Tamara Berg

Representing Images

Announcements

- Alex will release HW this weekend
- Make-up class for last week will be scheduled later this semester

Representing Images

Keep all the pixels!

Pros? Cons?

2^{nd} Try

Compute average pixel

Pros? Cons?

3rd Try

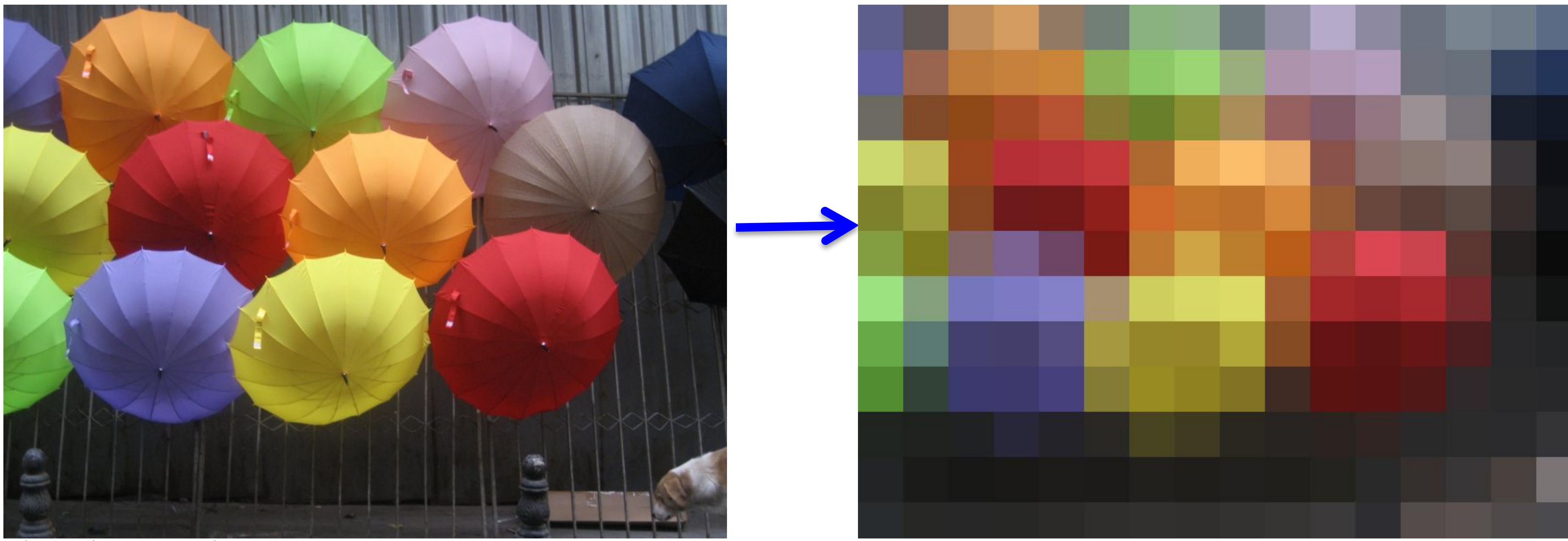


Photo by: marielito

Represent the image as a spatial grid of average pixel colors

Pros?

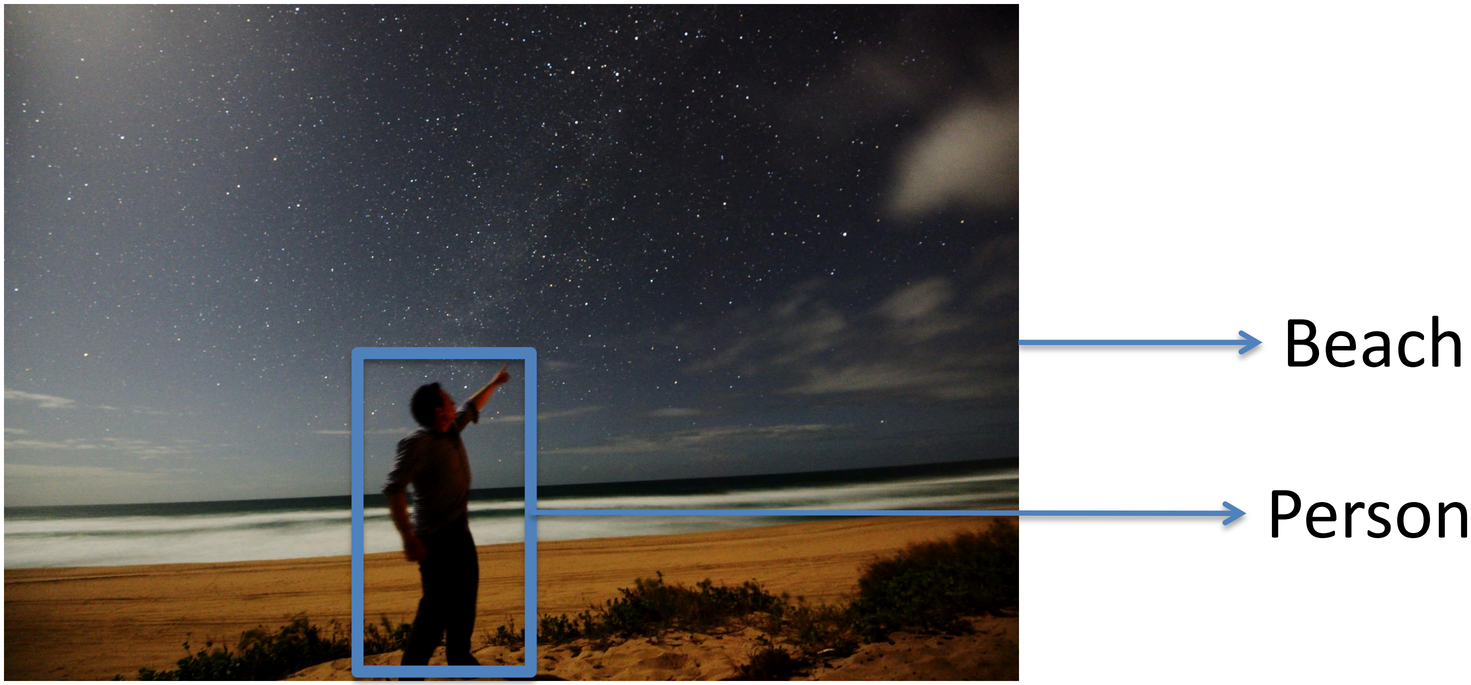
Cons?

QBIC system

- First content based image retrieval system
  - Query by image content (QBIC)
  - IBM 1995
  - QBIC interprets the virtual canvas as a grid of coloured areas, then matches this grid to other images stored in the database.

Feature Choices

Global vs Local?



Depends whether you want
whole or local image similarity

Feature types!

The representation of these two umbrella's should be similar….
Under a color based representation they look completely different!

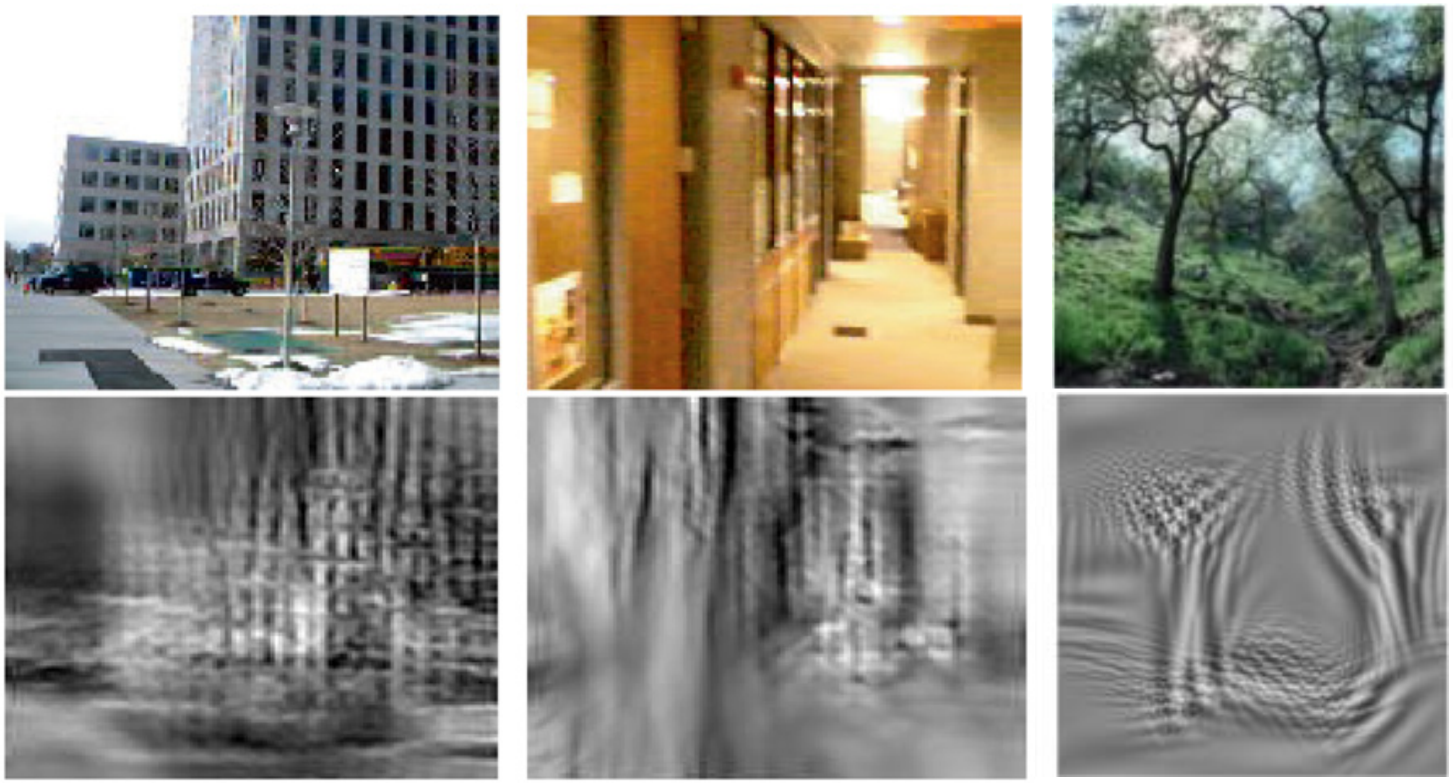
Outline

Image Representations

Global vs Local – depends on the task

Feature Type: color, shape, texture

Global Features



The “gist” of a scene: Oliva & Torralba, 2001

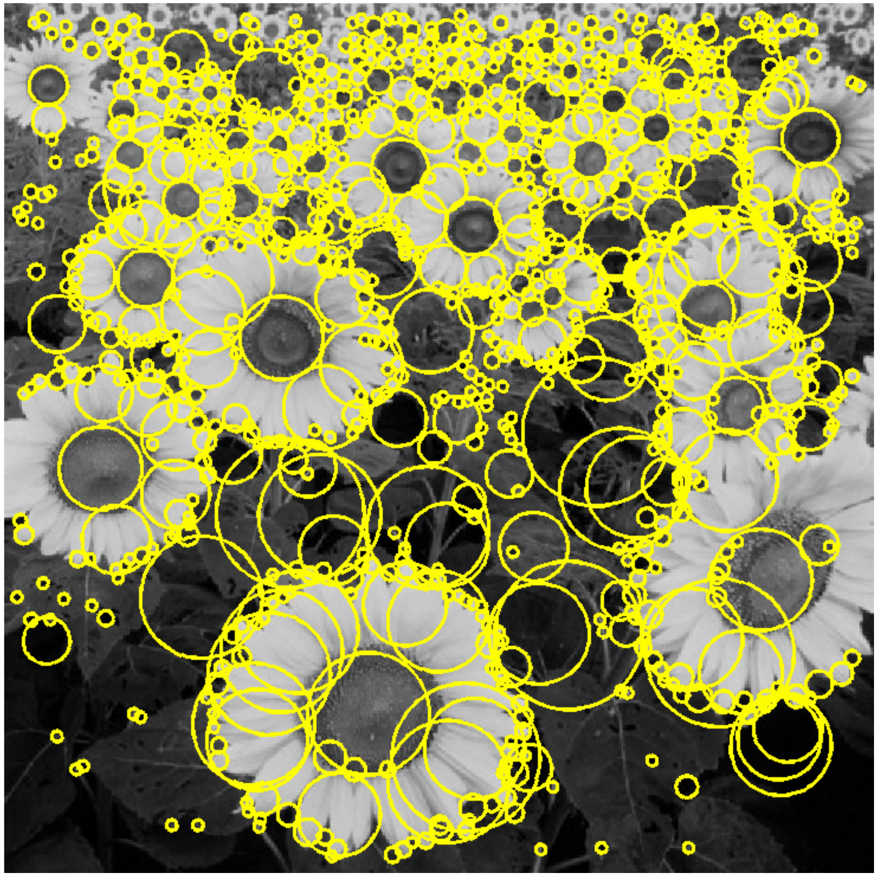
Limitations of global appearance models

- Can work on relatively simple patterns

- Not robust to clutter, occlusion

Local Features

Feature points (locations) + feature descriptors



source: Svetlana Lazebnik

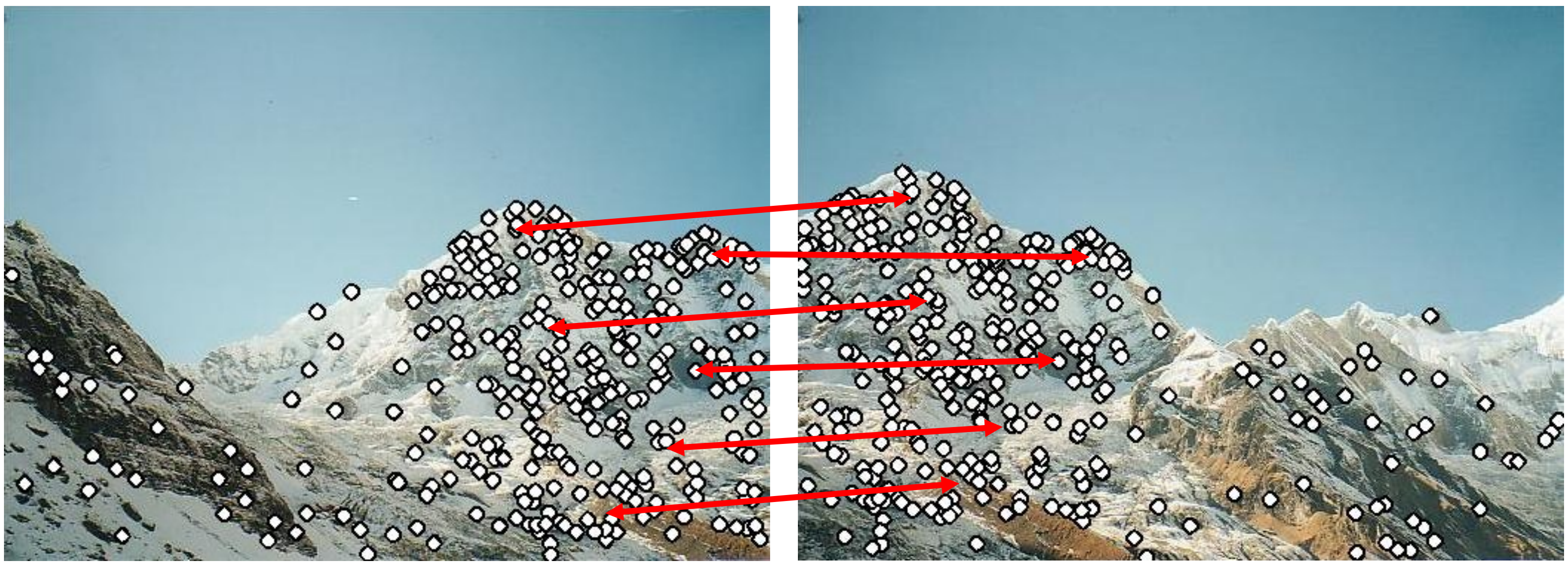
Why extract features?

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

source: Svetlana Lazebnik

Why extract features?

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



Step 1: extract features

Step 2: match features

source: Svetlana Lazebnik

Why extract features?

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

Step 1: extract features

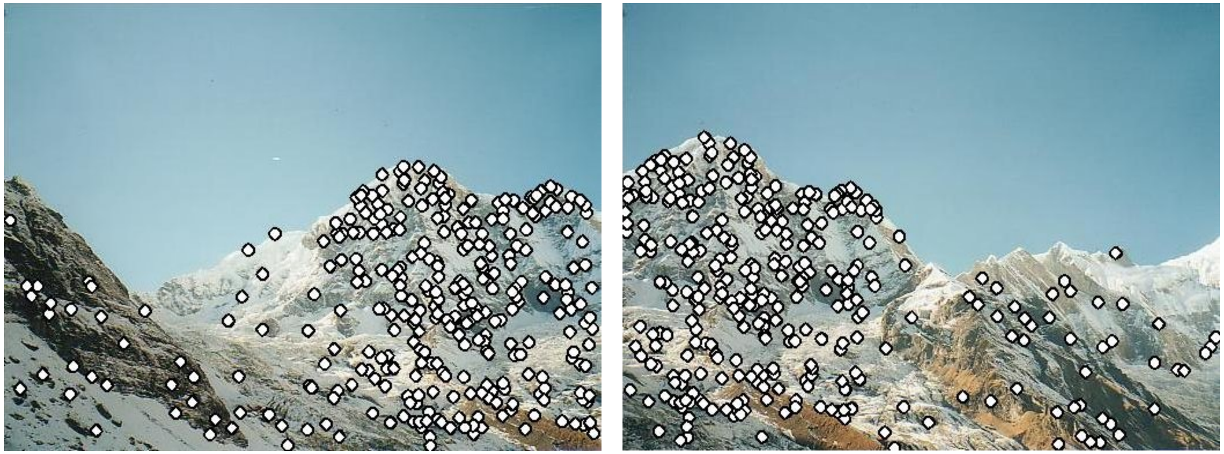
Step 2: match features

Step 3: align images

source: Svetlana Lazebnik

Local Features for object detection

Characteristics of good features



- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature has a distinctive description
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

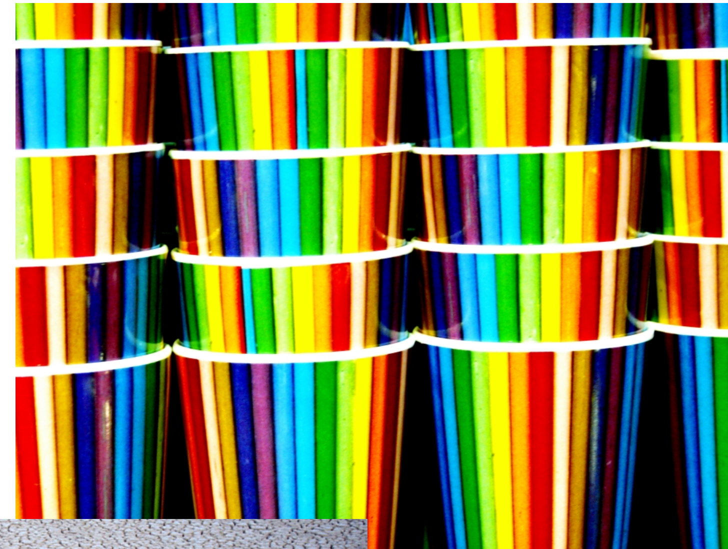
source: Svetlana Lazebnik

Local features - Applications

- Feature points are used for:
  - Retrieval
  - Tracking
  - Image alignment
  - 3D reconstruction
  - Object recognition
  - Object detection
  - Attribute recognition
  - Indexing and search
  - Robot navigation

Feature Types

Shape!

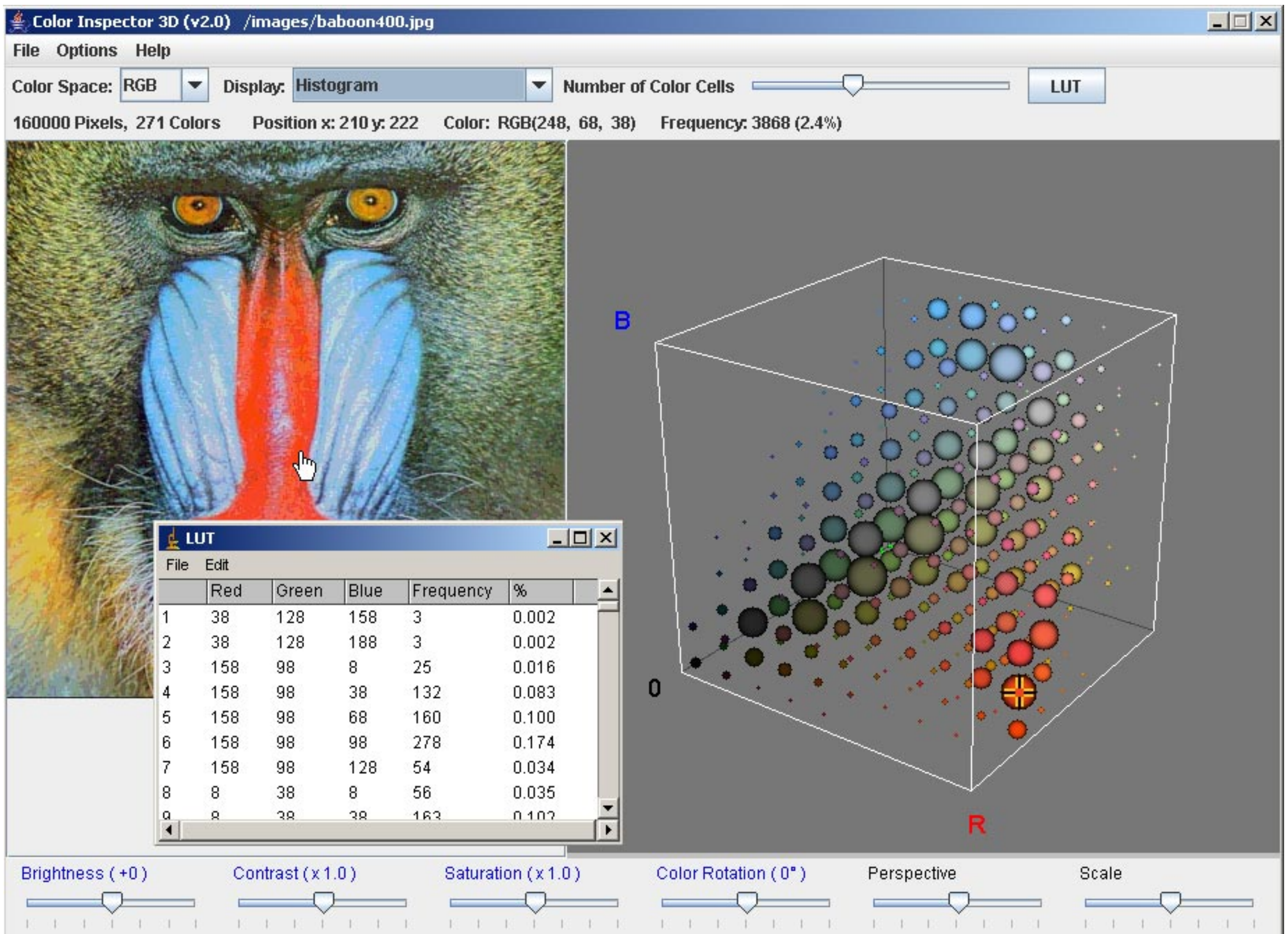


Color!

Texture!

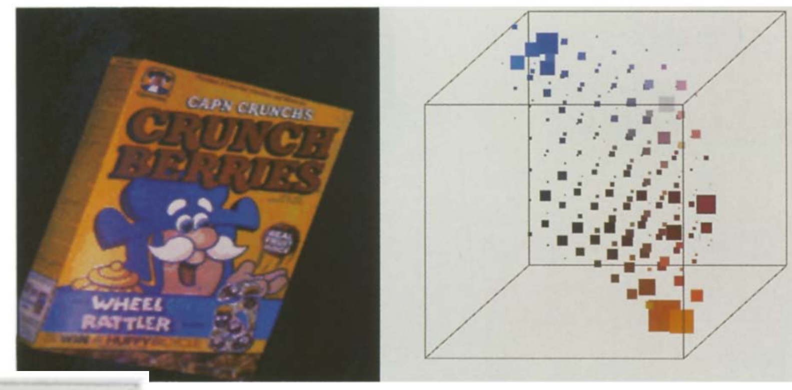
Color Features

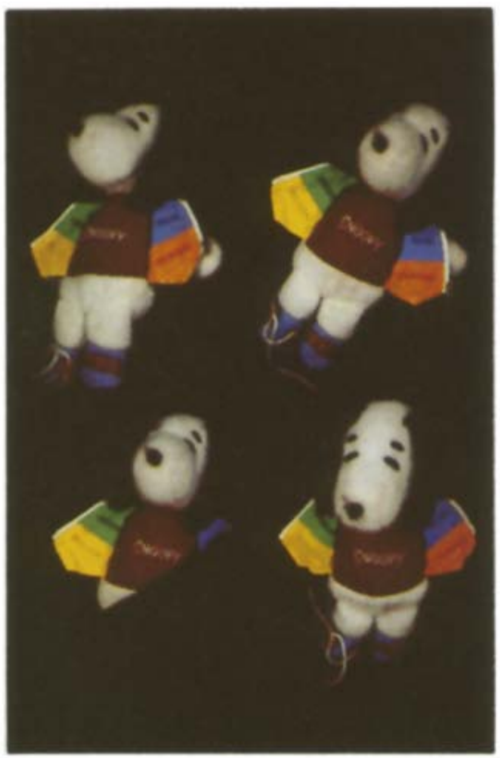
Color Histograms

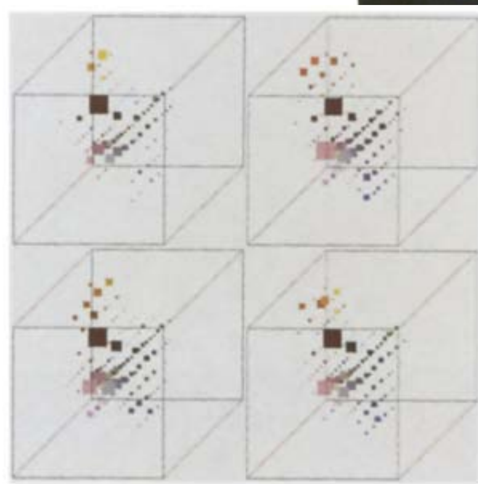


Uses of color in computer vision

Color histograms for indexing and retrieval





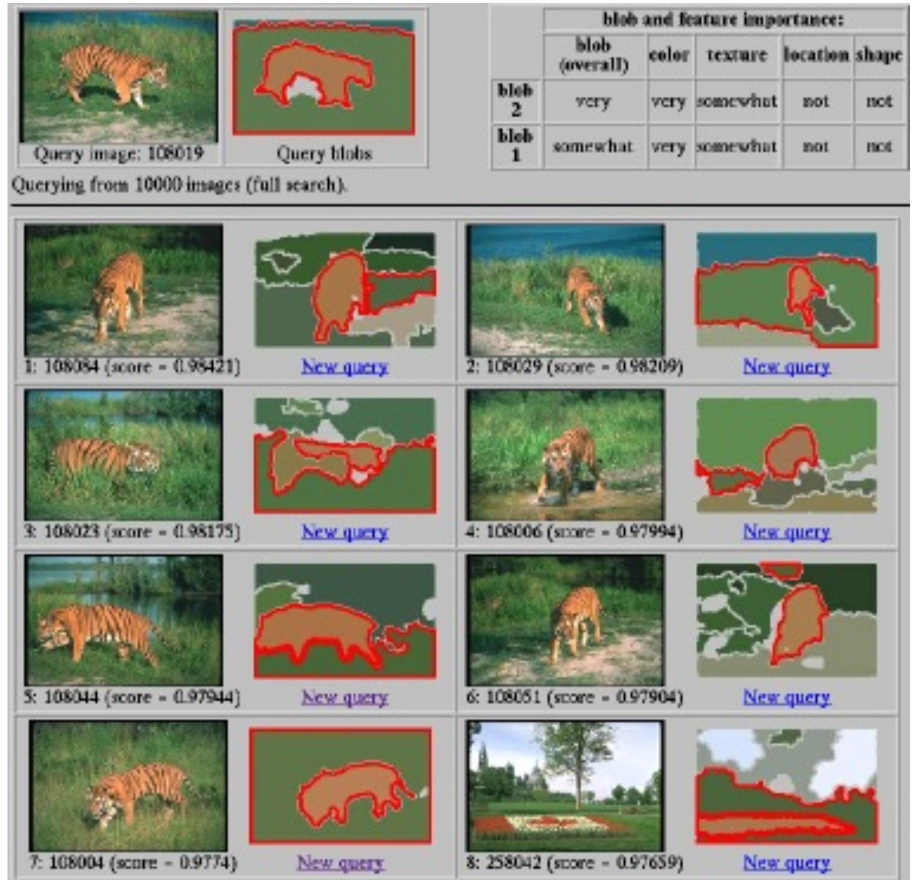


Swain and Ballard, [Color Indexing](), IJCV 1991.

source: Svetlana Lazebnik

Uses of color in computer vision

Image segmentation and retrieval

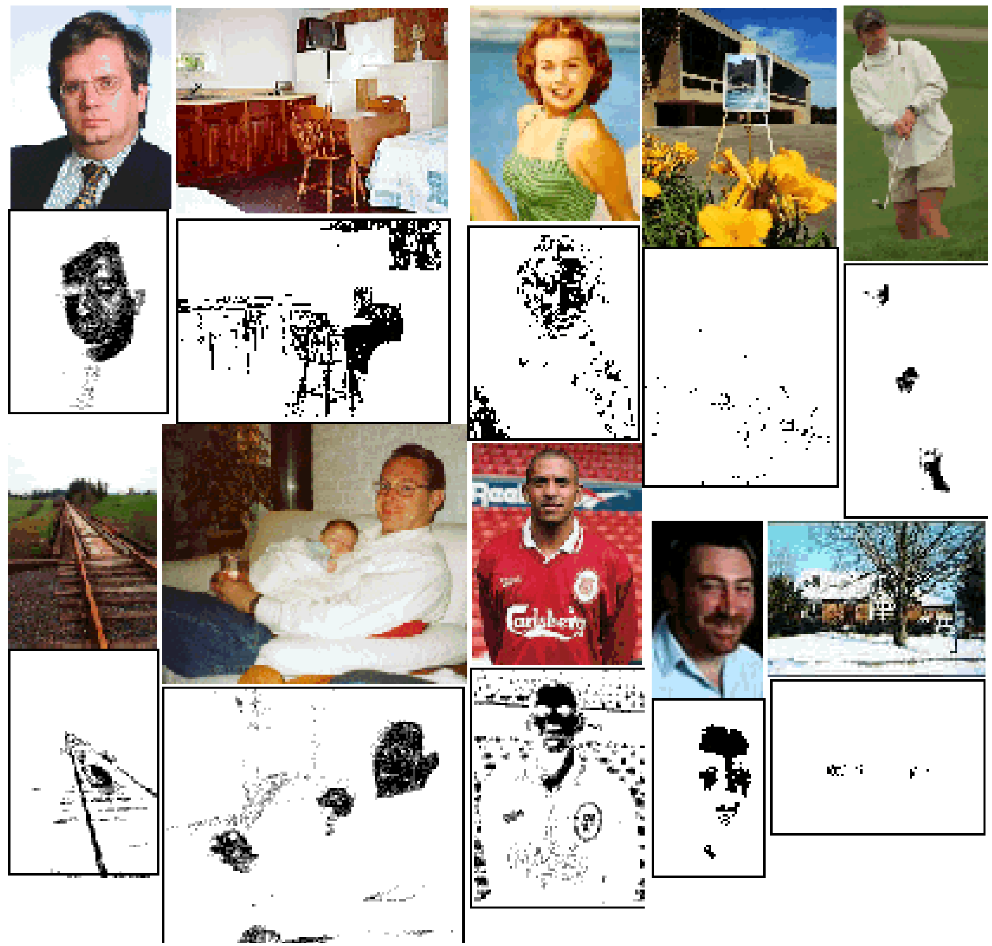


C. Carson, S. Belongie, H. Greenspan, and Ji. Malik, Blobworld: Image segmentation using Expectation-Maximization and its application to image querying, ICVIS 1999.

source: Svetlana Lazebnik

Uses of color in computer vision

Skin detection



M. Jones and J. Rehg, [Statistical Color Models with Application to Skin Detection](), IJCV 2002.

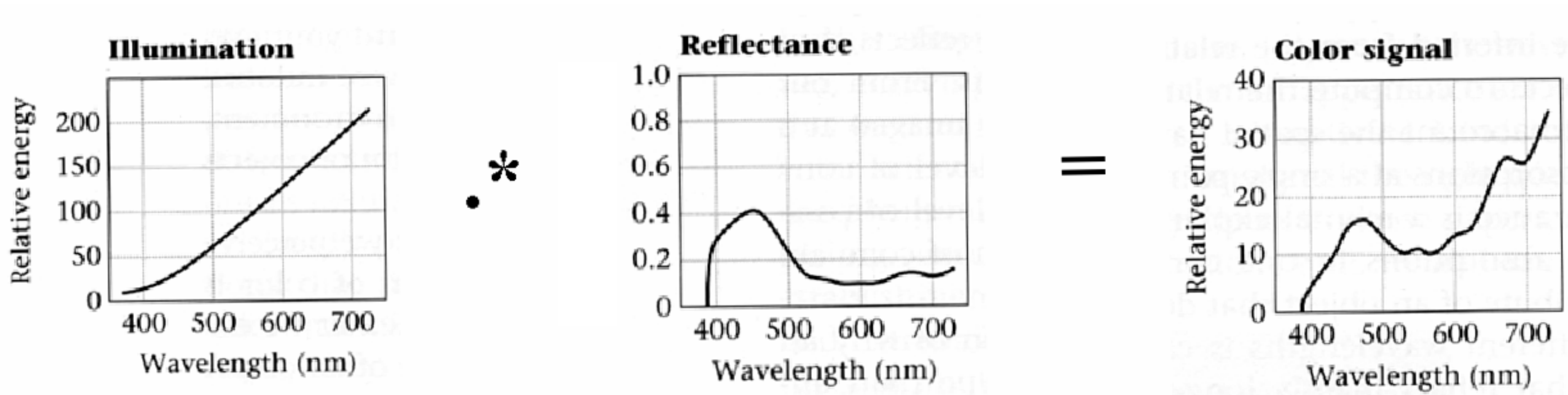
source: Svetlana Lazebnik

Color features

Pros/Cons?

Interaction of light and surfaces

- Reflected color is the result of interaction of light source spectrum with surface reflectance



source: Svetlana Lazebnik

Interaction of light and surfaces

Olafur Eliasson, *Room for one color*, *1997*

"In Olafur Eliasson's *Room for one color* (1997) we actually only see one color. The room is illuminated by yellow monofrequency light. In contrast to ordinary white light containing the full colour spectrum, it consists of a single wavelength at the yellow end of the spectrum. The room appears colorless because the light reduces all the other colours to a scale comprised of yellow. On the other hand, everything stands out with crystal sharpness because there is far less information than usual for the eye to process. After a while, the eye compensates for this deficiency by generating an excess of the missing primaries of red and blue – which together yield violet. So when we move on into the adjacent space we begin by seeing everything as imbued with violet."

source: Svetlana Lazebnik

Shape Features

Invariance

- We might want our features to be robust to variations that can be present in images.

Slide source: Tom Duerig

Types of invariance

- Illumination

Slide source: Tom Duerig

Types of invariance

- Illumination
- Scale

Slide source: Tom Duerig

Types of invariance

- Illumination
- Scale
- Rotation

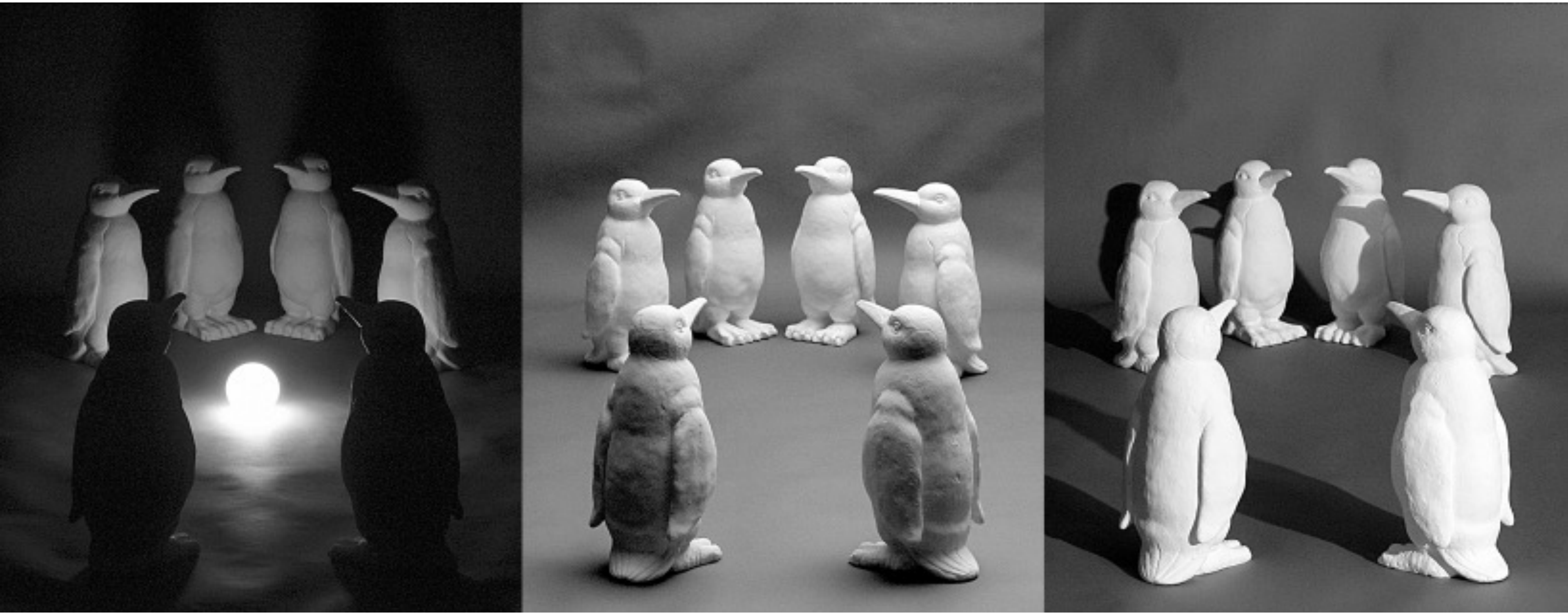
Slide source: Tom Duerig

Types of invariance

- Illumination
- Scale
- Rotation
- Affine

Slide source: Tom Duerig

Challenges: illumination

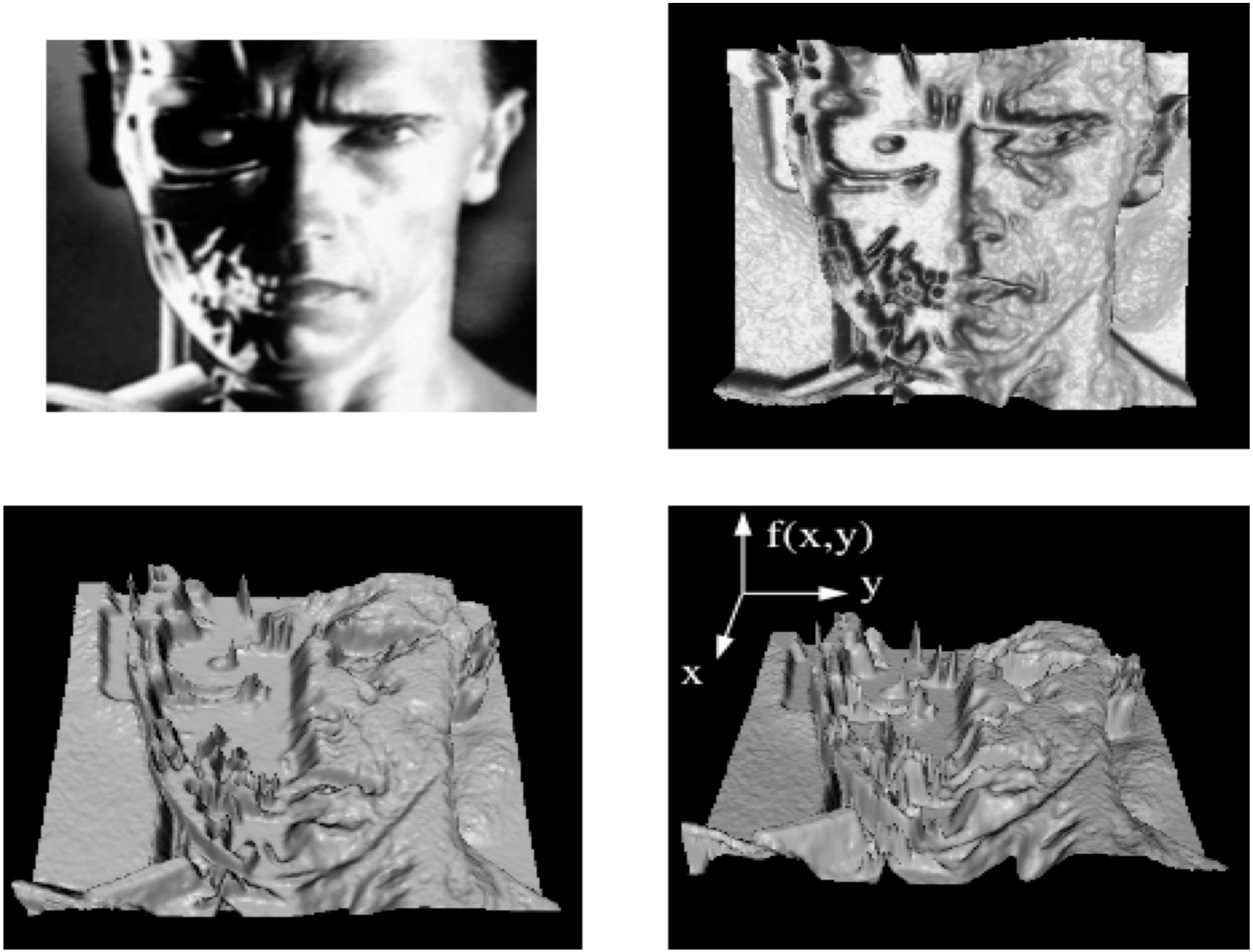


Illumination changes brightness of pixels. What remains unchanged?

image credit: J. Koenderink

Modeling images as functions

Images as functions



Intensity given position

Source: S. Seitz

Images as functions

- We can think of an **image** as a function, *f*, from R^2 to R:
  - $f(x, y)$ gives the **intensity** at position (x, y)

- A color image is just three functions pasted together. We can write this as a "vector-valued" function:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Source: S. Seitz

Motivation 1: Noise reduction

Given a camera and a still scene, how can you reduce noise?

Take lots of images and average them!

What's the next best thing?

Source: S. Seitz

First attempt at a solution

- Let's replace each pixel with an average of all the values in its neighborhood

Source: S. Marschner

Moving Average In 2D

$$F[x, y]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$G[x, y]$$

	0								

Source: S. Seitz

Moving Average In 2D

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10							

Source: S. Seitz

Moving Average In 2D

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20						

Source: S. Seitz

Moving Average In 2D

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30					

Source: S. Seitz

Moving Average In 2D

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30				

Source: S. Seitz

Moving Average In 2D

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Source: S. Seitz

Generalization of moving average (Convolution)

- Let’s replace each pixel with a *weighted* average of its neighborhood
- The weights are called the *filter kernel*
- What are the weights for a 3x3 moving average?

Source: D. Lowe

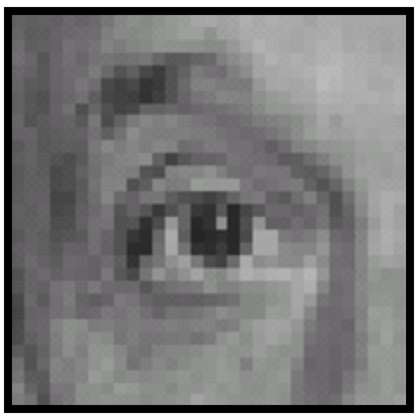
Generalization of moving average (Convolution)

- Let's replace each pixel with a *weighted* average of its neighborhood
- The weights are called the *filter kernel*
- What are the weights for a 3x3 moving average?

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Source: D. Lowe

Practice with linear filters



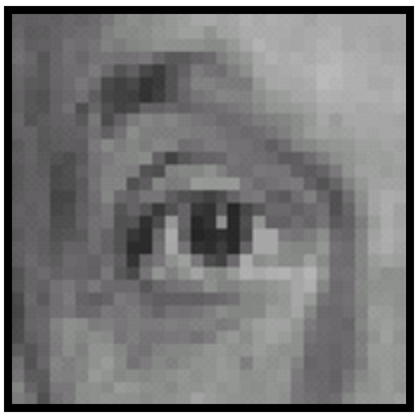
Original

0	0	0
0	1	0
0	0	0

?

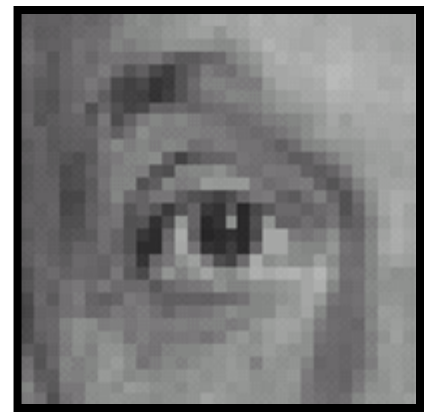
Source: D. Lowe

Practice with linear filters



Original

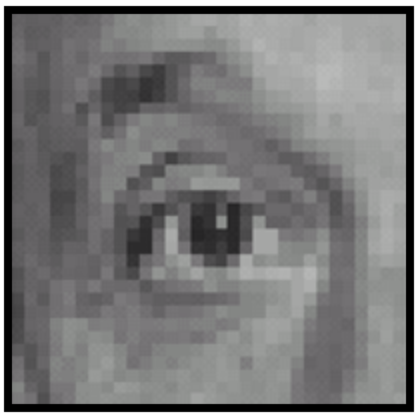
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters



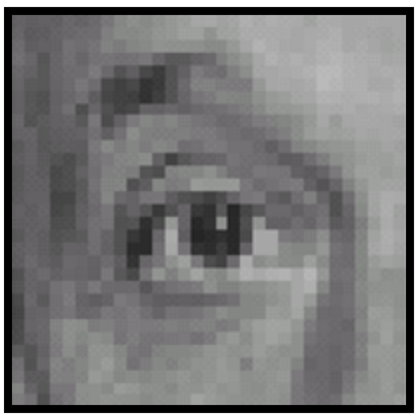
Original

0	0	0
0	0	1
0	0	0

?

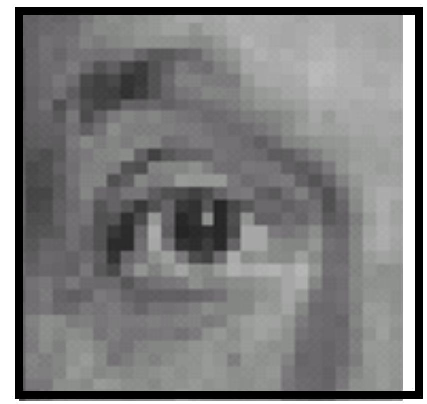
Source: D. Lowe

Practice with linear filters



Original

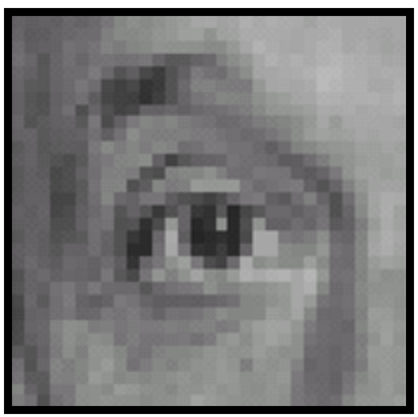
0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Practice with linear filters



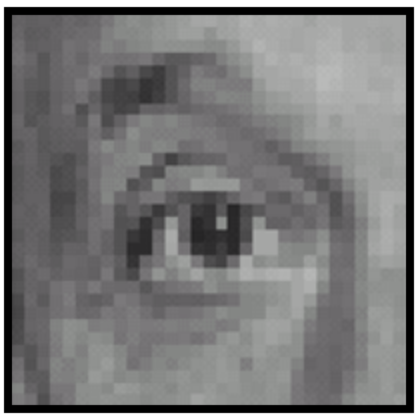
Original

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

?

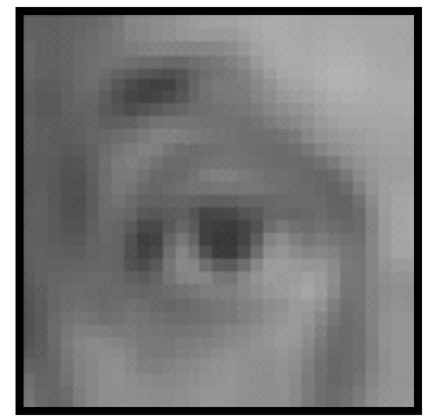
Source: D. Lowe

Practice with linear filters



Original

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

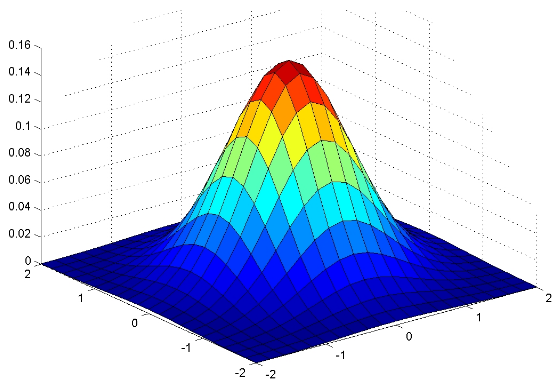


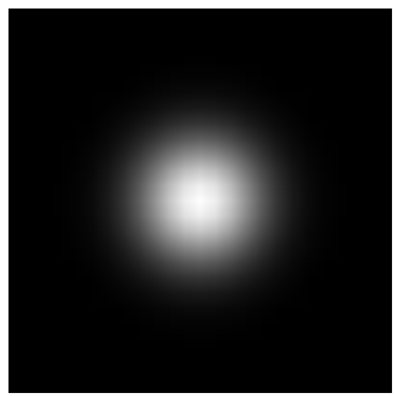
Blur (with a box filter)

Source: D. Lowe

Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$





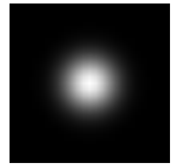
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

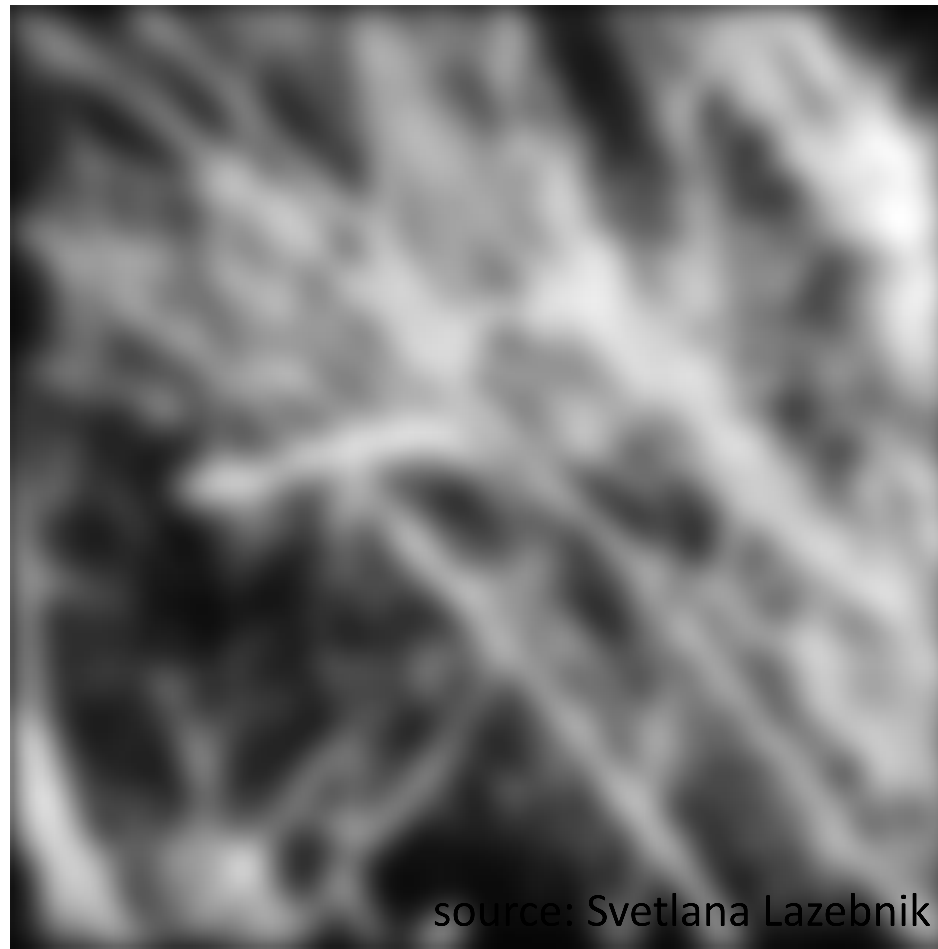
5 x 5, σ = 1

- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case)

Source: C. Rasmussen

Example: Smoothing with a Gaussian





source: Svetlana Lazebnik

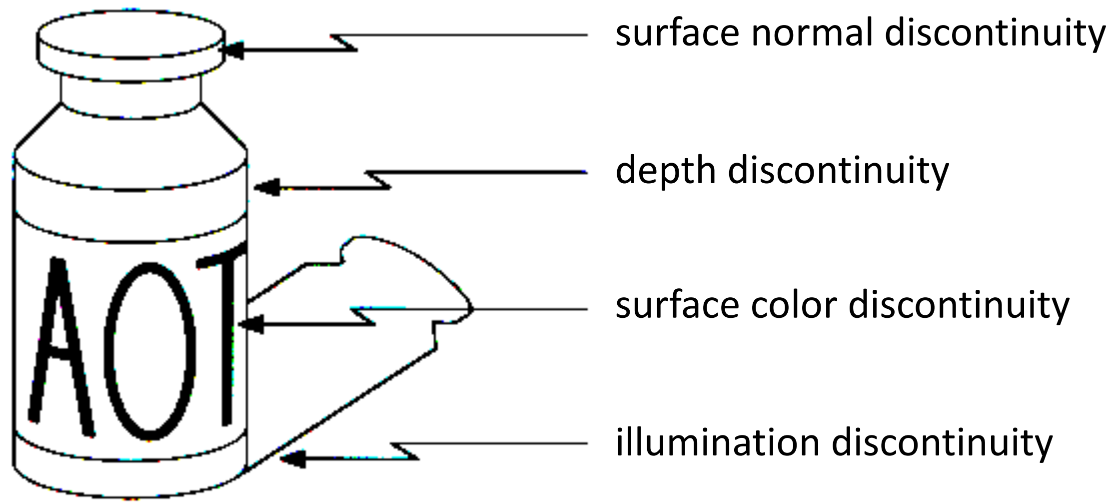
Edges

Edge detection

- **Goal:** Identify sudden changes (discontinuities) in an image
    - Intuitively, most semantic and shape information from the image can be encoded in the edges
    - More compact than pixels
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

Source: D. Lowe

Origin of Edges

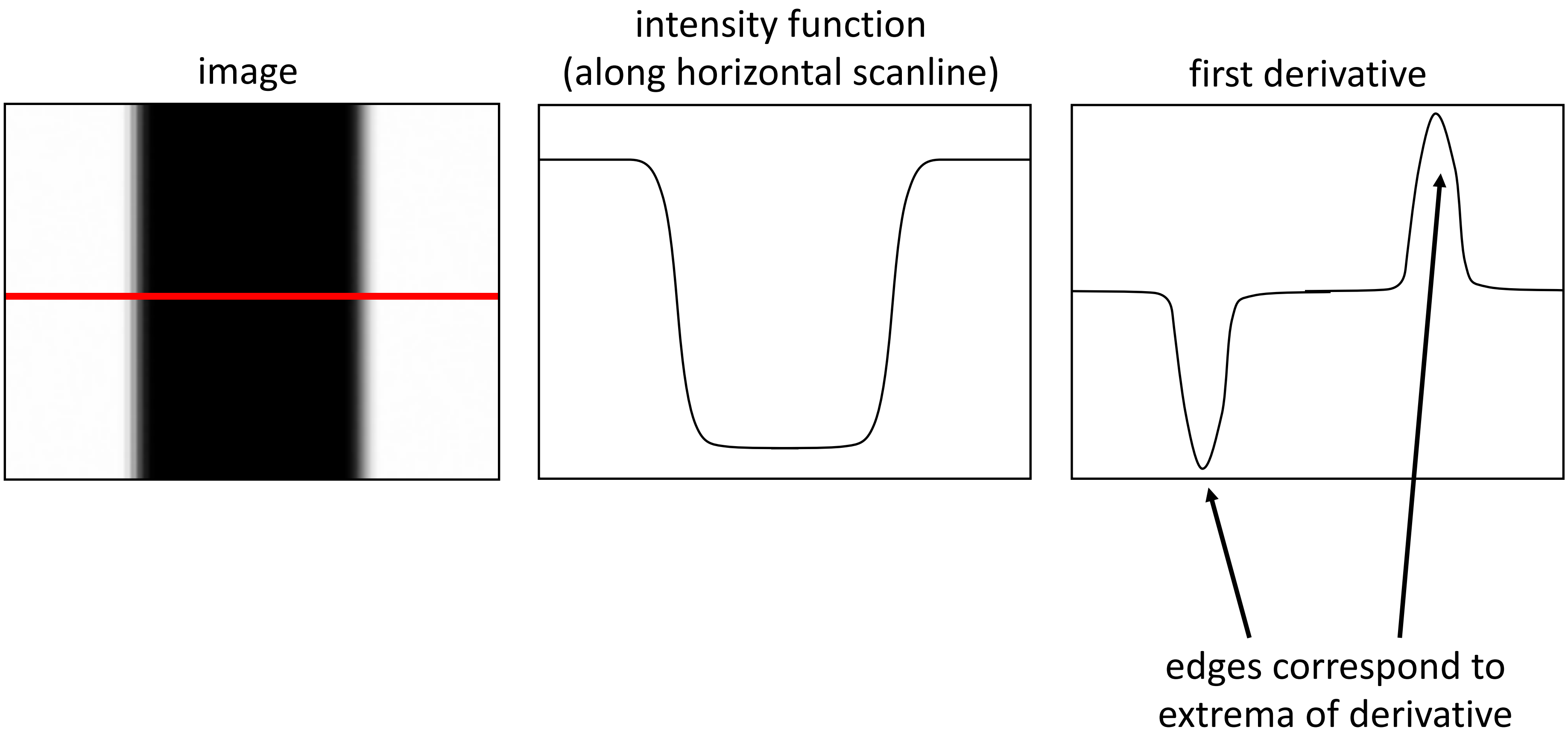


Edges are caused by a variety of factors

Source: Steve Seitz

Characterizing edges

- An edge is a place of rapid change in the image intensity function



source: Svetlana Lazebnik

Edge filters

Approximations of derivative filters:

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

Convolve filter with image to get edge map

Source: K. Grauman

Edge filters

Approximations of derivative filters:

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

Source: K. Grauman

Edge filters

Approximations of derivative filters:

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

Respond highly to vertical edges

Source: K. Grauman

Edge filters

Approximations of derivative filters:

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

Source: K. Grauman

Edge filters

Approximations of derivative filters:

Prewitt: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$

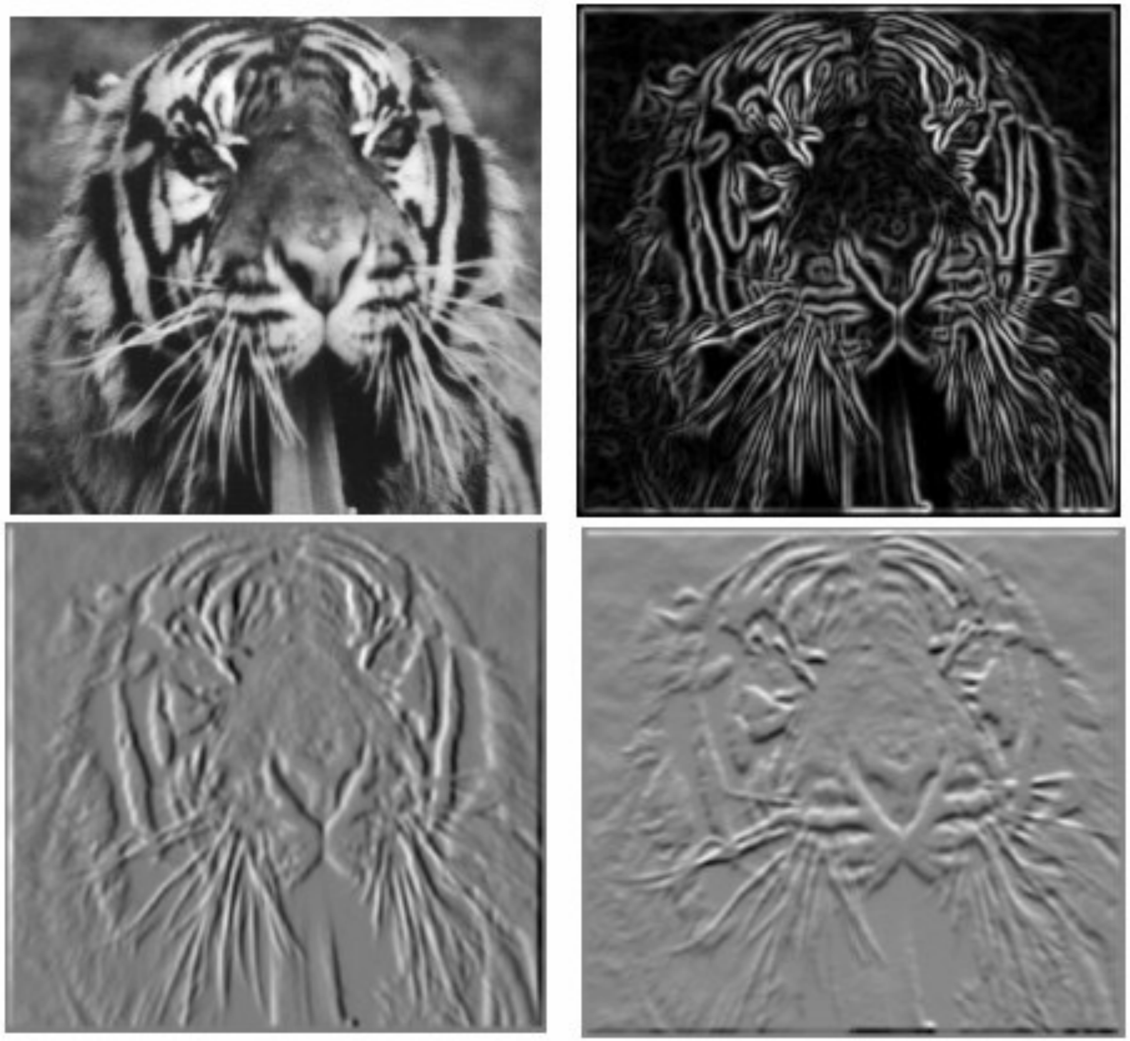
Sobel: $M_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$

Roberts: $M_x = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$; $M_y = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$

Respond highly to horizontal edges

Source: K. Grauman

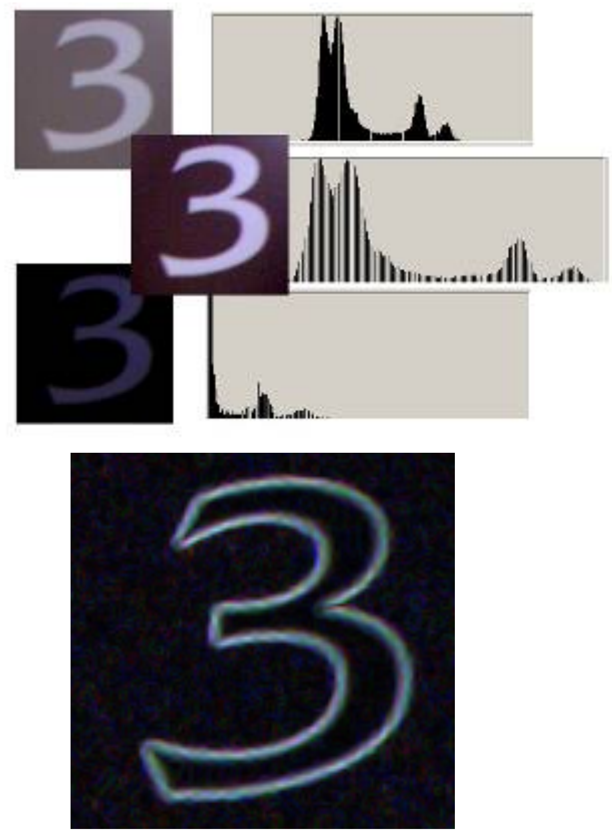
Edges: example



source: Svetlana Lazebnik

How to achieve illumination invariance

- Use edges instead of raw values



Slide source: Tom Duerig

Feature types!

The representation of these two umbrella’s should be similar….
Under a color based representation they look completely different!

Edges

Red umbrella

Gray umbrella

Edges extracted using convolution with Prewitt filter

Edges

Edges overlaid from red and gray umbrellas.

Local Features

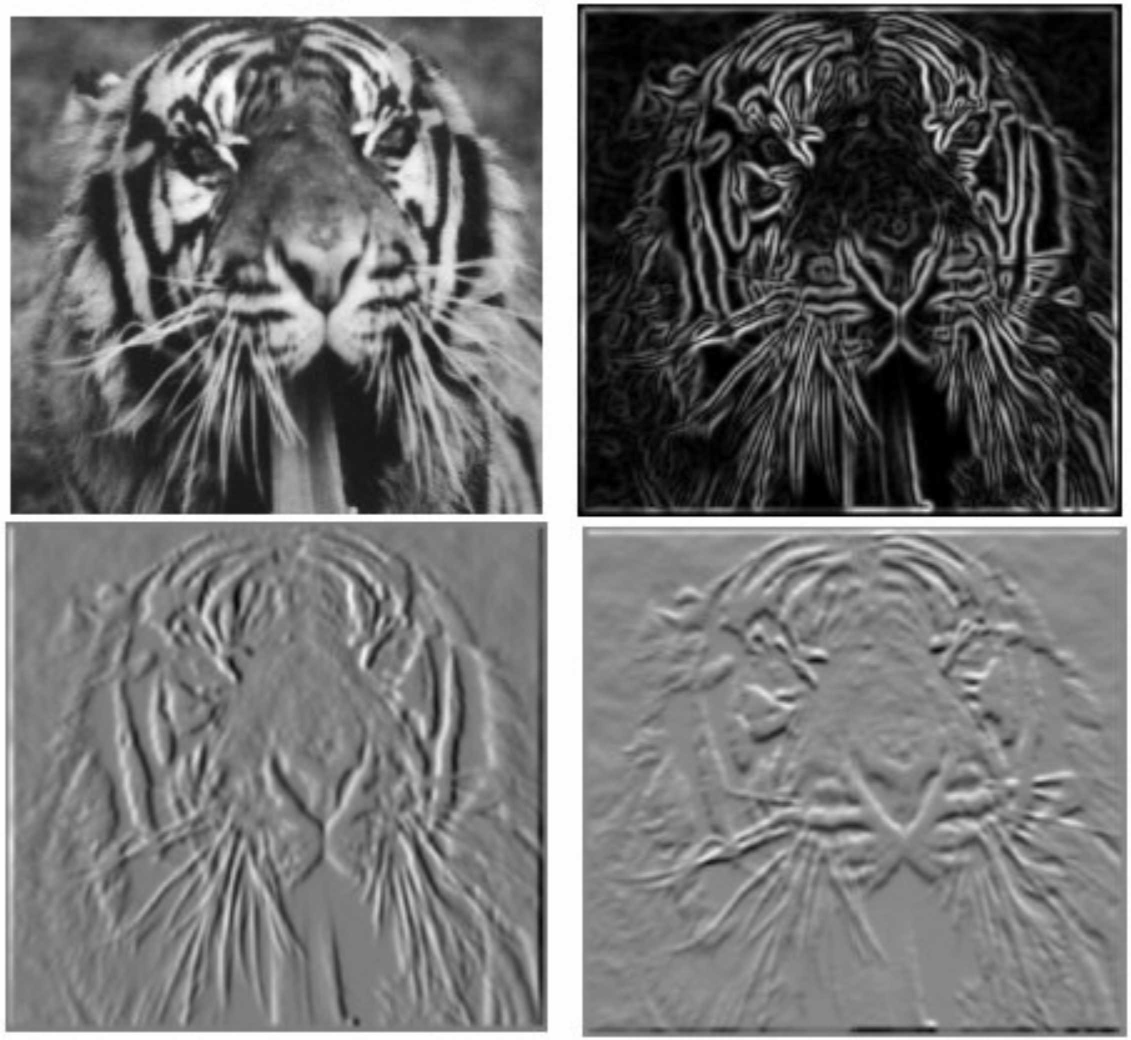
Feature points (locations) + feature descriptors:

1) Where should we put the features?

2) How should we describe the features

Where should we put features?

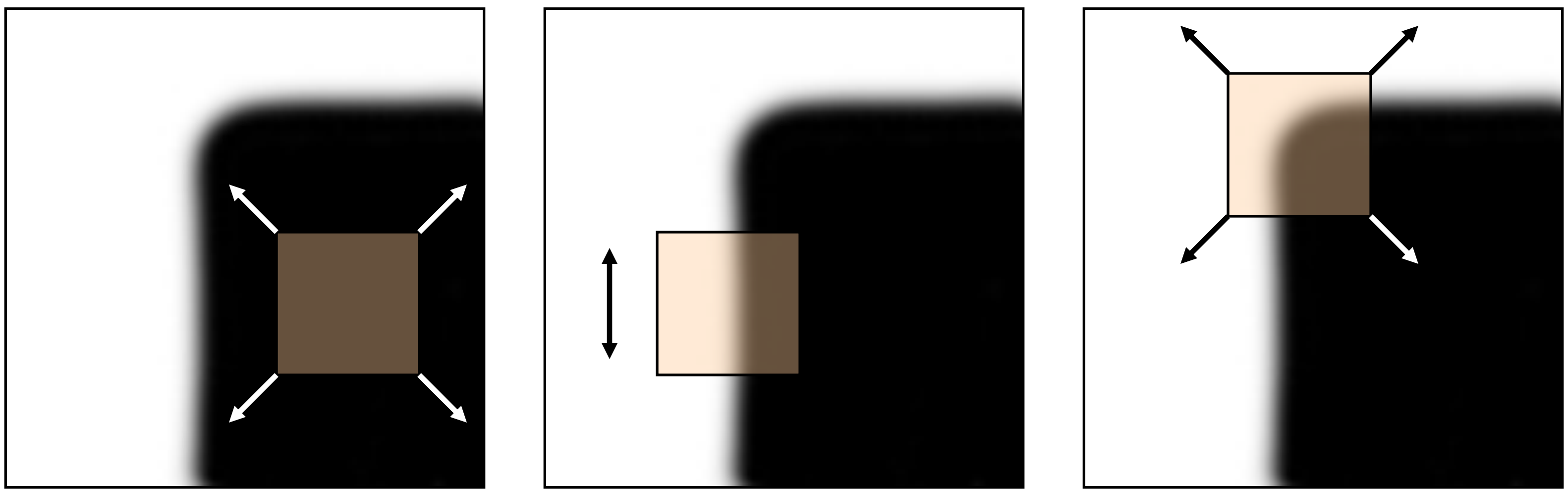
Where to put features?



source: Svetlana Lazebnik

Distinctiveness in x,y

- We should easily recognize the point 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

Source: A. Efros

Corner Detections

C.Harris and M.Stephens. "A Combined Corner and Edge Detector."
Proceedings of the 4th Alvey Vision Conference: pages 147--151.

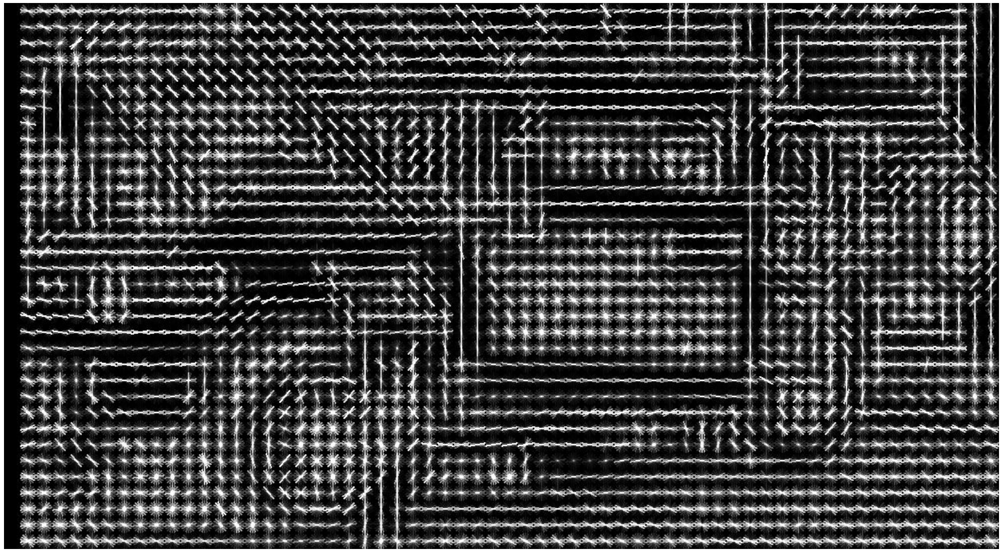
source: Svetlana Lazebnik

Toward denser sampling

Just sample points randomly on edges

Fully dense sampling

Or sample everywhere!



How should we describe features?

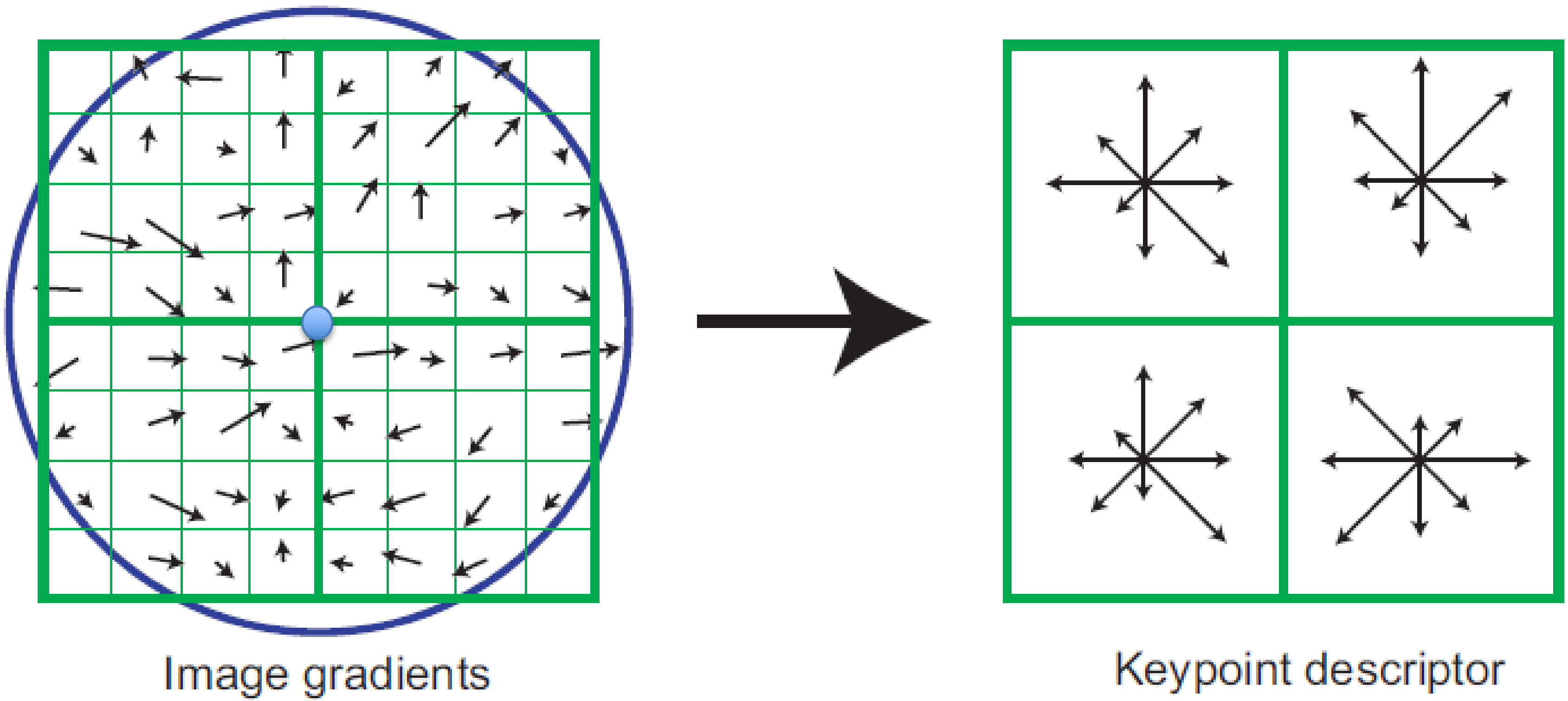
SIFT

- SIFT (Scale Invariant Feature Transform) - stable robust and distinctive local features
- One of the most popular shape (edge) based features

Where to put them:

- Originally sampled points distinctive in both position (x,y) and scale (SI – Scale Invariant)
- Often now used with sampling on a dense grid

Feature descriptor



Feature descriptor

- Based on 16*16 patches
- 4*4 subregions
- 8 orientation bins in each subregion
- 4*4*8=128 dimensions in total

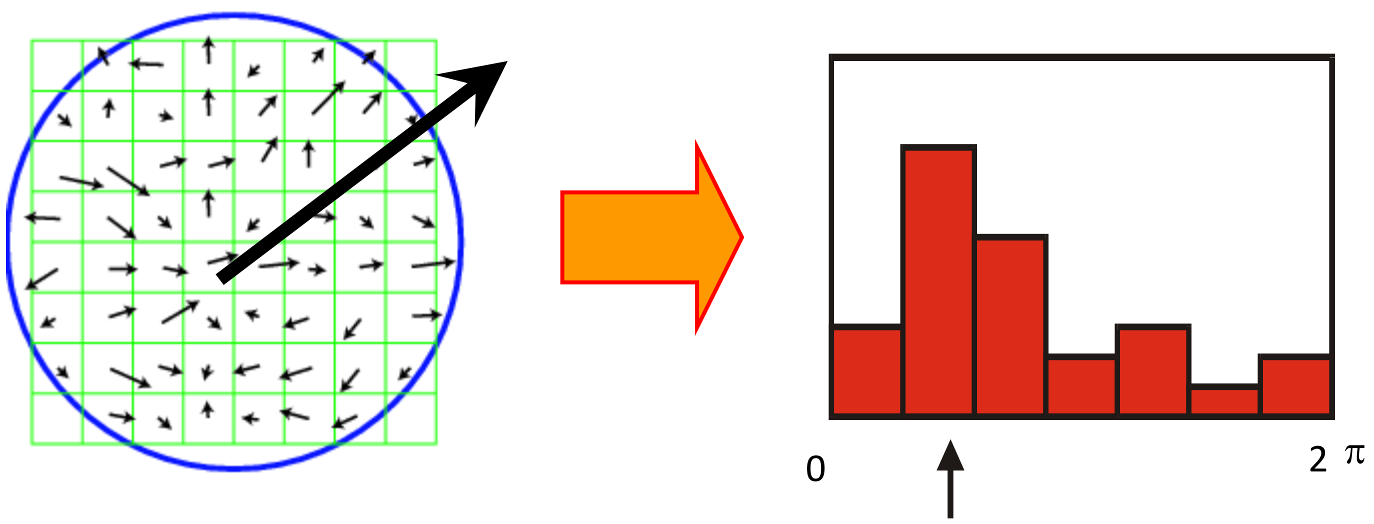
Rotation Invariance

- Rotate all features to go the same way in a determined manner
- A histogram is formed by quantizing the orientations into bins
- Features are rotated according to peak of orientation histogram

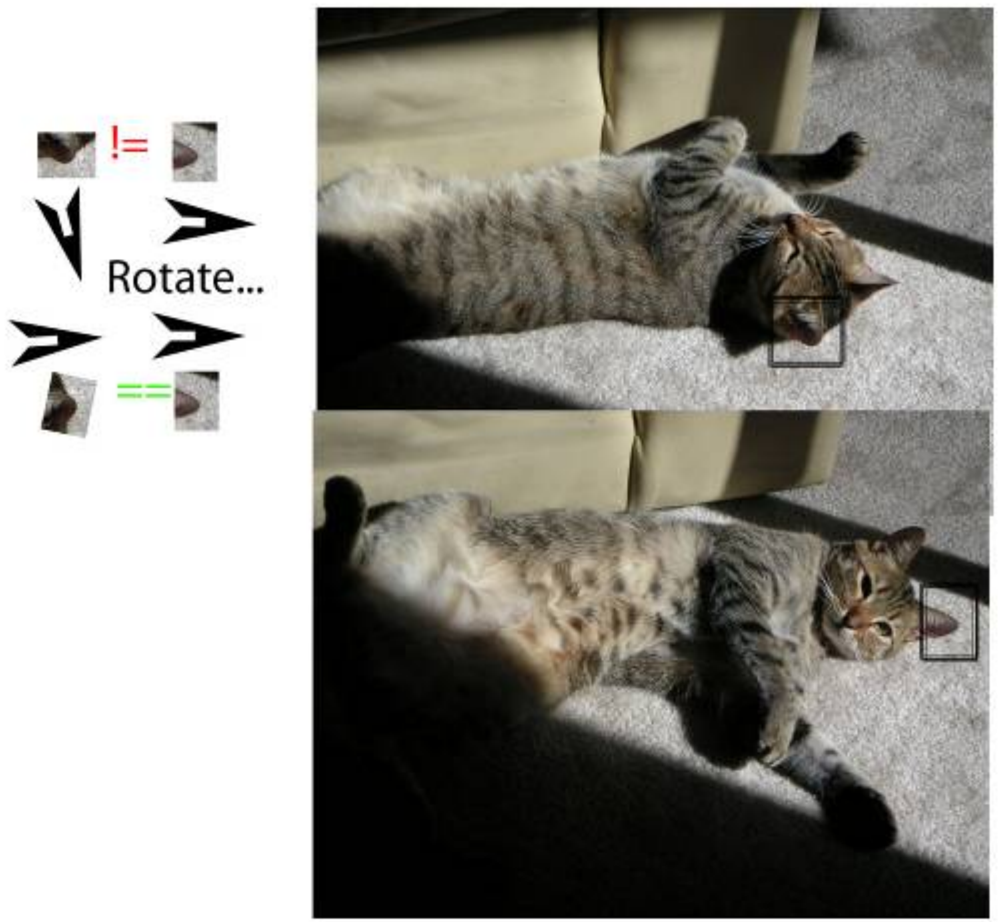
Eliminating rotation ambiguity

To assign a unique orientation:

- Create histogram of local gradient directions in the patch
- Assign canonical orientation at peak of smoothed histogram

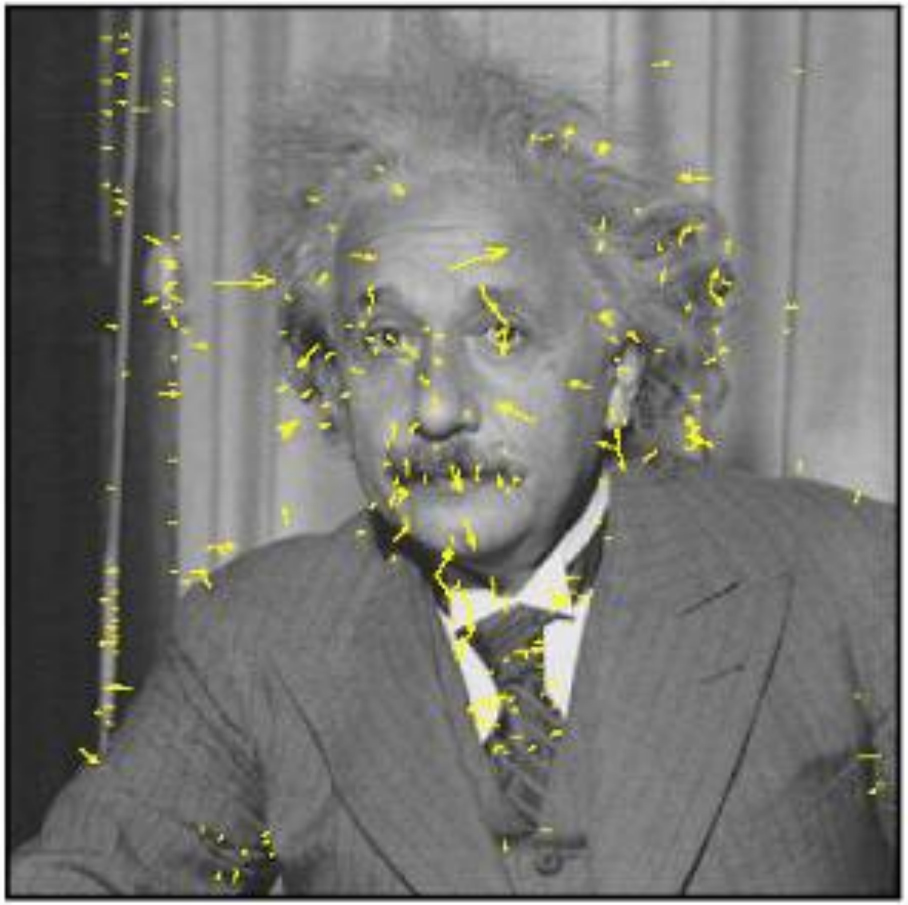


Rotation Invariance



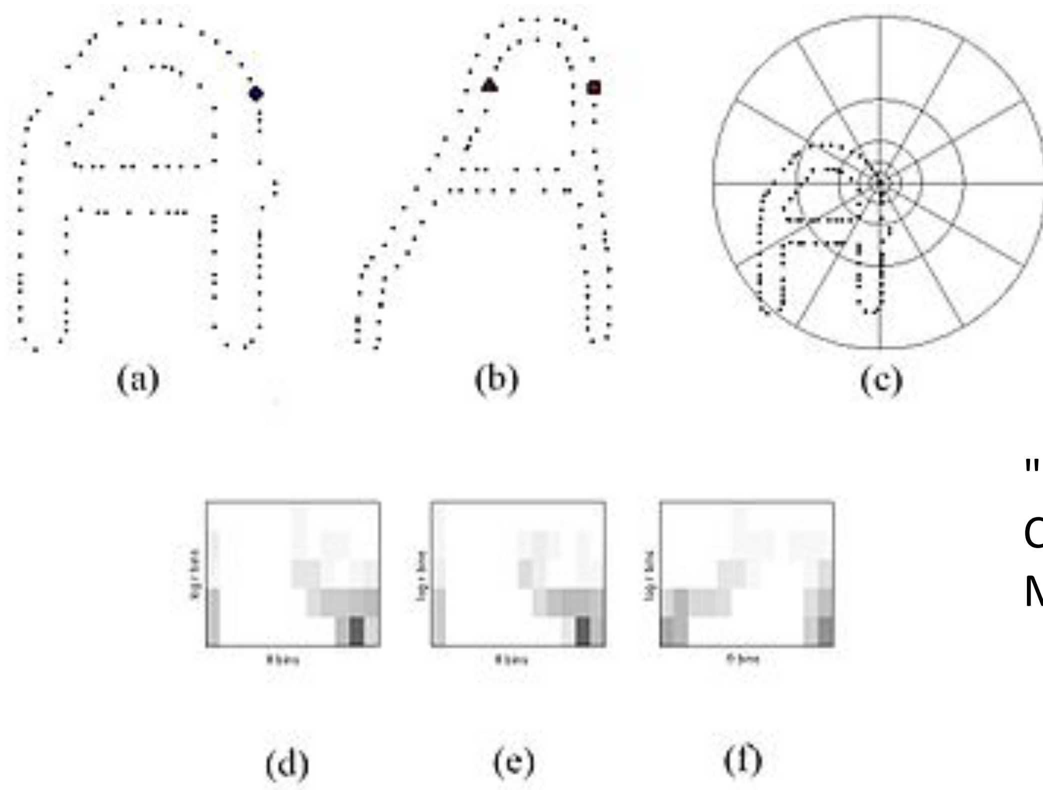
Slide source: Tom Duerig

SIFT output



Other shape features

Shape Context

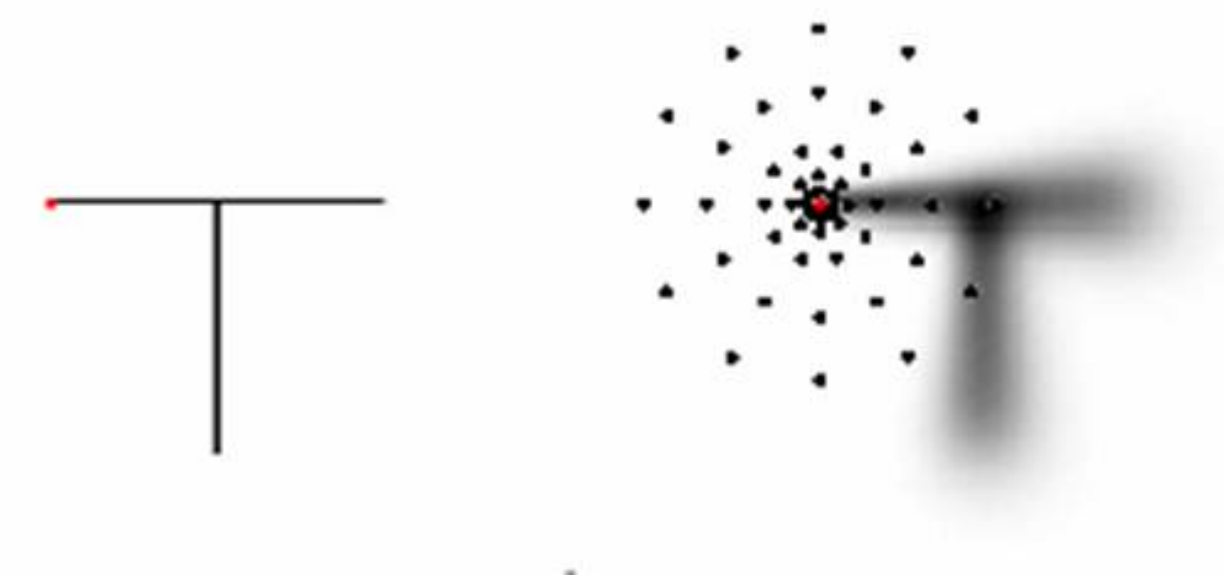


"Matching with Shape Contexts": Belongie, Malik 2000

- (a) and (b) are the sampled edge points of the two shapes. (c) is the diagram of the log-polar bins used to compute the shape context. (d) is the shape context for the circle, (e) is that for the diamond, and (f) is that for the triangle. As can be seen, since (d) and (e) are the shape contexts for two closely related points, they are quite similar, while the shape context in (f) is very different.

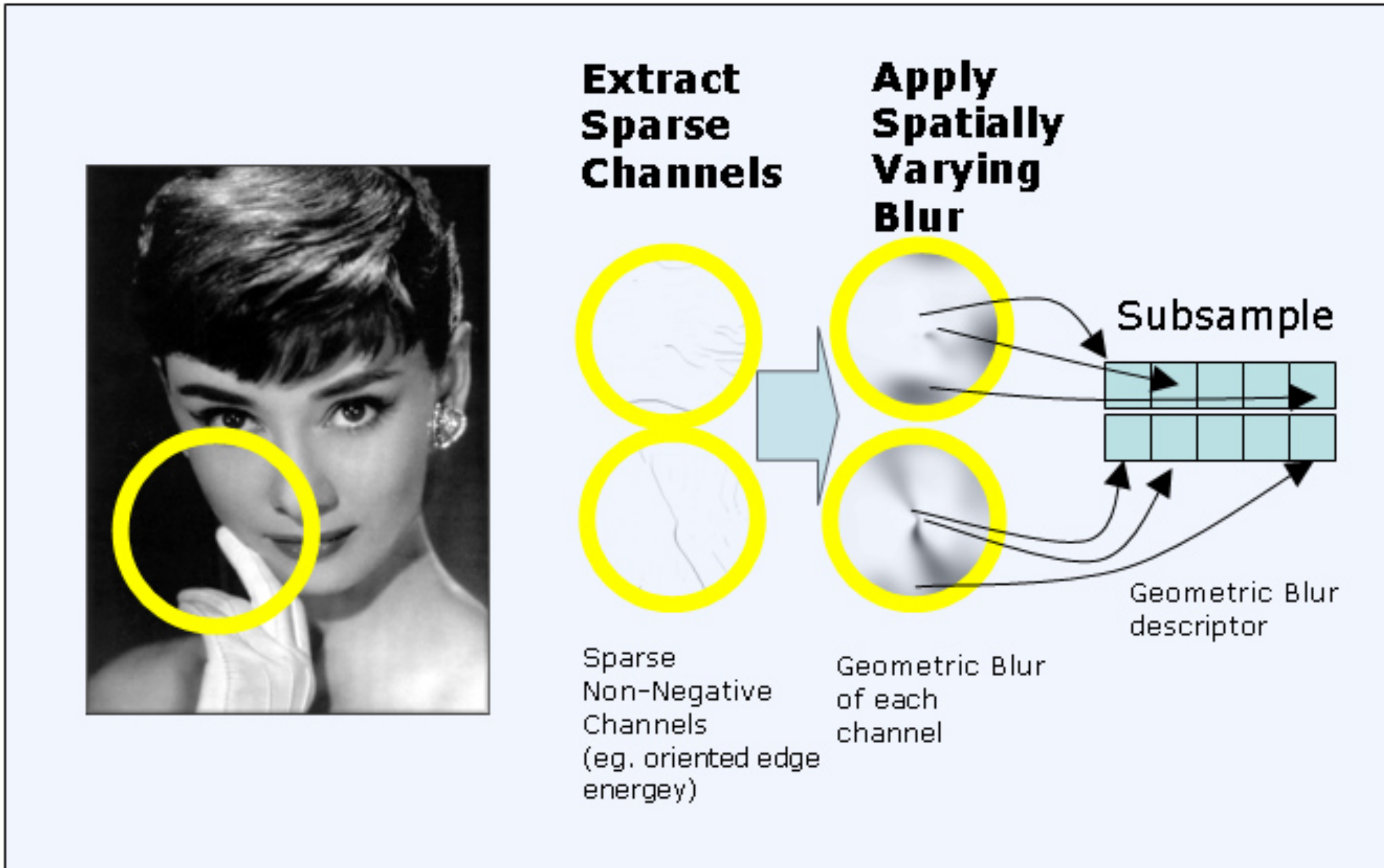
Geometric Blur

"Geometric Blur for Template Matching" A. Berg & Malik, 2001



Geometric Blur

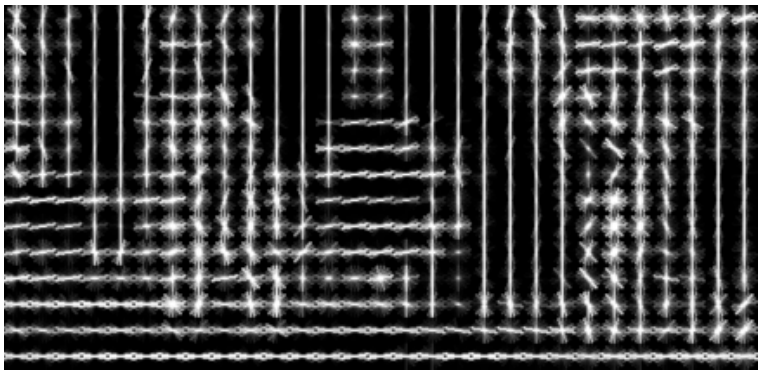
"Geometric Blur for Template Matching" A. Berg & Malik, 2001



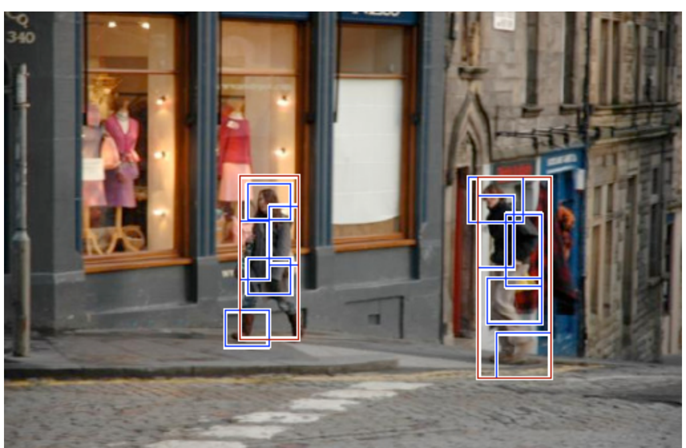
Histograms of Oriented Gradients

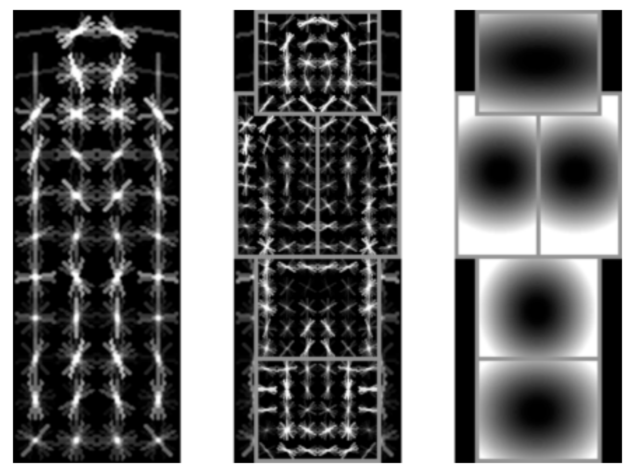
Navneet Dalal and Bill Triggs, 2005

HOG feature map



- Similar to SIFT
- Computed on dense grid of uniformly spaced cells
- Good exploration of parameter space (gradient scale, orientation binning, spatial binning, contrast normalization)



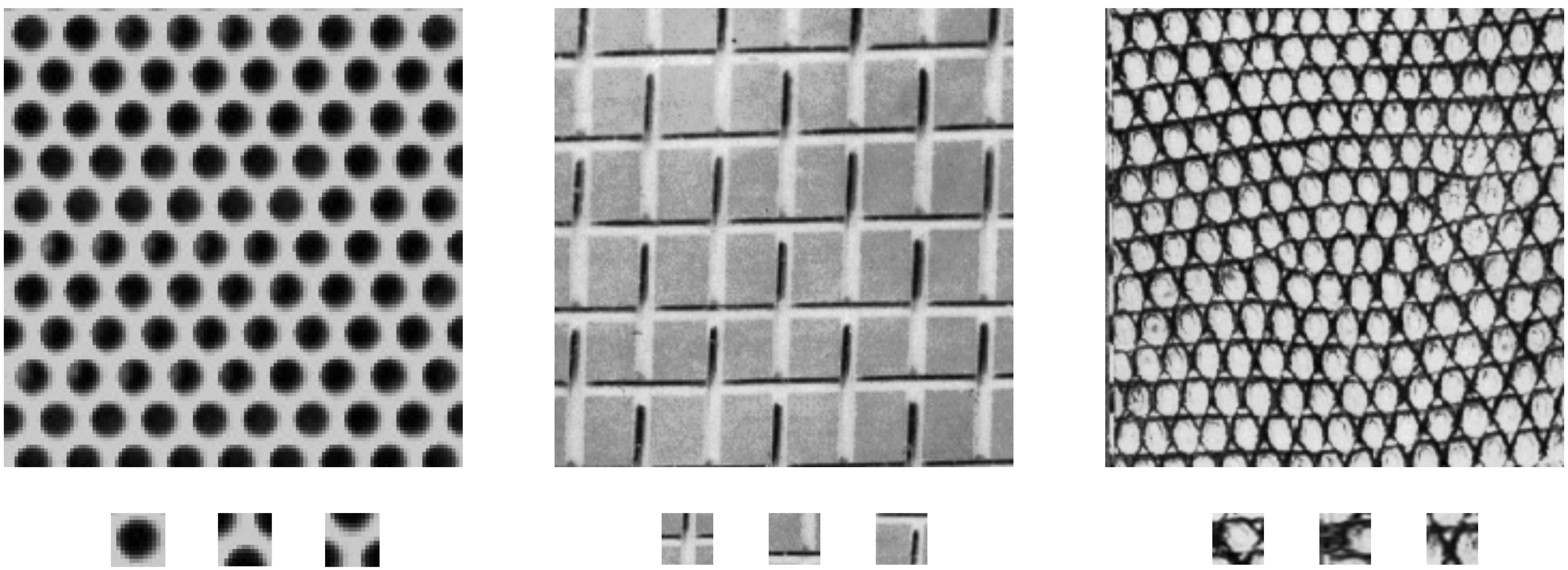


Root filter   Part filters   Deformation weights

Texture Features

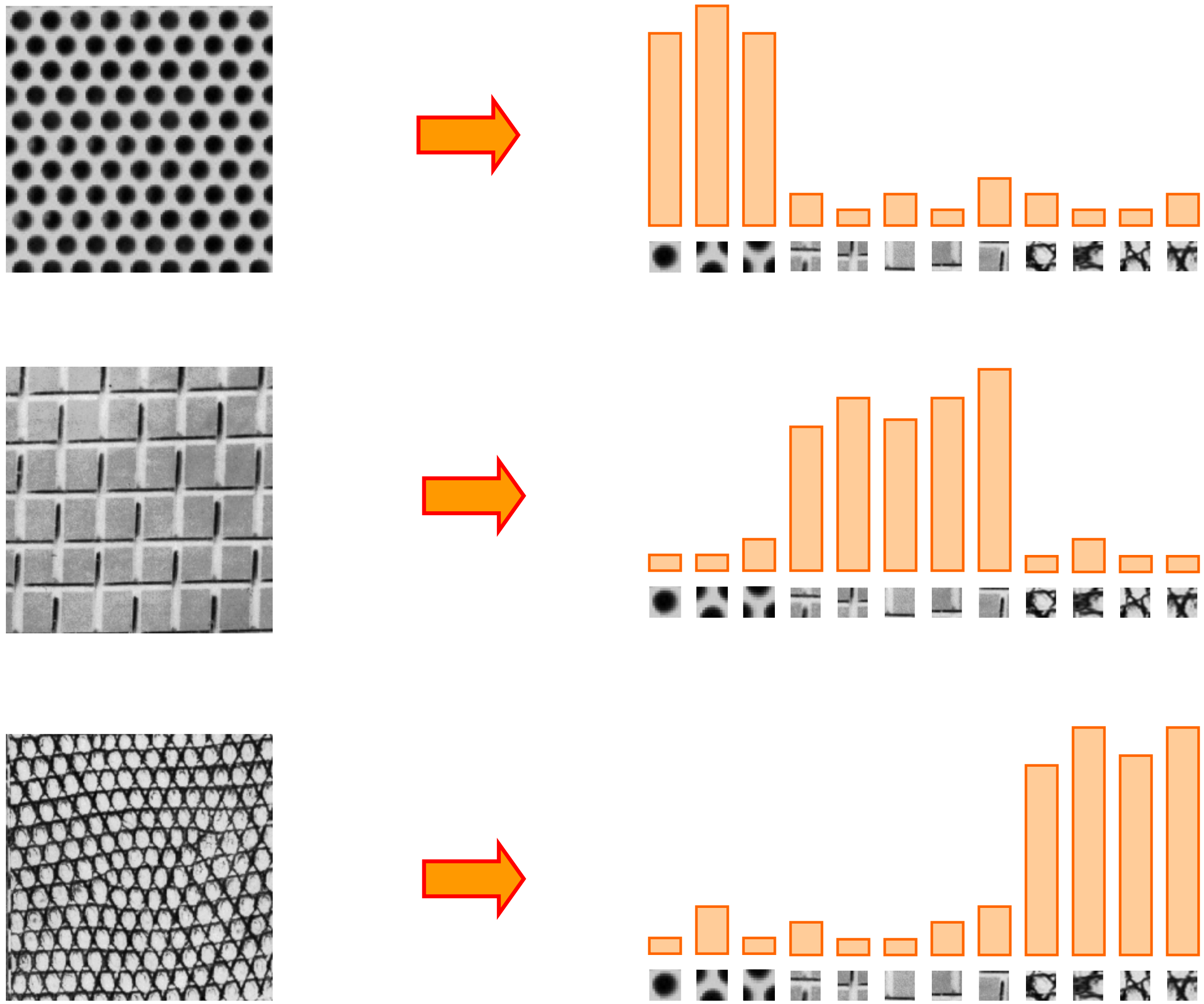
Texture Features

- Texture is characterized by the repetition of basic elements or *textons*
- Generally, it is the identity of the textons, not their spatial arrangement, that matters



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

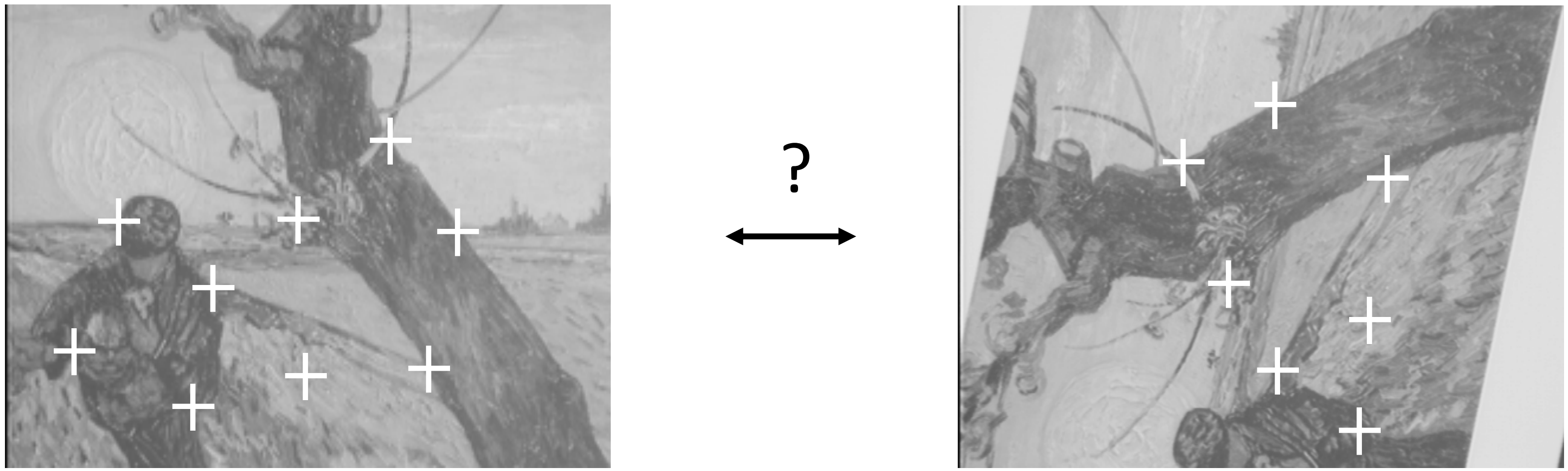
Texture Histograms



Can be computed locally or globally

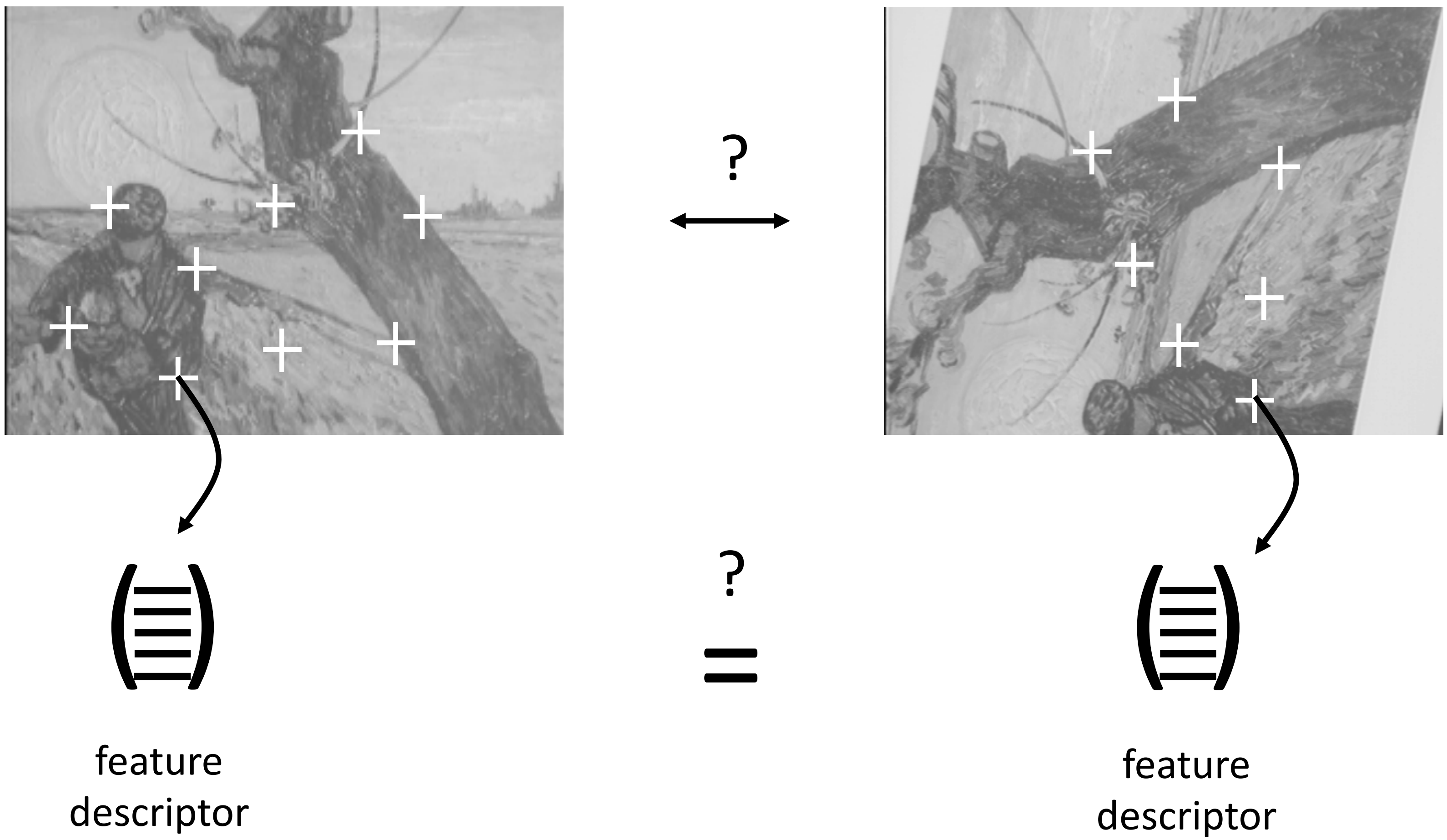
What can we do with features?

Finding correspondences



source: Svetlana Lazebnik

Finding correspondences



- Need to compare *feature descriptors* of local patches surrounding interest points

source: Svetlana Lazebnik

Comparing Feature descriptors

How to compare two such vectors?

- Sum of squared differences (SSD)

$$\mathrm{SSD}(u,v) = \sum_i \left(u_i - v_i\right)^2$$

  - Not invariant to intensity change

- Normalized correlation

$$\rho(u,v) = \frac{\sum_i (u_i - \overline{u})(v_i - \overline{v})}{\sqrt{\left(\sum_j (u_j - \overline{u})^2\right)\left(\sum_j (v_j - \overline{v})^2\right)}}$$

  - Invariant to intensity change

source: Svetlana Lazebnik

Example: object recognition

- The SIFT features of training images are extracted and stored
- For a query image

1. Extract SIFT feature
2. Find nearest neighbor match
3. Given 3 keypoint matches, perform geometric verification


ROAD ATLAS
CAA

Useful Code

VLFEAT (http://www.vlfeat.org/index.html)
open source library implementing various feature descriptors, clustering, matching algorithms

Some fun applications of features + matching

NN for estimating location

J. Hays and A. Efros, IM2GPS: estimating geographic information from a single image, CVPR 2008

Where?



What can you say about where these photos were taken?

How?

Collect a large collection of geo-tagged photos

6.5 million images with both GPS coordinates and geographic keywords, removing images with keywords like birthday, concert, abstract, …

Test set – 400 randomly sampled images from this collection. Manually removed abstract photos and photos with recognizable people – 237 test photos.

Nearest Neighbor Matching

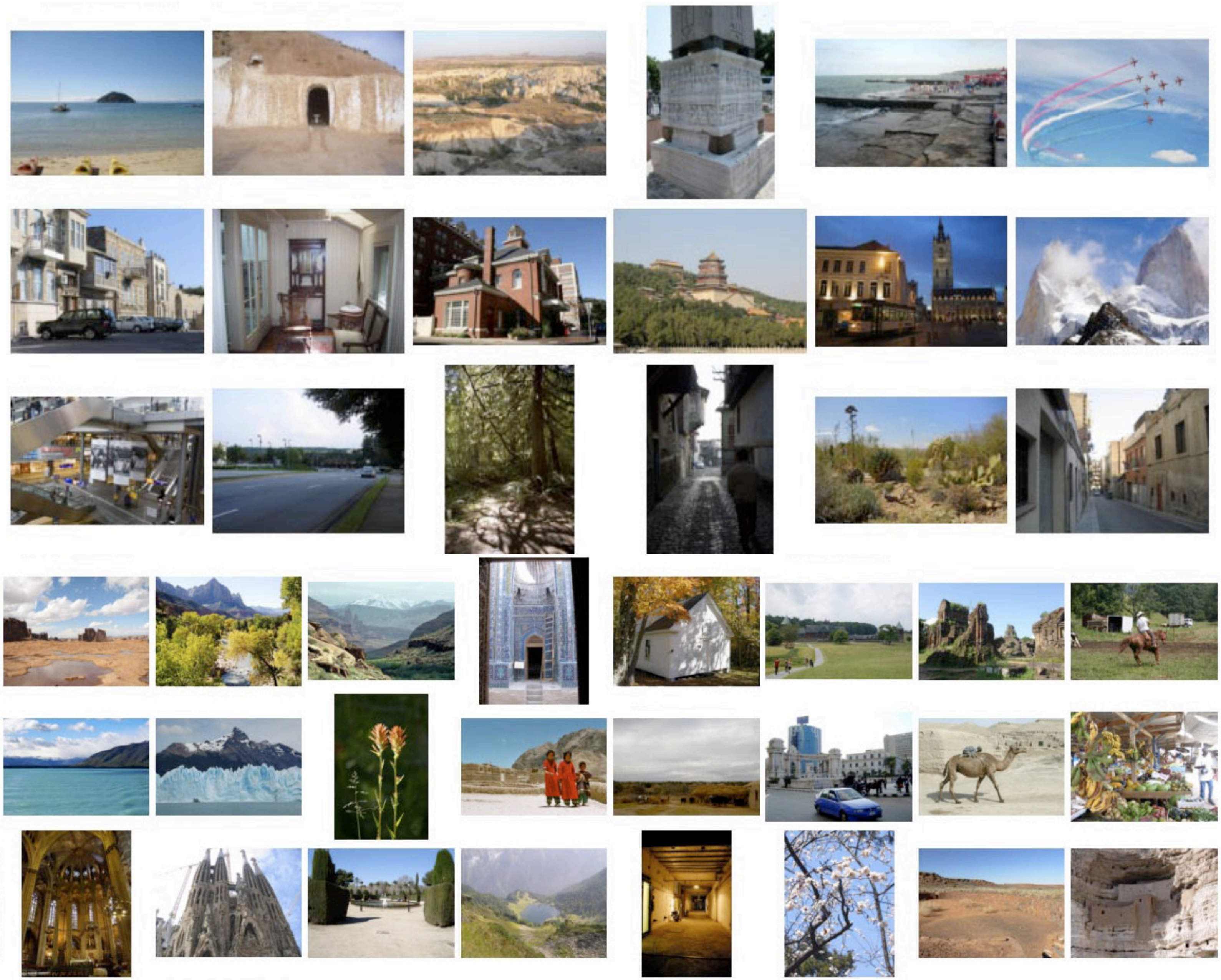
For each input image compute features (color, texture, shape)

Compute distance in feature space to all 6 million images in the database (each feature contributes equally).

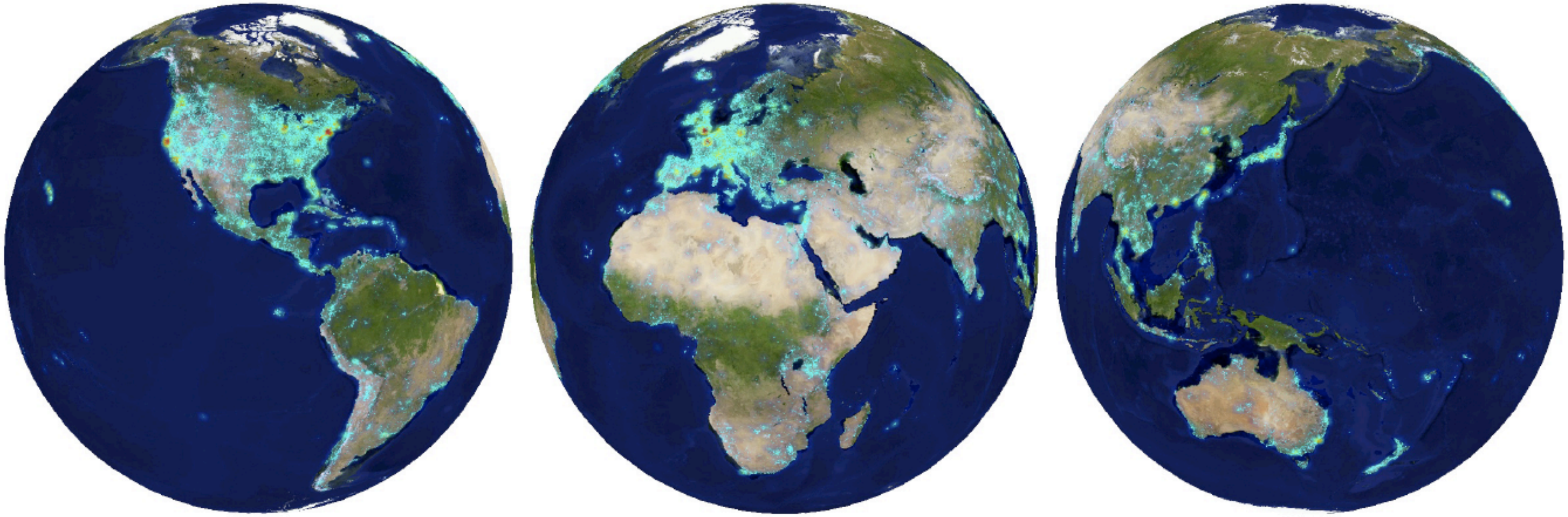
Label the image with GPS coordinates of:

- 1 nearest neighbor
- k=120 nearest neighbors – probability map over entire globe.

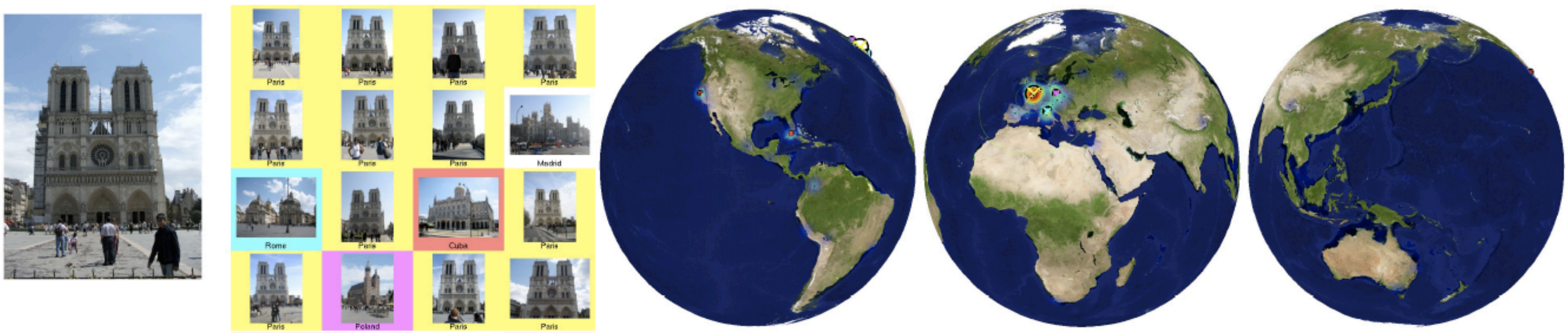
Test Images

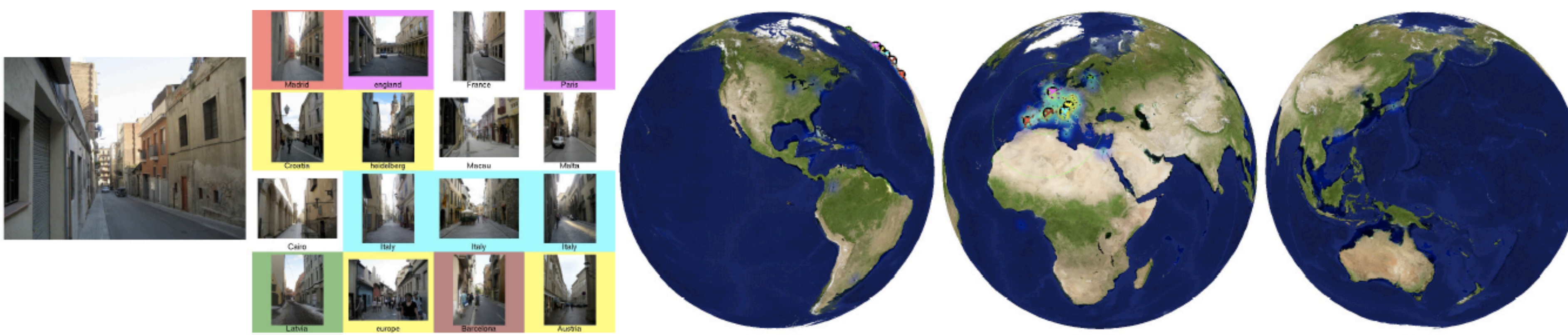


Distribution of photos



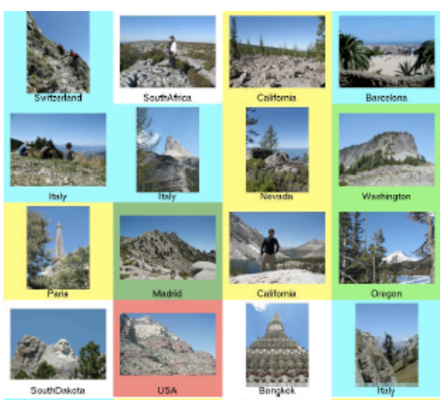
Results

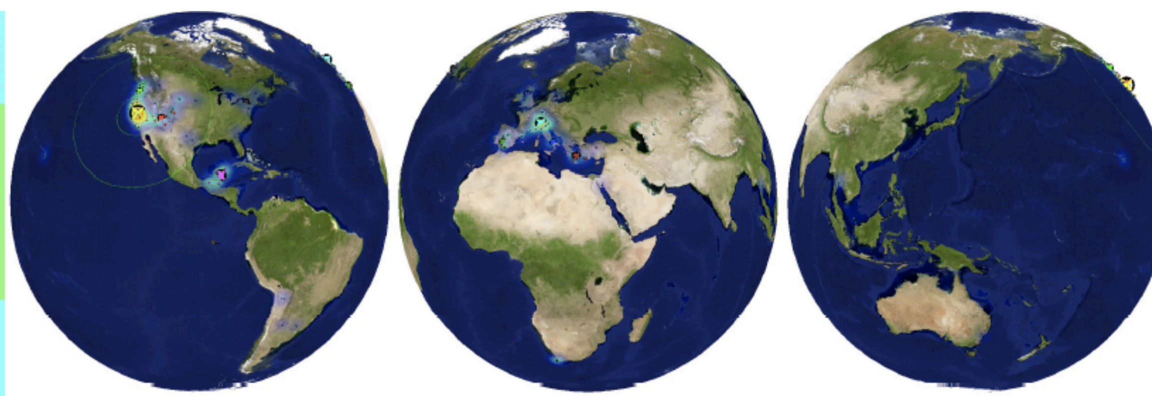

Paris
Paris
Paris
Paris
Paris
Paris
Paris
Madrid
Rome
Paris
Cuba
Paris
Paris
Poland
Paris
Paris


Madrid
england
France
Paris
Croatia
heidelberg
Macau
Malta
Cairo
Italy
Italy
Italy
Latvia
europe
Barcelona
Austria

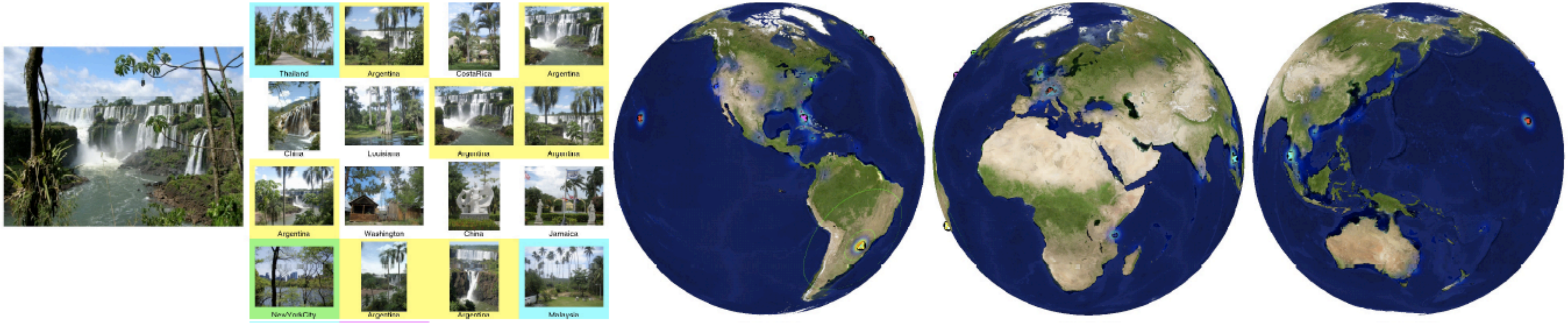
Results

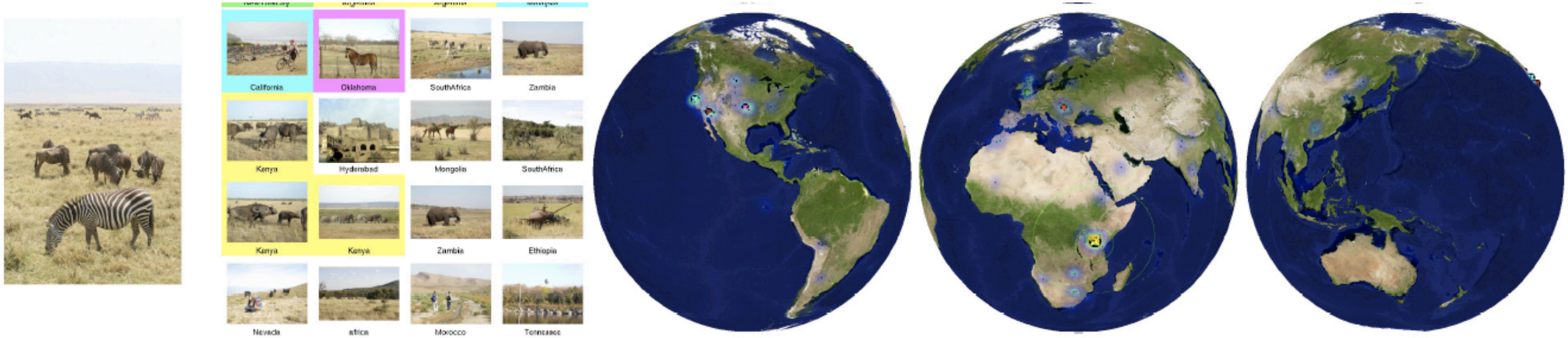

Philippines
Houston
Thailand
Houston
Maldives
Philippines
NewZealand
Bermuda
Palau
Mexico2
Brazil
Mendoza
Thailand
Arkansas


Switzerland
SouthAfrica
California
Barcelona
Italy
Italy
Nevada
Washington
Paris
Madrid
California
Oregon
SouthDakota
USA
Bangkok
Italy

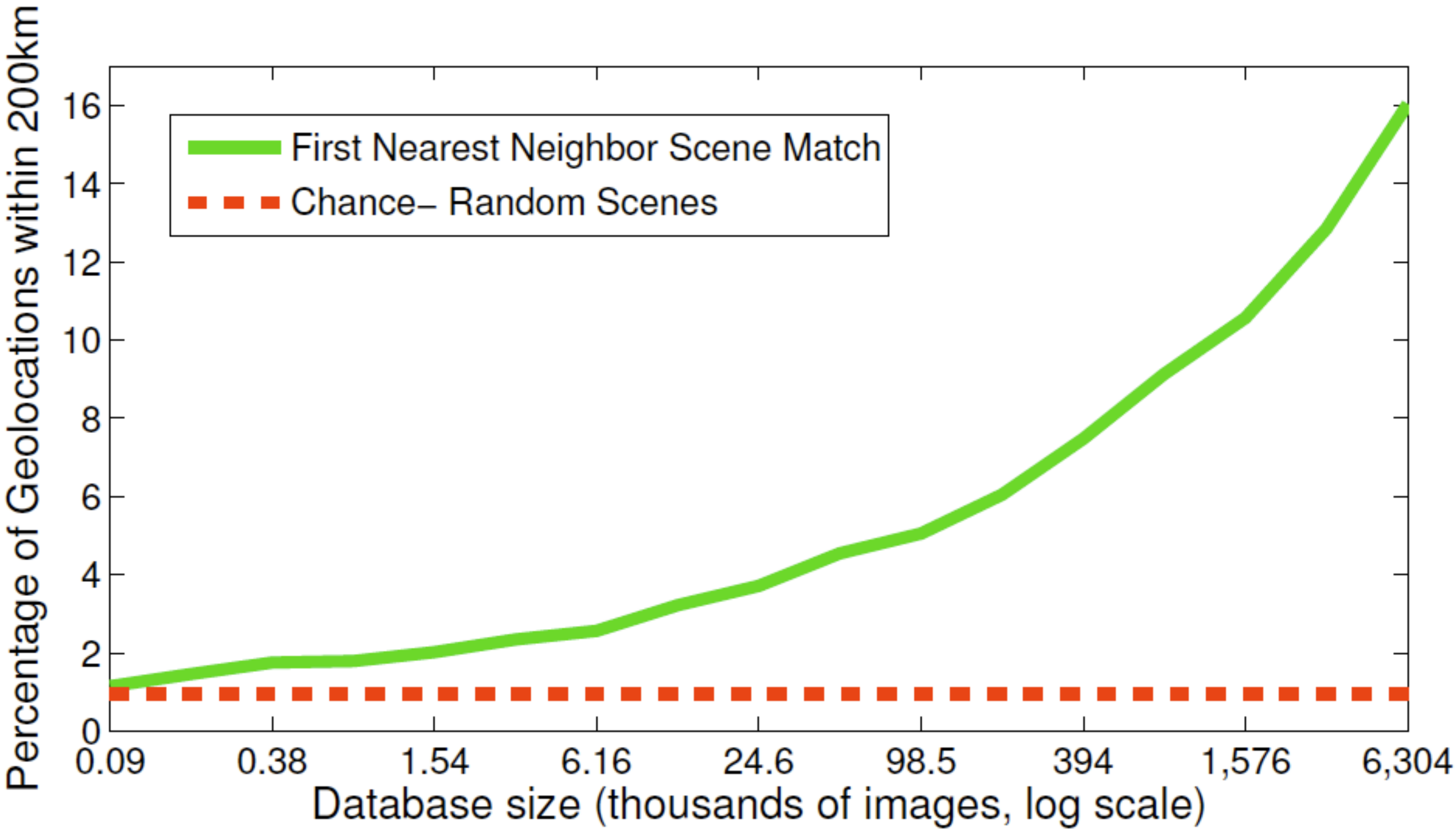


Results


Thailand
Argentina
CostaRica
Argentina
China
Louisiana
Argentina
Argentina
Argentina
Washington
China
Jamaica
NewYorkCity
Argentina
Argentina
Malaysia


California
Oklahoma
SouthAfrica
Zambia
Kenya
Hyderabad
Mongolia
SouthAfrica
Kenya
Kenya
Zambia
Ethiopia
Nevada
africa
Morocco
Tennessee

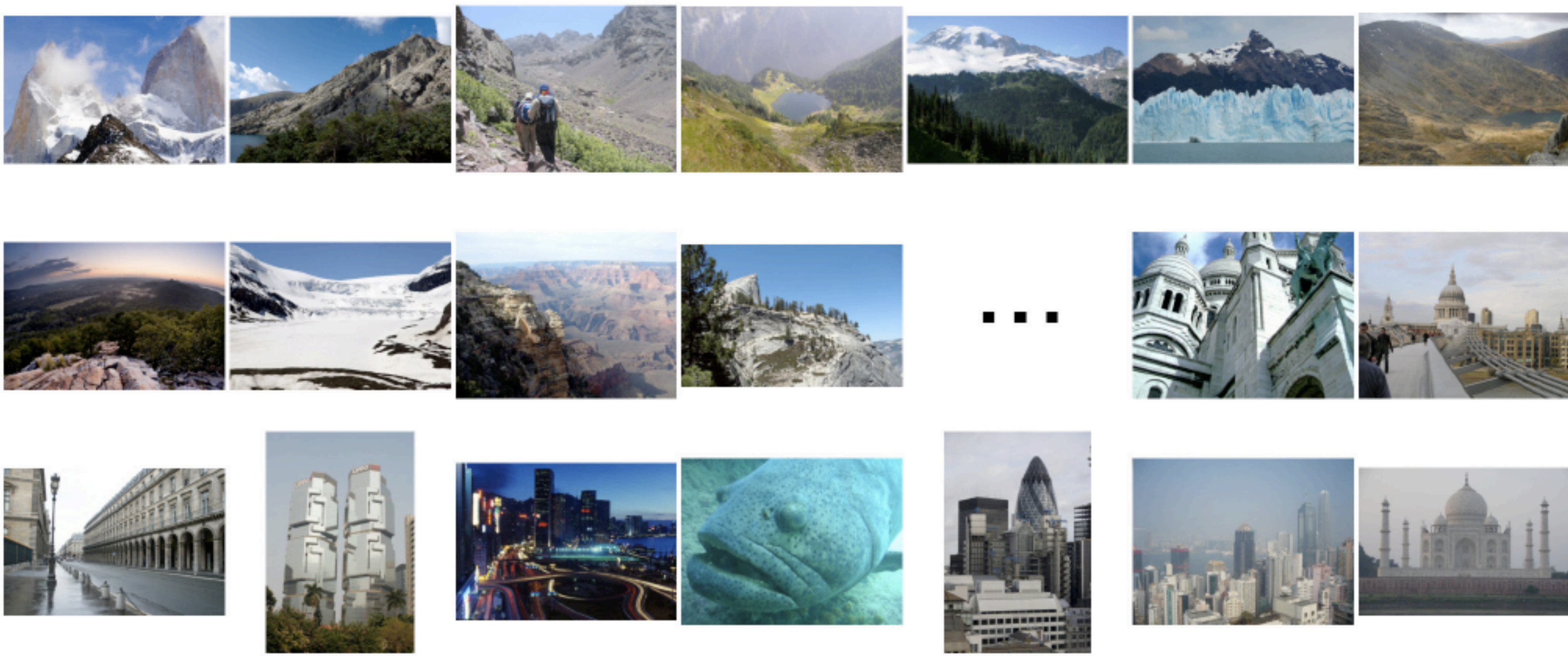
Performance across database size



Estimating population density

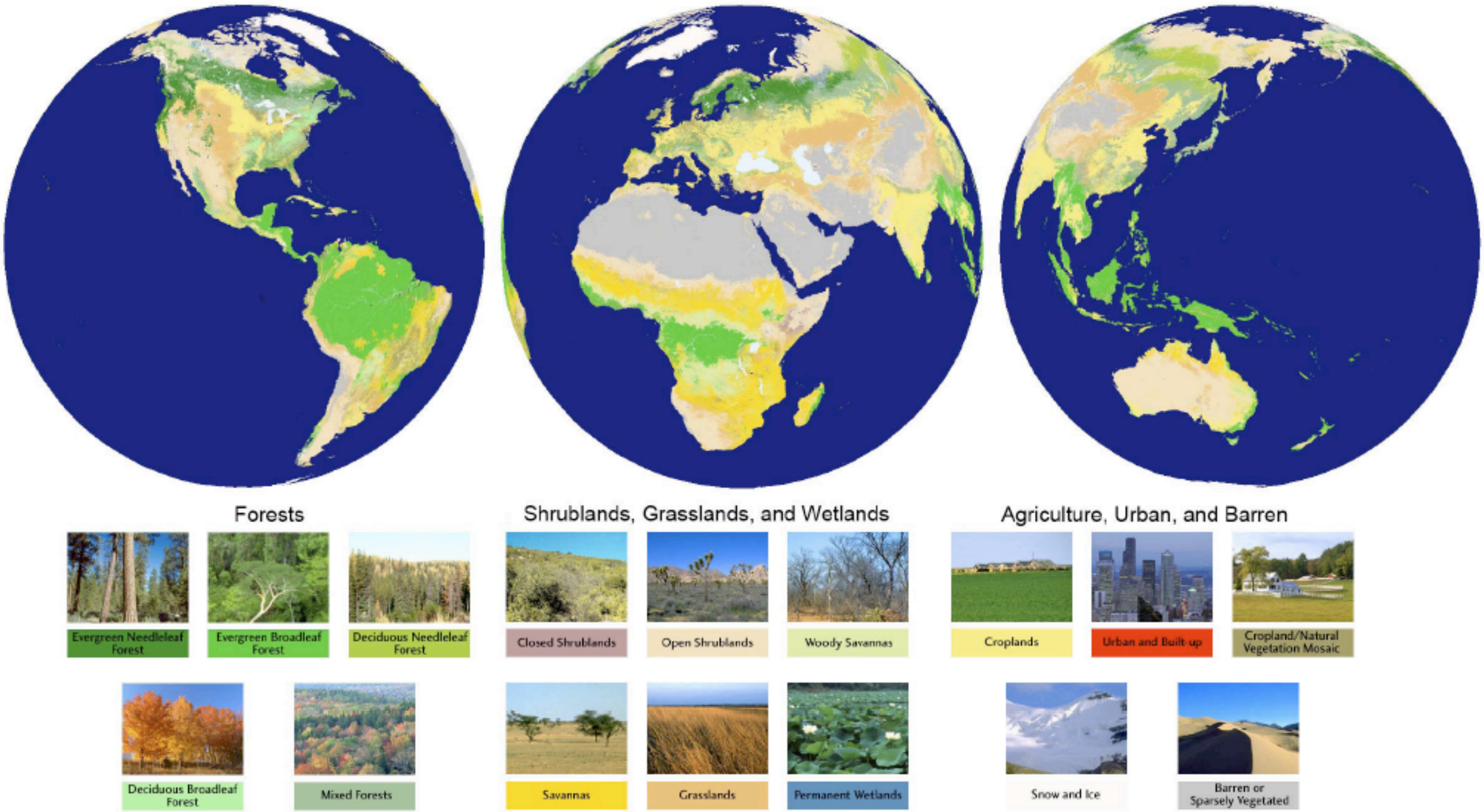
Given a population density map of world, estimate the population of an image by sampling at the estimated location. Images ranked by predicted population density.

Estimating elevation



Given an elevation map of world, they can predict the elevation of an image according to its location. Images ranked by their estimated elevation.

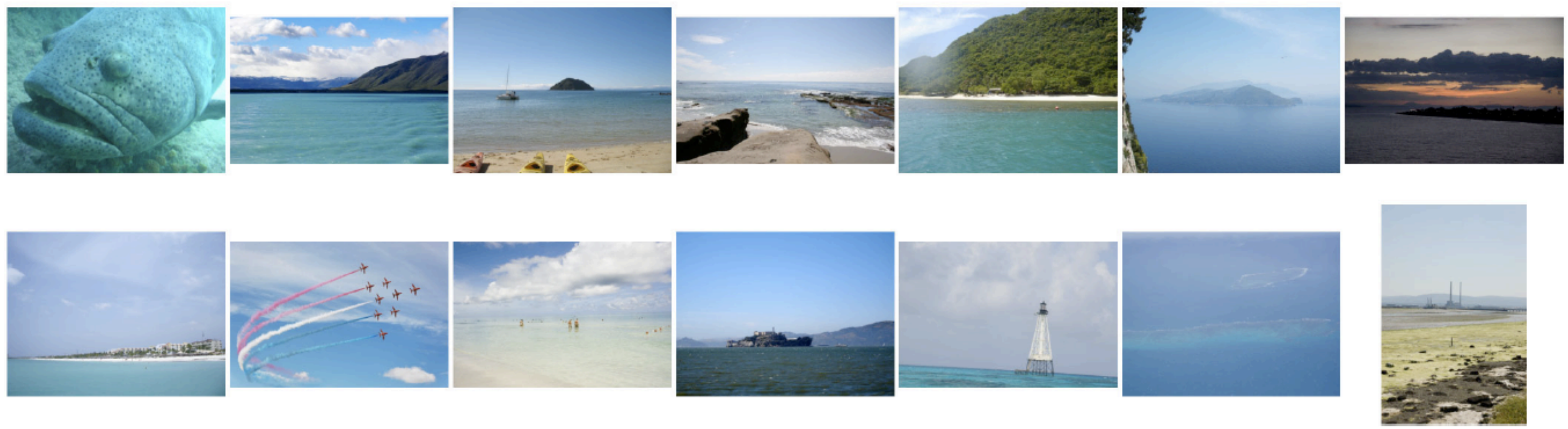
Landcover Classification



Given landcover map, predict which images are most likely to be examples of each category

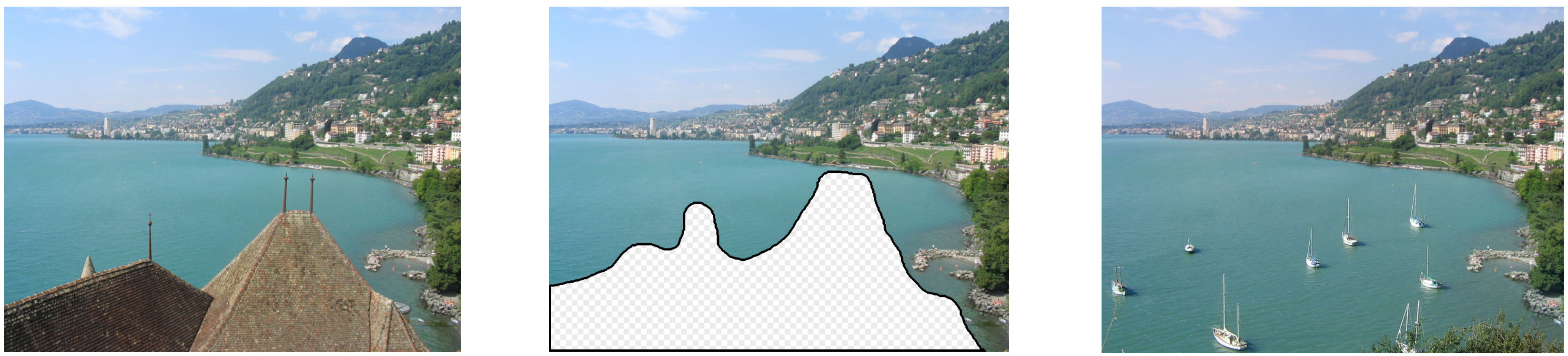
Forest

Water



Savanna

NN for Scene Completion Using Millions of Photographs

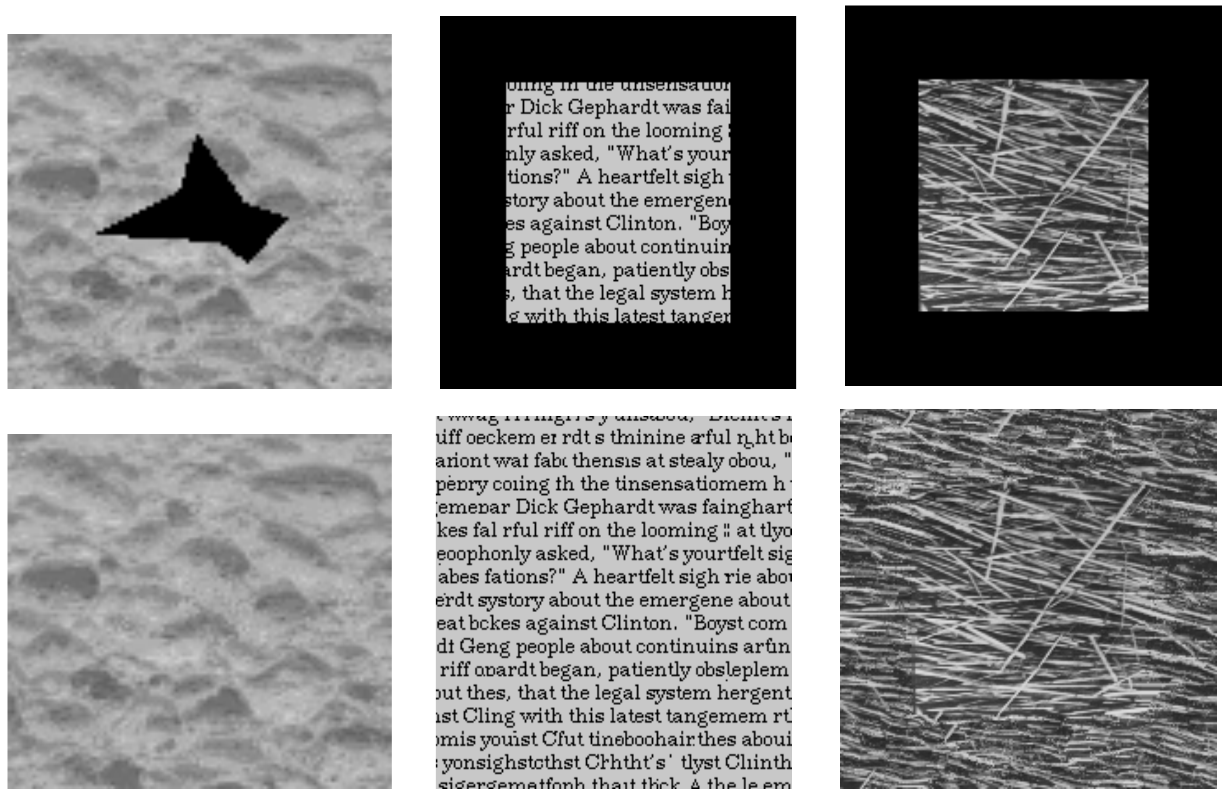


James Hays and Alexei A. Efros
Carnegie Mellon University

Thanks to James & Alyosha for slides!

Hays and Efros, SIGGRAPH 2007

Hays and Efros, SIGGRAPH 2007



Efros and Leung. Texture synthesis by non-parametric sampling. ICCV 1999.

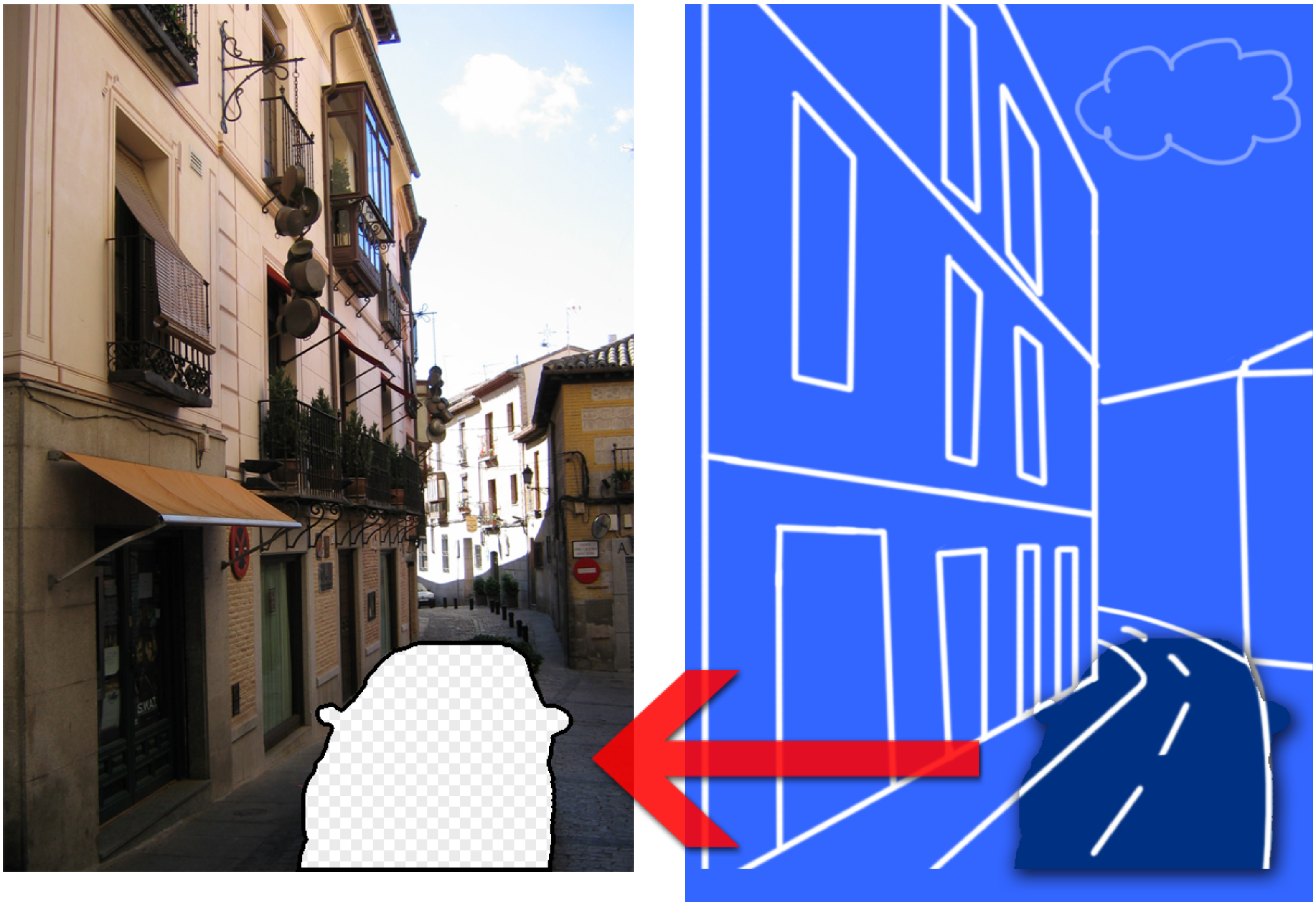
Fill in unknown region from source image parts

Hays and Efros, SIGGRAPH 2007

Efros and Leung result – no notion of semantics, also assumes necessary data is present elsewhere in the image

Hays and Efros, SIGGRAPH 2007

Scene Matching for Image Completion



Hays and Efros, SIGGRAPH 2007

Hays and Efros, SIGGRAPH 2007

Scene Completion Result

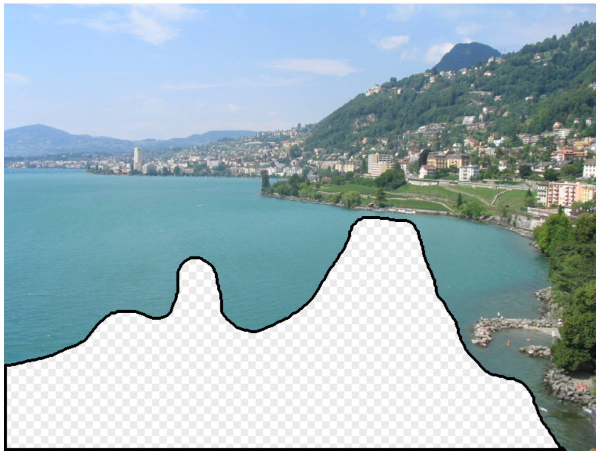
Challenges:

Computational costs of searching lots of images

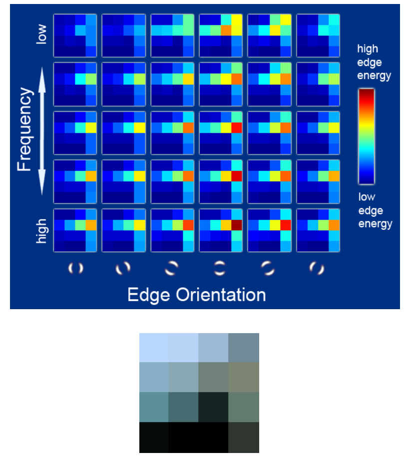
Should fill in missing regions with semantically valid fragments

Hays and Efros, SIGGRAPH 2007

The Algorithm



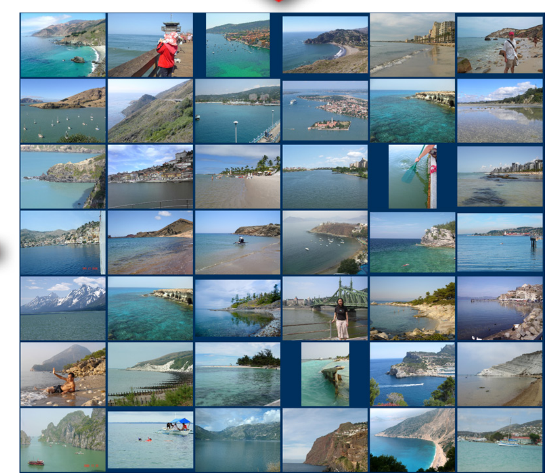
Input image



Scene Descriptor

...

Image Collection



...

200 matches

Context matching
+ blending

20 completions

Hays and Efros, SIGGRAPH 2007

Data

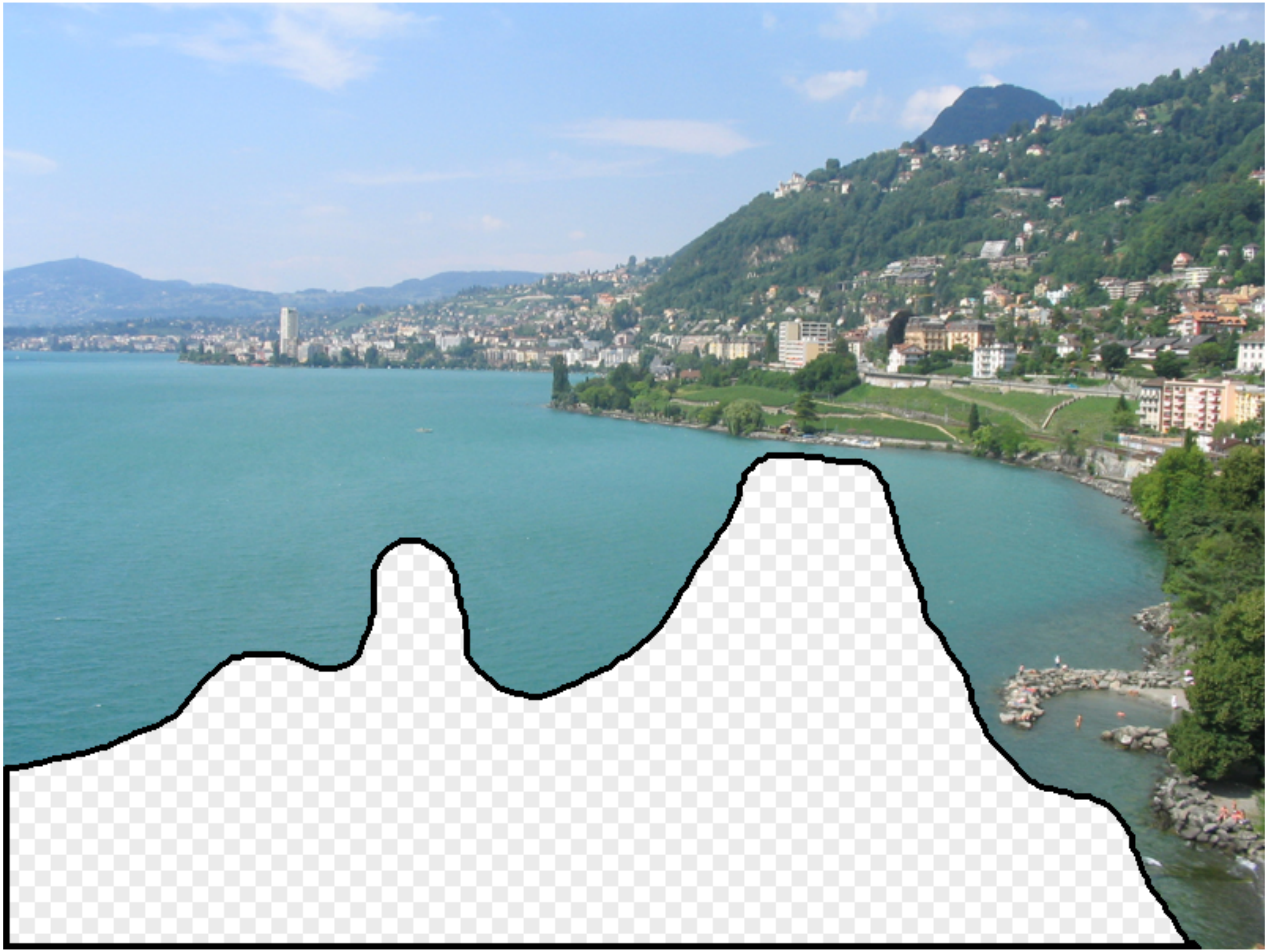
They downloaded <u>**2.3 Million**</u> unique images from Flickr groups and keyword searches.

Groups: lonelyplanet, urban-fragments, ruraldecay …
Keywords: outdoors, vacation, river…

Discard duplicates and small images

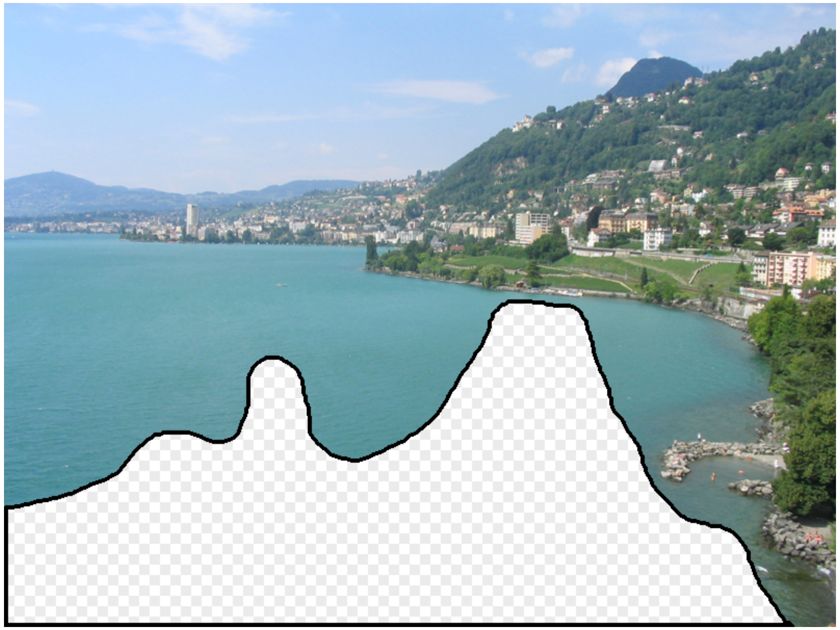
Hays and Efros, SIGGRAPH 2007

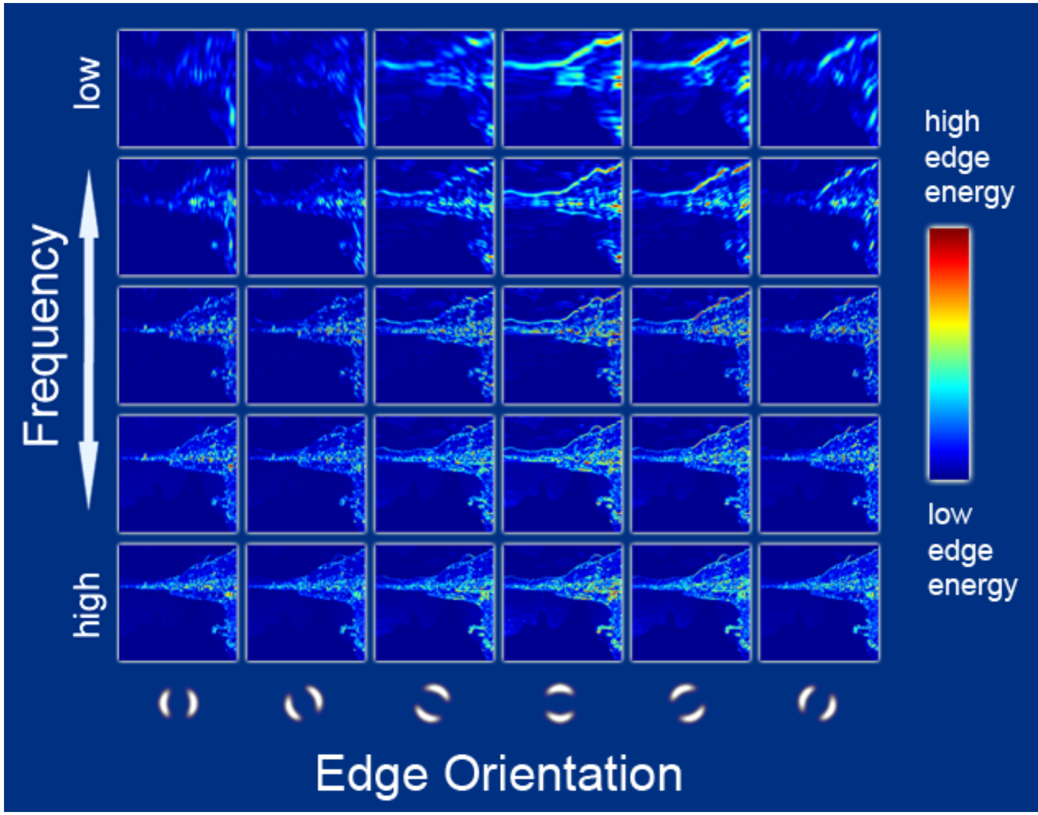
Scene Matching



Hays and Efros, SIGGRAPH 2007

Scene Descriptor

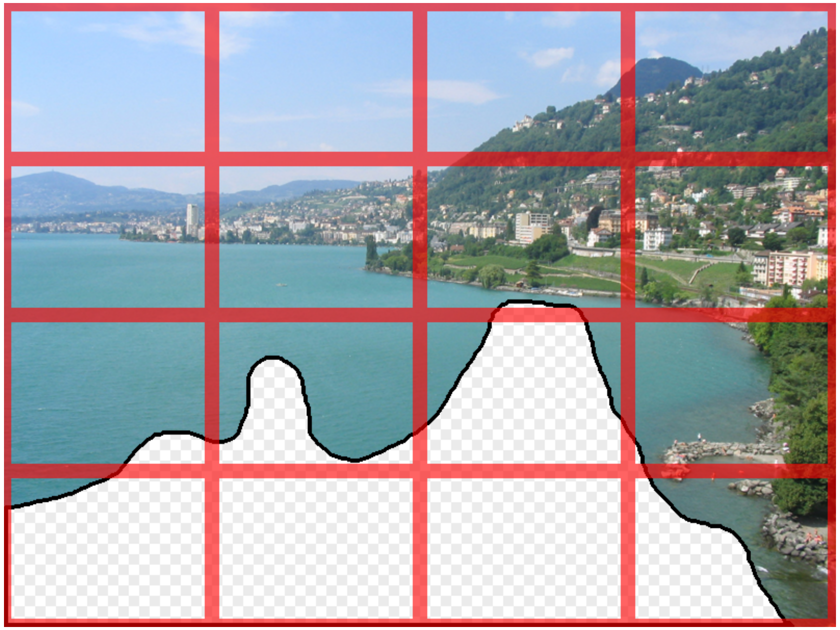


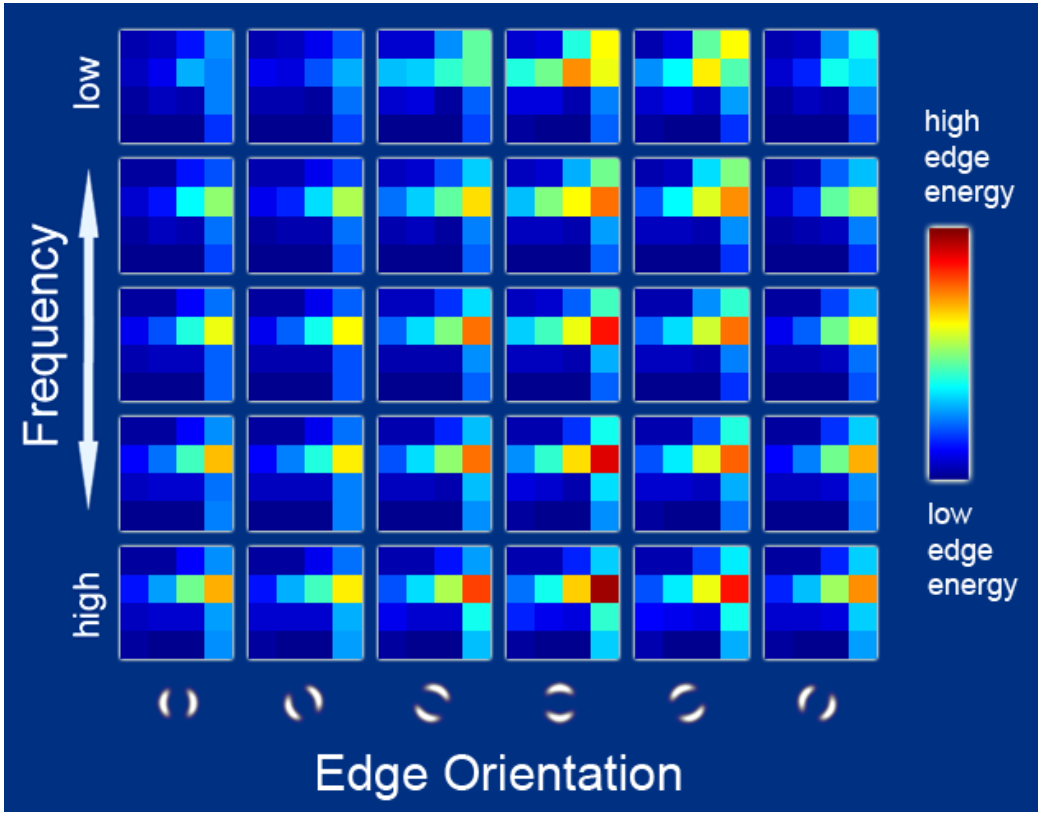


Compute oriented edge response at multiple scales (5 spatial scales, 6 orientations)

Hays and Efros, SIGGRAPH 2007

Scene Descriptor





Gist scene descriptor (Oliva and Torralba 2001)

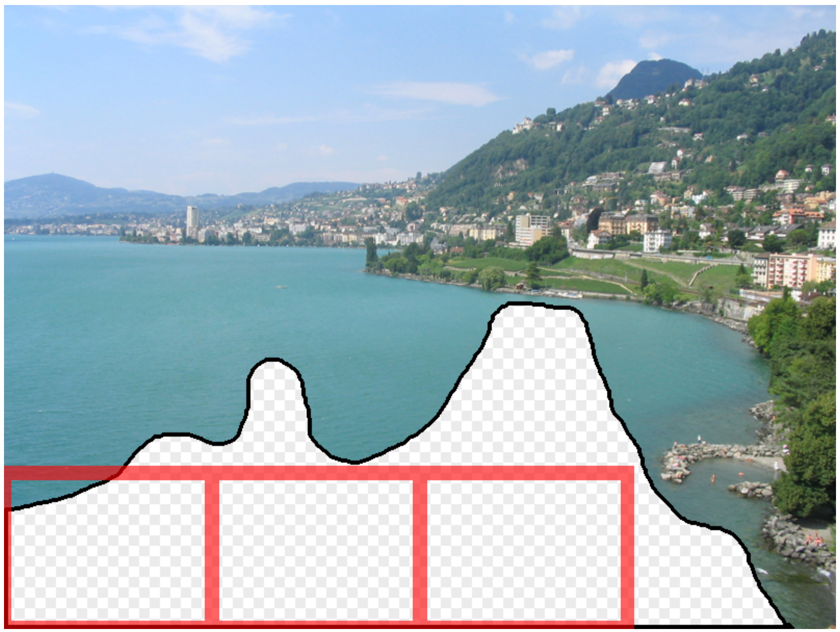
"semantic" descriptor of image composition

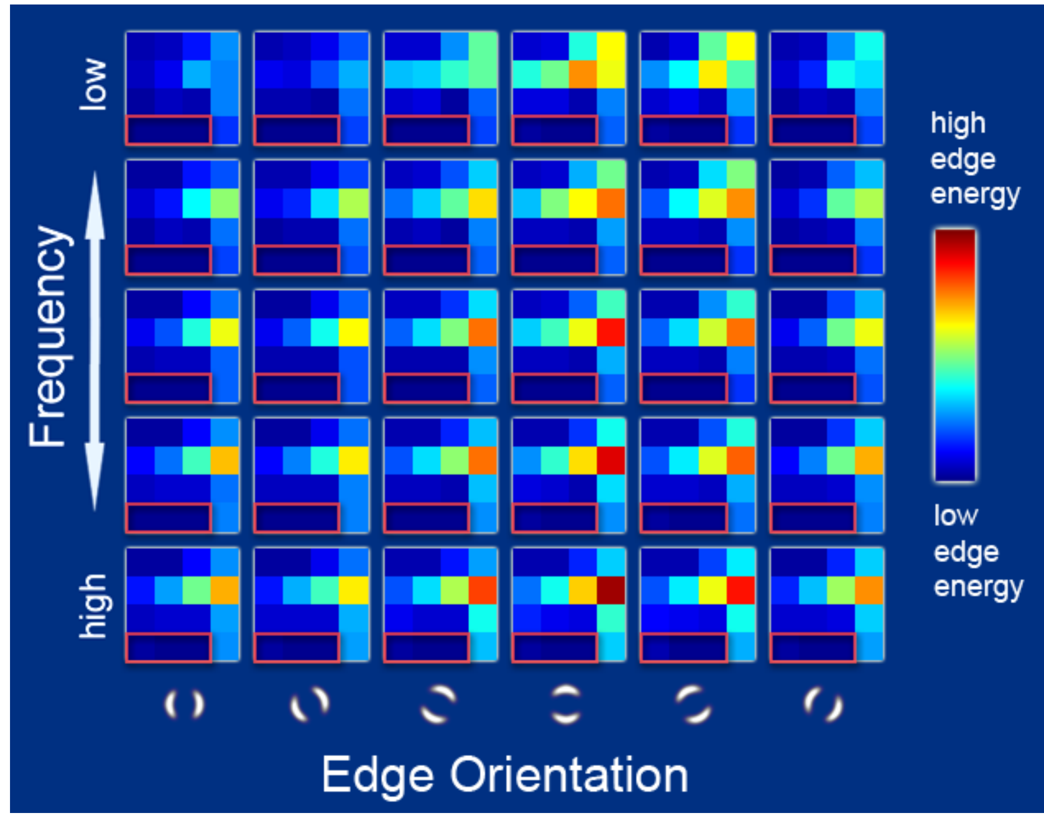
aggregated edge responses over 4x4 windows

scenes tend to be semantically similar under this descriptor if very close

Hays and Efros, SIGGRAPH 2007

Scene Descriptor

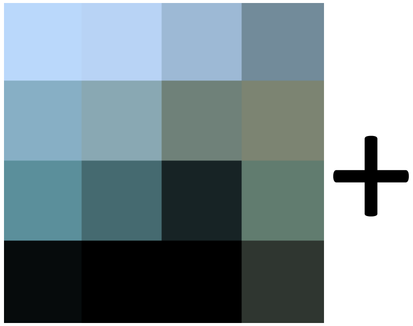




Gist scene descriptor - with missing regions masked (weighted based on percentage of valid pixels)

Hays and Efros, SIGGRAPH 2007

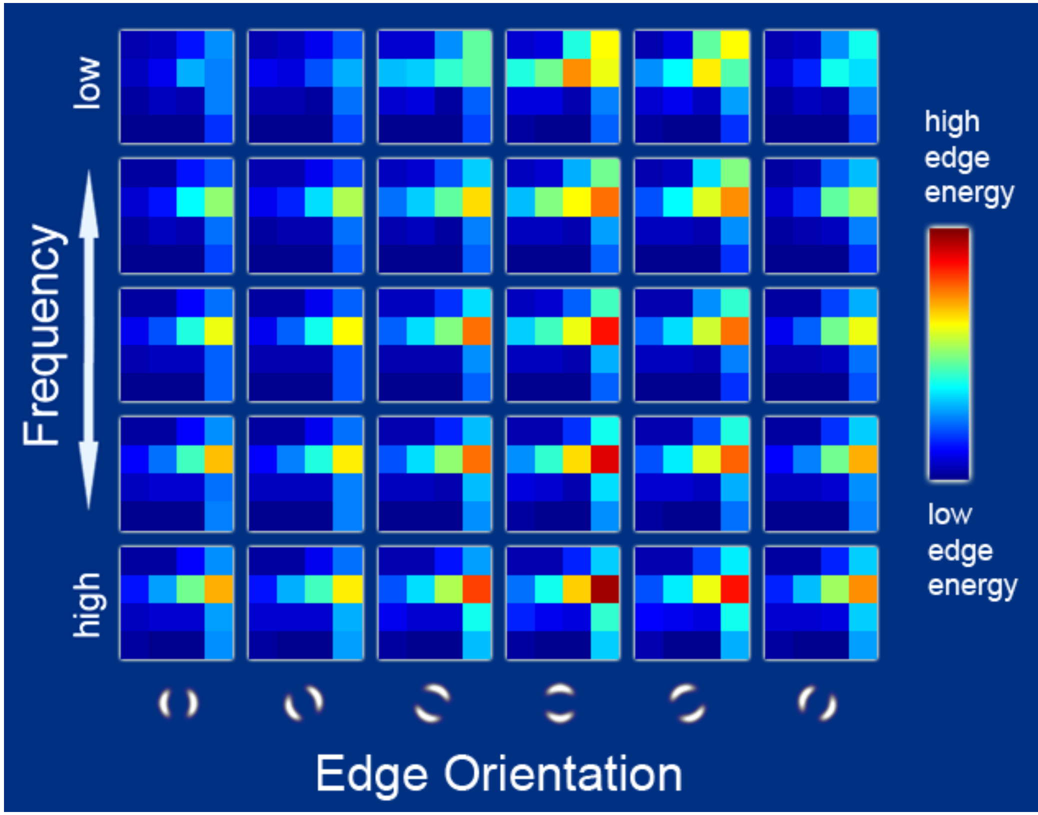
Scene Descriptor



Color descriptor – color of the query image downsampled to 4x4

Distances calculated by SSD between query image descriptors & imgs in database

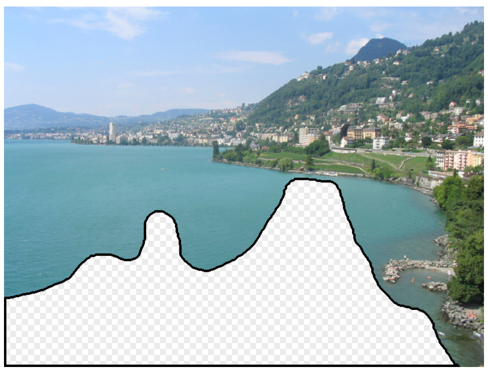
Total Dist = color dist + 2*gist dist

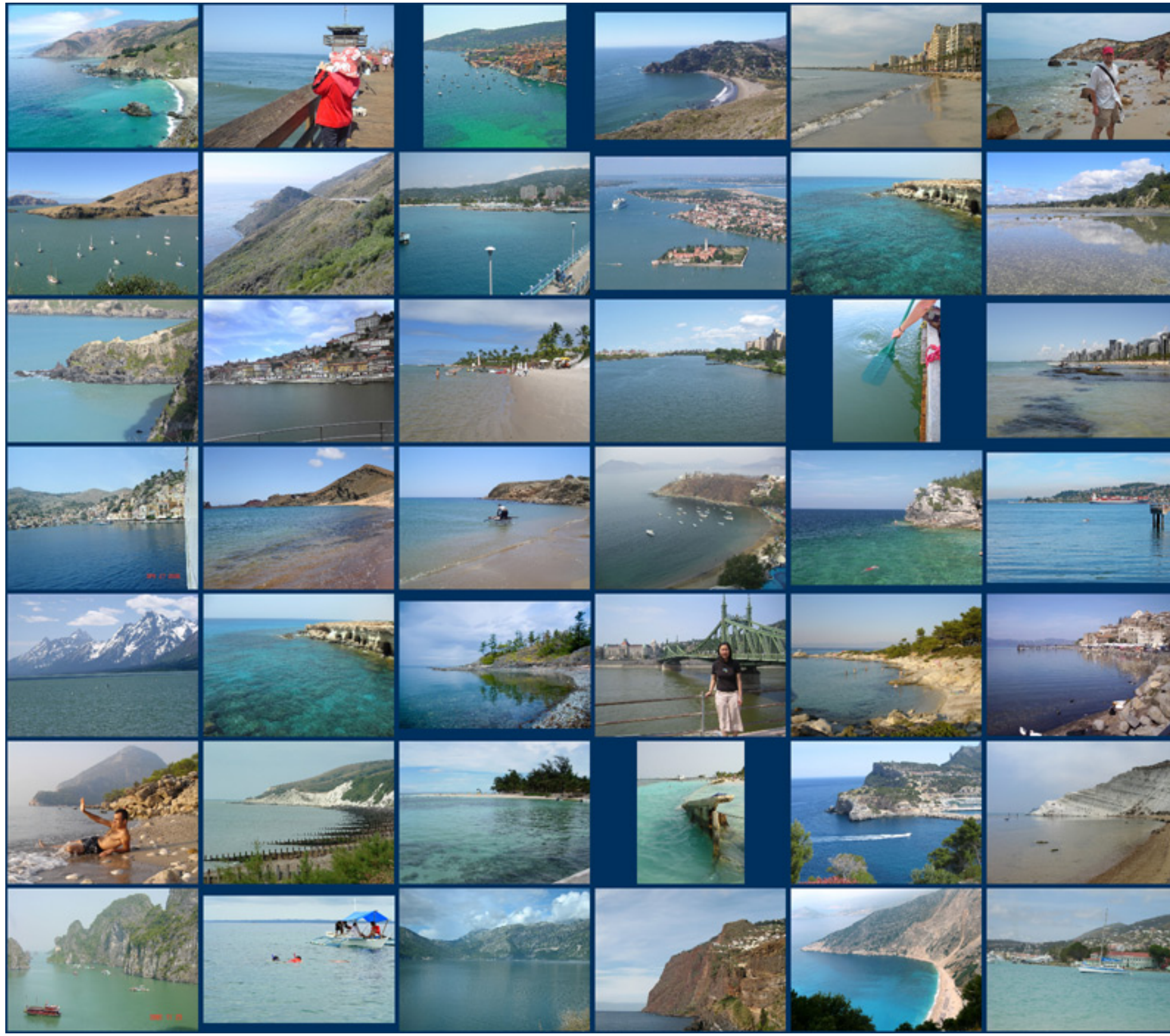


Gist scene descriptor
(Oliva and Torralba 2001)

Hays and Efros, SIGGRAPH 2007

Hays and Efros, SIGGRAPH 2007

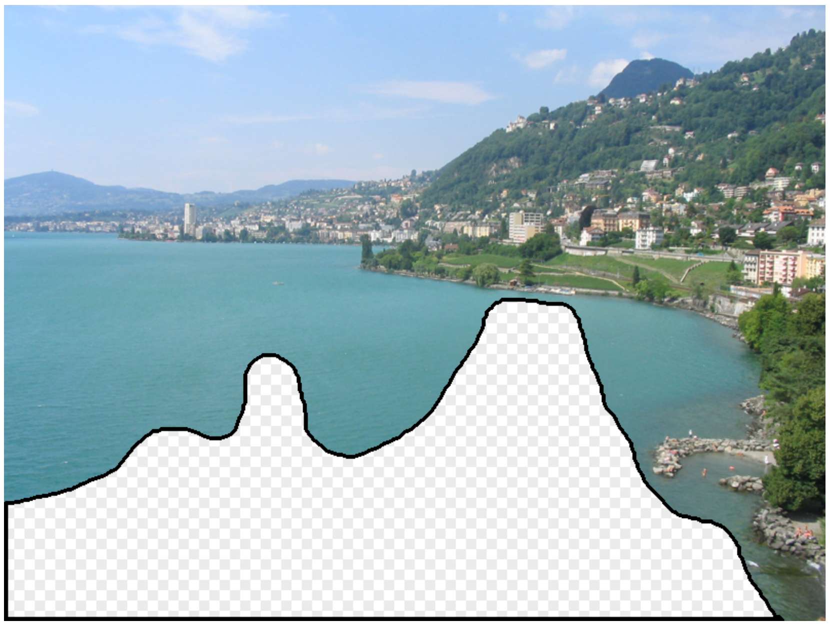




… 200 total

Hays and Efros, SIGGRAPH 2007

Context Matching



Need to more precisely align matching scenes to local img context around missing region
    local context = all pixels within 80 pixel radius of hole's boundary
Compute pixel-wise error of 200 best scene matches across all valid translations and 3 scales
Compute texture similarity of proposed fill-in to removed region

Hays and Efros, SIGGRAPH 2007


Graph cut
Hays and Efros, SIGGRAPH 2007

Final result – blended between the two images along the cut to merge seamlessly

Hays and Efros, SIGGRAPH 2007

Hays and Efros, SIGGRAPH 2007

Hays and Efros, SIGGRAPH 2007

Hays and Efros, SIGGRAPH 2007

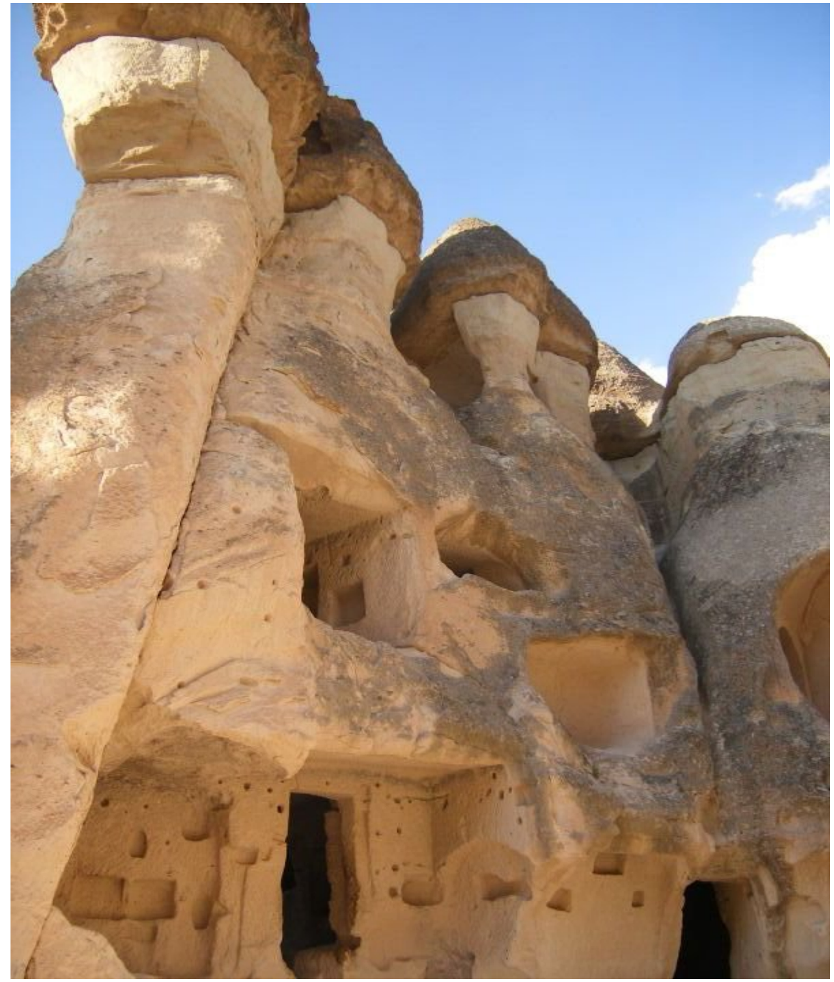
Failures

Hays and Efros, SIGGRAPH 2007

Failures

Hays and Efros, SIGGRAPH 2007

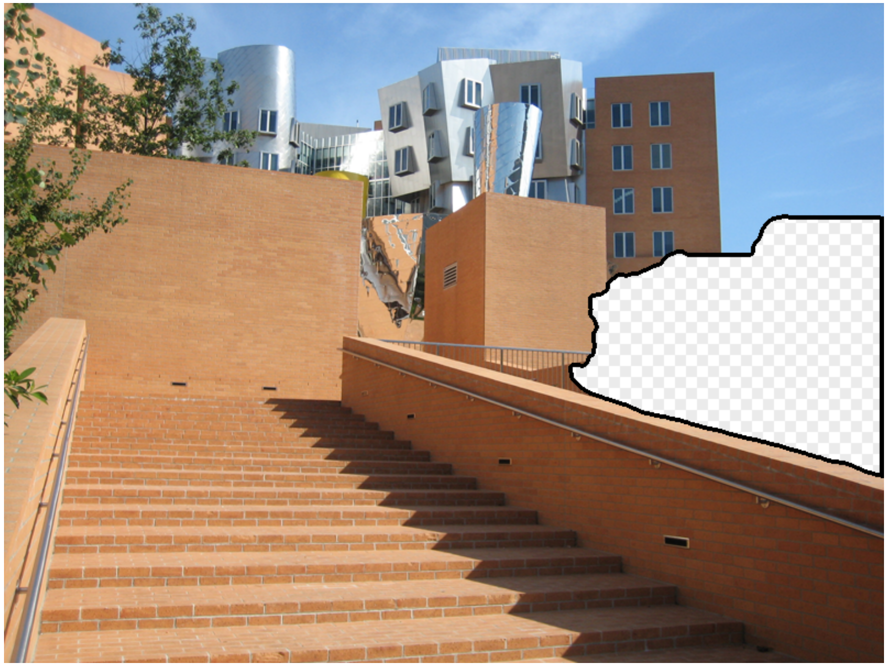
Failures



Cause of failure – atypical scene caused lack of good matches

Hays and Efros, SIGGRAPH 2007

Failures

Hays and Efros, SIGGRAPH 2007

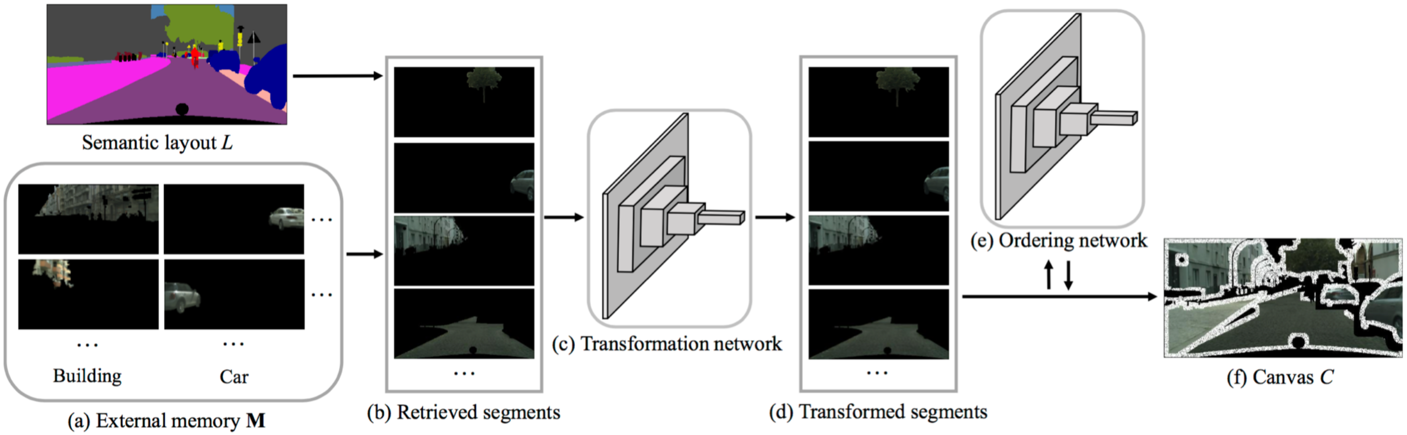
Related work in Deep Learning

Semantic layout L

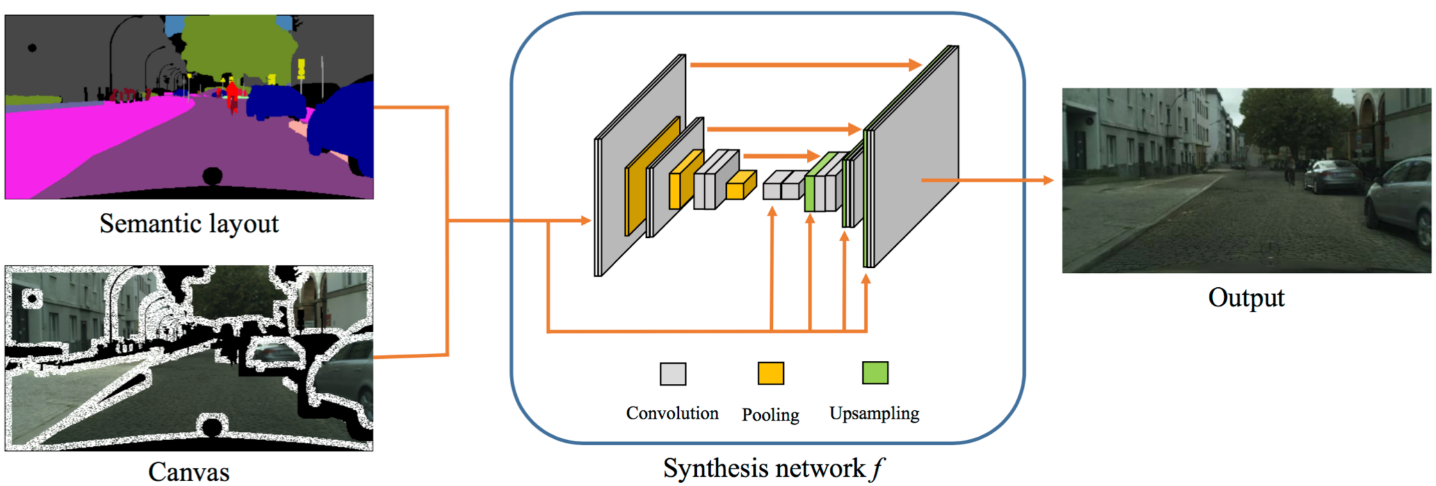
Output

Goal: Image synthesis from semantic layout

Related work in Deep Learning



Match to large number of training segments (M)



Generate output image using matched segments + generative model

NN for Image Captioning

Traditional Recognition...

person

car

shoe

Human centric recognition outputs

car

Human centric recognition outputs

pink car

Human centric recognition outputs

car on road

Human centric recognition outputs

Little pink smart car parked on the side of a road in a London shopping district.

Telling the *"story of an image"*

NN Captioning: Compose descriptions from recognized content

+ existing descriptions

(Captioning circa 2012-2013)

Through the smoke

Duna Portrait #5

Mirror and gold

the cat lounging in the sink

Data exists, but buried in junk!

Captions in the Wild

http://tamaraberg.com/sbucaptions

The Egyptian cat statue by the floor clock and perpetual motion machine in the pantheon


Little girl and her dog in northern Thailand. They both seemed interested in what we were doing


Interior design of modern white and brown living room furniture against white wall with a lamp hanging.


Man sits in a rusted car buried in the sand on Waitarere beach


Our dog Zoe in her bed


Emma in her hat looking super cute

Ordonez et al, NIPS 2011

Harness the Web

Ordonez et al, NIPS 2011

Global Matching (GIST + Color)

Captioned Photo Dataset
1 *million* captioned images!

The bridge over the lake on Suzhou Street.

Bridge to temple in Hoan Kiem lake.

A walk around the lake near our house with Abby.

Smallest house in paris between red (on right) and beige (on left).

Hangzhou bridge in West lake.

The daintree river by boat.

...

Transfer Whole Caption(s)

e.g. “The bridge over the lake on Suzhou Street.”

Transfer pieces of Captions

Object appearance → NP: the dirty sheep

Object pose → VP: meandered along a desolate road

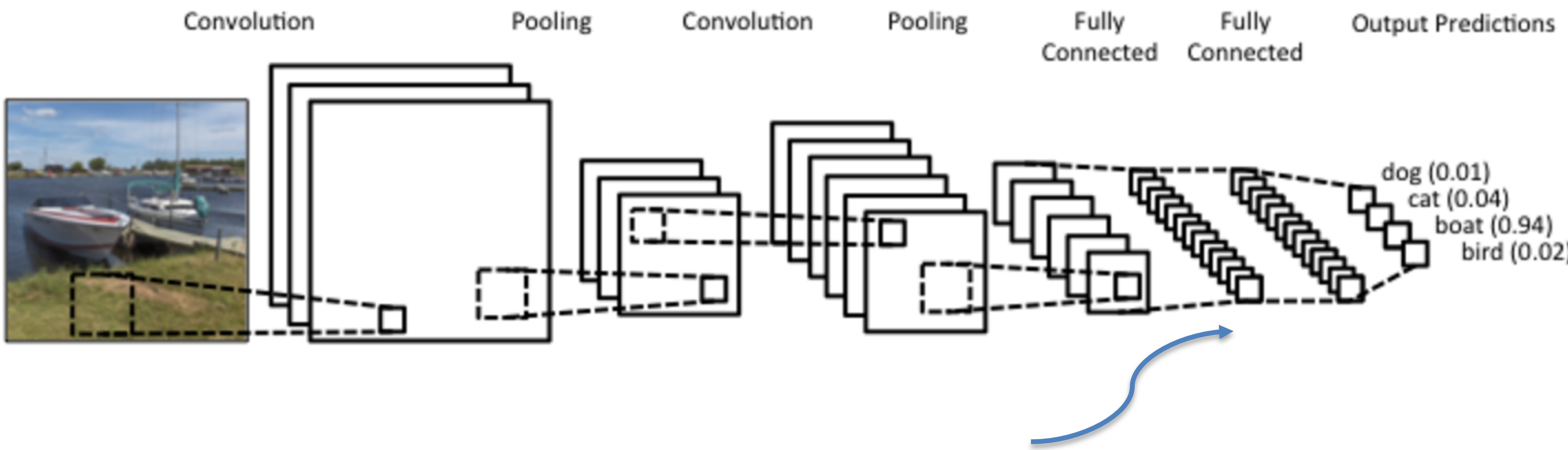
Scene appearance → PP: in the highlands of Scotland

Region appearance & relationship → PP: through frozen grass

Example Composed Description: the dirty sheep meandered along a desolate road in the highlands of Scotland through frozen grass

Kuznetsova et al, ACL 2012

CNNs (coming soon)



Feature representation extracted from some level of the CNN

With deep learning now features can be learned rather than hand-crafted.

Better features == better captions

A bedroom with a bed and a couch.

A hotel room with two beds and a table.

A train is stopped at a train station.

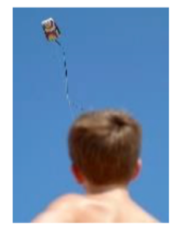
A red and white train parked in a train station.

A group of people sitting around in a living room.

A group of people sitting on a couch in a living room.

A man riding a wave on a surfboard.

A man riding a wave on a surfboard in the ocean.



A person flying a kite in the sky.

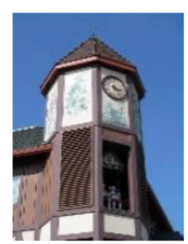
A person flying a kite in the sky.

A cat sitting in a bathroom sink.

A black and white cat sitting in a bathroom sink.

A baseball player holding a bat on a field.

A baseball player holding a bat on a field.



A building with a clock on the top.

A clock tower on the top of a building.

Brainstorming Session

- Form small groups and come up with applications where you could use feature matching for some task