

It's the features!

Nikhil Rasiwasia

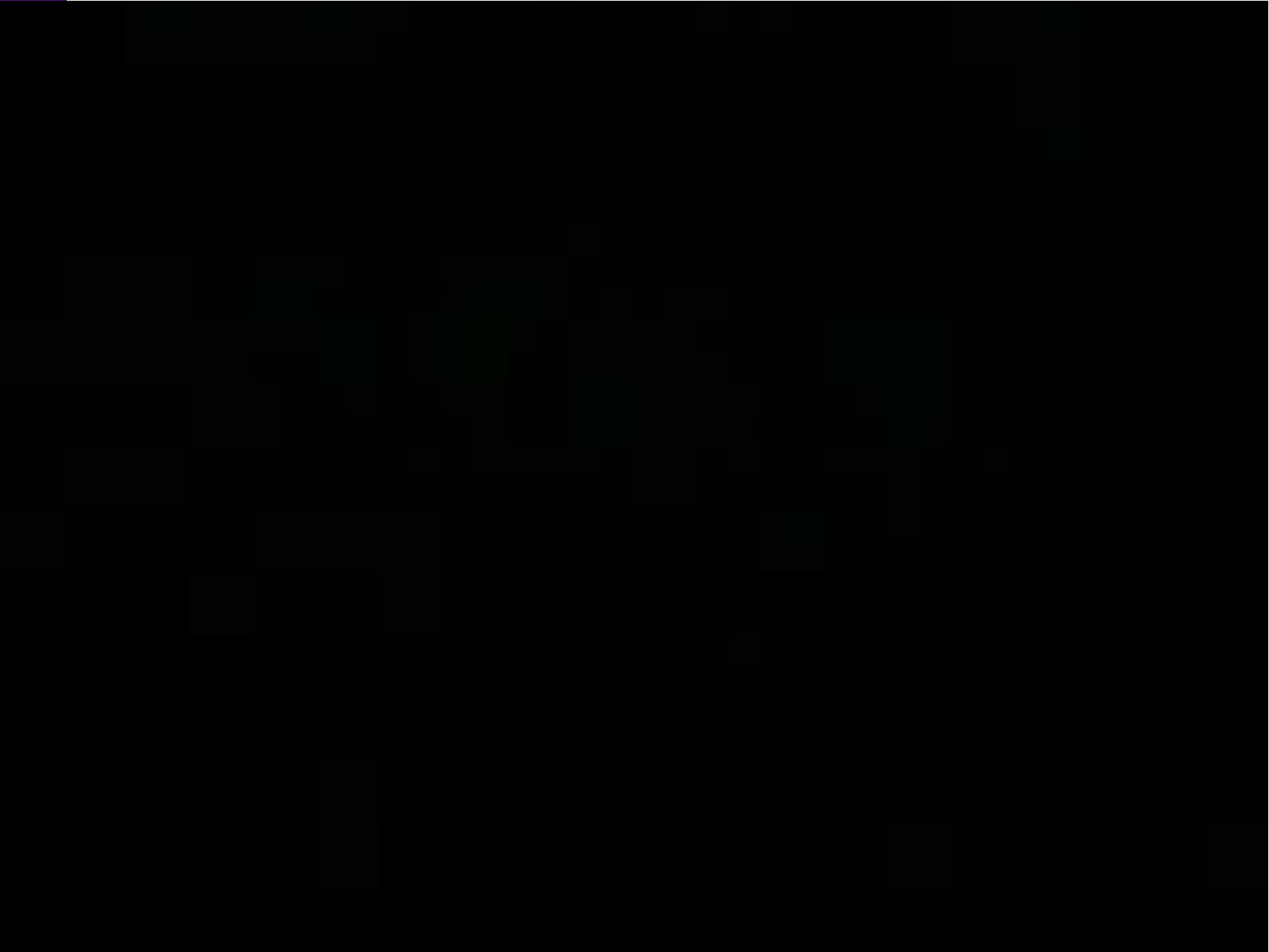
Scientist

Yahoo! Labs Bangalore

Slides contributed by Gaurav Aggarwal, Yahoo! Labs



YAHOO!



Features, Features, Features

In almost every case:

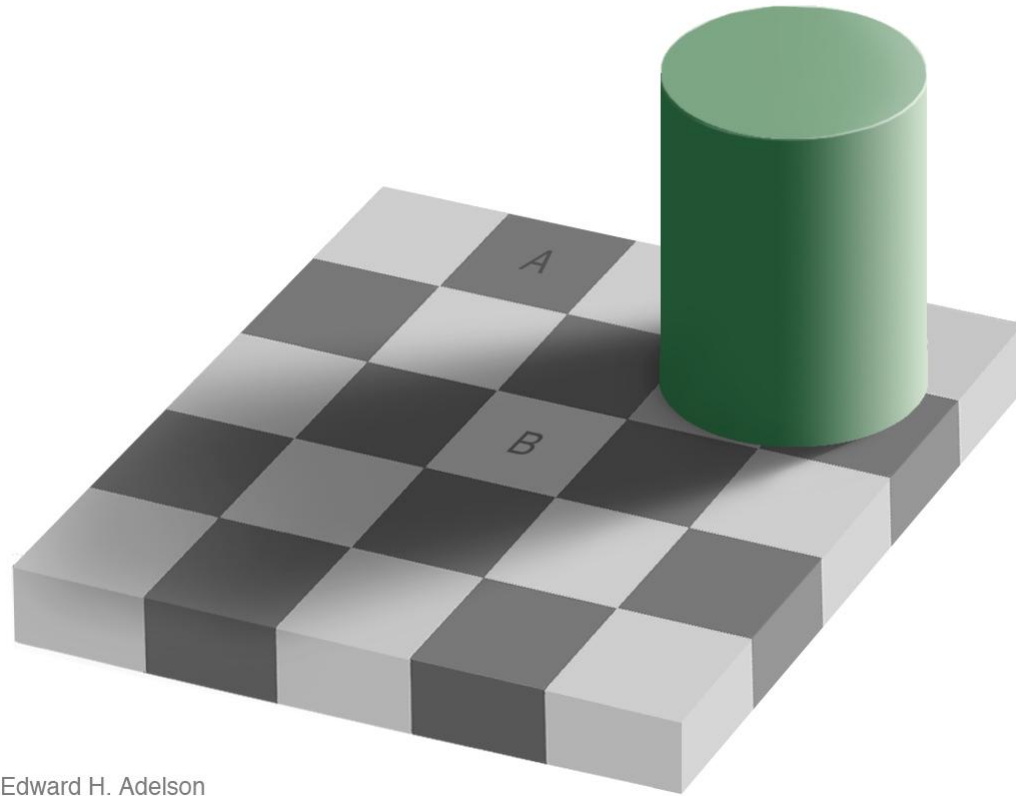
Good Features	beat	Good Learning
Learning	beats	No Learning

(Viola 2003)

Why do we need features?

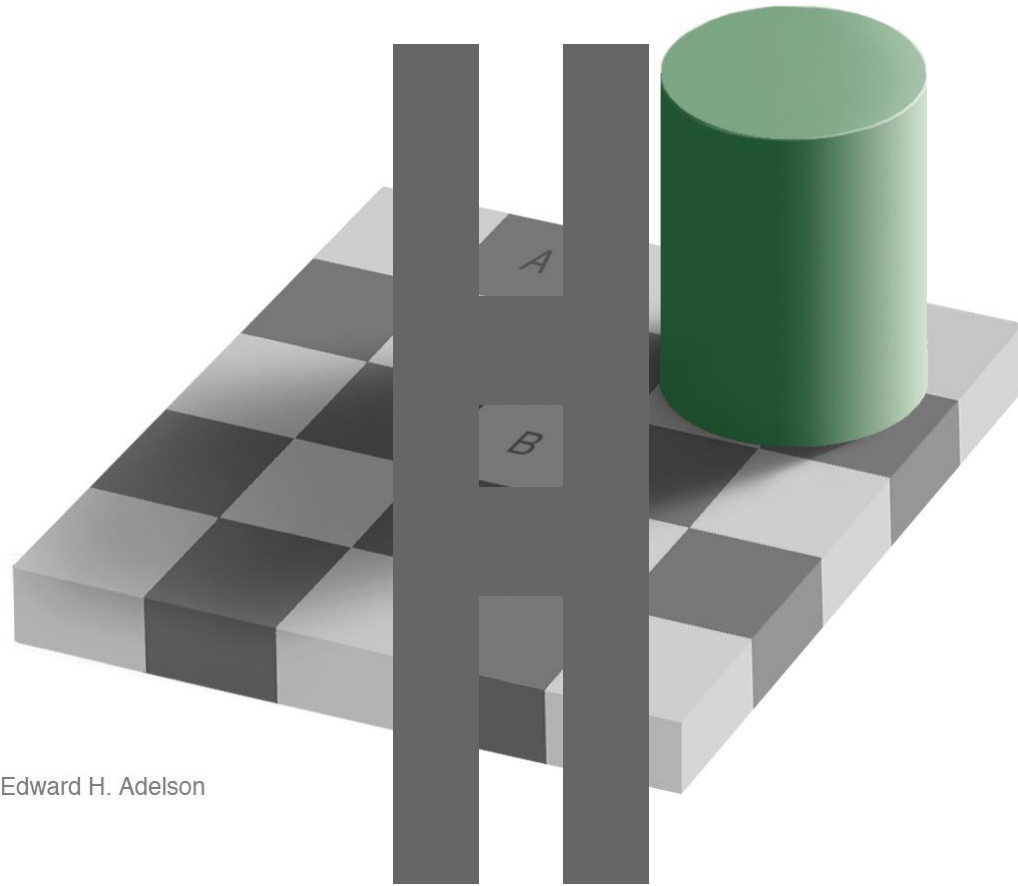


Brightness: Measurement vs. Perception



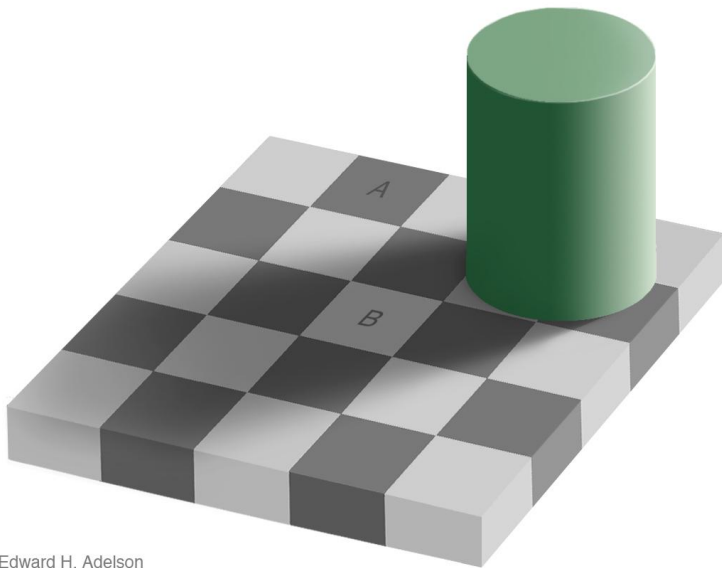
Edward H. Adelson

Brightness: Measurement vs. Perception

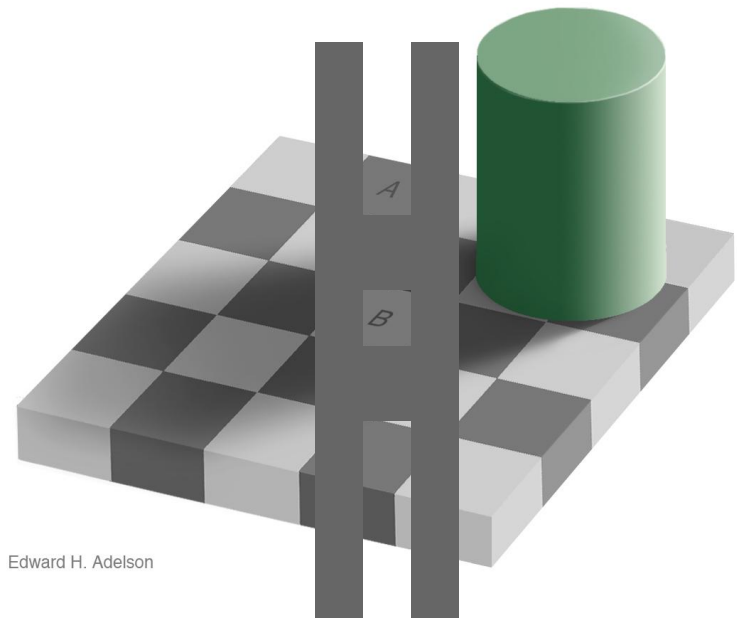


Edward H. Adelson

Brightness: Measurement vs. Perception



Edward H. Adelson



Edward H. Adelson

Proof!



A little story about Computer Vision (Recognition)



Founder, MIT AI project

In 1966, **Marvin Minsky** at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to **describe** what it saw”.

Recognize

We now know that the problem is more difficult than that.

(Szeliski, 2009, pg 33)



Challenges: view point variation



Pietà 1498-1499



Challenges: illumination

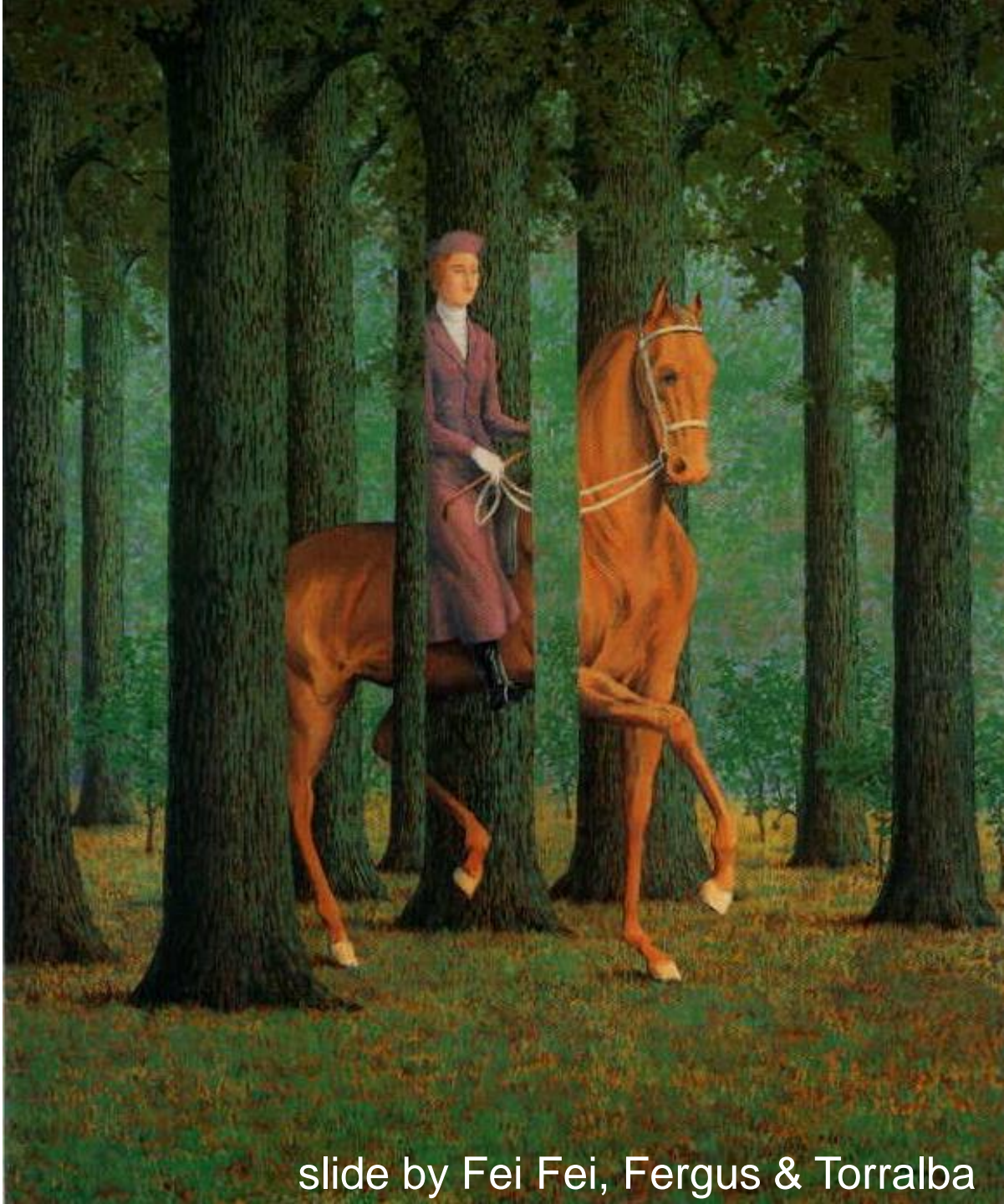




Challenges: occlusion



Magritte, 1957



slide by Fei Fei, Fergus & Torralba

Challenges: Texture grouping and segmentation





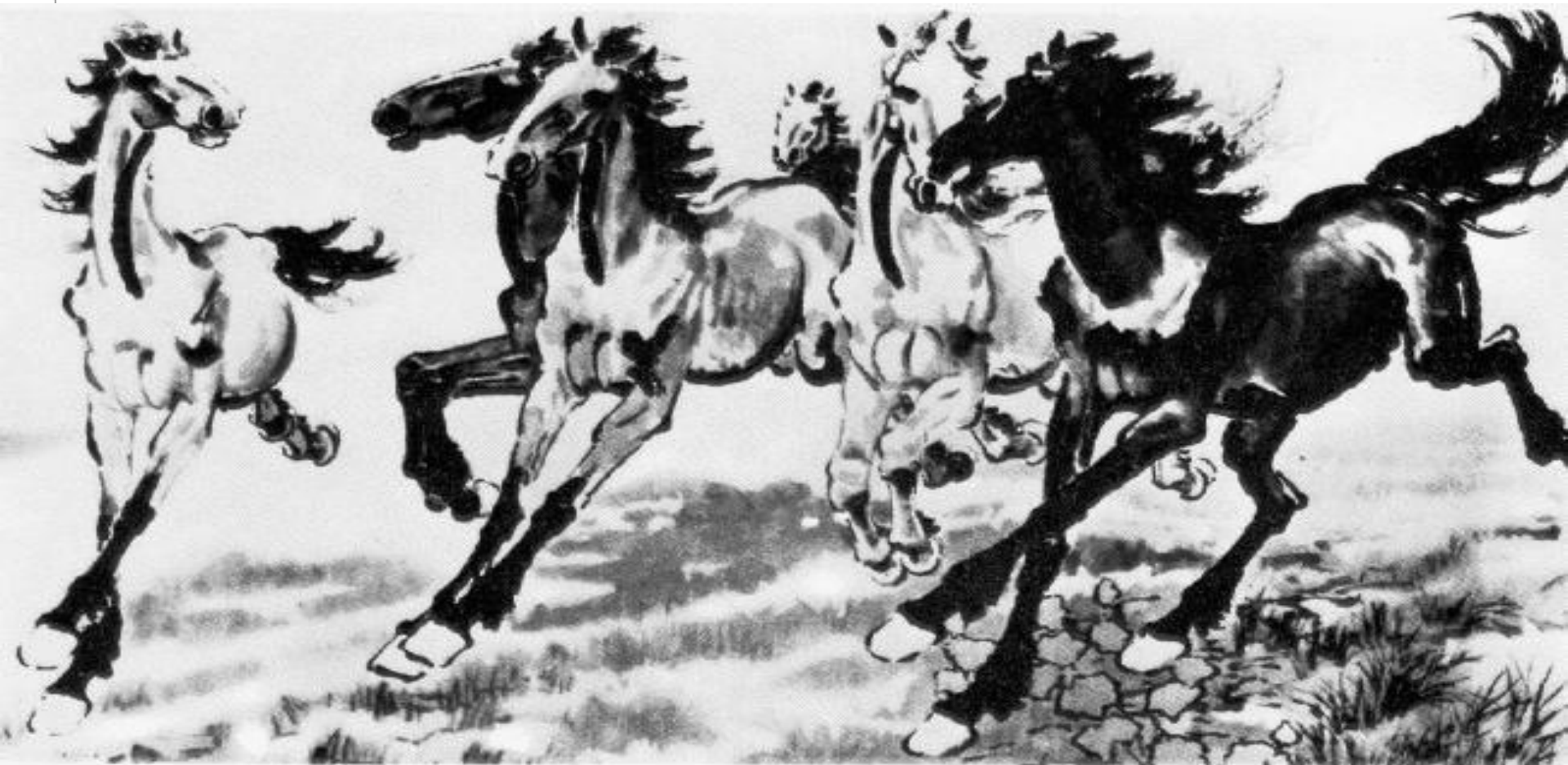
Challenges: scale



slide by Fei Fei, Fergus & Torralba



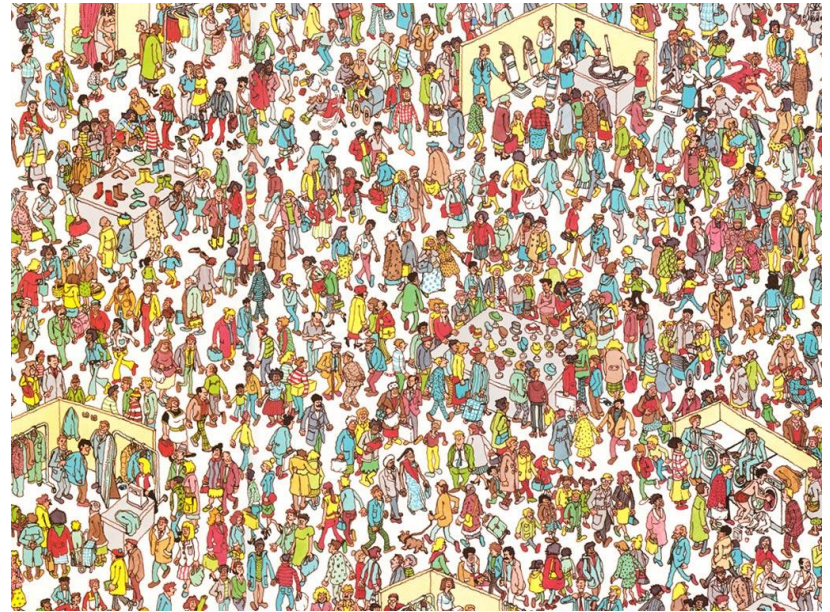
Challenges: deformation

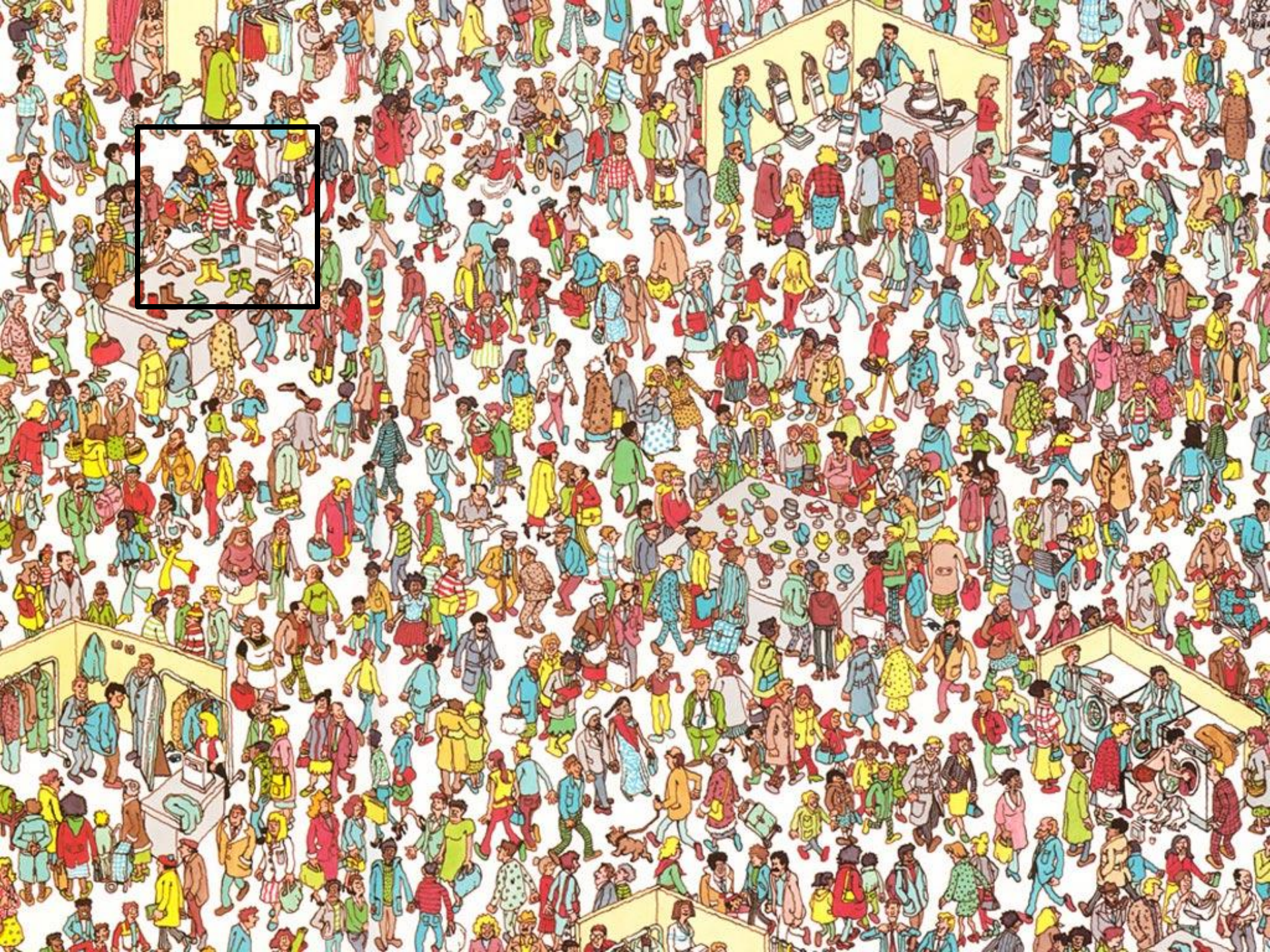


Challenges: background clutter



Where is Waldo?





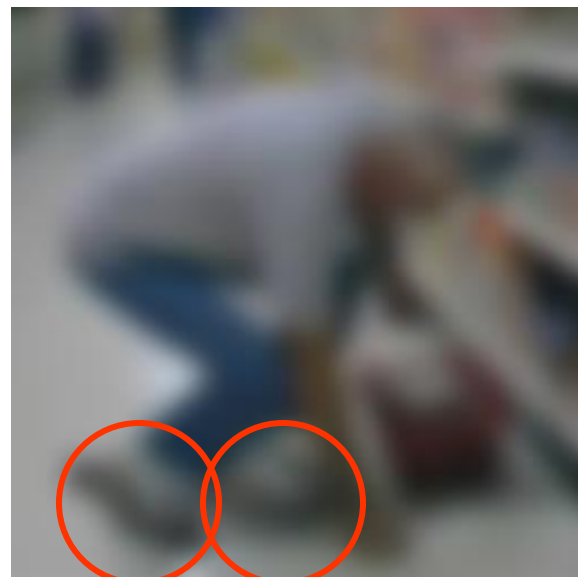
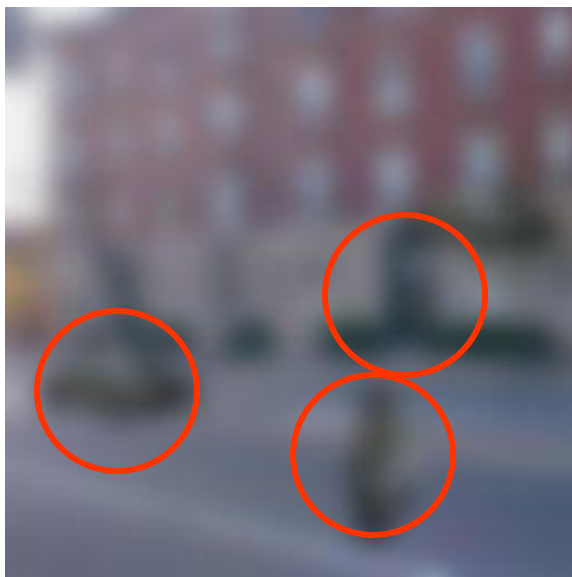
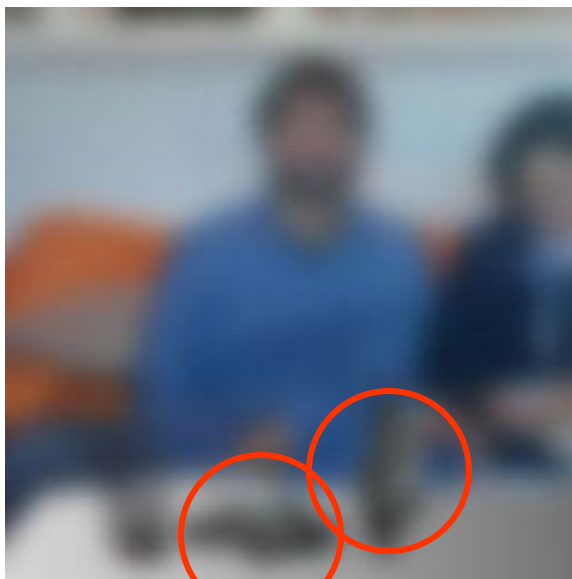
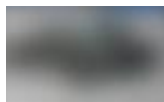


Challenges: object intra-class variation





Challenges: local ambiguity



slide by Fei-Fei, Fergus & Torralba

Challenges: Context

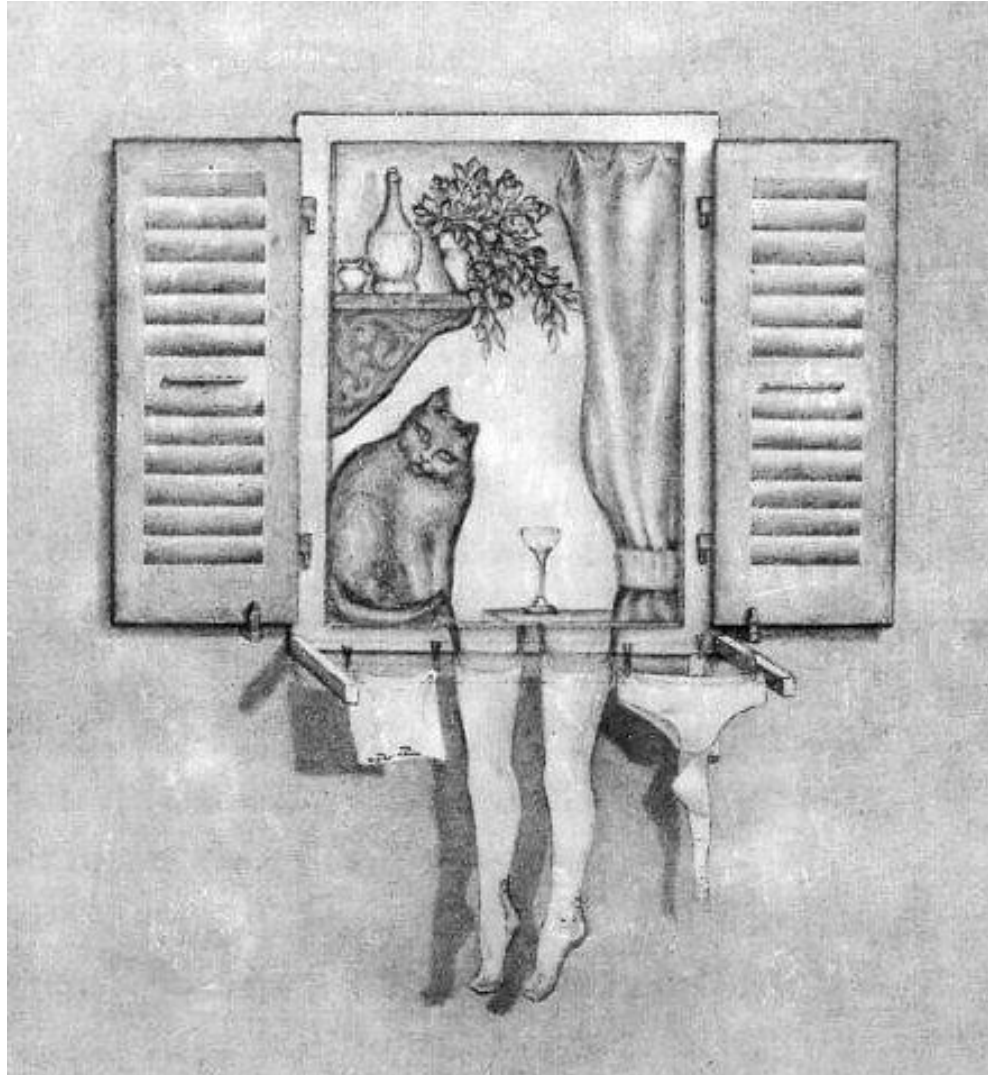
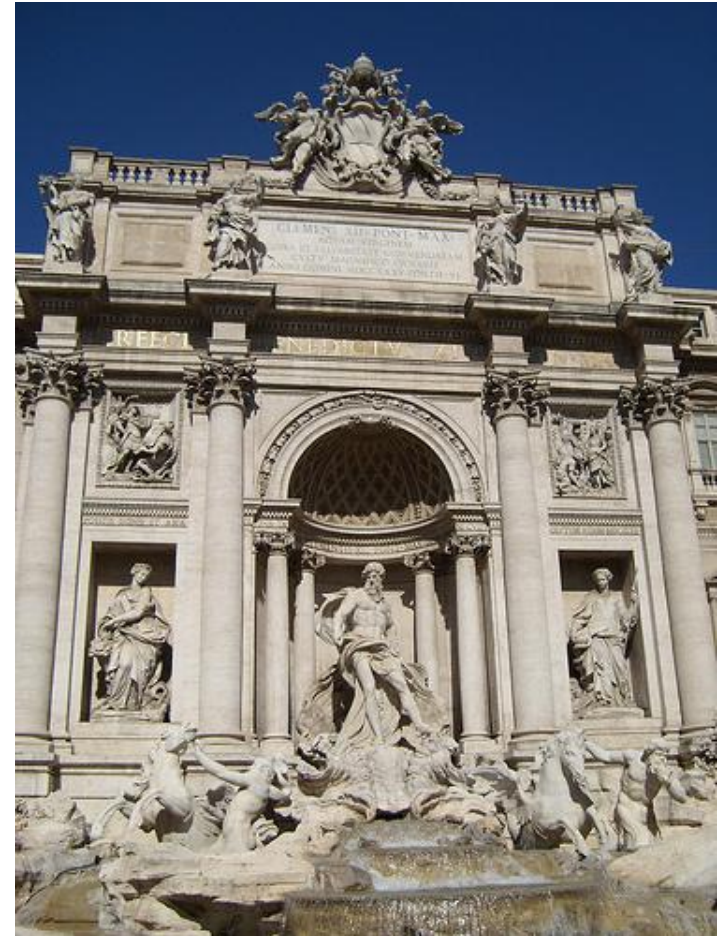


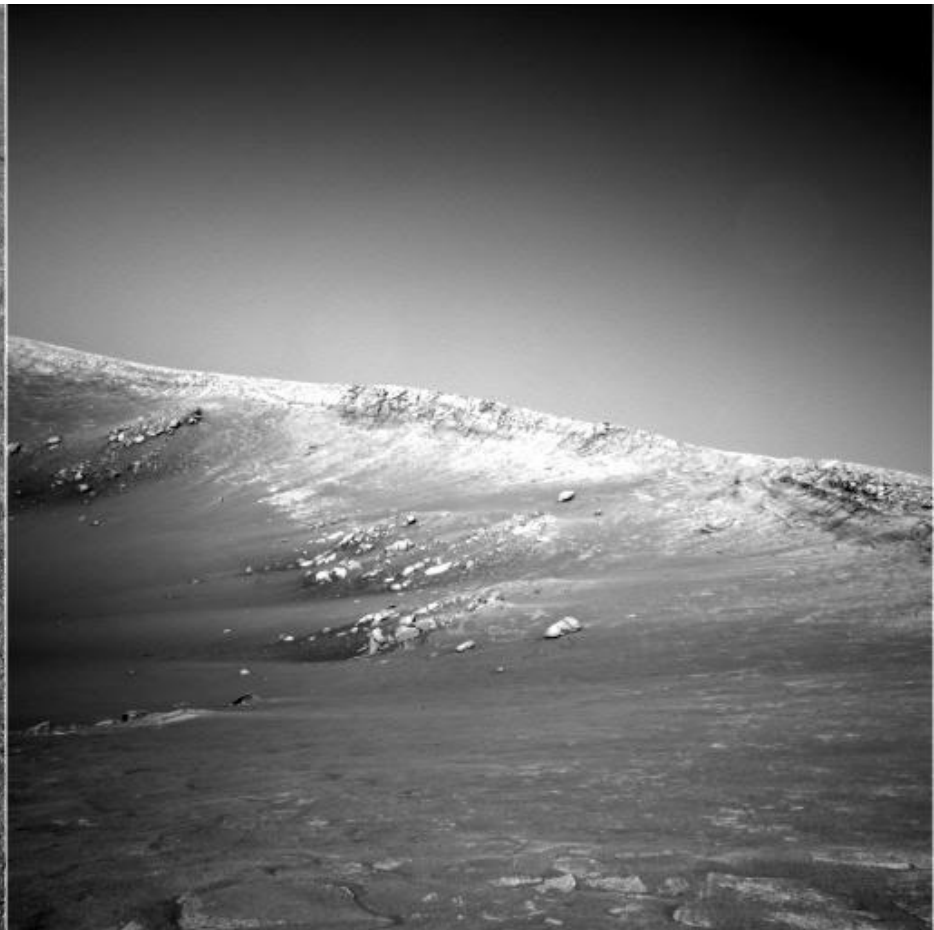
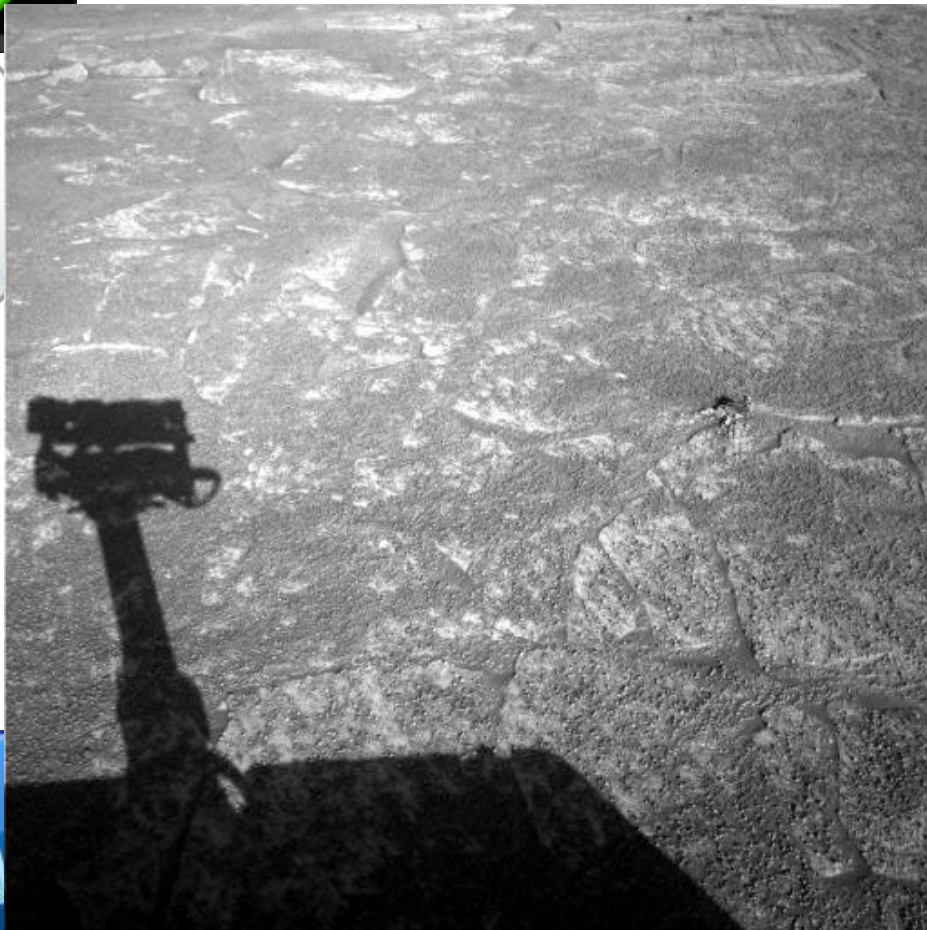
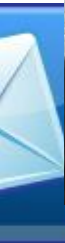
Image matching



Harder case



Harder!



CV works! (Look for tiny colored squares)

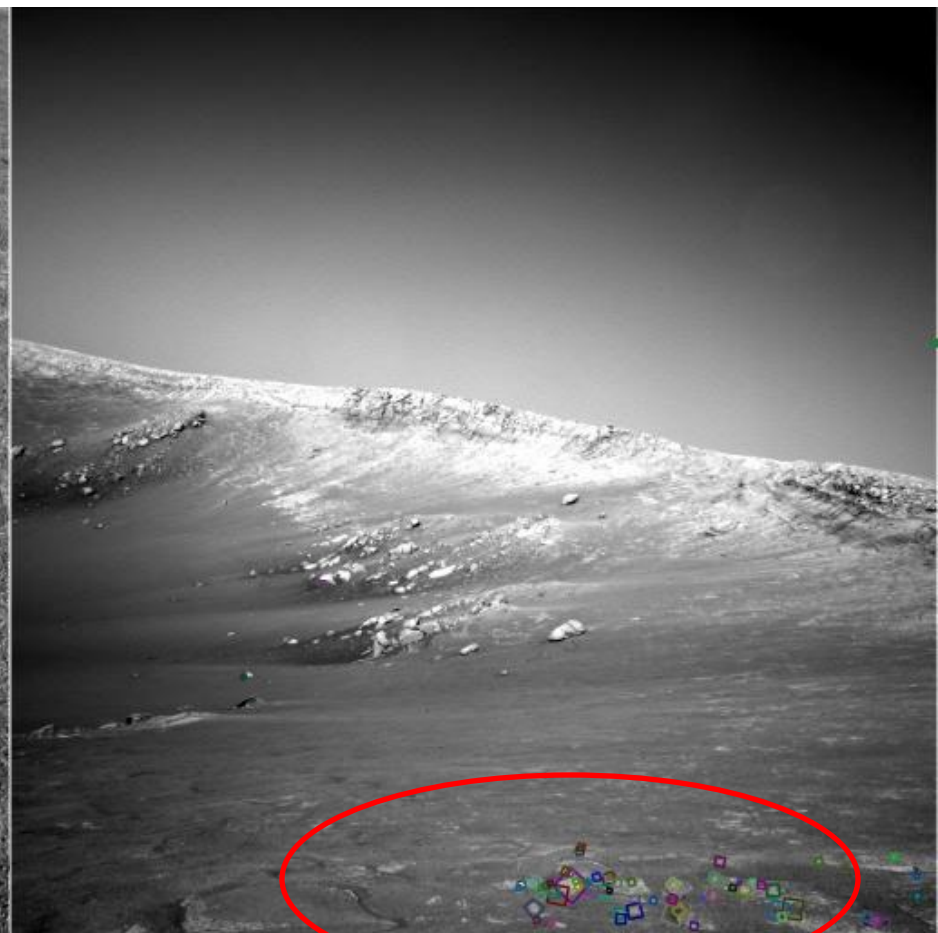
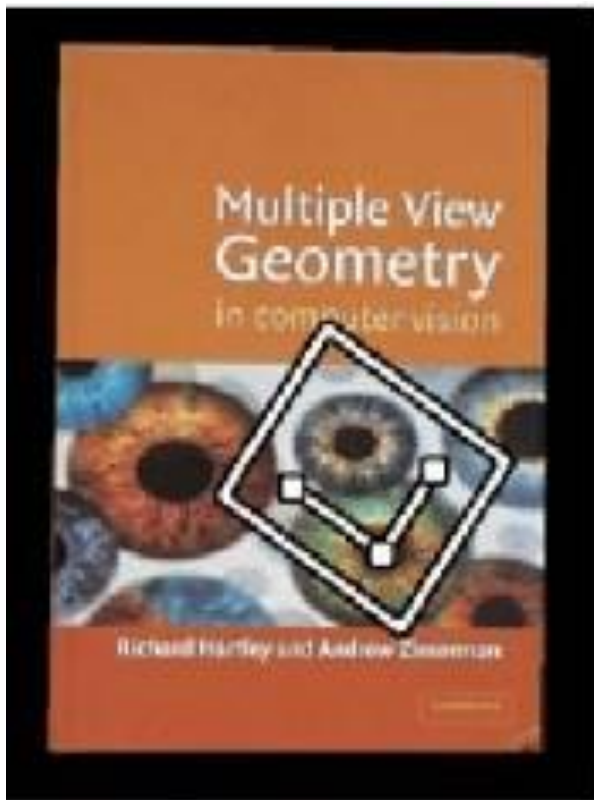
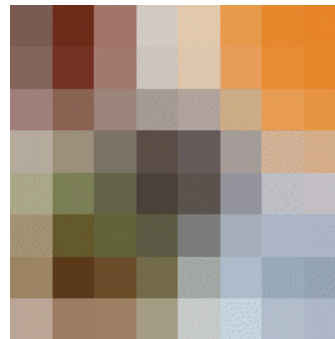
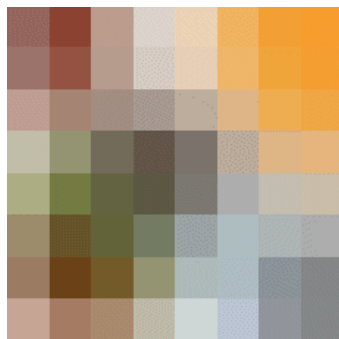
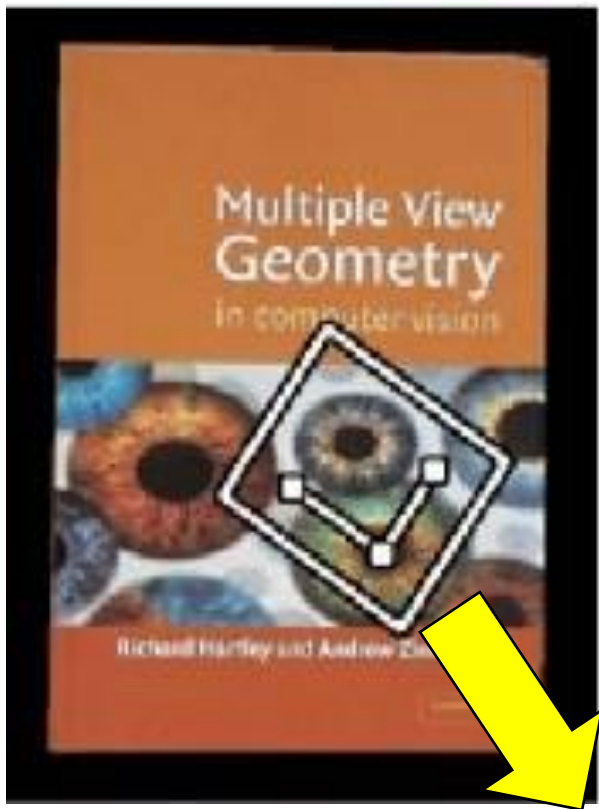


Image Matching

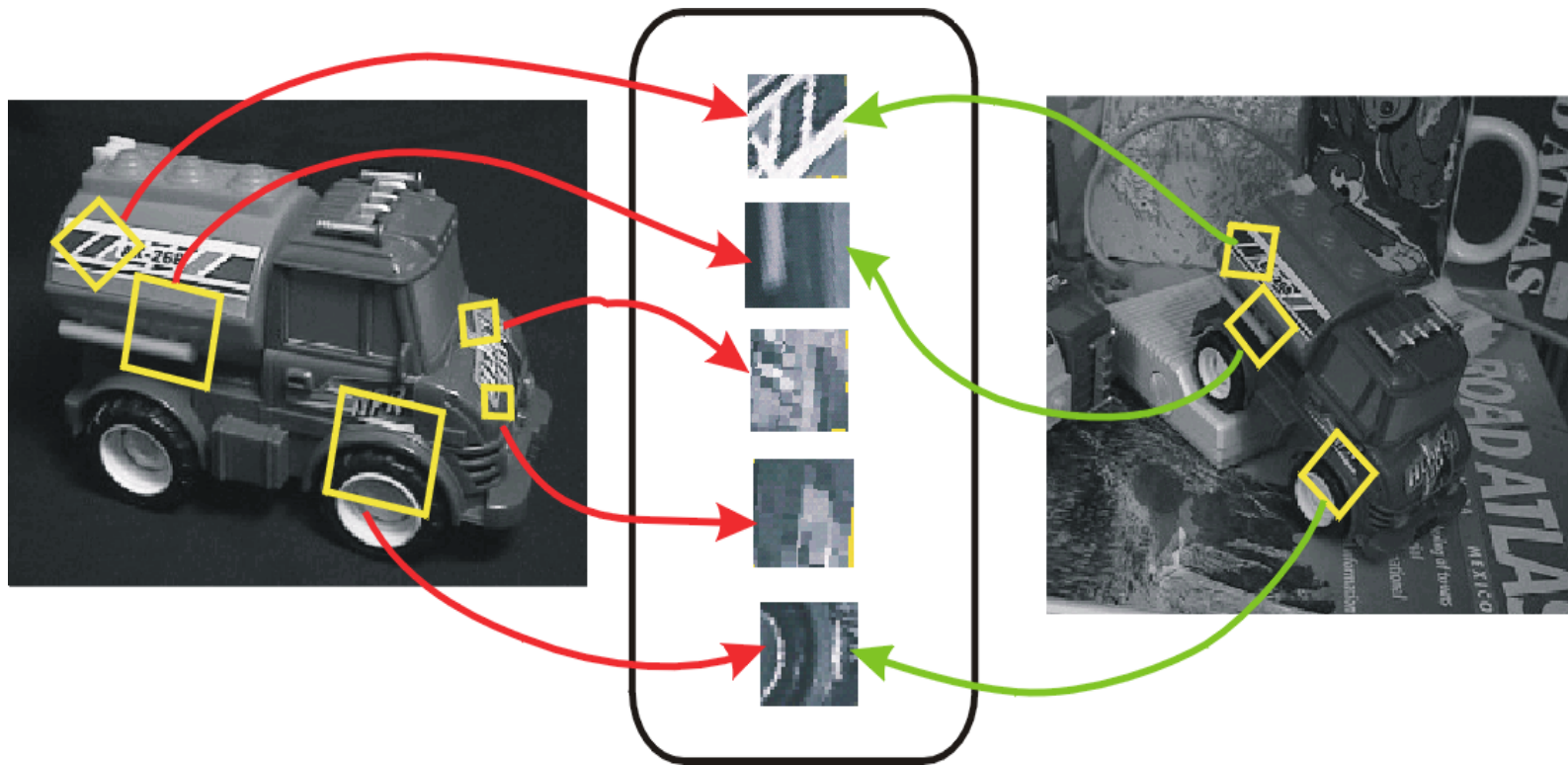


- 1) At an interesting point, let's define a coordinate system (x,y axis)
- 2) Use the coordinate system to pull out a patch at that point

Image Matching



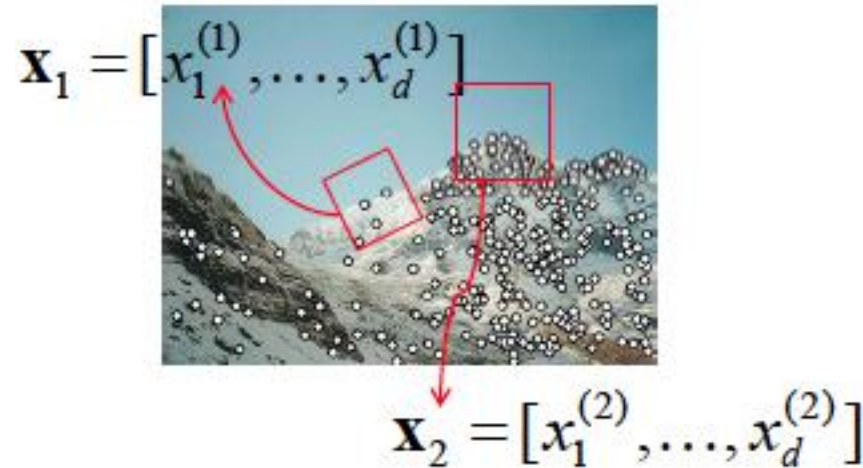
Invariant Local Features



Buzzword is invariance!

Local Features: main components

- Detection
 - Identify the interest points
- Description
 - Extract feature descriptor surrounding each point
- Matching
 - Determine correspondence between descriptors
 - (we will not cover this)
- Global Description
 - Bag-of-Words



Local Features: desired properties

- Repeatability
 - Can be found despite geometric and photometric transformations

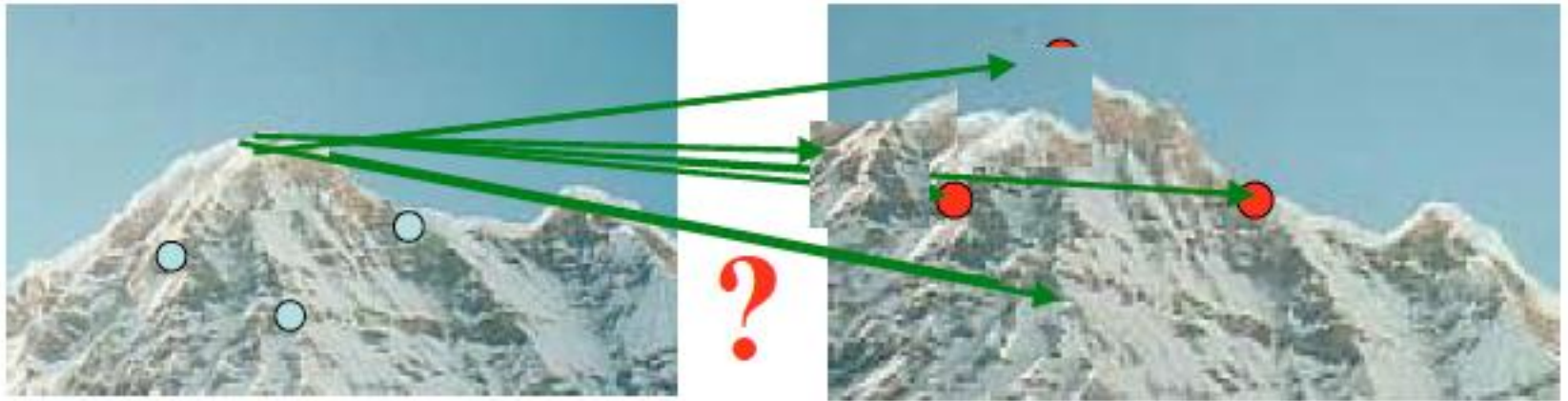


No chance to find true matches!

We have to be able to run the detection procedure *independently* per image.

Local Features: desired properties

- Saliency
 - Distinctive description



- Reliably determine which point goes with which.
- Must provide some invariance to geometric and photometric differences

Local Features: desired properties

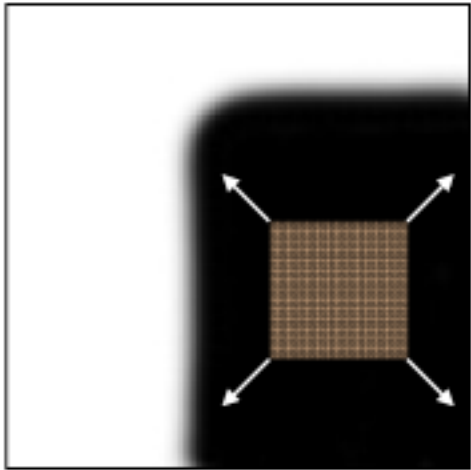
- Compactness and Efficiency
 - Many fewer features than image pixels
- Locality
 - Occupies relatively a small area in the image
 - Robust to clutter and occlusion

What points would you choose?

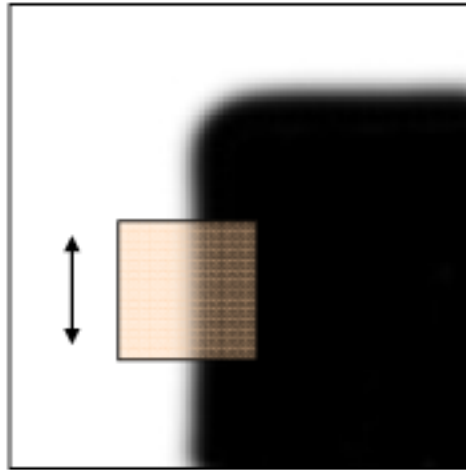


Uniqueness

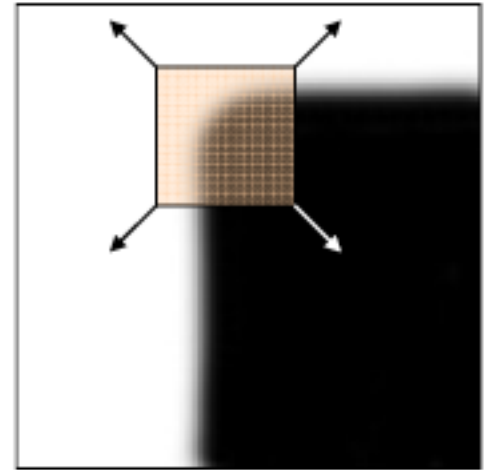
- How to define unique/unusual?
- Local measure of uniqueness
 - Corners as distinctive interest points



“flat” region:
no change in
all directions



“edge”:
no change
along the edge
direction

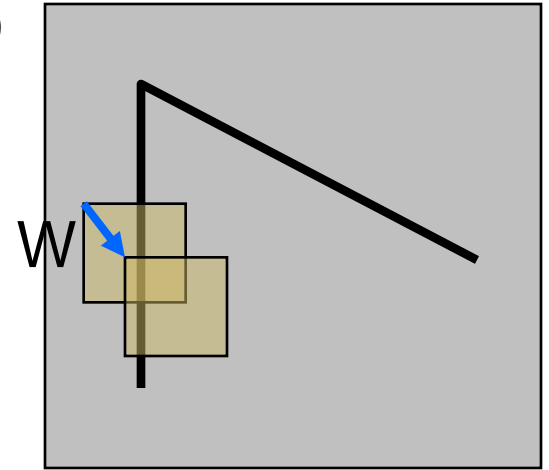


“corner”:
significant
change in all
directions

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD “error” of $E(u,v)$:



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

Small motion assumption

Taylor Series expansion of I:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

If the motion (u,v) is small, then first order approx is good

$$I(x + u, y + v) \approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

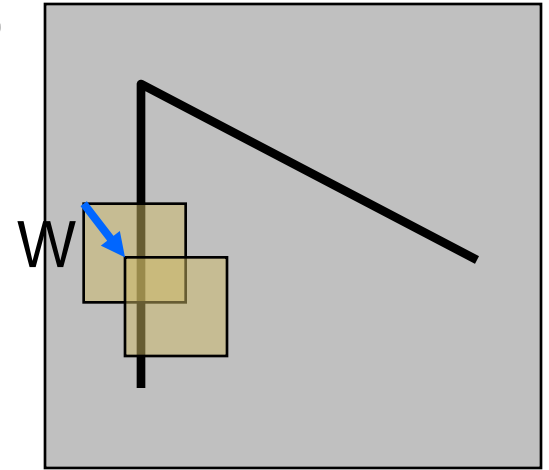
shorthand: $I_x = \frac{\partial I}{\partial x}$

Plugging this into the formula on the previous slide...

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an “error” of $E(u,v)$:

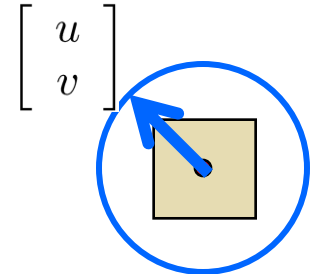
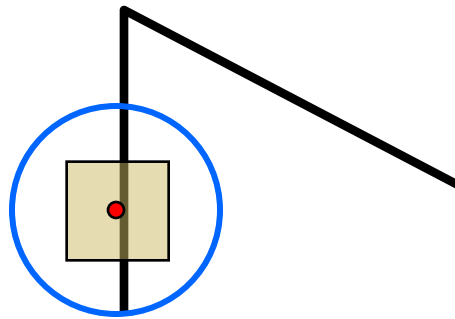


$$\begin{aligned} E(u, v) &= \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2 \\ &\approx \sum_{(x,y) \in W} [I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x, y)]^2 \\ &\approx \sum_{(x,y) \in W} \left[[I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2 \end{aligned}$$

Feature detection: the math

This can be rewritten:

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_H \begin{bmatrix} u \\ v \end{bmatrix}$$

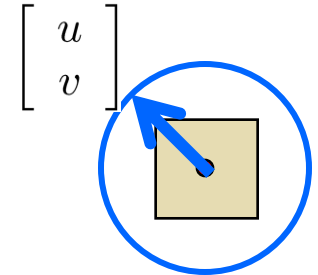
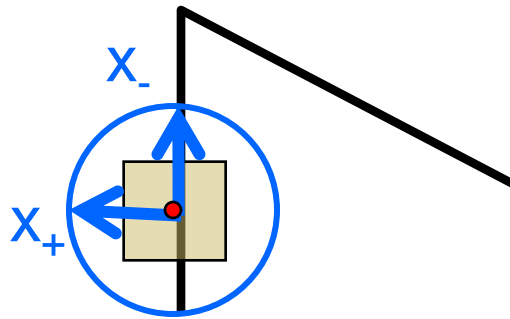


For the example above

- You can move the center of the green window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest E values?
- We can find these directions by looking at the eigenvectors of H

Feature detection: the math

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_H \begin{bmatrix} u \\ v \end{bmatrix}$$



Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of largest increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of smallest increase in E.
- λ_- = amount of increase in direction x_-

$$Hx_+ = \lambda_+ x_+$$

$$Hx_- = \lambda_- x_-$$

Feature detection: the math

How are λ_+ , x_+ , λ_- , and x_- relevant for feature detection?

- What's our feature scoring function?

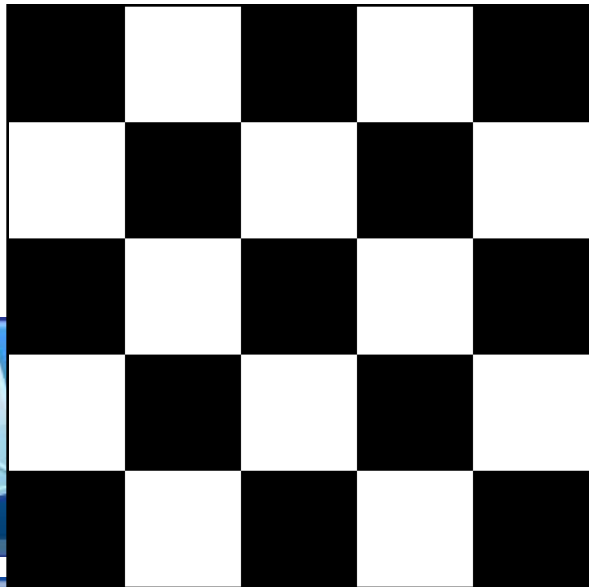
Feature detection: the math

How are λ_+ , x_+ , λ_- , and x_- relevant for feature detection?

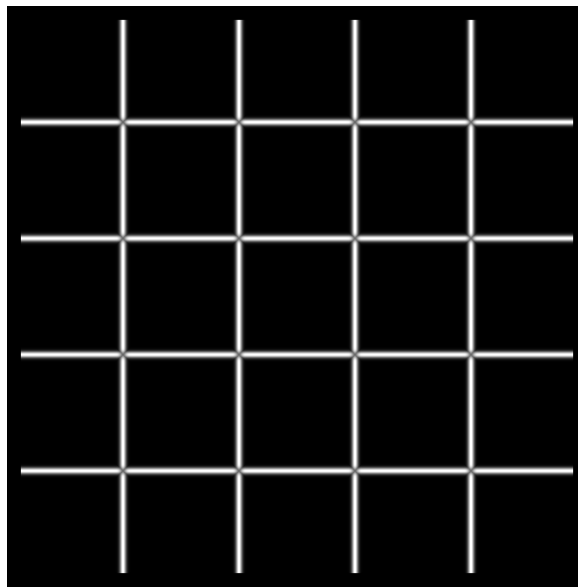
- What's our feature scoring function?

Want $E(u, v)$ to be *large* for small shifts in *all* directions

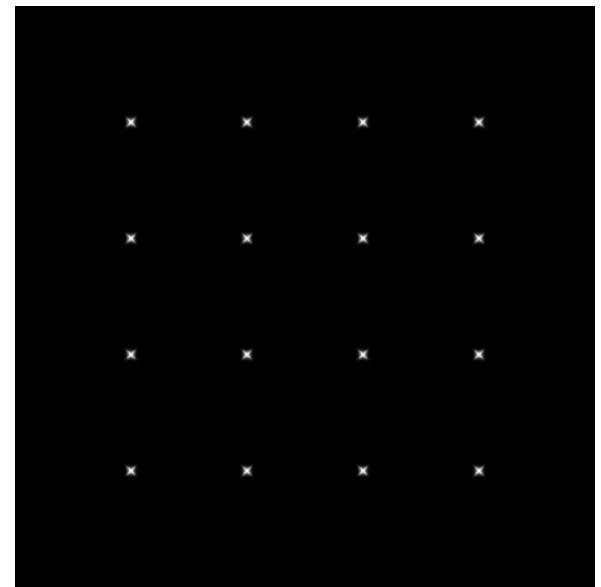
- the *minimum* of $E(u, v)$ should be large, over all unit vectors $[u \ v]$
- this minimum is given by the smaller eigenvalue (λ_-) of H



I



λ_+

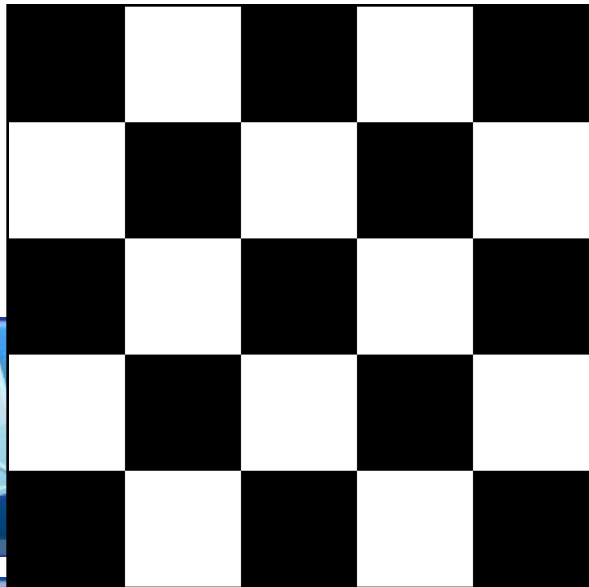


λ_-

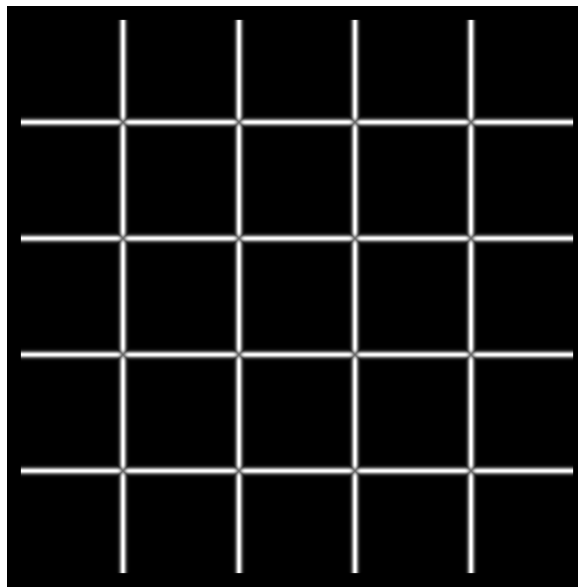
Feature detection summary

Here's what you do

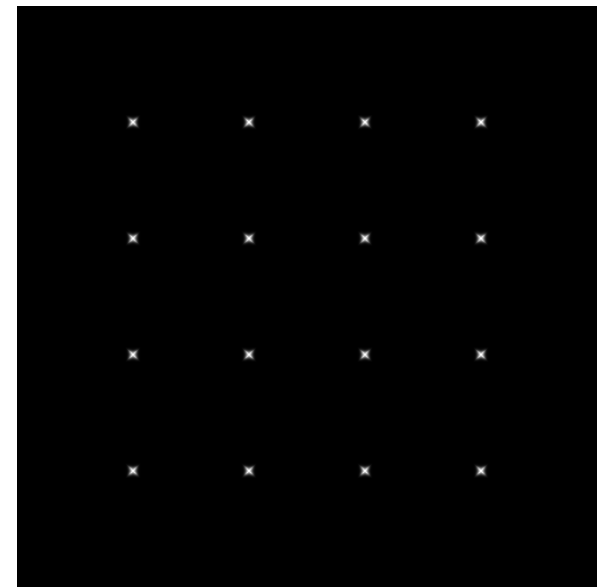
- Compute the gradient at each point in the image
- Create the H matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ($\lambda_- > \text{threshold}$)
- Choose those points where λ_- is a local maximum as features



I



λ_+

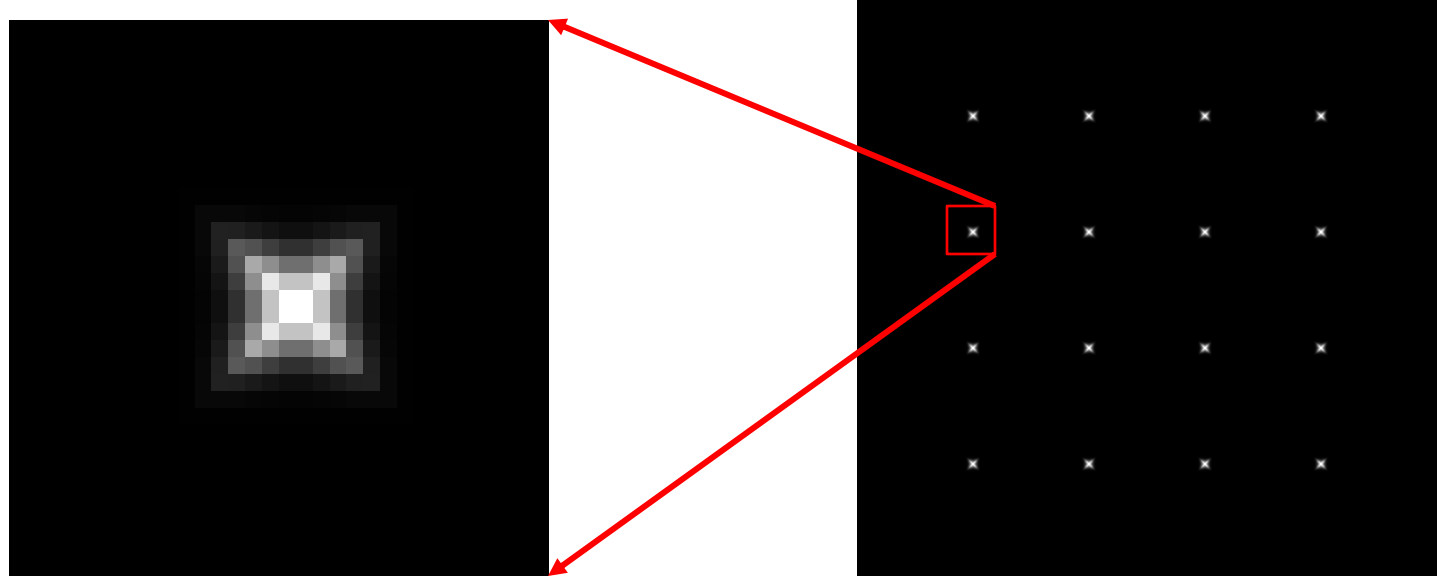


λ_-

Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the H matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ($\lambda_- > \text{threshold}$)
- Choose those points where λ_- is a local maximum as features



λ_-

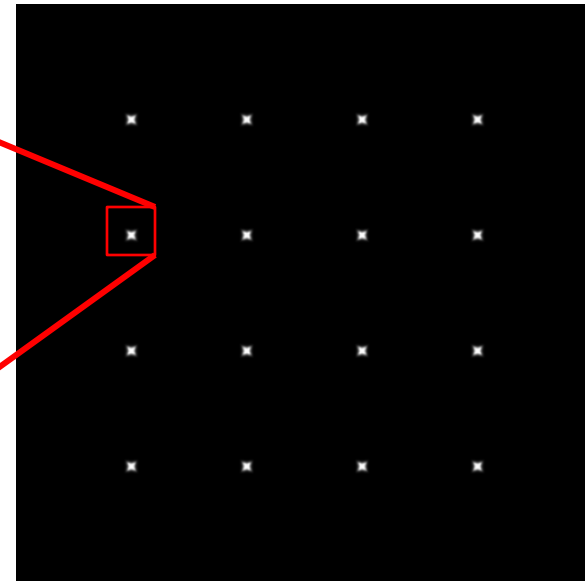
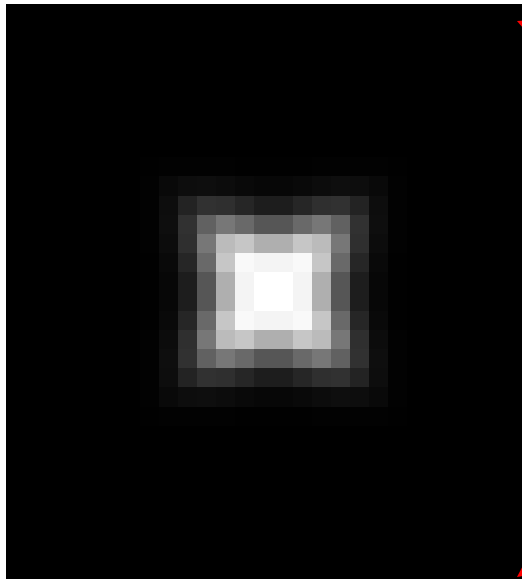
The Harris operator

λ_2 is a variant of the “Harris operator” for feature detection

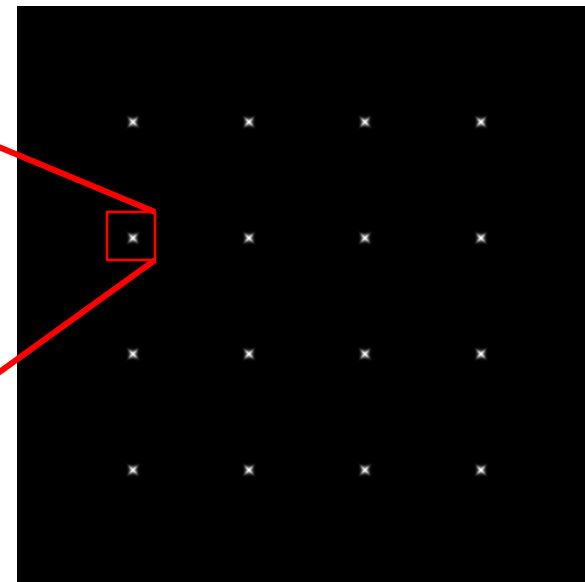
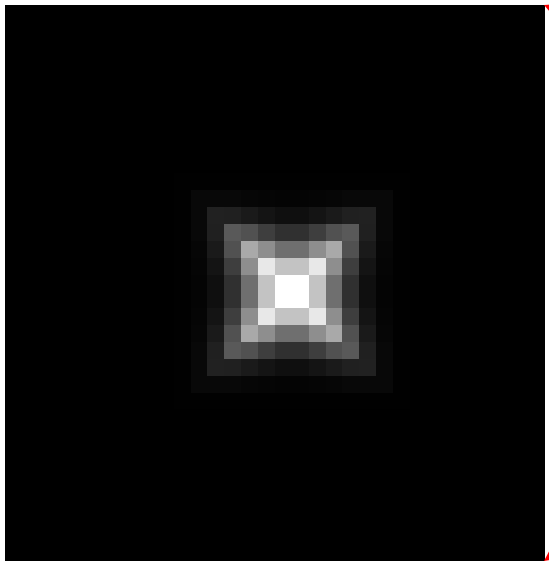
$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$
$$= \frac{\text{determinant}(H)}{\text{trace}(H)}$$

- The *trace* is the sum of the diagonals, i.e., $\text{trace}(H) = h_{11} + h_{22}$
- Called the “Harris Corner Detector” or “Harris Operator”
- Lots of other detectors, this is one of the most popular

The Harris operator

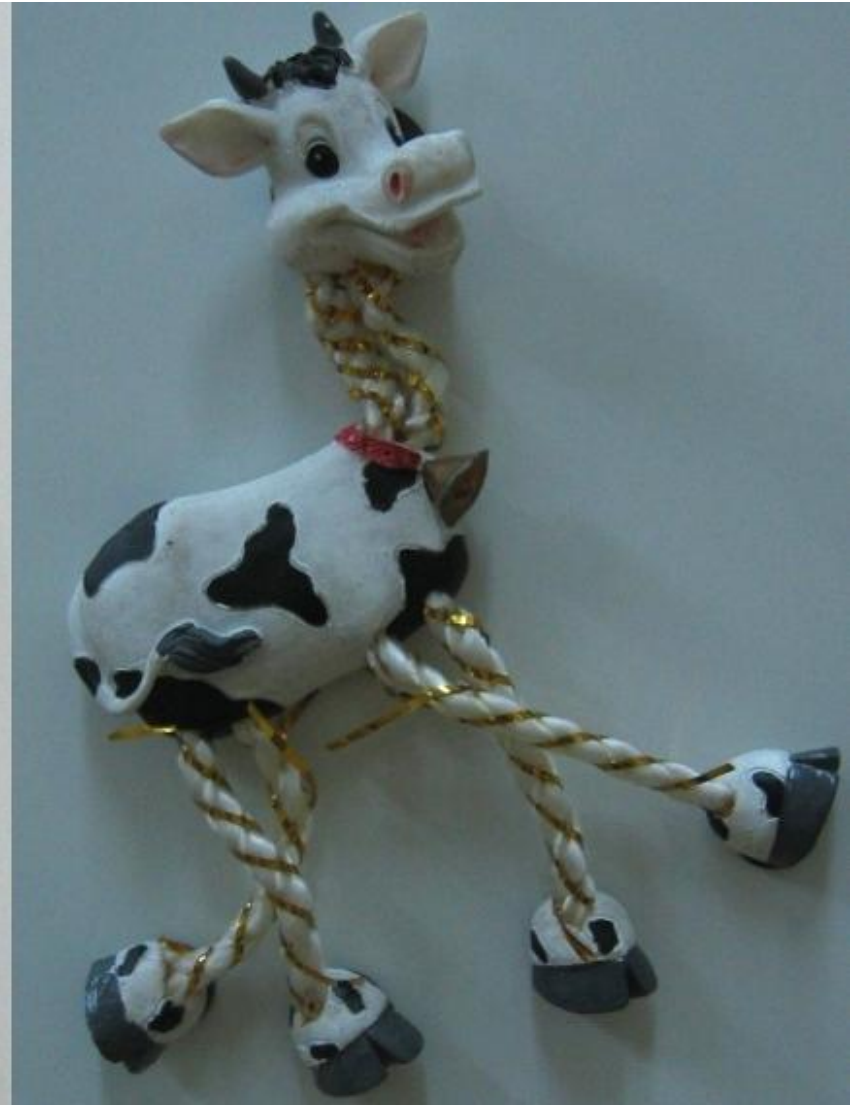


Harris
operator

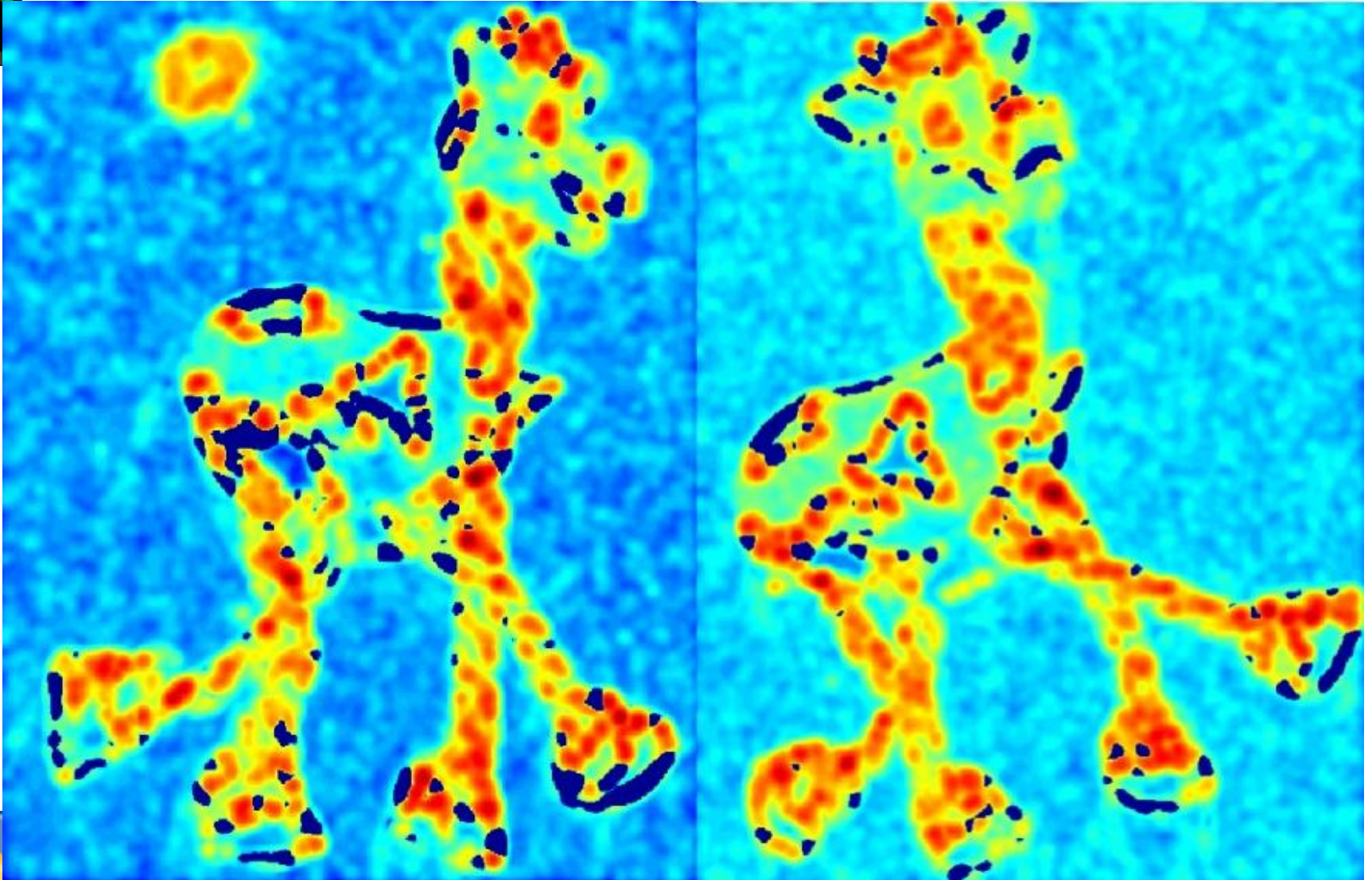


λ_-

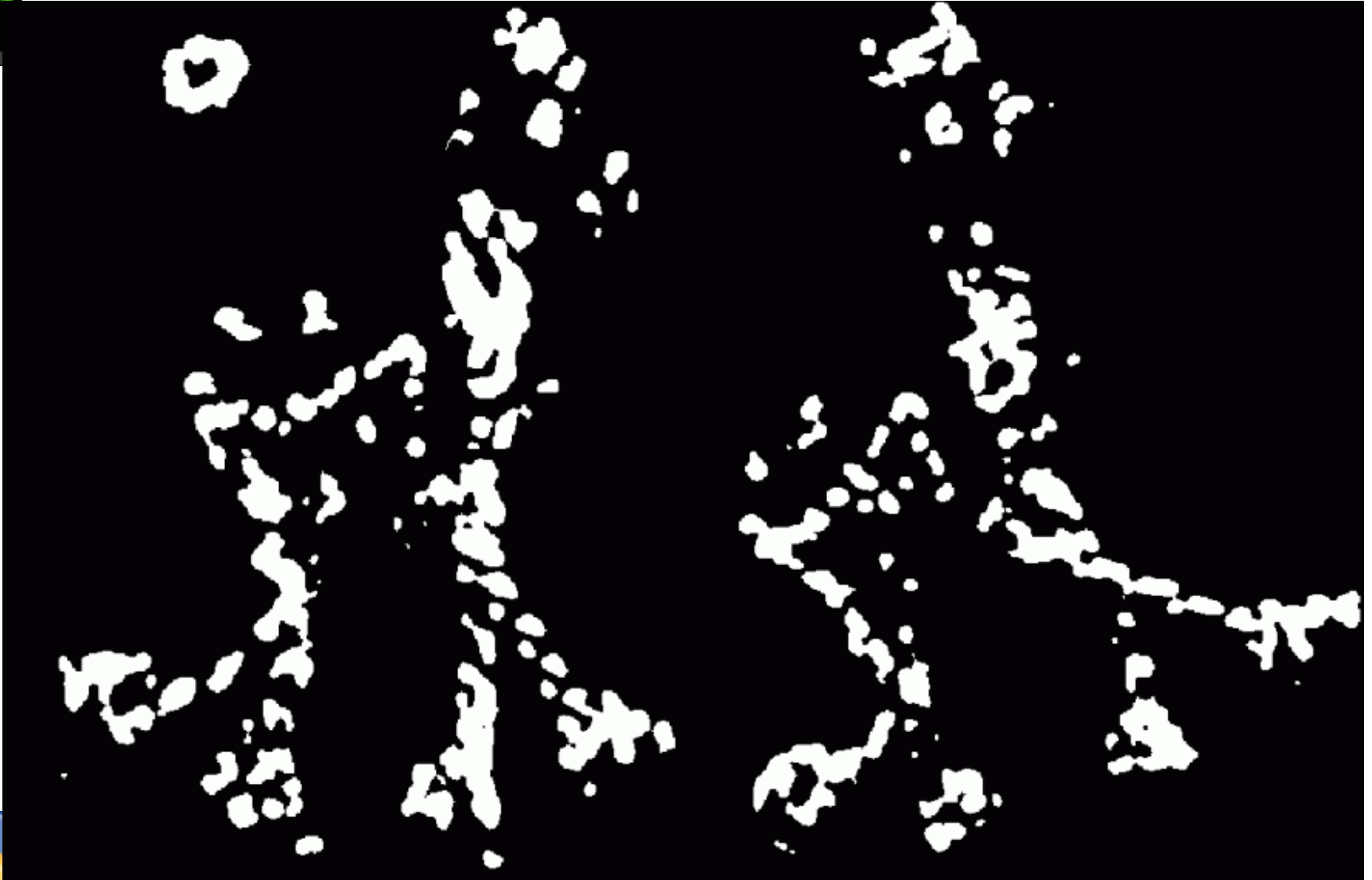
Harris detector example



f value (red high, blue low)



Threshold ($f > \text{value}$)



Find local maxima of f



Harris features (in red)



The tops of the horns are detected in both images

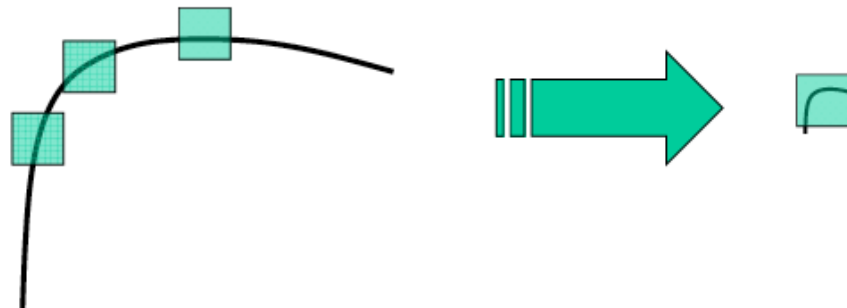
Invariance

Suppose you **rotate** the image by some angle

- Will you still pick up the same features?

What if you change the brightness?

Scale?



All points will be
classified as **edges**

Corner !

Scale-invariant Interest Points

- How can we independently select interest points in each image, such that the detections are repeatable across different scales?

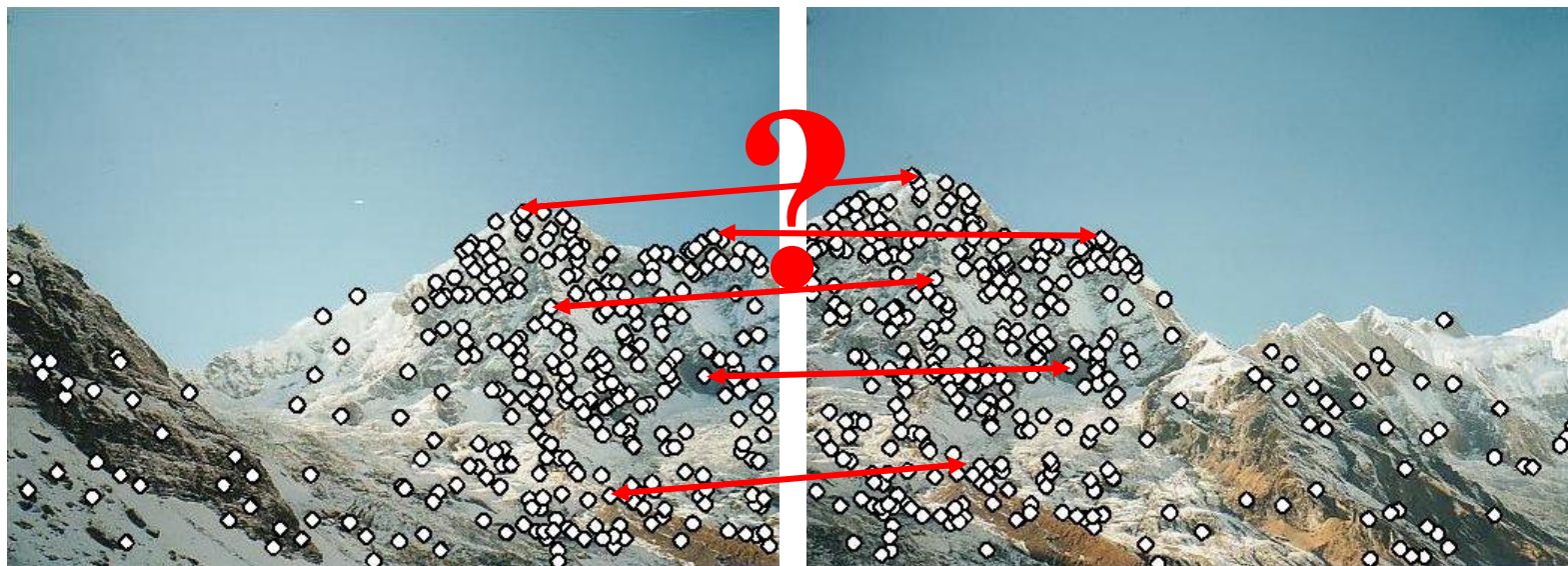


Will not be covered in this tutorial

Feature descriptors

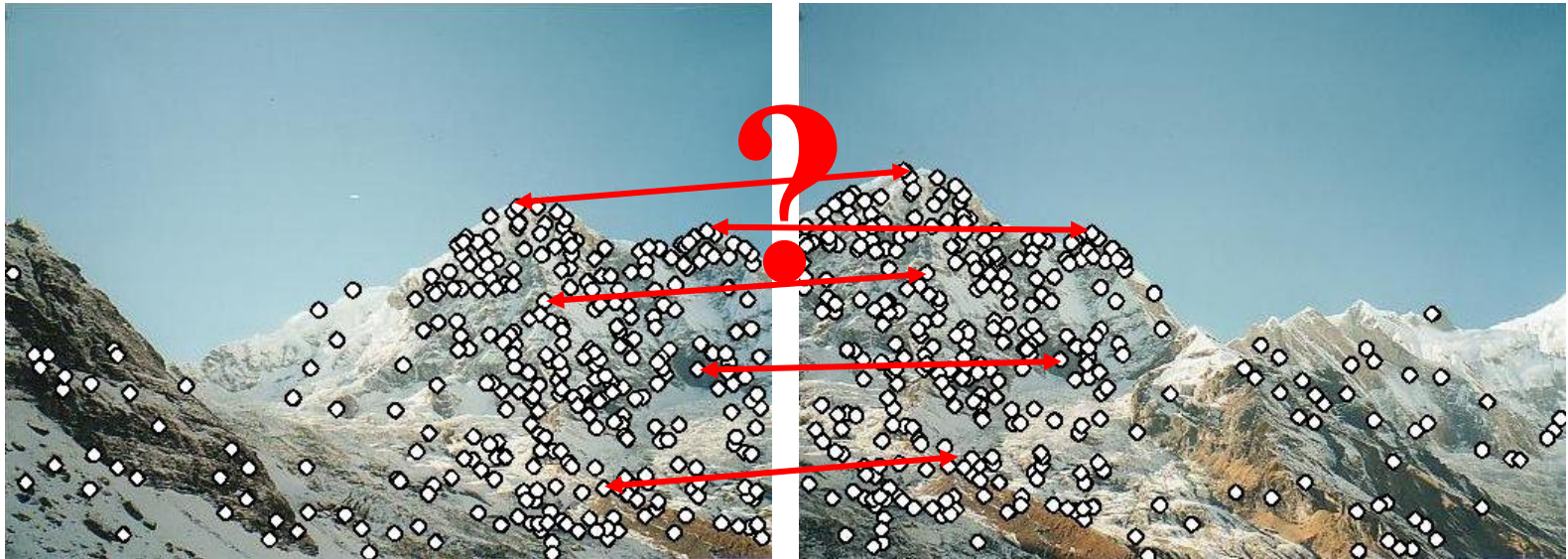
We know how to detect good points

Next question: **How to match them?**



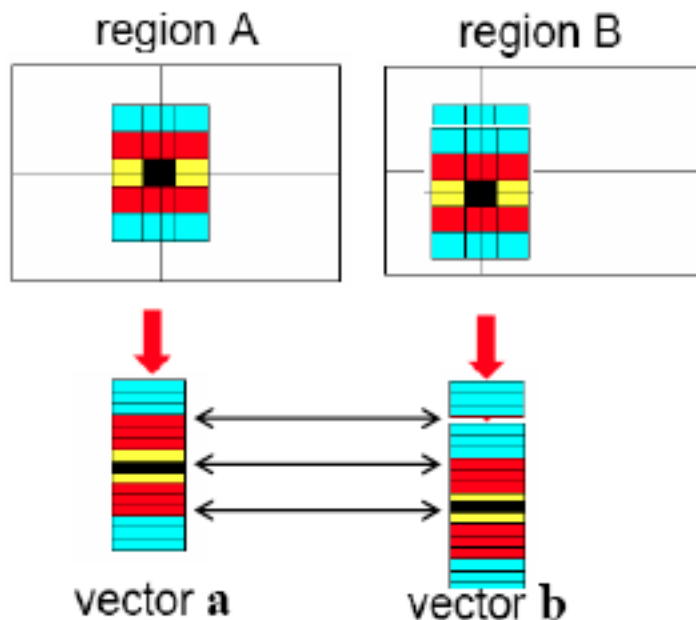
Feature descriptors

We know how to detect good points
Next question: **How to match them?**



- Simple option: match square windows around the point
- Better approach: SIFT
 - David Lowe, UBC <http://www.cs.ubc.ca/~lowe/keypoints/>

Raw image patches as descriptors



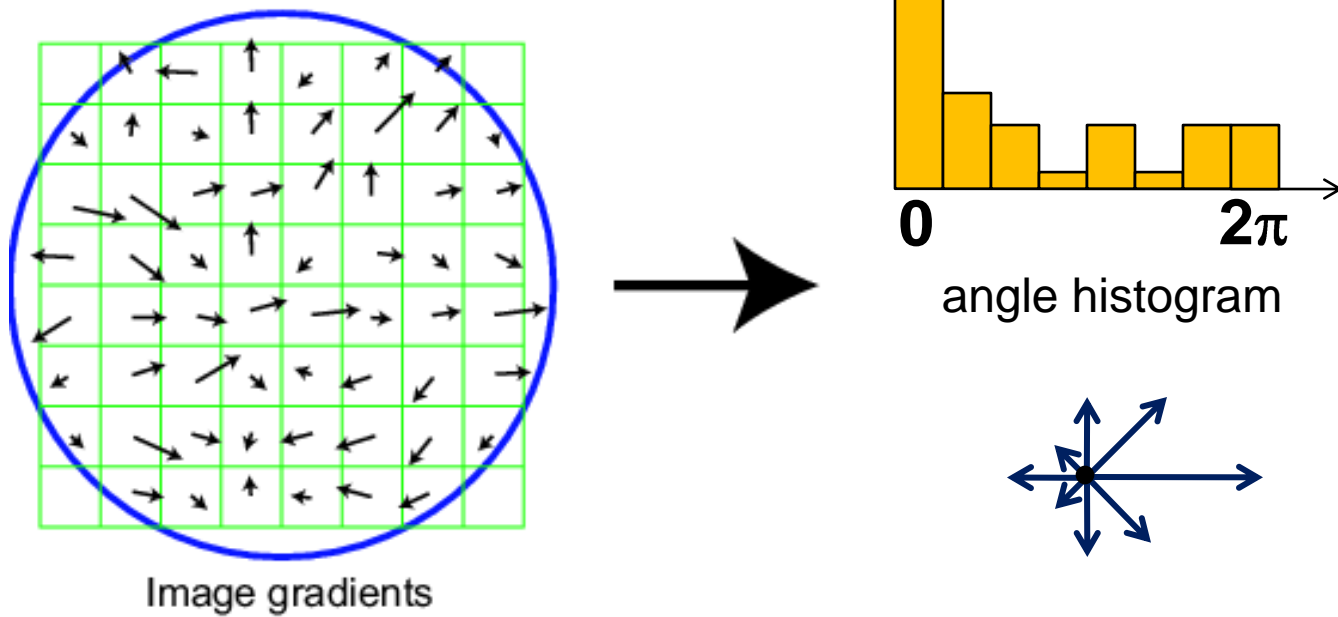
The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

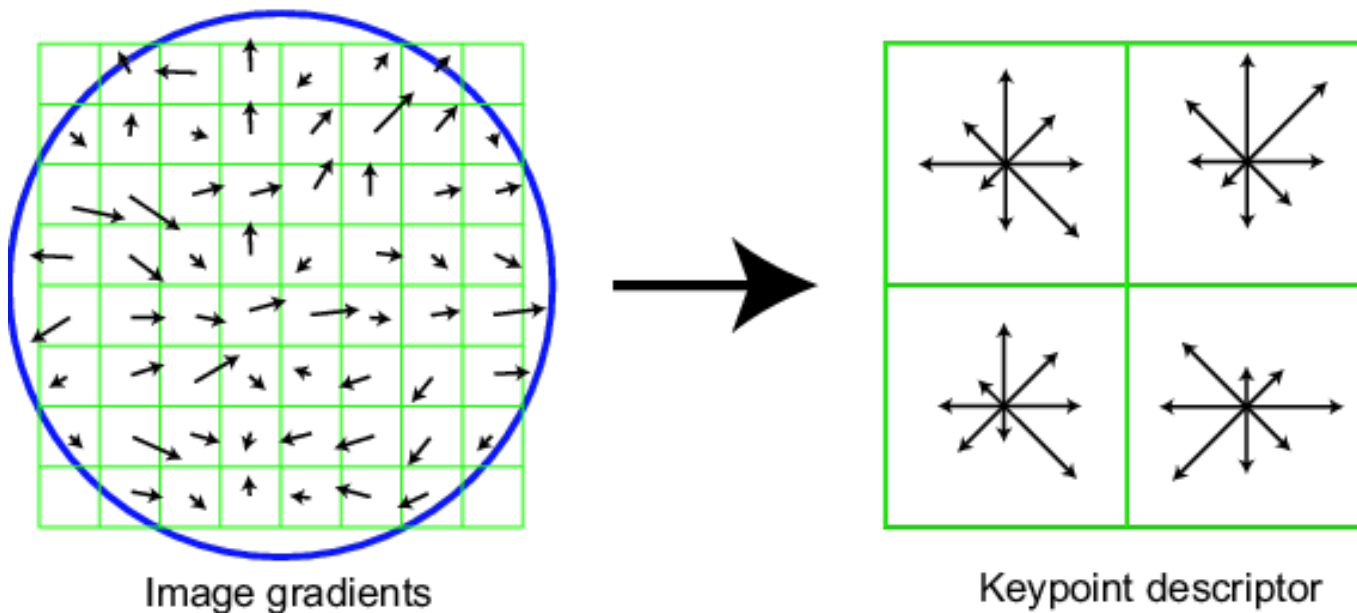


Adapted from slide by David Lowe

SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

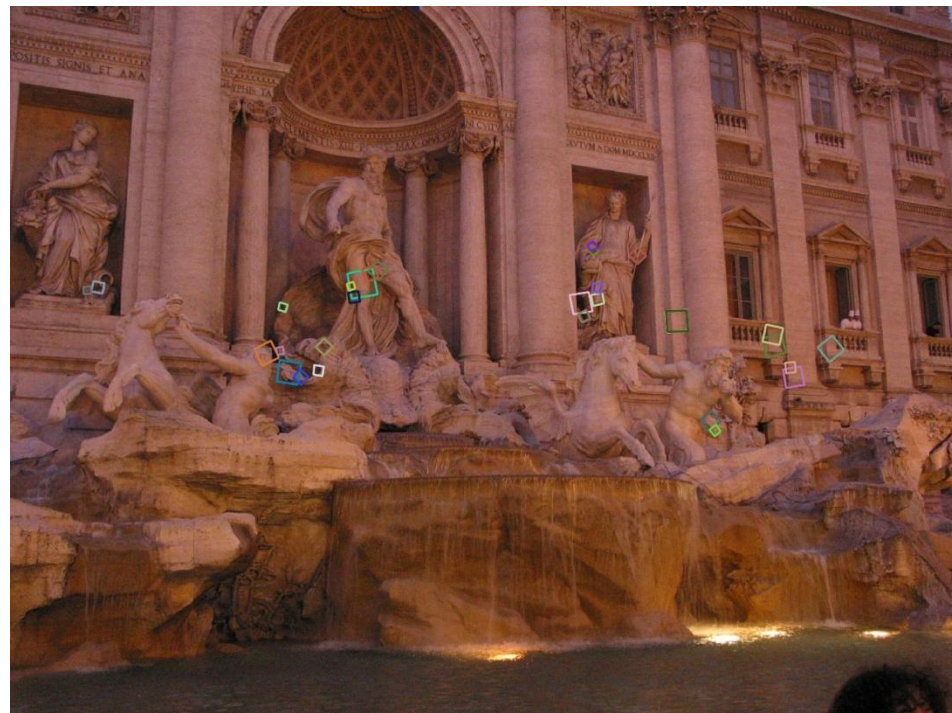


Adapted from slide by David Lowe

Properties of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
 - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
 - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
 - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT



Feature matching

Given a feature in I_1 , how to find the best match in I_2 ?

1. Define distance function that compares two descriptors
2. Test all the features in I_2 , find the one with min distance

Will not be covered in this tutorial

Lots of applications



Features are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- ... other



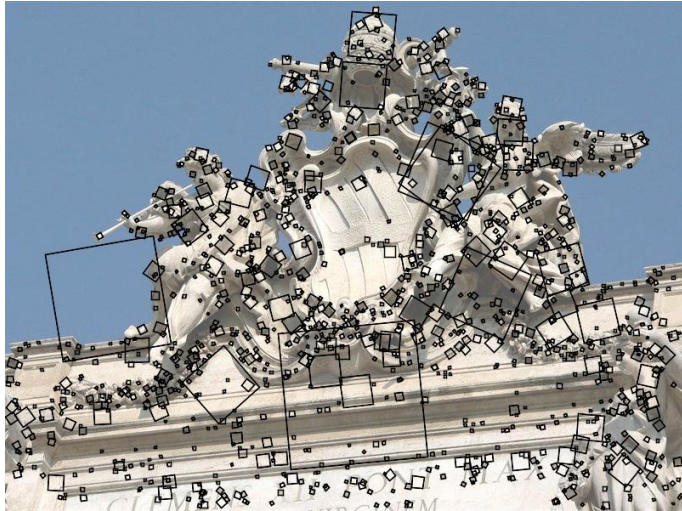
Automatic Mosaicing



Wide baseline stereo



Geometry Estimation



Snaveley, Seitz, & Szeliski 2006

Object Recognition



Schmid and Mohr 1997



Sivic and Zisserman, 2003

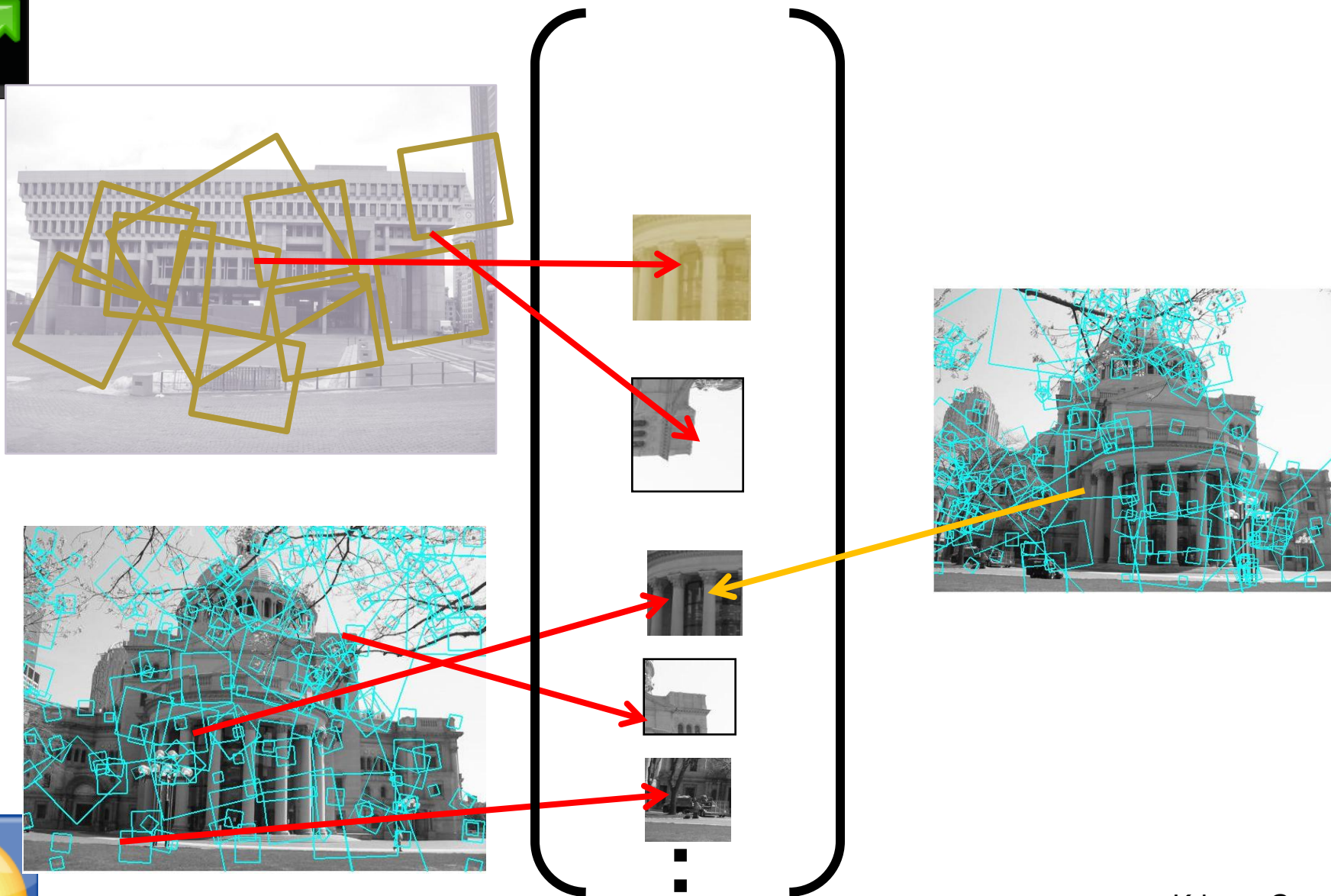


Rothganger et al. 2003



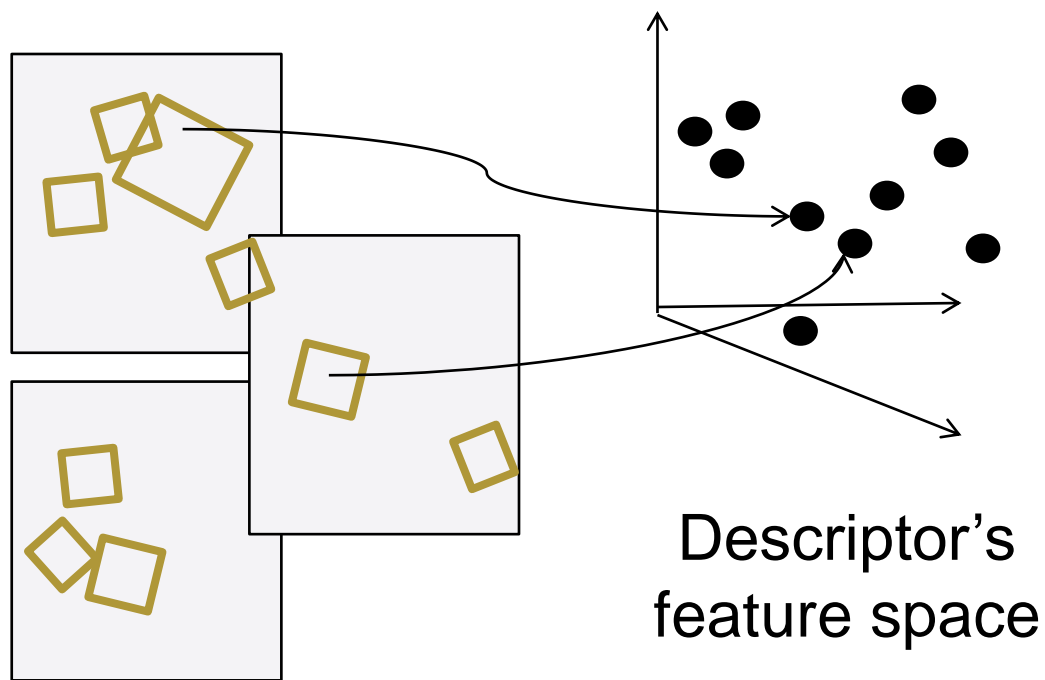
Lowe 2002

Indexing local features



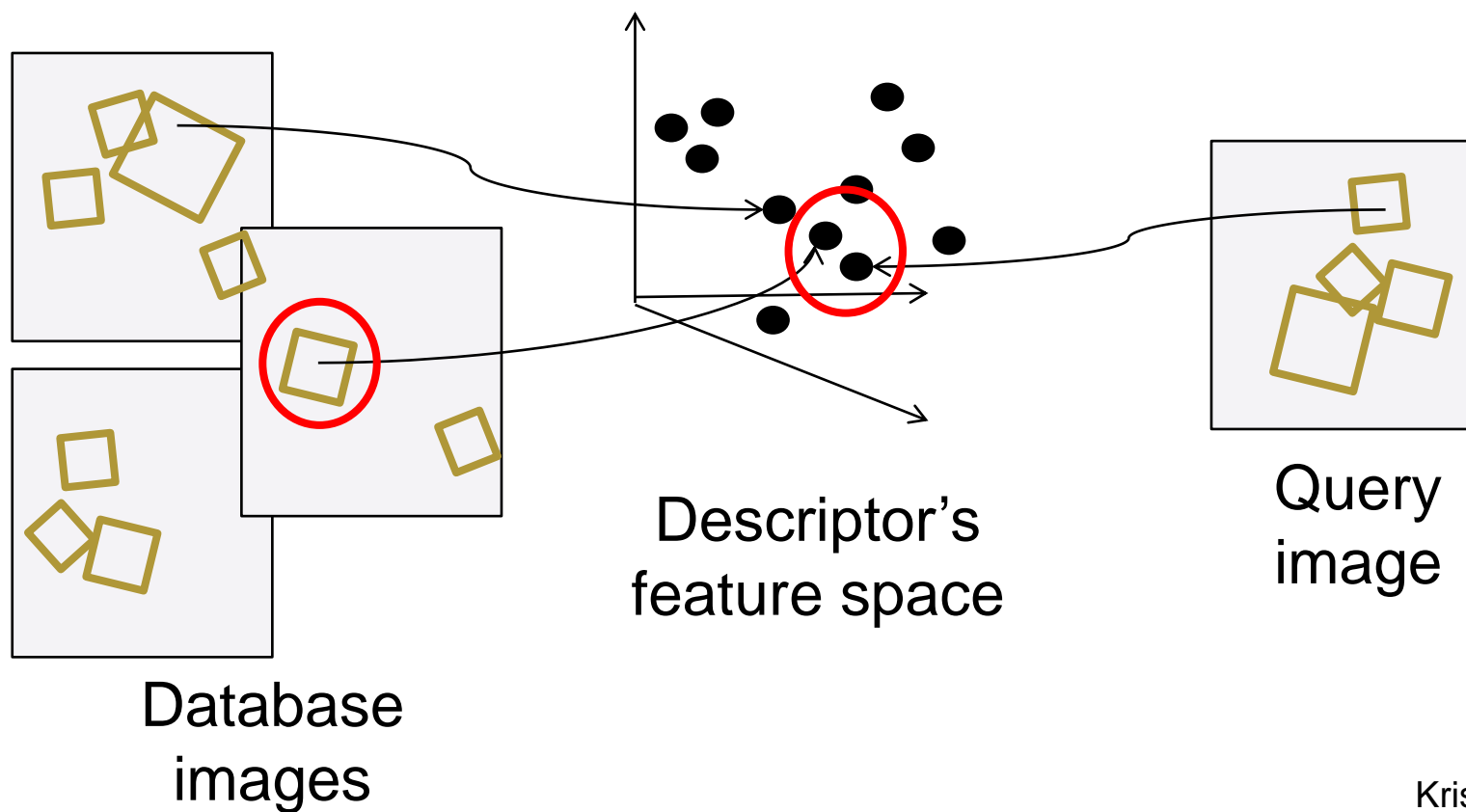
Indexing local features

Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



Indexing local features

When we see close points in feature space, we have similar descriptors, which indicates similar local content.



Indexing local features



With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?



Indexing local features: inverted file index

Index		
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134	Driving Lanes; 85
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)	Duval County; 163
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142	Eau Gallie; 175
511 Traffic Information; 83	Ca d'Zan; 147	Edison, Thomas; 152
A1A (Barrier Is) - I-95 Access; 86	Caloosahatchee River; 152	Eglin AFB; 116-118
AAA (and CAA); 83	Name; 150	Eight Reale; 176
AAA National Office; 88	Canaveral Natl Seashore; 173	Ellenton; 144-145
Abbreviations,	Cannon Creek Airpark; 130	Emanuel Point Wreck; 120
Colored 25 mile Maps; cover	Canopy Road; 106,169	Emergency Callboxes; 83
Exit Services; 196	Cape Canaveral; 174	Epiphytes; 142,148,157,159
Travelogue; 85	Castillo San Marcos; 169	Escambia Bay; 119
Africa; 177	Cave Diving; 131	Bridge (I-10); 119
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150	County; 120
Ah-Tah-Thi-Ki Museum; 160	Celebration; 93	Estero; 153
Air Conditioning, First; 112	Charlotte County; 149	Everglade,90,95,139-140,154-160
Alabama; 124	Charlotte Harbor; 150	Draining of; 156,161
Alachua; 132	Chautauqua; 116	Wildlife MA; 160
County; 131	Chipay; 114	Wonder Gardens; 154
Alafia River; 143	Name; 115	Falling Waters SP; 115
Alapaha, Name; 126	Choctawatchee, Name; 115	Fantasy of Flight; 95
Alfred B MacClay Gardens; 106	Circus Museum, Ringling; 147	Fayer Dykes SP; 171
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180	Fires, Forest; 166
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180	Fires, Prescribed ; 148
Alligator Hole (definition); 157	City Maps,	Fisherman's Village; 151
Alligator, Buddy; 155	Ft Lauderdale Expwys; 194-195	Flagler County; 171
Alligators; 100,135,138,147,156	Jacksonville; 163	Flagler, Henry; 97,165,167,171
Anastasia Island; 170	Kissimmee Expwys; 192-193	Florida Aquarium; 186
Anhaica; 109-109,146	Miami Expressways; 194-195	Florida,
Apalachicola River; 112	Orlando Expressways; 192-193	12,000 years ago; 187
Appleton Mus of Art; 136	Pensacola; 26	Cavern SP; 114
Aquifer; 102	Tallahassee; 191	Map of all Expressways; 2-3
Arabian Nights; 94	Tampa-St. Petersburg; 63	Mus of Natural History; 134
Art Museum, Ringling; 147	St. Augustine; 191	National Cemetery ; 141
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141	Part of Africa; 177
Aucilla River Project; 106	Clearwater Marine Aquarium; 187	Platform; 187
Babcock-Web WMA; 151	Collier County; 154	Sheriff's Boys Camp; 126
Bahia Mar Marina; 184	Collier, Barron; 152	Sports Hall of Fame; 130
Baker County; 99	Colonial Spanish Quarters; 168	Sun 'n Fun Museum; 97
Barefoot Mailmen; 182	Columbia County; 101,128	Supreme Court; 107
Barge Canal; 137	Coquina Building Material; 165	Florida's Turnpike (FTP), 178,189
Bee Line Expy; 80	Corkscrew Swamp, Name; 154	25 mile Strip Maps; 66
Belz Outlet Mall; 89	Cowboys; 95	Administration; 189
Bernard Castro; 136	Crab Trap II; 144	Coin System; 190
Big "I"; 165	Cracker, Florida; 88,95,132	Exit Services; 189
Big Cypress; 155,158	Crosstown Expy; 11,35,98,143	HEFT; 76,161,190
Big Foot Monster; 105	Cuban Bread; 184	History; 189
Billie Swamp Safari; 160	Dade Battlefield; 140	Names; 189
Blackwater River SP; 117	Dade, Maj. Francis; 139-140,161	Service Plazas; 190
Blue Angels	Dania Beach Hurricane; 184	Spur SR91; 76
	Daniel Boone, Florida Walk; 117	Ticket System; 190
	Daytona Beach; 172-173	Toll Plazas; 190
	De Land; 87	Ford, Henry; 152

For text documents,
an efficient way to
find all *pages* on
which a *word* occurs
is to use an index...

We want to find all
images in which a
feature occurs.

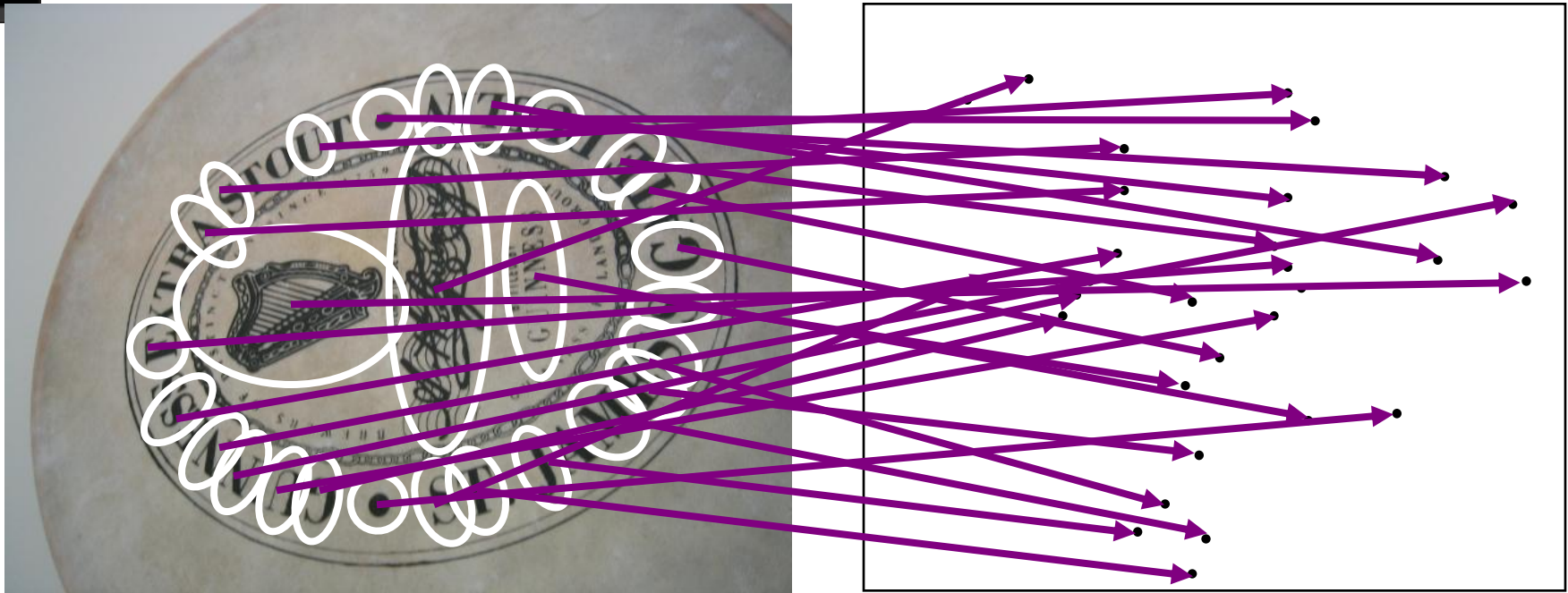
To use this idea, we'll
need to map our
features to "visual
words".

Text retrieval vs. image search

What makes the problems similar, different?

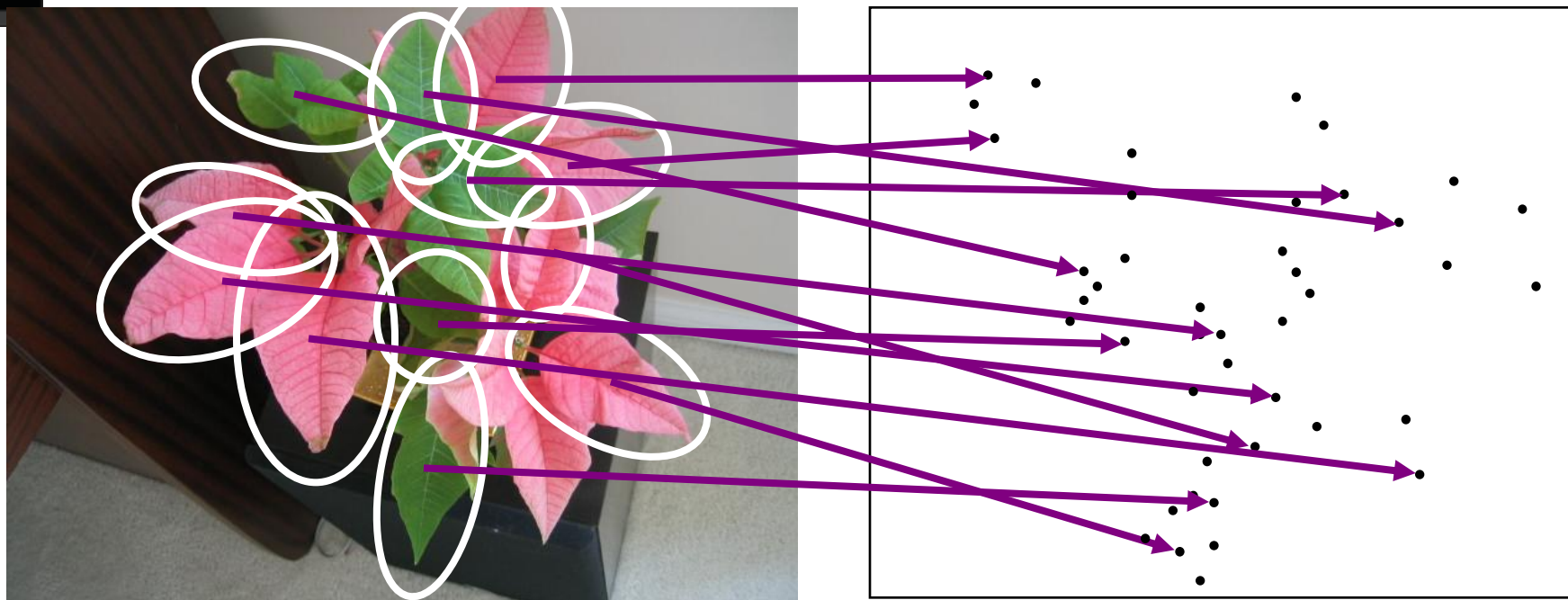
Visual words: main idea

Extract some local features from a number of images ...

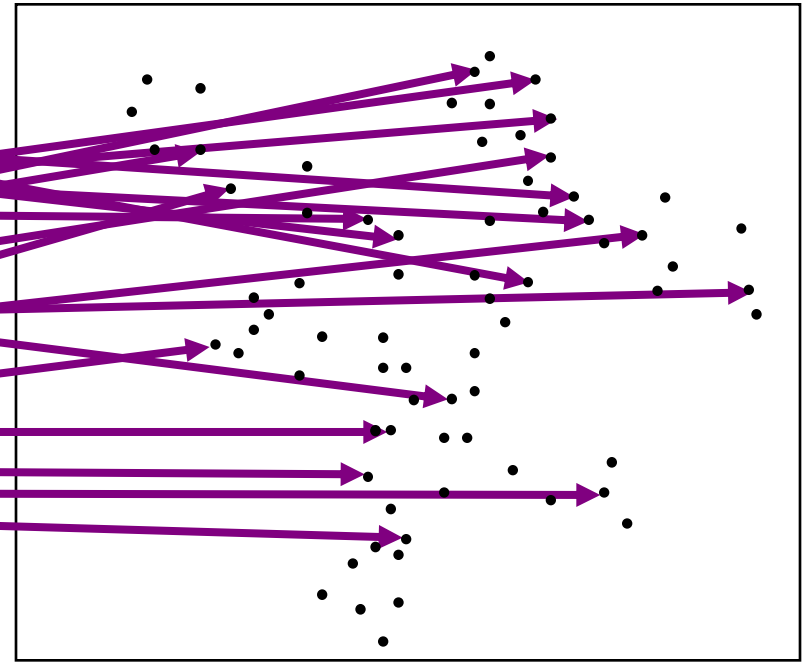
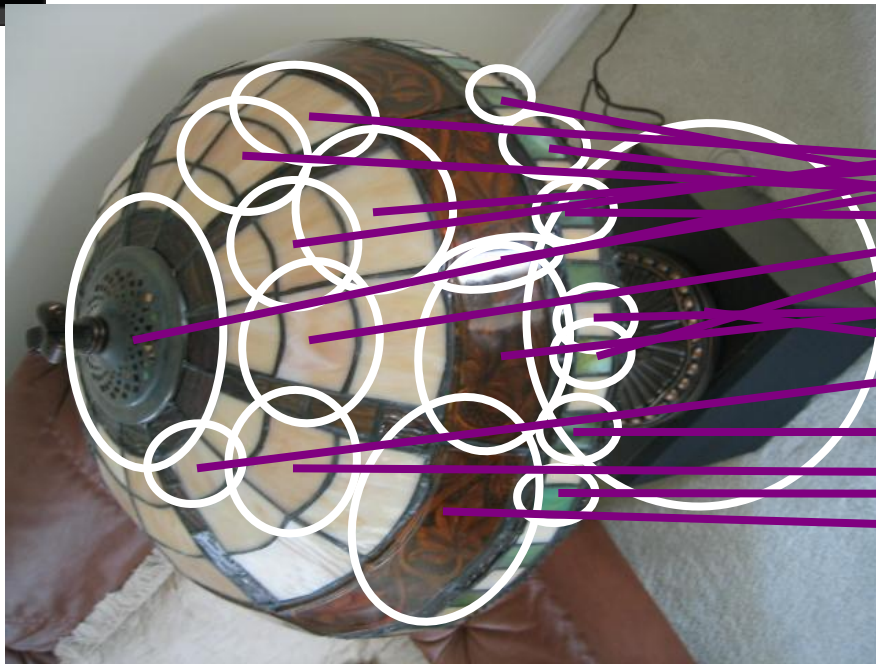


e.g., SIFT descriptor
space: each point is 128-
dimensional

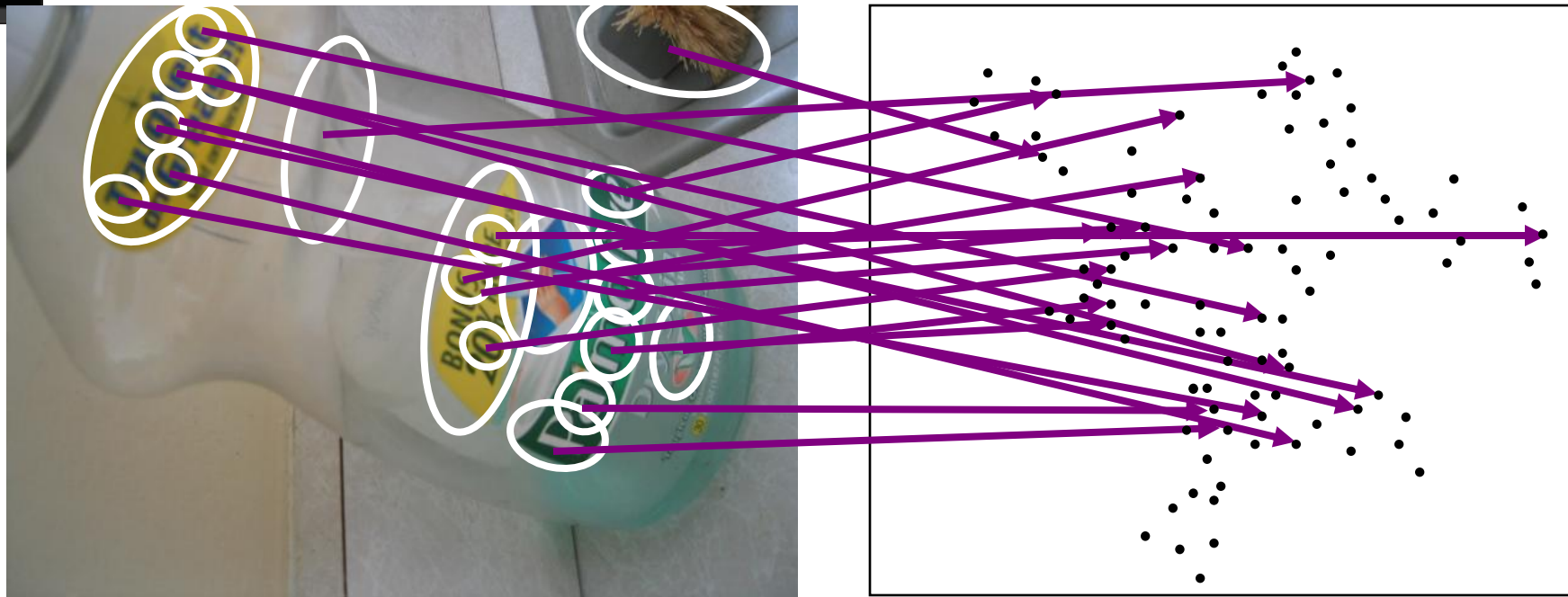
Visual words: main idea

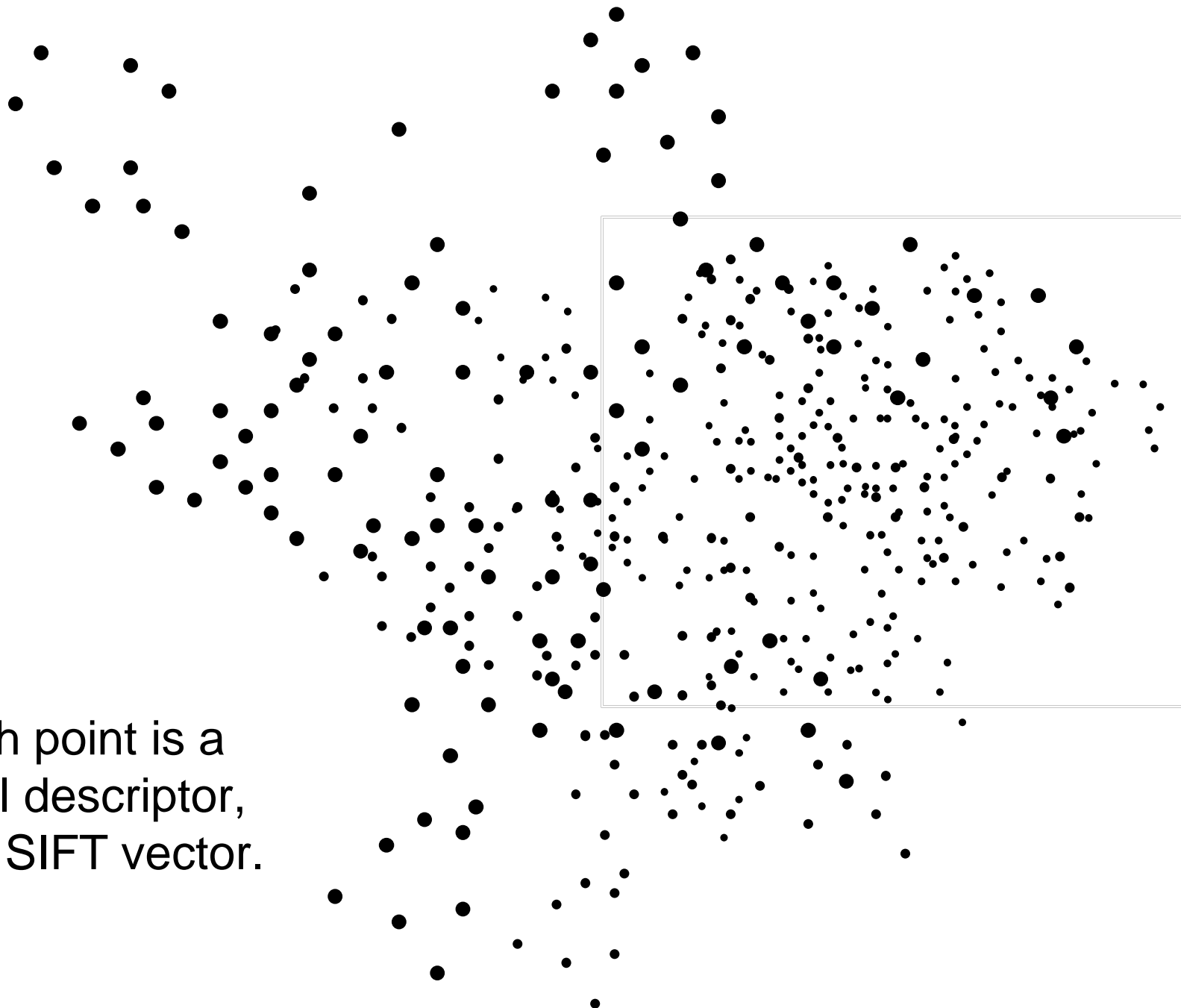


Visual words: main idea

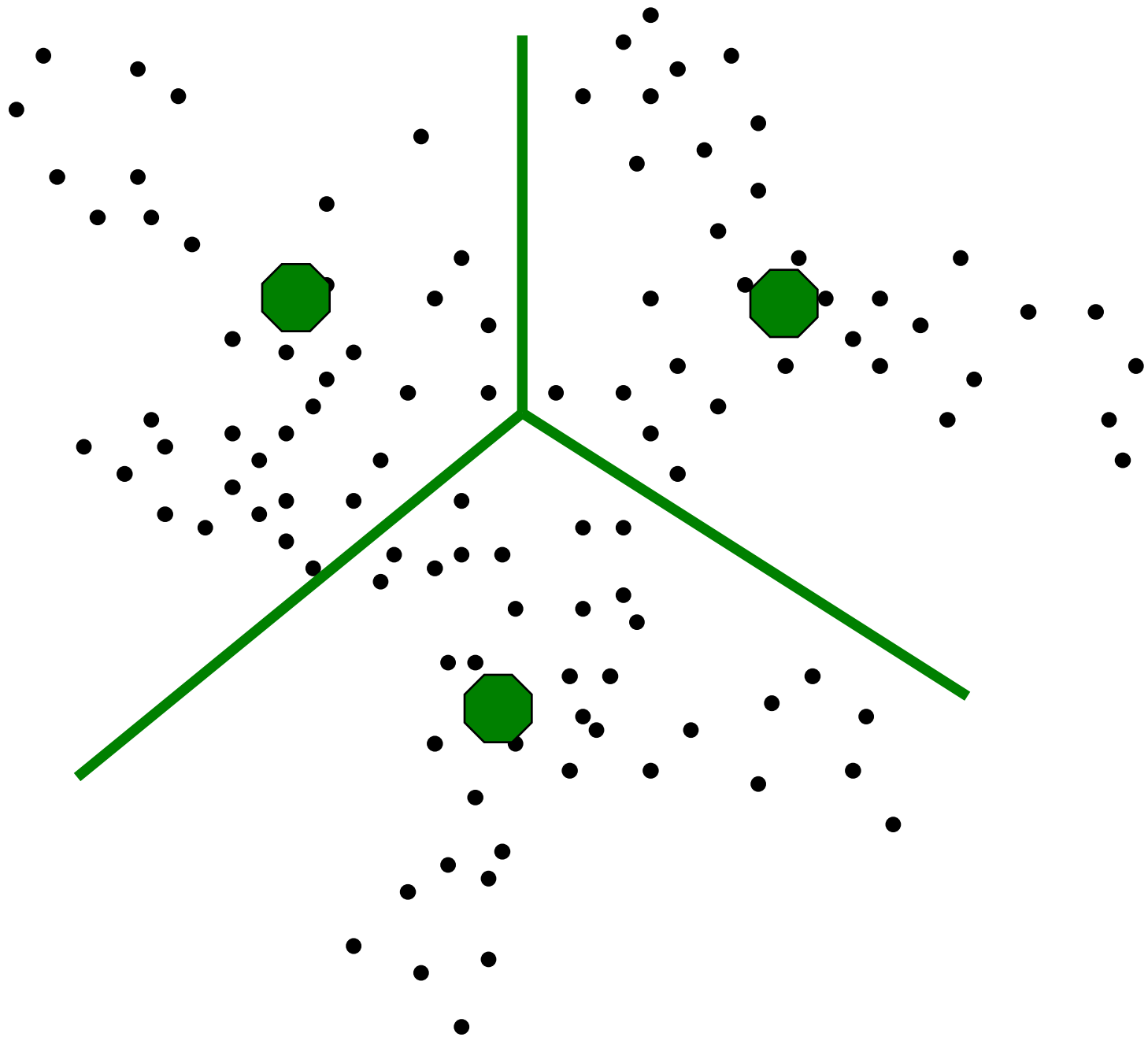


Visual words: main idea



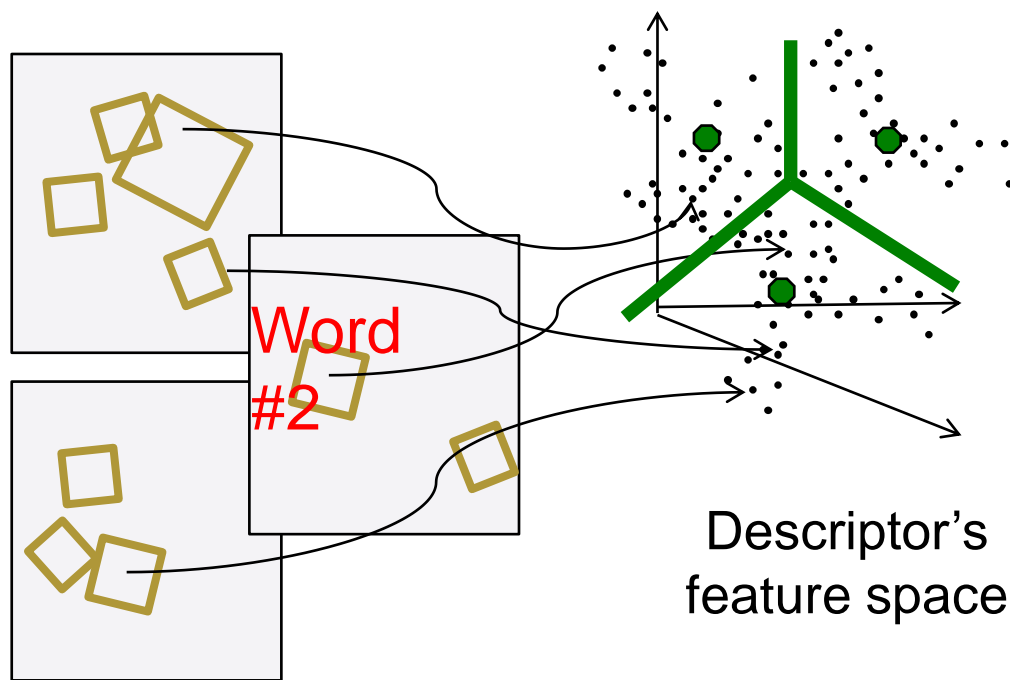


Each point is a
local descriptor,
e.g. SIFT vector.



Visual words

Map high-dimensional descriptors to tokens/words by quantizing the feature space



- Quantize via clustering, let cluster centers be the prototype “words”
- Determine which word to assign to each new image region by finding the closest cluster center.

Visual words

Example: each group of patches belongs to the same visual word

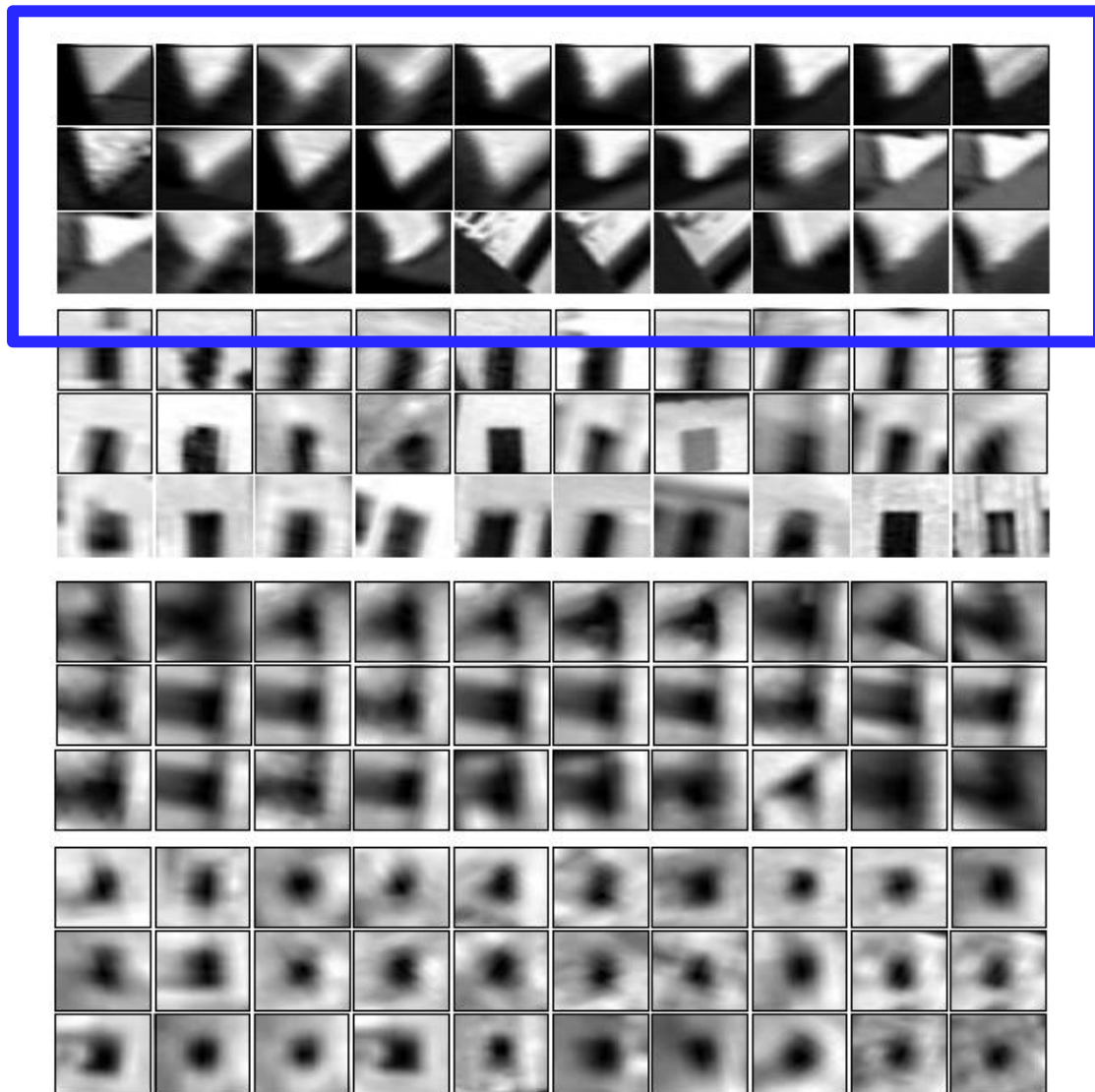
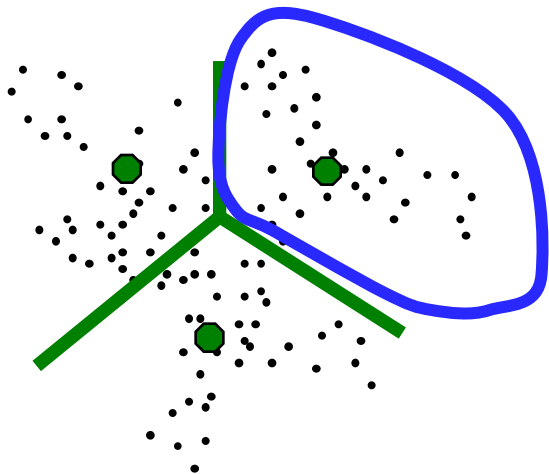


Figure from Sivic & Zisserman, ICCV 2003

Visual vocabulary formation

Issues:

Sampling strategy: where to extract features?

Clustering / quantization algorithm

Unsupervised vs. supervised

What corpus provides features (universal vocabulary?)

Vocabulary size, number of words



If a local image region is a visual word, how can we summarize an image (the document)?



Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach our eyes.

For a long time, the visual image was considered as a movie scene. The visual centers in the brain are a

retinal, cerebral cortex, eye, cell, optical

nerve, image

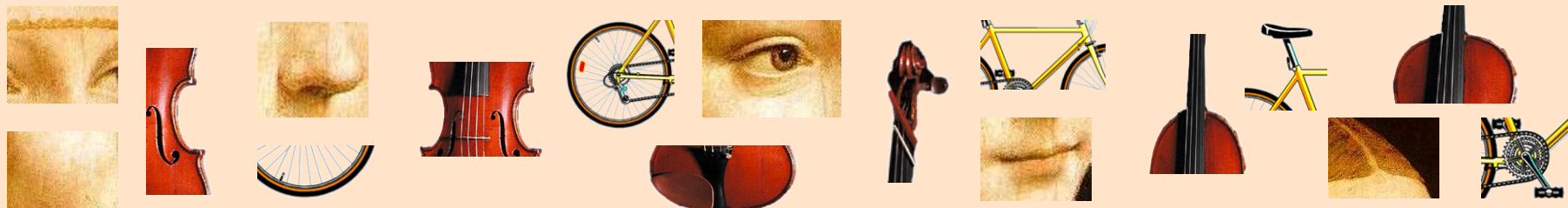
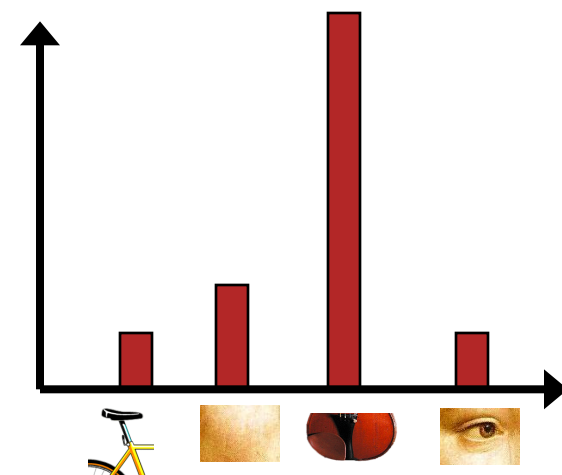
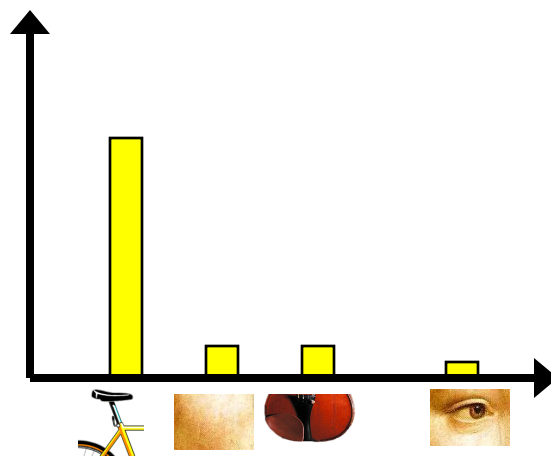
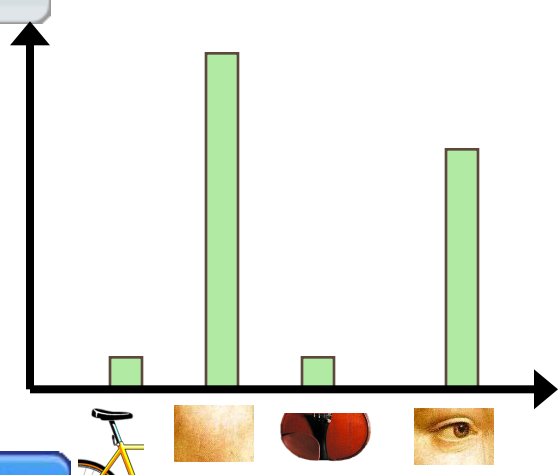
Hubel, Wiesel

following the discovery of the visual cortex, Hubel and Wiesel have demonstrated that the *message about the image falling on the retina undergoes a point-by-point analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.*

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$580bn in 2004, and a fall in imports to \$660bn. The increase in exports will annoy the US, but China's government has deliberately agreed to let the yuan rise against the dollar.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

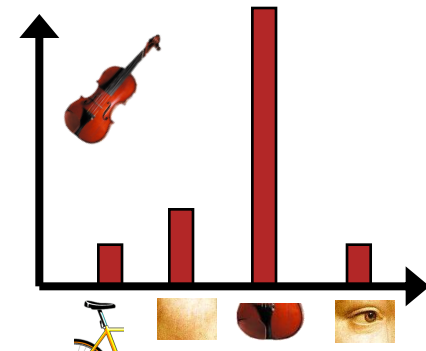
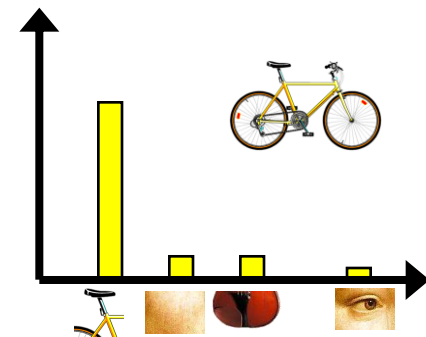
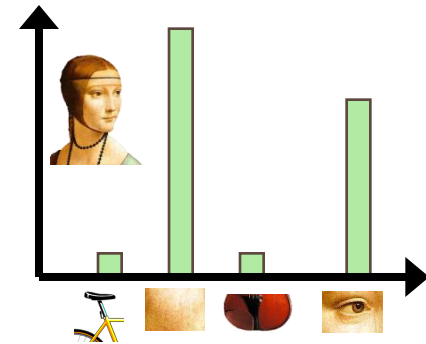
also needs to be taken into account. The demand for yuan is increasing in the country. China has been allowed to trade the yuan against the dollar within a narrow band and permitted it to trade within a narrow band but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.



Bags of visual words

Summarize entire image based on its distribution (histogram) of word occurrences.

Analogous to bag of words representation commonly used for documents.



Bags of words for content-based image retrieval



Visually defined query

“Find this clock”



“Find this place”



“Groundhog Day” [Rammis, 1993]



Example



retrieved shots



Start frame 52907



Key frame 53026



End frame 53028



Start frame 54342



Key frame 54376



End frame 54644



Start frame 51770



Key frame 52251



End frame 52348



Start frame 54079



Key frame 54201



End frame 54201



Start frame 38909



Key frame 39126



End frame 39300



Start frame 40760



Key frame 40826



End frame 41049



Start frame 39301



Key frame 39676



End frame 39730

Bags of words: pros and cons



- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + provides vector representation for sets
- + very good results in practice



- basic model ignores geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image



- optimal vocabulary formation remains unclear