

CS 1674/2074: Local features: detection, description and matching

PhD. Nils Murrugarra-Llerena
nem177@pitt.edu

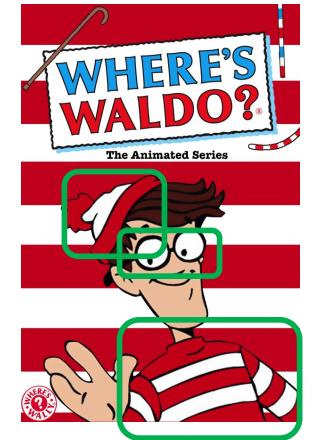


[Motivation] Local Features

The "Where's Waldo?" [\[Game\]](#)

The goal is to find Waldo in a crowded, complex scene.

How do you search for Waldo?

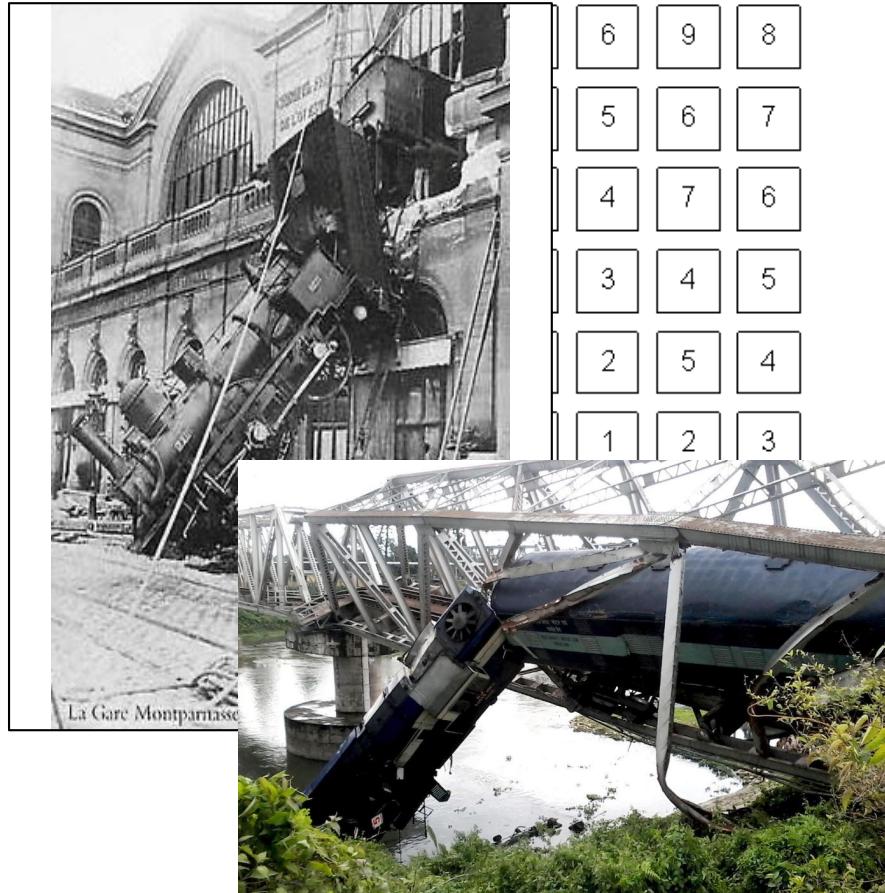


What are
Waldo's
distinctive
features?

Plan for this lecture

- Feature detection / keypoint extraction
 - Corner detection
- Feature description (of detected features)
- Matching features across images

An image is a set of pixels



Adapted from S. Narasimhan

Problems with pixel representation

- Not invariant to small changes
 - Translation
 - Illumination
 - etc.
- Some parts of an image are more important than others
- What do we want to represent?

Human eye movements



Yarbus eye tracking

Local features

- *Local* means that they only cover a small part of the image
- There will be many local features detected in an image; later we'll use those to compute a representation of the whole image
- Local features usually exploit image gradients, ignore color
- Feature $\sim=$ vector of gradient statistics for a window with *particular location and size*

Local features: desired properties

- Locality
 - A feature occupies a relatively small area of the image; robust to clutter and occlusion
- Repeatability and flexibility
 - Robustness to expected variations: the same feature can be found in several images despite geometric/photometric transformations
 - Maximize correct matches (panda to panda)
- Distinctiveness
 - Each feature has a distinctive description
 - Minimize wrong matches (panda to giraffe)
- Compactness and efficiency
 - Many fewer features than image pixels



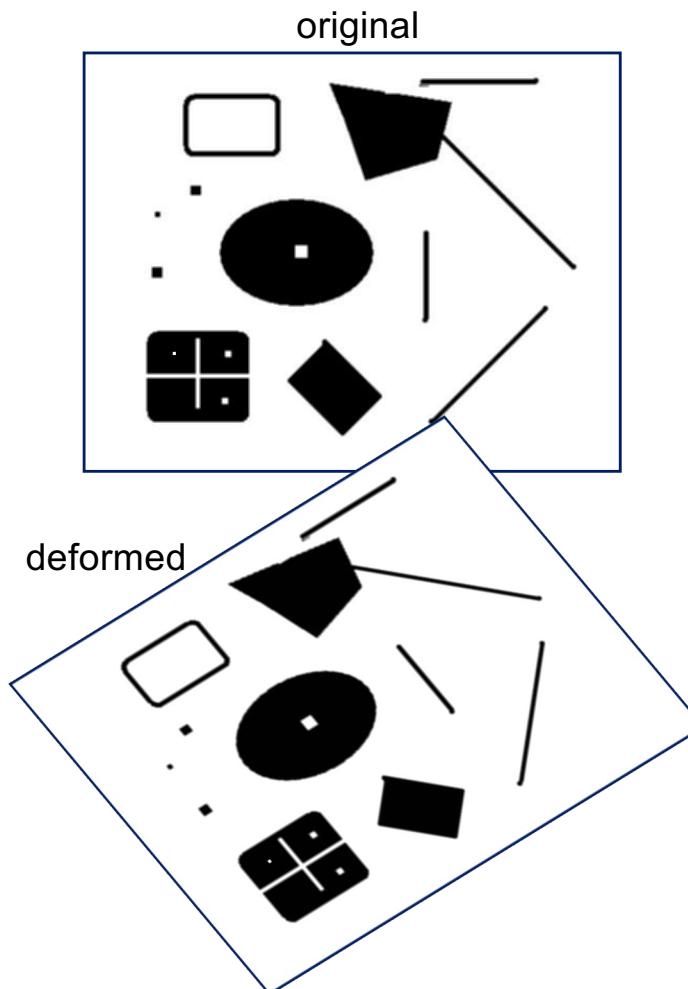
Adapted from K. Grauman and D. Hoiem

Interest(ing) points

- Note: “interest points” = “keypoints”, also sometimes called “features”
- Many applications
 - Recognition: which patches are likely to tell us something about the object category?
 - Image search: which points would allow us to match images between query and database?
 - 3D reconstruction: how to find correspondences across different views?
 - Tracking: which points are good to track?

Interest points

- Suppose you have to click on some point, go away and come back after I deform the image, and click on the same points again.
 - Which points would you choose?



Choosing interest points

Where would you tell
your friend to meet you?

→ Corner detection



Choosing interest points

Where would you tell
your friend to meet you?

→ Blob detection



D. Hoiem

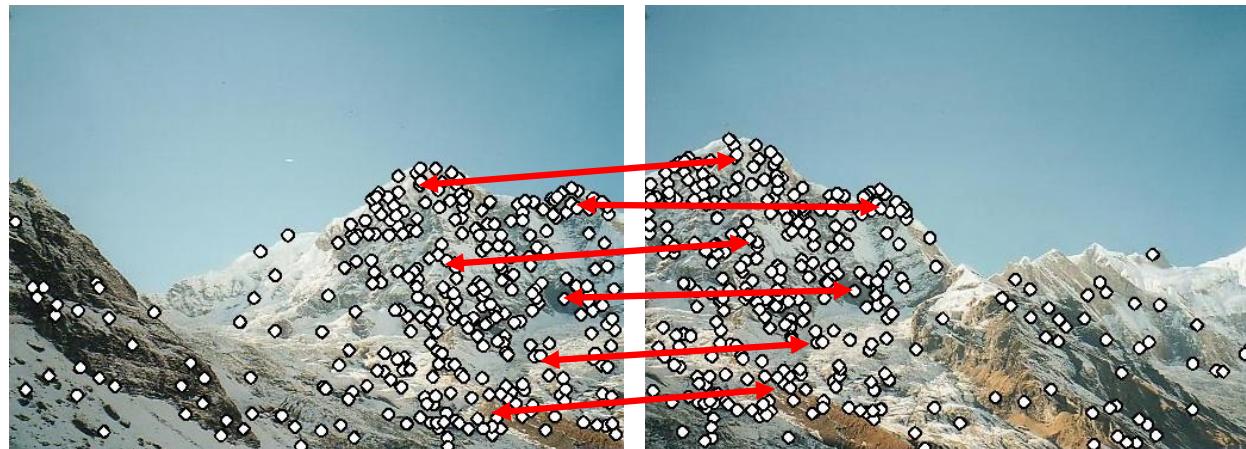
Application: Panorama stitching

- We have two images – how do we combine them?



Application: Panorama stitching

- We have two images – how do we combine them?



Step 1: extract features

Step 2: match features

Application: Panorama stitching

- We have two images – how do we combine them?



Step 1: extract features

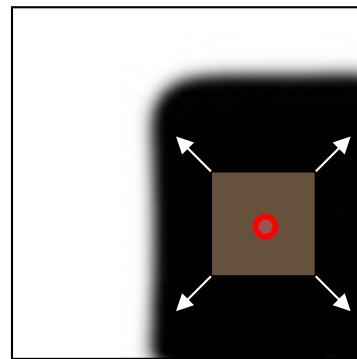
Step 2: match features

Step 3: align images

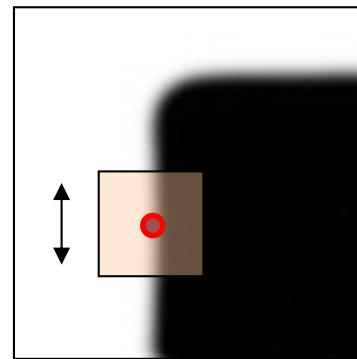
Corners are distinctive interest points

- We should easily recognize the keypoint by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity

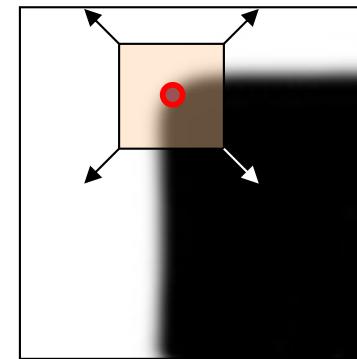
● Candidate keypoint



“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction

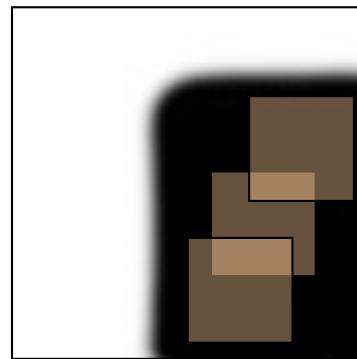


“corner”:
significant change
in all directions

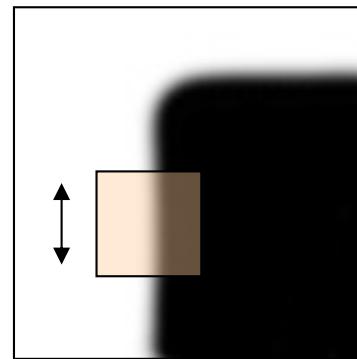
Adapted from A. Efros, D. Frolova, D. Simakov

Corners are distinctive interest points

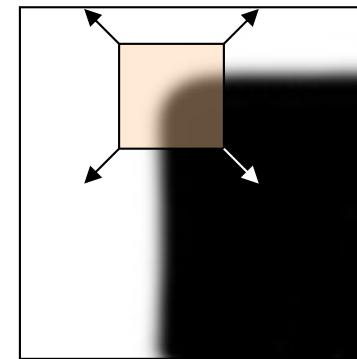
- We should easily recognize the keypoint by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity



“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction

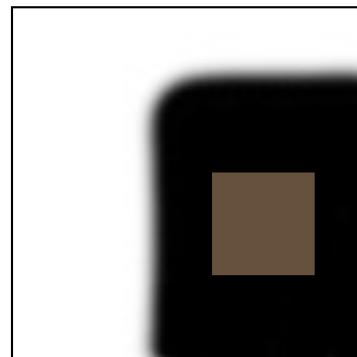


“corner”:
significant change
in all directions

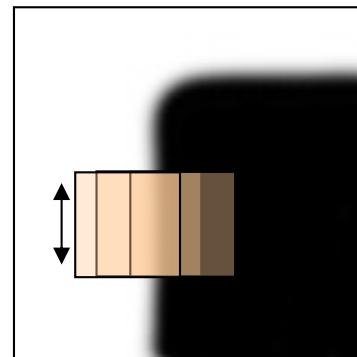
Adapted from A. Efros, D. Frolova, D. Simakov

Corners are distinctive interest points

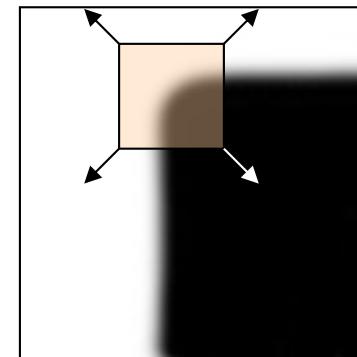
- We should easily recognize the keypoint by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity



“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction

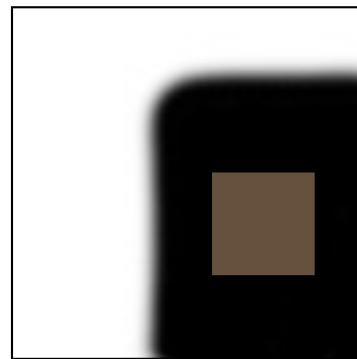


“corner”:
significant change
in all directions

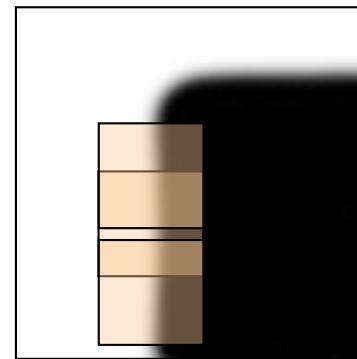
Adapted from A. Efros, D. Frolova, D. Simakov

Corners are distinctive interest points

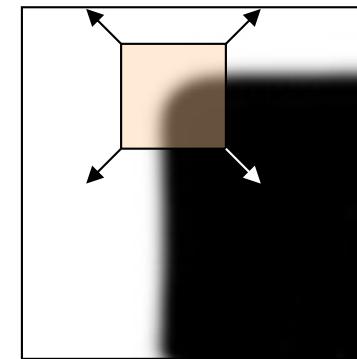
- We should easily recognize the keypoint by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity



“flat” region:
no change in
all directions



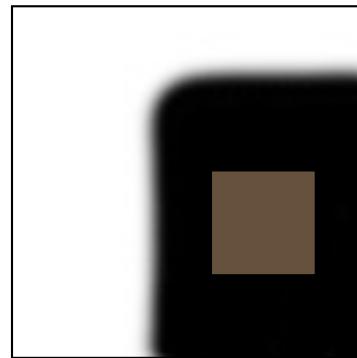
“edge”:
no change along
the edge direction



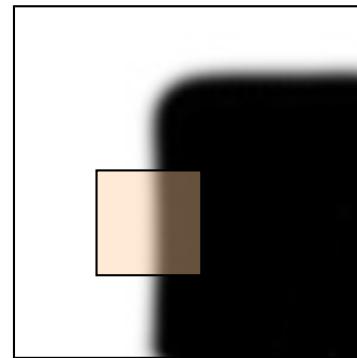
“corner”:
significant change
in all directions

Corners are distinctive interest points

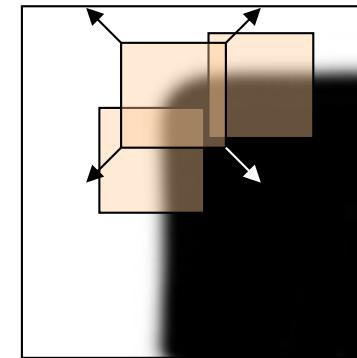
- We should easily recognize the keypoint by looking through a small window
- Shifting a window in *any direction* should give a *large change* in intensity



“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction

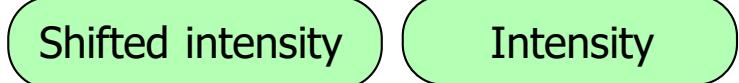


“corner”:
significant change
in all directions

Adapted from A. Efros, D. Frolova, D. Simakov

Harris Detector: Mathematics

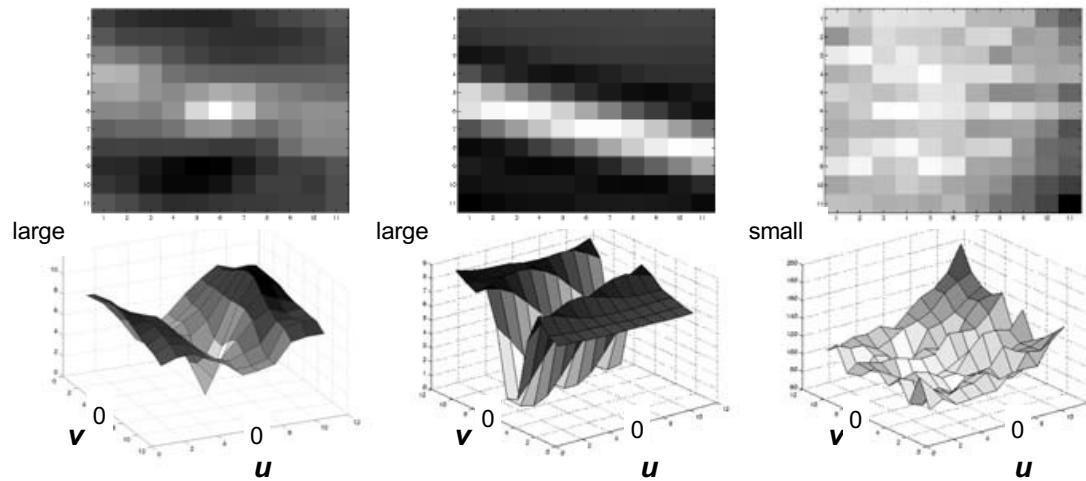
Window-averaged squared change of intensity induced by shifting the patch for a fixed candidate keypoint by [u,v]:

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


Harris Detector: Mathematics

Window-averaged squared change of intensity induced by shifting the patch for a fixed candidate keypoint by [u,v]:

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



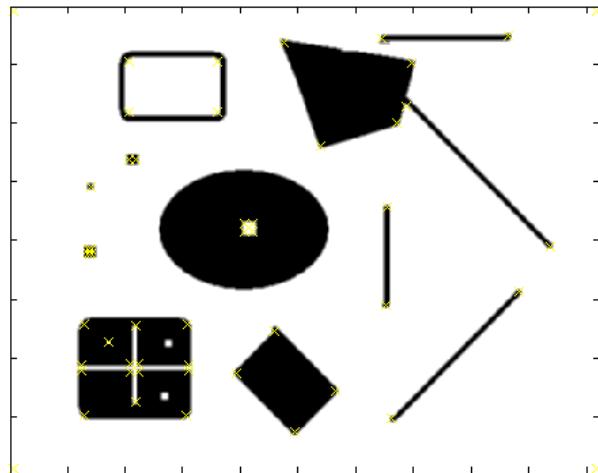
Adapted from D. Frolova, D. Simakov

Example of Harris Application

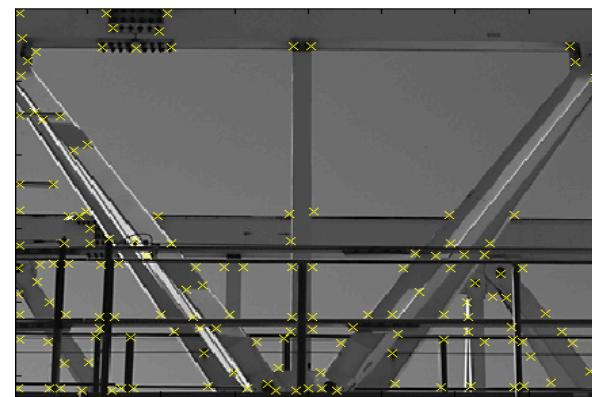


K. Grauman

More Harris Responses



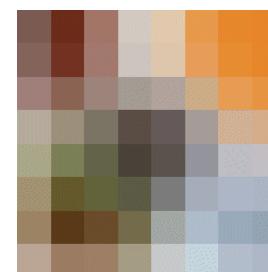
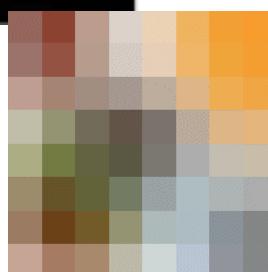
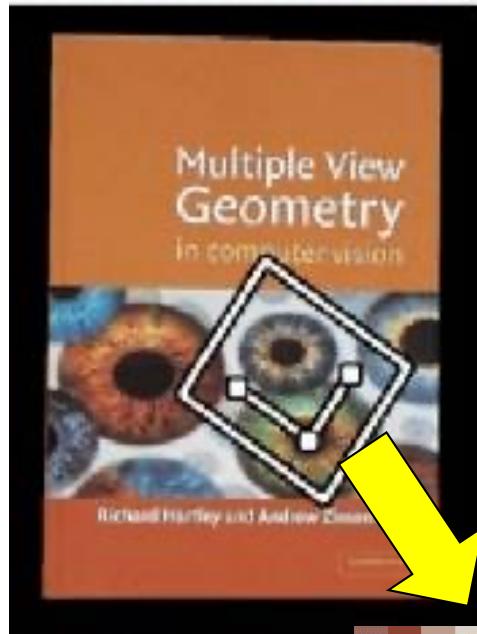
Effect: A very precise corner detector.



Plan for this lecture

- Feature detection / keypoint extraction
 - Corner detection
- Feature description (of detected features)
- Matching features across images

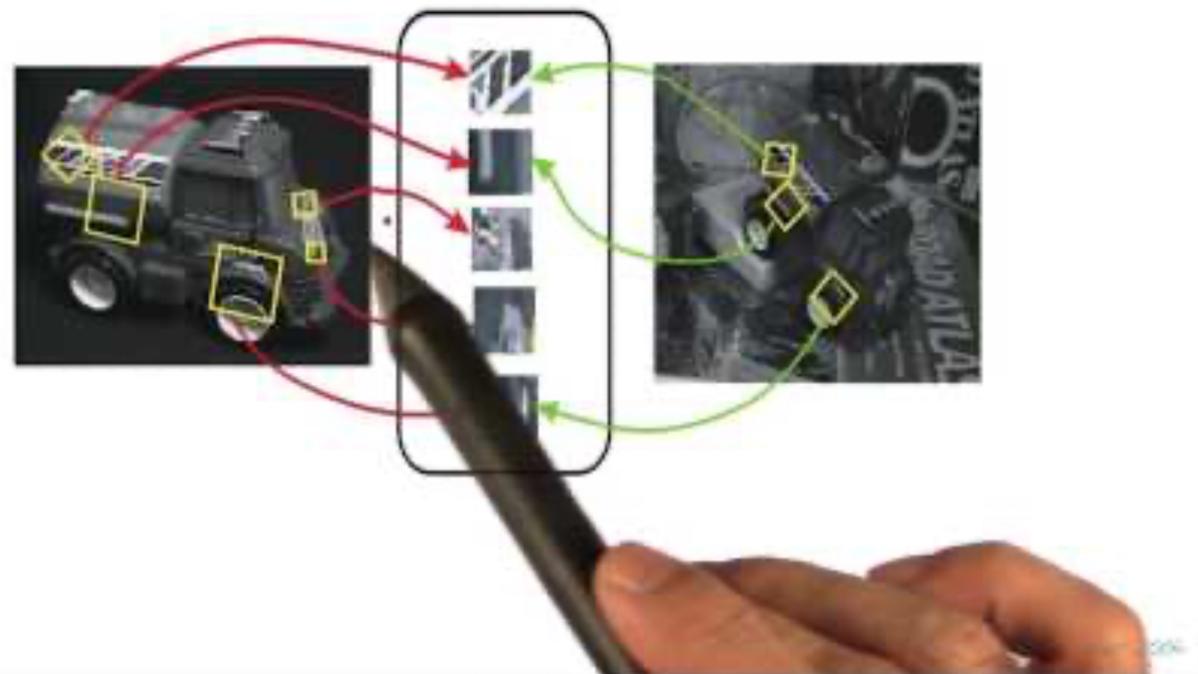
Geometric transformations



e.g. scale,
translation
, rotation

Short Video SIFT Descriptor

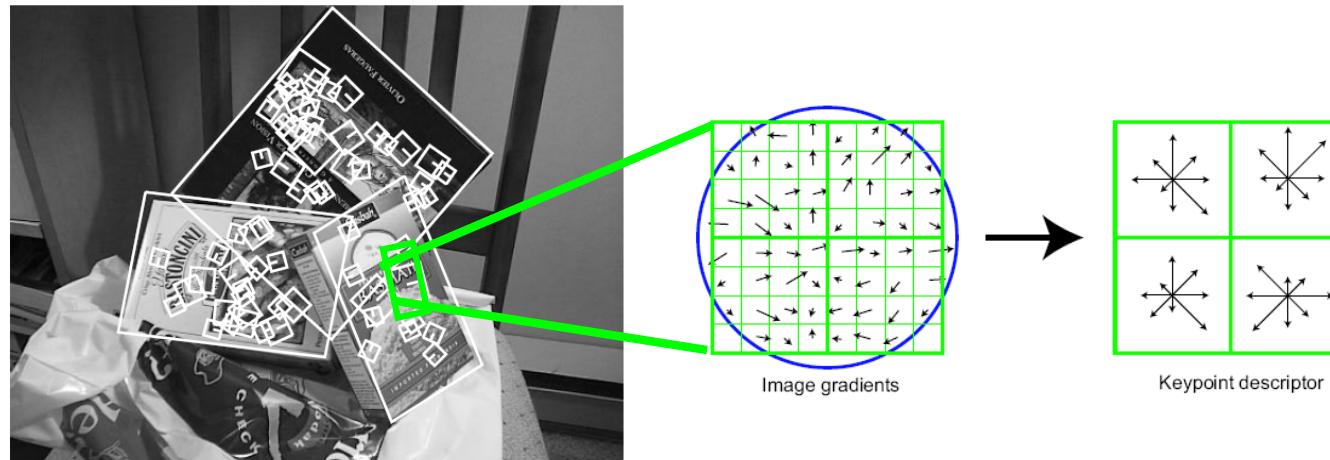
Invariant Local Features



https://www.youtube.com/watch?v=oKAnOzlu66c&ab_channel=Udacity

Scale-Invariant Feature Transform (SIFT) descriptor

Journal + conference versions: 87,527 citations (AlexNet paper has 93,821)



Histogram of oriented gradients

- Captures important texture information
- Robust to small translations / affine deformations

[Lowe, ICCV 1999]

K. Grauman, B. Leibe

Computing gradients

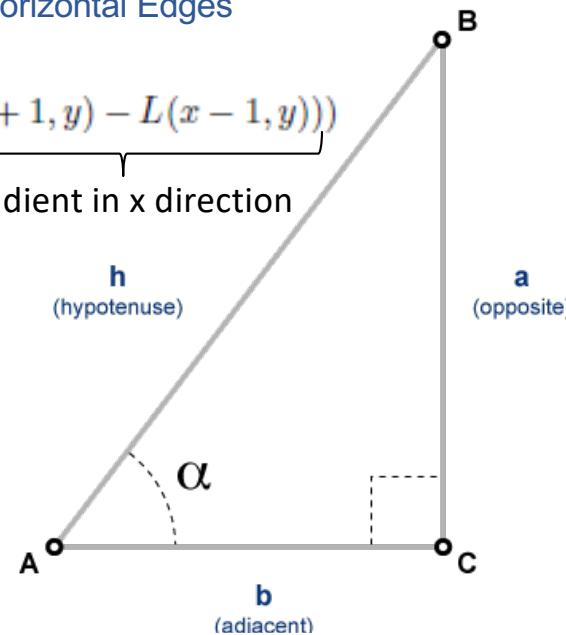
L = the image intensity

$$m(x, y) = \sqrt{\underbrace{(L(x+1, y) - L(x-1, y))^2}_{\text{gradient in x direction}} + \underbrace{(L(x, y+1) - L(x, y-1))^2}_{\text{gradient in y direction}}}$$

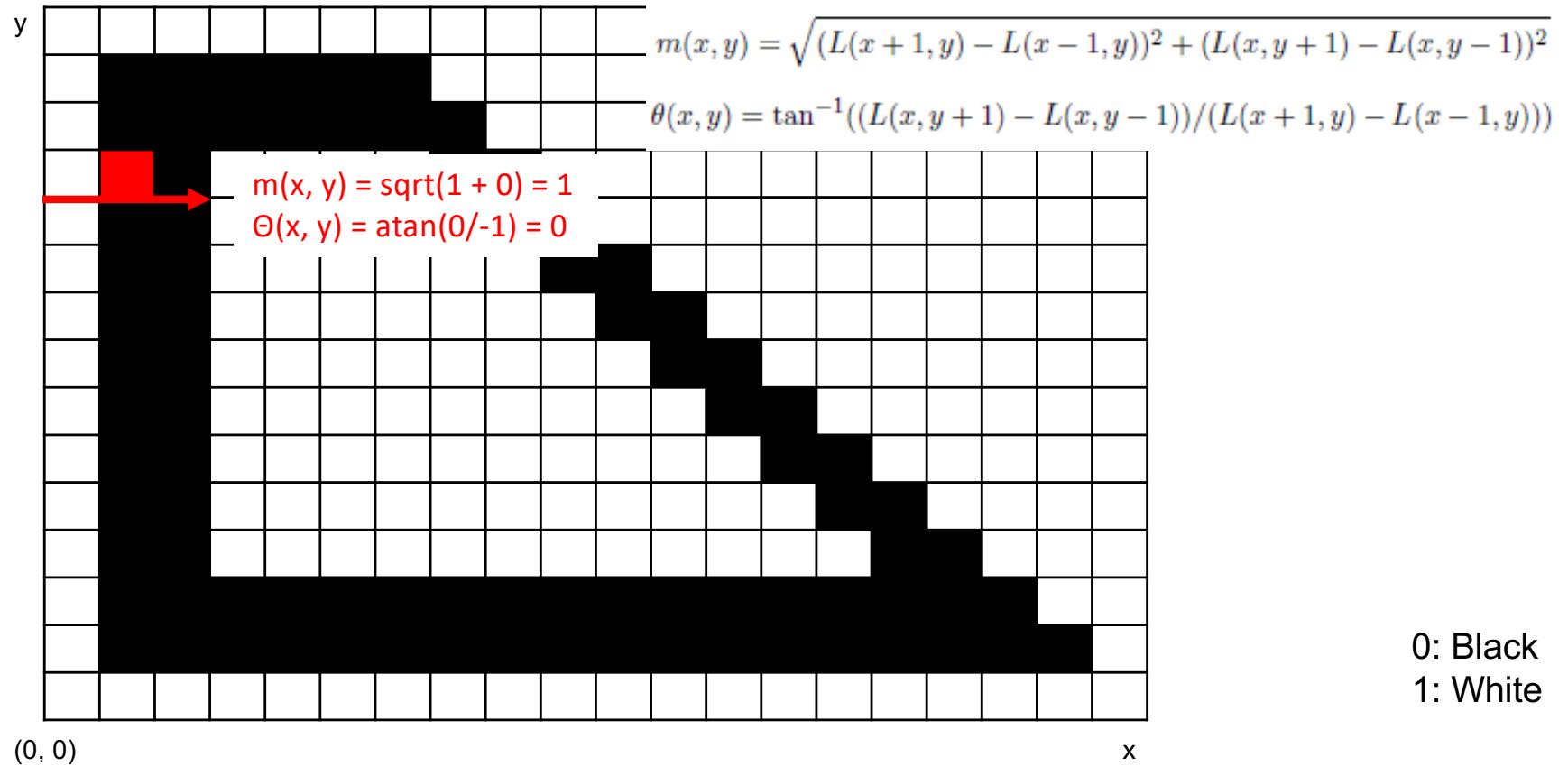
Vertical Edges Horizontal Edges

$$\theta(x, y) = \tan^{-1} \left(\frac{\underbrace{(L(x, y+1) - L(x, y-1))}_{\text{gradient in y direction}}}{\underbrace{(L(x+1, y) - L(x-1, y))}_{\text{gradient in x direction}}} \right)$$

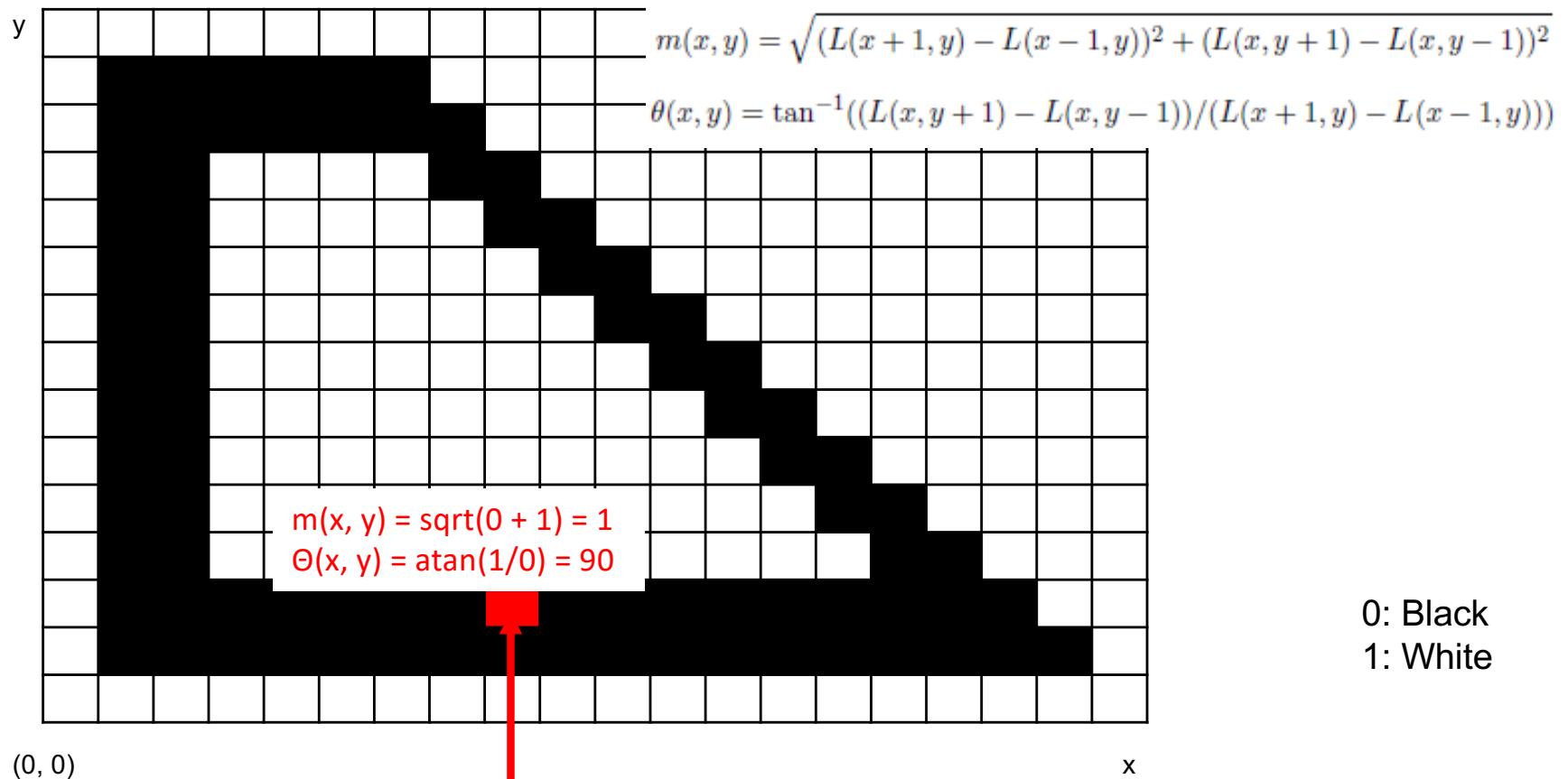
- $\tan(\alpha) = \frac{\text{opposite side}}{\text{adjacent side}}$



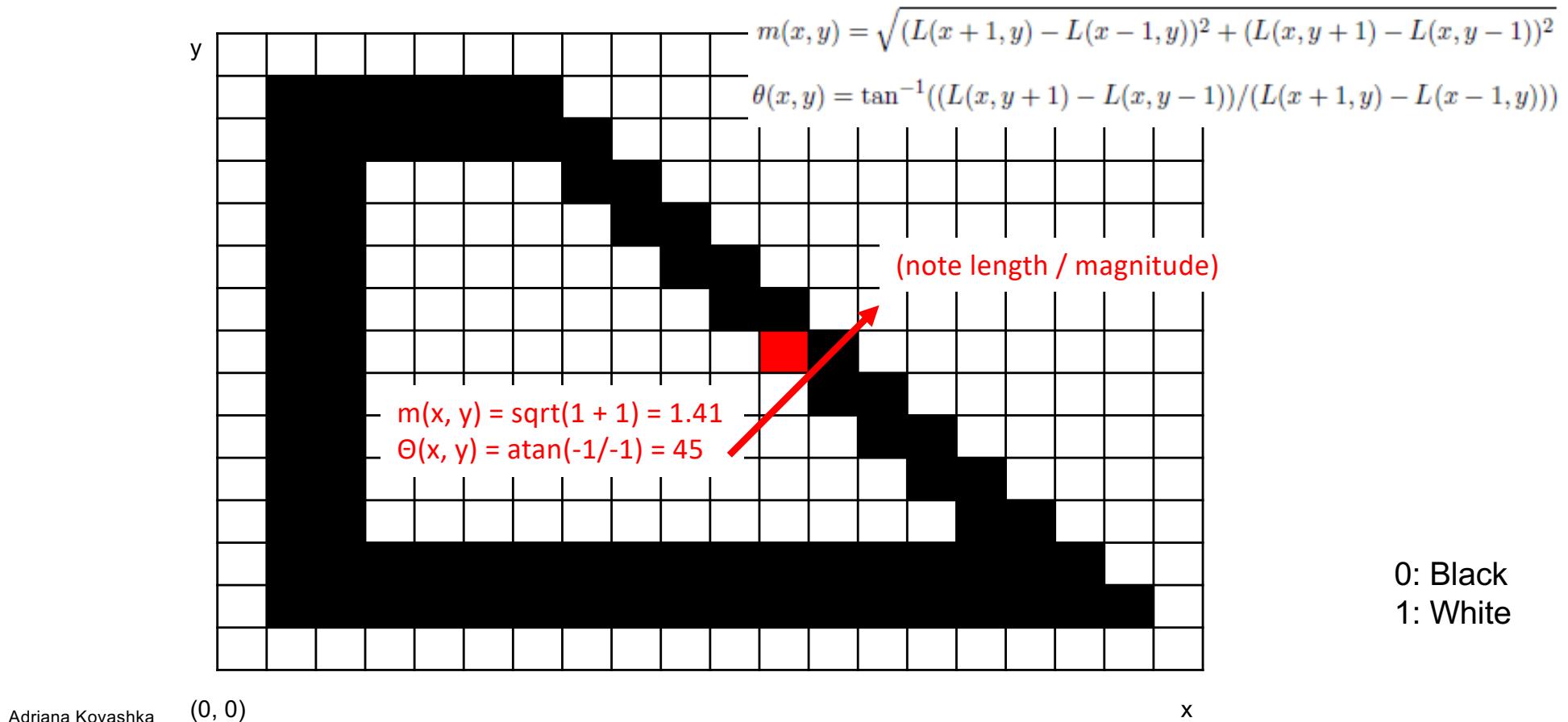
Gradients



Gradients



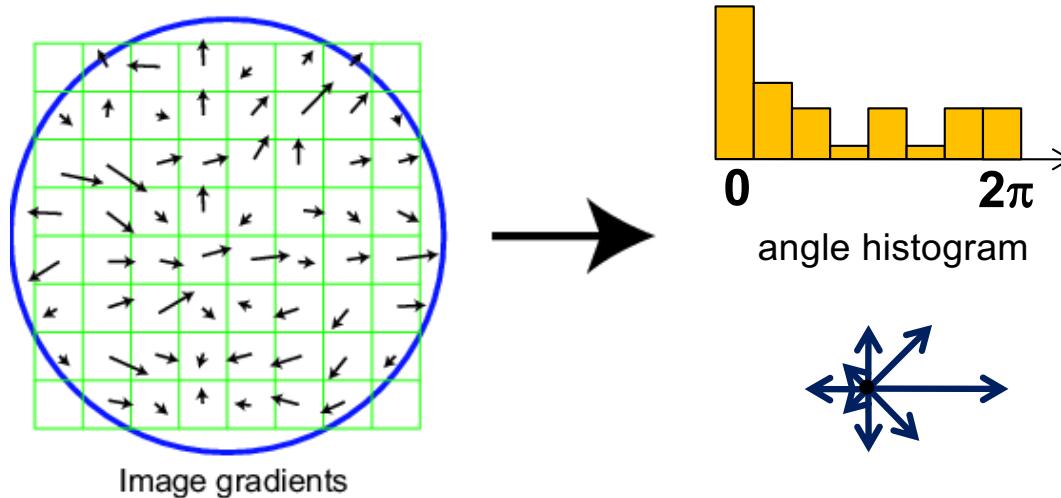
Gradients



Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute gradient orientation for each pixel
- Create histogram over edge orientations weighted by magnitude
- That's your feature descriptor!

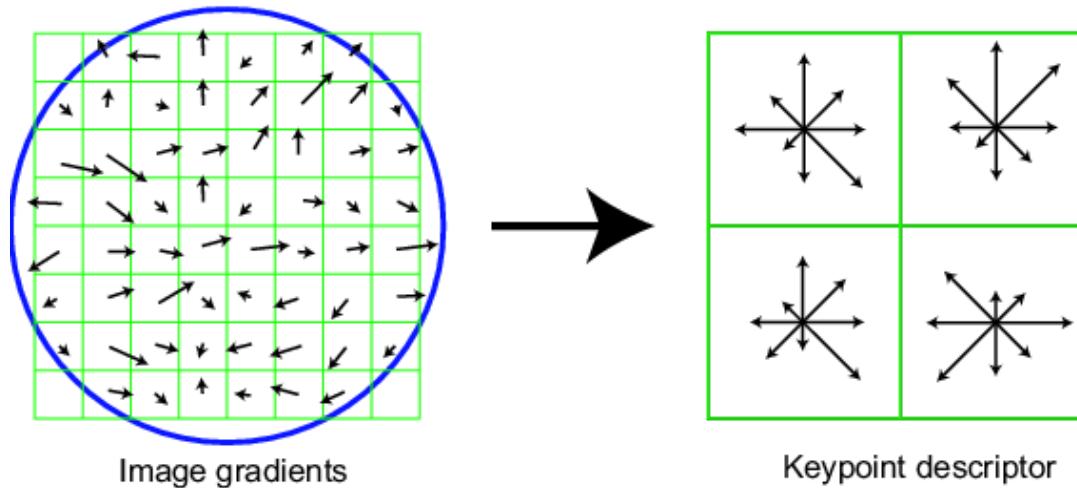


Adapted from L. Zitnick, D. Lowe

Scale Invariant Feature Transform

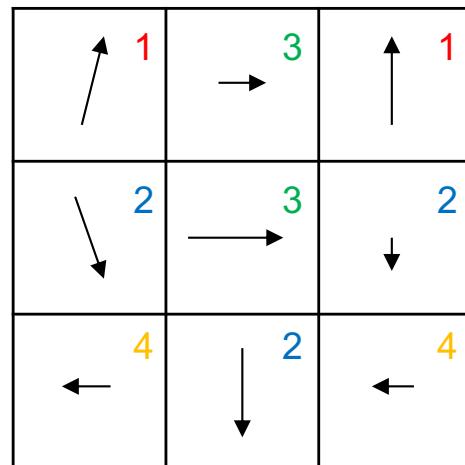
Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Quantize the gradient orientations i.e. snap each gradient to one of 8 angles
- Each gradient contributes not just 1, but magnitude(gradient) to the histogram, i.e. stronger gradients contribute more
- $16 \text{ cells} * 8 \text{ orientations} = 128 \text{ dimensional descriptor}$ for each detected feature

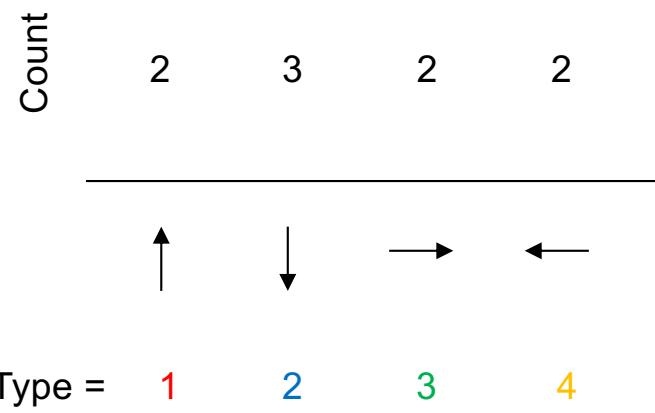


Adapted from L. Zitnick, D. Lowe

Scale Invariant Feature Transform



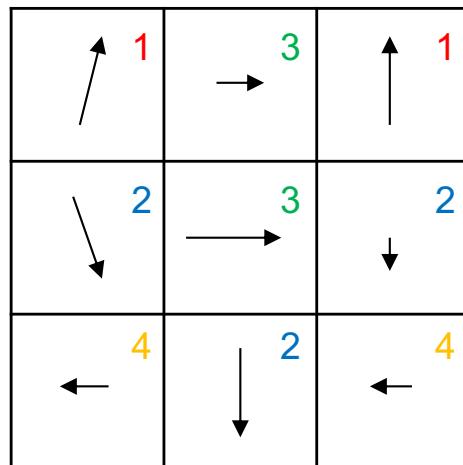
Uniform weight (ignore magnitude)



Gradients

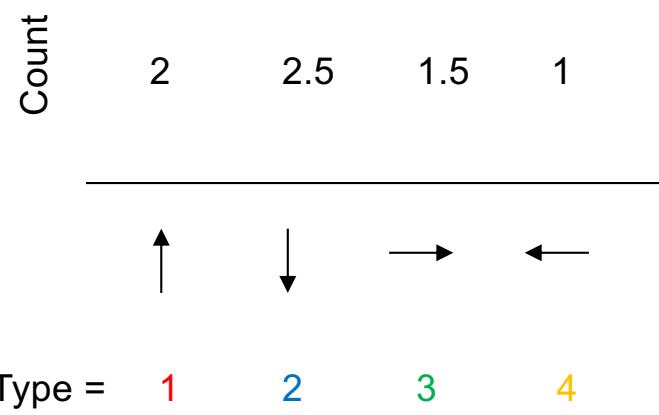
Histogram of gradients

Scale Invariant Feature Transform



Gradients

Weight contribution by magnitude
(e.g. long = 1, short = 0.5)



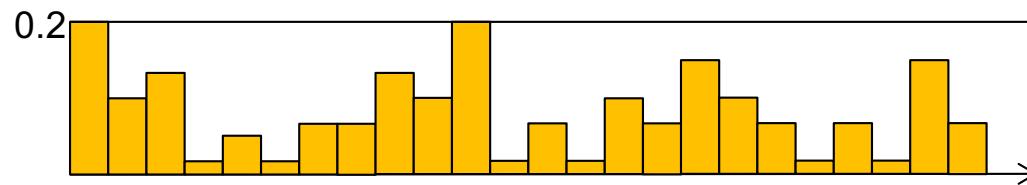
Histogram of gradients

Scale Invariant Feature Transform

Full version

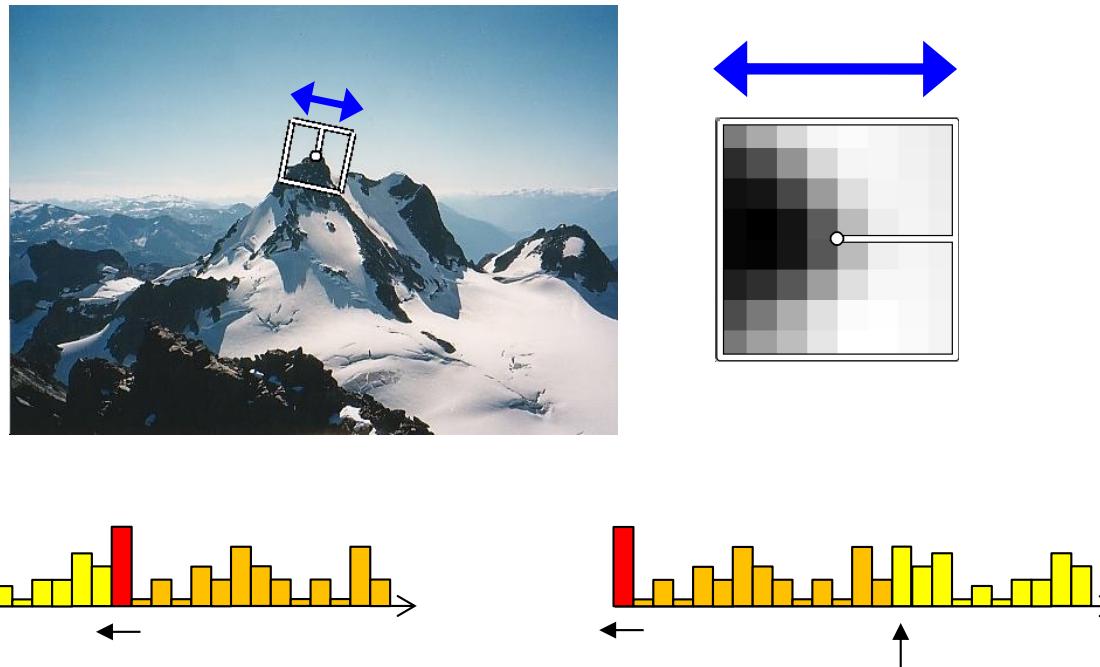
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Quantize the gradient orientations i.e. snap each gradient to one of 8 angles
- Each gradient contributes not just 1, but magnitude(gradients) to the histogram, i.e. stronger gradients contribute more
- 16 cells * 8 orientations = 128 dimensional descriptor for each detected feature
- Normalize + clip (threshold normalize to 0.2) + normalize the descriptor
- We want:

$$\sum_i d_i = 1 \quad \text{such that: } d_i < 0.2$$



Adapted from L. Zitnick, D. Lowe

Making descriptor rotation invariant



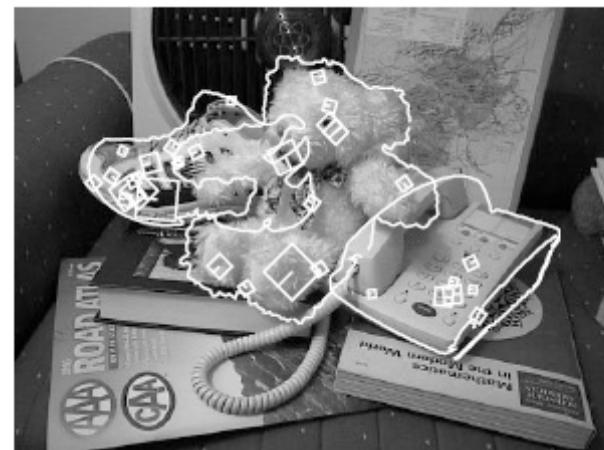
- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation

Adapted from K. Grauman, image from Matthew Brown

SIFT is robust

- 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
- **Fast and efficient**—can run in real time
- Can be made to work without feature detection, resulting in “**dense SIFT**” (more points means robustness to occlusion)
- One commonly used implementation
 - <http://www.vlfeat.org/overview/sift.html>

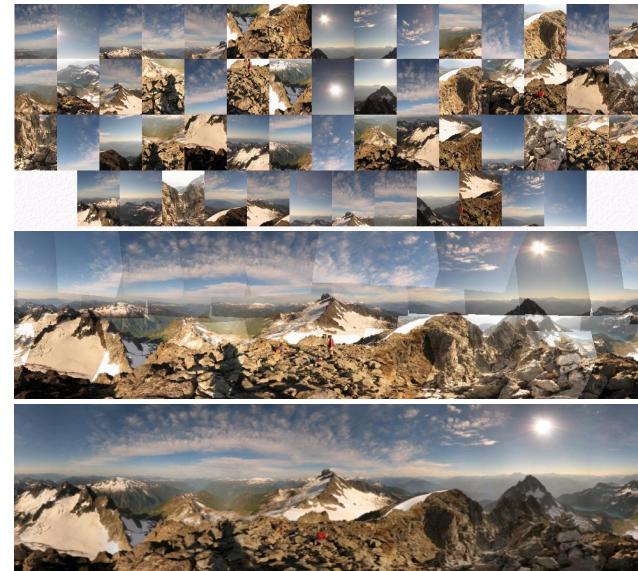
Examples of using SIFT



Adriana Kovashka

Applications of local invariant features

- Object recognition
- Indexing and retrieval
- Robot navigation
- 3D reconstruction
- Motion tracking
- Image alignment
- Panoramas and mosaics
- ...



<http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html>

Adapted from K. Grauman and L. Lazebnik

Lab 3: SIFT

Duration: 10 min



To join, go to: ahaslides.com/OGYZC



Please, select two images and draw SIFT matches among these images. Then, upload your result.

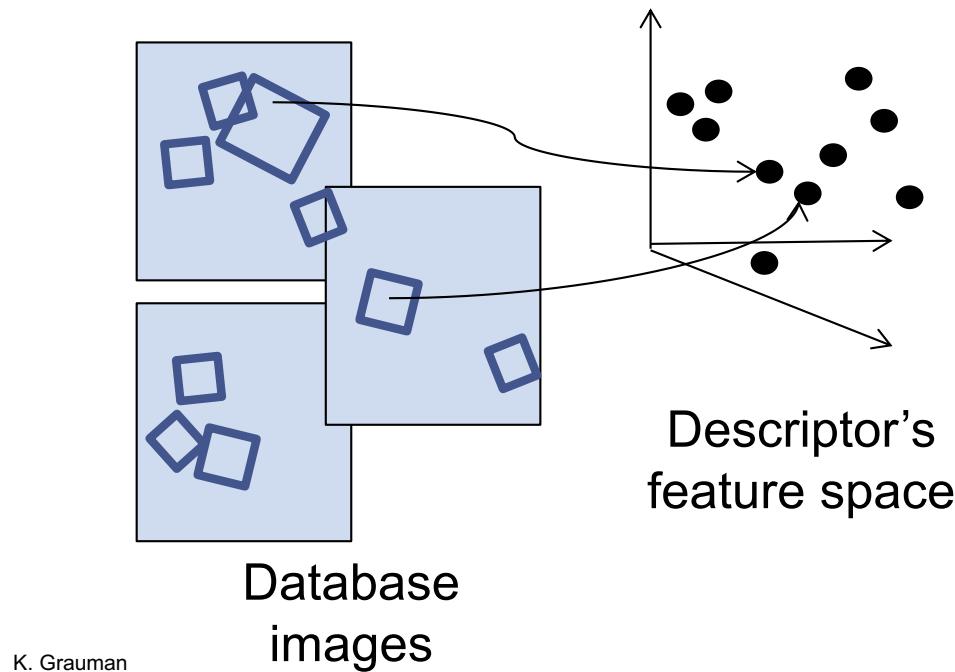
^ Get Feedback

Plan for this lecture

- Feature detection / keypoint extraction
 - Corner detection
 - Blob detection
- Feature description (of detected features)
- Matching features across images

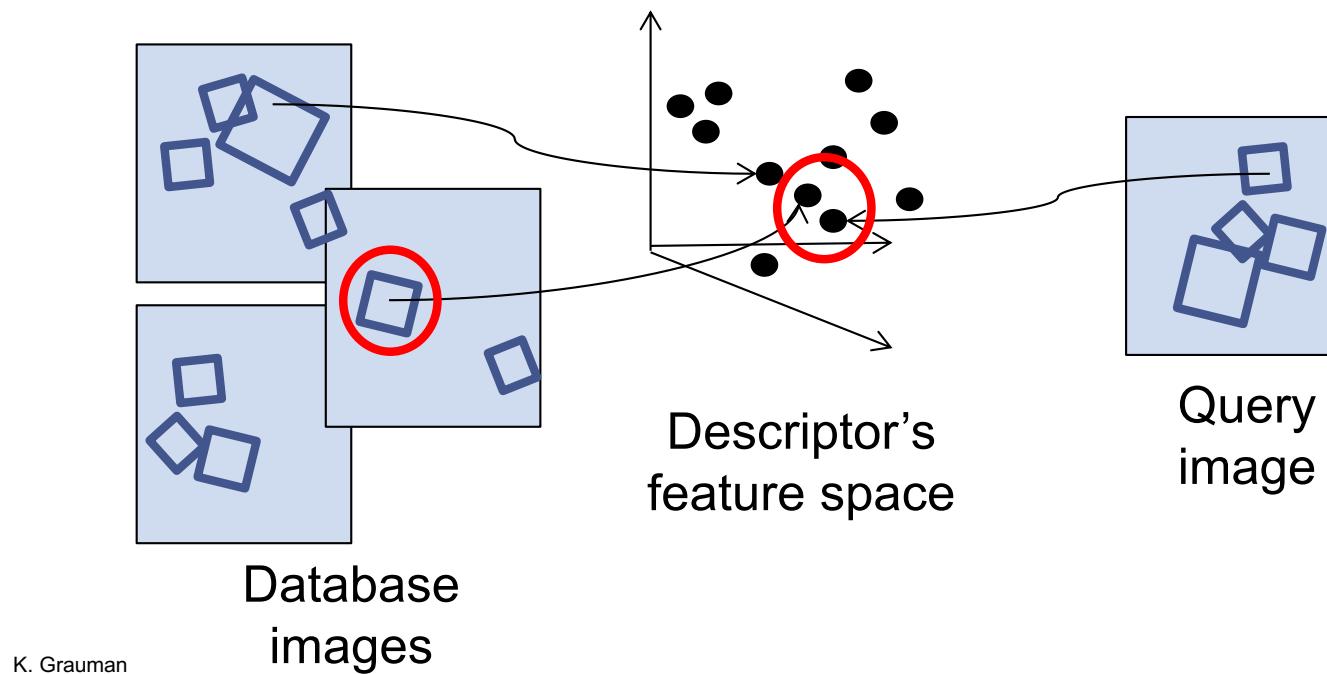
Matching Local Features Setup

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



Matching Local Features Setup

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



K. Grauman

Indexing local features

Index	
"Along I-75," From Detroit to Florida; <i>inside back cover</i>	Butterfly Center, McGuire; 134
"Drive I-95," From Boston to Florida; <i>inside back cover</i>	CAA (see AAA)
1929 Spanish Trail Roadway;	CCC, The; 111,113,115,135,142
101-102,104	Ca d'Zan; 147
511 Traffic Information; 83	Caloosahatchee River; 152
AIA (Barrier Isl) - I-95 Access; 86	Name; 150
AAA (and CAA); 83	Canaveral Natnl Seashore; 173
AAA National Office; 88	Cannon Creek Airpark; 130
Abbreviations,	Canopy Road; 106,169
Colored 25 mile Maps; cover	Cape Canaveral; 174
Exit Services; 196	Castillo San Marcos; 169
Travelogue; 85	Cave Diving; 131
Africa; 177	Cayo Costa, Name; 150
Agricultural Inspection Strs; 126	Celebration; 93
An-Tah-Thi-Ki Museum; 160	Charlotte County; 149
Air Conditioning, First; 112	Charlotte Harbor; 150
Alabama; 124	Chautauqua; 116
Alachua; 132	Chipley; 114
County; 131	Name; 115
Alafia River; 143	Choctawhatchee, Name; 115
Alapaha, Name; 126	Circus Museum, Ringling; 147
Alfred B Macay Gardens; 169	Citrus; 88,97,130,136,140,180
Alligator Alley; 154-155	CityPlace, W Palm Beach; 180
Alligator Farm, St Augustine; 169	City Maps,
Alligator Hole (definition); 157	Fl Lauderdale Expwy; 194-195
Alligator, Buddy; 155	Jacksonville; 163
Alligators; 100,135,138,147,156	Kissimmee Expwy; 192-193
Anastasia Island; 170	Miami Expressways; 194-195
Anhica; 108-109,146	Orlando Expressways; 192-193
Apalachicola River; 112	Pensacola; 26
Appleton Mus of Art; 136	Tallahassee; 191
Aquifer; 102	Tampa-St. Petersburg; 63
Arabian Nights; 94	St. Augustine; 191
Art Museum, Ringling; 147	Civil War; 100,108,127,138,141
Aruba Beach Cafe; 183	Clearwater Marine Aquarium; 187
Aucilla River Project; 106	Collier County; 154
Babcock-Web WMA; 151	Collier, Barron; 152
Bahia Mar Marina; 184	Colonial Spanish Quarters; 168
Baker County; 99	Columbia County; 101,128
Barefoot Mailmen; 182	Coquina Building Material; 165
Barge Canal; 137	Corkscrew Swamp, Name; 154
Bee Line Expy; 80	Cowboys; 95
Beltz Outlet Mall; 89	Crab Trap II; 144
Bernard Castro; 136	Cracker, Florida; 88,95,132
Big "I"; 165	Crossstown Expy; 11,35,98,143
Big Cypress; 155,158	Cuban Bread; 184
Big Foot Monster; 105	Dade Battlefield; 140
Billie Swamp Safari; 160	Dade, Mt; Francis; 139-140,161
Blackwater River SP; 117	Dania Beach Hurricane; 184
Blue Angels	Daniel Boone, Florida Walk; 117
A4-C Skyhawk; 117	Daytona Beach; 172-173
Atrium; 121	De Land; 87
Blue Springs SP; 87	De Soto, Hernando.
	Anhica; 108-109,146
	Countv; 149
	Driving Lanes; 85
	Duval County; 163
	Eau Gallie; 175
	Edison, Thomas; 152
	Eglin AFB; 116-118
	Eight Reale; 176
	Ellenton; 144-145
	Emanuel Point Wreck; 120
	Emergency Callboxes; 83
	Epiphytes; 142,148,157,159
	Escambia Bay; 119
	Bridge (I-10); 119
	County; 120
	Esterro; 153
	Everglade; 90,95,139-140,154-160
	Draining of; 156,181
	Wildlife MA; 160
	Wonder Gardens; 154
	Falling Waters SP; 115
	Fantasy of Flight; 95
	Fayer Dykes SP; 171
	Fires, Forest; 166
	Fires, Prescribed; 148
	Fisherman's Village; 151
	Flagler County; 171
	Flagler, Henry; 97,165,167,171
	Florida Aquarium; 186
	Florida,
	12,000 year ago; 187
	Cavern SP; 114
	Map of all Expressways; 2-3
	Mus of Natural History; 134
	National Cemetery; 141
	Part of Africa; 177
	Platform; 187
	Sheriff's Boys Camp; 126
	Sports Hall of Fame; 130
	Sun 'n Fun Museum; 97
	Supreme Court; 107
	Florida's Turnpike (FTP); 178,189
	25 mile Strip Maps; 66
	Administration; 189
	Coin System; 190
	Exit Services; 189
	HEFT; 76,161,190
	History; 189
	Names; 189
	Service Plazas; 190
	Spur SR91; 76
	Ticket System; 190
	Toll Plazas; 190
	Ford, Henry; 152
	Fort Barrancas; 122
	Buried Alive; 123
	Fort Caroline; 164

- 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”.

Visual Words: main idea

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

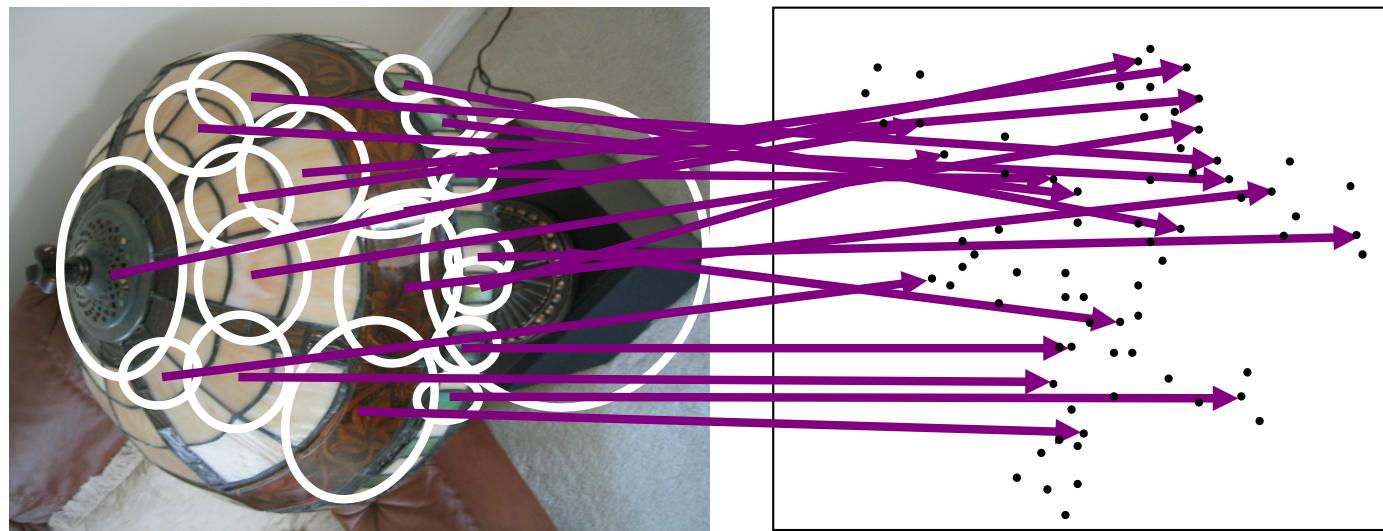


e.g., SIFT descriptor space: each point is 128-dimensional

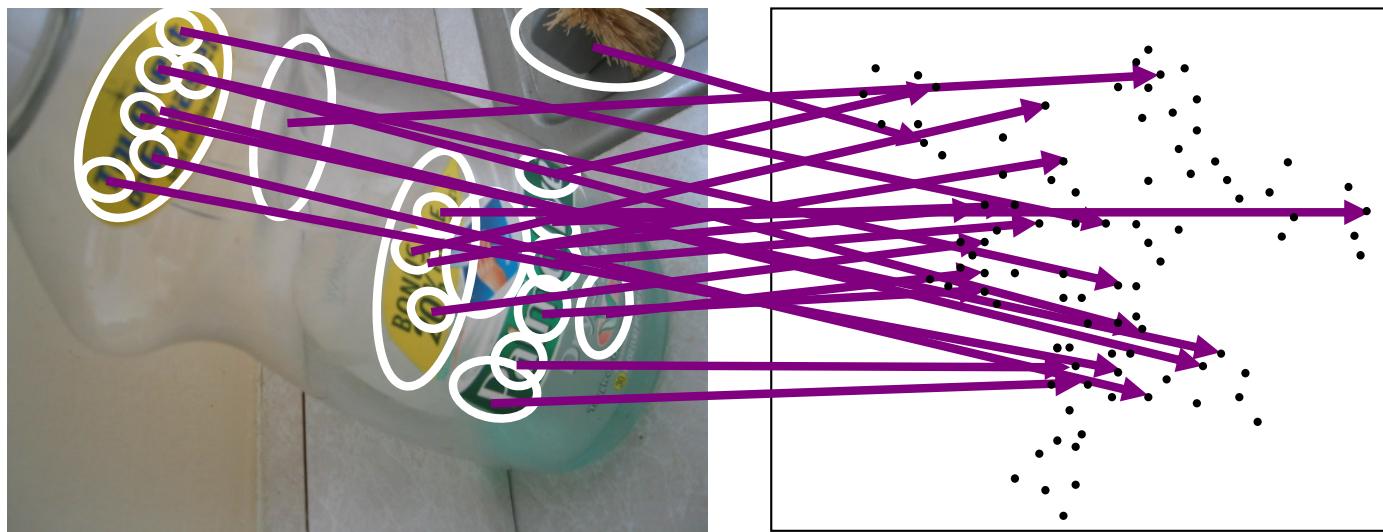
Visual Words: main idea

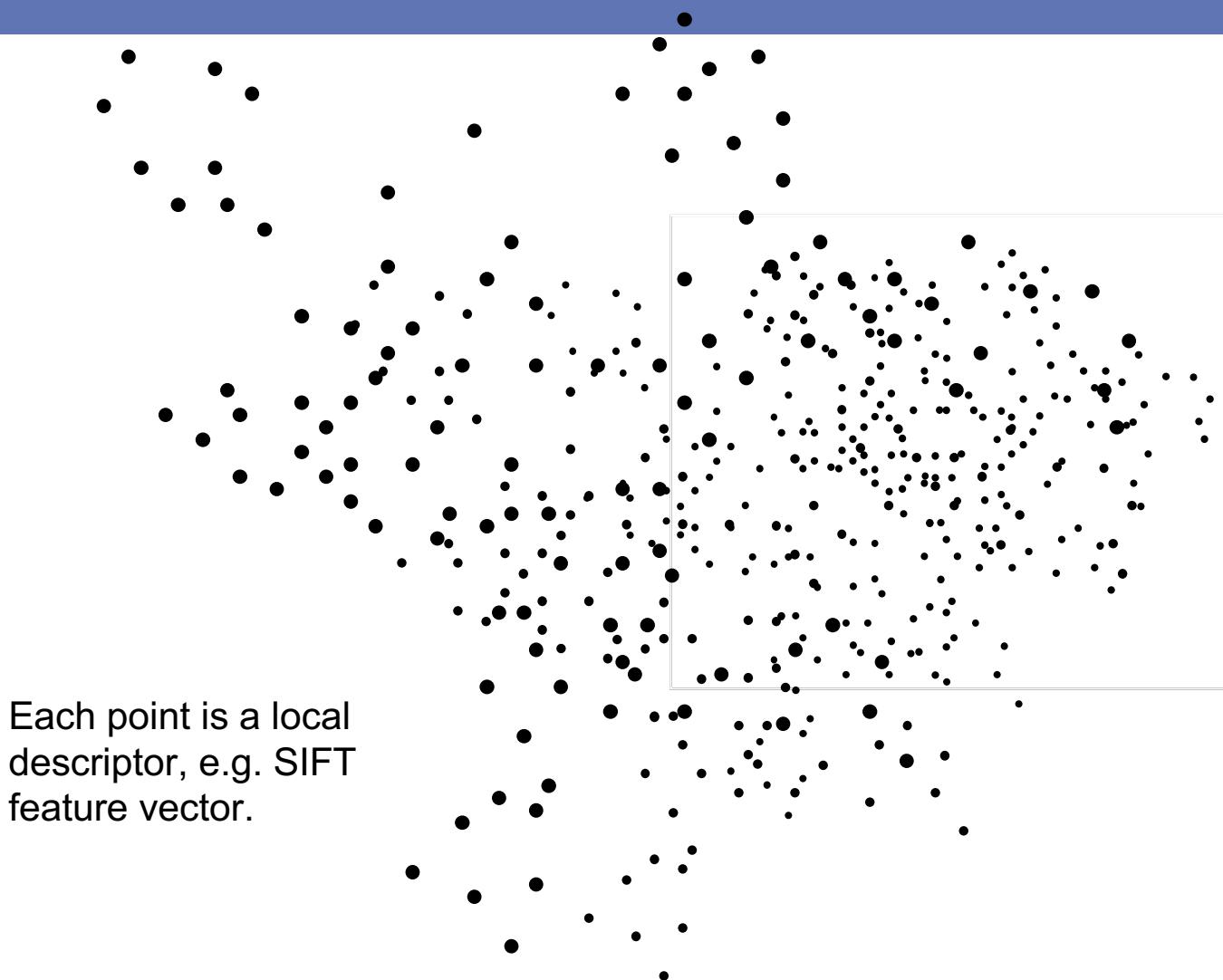


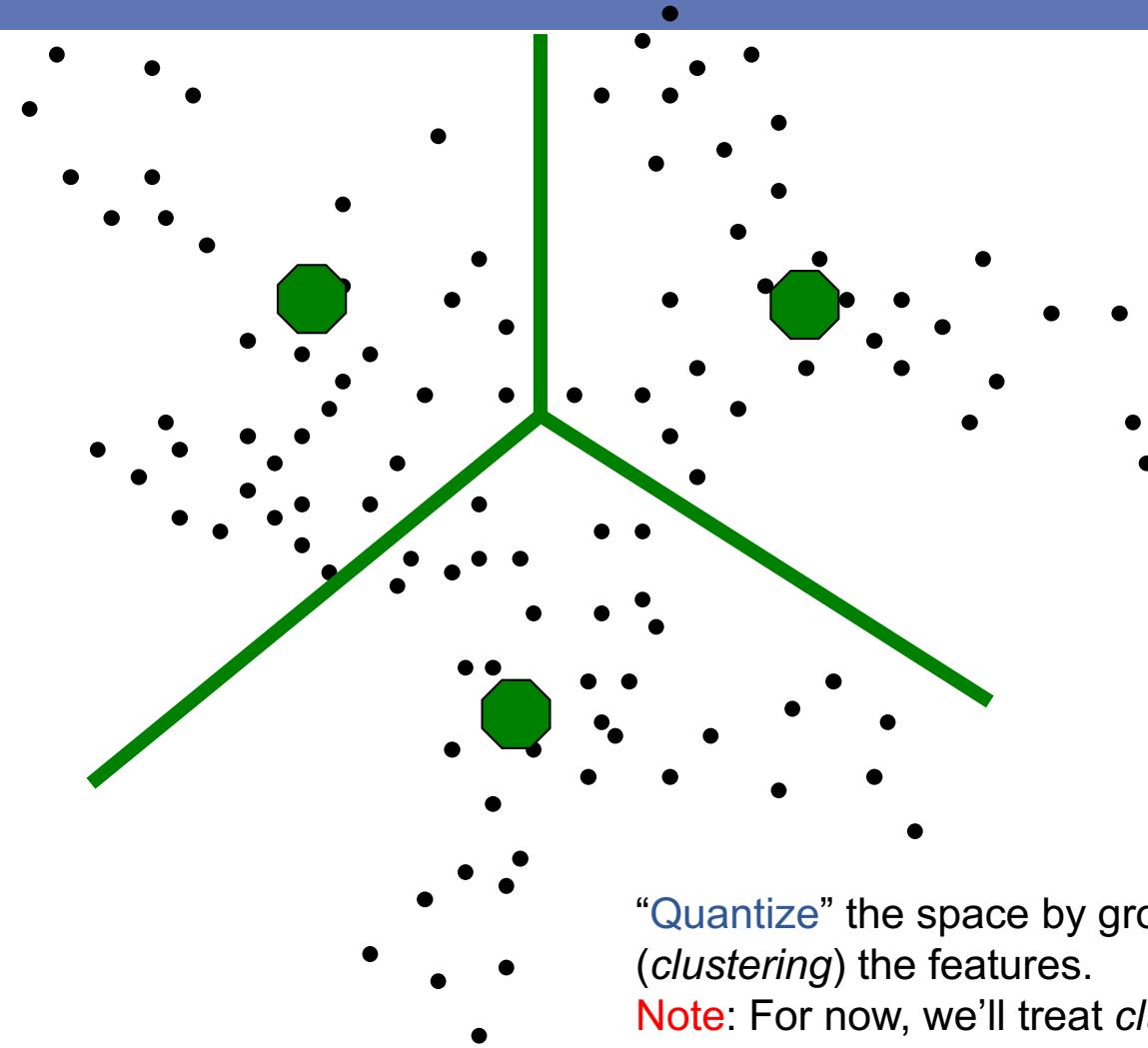
Visual Words: main idea



Visual Words: main idea



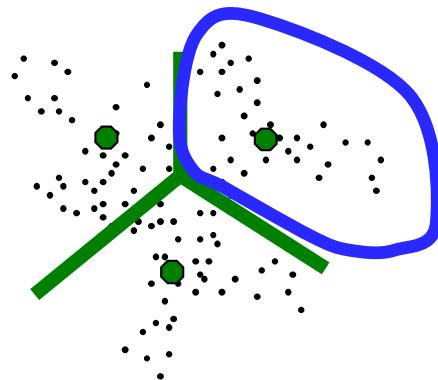




“Quantize” the space by grouping
(*clustering*) the features.
Note: For now, we’ll treat *clustering*
as a black box.

Visual Words

- Patches on the right = regions used to compute SIFT
- Each group of patches belongs to the same “visual word”



Adapted from K. Grauman

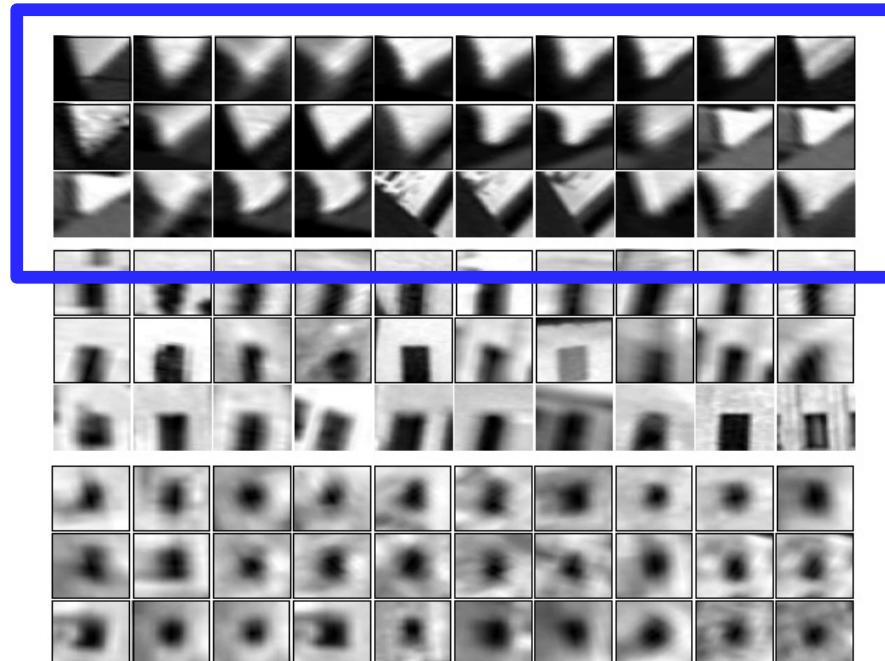
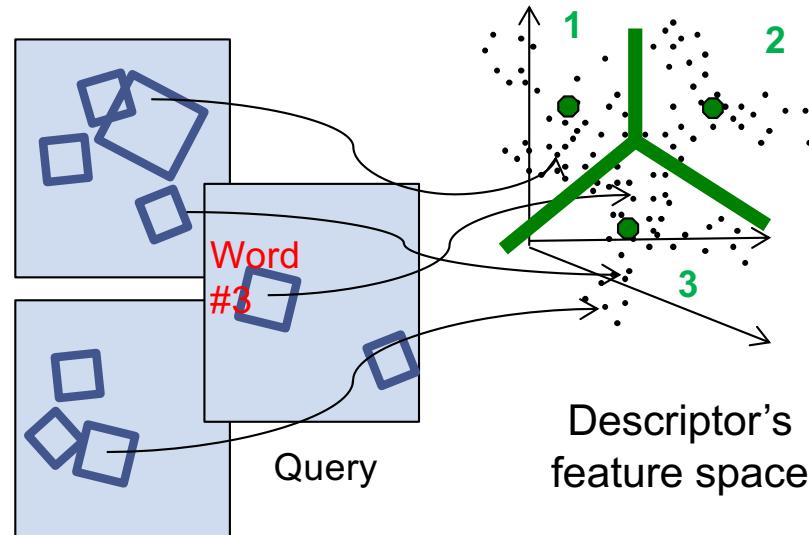


Figure from Sivic & Zisserman, ICCV 2003

Visual Words for Indexing

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



- Each cluster has a center
- To determine which word to assign to new image region (e.g. query), find closest cluster center
- To compare features:* Only compare query to others in same cluster, or just compare word IDs
- To compare images:* see next few slides

Adapted from K. Grauman

How to describe documents with words

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 us from our eyes. For a long time now we have known that the retinal image is projected point to point to visual areas in the cerebral cortex. Upon visual stimulation, the retina projects its messages to the brain, and Wiesel and Hubel have shown that the origin of the visual system lies in the eye. There is a course of events which starts with the impulses along the optic nerve, through the layers of the optical nerve, to the brain. Wiesel and Hubel have been able to demonstrate that the message about the image falling on the retina undergoes a step-wise analysis. In this 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.

**sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel**

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 \$560bn in imports to \$450bn. The ministry also said it would further avert a trade war with the US by urging that China's central bank, the People's Bank, take a deliberate policy to manage the yuan. It agreed that the Chinese government should allow the yuan to appreciate against the dollar. The Chinese government has been allowed to manage the yuan against the dollar since 1994 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.

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

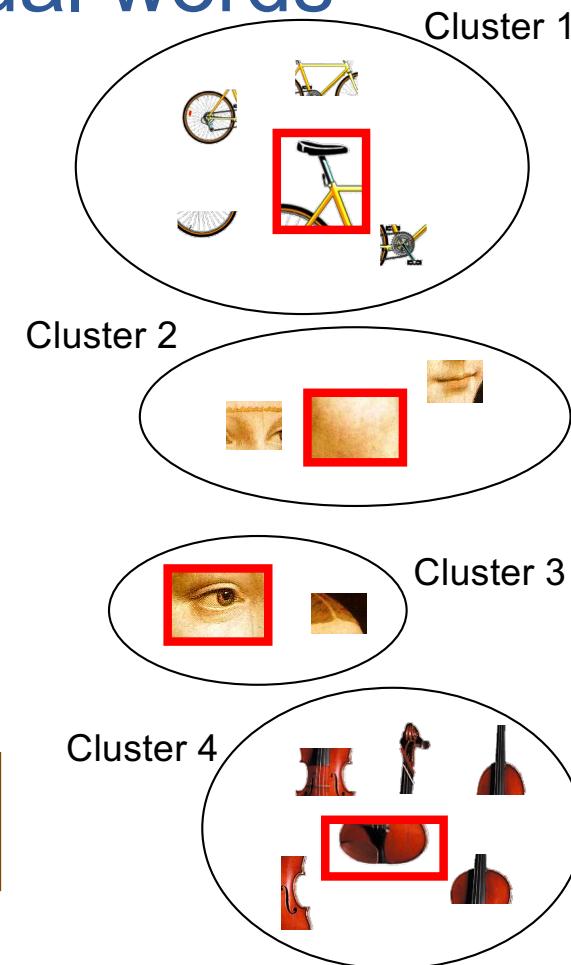
Describing images with visual words

- Summarize entire image based on its distribution (histogram) of word occurrences
- Analogous to bag of words representation commonly used for documents

Feature patches:



Adapted from K. Grauman



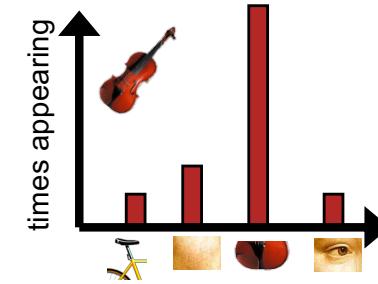
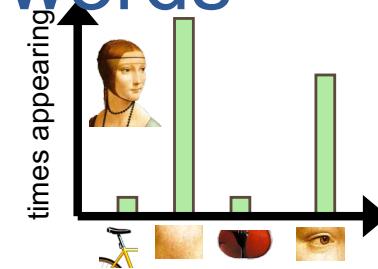
Describing images with visual words

- Summarize entire image based on its distribution (histogram) of word occurrences
- Analogous to bag of words representation commonly used for documents

Feature patches:



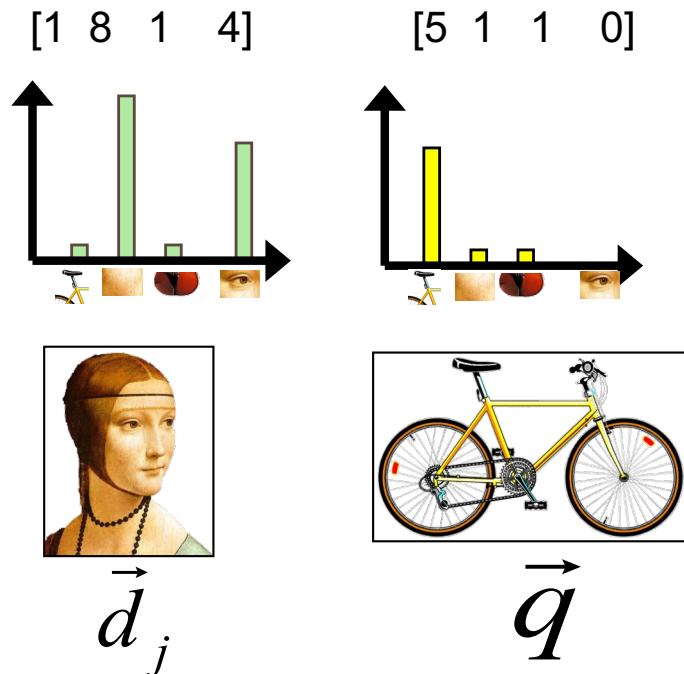
K. Grauman



Visual words

Comparing bags of words

- Similarity of images measured as normalized scalar product between their word occurrence counts
- Can be used to rank results (nearest neighbors of query)



$$\begin{aligned} sim(d_j, q) &= \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|} \\ &= \frac{\sum_{i=1}^V d_j(i) * q(i)}{\sqrt{\sum_{i=1}^V d_j(i)^2} * \sqrt{\sum_{i=1}^V q(i)^2}} \end{aligned}$$

for vocabulary of V words

Bags of words: pros and cons

- + Flexible to geometry / deformations / viewpoint
- + Compact summary of image content

- Basic model ignores geometry – verify afterwards
- What is the optimal vocabulary size?
- Background and foreground mixed when bag covers whole image

Summary: Inverted file index and bags of words similarity

Offline:

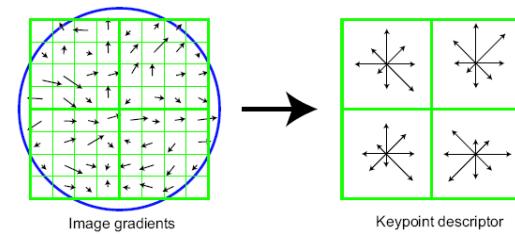
- Extract features in database images, cluster them to find words = cluster centers, make index

Online (during search):

1. Extract words in query (extract features and map each to closest cluster center)
2. Use inverted file index to find database images relevant to query
3. Rank database images by comparing word counts of query and database image

Summary

- Keypoint detection: repeatable and distinctive
 - Corners, blobs, stable regions
 - Laplacian of Gaussian, automatic scale selection
- Descriptors: robust and selective
 - Histograms for robustness to small shifts and translations (SIFT descriptor)
- Matching: cluster and index
 - Compare images through their feature distribution



Adapted from D. Hoiem, K. Grauman