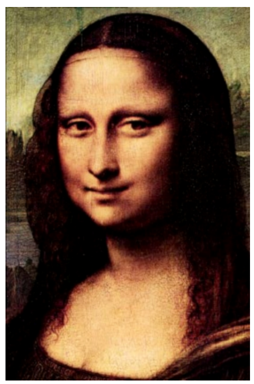
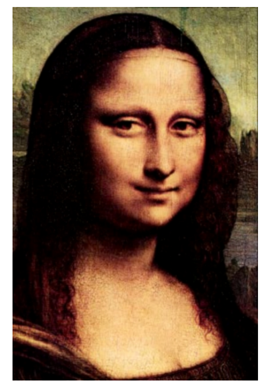
CS4670: Computer Vision

Kavita Bala

Lecture 5: Image Interpolation and Features





Announcements

- Tell us if conflicts on PA 1 grading

- iclickerGo
  - Tell us if you don’t have it

Upsampling

- This image is too small for this screen: 
- How can we make it 10 times as big?

- Simplest approach:
  repeat each row
  and column 10 times

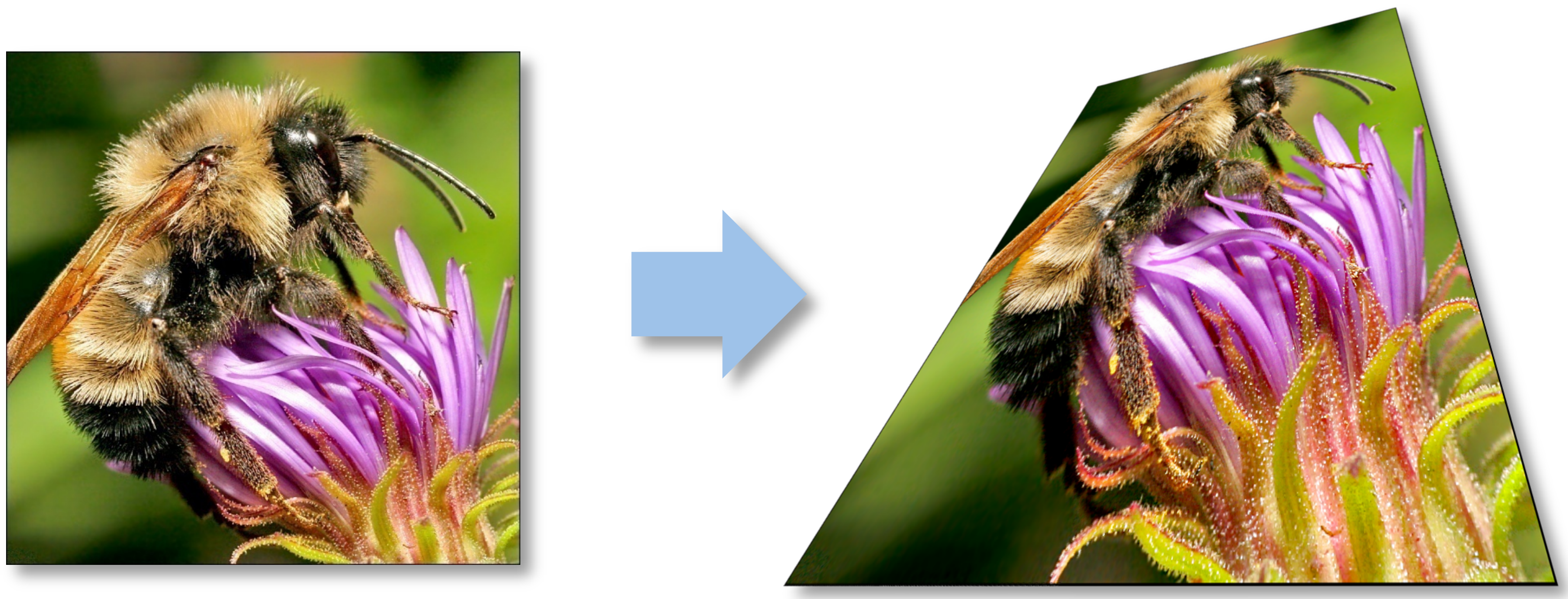
Image interpolation

Original image:  x 10



Image interpolation

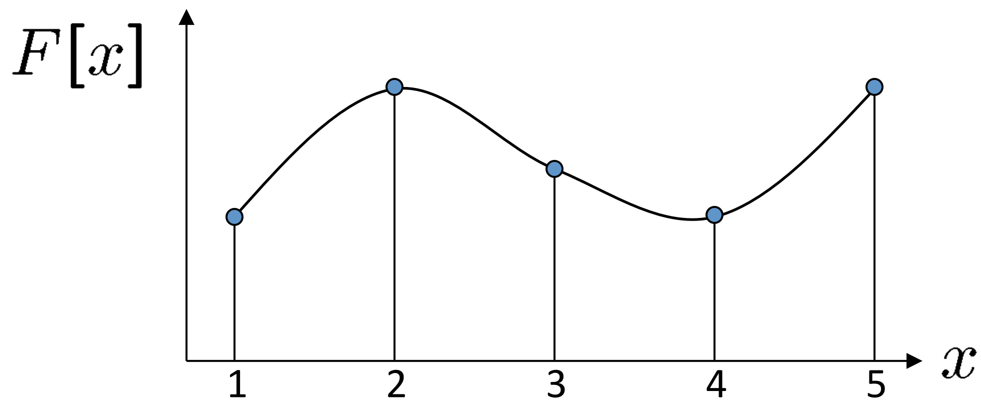
Also used for *resampling*



Upsampling

- This image is too small for this screen: 
- How can we make it 10 times as big?
- Simplest approach:
  repeat each row
  and column 10 times
- ("Nearest neighbor
  interpolation")

Image interpolation



d = 1 in this example

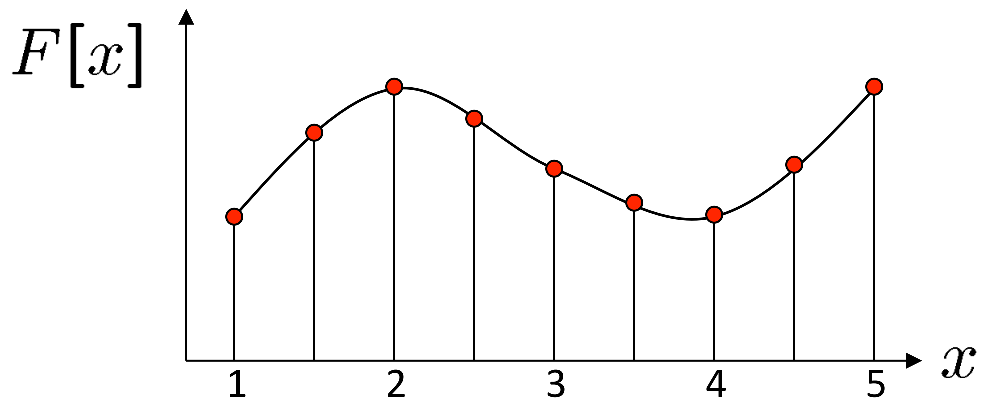
Recall how a digital image is formed

$$F[x,y] = \text{quantize}\{f(xd, yd)\}$$

- It is a discrete point-sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

Adapted from: S. Seitz

Image interpolation



d = 1 in this example

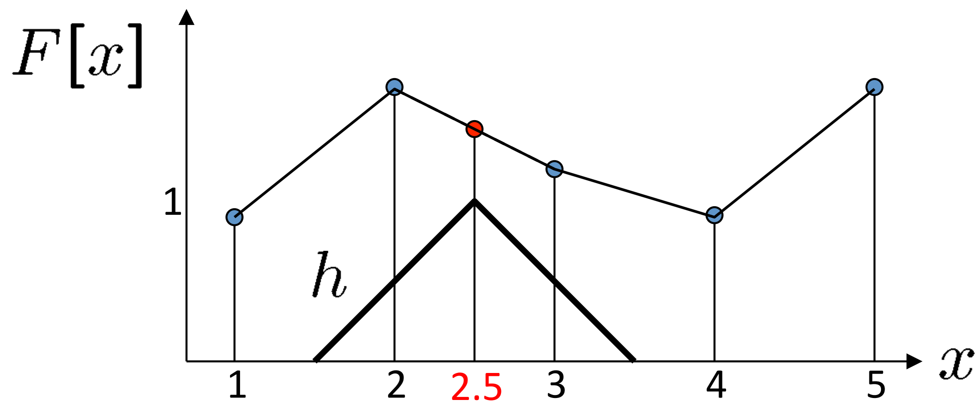
Recall how a digital image is formed

$$F[x,y] = \text{quantize}\{f(xd, yd)\}$$

- It is a discrete point-sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

Adapted from: S. Seitz

Image interpolation



d = 1 in this example

- What if we don't know f ?
  - Guess an approximation: $\tilde{f}$
  - Can be done in a principled way: filtering
  - Convert F to a continuous function:

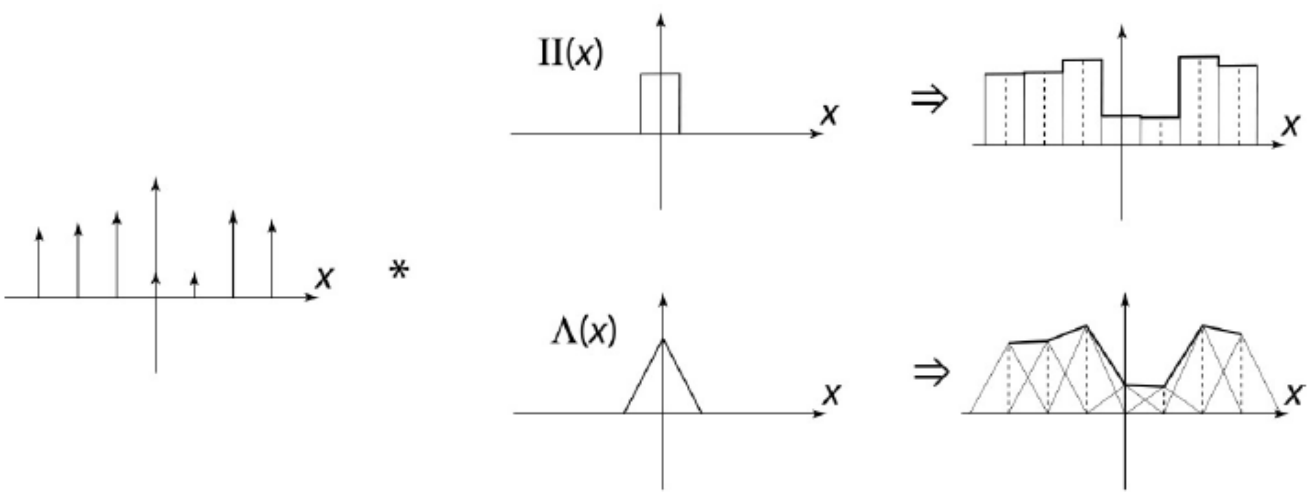
$$f_F(x) = F(\tfrac{x}{d}) \text{ when } \tfrac{x}{d} \text{ is an integer, 0 otherwise}$$

  - Reconstruct by convolution with a *reconstruction filter,* h

$$\tilde{f} = h * f_F$$

Adapted from: S. Seitz

Image interpolation



Nearest-neighbor interpolation

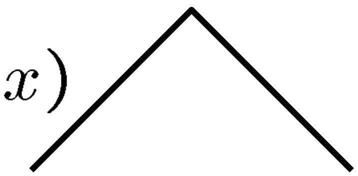
Linear interpolation

Source: B. Curless

Reconstruction filters

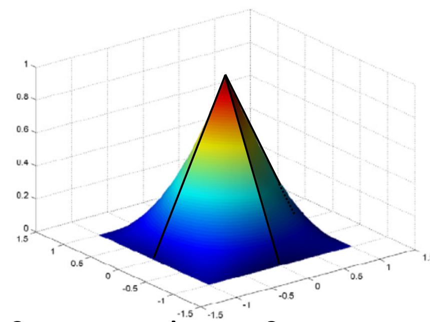
- What does the 2D version of this hat function look like?

$h(x)$



performs
linear interpolation

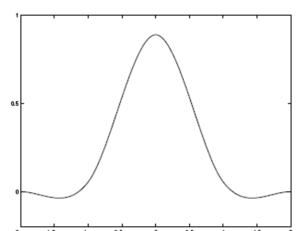
$h(x, y)$



(tent function) performs
bilinear interpolation

Better filters give better resampled images

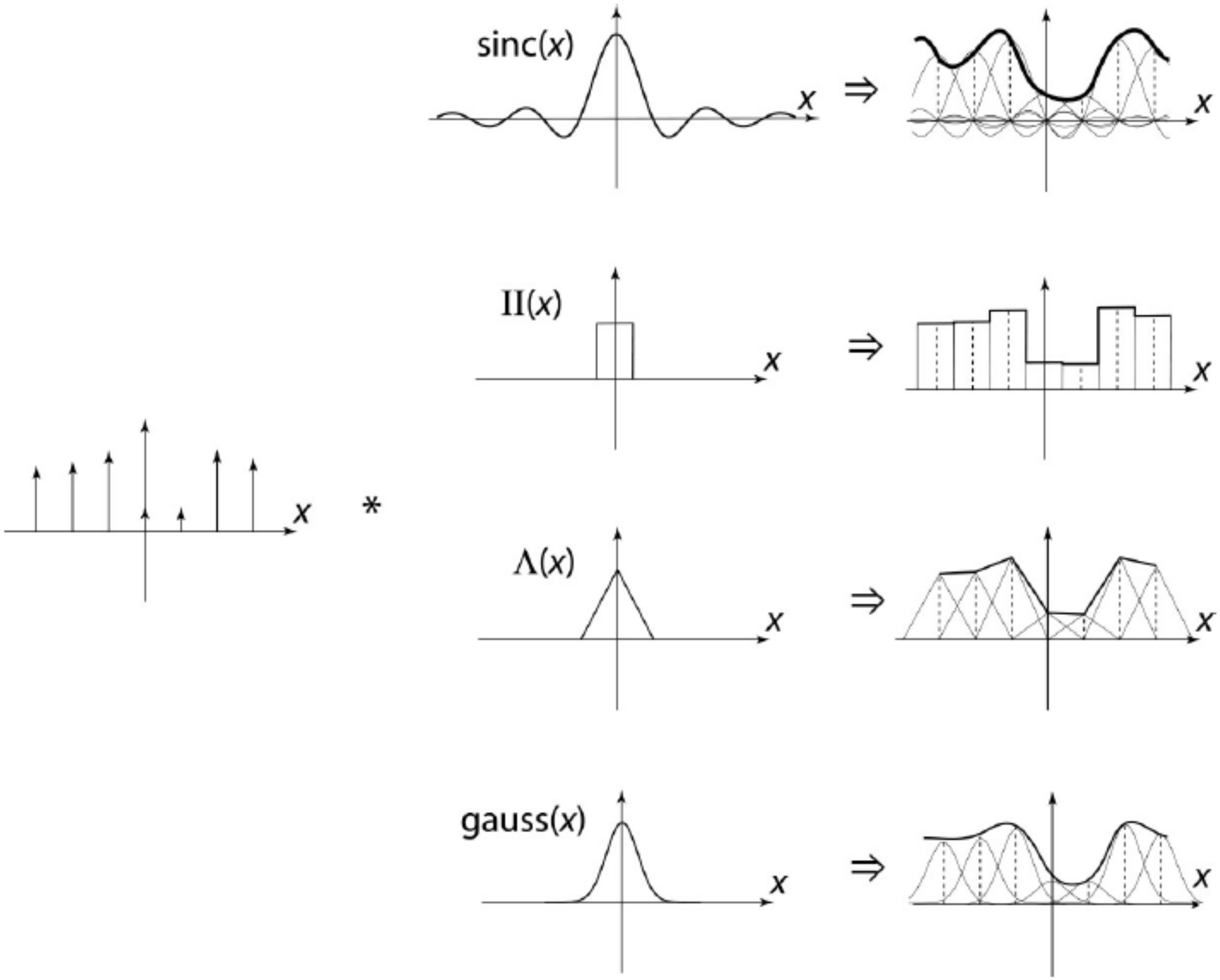
- **Bicubic** is common choice



Cubic reconstruction filter

$$r(x) = \frac{1}{6}\begin{cases} (12-9B-6C)|x|^3 + (-18+12B+6C)|x|^2 + (6-2B) & |x| < 1 \\ ((-B-6C)|x|^3 + (6B+30C)|x|^2 + (-12B-48C)|x| + (8B+24C) & 1 \leq |x| < 2 \\ 0 & otherwise \end{cases}$$

Image interpolation



“Ideal” reconstruction

Nearest-neighbor interpolation

Linear interpolation

Gaussian reconstruction

Source: B. Curless

Image interpolation

Original image:  x 10

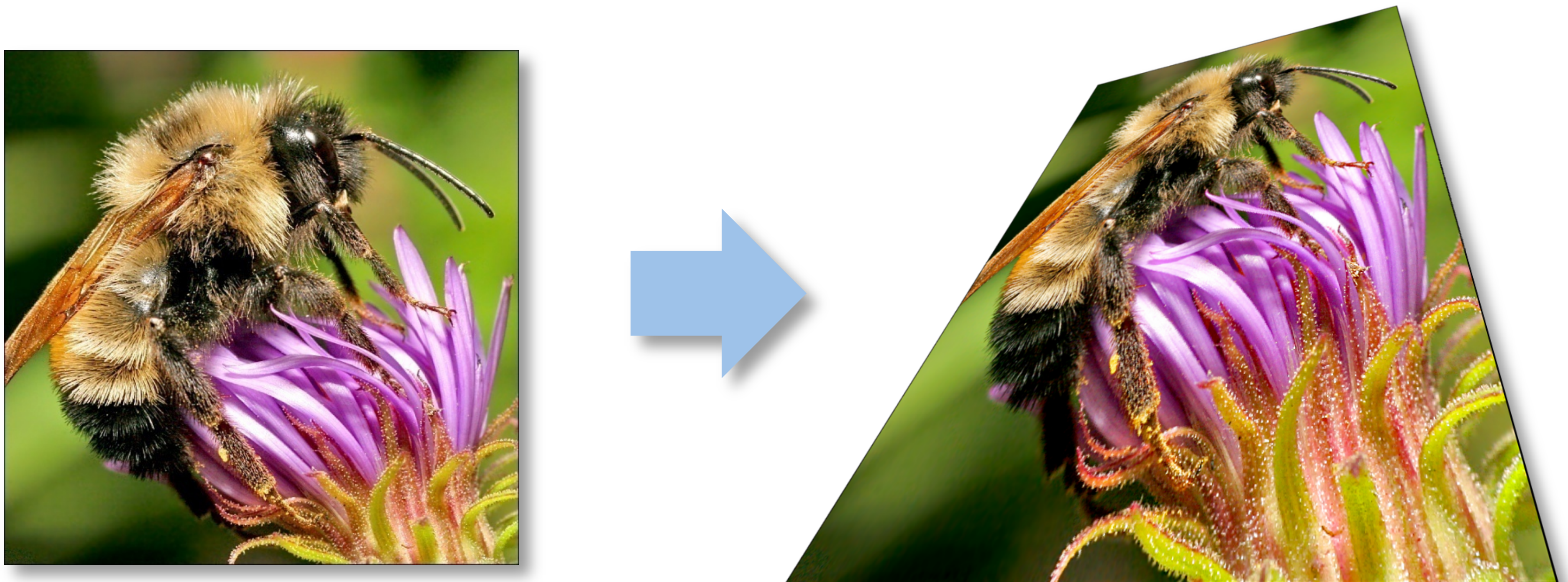
Nearest-neighbor interpolation

Bilinear interpolation

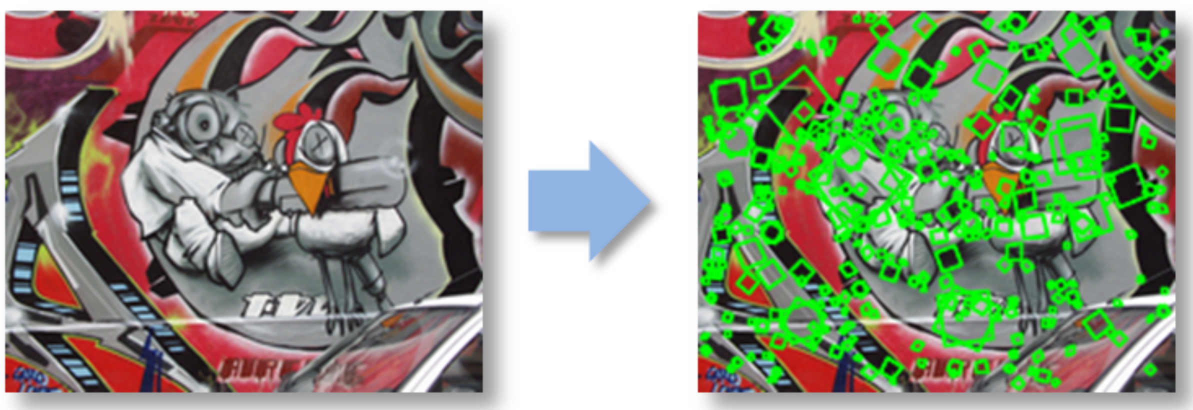
Bicubic interpolation

Image interpolation

Also used for *resampling*



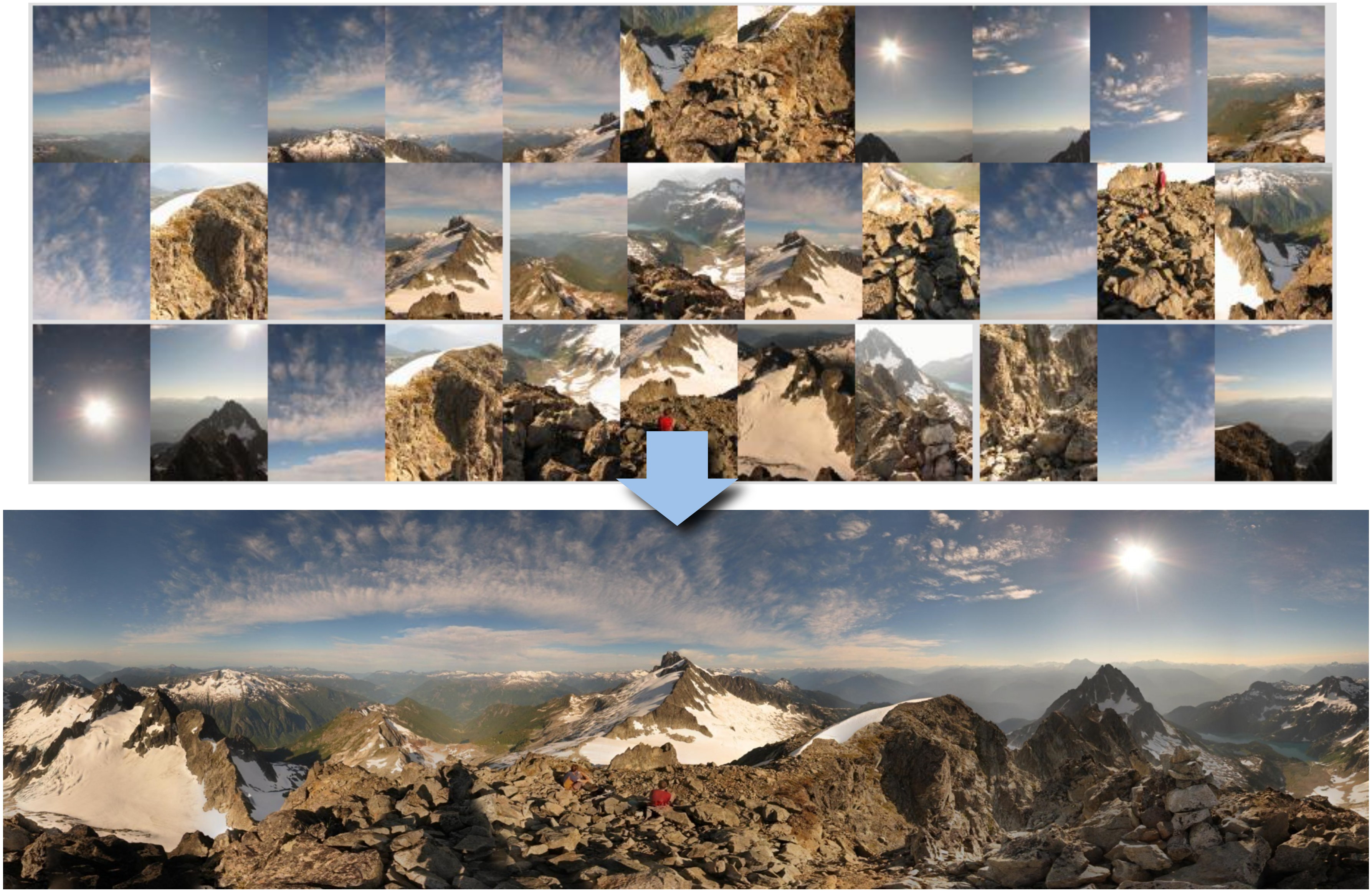
Feature detection and matching



Reading

- Szeliski: 4.1

Motivation: Automatic panoramas



Credit: Matt Brown

Motivation: Automatic panoramas

HD View
http://research.microsoft.com/en-us/um/redmond/groups/ivm/HDView/HDGigapixel.htm

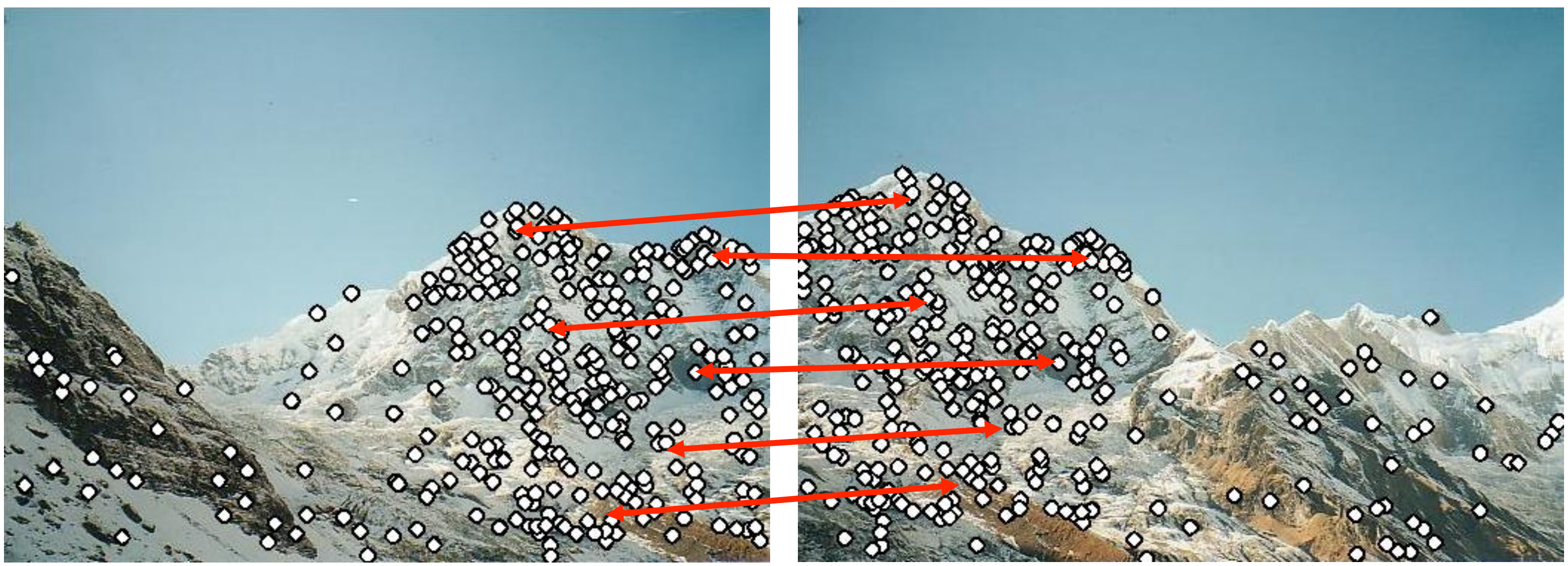
Also see GigaPan:
http://gigapan.org/

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?



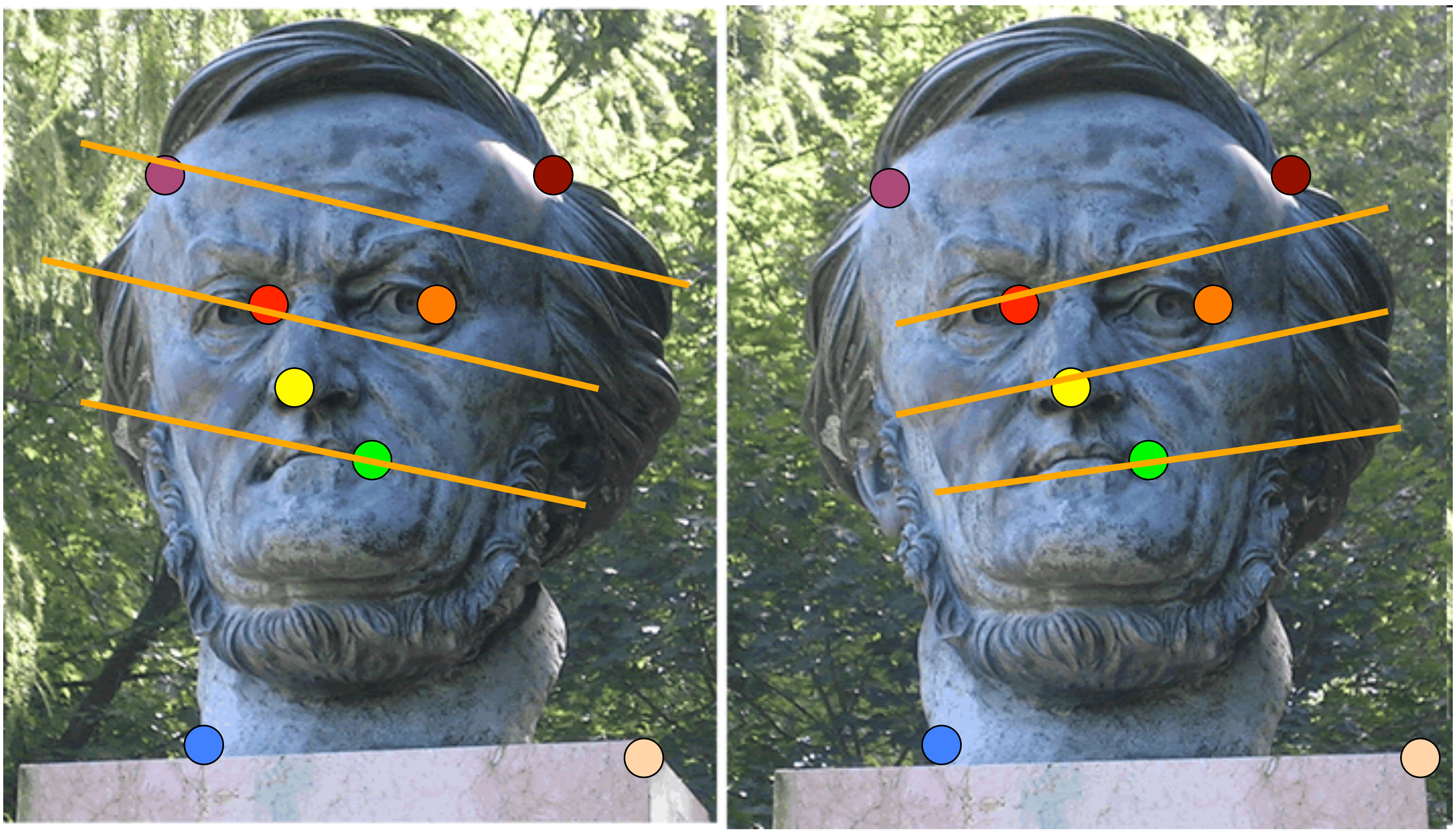
Step 1: extract features
Step 2: match features

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
Step 3: align images

Example: estimating “fundamental matrix” that corresponds two views



Slide from Silvio Savarese

Example: structure from motion

Applications

- Feature points are used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition

Matching can be challenging

Fei-Fei Li

Image matching

by Diva Sian

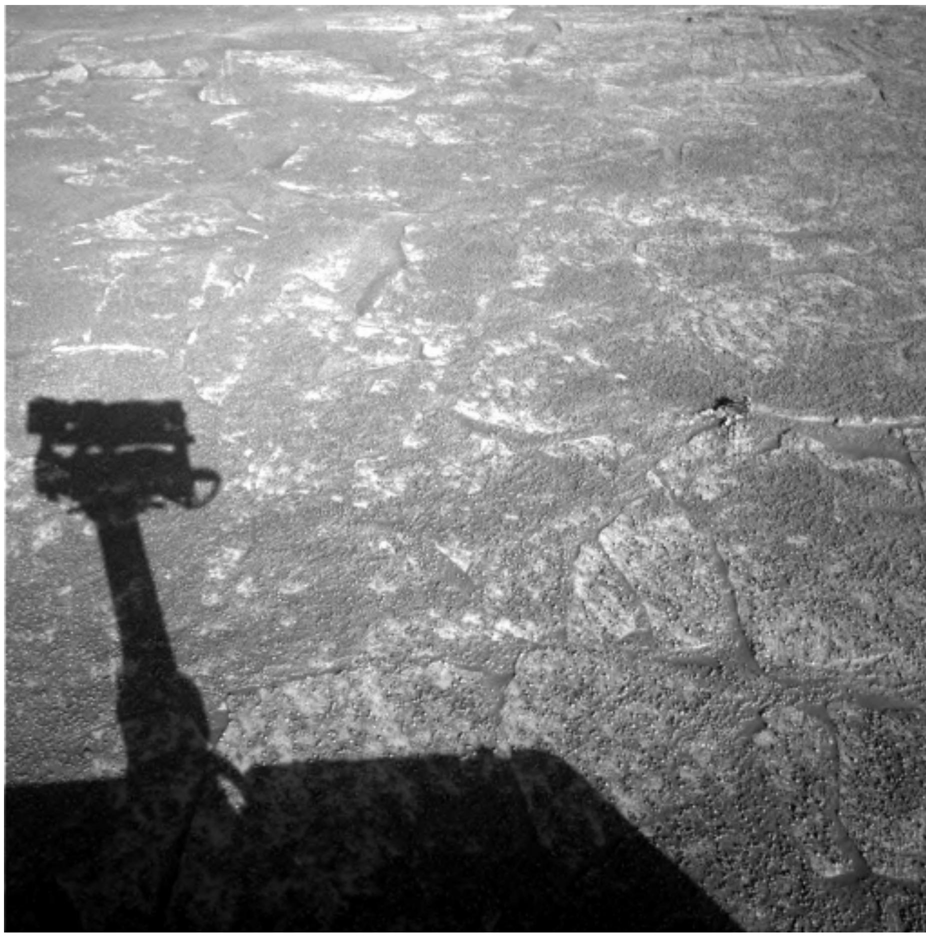
by swashford

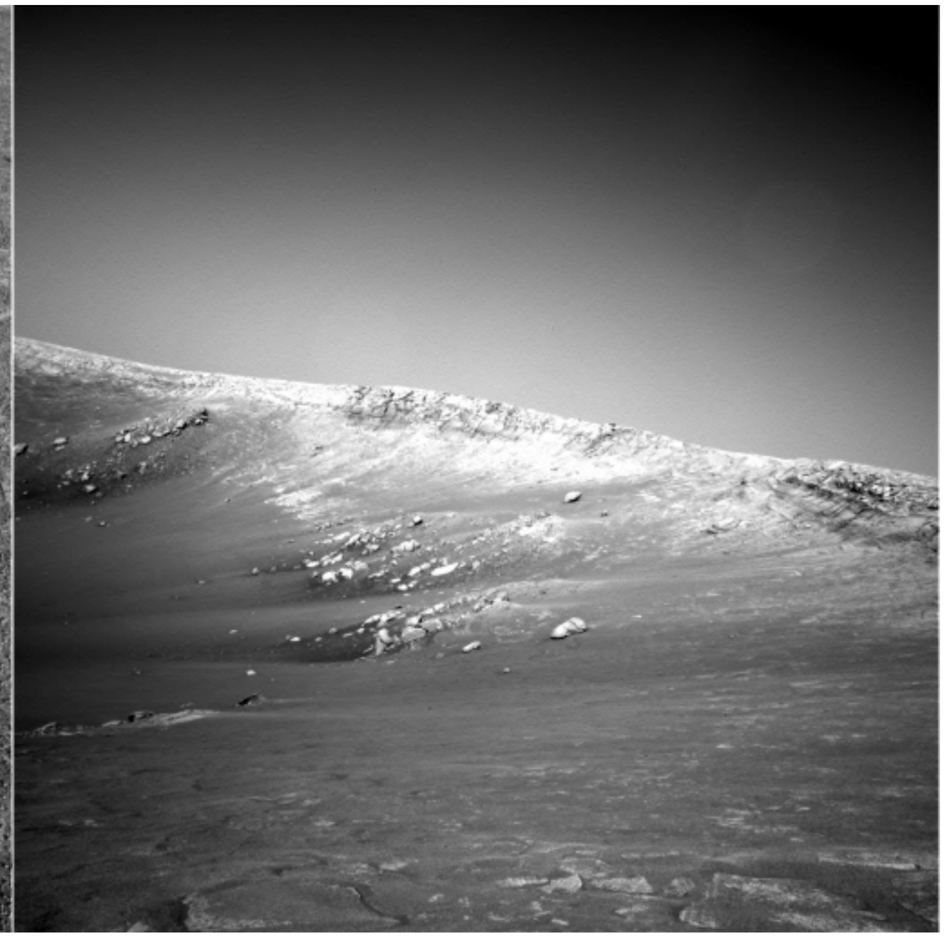
Harder case

by Diva Sian

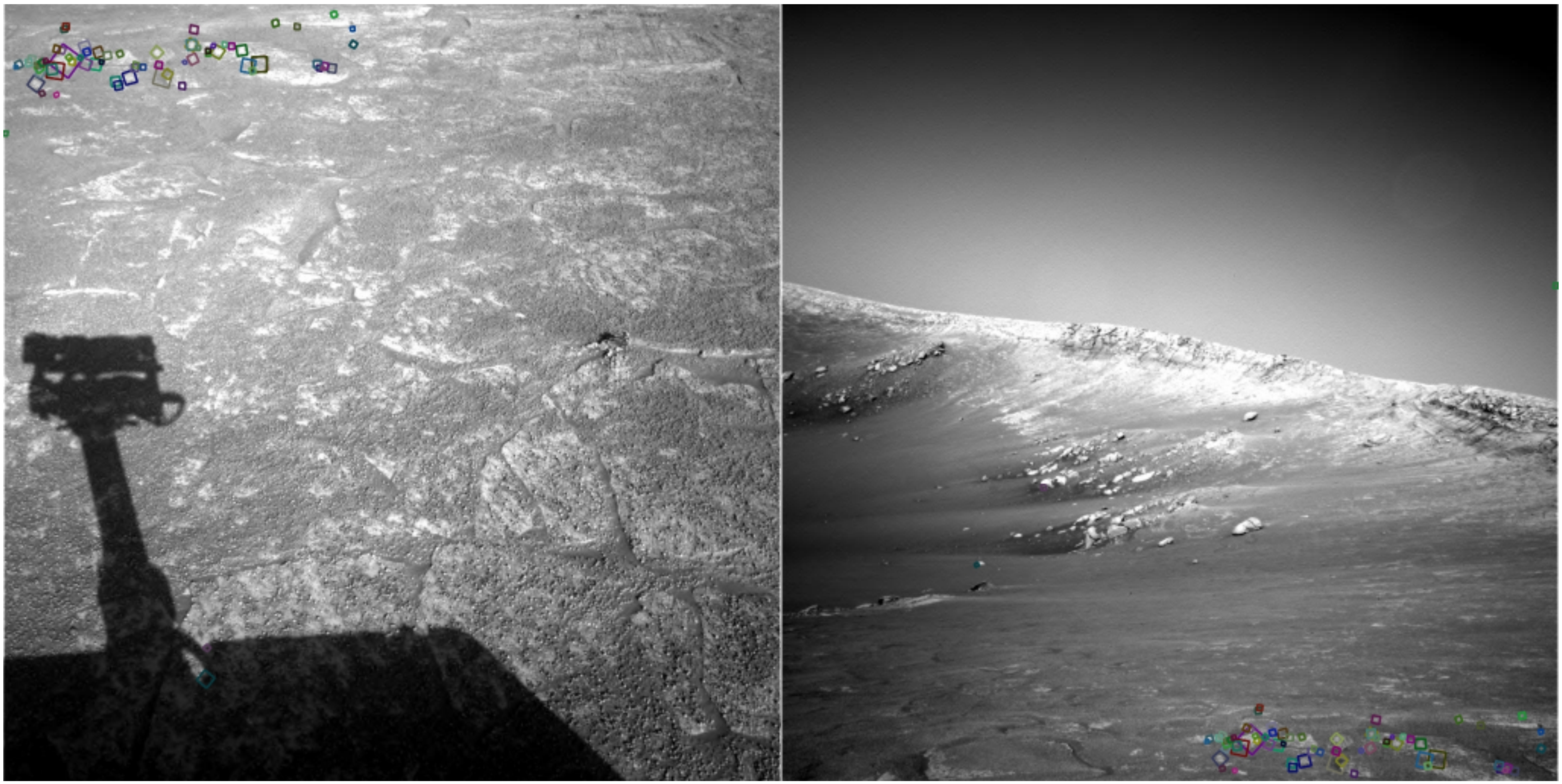
by scgbt

Harder still?





Answer below (look for tiny colored squares…)



NASA Mars Rover images
with SIFT feature matches

Approach

Feature detection: find it

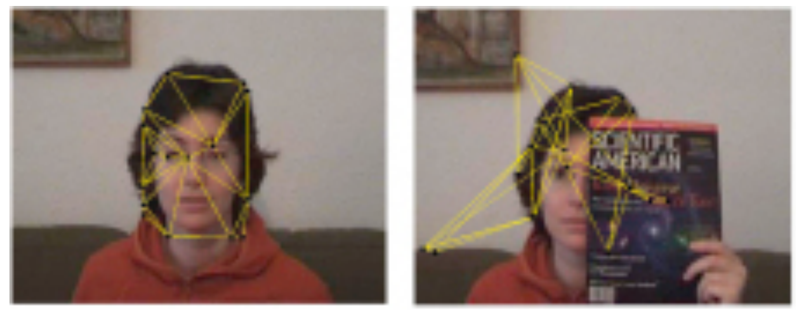
Feature descriptor: represent it

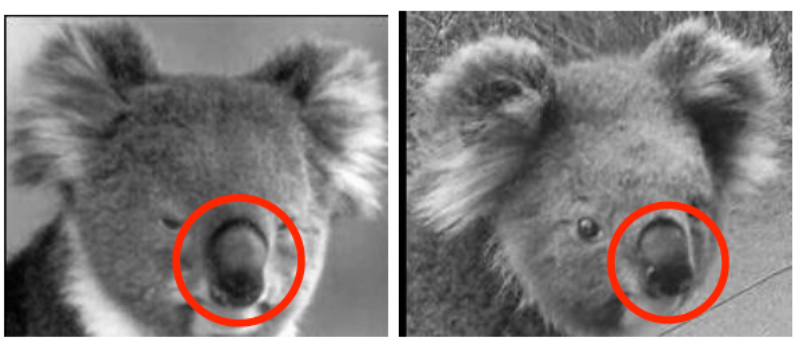
Feature matching: match it

Feature tracking: track it, when motion

Features

- Global vs. local representations
- Local: describe and match local regions
- Robust to
  - Occlusions
  - Articulation
  - Intra-category variation





Fei-Fei Li

Advantages of local features

Locality

– features are local, so robust to occlusion and clutter

Quantity

– hundreds or thousands in a single image

Distinctiveness:

– can differentiate a large database of objects

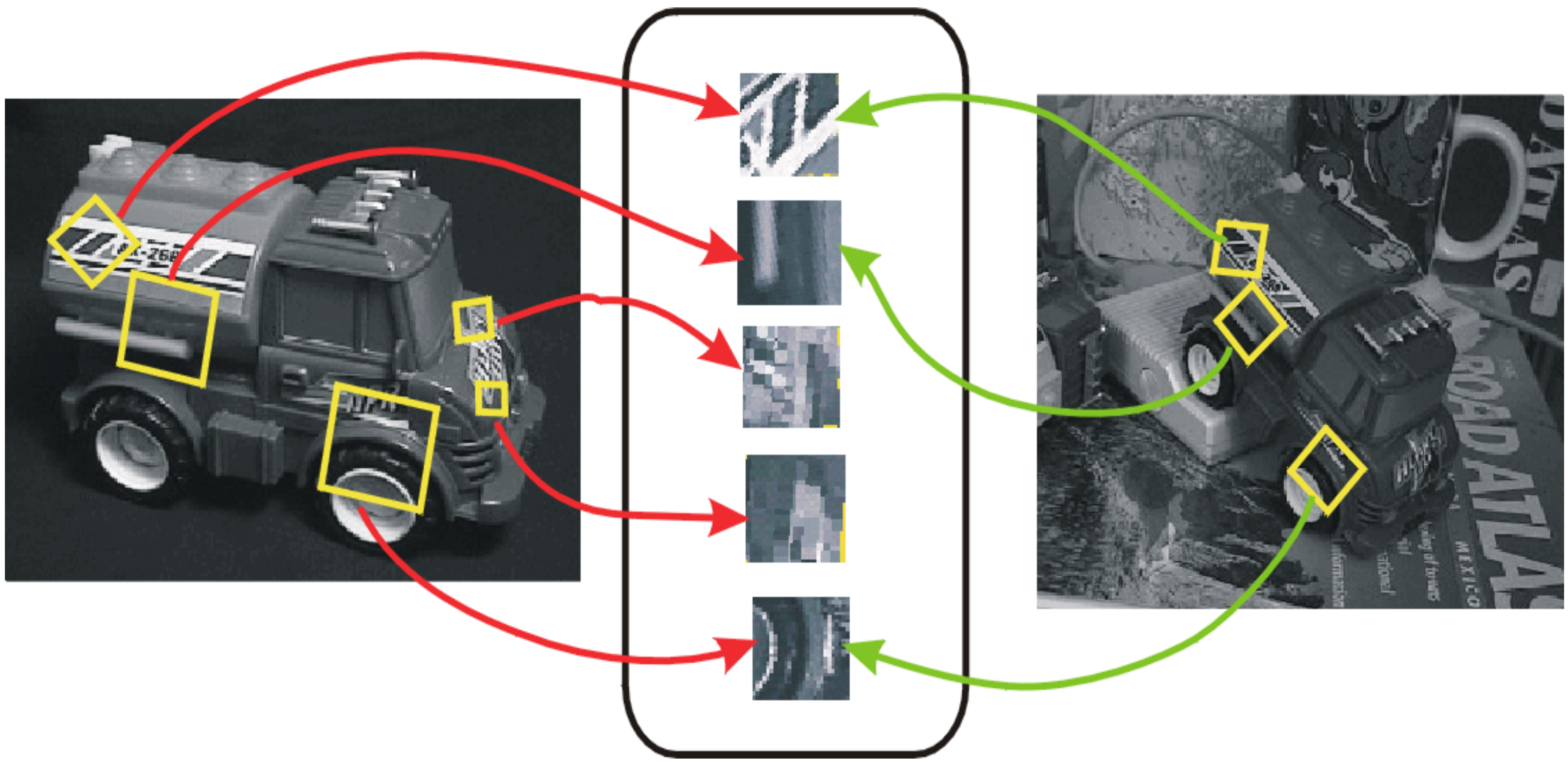
Efficiency

– real-time performance achievable

Invariant local features

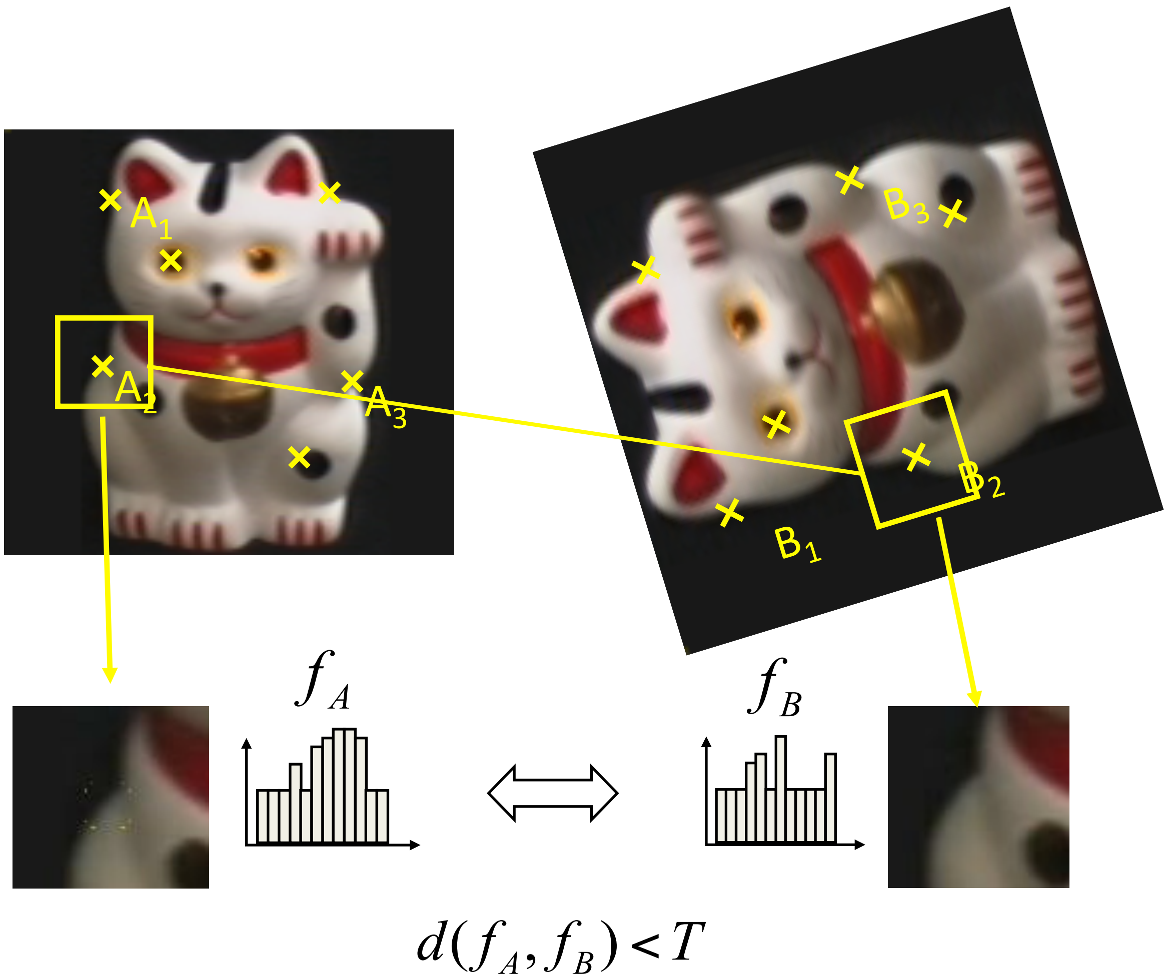
Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …



Feature Descriptors

Overview



1. **Find a set of distinctive features**
2. **Define a region around each feature**
3. **Extract and normalize the region content**
4. **Compute a local descriptor from the normalized region**
5. **Match local descriptors**

Goals for Features

Detect points that are *repeatable* and *distinctive*