# Computer Vision

# CSE/ECE 576
# Motion and Optical Flow

## Linda Shapiro
Professor of Computer Science & Engineering
Professor of Electrical & Computer Engineering

# We live in a moving world

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

# Motion and perceptual organization

- Even "impoverished" motion data can evoke a strong percept
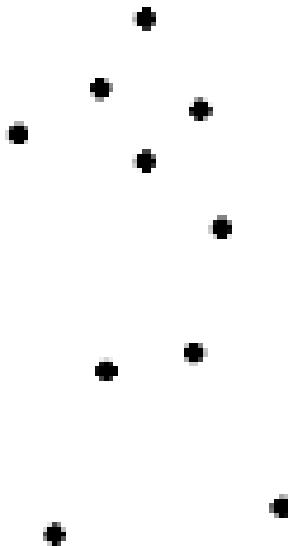
G. Johansson, "Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

# Motion and perceptual organization

- Even "impoverished" motion data can evoke a strong percept
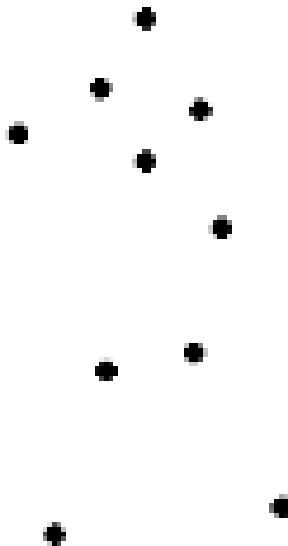
G. Johansson, "Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

# Seeing motion from a static picture?
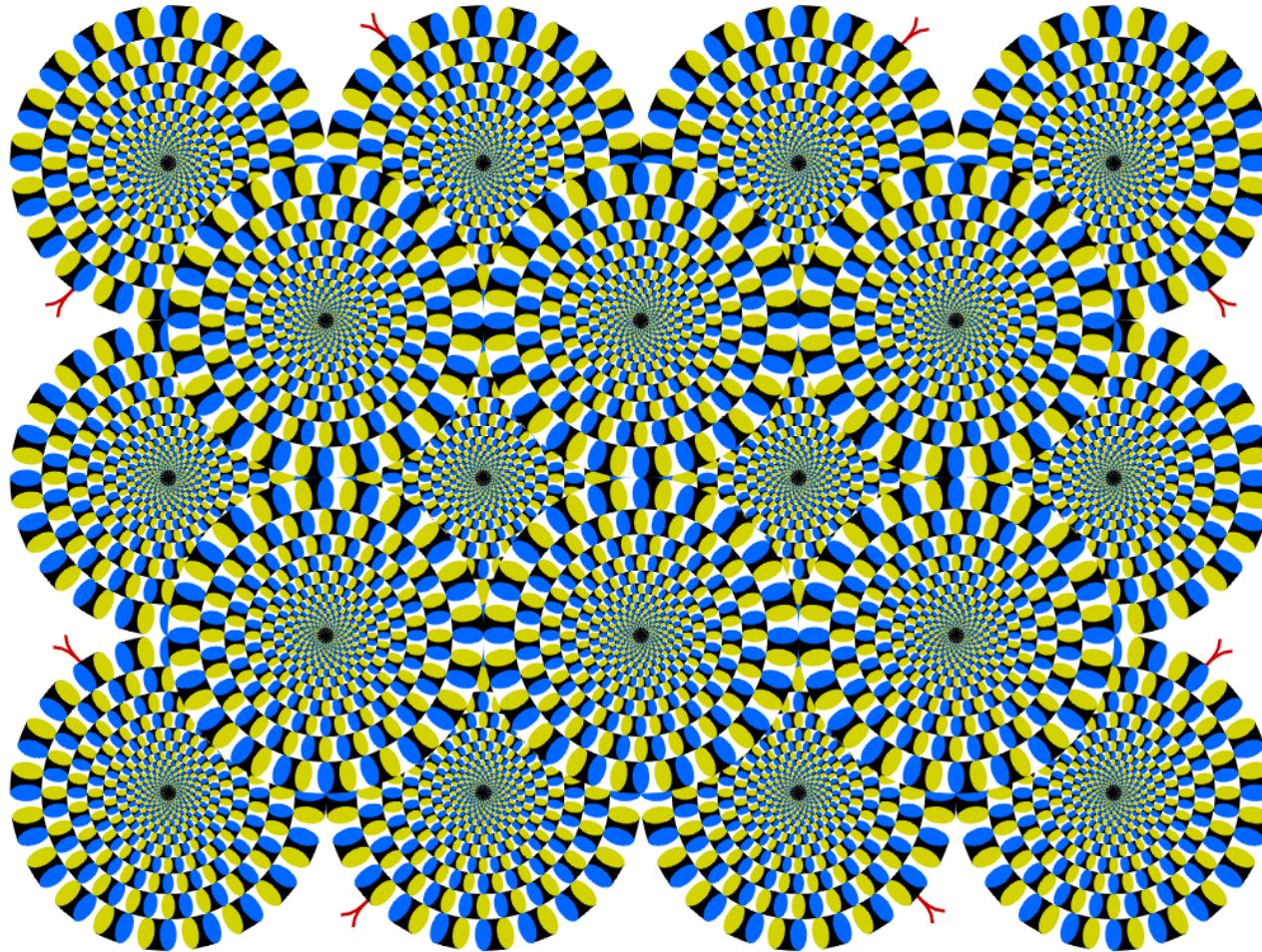
# More examples
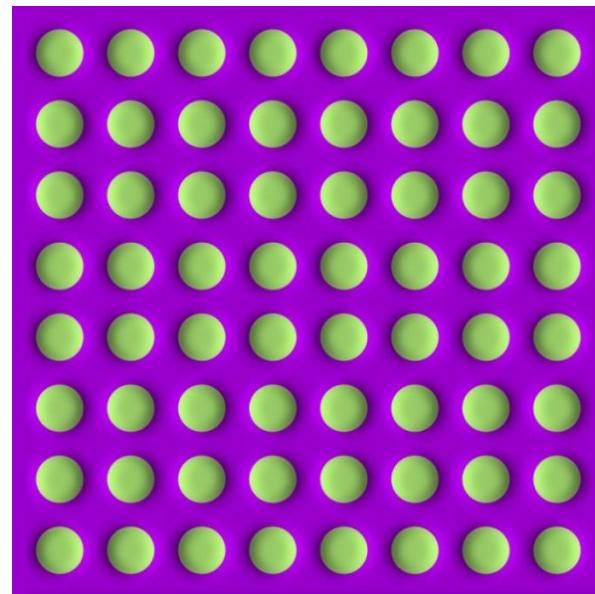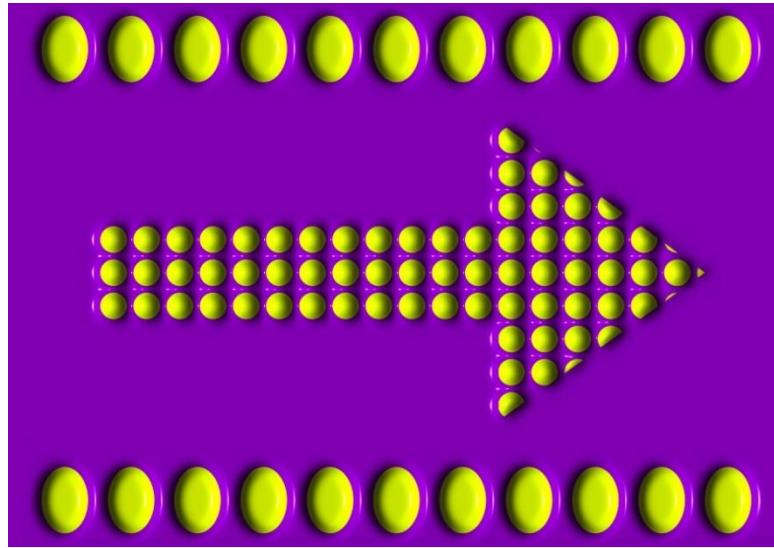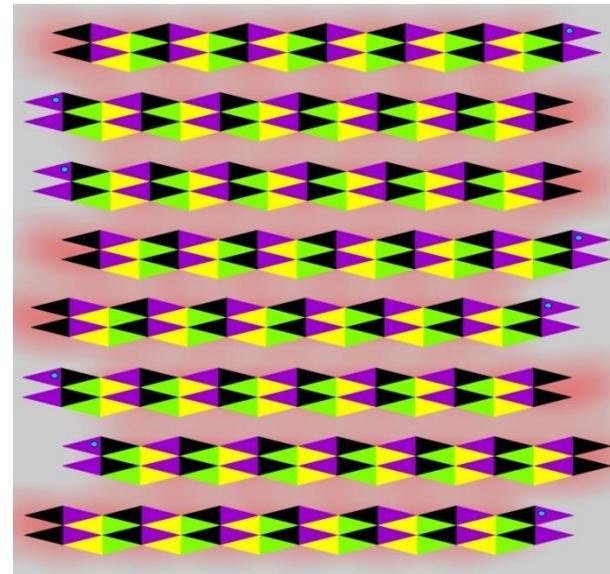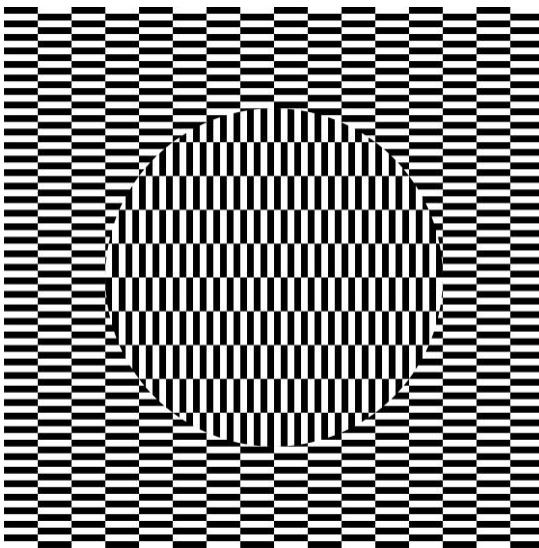
# How is this possible?

- The true mechanism is yet 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
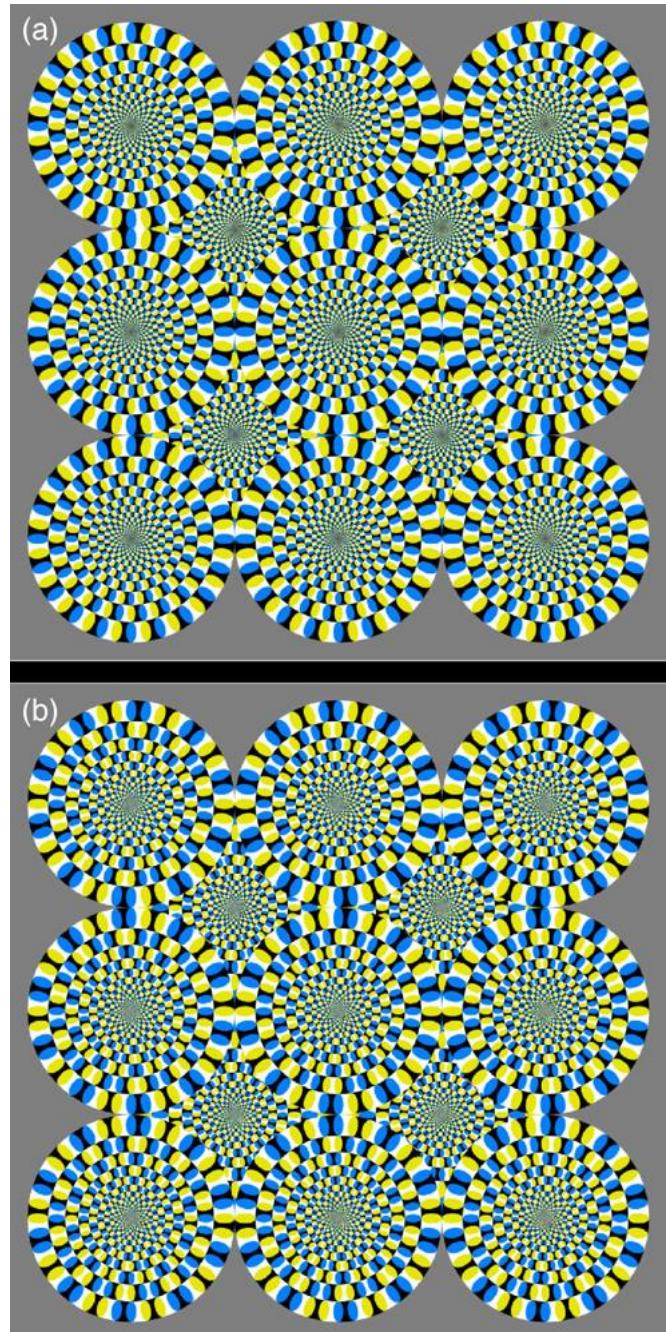
# The cause of motion

- Three factors in imaging process
  - Light
  - Object
  - Camera
- Varying either of them causes motion
  - Static camera, moving objects (surveillance)
  - Moving camera, static scene (3D capture)
  - Moving camera, moving scene (sports, movie)
  - Static camera, moving objects, moving light (time lapse)

# Motion scenarios (priors)



Static camera, moving scene

Moving camera, static scene

Moving camera, moving scene

Static camera, moving scene, moving light

# We still don't touch these areas

# How can we recover motion?

# Recovering motion

- ## Feature-tracking
  - Extract visual features (corners, textured areas) and "track" them over multiple frames

- ## Optical flow
  - Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)

Two problems, one registration method

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.

# Feature tracking

- Challenges
  - Figure out which features can be tracked
  - Efficiently track across frames
  - Some points may change appearance over time (e.g., due to rotation, moving into shadows, etc.)
  - Drift: small errors can accumulate as appearance model is updated
  - Points may appear or disappear: need to be able to add/delete tracked points

# What is Optical Flow?

# What is Optical Flow?

**Movement**

# What is Optical Flow?
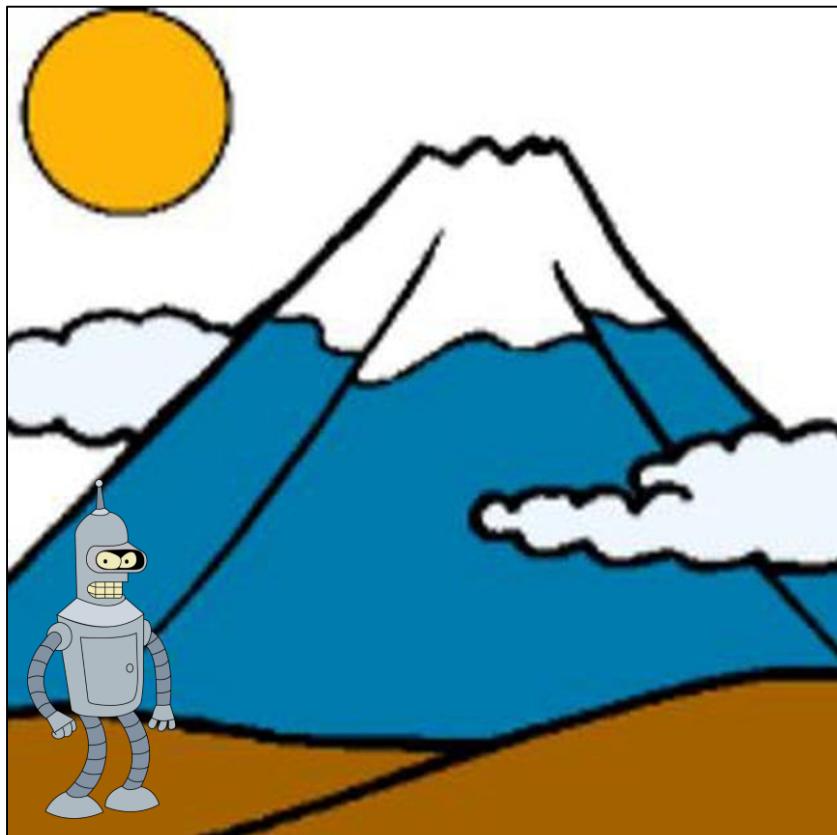
**Movement**

# What is Optical Flow?
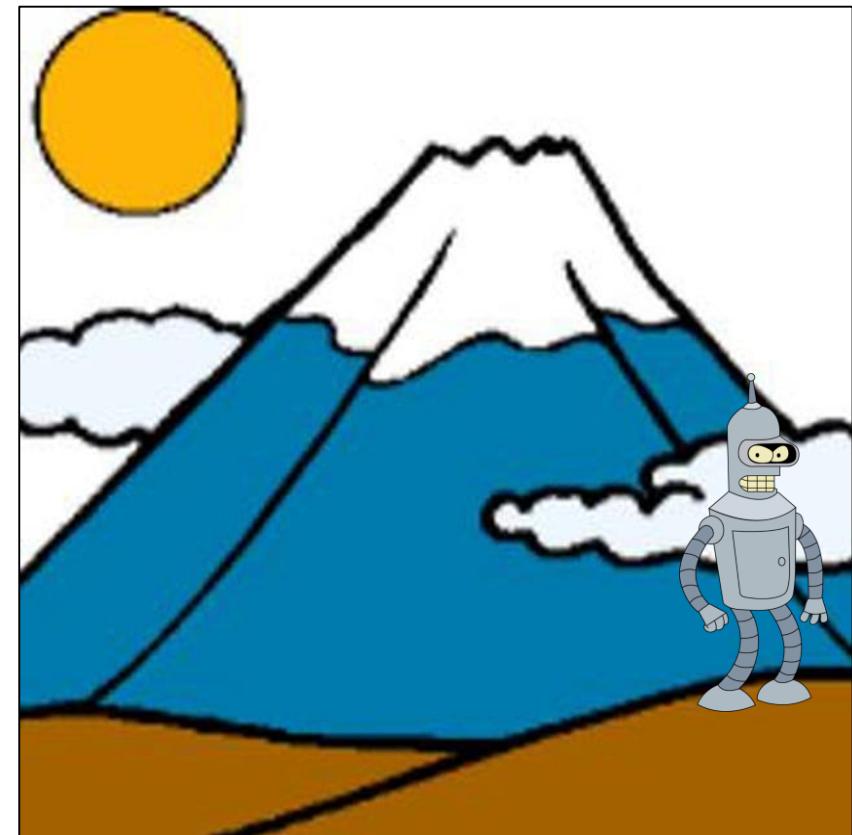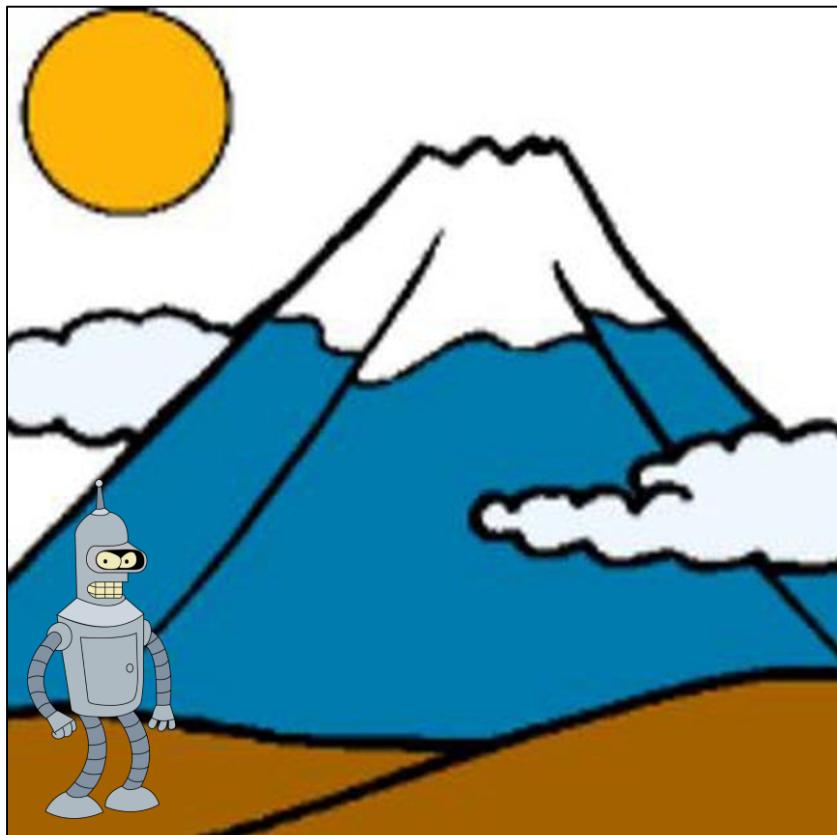
**Movement**

# What is Optical Flow?
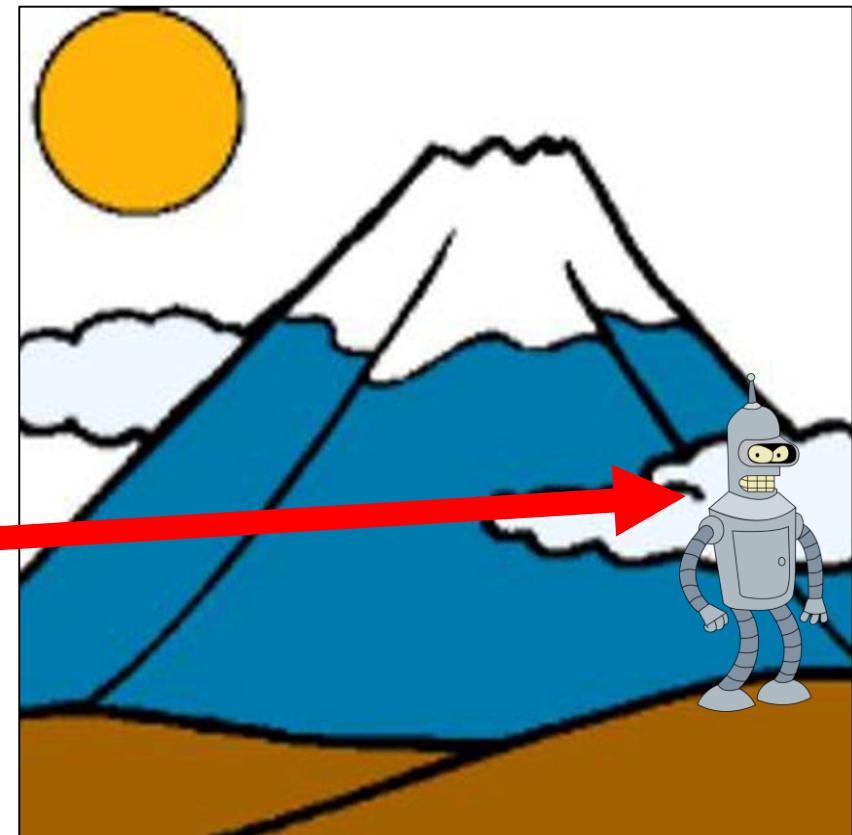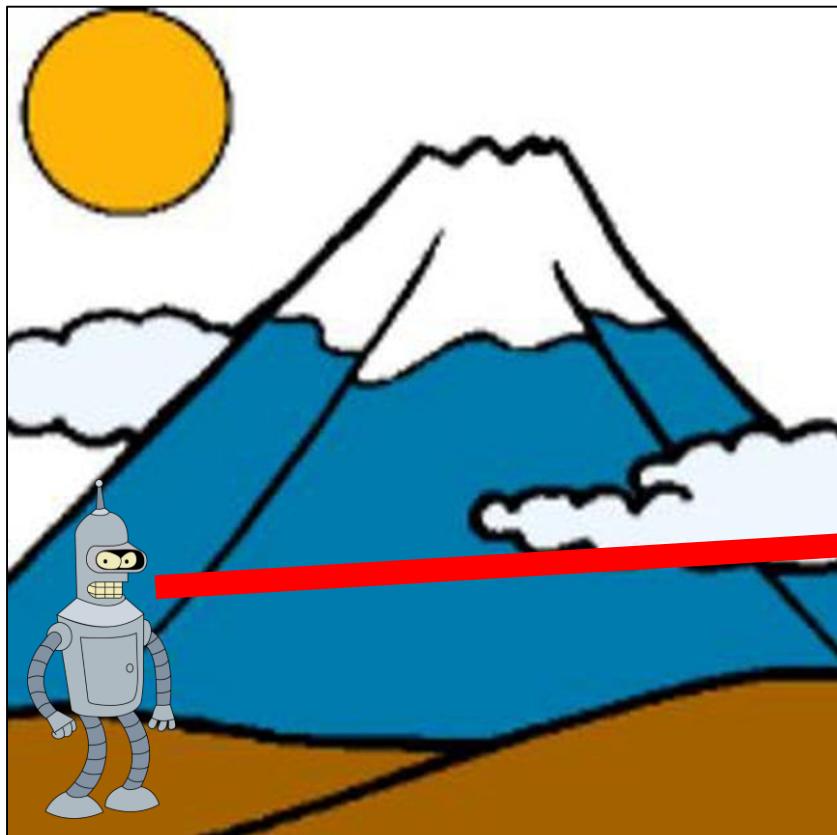
**Movement**

# What is Optical Flow?

**Movement**

**Object**
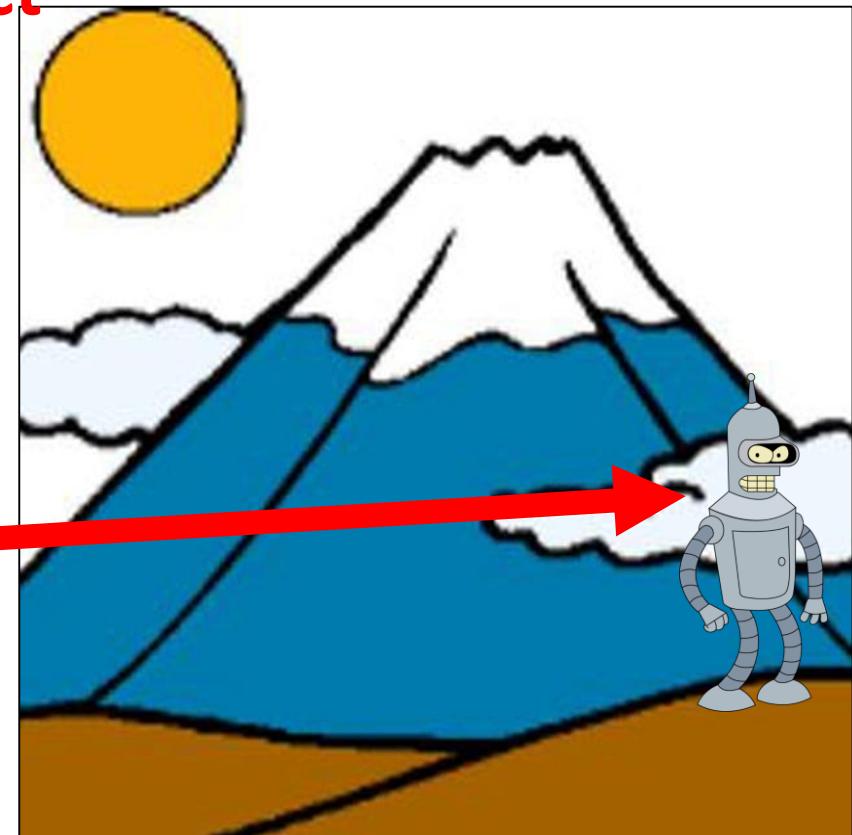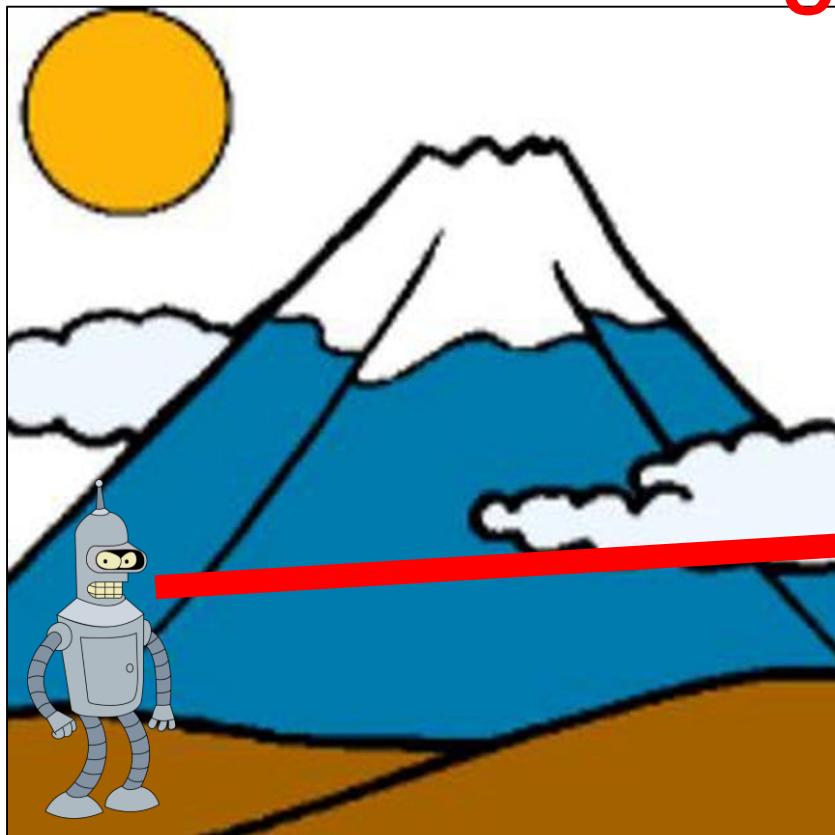
# What is Optical Flow?

**Pan**
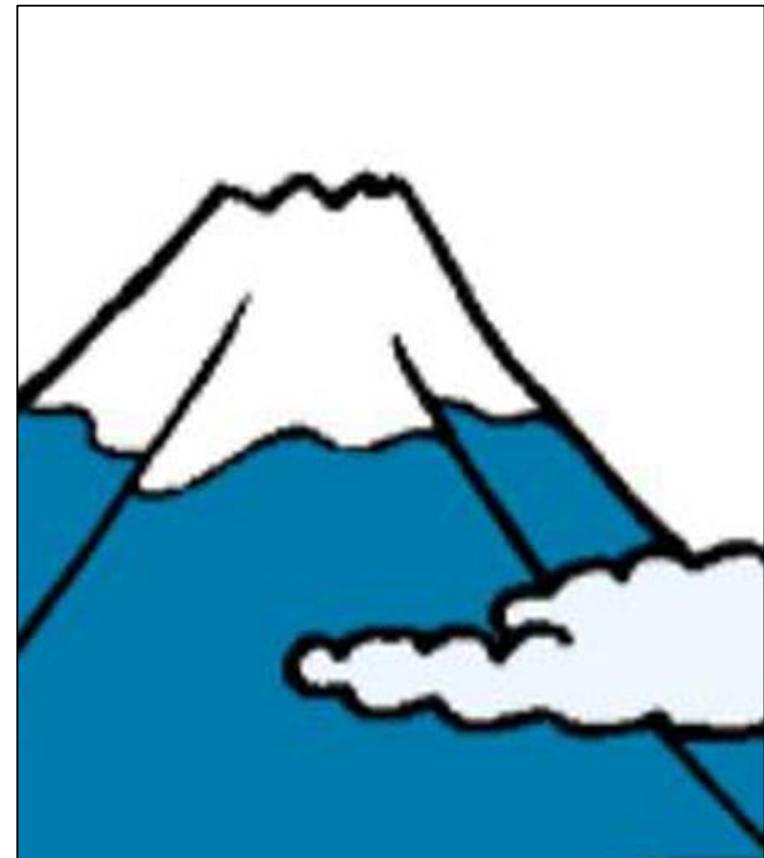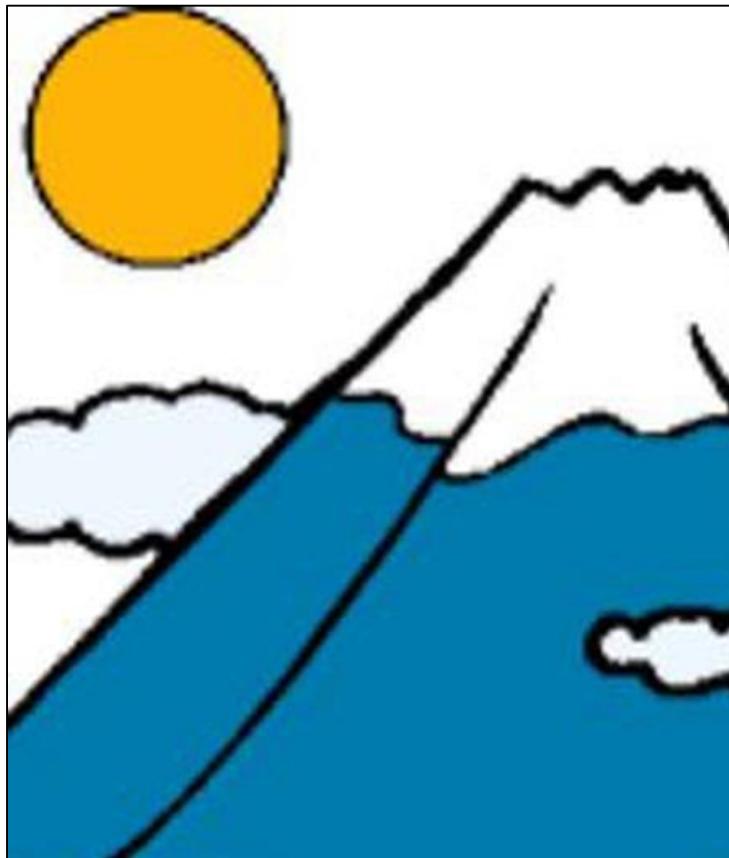
**Movement**

# What is Optical Flow?

**Forward** **Movement**
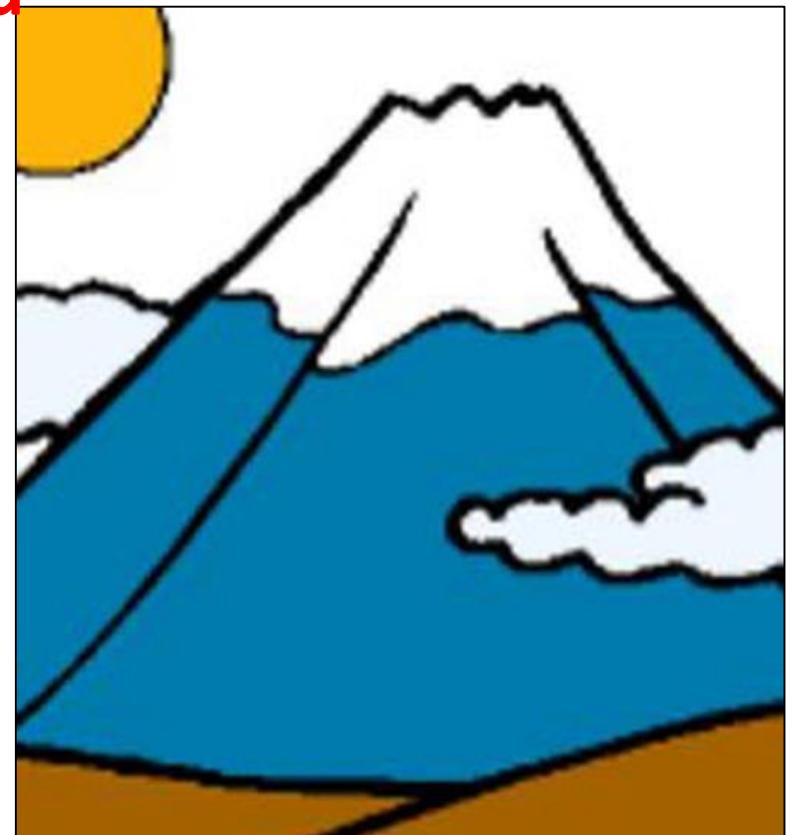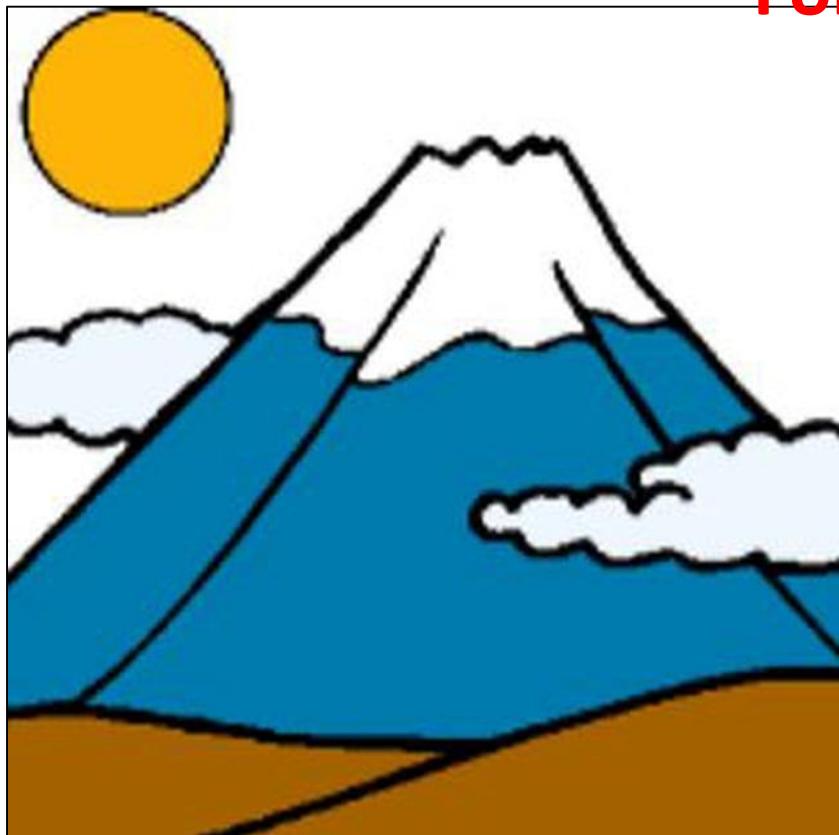
# What is Optical Flow?

**Movement**

# Why do we want Optical Flow?

# Why do we want Optical Flow?
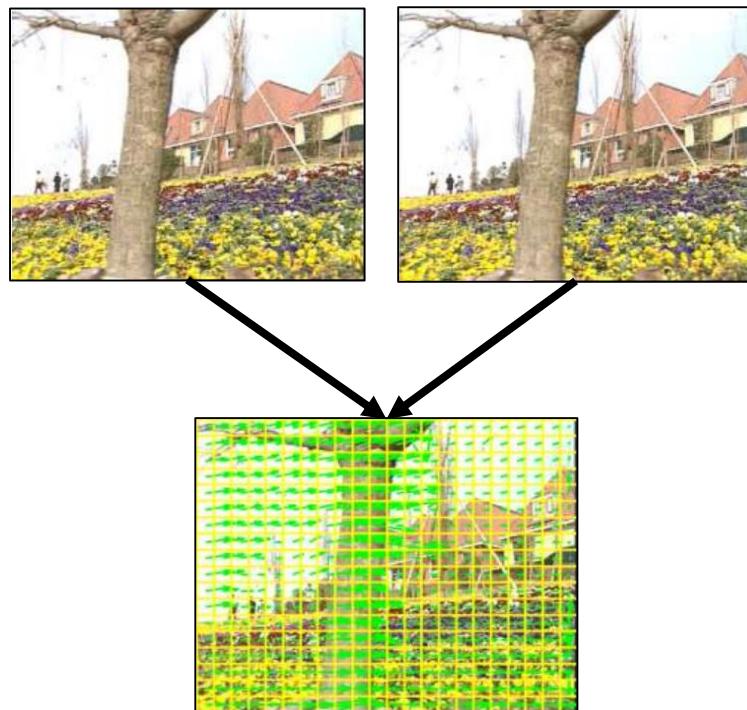
**Motion Estimation**

# Why do we want Optical Flow?
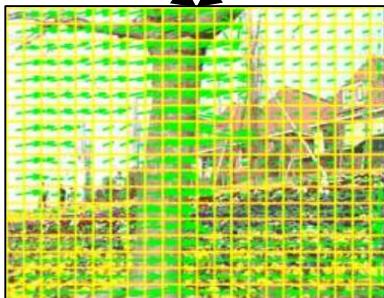
**Motion Estimation**



**Object Tracking**

# Why do we want Optical Flow?

**Motion Estimation**

**Object Tracking**
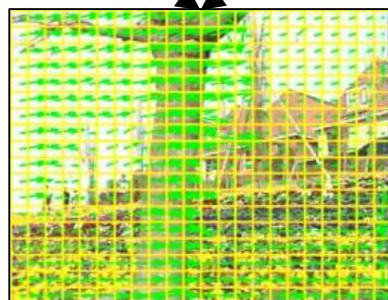
**Visual Odometry**
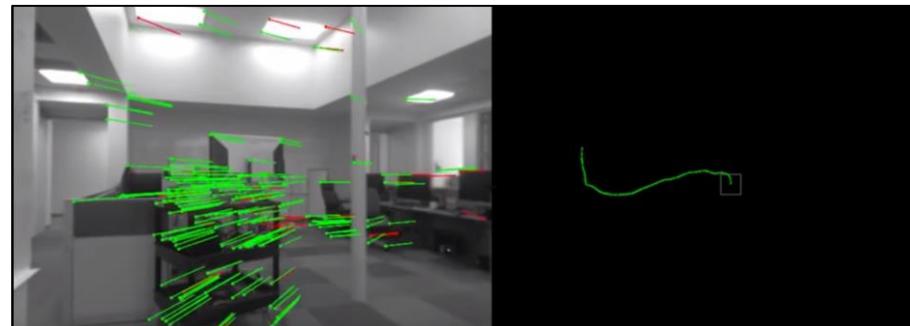
Estimating the position of a robot.

# How do we find the flow in an image?

# Feature Matching

# Previously: Features!

- Highly descriptive local regions
- Ways to describe those regions
- Useful for:
    - Matching
    - Recognition
    - Detection
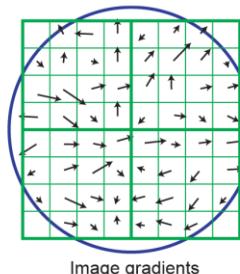
Image gradients

Keypoint descriptor
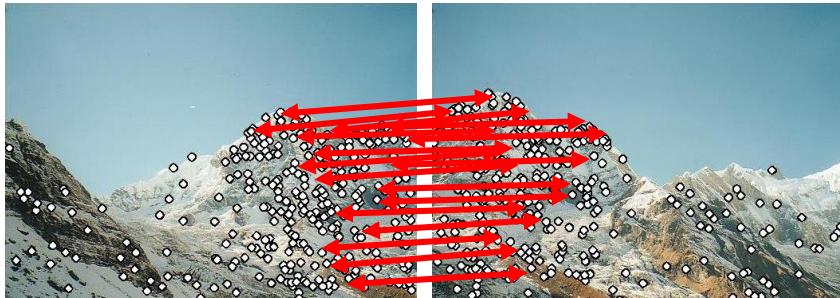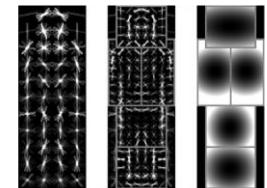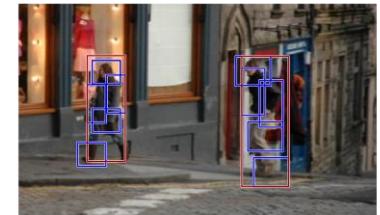
# Feature Matching
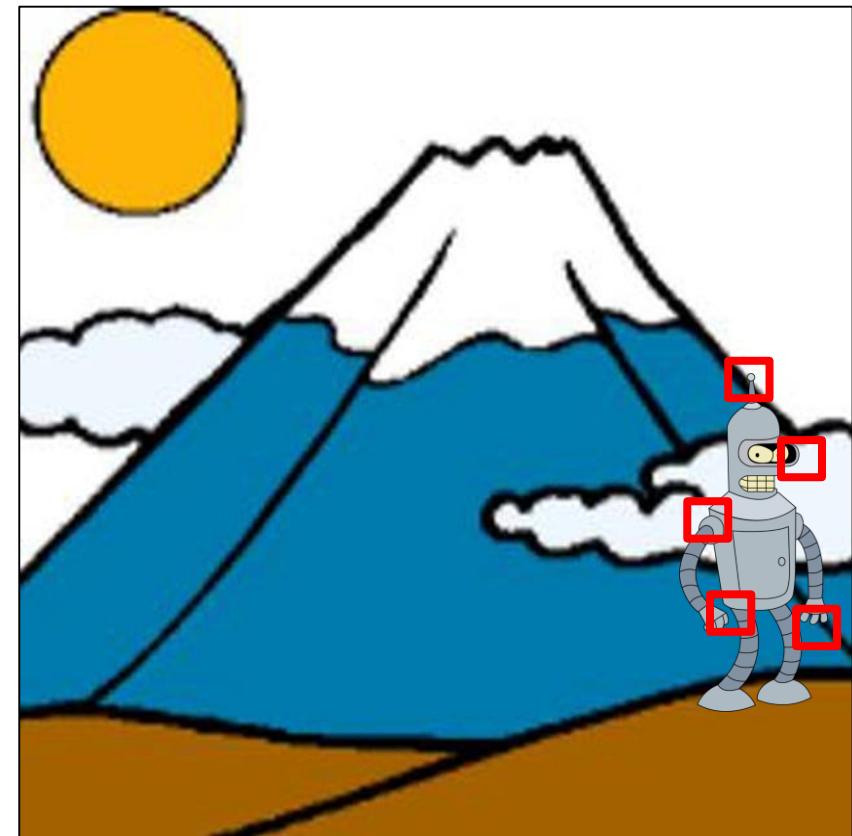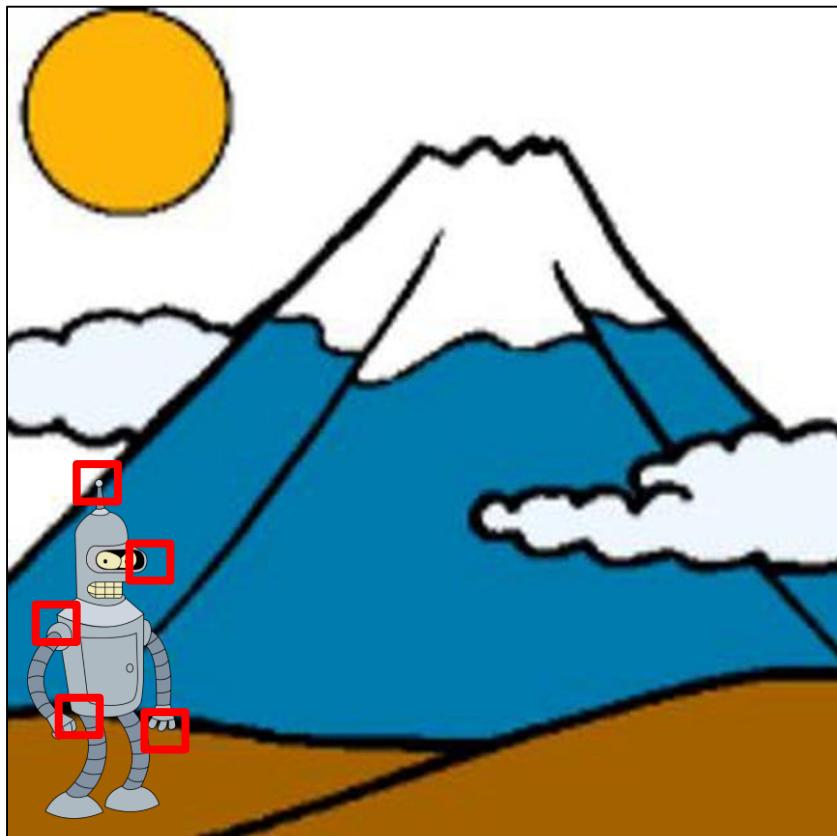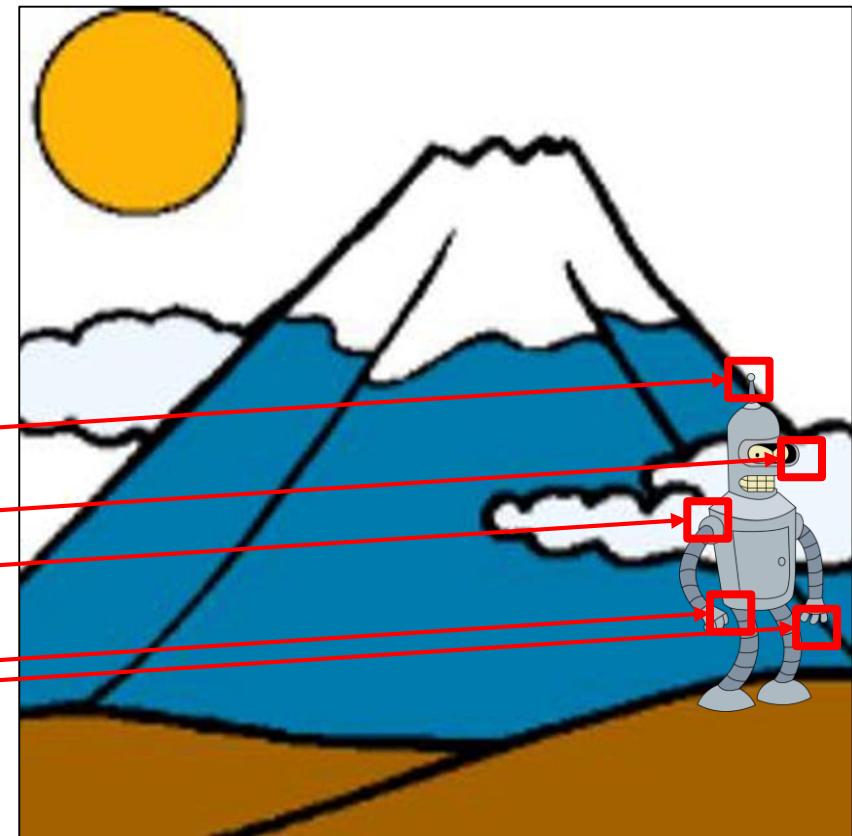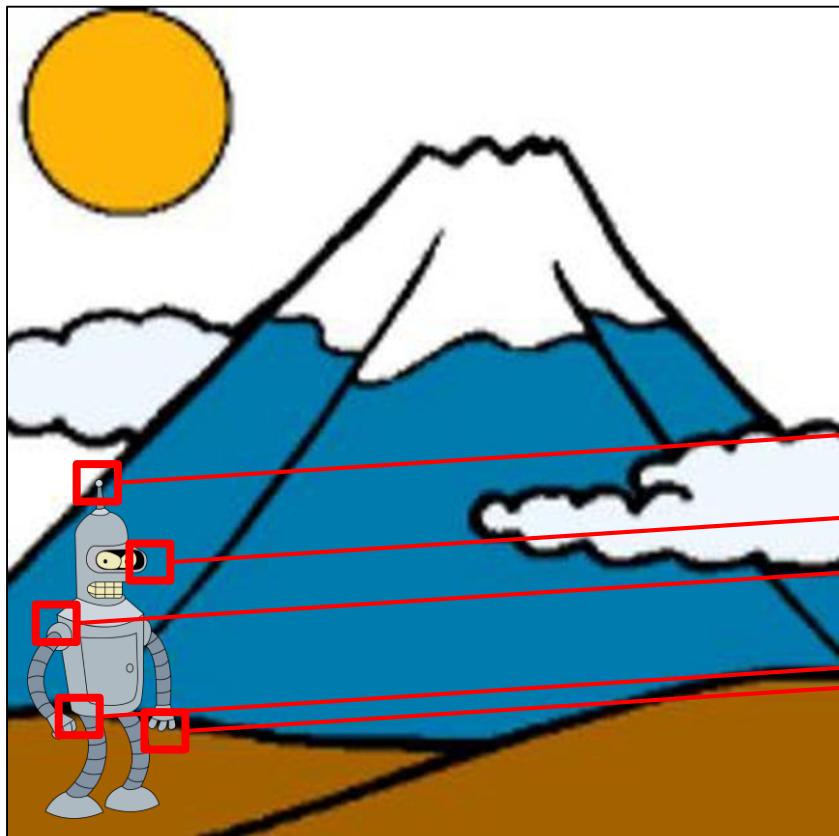
# Feature Matching
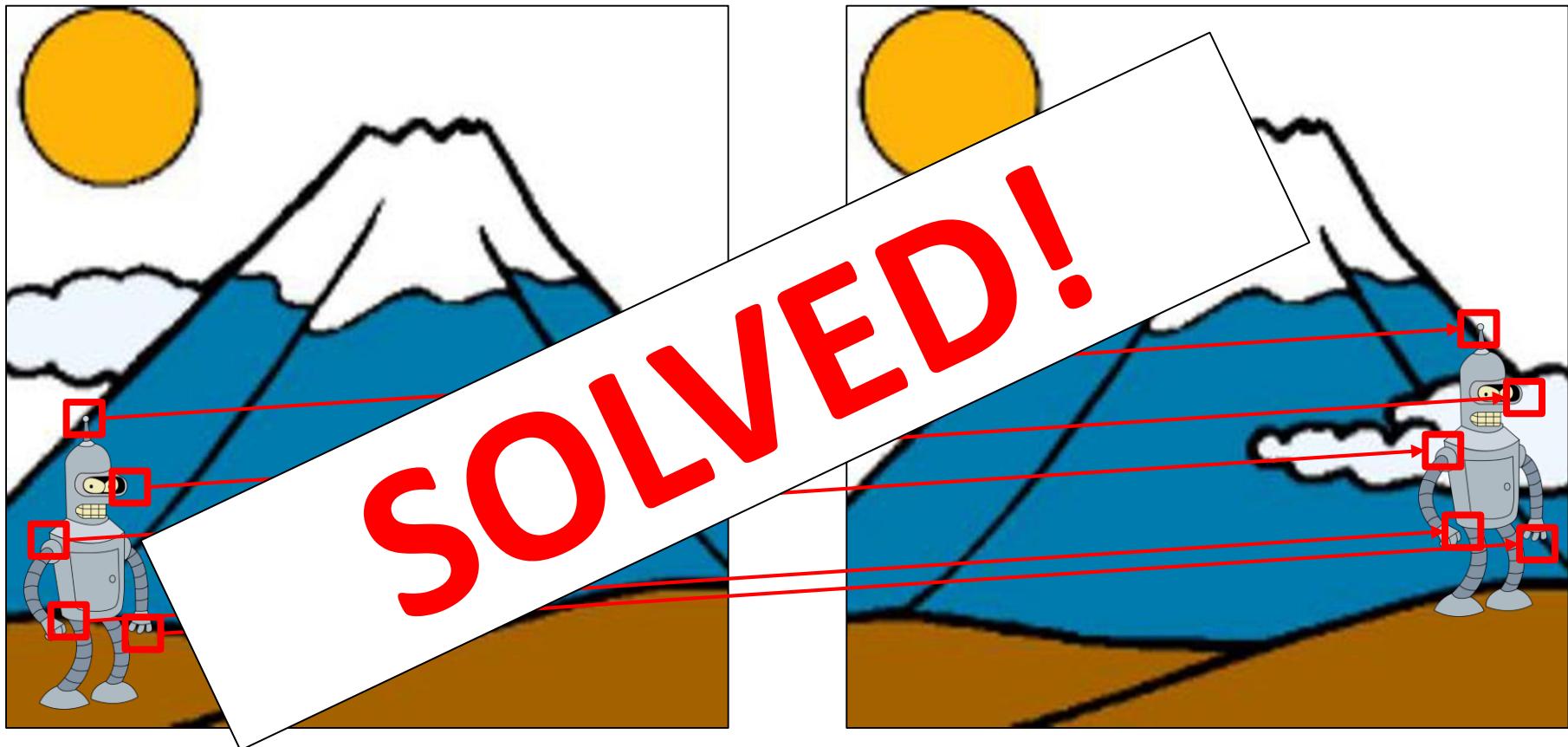
# Feature Matching

# Feature Matching

Disadvantages:

# Feature Matching

Disadvantages:

-Sparse!

# Feature Matching

Disadvantages:

-Sparse!

-Feature alignment not exact

# Feature Matching
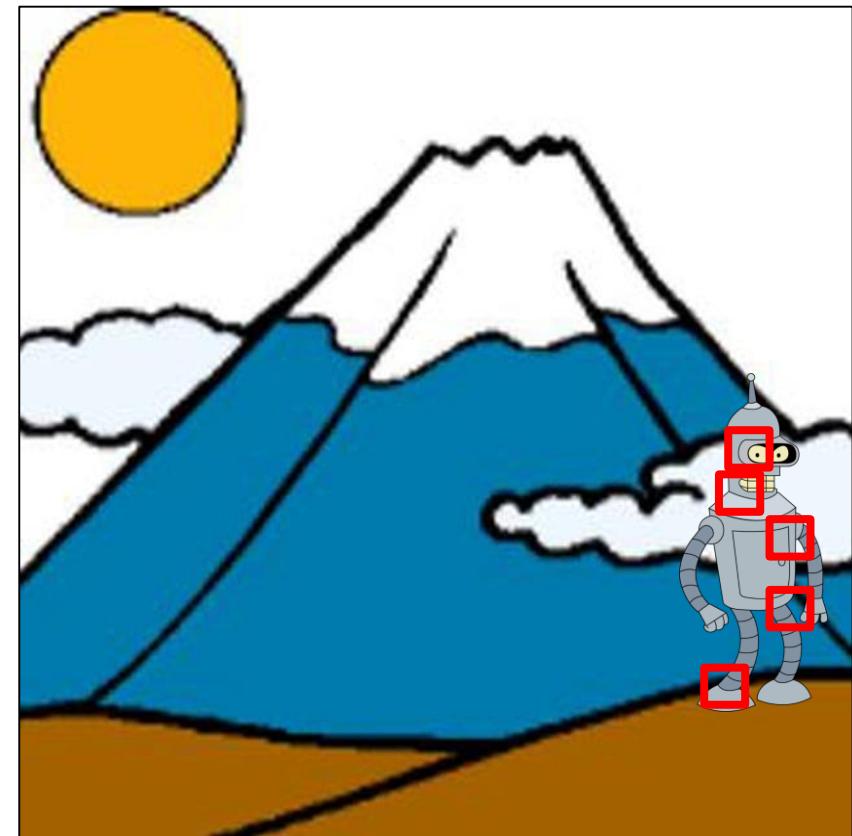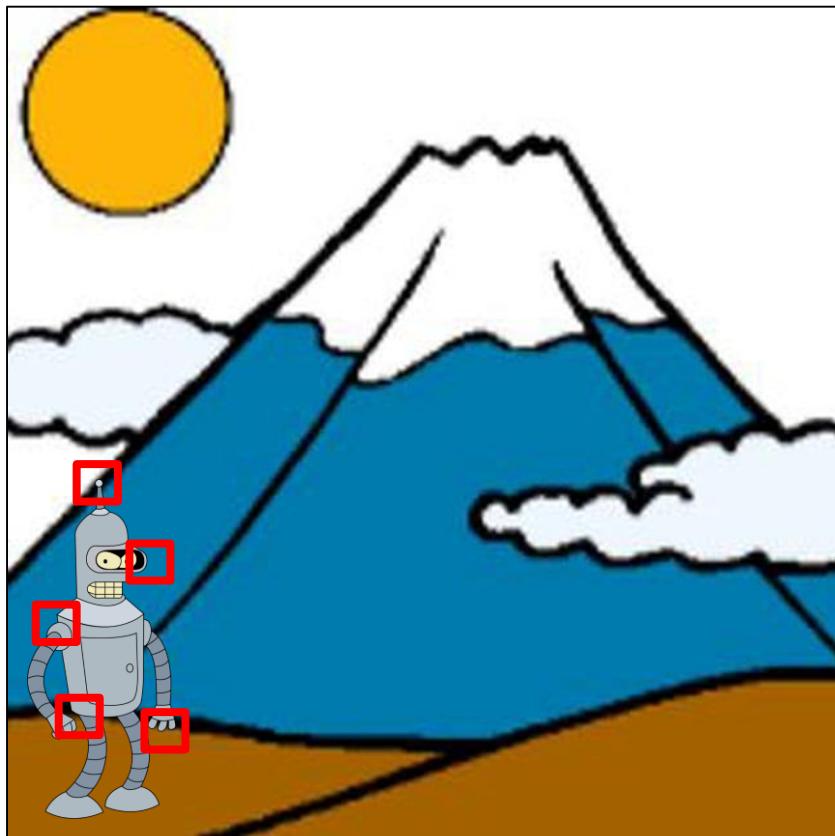
# Feature Matching

Disadvantages:

-Sparse!

-Feature alignment not exact

-Low accuracy

# Feature Matching

Disadvantages:

-Sparse!

-Feature alignment not exact

-Low accuracy

Advantages:

# Feature Matching

Disadvantages:

-Sparse!

-Feature alignment not exact

-Low accuracy

Advantages:

-Scale/rotation invariant

-*kinda* lighting invariant

-Can handle large movements

# Feature Matching

Disadvantages:

-Sparse!

-Feature alignment not exact

-Low accuracy

Advantages:

-Scale/rotation invariant

-*kinda* lighting invariant

-e movements

**Overall: Doesn't work very well for Optical Flow**

# What do we do instead?

# Feature tracking
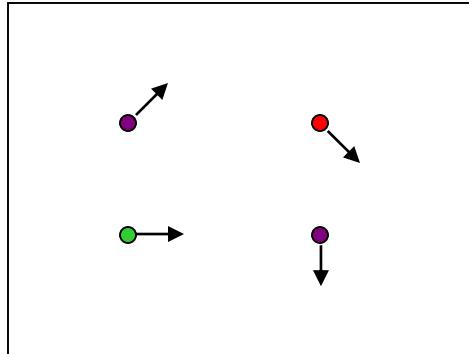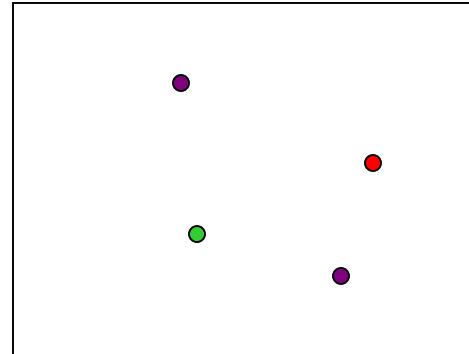


$I(x,y,t)$          $I(x,y,t+1)$

- Given two subsequent frames, estimate the point translation

- Key assumptions of Lucas-Kanade Tracker
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Small motion:** points do not move very far
  - **Spatial coherence:** points move like their neighbors

# The brightness constancy constraint
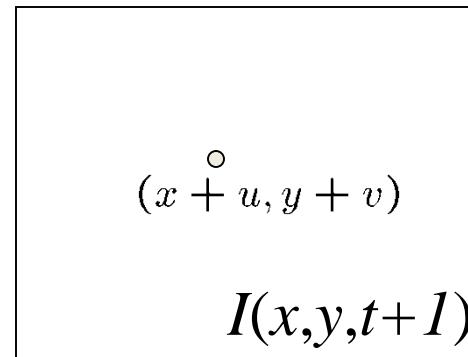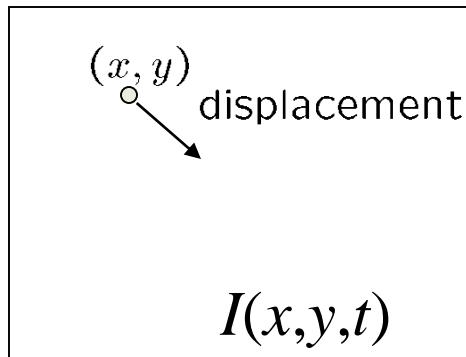


- ## Brightness Constancy Equation:

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

Take Taylor expansion of $I(x+u, y+v, t+1)$ at $(x,y,t)$ to linearize the right side:

Image derivative along x    Difference over frames

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

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

- Difference in intensity at the same pixel between one image and the previous one.

# The brightness constancy constraint

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

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

So: $\quad I_x \cdot u + I_y \cdot v + I_t \approx 0$

$$\rightarrow \nabla I \cdot \begin{bmatrix} u & v \end{bmatrix}^{\mathrm{T}} + I_t = 0$$

# The brightness constancy constraint

Can we use this equation to recover image motion (u,v) at each pixel?

$$\nabla I \cdot [\mathrm{u} \ \ \mathrm{v}]^{\mathrm{T}} + I_t = 0$$

- How many equations and unknowns per pixel?

  - One equation (this is a scalar equation!), two unknowns (u,v)

The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

# Solving the ambiguity…

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

- How to get more equations for a pixel?

- **Spatial coherence constraint**

- Assume the pixel's neighbors have the same (u,v)

  – If we use a 5x5 window, that gives us 25 equations per pixel

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

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

# Solving the ambiguity…

- Least squares problem:

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

$$A \quad d = b$$

25x2  2x1  25x1

# Matching patches across images

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

$A \quad d = b$

25x2  2x1  25x1

Least squares solution for *d* given by

$$(A^T A) \; d = A^T b$$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$A^T A \qquad\qquad\qquad A^T b$

The summations are over all pixels in the K x K window

$d = (A^T A)^{-1} A^T b$

# Conditions for solvability

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

$$\left[ \begin{array}{cc} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{array} \right] \left[ \begin{array}{c} u \\ v \end{array} \right] = - \left[ \begin{array}{c} \sum I_x I_t \\ \sum I_y I_t \end{array} \right]$$

$$A^T A \qquad\qquad\qquad A^T b$$

When is this solvable?  I.e., what are good points to track?

- **A$^T$A** should be invertible
- **A$^T$A** should not be too small due to noise
  - eigenvalues $\lambda_1$ and $\lambda_2$ of **A$^T$A** should not be too small
- **A$^T$A** should be well-conditioned
  - $\lambda_1 / \lambda_2$ should not be too large ($\lambda_1$ = larger eigenvalue)

Does this remind you of anything?

## Criteria for Harris corner detector

# Aperture problem
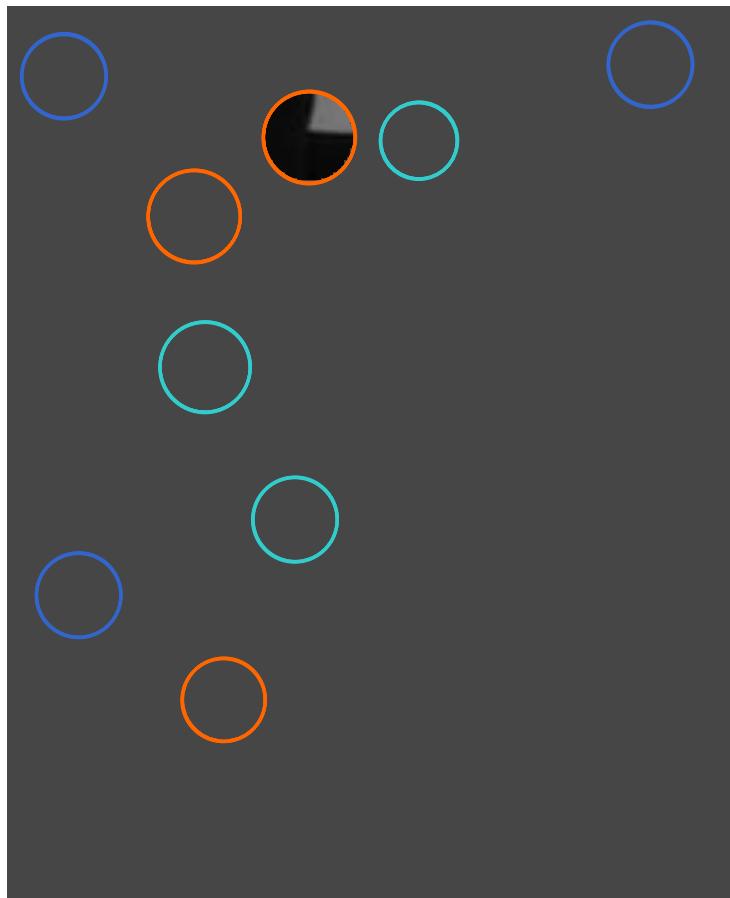


Corners | Lines | Flat regions

# Edge
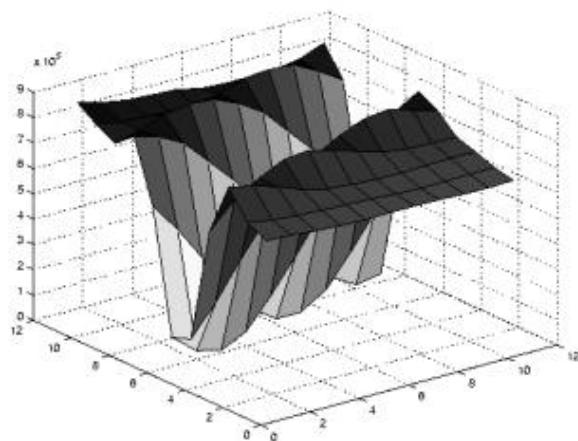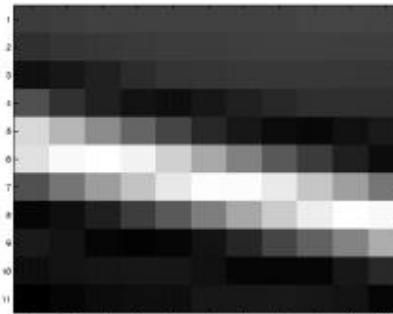


$$\sum \nabla I (\nabla I)^T$$

  – large gradients, all the same
  – large $\lambda_1$, small $\lambda_2$

# Low Texture Region
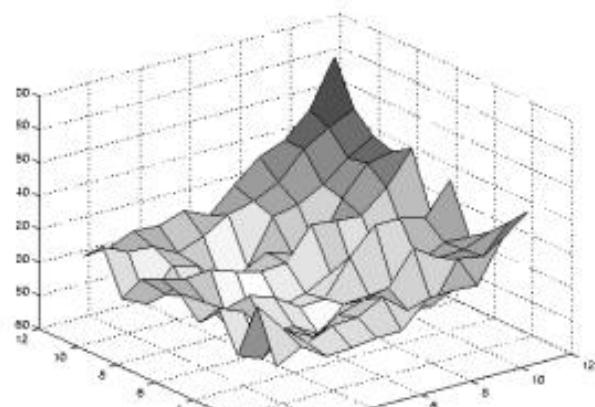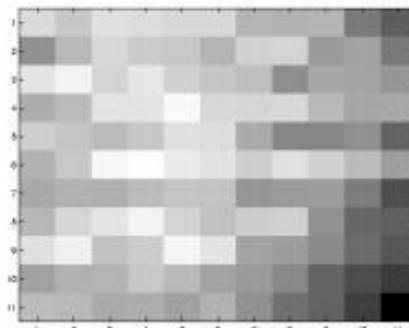


$$\sum \nabla I (\nabla I)^T$$

– gradients have small magnitude

– small $\lambda_1$, small $\lambda_2$

# High Texture Region
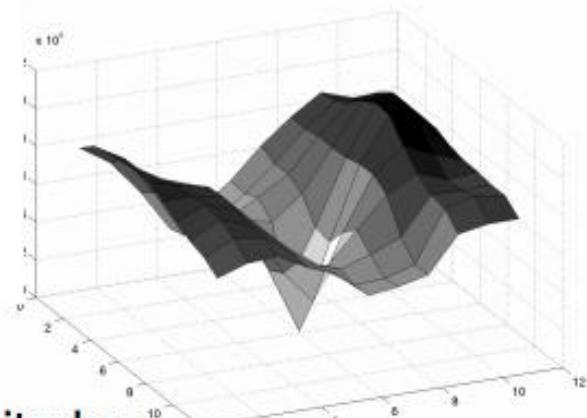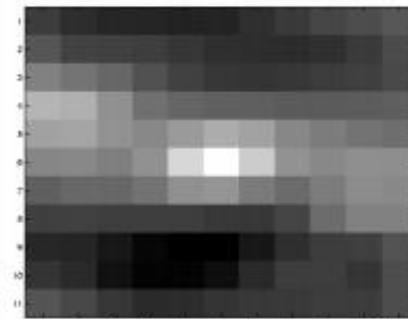


$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

– large $\lambda_1$, large $\lambda_2$

# Errors in Lukas-Kanade

- What are the potential causes of errors in this procedure?
  - Suppose $A^TA$ is easily invertible
  - Suppose there is not much noise in the image

When our assumptions are violated

- Brightness constancy is **not** satisfied

- The motion is **not** small

- A point does **not** move like its neighbors
  - window size is too large
  - what is the ideal window size?

# Revisiting the small motion assumption



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

# Reduce the resolution!

# Coarse-to-fine optical flow estimation



run iterative L-K

warp & upsample

run iterative L-K

**Gaussian pyramid of image 1 (t)**

**Gaussian pyramid of image 2 (t+1)**

image 1

image 2
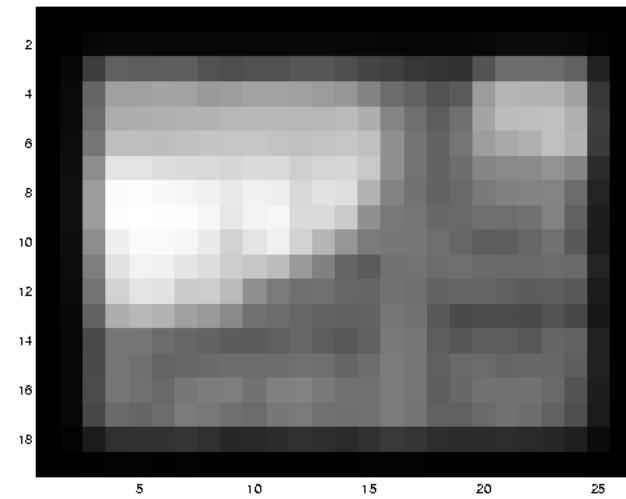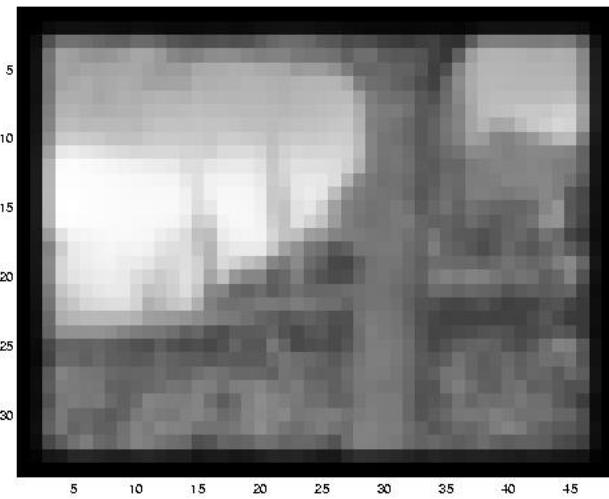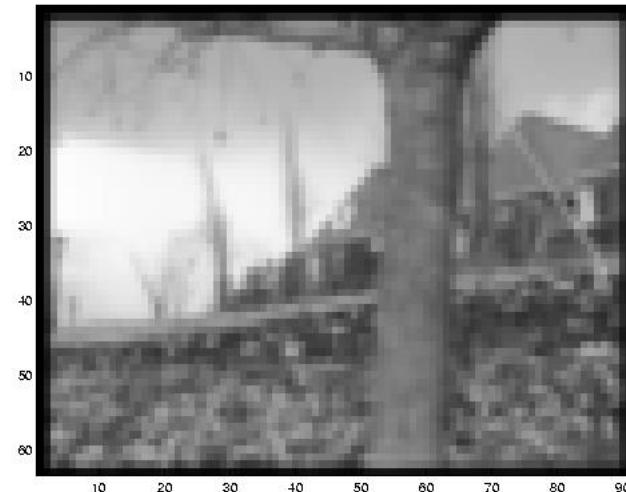
# A Few Details

- **Top Level**
  - Apply L-K to get a flow field representing the flow from the first frame to the second frame.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K on the new warped image to get a flow field from it to the second frame.
  - Repeat till convergence.
- **Next Level**
  - Upsample the flow field to the next level as the first guess of the flow at that level.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K and warping till convergence as above.
- **Etc.**

# Coarse-to-fine optical flow estimation
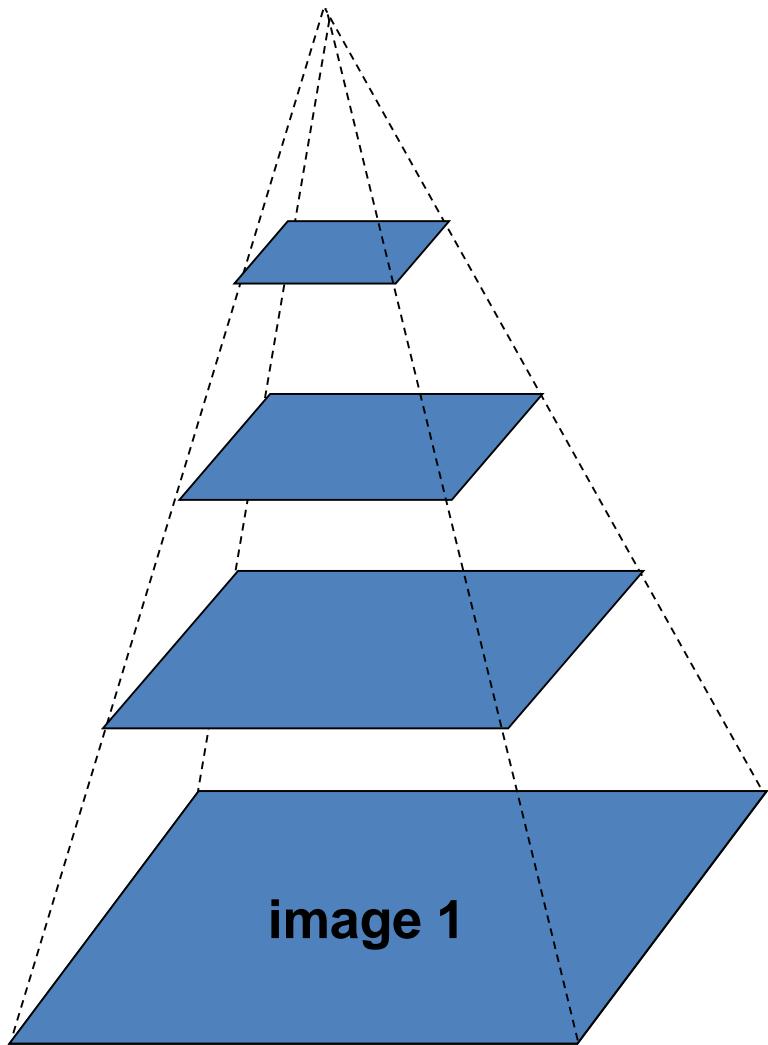


*u=1.25 pixels*

*u=2.5 pixels*

*u=5 pixels*

*u=10 pixels*

**image 1**

**image 2**

**Gaussian pyramid of image 1**
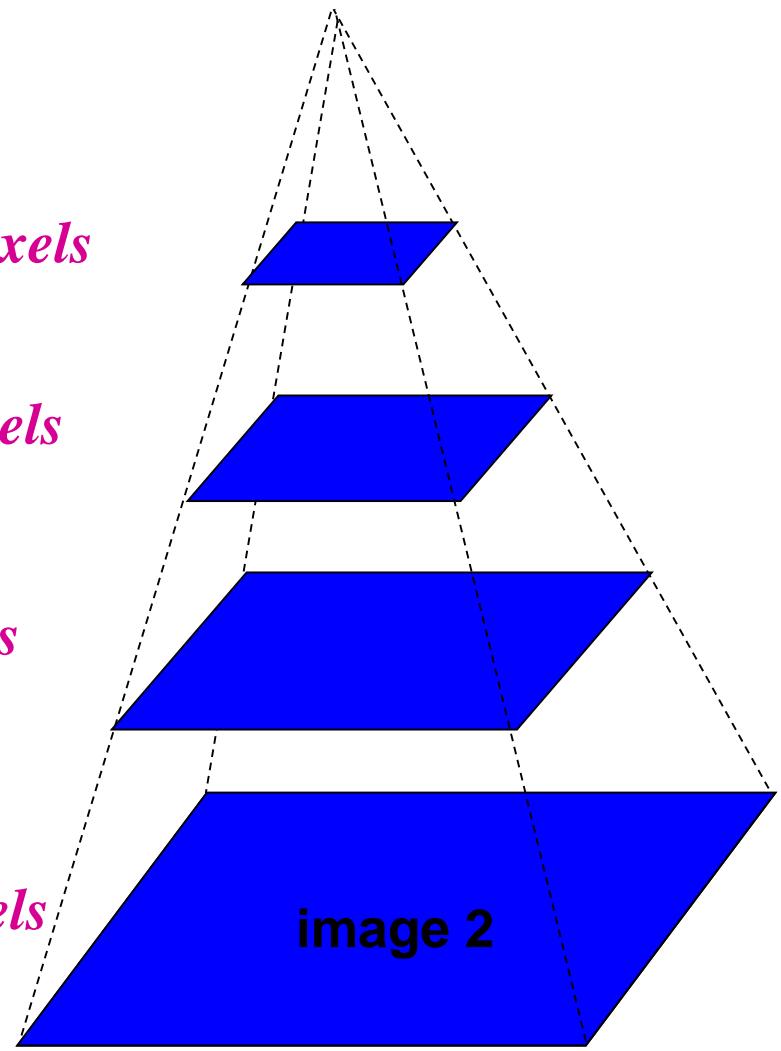
**Gaussian pyramid of image 2**

# The Flower Garden Video

What should the optical flow be?
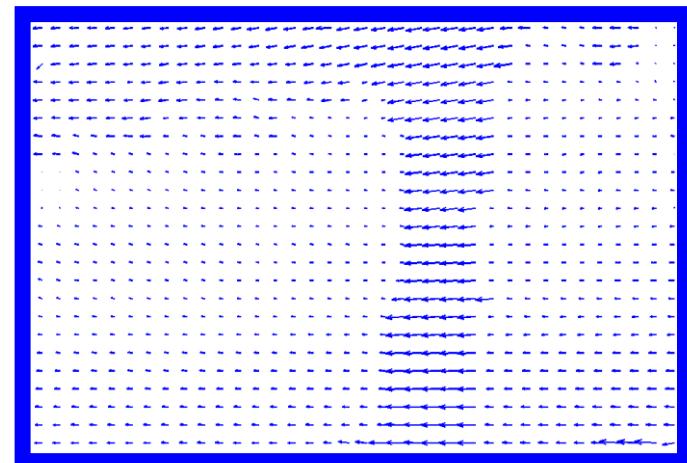
# Optical Flow Results
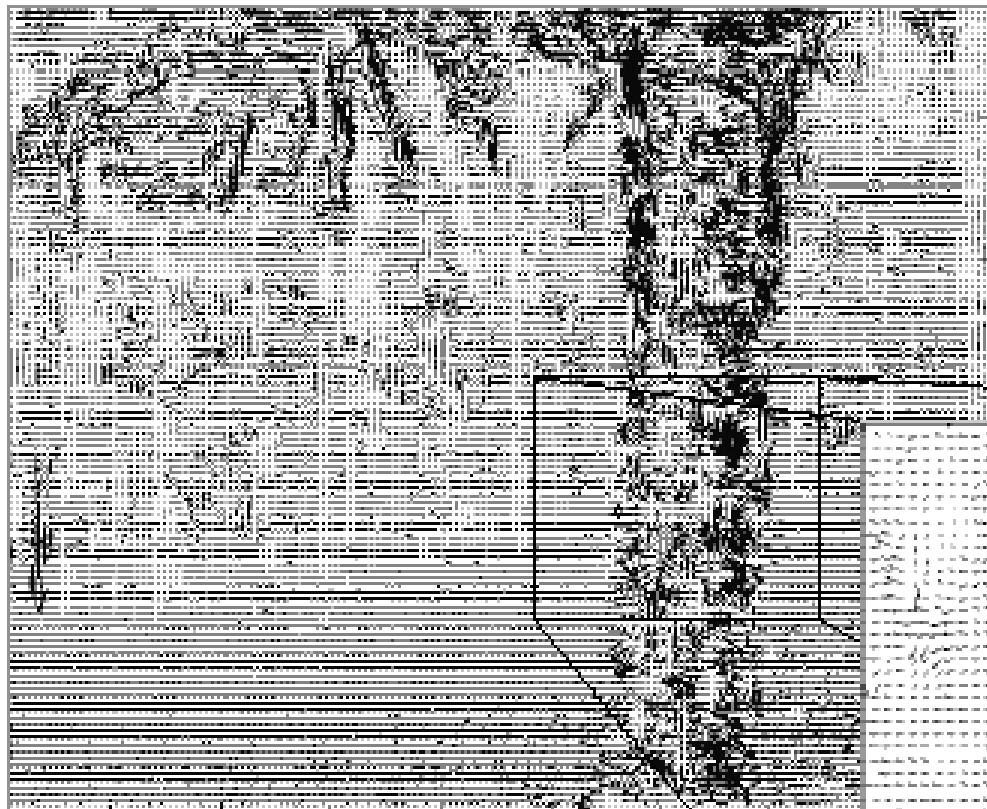
Lucas-Kanade
without pyramids

Fails in areas of large
motion
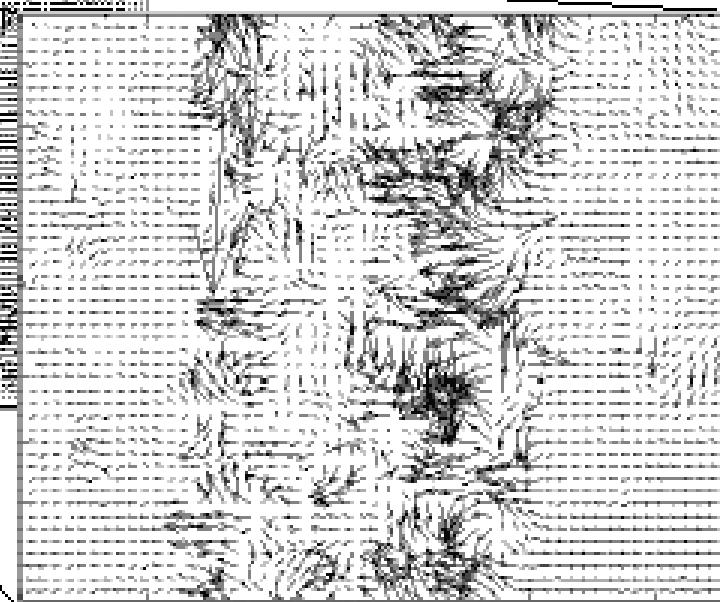
# Optical Flow Results
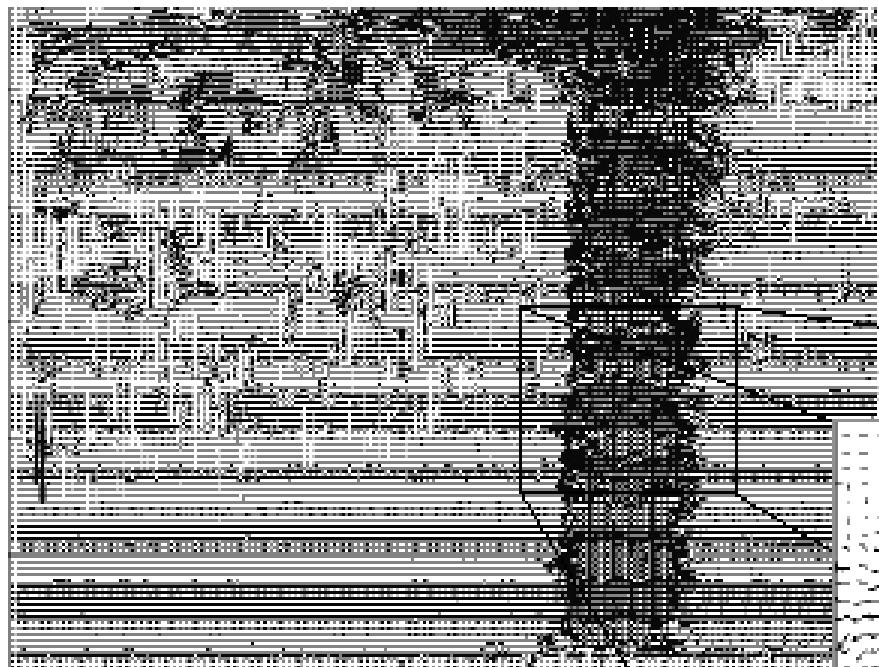


Lucas-Kanade with Pyramids
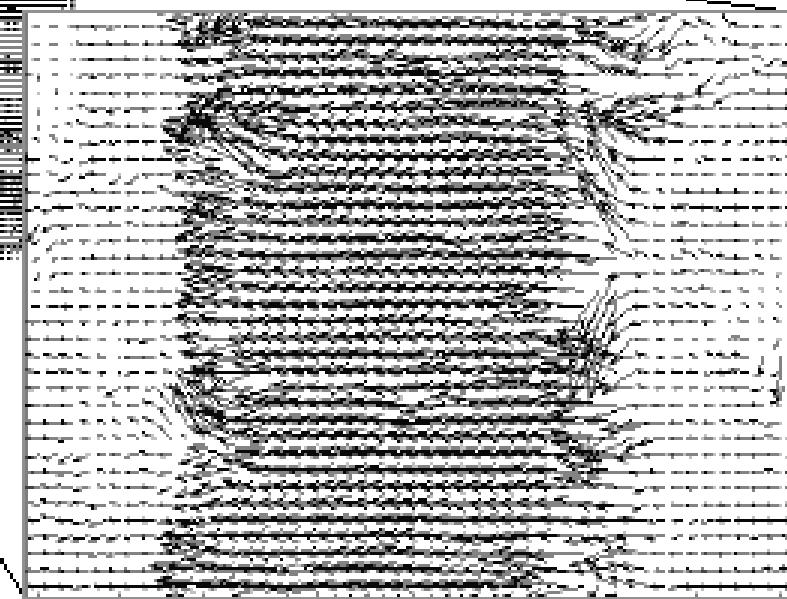
# Flow quality evaluation

# Flow quality evaluation

# Flow quality evaluation

- Middlebury flow page
  - http://vision.middlebury.edu/flow/





Ground Truth
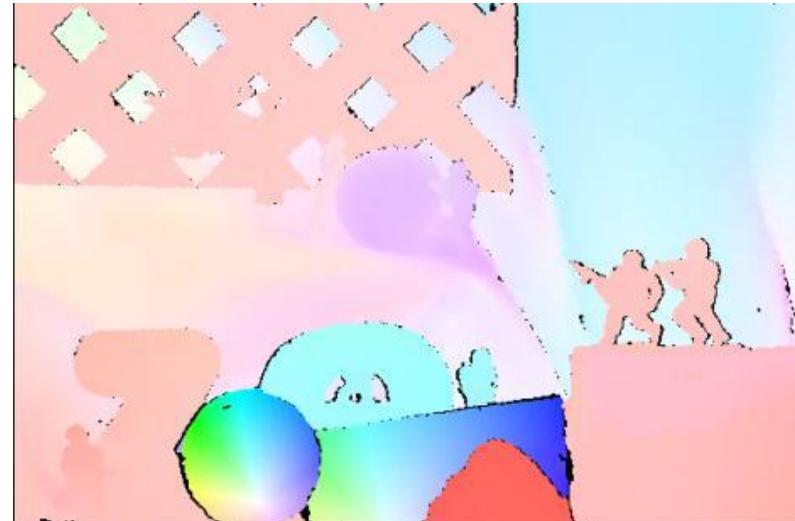
# Flow quality evaluation
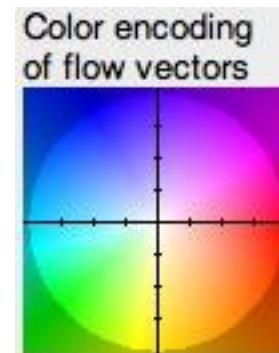
- Middlebury flow page
  - http://vision.middlebury.edu/flow/
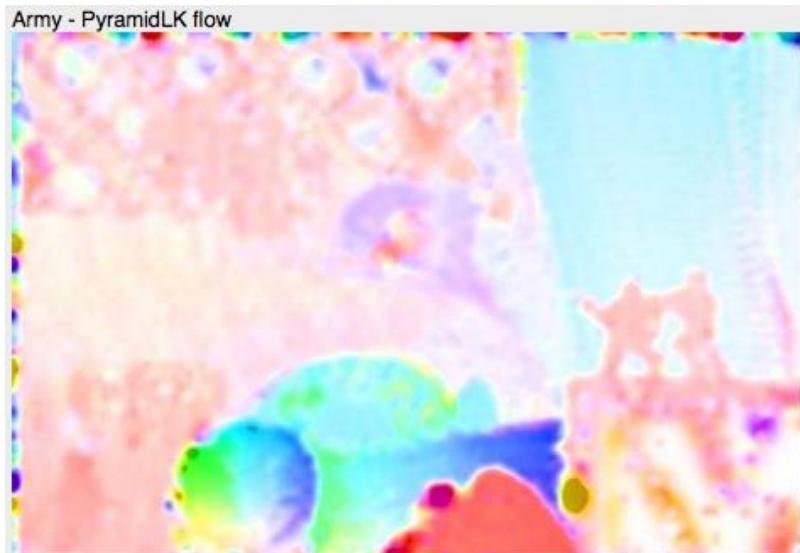


Lucas-Kanade flow



Ground Truth
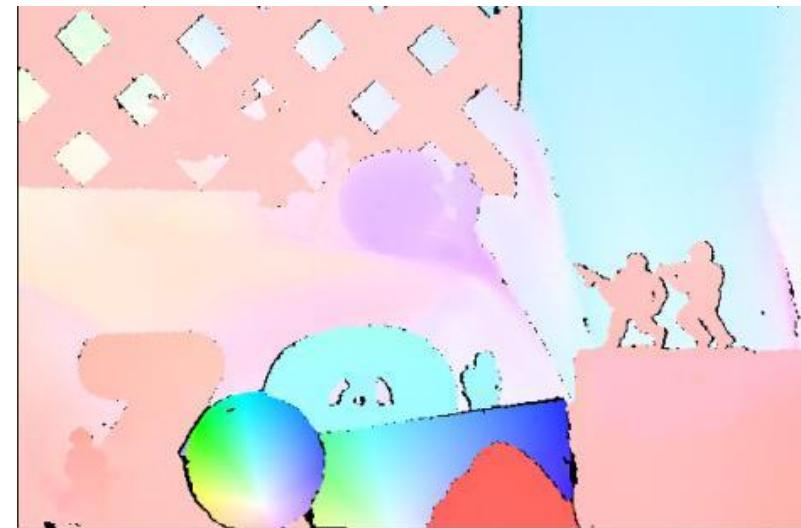
# Flow quality evaluation
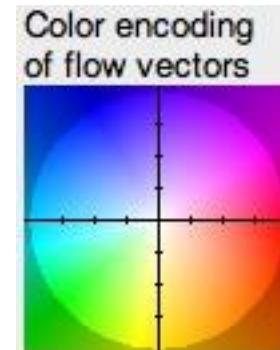
- Middlebury flow page
  - http://vision.middlebury.edu/flow/



Best-in-class alg
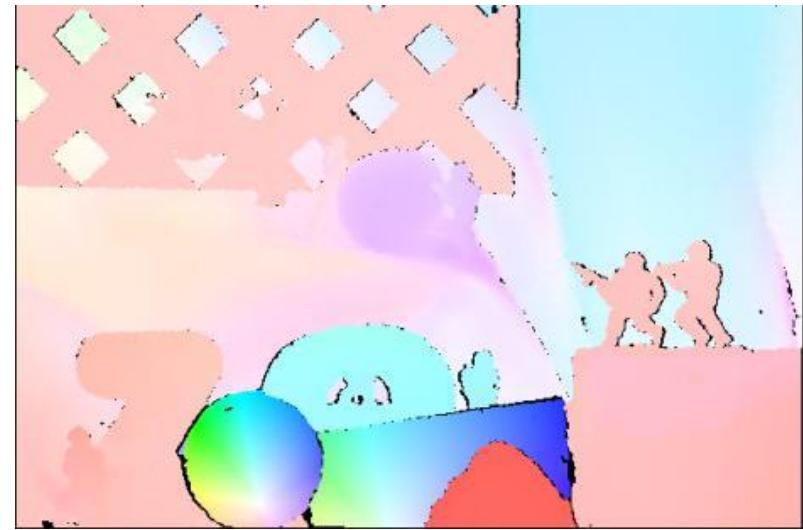


Ground Truth
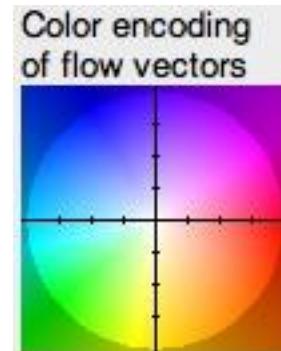
# Video stabilization

# Video denoising

# Video super resolution


Low-Res

# Robust Visual Motion Analysis:
## Piecewise-Smooth Optical Flow

## Ming Ye
## Electrical Engineering
## University of Washington

# Estimating Piecewise-Smooth Optical Flow
# with Global Matching and Graduated Optimization

**Problem Statement:**

**Assuming only brightness conservation and piecewise-smooth motion, find the optical flow to best describe the intensity change in three frames.**

# Approach: Matching-Based Global Optimization

- **Step 1.** **Robust local gradient-based method for high-quality initial flow estimate.**

  **Uses least median of squares instead of regular least squares.**

- **Step 2.** **Global gradient-based method to improve the flow-field coherence.**

  **Minimizes a global energy function $E = \Sigma (E_B(V_i) + E_S(V_i))$ where $E_B$ is the brightness difference and $E_S$ is the smoothness at flow vector $V_i$**

- **Step 3.** **Global matching that minimizes energy by a greedy approach.**

  **Visits each pixel and updates it to be consistent with neighbors, iteratively.**

# TT: Translating Tree



**150x150 (Barron 94)**

|  | $e_{\angle}(^{\circ})$ | $e_{|\bullet|}(\mathrm{pix})$ | $\overline{e}(\mathrm{pix})$ |
|---|---|---|---|
| **BA** | **2.60** | **0.128** | **0.0724** |
| **S3** | **0.248** | **0.0167** | **0.00984** |

**BA**
**S3**
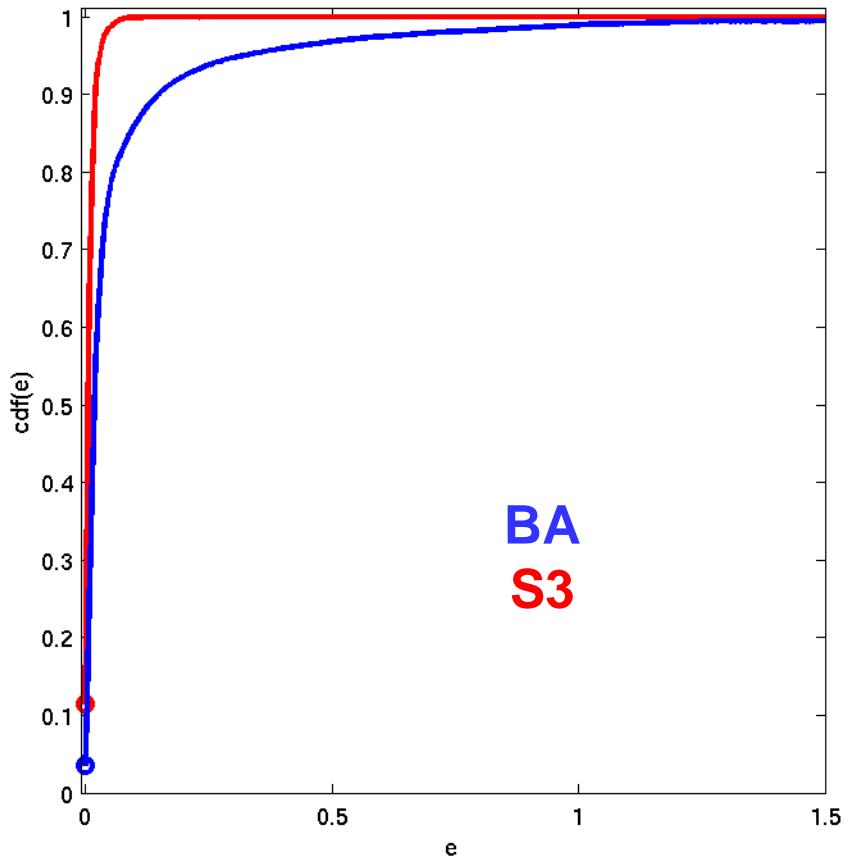
**e: error in pixels, cdf: culmulative distribution function for all pixels**

# DT: Diverging Tree



**150x150 (Barron 94)**

|     | $e_{\angle}(^{\circ})$ | $e_{|\bullet|}(\mathrm{pix})$ | $\overline{e}(\mathrm{pix})$ |
|-----|------|--------|--------|
| **BA** | 6.36 | 0.182 | 0.114 |
| **S3** | 2.60 | 0.0813 | 0.0507 |

**BA**
**S3**
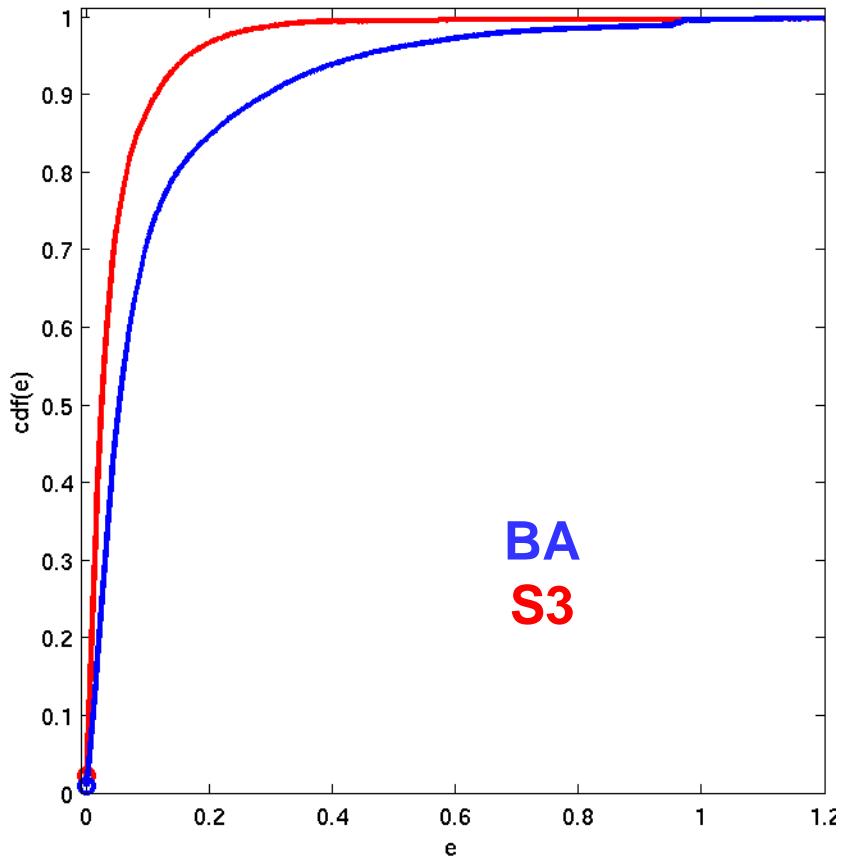
# YOS: Yosemite Fly-Through
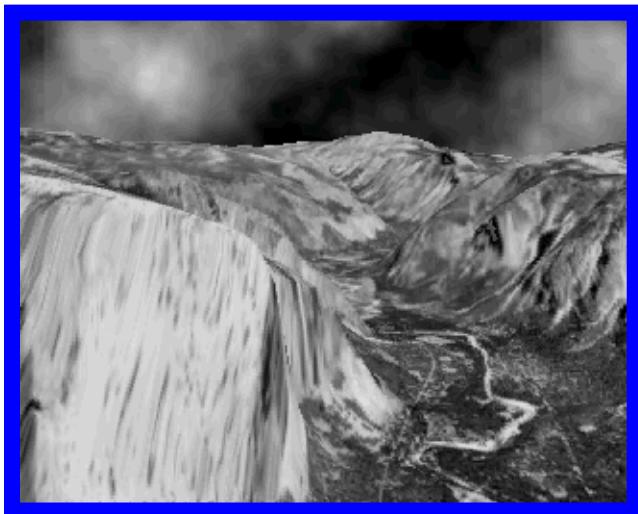


**316x252 (Barron, cloud excluded)**

| | $e_\angle (^\circ)$ | $e_{|\bullet|} (\mathrm{pix})$ | $\overline{e} (\mathrm{pix})$ |
|---|---|---|---|
| **BA** | **2.71** | **0.185** | **0.118** |
| **S3** | **1.92** | **0.120** | **0.0776** |



**BA**
**S3**
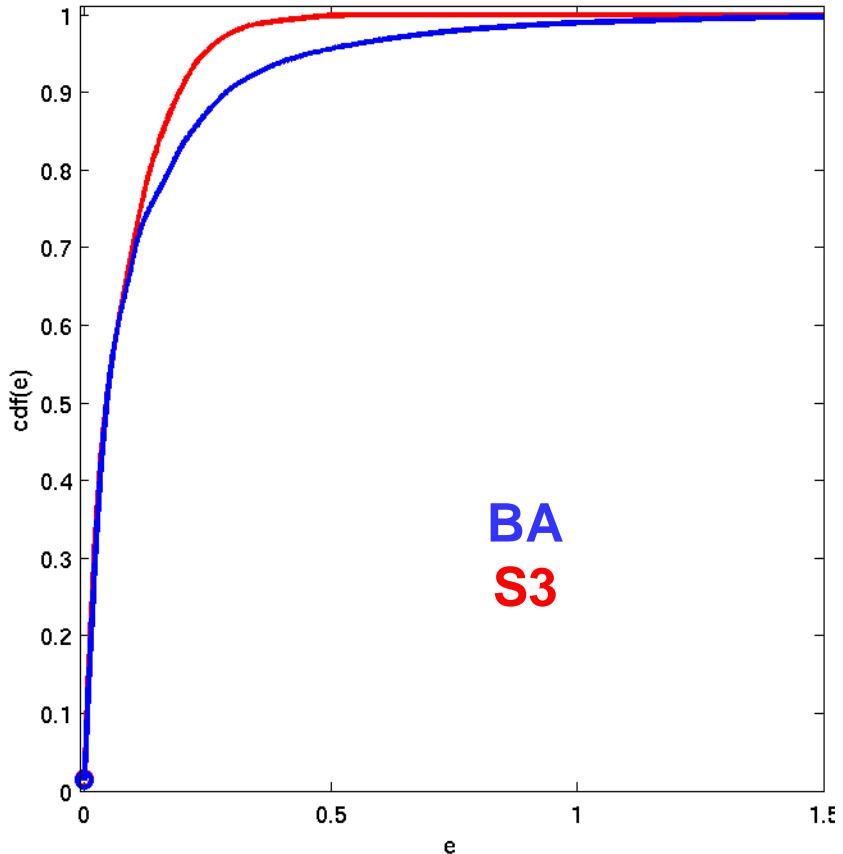
# TAXI: Hamburg Taxi



**256x190, (Barron 94)**
**max speed 3.0 pix/frame**

**LMS**

**BA**

**Ours**
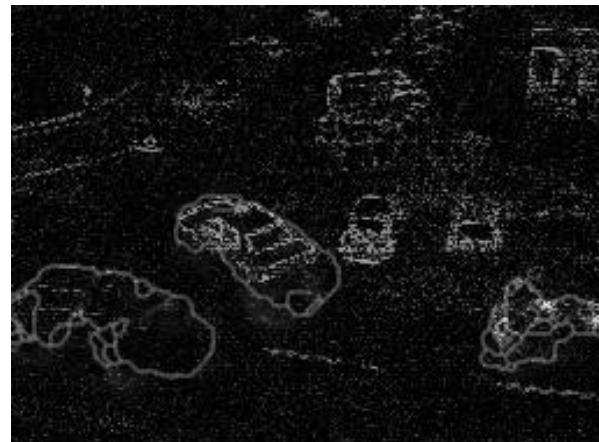
**Error map**

**Smoothness error**

# Traffic



512x512
(Nagel)
max speed:
6.0 pix/frame

**BA**
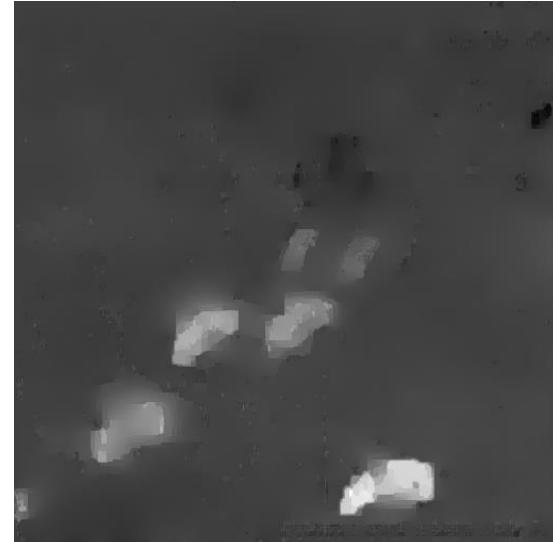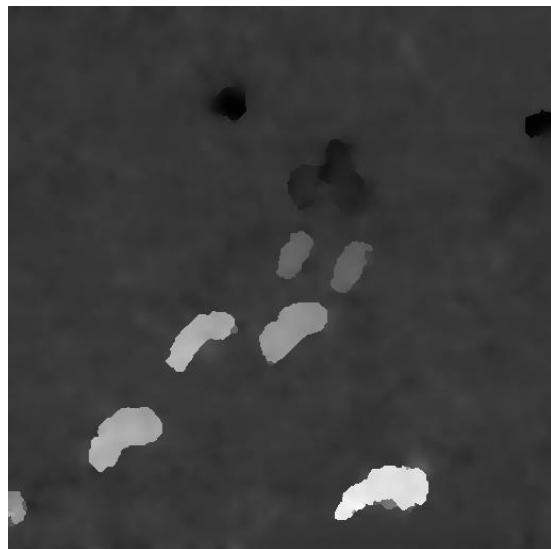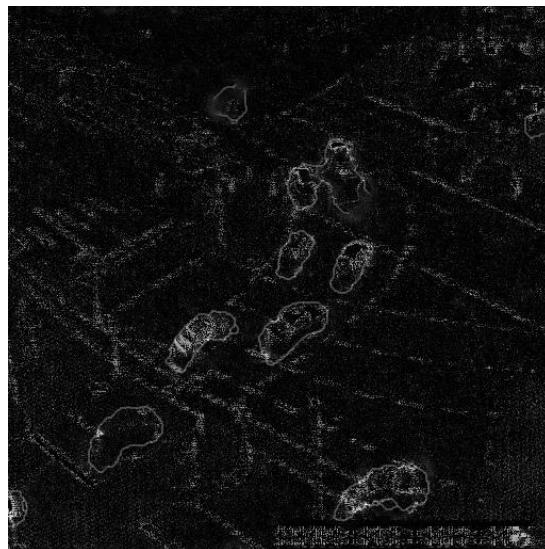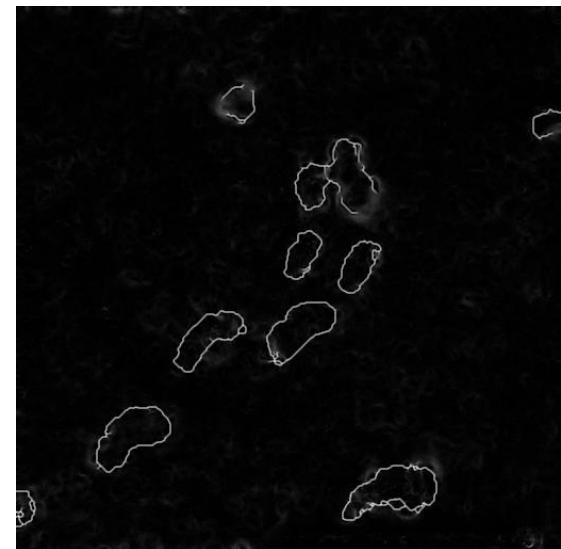
**Ours**            **Error map**            **Smoothness error**

# FG: Flower Garden
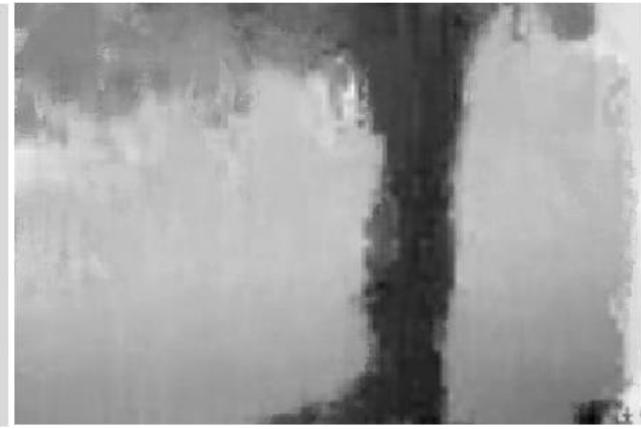


**360x240 (Black)**
**Max speed: 7pix/frame**

**BA**

**LMS**
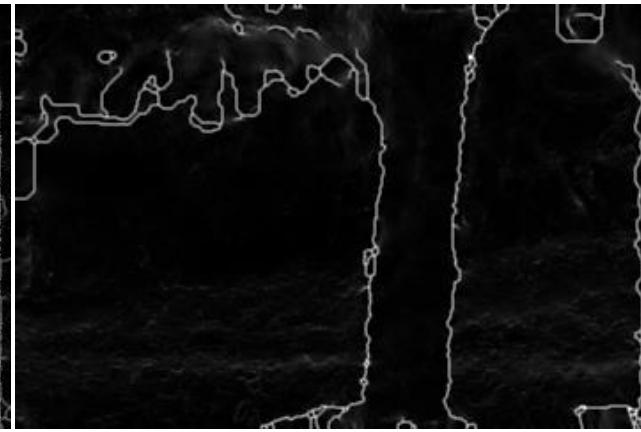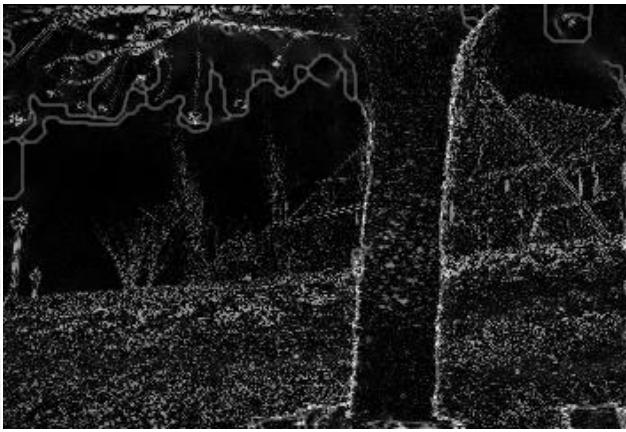
**Ours**

**Error map**

**Smoothness error**

# Representing Moving Images with Layers

J. Y. Wang and E. H. Adelson

MIT Media Lab

# Goal

- Represent moving images with sets of overlapping layers

- Layers are ordered in depth and occlude each other

- Velocity maps indicate how the layers are to be warped over time

# Simple Domain:
# Gesture Recognition



(a) Moving Hand

(b) Background

(c) Frame 1    Frame 2    Frame 3
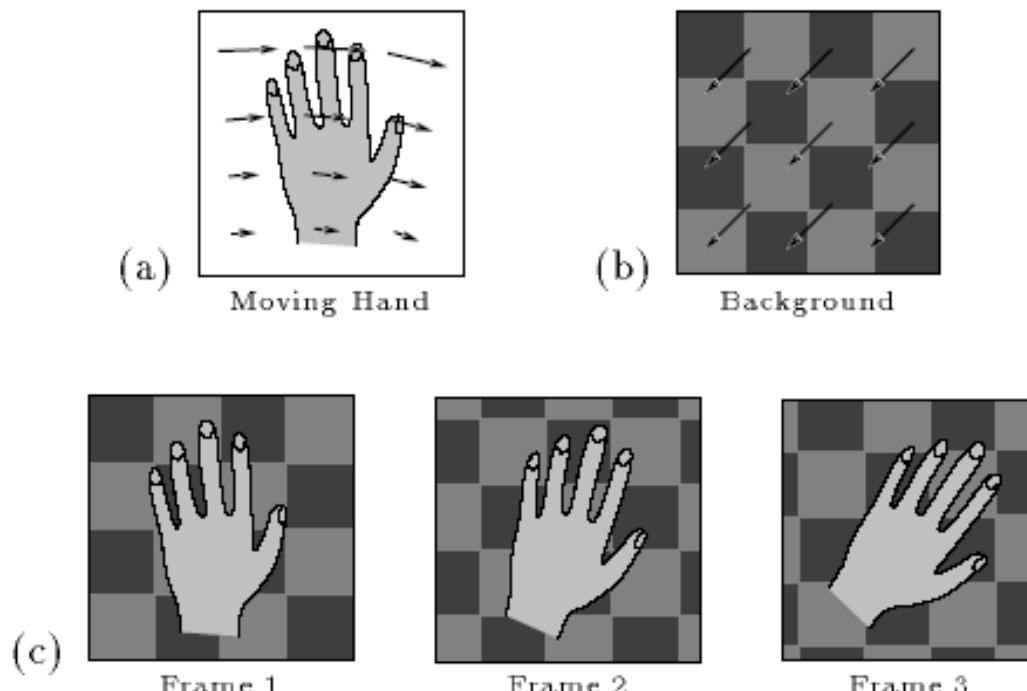
# More Complex:
# What are the layers?

# Motion Analysis Example
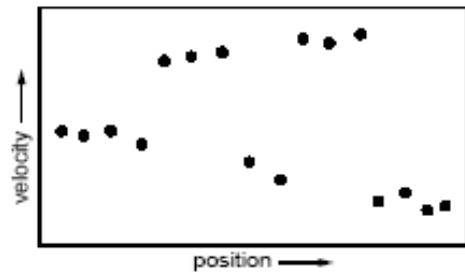


(a) velocity estimates
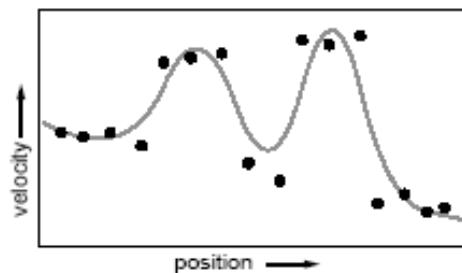
(b) velocity smoothing
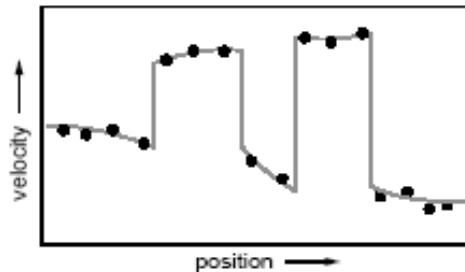
(c) regularization
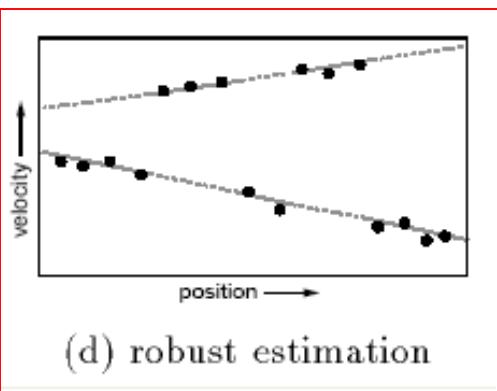
(d) robust estimation

2 separate layers shown as 2 affine models (lines);

The gaps show the occlusion.

# Motion Estimation Steps

1. Conventional optical flow algorithm and representation (uses multi-scale, coarse-to-fine Lucas-Kanade approach).

2. From the optical flow representation, determine a set of affine motions. Segment into regions with an affine motion within each region.

# Results
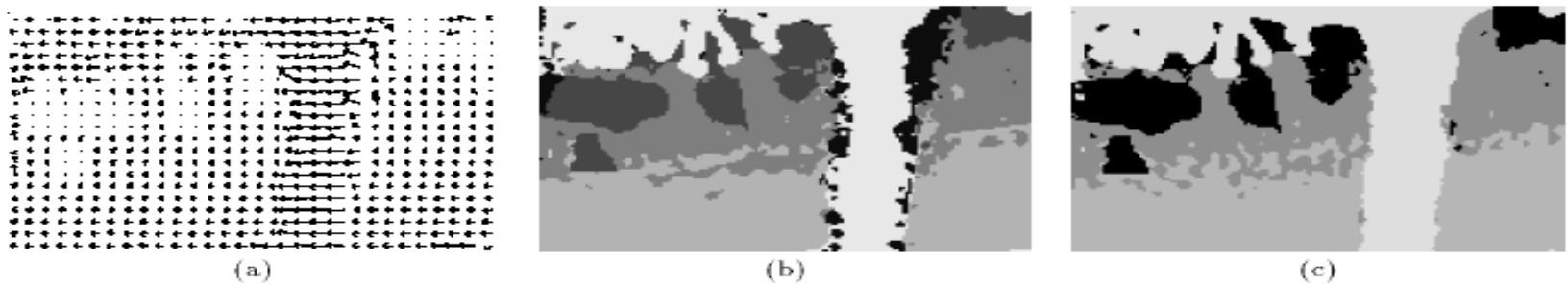


Figure 11: (a) The optic flow from multi-scale gradient method. (b) Segmentation obtained by clustering optic flow into affine motion regions. (c) Segmentation from consistency checking by image warping. Representing moving images with layers.
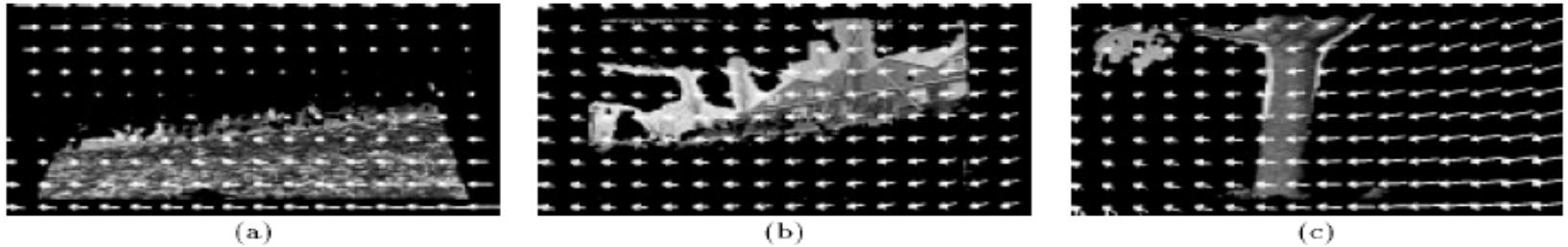


Figure 12: The layers corresponding to the tree, the flower bed, and the house shown in figures (a-c), respectively. The affine flow field for each layer is superimposed.

# Results
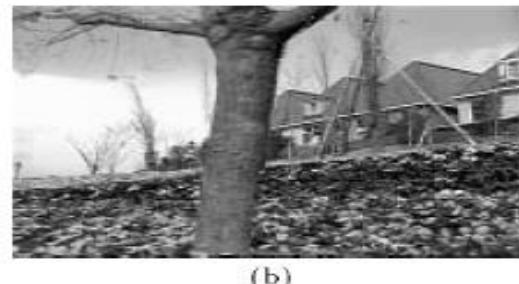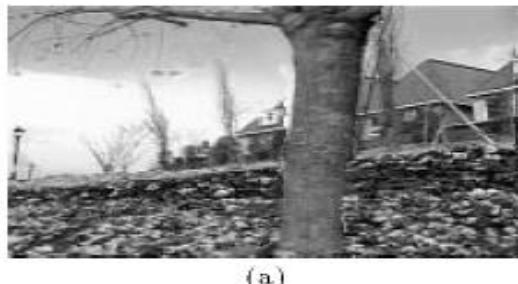


Figure 13: Frames 0, 15, and 30 as reconstructed from the layered representation shown in figures (a-c), respectively.
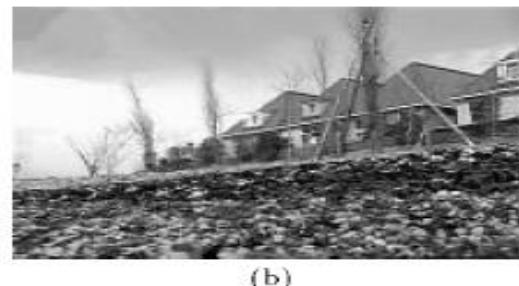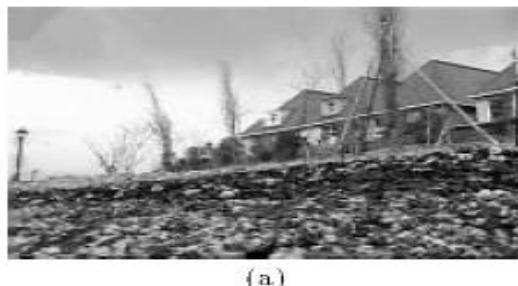


Figure 14: The sequence reconstructed without the tree layer shown in figures (a-c), respectively.

# Results
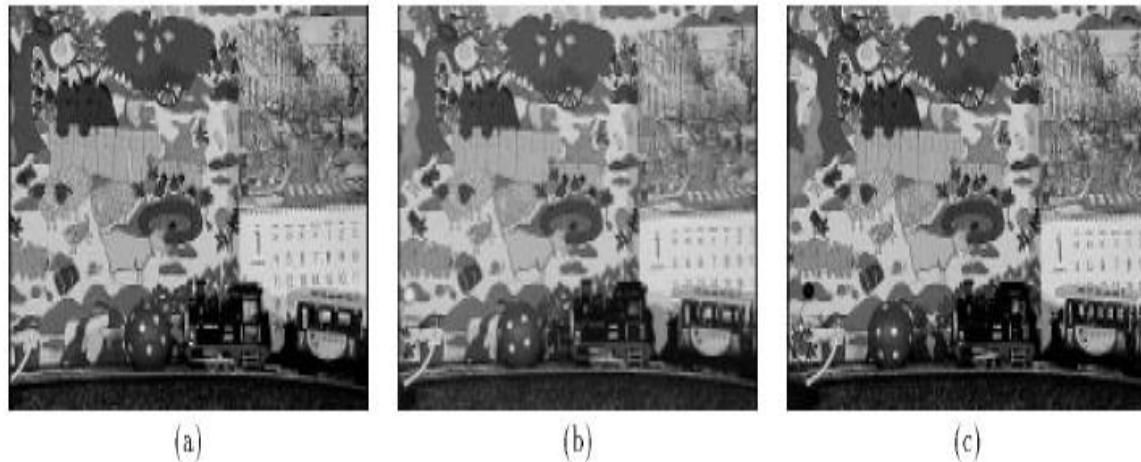


Figure 15: Frames 0, 15 and 30, of MPEG Calendar sequence shown in figures (a-c), respectively.
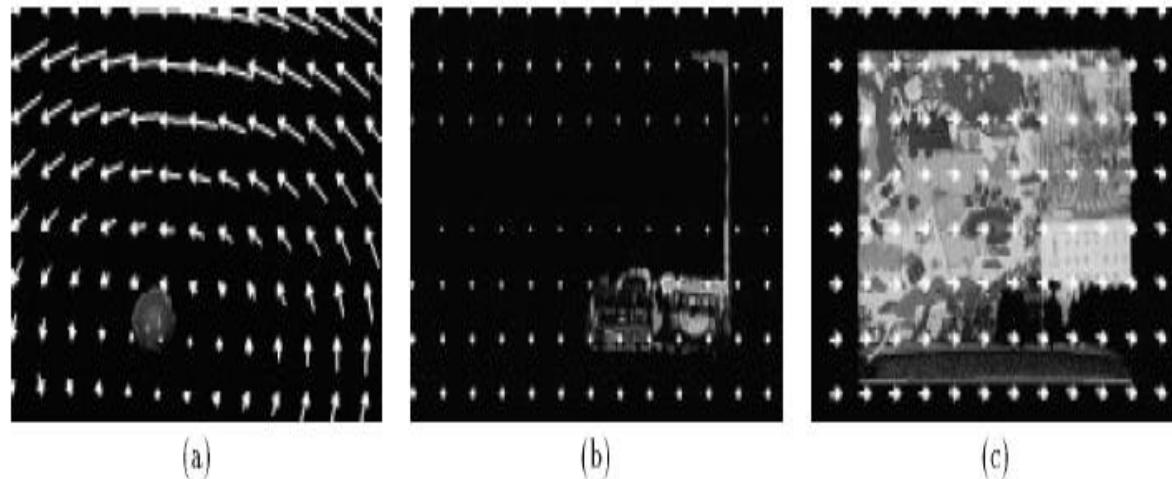


Figure 16: The layers corresponding to the ball, the train, and the background shown in figures (a-c), respectively.

# Summary

- Major contributions from Lucas, Tomasi, Kanade
  - Tracking feature points
  - Optical flow
  - Stereo
  - Structure from motion

- Key ideas
  - By assuming brightness constancy, truncated Taylor expansion leads to simple and fast patch matching across frames
  - Coarse-to-fine registration
  - Global approach by former EE student Ming Ye
  - Motion layers methodology by Wang and Adelson

# Homework 6
# Optical Flow
# WILL NOT BE ASSIGNED

# Motion