lecture 14

Scale space (3)

- SIFT (feature descriptors, image indexing)
- image registration

Image Indexing / Recognition

Suppose you are given a set of (training) images. You are then given a new image, and you wish to find the best match among the training images.

Challenges

- new image might have different focal length, orientation, exposure, etc. (it might be taken with a different camera!)

Image registration model is not suitable here, i.e.

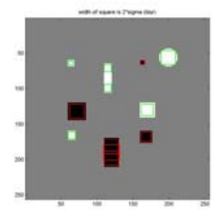
$$\sum (I(x,y) - J(x+h_x, y+h_y))^2$$

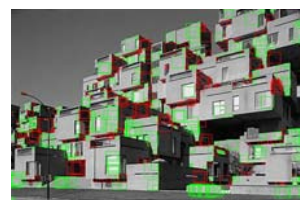
What to do?

Recall last lecture...

"Box detection": find maxima & minima of

$\nabla^2 g_\sigma(x,y) * I(x,y)$ over (x,y,σ)





The max/min are often called "keypoints" (x, y, σ)

"Keypoint" = "Feature"

- We wish to describe the intensity structure in the Ngd of keypoints, so that we can distinguish / recognize them.
- Many "feature descriptors" have been proposed. We will describe one of them: SIFT.

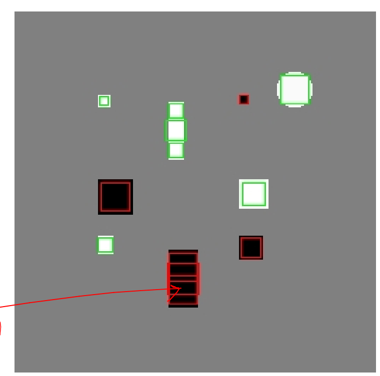
SIFT is based on

$\nabla I * g_\sigma(x,y)$

- at the scale σ of the keypoint,
- in a spatial Ngd (x,y). of width about $6 \cdot \sigma$ around keypoint

Recall....

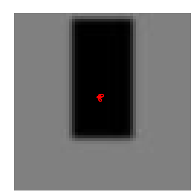
example: look at this keypoint (scale σ is 10 pixels)



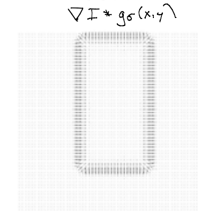
$I * G(x,y,\sigma)$

(x0,y0,sigma) = (120 , 182 , 1)

$\nabla I * g_\sigma(x,y)$



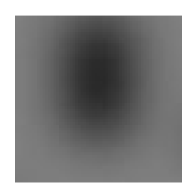
$\sigma = 1$



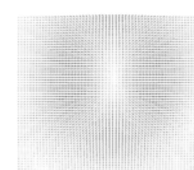
(x0,y0,sigma) = (120 , 182 , 10)

$\leftarrow 60 = 6 \cdot \sigma \rightarrow$

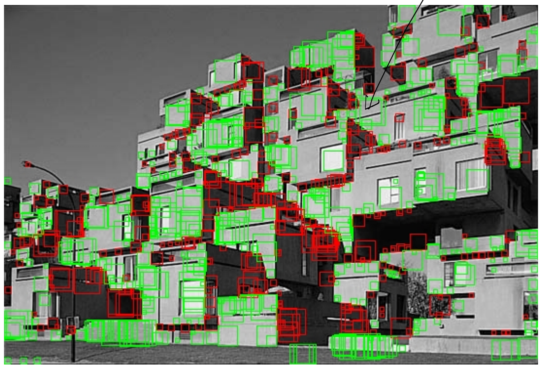
60



$\sigma = 10$

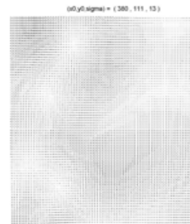


look at this
$\sigma = 13$ pixels



In fact, this is upside down.

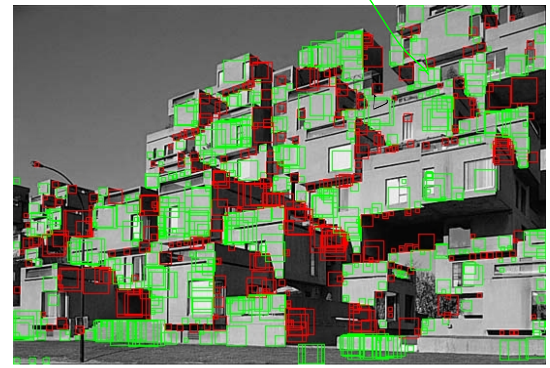
$\sigma = 1$.



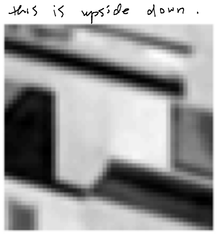
$\sigma = 13$

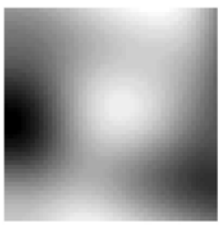
$6 \cdot \sigma = 78$ pixels

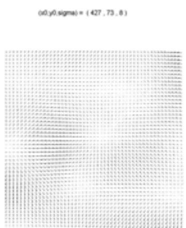
look at this
$\sigma = 8$ pixels



In fact, this is upside down.







$\sigma = 8$

$6 \cdot \sigma = 48$ pixels

Define a dominant orientation for each keypoint.

for each $(x,y) \in Ngd(x_0, y_0, \sigma)$ {

θ = direction of $\nabla I * g_\sigma(x,y)$

$\theta bin = \text{round}\left(\frac{\theta}{\text{bin width}}\right)$ // binwidth = 15 degrees

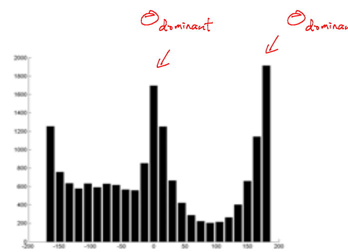
l = length of $\nabla I * g_\sigma(x,y)$

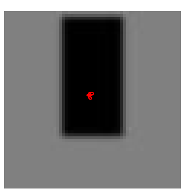
$hist(\theta_{bin}) \mathrel{+}= l \cdot g_\sigma(x - x_0, y - y_0)$

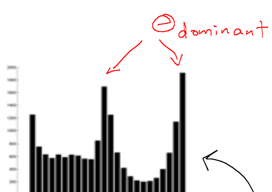
}

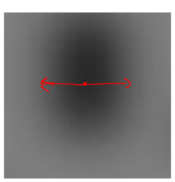
$$\theta_{dominant\,bin} = \underset{\theta bin}{\arg\max}\ hist(\theta bin)$$

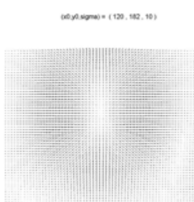
Allow multiple dominant orientations if maxes are similar $\Rightarrow$ multiple keypoints

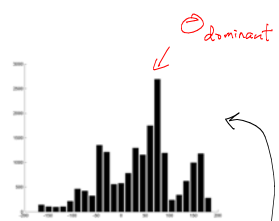


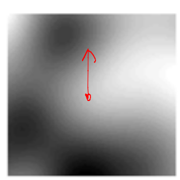


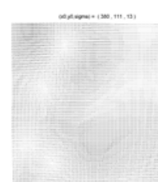


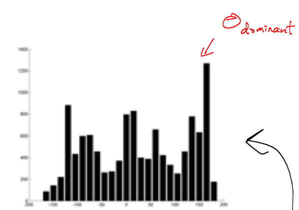


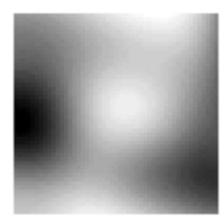


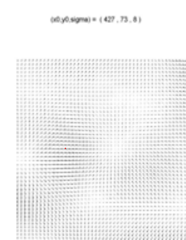








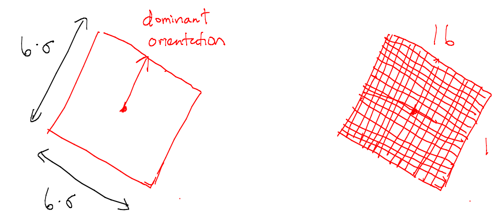




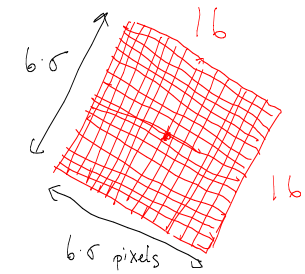
SIFT (scale invariant feature transform)

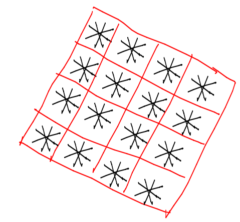
- find keypoints (max/min of $\nabla^2 I * g_\sigma(x,y)$)
- for each keypoint (x, y, σ)
  - find dominant orientation(s)
  - compute a "feature descriptor" for $\nabla I * g_\sigma$ in $Ngd(x, y, \sigma)$

Feature Descriptor



- Take a neighborhood of width 6σ and oriented in dominant direction.
- Define a 16x16 vector field $\nabla I * g_\sigma$





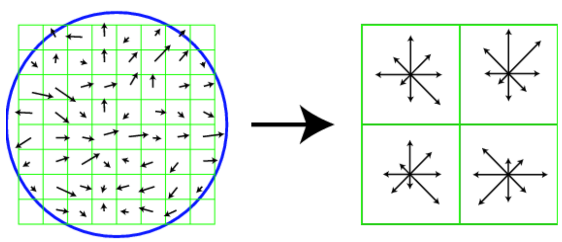
$16 \times 16 \times 2 = 512$

$\nabla I * g_\sigma(x,y)$

$4 \times 4 \times 8 = 128$

orientation histogram for each grid cell

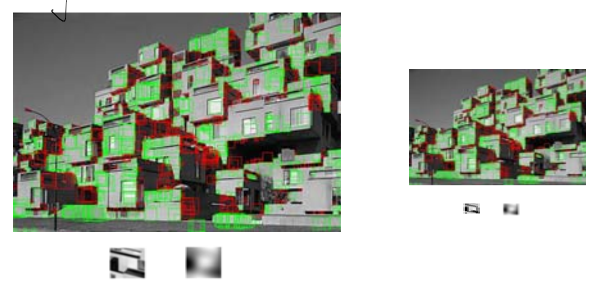
$\Rightarrow$ 128-d "feature descriptor"



$64 = 8 \times 8$ $\qquad$ $2 \times 2 \times 8$

- figure above taken from Lowe's 2004 paper
- often reproduced figure, but very confusing (heads up!) because different grid sizes are used. Also, $\nabla I * g_\sigma$ is smoother than what is shown.

Why is SIFT scale invariant?



Resizing the image doesn't change the feature descriptor!
(Mathematical proof omitted.)

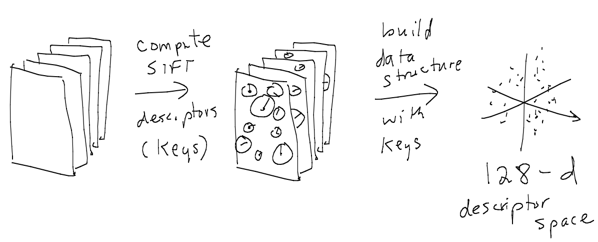
Applications

1) image indexing & recognition

2) image registration

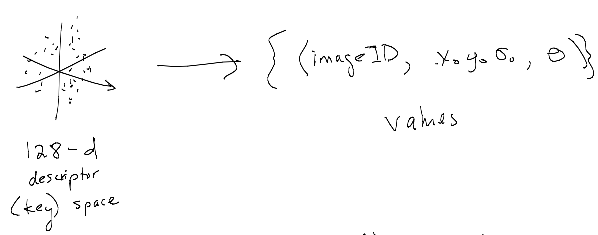
e.g. stitching panoramas

"Training" Images (Image database)



{128-d descriptor, imageID, $(x_0, y_0, \sigma_0, \theta)$}

Analogous to Hashing



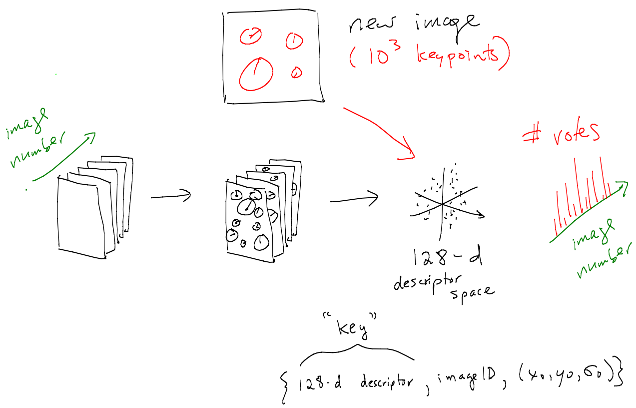
$\{\underbrace{\text{128-d descriptor}}_{\text{"key"}}, \underbrace{\text{imageID}, (x_0, y_0, \sigma_0, \theta)}_{\text{"value"}}\}$

Image Indexing / Recognition

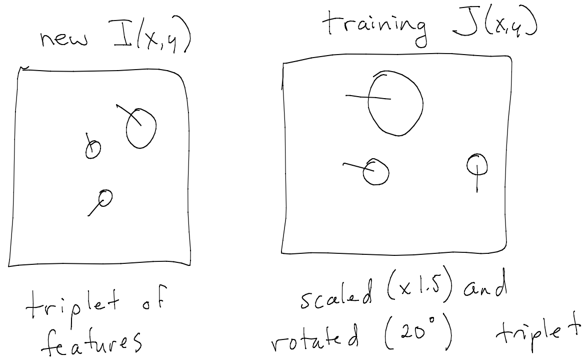
Given a new image $I(x,y)$, find the most similar image from training set.

- Compute keypoints and their feature descriptors (i.e. keys) from $I(x,y)$
- for each keypoint/descriptor, find similar descriptor points from training set.
- (naive) cast one vote for the image associated w/ each matching keypoint

Naive Voting Scheme



Non-naive: exploit relations between position and scale of key points



(Non-naive) Image Indexing / Recognition

Given a new image $I(x,y)$, find the most similar image from training set.

- Compute key points and their feature descriptors from $I(x,y)$
- for each keypoint/descriptor, find similar descriptor points from training set.
- Sample triplets of nearby keypoints in $I(x,y)$ and find consistent triplets in feature space (rotated and scaled, similar descriptors, one J)

For more details,

see paper by David Lowe

"Distinctive image features from scale invariant keypoints"

International Journal of Computer Vision 2004

Image Registration using SIFT

e.g. Panoramic Image Stitching

http://cvlab.epfl.ch/~brown/autostitch/autostitch.html

...

new topic

(still in scale space)

...

Image Registration (recall) NOT using SIFT

for each pixel (x_0, y_0)

find $(h_x\ h_y)$ that minimizes

$$\sum_{(x,y) \in Ngd(x_0 y_0)} \left(I(x+h_x, y+h_y) - J(x,y) \right)^2$$

Image Registration (recall)

for each pixel (x_0, y_0) {

  $k = 0$

  $(h_x^0, h_y^0) = (0,0)$

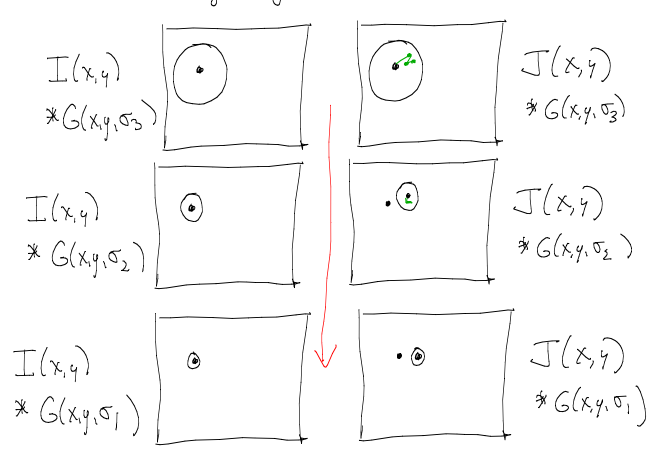
  repeat {

  - find $(h_x\ h_y)$ that minimizes
  $$\sum_{(x,y) \in Ngd(x_0 y_0)} \left(I(x + h_x^k + h_x,\ y + h_y^k + h_y) - J(x,y) \right)^2$$
  - $k = k+1$

  } until $(h_x\ h_y) \approx (0,0)$

}

Multiscale Image Registration ("Coarse to fine")



for $\sigma = \sigma_{big}$ down to σ_{small} {

for each pixel (x, y) {

- initialize (h_x^0, h_y^0) using the estimate from larger scale or $(0,0)$ if $\sigma == \sigma_{big}$
- estimate (h_x, h_y) at scale σ

}

}

Other Registration Approaches

Find 2x2 distortion matrix D and translation vector $(h_x\ h_y)$ that minimizes

$$\sum_{\vec{x} \in Ngd(\vec{x}_0)} \left(I(\vec{x} + D(\vec{x} - \vec{x}_0) + \vec{h}) - J(\vec{x}) \right)^2$$

where $\vec{x} = (x, y)$, $\vec{x}_0 = (x_0\ y_0)$, $\vec{h} = (h_x\ h_y)$

- 6 parameter motion model allows for rotation, scaling, shear, translation
- 1st order Taylor expansion about D_{ij}, $\vec{h}$
- solve a 6-d linear system
- again, you can do it multiscale