计算机视觉与模式识别

Computer Vision and Pattern Recognition

Motion Analysis

3D motion field, 2D motion field (image motion), Apparent motion, Optical flow computation (Nonparametric methods), KLT feature tracking

人工智能与机器人研究所

Institute of Artificial Intelligence and Robotics

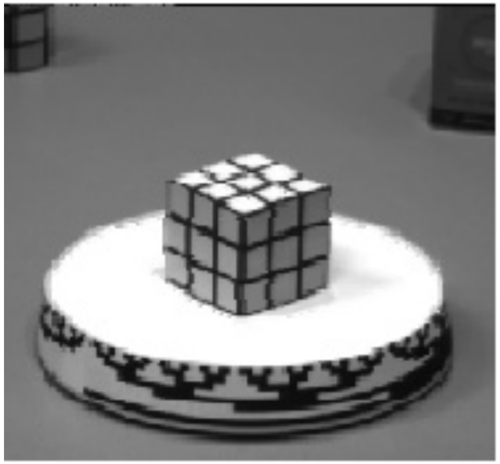
袁泽剑

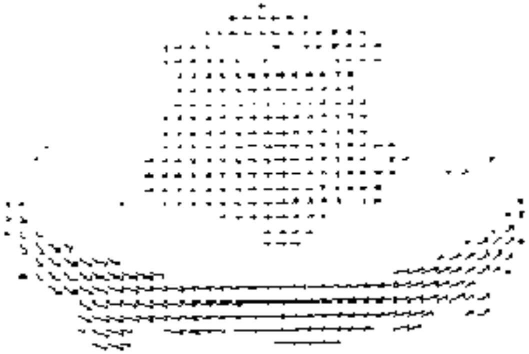
Email: yuan.ze.jian@xjtu.edu.cn

科学馆102室

Motion Field

- The motion field is the projection of the 3D scene motion into the image

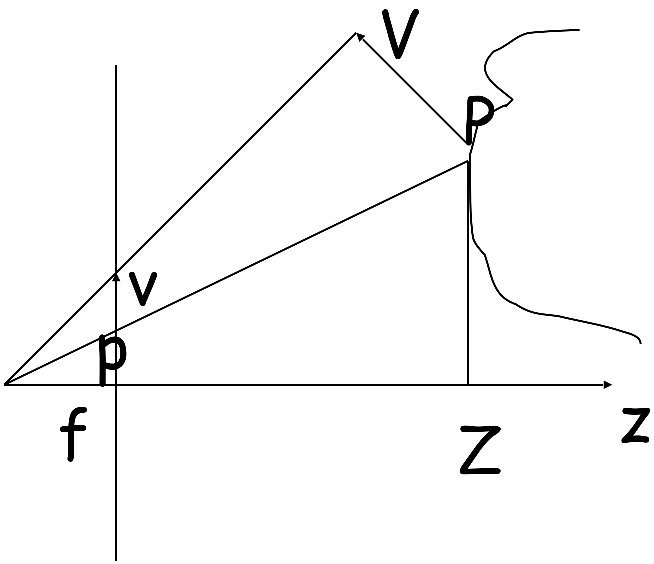




3D motion —> 2D motion

Slide credit: Svetlana Lazebnik

2D Motion Field



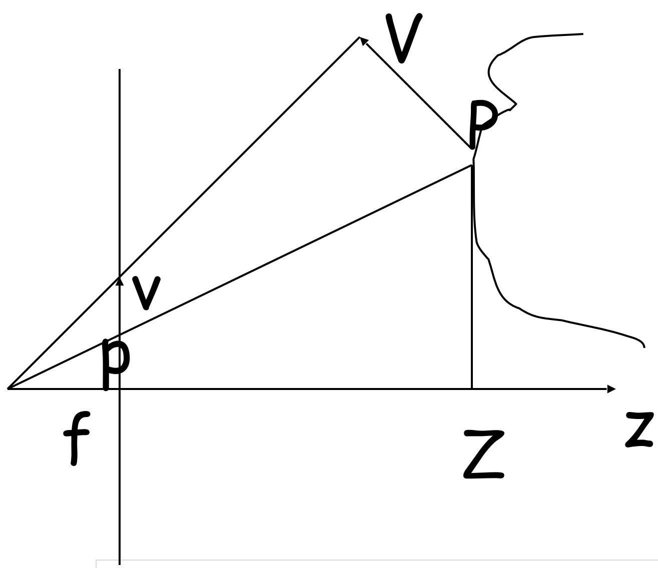
Translational component

Scaling ambiguity
(t and Z can only
be derived up to a scale
Factor)

$$u = \frac{t_z x - t_x f}{Z} + \frac{\omega_x x y}{f} - \omega_y\left(f + \frac{x^2}{f}\right) + \omega_z y$$

$$v = \frac{t_z y - t_y f}{Z} + \omega_x\left(f + \frac{y^2}{f}\right) - \frac{\omega_y x y}{f} - \omega_z x$$

2D Motion Field



Rotational component

$$u = \frac{t_z x - t_x f}{Z} + \frac{\omega_x x y}{f} - \omega_y\left(f + \frac{x^2}{f}\right) + \omega_z y$$

$$v = \frac{t_z y - t_y f}{Z} + \omega_x\left(f + \frac{y^2}{f}\right) - \frac{\omega_y x y}{f} - \omega_z x$$

NOTE: The rotational component is independent of depth Z !

Optical Flow Field

- Optical Flow
  - Brightness constancy constraint
  - Aperture problem
  - Lucas-Kanade flow
  - Iterative refinement
  - Coarse-to-fine estimation
- Global parametric motion

Optical Flow

- **Definition-1**: **optical flow is the apparent motion of brightness patterns in the image**.
- **Ideally**, optical flow would be the same as the motion field.
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion.
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination.

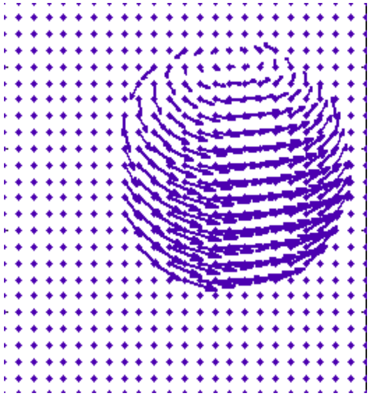
Optical Flow & Motion Field

▪ Definition-2

The **optical flow** is a velocity field in the image which transforms one image into the next image in a sequence [Horn&Schunck]

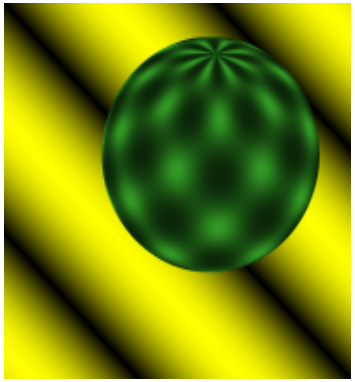
frame #1

\+



flow field

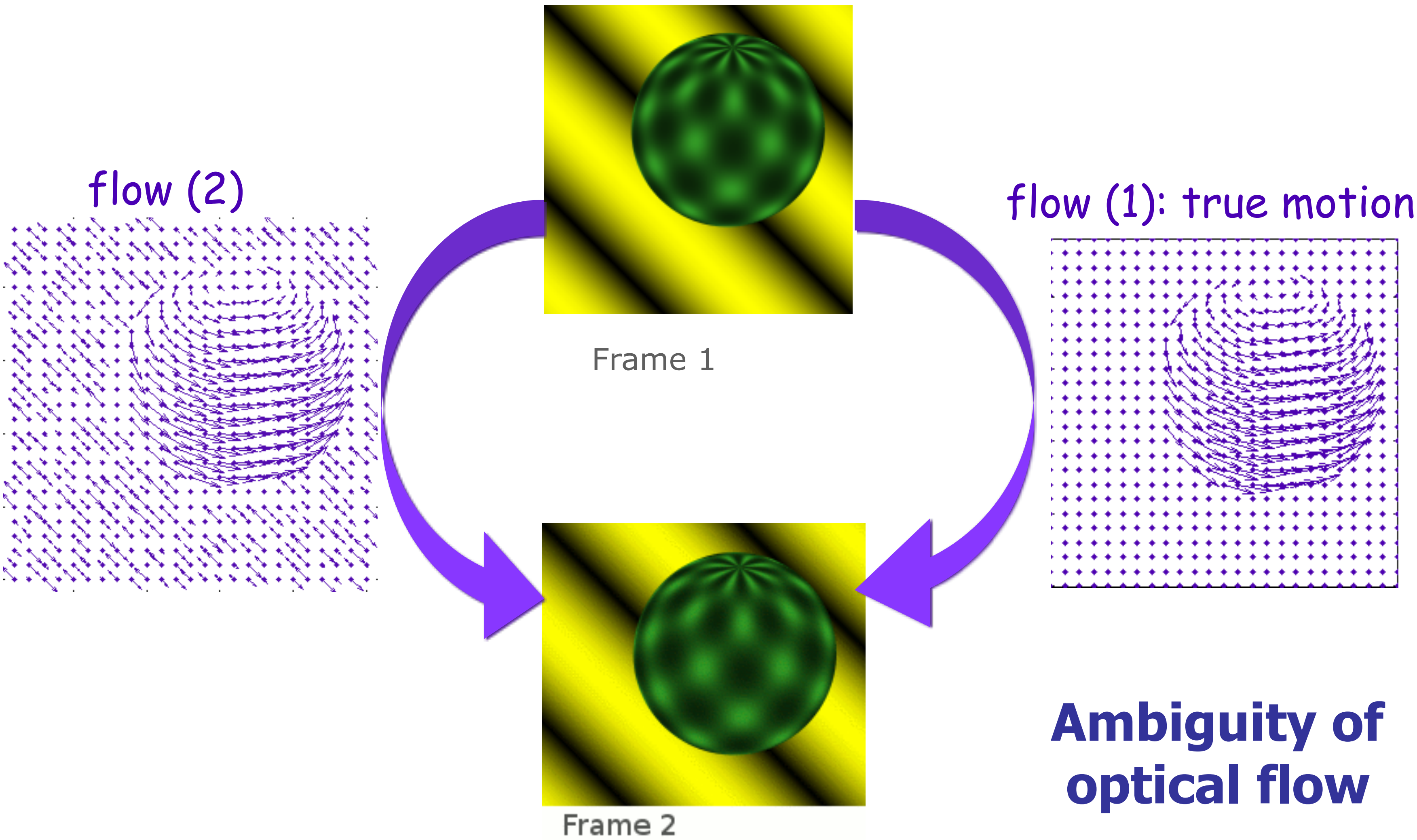
=



frame #2

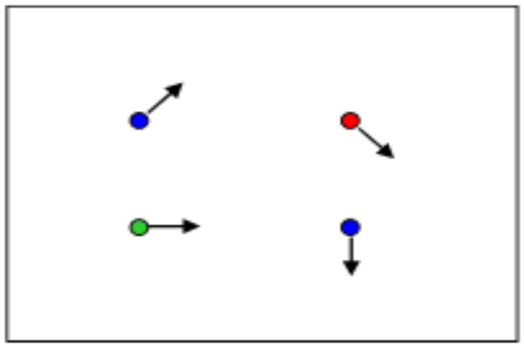
The **motion field** … is the projection into the image of three-dimensional motion vectors [Horn&Schunck]

Optical Flow & Motion Field

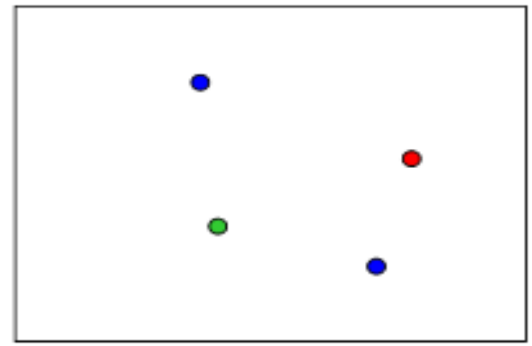


Ambiguity of optical flow

Estimating Optical Flow



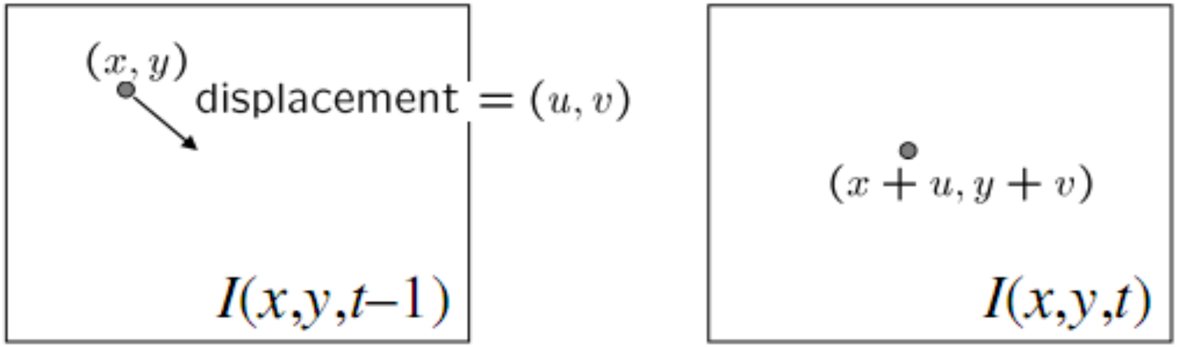
$I(x,y,t-1)$



$I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field u(x,y) and v(x,y) between them.
- **Key assumptions**
  - Brightness constancy: projection of the same point looks the same in every frame. (**Local image constraints**)
  - Small motion: points do not move very far.
  - Spatial coherence: points move like their neighbors.

Local image constraints



- **Brightness Constancy Equation:**

$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

- **Linearizing the right hand side using Taylor expansion:**

$$I(x, y, t-1) \approx I(x, y, t) + I_x \cdot u(x, y) + I_y \cdot v(x, y)$$

- **Hence,** $I_x \cdot u + I_y \cdot v + I_t \approx 0$

Spatial derivatives **Temporal derivative**

Local image constraints

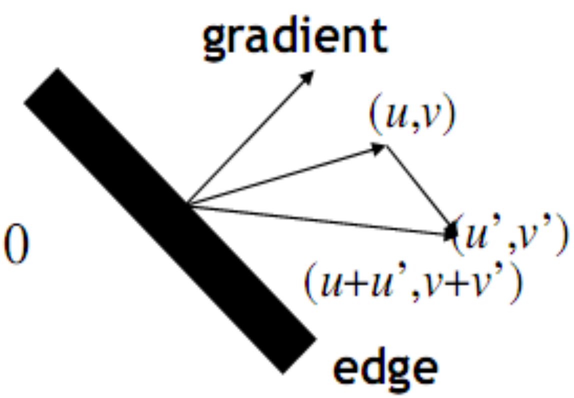
$$I_x \cdot u + I_y \cdot v + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation, two unknowns
- Intuitively, what does this constraint mean?

$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

If (u,v) satisfies the equation,
so does $(u+u', v+v')$ if $\nabla I \cdot (u', v') = 0$



The Aperture Problem

Solving the Aperture Problem

- How to get more equations for a pixel?
- **Spatial coherence constraint**: pretend the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

B. Lucas and T. Kanade. **An iterative image registration technique with an application to stereo vision**. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679, 1981

- **Least squares problem:**

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \qquad \underset{25\times 2}{A} \; \underset{2\times 1}{d} = \underset{25\times 1}{b}$$

- **Minimum least squares solution given by solution of**

$$\underset{2\times 2}{(A^T A)} \; \underset{2\times 1}{d} = \underset{2\times 1}{A^T b}$$

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

$\boxed{\text{(The summations are over all pixels in the K x K window)}}$

- Optimal (u, v) satisfies **Lucas-Kanade equation**

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

- When is this solvable?
  - A^TA should be invertible.
  - A^TA entries should not be too small (noise).
  - A^TA should be well-conditioned.

Eigenvectors of A^TA

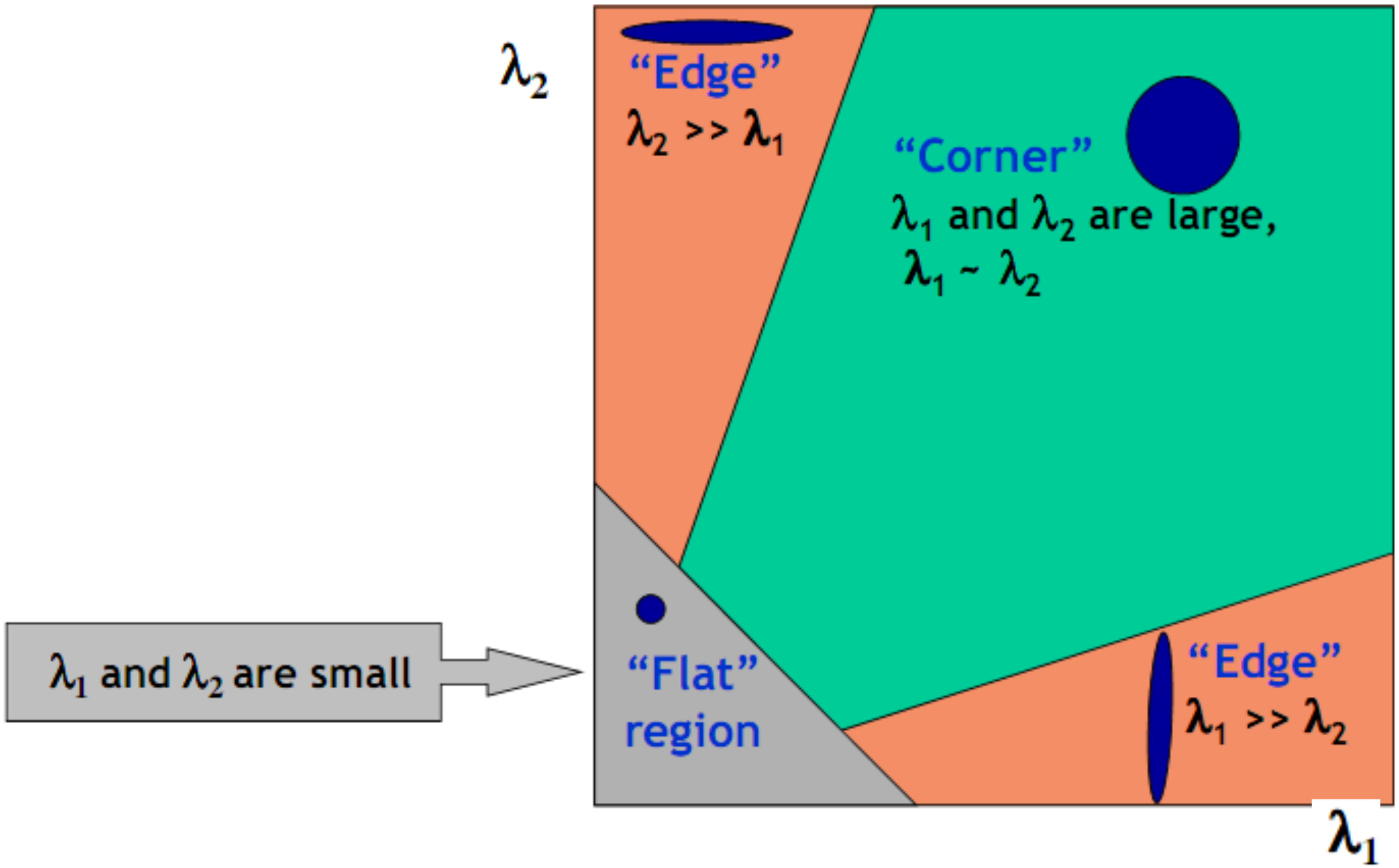
$$A^TA = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

- The eigenvectors and eigenvalues of M relate to edge direction and magnitude.
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change.
  - The other eigenvector is orthogonal to it.

Slide credit: Svetlana Lazebnik

Interpreting the Eigenvalues

- Classification of image points using eigenvalues of the second moment matrix:



Slide credit: Kristen Grauman

- Edge

$$\sum \nabla I (\nabla I)^T$$

- Gradients very large or very small
- Large λ_1, small λ_2

Slide credit: Svetlana Lazebnik

- # Low-Texture Region

$$\sum \nabla I (\nabla I)^T$$

- Gradients have small magnitude
- Small λ_1 , small λ_2

Slide credit: Svetlana Lazebnik

High-Texture Region

$$\sum \nabla I (\nabla I)^T$$

- Gradients are different, large magnitude
- Large λ_1 , large λ_2

Slide credit: Svetlana Lazebnik

Per-Pixel Estimation Procedure

- Let $M = \sum (\nabla I)(\nabla I)^T$ and $b = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$
- Algorithm: At each pixel compute U by solving $MU = b$
- M is singular if all gradient vectors point in the same direction
  - E.g., along an edge
  - Trivially singular if the summation is over a single pixel or if there is no texture
  - I.e., only normal flow is available (aperture problem)
- Corners and textured areas are OK

Iterative Refinement

1. Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation.

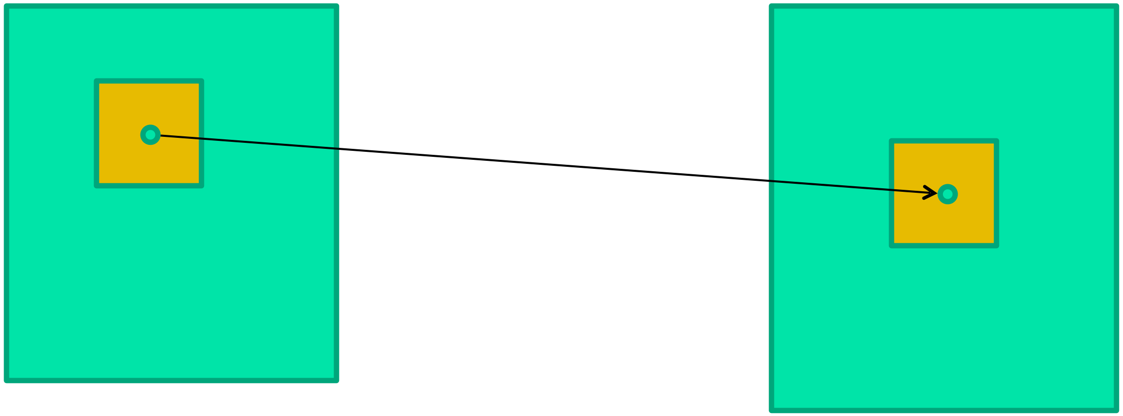
$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

2. Warp one image toward the other using the estimated flow field.

3. Refine estimate by repeating the process.

B. Lucas and T. Kanade. **An iterative image registration technique with an application to stereo vision**. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679, 1981

Lucas-Kanade flow



$$I_x \cdot u + I_y \cdot v + I_t = 0$$

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \qquad \underset{25\times2}{A} \; \underset{2\times1}{d} = \underset{25\times1}{b}$$

$$\underset{2\times2}{(A^T A)} \; \underset{2\times1}{d} = \underset{2\times1}{A^T b}$$

$$A^T A = \sum \nabla I (\nabla I)^T$$

Lucas-Kanade flow

- Brightness constant equation (Optical equation)

$$I\left(x,y,t-1\right)=I\left(x+u\left(x,y\right),y+v\left(x,y\right),t\right)$$

- Spatial coherence constraint (local): pretend the pixel's neighbors (Ω) have the same (u,v)

$$E\left(u,v\right)=\sum_{(x,y)\in\Omega} w\left(x,y\right)\left(I_x\left(x,y\right)\cdot u+I_y\left(x,y\right)\cdot v+I\left(x,y,t\right)-I\left(x,y,t-1\right)\right)^2$$

$$=\sum_{(x,y)\in\Omega} w\left(x,y\right)\left(\begin{bmatrix} I_x & I_y & I_t \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}\right)^2=\begin{bmatrix} u & v & 1 \end{bmatrix} M \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$M=\sum_{(x,y)\in\Omega} w\left(x,y\right)\begin{bmatrix} I_x^{\ 2} & I_xI_y & I_xI_t \\ I_yI_x & I_y^{\ 2} & I_yI_t \\ I_tI_x & I_tI_y & I_t^{\ 2} \end{bmatrix}$$

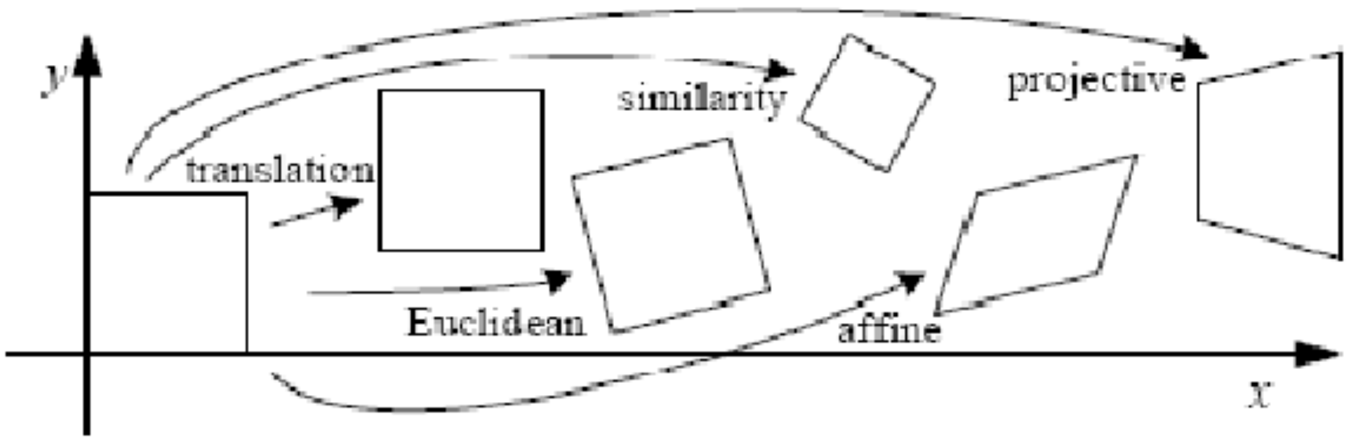
Lucas-Kanade flow

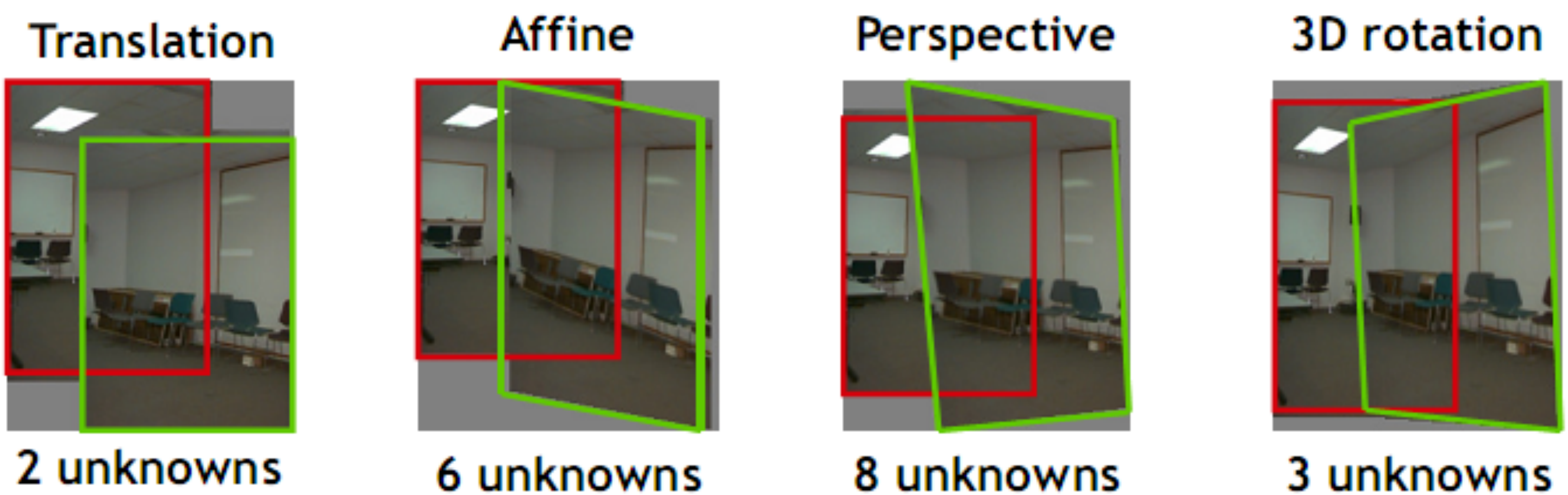
- Solve independently for each point [Lucas & Kanade 1981]

$$\frac{\partial E(u,v)}{\partial(u,v)} = 0 \Rightarrow \begin{bmatrix} \sum_{(x,y)\in\Omega} w(x,y) I_x^2 & \sum_{(x,y)\in\Omega} w(x,y) I_x I_y \\ \sum_{(x,y)\in\Omega} w(x,y) I_y I_x & \sum_{(x,y)\in\Omega} w(x,y) I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_{(x,y)\in\Omega} w(x,y) I_x I_t \\ \sum_{(x,y)\in\Omega} w(x,y) I_y I_t \end{bmatrix}$$

$$G_\sigma * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -G_\sigma * \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

Extension: Global Parametric Motion Models





Slide credit: Steve Seitz

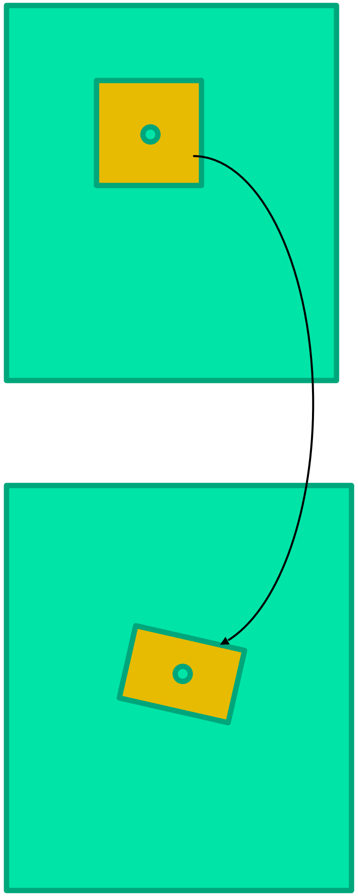
Affine Motion

- Affine Motion

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

- Substituting into the brightness constancy equation:

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



Affine Motion

- Affine Motion

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

- Substituting into the brightness constancy equation:

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

- Each pixel provides 1 linear constraint in 6 unknowns.

- Spatial coherence constrains, Least squares minimization:

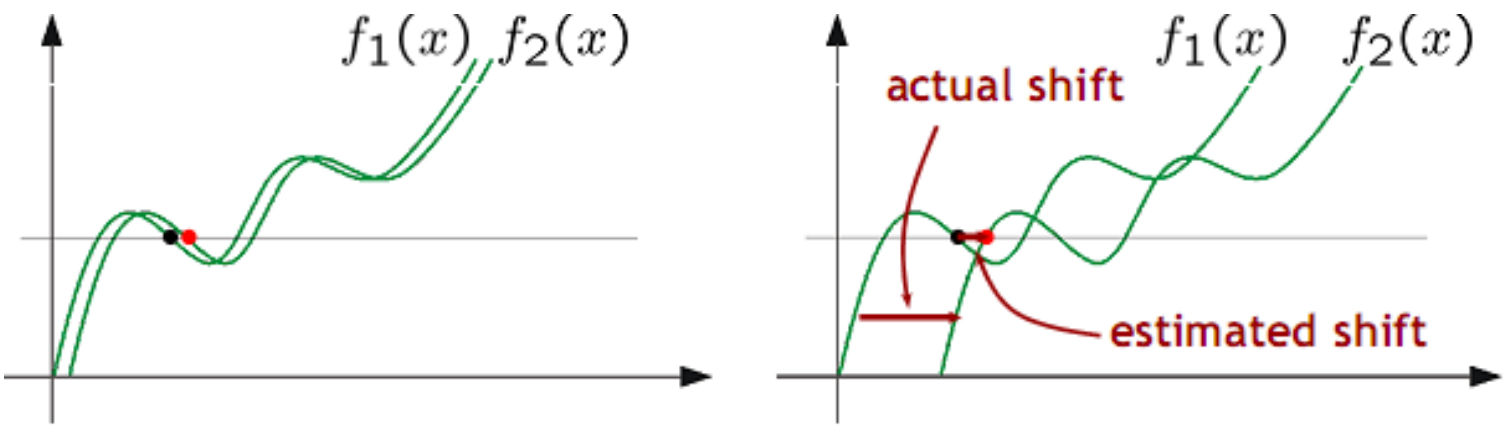
$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

Problem Cases in Lucas-Kanade

- **The motion is large** (larger than a pixel)
  - Iterative refinement, coarse-to-fine estimation
- **A point does not move like its neighbors**
  - Motion segmentation
- **Brightness constancy does not hold**
  - Do exhaustive neighborhood search with normalized correlation.

Dealing with Large Motions/ Temporal Aliasing

- Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity.
  - I.e., how do we know which 'correspondence' is correct?

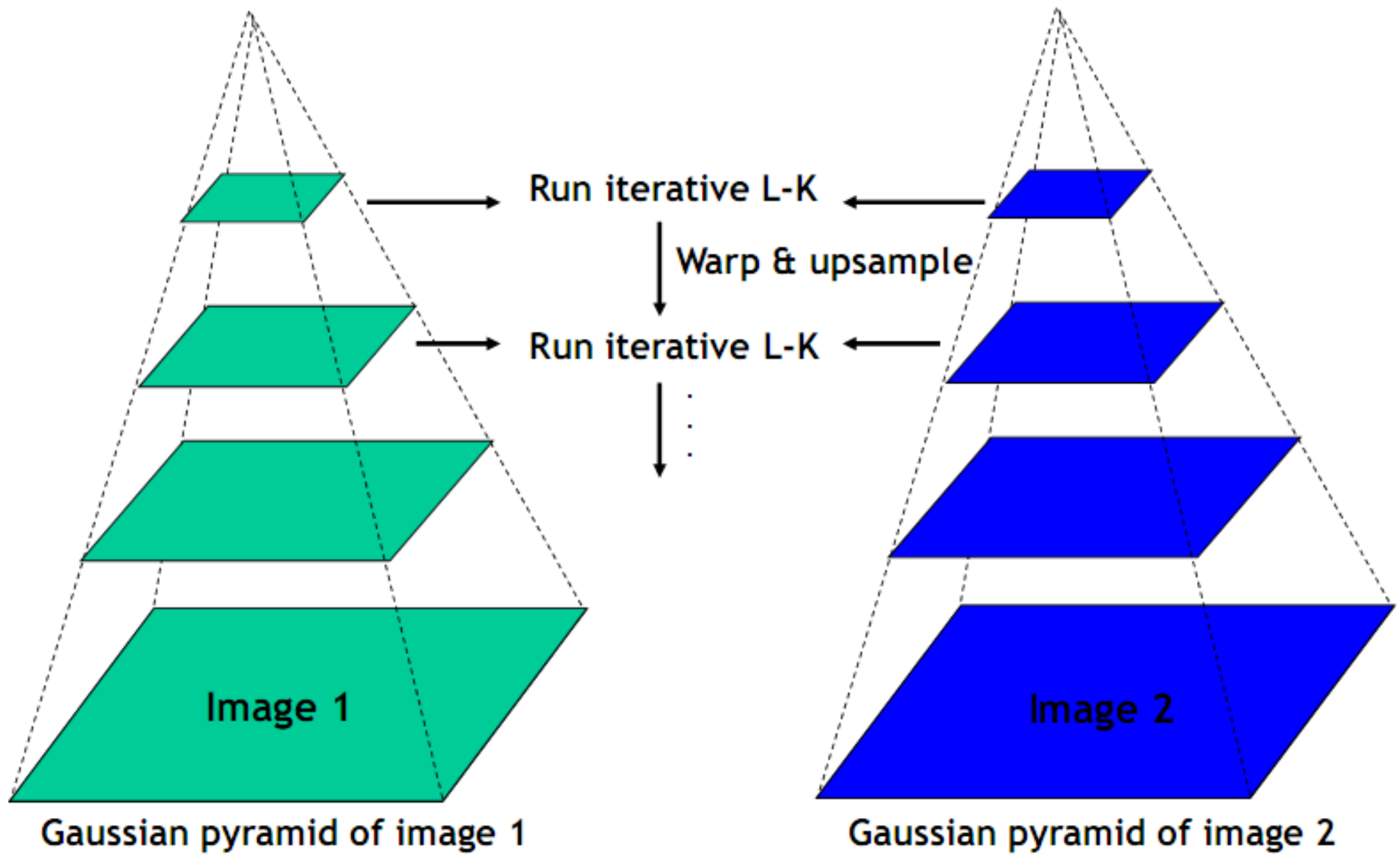


Nearest match is correct (no aliasing)

Nearest match is incorrect (aliasing)

To overcome aliasing: coarse-to-fine estimation.

Coarse-to-fine Optical Flow Estimation



Jean-Yves Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker*, TR, Intel, , 1997

Slide credit: Steve Seitz

Extension: Gradient constancy

Brightness is not always constant

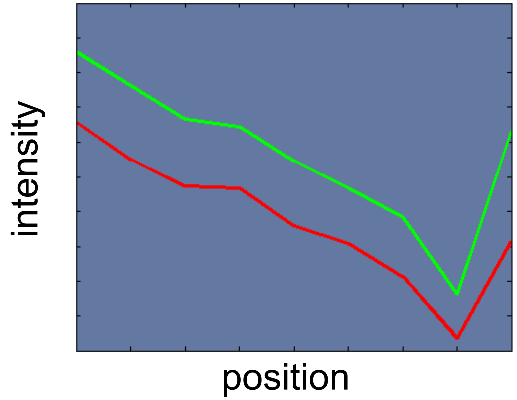
Rotating cylinder

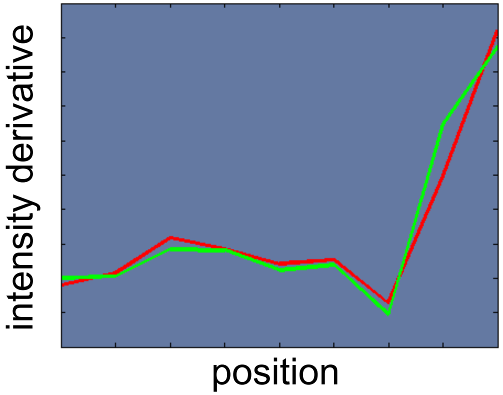
Brightness constancy does not always hold

Gradient constancy holds

$$I(x+u, y+v, t+1) \neq I(x, y, t)$$

$$\nabla I(x+u, y+v, t+1) = \nabla I(x, y, t)$$





Slide credit: D. Frolova and D. Simakov

- **Local constraints (data) + Local spatial coherence**

➢ **Brightness constancy**

$$I(x+u, y+v, t+1) - I(x, y, t) = 0$$

linearized

$$[u \quad v \quad 1] \quad J \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong 0$$

averaged linearized

$$\delta^2_{LIN+GAUSS} = [u \quad v \quad 1] \ (G_\rho * J) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong 0$$

Local spatial coherence

➢ **Gradient constancy**

$$\nabla I(x+u, y+v, t+1) - \nabla I(x, y, t) = 0$$

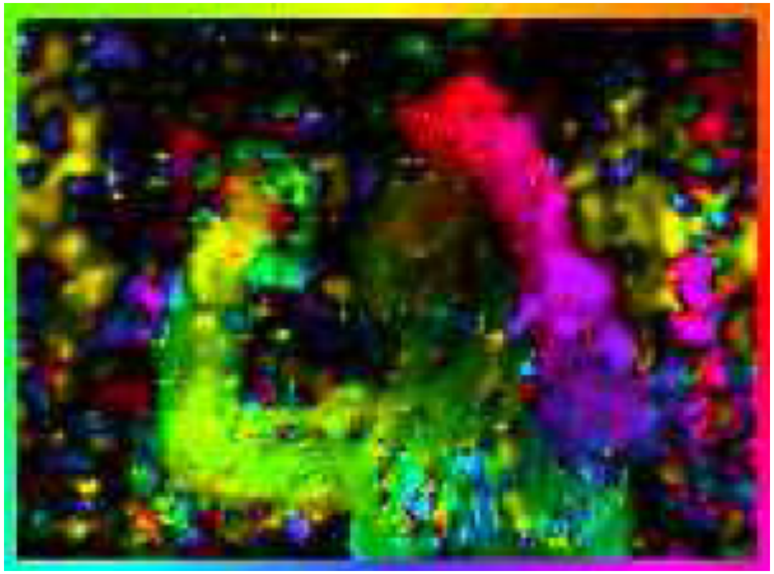
Slide credit: D. Frolova and D. Simakov

Local constraints are not enough!

- Local constraints work poorly

input video

Optical flow direction using only local constraints



color encodes direction as marked on the boundary

Slide credit: D. Frolova and D. Simakov

Where local constraints fail

1） Uniform regions

Motion is not observable in the image (locally)

2） “Aperture problem”

We can estimate only one flow component (normal)

Where local constraints fail

3） *Occlusions*

We have not seen where some points moved

Occluded regions are marked in red

Obtaining support from neighbors

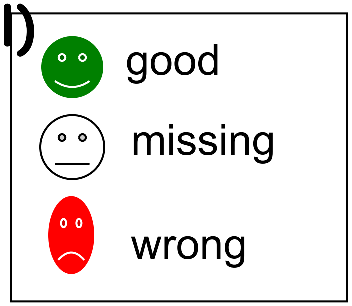
Two main problems with local constraints:

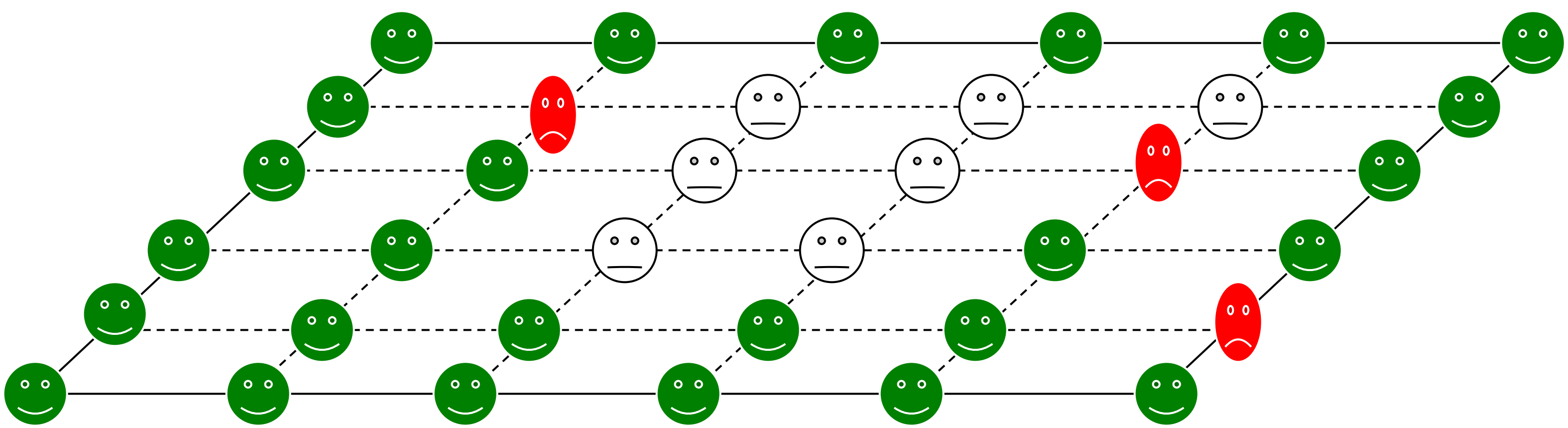
- information about motion is **missing** in some points

  => need **spatial coherency (large region/global)**

- constraints do not hold everywhere

  => need methods to **combine them robustly**





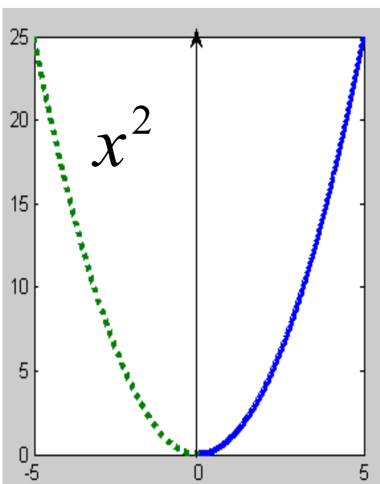
Robust combination of partially reliable data

Combination of two **flow constraints**

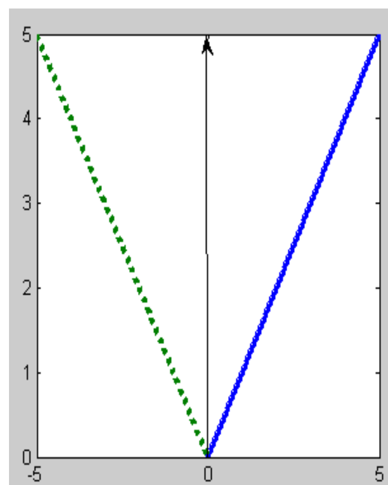
$$\min \int_{video} \phi\left(\left|I_{warped} - I\right|\right) + \alpha\phi\left(\left|\nabla I_{warped} - \nabla I\right|\right)$$

$$I_{warped} = I(x+u, y+v, t+1); \quad I = I(x,y,t)$$

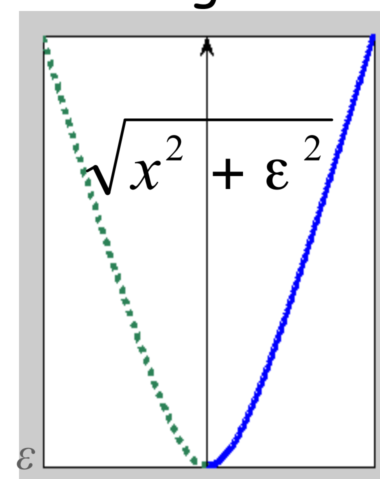
usual: L2



robust: L1



robust regularized



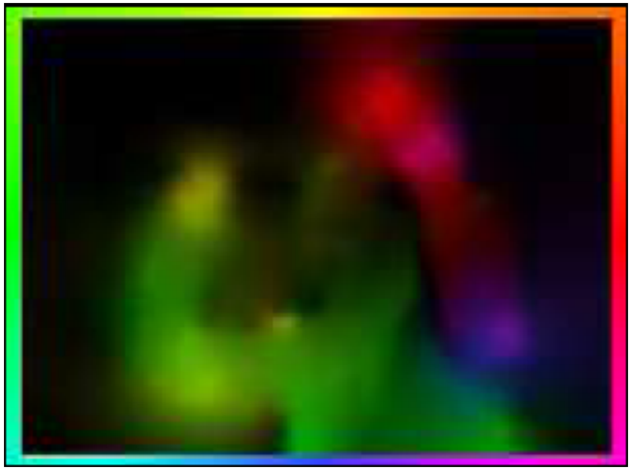
✓ easy to analyze and minimize – sensitive to outliers	✓ robust in presence of outliers – non-smooth: hard to analyze	✓ smooth: easy to analyze ✓ robust in presence of outliers

[A. Bruhn, J. Weickert, 2005]
Towards ultimate motion estimation: Combining highest accuracy with real-time performance

Homogeneous propagation

$$\min \int_{video} |\nabla u|^2 + |\nabla v|^2$$

$u(x, y, t)$- flow in the x direction
$v(x, y, t)$- flow in the y direction
∇ - gradient



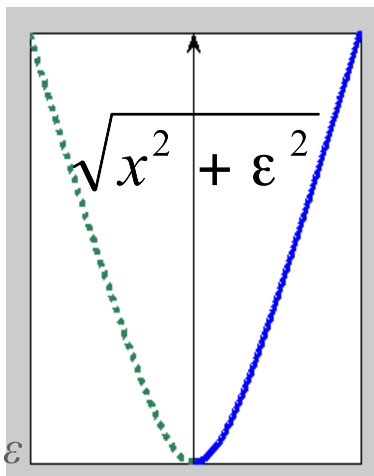
This constraint is not correct on motion boundaries
=> over-smoothing of the resulting flow

[Horn&Schunck 1981]

Robustness to flow discontinuities

$$\min \int_{video} \phi\left(\sqrt{|\nabla u|^2 + |\nabla v|^2}\right)$$

ϕ :



(also known as robust flow-driven regularization)

[T. Brox, A. Bruhn, N. Papenberg, J. Weickert, 2004]
High accuracy optical flow estimation based on a theory for warping

Combining ingredients

Local constraints

–Brightness constancy

–Image gradient constancy

–Local motion coherence

Spatial coherency (Global)

–Homogeneous

–Flow-driven (adaptive)

–Bilateral filtering + occlusions

– Image + flow driven

$$\text{Energy} = \int \phi\,(\text{Data}) + \int \phi\,(\text{“Smoothness”})$$

Combined using robust statistics ϕ

Computed coarse-to-fine

Use several frames

Slide credit: D. Frolova and D. Simakov

Combining Local and Global

Remember: $J = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_t \\ I_y I_x & I_y^2 & I_y I_t \\ I_t I_x & I_t I_y & I_t^2 \end{bmatrix}$

Lucas&Kanade (local data+smoothing)

$$\int_{video} [u \;\; v \;\; 1]\,(G_\rho * J)\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Horn&Schunk

$$\int_{video} [u \;\; v \;\; 1]\; J \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \alpha \int_{video} |\nabla u|^2 + |\nabla v|^2$$

$$\int_{video} [u \;\; v \;\; 1]\,(G_\rho * J)\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \alpha \int_{video} |\nabla u|^2 + |\nabla v|^2$$

Basic "Combining local and global"

[A. Bruhn, J. Weickert, C. Schnörr, 2002]

Slide credit: D. Frolova and D. Simakov

The more ingredients - the better

brightness constancy

gradient constancy

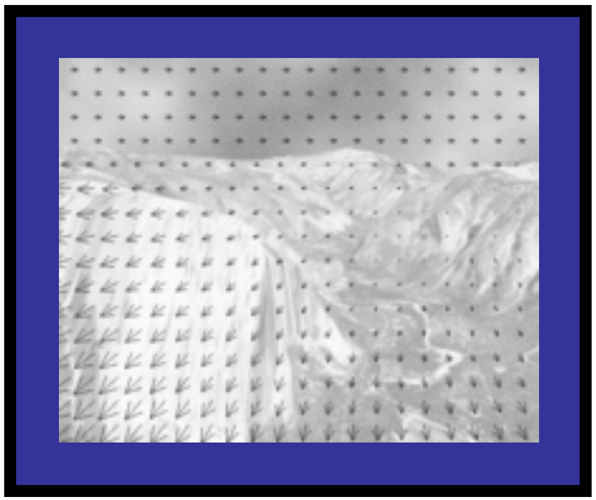
spatial coherence

$$\int_{video} \phi\left(\left|I_{warped} - I\right|\right) + \alpha\phi\left(\left|\nabla I_{warped} - \nabla I\right|\right) + \beta \int_{video} \phi\left(\left\|\nabla \mathrm{flow}\right\|\right)$$

Towards ultimate motion estimation: Combining highest accuracy with real-time performance [Bruhn, Weickert, 2005]

Slide credit: D. Frolova and D. Simakov

Quantitative results

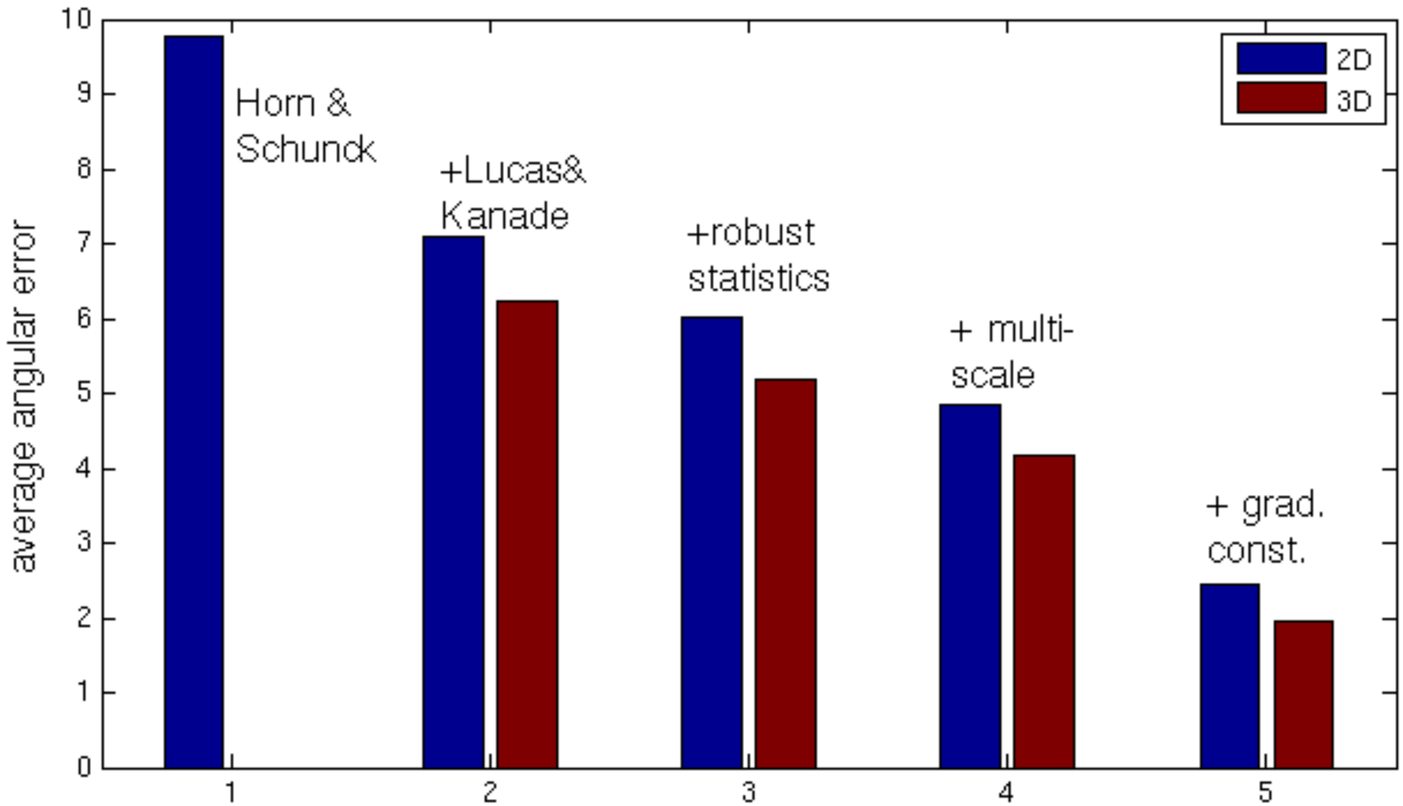


Yosemite sequence with clouds

Method / Angular error	Average	Standard Deviation
Lucas&Kanade	4.3 (density 35%)	
Horn&Schunk	9.8	16.2
Combining local and global	4.2	7.7
"Towards ultimate …"	2.4	6.7

Average error decreases, but standard deviation is still high….

Influence of each ingredient



For Yosemite sequence with clouds

- Feature Tracking
  (sparse optical flow)

Tracking Challenges

- Ambiguity of optical flow
  - Find good features to track
- Large motions
  - Discrete search instead of Lucas-Kanade
- Changes in shape, orientation, color
  - Allow some matching flexibility
- Occlusions, disocclusions
  - Need mechanism for deleting, adding new features
- Drift – errors may accumulate over time
  - Need to know when to terminate a track

Handling Large Displacements

- Define a small area around a pixel as the template.
- Match the template against each pixel within a search area in next image – just like stereo matching!
- Use a match measure such as SSD or correlation.
- After finding the best discrete location, can use Lucas-Kanade to get **sub-pixel estimate**. Kanade to get sub-pixel estimate.

Tracking Over Many Frames

- Select features in first frame
- For each frame:
  - Update positions of tracked features
    - Discrete search or Lucas-Kanade
  - Terminate inconsistent tracks
    - Compute similarity with corresponding feature in the previous frame or in the first frame where it's visible
- Start new tracks if needed
    - Typically every ~10 frames, new features are added to "refill the ranks"

Shi-Tomasi Feature Tracker

- Find good features using eigenvalues of second-moment matrix
  - Key idea: "good" features to track are the ones that can be tracked reliably.
- From frame to frame, track with Lucas-Kanade and a pure translation model.
  - More robust for small displacements, can be estimated from smaller neighborhoods.
- Check consistency of tracks by affine registration to the first observed instance of the feature.
  - Affine model is more accurate for larger displacements.
  - Comparing to the first frame helps to minimize drift.

J. Shi and C. Tomasi. **Good Features to Track**. CVPR 1994

Real-Time GPU Implementations

- This basic feature tracking framework (Lucas-Kanade + Shi-Tomasi) is commonly referred to as "KLT tracking".
  - Used as preprocessing step for many applications (recall the Structure-from-Motion pipeline)
  - Lends itself to easy parallelization
- Very fast GPU implementations available
  - C. Zach, D. Gallup, J.-M. Frahm, Fast Gain-Adaptive KLT tracking on the GPU. In CVGPU'08 Workshop, Anchorage, USA, 2008
  - 216 fps with automatic gain adaptation
  - 260 fps without gain adaptation

http://www.cs.unc.edu/~ssinha/Research/GPU_KLT/
http://cs.unc.edu/~cmzach/opensource.html

KLT—Pyramidal tracking algorithm (Coarse-to-fine)

Goal: Let $\mathbf{u}$ be a point on image I. Find its corresponding location $\mathbf{v}$ on image J

Build pyramid representations of I and J: $\{I^L\}_{L=0,\dots,L_m}$ and $\{J^L\}_{L=0,\dots,L_m}$

Initialization of pyramidal guess: $\mathbf{g}^{L_m} = [g_x^{L_m} \;\; g_x^{L_m}]^T = [0 \;\; 0]^T$

for $L = L_m$ **down to** 0 **with step of** -1

Location of point $\mathbf{u}$ on image I^L: $\mathbf{u}^L = [p_x \;\; p_y]^T = \mathbf{u}/2^L$

Derivative of I^L with respect to x: $I_x(x,y) = \dfrac{I^L(x+1,y) - I^L(x-1,y)}{2}$

Derivative of I^L with respect to y: $I_y(x,y) = \dfrac{I^L(x,y+1) - I^L(x,y-1)}{2}$

Spatial gradient matrix:

$$G = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)\, I_y(x,y) \\ I_x(x,y)\, I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

Initialization of iterative L-K: $\overline{\nu}^0 = [0 \;\; 0]^T$

for $k = 1$ **to** K **with step of** 1 (or until $\|\overline{\eta}^k\| <$ accuracy threshold)

Image difference: $\delta I_k(x,y) = I^L(x,y) - J^L(x + g_x^L + \nu_x^{k-1}, y + g_y^L + \nu_y^{k-1})$

Image mismatch vector: $$\overline{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x,y)\, I_x(x,y) \\ \delta I_k(x,y)\, I_y(x,y) \end{bmatrix}$$

Optical flow (Lucas-Kanade): $\overline{\eta}^k = G^{-1}\,\overline{b}_k$

Guess for next iteration: $\overline{\nu}^k = \overline{\nu}^{k-1} + \overline{\eta}^k$

end of for-loop on k

Final optical flow at level L*:* $\mathbf{d}^L = \overline{\nu}^K$

Guess for next level $L-1$*:* $\mathbf{g}^{L-1} = [g_x^{L-1} \;\; g_y^{L-1}]^T = 2\,(\mathbf{g}^{\mathbf{L}} + \mathbf{d}^L)$

end of for-loop on L

Final optical flow vector: $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$

Location of point on J*:* $\mathbf{v} = \mathbf{u} + \mathbf{d}$

Solution: The corresponding point is at location $\mathbf{v}$ on image J

Dense Optical Flow

- Dense measurements can be obtained by adding smoothness constraints.

T. Brox, C. Bregler, J. Malik, **Large displacement optical flow**, CVPR'09, Miami, USA, June 2009.

Motion vs. Stereo

Similarities

- Both involve solving
  - Correspondence:
    (disparities, motion vectors)
  - Reconstruction
    (Depth, SFM, image alignment)

Differences

- Motion:
  - Uses velocity: consecutive frames must be close to get good approximate time derivative.
  - 3D movement between camera and scene not necessarily single 3D rigid transformation.
- Stereo:
  - Could have any disparity value.
  - View pair separated by a single 3d transformation. (global transformation)

Summary

- **30 years** of Optical Flow : a lot of useful ingredients were developed:
  - ➢ Local constraints:
    - ✓ brightness constancy
    - ✓ gradient constancy
    - ✓ Multi-channel & complex features & CNN features
  - ➢ Smoothing techniques:
    - ✓ homogeneous (local coherence)
    - ✓ flow (&image)-driven (preserving discontinuities)
    - ✓ bilateral filters
    - ✓ handling of occlusions
  - ➢ Learning flow prior (& Ground-Truth flow dataset)
  - ➢ Interactive labeling
  - ➢ Robust functions
  - ➢ Multi-scale trick
  - ➢ **Spatial** & Temporal Constraints
- All ingredients are combined an a global Energy Minimization approach
- This difficult global optimization can be done very fast using Multigrid.

References and Further Reading

[1] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proc. IJCAI, pp. 674–679, 1981.

[2] B.K.P. Horn and B.G. Schunck, Determining optical flow, Artificial Intelligence, Vol.17, pp.185-204, 1981.

[3] A. Bruhn, J. Weickert, C. Schnörr, Combining the advantages of local and global optic flow methods (“Lucas/Kanade meets Horn/Schunck”) , 2002 - 2005

[4] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, 2004 - 2005

[5] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, C. Schnörr, Real-Time Optic Flow Computation with Variational Methods, 2003 - 2005

[6] A. Bruhn, J. Weickert, Towards ultimate motion estimation: Combining highest accuracy with real-time performance, 2005

[7] J.Xiao, H.Cheng, H.Sawhney, C.Rao, M.Isnardi, Bilateral filtering-based optical flow estimation with occlusion detection, 2006

[8] Roth and Black, On the Spatial Statistics of Optical Flow, ICCV 2005.

[9] J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

A Hierarchy of Models

Taxonomy by Bergen,Anandan et al.'92

- **Parametric motion models**
  - 2D translation, affine, projective, 3D pose [Bergen, Anandan, et.al.'92]
- Piecewise parametric motion models
  - 2D parametric motion/structure layers [Wang&Adelson'93, Ayer&Sawhney'95]
- Quasi-parametric
  - 3D R, T & depth per pixel. [Hanna&Okumoto'91]
  - Plane+parallax [Kumar et.al.'94, Sawhney'94]
- Piecewise quasi-parametric motion models
  - 2D parametric layers + parallax per layer [Baker et al.'98]
- **Non-parametric**

– Optic flow: 2D vector per pixel [Lucas&Kanade'81, Bergen,Anandan et.al.'92]

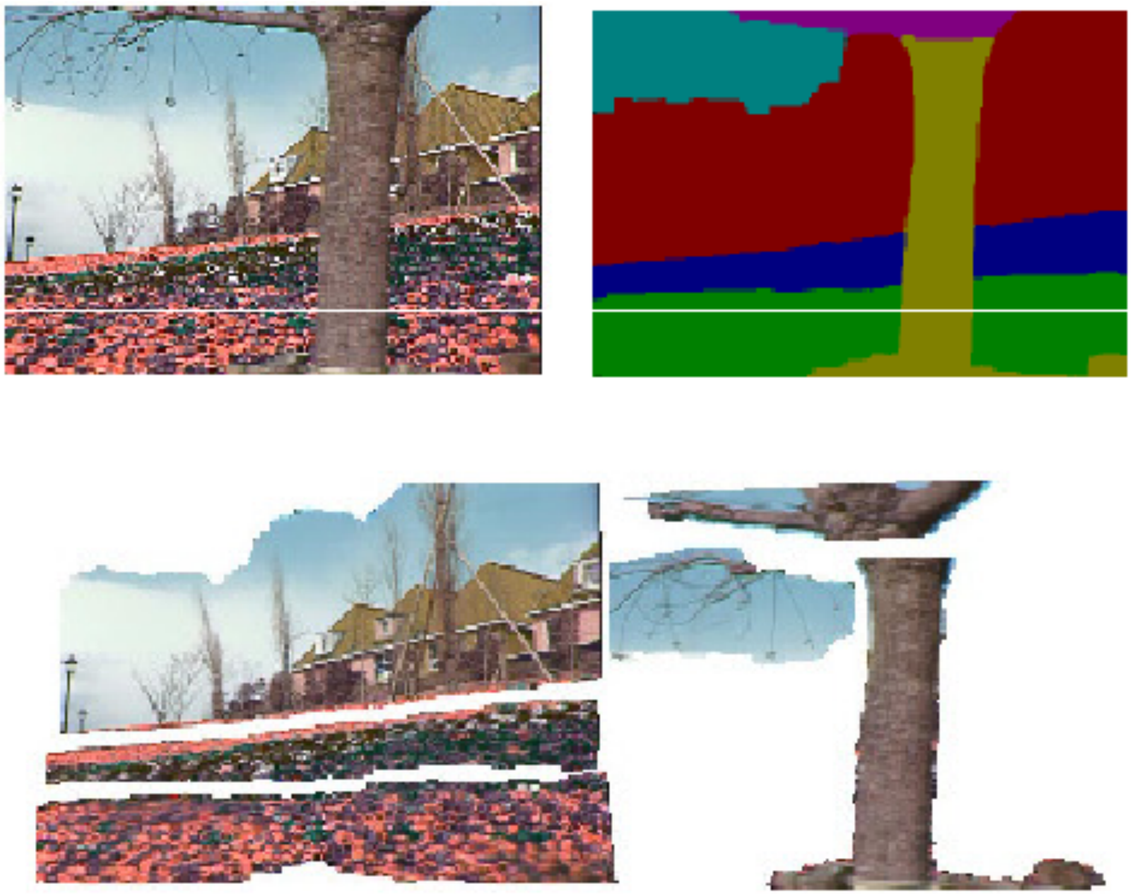
Motion Segmentation

- How do we represent the motion in this scene?

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

Layered Motion

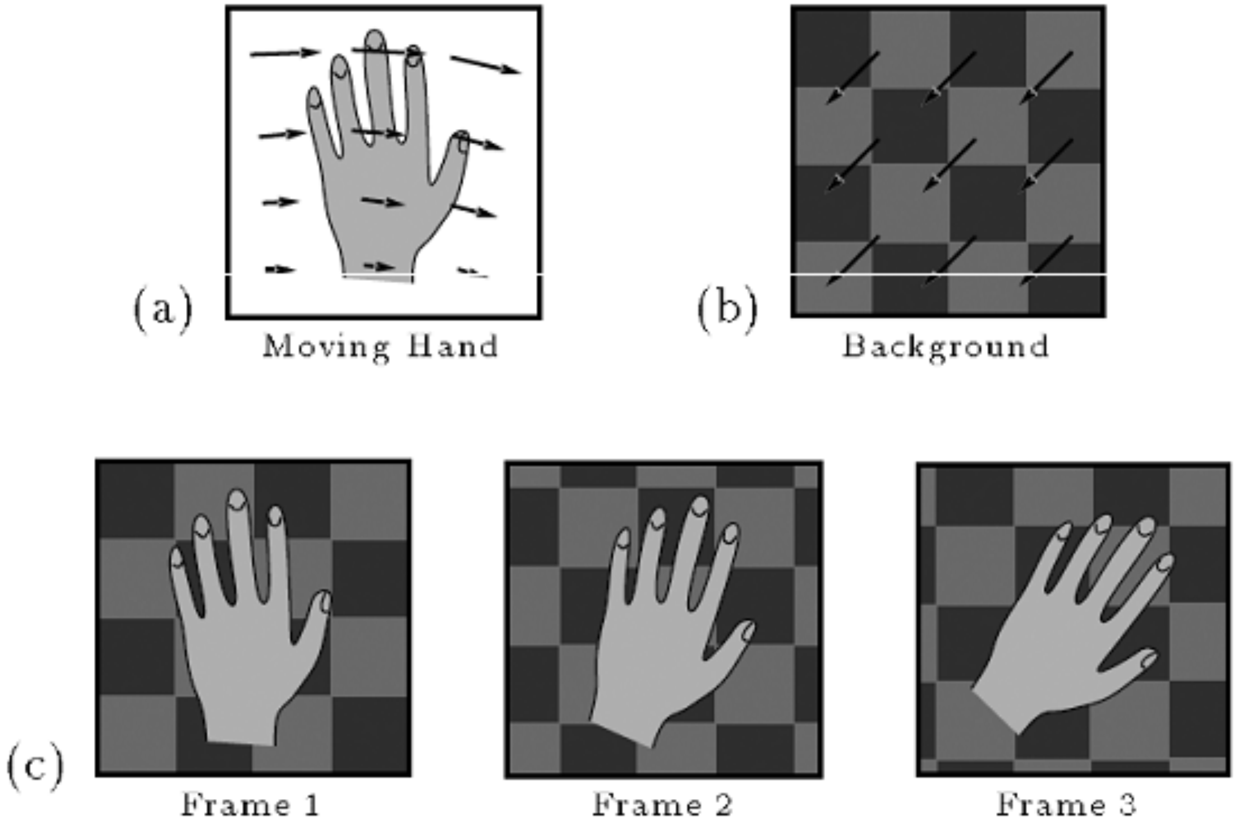
- Break image sequence into “layers” each of which has a **coherent motion**



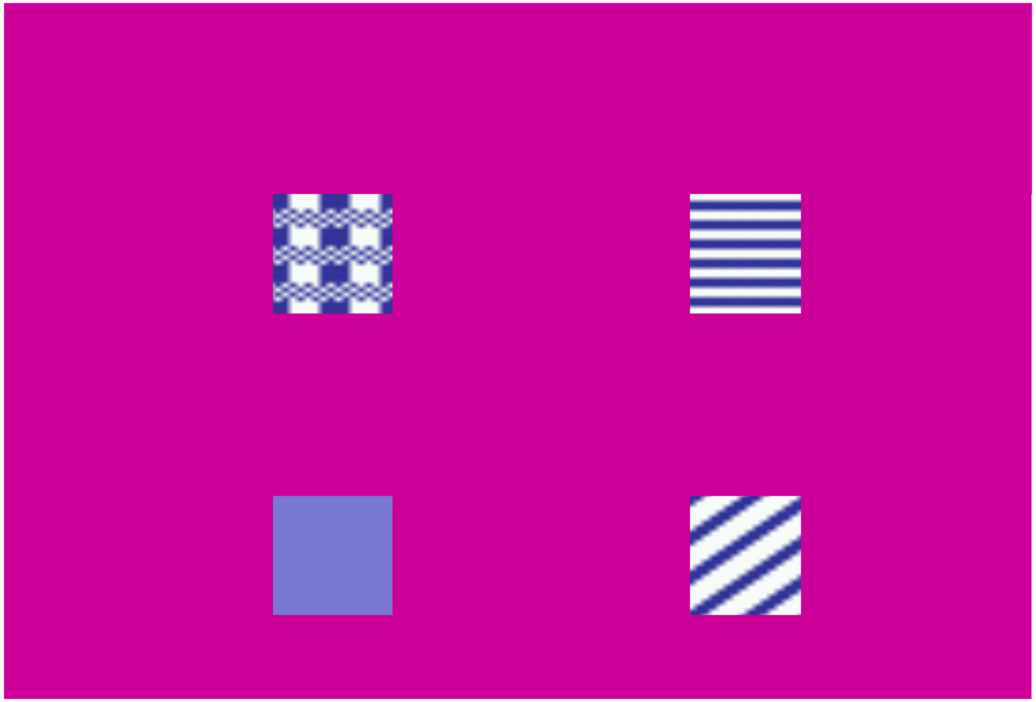
J. Wang and E. Adelson. Layered Representation for Motion Analysis. CVPR 1993.

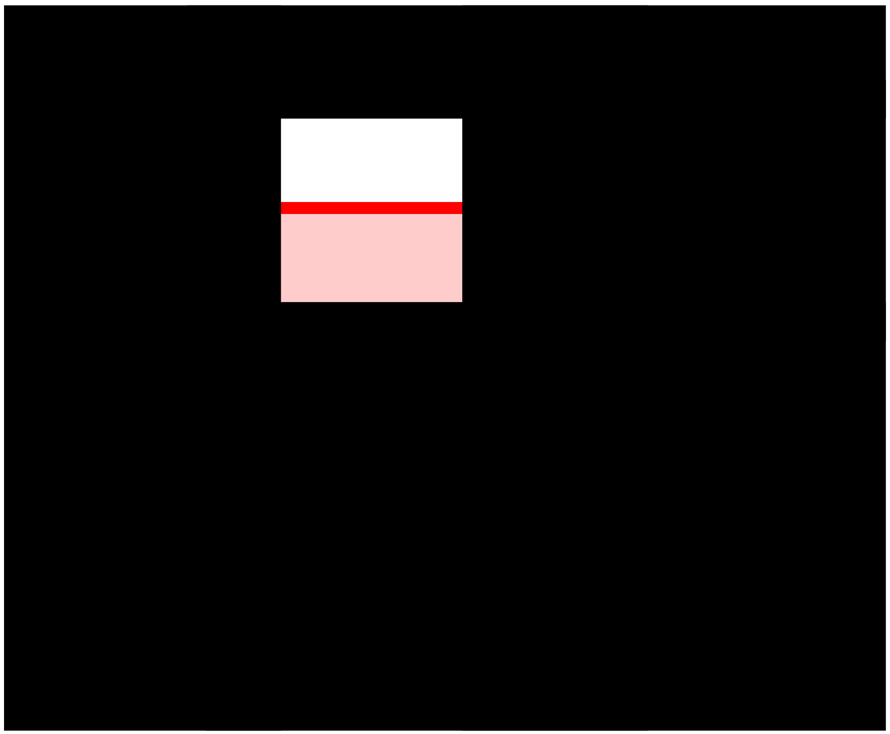
What Are Layers?

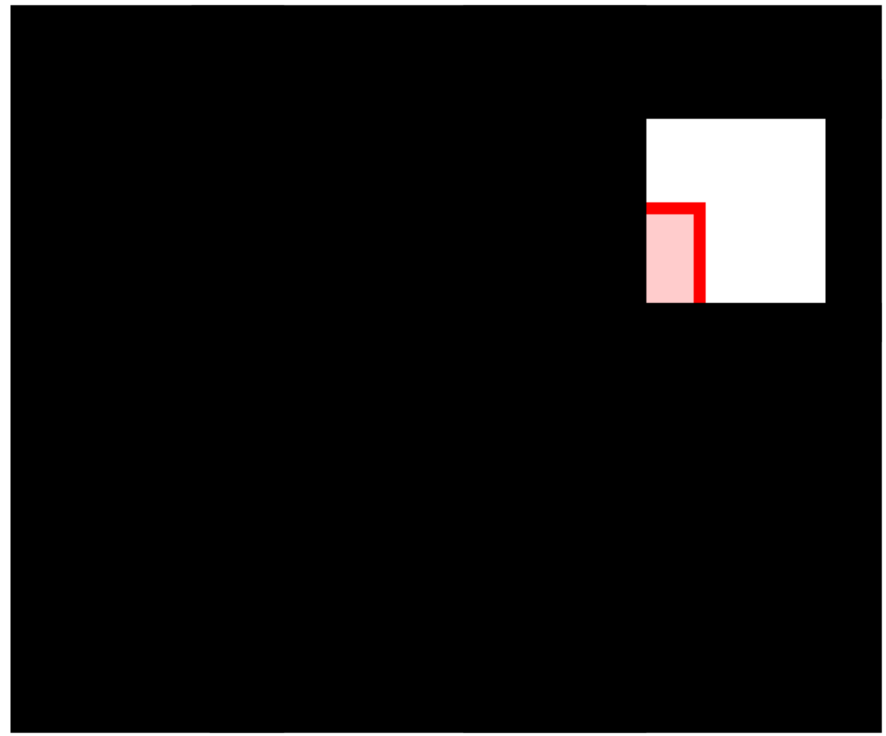
- Each layer is defined by an alpha mask and an affine motion model



The Aperture Problem







Information is propagated from regions with high certainty (e.g., corners) to regions with low certainty.

Such info propagation can cause optical illusions…