CAP 5415 Computer Vision
Fall 2005

Dr. Alper Yilmaz
Univ. of Central Florida
www.cs.ucf.edu/courses/cap5415/fall2005
Office: CSB 250

Recap
Motion

- Brightness constancy constraint

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

- Optical flow equation, brightness constancy equation
  - Normal flow **can be** computed
  - Parallel flow **cannot**

$$uI_x + vI_y + I_t = 0 \qquad v = u\frac{I_x}{I_y} + \frac{I_t}{I_y} = 0 \text{ line equation}$$

Recap
Horn & Schunck

$$E(x,y) = (uI_x + vI_y + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)$$

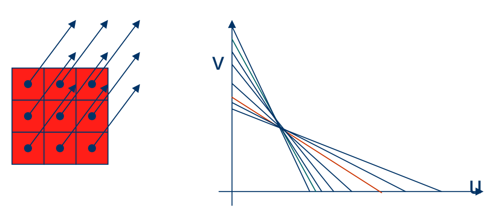
$$I_x(uI_x + vI_y + I_t) + \Delta^2 u = 0 \qquad I_y(uI_x + vI_y + I_t) + \Delta^2 v = 0$$

$$\Delta^2 u = u - u_{avg} \qquad \Delta^2 v = v - v_{avg}$$

$$\boxed{\begin{aligned} u\left(\lambda + I_x^2\right) + vI_xI_y + I_xI_t - \lambda u_{avg} &= 0 \\ v\left(\lambda + I_y^2\right) + uI_xI_y + I_yI_t - \lambda v_{avg} &= 0 \end{aligned}}$$

$$\boxed{\begin{aligned} u &= u_{avg} - I_x\left(\frac{I_x u_{avg} + I_y v_{avg} + I_t}{I_x^2 + I_y^2 + \lambda}\right) \\ v &= v_{avg} - I_y\left(\frac{I_x u_{avg} + I_y v_{avg} + I_t}{I_x^2 + I_y^2 + \lambda}\right) \end{aligned}}$$

Recap
Schunck



Find clusters of intersection points, select the cluster with maximum number of intersection as true flow

Recap
Lucas & Kanade

$$E=\sum(uI_x+vI_y+I_t)^2 \xrightarrow{\frac{\partial E}{\partial u}} \sum 2I_x(uI_x+vI_y+I_t)=0$$

$$E=\sum(uI_x+vI_y+I_t)^2 \xrightarrow{\frac{\partial E}{\partial v}} \sum 2I_y(uI_x+vI_y+I_t)=0$$

$$\boxed{\overbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} u \\ v \end{bmatrix}}^{u} = \overbrace{\begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}}^{B}}$$

$$u=A^{-1}B$$

Global Motion

Alper Yilmaz, Fall 2005 UCF

Global motion

- Common motion observed in the frame
  - Motion of **all** points in the scene
  - Motion of **most** of the points in the scene
- Reasons
  - Motion of sensor (Ego Motion)
  - Motion of a rigid scene
- Parametric flow describes optical flow for each pixel
  - Affine
  - Projective
- Global motion can be used to
  - Visual mosaics
  - Image registration
  - Removing camera jitter
  - Object tracking
  - Video segmentation

Alper Yilmaz, Fall 2005 UCF

Affine Motion

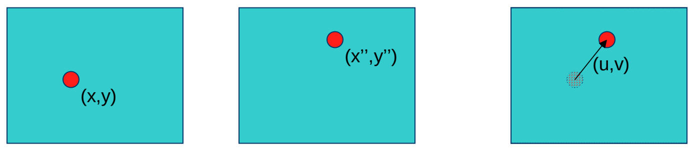


image at time t

image at time t+1

$$u = x'' - x$$
$$v = y'' - y$$

Affine motion:

$$u(x, y) = a_1 x + a_2 y + b_1$$
$$v(x, y) = a_3 x + a_4 y + b_2$$

$\boxed{a_1, a_2, b_1, a_3, a_4, b_2}$ Affine motion parameters

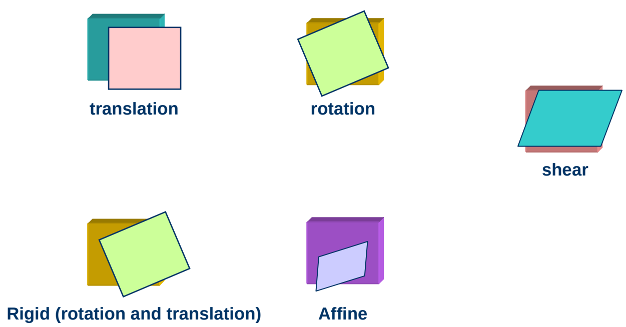
Global Affine Motion

$$u(x, y) = a_1 x + a_2 y + b_1$$
$$v(x, y) = a_3 x + a_4 y + b_2$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Spatial Transformations

- Transformations in image space



James R. Bergen, P. Anandan, Keith J. Hanna, Rajesh Hingorani: "Hierarchical Model-Based Motion Estimation," ECCV 1992: 237-252

Anandan's Approach

$$\boxed{\begin{aligned} u(x,y) &= a_1 x + a_2 y + b_1 \\ v(x,y) &= a_3 x + a_4 y + b_2 \end{aligned}}$$

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ a_3 \\ a_4 \\ b_2 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mathbf{u(x) = X(x)a}$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{X}(\mathbf{x})\mathbf{a}$$

Anandan's Approach

- Optical flow equation

$$I_x u + I_y v = -I_t \quad \Rightarrow \quad \underbrace{\begin{bmatrix} I_x & I_y \end{bmatrix}}_{\Delta I^T} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\mathbf{u}} = -I_t$$

transpose

- Energy functional

$$E(\mathbf{u}) = \sum_{\text{all pixels}} (I_t + \Delta I^T \mathbf{u})^2 \qquad E(\mathbf{a}) = \sum_{\text{all pixels}} (I_t + \Delta I^t \mathbf{X}\mathbf{a})^2$$

- Minimize energy by taking derivative and equal it to 0

Alper Yilmaz, Fall 2005 UCF

Anandan's Approach

$$E(\mathbf{a}) = \sum_{\text{all pixels}} (I_t + \Delta I^T \mathbf{X}\mathbf{a})^2$$

$$\frac{\partial E}{\partial \mathbf{a}} = 2 \sum_{\text{all pixels}} (\Delta I^T \mathbf{X})^T (I_t + \Delta I^T \mathbf{X}\mathbf{a}) = 0$$

$$\sum_{\text{all pixels}} \mathbf{X}^T \Delta I I_t + \sum_{\text{all pixels}} \mathbf{X}^T \Delta I \Delta I^T \mathbf{X}\mathbf{a} = 0$$

$$\boxed{\sum_{\text{all pixels}} \mathbf{X}^T \Delta I \Delta I^T \mathbf{X}\mathbf{a} = -\sum_{\text{all pixels}} \mathbf{X}^T \Delta I \; I_t}$$

Alper Yilmaz, Fall 2005 UCF

Anandan's Approach

$$\sum_{\text{all pixels}} \mathbf{X}^T \Delta I \Delta I^T \mathbf{X} \mathbf{a} = - \sum_{\text{all pixels}} \mathbf{X}^T \Delta I \; I_t$$

2x1 matrix 2x6 matrix

$$\mathbf{A}_{6\times6}\mathbf{a}_{6\times1} = \mathbf{B}_{6\times1} \Rightarrow \mathbf{a} = \mathbf{A}^{-1}\mathbf{B} \qquad \textbf{(A)}$$

We will iteratively compute affine parameters **a**

Anandan's Approach

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \qquad \begin{bmatrix} u \\ v \end{bmatrix} = \overbrace{\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}}^{A} \begin{bmatrix} x \\ y \end{bmatrix} + \overbrace{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}^{B}$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

$$X'' = (A + I)X + B$$

Anandan's Approach

- Initially assume a_1 and a_4 is 1 others are 0
- Compute $\mathbf{a}_i$ where *i* is iteration number

$$A_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- *X* is original position.
  - Let using A_0 and X we compute X'. Our goal is to compute X''

    $$X' = (A_0 + I)X + B_0$$

  - Using X' we compute A_1

    $$X'' = (A_1 + I)X' + B_1$$

Alper Yilmaz, Fall 2005 UCF

Anandan's Approach

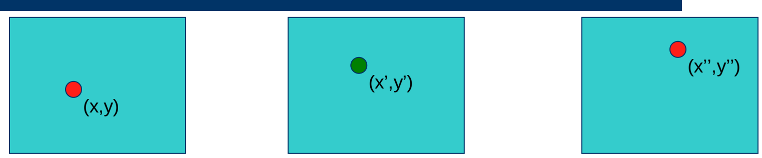


image at time t | intermediate iteration | image at time t+1

$$X' = \underbrace{(A_0 + I)}_{A'_0} X + B_0 \qquad X'' = \underbrace{(A_1 + I)}_{A'_1} X' + B_1$$

(B) $\boxed{X'' = A'_1 A'_0 X + A'_1 B_0 + B_1}$

$$\boxed{\hat{A} = A'_1 A'_0}$$

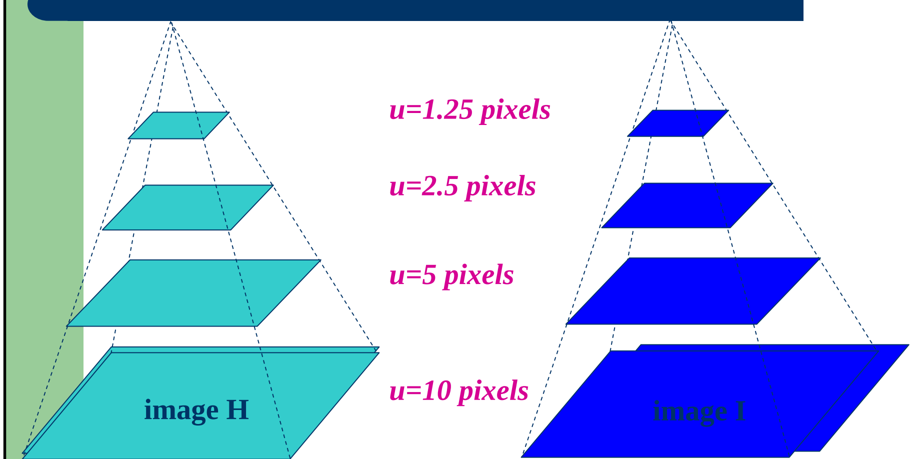
$$\boxed{\hat{B} = A'_1 B_0 + B_1}$$

Alper Yilmaz, Fall 2005 UCF

Algorithm

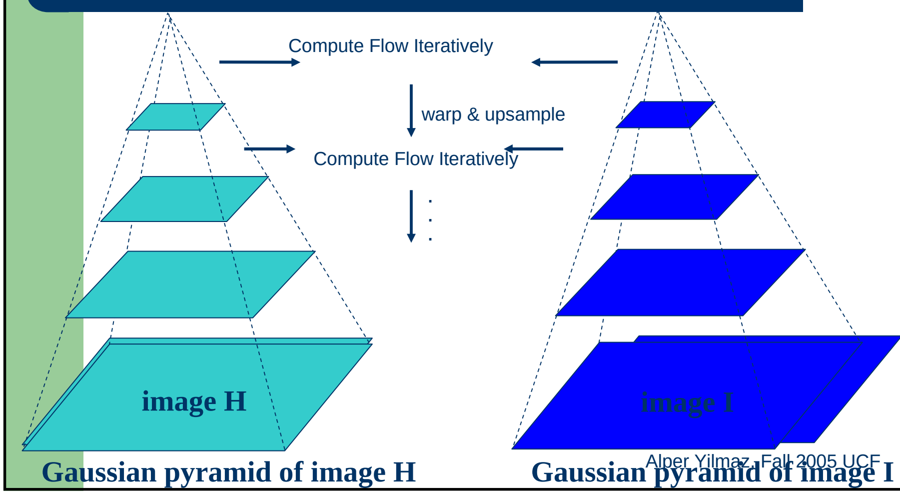
- Initialize affine parameters
- Compute affine parameters iteratively
  - Compute new affine parameters using (A)
  - At each iteration update the global affine solution using (B)
  - Stop when affine parameters do not update
- If motion in between frames is high construct pyramid representation.

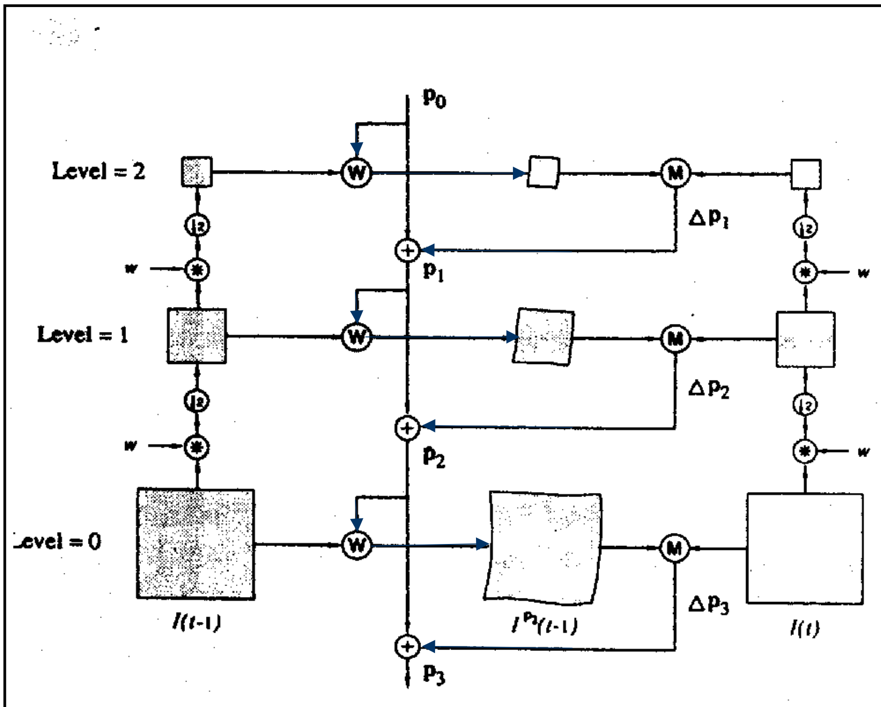
Using Pyramids



Gaussian pyramid of image H **Gaussian pyramid of image I**

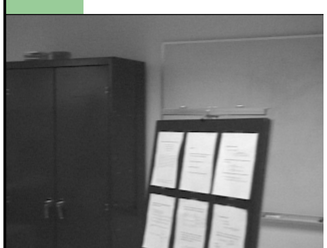
Using Pyramids

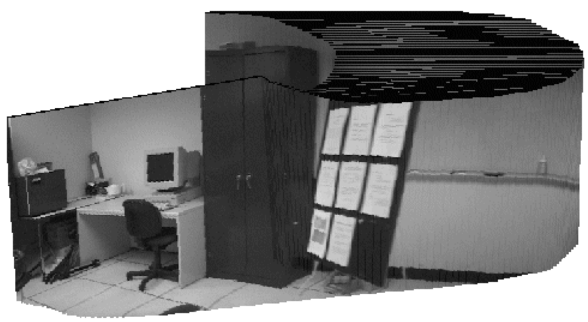




Examples

Examples





Examples

Examples

Programming assignment

- Implement Lucas Kanade optical flow algorithm
  - Deliverables: Report including the input images (two consecutive images), pyramid levels (2 levels), flow field computed for each level independently.
  - Due date November 16, 2005