# Motion Estimation (I)

Ce Liu
celiu@microsoft.com
Microsoft Research New England

# We live in a moving world

- Perceiving, understanding and predicting motion is an important part of our daily lives

# Motion estimation: a core problem of computer vision

- Related topics:
  - Image correspondence, image registration, image matching, image alignment, …
- Applications
  - Video enhancement: stabilization, denoising, super resolution
  - 3D reconstruction: structure from motion (SFM)
  - Video segmentation
  - Tracking/recognition
  - Advanced video editing (label propagation)

# Contents (today)

- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

# Contents (next time)

- Discrete optical flow
- Layer motion analysis
- Contour motion analysis
- Obtaining motion ground truth
- SIFT flow: generalized optical flow
- Applications (2)

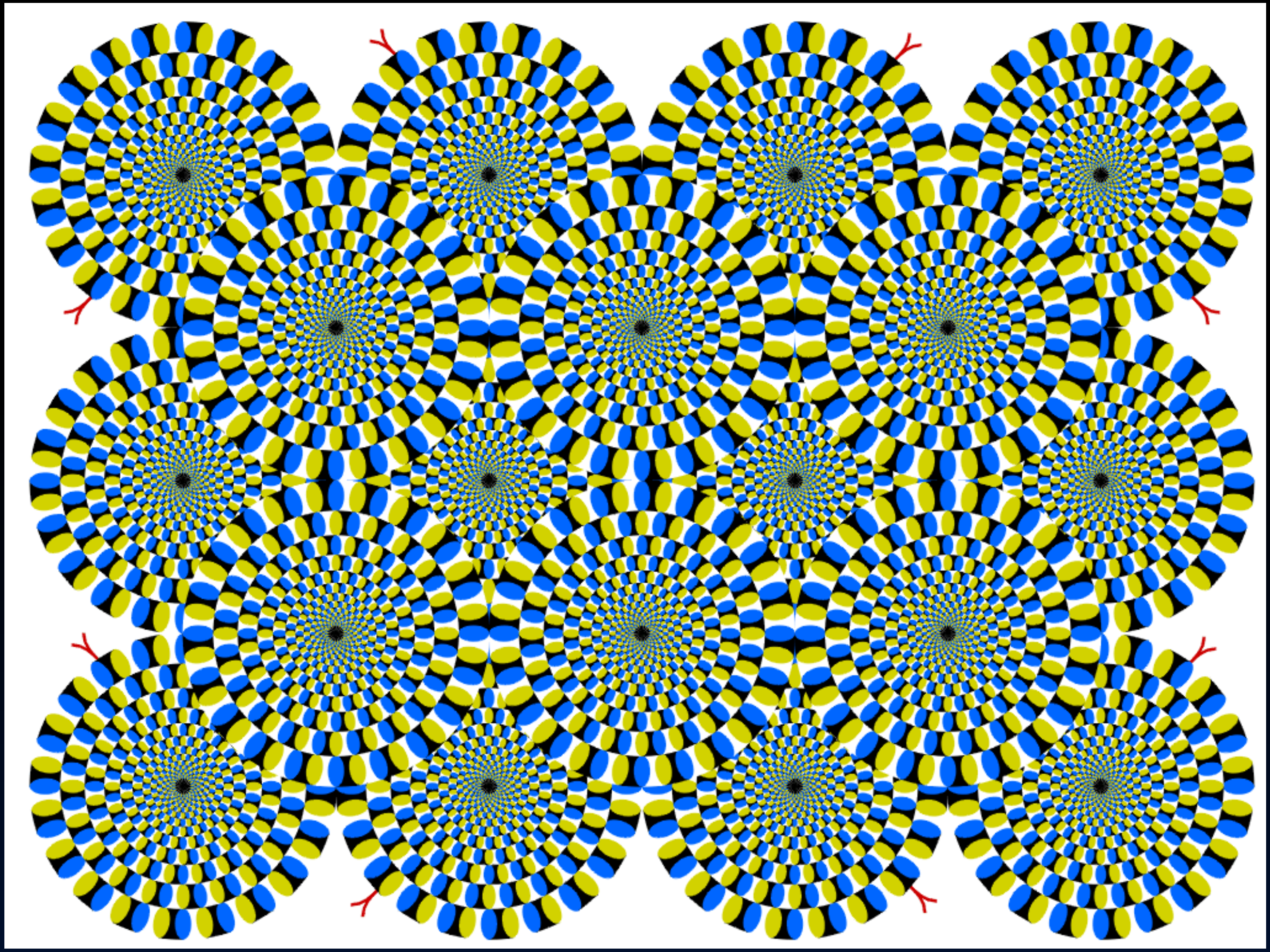
# Readings

- Rick's book: Chapter 8
- Ce Liu's PhD thesis (appendix A & B)
- S. Baker and I. Matthews. Lucas-Kanade 20 years on: a unifying framework. IJCV 2004
- Horn-Schunck (wikipedia)
- A. Bruhn, J. Weickert, C. Schnorr. Lucas/Kanade meets Horn/Schunk: combining local and global optical flow methods. IJCV 2005

# Contents

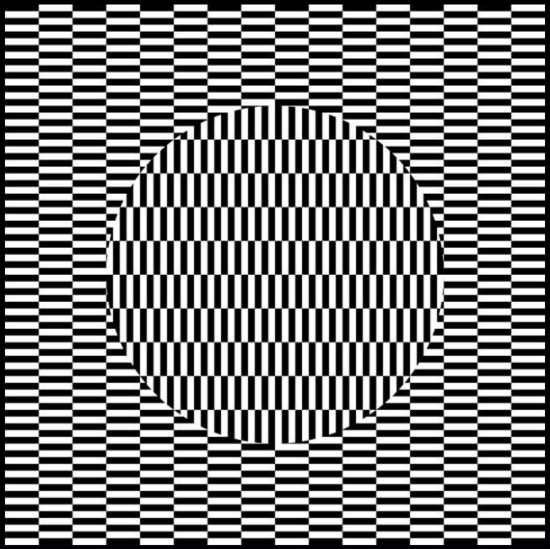
- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

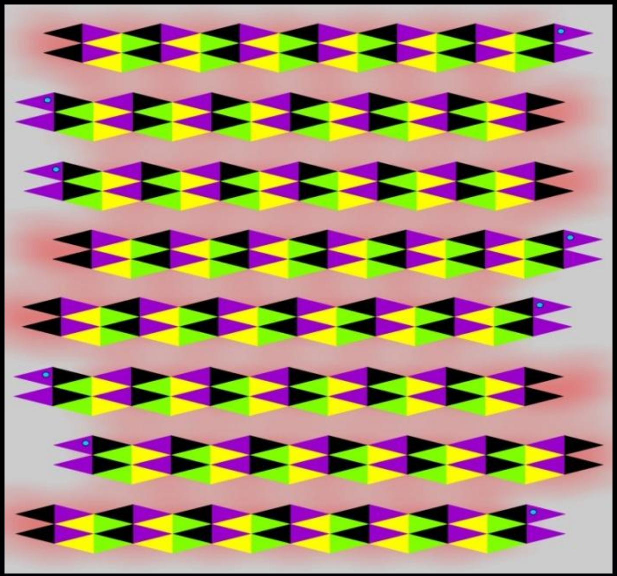
# Seeing motion from a static picture?

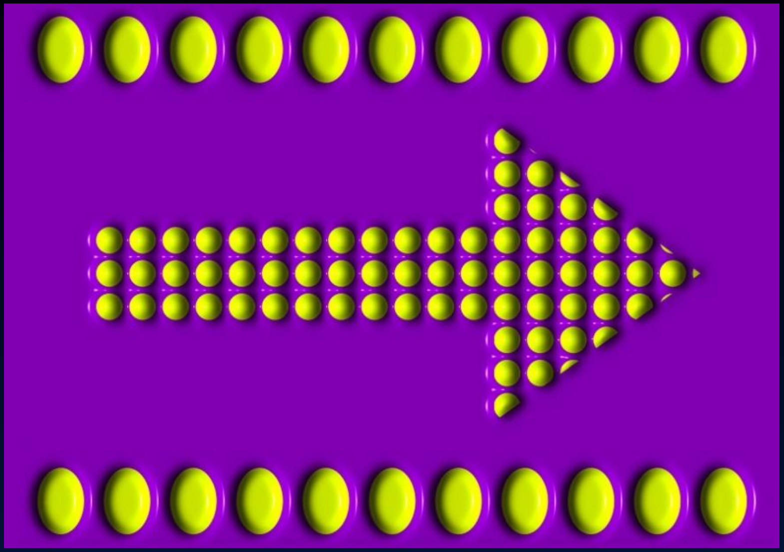


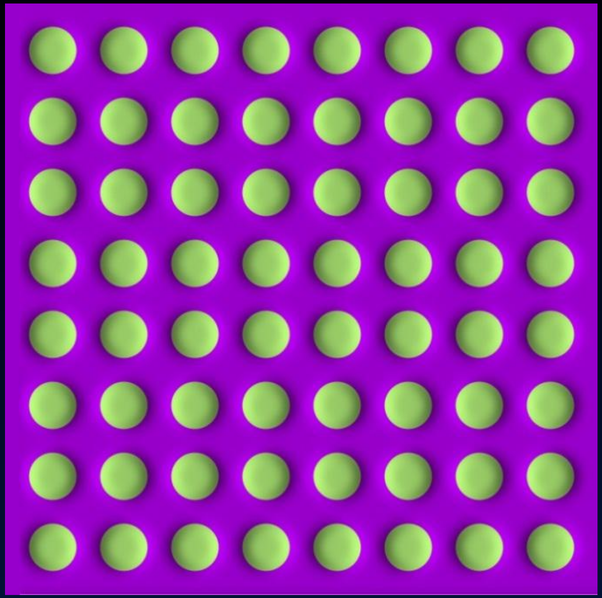
http://www.ritsumei.ac.jp/~akitaoka/index-e.html

# More examples



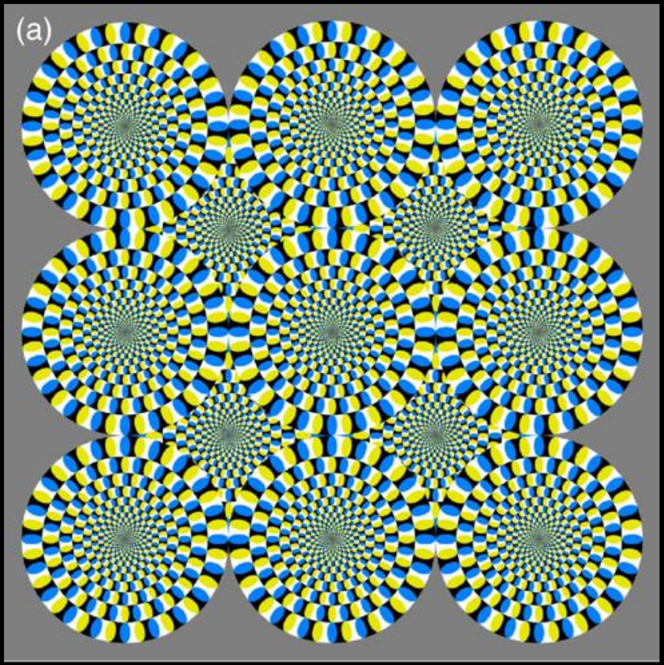


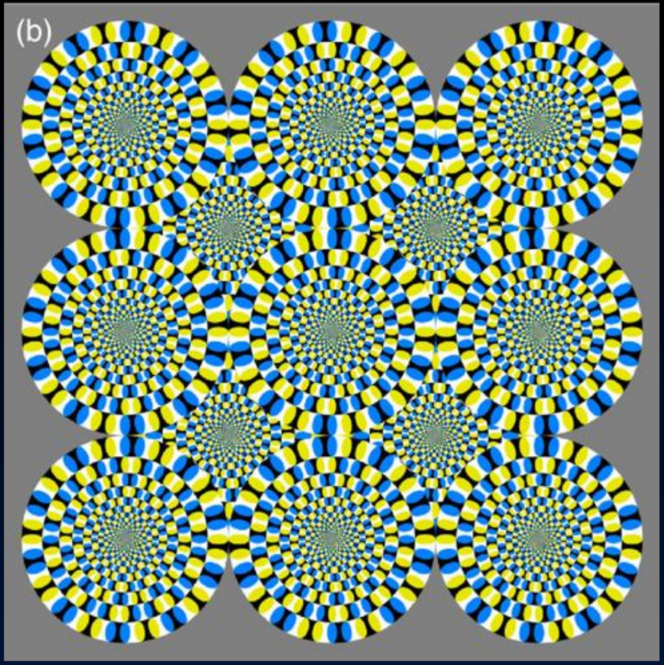




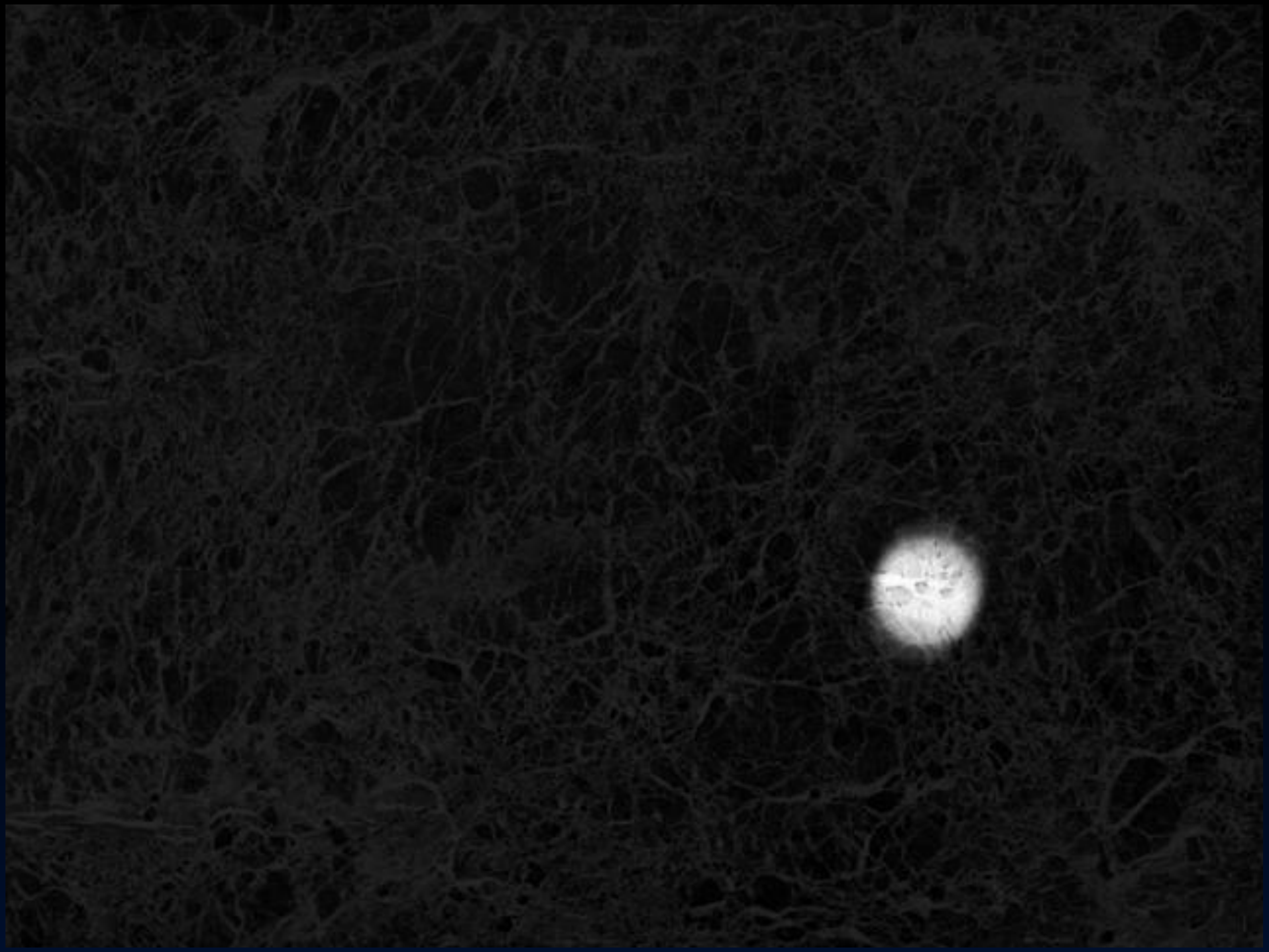
# How is this possible?

- The true mechanism is to be revealed
- FMRI data suggest that illusion is related to some component of eye movements
- We don't expect computer vision to "see" motion from these stimuli, yet

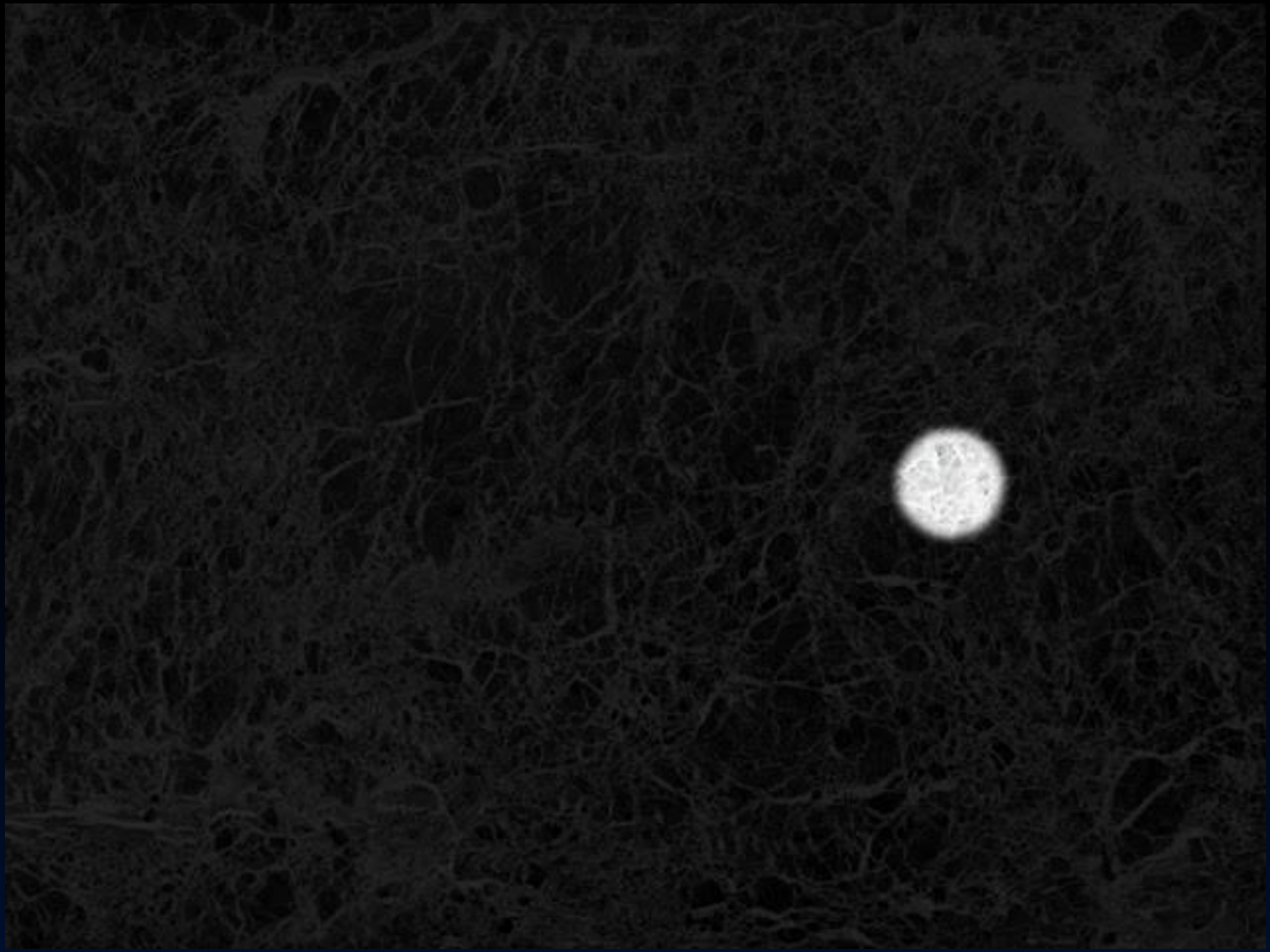


# What do you see?



# In fact, …



# We still don't touch these areas

# Motion analysis: human vs. computer

- Computers can only analyze motion for opaque and solid objects
- Challenges:
  - Shapeless or transparent scenes
- Key: motion representation

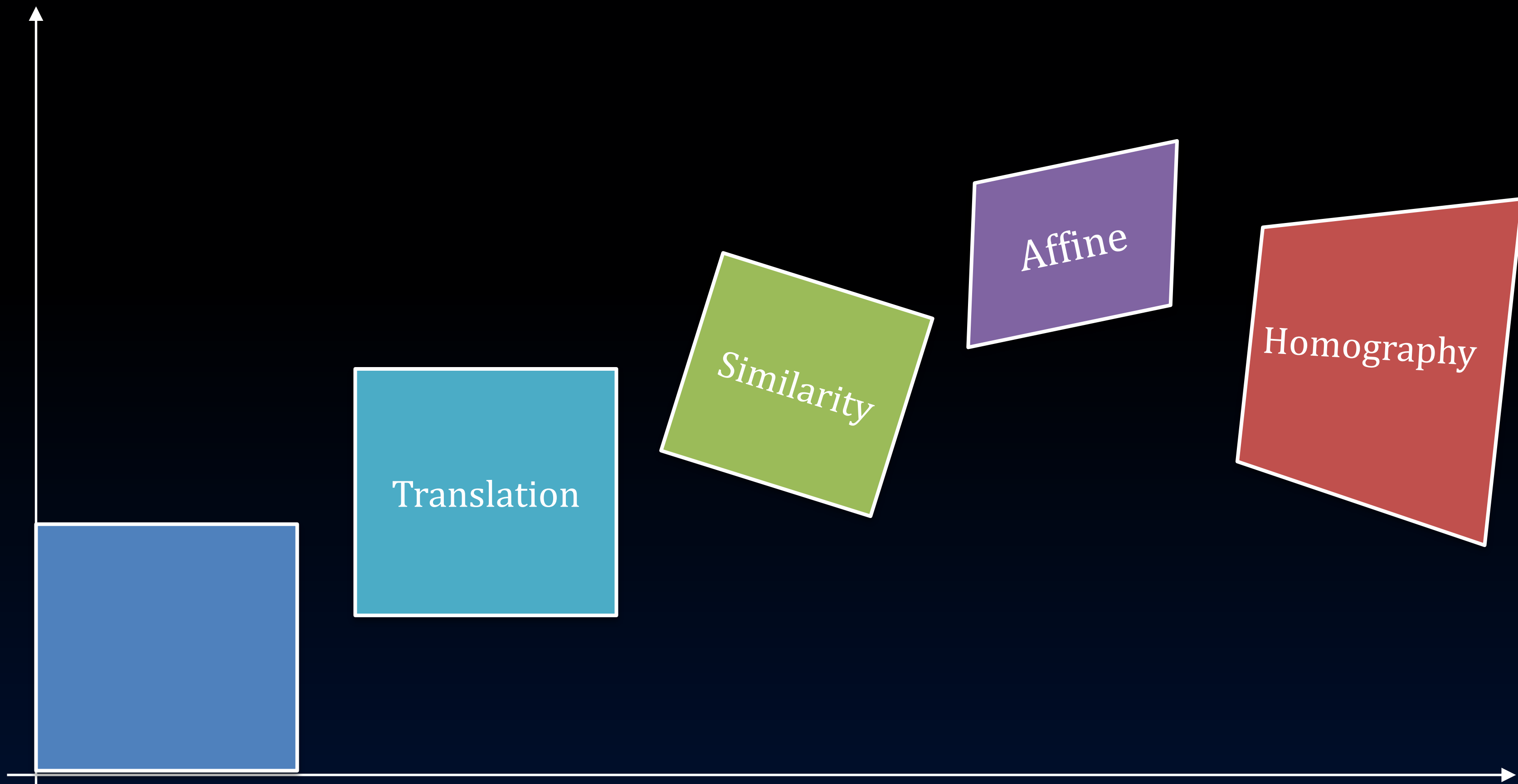
# Contents

- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

# Motion forms

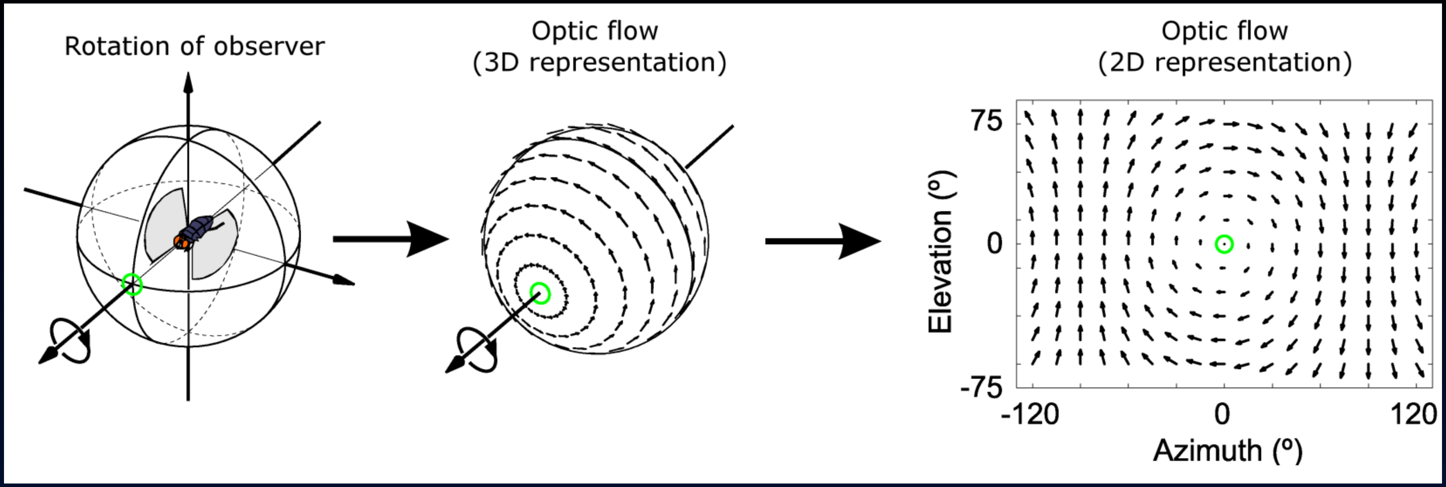
- Mapping: $(x_1, y_1) \rightarrow (x_2, y_2)$
- Global parametric motion: $(x_2, y_2) = f(x_1, y_1; \theta)$
- Motion types
  - Translation: $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + a \\ y_1 + b \end{bmatrix}$
  - Similarity: $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = s \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_1 + a \\ y_1 + b \end{bmatrix}$
  - Affine: $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} ax_1 + by_1 + c \\ dx_1 + ey_1 + f \end{bmatrix}$
  - Homography: $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} ax_1 + by_1 + c \\ dx_1 + ey_1 + f \end{bmatrix}, z = gx_1 + hy_1 + i$

# Illustration of motion types



# Optical flow field

- Parametric motion is limited and cannot describe the motion of arbitrary videos
- Optical flow field: assign a flow vector $\left(u(x,y),v(x,y)\right)$ to each pixel $(x,y)$
- Projection from 3D world to 2D

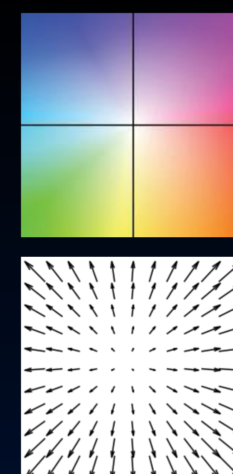


# Optical flow field visualization

- Too messy to plot flow vector for every pixel
- Map flow vector to color
  - Magnitude: saturation
  - Orientation: hue

Input

Ground-truth flow field



Visualization code
[Baker et al. 2007]

# Matching criterion

- Brightness constancy assumption

  $$I_1(x, y) = I_2(x + u, y + v) + r + g$$

  $$r \sim N(0, \sigma^2), g \sim U(-1,1)$$

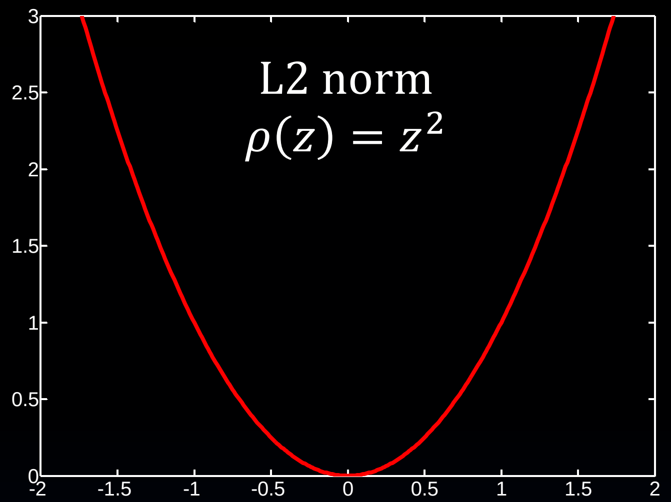
  Noise $r$, outlier $g$ (occlusion, lighting change)

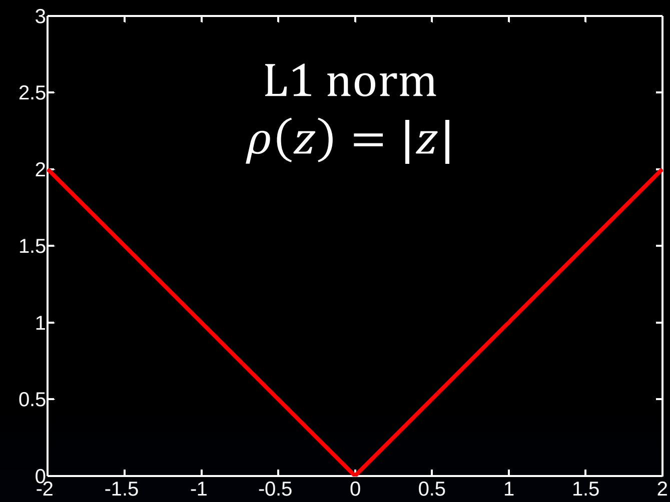
- Matching criteria
  - What's invariant between two images?
    - Brightness, gradients, phase, other features…
  - Distance metric (L2, L1, truncated L1, Lorentzian)

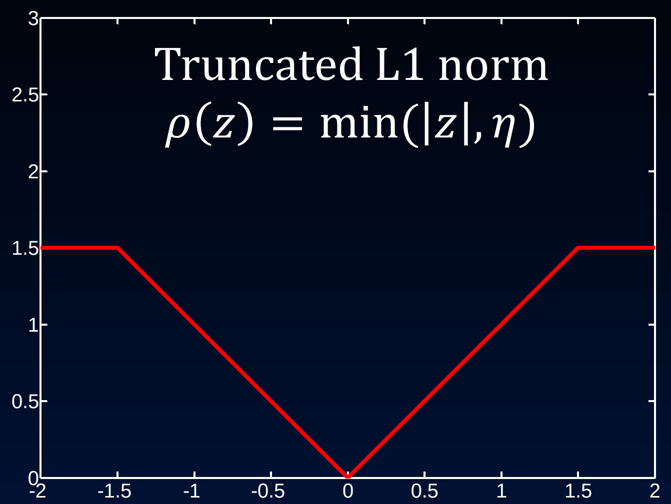
    $$E(u, v) = \sum_{x,y} \rho(I_1(x, y) - I_2(x + u, y + v))$$

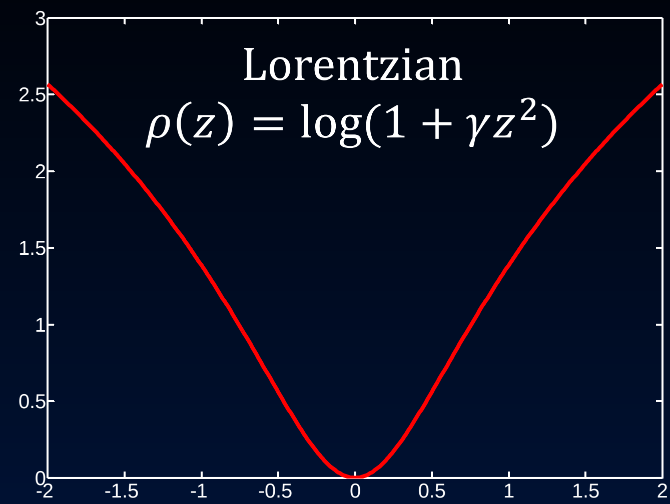
  - Correlation, normalized cross correlation (NCC)

# Error functions



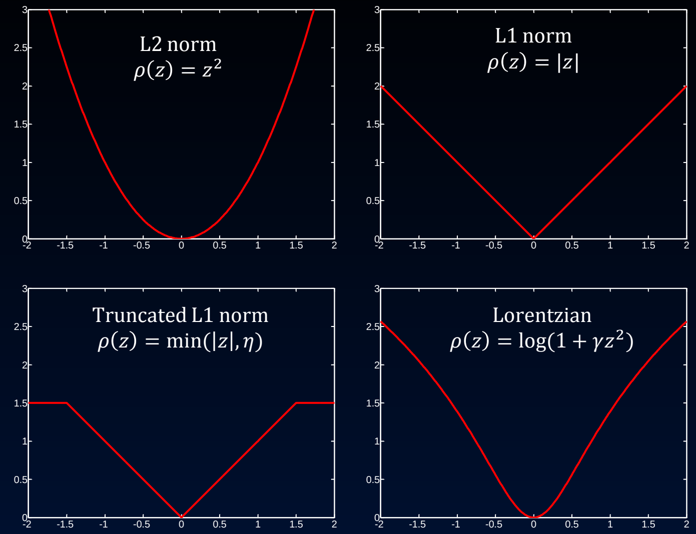



# Robust statistics

- Traditional L2 norm: only noise, no outlier
- Example: estimate the average of
  0.95, 1.04, 0.91, 1.02, 1.10, 20.01
- Estimate with minimum error

$$z^* = \arg\min_z \sum_i \rho(z - z_i)$$

  – L2 norm: $z^* = 4.172$
  – L1 norm: $z^* = 1.038$
  – Truncated L1: $z^* = 1.0296$
  – Lorentzian: $z^* = 1.0147$



# Contents

- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

# Lucas-Kanade: problem setup

- Given two images $I_1(x, y)$ and $I_2(x, y)$, estimate a parametric motion that transforms $I_1$ to $I_2$
- Let $\mathbf{x} = (x, y)^T$ be a column vector indexing pixel coordinate
- Two typical transforms
  - Translation: $W(\mathrm{x}; \mathrm{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$
  - Affine: $W(\mathrm{x}; \mathrm{p}) = \begin{bmatrix} p_1 x + p_2 y + p_3 \\ p_4 x + p_5 y + p_6 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
- Goal of the Lucas-Kanade algorithm

$$\mathrm{p}^* = \arg\min_{\mathrm{p}} \sum_{\mathrm{x}} \left[ I_2\big(W(\mathrm{x}; \mathrm{p})\big) - I_1(\mathrm{x}) \right]^2$$

# An incremental algorithm

- Difficult to directly optimize the objective function

$$\mathrm{p}^* = \arg\min_{\mathrm{p}} \sum_{\mathrm{x}} \left[ I_2\big(W(\mathrm{x};\mathrm{p})\big) - I_1(\mathrm{x}) \right]^2$$

- Instead, we try to optimize each step

$$\Delta\mathrm{p}^* = \arg\min_{\Delta\mathrm{p}} \sum_{\mathrm{x}} \left[ I_2\big(W(\mathrm{x};\mathrm{p}+\Delta\mathrm{p})\big) - I_1(\mathrm{x}) \right]^2$$

- The transform parameter is updated:

$$\mathrm{p} \leftarrow \mathrm{p} + \Delta\mathrm{p}^*$$

# Taylor expansion

- The term $I_2\big(W(\mathrm{x};\mathrm{p}+\Delta\mathrm{p})\big)$ is highly nonlinear
- Taylor expansion:

$$I_2\big(W(\mathrm{x};\mathrm{p}+\Delta\mathrm{p})\big) \approx I_2\big(W(x;p)\big) + \nabla I_2 \frac{\partial W}{\partial \mathrm{p}} \Delta\mathrm{p}$$

- $\frac{\partial W}{\partial \mathrm{p}}$ : *Jacobian* of the warp
- If $W(\mathrm{x};\mathrm{p}) = \big(W_x(\mathrm{x};\mathrm{p}), W_y(\mathrm{x};\mathrm{p})\big)^T$ , then

$$\frac{\partial W}{\partial \mathrm{p}} = \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \cdots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \cdots & \frac{\partial W_y}{\partial p_n} \end{bmatrix}$$

# Jacobian matrix

- For affine transform: $W(\mathrm{x};\mathrm{p}) = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

  The Jacobian is $\frac{\partial W}{\partial \mathrm{p}} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix}$

- For translation : $W(\mathrm{x};\mathrm{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$

  The Jacobian is $\frac{\partial W}{\partial \mathrm{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

# Taylor expansion

- $\nabla I_2 = \left[I_x \; I_y\right]$ is the gradient of image $I_2$ evaluated at $W(\mathrm{x};\mathrm{p})$: compute the gradients in the coordinate of $I_2$ and warp back to the coordinate of $I_1$

- For affine transform $\frac{\partial W}{\partial \mathrm{p}} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix}$

$$\nabla I_2 \frac{\partial W}{\partial \mathrm{p}} = [I_x x \quad I_x y \quad I_x \quad I_y x \quad I_y y \quad I_y]$$

- Let matrix $\mathbf{B} = [\mathbf{I}_x\mathbf{X} \;\; \mathbf{I}_x\mathbf{Y} \;\; \mathbf{I}_x \;\; \mathbf{I}_y\mathbf{X} \;\; \mathbf{I}_y\mathbf{Y} \;\; \mathbf{I}_y] \in \mathbb{R}^{n\times 6}$, $\mathbf{I}_x$ and $\mathbf{X}$ are both column vectors. $\mathbf{I}_x\mathbf{X}$ is element-wise vector multiplication.

# Gauss-Newton

- With Taylor expansion, the objective function becomes

$$\Delta \mathrm{p}^* = \arg\min_{\Delta \mathrm{p}} \sum_{\mathrm{x}} \left[ I_2\big(W(x;p)\big) + \nabla I_2 \frac{\partial W}{\partial \mathrm{p}} \Delta \mathrm{p} \; - I_1(\mathrm{x}) \right]^2$$

Or in a vector form:

$$\boxed{\Delta \mathrm{p}^* = \arg\min_{\Delta \mathrm{p}} (\mathbf{I}_t + \mathbf{B}\Delta \mathrm{p})^T (\mathbf{I}_t + \mathbf{B}\Delta \mathrm{p})}$$

Where $\mathbf{B} = [\mathbf{I}_x\mathbf{X} \;\; \mathbf{I}_x\mathbf{Y} \;\; \mathbf{I}_x \;\; \mathbf{I}_y\mathbf{X} \;\; \mathbf{I}_y\mathbf{Y} \;\; \mathbf{I}_y] \in \mathbb{R}^{n\times 6}$

$$\mathbf{I}_t = \mathbf{I}_2\big(\mathbf{W}(\mathrm{p})\big) - \mathbf{I}_1$$

- Solution:

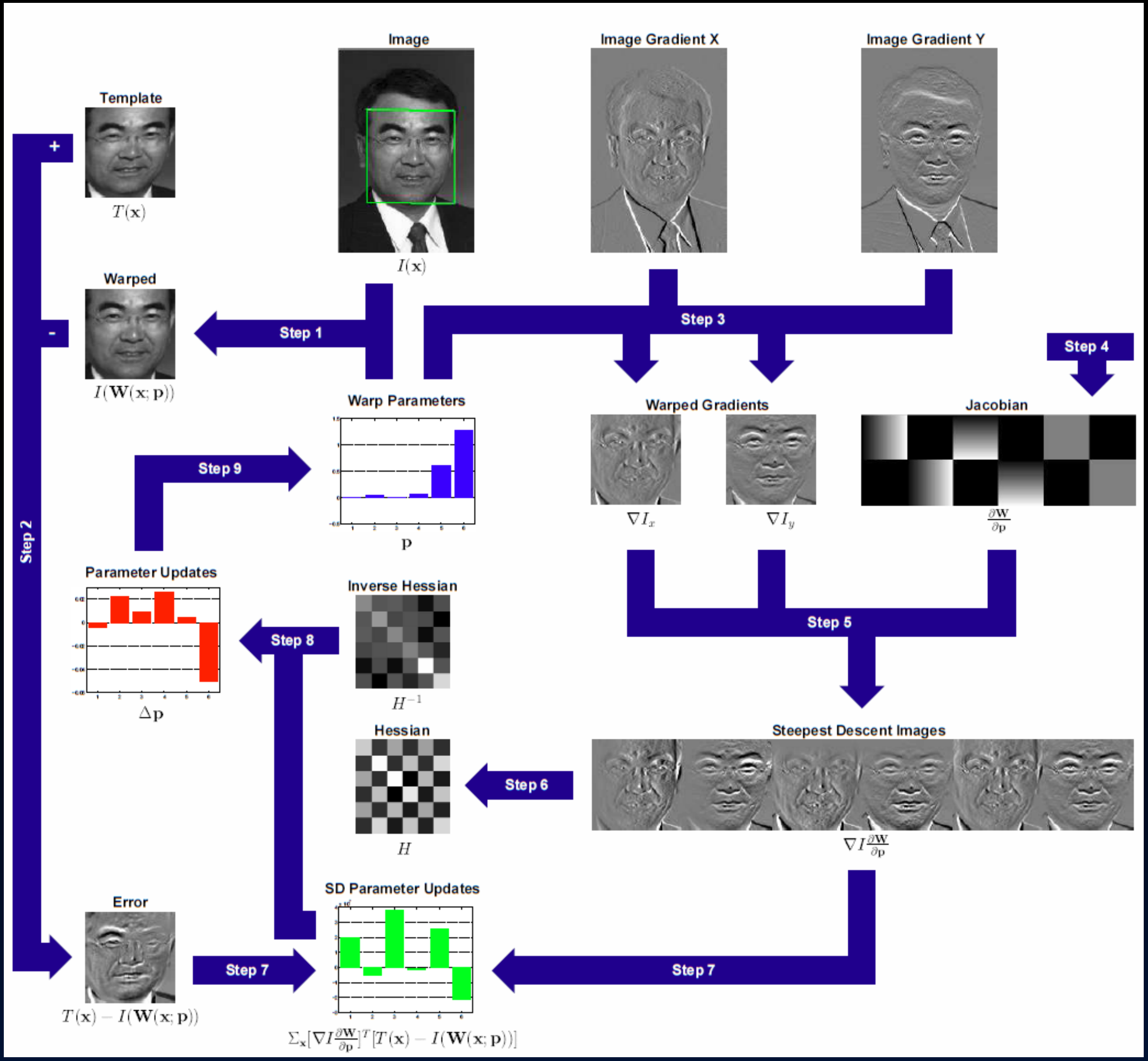
$$\boxed{\Delta \mathrm{p}^* = -(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{I}_t}$$

# Translation

- Jacobian: $\frac{\partial W}{\partial \mathrm{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- $\nabla I_2 \frac{\partial W}{\partial \mathrm{p}} = [I_x \quad I_y]$
- $\mathbf{B} = [\mathbf{I}_x \ \ \mathbf{I}_y] \in \mathbb{R}^{n \times 2}$
- Solution:

$$\begin{aligned} \Delta \mathrm{p}^* &= -(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{I}_t \\ &= -\begin{bmatrix} \mathbf{I}_x^T\mathbf{I}_x & \mathbf{I}_x^T\mathbf{I}_y \\ \mathbf{I}_x^T\mathbf{I}_y & \mathbf{I}_y^T\mathbf{I}_y \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_x^T\mathbf{I}_t \\ \mathbf{I}_y^T\mathbf{I}_t \end{bmatrix} \end{aligned}$$

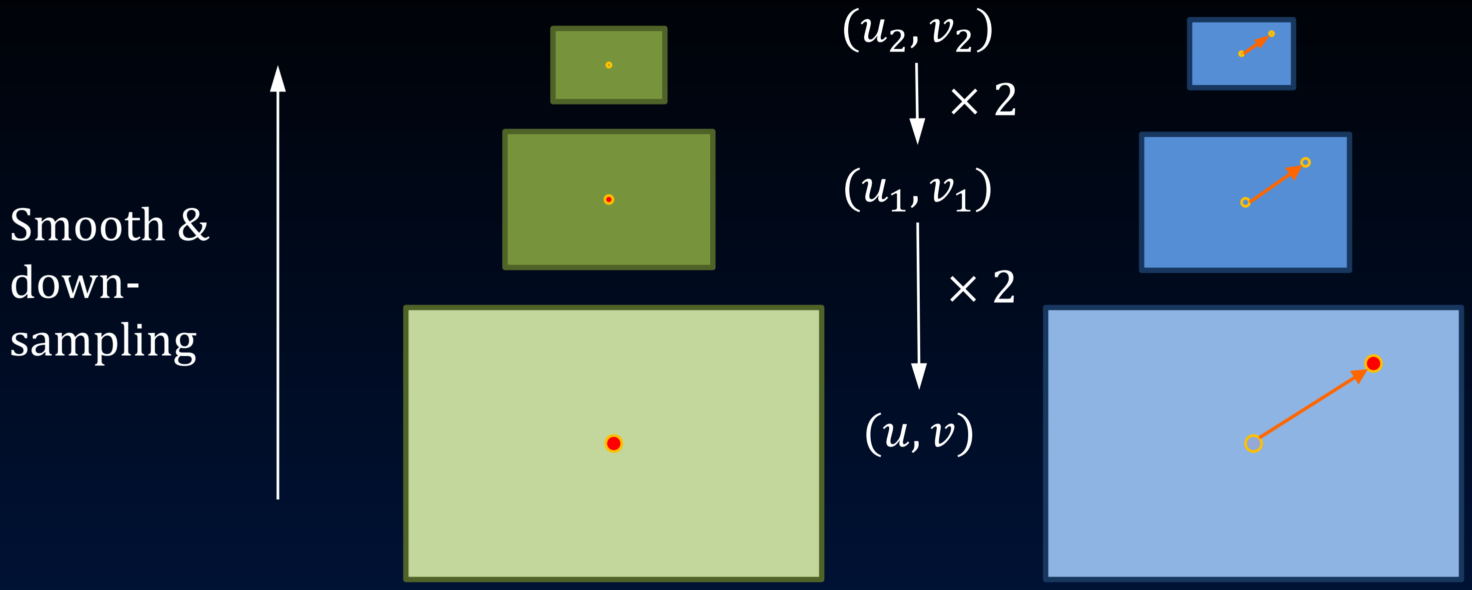
# How it works



Template
$T(\mathbf{x})$
Image
$I(\mathbf{x})$
Image Gradient X
Image Gradient Y
Warped
$I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
Step 1
Step 3
Step 4
Warp Parameters
$\mathbf{p}$
Warped Gradients
$\nabla I_x$
$\nabla I_y$
Jacobian
$\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
Step 9
Step 2
+
-
Parameter Updates
$\Delta\mathbf{p}$
Inverse Hessian
$H^{-1}$
Step 8
Step 5
Hessian
$H$
Step 6
Steepest Descent Images
$\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
SD Parameter Updates
Error
$T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
Step 7
Step 7
$\Sigma_{\mathbf{x}}[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$

# Coarse-to-fine refinement

- Lucas-Kanade is a greedy algorithm that converges to local minimum
- Initialization is crucial: if initialized with zero, then the underlying motion must be small
- If underlying transform is significant, then coarse-to-fine is a must



# Variations

- Variations of Lucas Kanade:
  - Additive algorithm [Lucas-Kanade, 81]
  - Compositional algorithm [Shum & Szeliski, 98]
  - Inverse compositional algorithm [Baker & Matthews, 01]
  - Inverse additive algorithm [Hager & Belhumeur, 98]
- Although inverse algorithms run faster (avoiding re-computing Hessian), they have the same complexity for robust error functions!

# From parametric motion to flow field

- Incremental flow update $(du, dv)$ for pixel $(x, y)$

$$I_2(x + u + du, y + v + dv) - I_1(x, y)$$
$$= I_2(x + u, y + v) + I_x(x + u, y + v)du + I_y(x + u, y + v)dv - I_1(x, y)$$

$$\boxed{I_x du + I_y dv + I_t = 0}$$

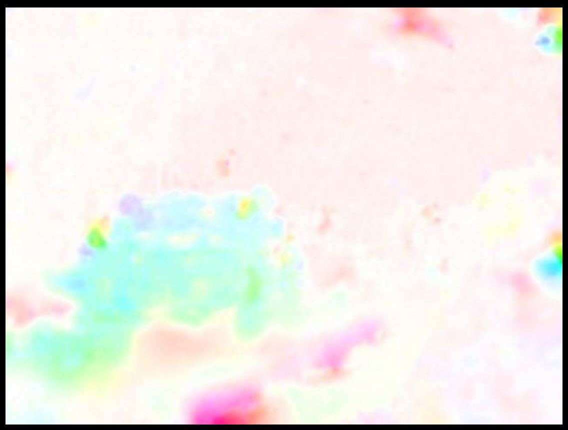
- We obtain the following function within a patch

$$\boxed{\begin{bmatrix} du \\ dv \end{bmatrix} = -\begin{bmatrix} \mathbf{I}_x^T\mathbf{I}_x & \mathbf{I}_x^T\mathbf{I}_y \\ \mathbf{I}_x^T\mathbf{I}_y & \mathbf{I}_y^T\mathbf{I}_y \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_x^T\mathbf{I}_t \\ \mathbf{I}_y^T\mathbf{I}_t \end{bmatrix}}$$

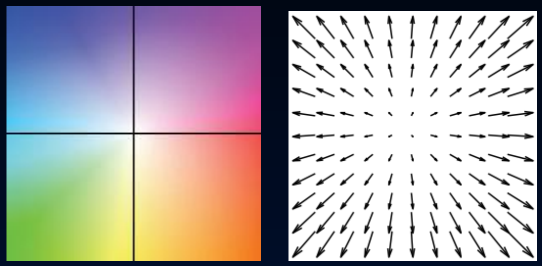
- The flow vector of each pixel is updated independently
- Median filtering can be applied for spatial smoothness

# Example

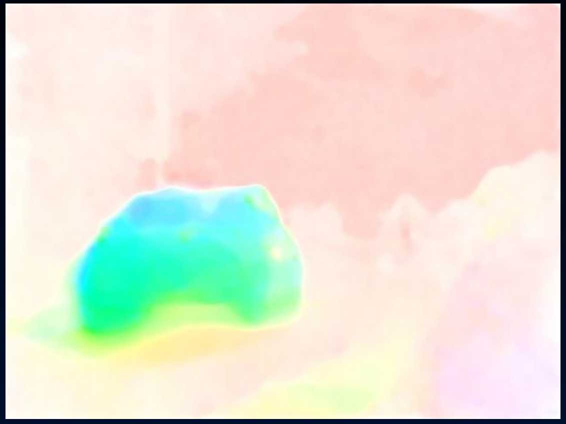
Input two frames



Coarse-to-fine LK



Flow visualization



Coarse-to-fine LK with median filtering

# Contents

- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

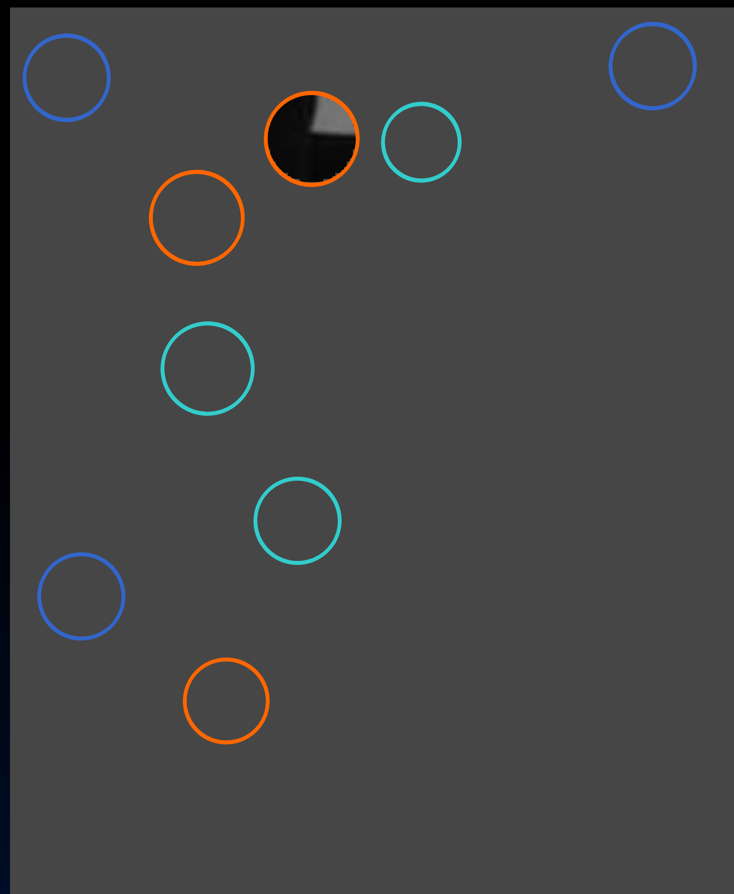
# Motion ambiguities

- When will the Lucas-Kanade algorithm fail?

$$\begin{bmatrix} \mathrm{d}u \\ \mathrm{d}v \end{bmatrix} = -\begin{bmatrix} \mathbf{I}_x^T\mathbf{I}_x & \mathbf{I}_x^T\mathbf{I}_y \\ \mathbf{I}_x^T\mathbf{I}_y & \mathbf{I}_y^T\mathbf{I}_y \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_x^T\mathbf{I}_t \\ \mathbf{I}_y^T\mathbf{I}_t \end{bmatrix}$$

- The inverse may not exist!!!
- How?
  - All the derivatives are zero: *flat regions*
  - X- and y- derivatives are linearly correlated: *lines*

# The aperture problem



Corners

Lines

Flat regions

# Dense optical flow with spatial regularity

- Local motion is inherently ambiguous
  - Corners: definite, no ambiguity
  - Lines: definite along the normal, ambiguous along the tangent
  - Flat regions: totally ambiguous
- Solution: imposing spatial smoothness to the flow field
  - Adjacent pixels should move together as much as possible
  - Horn & Schunck equation

$$\boxed{(u,v) = \arg\min \iint \underbrace{\left(I_x u + I_y v + I_t\right)^2}_{\text{Data term}} + \alpha \underbrace{\left(|\nabla u|^2 + |\nabla v|^2\right)}_{\text{Smoothness term}} dxdy}$$

  - $|\nabla u|^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 = u_x^2 + u_y^2$
  - $\alpha$: smoothness coefficient

# 2D Euler Lagrange

- 2D Euler Lagrange: the functional

$$S = \iint_{\Omega} L(x, y, f, f_x, f_y)\,dxdy$$

is minimized only if $f$ satisfies the partial differential equation (PDE)

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x}\frac{\partial L}{\partial f_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial f_y} = 0$$

- In Horn-Schunck
  - $L(u, v, u_x, u_y, v_x, v_y) = (I_x u + I_y v + I_t)^2 + \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2)$
  - $\frac{\partial L}{\partial u} = 2(I_x u + I_y v + I_t)I_x$
  - $\frac{\partial L}{\partial u_x} = 2\alpha u_x, \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} = 2\alpha u_{xx}, \frac{\partial L}{\partial u_y} = 2\alpha u_y, \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 2\alpha u_{yy}$

# Linear PDE

- The Euler-Lagrange PDE for Horn-Schunck is

$$\begin{cases}(I_x u + I_y v + I_t)I_x - \alpha(u_{xx} + u_{yy}) = 0 \\ (I_x u + I_y v + I_t)I_y - \alpha(v_{xx} + v_{yy}) = 0\end{cases}$$

- $u_{xx} + u_{yy}$ can be obtained by a Laplacian operator:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

- In the end, we solve a large linear system

$$\boxed{\begin{bmatrix} \mathbf{I}_x^2 + \alpha\mathbf{L} & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y^2 + \alpha\mathbf{L} \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = -\begin{bmatrix} \mathbf{I}_x I_t \\ \mathbf{I}_y I_t \end{bmatrix}}$$

# How to solve a large linear system?

$$\begin{bmatrix} \mathbf{I}_x^2 + \alpha\mathbf{L} & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y^2 + \alpha\mathbf{L} \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = -\begin{bmatrix} \mathbf{I}_x I_t \\ \mathbf{I}_y I_t \end{bmatrix}$$

- With $\alpha > 0$, this system is positive definite!
- You can use your favorite solver
  - Gauss-Seidel, successive over-relaxation (SOR)
  - (Pre-conditioned) conjugate gradient
- No need to wait for the solver to converge completely

# Condition for convergence

- In the objective function

$$(u, v) = \arg\min \iint \left(I_x u + I_y v + I_t\right)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)dxdy$$

  The displacement $(u, v)$ has to be small for the Taylor expansion to be valid.

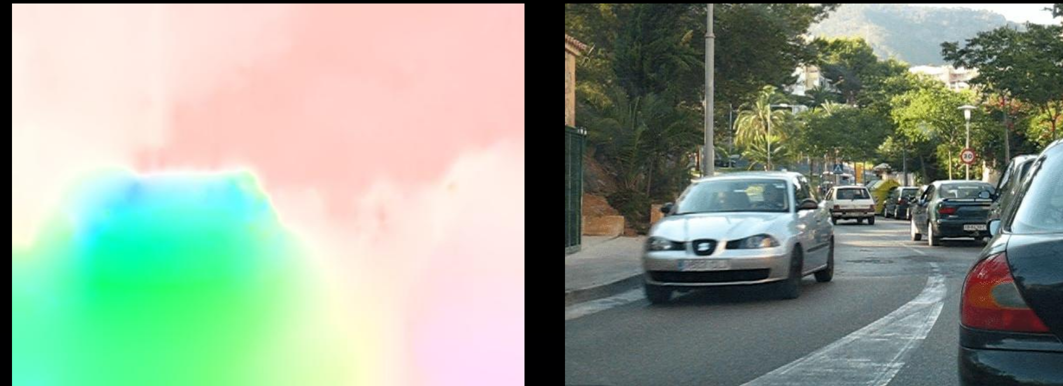
- More practically, we can estimate the optimal incremental change

$$\iint \left(I_x du + I_y dv + I_t\right)^2 + \alpha(|\nabla(u + du)|^2 + |\nabla(v + dv)|^2)dxdy$$

- The solution becomes

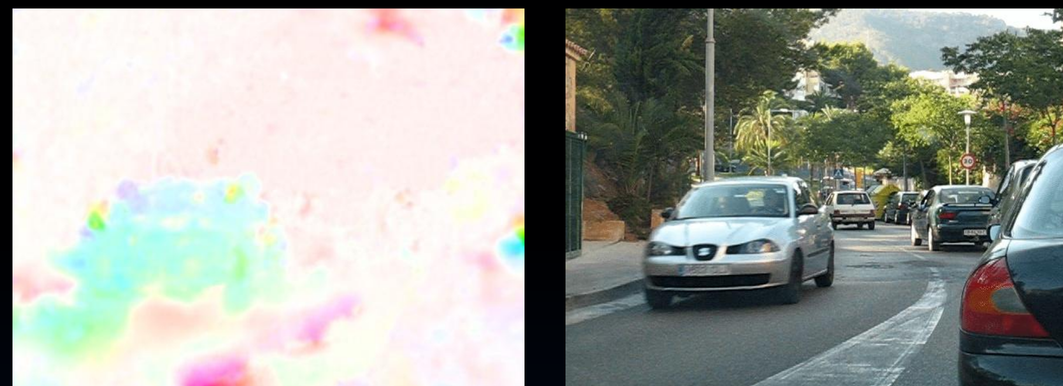
$$\boxed{\begin{bmatrix} \mathbf{I}_x^2 + \alpha\mathbf{L} & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y^2 + \alpha\mathbf{L} \end{bmatrix} \begin{bmatrix} dU \\ dV \end{bmatrix} = -\begin{bmatrix} \mathbf{I}_x I_t + \alpha\mathbf{L}U \\ \mathbf{I}_y I_t + \alpha\mathbf{L}V \end{bmatrix}}$$

# Examples

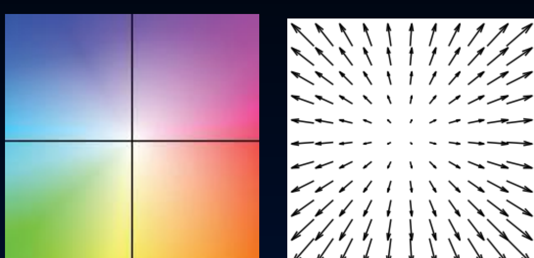


Horn-Schunck

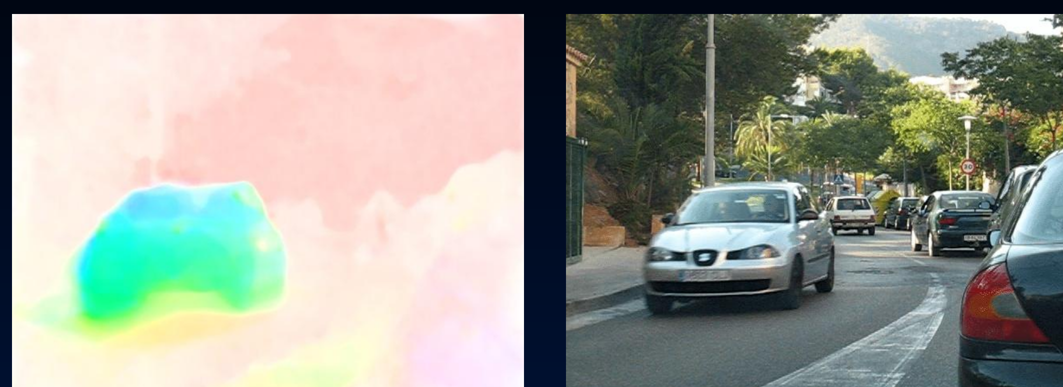
Input two frames



Coarse-to-fine LK



Flow visualization



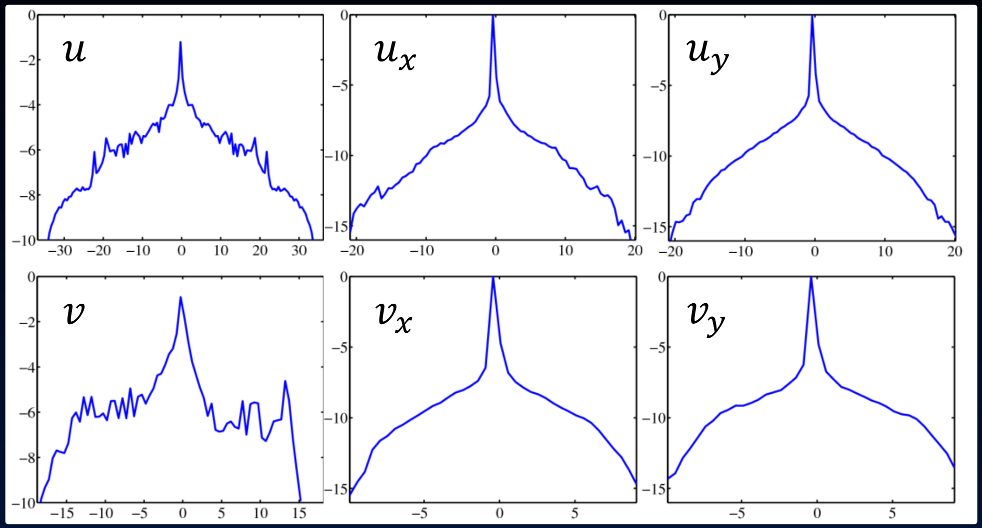
Coarse-to-fine LK with median filtering

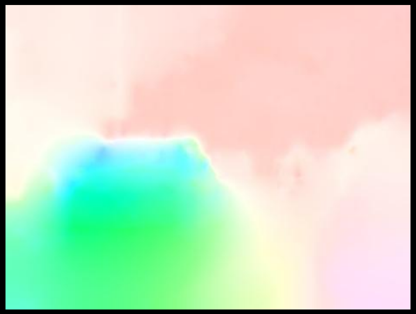
# The source of over-smoothness

- Horn-Schunck is a Gaussian Markov random field (GMRF)

$$\iint \left(I_x u + I_y v + I_t\right)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)dxdy$$

- Spatial over-smoothness is caused by quadratic smoothness term
- Nevertheless, optical flow fields are sparse!



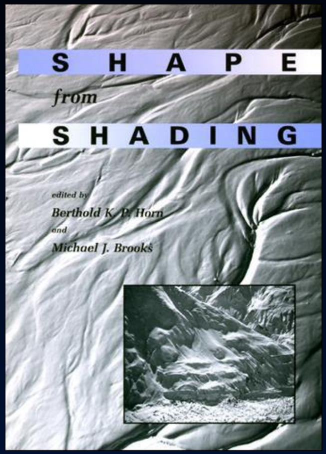



Horn-Schunck

Ground truth

# Continuous Markov Random Fields

- Horn-Schunck started 30 years of research on continuous Markov random fields
  - Optical flow estimation
  - Image reconstruction, e.g. denoising, super resolution
  - Shape from shading, inverse rendering problems
  - Natural image priors
- Why continuous?
  - Many signals are differentiable
  - More complicated spatial relationships
- Fast solvers
  - Multi-grid
  - Preconditioned conjugate gradient
  - FFT + annealing



# Contents

- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

# Modification to Horn-Schunck

- Let $\mathrm{x} = (x, y, t)$, and $\mathrm{w}(\mathrm{x}) = (u(\mathrm{x}), v(\mathrm{x}), 1)$ be the flow vector
- Horn-Schunck (recall)

$$\iint \left(I_x u + I_y v + I_t\right)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) dxdy$$

- Robust estimation

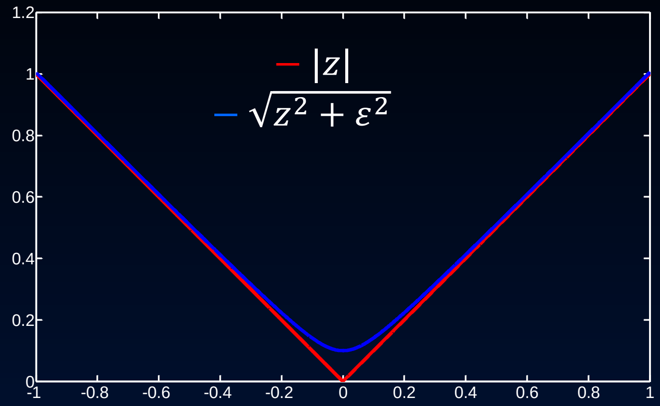
$$\iint \psi(|I(\mathrm{x} + \mathrm{w}) - I(\mathrm{x})|^2) + \alpha\phi(|\nabla u|^2 + |\nabla v|^2) dxdy$$

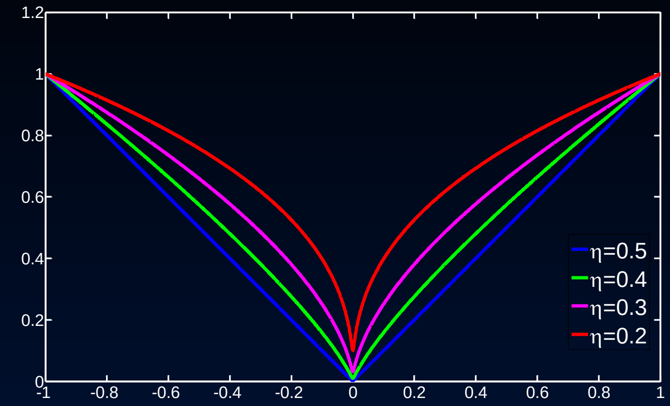
- Robust estimation with Lucas-Kanade

$$\iint g * \psi(|I(\mathrm{x} + \mathrm{w}) - I(\mathrm{x})|^2) + \alpha\phi(|\nabla u|^2 + |\nabla v|^2) dxdy$$

# Robust functions

- Various forms of robust functions
  - L1 norm: $\psi(z^2) = \sqrt{z^2 + \varepsilon^2}, \phi(z^2) = \sqrt{z^2 + \varepsilon^2}$
  - Sub L1: $\psi(z^2; \eta) = (z^2 + \varepsilon^2)^\eta, \eta < 0.5$
  - Lorentzian: $\psi(z^2) = \log(1 + z^2)$



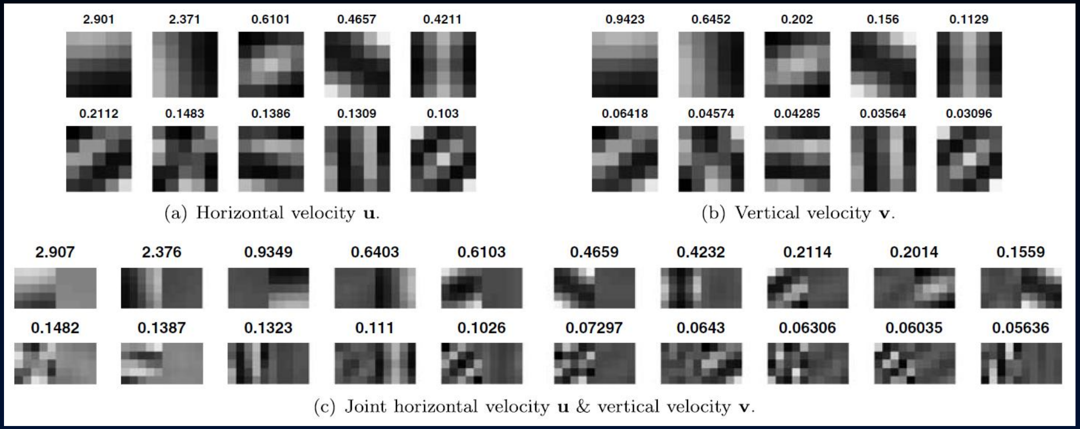
# Special cases

- The robust objective function

$$\iint g * \psi(|I(\mathrm{x}+\mathrm{w}) - I(\mathrm{x})|^2) + \alpha\phi(|\nabla u|^2 + |\nabla v|^2)dxdy$$

  - Lucas-Kanade: $\alpha = 0, \psi(z^2) = z^2$
  - Robust Lucas-Kanade: $\alpha = 0, \psi(z^2) = \sqrt{z^2 + \varepsilon^2}$
  - Horn-Schunck: $g = 1, \psi(z^2) = z^2, \phi(z^2) = z^2$

- One can also learn the filters (other than gradients), and robust function $\psi(\cdot), \phi(\cdot)$ [Roth & Black 2005]



# Derivation strategies

- Euler-Lagrange
  - Derive in continuous domain, discretize in the end
  - Nonlinear PDE’s
  - Outer and inner fixed point iterations
  - Cannot generalize to general filters
- Variational optimization
- Iterative reweighted least square (IRLS)
  - Discretize first and derive in matrix form
  - Easy to understand and derive
- These three approaches are equivalent!

# Iterative reweighted least square (IRLS)

- Let $\phi(z) = (z^2 + \varepsilon^2)^\eta$ be a robust function
- We want to minimize the objective function

$$\Phi(\mathbf{A}x + b) = \sum_{i=1}^{n} \phi\left(\left(a_i^T x + b_i\right)^2\right)$$

where $x \in \mathbb{R}^d$, $\mathbf{A} = [a_1\ a_2 \cdots a_n]^T \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$

- By setting $\frac{\partial \Phi}{\partial x} = 0$, we can derive

$$\frac{\partial \Phi}{\partial x} = \sum_{i=1}^{n} \phi'\left(\left(a_i^T x + b_i\right)^2\right)\left(a_i^T x + b_i\right) a_i$$

$$= \sum_{i=1}^{n} \mathrm{w}_{ii} a_i^T x a_i + \mathrm{w}_{ii} b_i a_i \qquad \mathrm{w}_{ii} = \phi'\left(\left(a_i^T x + b_i\right)^2\right)$$

$$= \sum_{i=1}^{n} a_i^T \mathrm{w}_{ii} x a_i + b_i \mathrm{w}_{ii} a_i$$

$$= \mathbf{A}^T \mathbf{W} \mathbf{A} x + \mathbf{A}^T \mathbf{W} b \qquad \mathbf{W} = \mathrm{diag}(\Phi'(\mathbf{A}x + b))$$

# Iterative reweighted least square (IRLS)

- Derivative: $\frac{\partial \Phi}{\partial x} = \mathbf{A}^T\mathbf{W}\mathbf{A}x + \mathbf{A}^T\mathbf{W}b$
- Iterate between *reweighting* and *least square*

1. Initialize $x = x_0$
2. Compute weight matrix $\mathbf{W} = \mathrm{diag}(\Phi'(\mathbf{A}x + b))$
3. Solve the linear system $\mathbf{A}^T\mathbf{W}\mathbf{A}x = -\mathbf{A}^T\mathbf{W}b$
4. If $x$ converges, return; otherwise, go to 2

- Convergence is guaranteed (local minima)

# IRLS for robust optical flow

- Objective function

$$\iint g * \psi(|I(\mathrm{x}+\mathrm{w}) - I(\mathrm{x})|^2) + \alpha\phi(|\nabla u|^2 + |\nabla v|^2)dxdy$$

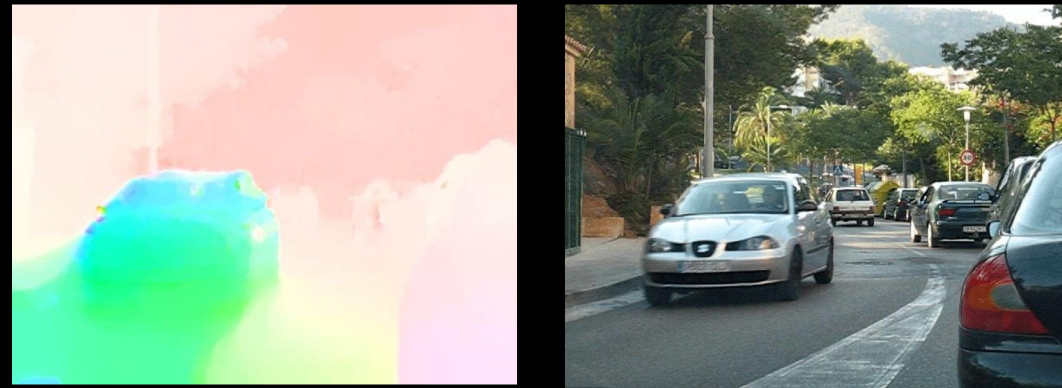
- Discretize, linearize and increment

$$\sum_{x,y} g * \psi(|I_t + I_x du + I_Y dv|^2) + \alpha\phi(|\nabla(u+du)|^2 + |\nabla(v+dv)|^2)$$

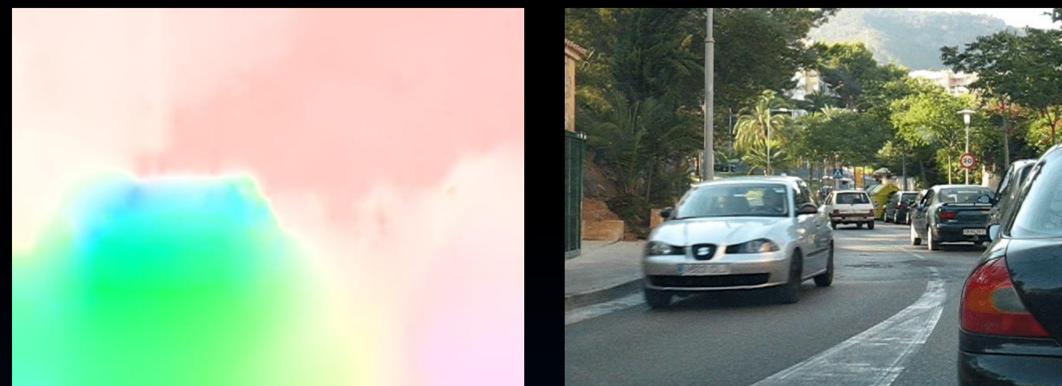
- IRLS (initialize $du = dv = 0$)
  - Weight: $\boldsymbol{\Psi}'_{xx} = \mathrm{diag}(g * \psi' \mathbf{I}_x \mathbf{I}_x), \boldsymbol{\Psi}'_{xy} = \mathrm{diag}\left(g * \psi' \mathbf{I}_x \mathbf{I}_y\right),$
    $\boldsymbol{\Psi}'_{yy} = \mathrm{diag}\left(g * \psi' \mathbf{I}_y \mathbf{I}_y\right), \boldsymbol{\Psi}'_{xt} = \mathrm{diag}(g * \psi' \mathbf{I}_x \mathbf{I}_t),$
    $\boldsymbol{\Psi}'_{yt} = \mathrm{diag}(g * \psi' \mathbf{I}_y \mathbf{I}_t),\quad \mathbf{L} = \mathbf{D}_x^T \boldsymbol{\Phi}' \mathbf{D}_x + \mathbf{D}_y^T \boldsymbol{\Phi}' \mathbf{D}_y$
  - Least square:

$$\boxed{\begin{bmatrix} \boldsymbol{\Psi}'_{xx} + \alpha\mathbf{L} & \boldsymbol{\Psi}'_{xy} \\ \boldsymbol{\Psi}'_{xy} & \boldsymbol{\Psi}'_{yy} + \alpha\mathbf{L} \end{bmatrix} \begin{bmatrix} dU \\ dV \end{bmatrix} = -\begin{bmatrix} \boldsymbol{\Psi}'_{xt} + \alpha\mathbf{L}U \\ \mathbf{I}_y I_t + \alpha\mathbf{L}V \end{bmatrix}}$$

# Examples

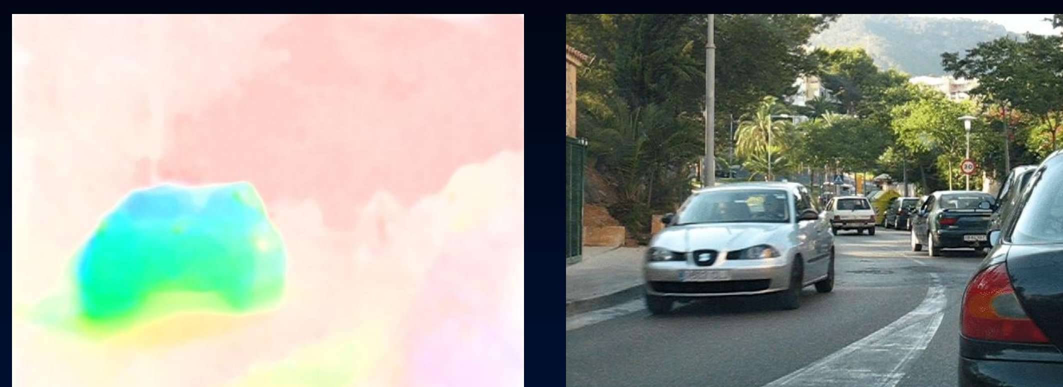


Robust optical flow

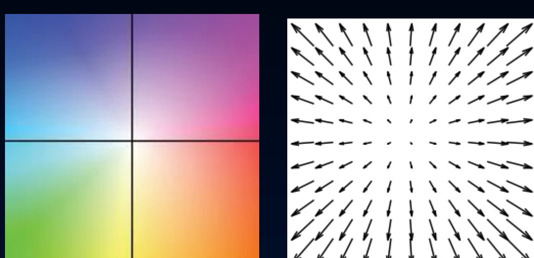


Horn-Schunck

Input two frames



Coarse-to-fine LK with median filtering



Flow visualization

# Contents

- Motion perception
- Motion representation
- Parametric motion: Lucas-Kanade
- Dense optical flow: Horn-Schunck
- Robust estimation
- Applications (1)

# Video stabilization

# Video denoising

- Use multiple frames for temporal coherence
- Non-local mean

# Video denoising

# Video super resolution

- Merge information from adjacent frames
- Reconstruction depends on flow accuracy

# Summary

- Lucas-Kanade
  - Parametric motion
  - Dense flow field (with median filtering)
- Horn-Schunck
  - Gaussian Markov random field
  - Euler-Lagrange
- Robust flow estimation
  - Robust function
    - Account for outliers in data term
    - Encourage piecewise smoothness
  - IRLS (= nonlinear PDE = variational optimization)

# Next time

- Discrete optical flow
- Layer motion analysis
- Contour motion analysis
- Obtaining motion ground truth
- SIFT flow: generalized optical flow
- Applications (2)