This lecture: Motion Analysis

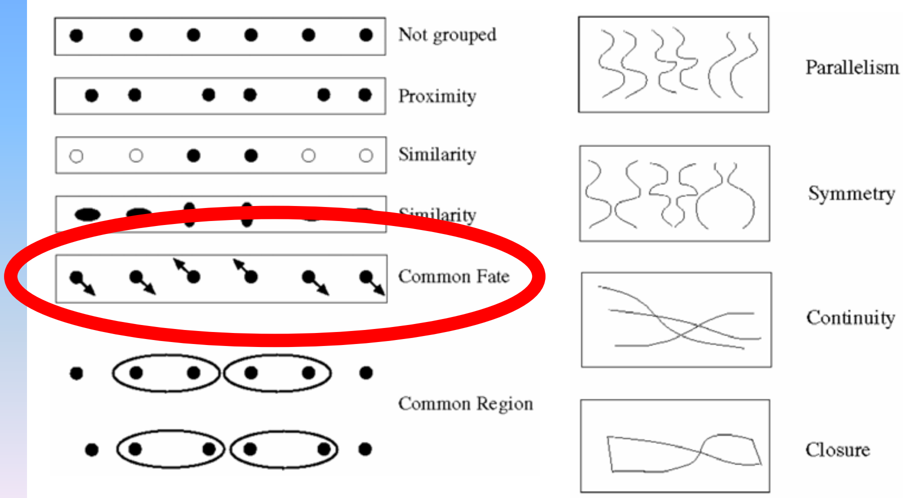
- Remainder of segmentation lecture
- Motion Analysis
  - Optical Flow
  - Aperture Problem
  - Lucas-Kanade
  - Horn-Shunk & Variational Methods
  - Layered Models & Segmentation

  - By Prof. I. Kokkinos

1

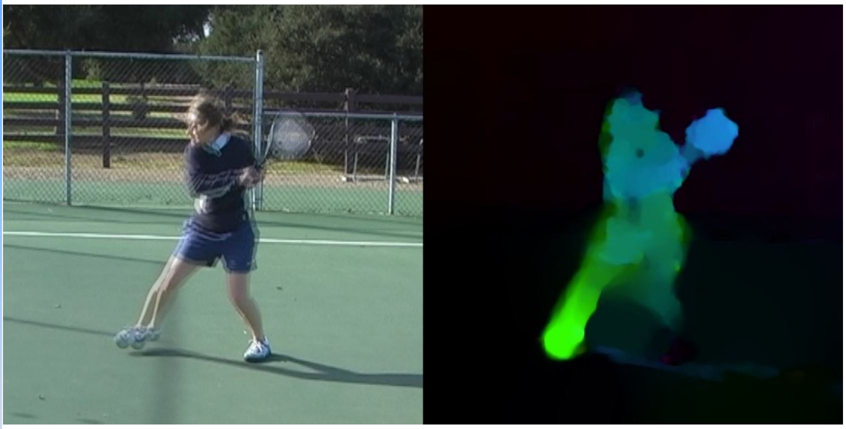
1

Motion and perceptual organization



2

Motion Estimation



Brox et. al. PAMI 2011

3

3

Why estimate visual motion?

- Recover dynamic information in a scene
  - Track objects and analyze trajectories
  - Action recognition
  - Segment objects based on motion
  - Applications: surveillance & security, car industry, robotics, multimedia, …
- Improve video quality
  - Camera instabilities cause jitter
  - Measure it and remove it (stabilize)
- Estimate 3D structure from motion pattern

4

Action Recognition

- Even impoverished input can reveal action class

G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

5

Action Recognition

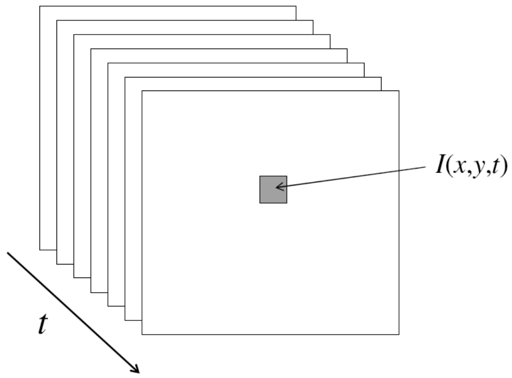
- Even impoverished input can reveal action class

G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

6

Video

- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)

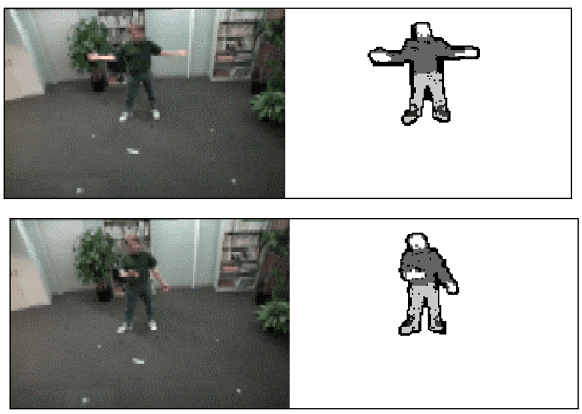


7

7

Motion Analysis & Video processing

- Background subtraction
  - A static camera is observing a scene.
  - Goal: separate the static *background* from the moving *foreground.*



How to come up with background frame estimate without access to "empty" scene?

8

8

Motion Segmentation & Surveillance

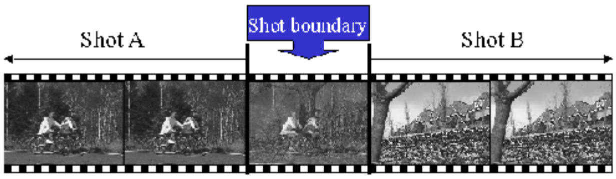
- Static camera observing crowds
  - Example application: count number of people going in two directions and detect emergencies

9

9

Motion Segmentation & Video Analysis

- Shot boundary detection
  - Commercial video is usually composed of *shots* or sequences showing the same objects or scene.
  - Goal: segment video into shots for summarization and browsing (each shot can be represented by a single keyframe).



- Customized delivery of multimedia
  - Detect beginnings of commercials/TV shows/weather/music
  - Deliver user-specific content (e.g. no R & B !)

10

10

This lecture: Motion Analysis

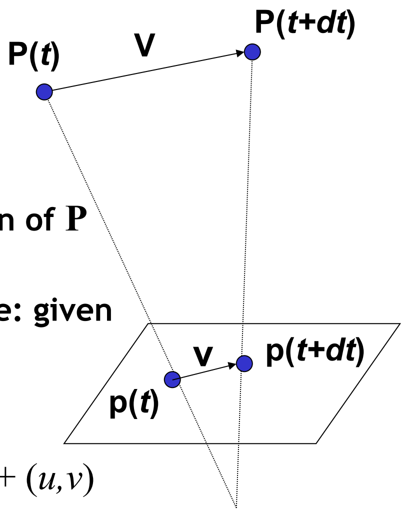
- Remainder of segmentation lecture
- Motion Analysis
  - Optical Flow
  - Aperture Problem
  - Lucas-Kanade
  - Horn-Shunk & Variational Methods
  - Layered Models & Segmentation

11

11

Motion Field

- $\mathbf{P}(t)$ is a moving 3D point
- Velocity of scene point: $\mathbf{V} = \mathrm{d}\mathbf{P}/\mathrm{d}t$
- $\mathbf{p}(t) = (x(t),y(t))$ is the projection of $\mathbf{P}$ in the image.
- Apparent velocity $\mathbf{v}$ in the image: given by components $u = \mathrm{d}x/\mathrm{d}t$ and $v = \mathrm{d}y/\mathrm{d}t$
- Considering $\mathrm{d}t = 1$: $\mathbf{p}(t+1) = \mathbf{p}(t) + (u,v)$
- Motion estimation task:
  - Estimate (u,v)



12

12

Optical flow

- Optical flow is the *apparent* motion of brightness patterns in the image
- Optical flow should ideally equal the motion field
- But apparent motion can be caused by lighting changes without any actual motion

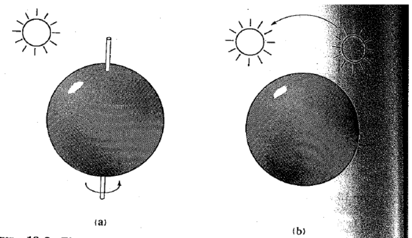
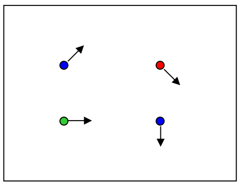


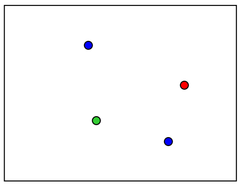
Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field is zero.

13

Estimating Optical Flow



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



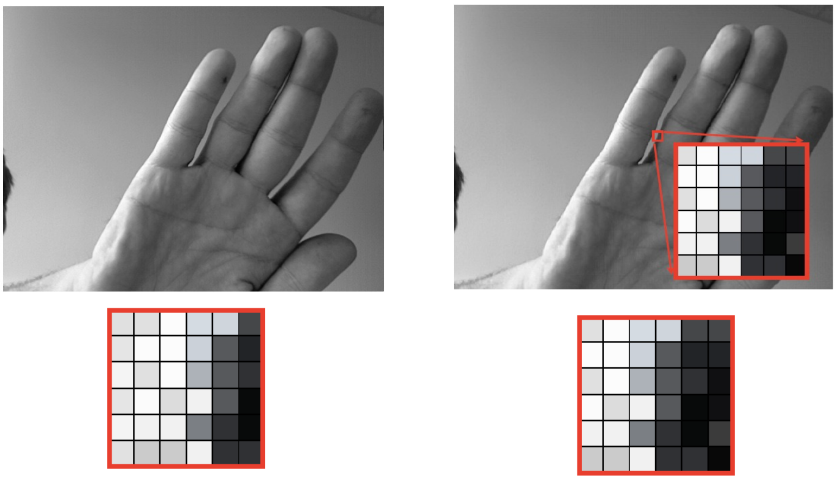
$I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them.

- Consider 1D case first

14

Optical flow estimation - 2D to 1D

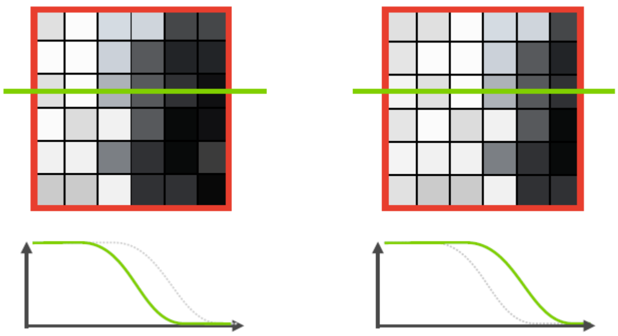


15

15

Optical flow estimation - 1D case

- How can we estimate the displacement?

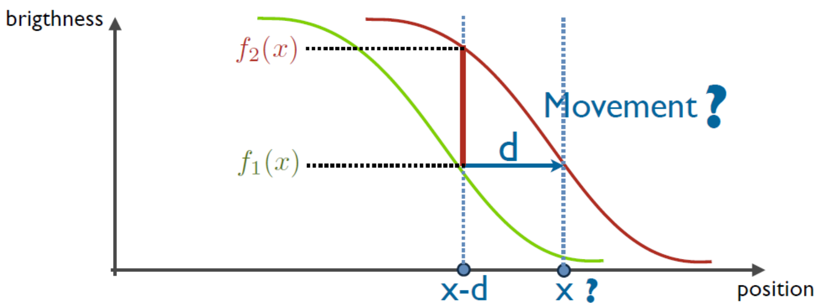


16

16

Optical Flow Estimation - 1D case

- Known: Gradient, Difference of intensities at x-d
- Unknown: d



$$d = \frac{f_1(x) - f_2(x)}{f_1'(x)}$$

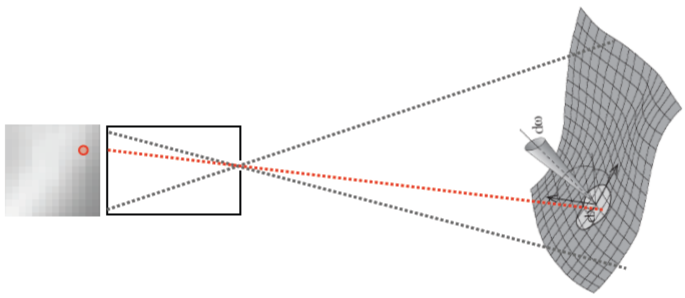
How about 2D?

17

17

Brightness constancy Constraint

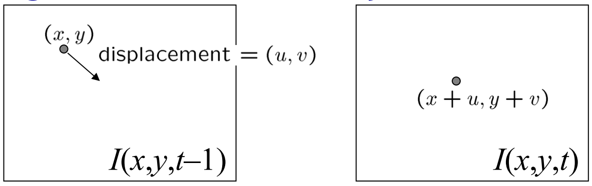
- Image is projection of 3D environment
- Each pixel: projection of a surface patch
- Pixel intensity influenced by 3D surface, incident light, camera …
- Assumption: intensity of surface patch remains constant



18

18

The Brightness Constancy Constraint



- **Brightness Constancy Equation:**

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

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

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

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

The Brightness Constancy Constraint

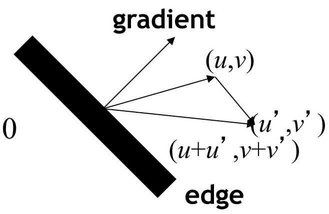
$$I_x \cdot u + I_y \cdot v + I_t = 0$$

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

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

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

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



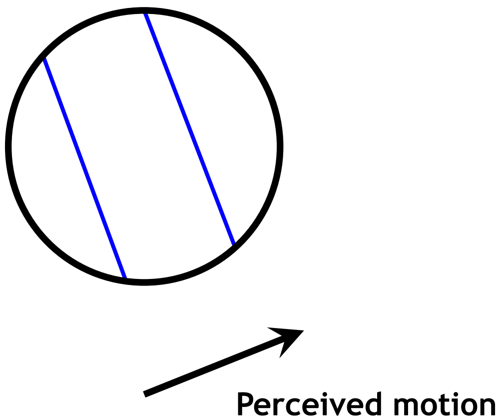
This lecture: Motion Analysis

- Remainder of segmentation lecture
- Motion Analysis
  - Optical Flow
  - Aperture Problem
  - Lucas-Kanade
  - Horn-Shunk & Variational Methods
  - Layered Models & Segmentation

21

21

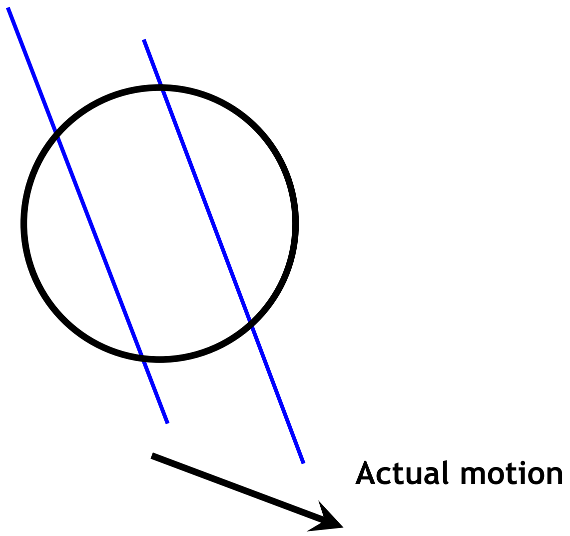
The Aperture Problem



22

22

The Aperture Problem



23

23

The Barber Pole Illusion

24

24

The Barber Pole Illusion

25

25

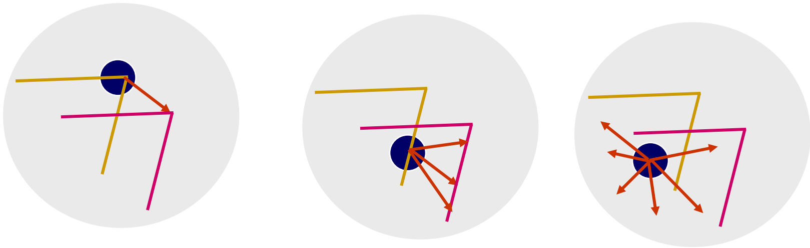
The Barber Pole Illusion

26

26

Local Patch Analysis

- How *certain* are the motion estimates?



27

This lecture: Motion Analysis

- Remainder of segmentation lecture
- Motion Analysis
  - Optical Flow
  - Aperture Problem
  - Lucas-Kanade
  - Horn-Shunk & Variational Methods
  - Layered Models & Segmentation

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

28

28

Solving the aperture problem

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

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

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

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

29

Lucas-Kanade flow

- Least squares problem:

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

- Minimum least squares solution given by solution of

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

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

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

30

30

Conditions for Solvability

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

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

- **When is this solvable?**
  - A^TA should be invertible.

31

31

Conditions for Solvability

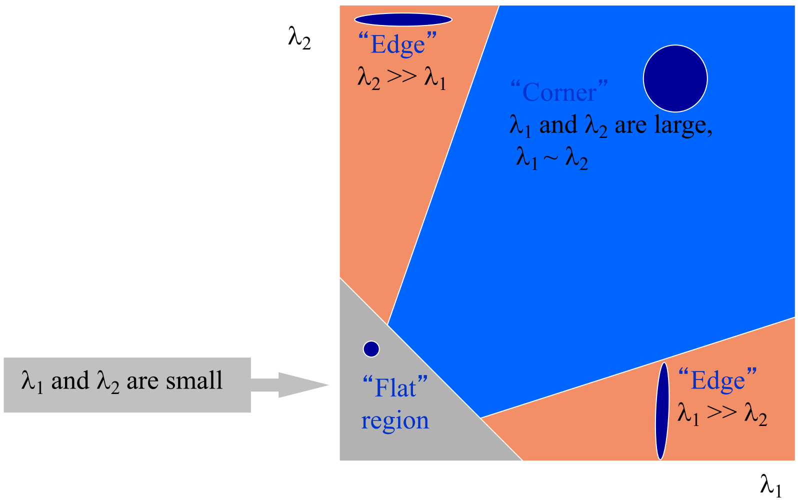
$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

- Recall the Harris corner detector: $M = A^TA$ is the *second moment matrix*
- We can figure out whether the system is solvable by looking at the eigenvalues of the second moment matrix
  - The eigenvectors and eigenvalues of *M* relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change, and the other eigenvector is orthogonal to it

32

Interpreting the eigenvalues

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

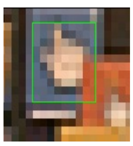
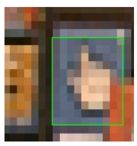


33

SSD measure reminder

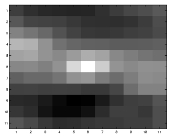
- **Sum of squared differences**

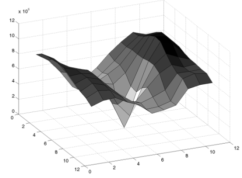
$$E(x, y; d) = \sum_{(x', y') \in N(x,y)} [I_L(x'+d, y') - I_R(x', y')]^2$$

34

High-Texture Region



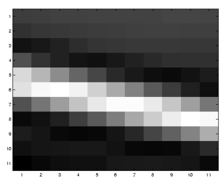


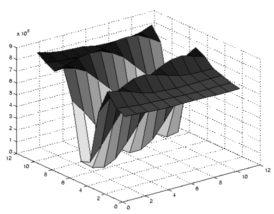
- Gradients are different, large magnitude
- Large λ_1 , large λ_2

35

35

Edge



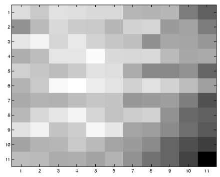


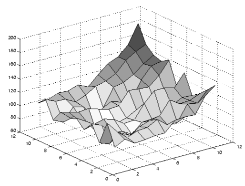
- Gradients very large or very small
- Large λ_1 , small λ_2

36

36

Low-Texture Region





- Gradients have small magnitude
- Small λ_1 , small λ_2

37

37

Shi-Tomasi feature tracker

1. Find good features (min eigenvalue of 2×2 Hessian)
2. Use Lucas-Kanade to track with pure translation
3. Use affine registration with first feature patch
4. Terminate tracks whose dissimilarity gets too large
5. Start new tracks when needed

38

Tracking example

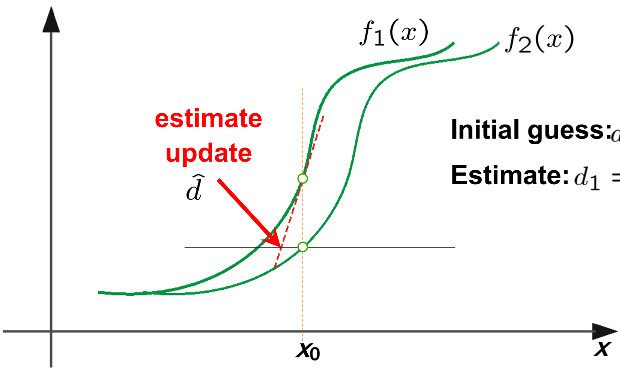
Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

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

39

Optical Flow: Iterative Estimation



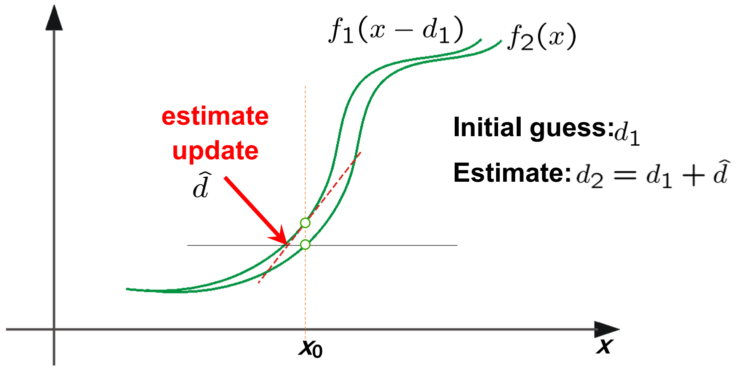
Initial guess: $d_0 = 0$

Estimate: $d_1 = d_0 + \hat{d}$

(using *d* for *displacement* here instead of *u*)

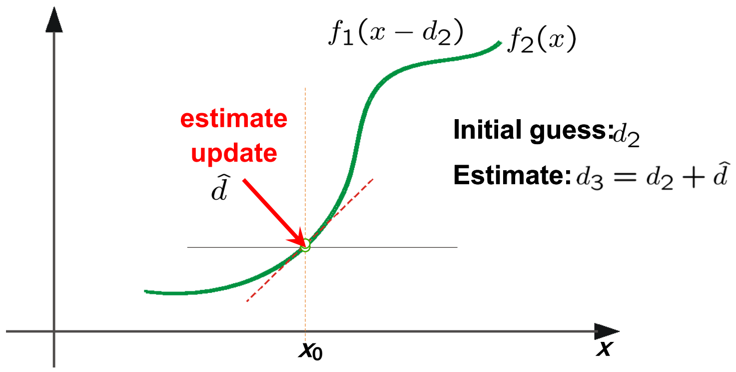
40

Optical Flow: Iterative Estimation



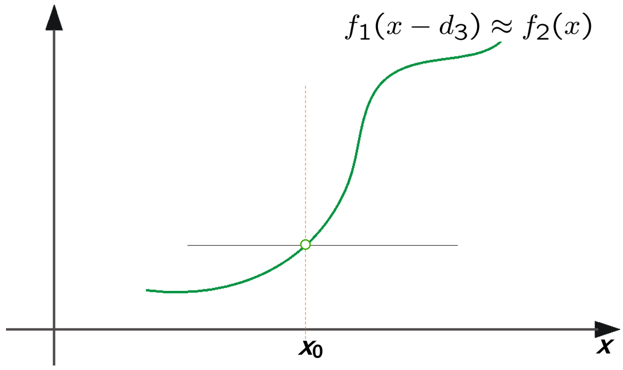
41

Optical Flow: Iterative Estimation



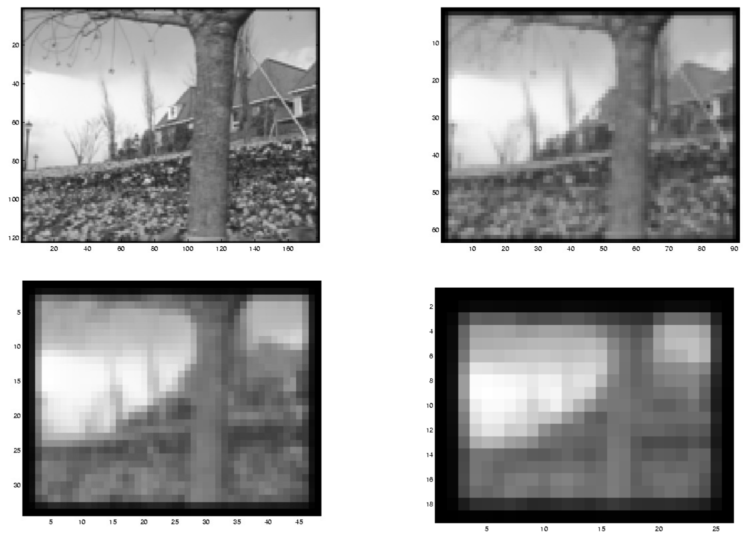
42

Optical Flow: Iterative Estimation



43

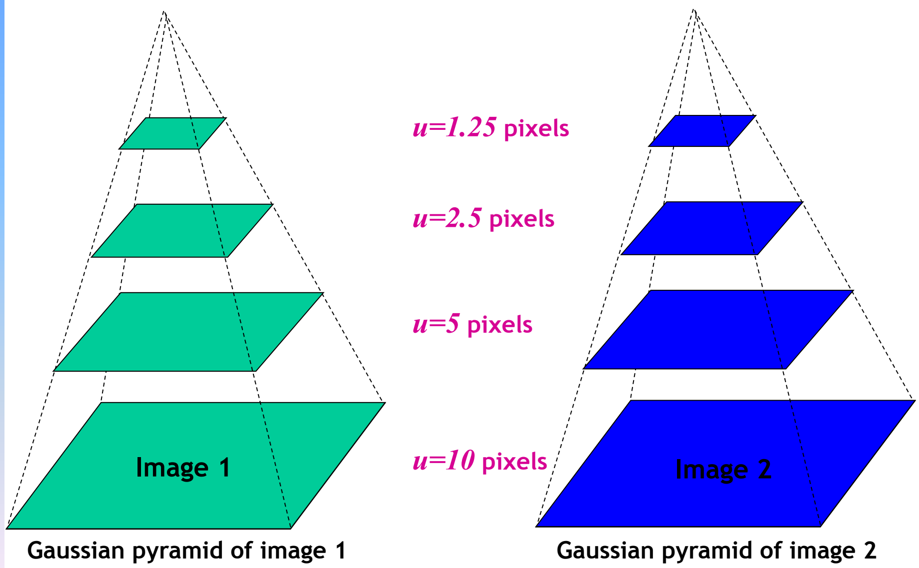
Large Displacements: Reduce Resolution!



44

44

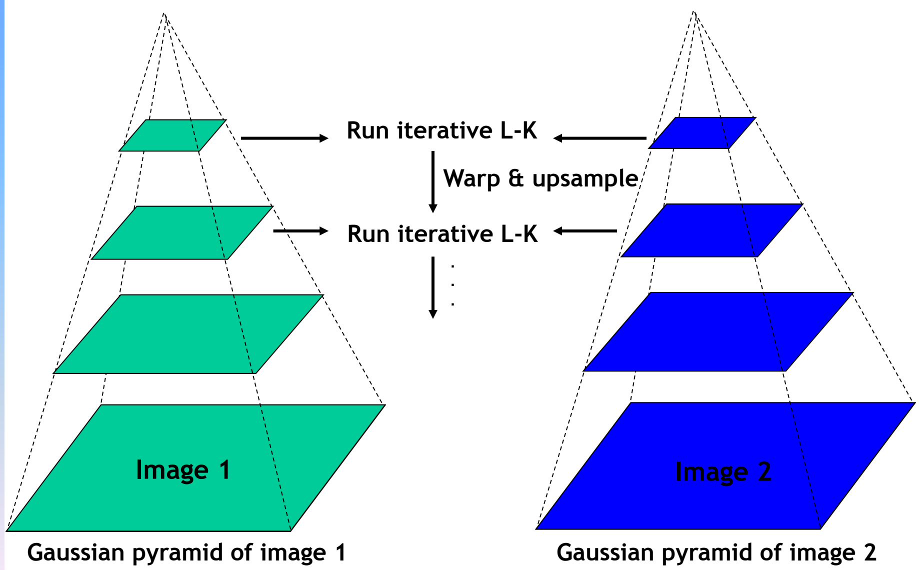
Coarse-to-fine Optical Flow Estimation



45

45

Coarse-to-fine Optical Flow Estimation

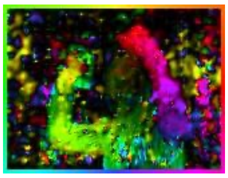


46

46

This lecture: Motion Analysis

- Remainder of segmentation lecture
- Motion Analysis
  - Optical Flow
  - Aperture Problem
  - Lucas-Kanade
  - Horn-Shunk & Variational Methods
  - Layered Models & Segmentation



Lucas Kanade

Horn Schunk

Anisotropic

47

47

Solving the aperture problem, revisted

- Lucas Kanade: pixel's neighbors have the same (u,v)
  - Brighntess constancy for a 5x5 window

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v] \qquad \begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

  - Solve for (u,v) separately at each pixel using least squares

- Horn-Schunk: neighboring pixels have similar (u,v)

$$J(u,v) = \frac{1}{2} \iint_\Omega \underbrace{[I_t + \nabla I \cdot (u,v)]^2}_{\text{Brightness Constancy}}$$

  - (u,v) estimates become coupled through Spatial Smoothness term
- Filling-in: smoothness term resolves aperture problem

48

Variational Optical Flow-I

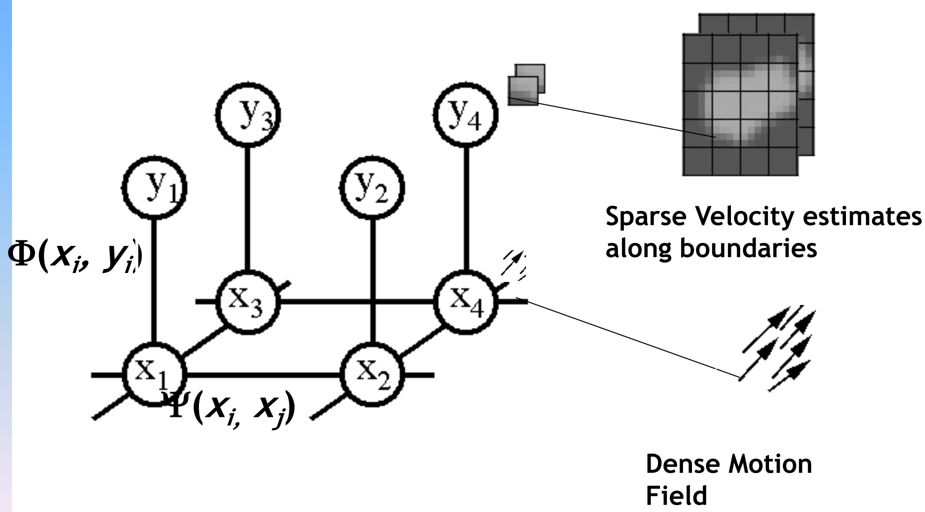
- Cost functional

$$J(u,v) = \frac{1}{2}\iint_{\Omega} \underbrace{\left[I_t + \nabla I \cdot (u,v)\right]^2}_{\text{Brightness Constancy}} + \lambda \underbrace{\left[(\nabla u)^2 + (\nabla v)^2\right]}_{\text{Spatial Smoothness}} \, dxdy$$

- This is a probabilistic graphical model!
- Can solve using least squares (as in Lucas-Kanade)
- Or using discrete optimization (e.g. belief propagation)

49

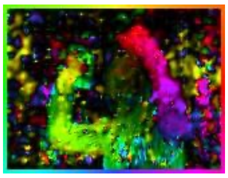
MRFs for Motion



55

Pre-Deep Learning State-of-the-art

- Bruhn, Weickert et. al.



Local Method

Horn-Schunck
(isotropic)

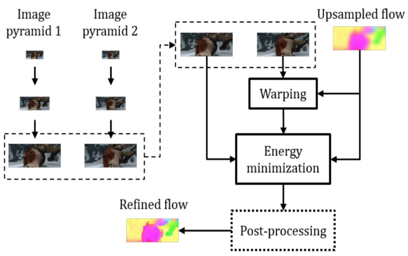
Total Variation
(anisotropic,
image-based)

Proposed
(anisotropic, flow-based)

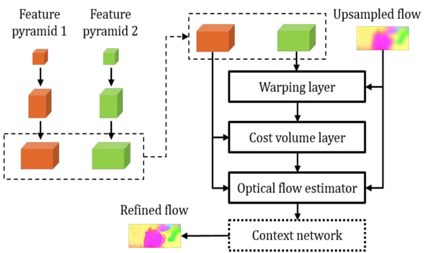
56

56

Flow Networks



Traditional coarse-to-fine flow



PWC-net

[Sun et al., “PWC-Net”, 2018]

57

57

Fun with flow

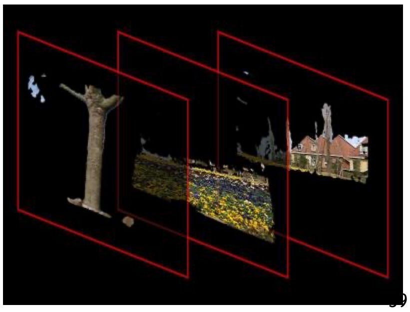
- http://www.youtube.com/watch?v=TbJrc6QCeU0&feature=related

- http://www.youtube.com/watch?v=pckFacsIWg4

58

This lecture: Motion Analysis

- Remainder of segmentation lecture
- Motion Analysis
  - Optical Flow
  - Aperture Problem
  - Lucas-Kanade
  - Horn-Shunk & Variational Methods
  - Layered Models & Segmentation



59

Motion Segmentation

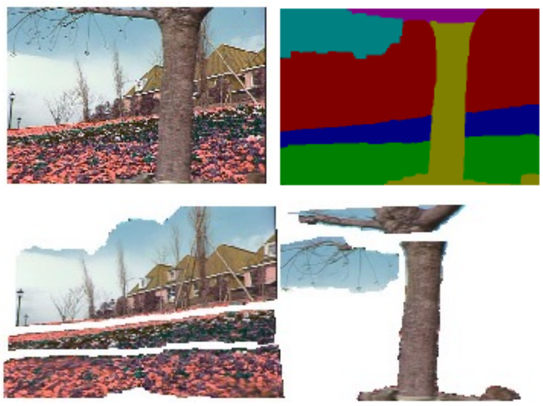
- How do we represent the motion in this scene?

J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993.* 60

60

Layered Motion

- Break image sequence into "layers" each of which has a coherent motion



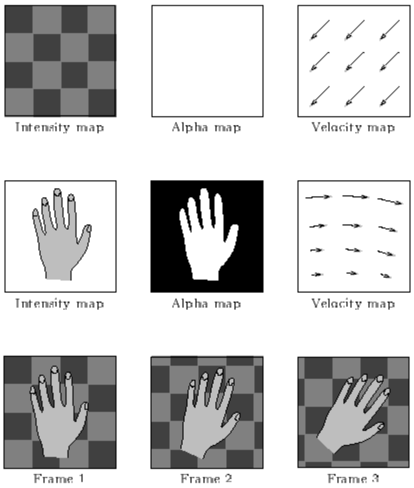
61

61

What is in each layer?

- **Intensities**
- **Alpha maps**
  - Transparency
- **Velocities**
  - Affine motion model

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



62

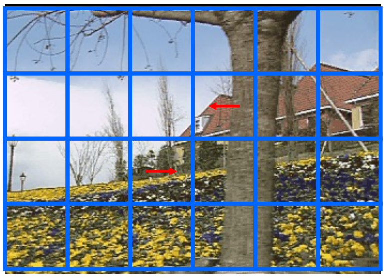
Layered motion

- **Advantages:**
  - can represent occlusions / disocclusions
  - each layer's motion can be smooth
  - video segmentation for semantic processing
- **Difficulties:**
  - how do we determine the correct number?
  - how do we assign pixels?
  - how do we model the motion?

63

Block-based motion prediction

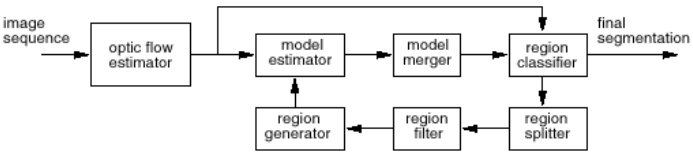
- Break image up into square blocks
- Estimate translation for each block
- Use this to predict next frame, code difference (MPEG-2)



64

How do we estimate the layers?

1. compute coarse-to-fine flow
2. estimate affine motion in blocks (regression)
3. cluster with *k-means*
4. assign pixels to best fitting affine region
5. re-estimate affine motions in each region…



65

Motion Segmentation with an Affine Model

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

Local flow estimates

66

66

Motion Segmentation with an Affine Model

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

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

67

67

Motion Segmentation with an Affine Model

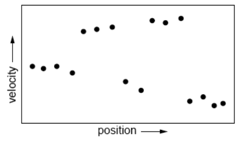
$$\boxed{u(x,y) = a_1 + a_2 x + a_3 y}$$

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

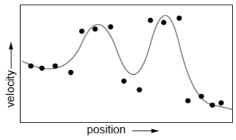
Equation of a plane (parameters a_1, a_2, a_3 can be found by least squares)

1D example

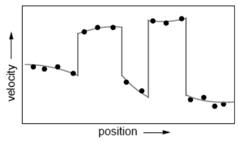
$u(x,y)$



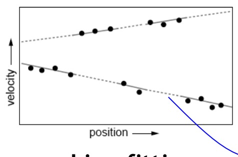
True flow



Local flow estimate



Segmented estimate



Line fitting

"Foreground"

"Background"

Occlusion

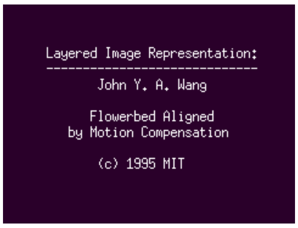
68

Segmentation results

69

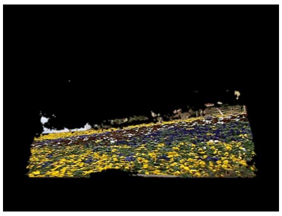
Once layered motion model is available

- Layer-based Stabilization



- Filling-in occlusions





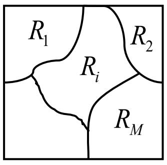
70

70

Motion Segmentation

- Probabilistic model for optical flow within each region
  - Affine motion model: special case
- Motion Competition: Region Competition for optical flow observations

$$J[\theta, \Gamma] = \sum_{i=1}^{K} \iint_{R_i} - \log \underset{\uparrow}{P(I; \theta_i)} + \frac{\beta}{2} \int_{\partial R_i} ds$$



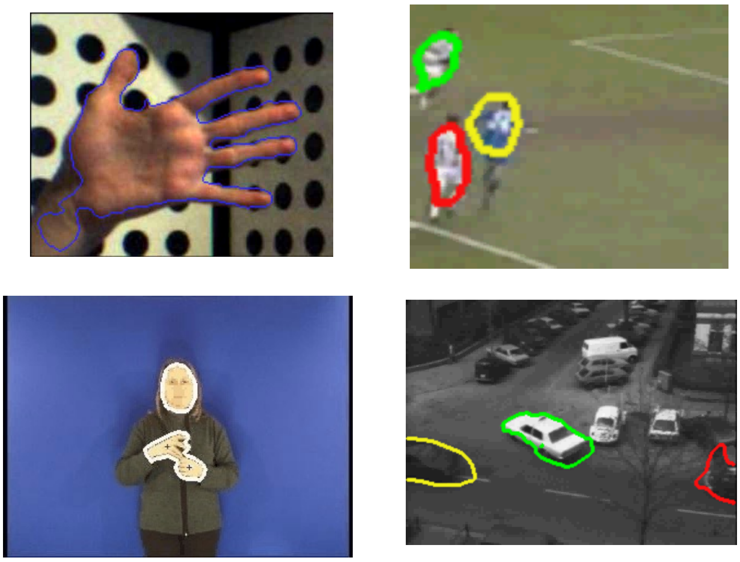
Pixel Velocity estimates

$$\frac{\partial \Gamma_i}{\partial t} = \log \frac{P(I; \theta_i)}{P(I; \theta_j)} \mathcal{N} - \beta \kappa \mathcal{N} \qquad \Gamma_i = \partial R_i$$

71

71

Variational Motion Segmentation Results

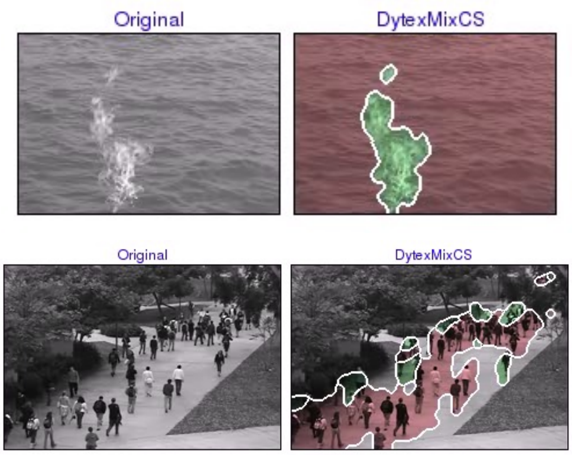


72

72

Dynamic Texture Segmentation

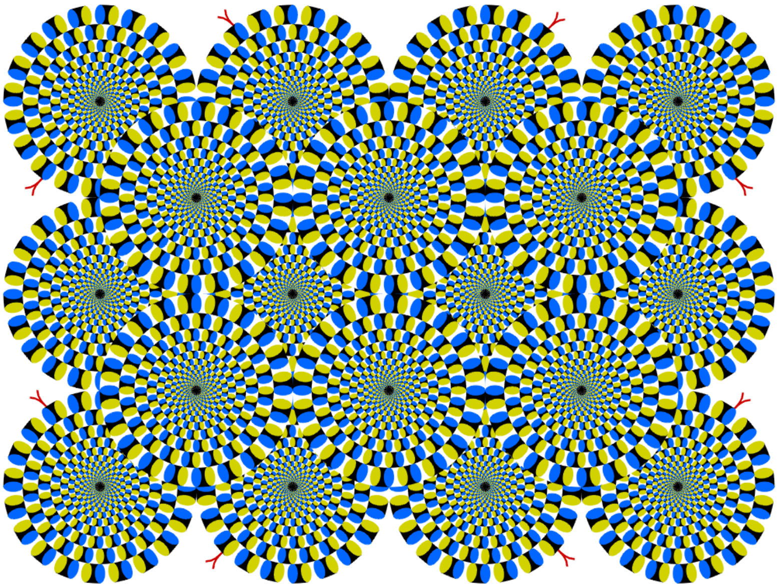
- Probabilistic model for texture within each region
  - Wait until next lecture for model



73

73

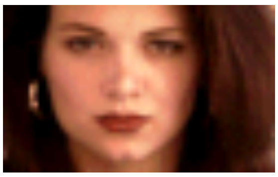
Could/Should a computer see motion here?



74

74

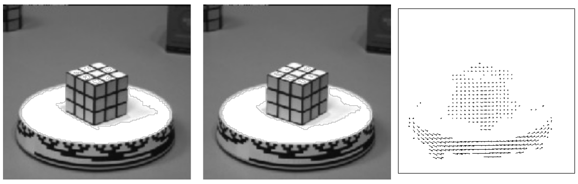
Image Morphing



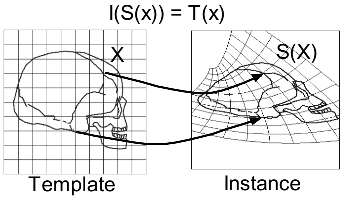
75

Motion Field & Deformations

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



- Image registration: a mapping from the domain of one image into that of another

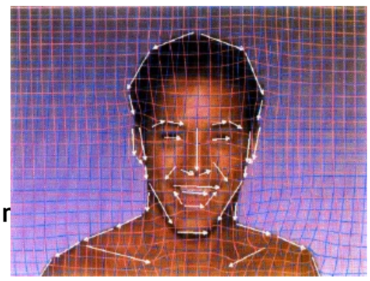


- Different physical problems, common techniques

76

Image Warping - non-parametric

- Specify more detailed warp function
- Examples:
  - splines
  - triangles
  - optical flow (per-pixel r



78

Image Warping - non-parametric

- Move control points to specify spline warp

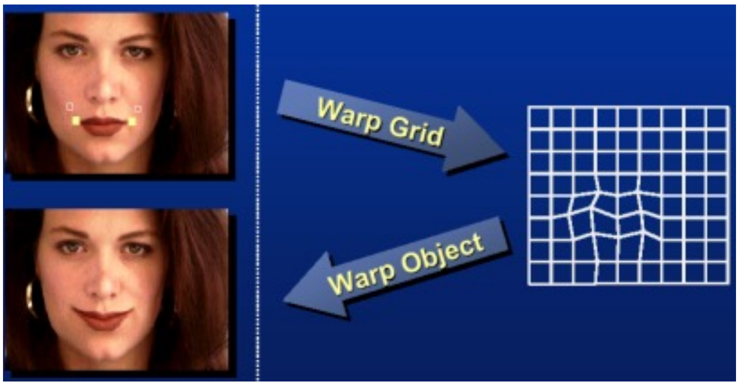


Image Morphing

- How can we *mix* two images?
  1. Cross-dissolve

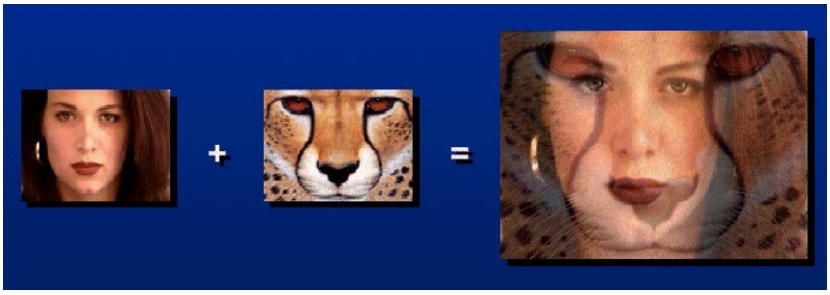
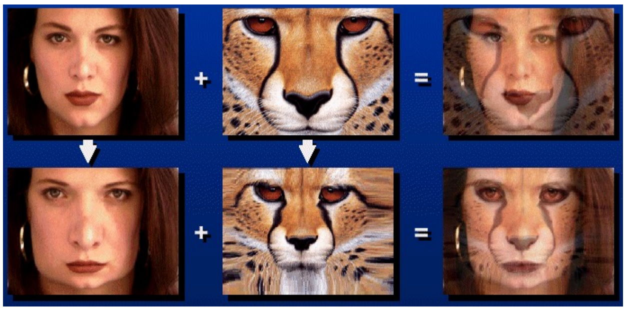


Image Morphing

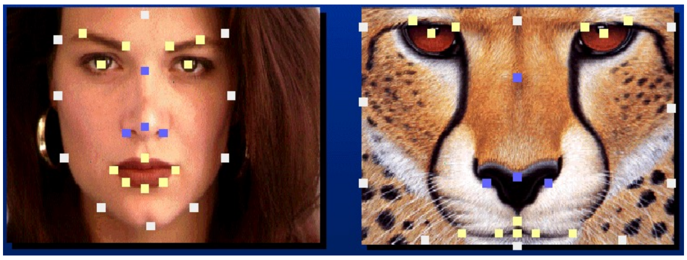
- How can we *mix* two images?
  2. Warp then cross-dissolve = *morph*



81

Warp specification

- How can we specify the warp?
  1. Specify corresponding *points*
     - *interpolate* to a complete warping function



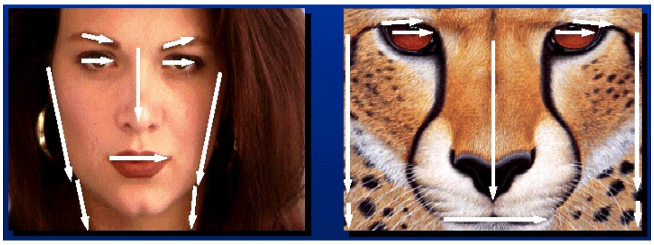
Motion estimation

82
CSE 576, Spring 2008

82

Warp specification

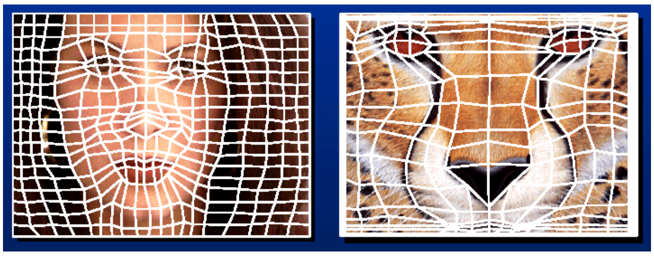
- How can we specify the warp?
  2. Specify corresponding *vectors*
     - *interpolate* to a complete warping function



83

Warp specification

- How can we specify the warp?
  3. Specify corresponding *spline control points*
     - *interpolate* to a complete warping function



84

Morphing Results

85