Optical Flow Ib

Guido Gerig

CS 6320, Spring 2012

(credits: slides modified from Marc Pollefeys
UNC Chapel Hill, Comp 256, and from K.H. Shafique,
UCSF, CAP5415, and from S. Narasimhan, CMU)

Materials

- Gary Bradski & Sebastian Thrun, Stanford CS223 http://robots.stanford.edu/cs223b/index.html
- S. Narasimhan, CMU: http://www.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/lec-16.ppt
- M. Pollefeys, ETH Zurich/UNC Chapel Hill: http://www.cs.unc.edu/Research/vision/comp256/vision10.ppt
- K.H. Shafique, UCSF: http://www.cs.ucf.edu/courses/cap6411/cap5415/
  - Lecture 18 (March 25, 2003), Slides: PDF/ PPT
- Jepson, Toronto: http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf
- Original paper Horn&Schunck 1981: http://www.csd.uwo.ca/faculty/beau/CS9645/PAPERS/Horn-Schunck.pdf
- MIT AI Memo Horn& Schunck 1980: http://people.csail.mit.edu/bkph/AIM/AIM-572.pdf
- Bahadir K. Gunturk, EE 7730 Image Analysis II
- Some slides and illustrations from L. Van Gool, T. Darell, B. Horn, Y. Weiss, P. Anandan, M. Black, K. Toyama

Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

Lucas & Kanade

•Assume single velocity for all pixels within a patch.
•Integrate over a patch.

- Similar to line fitting we have seen
  - Define an energy functional
    - Take derivatives equate it to 0
    - Rearrange and construct an observation matrix

$$E = \sum (uI_x + vI_y + I_t)^2$$

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

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

Lucas & Kanade

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

$$\sum uI_x^2 + \sum vI_xI_y + \sum I_xI_t = 0$$

$$u\sum I_x^2 + v\sum I_xI_y = -\sum I_xI_t$$

$$\begin{bmatrix} \sum I_x^2 & \sum I_xI_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\sum I_xI_t$$

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

$$\sum uI_xI_y + \sum vI_y^2 + \sum I_yI_t = 0$$

$$u\sum I_xI_y + v\sum I_y^2 = -\sum I_yI_t$$

$$\begin{bmatrix} \sum I_xI_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\sum I_yI_t$$

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

Lucas & Kanade

$$Au = B \qquad A^{-1}Au = A^{-1}B \qquad Iu = A^{-1}B \qquad u = A^{-1}B$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

$$\boxed{\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\sum I_x^2 \sum I_y^2 - \left(\sum I_x I_y\right)^2} \begin{bmatrix} \sum I_y^2 & -\sum I_x I_y \\ -\sum I_x I_y & \sum I_x^2 \end{bmatrix} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}}$$

Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

$$E(u,v) = \sum_{x,y\in\Omega} \left(I_x(x,y)u + I_y(x,y)v + I_t \right)^2$$

$$\frac{dE(u,v)}{du} = \sum 2I_x \left(I_x u + I_y v + I_t \right) = 0$$

Solve with:

$$\frac{dE(u,v)}{dv} = \sum 2I_y \left(I_x u + I_y v + I_t \right) = 0$$

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

On the LHS: sum of the 2x2 outer product tensor of the gradient vector

$$\left(\sum \nabla I \nabla I^T \right) \vec{U} = -\sum \nabla I I_t$$

Lucas-Kanade: Singularities and the Aperture Problem

Let $M = \sum (\nabla I)(\nabla I)^T$ and $b = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$

- Algorithm: At each pixel compute U by solving $MU = b$

- M is singular if all gradient vectors point in the same direction
  - -- e.g., along an edge
  - -- of course, trivially singular if the summation is over a single pixel
  - -- i.e., only *normal flow* is available (aperture problem)

- Corners and textured areas are OK

Discussion

- Horn-Schunck: Add smoothness constraint.

$$\int_D (\nabla I \cdot \vec{v} + I_t)^2 + \lambda^2 \left[\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 \right] dx dy$$

- Lucas-Kanade: Provide constraint by minimizing over local neighborhood:

$$\sum_{x,y \in \Omega} W^2(x,y)[\nabla I(x,y,t) \cdot \vec{v} + I_t(x,y,t)]^2$$

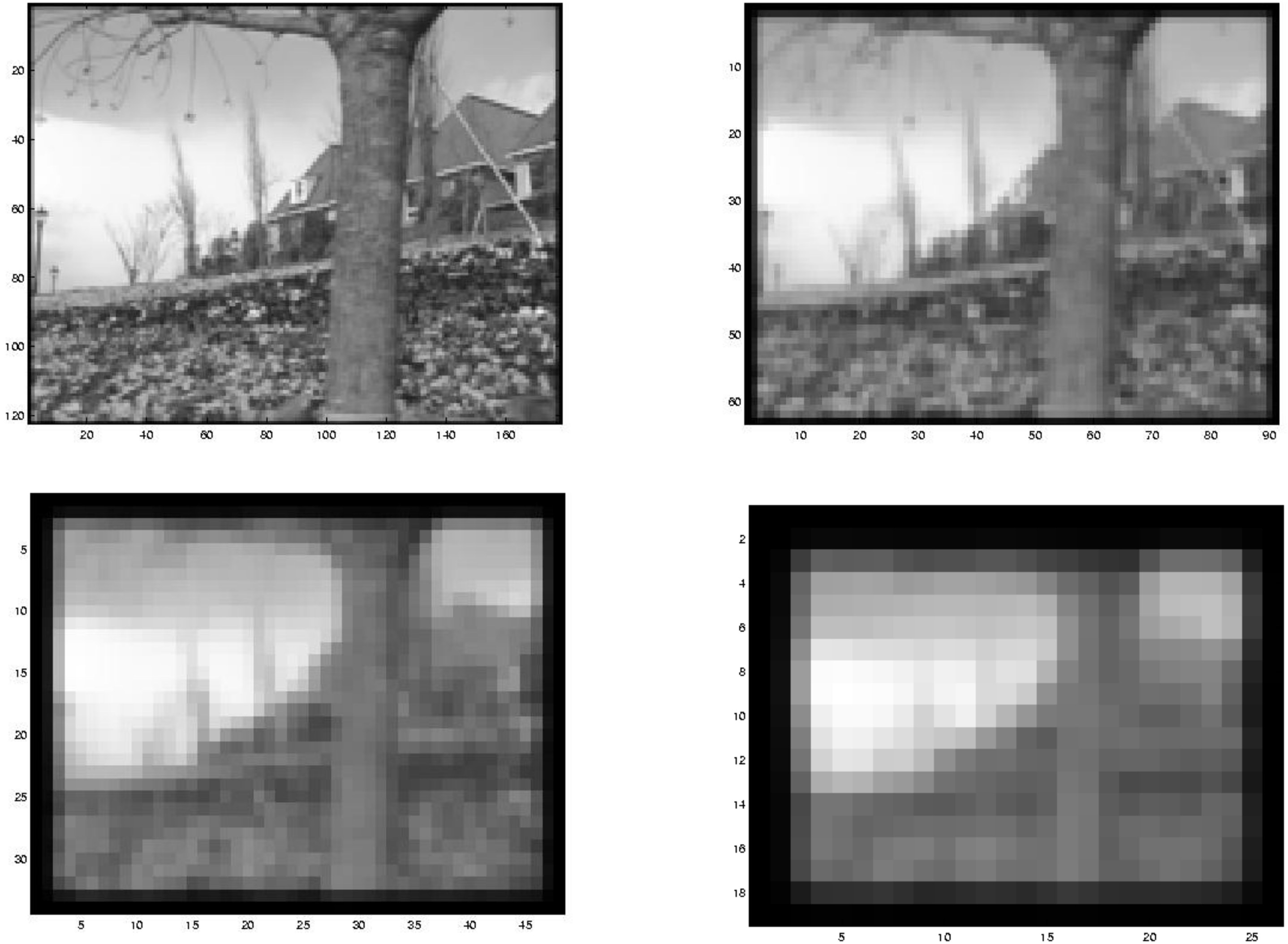
- Horn-Schunck and Lucas-Kanade optical methods work only for small motion.
- If object moves faster, the brightness changes rapidly, derivative masks fail to estimate spatiotemporal derivatives.
- Pyramids can be used to compute large optical flow vectors.

Iterative Refinement

- Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation
- Warp one image toward the other using the estimated flow field

  (easier said than done)
- Refine estimate by repeating the process

Reduce the Resolution!



Optical Flow

- Brightness Constancy
- The Aperture problem
- Regularization
- Lucas-Kanade
- Coarse-to-fine
- Parametric motion models
- Direct depth
- SSD tracking
- Robust flow
- Bayesian flow

Limits of the (local) gradient method

1. Fails when intensity structure within window is poor
2. Fails when the displacement is large (typical operating range is motion of 1 pixel per iteration!)
   - *Linearization of brightness is suitable only for small displacements*

Also, brightness is not strictly constant in images

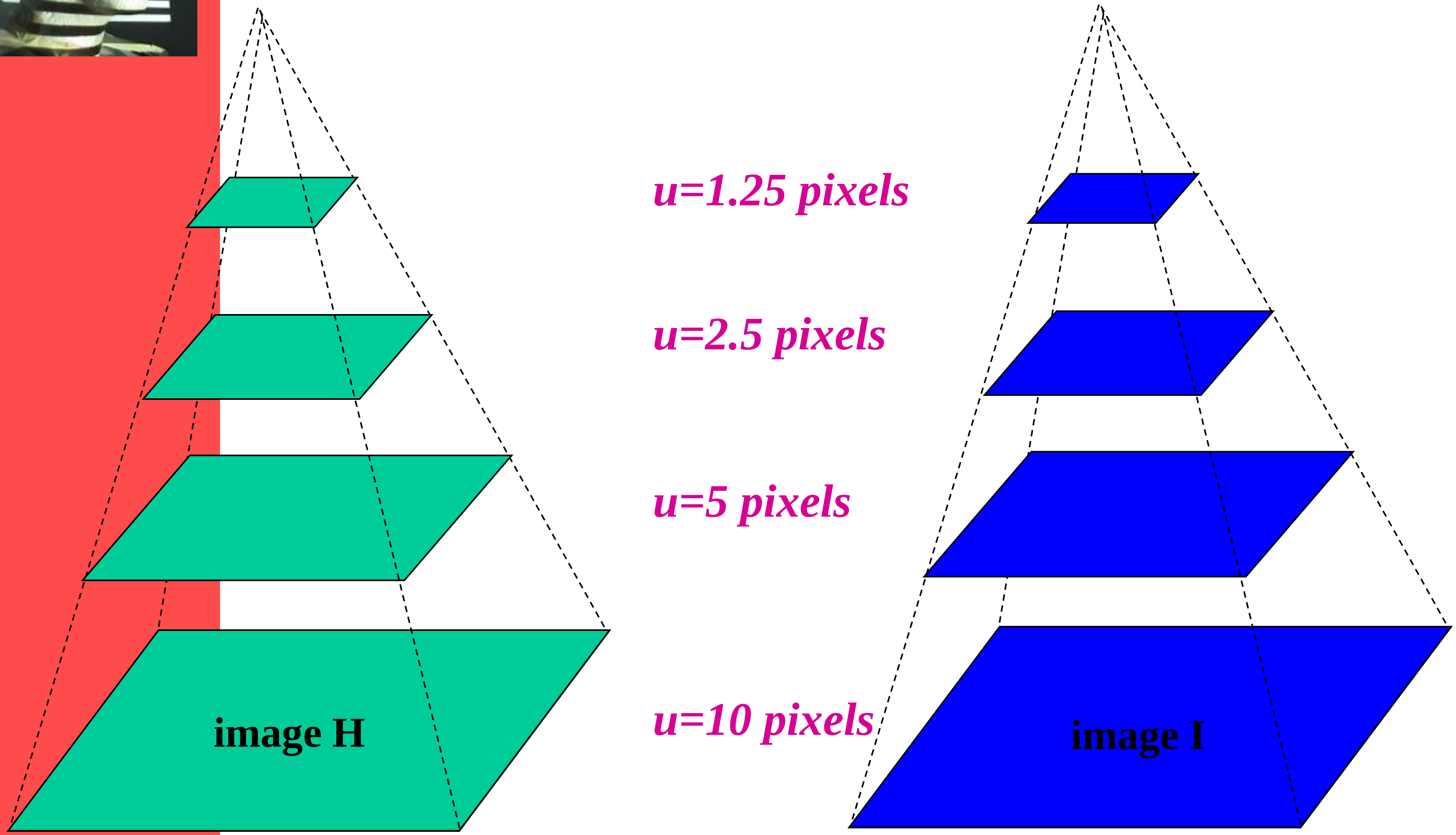
- *actually less problematic than it appears, since we can pre-filter images to make them look similar*

Results

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?

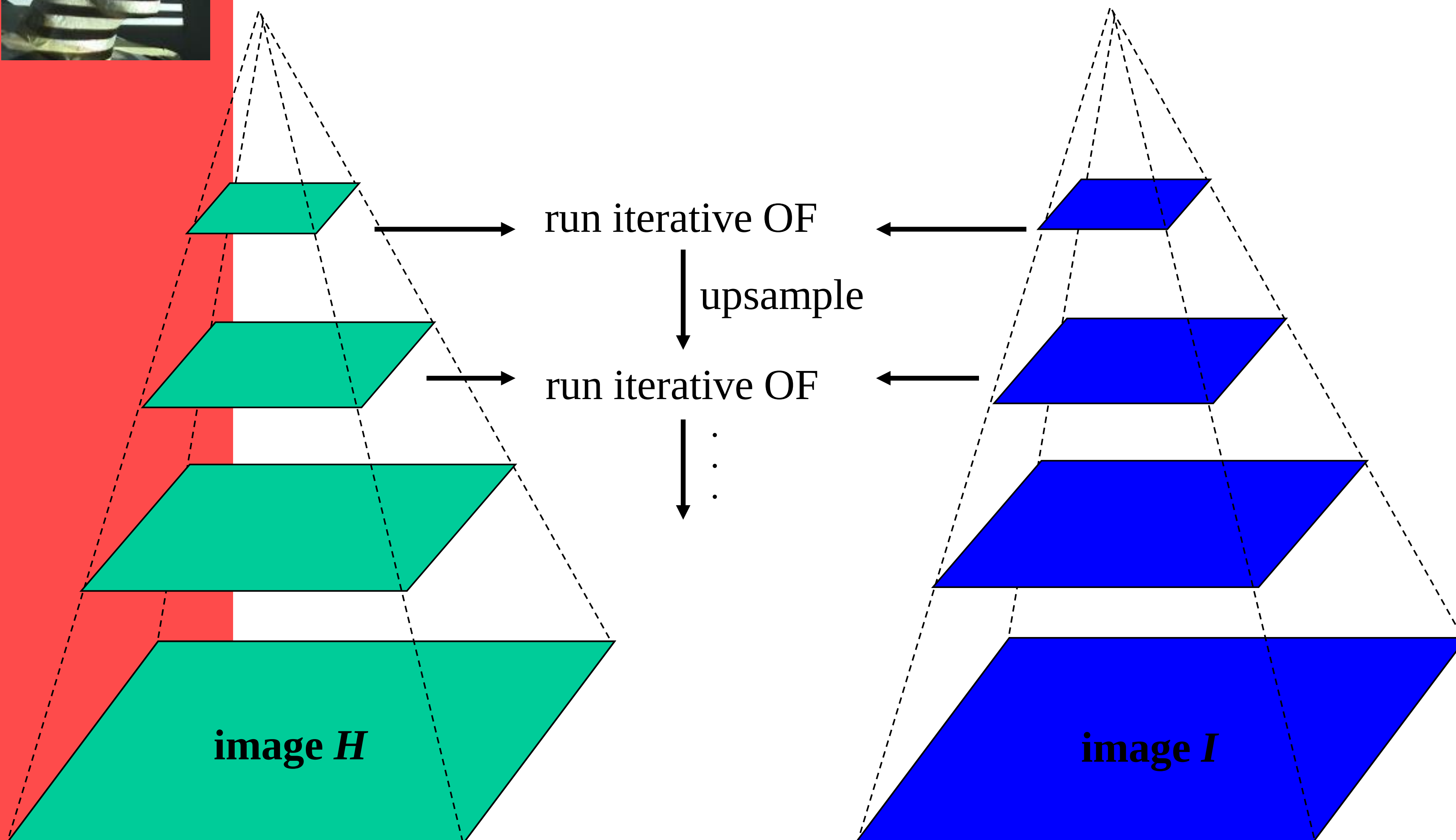
Coarse-to-fine Optical Flow Estimation



Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

Coarse-to-fine Optical Flow Estimation

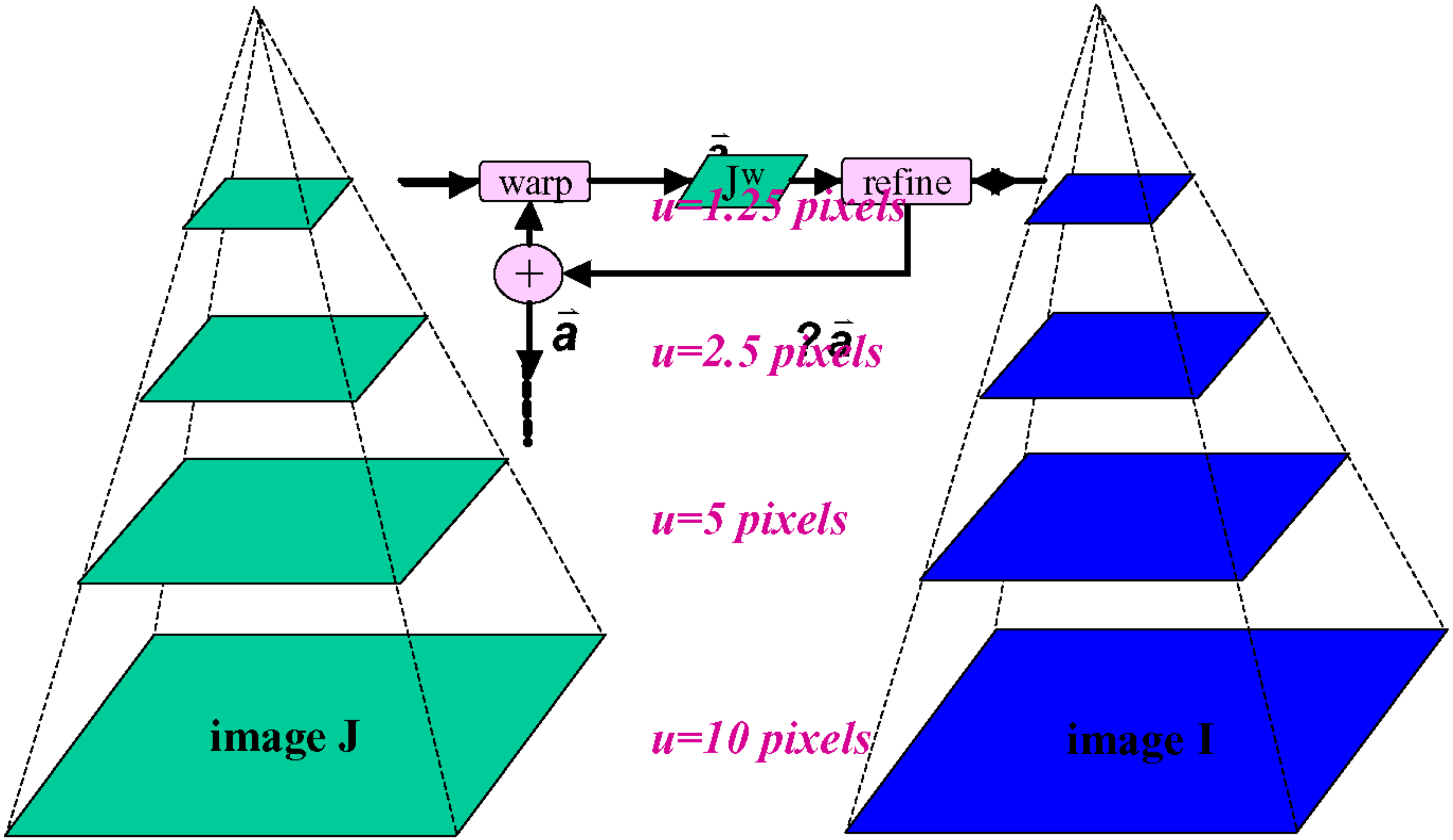


Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

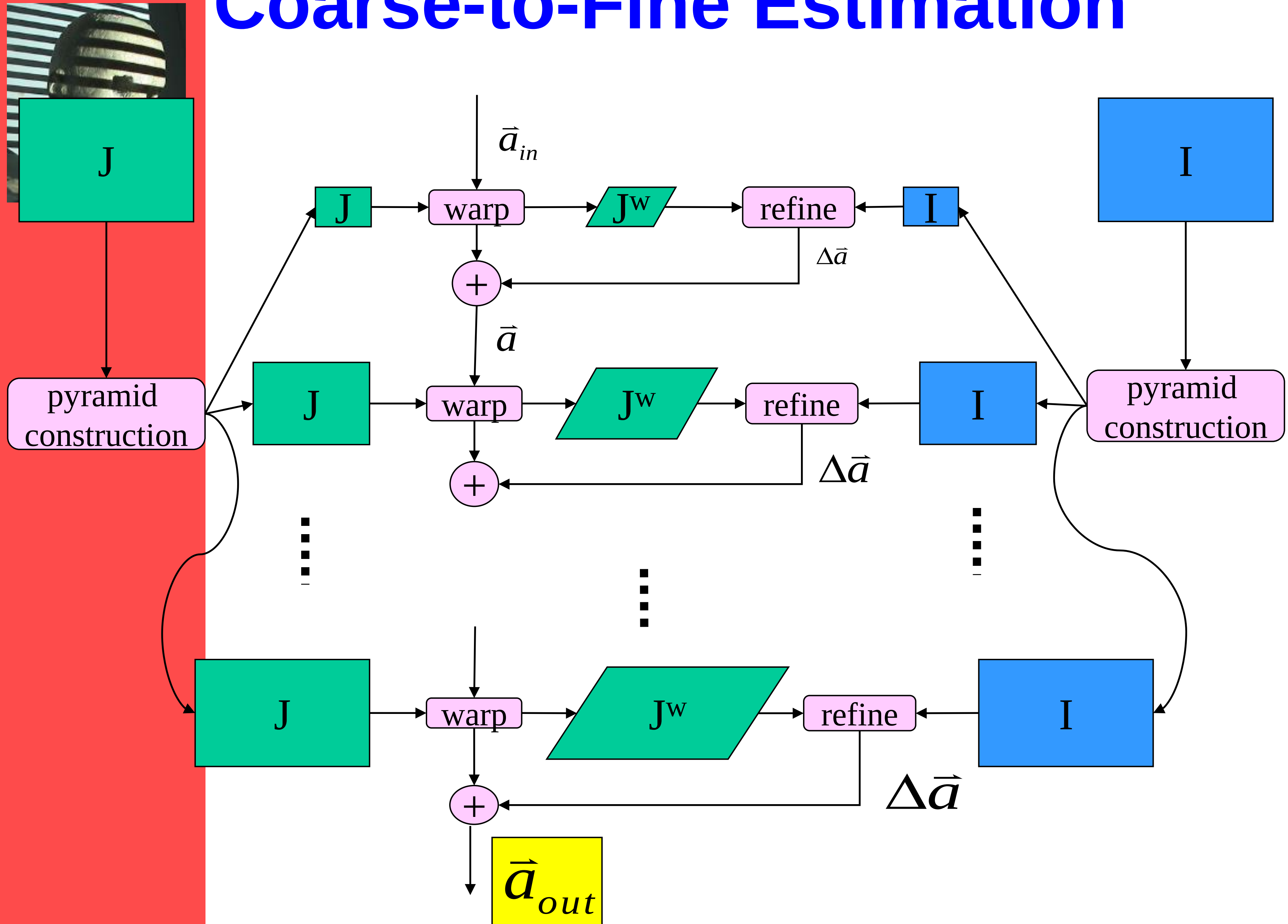
Coarse-to-Fine Estimation

$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0}$$ ==> **small *u* and *v* ...**

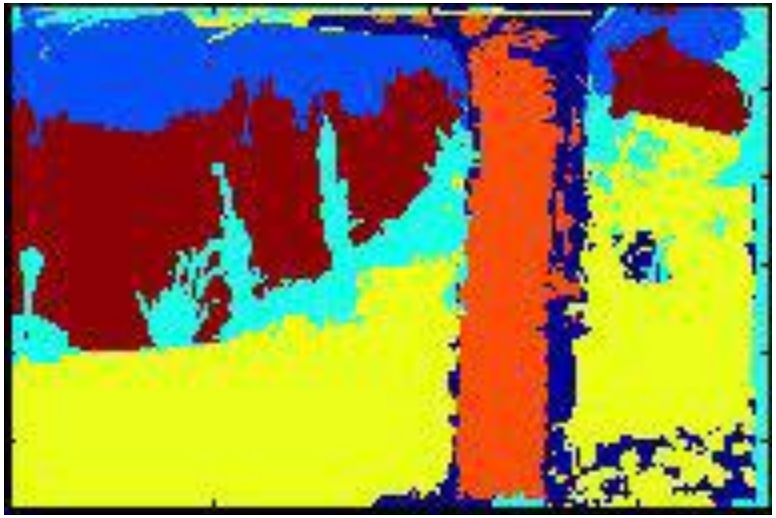


Pyramid of image J **Pyramid of image I**

Coarse-to-Fine Estimation



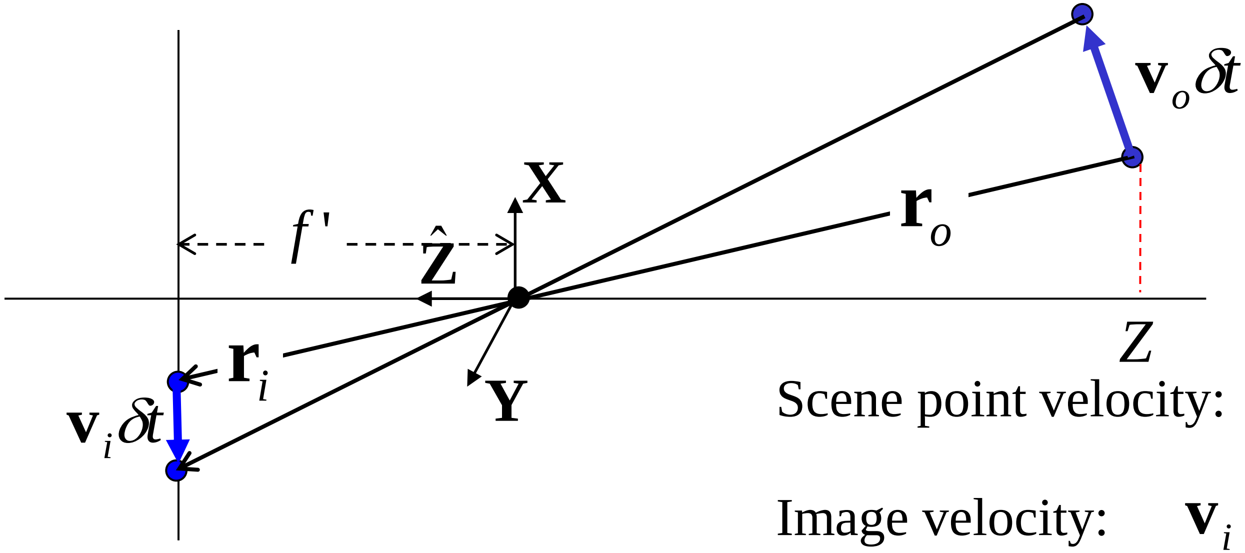
Video Segmentation



Next:
Motion Field
Structure from Motion

Motion Field

- Image velocity of a point moving in the scene



Scene point velocity: $\mathbf{v}_o = \dfrac{d\mathbf{r}_o}{dt}$

Image velocity: $\mathbf{v}_i = \dfrac{d\mathbf{r}_i}{dt}$

Perspective projection: $\dfrac{1}{f'}\mathbf{r}_i = \dfrac{\mathbf{r}_o}{\mathbf{r}_o \cdot \hat{\mathbf{Z}}} = \dfrac{\mathbf{r}_o}{Z}$

Motion field

$$\mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} = f' \frac{(\mathbf{r}_o \cdot \mathbf{Z})\mathbf{v}_o - (\mathbf{v}_o \cdot \mathbf{Z})\mathbf{r}_o}{(\mathbf{r}_o \cdot \mathbf{Z})^2} = f' \frac{(\mathbf{r}_o \times \mathbf{v}_o) \times \mathbf{Z}}{(\mathbf{r}_o \cdot \mathbf{Z})^2}$$