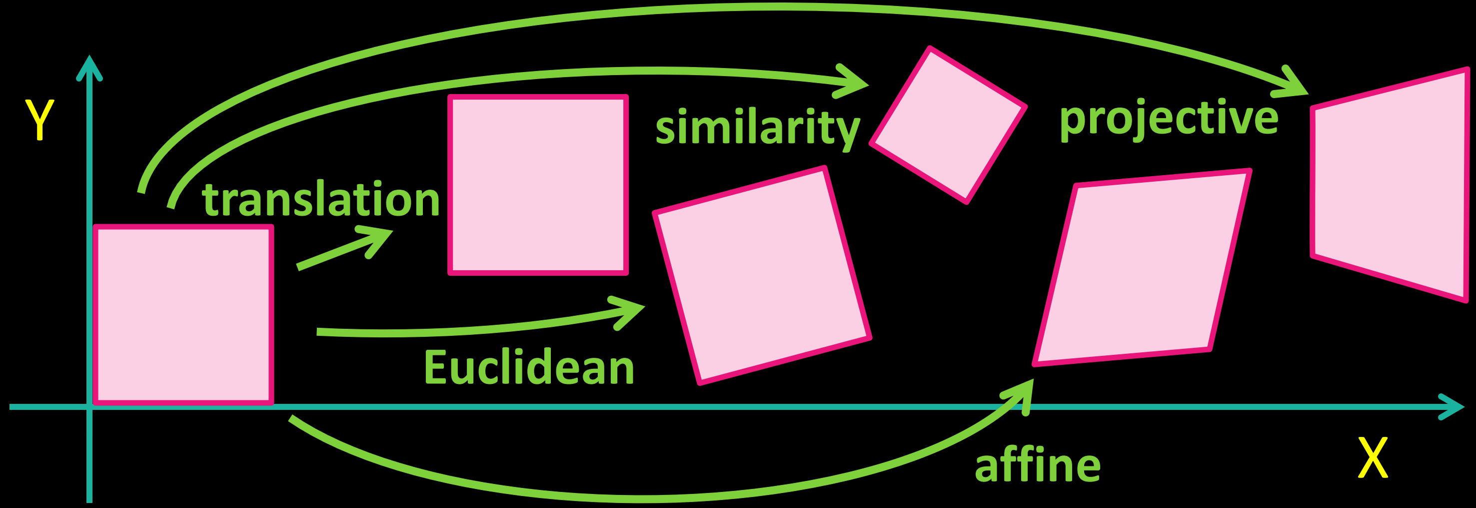
CS4495/6495
Introduction to Computer Vision

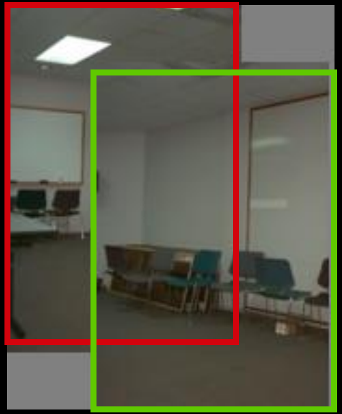
6B-L4 *Motion models*

Motion models



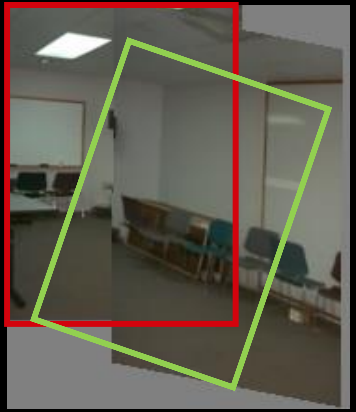
Motion models

Translation



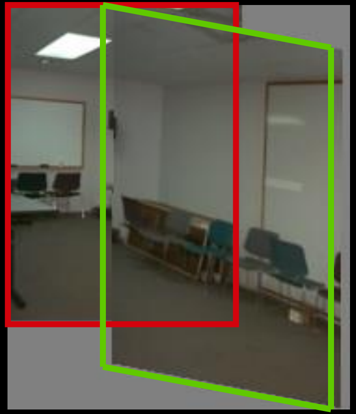
2 unknowns

Similarity



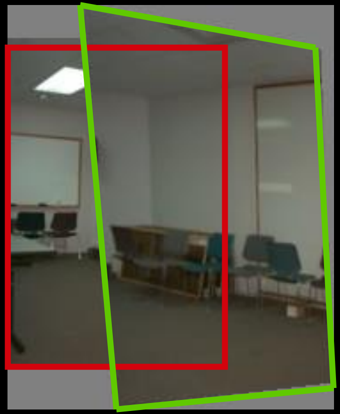
4 unknowns

Affine



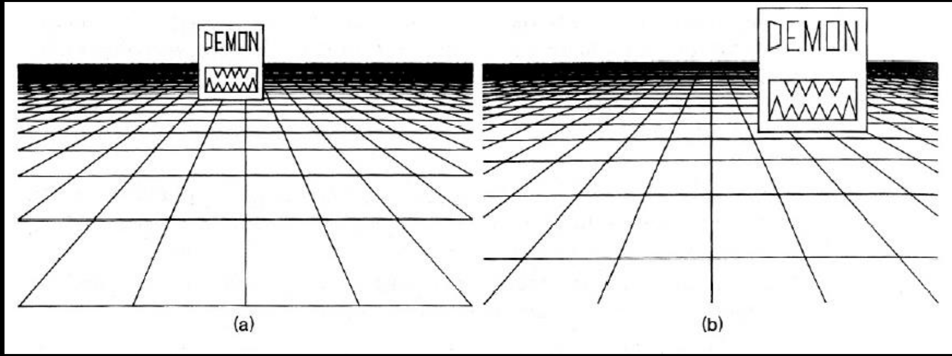
6 unknowns

Perspective



8 unknowns

Focus of Expansion (FOE) - Example



Focus of Expansion (FOE) - Example

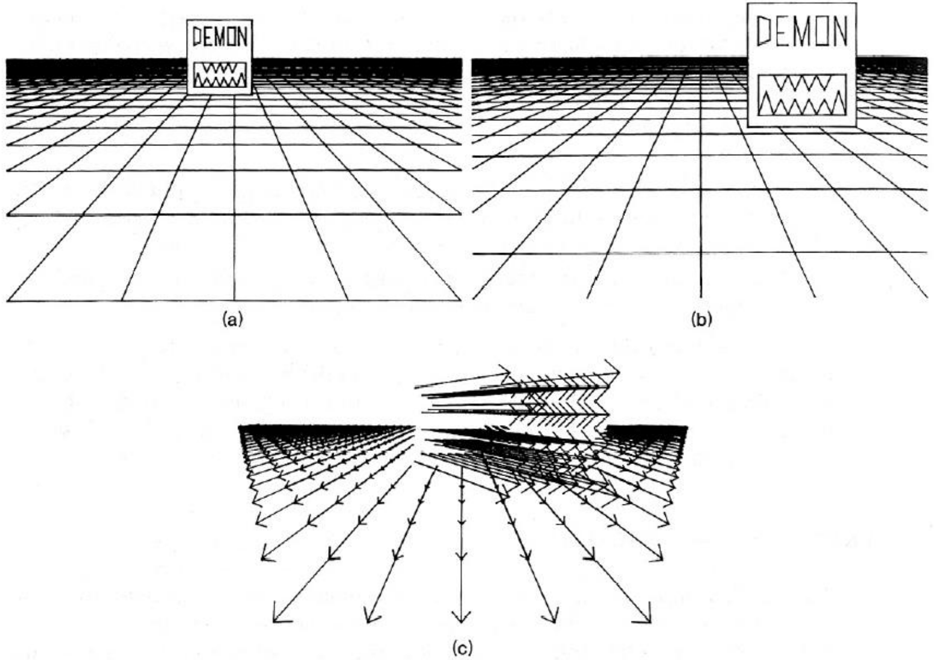


Fig. 7.3 FOE for rectilinear observer motion. (a) An image. (b) Later image. (c) Flow shows different FOEs for static floor and moving object.

Full motion model

From physics or elsewhere:

$$\boxed{V = \Omega \times R + T}$$

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} = \begin{bmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_{T_x} \\ V_{T_Y} \\ V_{T_Z} \end{bmatrix}$$

$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix}$ Velocity Vector

$\begin{bmatrix} V_{T_X} \\ V_{T_Y} \\ V_{T_Z} \end{bmatrix}$ Translational Component of Velocity

$\begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}$ Angular Velocity

General motion

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

$$u = v_x = f\frac{ZV_X - XV_Z}{Z^2} = f\frac{V_X}{Z} - \left(f\frac{X}{Z}\right)\frac{V_Z}{Z} = f\frac{V_X}{Z} - x\frac{V_Z}{Z}$$

$$v = v_y = f\frac{ZV_Y - YV_Z}{Z^2} = f\frac{V_Y}{Z} - \left(f\frac{Y}{Z}\right)\frac{V_Z}{Z} = f\frac{V_Y}{Z} - y\frac{V_Z}{Z}$$

General motion

$$x = f\frac{X}{Z} \qquad u = v_x = f\frac{ZV_X - XV_Z}{Z^2} = f\frac{V_X}{Z} - \left(f\frac{X}{Z}\right)\frac{V_Z}{Z} = f\frac{V_X}{Z} - x\frac{V_Z}{Z}$$

$$y = f\frac{Y}{Z} \qquad v = v_y = f\frac{ZV_Y - YV_Z}{Z^2} = f\frac{V_Y}{Z} - \left(f\frac{Y}{Z}\right)\frac{V_Z}{Z} = f\frac{V_Y}{Z} - y\frac{V_Z}{Z}$$

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{1}{Z(x,y)}\mathbf{A}(x,y)\mathbf{T} + \mathbf{B}(x,y)\mathbf{\Omega}$$

where **T** is the translation vector, Ω is rotation

General motion

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \frac{1}{Z(x,y)} \mathbf{A}(x,y)\mathbf{T} + \mathbf{B}(x,y)\mathbf{\Omega}$$

Why is Z only here?

where **T** is the translation vector, Ω is rotation

$$\mathbf{A}(x,y) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \quad \mathbf{B}(x,y) = \begin{bmatrix} (xy)/f & -(f+x^2)/f & y \\ (f+y^2)/f & -(xy)/f & -x \end{bmatrix}$$

If a plane and perspective...

$$aX + bY + cZ + d = 0$$

$$u(x, y) = a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy$$

$$v(x, y) = a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2$$

If a plane and orthographic…

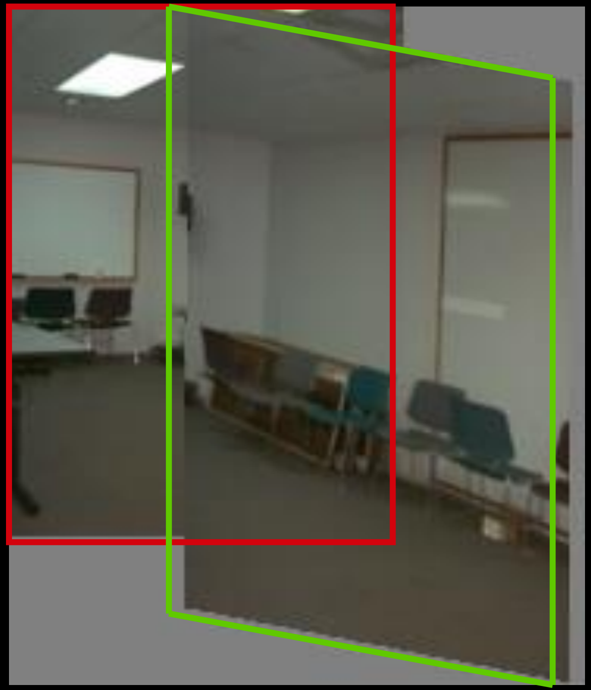
$$u(x, y) = a_1 + a_2 x + a_3 y$$

$$v(x, y) = a_4 + a_5 x + a_6 y$$

Affine!

$$u(x,y) = a_1 + a_2 x + a_3 y$$

$$v(x,y) = a_4 + a_5 x + a_6 y$$



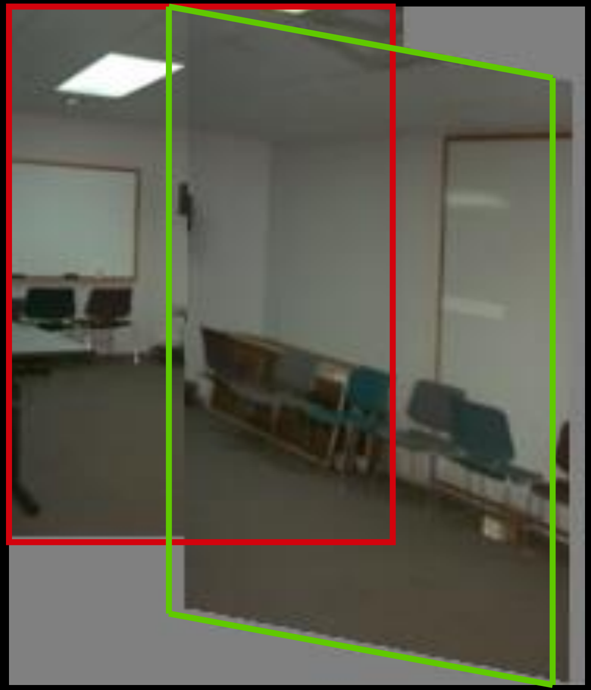
Substituting into the brightness constancy equation:

$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0}$$

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0}$$

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$



- Each pixel provides 1 linear constraint in 6 unknowns
- Least squares minimization:

$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

- Can sum gradients over window or entire image:

$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

- Minimize squared error (robustly)

$$\begin{bmatrix} I_x & I_x x_1 & I_x y_1 & I_y & I_y x_1 & I_y y_1 \\ I_x & I_x x_2 & I_x y_2 & I_y & I_y x_2 & I_y y_2 \\ & & \vdots & & & \\ & & \vdots & & & \\ I_x & I_x x_n & I_x y_n & I_y & I_y x_n & I_y y_n \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = - \begin{bmatrix} I_t^1 \\ I_t^2 \\ \vdots \\ \vdots \\ I_t^n \end{bmatrix}$$

Hierarchical model-based flow

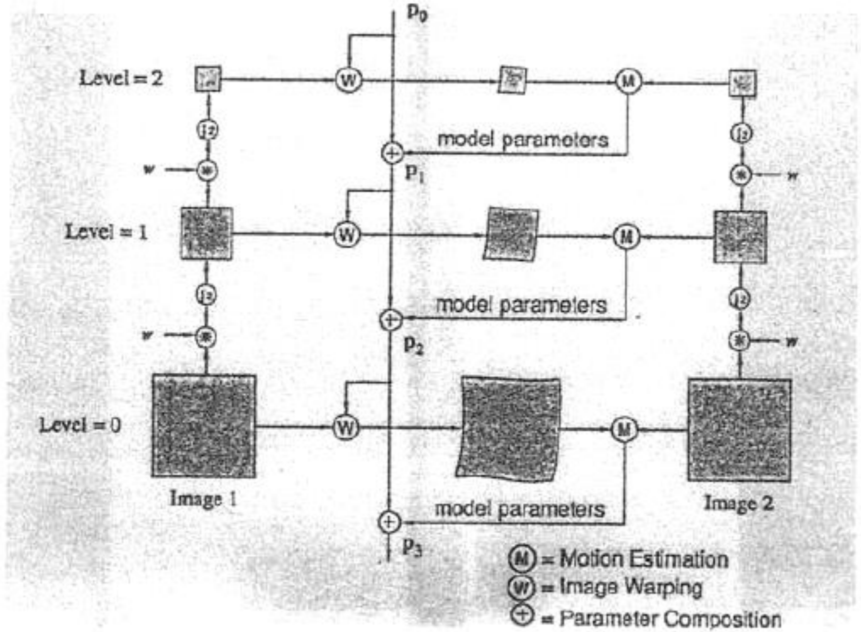


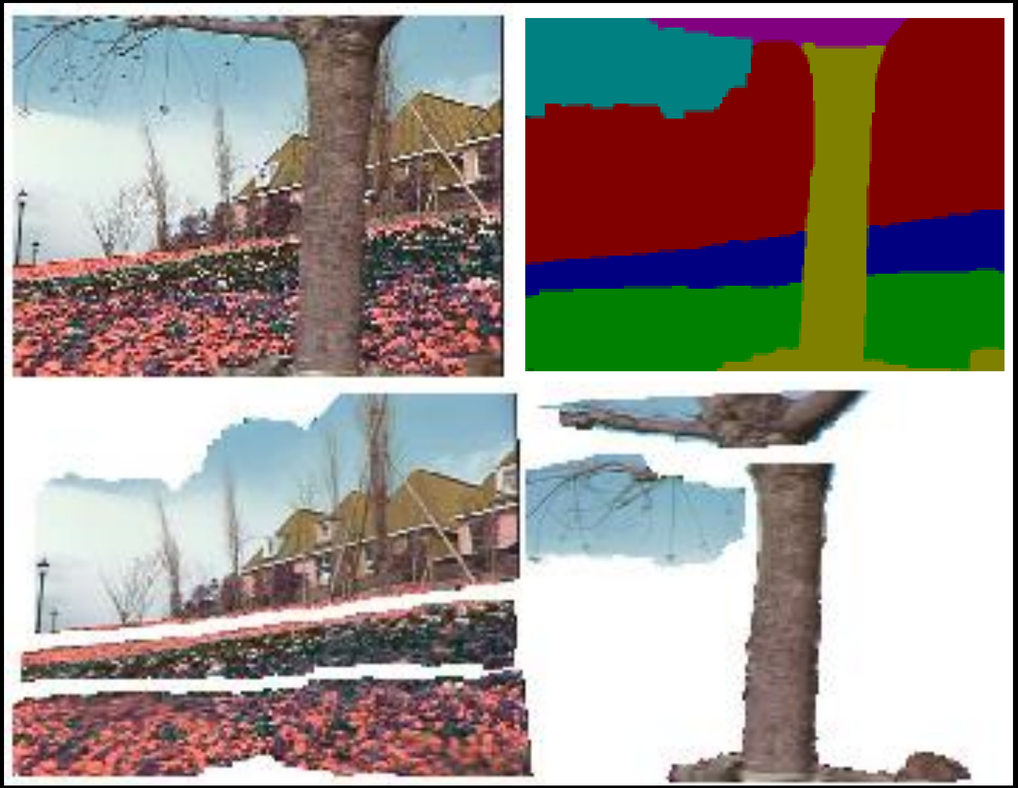
Fig. 1. Diagram of the hierarchical motion estimation framework.

James R. Bergen, P. Anandan, Keith J. Hanna, Rajesh Hingorani: "Hierarchical Model-Based Motion Estimation," ECCV 1992: 237-252

Now, if different motion regions…

Layered motion: Basic idea

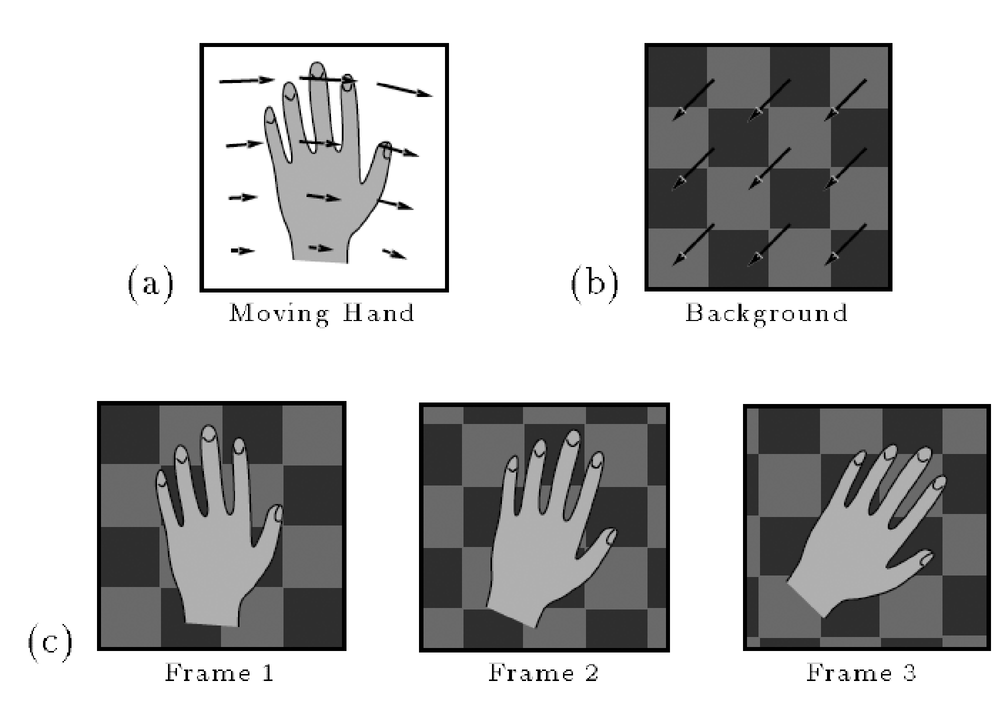
Break image sequence into “layers” – each of which has a coherent motion



J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

What are layers?

Each layer is defined by an alpha mask and an affine motion model



J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

$$\boxed{\begin{aligned} u(x,y) \\ v(x,y) \end{aligned}} \begin{aligned} &= a_1 + a_2 x + a_3 y \\ &= a_4 + a_5 x + a_6 y \end{aligned}$$

Local flow
estimates

$$\boxed{u(x,y) = a_1 + a_2 x + a_3 y}$$
$$v(x,y) = a_4 + a_5 x + a_6 y$$

Equation of a plane (parameters a_1, a_2, a_3 can be found by least squares)

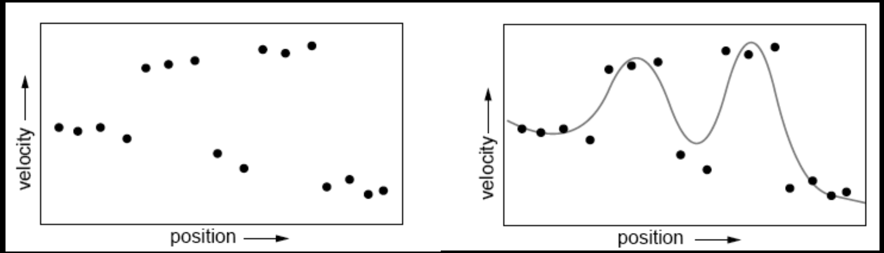
$$\boxed{u(x,y) = a_1 + a_2 x + a_3 y}$$

$$v(x,y) = a_4 + a_5 x + a_6 y$$

Equation of a plane (parameters a_1, a_2, a_3 can be found by least squares)

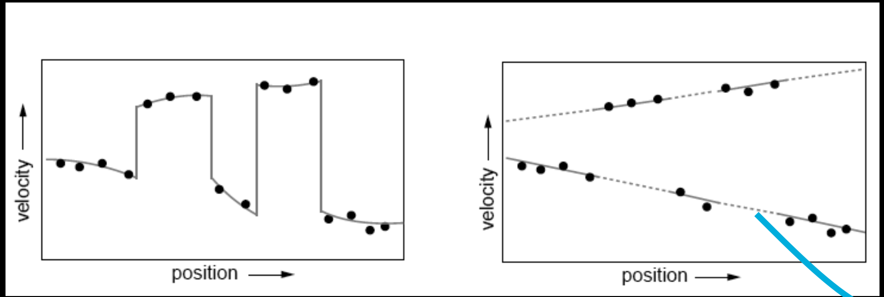
1D example:

$u(x,y)$



True flow

Local flow estimate



Segmented estimate

Line fitting

“Foreground”

“Background”

Occlusion

How do we estimate the layers?

1. Compute local flow in a coarse-to-fine fashion

How do we estimate the layers?

2. Obtain a set of initial affine motion hypotheses
   - Divide the image into blocks and estimate affine motion parameters in each block by least squares
   - Perform k-means clustering on affine motion parameters

How do we estimate the layers?

3. Iterate until convergence:
   - Assign each pixel to best hypothesis
     - Pixels with high residual error remain unassigned
   - Perform region filtering to enforce spatial constraints
   - Re-estimate affine motions in each region

Example result

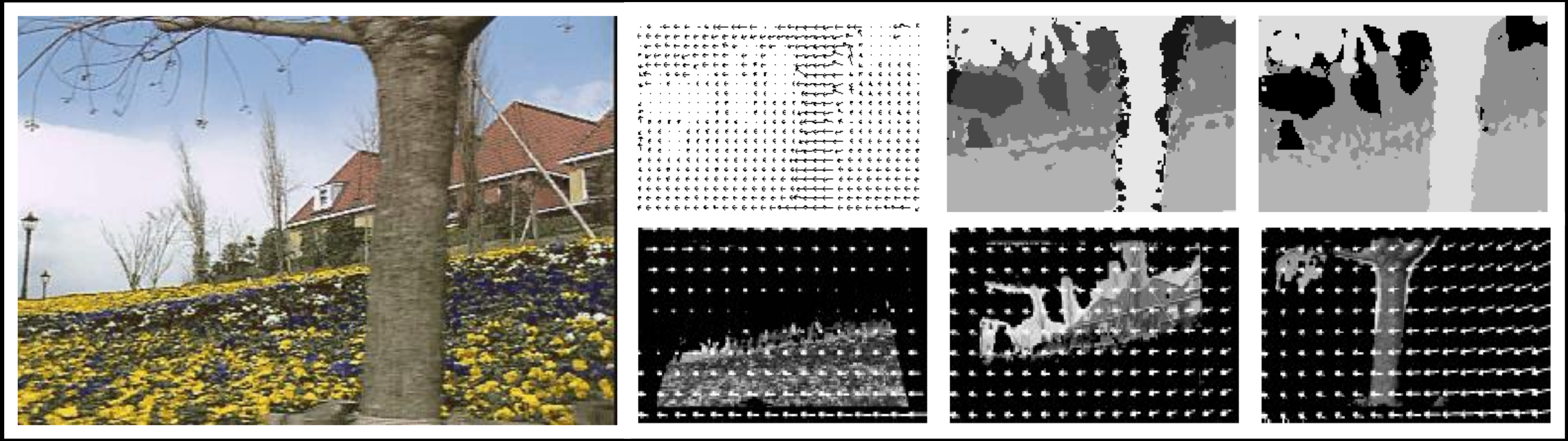


Image motion: Summary

- Feature-based methods (e.g. SIFT, RANSAC, regression)
  - Extract visual features (corners, textured areas), and track them – sometimes over multiple frames
  - Sparse motion fields, but possibly robust tracking – good for global motion
  - Suitable especially when image motion is large (10s of pixels)

- Direct-methods (e.g. optical flow)
  - Directly recover motion from spatio-temporal image brightness variations
  - Dense, local motion fields, but more sensitive to appearance variations
  - Suitable for video and when image motion is small (< 10 pixels)