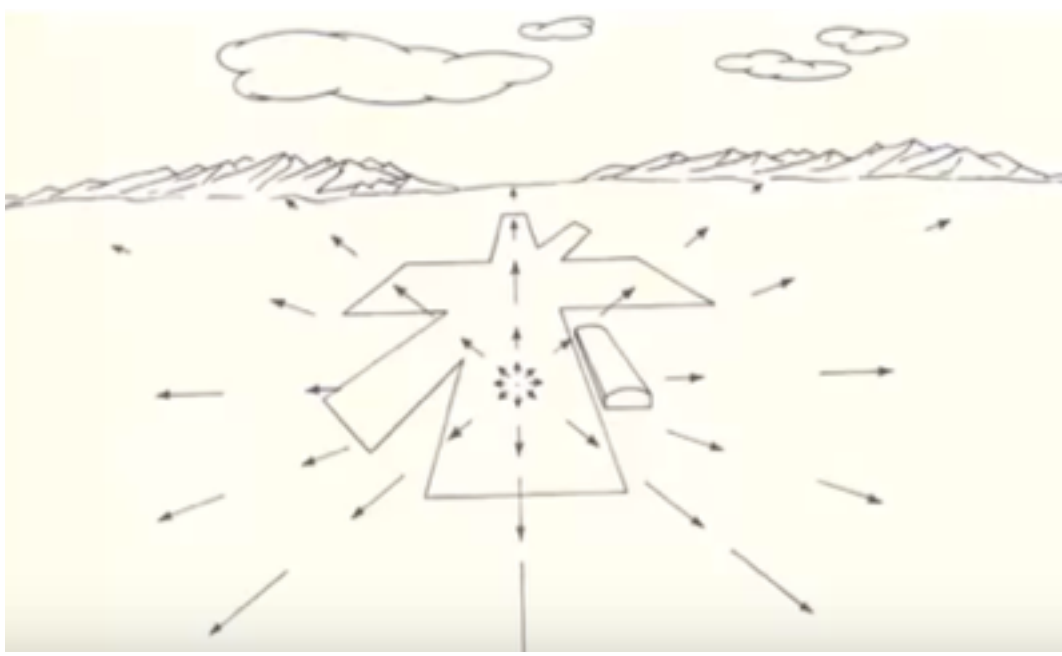
Estimating optical flow

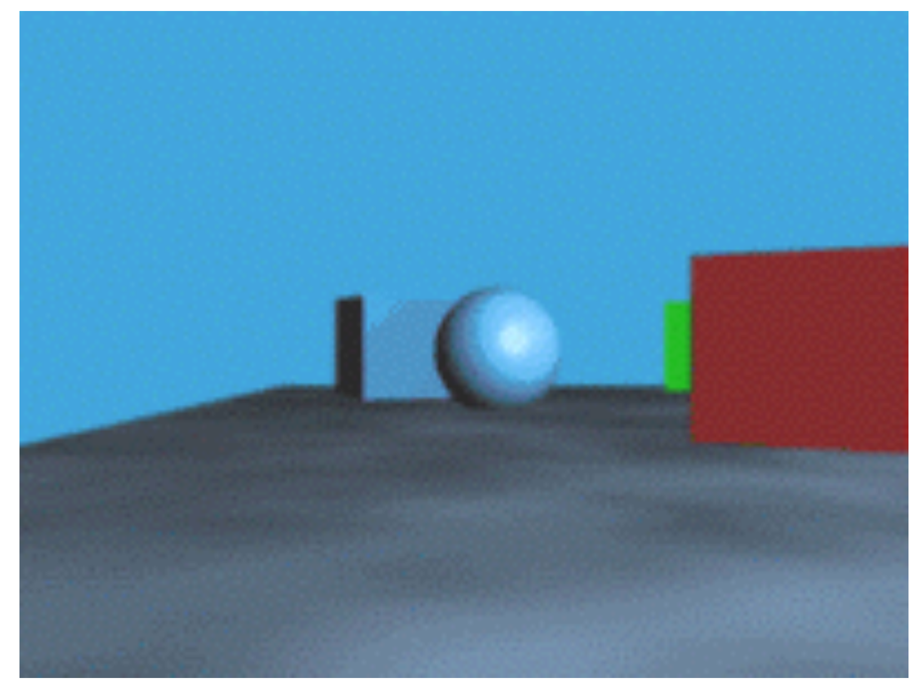
Outline

- Bightness constancy
- Aperture problem
- Small-motion assumption
- Motion segmentation

Biological importance of optical flow



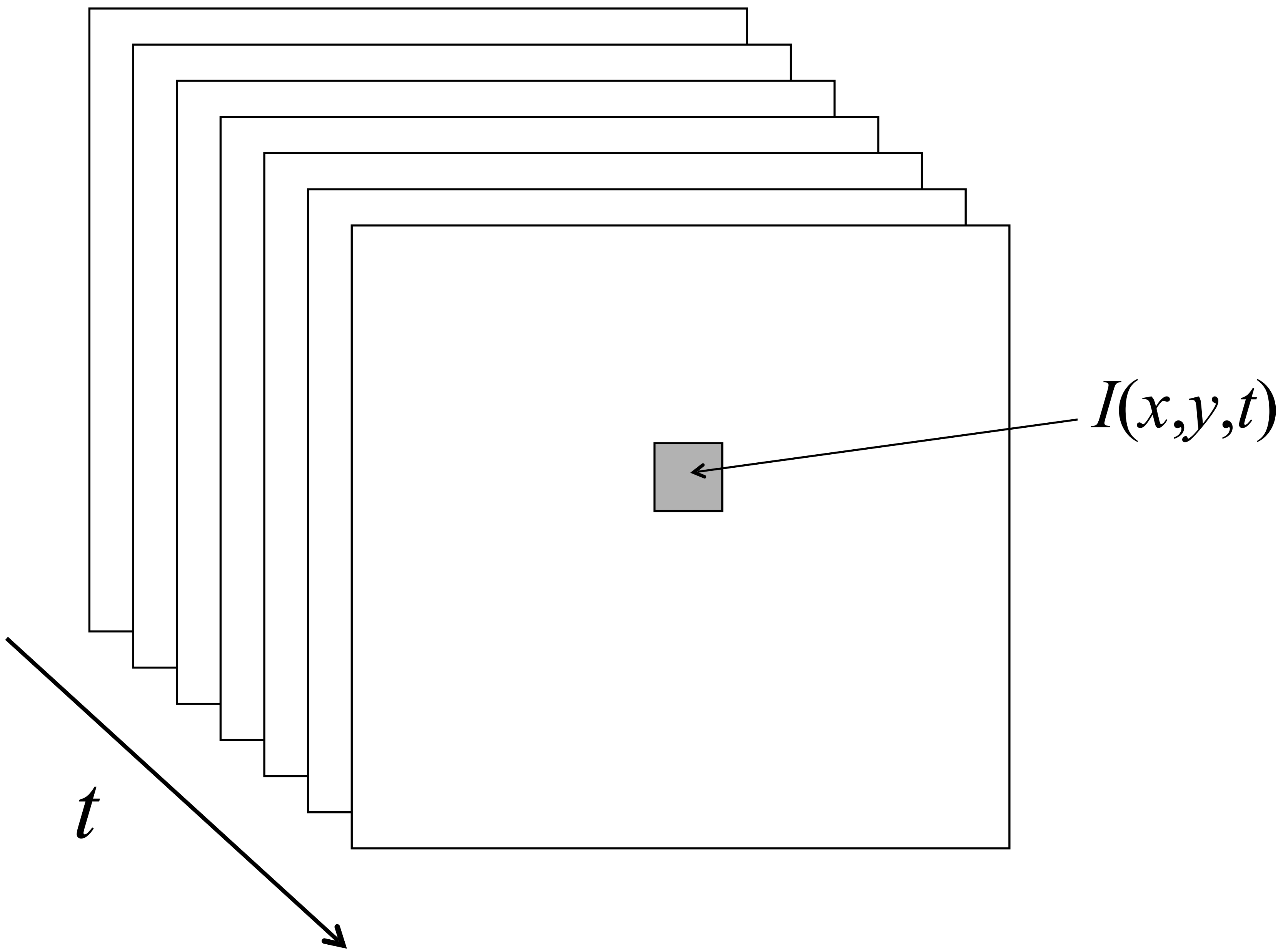
Time-to-contact



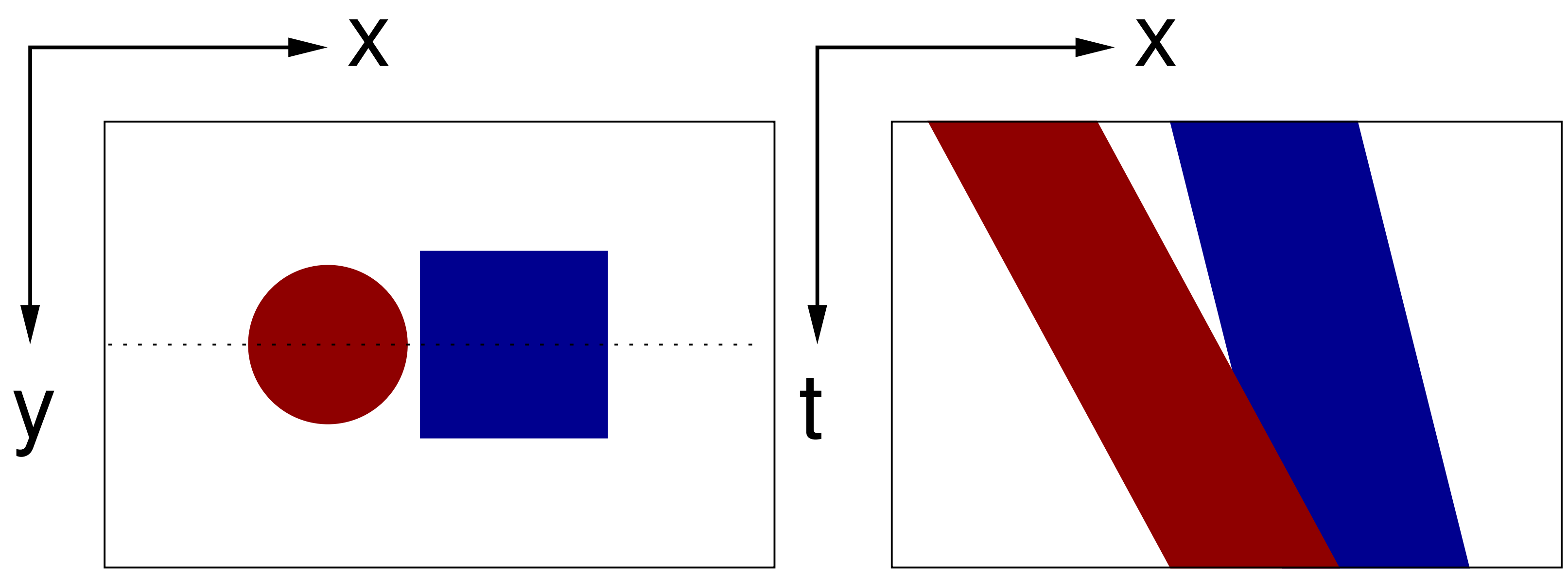
Parallax reveals depth

Importance of low-level motion

Videos as spacetime cubes



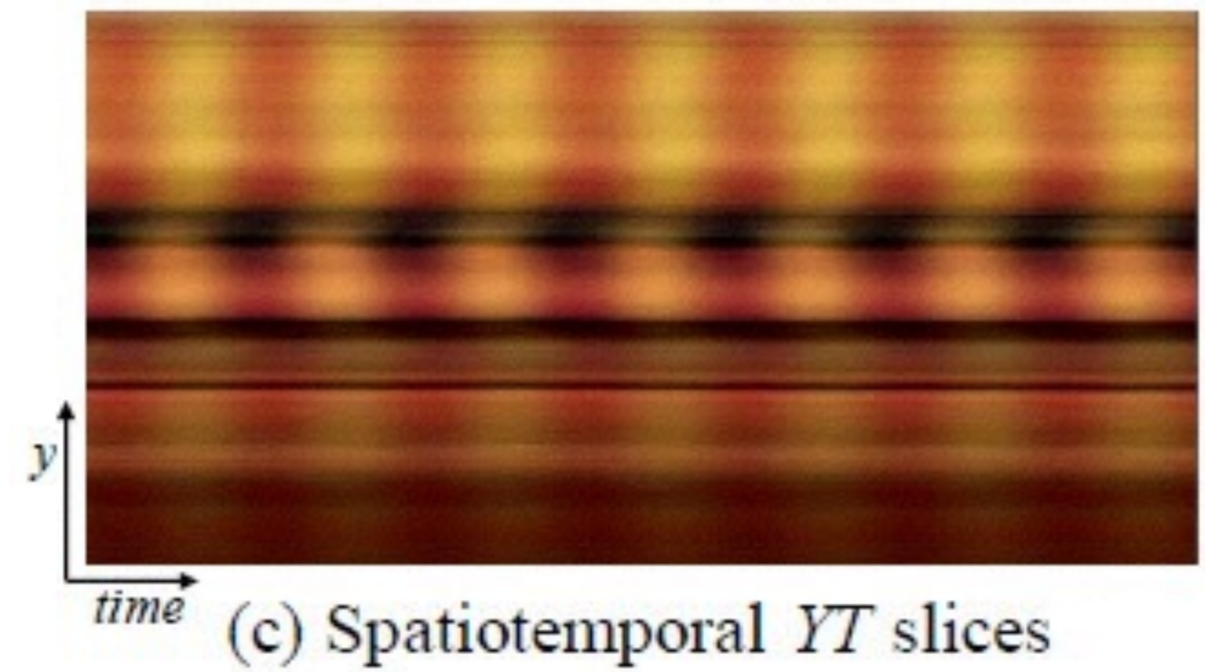
Visualizing spacetime cubes



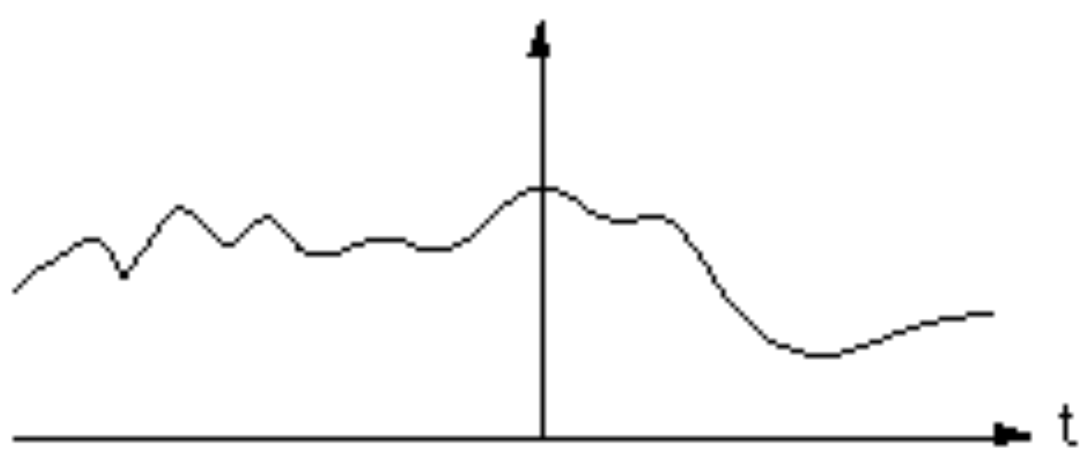
In this example, the circle is in front of the square and the camera is moving horigontally to the left

Digression: visualizing space-time cube

Plot I(x,y,t) for a fixed t

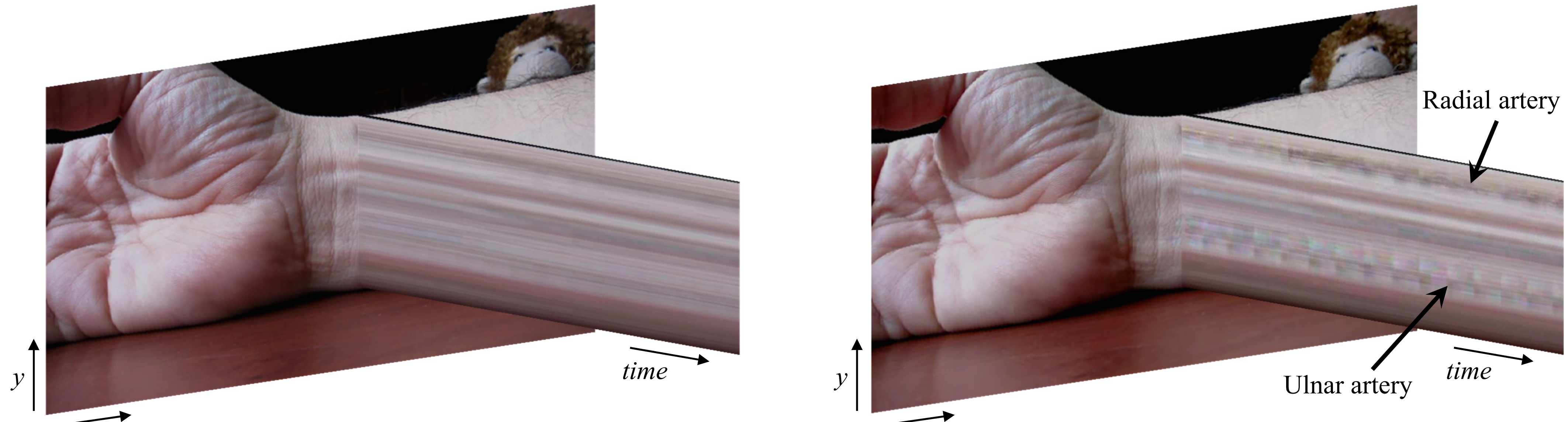


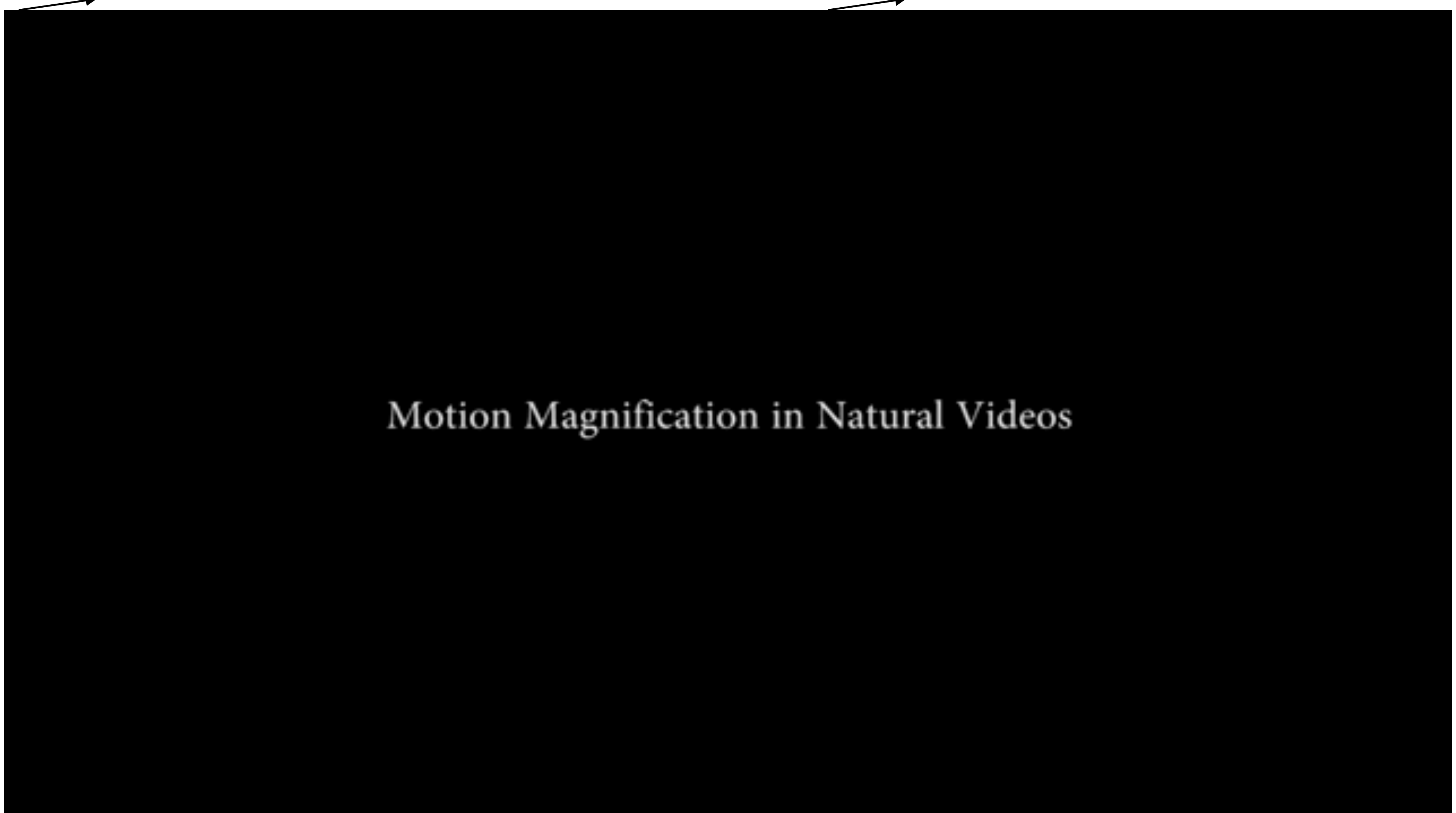
Plot I(x,y,t) for a fixed x



Plot I(x,y,t) for a fixed (x,y)

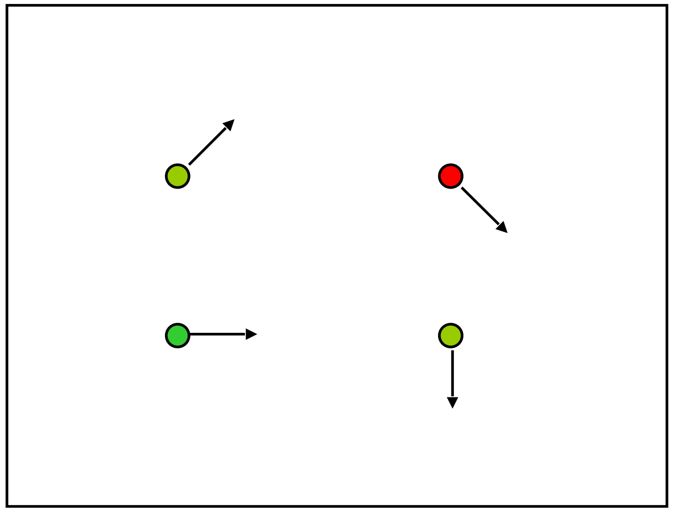
Amplifying temporal signals



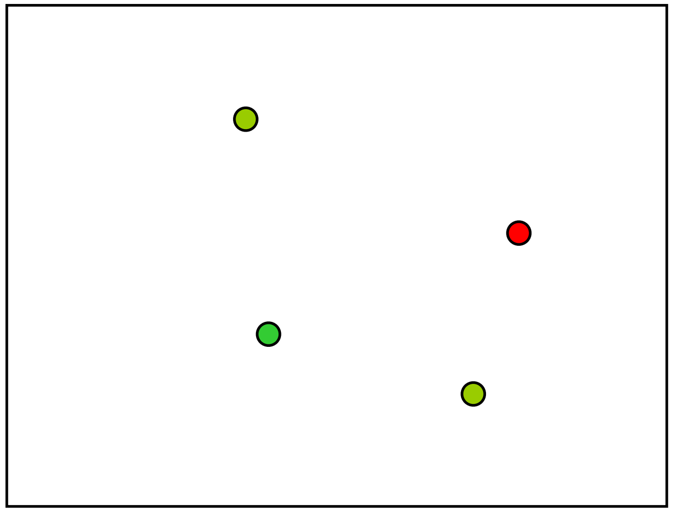


Eulerian Video Magnification for Revealing Subtle Changes in the World

Problem Definition: Optical Flow



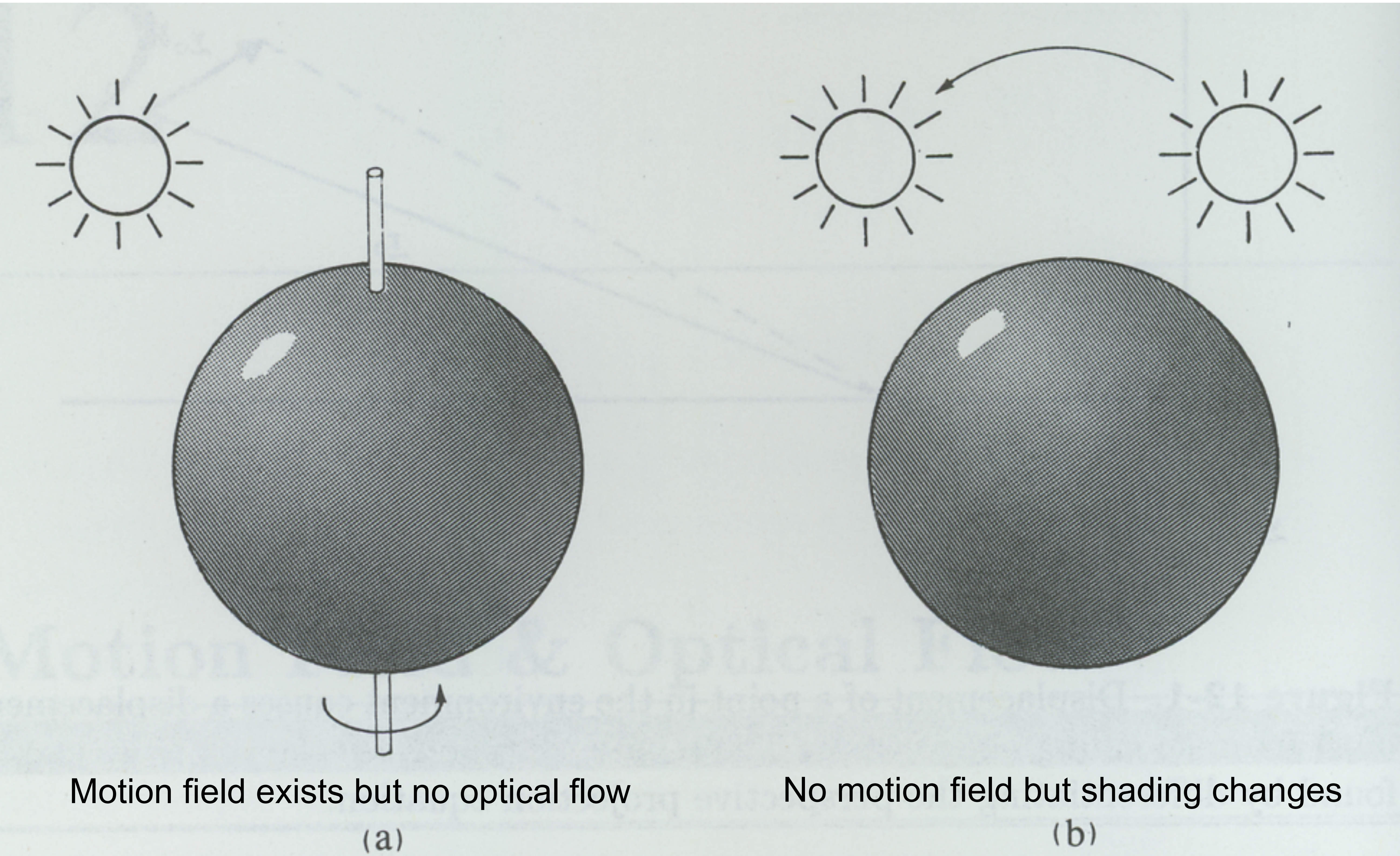
$H(x, y)$



$I(x, y)$

- How to estimate pixel motion from image H to image I?
  - Find pixel correspondences
    - Given a pixel in H, look for nearby pixels of the same color in I
- Key assumption
  - **color constancy**: a point in H looks "the same" in image I
    - For grayscale images, this is **brightness constancy**

Caution:
2D measured optical flow ≠ 3D scene flow



Brightness constancy

$$I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t) = 0$$

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \approx 0$$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad \text{where} \quad u = \frac{\Delta x}{\Delta t}, v = \frac{\Delta y}{\Delta t}$$

$$\nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \frac{\partial I}{\partial t} = 0$$

Brightness constancy equation gives us:

1) a constraint on flow vector (u,v)
2) a linear approximation of pixel error

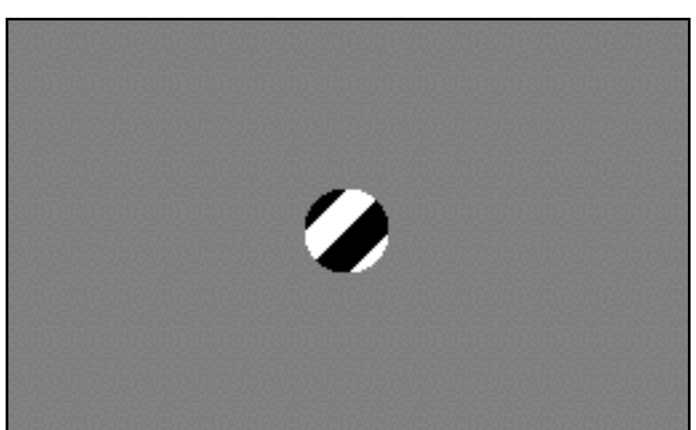
Aperature problem

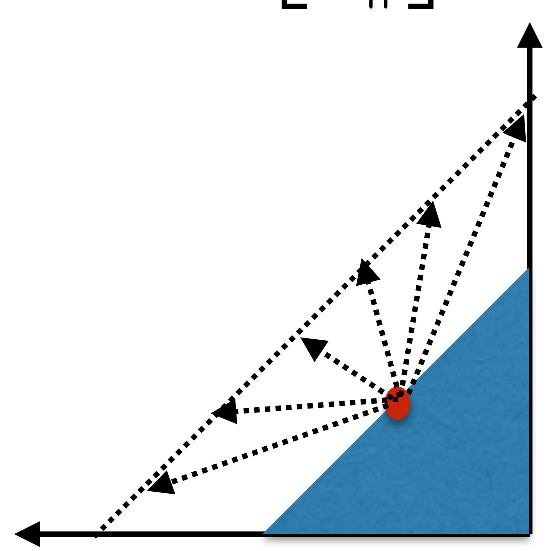
We can only determine flow in direction parallel to gradient

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_\perp \\ v_\perp \end{bmatrix} + \begin{bmatrix} u_{||} \\ v_{||} \end{bmatrix}$$

$$\nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \nabla I \cdot \begin{bmatrix} u_\perp \\ v_\perp \end{bmatrix} + \nabla I \cdot \begin{bmatrix} u_{||} \\ v_{||} \end{bmatrix}$$

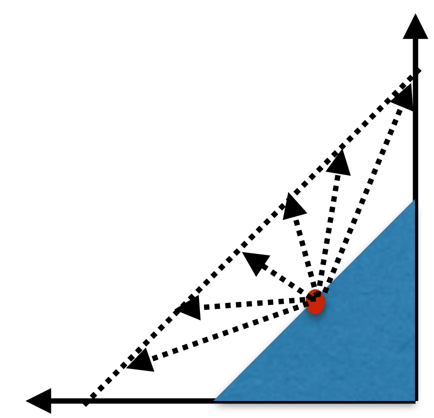
$$= \nabla I \cdot \begin{bmatrix} u_{||} \\ v_{||} \end{bmatrix}$$





Challenges

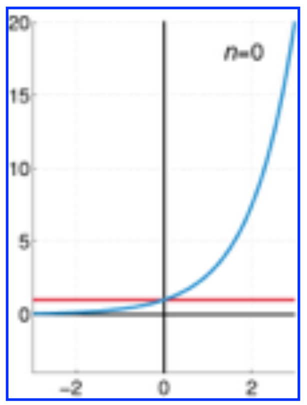
- Aperture problem



Soln to brightness constancy equation may not be unique

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

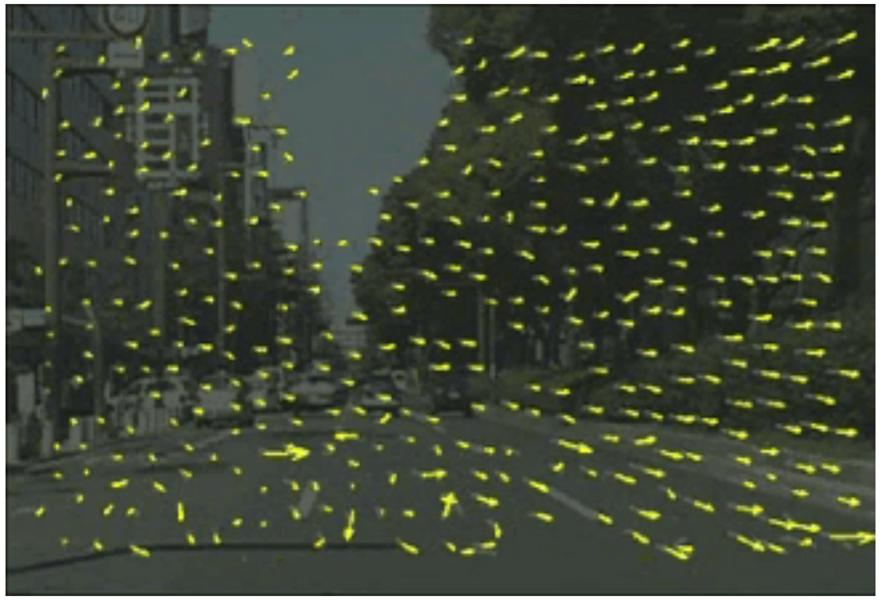
- Small motion assumption



First-order taylor approximation does not hold for large motions

$$\nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \frac{\partial I}{\partial t} = 0$$

Soln for aperture problem

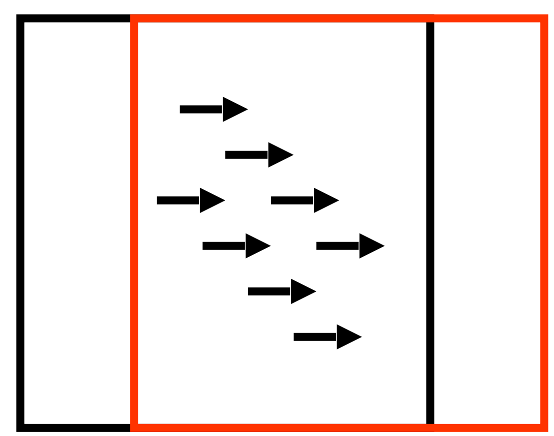


1. Don’t try to estimate flow at unreliable points (sparse flow)

2. Assume neighboring flow vectors are similar (enforce *spatial smoothness* in dense flow feild)

Simple approach: assume flow is constant over a neighborhood

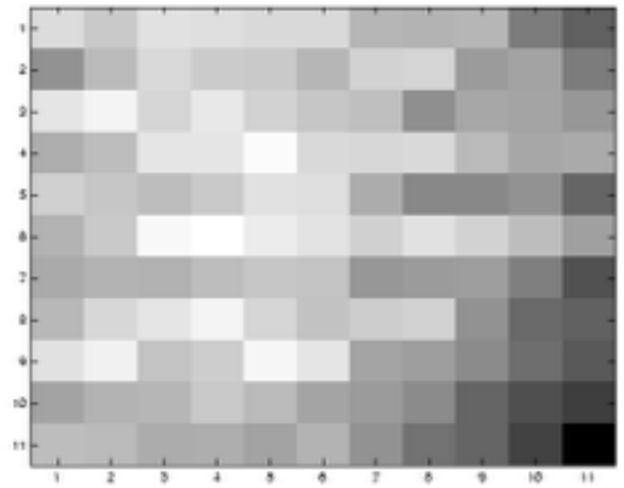
$$\min_{u,v} \sum_{x,y \in W} \Big(I_2(x+u, y+v) - I_1(x,y) \Big)^2$$

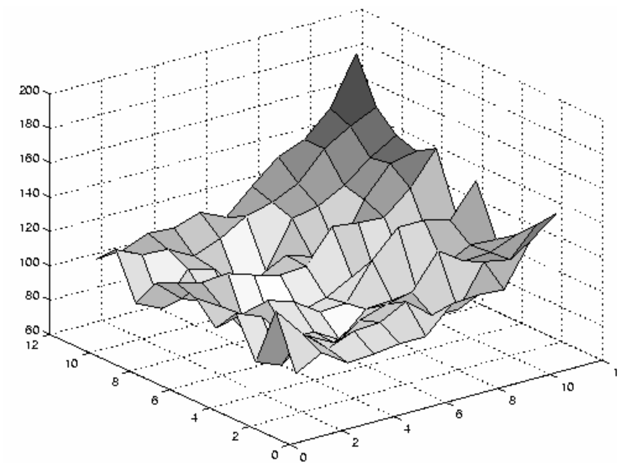


$u(x,y) = u$

$v(x,y) = v$

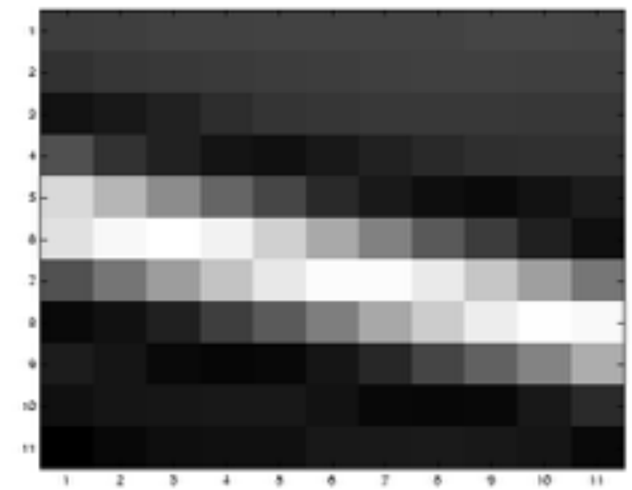
Low Texture Region - Bad

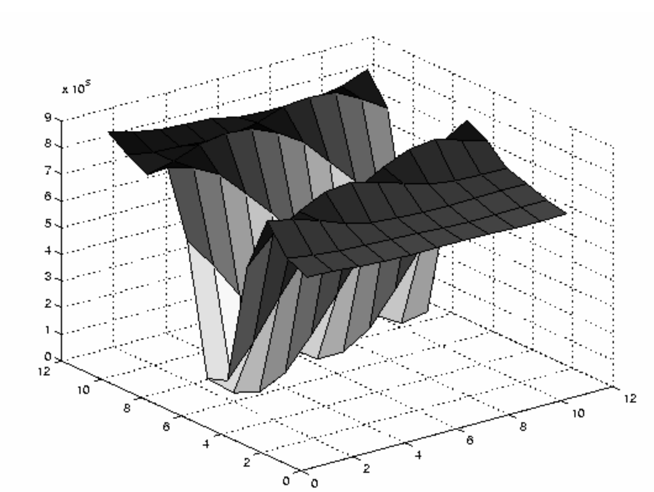




SSD surface

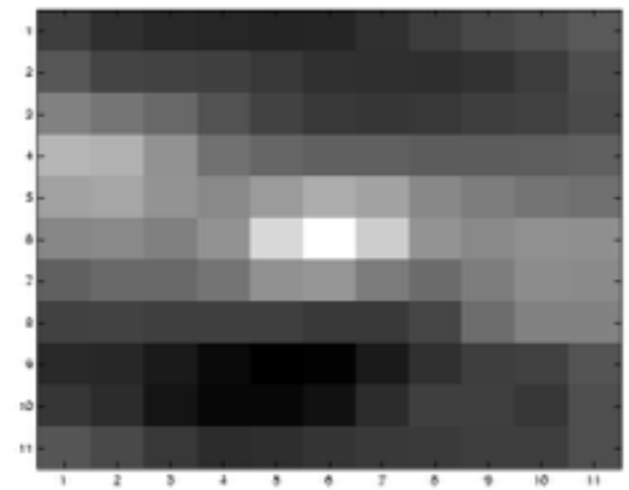
Edges – so,so (aperture problem)

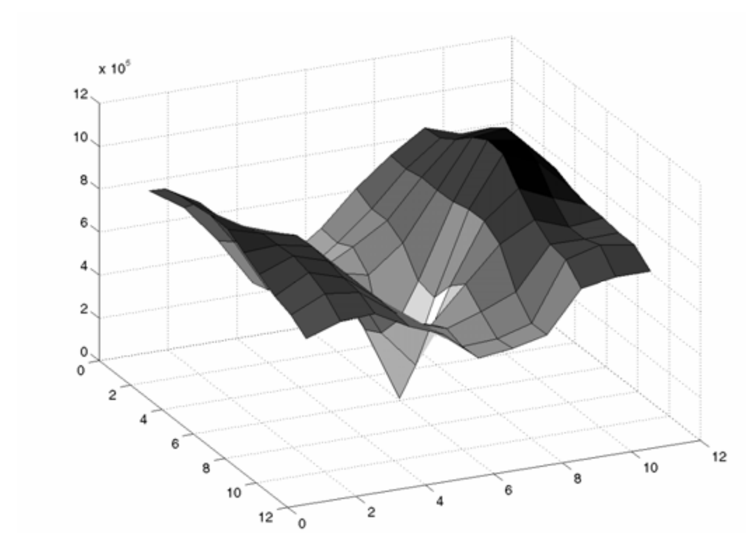




SSD surface

High Textured Region - Good





SSD surface

Sparse flow estimation (feature tracking)

1. User Harris corner score to find trackable patches

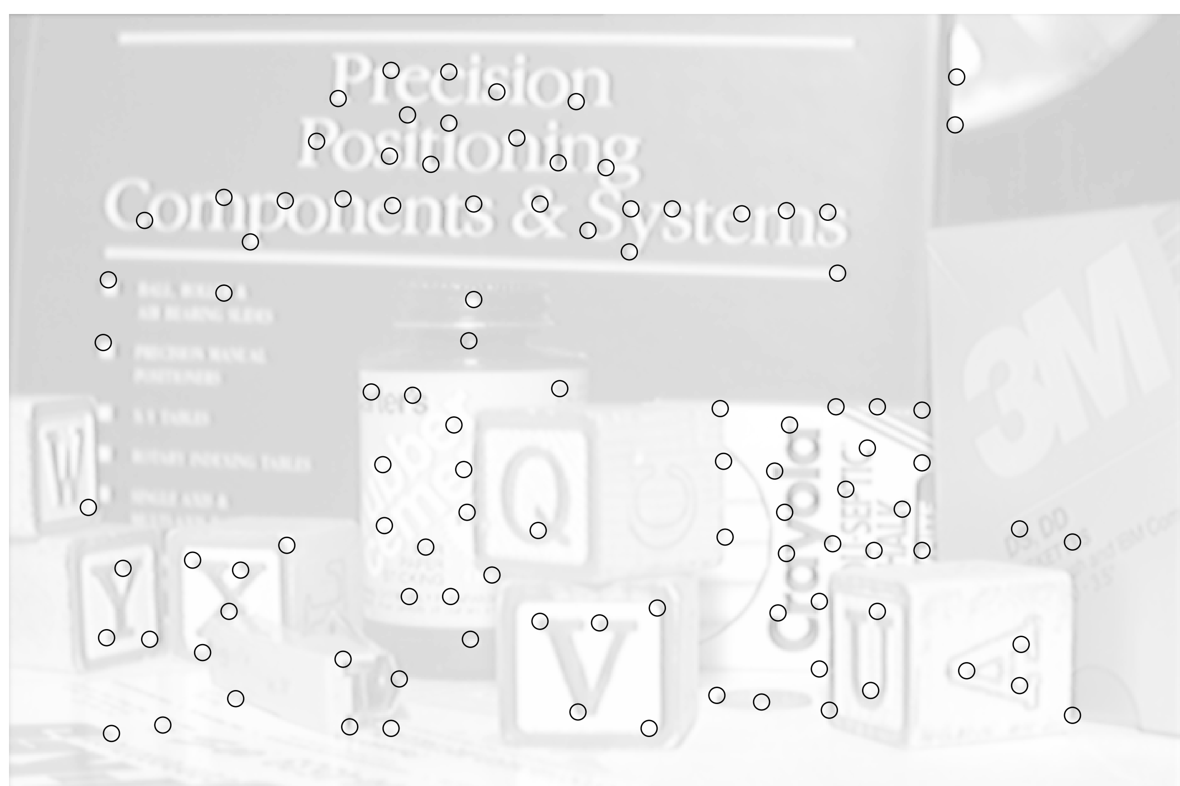
$$I_2(x+u, y+v) - I_1(x,y) \approx \nabla I(x,y) \begin{bmatrix} u \\ v \end{bmatrix} + I_t(x,y)$$

2. Appy Lucas Kanade on those patches

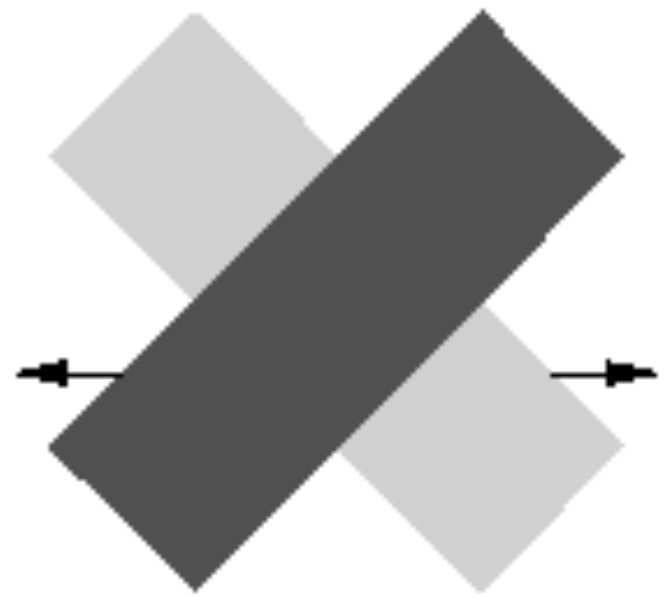
Good Features to Track

Jianbo Shi
Computer Science Department
Cornell University
Ithaca, NY 14853

Carlo Tomasi
Computer Science Department
Stanford University
Stanford, CA 94305



Local motion estimation is hard



Where does false “t-junctions” appear to move?

We’d like to integrate local signals globally

Dense flow (I)

$$E(\mathbf{p}) = \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})]^2$$

Apply Lucas Kanade on successive frames of a video sequence

Generalize translation to other 2D warps (affine, homographies,…)

Applications: mosaicing

Homography warp works for some cases (rotations, planar scenes).
We'll discuss a solution for others in a bit…

Dense flow (II)

Solve for global flow feild

$$\min_{\substack{u(x,y)\\v(x,y)}} \sum_{x,y} [I_2(x+u(x,y), y+v(x,y)) - I_1(x,y)]^2$$

Aside: continuous case

$$\min_{u,v} \int\int \big(I_2(x+u, y+v) - I_1(x,y)\big)^2 dxdy$$

Formal math is known as calculus of variations (we're minimizing over the *space of functions*)

https://en.wikipedia.org/wiki/Calculus_of_variations

Dense *variational* flow

If we assume small motions….

$$I_2(x+u, y+v) - I_1(x,y) \approx \nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} + I_t$$

$$\min_{u,v} \int \int (\nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} + I_t)^2 dxdy$$

above is “shorthand” for…

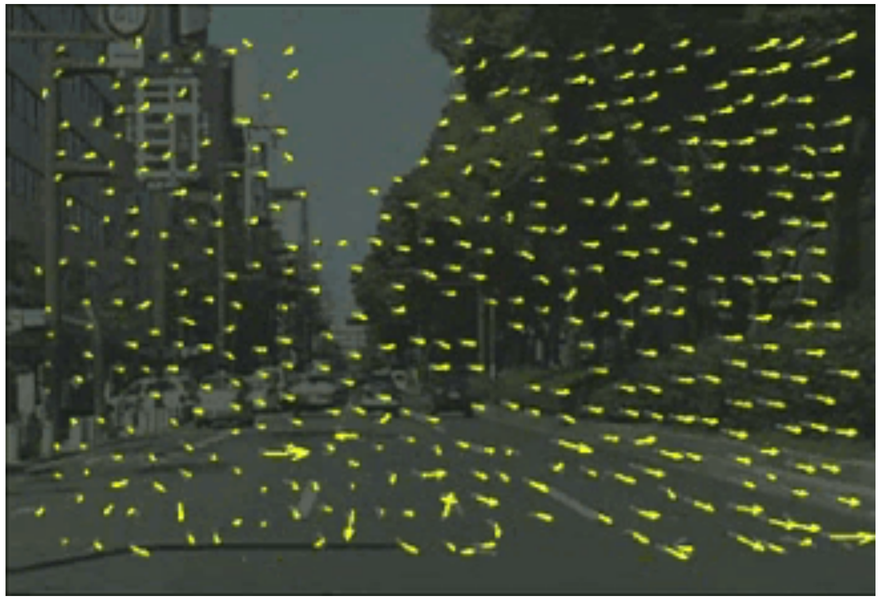
$$\min_{\substack{u(x,y) \\ v(x,y)}} \sum_{x,y} \left[\nabla I(x,y) \cdot \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} + I_t(x,y) \right]^2$$

Spatial regularization

Penalize differences in nearby flow vectors

$$\min_{u,v} E_{intensity} + E_{smooth}$$

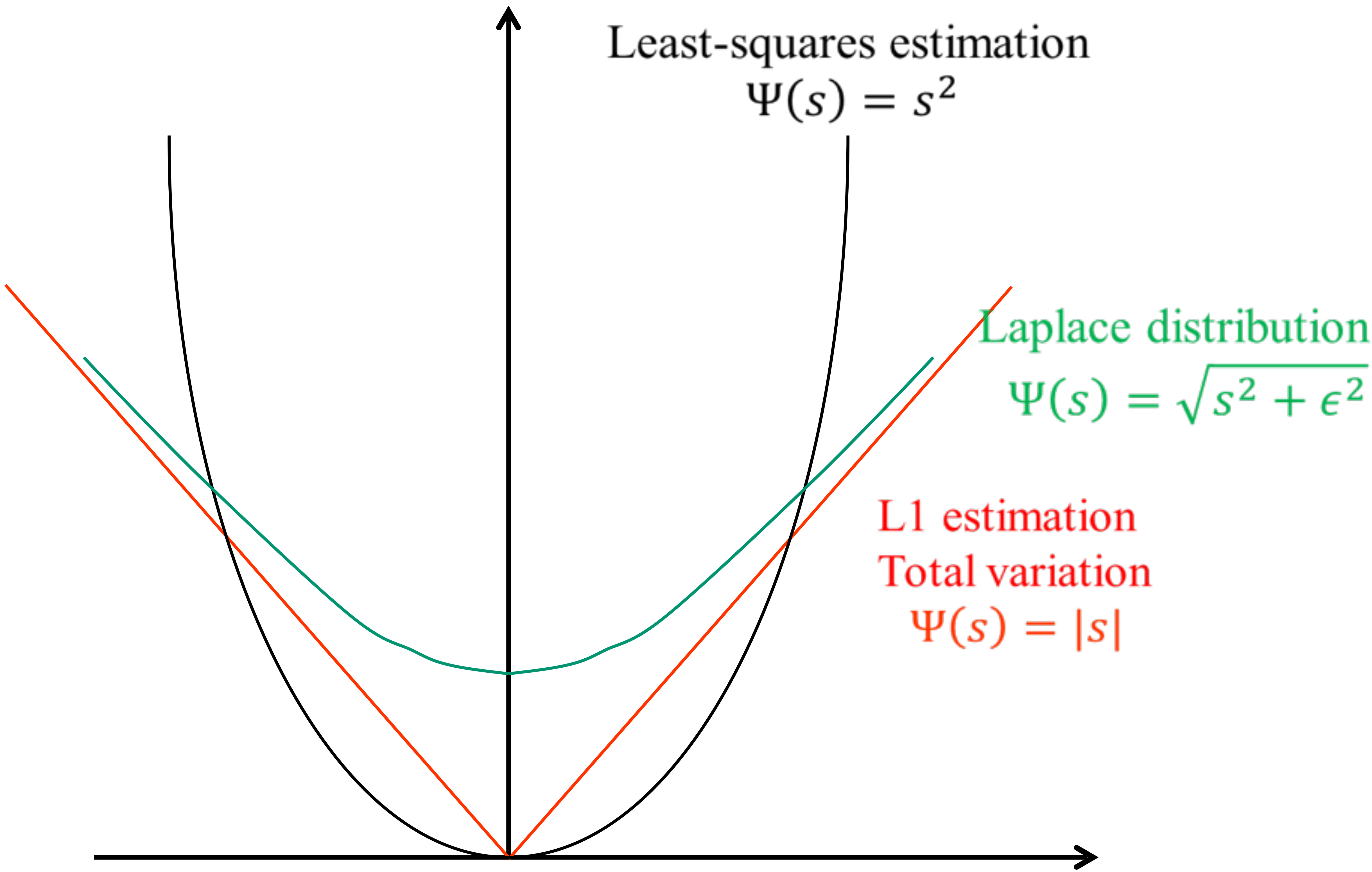
$$E_{intensity}(u,v) = \int\int (\nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} + I_t)^2 dxdy \qquad E_{smooth}(u,v) = \int\int ||\nabla u||^2 + ||\nabla v||^2 dxdy$$



1. Unknowns (u,v) appear quadratically in above expression => discretize above and solve for them with a giant linear system

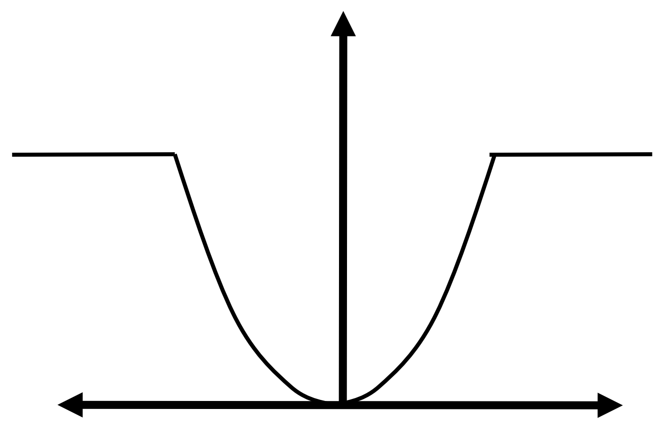
2. Challenge: outliers will overwhelm squared error term

Robust statistics



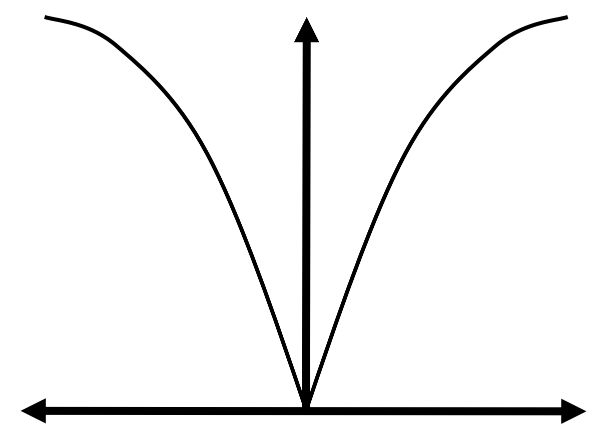
Energy function(u,v) is still convex (and globally optimizable with local search)

Robust statistics (cont'd)



truncated quadratic

$$\rho_{\alpha,\lambda}(x) = \begin{cases} \lambda x^2 & \text{if } |x| < \frac{\sqrt{\alpha}}{\sqrt{\lambda}} \\ \alpha & \text{otherwise} \end{cases}$$



lorentzian

$$\rho_\sigma(x) = log\left(1 + \frac{1}{2}(\frac{x}{\sigma})^2\right)$$

Energy function(u,v) not convex

Robust variational optical flow

$$\min_{u,v} \int\int \rho(I_2(x+u, y+v) - I_1(x,y)) + \rho(||\nabla u||) + \rho(||\nabla v||) dxdy$$

first image quadratic flow lorentzian flow detected outliers

Reference

- Black, M. J. and Anandan, P., A framework for the robust estimation of optical flow, *Fourth International Conf. on Computer Vision* (ICCV), 1993, pp. 231-236 http://www.cs.washington.edu/education/courses/576/03sp/readings/black93.pdf

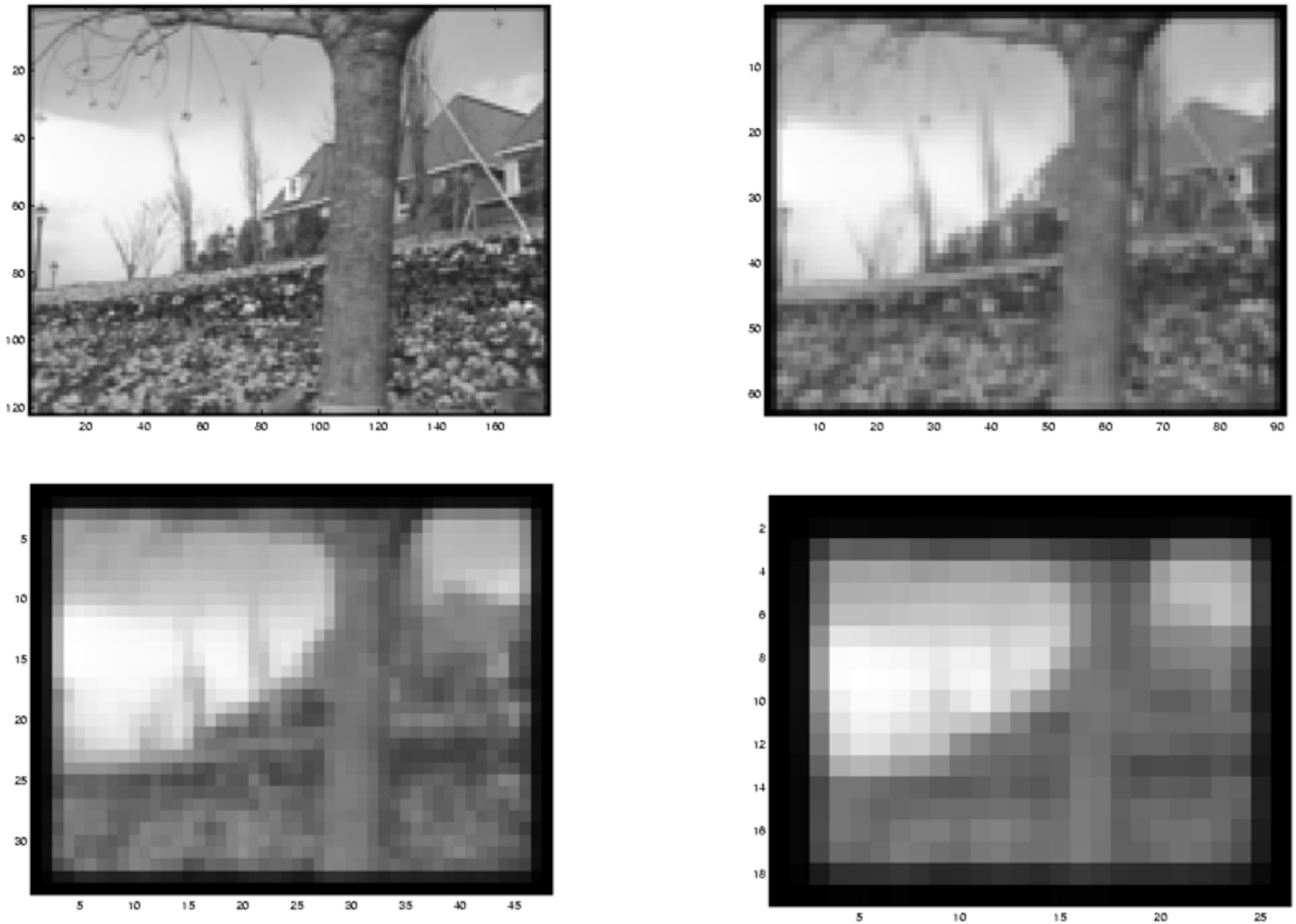
Outline

- Bightness constancy
- Aperture problem (sparse flow, spatial regularization)
- Small-motion assumption
- Motion segmentation

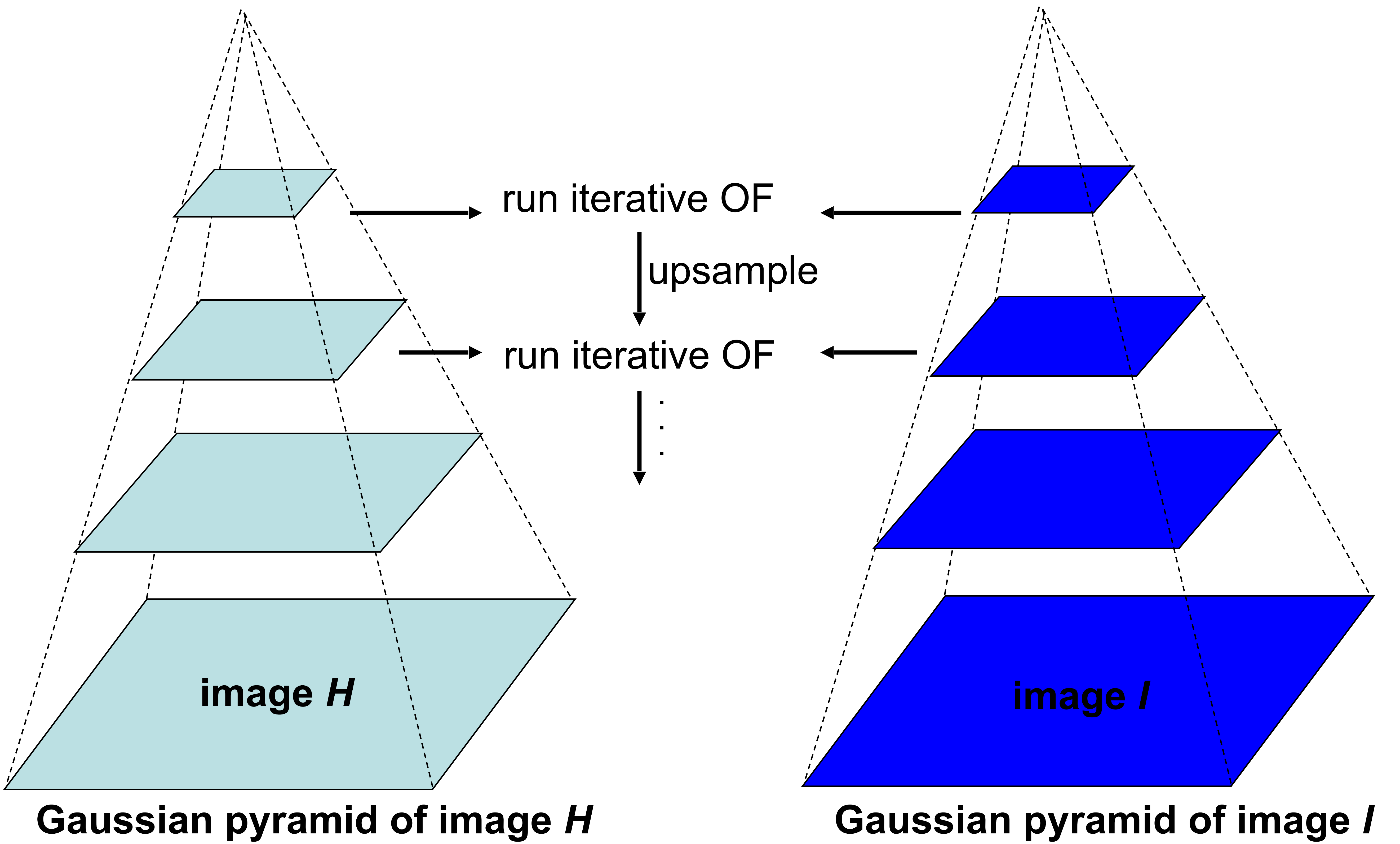
Revisiting the Small Motion Assumption

- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2nd order terms dominate)
  - How might we solve this problem?

Reduce the Resolution!



Soln 1: Coarse-to-fine Optical Flow

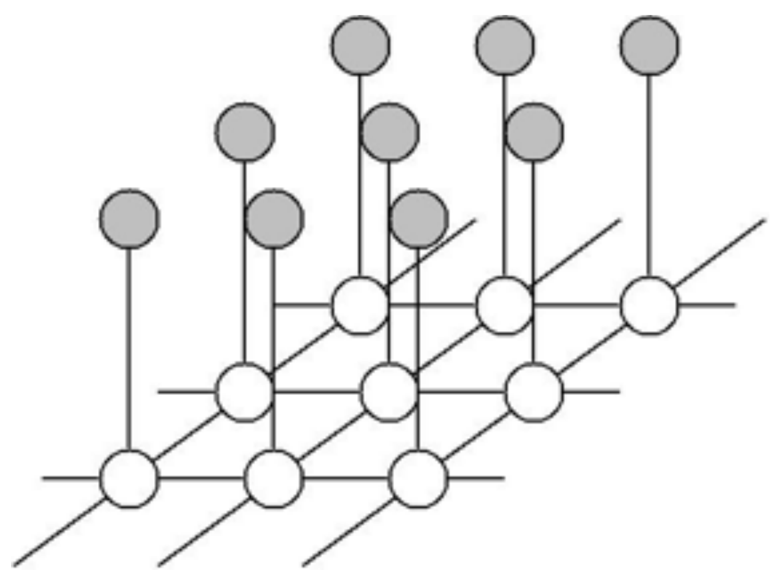


Soln 2: discrete optical flow estimation

$$u_i \in \{-5 \dots 5\}$$
$$v_i \in \{-5 \dots 5\}$$
$$z_i = (u_i, v_i)$$

$$\phi_i(z_i) = \rho(||I_2(x_i + u_i, y_i + v_i) - I(x_i, y_i)||)$$
$$\psi_{ij}(z_i, z_j) = \rho(u_i - u_j, v_i - v_j)$$

$$E(z) = \sum_{i \in V} \phi_i(z_i) + \sum_{ij \in E} \psi_{ij}(z_i, z_j)$$



Discrete Markov Random Feild (MRF) with pixel-grid graph G=(V,E)

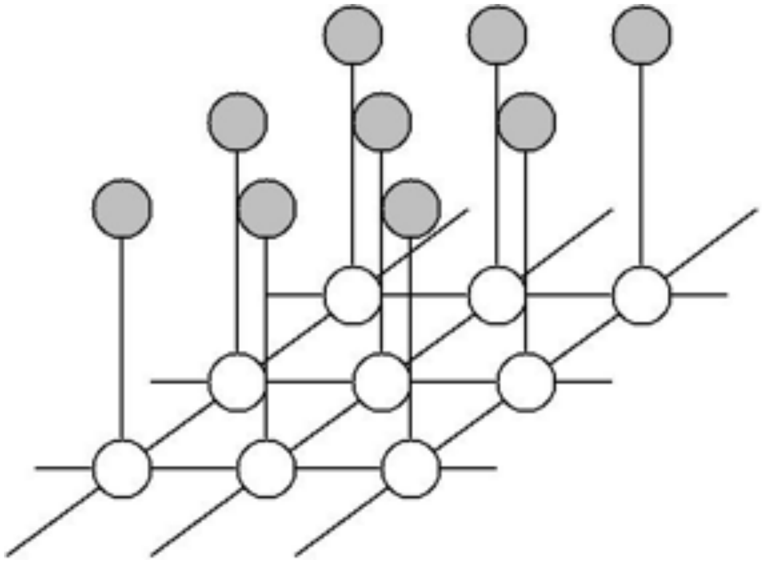
A Database and Evaluation Methodology for Optical Flow

Simon Baker · Daniel Scharstein · J.P. Lewis ·
Stefan Roth · Michael J. Black · Richard Szeliski

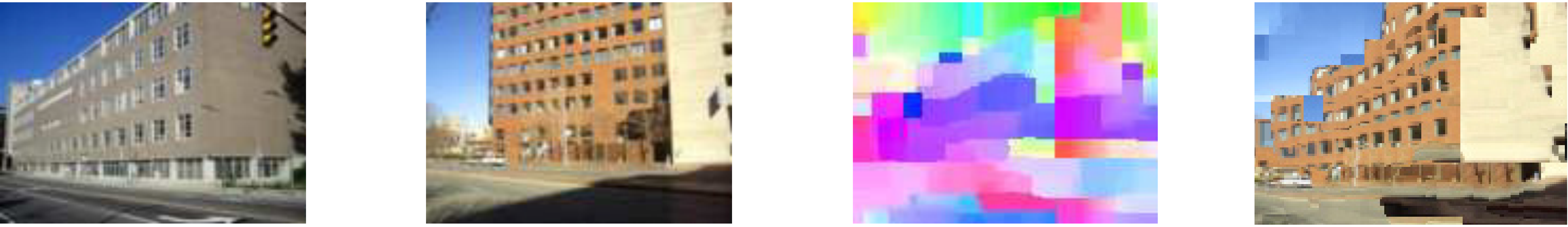
Example: SIFTFlow

Measure local appearances of patches using SIFT descriptors

Turns out that this can be used to align images of different scenes!



Liu et al, PAMI 2011



Allows us to do nearest-neighbor label transfer for scene analysis

Outline

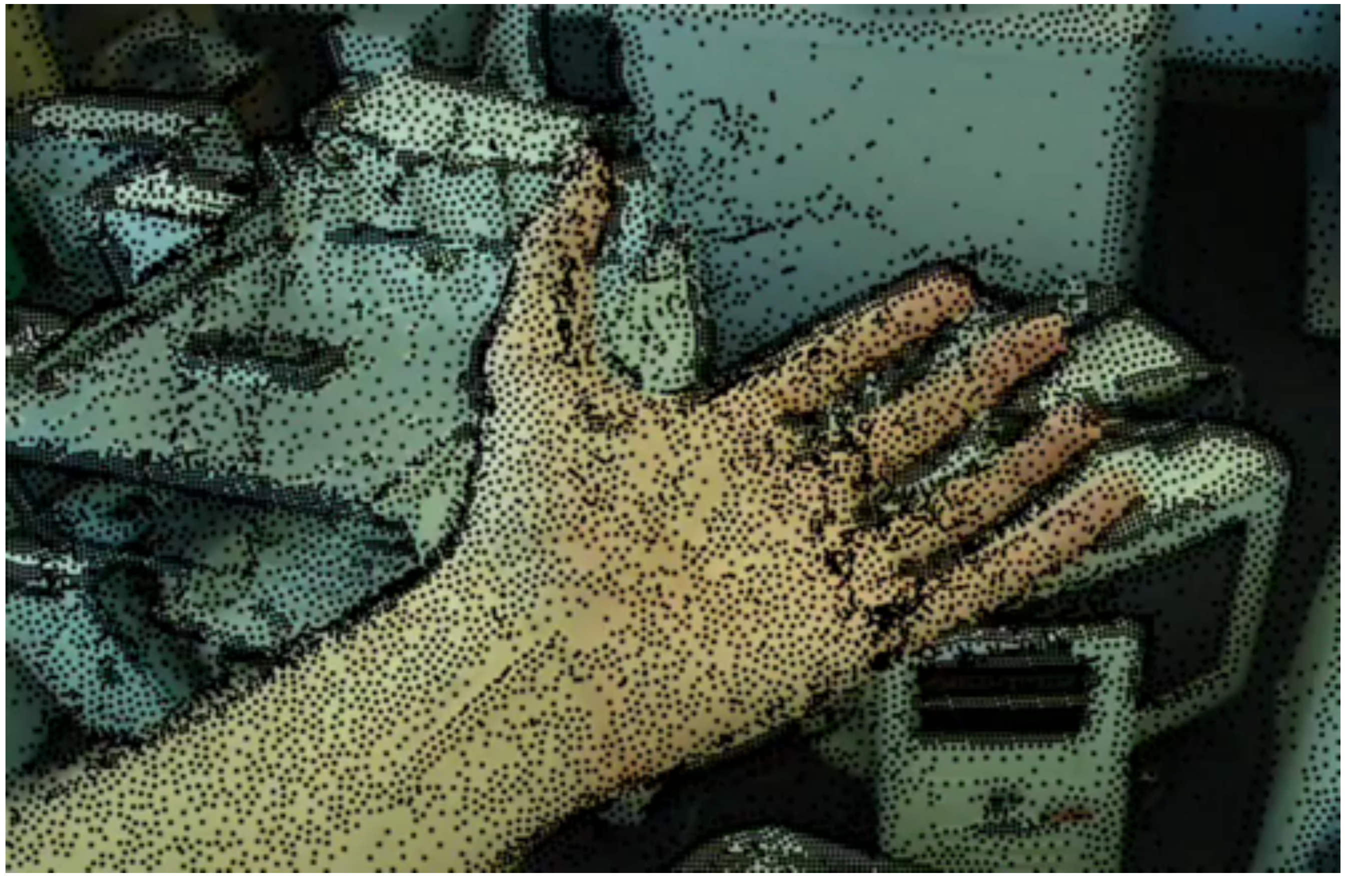
- Bightness constancy
- Aperture problem (sparse flow, spatial regularization)
- Small-motion assumption (coarse-to-fine, discrete optimization)
- Motion segmentation

[Some remaining challenges]

Remaining challenges: long-term optical flow

Combine long-term sparse feature tracking with variational flow regularization

(http://rvsn.csail.mit.edu/pv/)



Note the difficulty in getting regularization “right”!

Remaining challenges: small things that move fast

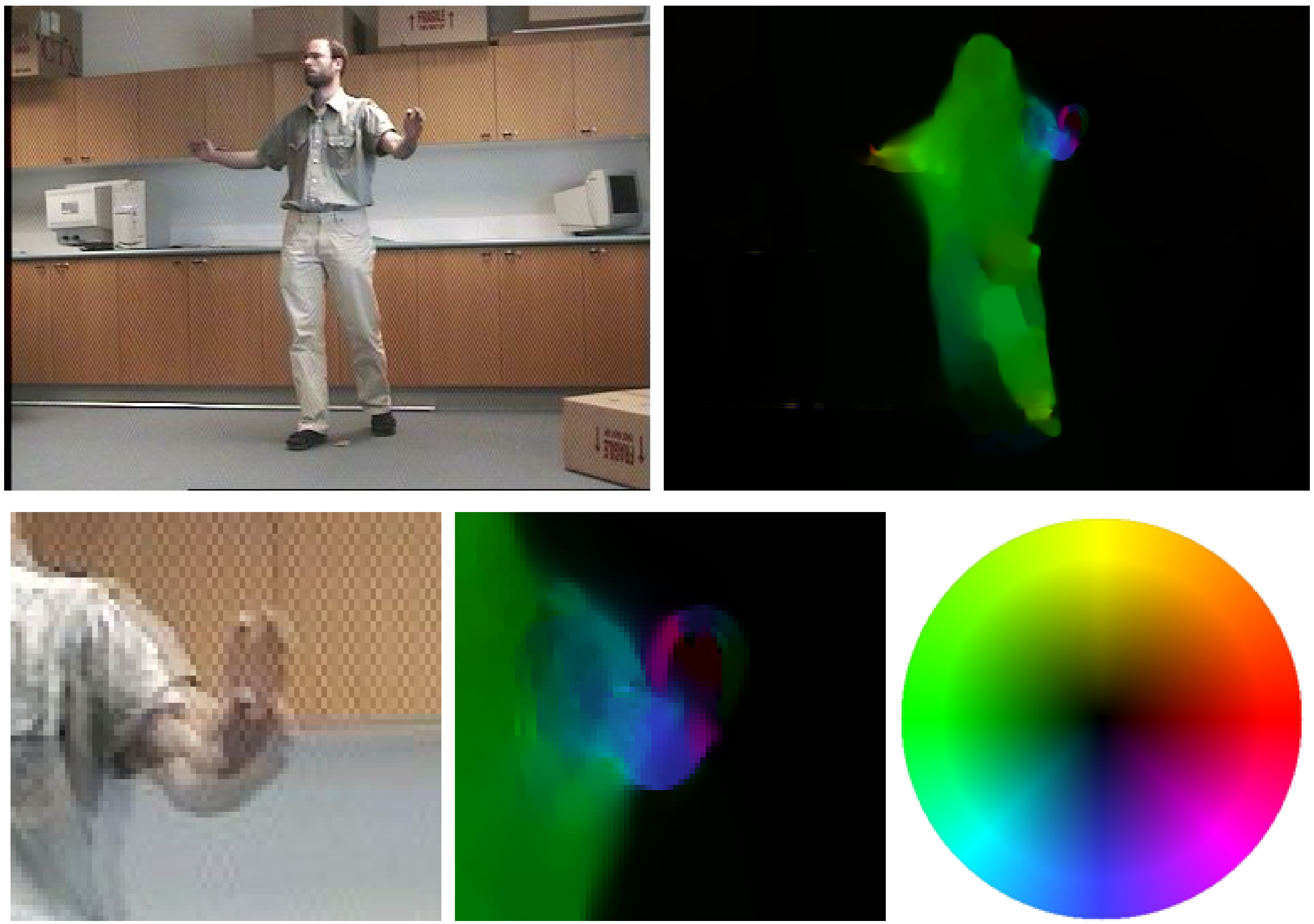
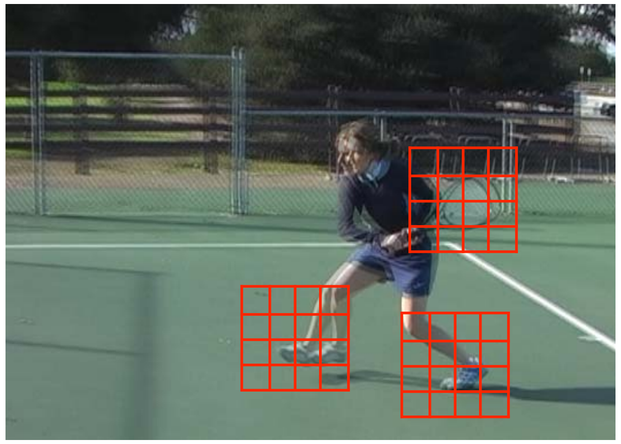
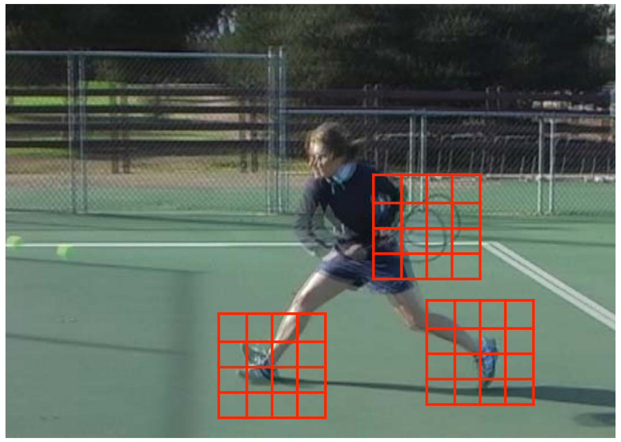


Figure 1. **Top row:** Image of a sequence where the person is stepping forward and moving his hands. The optical flow estimated with the method from [4] is quite accurate for the main body and the legs, but the hands are not accurately captured. **Bottom row,**

Large Displacement Optical Flow*

Thomas Brox[1] Christoph Bregler[2] Jitendra Malik[1]
[1]University of California, Berkeley
Berkeley, CA, 94720, USA
{brox,malik}@eecs.berkeley.edu
[2]Courant Institute, New York University
New York, NY, 10003, USA
bregler@courant.nyu.edu

Estimate dense or sparse correspondences across 2 frames with classic descriptor matching

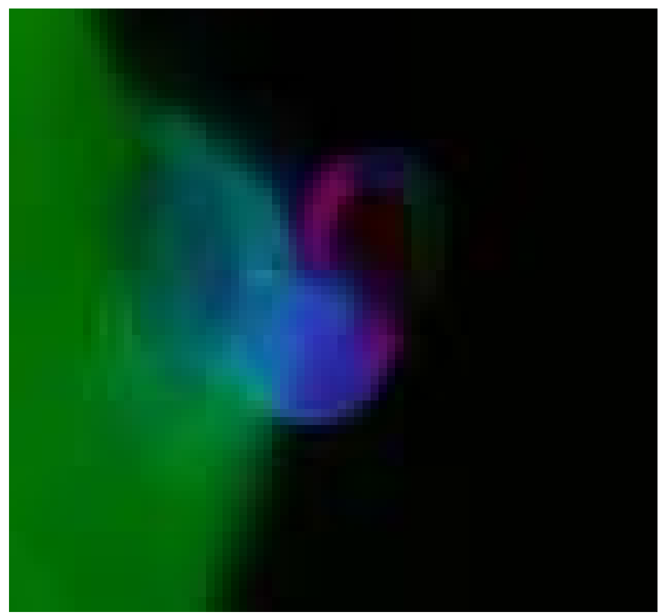



Set of matchable points and estimated offsets: $\{(x_i, y_i, u_i, v_i)\}$

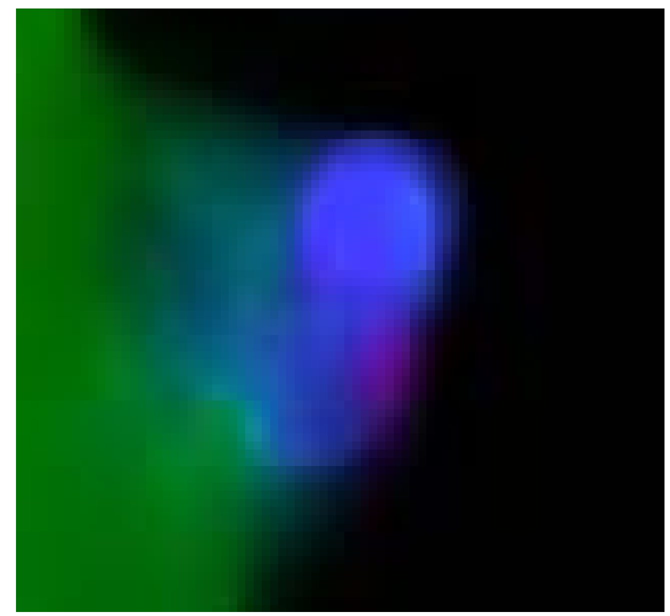
$$E_{match}(u,v) = \sum_i (u(x_i,y_i) - u_i)^2 + (v(x_i,y_i) - v_i)^2$$

$$\min_{u,v} E_{intensity} + E_{smooth} + E_{match}$$

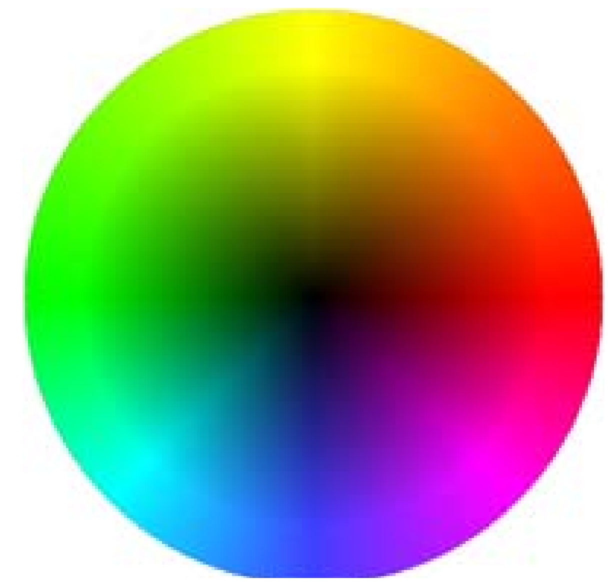
Examples



no E_{match}

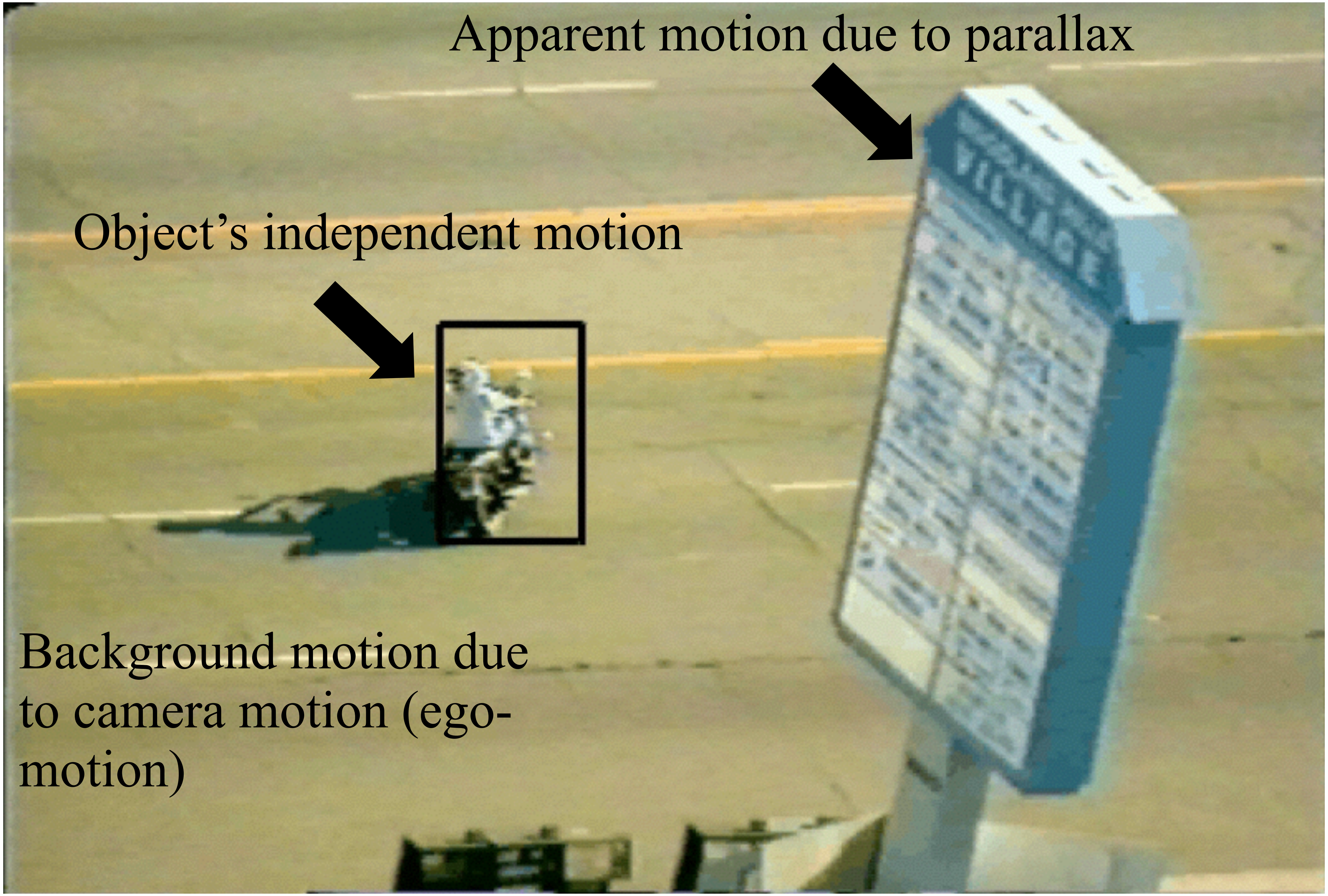


with E_{match}



Outline

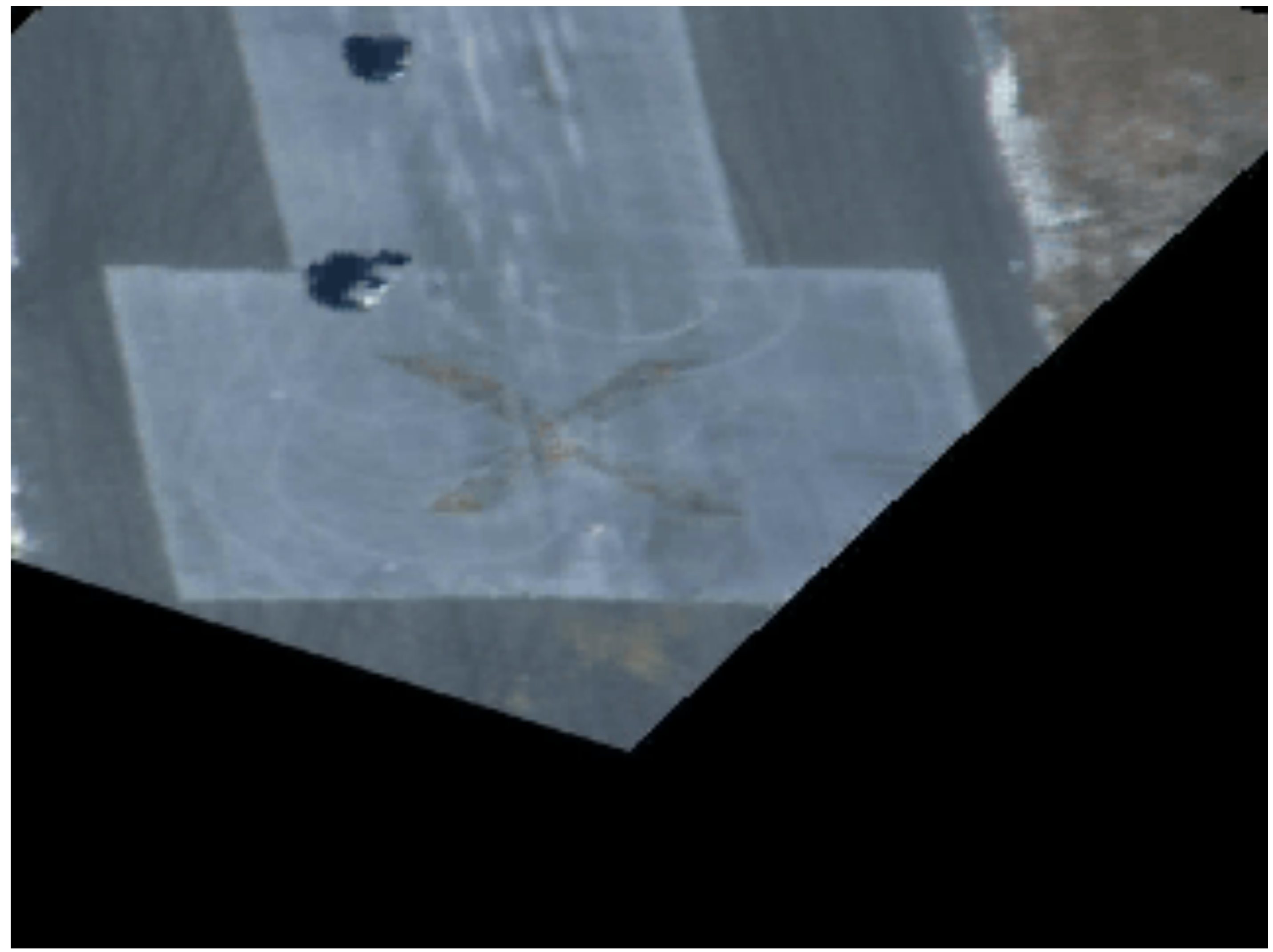
- Bightness constancy
- Aperture problem (sparse flow, spatial regularization)
- Small-motion assumption (coarse-to-fine, discrete optimization)
- Motion segmentation


Apparent motion due to parallax
Object’s independent motion
Background motion due to camera motion (ego-motion)
VILLAGE

Motion segmentation (I): robustly estimate dominant motion

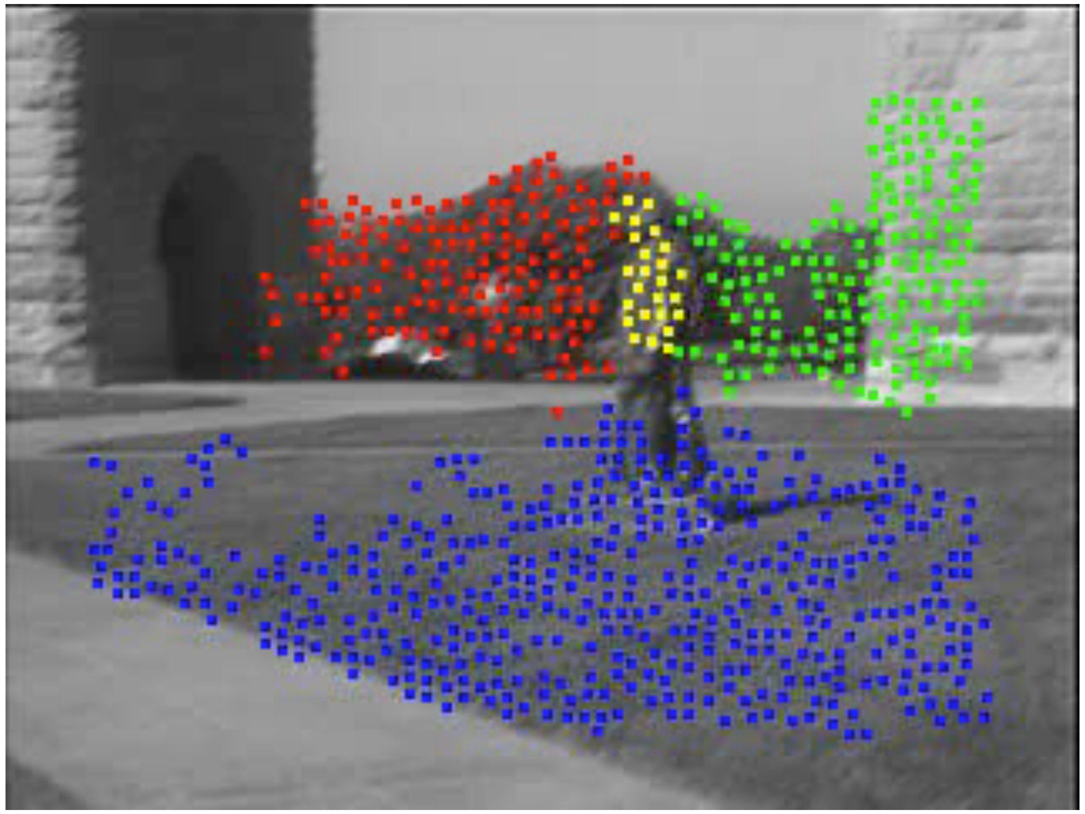
1. Assume parametric warp (typically homography)
2. Treat moving/non-planar objects as outliers in robust error function

$$E(\mathbf{p}) = \sum_{\mathbf{x}} \rho(I(\mathbf{W}(\mathbf{x};\mathbf{p}))) - T(\mathbf{x}))$$



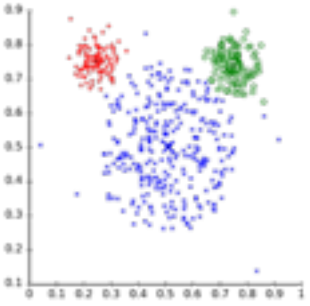
Motion segmentation (II)

Treat as clustering problem



1. Obtain an initial estimate of flow (sparse or dense)
2. Cluster pixels using feature vectors (consisting of flow, RGB, etc.)

Generalize K-means to fit a parametric model (e.g., affine warp) rather than a centroid

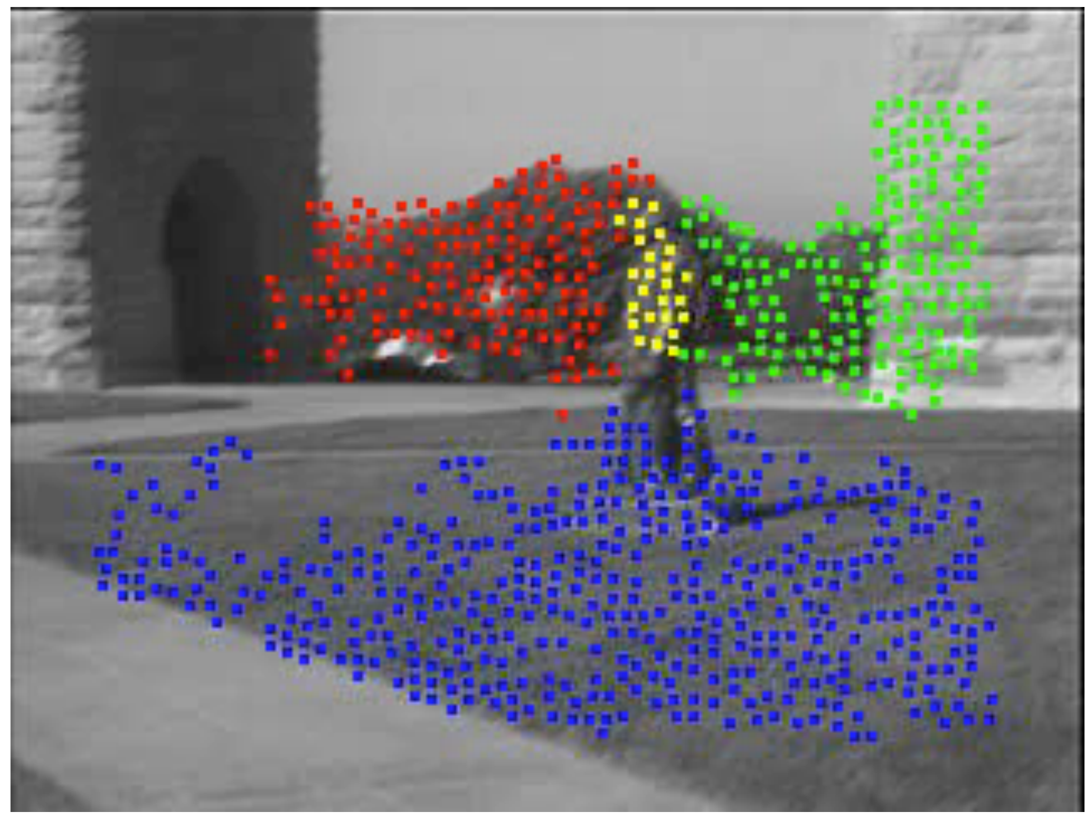


Weiss & Adelson, CVPR 96

Uses “soft” K-means or EM algorithm

Motion segmentation (II)

Treat as clustering problem

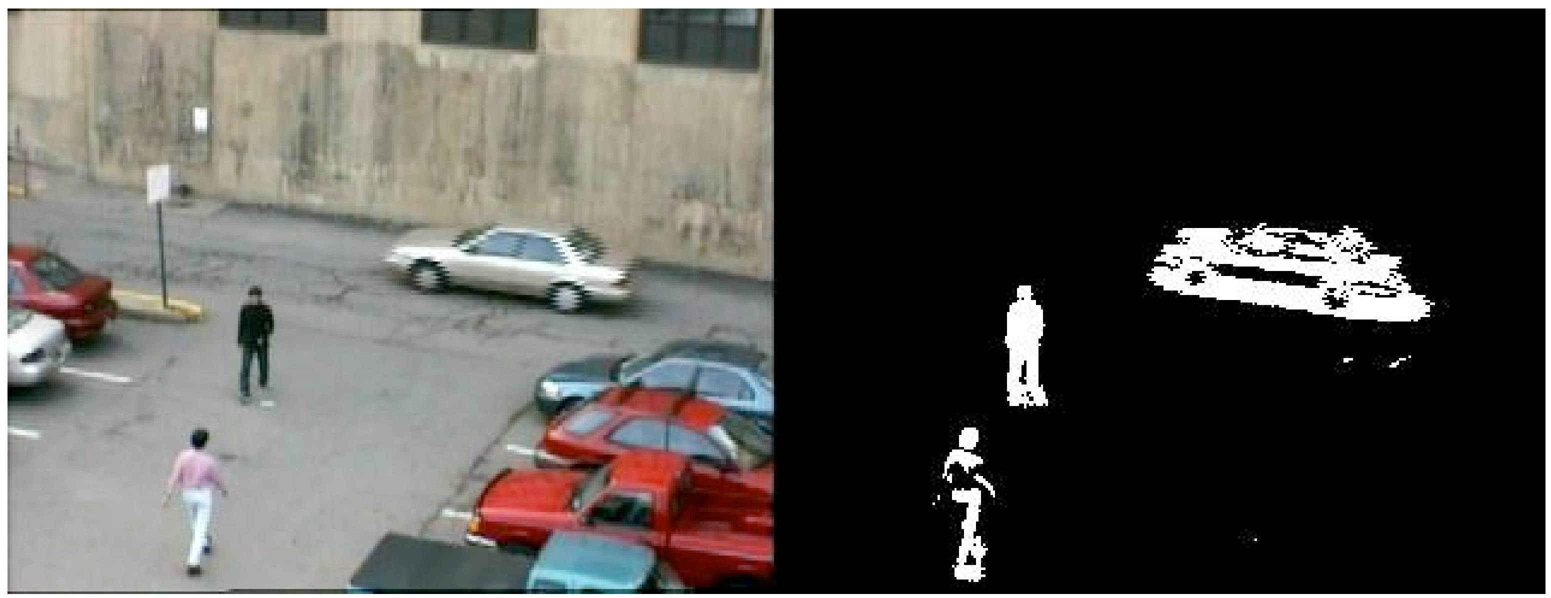


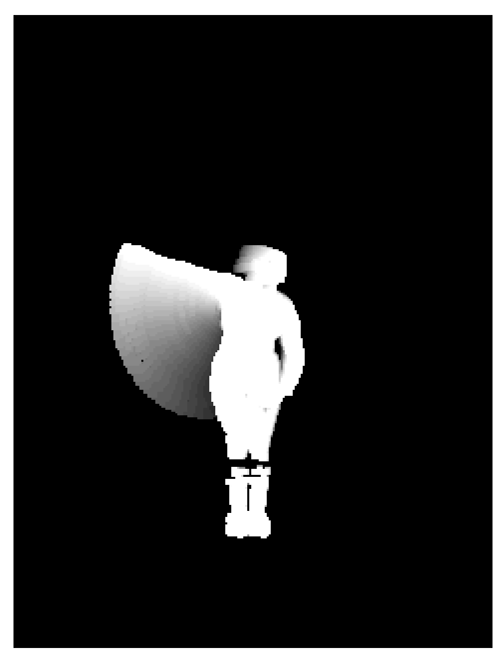
Ideally, estimate flow and warp parameters jointly in one giant variational optimization

(I haven't seen this; looks hard because of joint discrete / continous optimization)

Background subtraction

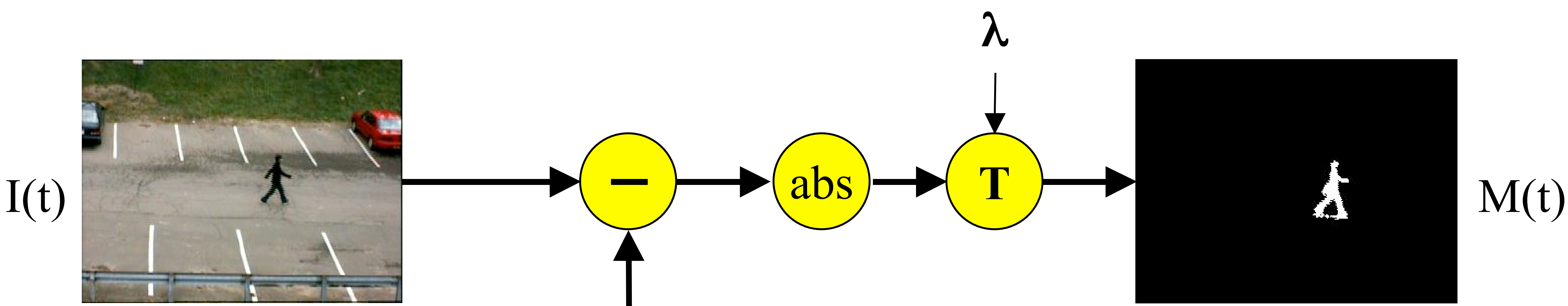
Once we have background image/mosiac (trivial for a stationary camera), how do we identify foreground?

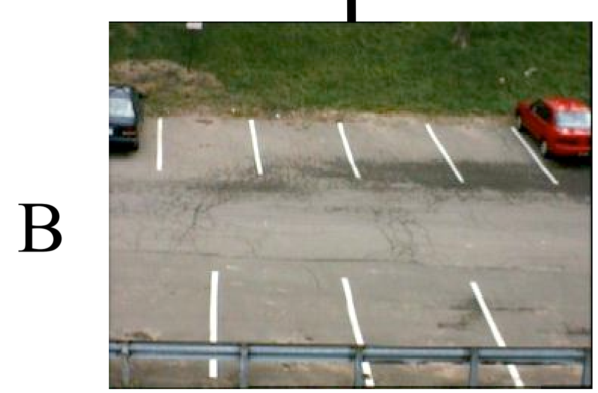




Very commonly-used technique, so we'll spend a few slides on it…

A naive approach






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

(Note: pseudocode is written for grayscale images)

Difficulties (I)

Objects that enter the scene and stop continue to be detected, making it difficult to detect new objects that pass in front of them.

If part of the assumed static background starts moving, both the object and its negative ghost (the revealed background) are detected

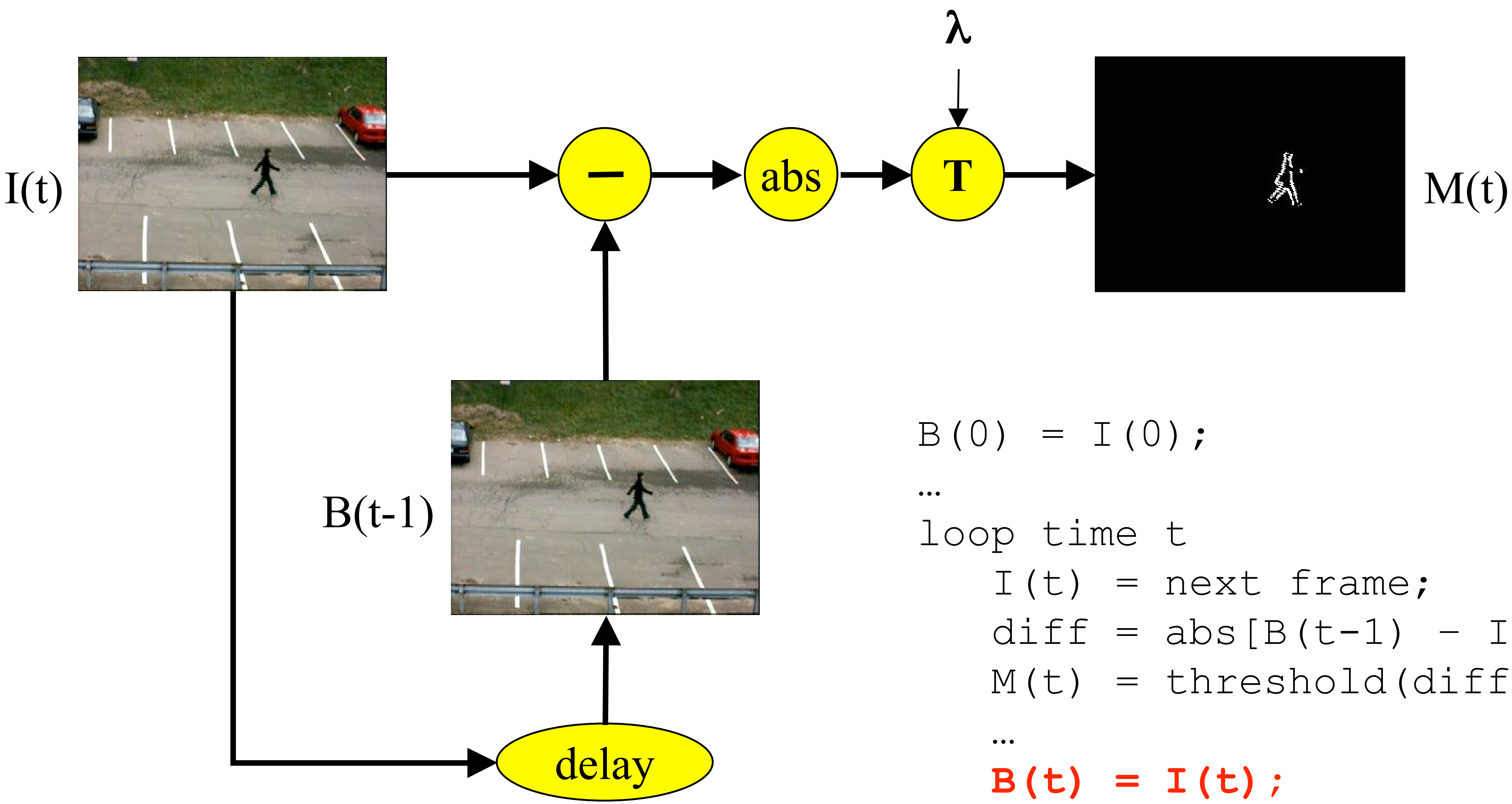
Difficulties (II)

Background subtraction is sensitive to changing illumination and unimportant movement of the background (for example, trees blowing in the wind, reflections of sunlight off of cars or water).

Background subtraction cannot handle movement of the camera.

Frame-differencing

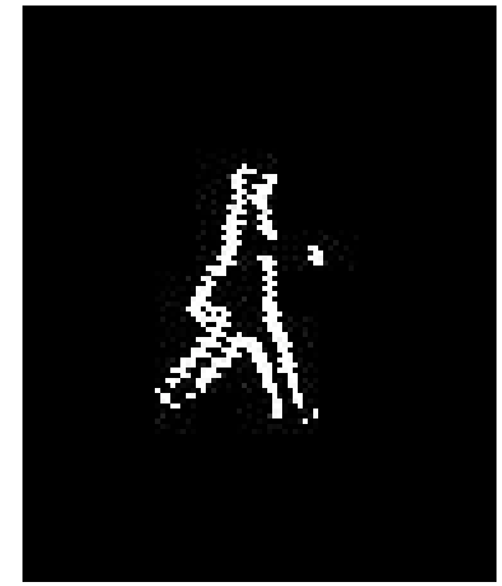
- Background model is replaced with the previous image.




```
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
```

How well does it work?

Frame differencing is very quick to adapt to changes in lighting or camera motion.



Objects that stop are no longer detected. Objects that start up do not leave behind ghosts.

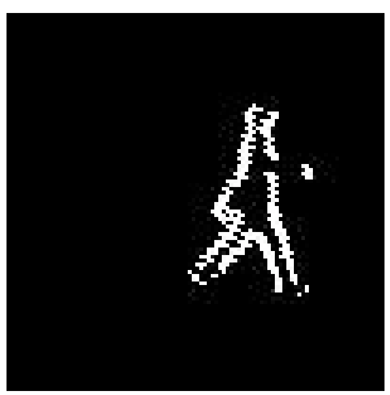
However, frame differencing only detects the leading and trailing edge of a uniformly colored object. As a result very few pixels on the object are labeled, and it is very hard to detect an object moving towards or away from the camera.

Adjusting temporal scale of differencing

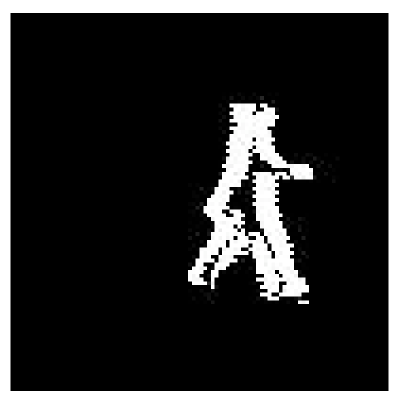
Note what happens when we adjust the temporal scale (frame rate) at which we perform two-frame differencing …

Define D(N) = || I(t) - I(t+N) ||

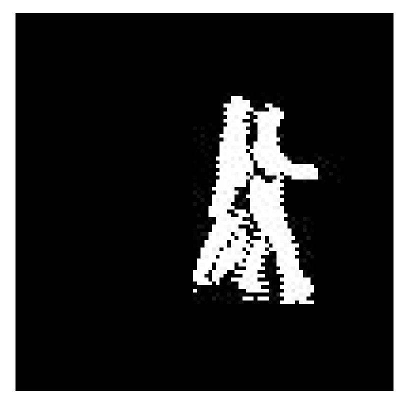
I(t)



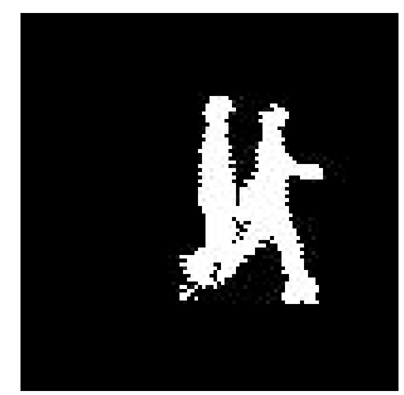
D(-1)



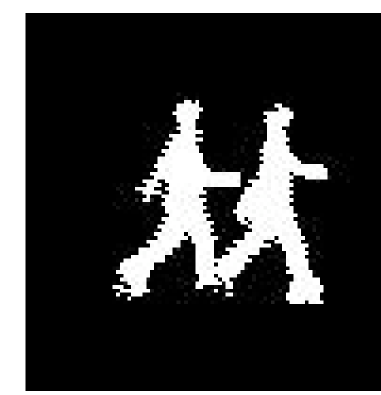
D(-3)



D(-5)



D(-9)

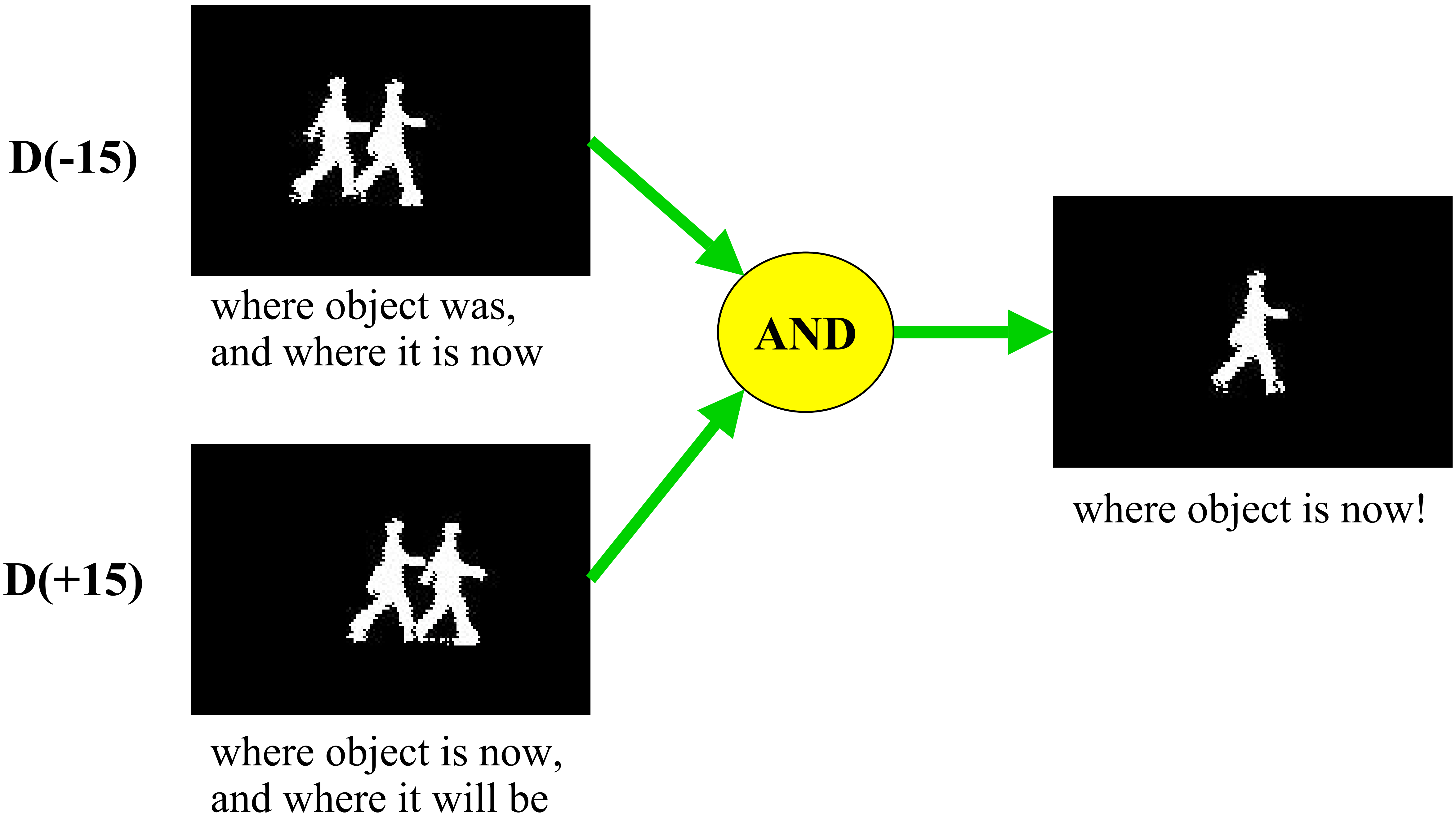


D(-15)

more complete object silhouette, but two copies (one where object used to be, one where it is now).

A neat “trick”: 3-frame differencing

The previous observation is the motivation behind three-frame differencing



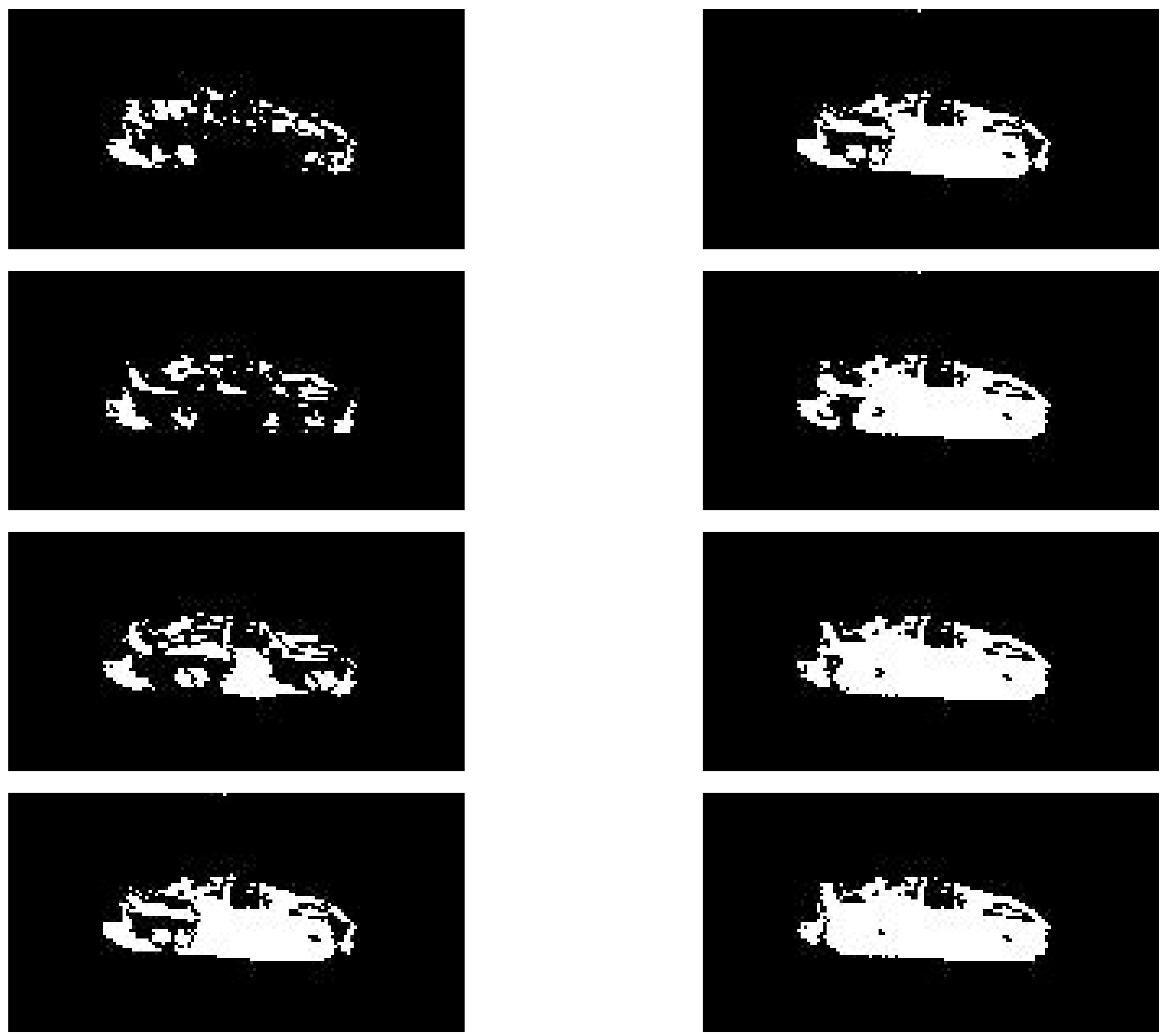
But its hard to find a good frame rate

Choice of good frame-rate for three-frame differencing depends on the size and speed of the object

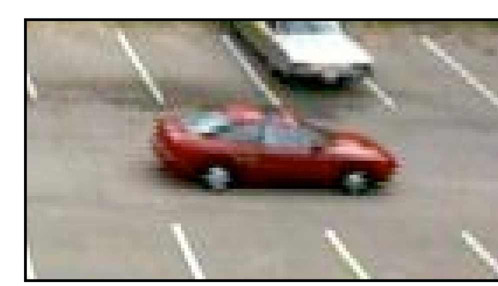
frames skipped

1 **5** **15** **25**

This worked well for the person → 15

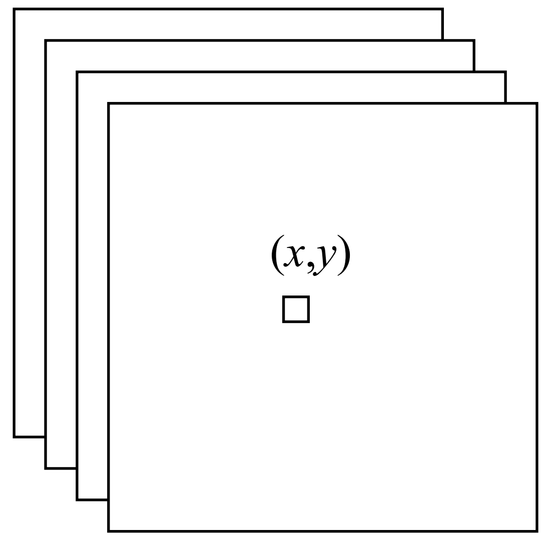


35 **45** **55** **65**



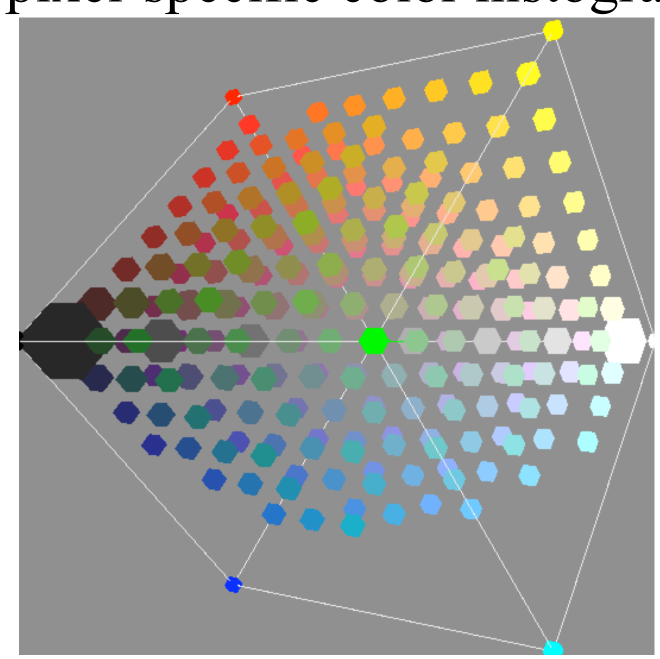
What's a "principled" way to build background model?

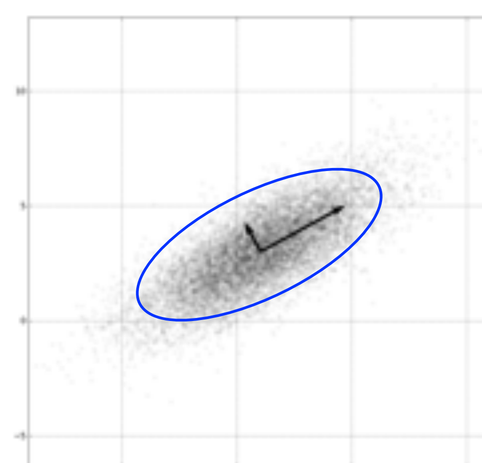
Statistical color models



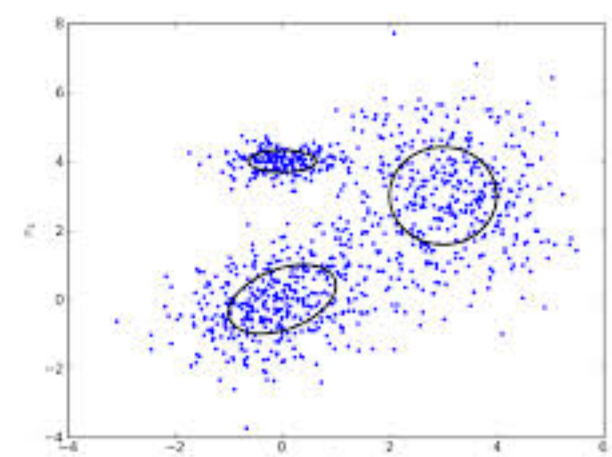
$$P(I(x,y)|bg) > threshold$$

pixel-specific color histogram





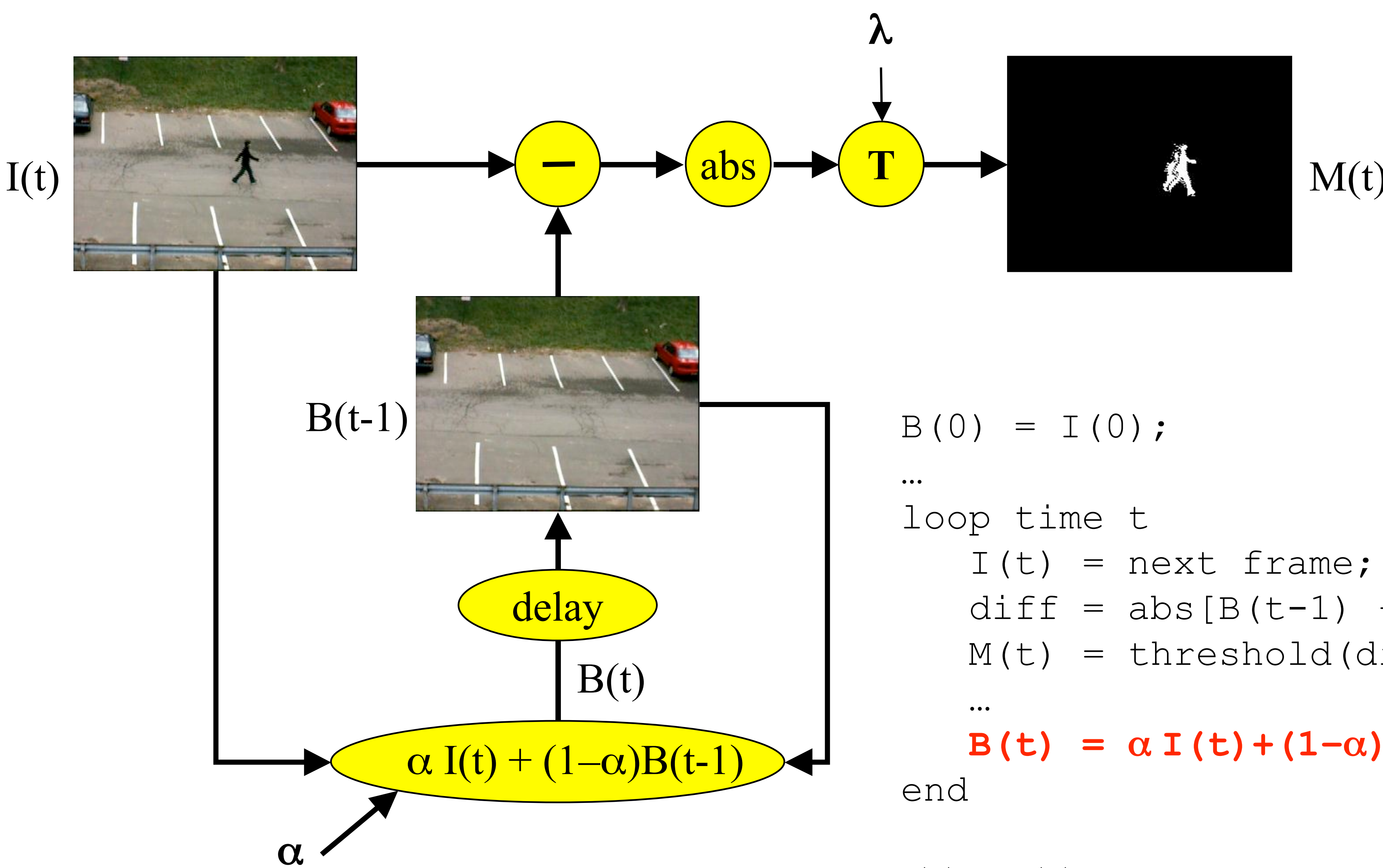
$$P(I) = N(I; \mu, \Sigma)$$



$$P(I) = \sum_i \pi_i N(I; \mu, \Sigma)$$

Online statistical learning (of say, mean)

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

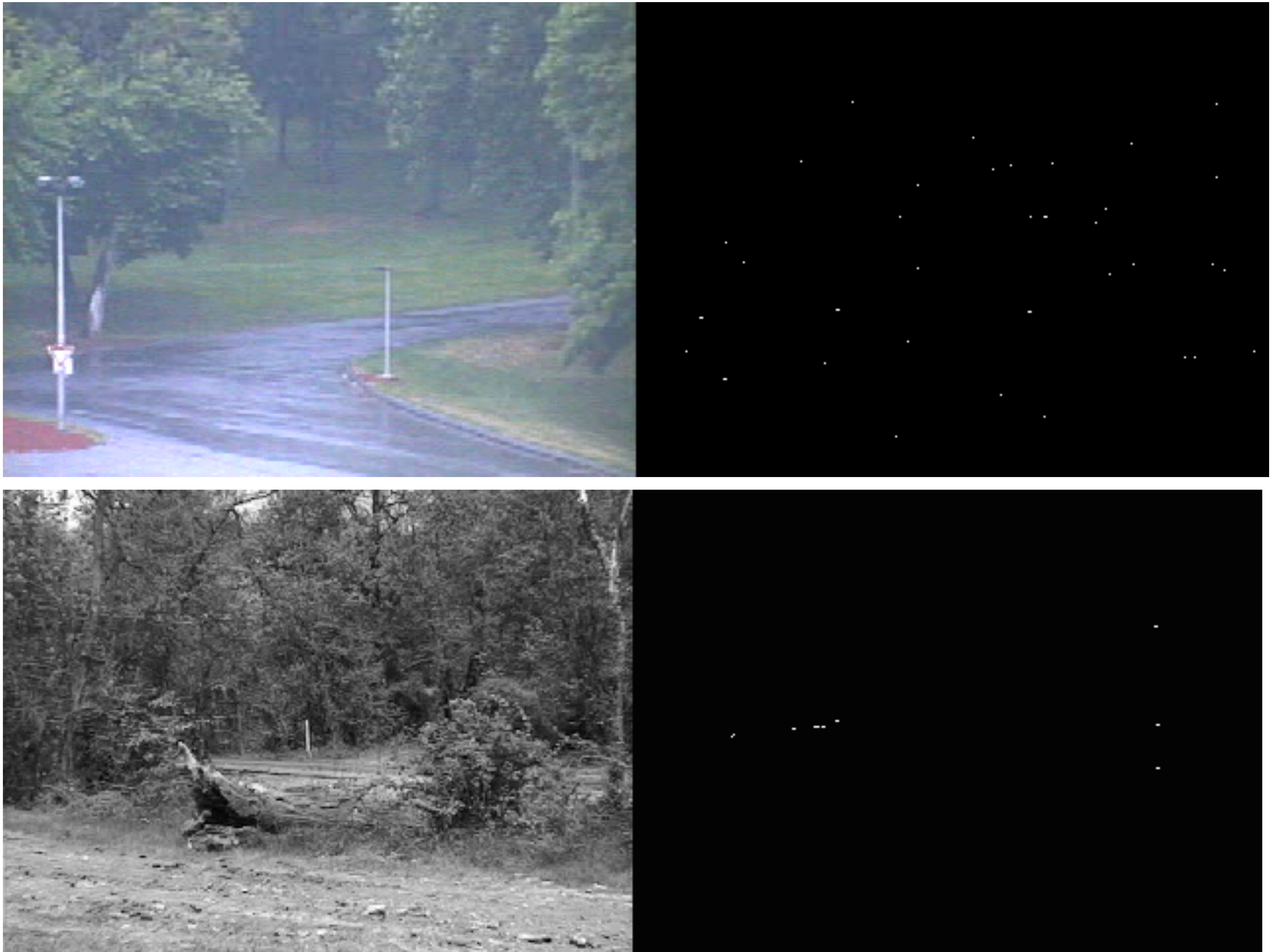



```
B(0) = I(0);
…
loop time t
    I(t) = next frame;
    diff = abs[B(t-1) - I(t)];
    M(t) = threshold(diff,λ);
    …
    B(t) = α I(t)+(1–α)B(t-1);
end
```

$$B(1) = I(1)$$
$$B(2) = .5I(2) + .5I(1)$$
$$B(3) = .5I(3) + .25I(2) + .5I(1)$$
$$\vdots$$

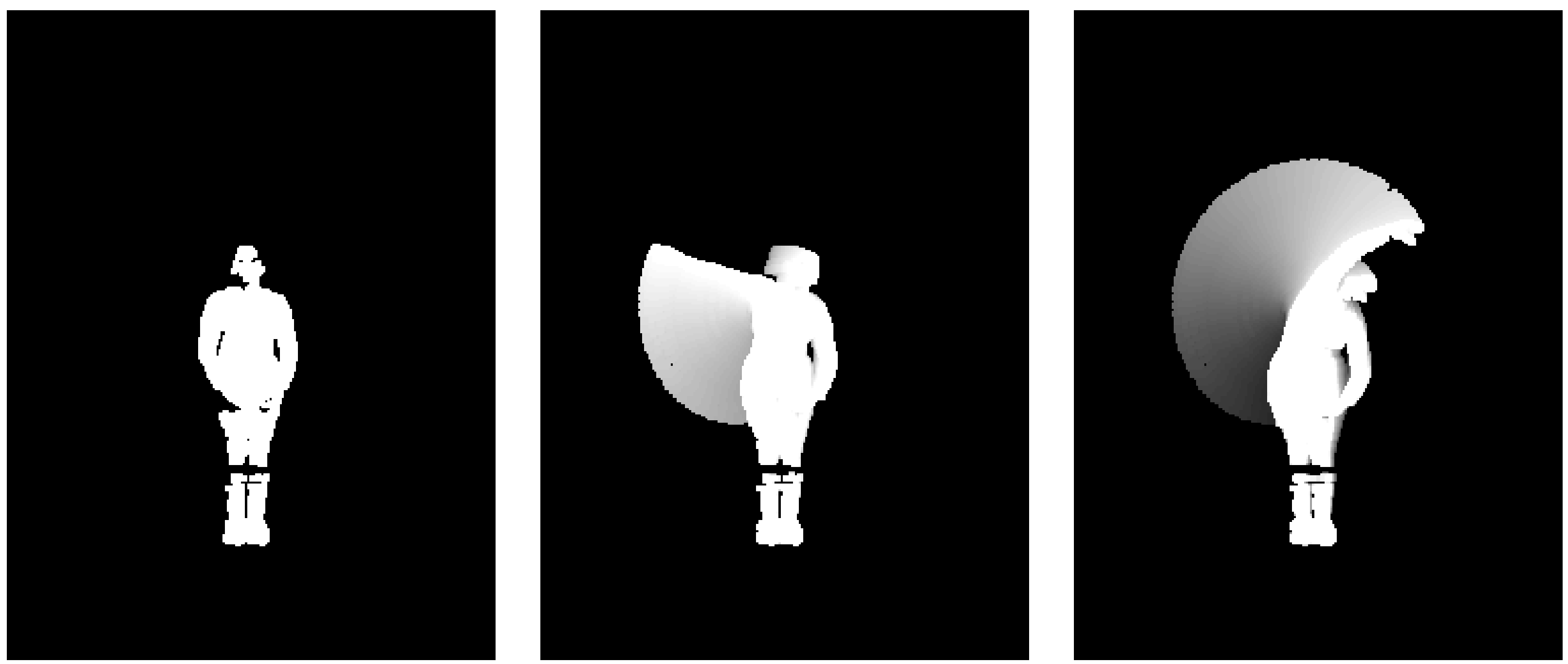
exponential decay

Adaptive background subtraction



Nifty visualizations: persistant frame differencing

FRAME-0 FRAME-35 FRAME-70



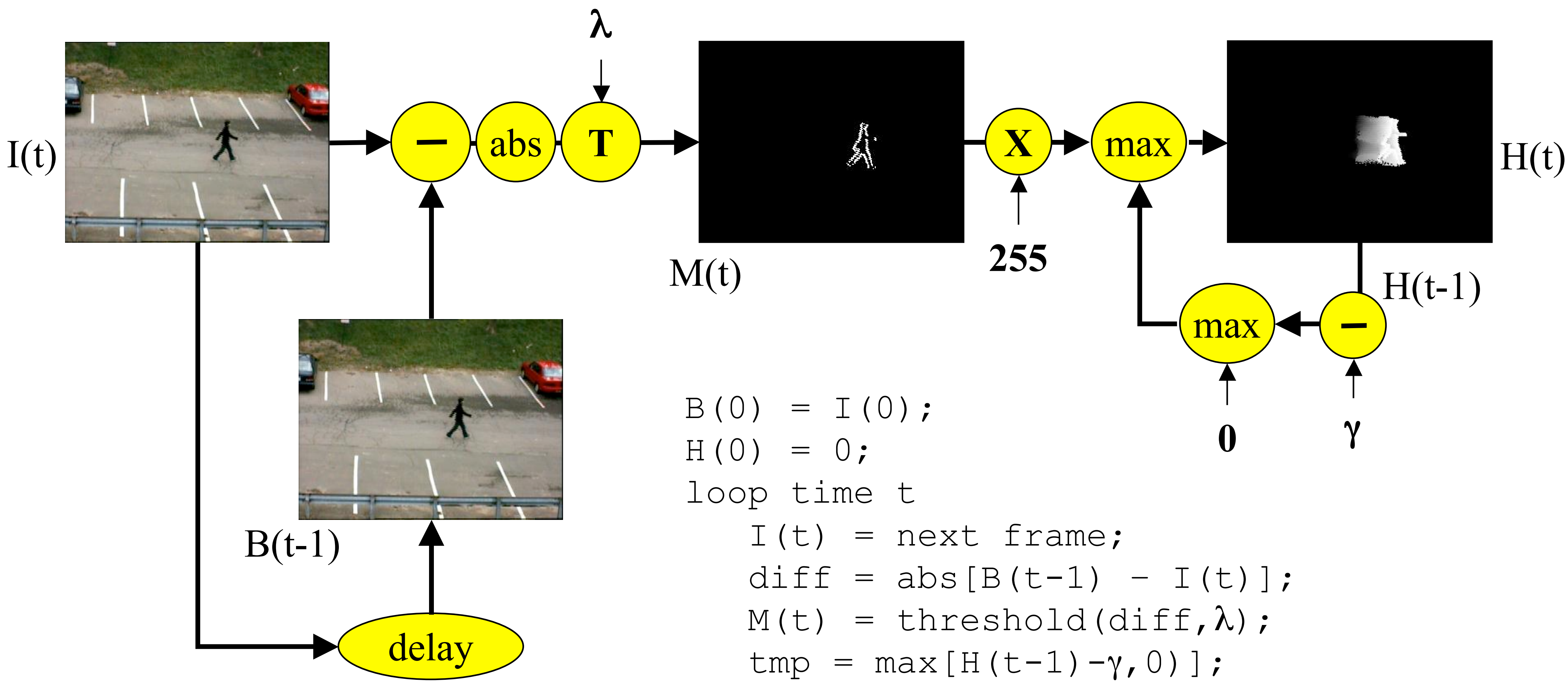
MHI-0 MHI-35 MHI-70

Use some previous method to identify foreground/background pixels

Mark each pixel with the last “time” is was declared foreground

Efficient implementation

- Motion images are combined with a linear decay term
- also known as motion history images (Davis and Bobick)



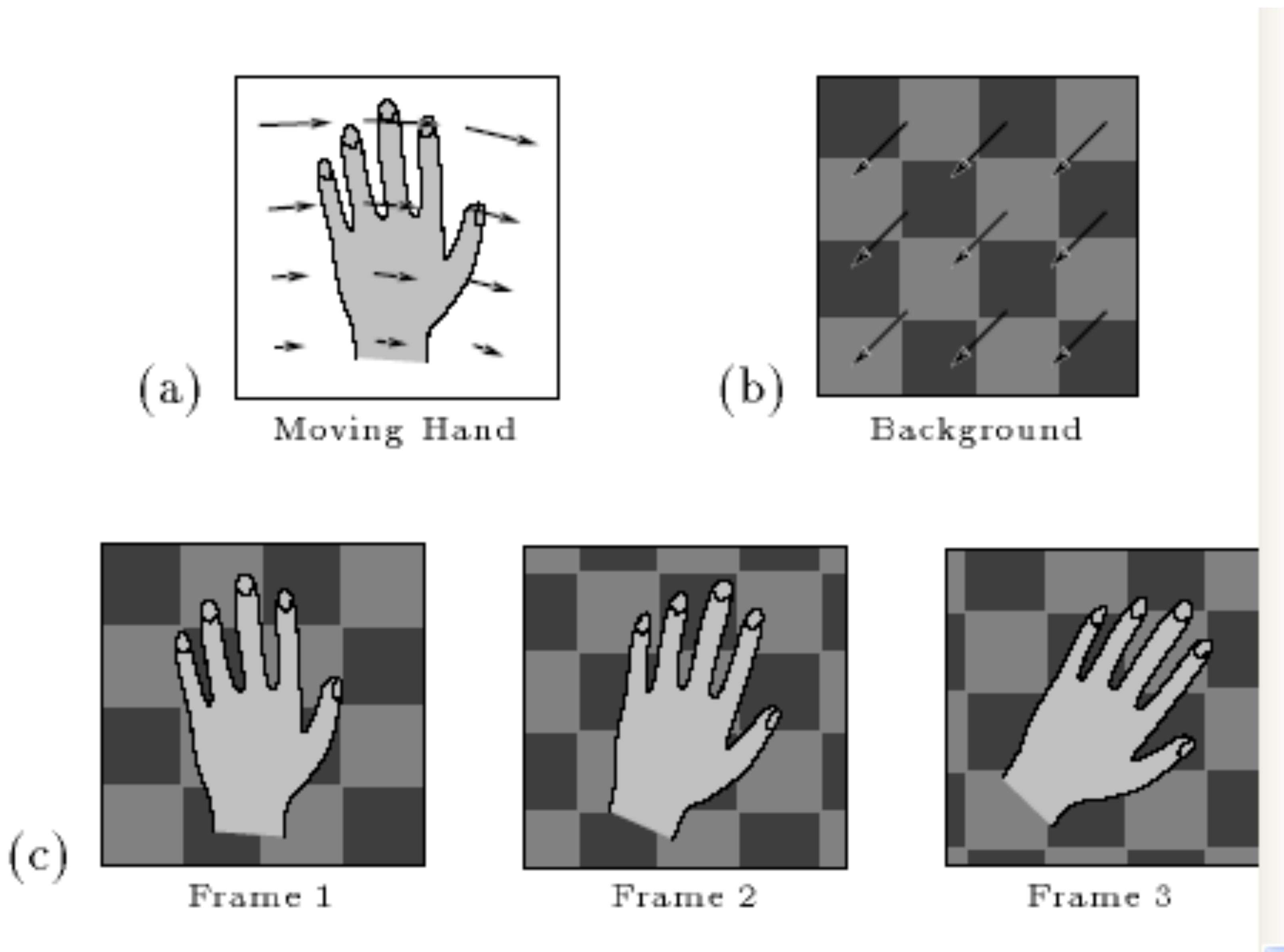

```
B(0) = I(0);
H(0) = 0;
loop time t
    I(t) = next frame;
    diff = abs[B(t-1) - I(t)];
    M(t) = threshold(diff,λ);
    tmp = max[H(t-1)-γ,0)];
    H(t) = max[255*M(t),tmp)];
    …
    B(t) = I(t);
end
```

Motion History Images

Outline

- Bightness constancy
- Aperture problem (sparse flow, spatial regularization)
- Small-motion assumption (coarse-to-fine, discrete optimization)
- Motion segmentation (dominant motion estimation, background subtraction, layered models)

Layered model

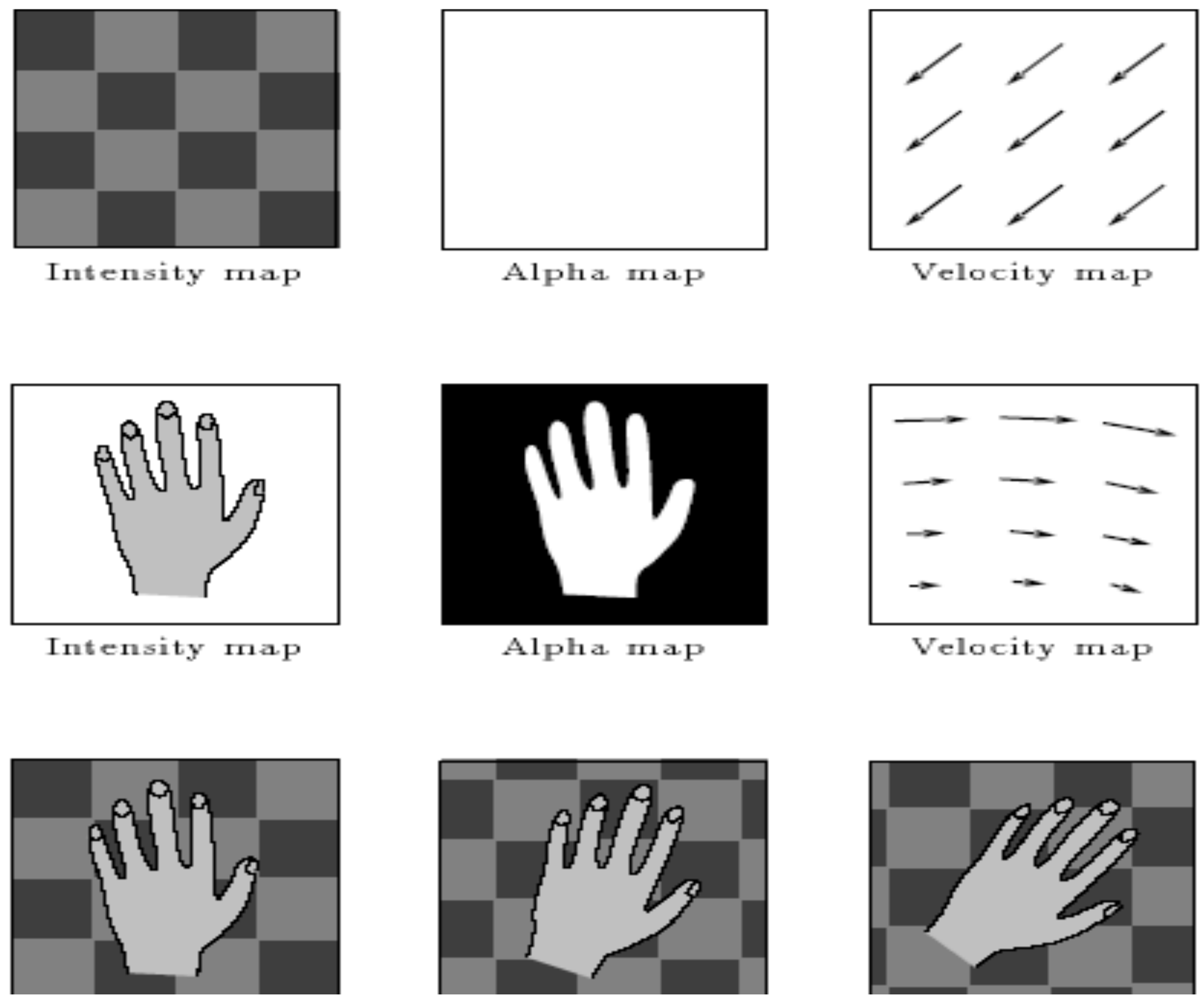


Mathematical formalism

Layer 0 (BG)

Layer 1

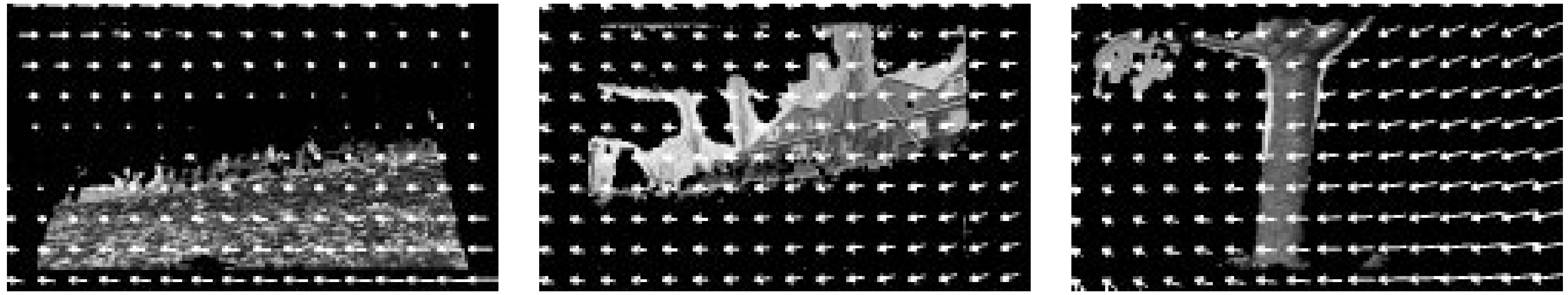
Alpha composite



$$I_i(x, y) = \alpha_i(x, y)L_i(x, y) + (1 - \alpha_i(x, y))I_{i-1}(x, y)$$

Representing Moving Images with Layers

John Y. A. Wang AND Edward H. Adelson



(a) (b) (c)

Figure 12: The layers corresponding to the tree, the flower bed, and the house shown in figures (a-c), respectively. The affine flow field for each layer is superimposed.

(a) (b) (c)

Figure 13: Frames 0, 15, and 30 as reconstructed from the layered representation shown in figures (a-c), respectively.

(a) (b) (c)

Figure 14: The sequence reconstructed without the tree layer shown in figures (a-c), respectively.

Inferring layers, motion, and appearance with EM

Learning Flexible Sprites in Video Layers

Nebojsa Jojic
Microsoft Research
http://www.ifp.uiuc.edu/~jojic

Brendan J. Frey
University of Toronto
http://www.psi.toronto.edu

Takeaways

- Bightness constancy
- Aperture problem (sparse flow, spatial regularization)
- Small-motion assumption (coarse-to-fine, discrete optimization)
- Motion segmentation (dominant motion estimation, background subtraction, layered models)