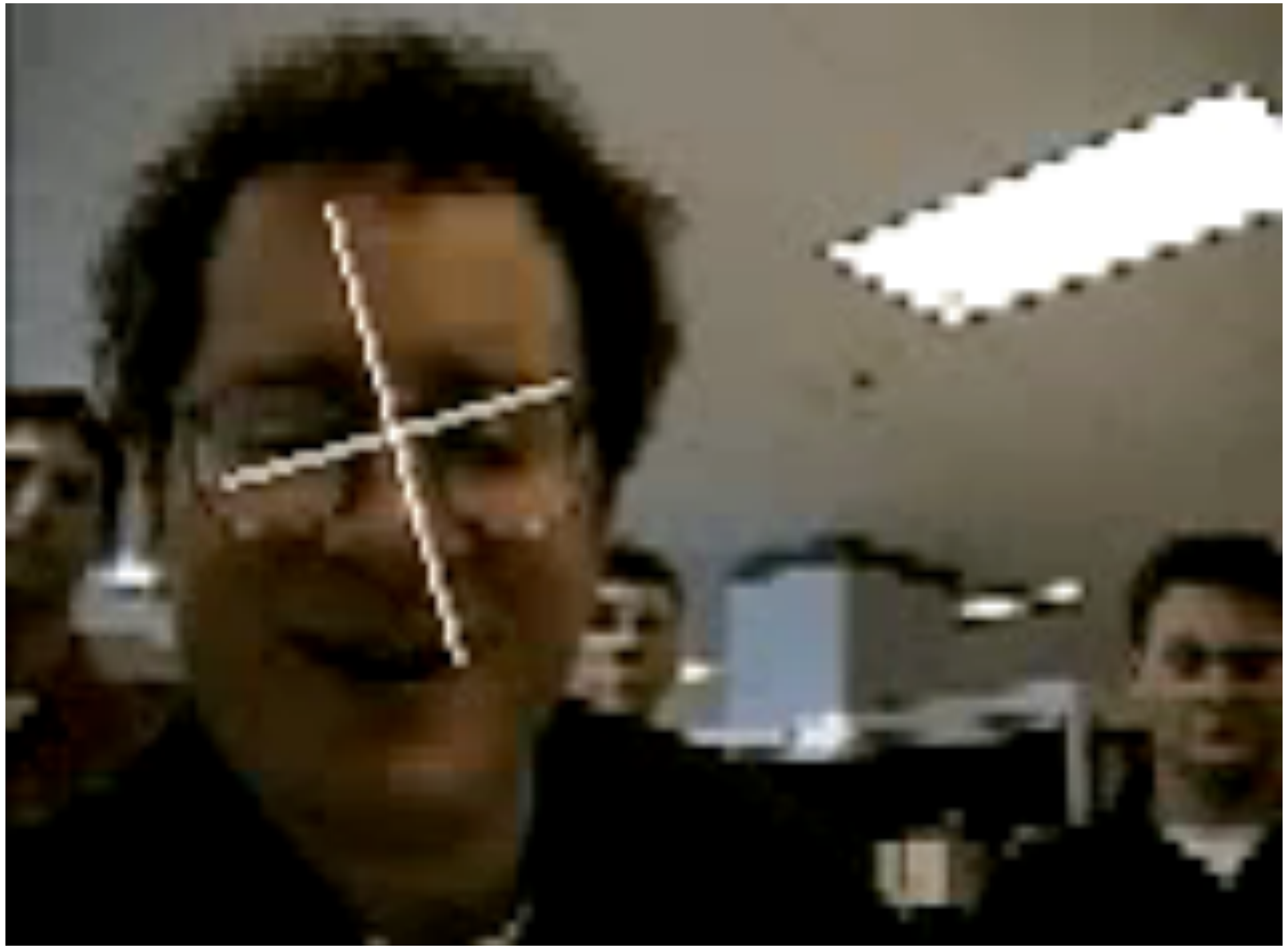


- The algorithm is performing hill-climbing on an implicit density function.

Mean-Shift Tracking Example



Mean-Shift Tracking Example

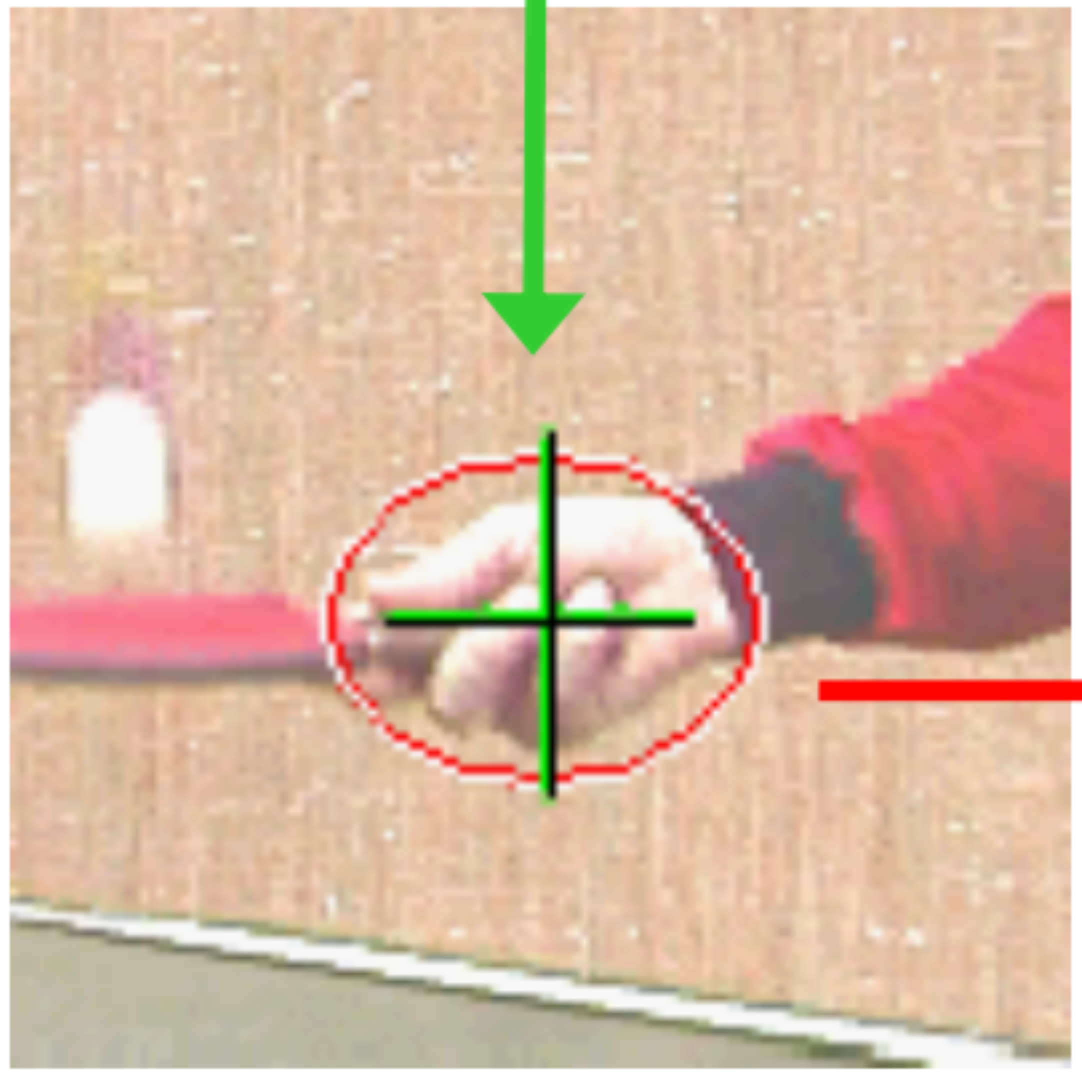


Using Mean-Shift on Color Models

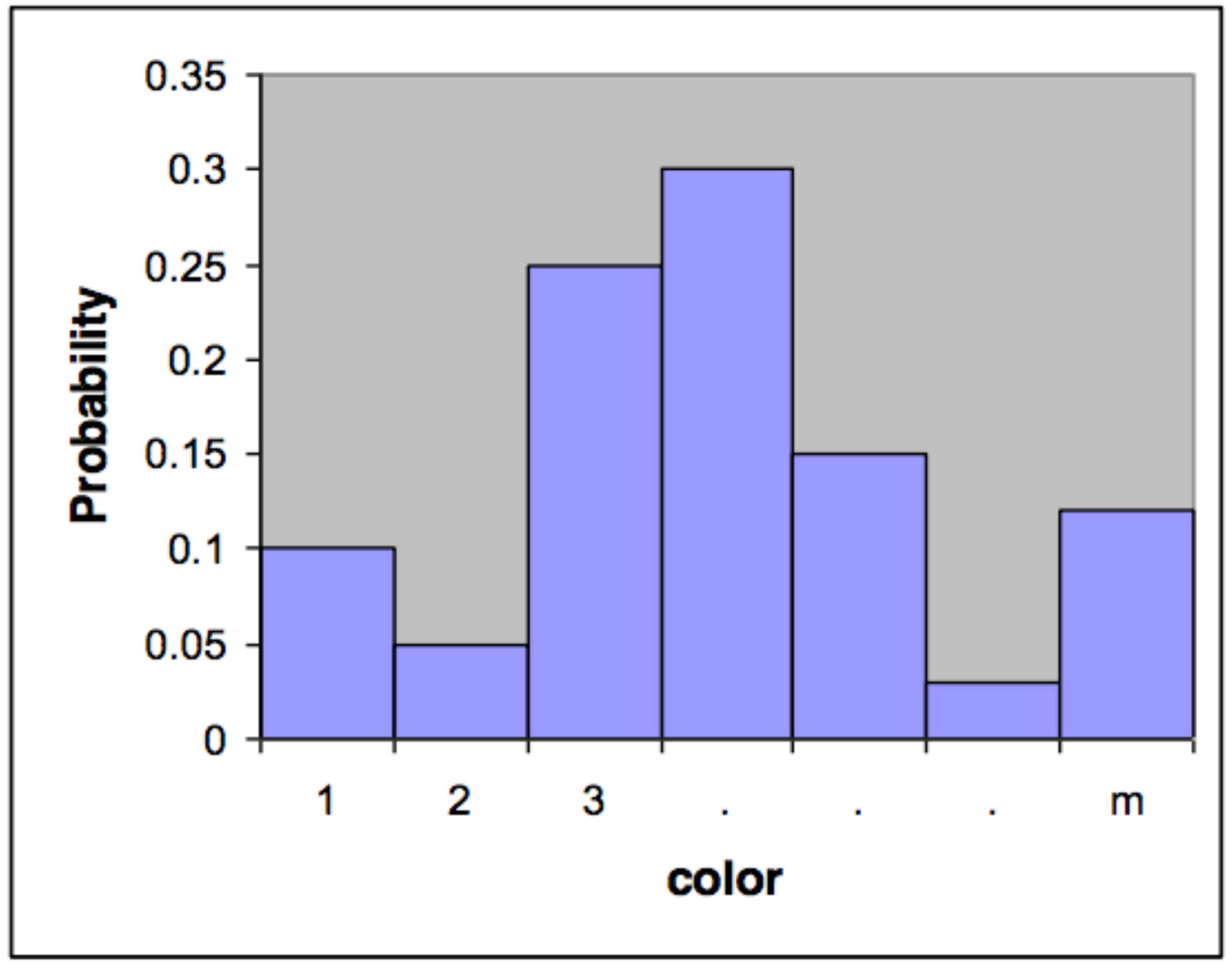
Two approaches:

- 1) Create a color “likelihood” image, with pixels weighted by similarity to the desired color (best for unicolored objects)
- 2) Represent color distribution with a histogram. Use mean-shift to find region that has most similar distribution of colors.

Choose a reference target model → Choose a feature space → Represent the model by its PDF in the feature space

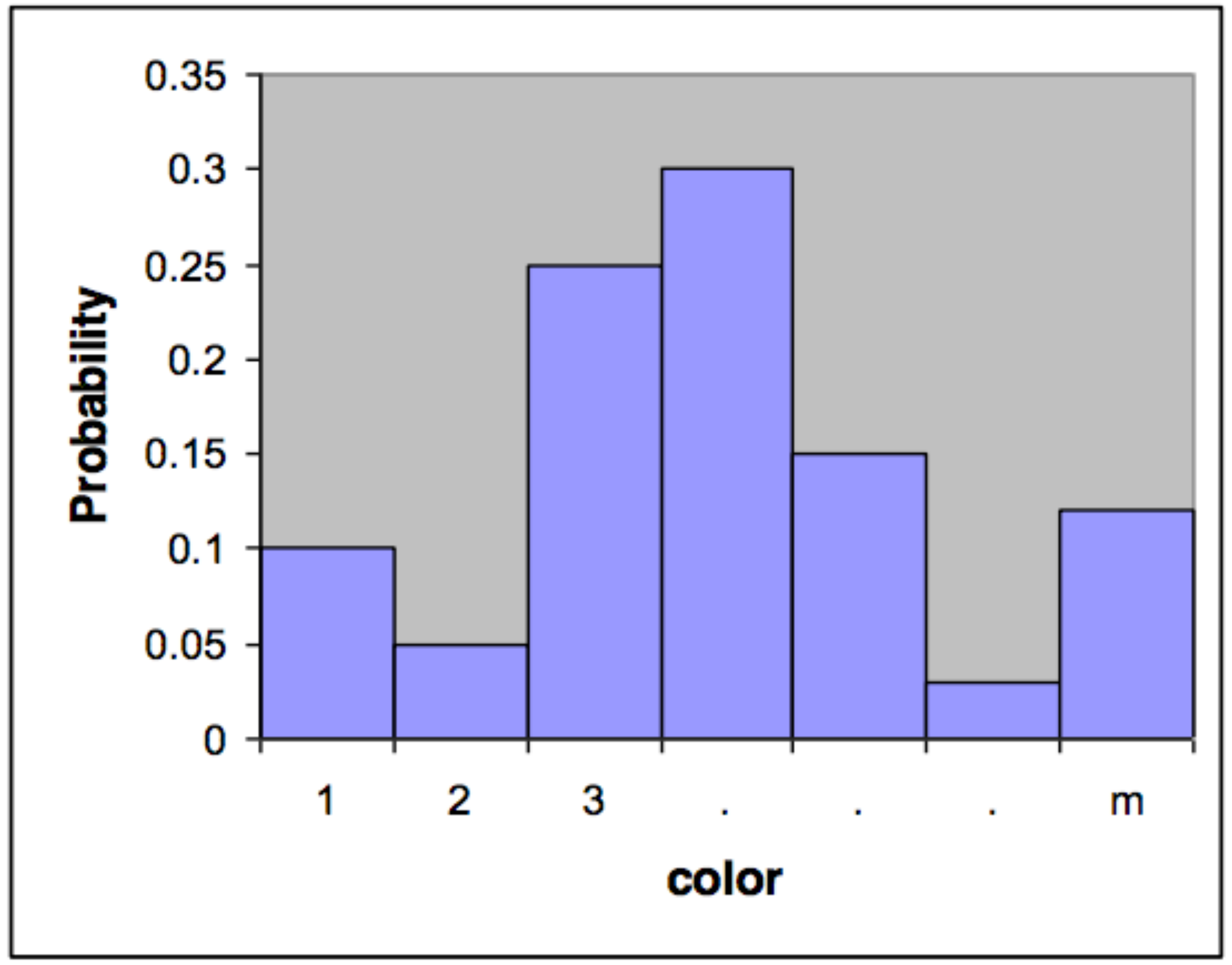


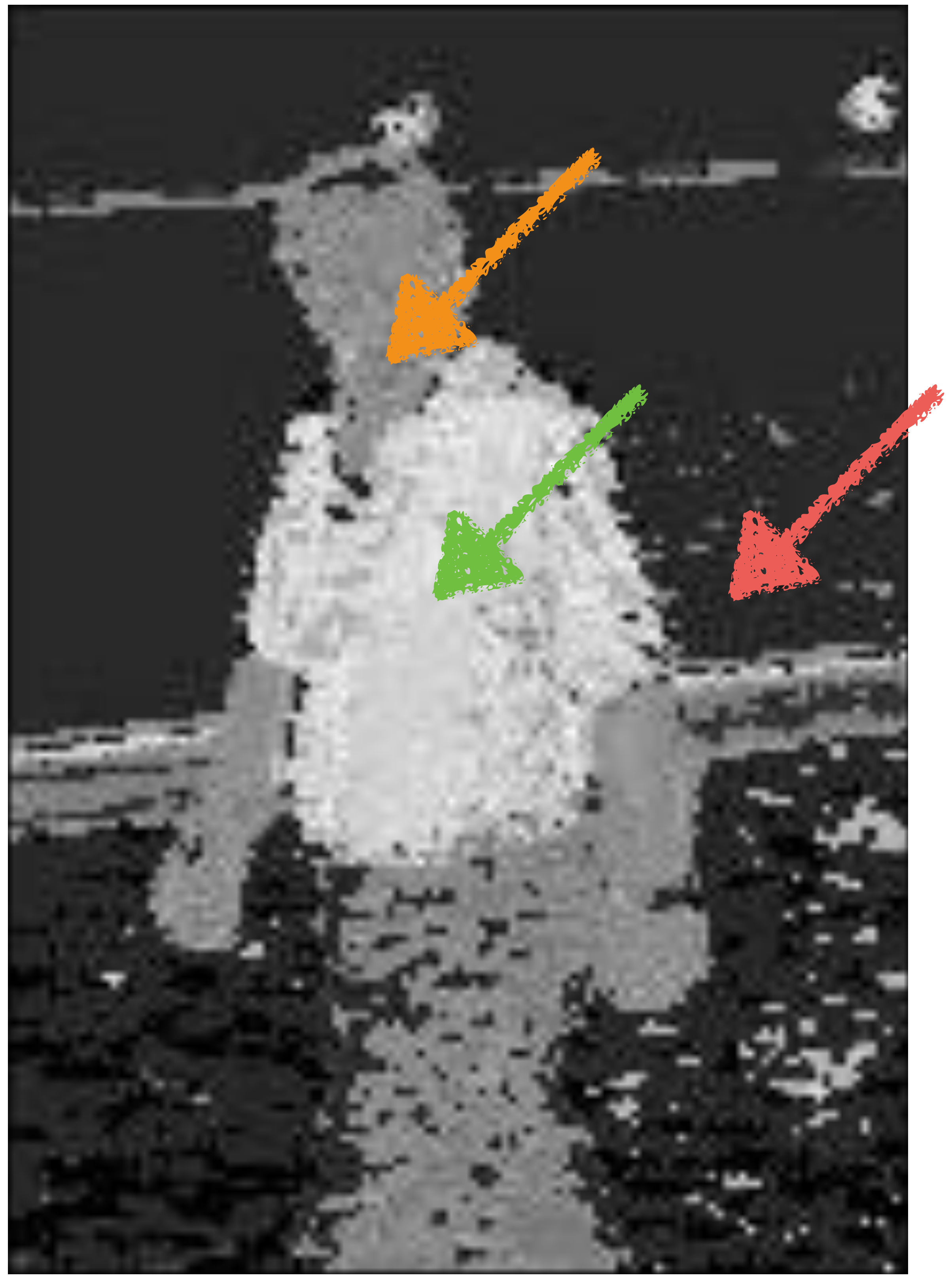
Quantized Color Space



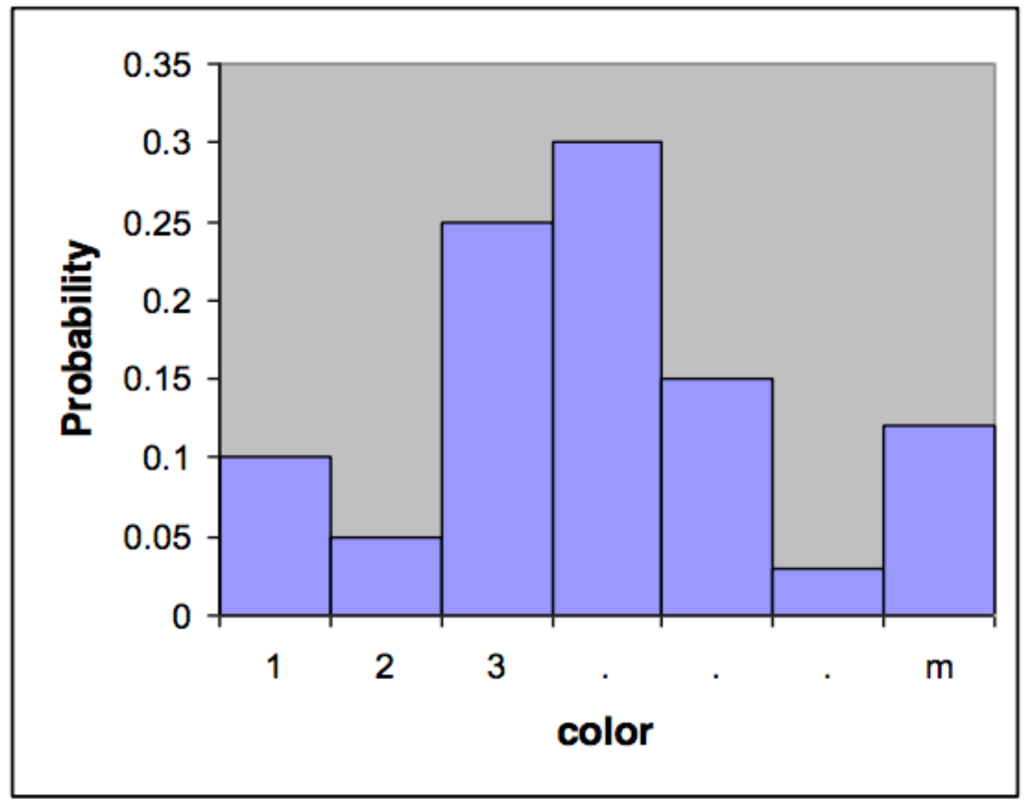
Kernel Based Object Tracking, by Comaniniu, Ramesh, Meer

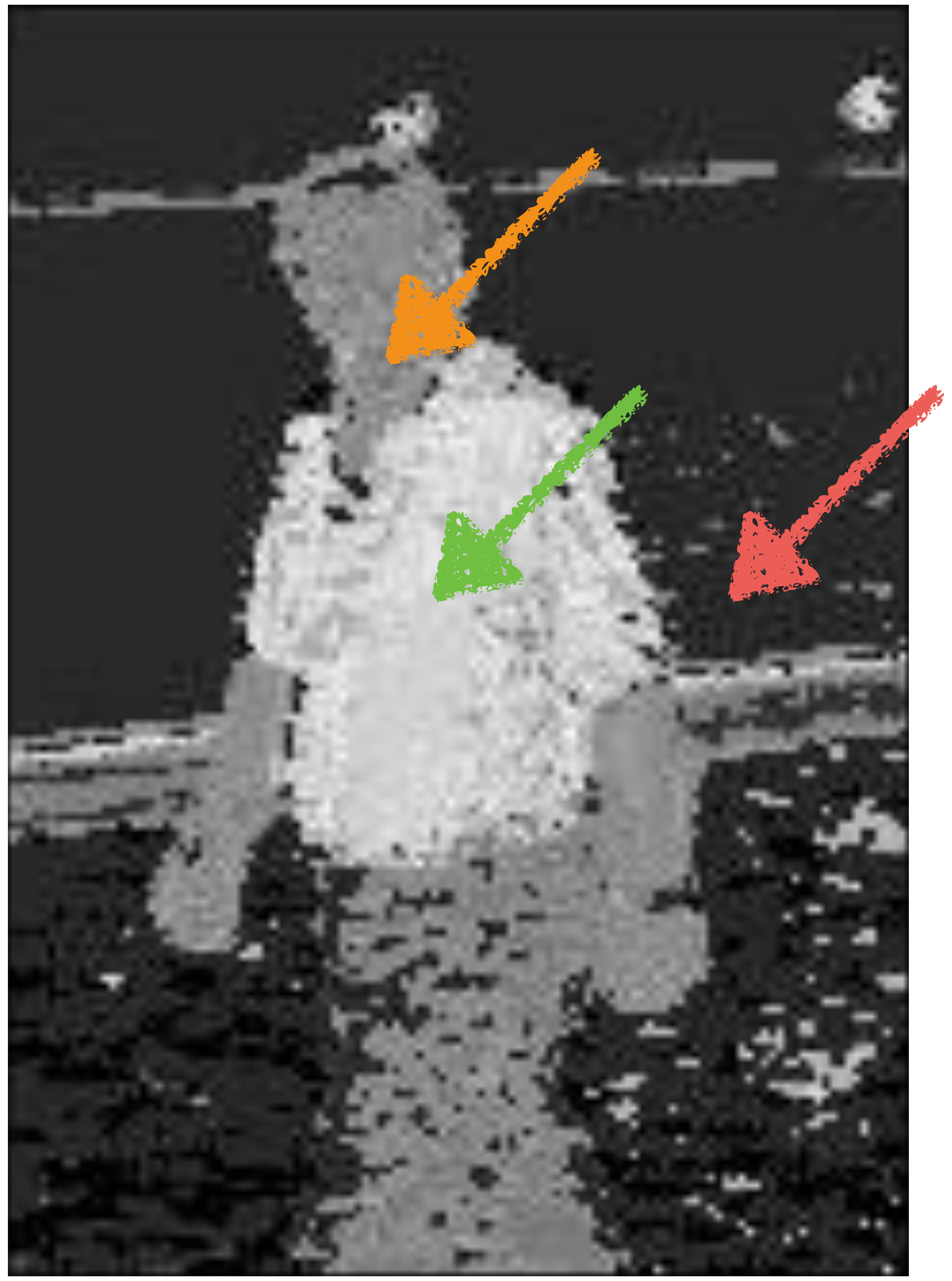
1) Color Likelihood Image



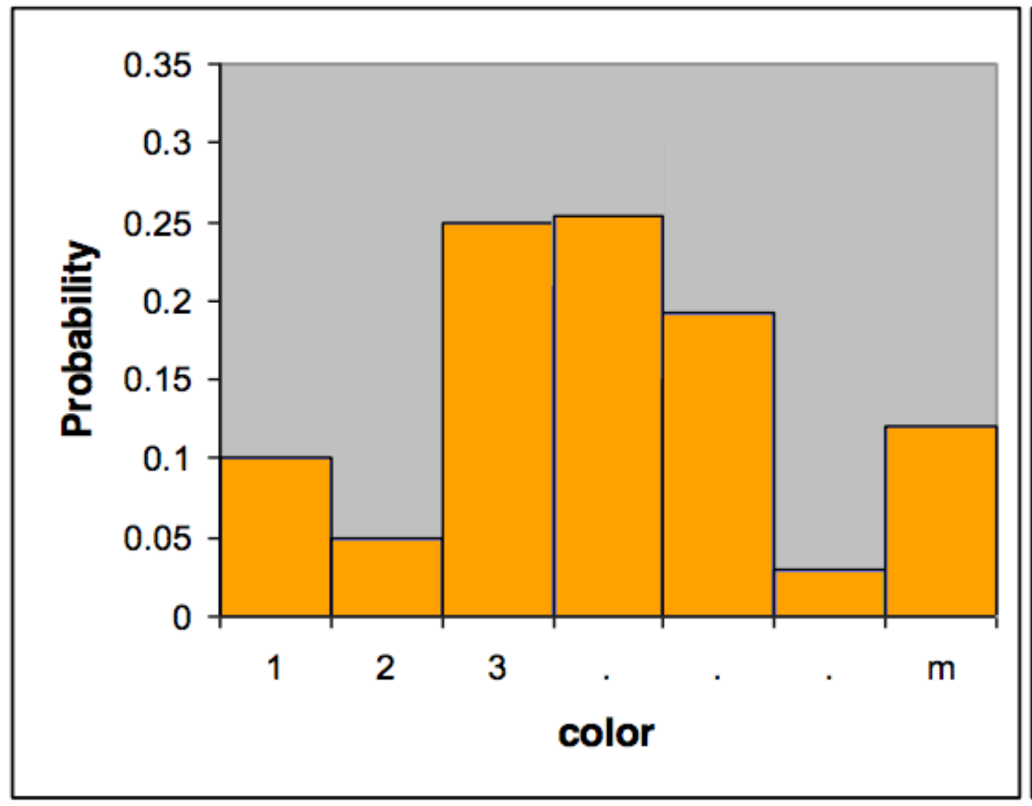


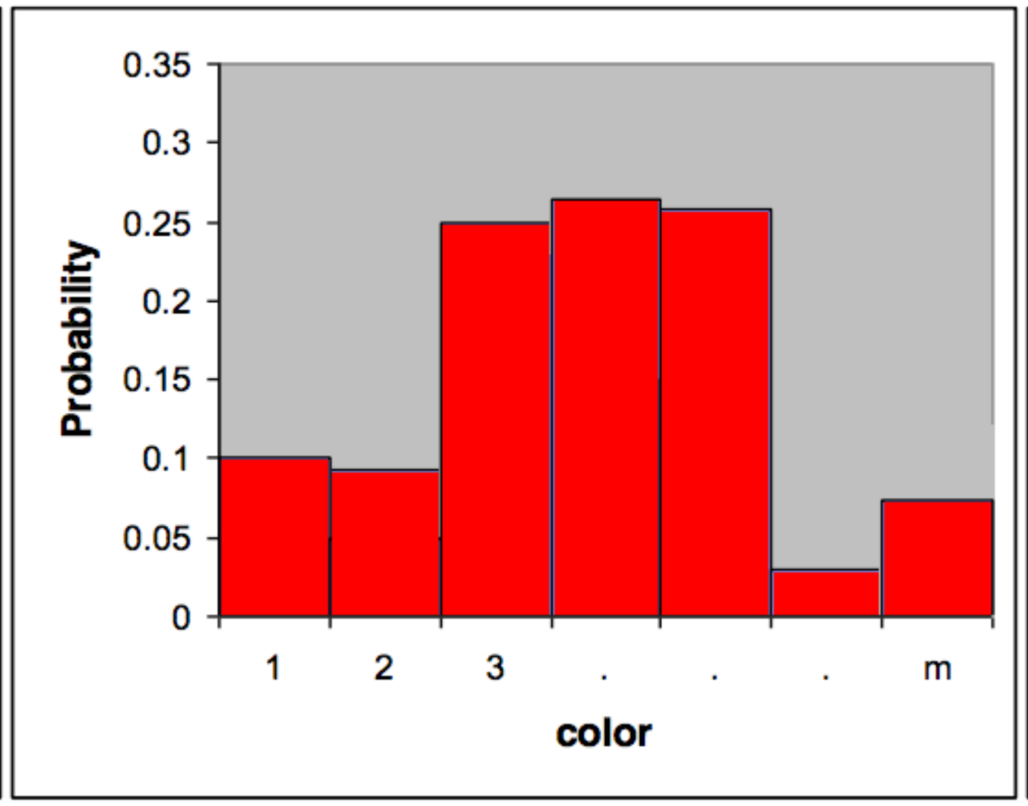
2) Color Distribution + Distance Metric

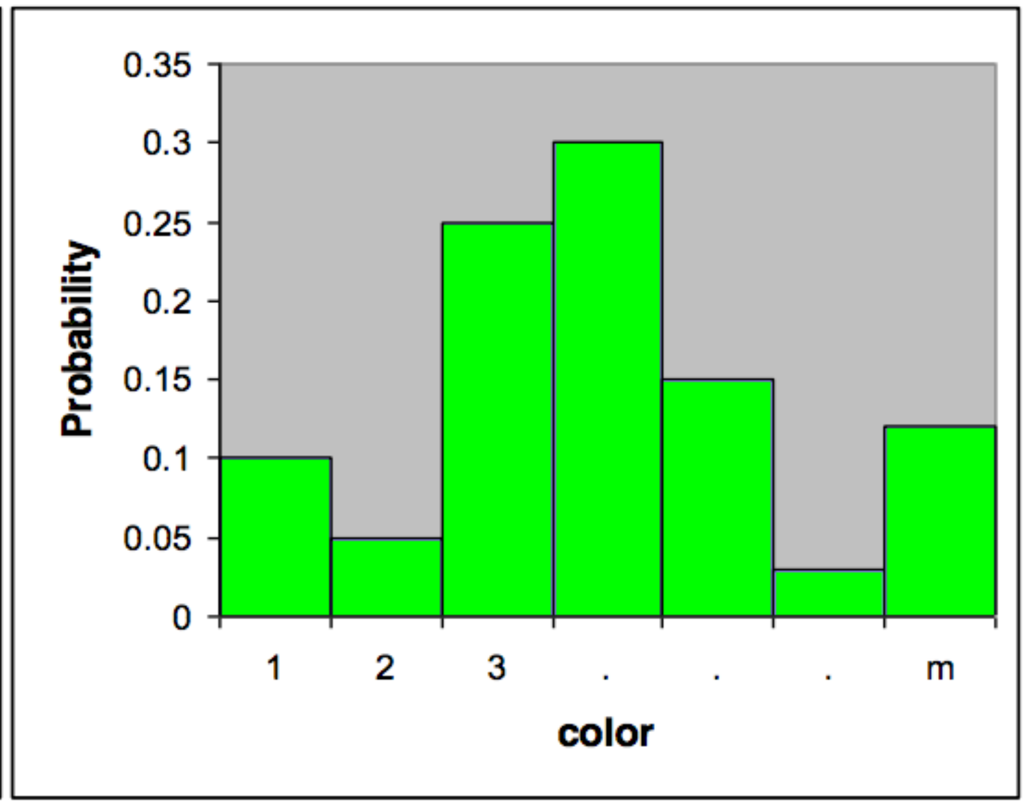




Distance between Histograms





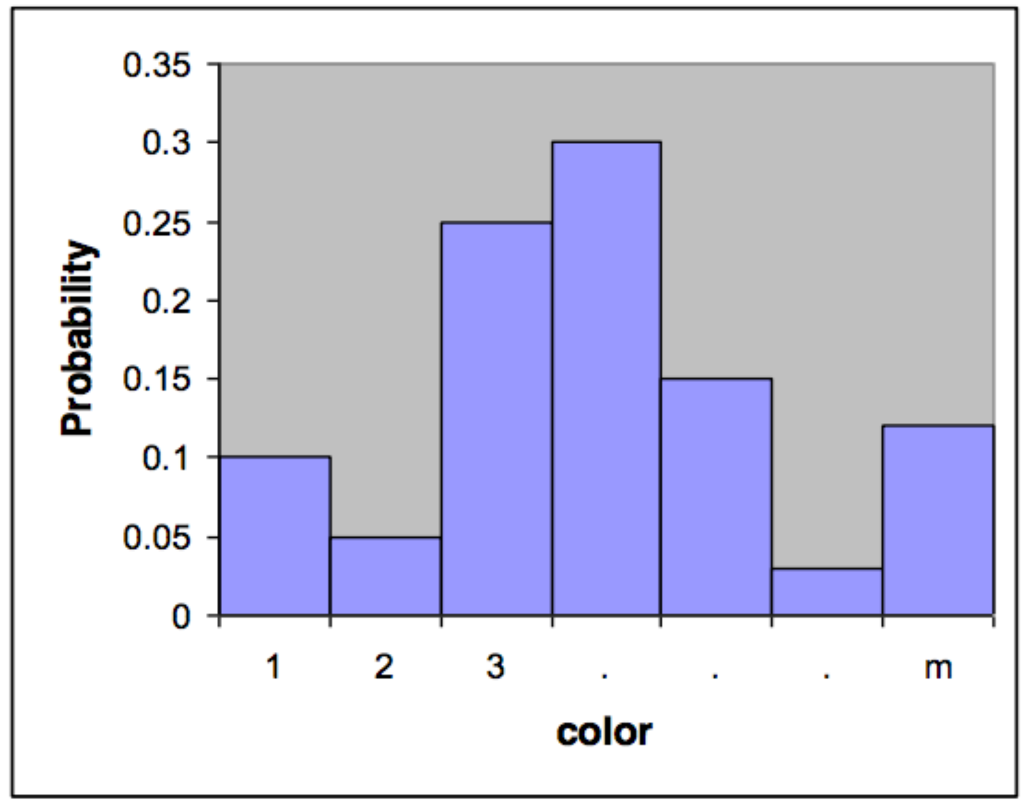


Using Mean-Shift on Color Models

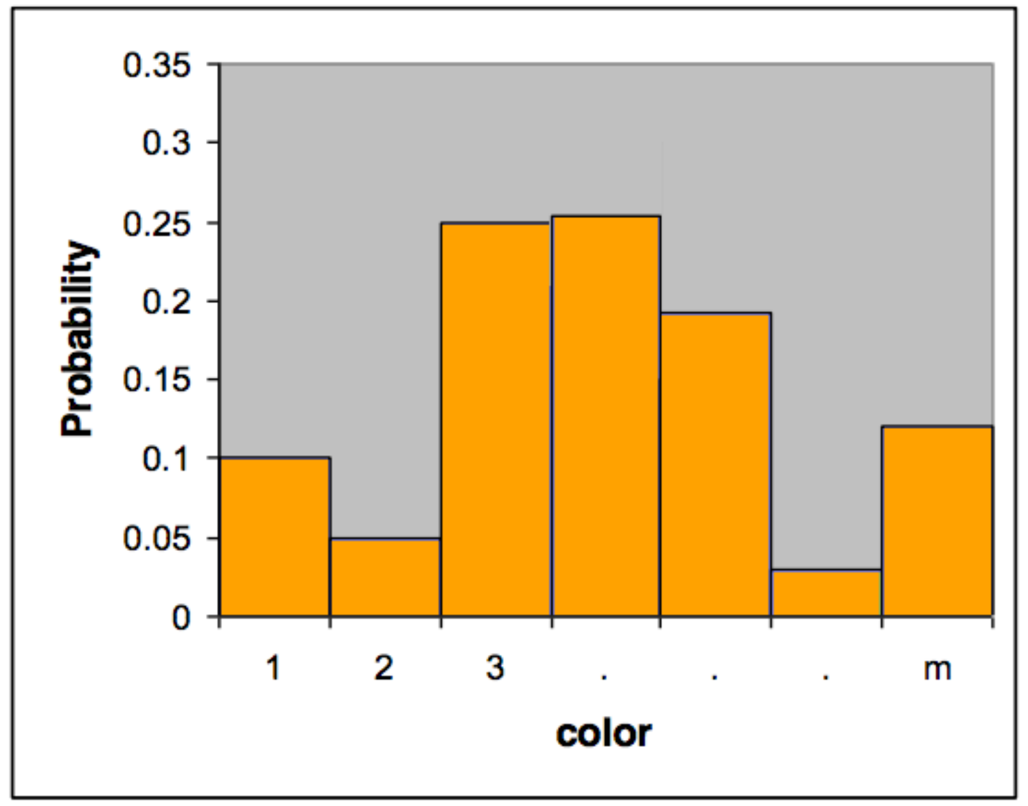
Two approaches:

- 1) Create a color “likelihood” image, with pixels weighted by similarity to the desired color (best for unicolored objects)
- 2) Represent color distribution with a histogram. Use mean-shift to find region that has most similar distribution of colors.
- Think about it… discuss the advantages of both, can you think of other color models that might be useful?

What Distance Metric?



q



$p(y)$

Bhattacharyya Distance

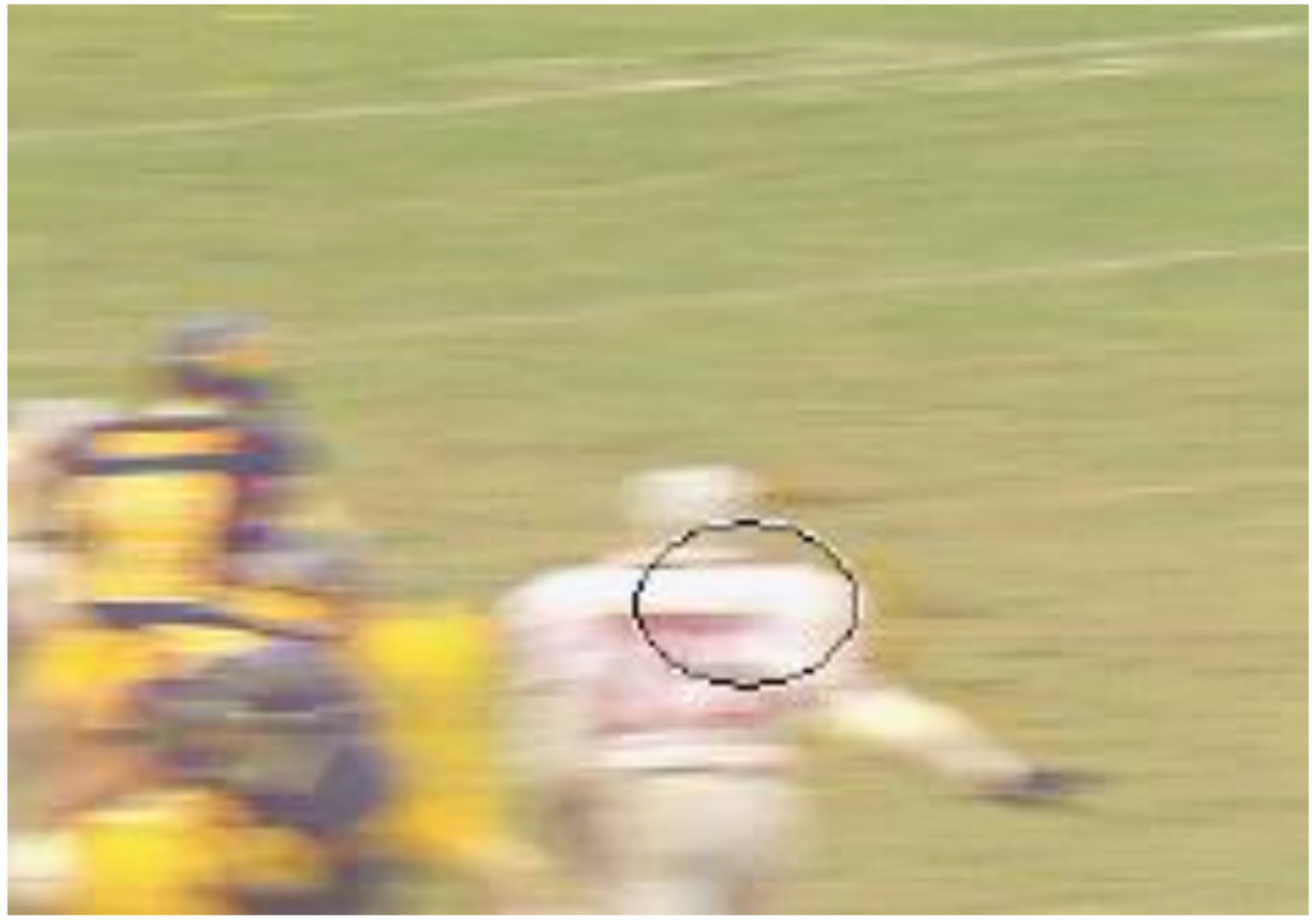
$$f(q, p(y)) = \sum_u \sqrt{q_u p_u(y)}$$

From Comaniciu, Ramesh, Meer

Feature space: 16×16×16 quantized RGB
Target: manually selected on 1st frame
Average mean-shift iterations: 4

Partial occlusion

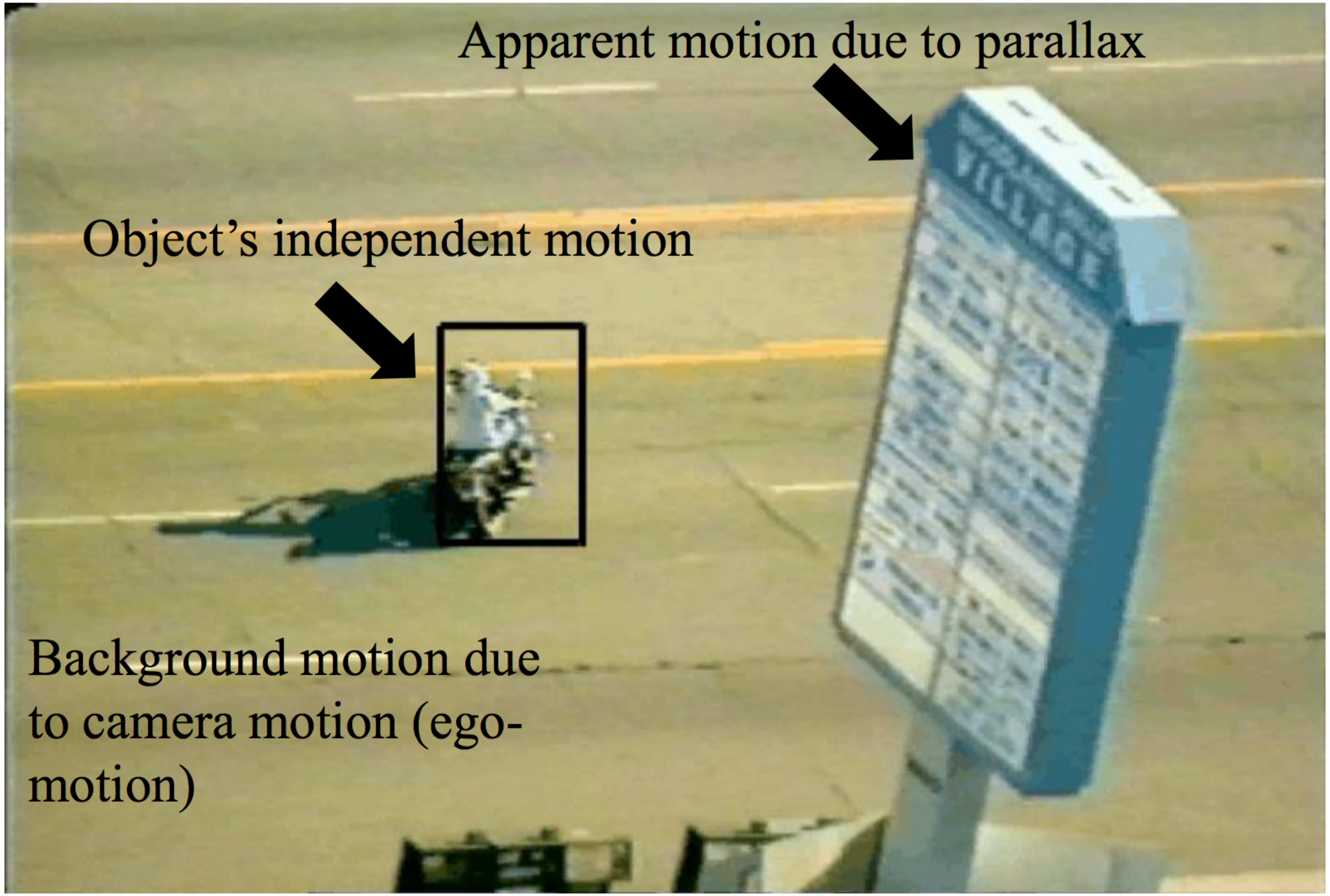
Distraction



Motion blur

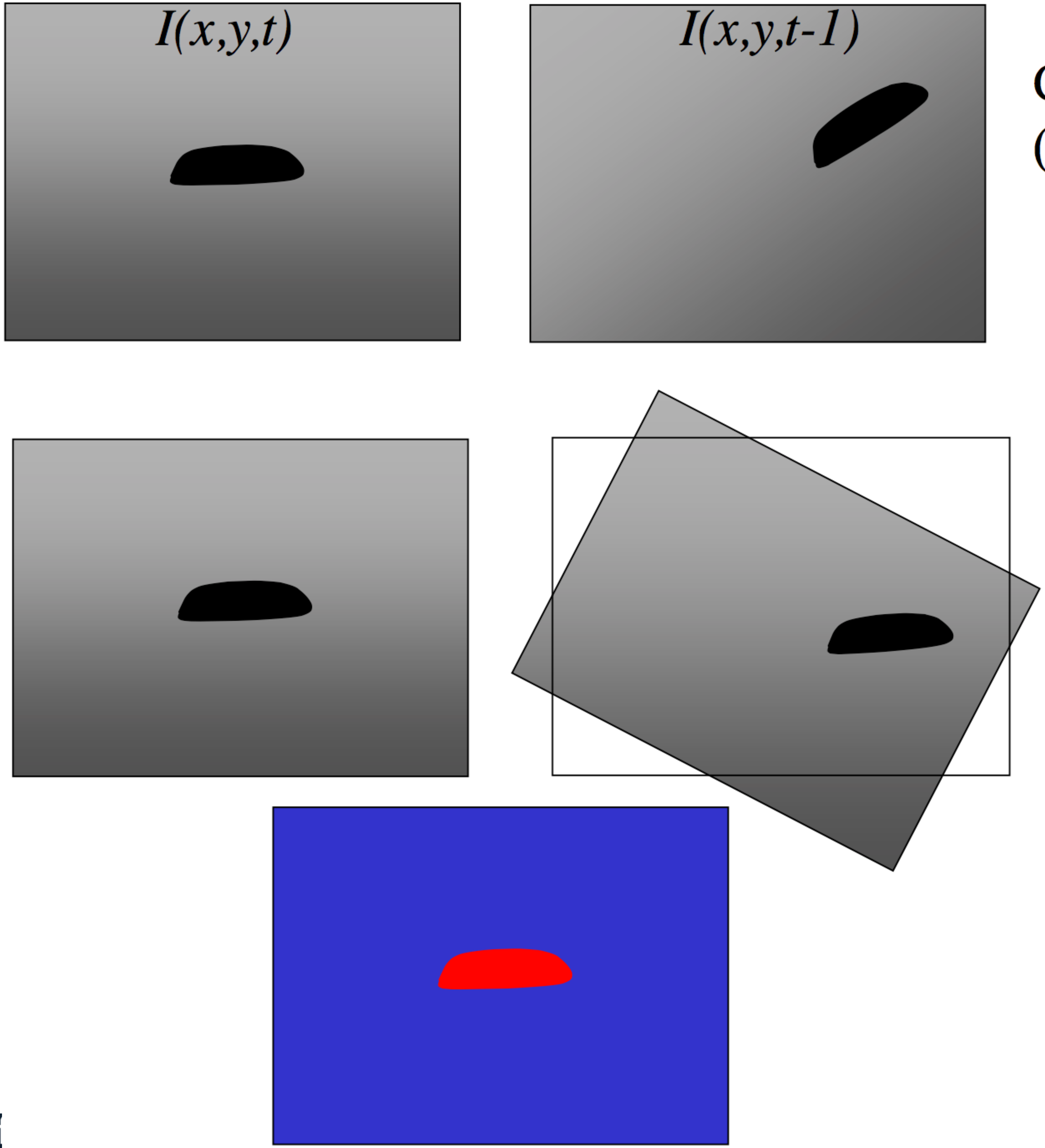
Motion Segmentation

Motion Segmentation



Separating Motion

- Consider removing the dominant motion in the image as "background"
  - Assumes a large static "background" + "small" moving objects
- How about segmentation from independent motion and parallax: i.e. "Layered" representations

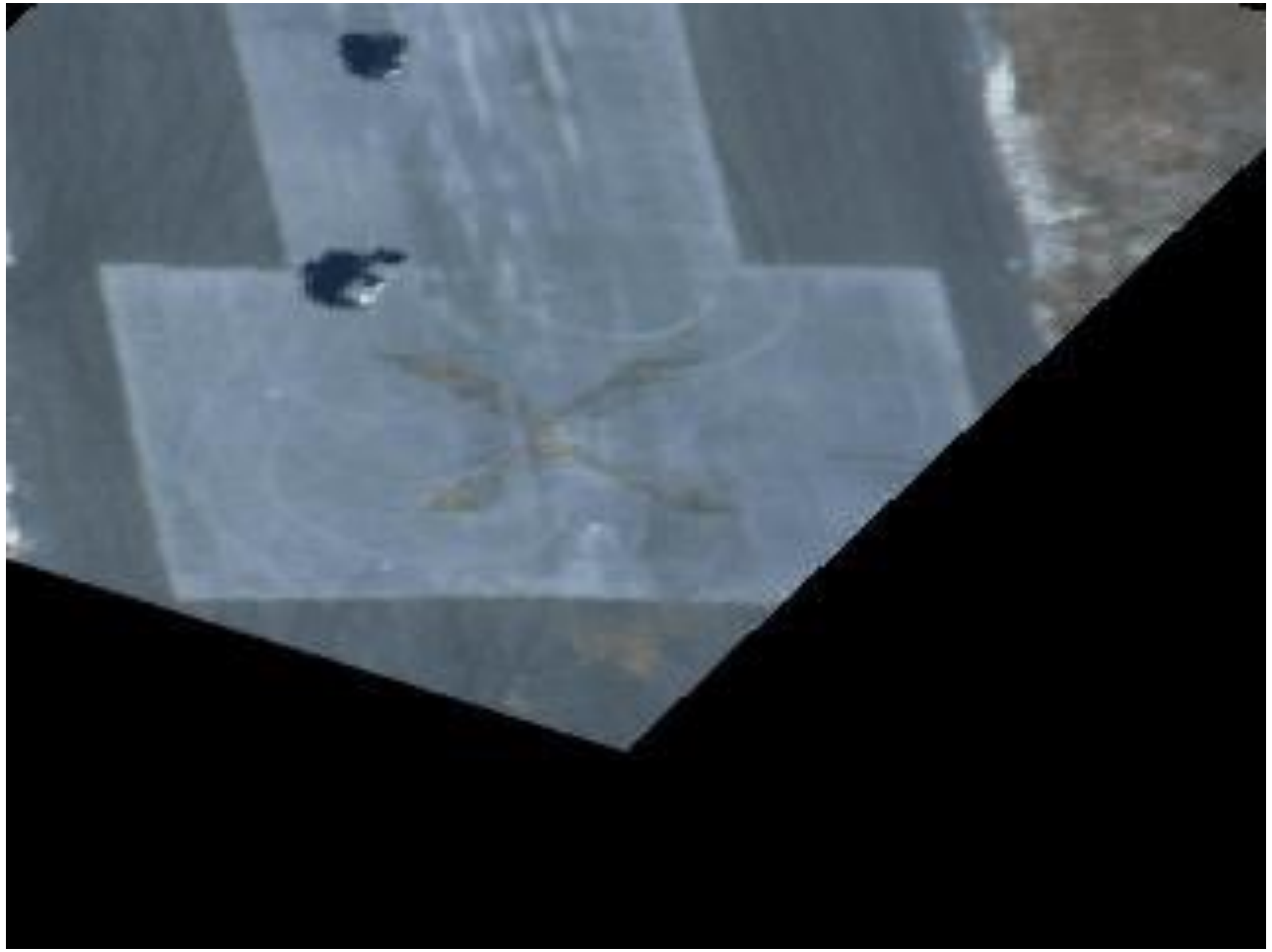


Compute flow
$(u(x,y),v(x,y))$

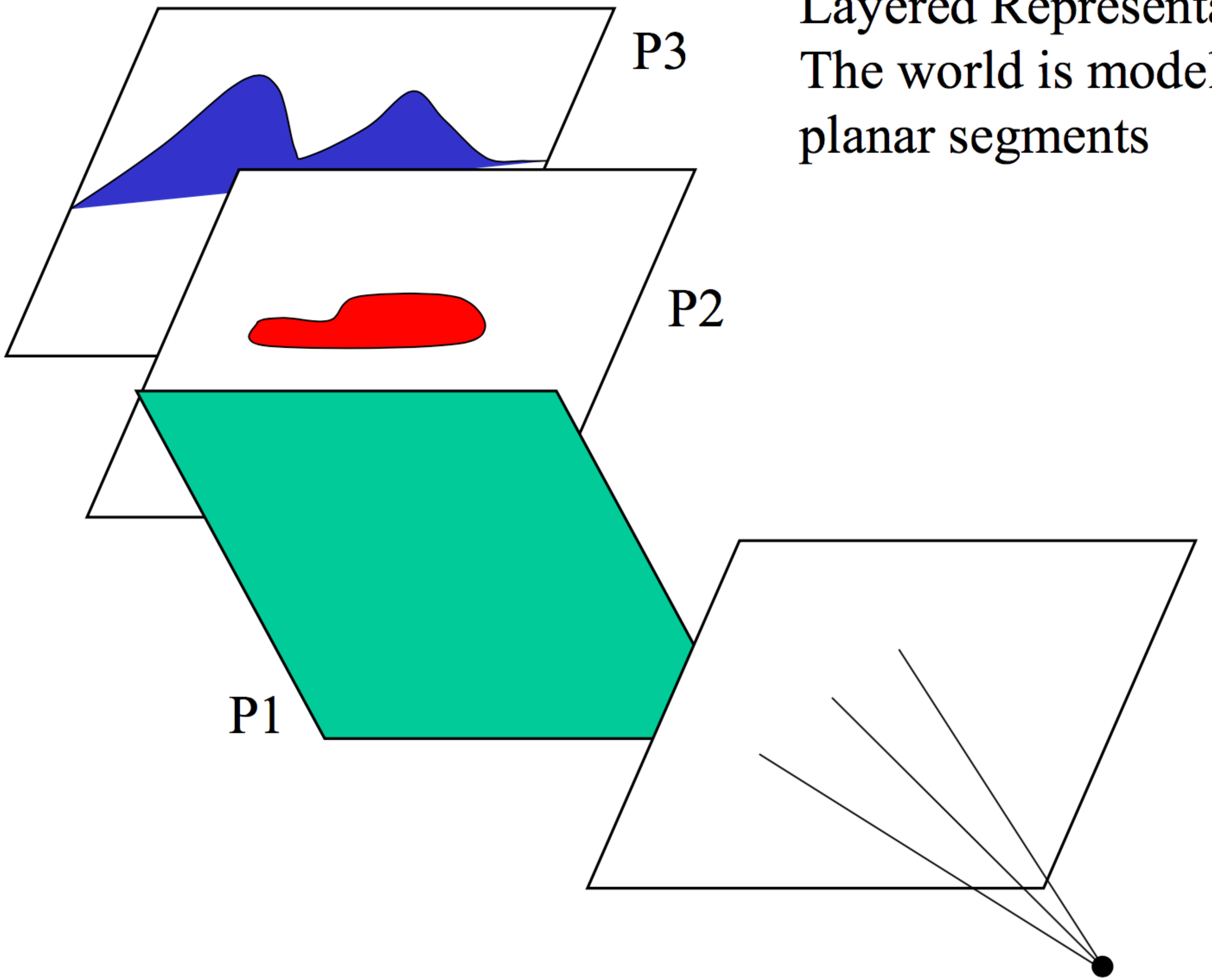
Warp image

Compare images

Warp/align the dominant background



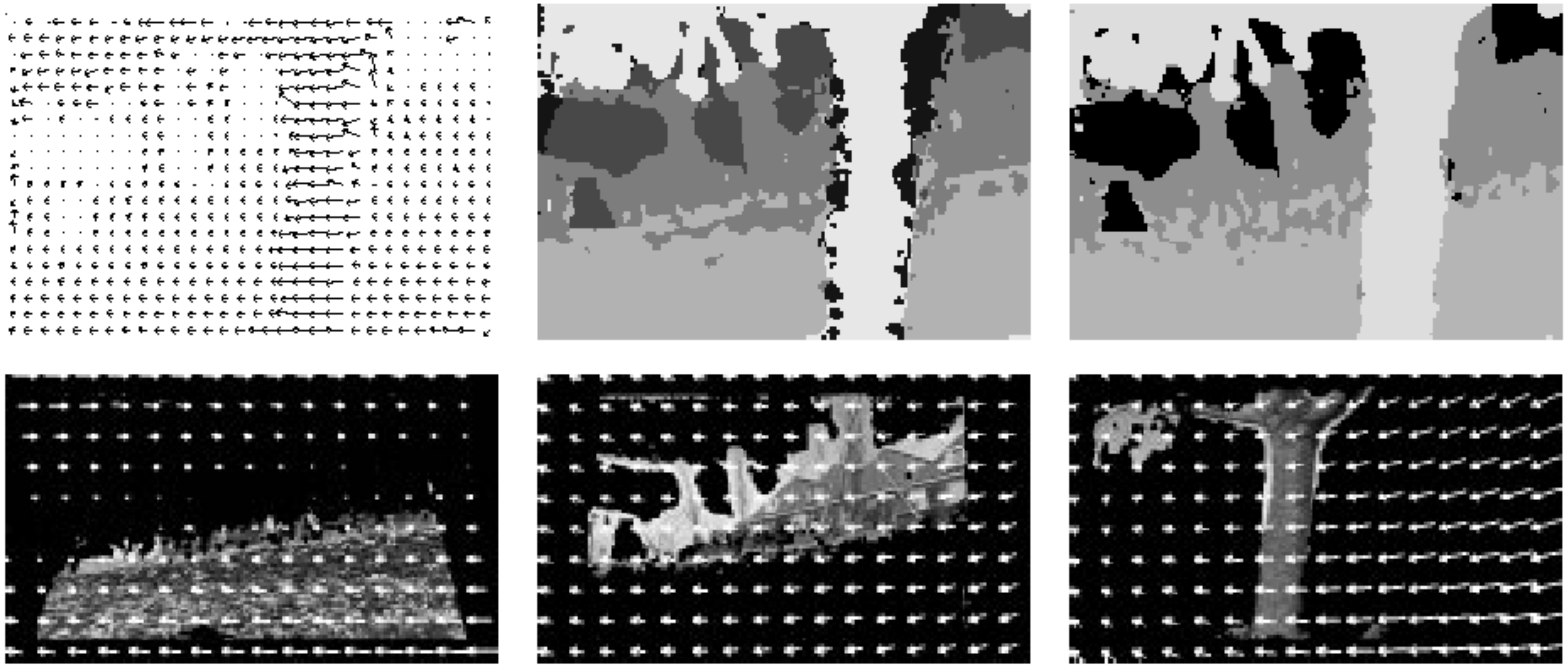
Layered Representation:
The world is modeled as a set of planar segments



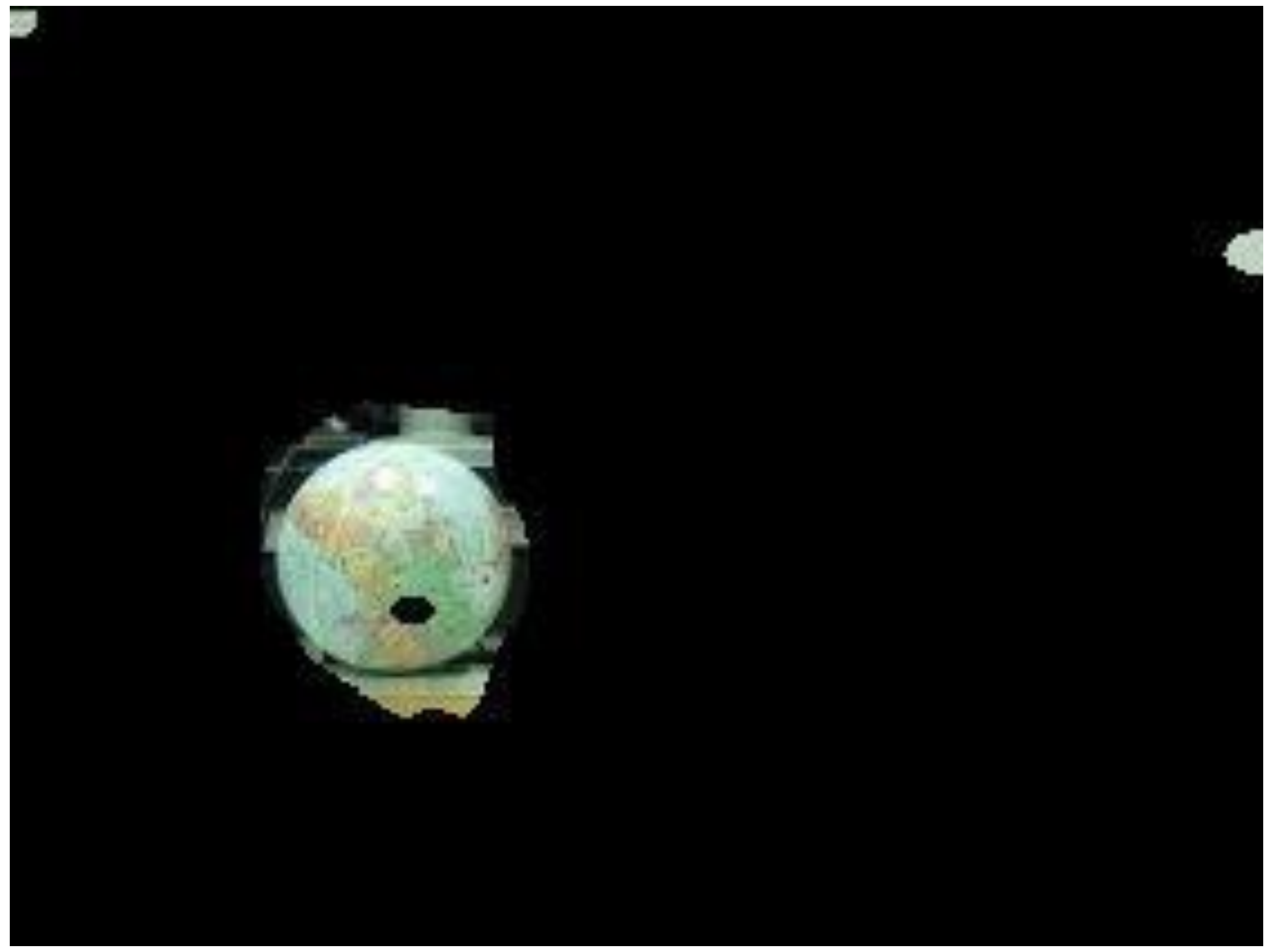
Layered Representations : Outline

- Estimate the dominant motion a over the entire image.
- Warp the images according to the dominant motion.
- Extract the regions that do not agree with the dominant motion.
- For each region:
  - Estimate the dominant motion again over that region
  - Extract the moving objects in the region by warping and differencing.
- This assumes that there is a dominant motion, i.e., one region is very large relative to the others. This might not be the case, so another approach is to first divide into K different motion models and find the corresponding pixels. A sketch of the algorithm is:
  - Estimate the motion in local blocks (e.g., NxN windows in the image)
  - Each window produces a vector representing the estimate motion (e.g. 6-dim if affine model)
  - Cluster the vectors into K clusters (corresponding to K motions) Assign each pixel to the cluster that best agrees with the local motion
  - Re-estimate the motion in each cluster and iterate

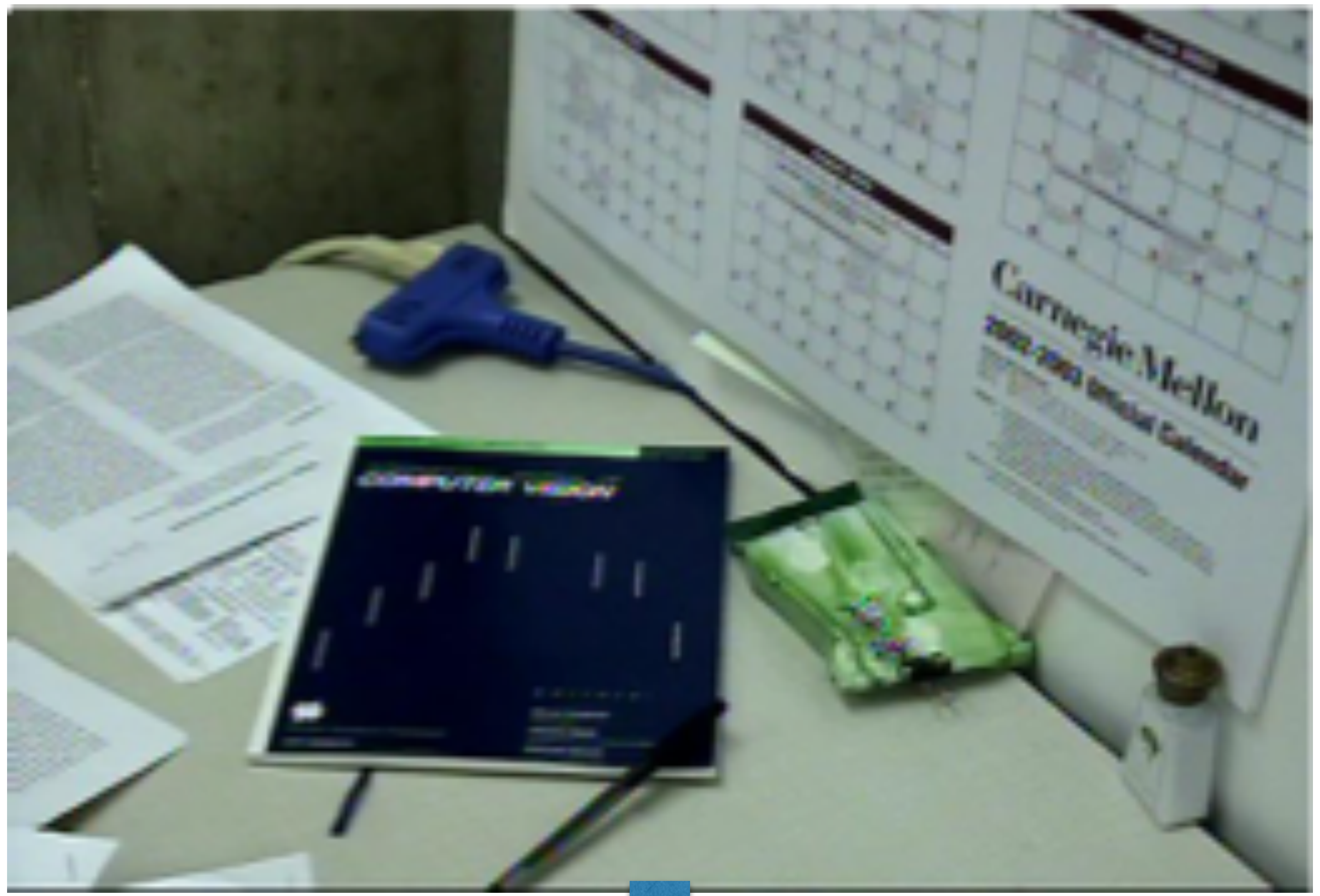
Motion Segmentation in Layers



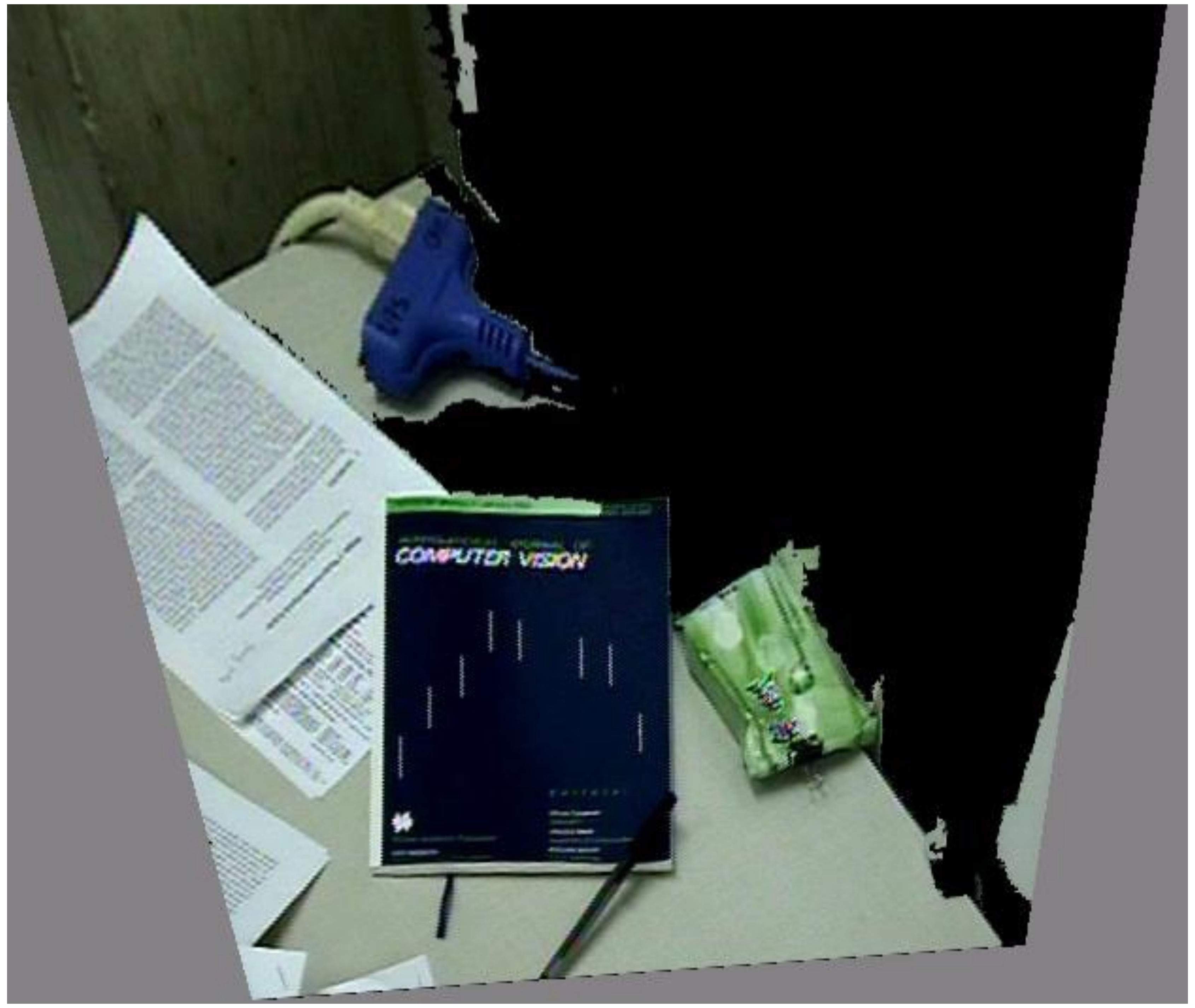
Example from Adelson et al.



From M. Irani, Weizmann Inst.


Carnegie Mellon
2002-2003 Official Calendar
COMPUTER VISION


Carnegie Mellon
2002-2003 Official Calendar


COMPUTER VISION

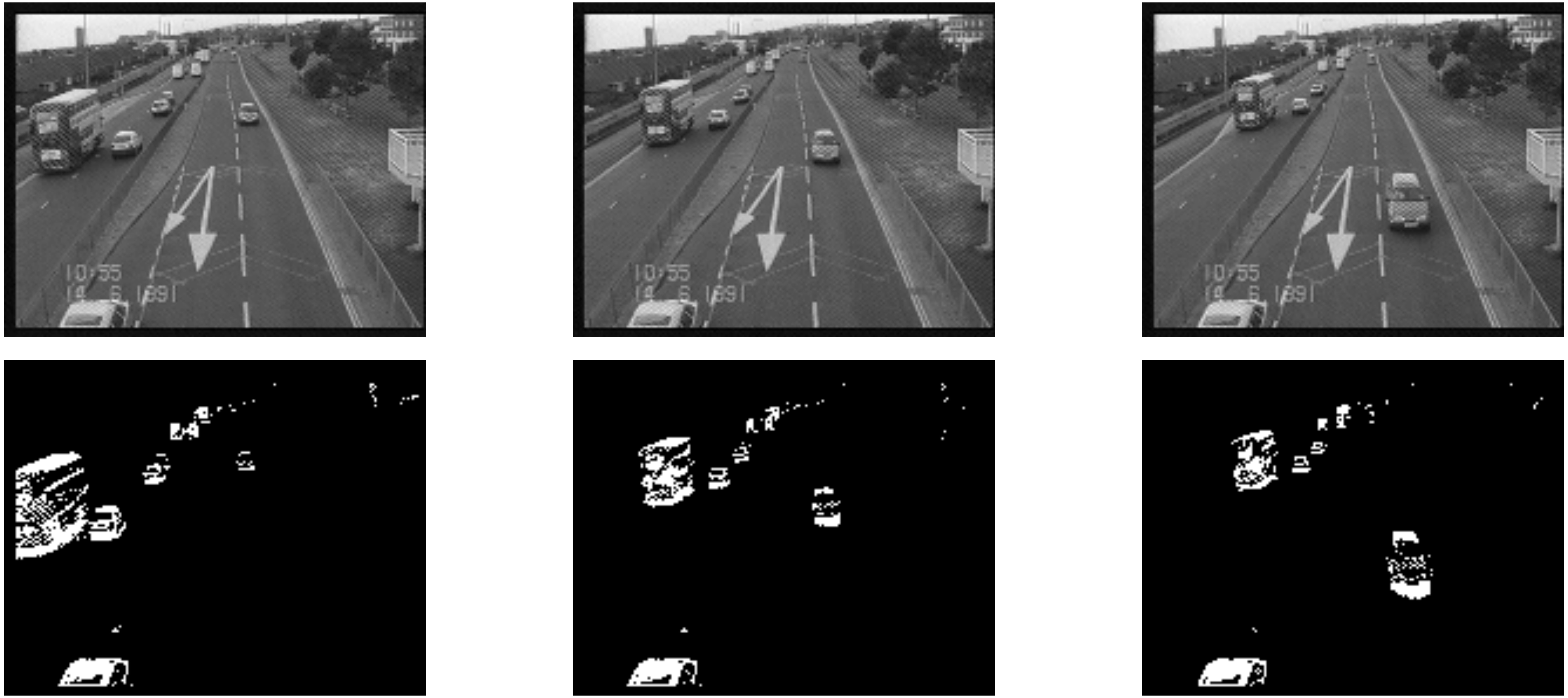
Q. Ke & T. Kanade

Side Note

- Drone footage as which is currently a huge expansion area in computer vision is particularly well suited to exploring these techniques.

Change Detection/Background Models

Simple Background Subtraction



$$|I(x,y) - I_b(x,y)| > \tau \ ??$$

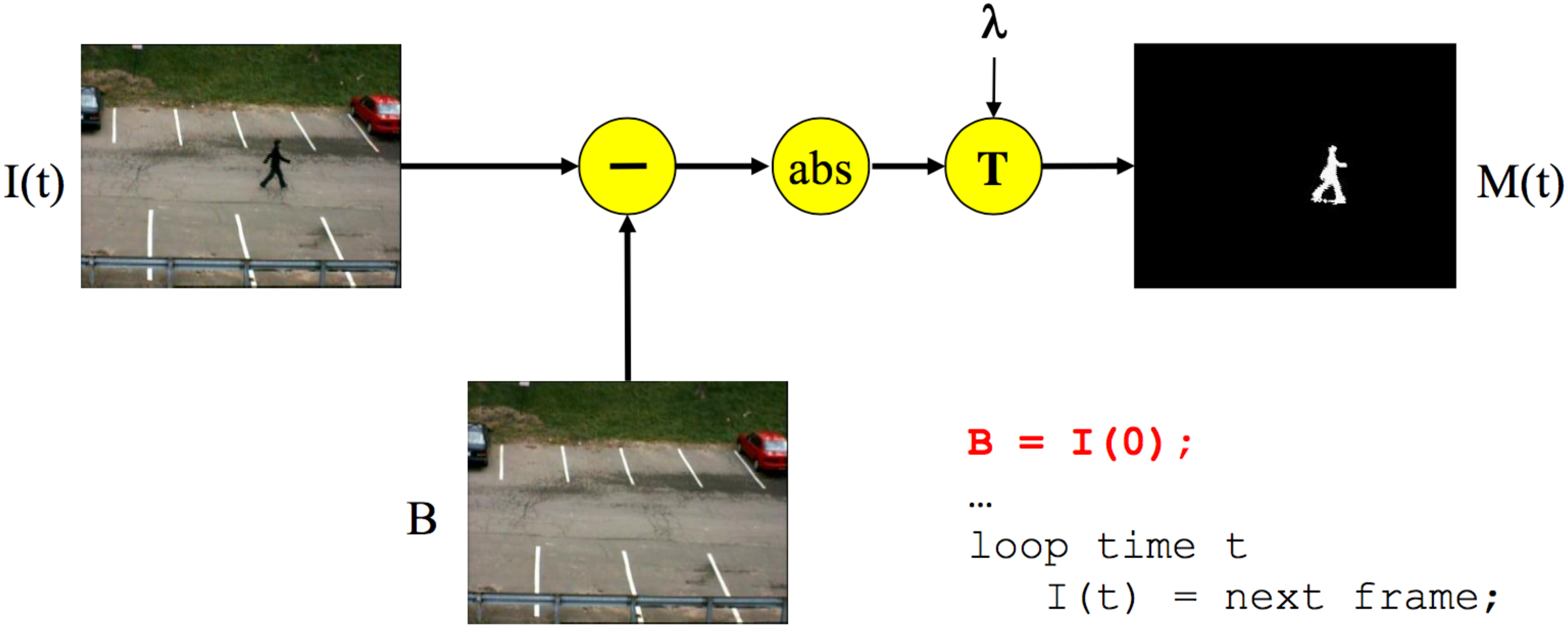
Simple Background Subtraction

-

=

Example from A. Efros

Simple Background Subtraction

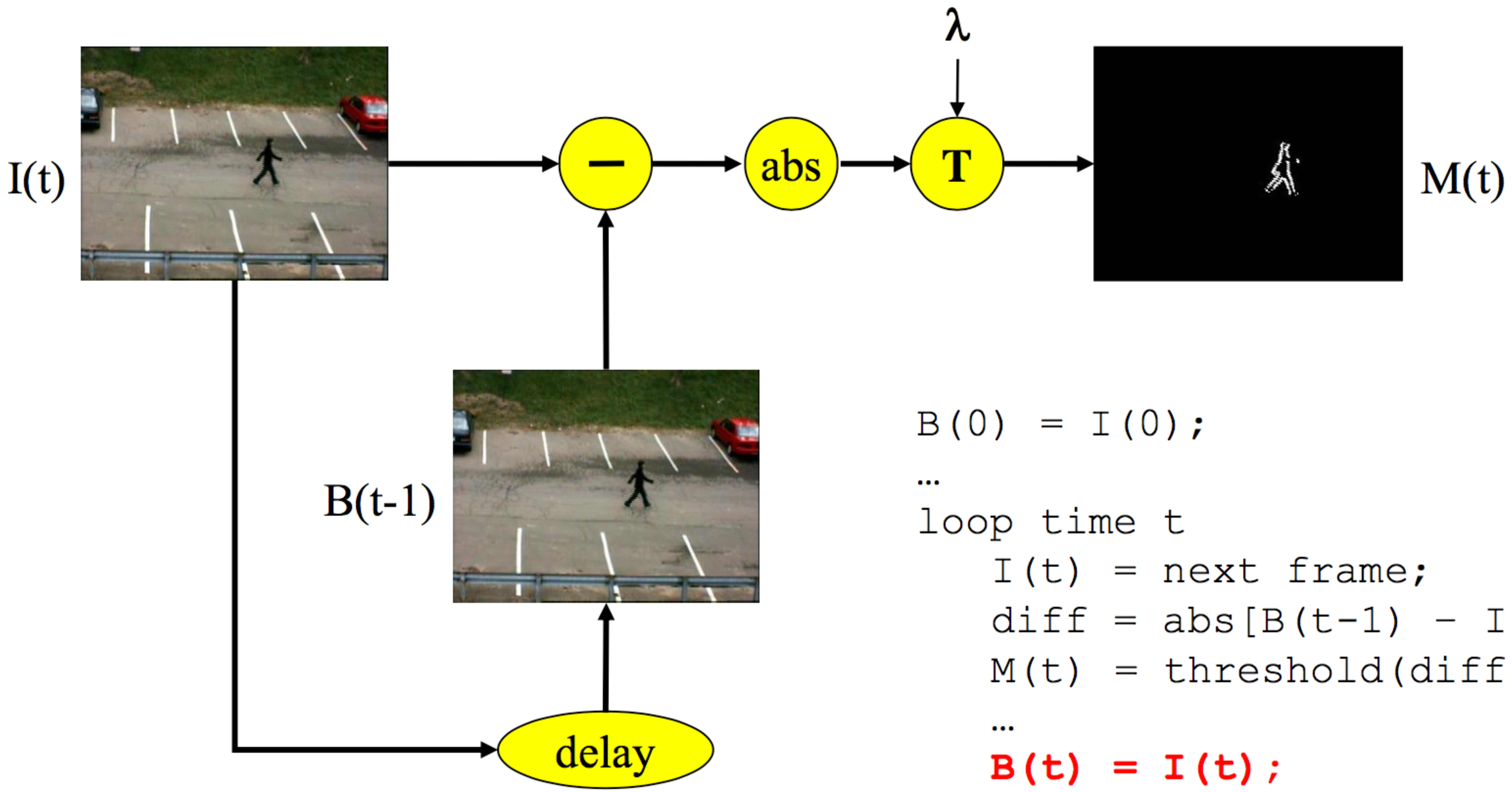



```
B = I(0);
…
loop time t
    I(t) = next frame;
    diff = abs[B - I(t)];
    M(t) = threshold(diff,λ);
    …
end
```

- Background model is a static image (assumed to have no objects present).
- Pixels are labeled as object (1) or not object (0) based on thresholding the absolute intensity difference between current frame and background.

Slide adapted from B. Collins

Simple Frame Differencing

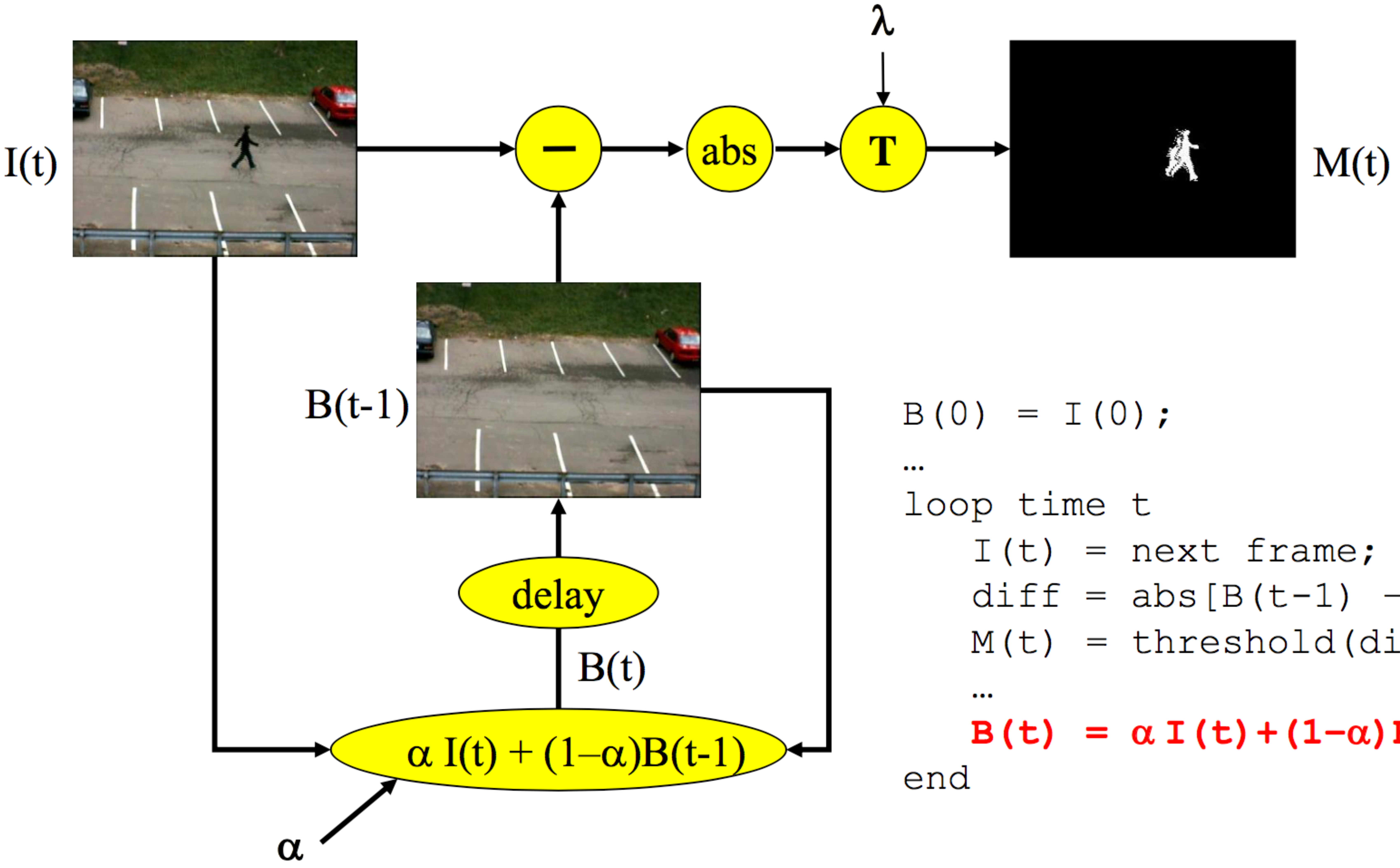



```
B(0) = I(0);
…
loop time t
    I(t) = next frame;
    diff = abs[B(t-1) - I(t)];
    M(t) = threshold(diff,λ);
    …
    B(t) = I(t);
end
```

- Background model is replaced with the previous image.

Credit: B. Collins

Adaptive Background Subtraction



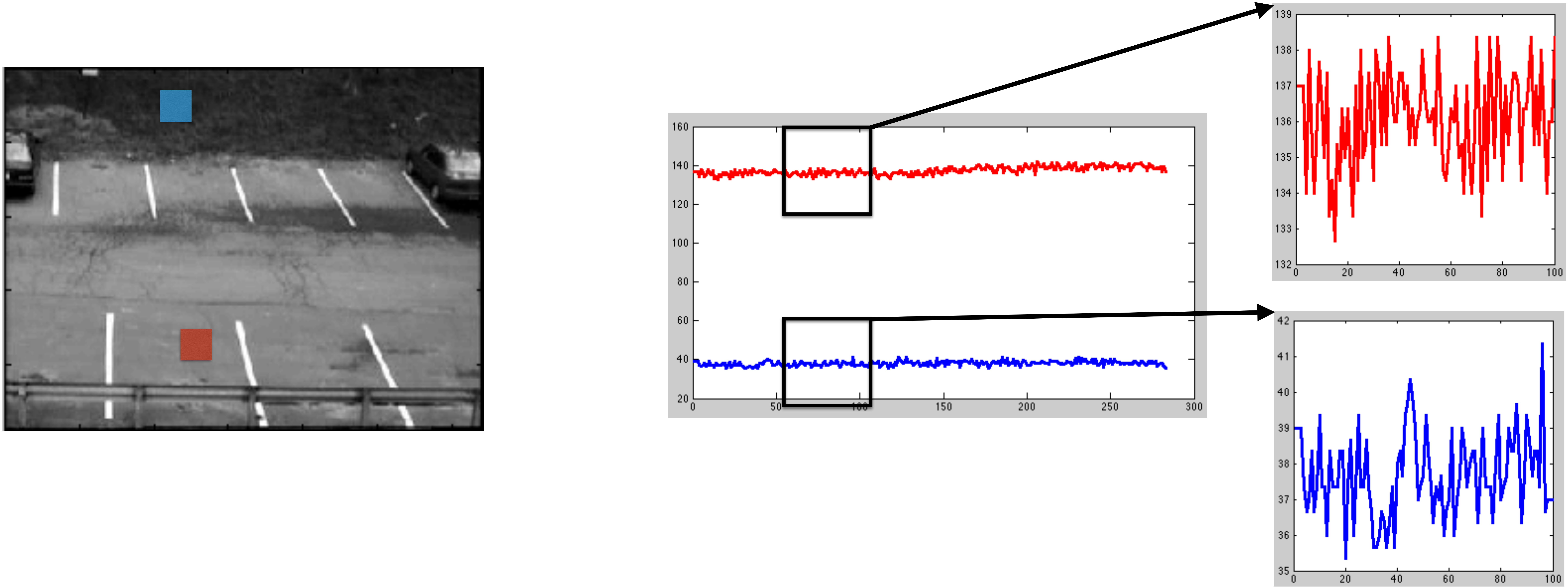

```
B(0) = I(0);
…
loop time t
    I(t) = next frame;
    diff = abs[B(t-1) - I(t)];
    M(t) = threshold(diff,λ);
    …
    B(t) = α I(t)+(1–α)B(t-1);
end
```

- Current image is “blended” into the background model with parameter a
- a = 0 yields simple background subtraction, a = 1 yields frame differencing

Credit: B. Collins

Background Models

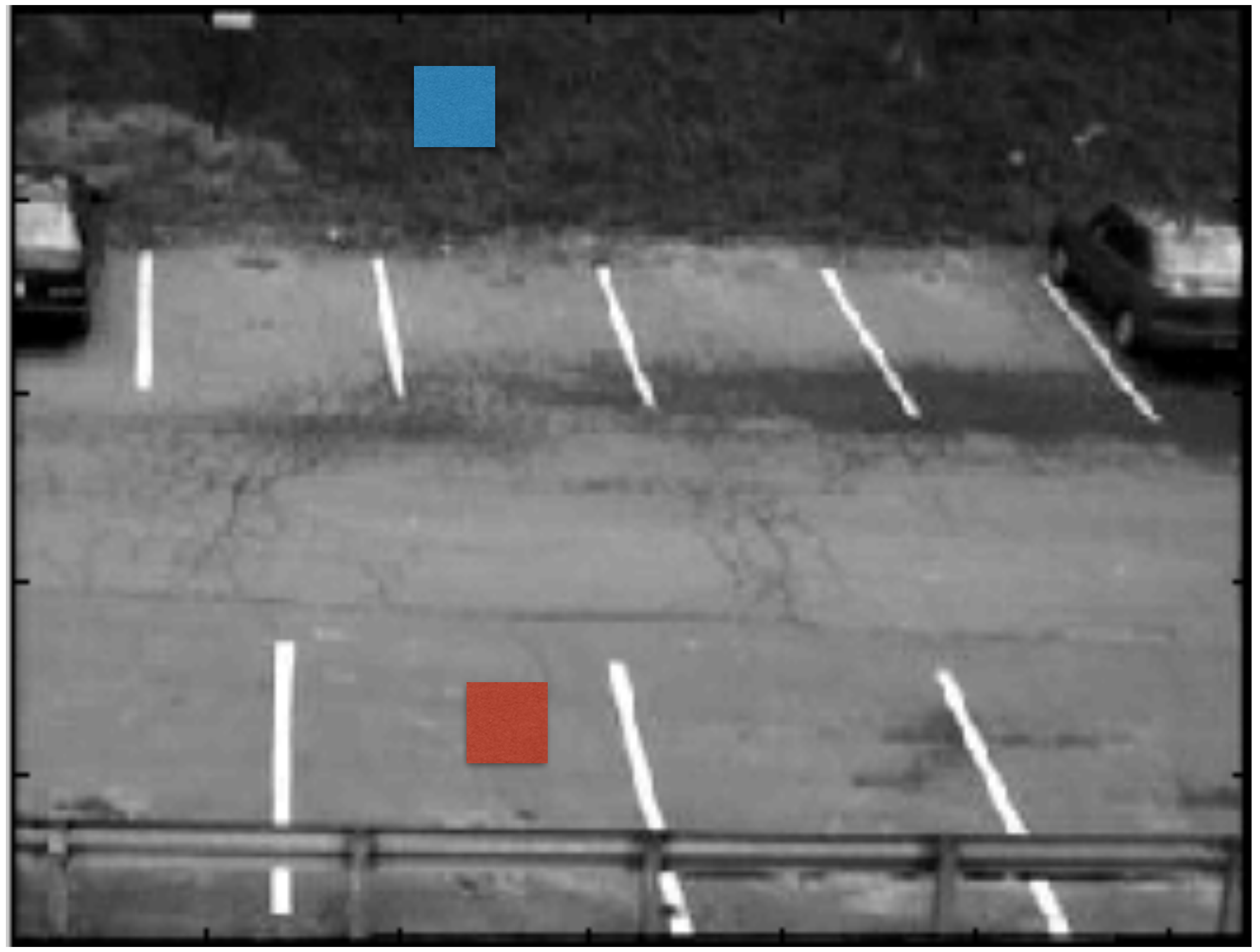
- What is a good statistical model of the value of a pixel in the unchanging background?

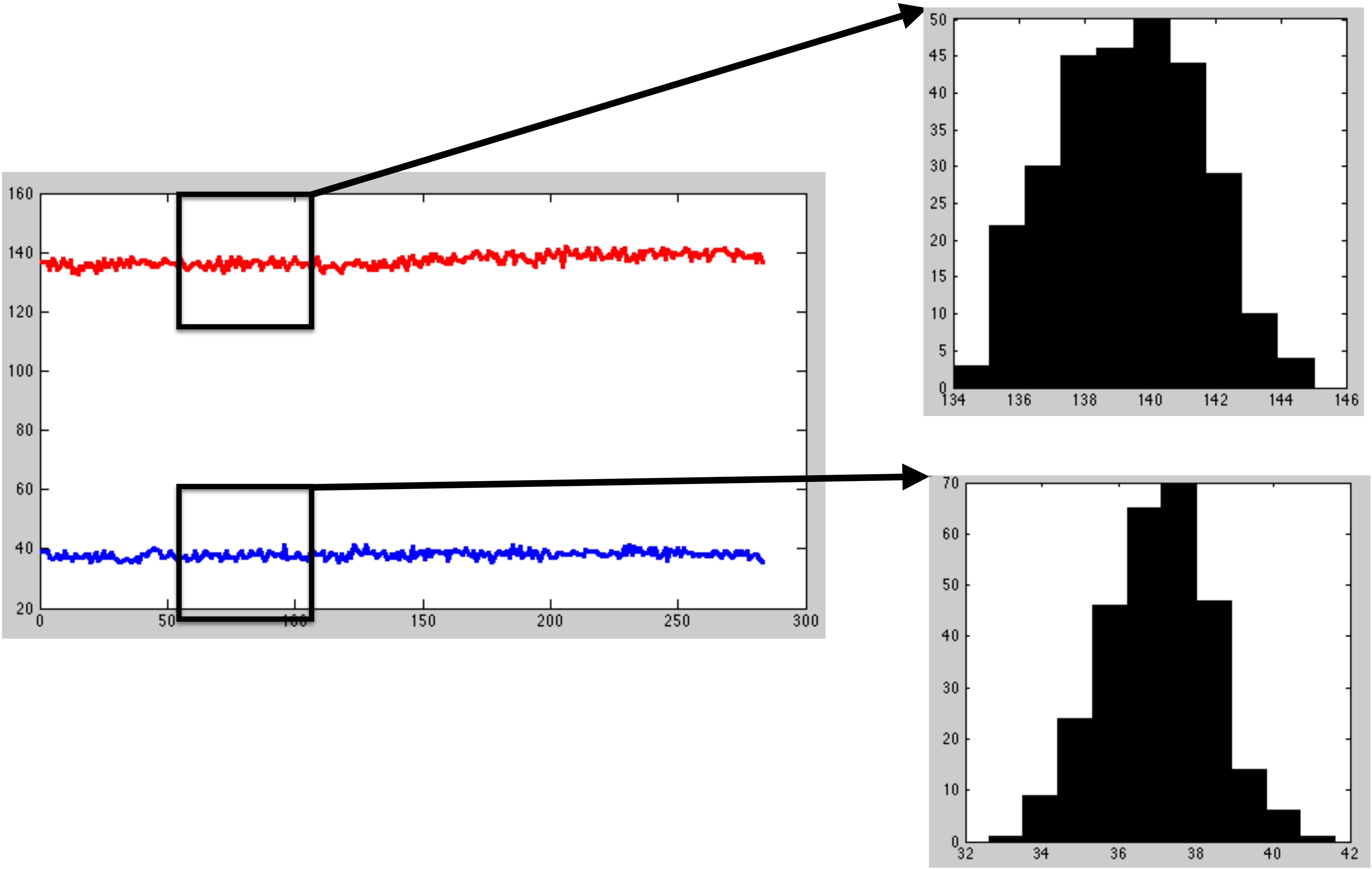


Credit: B. Collins

Background Models

- What is a good statistical model of the value of a pixel in the unchanging background?

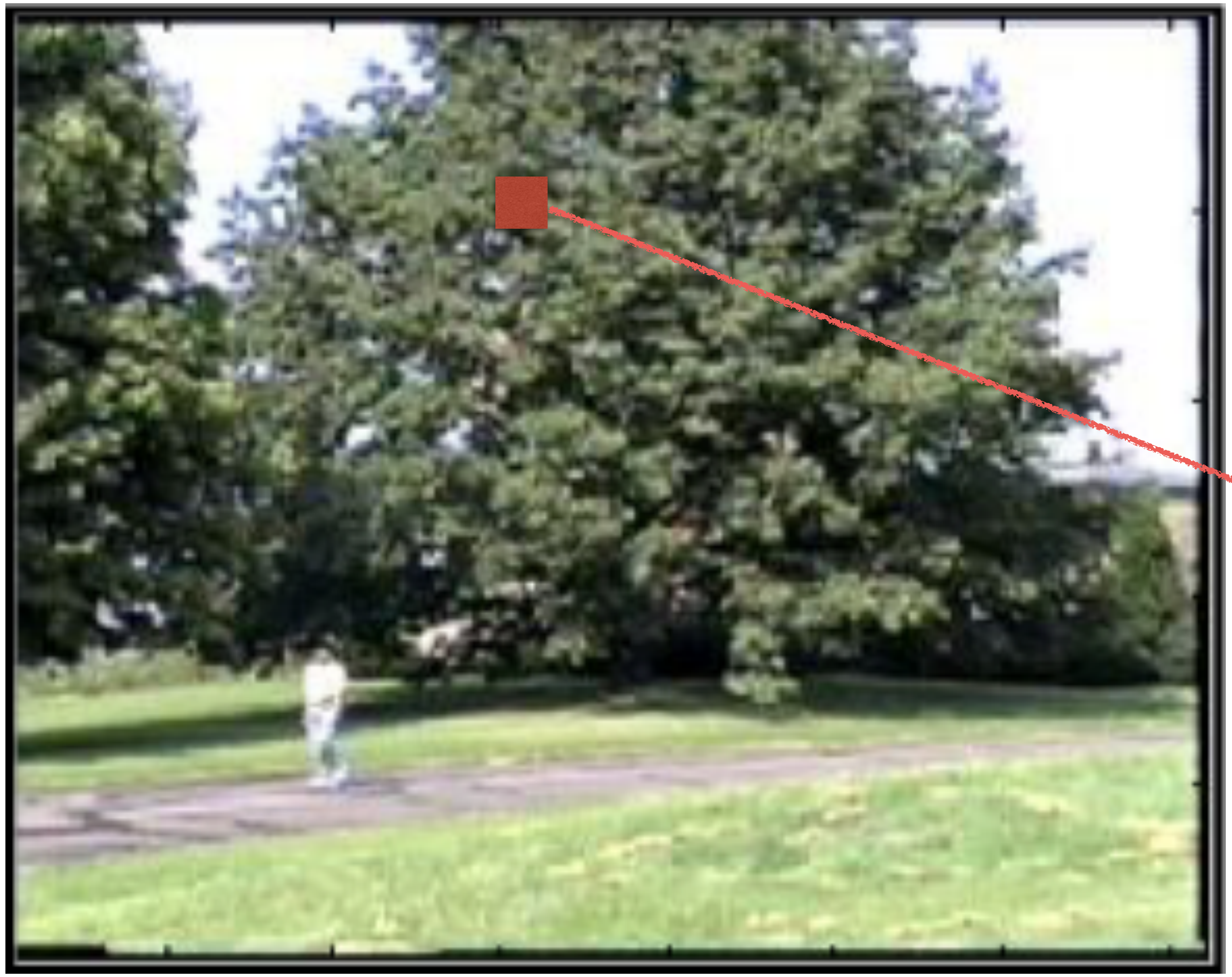


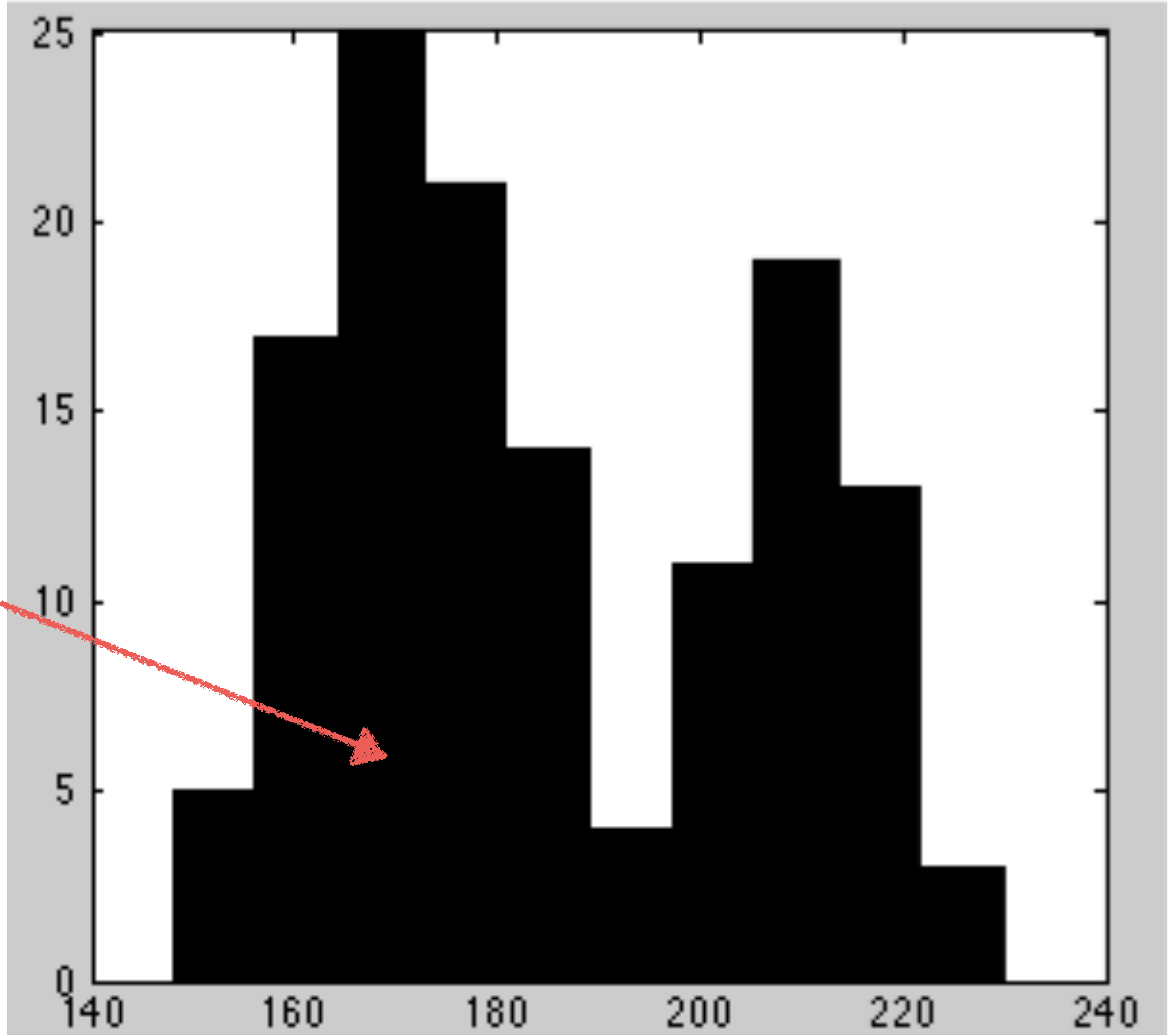


Credit: B. Collins

Limitation of Gaussian Assumption

- There is a problem with multimodal pixels
- Examples: trees in the wind; rippling water





Credit: B. Collins

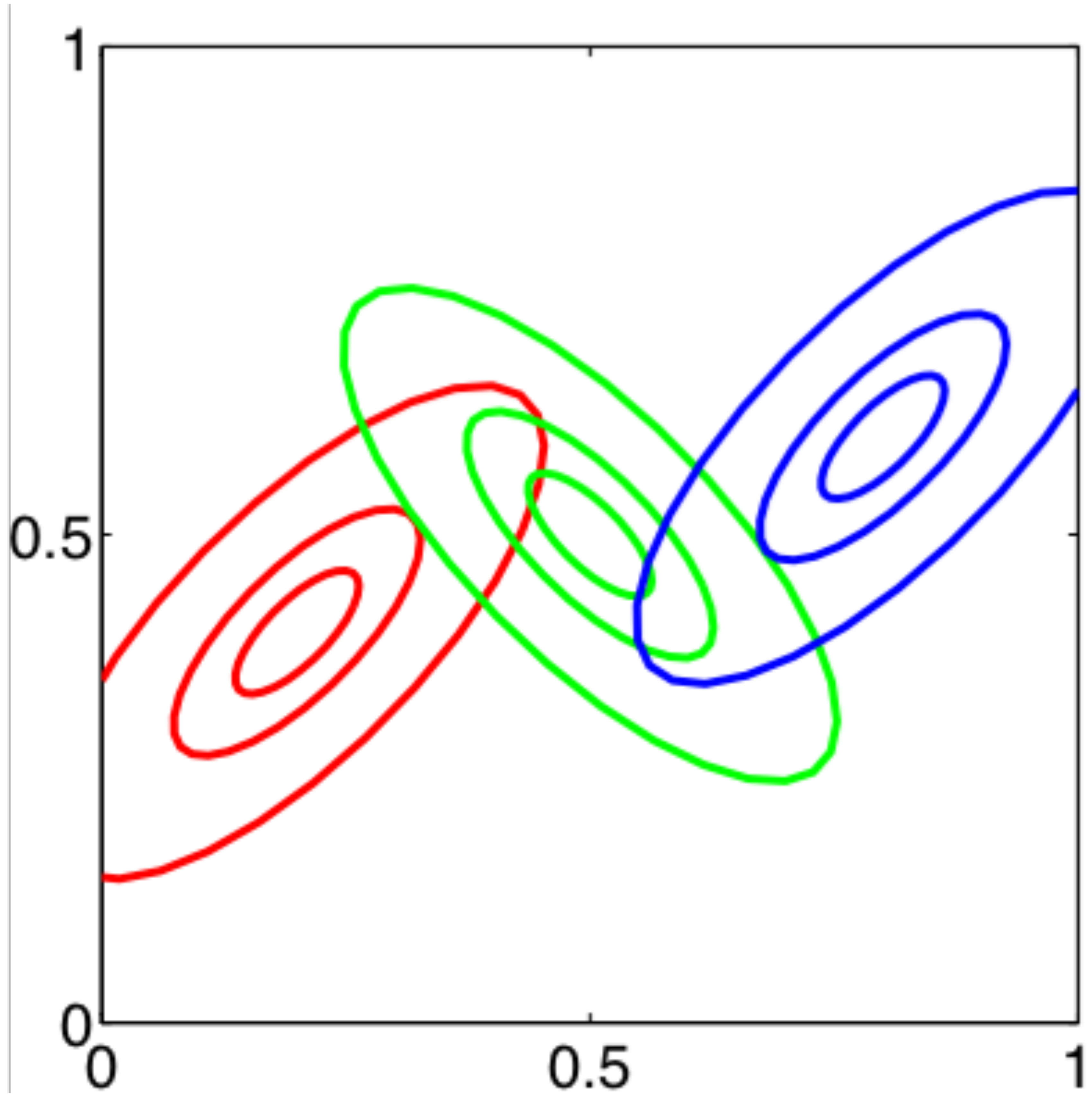
Use a Mixture of Gaussians!

- Linear combination of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

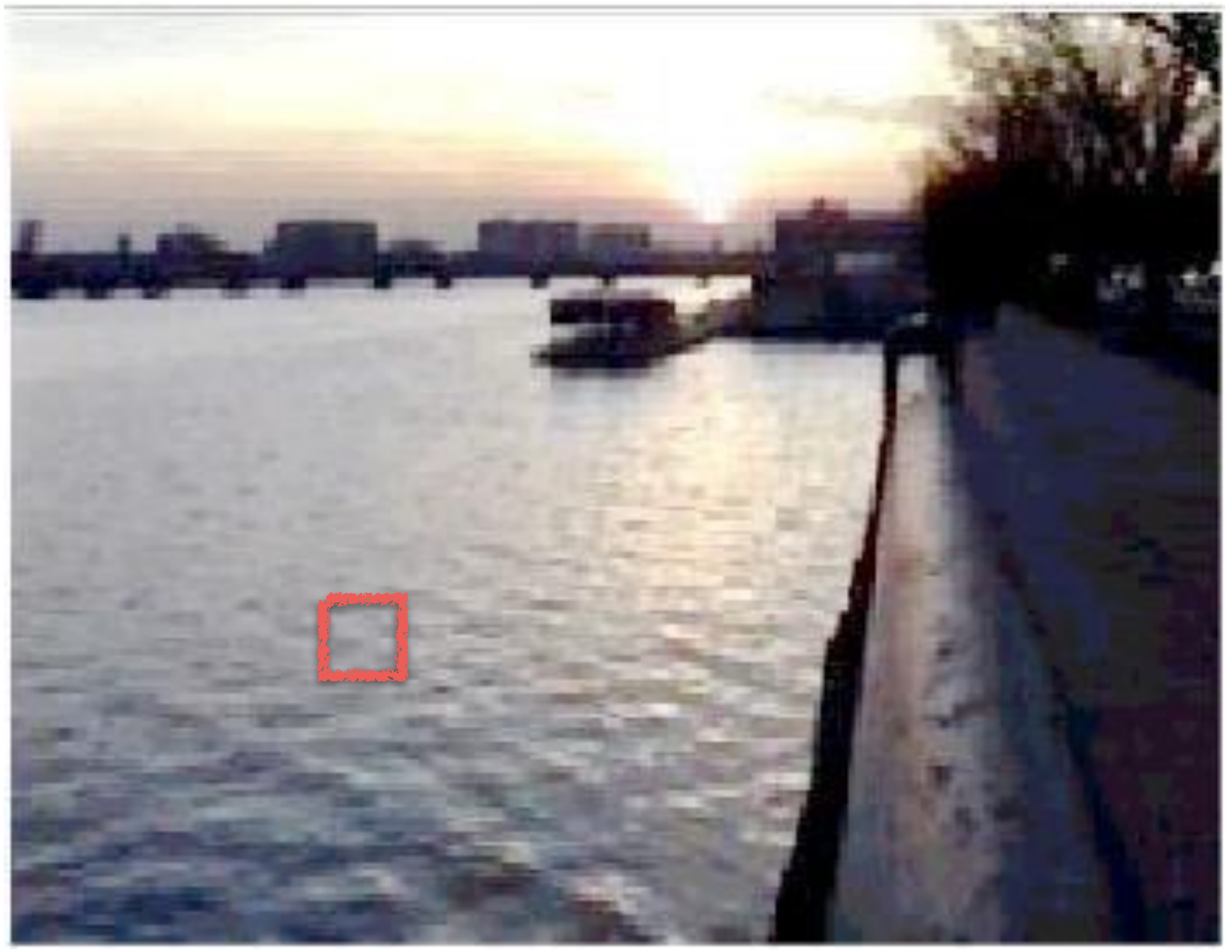
- Normalization and positivity requirements

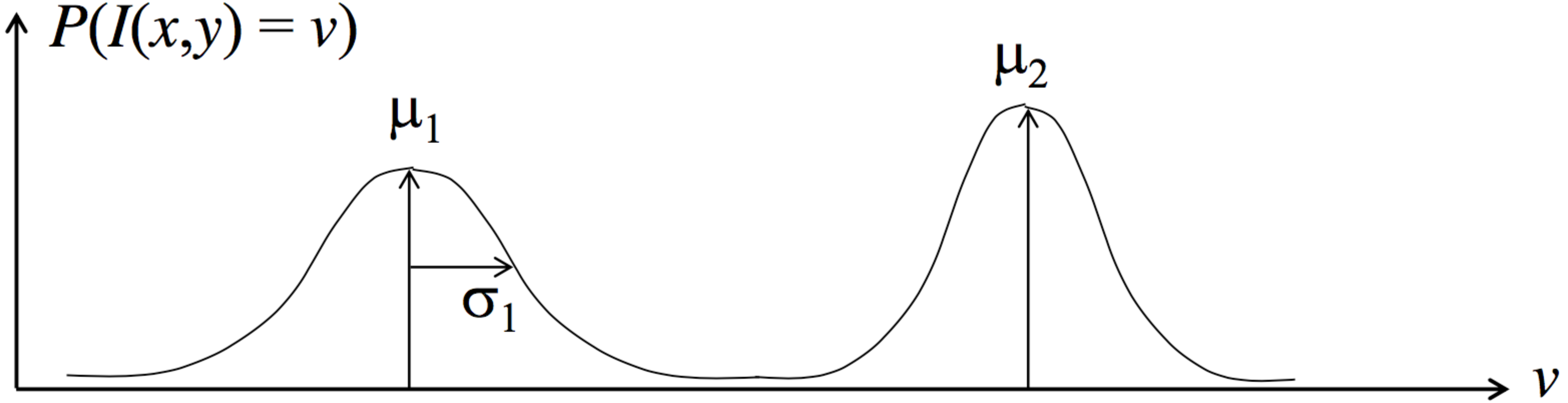
$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$



Credit: B. Collins

Use a Mixture of Gaussians!



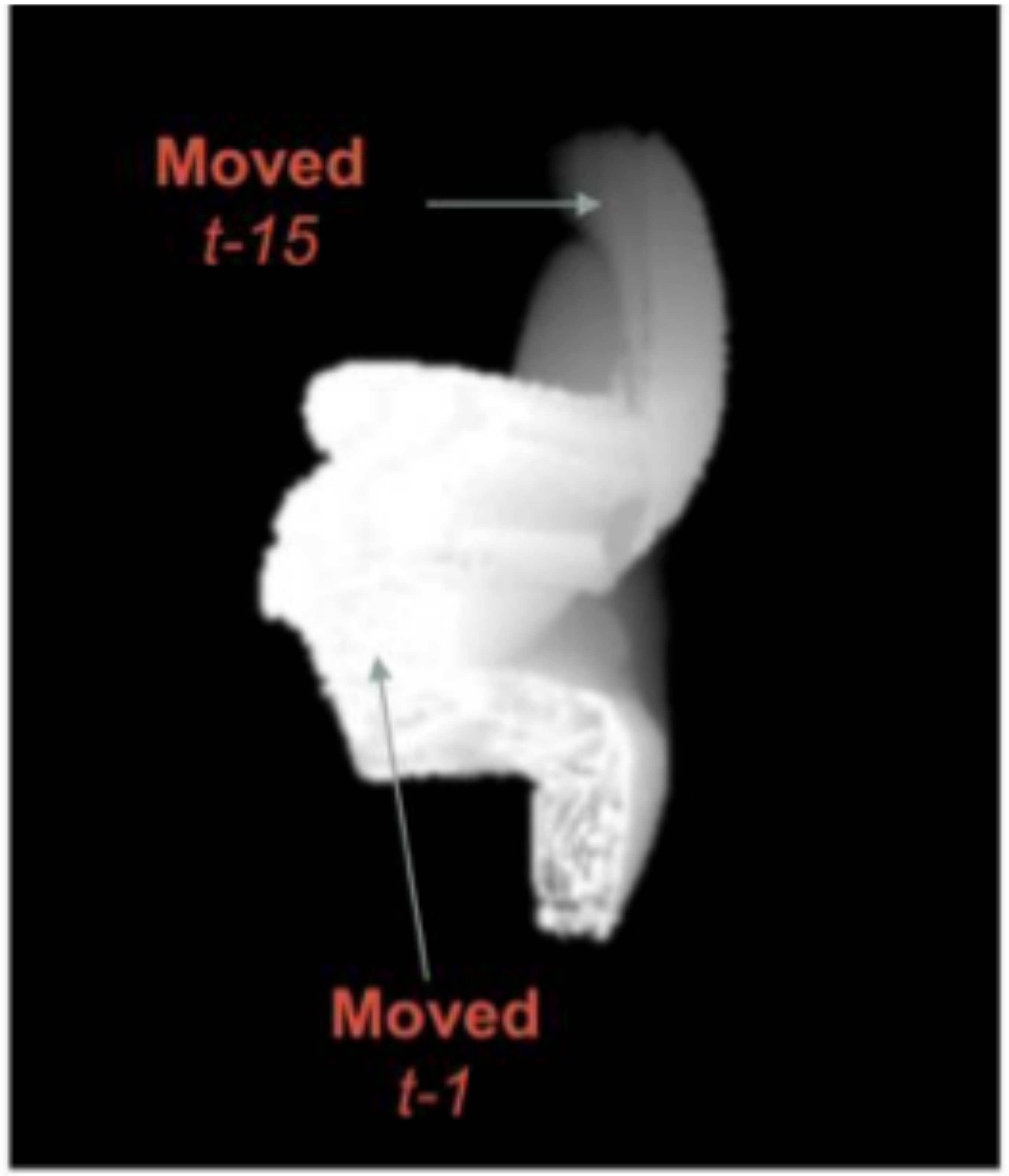


Persistent Frame Differencing

- Motion history images are a different function of temporal volume.
- Pixel operator is replacement decay:

`if moving` $I_\tau(x,y,t) = \tau$
`otherwise`
$I_\tau(x,y,t) = \max(I_\tau(x,y,t-1)-1\ ,0)$

- Trivial to construct $I_{\tau-k}(x,y,t)$ from $I_\tau(x,y,t)$ so can process multiple time window lengths without more search.
- MEI is thresholded MHI



Persistent Frame Differencing