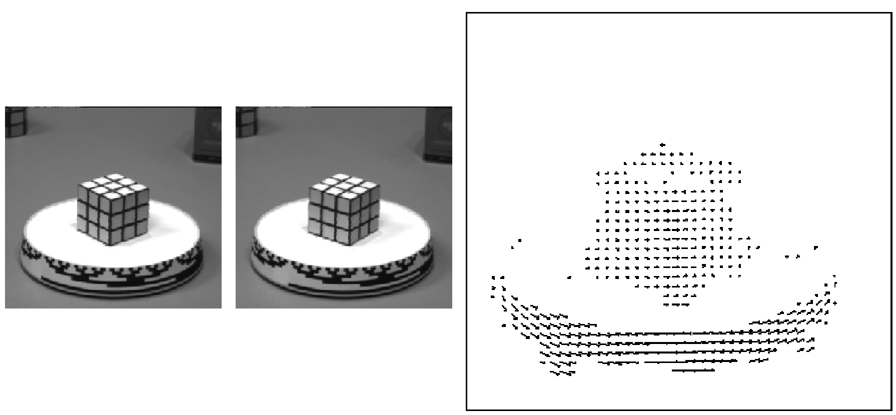
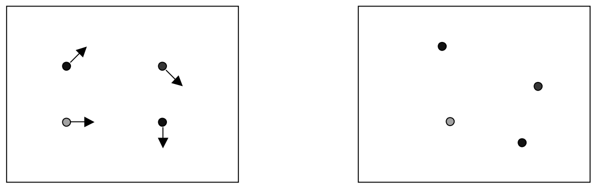
Motion Estimation

Optical flow

Measurement of motion at every pixel



Problem definition: optical flow



$H(x, y)$ $\qquad$ $I(x, y)$

How to estimate pixel motion from image H to image I?

- Solve pixel correspondence problem
  - given a pixel in H, look for nearby pixels of the same color in I

Key assumptions

- **color constancy:** a point in H looks the same in I
  - For grayscale images, this is **brightness constancy**
- **small motion**: points do not move very far

This is called the **optical flow** problem

Optical flow equation

$$0 = I_t + \nabla I \cdot [u \;\; v]$$

Lukas-Kanade flow

Prob: we have more equations than unknowns

$$\underset{25\times2}{A}\ \underset{2\times1}{d} = \underset{25\times1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

Solution: solve least squares problem

- minimum least squares solution given by solution (in d) of:

$$\underset{2\times2}{(A^T A)}\ \underset{2\times1}{d} = \underset{2\times1}{A^T b}$$

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

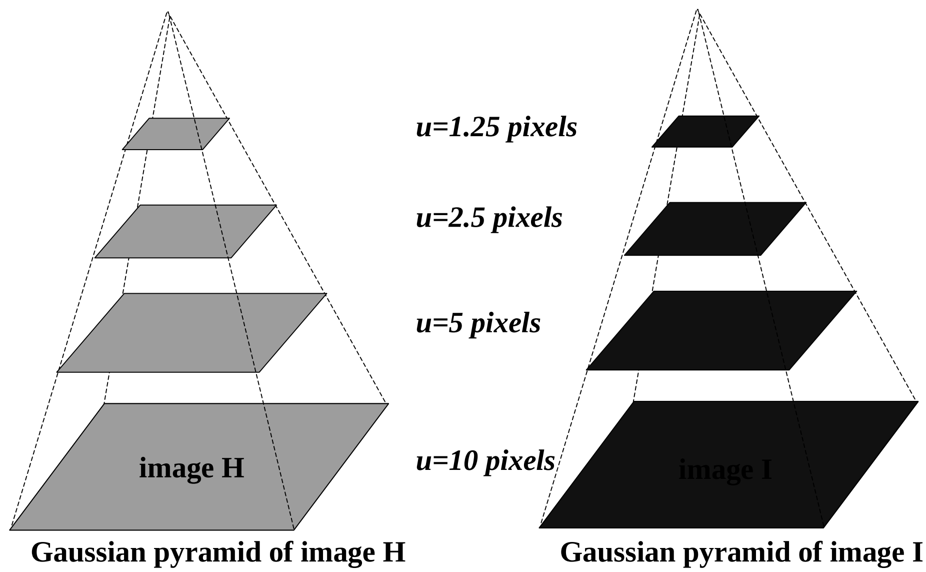
- The summations are over all pixels in the K x K window
- This technique was first proposed by Lukas & Kanade (1981)
  - described in Trucco & Verri reading

Iterative Refinement

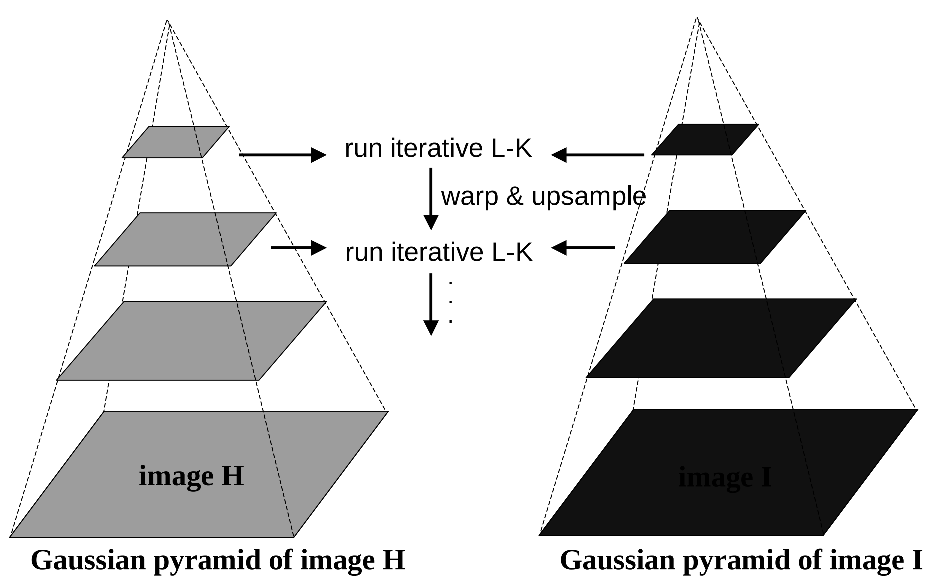
Iterative Lukas-Kanade Algorithm

1. Estimate velocity at each pixel by solving Lucas-Kanade equations
2. Warp H towards I using the estimated flow field
   - use image warping techniques
3. Repeat until convergence

Coarse-to-fine optical flow estimation



Coarse-to-fine optical flow estimation

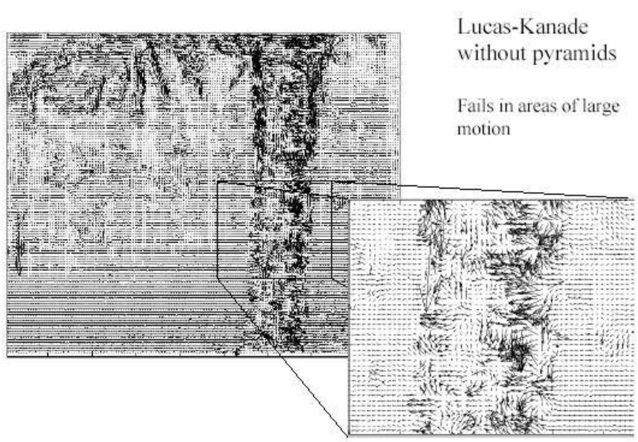


Multi-resolution Lucas Kanade Algorithm

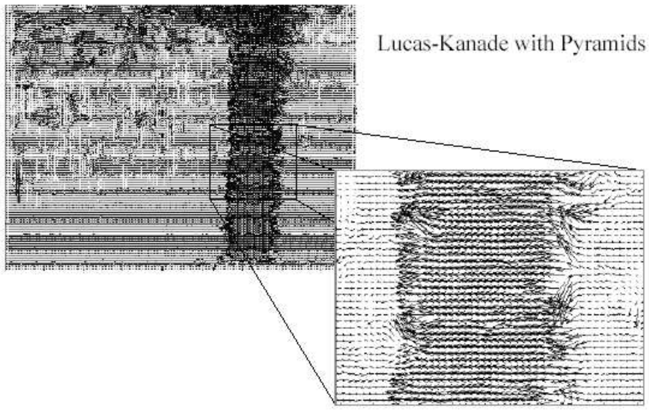
Compute Iterative LK at highest level
For Each Level *i*

- Take flow $u(i-1)$, $v(i-1)$ from level $i-1$
- Upsample the flow to create $u^*(i)$, $v^*(i)$ matrices of twice resolution for level *i*.
- Multiply $u^*(i)$, $v^*(i)$ by 2
- Compute I_t from a block displaced by $u^*(i)$, $v^*(i)$
- Apply LK to get $u'(i)$, $v'(i)$ (the correction in flow)
- Add corrections $u'(i)$, $v'(i)$ to obtain the flow $u(i)$, $v(i)$ at i^{th} level, i.e., $u(i)=u^*(i)+u'(i)$, $v(i)=v^*(i)+v'(i)$

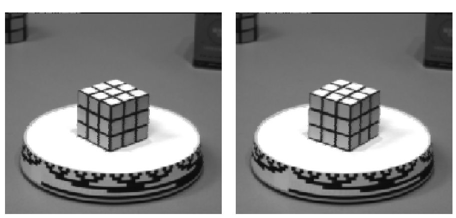
Optical Flow Results

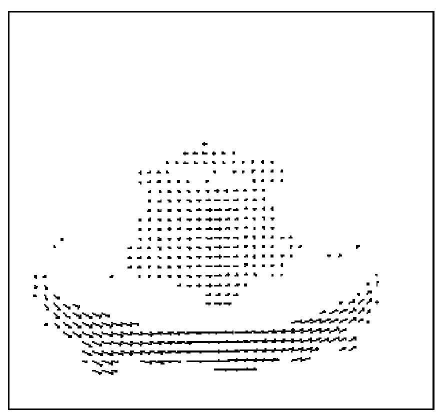


Optical Flow Results



Optical flow Results





Global Flow

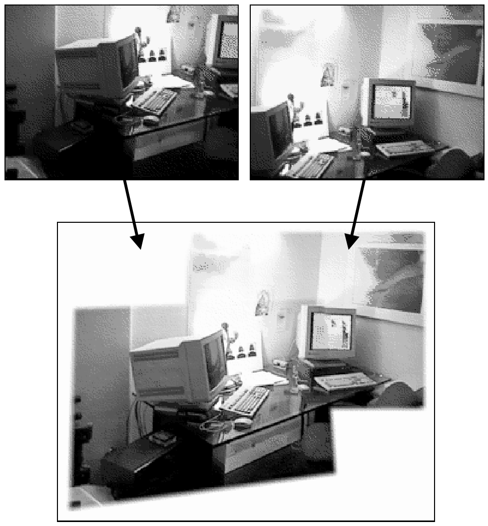
Global Flow

- Dominant Motion in the image
  - Motion of all points in the scene
  - Motion of most of the points in the scene
  - A Component of motion of all points in the scene

- Global Motion is caused by
  - Motion of sensor (Ego Motion)
  - Motion of a rigid scene

- Estimation of Global Motion can be used to
  - Video Mosaics
  - Image Alignment (Registration)
  - Removing Camera Jitter
  - Tracking (By neglecting camera motion)
  - Video Segmentation etc.

Global Flow



Application: Image Alignment

Global Flow

- Special Case of General Optical Flow Problem
- Can be solved by using Lucas Kanade algorithm.
- Specialized algorithms exist that perform better by further constraining the problem.

Motion Models

First we look for a parametric form of global flow vector.

Global Flow occurs because of 3D rigid motion of either the sensor or the scene.

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T = R_Z^\gamma R_Y^\beta R_X^\alpha \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T$$

3D Rigid Motion

$$= \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$\cos\theta \approx 1$, $\sin\theta \approx \theta$ (If θ is small)

Also neglecting the higher order terms

Motion Models

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

3D Rigid Motion

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \approx \left(\begin{bmatrix} 0 & -\gamma & \beta \\ \gamma & 0 & -\alpha \\ -\beta & \alpha & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} X' - X \\ Y' - Y \\ Z' - Z \end{bmatrix} \approx \begin{bmatrix} 0 & -\gamma & \beta \\ \gamma & 0 & -\alpha \\ -\beta & \alpha & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} \approx \begin{bmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_{T_x} \\ V_{T_Y} \\ V_{T_Z} \end{bmatrix}$$

$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} =$ Velocity Vector

$\begin{bmatrix} V_{T_X} \\ V_{T_Y} \\ V_{T_Z} \end{bmatrix} =$ Translational Component of Velocity

$\begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix} =$ Angular Velocity

Motion Models

Orthographic Projection

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} \approx \begin{bmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_{T_x} \\ V_{T_Y} \\ V_{T_Z} \end{bmatrix}$$

$$V_X = -\omega_Z Y + \omega_Y Z + V_{T_X}$$
$$V_Y = \omega_Z X - \omega_X Z + V_{T_Y}$$
$$V_Z = -\omega_Y X + \omega_X Y + V_{T_Z}$$

$$u = v_x = -\omega_Z y + \omega_Y Z + V_{T_X} \qquad x = X$$
$$v = v_y = \omega_Z x - \omega_X Z + V_{T_Y} \qquad y = Y$$

(Orthographic Projection)

Motion Models

Perspective Projection (Arbitrary Flow)

$$x = f\frac{X}{Z} \qquad u = v_x = f\,\frac{ZV_X - XV_Z}{Z^2} = f\,\frac{V_X}{Z} - \left(f\,\frac{X}{Z}\right)\frac{V_Z}{Z} = f\,\frac{V_X}{Z} - x\frac{V_Z}{Z}$$

$$y = f\frac{Y}{Z} \qquad v = v_y = f\,\frac{ZV_Y - YV_Z}{Z^2} = f\,\frac{V_Y}{Z} - \left(f\,\frac{Y}{Z}\right)\frac{V_Z}{Z} = f\,\frac{V_Y}{Z} - y\frac{V_Z}{Z}$$

$$V_X = -\omega_Z Y + \omega_Y Z + V_{T_X}$$
$$V_Y = \omega_Z X - \omega_X Z + V_{T_Y}$$
$$V_Z = -\omega_Y X + \omega_X Y + V_{T_Z}$$

$$u = v_x = \frac{V_{T_Z} x - V_{T_X} f}{Z} - \omega_Y f + \omega_Z y + \frac{\omega_X xy}{f} - \frac{\omega_Y x^2}{f}$$

$$v = v_y = \frac{V_{T_Z} y - V_{T_Y} f}{Z} + \omega_X f - \omega_Z x - \frac{\omega_Y xy}{f} + \frac{\omega_X y^2}{f}$$

Basic Equations of the Motion Field

Motion Models

Planar Scene + Orthographic Projection (Affine Flow)

$$u = v_x = -\omega_Z y + \omega_Y Z + V_{T_X}$$
$$v = v_y = \omega_Z x - \omega_X Z + V_{T_Y}$$

(Orthographic Projection +3D Rigid Motion)

$$Z = a + bX + cY$$ (Equation of Plane)

$$x = X$$
$$y = Y$$

$$u = v_x = -\omega_Z y + \omega_Y (a + bx + cy) + V_{T_X}$$
$$v = v_y = \omega_Z x - \omega_X (a + bx + cy) + V_{T_Y}$$

$$u = v_x = a_1 x + a_2 y + b_1$$
$$v = v_y = a_3 x + a_4 y + b_2$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + B$$

$$a_1 = b\omega_Y$$
$$a_2 = c\omega_Y - \omega_Z$$
$$b_1 = a\omega_Y + V_{T_X}$$
$$a_3 = \omega_Z - b\omega_X$$
$$a_2 = -c\omega_X$$
$$b_1 = -a\omega_X + V_{T_Y}$$

Motion Models

Planar Scene + Perspective Projection (Pseudo-Perspective)

$$u = v_x = \frac{V_{T_Z} x - V_{T_X} f}{Z} - \omega_Y f + \omega_Z y + \frac{\omega_X xy}{f} - \frac{\omega_Y x^2}{f}$$
$$v = v_y = \frac{V_{T_Z} y - V_{T_Y} f}{Z} + \omega_X f - \omega_Z x - \frac{\omega_Y xy}{f} + \frac{\omega_X y^2}{f}$$

(Perspective Flow)

$$Z = a + bX + cY$$ (Equation of Plane)

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}$$

$$\frac{1}{Z} = \frac{1}{a} - \frac{b}{af}x - \frac{c}{af}y$$

$$u = v_x = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy$$
$$v = v_y = a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2$$

Estimation of Global Flow

Assume Affine Flow:

$$\begin{bmatrix} u \\ v \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + B$$

$$u = v_x = a_1 x + a_2 y + b_1$$

$$v = v_y = a_3 x + a_4 y + b_2$$

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

$$\mathbf{U} = \mathbf{Xa}$$

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

Estimation of Global Flow

$$(\nabla I)^T \begin{bmatrix} u \\ v \end{bmatrix} + I_t = 0$$ (Optical Flow Constraint Equation)

$$(\nabla I)^T \mathbf{Xa} + I_t = 0$$

$$\sum_{\forall (x,y) \in I} \left[(\nabla I)^T \mathbf{Xa} + I_t \right]^2 = 0$$

$$\min \sum_{\forall (x,y) \in I} \left[(\nabla I)^T \mathbf{Xa} + I_t \right]^2 \Longrightarrow \left[\sum_{\forall (x,y) \in I} \mathbf{X}^T (\nabla I)(\nabla I)^T \mathbf{X} \right] \mathbf{a} = - \sum_{\forall (x,y) \in I} I_t \mathbf{X}^T \nabla I$$

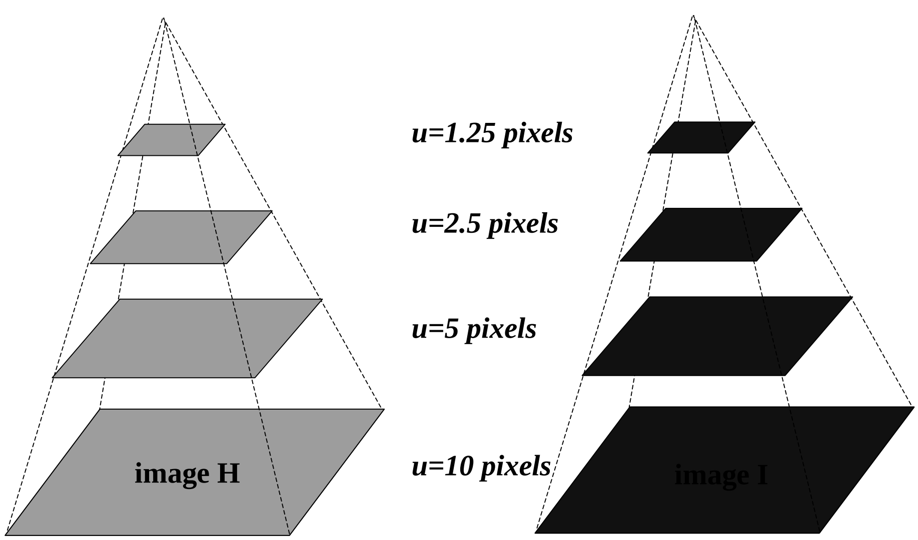
$$\mathbf{Aa} = \mathbf{B}$$

Iterative Refinement

Iterative Algorithm

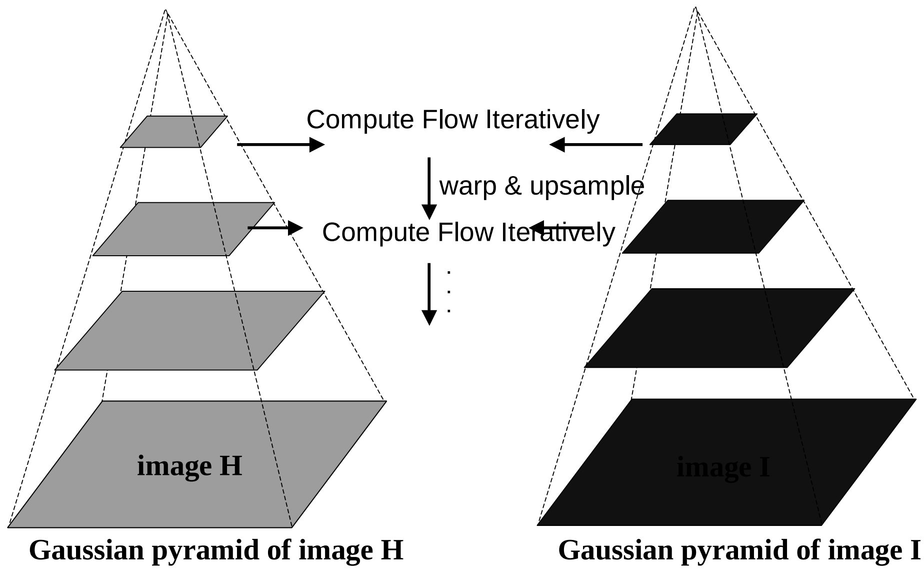
1. Estimate global flow by solving linear system Aa=B
2. Warp H towards I using the estimated flow
   - use image warping techniques (to be covered later)
3. Repeat until convergence or a fixed number of iterations

Coarse-to-fine global flow estimation

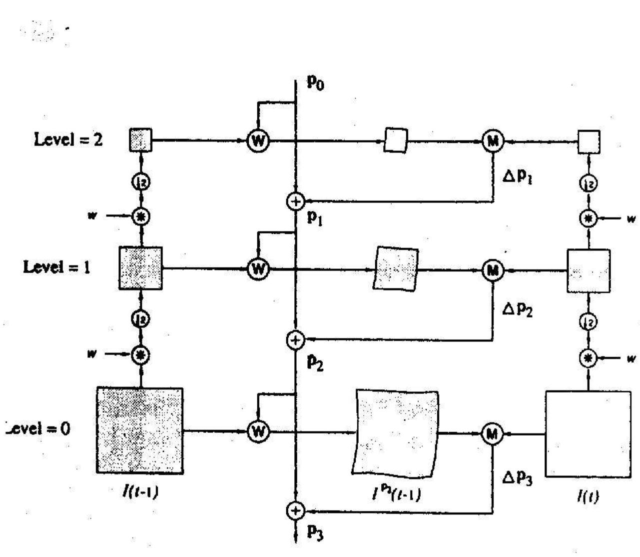


Gaussian pyramid of image H **Gaussian pyramid of image I**

Coarse-to-fine global flow estimation



Coarse-to-fine global motion estimation



Basic Components

- Pyramid Construction
- Motion Estimation
- Image Warping
- Coarse to Fine Refinement

Result of Global Motion Estimation

Image ‘t’

Image ‘t+1’

Affine Model

output

4 Pyramids Level
5 Iterations/Pyramid Level

Video Mosaic

Suggested Readings

- Chapter 8, Emanuele Trucco, Alessandro Verri, “Introductory Techniques for 3-D Computer Vision”
- James R. Bergen, P. Anandan, Keith J. Hanna, Rajesh Hingorani: “Hierarchical Model-Based Motion Estimation," ECCV 1992: 237-252