Computational Photography

Prof. Feng Liu

Spring 2022

http://www.cs.pdx.edu/~fliu/courses/cs510/

04/21/2022

Last Time

- Re-lighting
  - HDR

Today

- Panorama
  - Overview
  - Feature detection

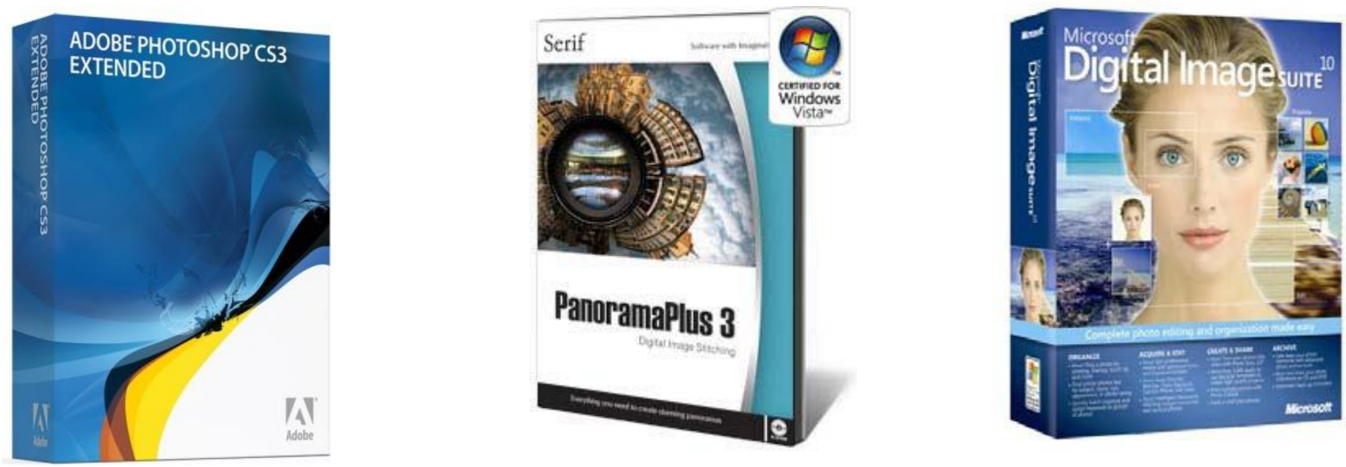
Panorama Building: History

Along the River During Ching Ming Festival
by Z.D Zhang (1085-1145)

San Francisco from Rincon Hill, 1851,
by Martin Behrmanx

Panorama Building: A Concise History

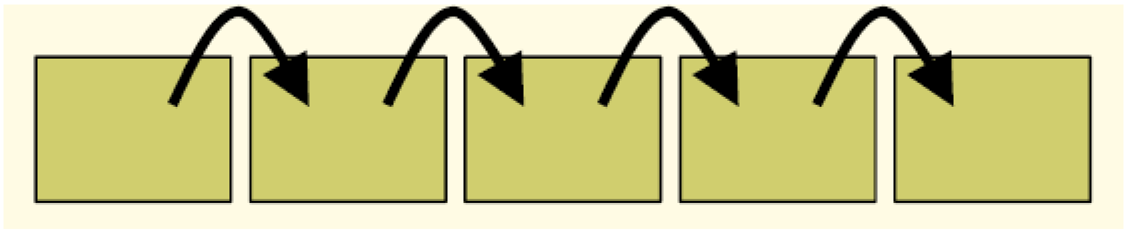
- The state of the art and practice is good at assembling images into panoramas



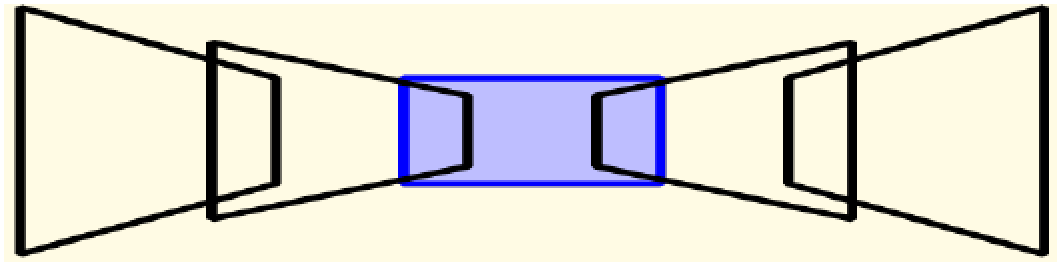
- Mid 90s -Commercial Players (e.g. QuicktimeVR)
- Late 90s -Robust stitchers (in research)
- Early 00s -Consumer stitching common
- Mid 00s -Automation

Stitching Recipe

- Align pairs of images



- Align all to a common frame

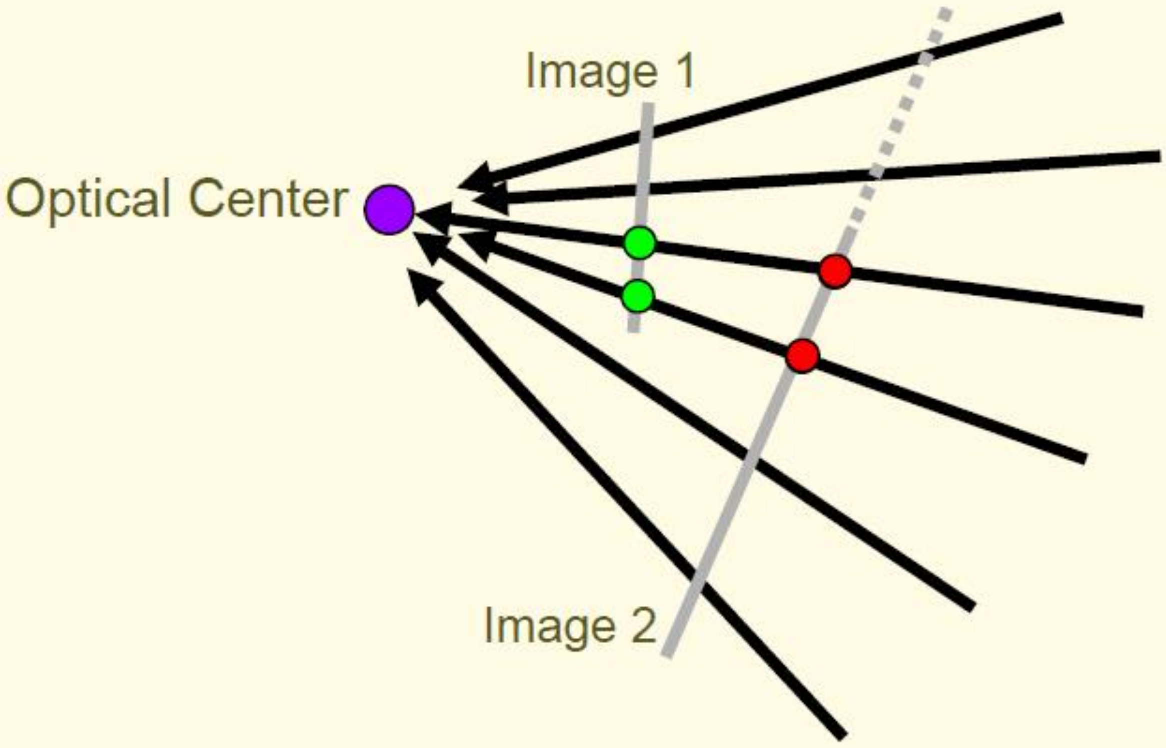


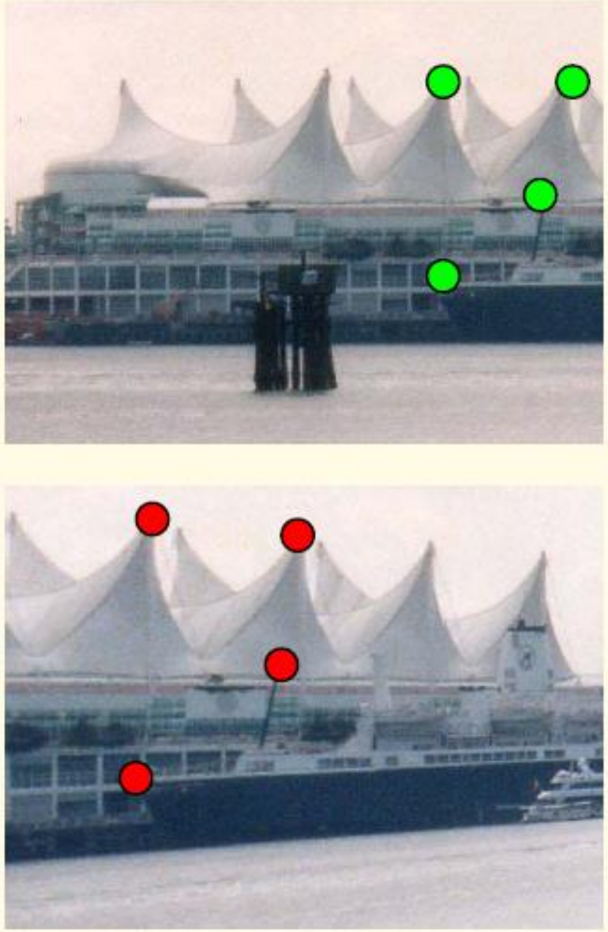
- Adjust (Global) & Blend

Stitching Images Together

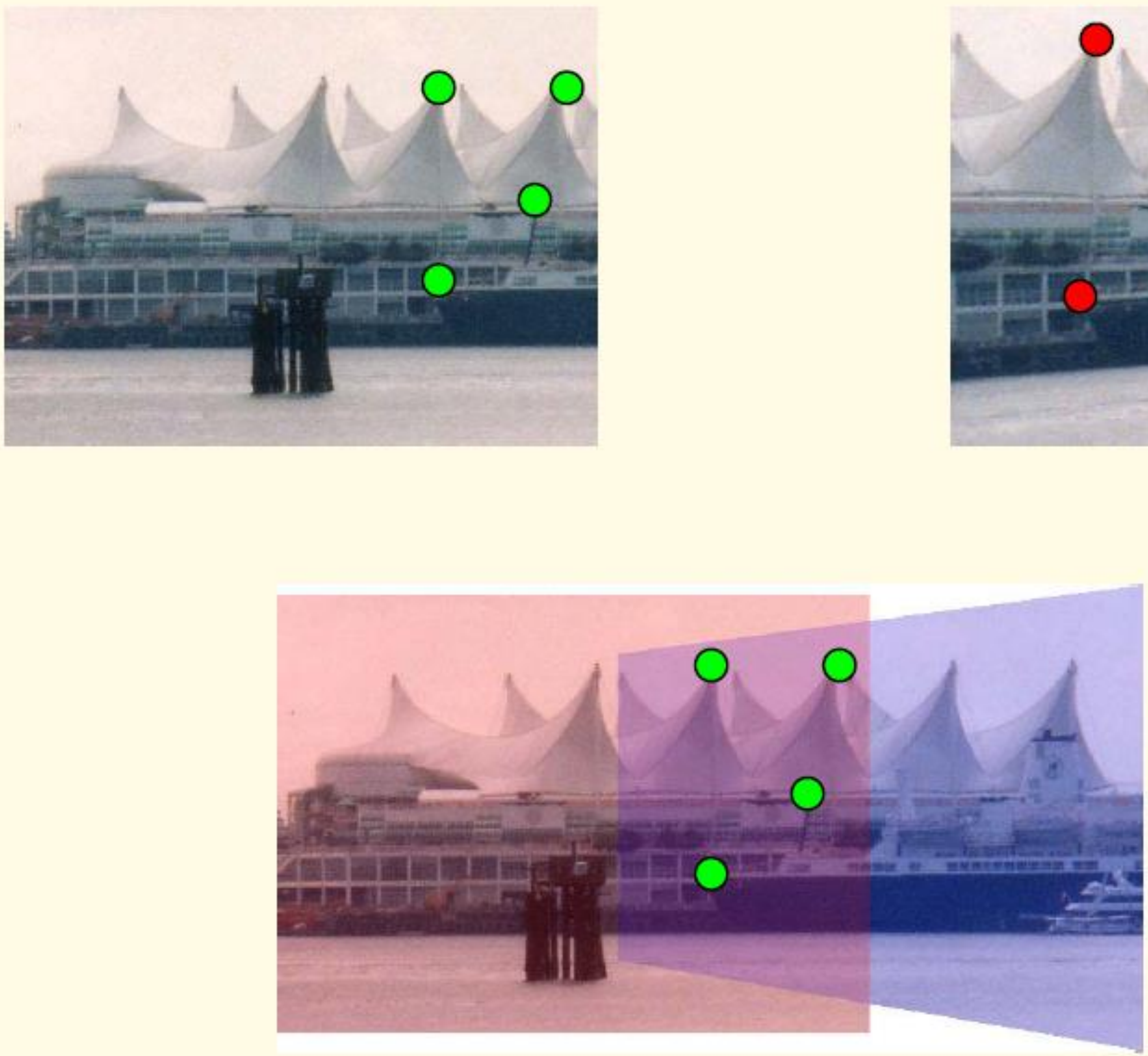
When do two images “stitch”?

Images taken from the same viewpoint are related





Images can be transformed to match



Images are related by *Homographies*

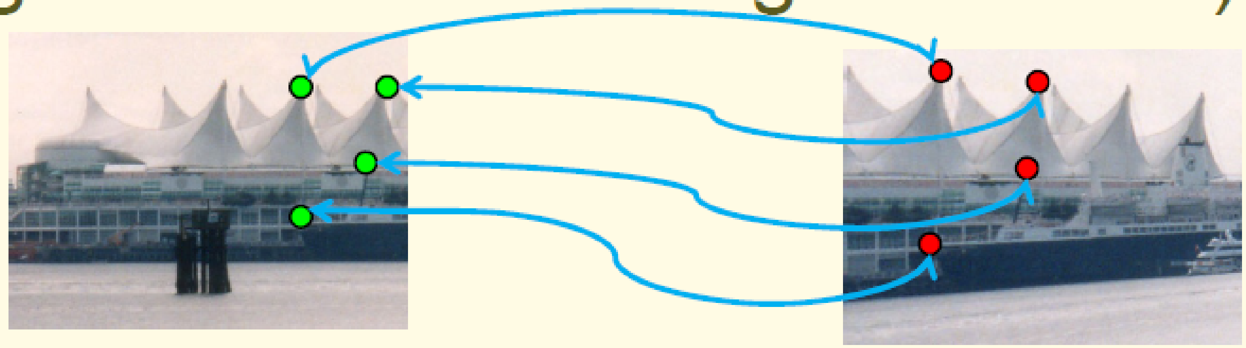
- 8 parameter, 2D Image Transformation

$$x', y' = \frac{ax + by + c}{gx + hy + 1}, \frac{dx + ey + f}{gx + hy + 1}$$

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Compute Homographies

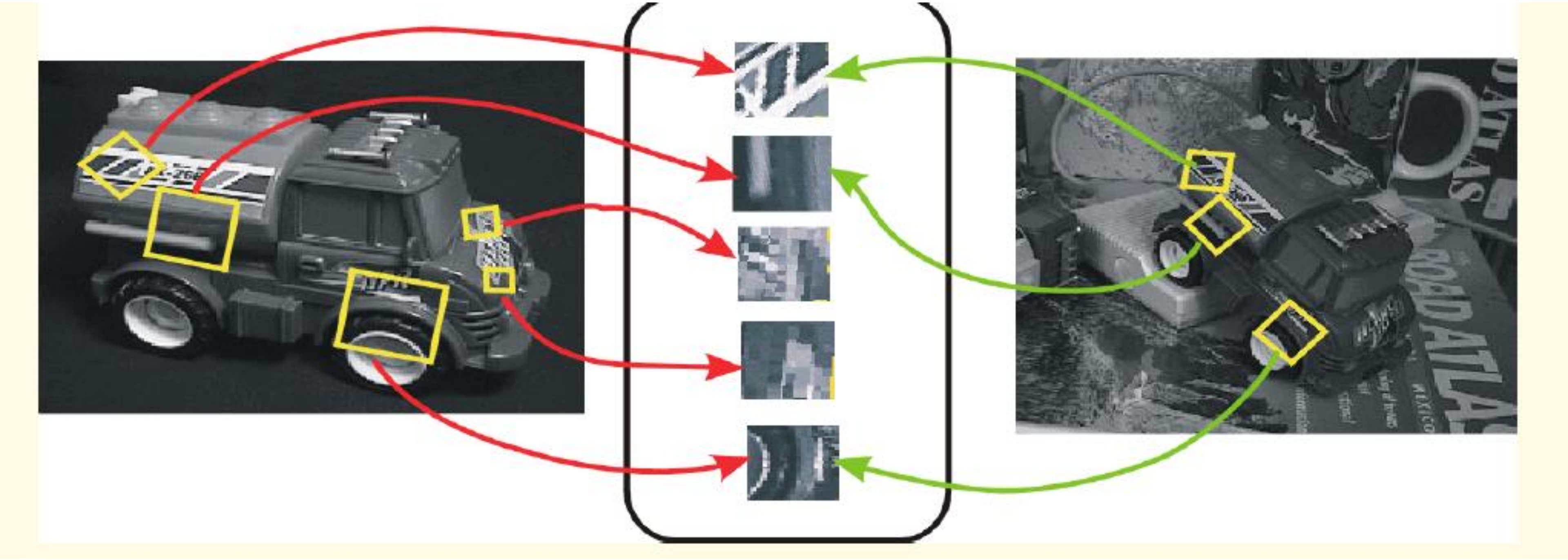
- Find Corresponding Features*
- Compute Best-Fit Homography (using robust statistics e.g. RANSAC)



- Two images stitch if and only if the best fit homography is a good fit
- If the best fit homography is a bad fit, the resulting panorama will be bad.

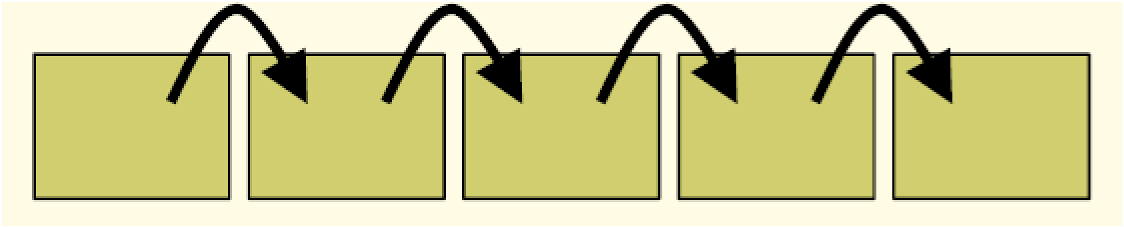
Automatic Feature Points Matching

- Match local neighborhoods around points
- Use descriptors to efficiently compare SIFT
  - [Lowe 04] most common choice

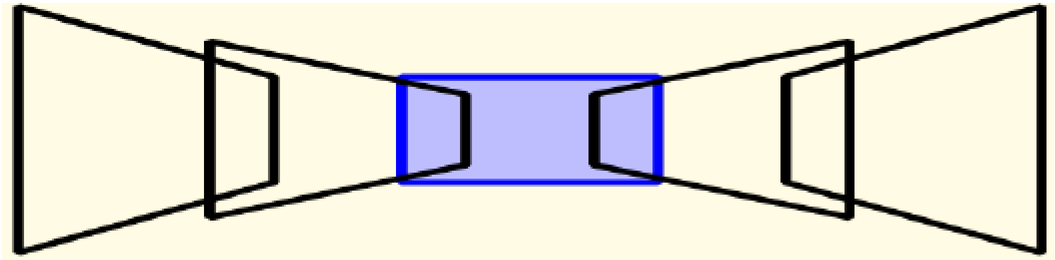


Stitching Recipe

- Align pairs of images



- Align all to a common frame

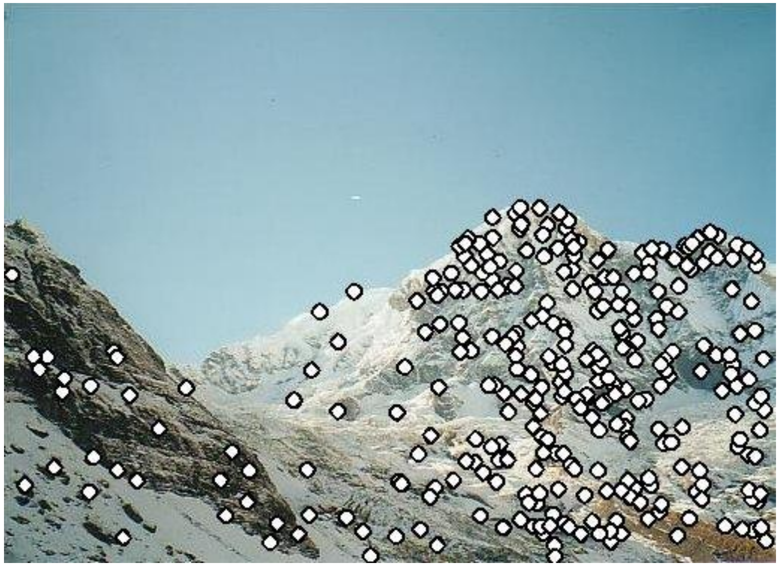


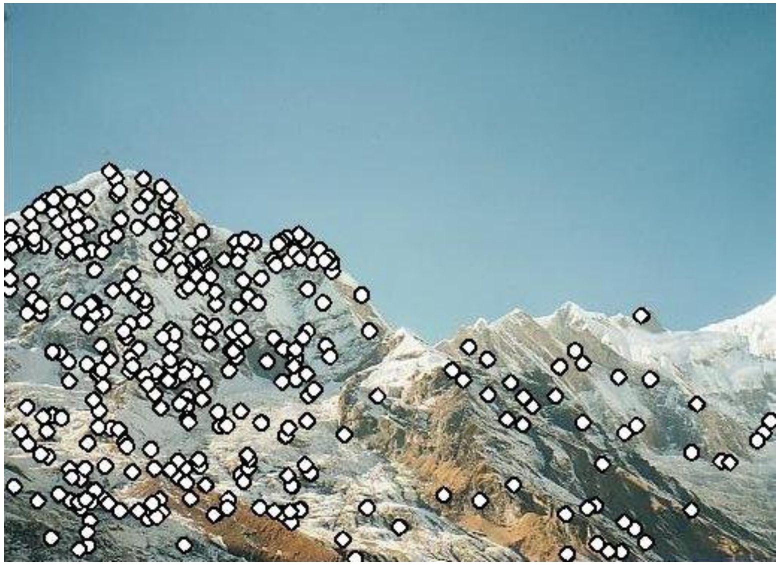
- Adjust (Global) & Blend

Wide Baseline Matching

- Images taken by cameras that are far apart make the correspondence problem very difficult

- Feature-based approach: Detect and match feature points in pairs of images

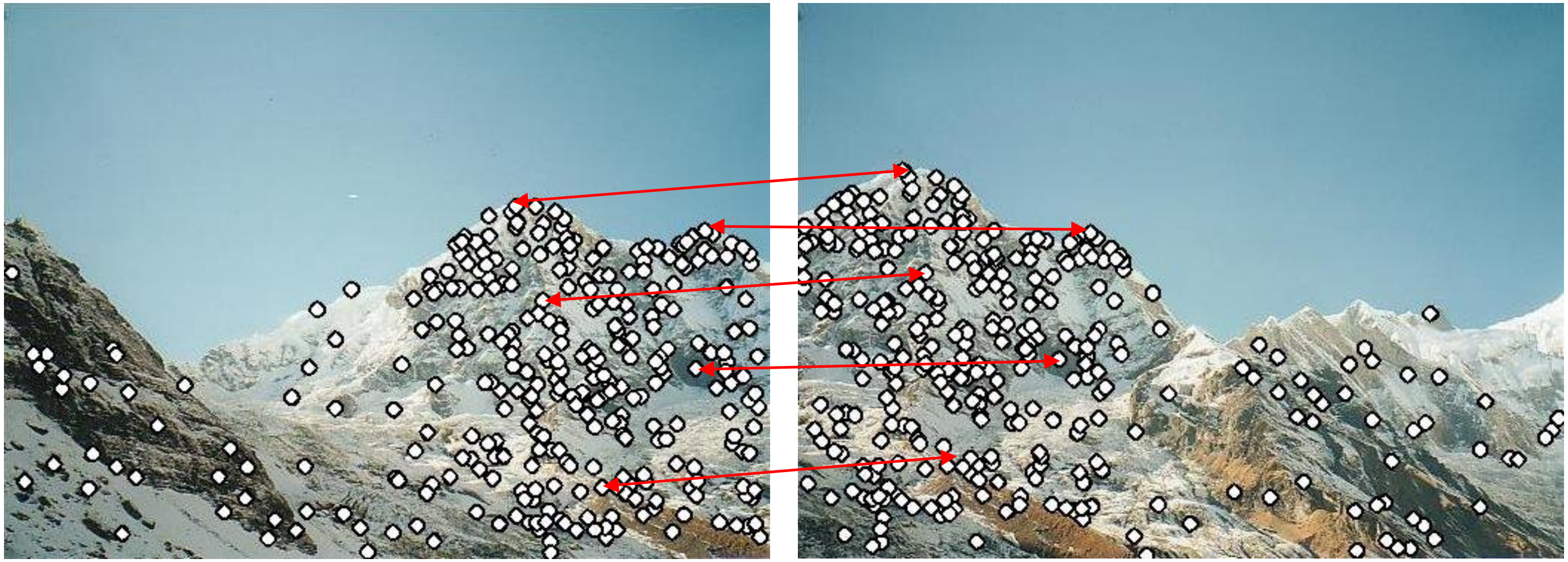




Credit: C. Dyer

Matching with Features

- Detect feature points
- Find corresponding pairs



Credit: C. Dyer

Matching with Features

- Problem 1:
  - Detect the *same* point *independently* in both images

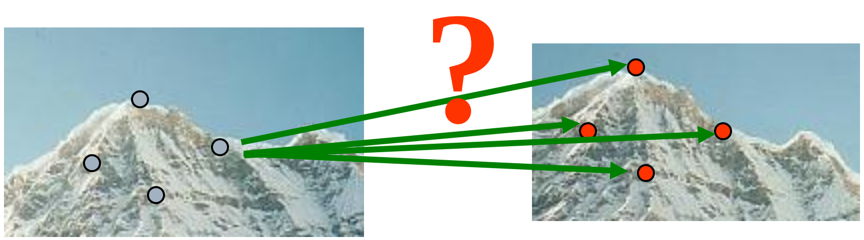
no chance to match!

We need a **repeatable detector**

Credit: C. Dyer

Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding point



We need a reliable and distinctive **descriptor**

Credit: C. Dyer

Properties of an Ideal Feature

- Local: features are local, so robust to occlusion and clutter (no prior segmentation)
- Invariant (or covariant) to many kinds of geometric and photometric transformations
- Robust: noise, blur, discretization, compression, etc. do not have a big impact on the feature
- Distinctive: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Accurate: precise localization
- Efficient: close to real-time performance

Credit: C. Dyer

Problem 1: Detecting Good Feature Points

[Image from T. Tuytelaars ECCV 2006 tutorial]

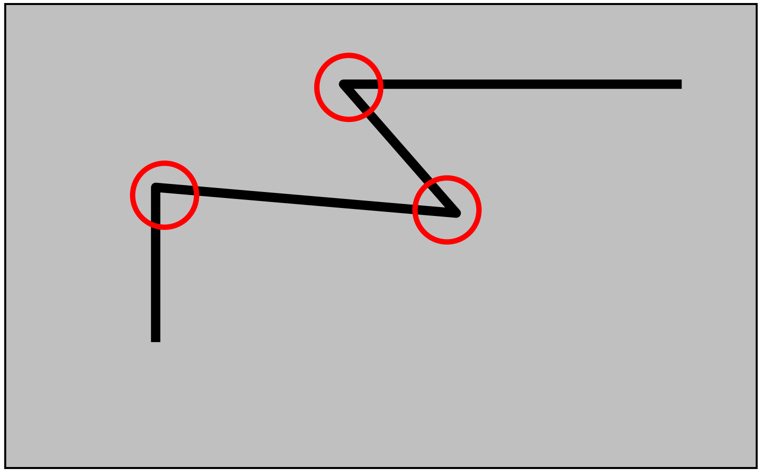
Credit: C. Dyer

Feature Detectors

- Hessian
- **Harris**
- Lowe: SIFT (DoG)
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

Credit: C. Dyer

Harris Corner Point Detector

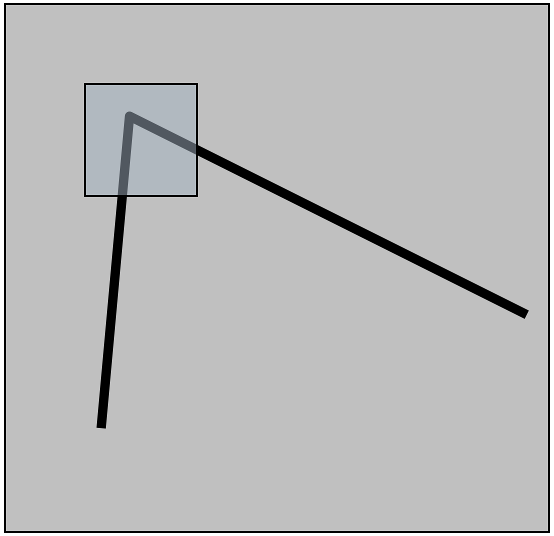


C. Harris, M. Stephens, “A Combined Corner and Edge Detector,” 1988

Credit: C. Dyer

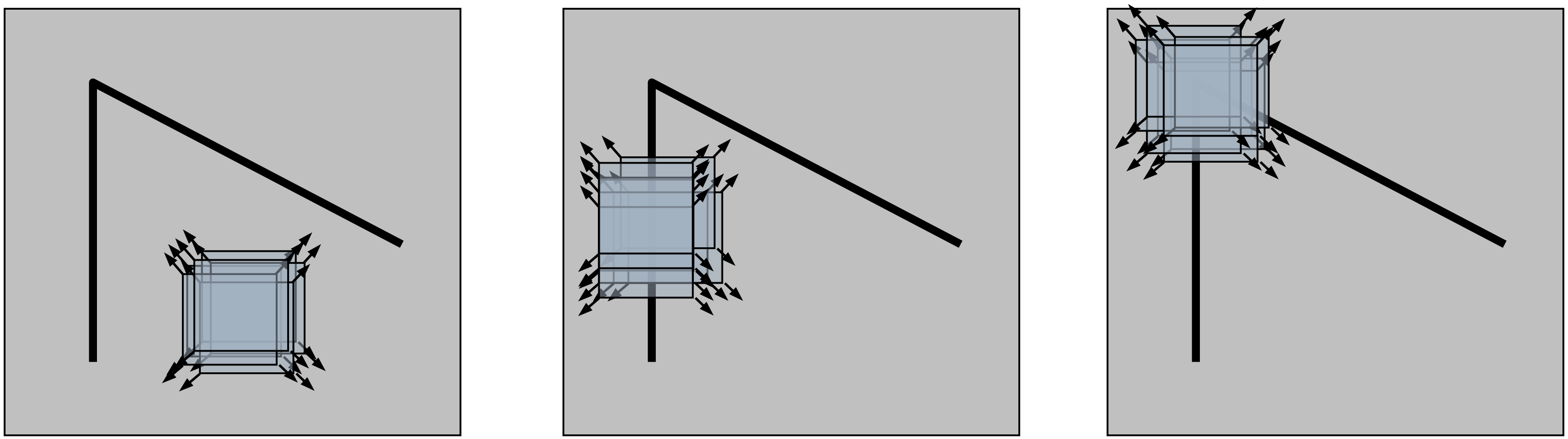
Harris Detector: Basic Idea

- We should recognize the point by looking through a small window
- Shifting a window in ***any** direction* should give *a **large** change* in response



Credit: C. Dyer

Harris Detector: Basic Idea



"flat" region:
no change in
all directions

"edge":
no change along
the edge direction

"corner":
significant change
in *all* directions

Credit: C. Dyer

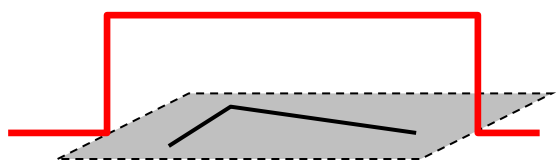
Harris Detector: Derivation

Change of intensity for a (small) shift by [*u*,*v*] in image *I*:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

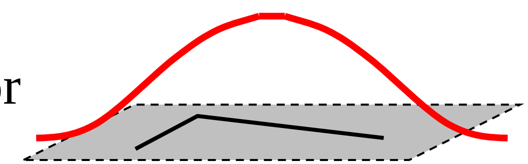
Weighting function — Shifted intensity — Intensity

Weighting function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Credit: R. Szeliski

Calculus: Taylor Series Expansion

A real function $f(x+u)$ can be approximated as the 2nd order of its Taylor series expansion at a point x.

$$f(x+u) = f(x) + uf'(x) + O(u^2)$$

Derivatives

For 1D function f(x), the derivative is:

$$\frac{\partial f(x)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

For 2D function f(x,y), the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

Source: K. Grauman

Derivatives

For discrete data, we can approximate using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x,y)}{1}$$

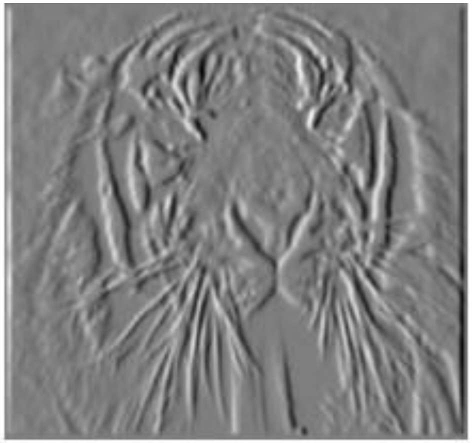
To implement above as convolution, what would be the associated filter?

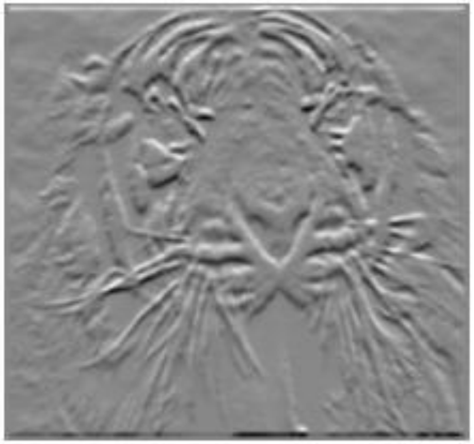
Source: K. Grauman

Partial derivatives of an image

$$\frac{\partial f(x,y)}{\partial x}$$

-1	1





$$\frac{\partial f(x,y)}{\partial y}$$

-1
1

or

1
-1

Which shows changes with respect to x?

Source: S. Lazebnik

Finite difference filters

Other approximations of derivative filters exist:

Prewitt: $$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \; ; \quad M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

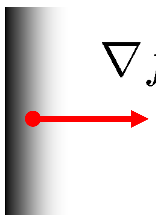
Sobel: $$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; ; \quad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Roberts: $$M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \; ; \quad M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

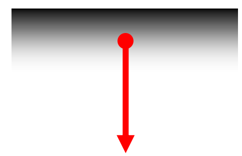
Source: K. Grauman

Image gradient

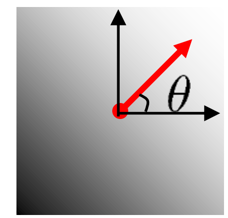
The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

The gradient points in the direction of most rapid increase in intensity

- How does this direction relate to the direction of the edge?

The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Harris Detector

Apply 2nd order Taylor series expansion:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

$$= \sum_{x,y} w(x,y)\left[I_x u + I_y v + O(u^2, v^2)\right]^2$$

$$E(u,v) = Au^2 + 2Cuv + Bv^2$$

$$A = \sum_{x,y} w(x,y) I_x^2(x,y)$$

$$B = \sum_{x,y} w(x,y) I_y^2(x,y)$$

$$C = \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y)$$

$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & C \\ C & B \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$I_x = \partial I(x,y) / \partial x$$

$$I_y = \partial I(x,y) / \partial y$$

Credit: R. Szeliski

Harris Corner Detector

Expanding $E(u,v)$ in a 2nd order Taylor series, we have, for small shifts, $[u,v]$, a *bilinear* approximation:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where **M** is a 2 × 2 matrix computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$I_x = \partial I(x,y) / \partial x$$
$$I_y = \partial I(x,y) / \partial y$$

Note: Sum computed over small neighborhood around given pixel

Credit: R. Szeliski

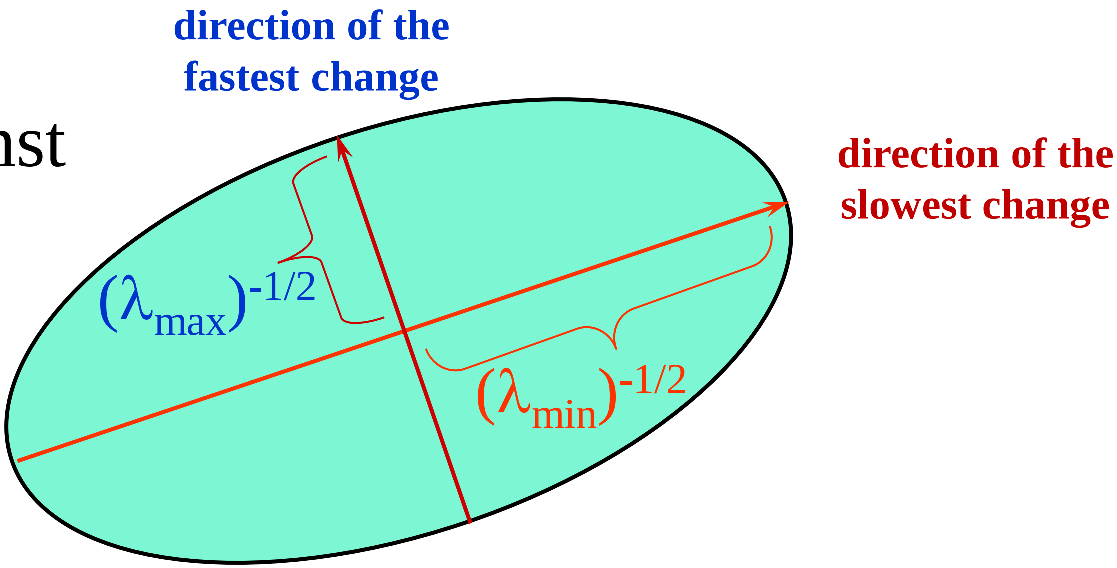
Harris Corner Detector

Intensity change in shifting window: eigenvalue analysis

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

λ_1, λ_2 – eigenvalues of $\mathbf{M}$

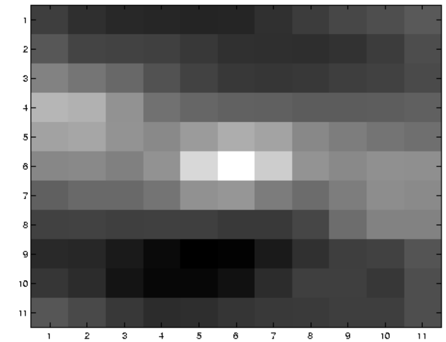
Ellipse $E(u,v)$ = const



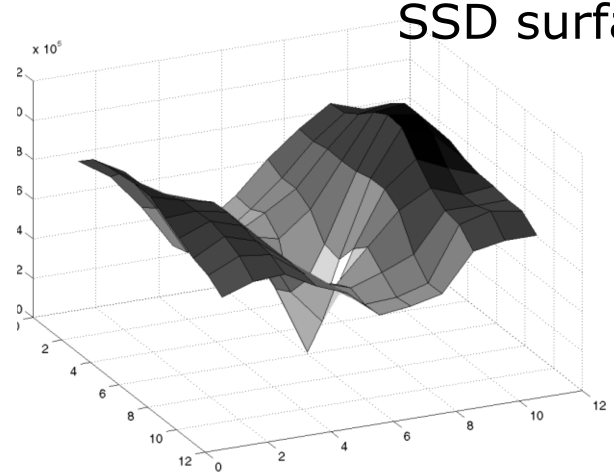
Credit: R. Szeliski

Selecting Good Features

Image patch



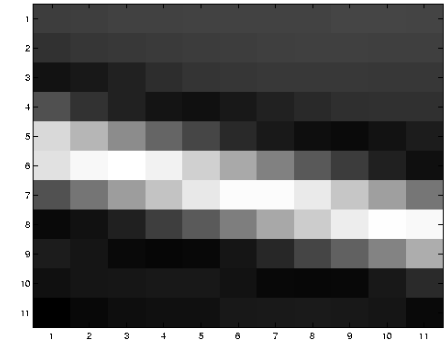
SSD surface



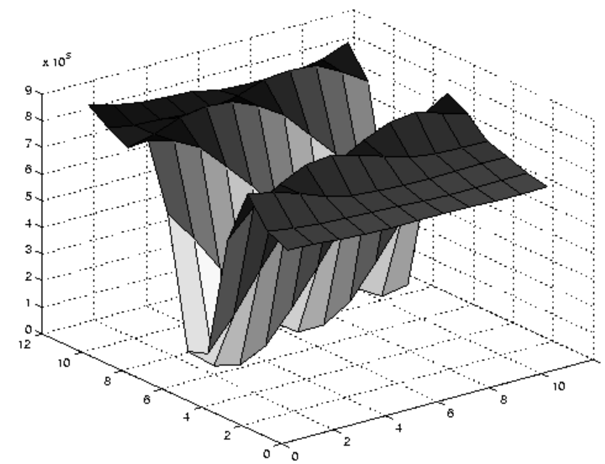
λ_1 and λ_2 both large

Credit: C. Dyer

Selecting Good Features



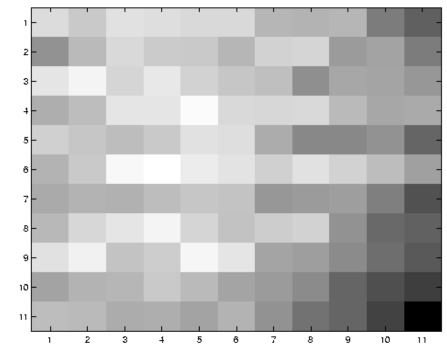
SSD surface



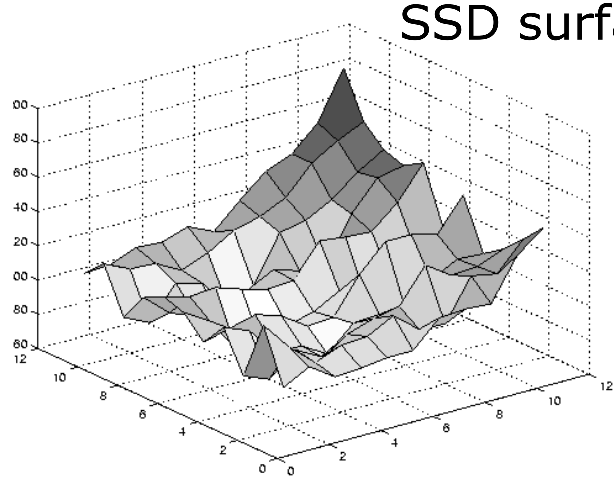
large λ_1, small λ_2

Credit: C. Dyer

Selecting Good Features



SSD surface



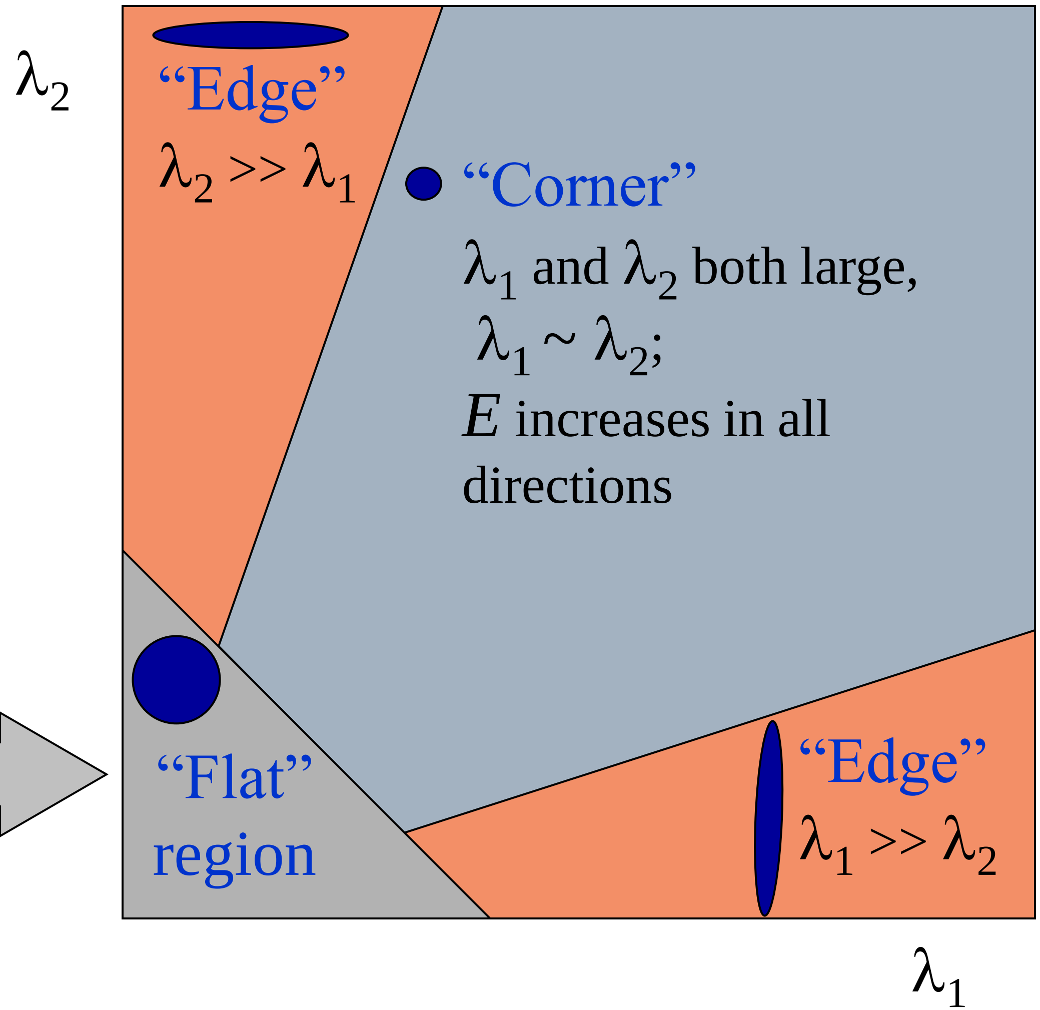
small λ_1, small λ_2

Credit: C. Dyer

Harris Corner Detector

Classification of image points using eigenvalues of **M**:

λ_1 and λ_2 are small;
E is almost constant in all directions



Credit: C. Dyer

Harris Corner Detector

Measure of corner response:

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

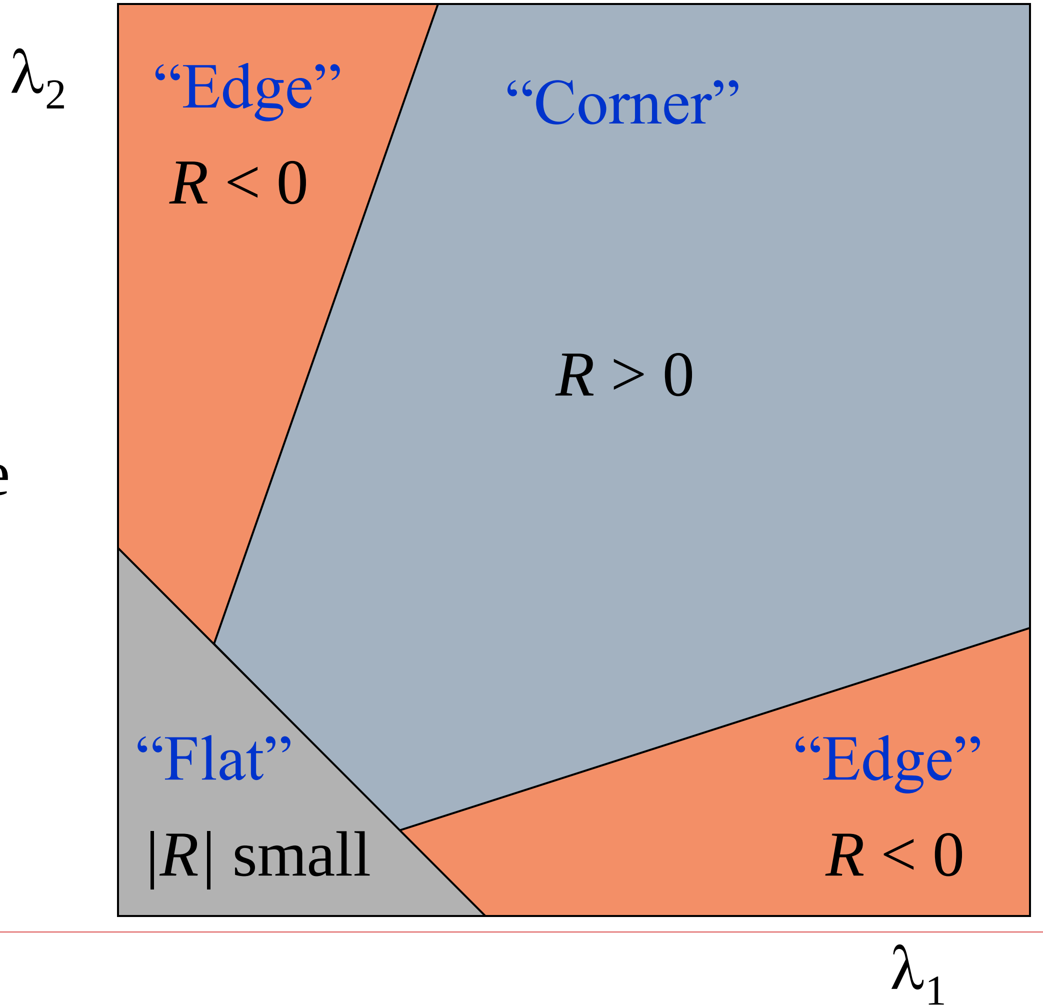
$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

k is an empirically-determined constant; e.g., $k = 0.05$

Credit: C. Dyer

Harris Corner Detector

- R depends only on eigenvalues of **M**
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Credit: C. Dyer

Harris Corner Detector: Algorithm

- Algorithm:
  1. Find points with large corner response function R

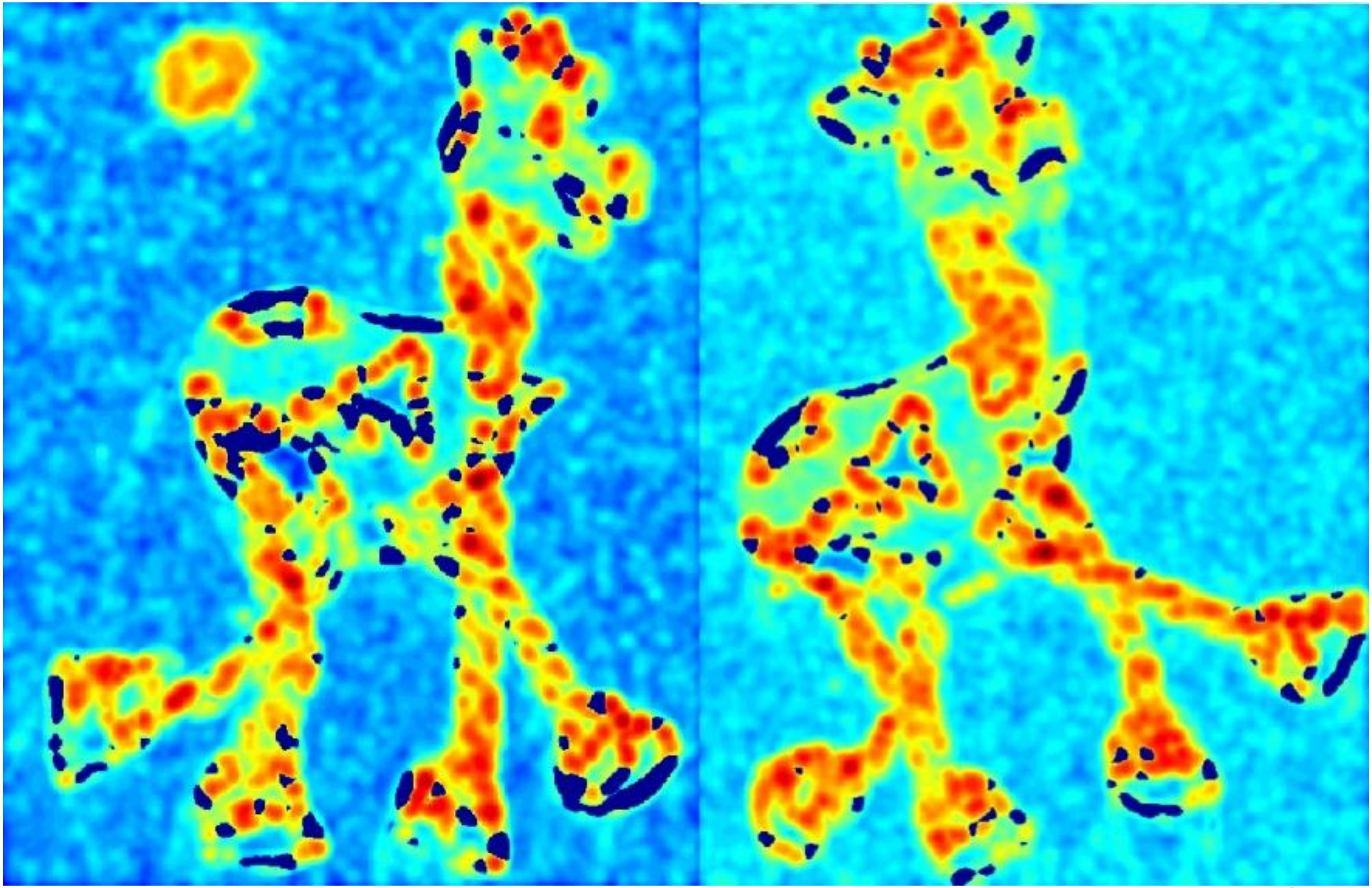
     (i.e., R > threshold)
  2. Take the points of local maxima of R (for localization) by non-maximum suppression

Credit: C. Dyer

Harris Detector: Example

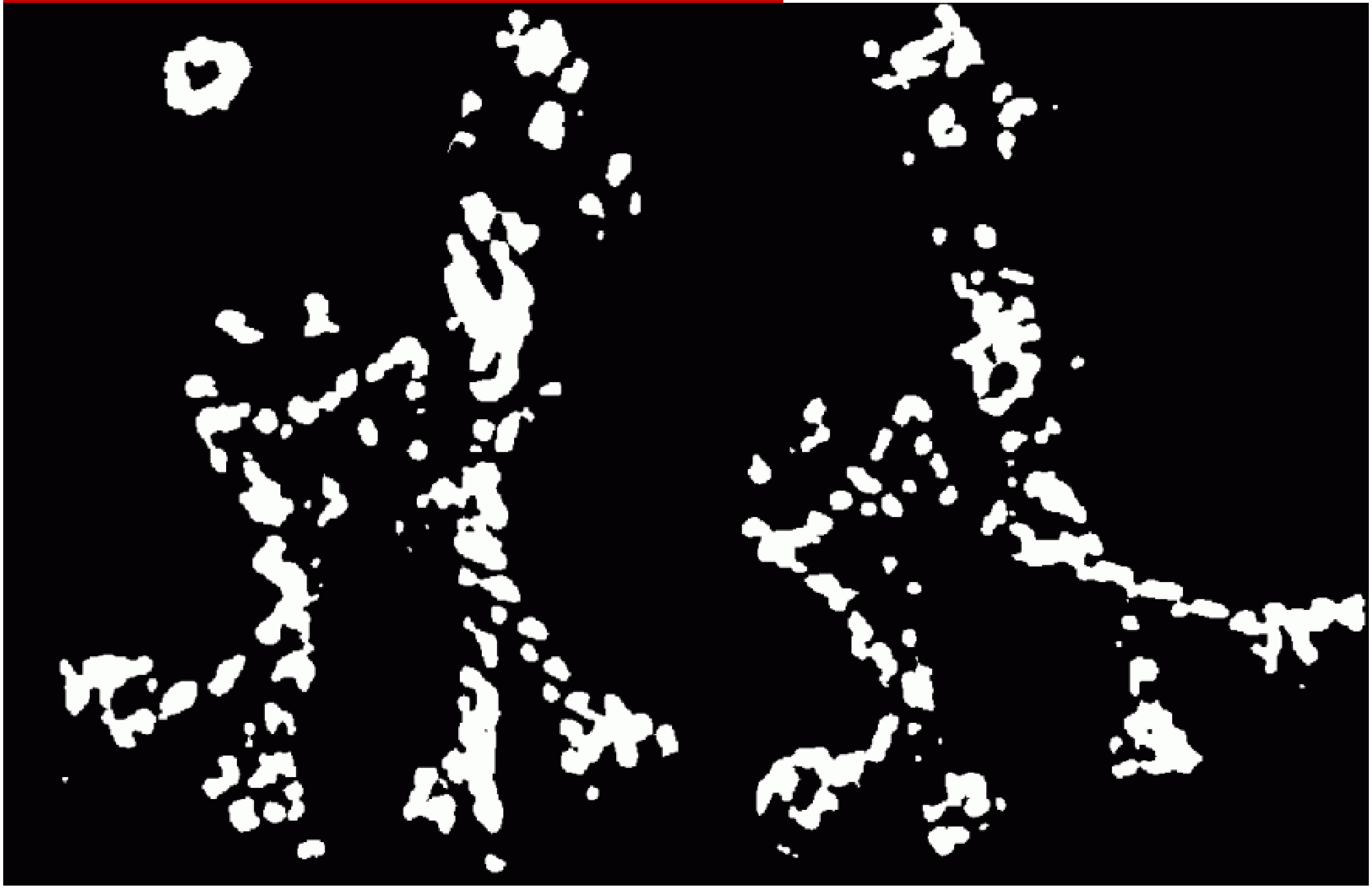
Credit: C. Dyer

Harris Detector: Example



Credit: C. Dyer Compute corner response $R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$

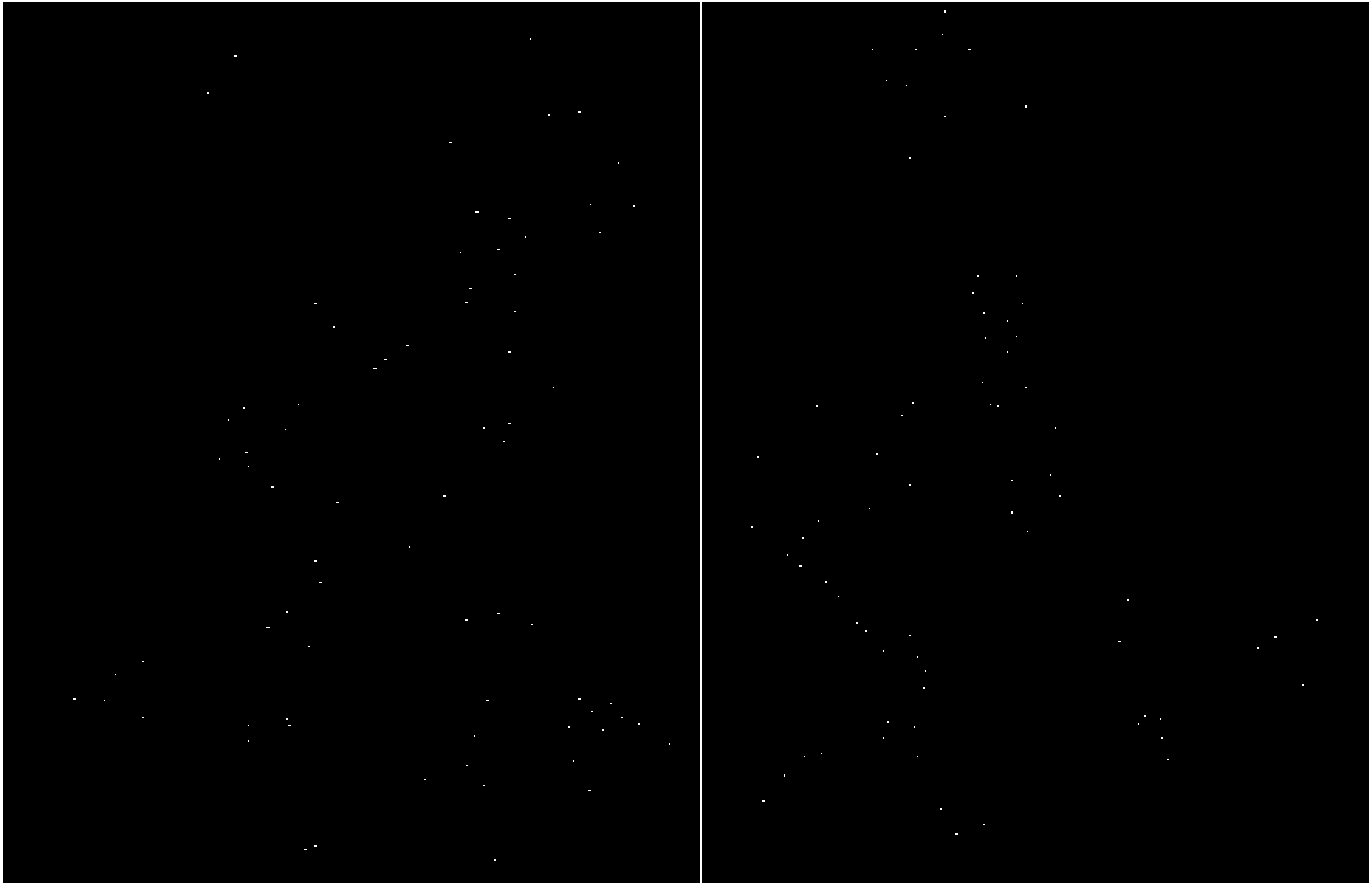
Harris Detector: Example



Credit: C. Dyer

Find points with large corner response: R > threshold

Harris Detector: Example

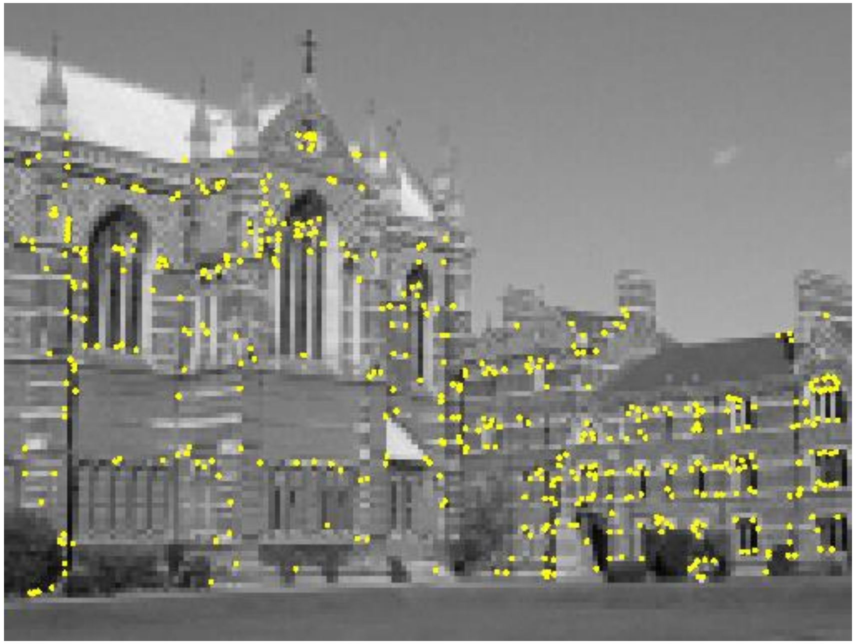
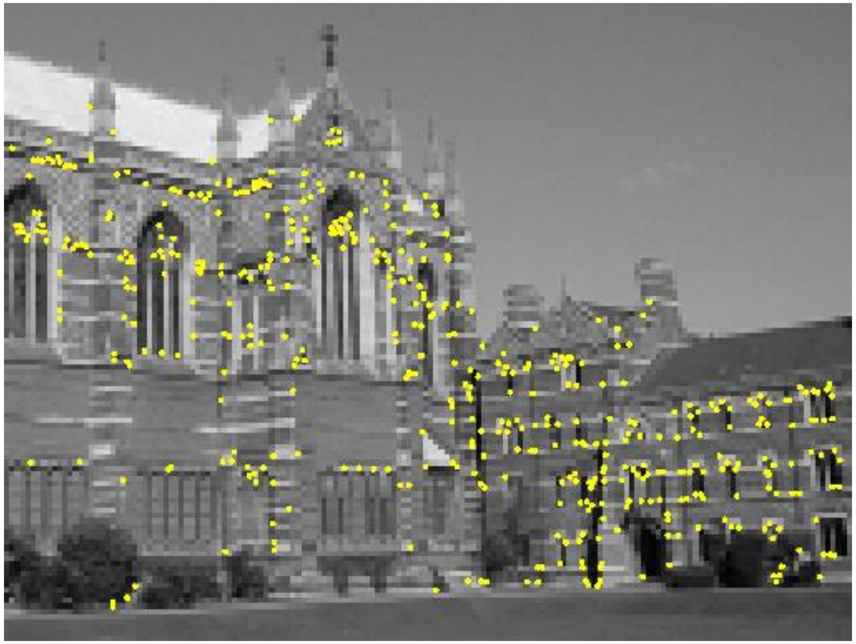


Credit: C. Dyer Take only the points of local maxima of R

Harris Detector: Example

Credit: C. Dyer

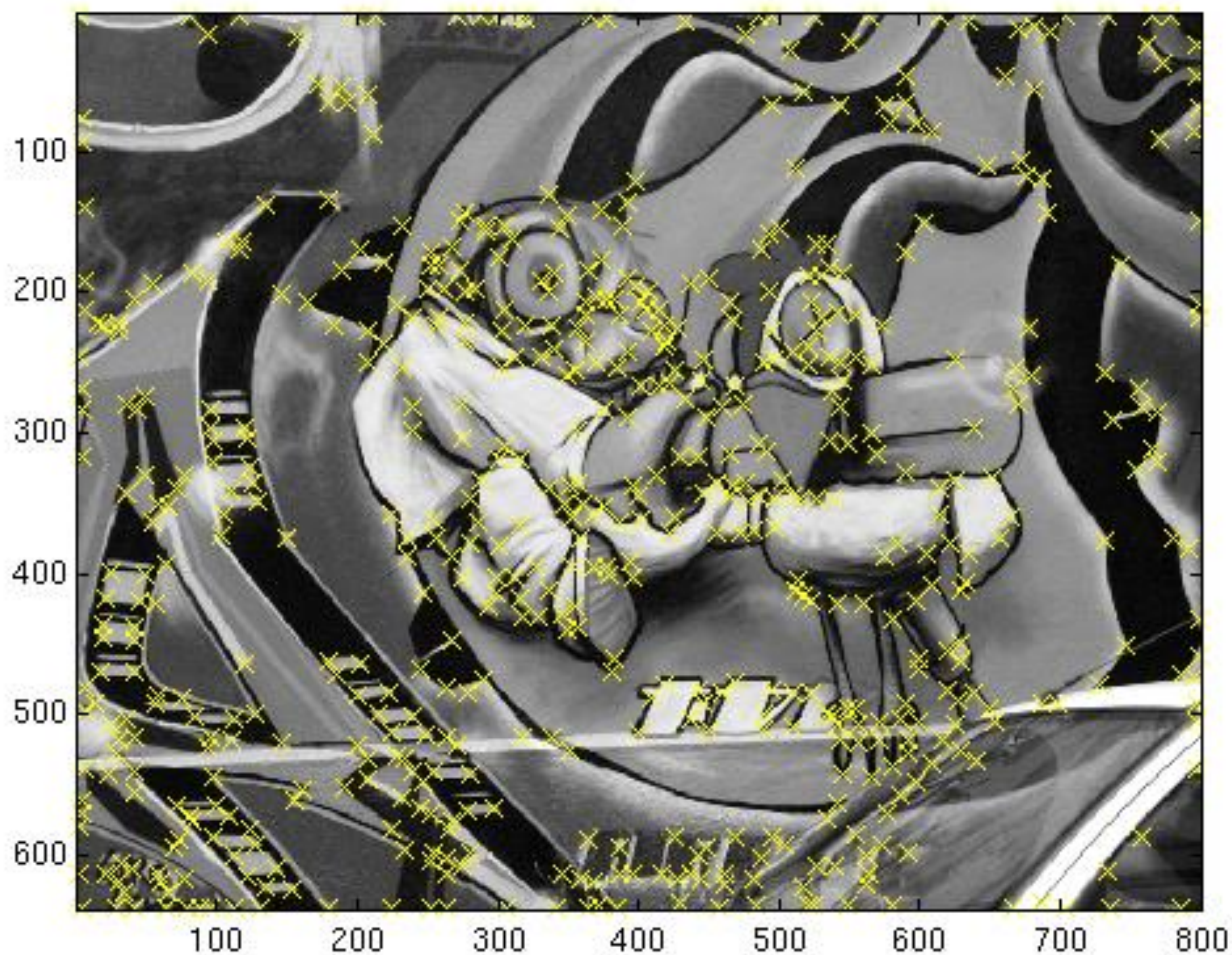
Harris Detector: Example

Interest points extracted with Harris (~ 500 points)

Credit: C. Dyer

Harris Detector: Example



Credit: C. Dyer

Harris Detector: Summary

- Average intensity change in direction [*u,v*] can be expressed in bilinear form:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

- Describe a point in terms of eigenvalues of **M**: *measure of corner response:*

$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

- A good (corner) point should have a *large intensity change* in *all directions*, i.e., *R* should be a large positive value

Credit: C. Dyer

Student paper presentation

Color harmonization

D. Cohen-Or, O. Sorkine, R. Gal, T. Leyv, and, Y. Xu
ACM SIGGRAPH 2006

Presenter: Gerendasy, Daniel R

Student paper presentation

Colorization Using Optimization

A. Levin, D. Lischinski, and Y. Weiss
ACM SIGGRAPH 2004

Presenter: Nguyen, Henry

Next Time

- Panorama
  - Feature and matching
- Student paper presentations
  - 04/26: Li, Chi G
    - Color Conceptualization
      X. Hou and L. Zhang, ACM Multimedia 2007
  - 04/26: Liu, Jiawei
    - Photographic tone reproduction for digital images
      E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, SIGGRAPH 2012