Microsoft Research

Massachusetts Institute of Technology

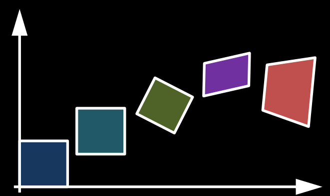
CSAIL

Motion Estimation (II)

Ce Liu
celiu@microsoft.com
Microsoft Research New England

Last time

- Motion perception
- Motion representation
- Parametric motion: *Lucas-Kanade*
- Dense optical flow: *Horn-Schunck*



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

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

Who are they?

Berthold K. P. Horn

Takeo Kanade

Content

- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Other representations

Content

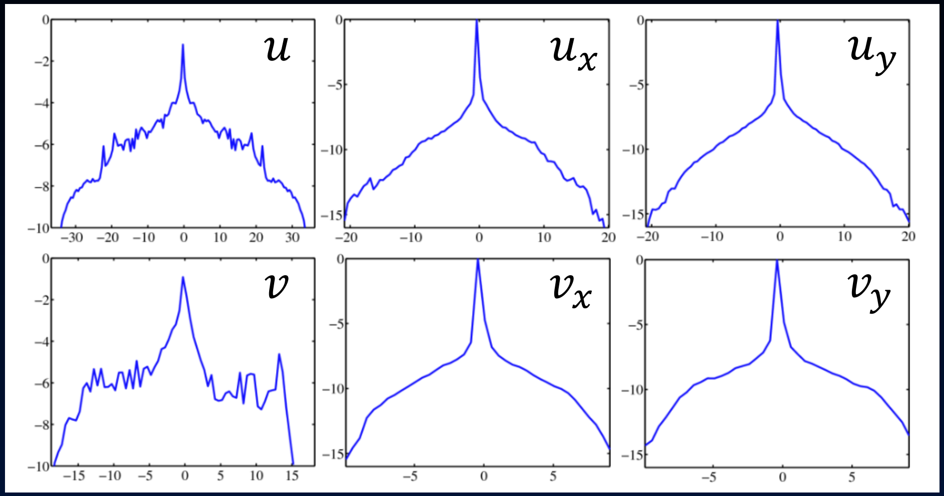
- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Other representations

Spatial regularity

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

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

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



Data term

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

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

- Quadratic data term implies Gaussian white noise
- Nevertheless, the difference between two corresponded pixels is caused by
  - Noise (majority)
  - Occlusion
  - Compression error
  - Lighting change
  - …

- The error function needs to account for these factors

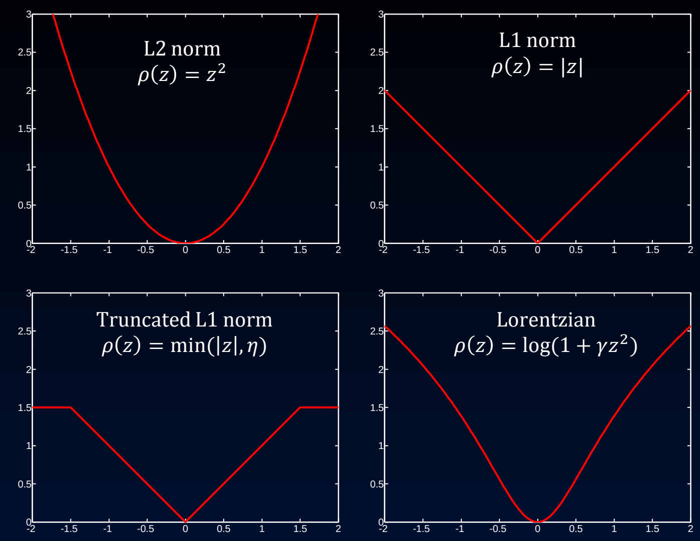
Robust statistics

- 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

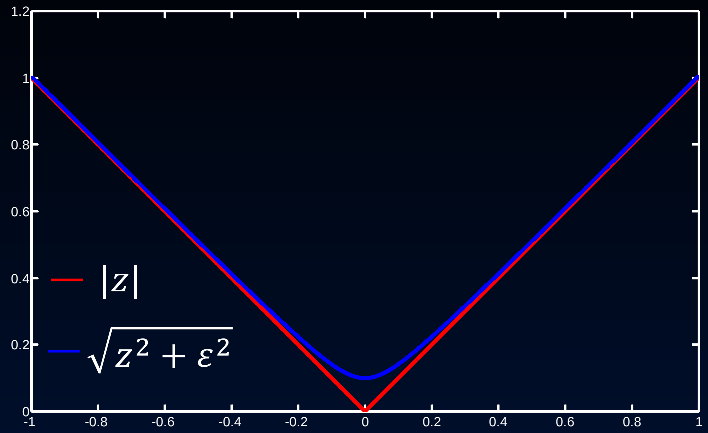
$$z^* = \arg\min_z \sum_i \rho(z - z_i)$$

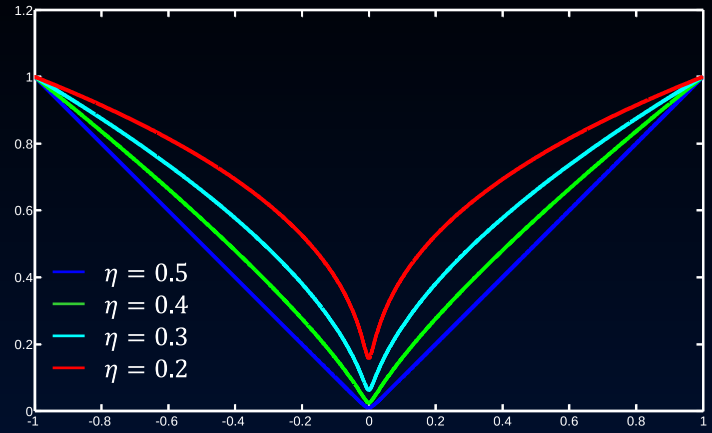
  - L2 norm: $z^* = 4.172$
  - L1 norm: $z^* = 1.038$
  - Truncated L1: $z^* = 1.0296$
  - Lorentzian: $z^* = 1.0147$



The family of robust power functions

- Can we directly use L1 norm $\psi(z) = |z|$?
  - Derivative is not continuous
- Alternative forms
  - L1 norm: $\psi(z^2) = \sqrt{z^2 + \varepsilon^2}$
  - Sub L1: $\psi(z^2; \eta) = (z^2 + \varepsilon^2)^\eta, \eta < 0.5$





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

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

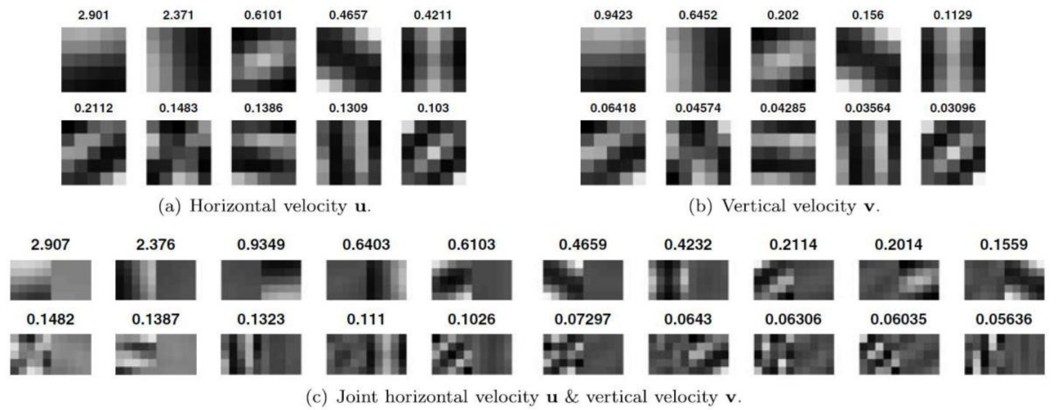
A unifying framework

- The robust object 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]



(a) Horizontal velocity **u**. (b) Vertical velocity **v**.

(c) Joint horizontal velocity **u** & vertical velocity **v**.

Derivation strategies

- Euler-Lagrange
  - Derive in continuous domain, discretize in the end
  - Nonlinear PDE's
  - Outer and inner fixed point iterations
  - Limited to derivative filters; cannot generalize to arbitrary filters
- Energy minimization
  - Discretize first and derive in matrix form
  - Easy to understand and derive
- Variational optimization
- Iteratively reweighted least square (IRLS)
- Euler-Lagrange = Variational optimization = IRLS

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

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

$$\begin{aligned}
\frac{\partial \Phi}{\partial x} &= \sum\nolimits_{i=1}^{n} \phi'\left(\left(a_i^T x + b_i\right)^2\right)\left(a_i^T x + b_i\right)^2 a_i \\
&= \sum\nolimits_{i=1}^{n} w_{ii} a_i^T x a_i + w_{ii} b_i a_i \\
&= \sum\nolimits_{i=1}^{n} a_i^T w_{ii} x a_i + b_i w_{ii} a_i \\
&= \mathbf{A}^T \mathbf{W} \mathbf{A} x + \mathbf{A}^T \mathbf{W} b
\end{aligned}$$

$$w_{ii} = \phi'\left(\left(a_i^T x + b_i\right)^2\right)$$

$$\mathbf{W} = \operatorname{diag}(\Phi'(\mathbf{A}x + b))$$

Iteratively reweighted least square (IRLS)

- Derivative: $\frac{\partial \Phi}{\partial x} = \mathbf{A}^T \mathbf{W} \mathbf{A} x + \mathbf{A}^T \mathbf{W} b = 0$
- Iterate between *reweighting* and *least square*

1. Initialize $x = x_0$
2. Compute weight matrix $\mathbf{W} = \text{diag}(\Phi'(\mathbf{A}x + b))$
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

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

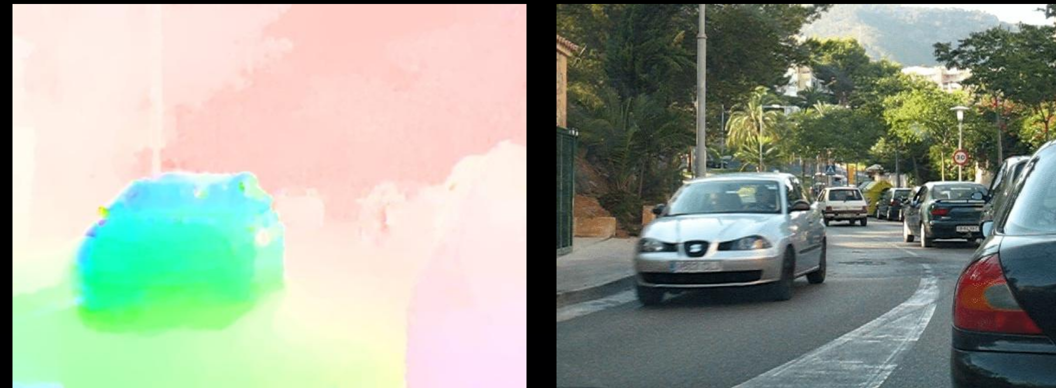
- Discretize, linearize and increment

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

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

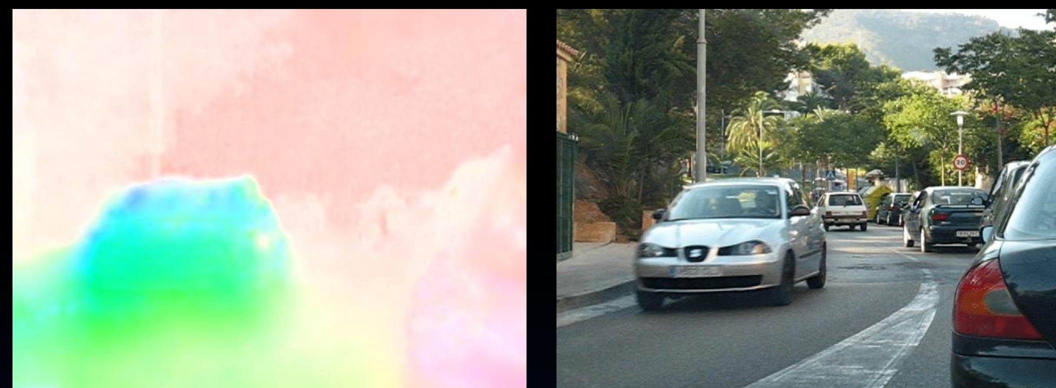
$$\begin{bmatrix} \mathbf{\Psi}'_{xx} + \alpha\mathbf{L} & \mathbf{\Psi}'_{xy} \\ \mathbf{\Psi}'_{xy} & \mathbf{\Psi}'_{yy} + \alpha\mathbf{L} \end{bmatrix} \begin{bmatrix} dU \\ dV \end{bmatrix} = -\begin{bmatrix} \mathbf{\Psi}'_{xt} + \alpha\mathbf{L}U \\ \mathbf{\Psi}'_{yt} + \alpha\mathbf{L}V \end{bmatrix}$$

Example

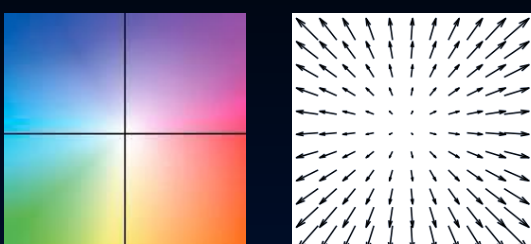


Robust optical flow

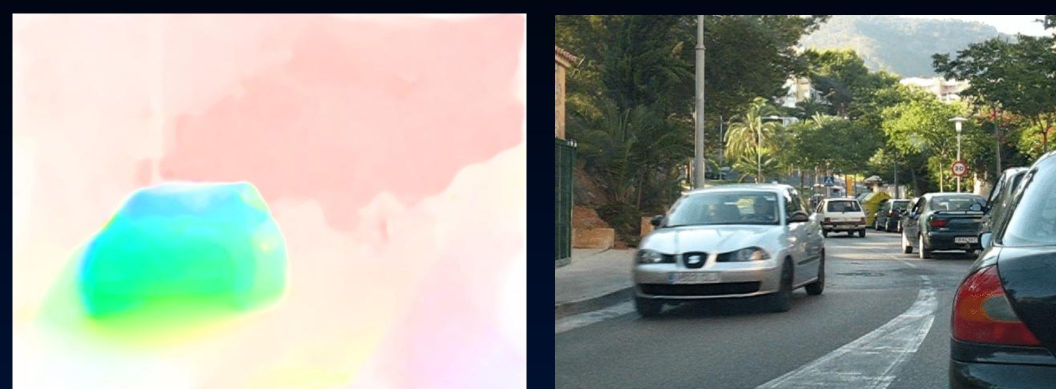
Input two frames



Horn-Schunck



Flow visualization



Coarse-to-fine LK with median filtering

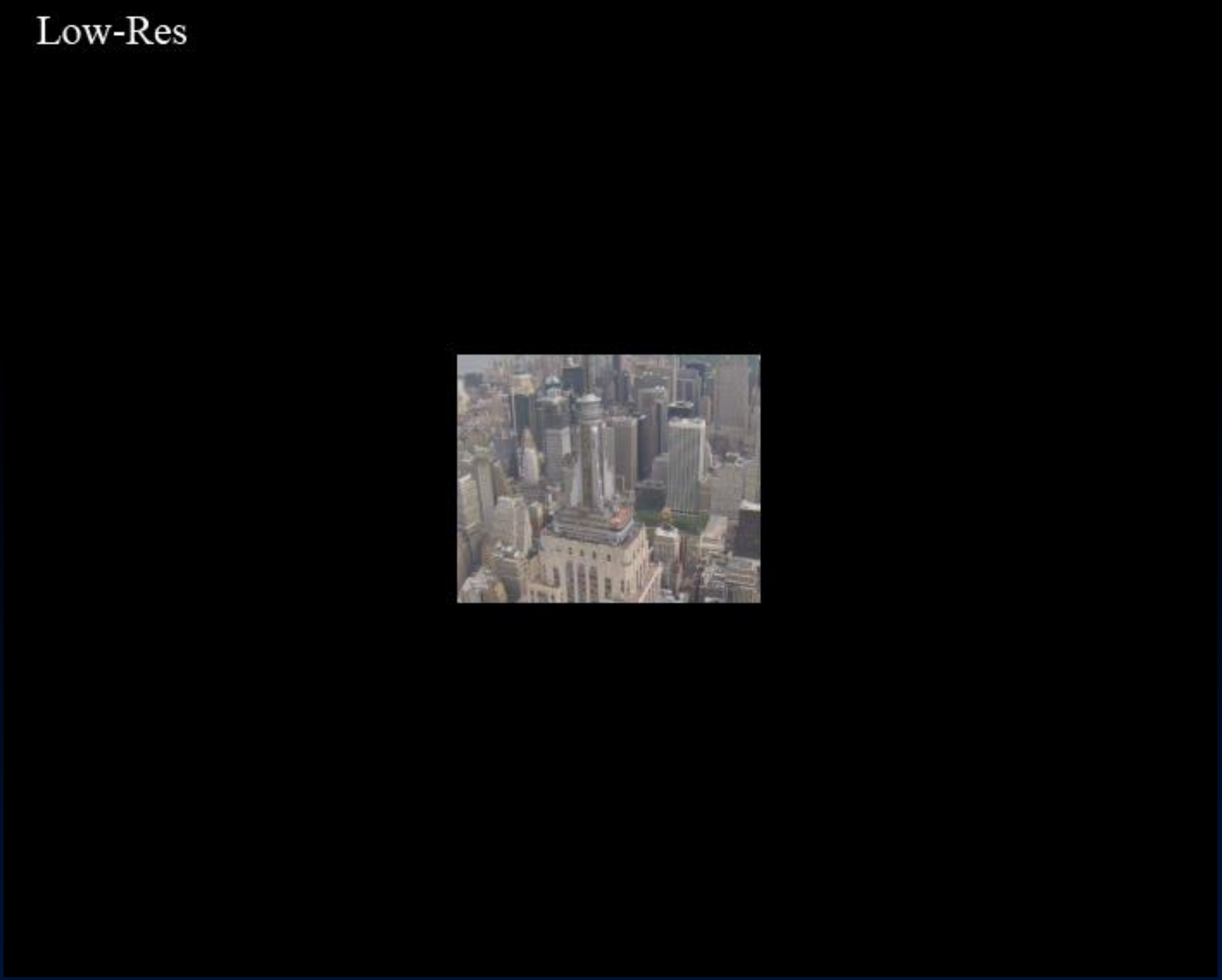
Content

- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Other representations

Video stabilization

Video denoising

Video super resolution



Content

- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Contour motion analysis
- Obtaining motion ground truth

Block matching

- Both Horn-Schunck and Lucas-Kanade are sub-pixel accuracy algorithms
- But in practice we may not need sub-pixel accuracy
- MPEG: 16×16 block matching using MMSE
- H264: variable block size and quarter-pixel precision

Tracking reliable features

- Idea: no need to work on ambiguous region pixels (flat regions & line structures)
- Instead, we can track features and then propagate the tracking to ambiguous pixels
- Good features to track [Shi & Tomashi 94]

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

- Block matching + Lucas-Kanade refinement

Feature detection & tracking

From sparse to dense

- Interpolation: given values $\{d_i\}$ at $\{(x_i, y_i)\}$, reconstruct a smooth plane $f(x, y)$
- Membrane model (first order smoothness)

$$\iint \sum_i \left(w_i(f(x_i, y_i) - d_i)^2 + \alpha\left(f_x^2 + f_y^2\right)\right) dxdy$$

- Thin plate model (second order smoothness)

$$\iint \sum_i \left(w_i(f(x_i, y_i) - d_i)^2 + \alpha\left(f_{xx}^2 + f_{xy}^2 + f_{yy}^2\right)\right) dxdy$$

Membrane vs. thin plate

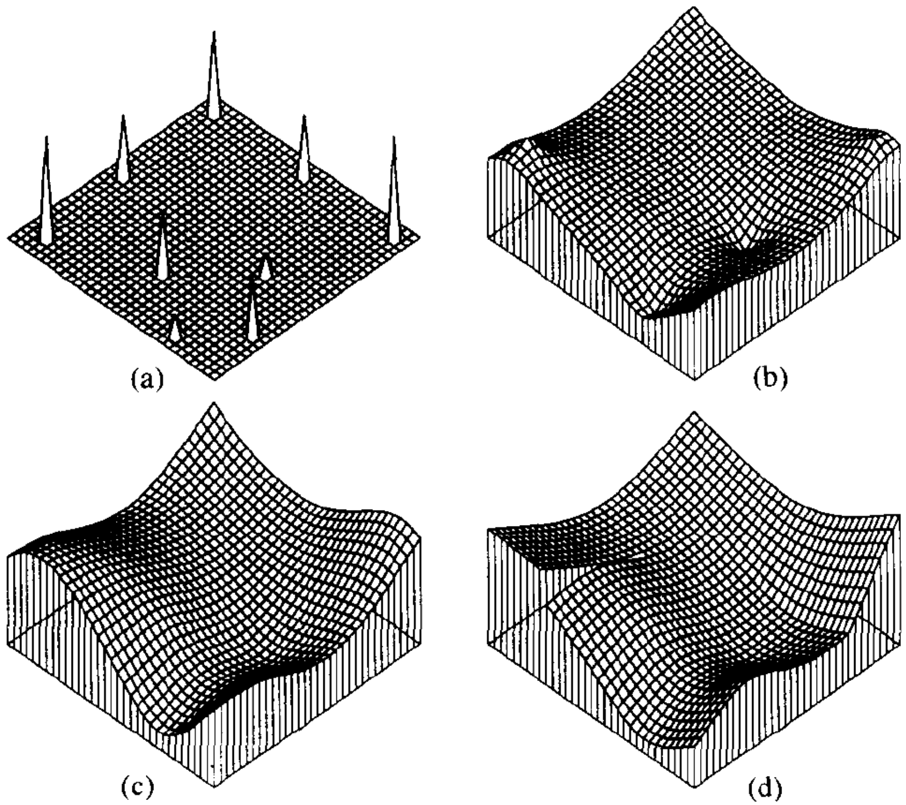
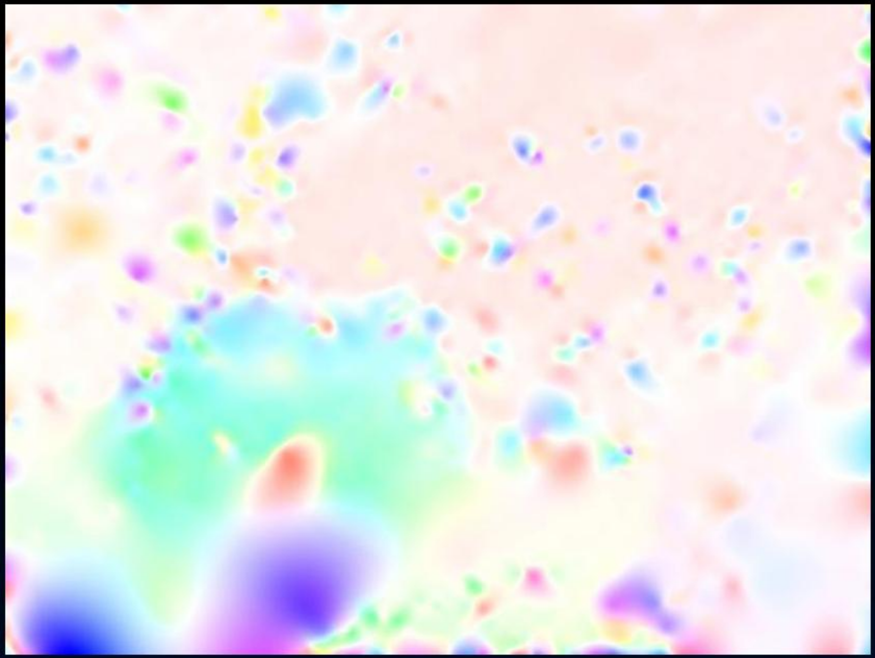


Fig. 1. Sample data points and interpolated solutions: (a) sample data points, (b) membrane interpolant, (c) thin plate interpolant, (d) controlled continuity spline (thin plate with discontinuities and creases).

Dense flow field from sparse tracking



Pros and Cons of Feature Matching

- Pros
  - Efficient (a few feature points vs. all pixels)
  - Reliable (with advanced feature descriptors)
- Cons
  - Independent tracking (tracking can be unreliable)
  - Not all information is used (may not capture weak features)
- How to improve
  - Track every pixel with uncertainty
  - Integrate spatial regularity (neighboring pixels go together)

Content

- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Other representations

Discrete optical flow

- The objective function is similar to that of continuous flow
- $\mathrm{x} = (x, y)$ is pixel coordinate, $\mathrm{w} = (u, v)$ is flow vector

$$E(\mathrm{w}) = \sum_{\mathrm{x}} \min\left(\left|I_1(\mathrm{x}) - I_2\left(\mathrm{x} + \mathrm{w}(\mathrm{x})\right)\right|, t\right) +$$

$$\sum_{\mathrm{x}} \eta(|u(\mathrm{x})| + |v(\mathrm{x})|) +$$

$$\sum_{(\mathrm{x}_1, \mathrm{x}_2) \in \varepsilon} \min(\alpha|u(\mathrm{x}_1) - u(\mathrm{x}_2)|, d) + \min(\alpha|v(\mathrm{x}_1) - v(\mathrm{x}_2)|, d)$$

Data term

Small displacement

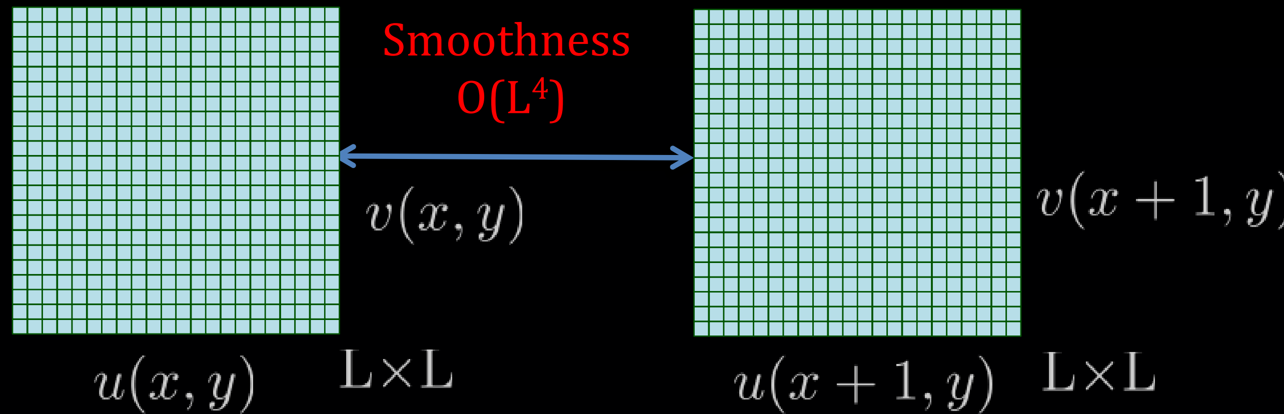
Spatial regularity

- Truncated L1 norms:
  - Account for outliers in the data term
  - Encourage piecewise smoothness in the smoothness term

Decoupled smoothness

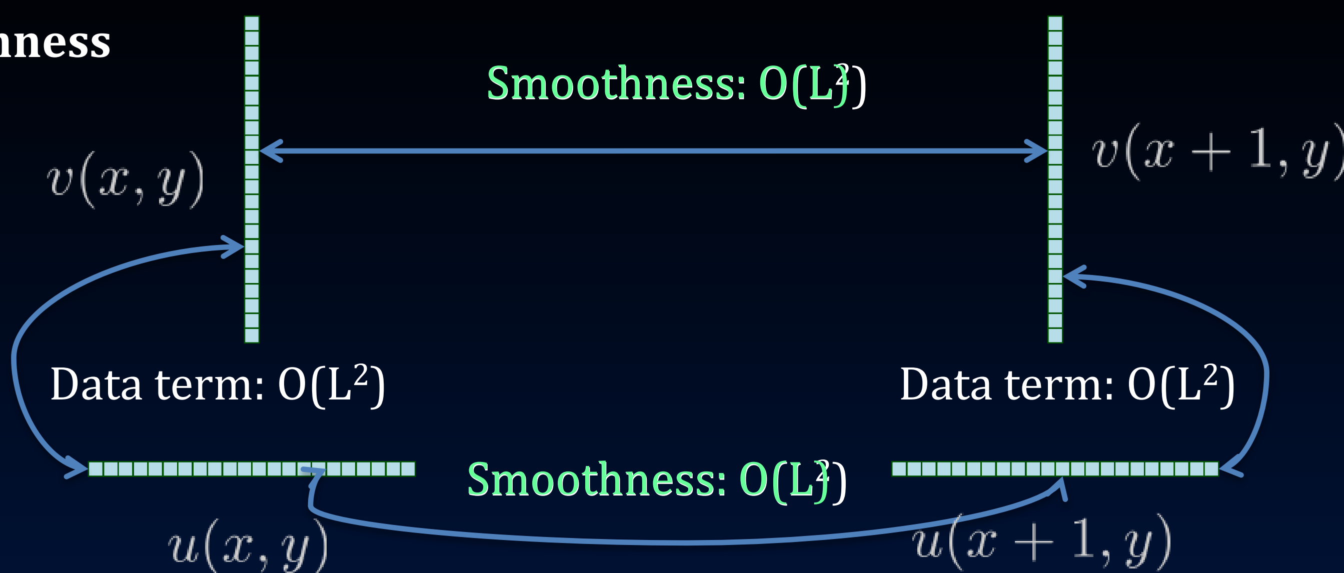
Coupled smoothness

$$\sqrt{u_x^2 + v_x^2}$$



Decoupled smoothness

$$|u_x| + |v_x|$$

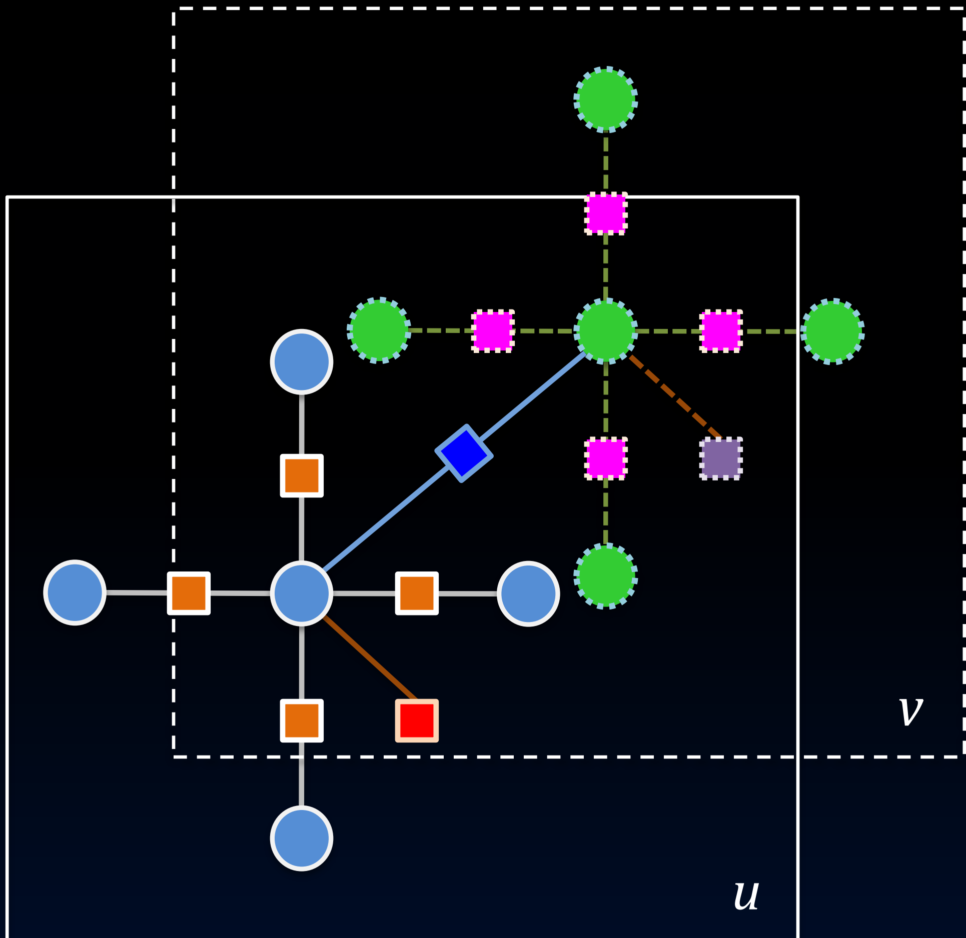


Combinatorial optimization on graph

$$E(\mathrm{w}) = \sum_{\mathrm{x}} \min\left(\left|I_1(\mathrm{x}) - I_2\big(\mathrm{x} + \mathrm{w}(\mathrm{x})\big)\right|, t\right) + \sum_{\mathrm{x}} \eta(|u(\mathrm{x})| + |v(\mathrm{x})|) + \sum_{(\mathrm{x}_1,\mathrm{x}_2)\in\varepsilon} \min(\alpha|u(\mathrm{x}_1) - u(\mathrm{x}_2)|, d) + \min(\alpha|v(\mathrm{x}_1) - v(\mathrm{x}_2)|, d)$$

- Optimization strategies
  - Belief propagation
  - Graph cuts
  - MCMC (simulated annealing)

Dual-layer belief propagation



Horizontal flow u

Vertical flow v

$\mathrm{w} = (u, v)$

Data term

$$\min(|I_1(\mathrm{x}) - I_2(\mathrm{x} + \mathrm{w})|, t)$$

Smoothness term on u

$$\min(\alpha|u(\mathrm{x}_1) - u(\mathrm{x}_2)|, d)$$

Smoothness term on v

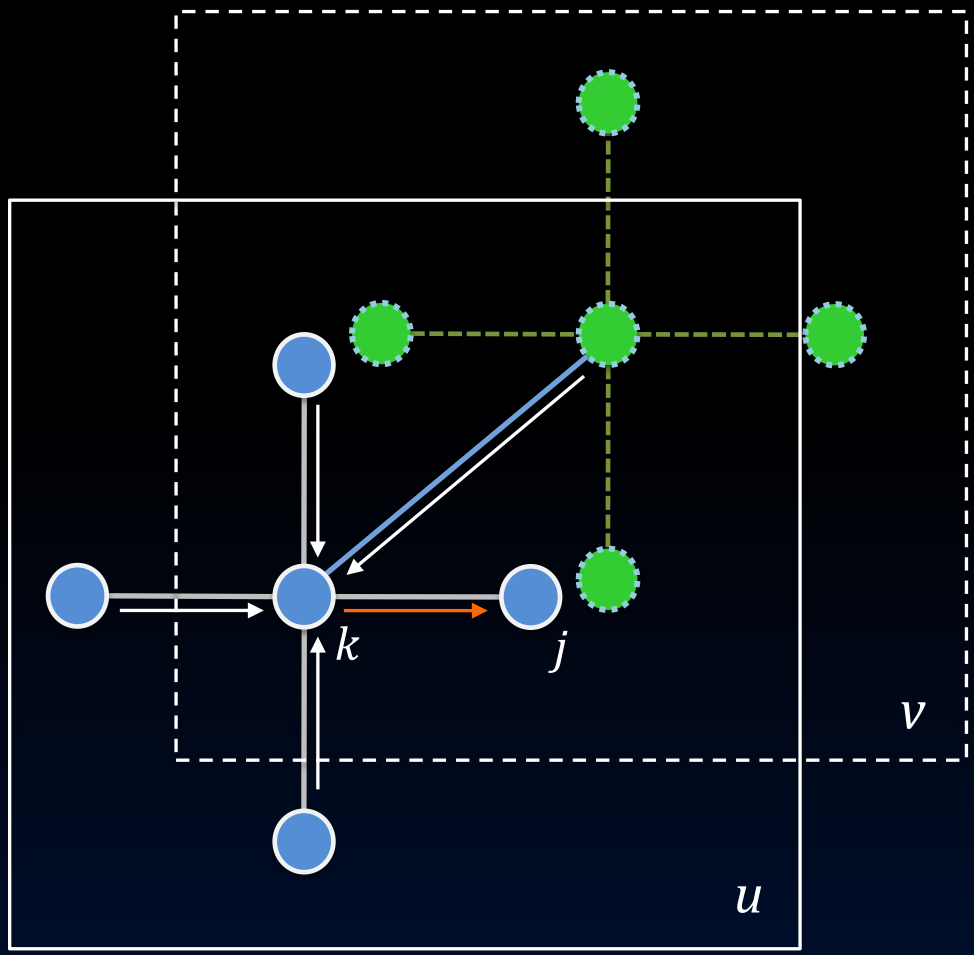
$$\min(\alpha|v(\mathrm{x}_1) - v(\mathrm{x}_2)|, d)$$

Regularization term on u $\eta|u(\mathrm{x})|$

Regularization term on v $\eta|v(\mathrm{x})|$

[Shekhovtsov et al. CVPR 07]

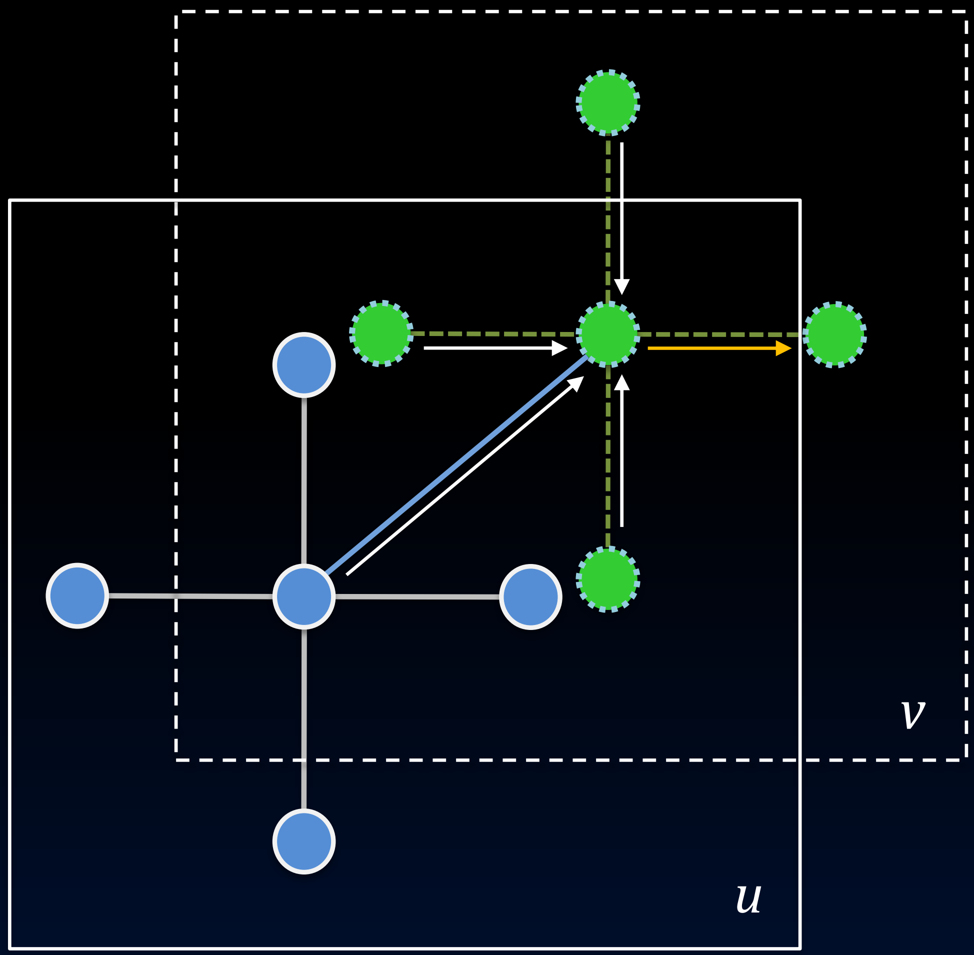
Dual-layer belief propagation



Message M_j^k: given all the information at node k, predict the distribution at node j

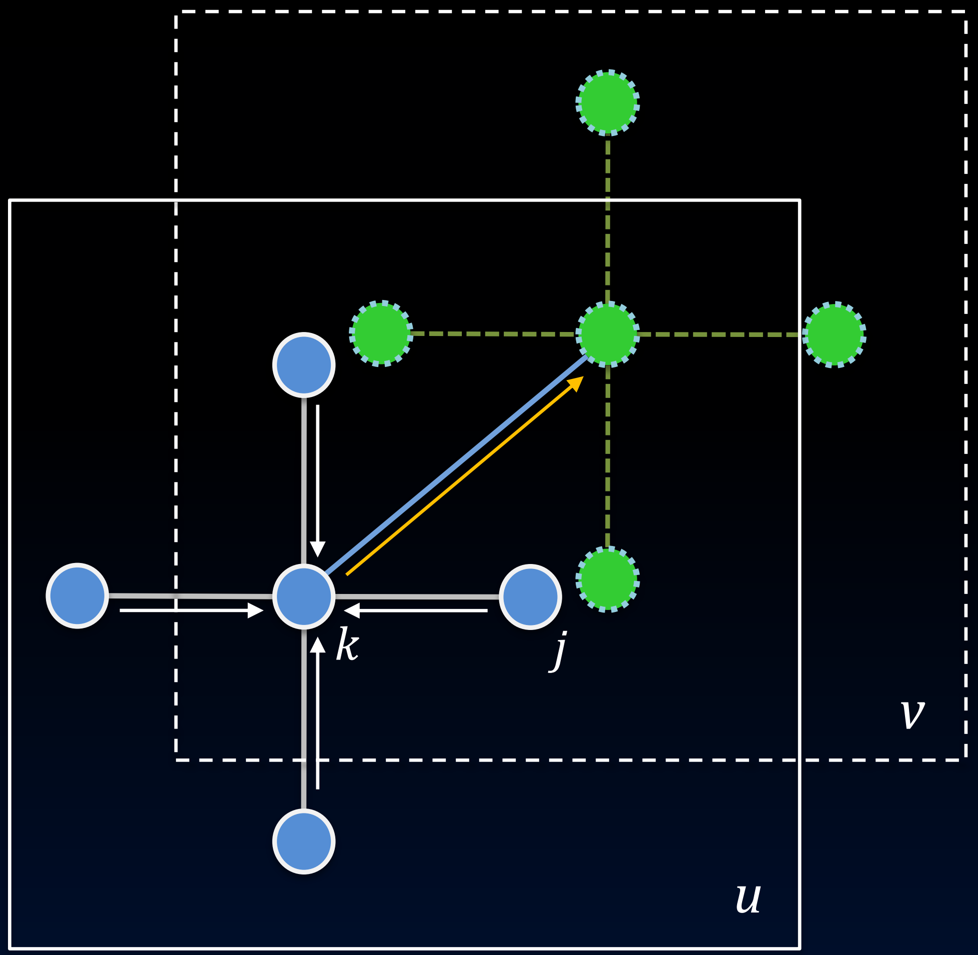
Update within u plane

Dual-layer belief propagation



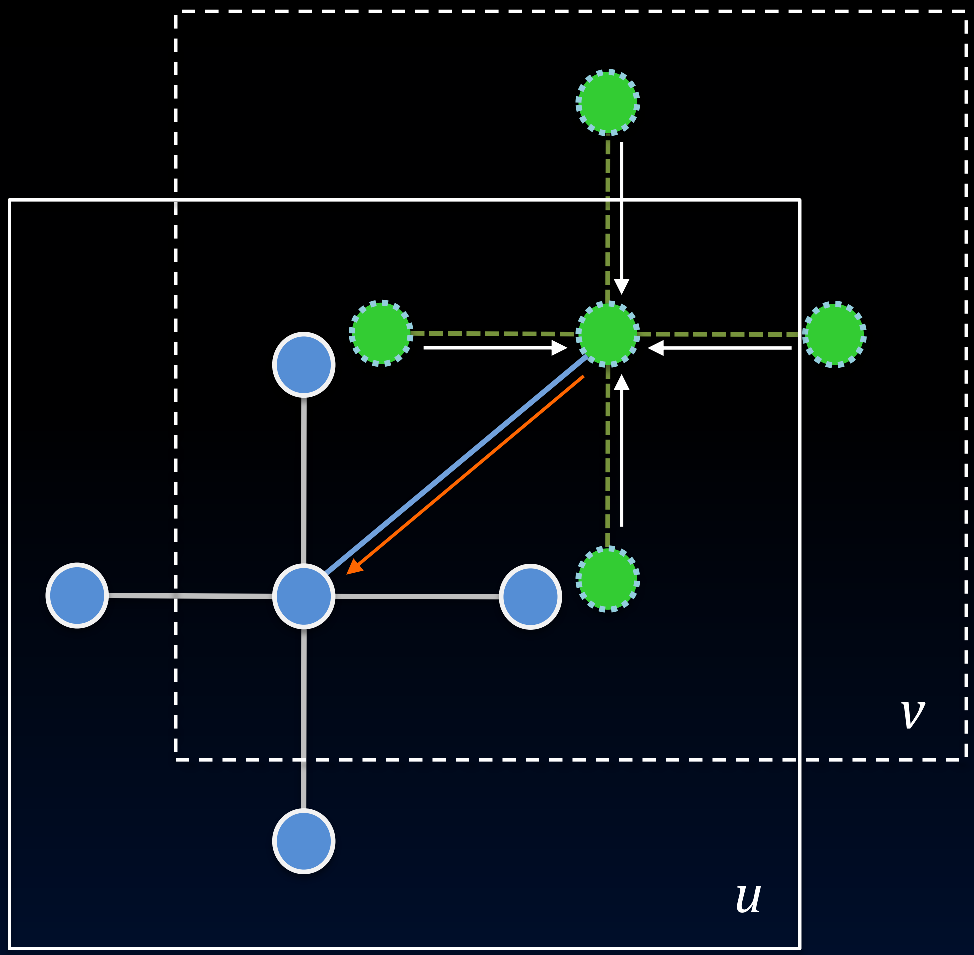
Update within v plane

Dual-layer belief propagation



Update from u plane to v plane

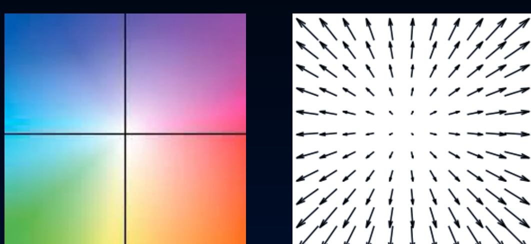
Dual-layer belief propagation



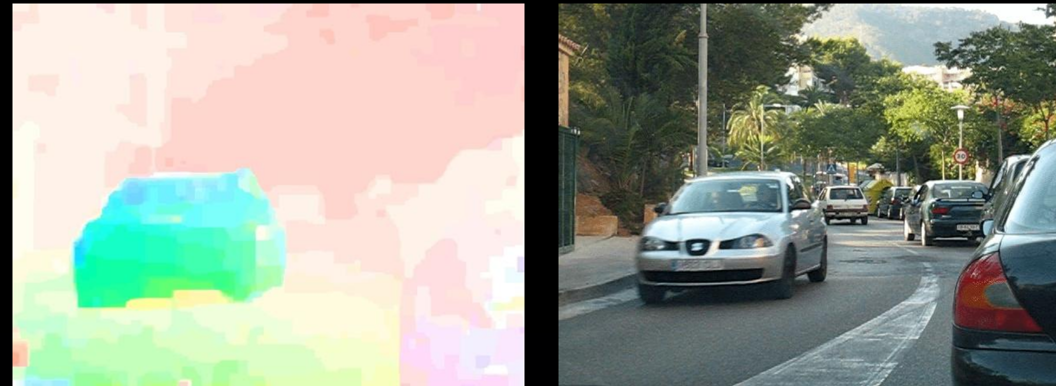
Update from v plane to u plane

Example

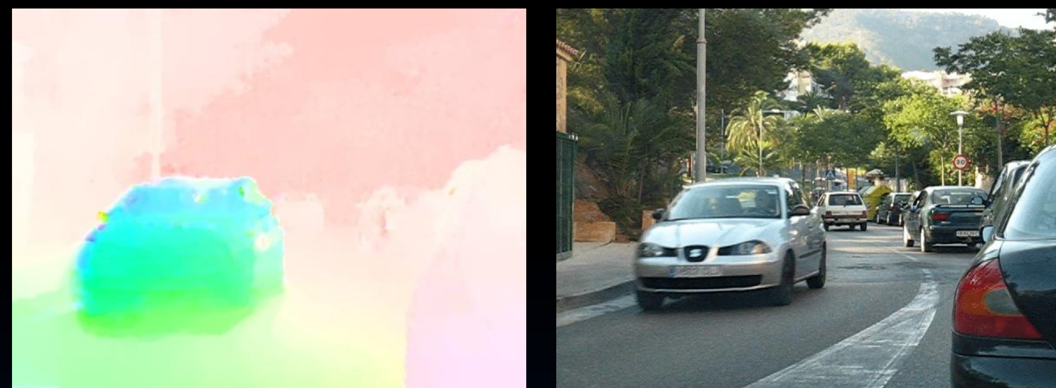
Input two frames



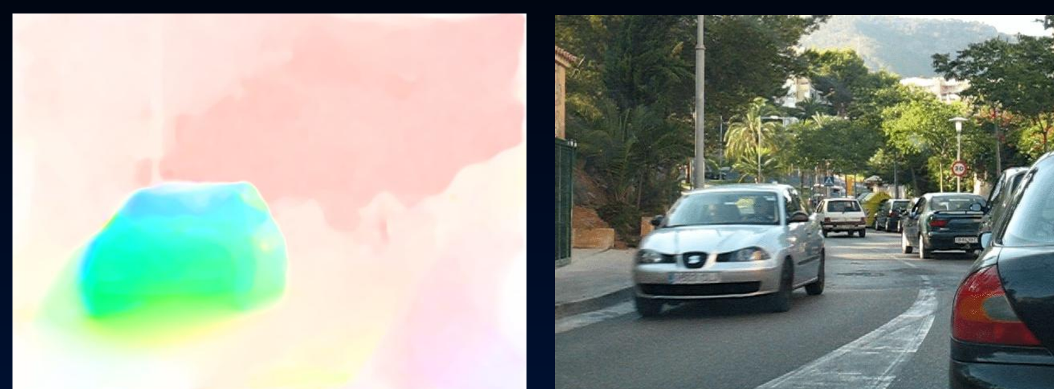
Flow visualization



Discrete optical flow



Robust optical flow



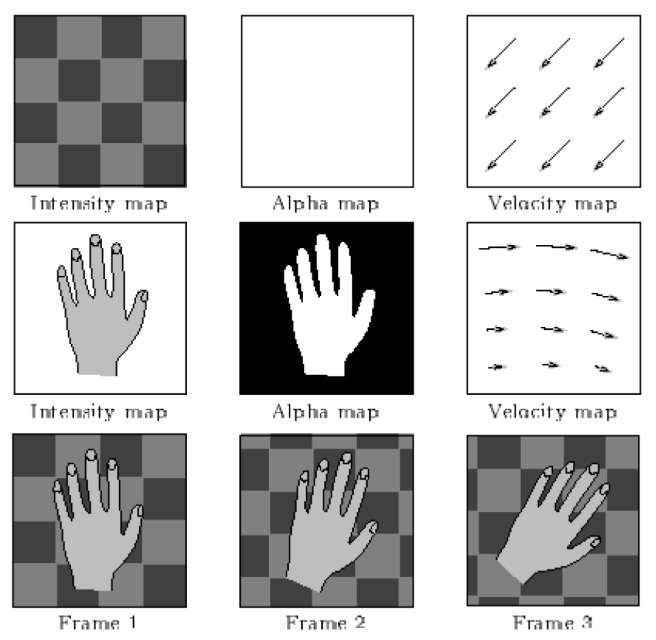
Coarse-to-fine LK with median filtering

Content

- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Other representations

Layer representation

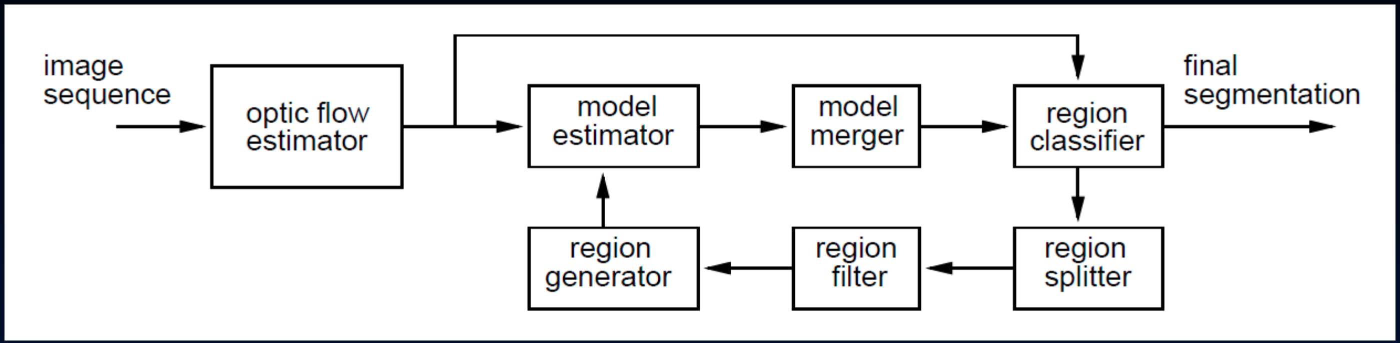
- Optical flow field is able to model complicated motion
- Different angle: a video sequence can be a composite of several moving layers
- Layers have been widely used
  - Adobe Photoshop
  - Adobe After Effect
- Compositing is straightforward, but inference is hard



Wang & Adelson, 1994

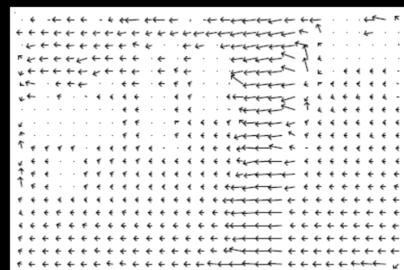
Wang & Adelson, 1994

- Strategy
  - Obtaining dense optical flow field
  - Divide a frame into non-overlapping regions and fit affine motion for each region
  - Cluster affine motions by k-means clustering
  - Region assignment by hypothesis testing
  - Region splitter: disconnected regions are separated

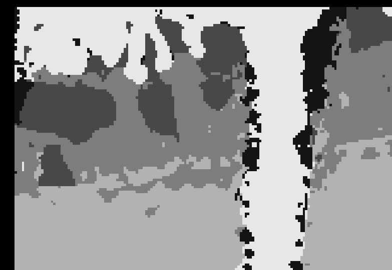


Results

Flower garden

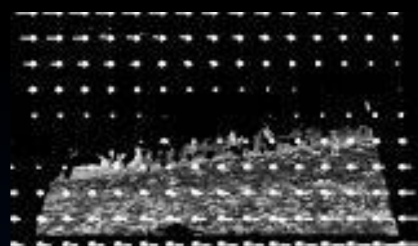


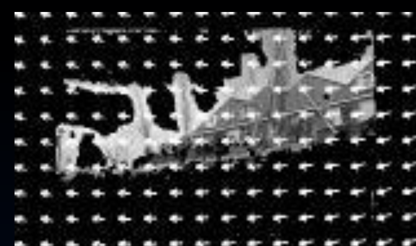
Optical flow field

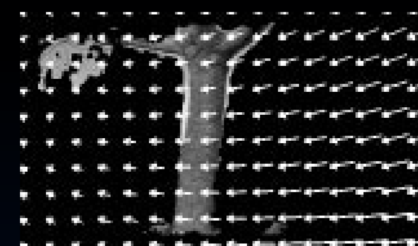


Clustering to affine regions

Clustering with error metric







Three layers with affine motion superimposed

Reconstructed background layer

Weiss & Adelson, 1996

- Chicken & egg problem
  - Good motion → good segmentation
  - Good segmentation → good motion
- We don't have either of them, so iterate!
- Perceptually organized expectation & maximization (POEM)
  - E-step: estimate the motion parameter of each layer
  - M-step: estimate the likelihood that a pixel belongs to each of the layers (segmentation)

Liu et. al. 2005

- Reliable layer segmentation for motion magnification
- Layer segmentation pipeline

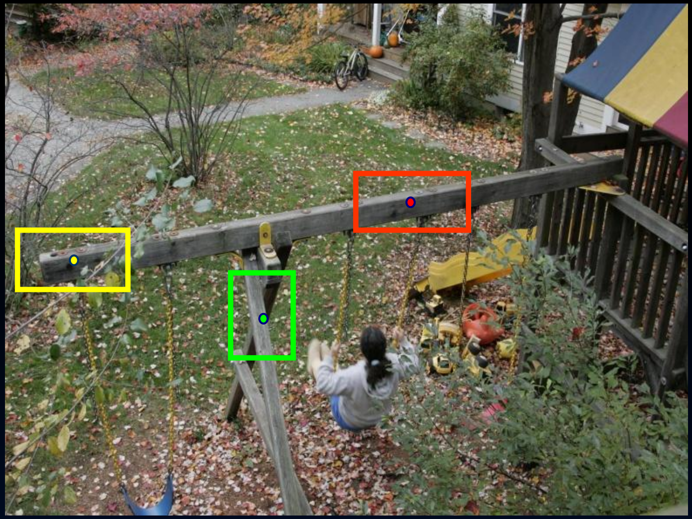
Feature point tracking → Trajectory clustering → Dense optical flow interpolation → Layer segmentation


Normalized Complex Correlation

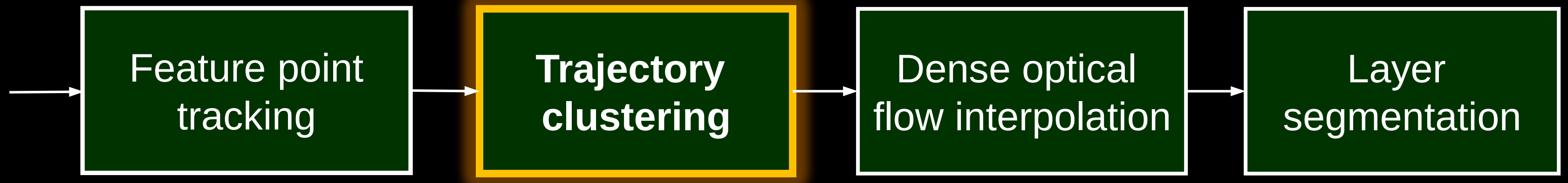
Feature point tracking → **Trajectory clustering** → Dense optical flow interpolation → Layer segmentation

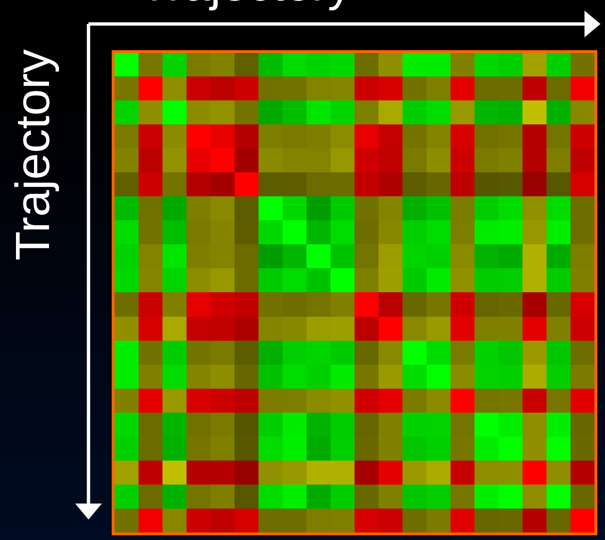
- The similarity metric should be independent of phase and magnitude
- Normalized complex correlation

$$S(C_1,C_2)=\frac{|\sum_t C_1(t)\overline{C}_2(t)|^2}{\sqrt{\sum_t C_1(t)\overline{C}_1(t)}\sqrt{\sum_t C_2(t)\overline{C}_2(t)}}$$



Spectral Clustering

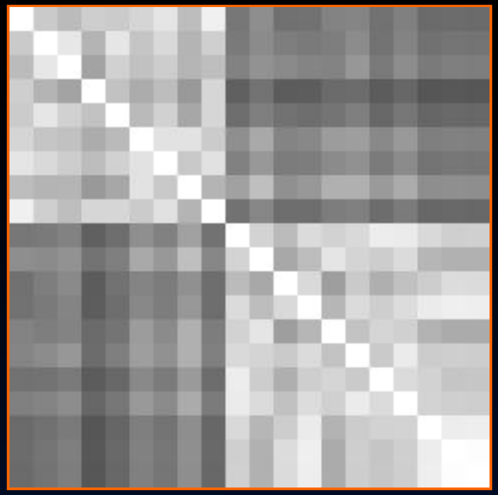




Affinity matrix

Two clusters

Clustering



Reordering of affinity matrix

Clustering Results

Feature point tracking → **Trajectory clustering** → Dense optical flow interpolation → Layer segmentation

From Sparse Feature Points to Dense Optical Flow Field

Feature point tracking → Trajectory clustering → **Dense optical flow interpolation** → Layer segmentation

- Interpolate dense optical flow field using locally weighted linear regression

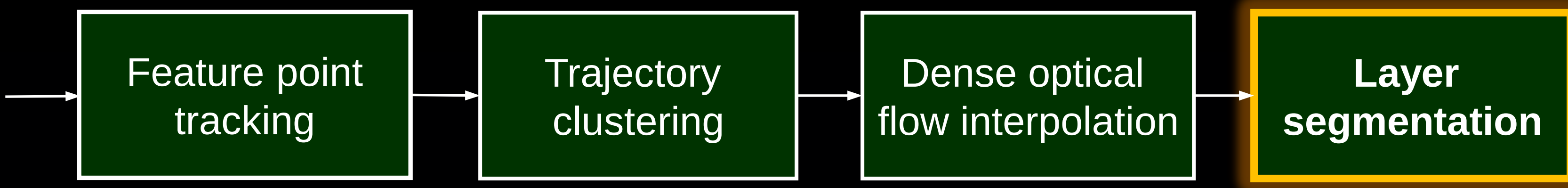
Flow vectors of clustered sparse feature points

Dense optical flow field of cluster 2 (swing)

Cluster 1: leaves

Cluster 2: swing

Motion Layer Assignment



- Assign each pixel to a motion cluster layer, using four cues:
  - **Motion likelihood**—consistency of pixel's intensity if it moves with the motion of a given layer (dense optical flow field)
  - **Color likelihood**—consistency of the color in a layer
  - **Spatial connectivity**—adjacent pixels favored to belong the same group
  - **Temporal coherence**—label assignment stays constant over time
- Energy minimization using graph cuts

Motion Magnification

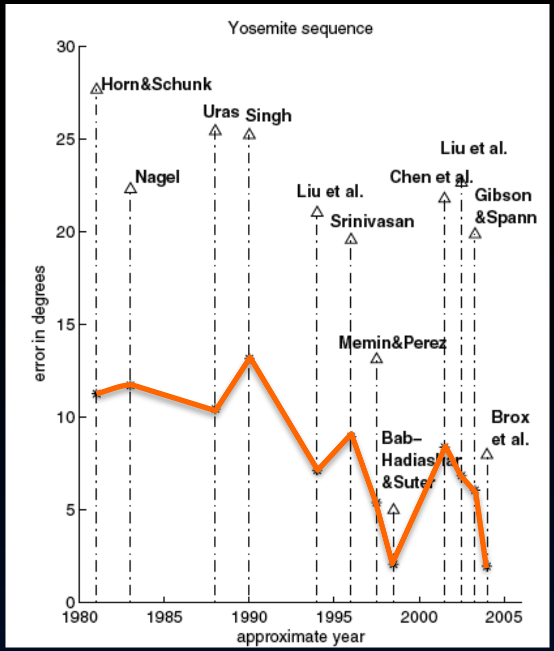
Ce Liu
Antonio Torralba
William T. Freeman
Fredo Durand
Edward H. Adelson

Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

SIGGRAPH2005
The 32nd International Conference on Computer Graphics and Interactive Techniques

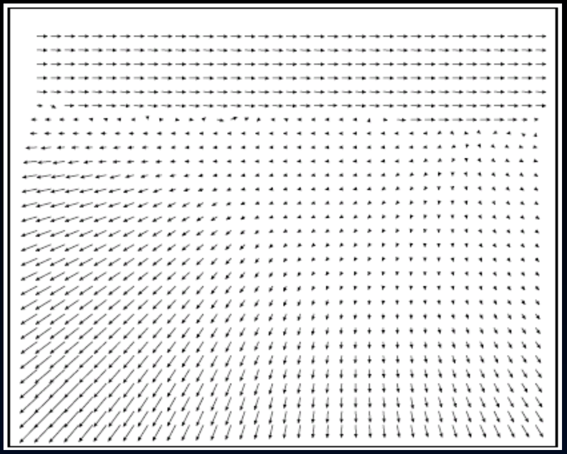
How good is optical flow?

- The AAE (average angular error) race on the *Yosemite* sequence for over 15 years



Improvement#

Yosemite sequence



State-of-the-art optical flow*

#I. Austvoll. Lecture Notes in Computer Science, 2005
*Brox *et al. ECCV*, 2004.

Middlebury flow database

Baker et. al. A Database and Evaluation Methodology for Optical Flow. ICCV 2007

Middlebury flow database

Optical flow evaluation results

Statistics: Average SD R0.5 R1.0 R2.0 A50 A75 A95

Error type: endpoint angle interpolation normalized interpolation

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1 all	disc	untext	Mequon (Hidden texture) GT im0 im1 all	disc	untext	Schefflera (Hidden texture) GT im0 im1 all	disc	untext	Wooden (Hidden texture) GT im0 im1 all	disc	untext
Adaptive [20]	4.4	0.09 1	0.26 1	0.06 1	0.23 5	0.78 4	[illegible] 5	0.54 8	[illegible] 10	0.21 3	0.18 1	0.91 3	0.10 1
Complementary OF [21]	5.7	0.11 5	0.28 3	0.10 9	0.18 1	0.63 [illegible]	0.12 1	0.31 3	0.75 [illegible]	0.18 1	0.19 2	0.97 5	0.12 3
Aniso. Huber-L1 [22]	5.8	0.10 3	0.28 3	0.08 3	0.31 11	0.88 [illegible]	0.28 12	[illegible] 10	[illegible]	0.29 12	0.20 4	0.92 4	0.13 5
DPOF [18]	6.1	0.13 12	0.35 12	0.09 4	0.25 6	0.79 5	[illegible]	[illegible]	[illegible] 1	0.21 3	0.19 2	0.62 1	0.15 11
TV-L1-improved [17]	7.2	0.09 1	0.26 1	0.07 2	0.20 3	0.71 3	0.16 2	0.53 7	1.18 9	0.22 5	0.21 7	1.24 11	0.11 2
CBF [12]	7.8	0.10 3	0.28 3	0.09 4	0.34 12	0.80 6	0.37 13	0.43 5	0.95 5	0.26 8	0.21 7	1.14 8	0.13 5
Brox et al. [5]	8.4	0.11 5	0.32 8	0.11 12	0.27 9	0.93 10	0.22 9	0.39 4	0.94 4	0.24 7	0.24 9	1.25 12	0.13 5
Rannacher [23]	8.5	0.11 5	0.31 6	0.09 4	0.25 6	0.84 7	0.21 8	0.57 12	1.27 15	0.26 8	0.24 9	1.32 14	0.13 5
F-TV-L1 [15]	8.8	0.14 13	0.35 12	0.14 15	0.34 12	0.98 12	0.26 11	0.59 14	1.19 10	0.26 8	0.27 13	1.36 15	0.16 12
Second-order prior [8]	9.0	0.11 5	0.31 6	0.09 4	0.26 8	0.93 10	0.20 7	0.57 12	1.25 14	0.26 8	0.20 4	1.04 6	0.12 3
Fusion [6]	9.4	0.11 5	0.34 10	0.10 9	0.19 2	0.69 2	0.16 2	0.29 2	0.66 2	0.23 6	0.20 4	1.19 10	0.14 9
Dynamic MRF [7]	11.1	0.12 11	0.34 10	0.11 12	0.22 4	0.89 9	0.16 2	0.44 6	1.13 7	0.20 2	0.24 9	1.29 13	0.14 9
SegOF [10]	11.7	0.15 14	0.36 14	0.10 9	0.57 15	1.16 15	0.59 19	0.68 15	1.24 12	0.64 14	0.32 15	0.86 2	0.26 15
Learning Flow [11]	13.3	0.11 5	0.32 8	0.09 4	0.29 10	0.99 13	0.23 10	0.55 9	1.24 12	0.29 12	0.36 16	1.56 17	0.25 14
Filter Flow [19]	14.3	0.17 16	0.39 16	0.13 14	0.43 14	1.09 14	0.38 14	0.75 16	1.34 16	0.78 19	0.70 19	1.54 16	0.68 19
GraphCuts [14]	14.5	0.16 15	0.38 15	0.14 15	0.59 18	1.36 19	0.46 15	0.56 10	1.07 6	0.64 14	0.26 12	1.14 8	0.17 13
Black & Anandan [4]	15.0	0.18 17	0.42 17	0.19 18	0.58 17	1.31 17	0.50 16	0.95 19	1.58 18	0.70 16	0.49 17	1.59 18	0.45 17
SPSA-learn [13]	15.7	0.18 17	0.45 18	0.17 17	0.57 15	1.32 18	0.51 17	0.84 17	1.50 17	0.72 17	0.52 18	1.64 19	0.49 18
GroupFlow [9]	15.9	0.21 19	0.51 19	0.21 19	0.79 21	1.69 21	0.72 21	0.86 18	1.64 19	0.74 18	0.30 14	1.07 7	0.26 15
2D-CLG [1]	17.4	0.28 21	0.62 22	0.21 19	0.67 20	1.21 16	0.70 20	1.12 21	1.80 21	0.99 22	1.07 22	2.06 21	1.12 22
Horn & Schunck [3]	18.6	0.22 20	0.55 20	0.22 21	0.61 19	1.53 20	0.52 18	1.01 20	1.73 20	0.80 20	0.78 20	2.02 20	0.77 20
TI-DOFE [24]	19.6	0.38 23	0.64 23	0.47 23	1.16 22	1.72 22	1.26 22	1.39 23	2.06 24	1.17 23	1.29 23	2.21 23	1.41 23
FOLKI [16]	22.6	0.29 22	0.73 24	0.33 22	1.52 23	1.96 24	1.80 23	1.23 22	2.04 23	0.95 21	0.99 21	2.20 22	1.08 21
Pyramid LK [2]	23.7	0.39 24	0.61 21	0.61 24	1.67 24	1.78 23	2.00 24	1.50 24	1.97 22	1.38 24	1.57 24	2.39 24	1.78 24

Average endpoint error	Grove (Synthetic) GT im0 im1 all	disc	untext	Urban (Synthetic) GT im0 im1 all	disc	untext	Yosemite (Synthetic) GT im0 im1 all	disc	untext	Teddy (Stereo) GT im0 im1 all	disc	untext
Adaptive [20]	0.88 3	1.25 3	0.73 5	0.50 3	1.28 3	0.31 3	0.14 10	0.16 12	0.22 10	0.65 3	1.37 3	0.79 4
Complementary OF [21]	0.97 10	1.31 6	1.00 11	1.78 20	1.73 7	0.87 14	0.11 4	0.12 2	0.22 10	0.68 4	1.48 4	0.95 8
Aniso. Huber-L1 [22]	0.84 2	1.20 2	0.70 2	0.39 1	1.23 1	0.28 1	0.17 15	0.15 9	0.27 16	0.64 2	1.36 2	0.79 4
DPOF [18]	0.74 1	1.09 1	0.49 1	0.66 7	1.80 10	0.63 8	0.19 17	0.17 14	0.35 20	0.50 1	1.08 1	0.55 1
TV-L1-improved [17]	0.90 4	1.31 6	0.72 3	1.51 14	1.93 11	0.84 11	0.18 16	0.17 14	0.31 17	0.73 8	1.62 9	0.87 7
CBF [12]	0.90 4	1.27 4	0.82 7	0.41 2	1.23 1	0.30 2	0.23 22	0.19 20	0.39 21	0.76 9	1.56 6	1.02 9
Brox et al. [5]	1.10 13	1.39 12	1.43 17	0.89 8	1.77 8	0.55 7	0.10 2	0.13 4	0.11 1	0.91 11	1.83 12	1.13 12
Rannacher [23]	0.91 7	1.33 8	0.72 3	1.49 13	1.95 13	0.78 9	0.15 12	0.14 7	0.26 13	0.69 6	1.58 8	0.86 6
F-TV-L1 [15]	0.90 4	1.30 5	0.76 6	0.54 4	1.62 6	0.36 4	0.13 6	0.15 9	0.20 9	0.68 4	1.56 6	0.66 2
Second-order prior [8]	0.94 8	1.34 9	0.83 9	0.61 6	1.93 11	0.47 6	0.20 18	0.16 12	0.34 19	0.77 10	1.64 10	1.07 10
Fusion [6]	1.07 11	1.42 13	1.22 13	1.35 10	1.49 5	0.86 13	0.20 18	0.20 21	0.26 13	1.07 14	2.07 16	1.39 16
Dynamic MRF [7]	1.11 14	1.52 17	1.13 12	1.54 15	2.37 20	0.93 15	0.13 6	0.12 2	0.31 17	1.27 18	2.33 20	1.66 17
SegOF [10]	1.18 17	1.50 16	1.47 18	1.63 18	2.09 14	0.96 16	0.08 1	0.13 4	0.12 2	0.70 7	1.50 5	0.69 3
Learning Flow [11]	1.25 19	1.64 21	1.41 16	1.55 17	2.32 19	0.85 12	0.14 10	0.18 18	0.24 12	1.09 15	2.09 18	1.27 13
Filter Flow [19]	1.13 16	1.38 11	1.51 19	0.57 5	1.32 4	0.44 5	0.22 20	0.23 23	0.26 13	0.96 12	1.66 11	1.12 11
GraphCuts [14]	0.96 9	1.35 10	0.84 10	2.25 23	1.79 9	1.22 21	0.22 20	0.17 14	0.43 22	1.22 17	2.05 15	1.78 19
Black & Anandan [4]	1.08 12	1.42 13	1.22 13	1.43 11	2.28 17	0.83 10	0.15 12	0.17 14	0.17 6	1.11 16	1.98 14	1.30 14
SPSA-learn [13]	1.12 15	1.42 13	1.39 15	1.75 19	2.14 15	1.06 20	0.13 6	0.13 4	0.19 7	1.32 19	2.08 17	1.73 18
GroupFlow [9]	1.29 22	1.81 22	0.82 7	1.94 21	2.30 18	1.36 22	0.11 4	0.14 7	0.19 7	1.06 13	1.96 13	1.35 15
2D-CLG [1]	1.23 18	1.52 17	1.62 22	1.54 15	2.15 16	0.96 16	0.10 2	0.11 1	0.16 4	1.38 20	2.26 19	1.83 20
Horn & Schunck [3]	1.26 20	1.58 19	1.55 20	1.43 11	2.59 22	1.00 18	0.16 14	0.18 18	0.15 3	1.51 21	2.50 21	1.88 21
TI-DOFE [24]	1.27 21	1.61 20	1.57 21	1.28 9	2.57 21	1.01 19	0.13 6	0.15 9	0.16 4	1.87 22	2.71 22	2.53 22
FOLKI [16]	1.53 23	1.85 23	2.07 23	2.14 22	3.23 24	1.60 23	0.26 23	0.21 22	0.68 23	2.67 23	3.27 23	4.32 23
Pyramid LK [2]	2.94 24	3.72 24	2.98 24	3.33 24	2.74 23	2.43 24	0.30 24	0.24 24	0.73 24	3.80 24	5.08 24	4.88 24

Move the mouse over the numbers in the table to see the corresponding images. Click to compare with the ground truth.

Measuring motion for real-life videos

- Challenging because of occlusion, shadow, reflection, motion blur, sensor noise and compression artifacts

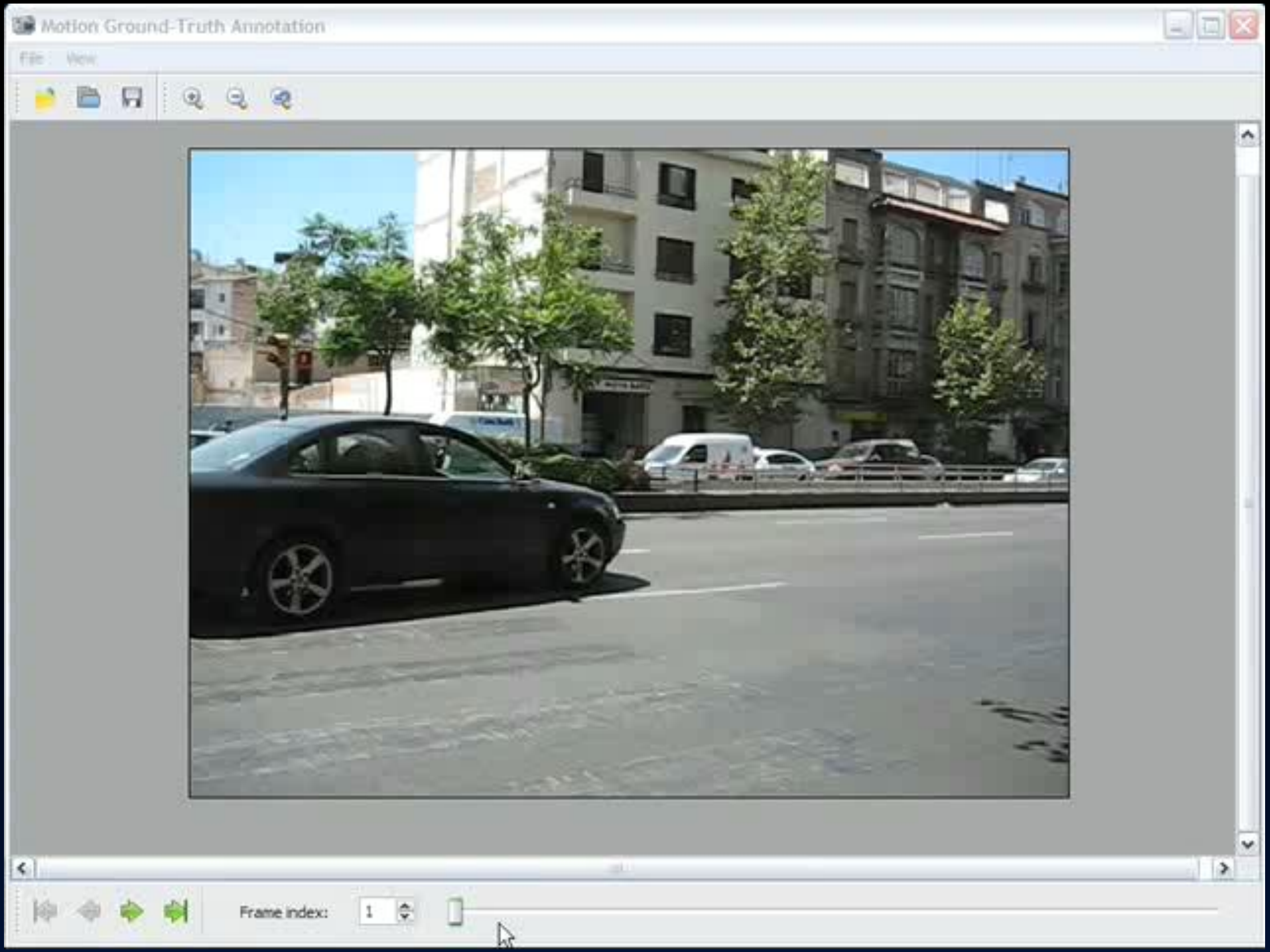
[Video courtesy: Antonio Torralba]

- Accurately measuring motion also has great impact in scientific measurement and graphics applications
- Humans are experts in perceiving motion. Can we use human expertise to annotate motion?

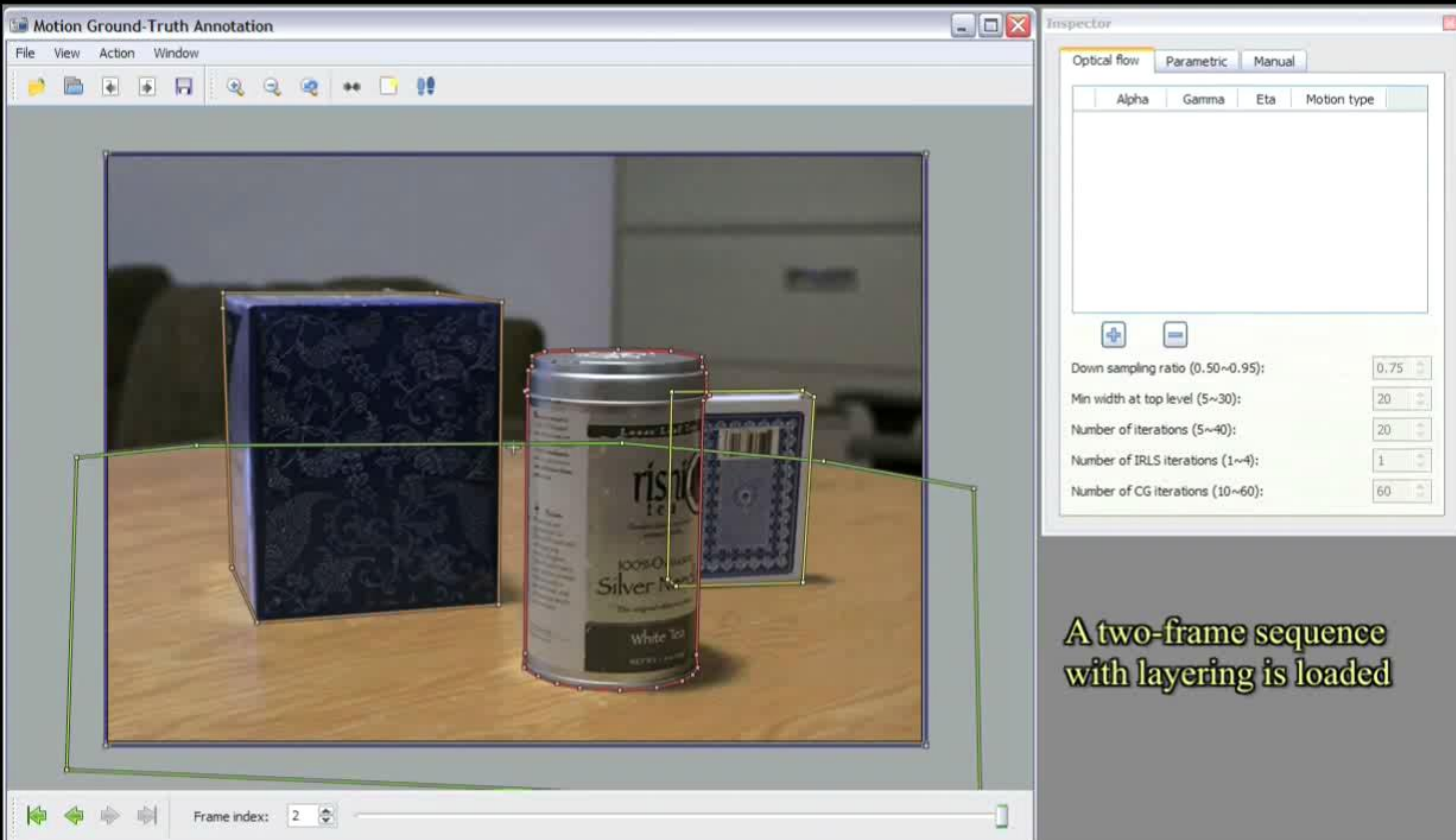
Human-assisted motion annotation

- Our approach: an interactive system to combine human perception and the state-of-the-art computer vision algorithms to annotate motion
- Use layers as the interface for user interaction
  - Decompose a video sequence into layers
  - Motion analysis for each layer

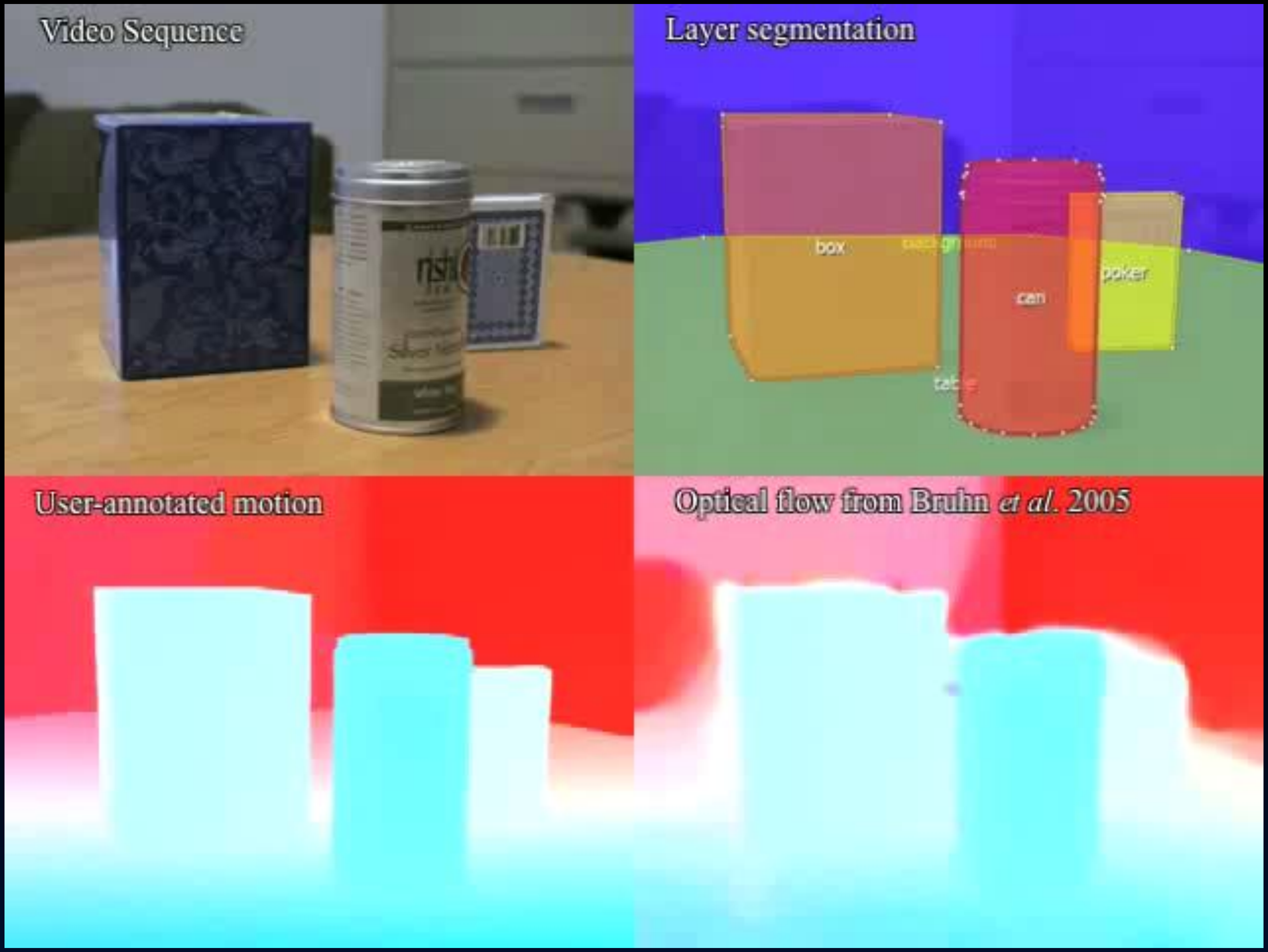
Demo: interactive layer segmentation

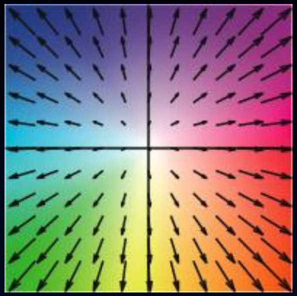


Demo: interactive motion labeling



Motion database of natural scenes

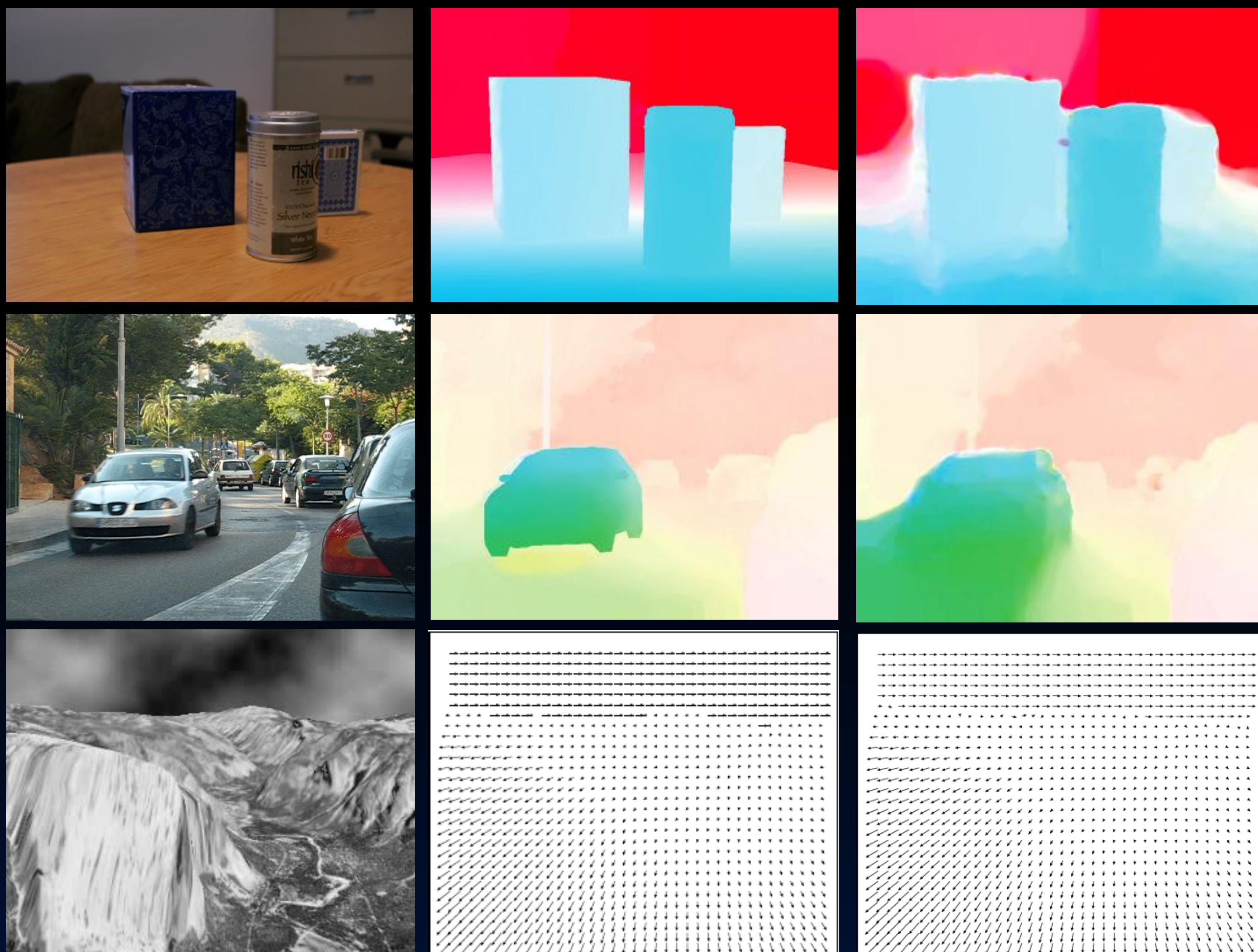




Color map

Bruhn *et al*. Lucas/Kanade meets Horn/Schunck: combining local and global optical flow methods. *IJCV*, 2005

Optical flow is far from being solved



AAE=8.99°

AAE=5.24°

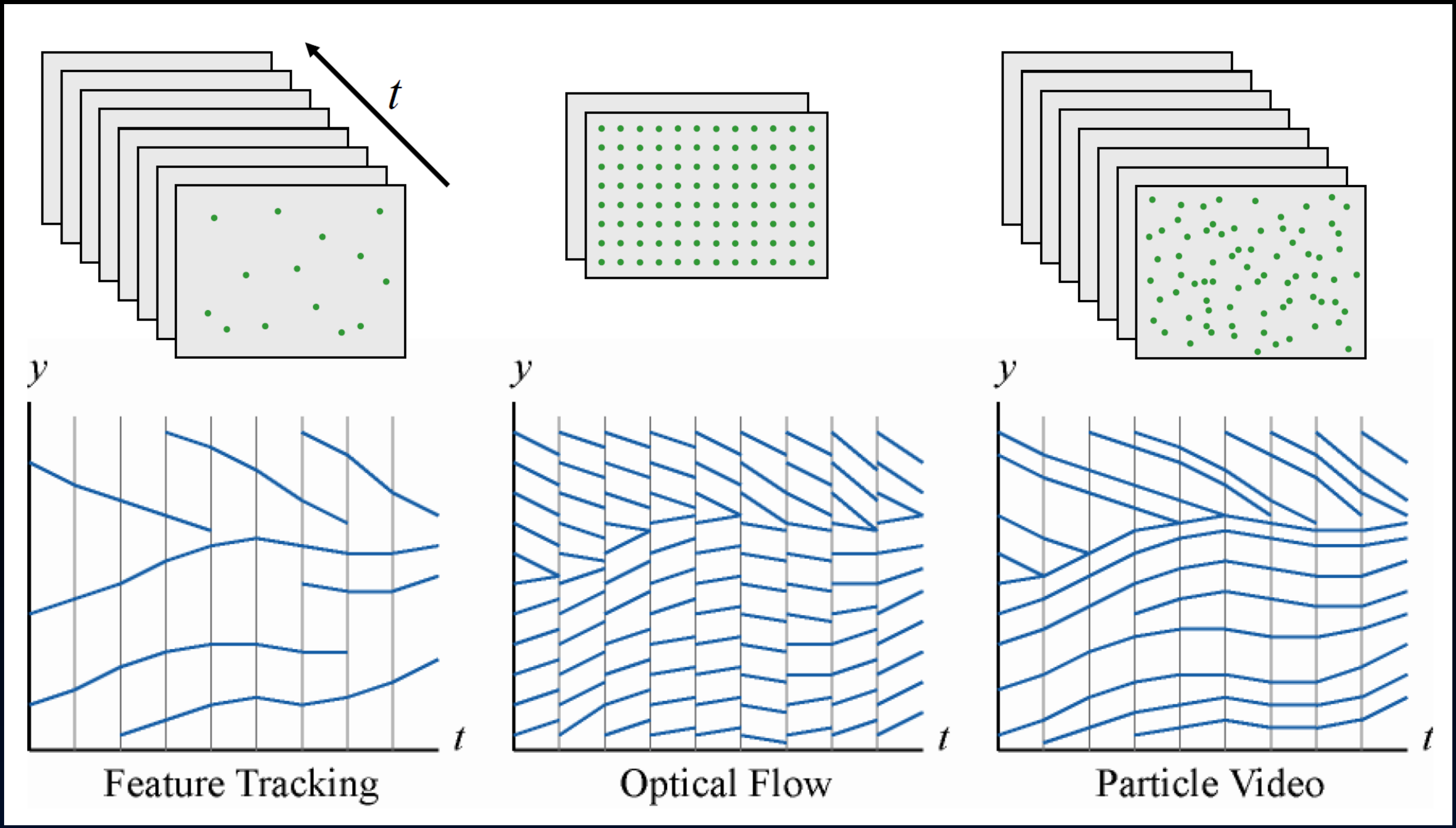
AAE=1.94°

Frame | Ground-truth motion | Optical flow

Content

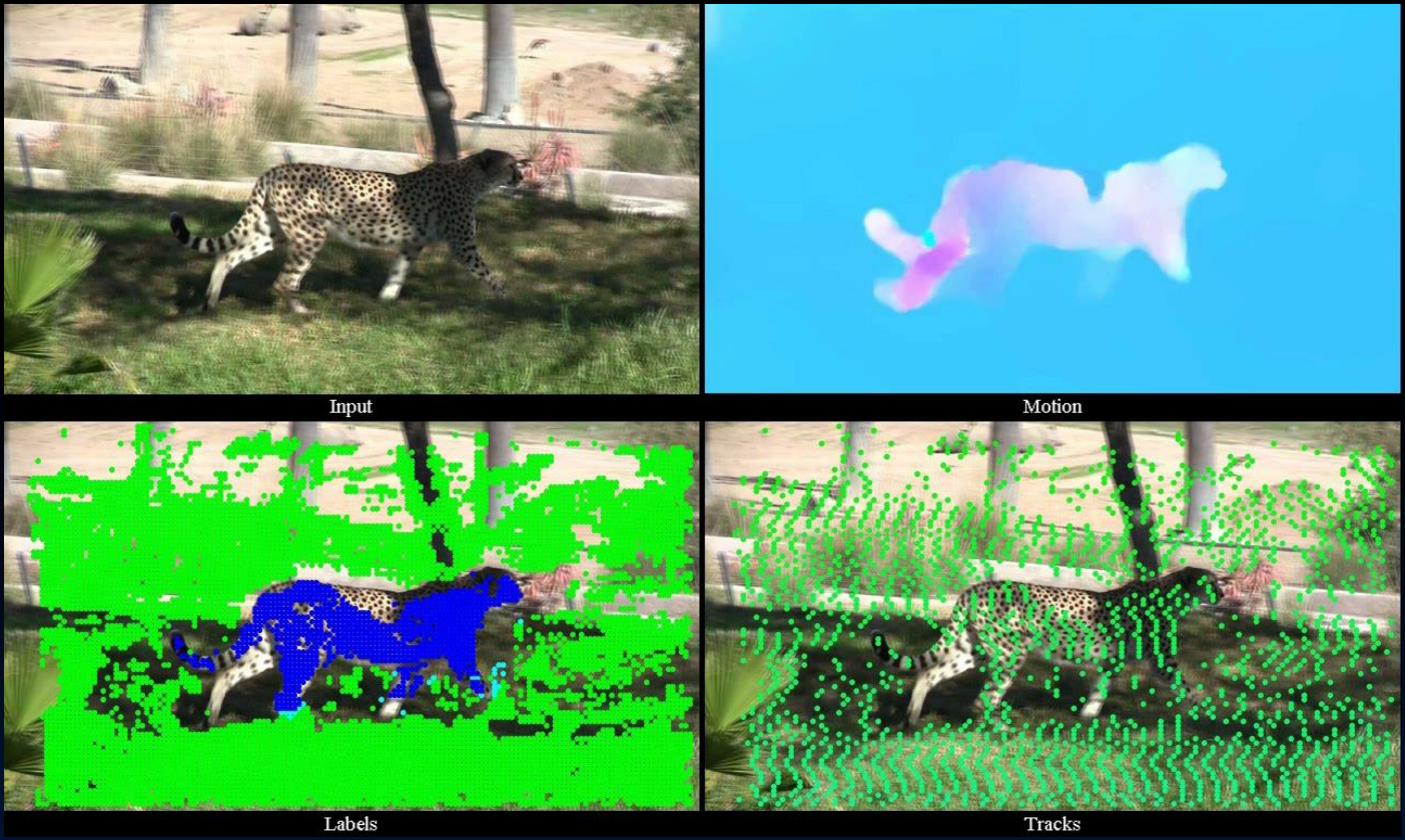
- Robust optical flow estimation
- Applications
- Feature matching
- Discrete optical flow
- Layer motion analysis
- Other representations

Particle video

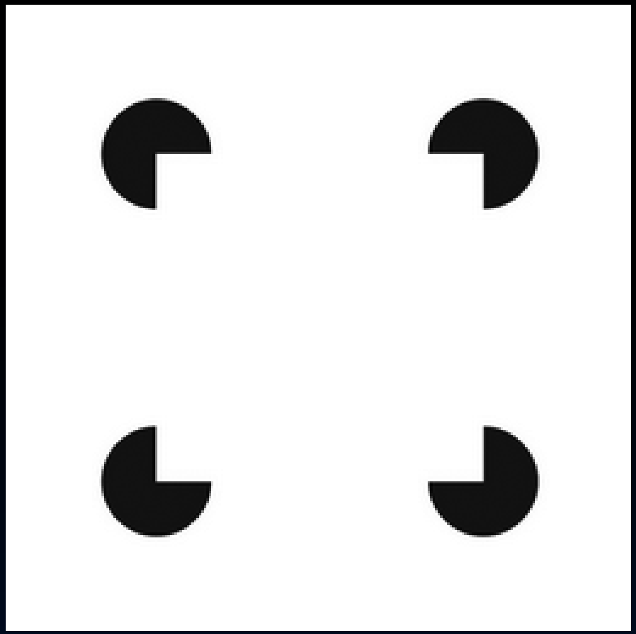


P. Sand and S. Teller. Particle Video: Long-Range Motion Estimation using Point Trajectories. CVPR 2006

Particle video



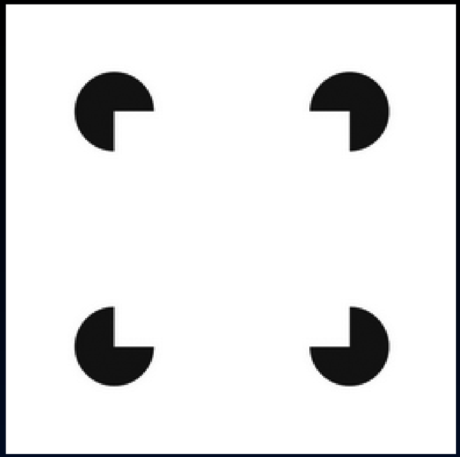
Seemingly Simple Examples



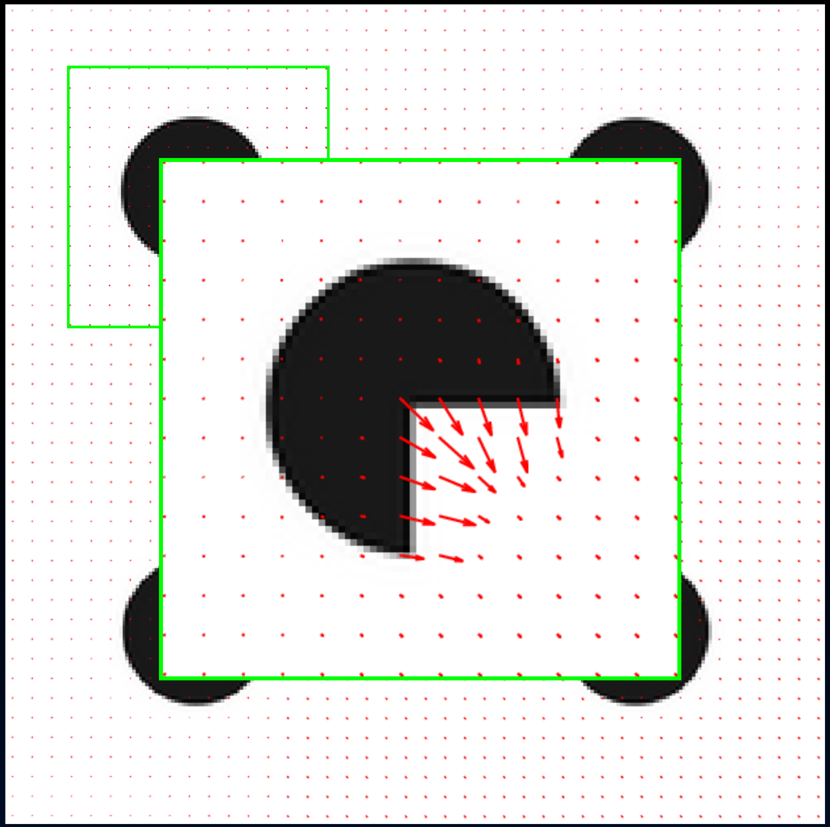
Kanizsa square

From real video

Output from the State-of-the-Art Optical Flow Algorithm



Kanizsa square

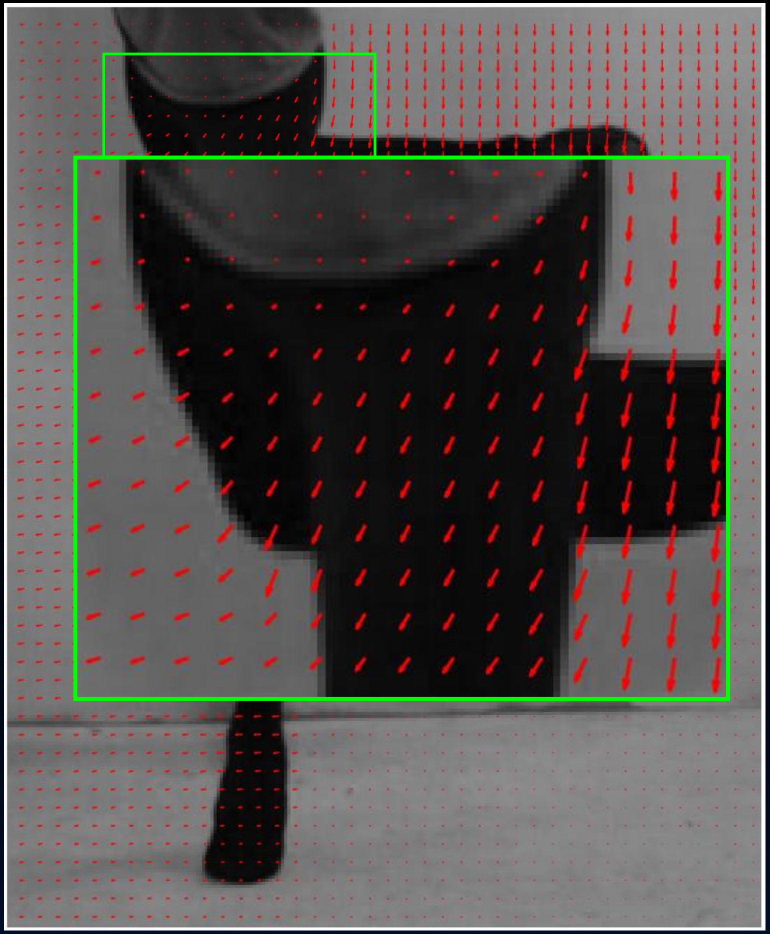


Optical flow field

T. Brox *et al*. High accuracy optical flow estimation based on a theory for warping. *ECCV* 2004

Output from the State-of-the-Art Optical Flow Algorithm

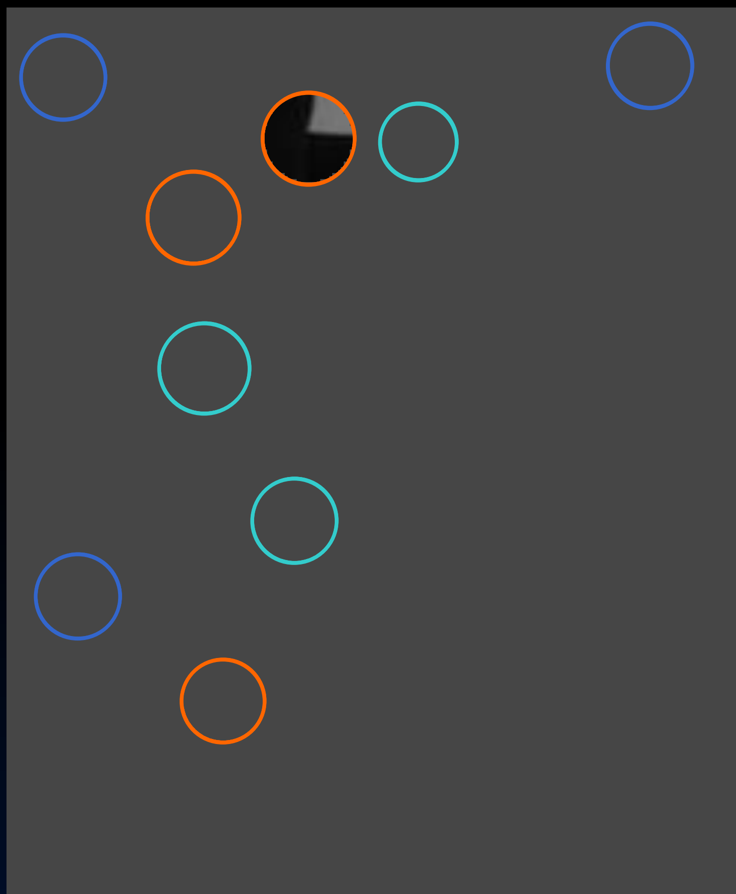
Dancer

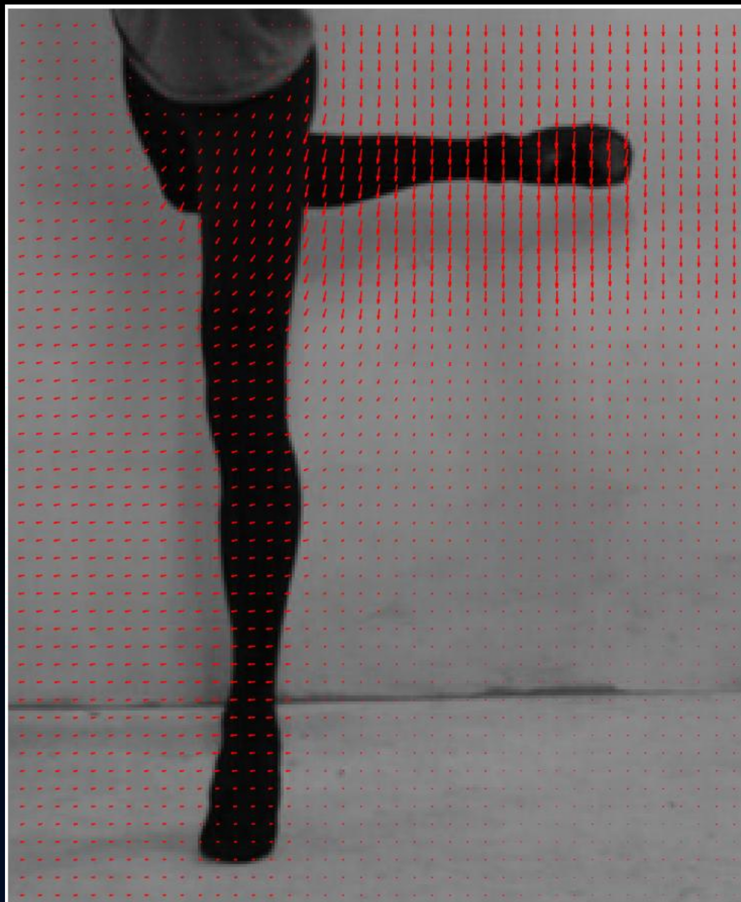


Optical flow field

T. Brox *et al*. High accuracy optical flow estimation based on a theory for warping. *ECCV* 2004

Optical flow representation: aperture problem





Corners

Lines

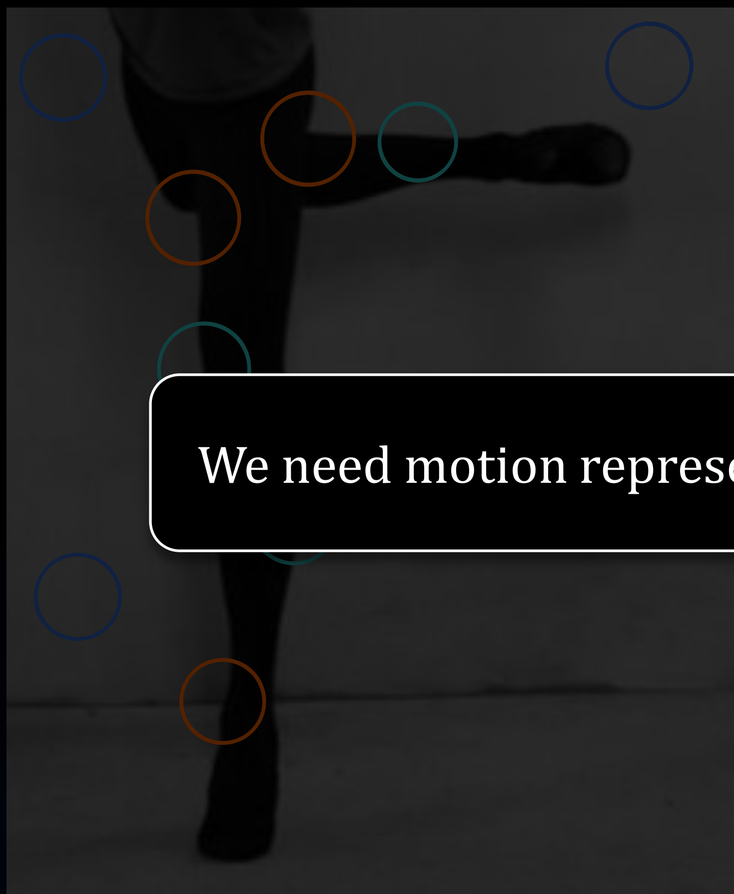
Flat regions

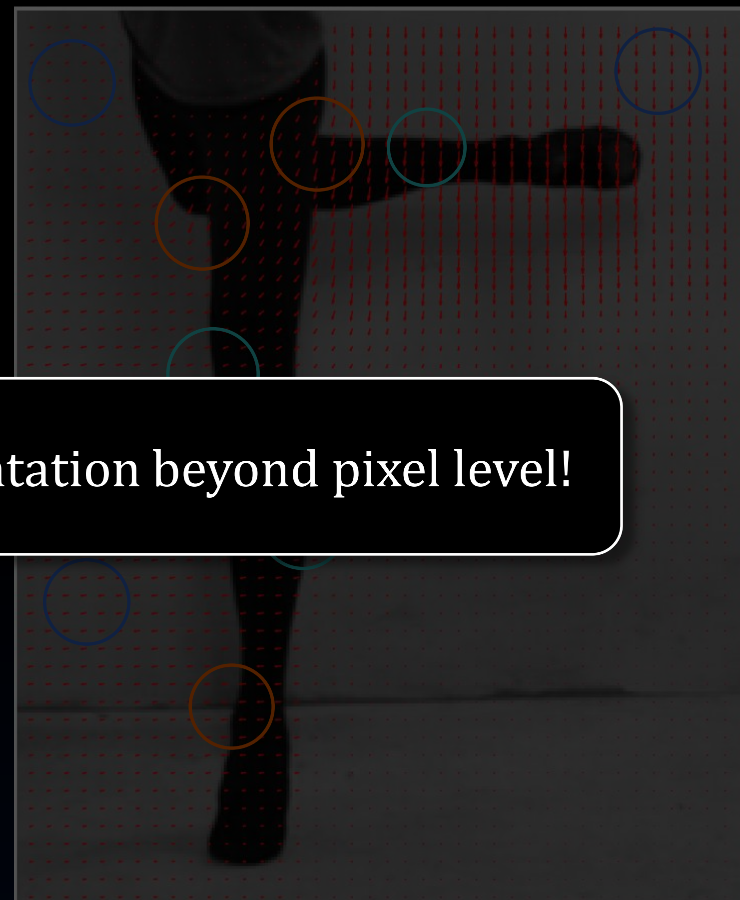
Spurious junctions

Boundary ownership

Illusory boundaries

Optical flow representation: aperture problem





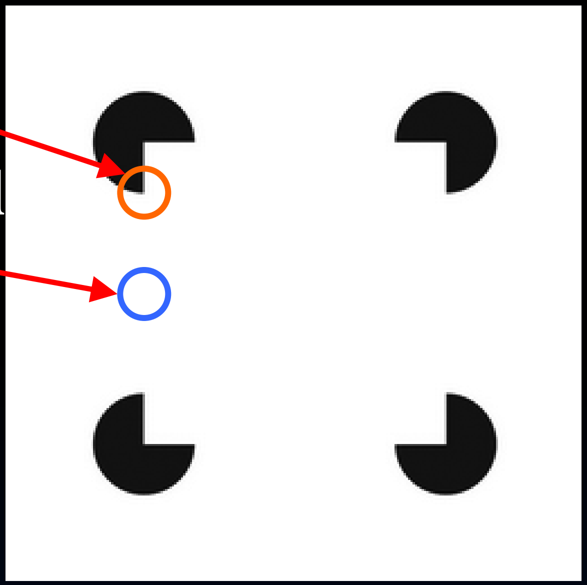
We need motion representation beyond pixel level!

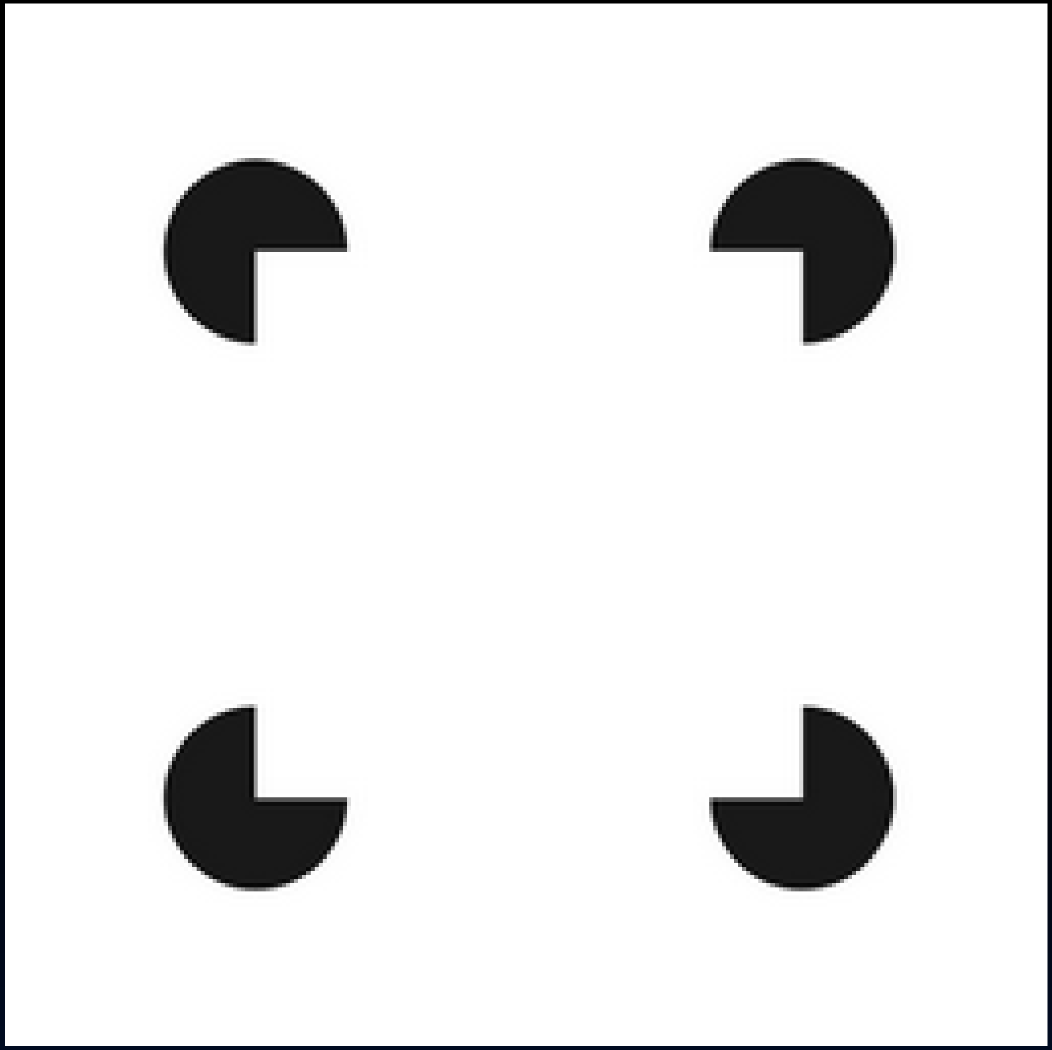
Corners | Lines | Flat regions

Spurious junctions | Boundary ownership | Illusory boundaries

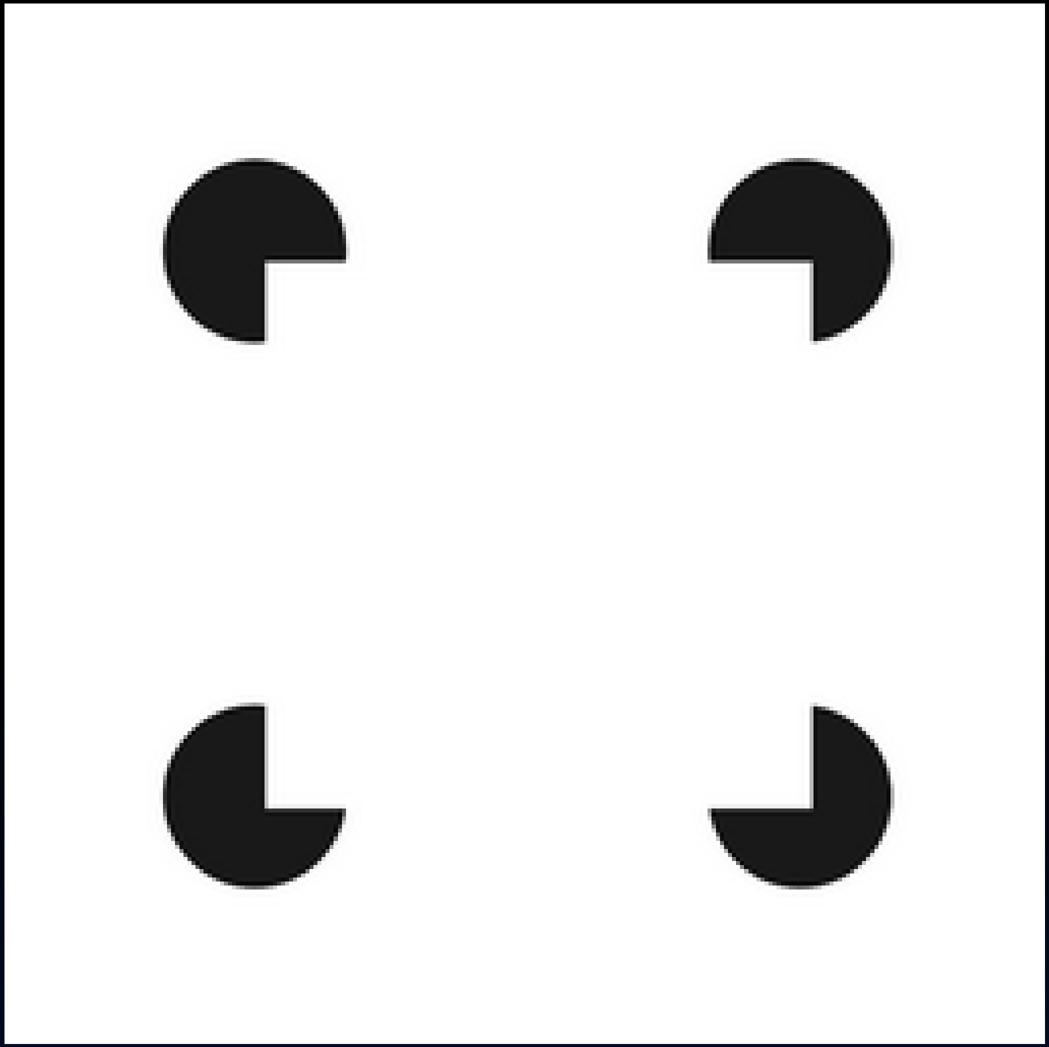
Challenge: Textureless Objects under Occlusion

- Corners are not always trustworthy (junctions)
- Flat regions do not always move smoothl (discontinuous at illusory boundaries)
- How about boundaries?
  - Easy to detect and track for textureless objects
  - Able to handle junctions with illusory boundaries

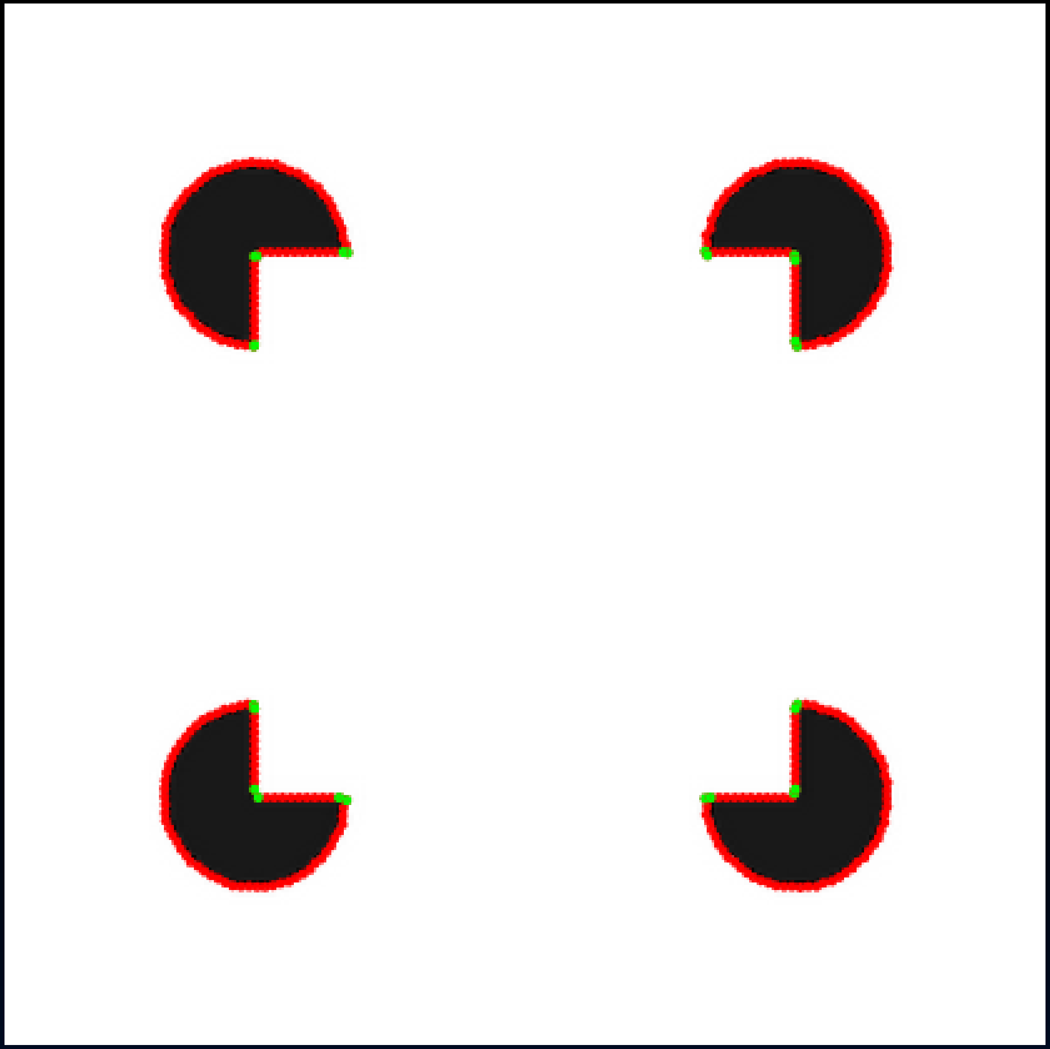




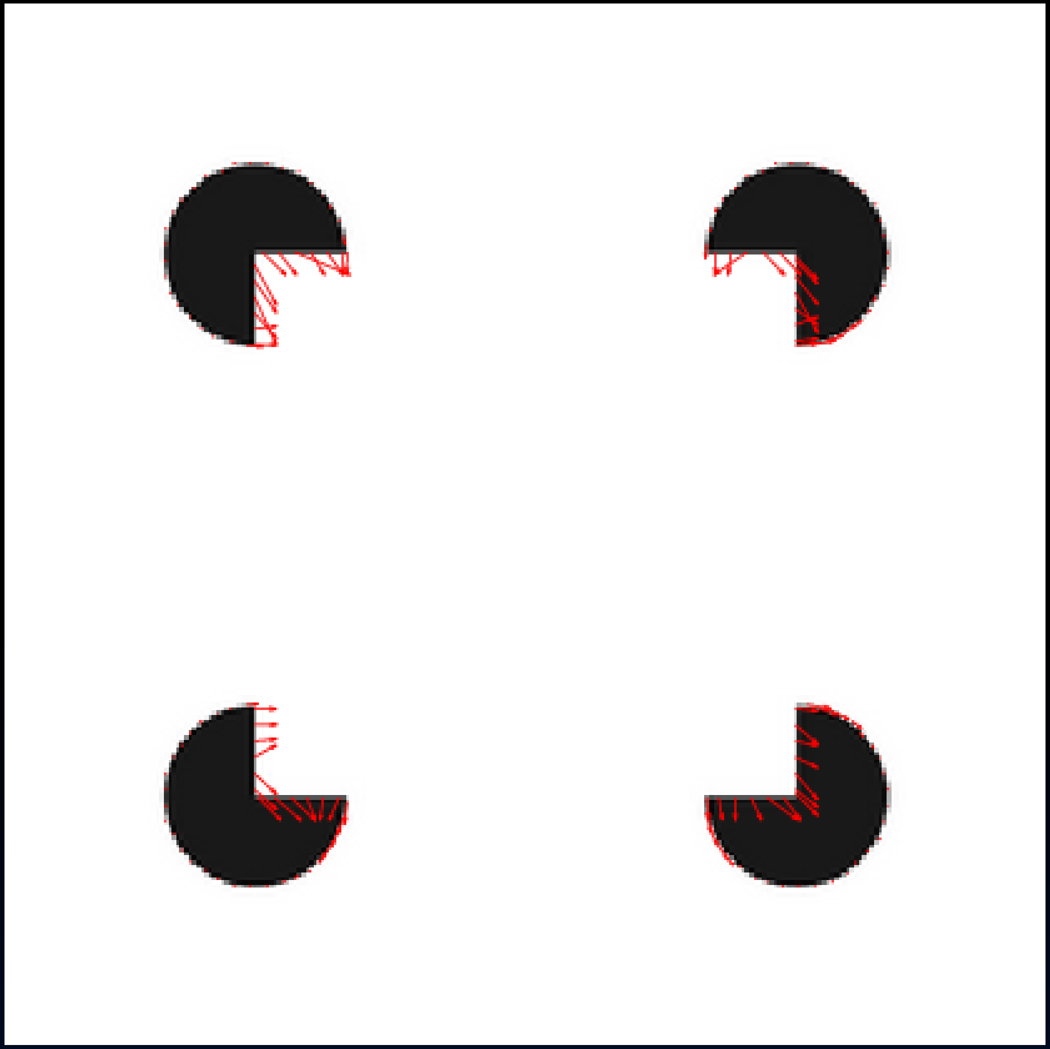
Frame 1



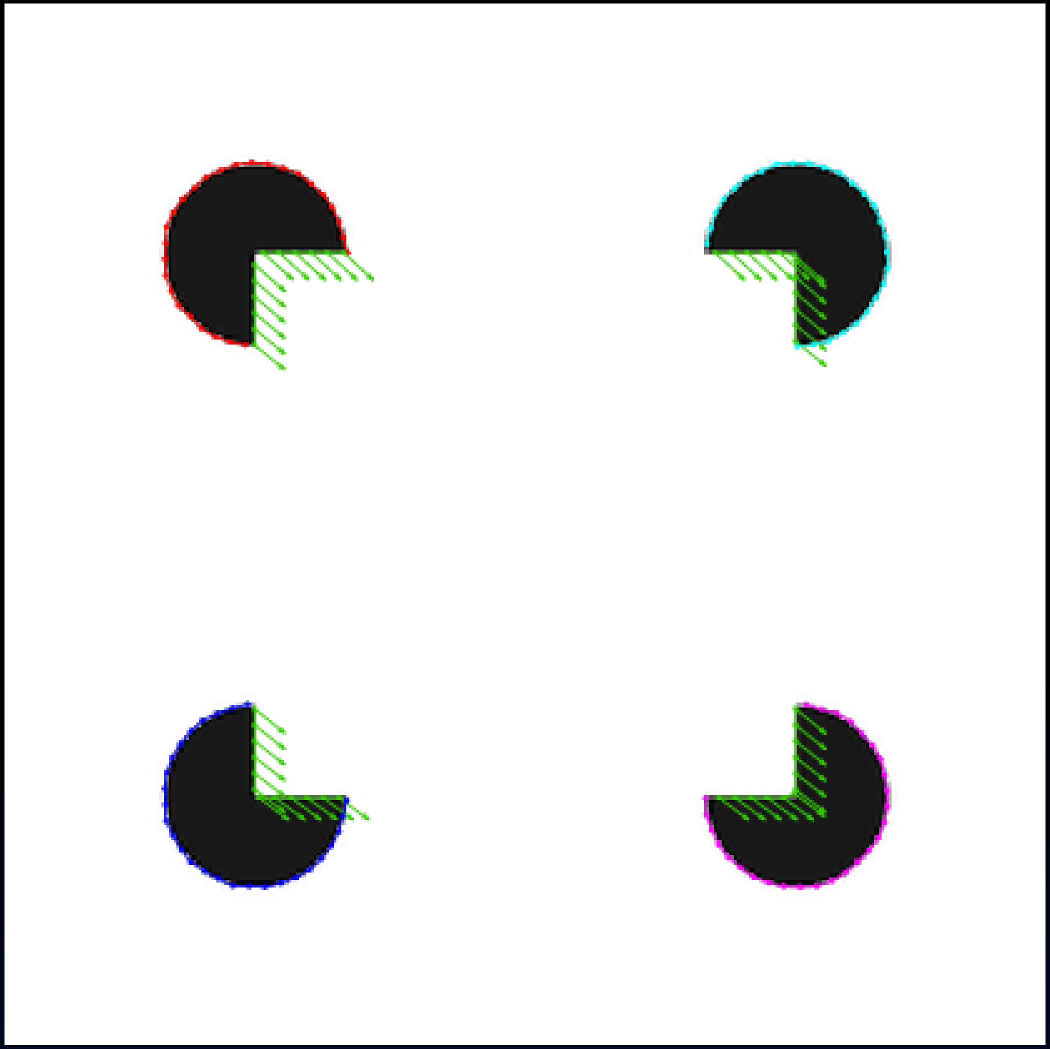
Frame 2



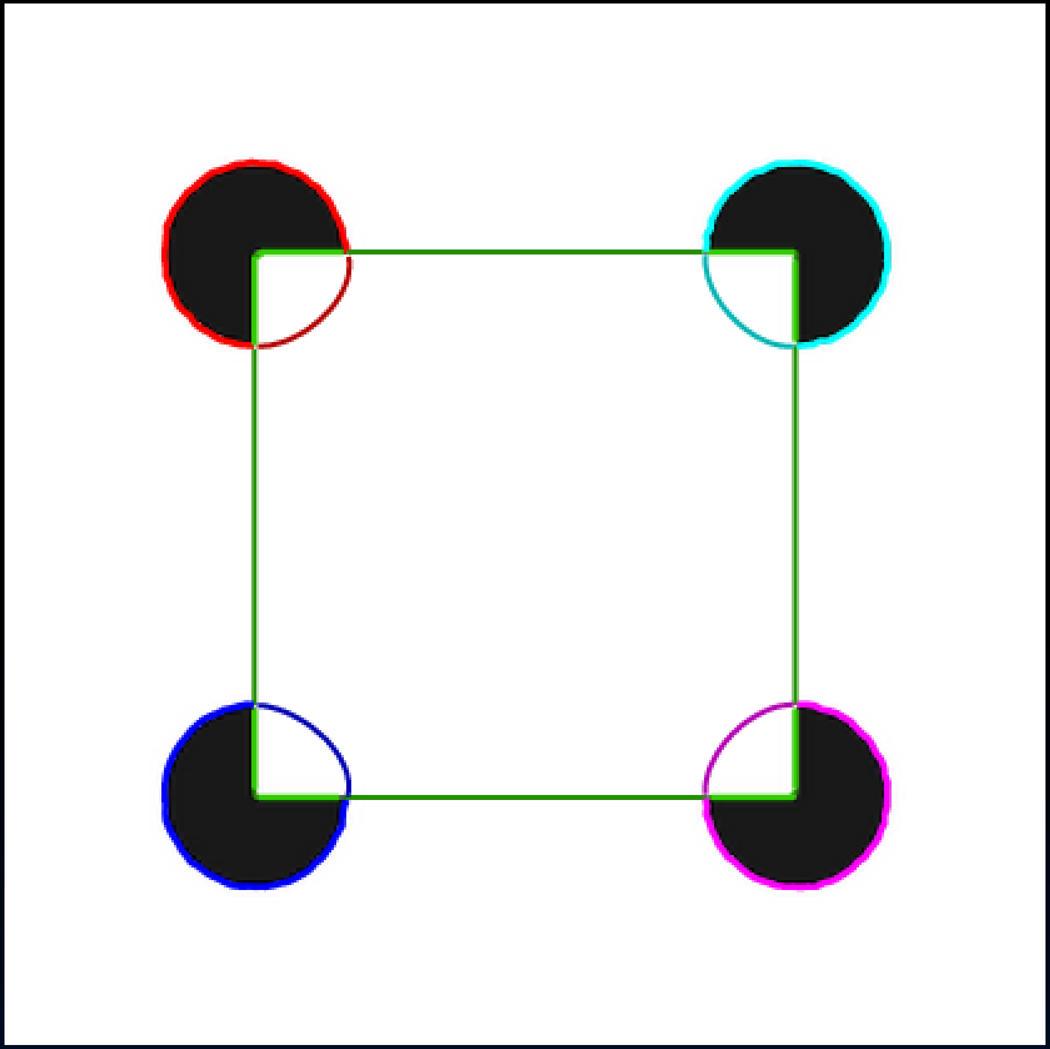
Extracted boundary fragments



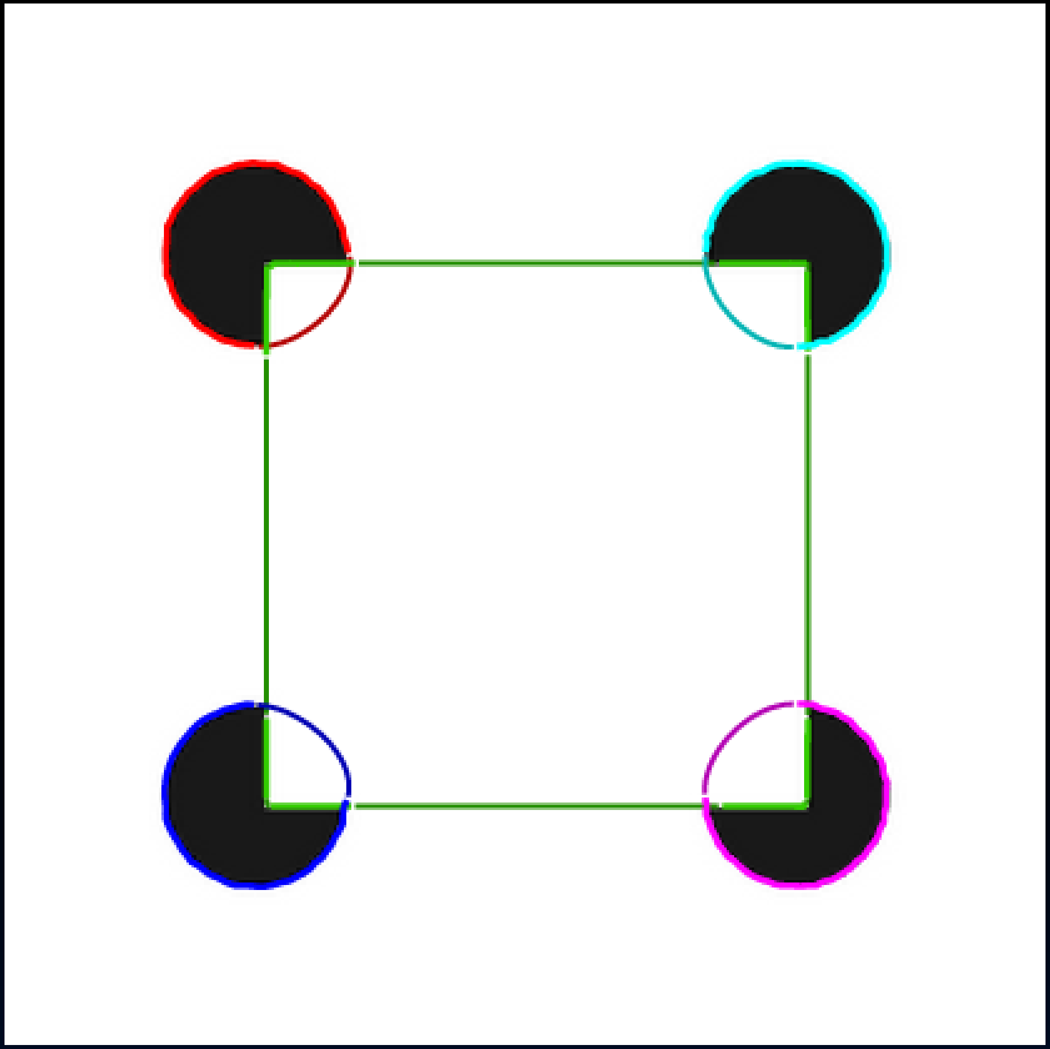
Optical flow from Lucas-Kanade algorithm



Estimated motion by our system, after grouping



Boundary grouping and illusory boundaries (frame 1)



Boundary grouping and illusory boundaries (frame 2)

Rotating Chair

Frame 1

Frame 2

Extracted boundary fragments

Estimated flow field from Brox et al.

Estimated motion by our system, after grouping

Boundary grouping and illusory boundaries (frame 1)

Boundary grouping and illusory boundaries (frame 2)