CS4243
Computer Vision
&
Pattern Recognition

Change Detection,
Corner Detectors
&
Motion Tracking

Change Detection

- Detects changes in two video frames.
- Straightforward method:
  - Compute difference between corresponding pixels:

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

  - $I(x, y, t)$: intensity / colour at (*x*, *y*) in frame *t*.
  - If $D_t(x, y)$ > threshold, declare "changed"

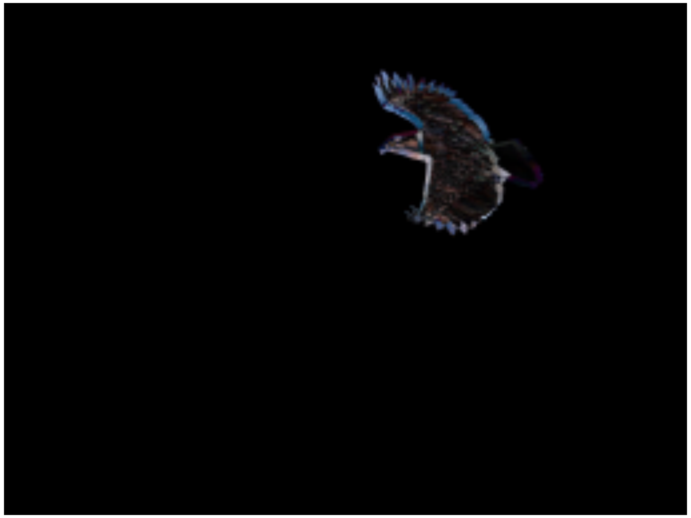
Any difference?

No

Any difference?

Yes, illumination change

Any difference?



Yes,
position change

Change Detection

- Can detect
  - Illumination change
  - Position change
  - Illumination and position change
- But, cannot distinguish between them.
- Need to detect and measure position change.

Motion Tracking

- Two approaches
  - feature based
  - intensity gradient-based

Feature based Tracking

Feature-based Motion Tracking

- Look for distinct features (eg. eagle's wing tip) that change positions.

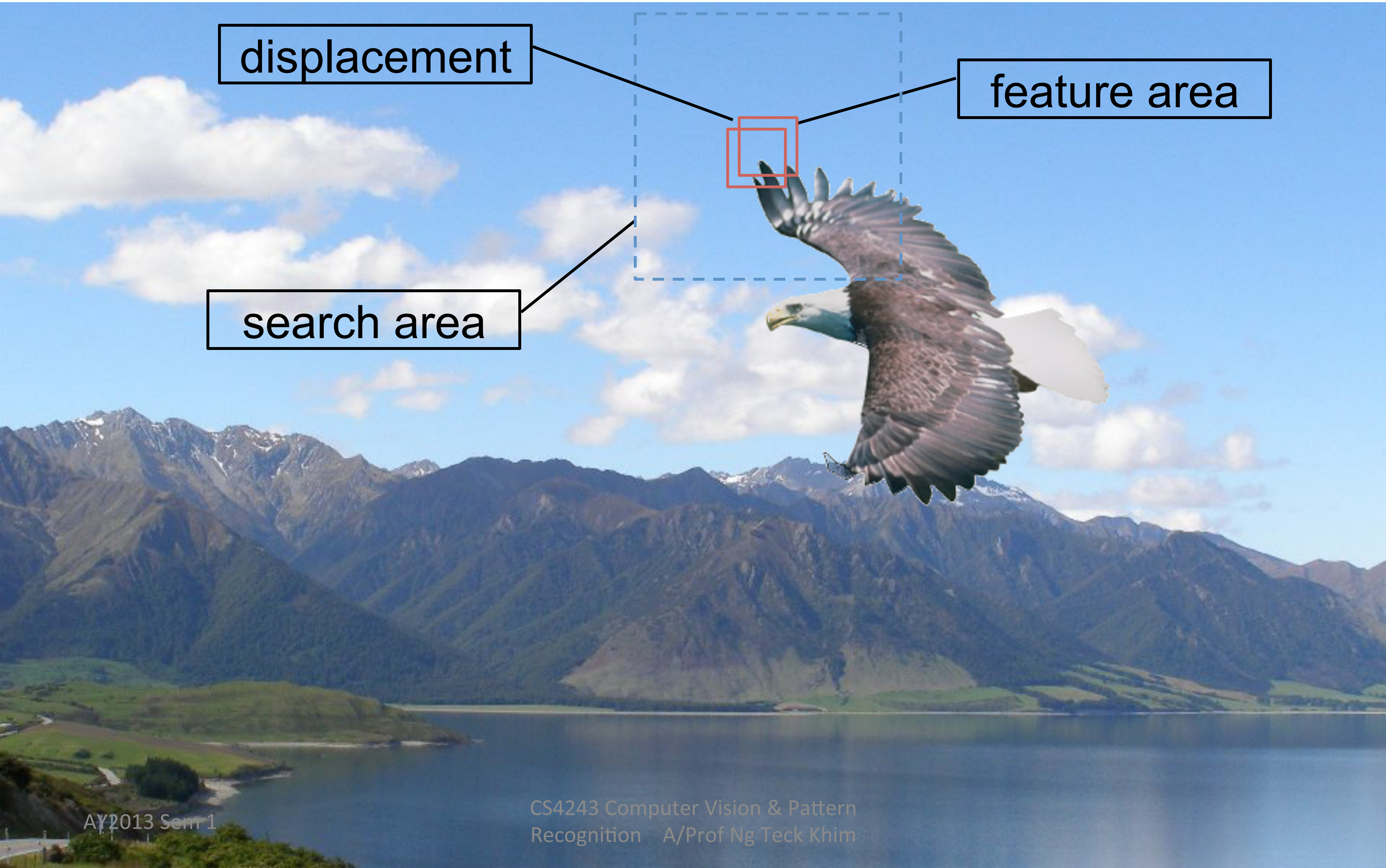
Basic Ideas

1. Look for distinct features in current frame.

   eg. wing tip of eagle

1. For each feature
   – Search for matching feature in next frame.
   – Difference in positions → displacement

Basic Ideas



What is a good feature?

“CORNERS”

- Harris corner detector (most popular)

- Tomasi corner detector (same rationale as Harris but with direct computation of eigenvalues)

$$\frac{1}{N}\sum_{u}\sum_{v}\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$= \frac{1}{N}\sum_{u}\sum_{v}\begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}$$

where

N : total number of pixels in window of interest

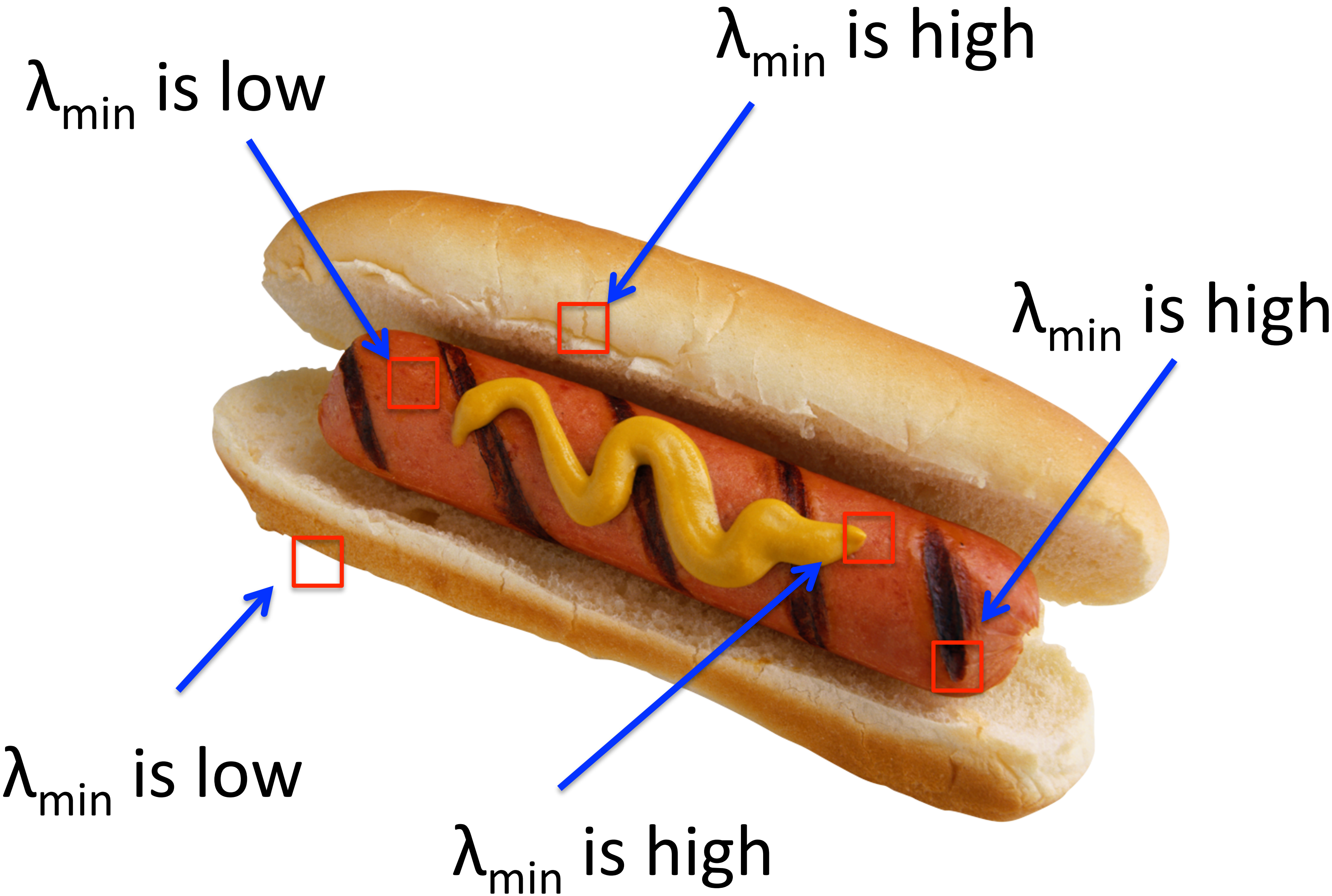
u : horizontal index of pixel in window of interest

v : vertical index of pixel in window of interest

$$\frac{1}{N}\sum_{u}\sum_{v}\begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}$$

Let the eigenvalues of the above matrix be λ_{max} and λ_{min}

The greater λ_{min} is, the more "cornerness" is the feature


λmin is low
λmin is high
λmin is high
λmin is low
λmin is high

Feature Descriptors

- SIFT
- SURF
- GLOH, etc.

Intensity Gradient based Motion Tracking

Gradient-based Motion Tracking

- Two basic assumptions
  - Intensity changes smoothly with position.
  - Brightness constancy (Intensity of object doesn't change over time.)
- Suppose an object is in motion.
  - position changed by (dx, dy) over time dt.
  - Then, from 2nd assumption:

$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \cdots$$

– Omit higher order terms and divide by d*t*

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

– Denote

$$u = \frac{dx}{dt} \ , \ v = \frac{dy}{dt} \ , \ I_x = \frac{\partial I}{\partial x} \ , \ I_y = \frac{\partial I}{\partial y} \ , \ I_t = \frac{\partial I}{\partial t} \ .$$

– Then,

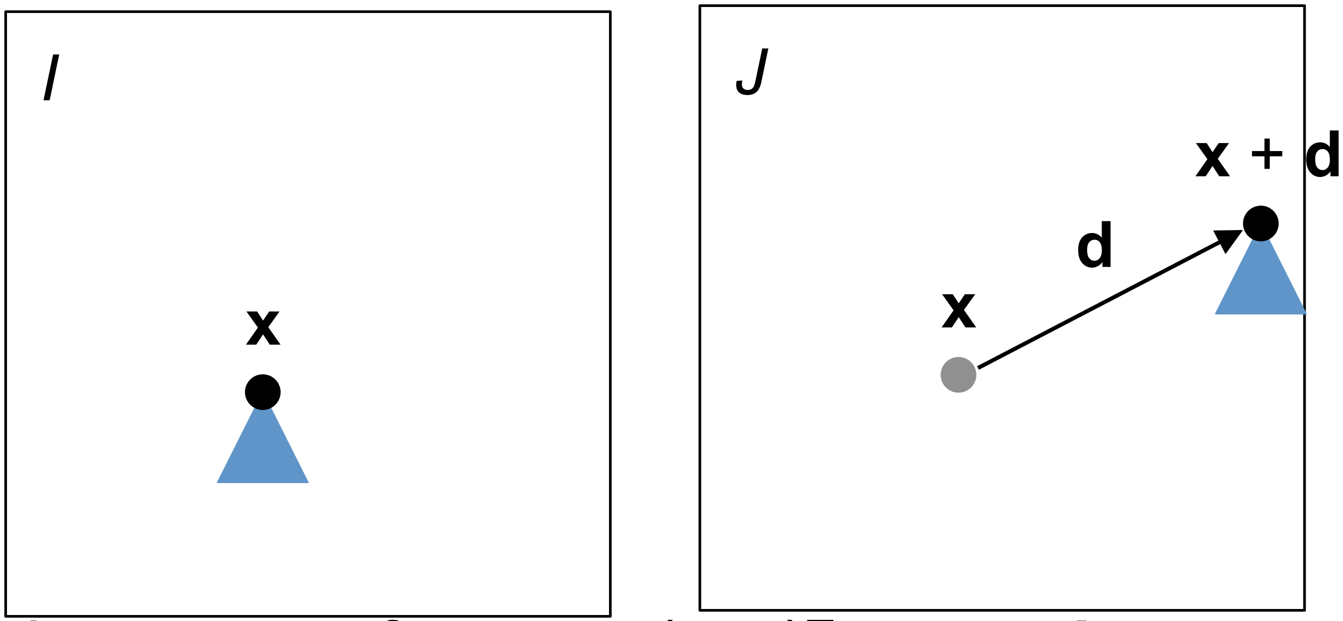
$$I_x u + I_y v + I_t = 0$$

– *u*, *v* are unknown.

– 2 unknowns, 1 eqn -> cannot solve

Lucas-Kanade Method

- Consider two consecutive image frames *I* and *J*:



  - Object moves from $\mathbf{x} = (x, y)^T$ to $\mathbf{x} + \mathbf{d}$.
  - $\mathbf{d} = (u, v)^T$

- So,

$$J(\mathbf{x}+\mathbf{d}) = I(\mathbf{x})$$

- Or

$$J(\mathbf{x}) = I(\mathbf{x}-\mathbf{d})$$

- Due to noise, there's an error at position **x**:

$$e(\mathbf{x}) = I(\mathbf{x}-\mathbf{d}) - J(\mathbf{x})$$

- Sum error over small window *W* at position **x**:

$$E(\mathbf{x}) = \sum_{\mathbf{x}\in W} w(\mathbf{x})\left[I(\mathbf{x}-\mathbf{d}) - J(\mathbf{x})\right]^2$$

Similar to template matching

weight

- If E is small, patterns in I and J match well.
- So, find $\mathbf{d}$ that minimises E:
  - Set $\partial E / \partial \mathbf{d} = 0$, compute $\mathbf{d}$ that minimises E.
  - First, expand $I(\mathbf{x} - \mathbf{d})$ by Taylor's series

$$I(x-u, y-v) = I(x, y) - u\, I_x(x, y) - v\, I_y(x, y) + \cdots$$

  - Omit higher order terms:

$$I(x-u, y-v) = I(x, y) - u\, I_x(x, y) - v\, I_y(x, y)$$

  - Write in matrix form:

$$I(\mathbf{x} - \mathbf{d}) = I(\mathbf{x}) - \mathbf{d}^\top \mathbf{g}(\mathbf{x}) \qquad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} I_x(\mathbf{x}) \\ I_y(\mathbf{x}) \end{bmatrix}$$

Intensity gradient

– Now, error *E* at position **x** is:

$$E(\mathbf{x}) = \sum_{\mathbf{x} \in W} w(\mathbf{x}) \left[I(\mathbf{x}) - J(\mathbf{x}) - \mathbf{d}^\top \mathbf{g}(\mathbf{x}) \right]^2$$

– Now, differentiate *E* with respect to **d**

$$\frac{\partial E}{\partial \mathbf{d}} = -2 \sum_{\mathbf{x} \in W} w(\mathbf{x}) \left[I(\mathbf{x}) - J(\mathbf{x}) - \mathbf{d}^\top \mathbf{g}(\mathbf{x}) \right] \mathbf{g}(\mathbf{x})$$

– Setting $\partial E \,/\, \partial \mathbf{d} = 0$ gives

the only unknown

b

z

$$\boxed{\sum_{\mathbf{x} \in W} w(\mathbf{x}) \left[I(\mathbf{x}) - J(\mathbf{x}) \right] \mathbf{g}(\mathbf{x})}$$

$$= \sum_{\mathbf{x} \in W} w(\mathbf{x})\, \mathbf{d}^\top \mathbf{g}(\mathbf{x})\, \mathbf{g}(\mathbf{x}) = \boxed{\sum_{\mathbf{x} \in W} w(\mathbf{x})\, \mathbf{g}(\mathbf{x})\, \mathbf{g}^\top(\mathbf{x})\, \mathbf{d}}$$

– So, we get

$$\mathbf{Z}\,\mathbf{d} = \mathbf{b}$$

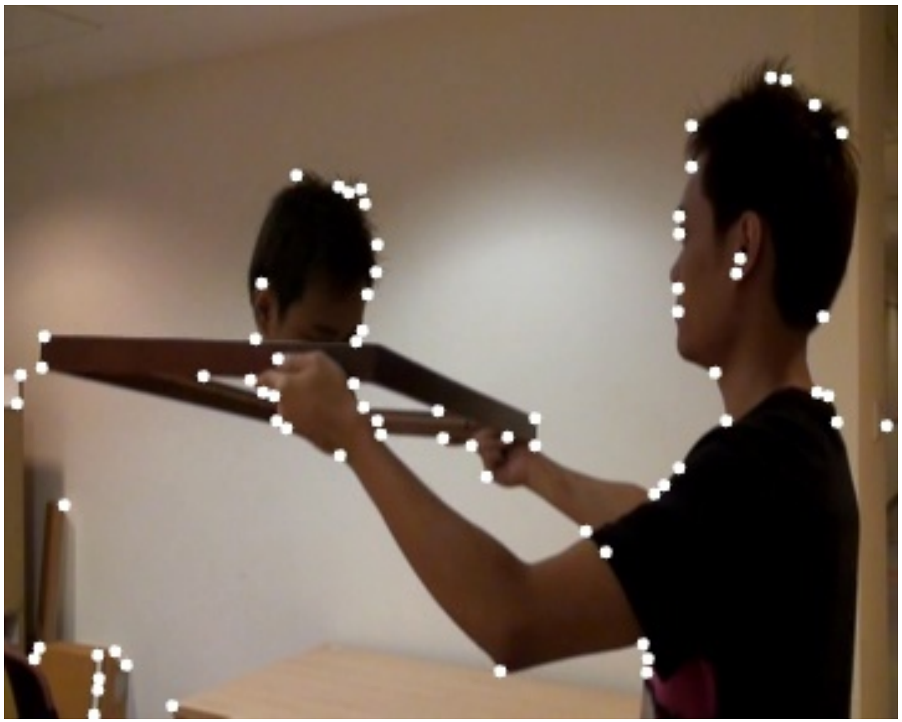
$$\mathbf{Z} = \begin{bmatrix} \sum_{\mathbf{x}\in W} w\, I_x^2 & \sum_{\mathbf{x}\in W} w\, I_x\, I_y \\ \sum_{\mathbf{x}\in W} w\, I_x\, I_y & \sum_{\mathbf{x}\in W} w\, I_y^2 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \sum_{\mathbf{x}\in W} w\,(I - J)\, I_x \\ \sum_{\mathbf{x}\in W} w\,(I - J)\, I_y \end{bmatrix}$$

– 2 unknowns, 2 equations. Can solve for **d** = (*u*, *v*).

Lucas-Kanade + Tomasi/Harris

- Lucas-Kanade algorithm is often used with Harris/Tomasi's corner detectors
  – Apply corner detectors to detect good features.
  – Apply LK method to compute **d** for each pixel.
  – Accept **d** only for good features.

Example



Can you spot tracking errors?

Constraints

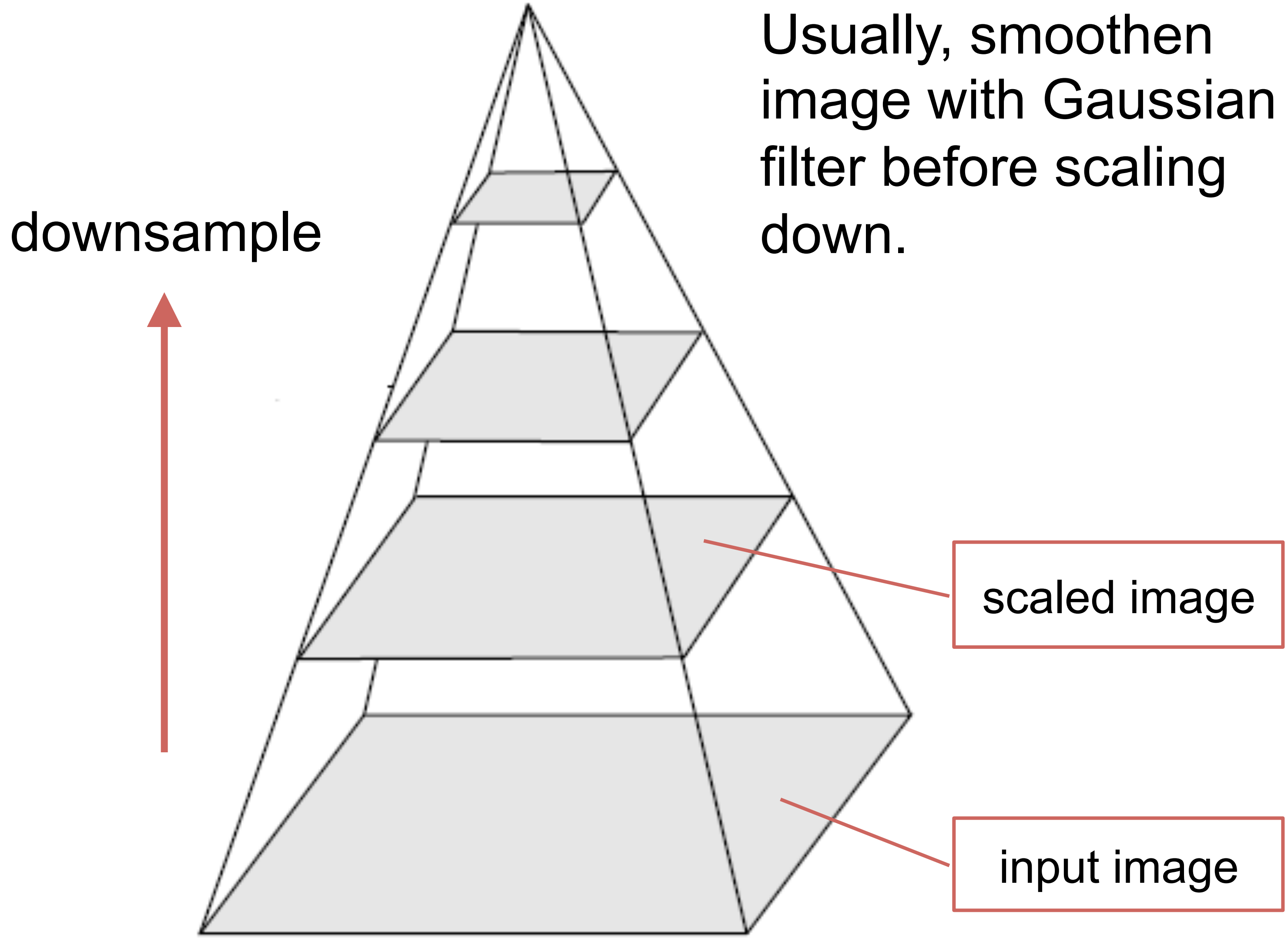
- Math of LK tracker assumes **d** is small.
- In implementation, *W* is also small.
- LK tracker is good only for small displacement.

How to handle large displacement?

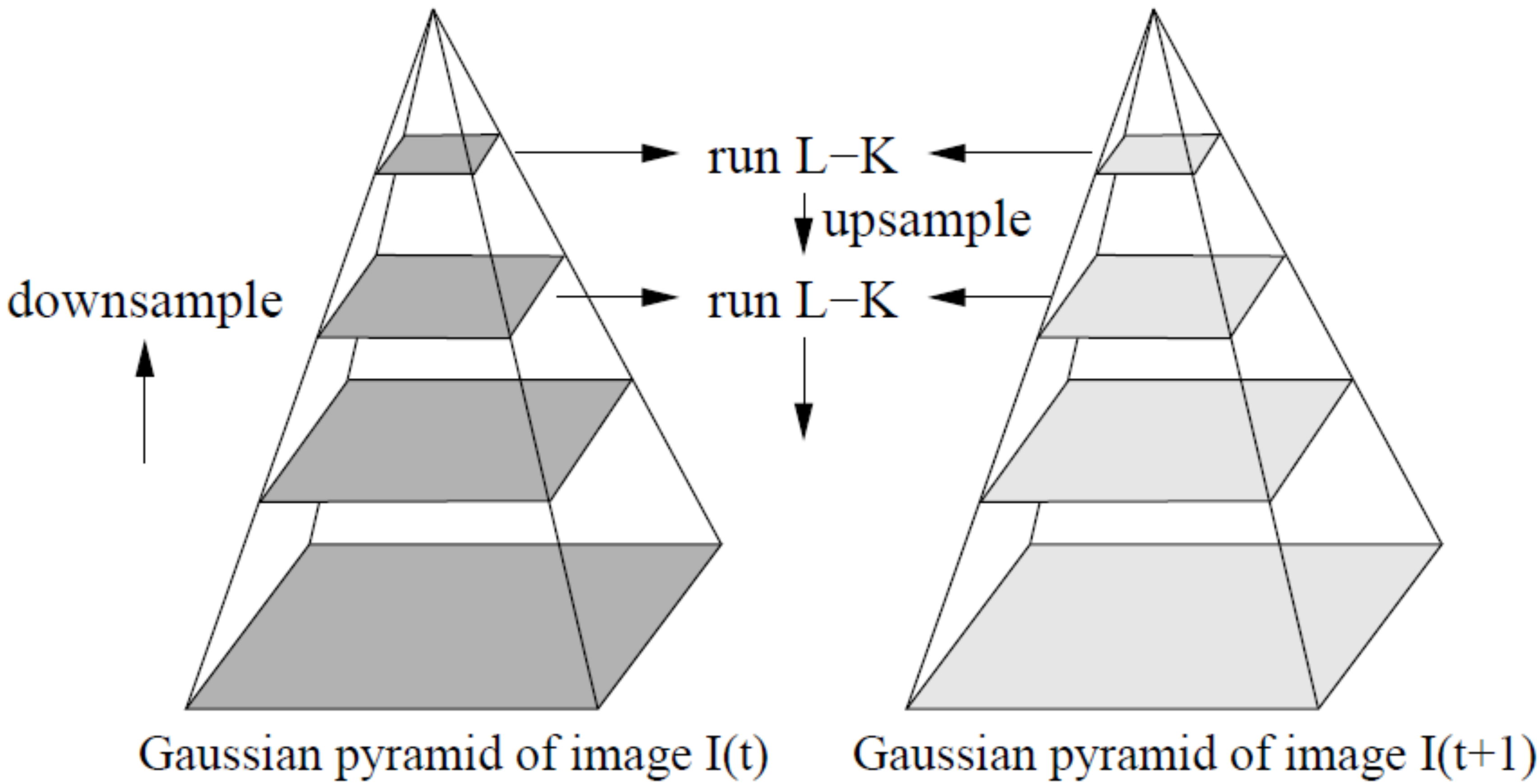
- What if we scale down images?
  - Displacements are smaller!

Image Pyramid

Usually, smoothen image with Gaussian filter before scaling down.

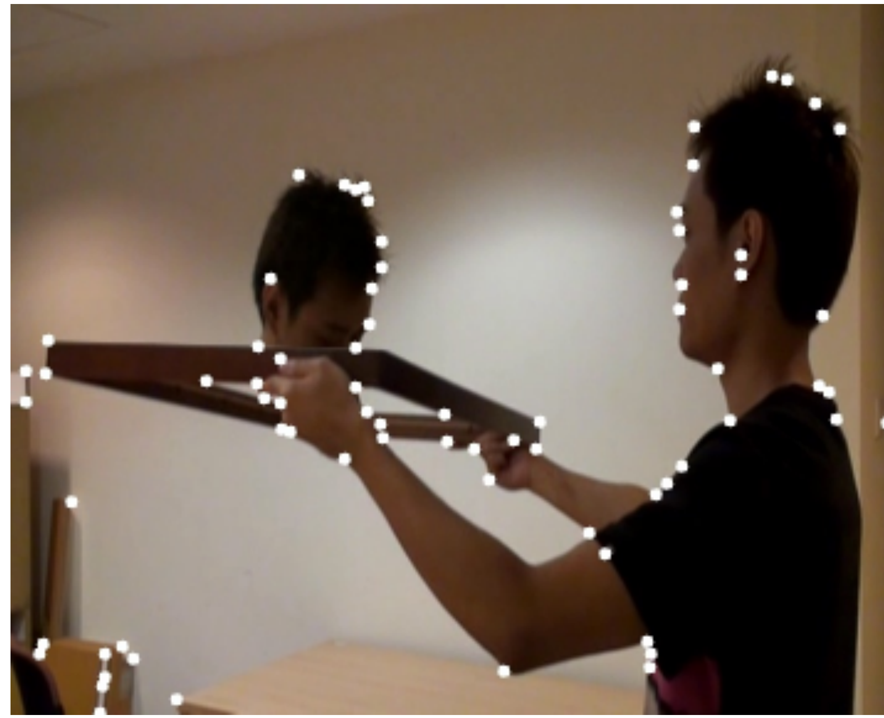


LK Tracking with Image Pyramid



- Construct image pyramids.
- Apply LK tracker to low-resolution images.
- Propagate results to higher-resolution images.
- Apply LK tracker to higher-resolution images.

Example



Tracking results are more accurate (can handle larger displacements).

Summary

- Efficient algorithm, no explicit search.
- Has aperture problem (not unique to LK); track good features only
- LK tracker alone can't track large displacement.
- Use LK + image pyramid for large displacement.

Software

- OpenCV supports LK (cv only) and LK with pyramid (cv and cv2)
- [Bir] offers LK with Tomasi's features & pyramid.

Appendix

- Calculation of I_x, I_y, I_t
  - Use finite difference method
  - Forward difference

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

  - Backward difference

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

Further Readings

- Lucas-Kanade tracking with pyramid: [BK08] Chapter 10.
- Optical flow: [Sze10] Section 8.4.
- Hierarchical motion estimation (with image pyramid): [Sze10] Section 8.1.1.

References

- [Bir] S. Birchfield. *KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker*. http://vision.stanford.edu/~birch/klt/.
- [BK08] Bradski and Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.
- [LK81] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *In Proceedings of 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [ST94] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
- [Sze10] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

References

- [TK91] C. Tomasi and T. Kanade. *Detection and tracking of point features*. Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, 1991.