EE795: Computer Vision and Intelligent Systems

Spring 2012
TTh 17:30-18:45 FDH 204

Lecture 17
130402

http://www.ee.unlv.edu/~b1morris/ecg795/

Outline

- Review
  - Background Subtraction
  - Stauffer and Grimson

Slides from Birgi Tamersoy, UT Austin

- Object Recognition Intro (Chapter 14)

Slides from Steve Seitz, Washington

- Excellent References
  - http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html
  - http://web.eecs.umich.edu/~silvio/teaching/lectures/lecture19.pdf

Background Subtraction

- Motion is an important
  - Indicates an object of interest

- Background subtraction
  - Given an image (usually a video frame), identify the **foreground objects** in that image
    - Assume that foreground objects are moving
    - Typically, moving objects more interesting than the scene
    - Simplifies processing – less processing cost and less room for error

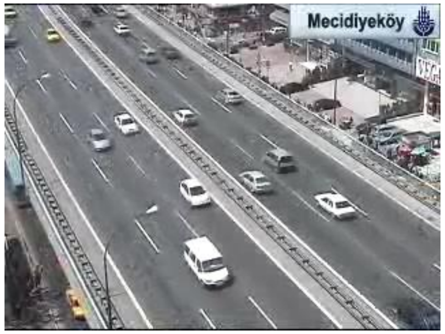
Requirements

- A reliable and robust background subtraction algorithm should handle:
  - Sudden or gradual illumination changes
    - Light turning on/off, cast shadows through a day
  - High frequency, repetitive motion in the background
    - Tree leaves blowing in the wind, flag, etc.
  - Long-term scene changes
    - A car parks in a parking spot

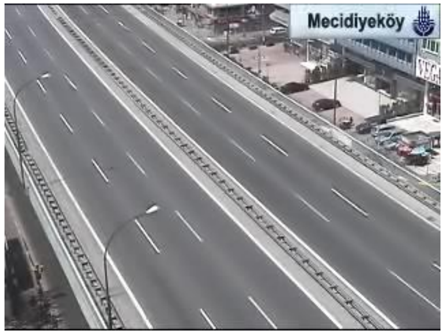
Basic Approach

- Estimate the background at time t
- Subtract the estimated background from the current input frame
- Apply a threshold, Th, to the absolute difference to get the foreground mask.
  - $I(x,y,t) - B(x,y,t)| > Th = F(x,y,t)$

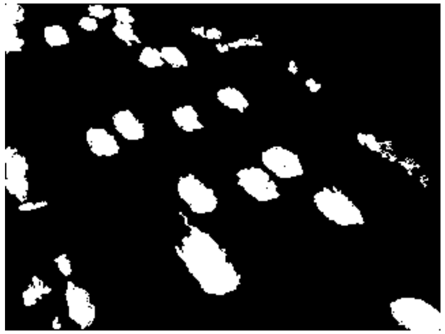
$|$



$-$



$| > Th =$

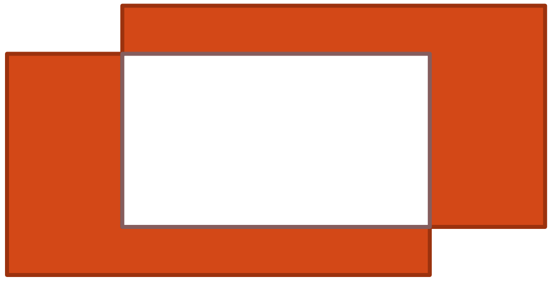


$I(x,y,t)$ $B(x,y,t)$ $F(x,y,t)$

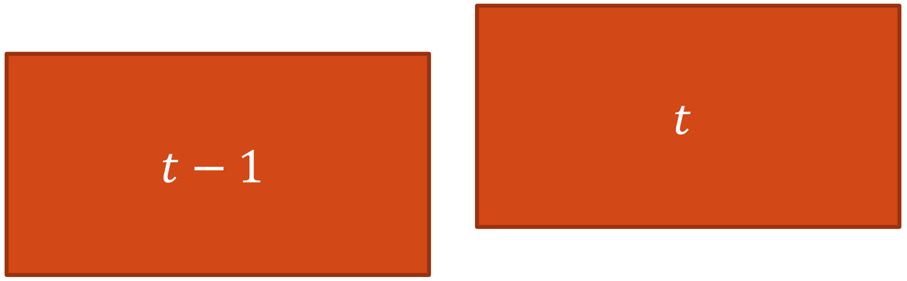
How can we estimate the background?

Frame Differencing

- Background is estimated to be the previous frame
  - $B(x, y, t) = I(x, y, t-1)$
- Depending on the object structure, speed, frame rate, and global threshold, may or may not be useful
  - Usually not useful – generates impartial objects and ghosts



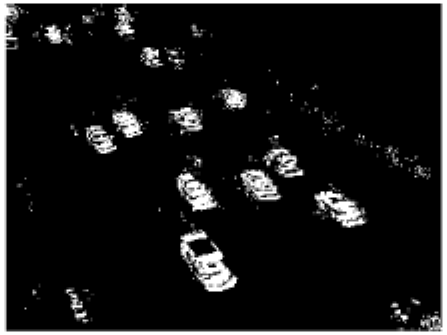
Incomplete object



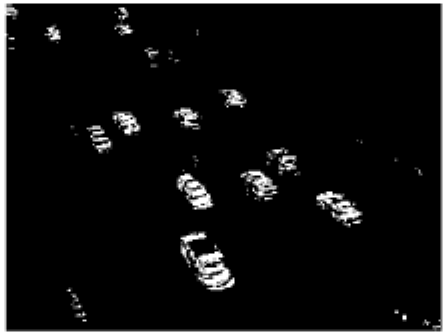
ghosts

Frame Differencing Example

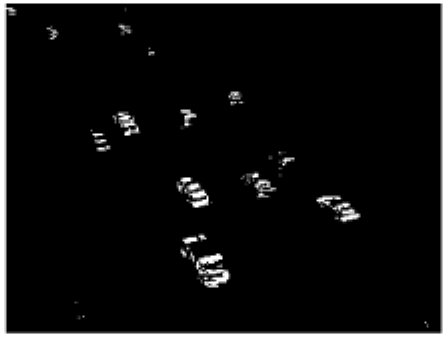
$Th = 25$



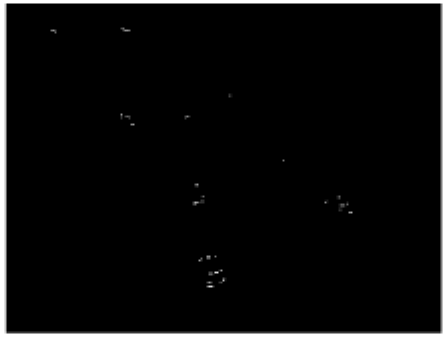
$Th = 50$



$Th = 100$



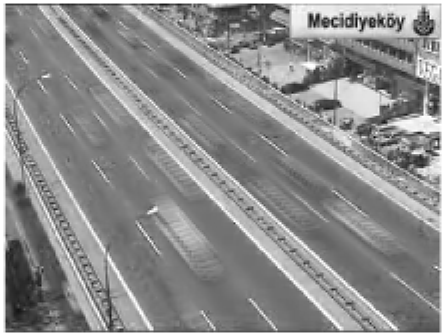
$Th = 200$



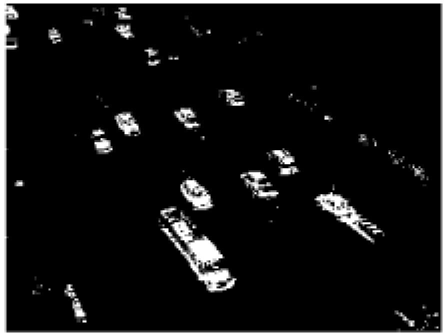
Mean Filter

- Background is the mean of the previous N frames
  - $B(x,y,t) = \frac{1}{N}\sum_{i=0}^{N-1} I(x,y,t-i)$
  - Produces a background that is a temporal smoothing or "blur"
- $N = 10$

Estimated Background



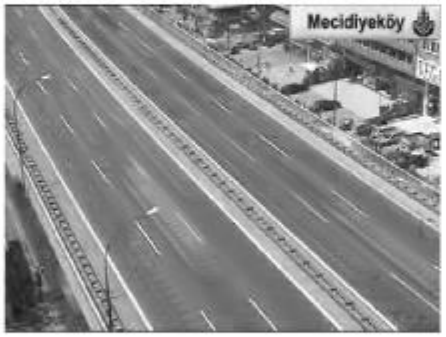
Foreground Mask



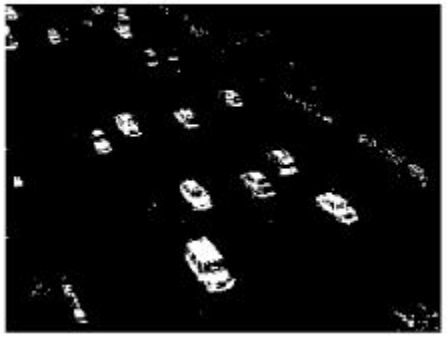
Mean Filter

- $N = 20$

Estimated Background

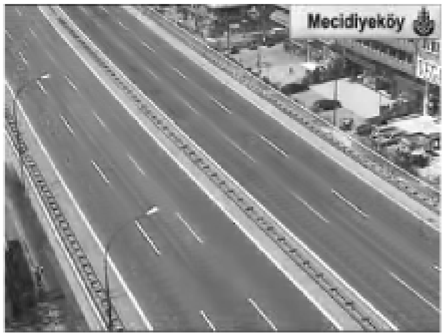


Foreground Mask

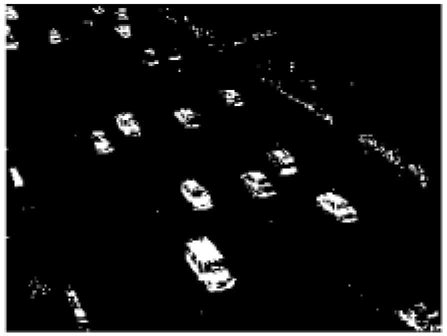


- $N = 50$

Estimated Background



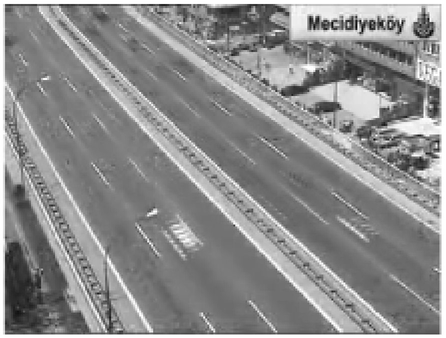
Foreground Mask



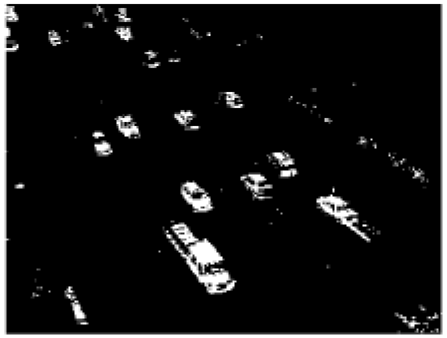
Median Filter

- Assume the background is more likely to appear than foreground objects
  - $B(x, y, t) = median\big(I(x, y, t - i)\big),\ i \in \{0, N - 1\}$
- $N = 10$

Estimated Background



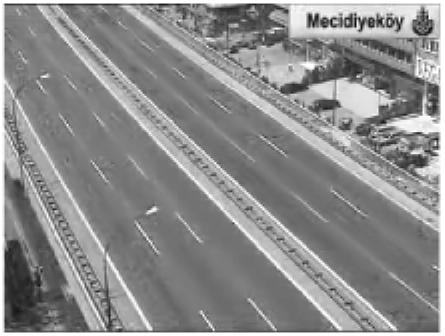
Foreground Mask



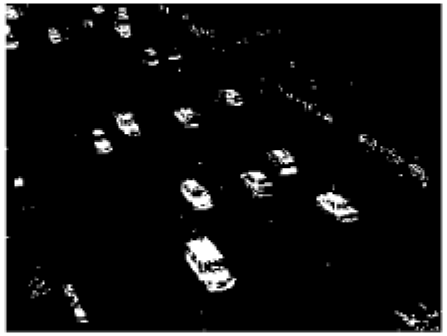
Median Filter

- $N = 20$

Estimated Background

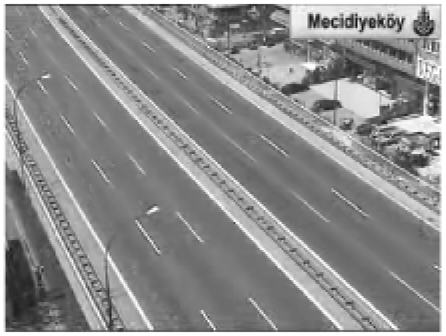


Foreground Mask

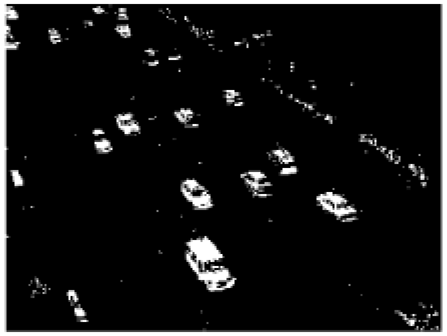


- $N = 50$

Estimated Background



Foreground Mask



Frame Difference Advantages

- Extremely easy to implement and use
- All the described variants are pretty fast
- The background models are not constant
    - Background changes over time

Frame Differencing Shortcomings

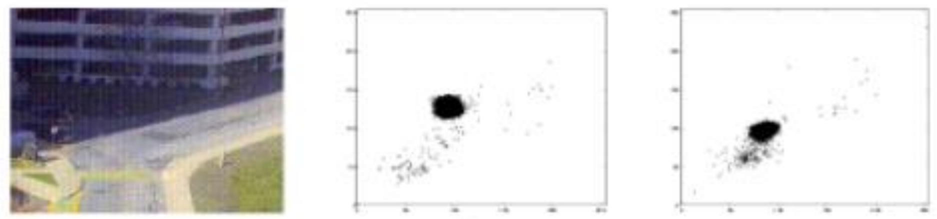
- Accuracy depends on object speed/frame rate
- Mean and median require large memory
  - Can use a running average
  - $B(x,y,t) = (1-\alpha)B(x,y,t-1) + \alpha I(x,y,t)$
    - α – is the learning rate
- Use of a global threshold
  - Same for all pixels and does not change with time
  - Will give poor results when the:
    - Background is bimodal
    - Scene has many slow moving objects (mean, median)
    - Objects are fast and low frame rate (frame diff)
    - Lighting conditions change with time

Improving Background Subtraction

- Adaptive Background Mixture Models for Real-Time Tracking
  - Chris Stauffer and W.E.L. Grimson
- The paper on background subtraction
  - Over 4000 citations since 1999

Motivation

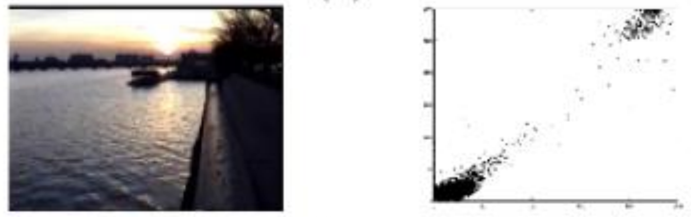
- Robust background subtraction should handle lighting changes, repetitive motion from clutter and long term scene changes



(a)

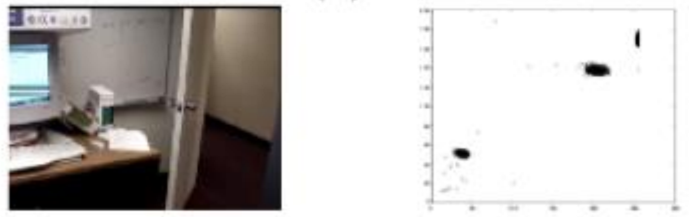
Differing threshold over time

RG plots of a single pixel



(b)

Bimodal distribution over time



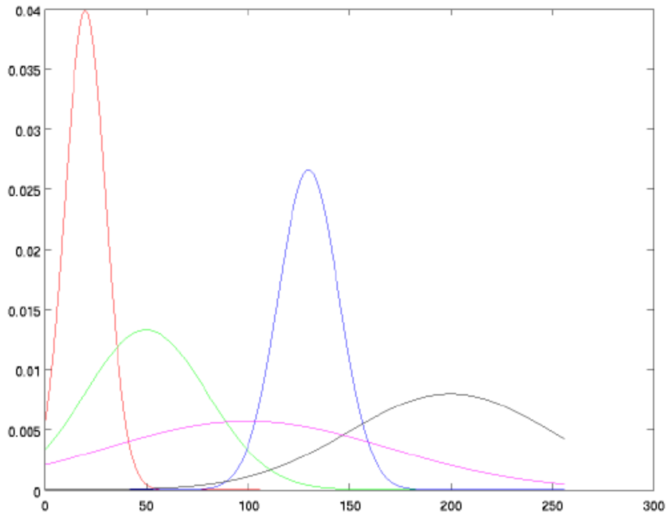
(c)

Algorithm Overview

- Pixel value is modeled as a mixture of adaptive Gaussian distributions
  - Why a mixture?
    - Multiple surfaces appear in a pixel (mean background assumes a single pixel distribution)
  - Why adaptive?
    - Lighting conditions change
- Gaussians are evaluated to determine which ones are most likely to correspond to the background
- Pixels that do not match the background Gaussians are classified as foreground

Online Mixture Model

- History of a pixel is known up to current time t
  - $\{X_1, \dots, X_t\} = \{I(x_o, y_o, i) : 1 \leq i \leq t\}$
- Model the history as a mixture of K Gaussian Distributions
  - $P(X_t) = \sum_{i=1}^{K} w_{i,t} \mathcal{N}(X_t | u_{i,t}, \Sigma_{i,t})$
    - $w_{i,t}$ - prior probability (weight) of Gaussian i
  - For a grayscale image with $K = 5$



Model Adaption

- Online K-means approximation is used to update the Gaussians
- Match a new pixel X_{t+1} to an existing Gaussian and update
  - Must be within 2.5σ
  - $\mu_{i,t+1} = (1-\rho)\mu_{i,t} + \rho X_{t+1}$
  - $\sigma^2_{i,t+1} = (1-\rho)\sigma^2_{i,t} + \rho\left(X_{t+1} - \mu_{i,t}\right)^2$
    - $\rho = \alpha\mathcal{N}\left(\mathrm{X}_{\mathrm{t+1}} \middle| \mu_{i,t}, \sigma^2_{i,t}\right)$
    - α – is a learning rate
- Prior weights of Gaussians are updated
  - $w_{i,t+1} = (1-\alpha)w_{i,t} + \alpha\left(M_{i,t+1}\right)$
  - $M_{i,t+1} = 1$ for matching Guassian or $M_{i,t+1} = 0$ for all others

Model Adaption

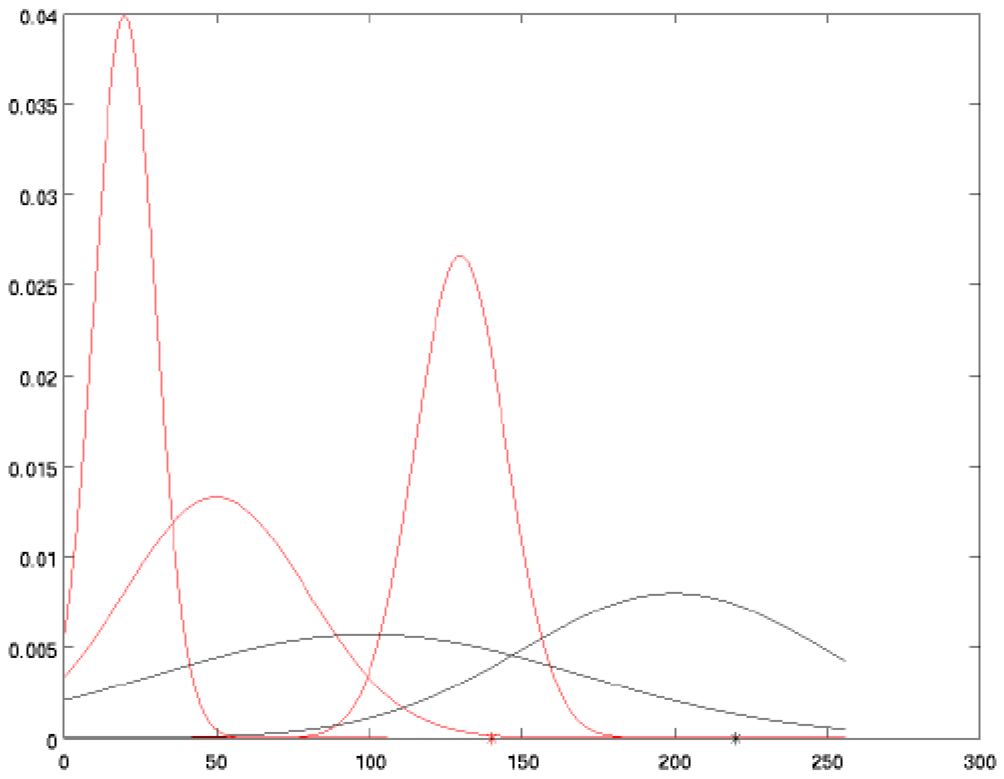
- If X_{t+1} do not match and of the K Gaussians, there is no matching mixture
- Replace the least probable distribution with a new one
  - Least probable in the ω/σ sense (to be explained)
  - The newly created distribution has
    - $\mu_{t+1} = X_{t+1}$
    - Has high variance and low prior weight

Background Model Estimation

- Heuristic: Gaussians with the most **supporting evidence** and **least variance** should correspond to the background
  - Why?
- Gaussians are ordered by the value of ω/σ
  - High support and smaller variance give larger value
- First B distributions are selected as the background model
  - $B = argmin_b(\sum_{i=1}^{b} w_i > T)$
    - T minimum portion of image expected to be background

Background Estimation Example

- After background estimation, red are the background and black are foreground

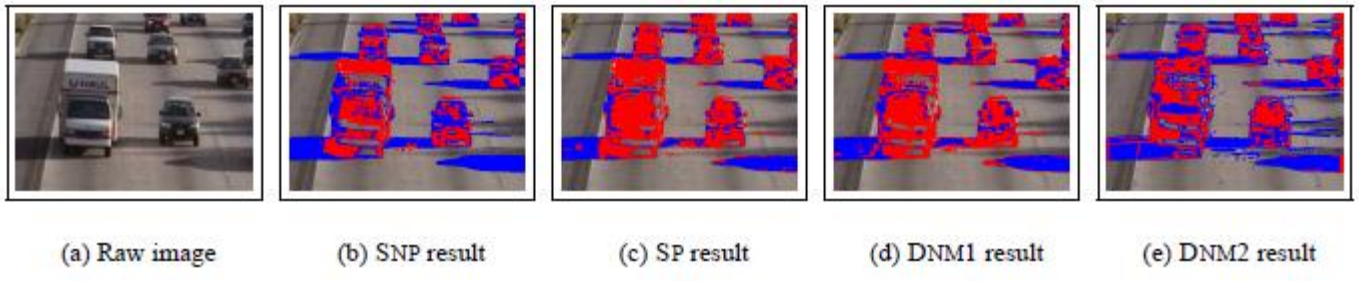


Discussion

- Advantages
  - Different threshold for each pixel
  - Pixel-wise thresholds adapt over time
  - Objects are allowed to become part of the background without destroying the existing background model
  - Provides fast recovery
- Disadvantages
  - Cannot handle sudden, drastic lighting changes
  - Must have good Gaussian initialization (median filtering)
  - There are a number of parameters to tune

More Issues?

- Shadows detection
  - [Prati, Mikic, Trivedi, Cucchiara 2003]



- Chen & Aggarwal: The likelihood of a pixel being covered or uncovered is decided by the relative coordinates of optical flow vector vertices in its neighborhood.
- Oliver et al.: “Eigenbackgrounds" and its variations.
- Seki et al.: Image variations at neighboring image blocks have strong correlation.

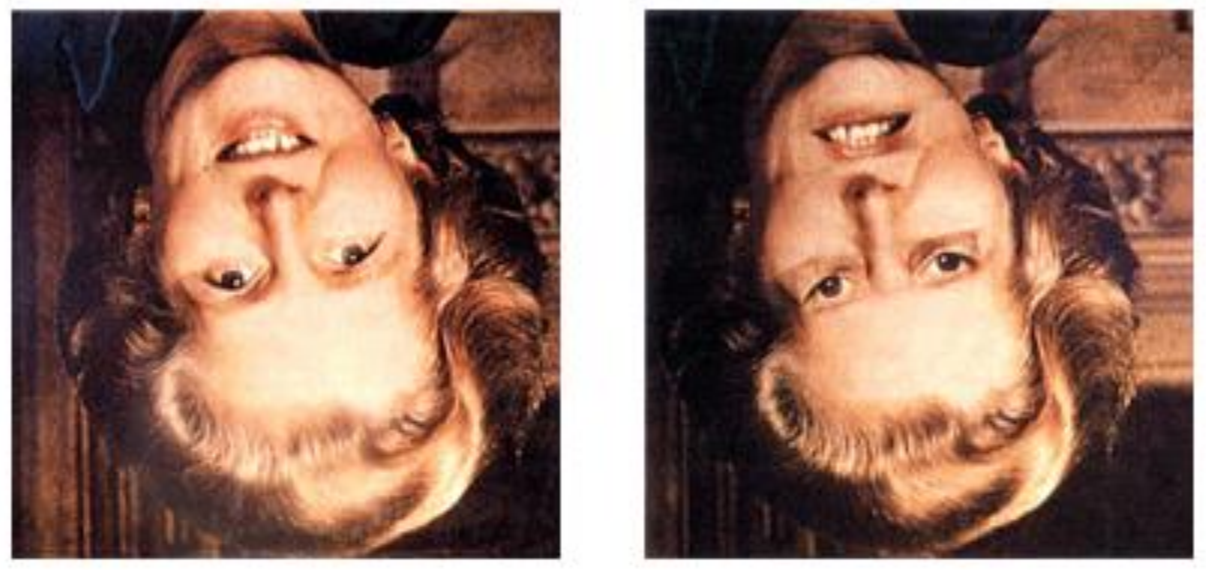
Simple Improvement

- Adaptive background mixture model + 3D connected component analysis [Goo et al.]
  - 3rd dimension is time
- Incorporate both spatial and temporal information into the background model

Summary

- Simple background subtraction approaches such as fame diff, mean, and median filtering are fast
  - Constant thresholds make them ill-suited for challenging real-world problems
- Adaptive background mixture model approach can handle challenging situations
  - Bimodal backgrounds, long-term scene changes, and repetitive motion
- Improvements include upgrade the approach with temporal information or using region-based techniques

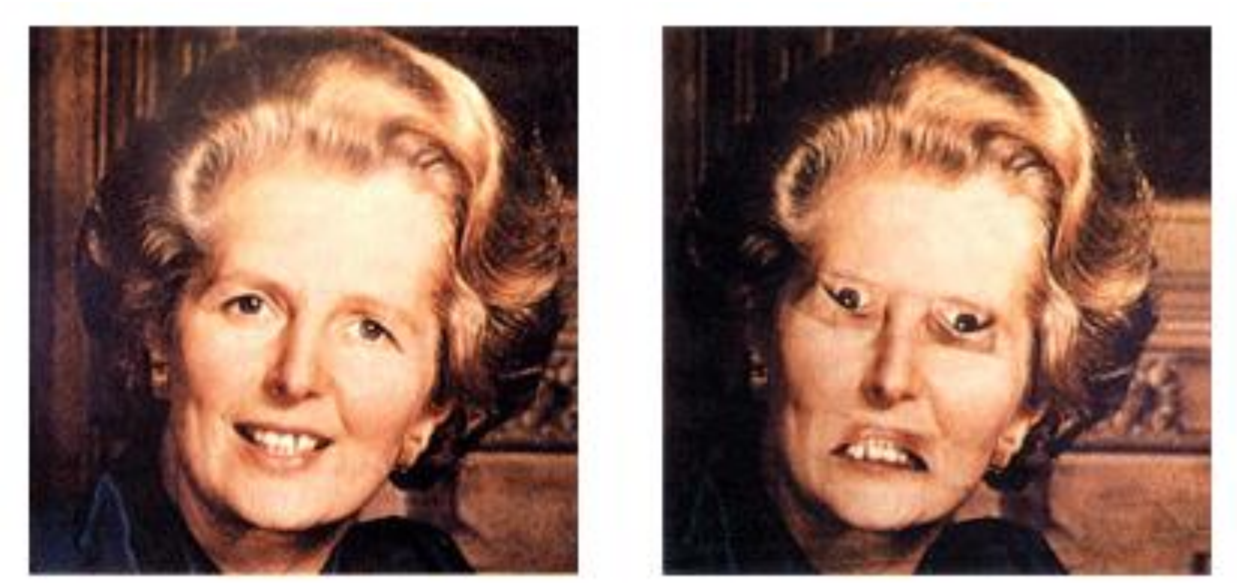
Object Recognition Introduction



The "Margaret Thatcher Illusion", by Peter Thompson

Steve Seitz

Object Recognition Introduction

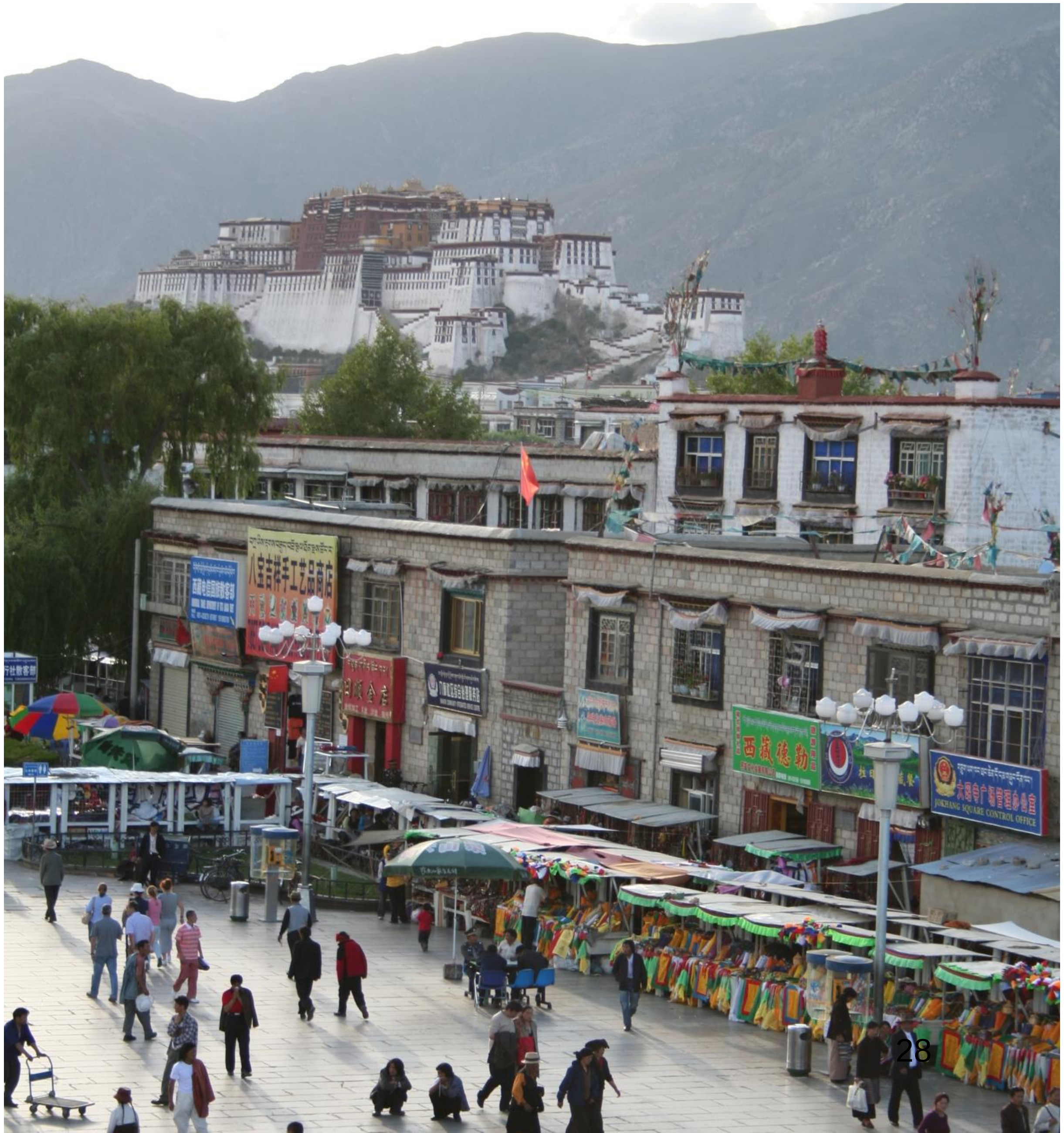


The “Margaret Thatcher Illusion”, by Peter Thompson

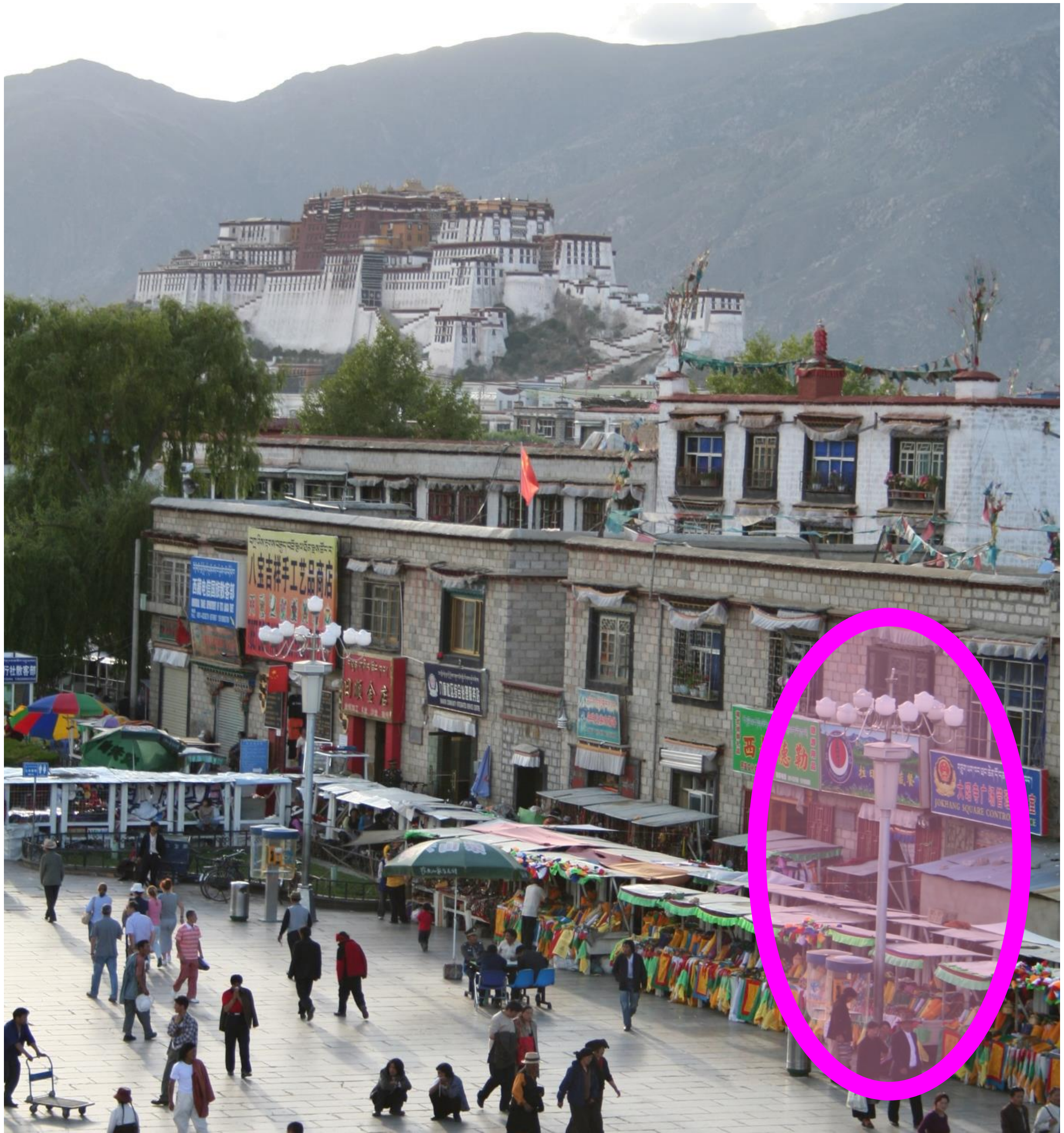
Steve Seitz

What do we mean by “object recognition”?

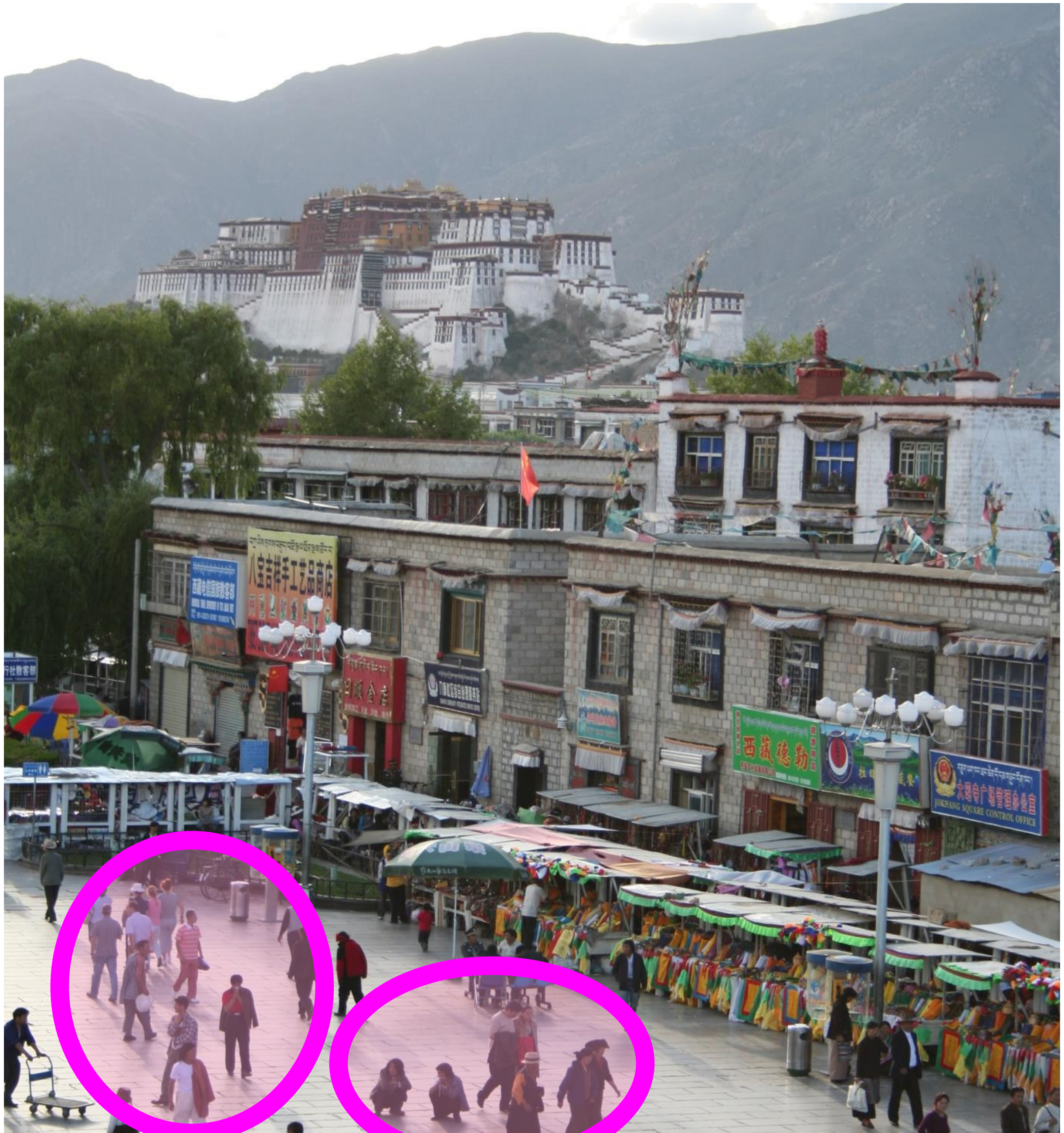
Next 15 slides adapted from Li, Fergus, & Torralba’s excellent short course on category and object recognition



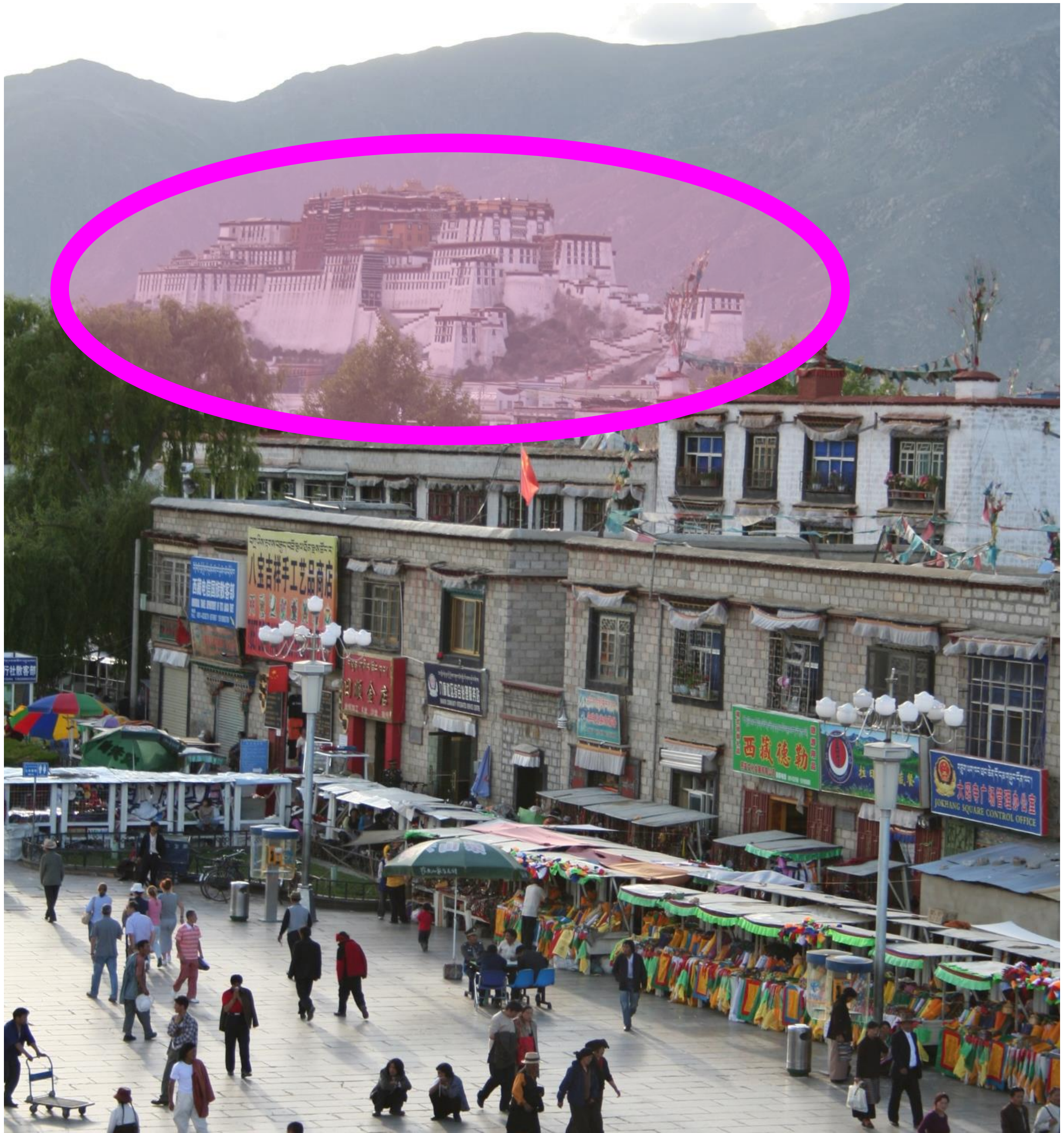
Verification: is that a lamp?



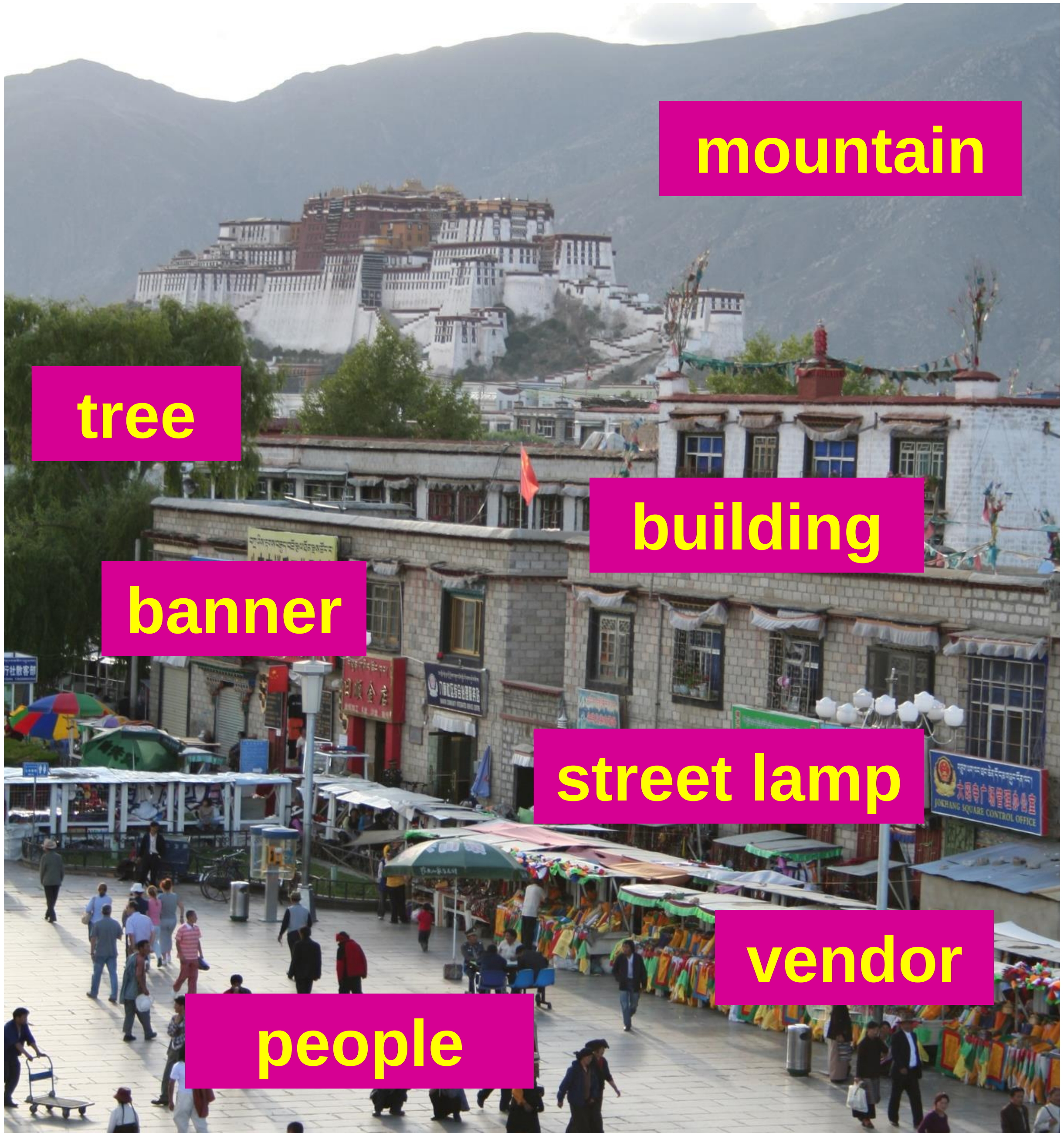
Detection: are there people?



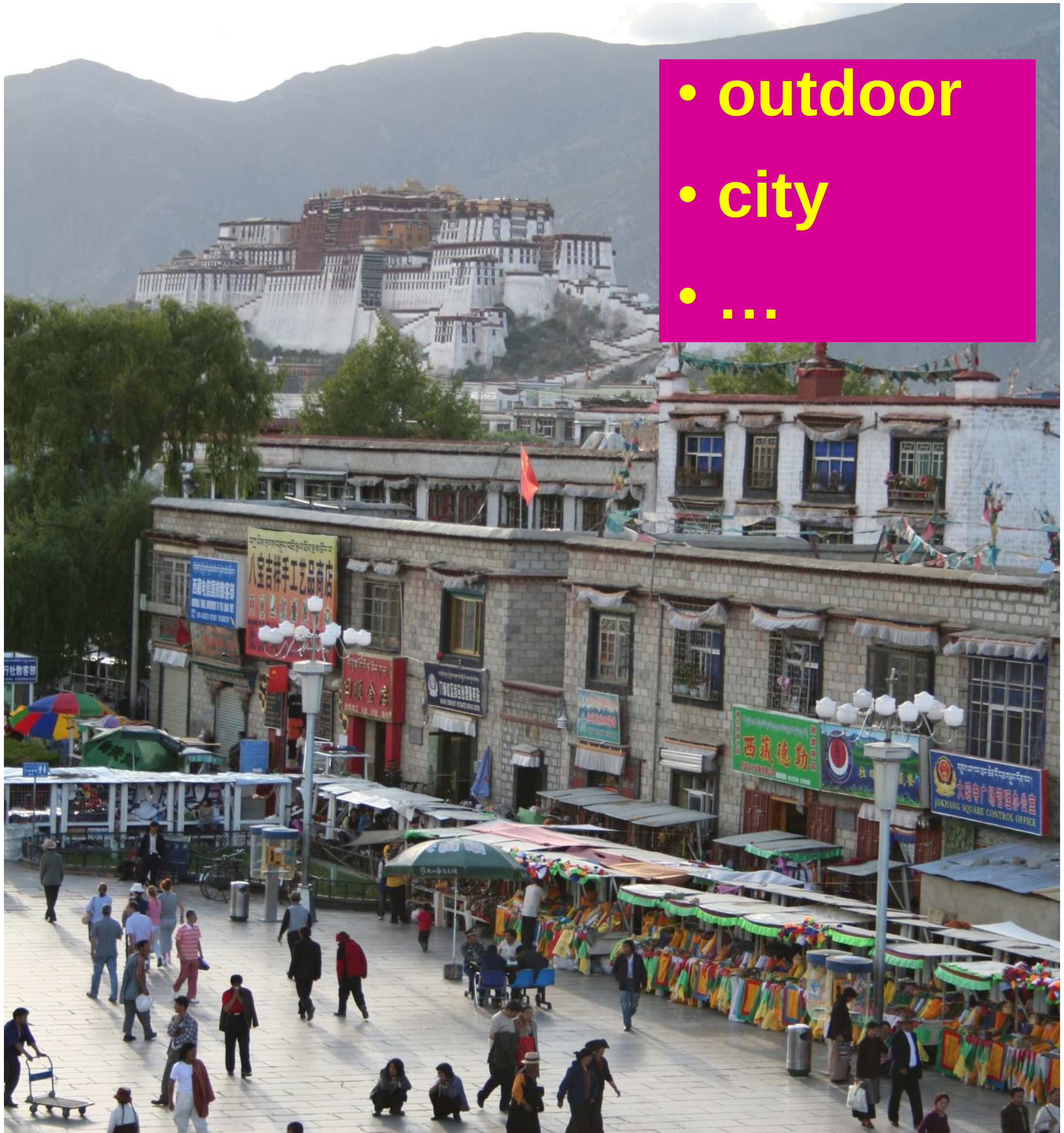
Identification: is that Potala Palace?



Object categorization



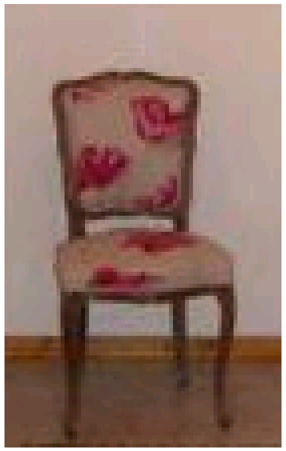
Scene and context categorization



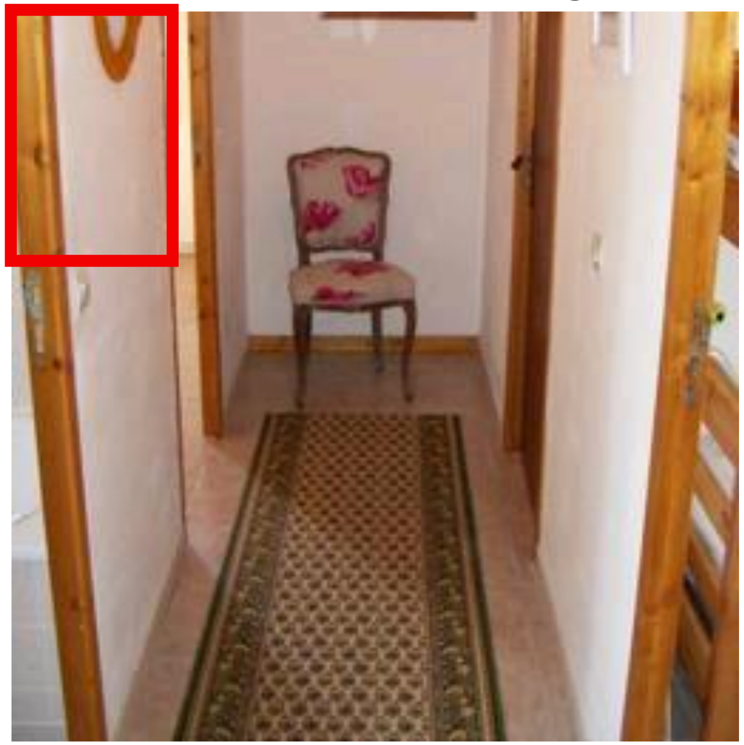
- outdoor
- city
- …

Object recognition
Is it really so hard?

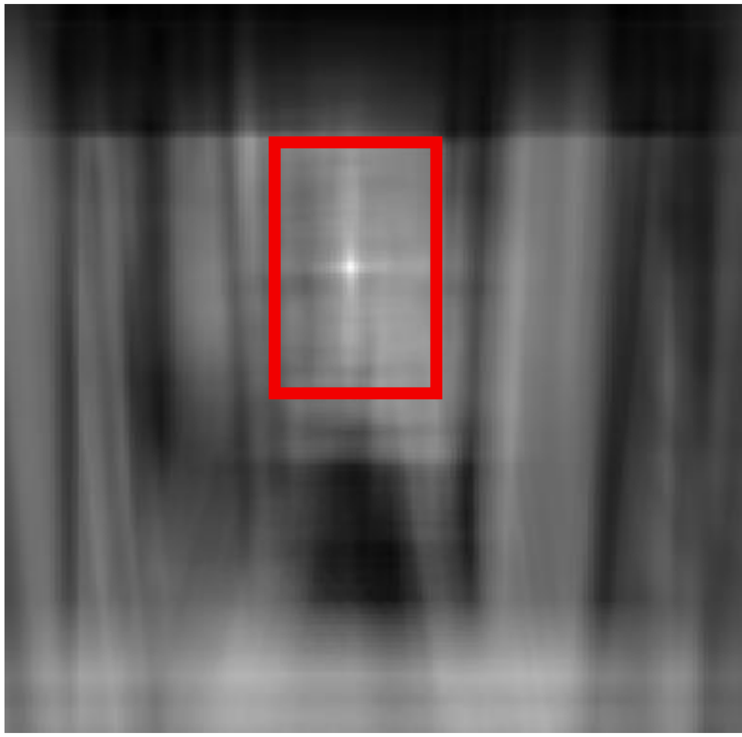
This is a chair

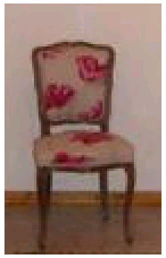


Find the chair in this image



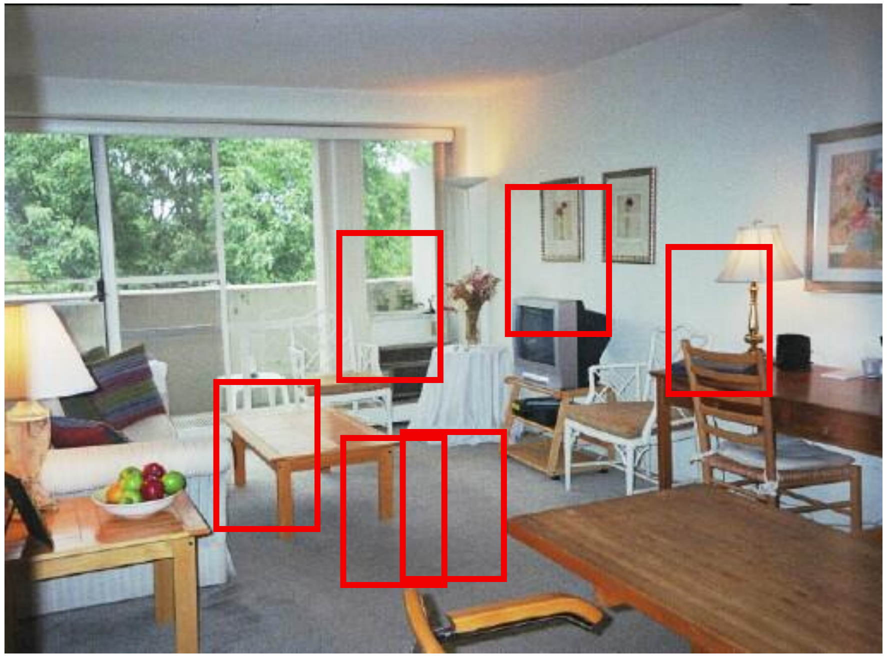
Output of normalized correlation

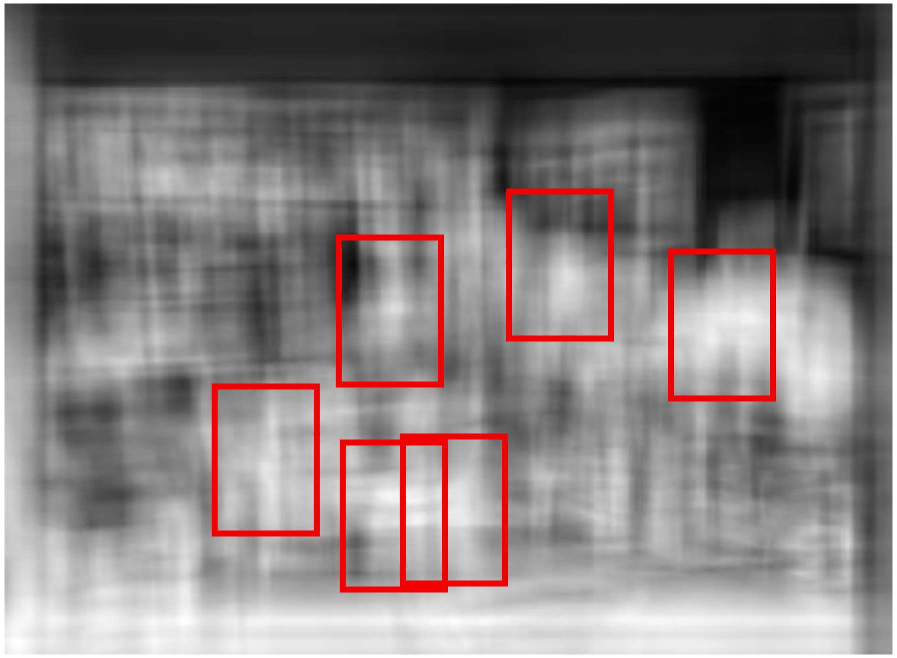




Object recognition Is it really so hard?

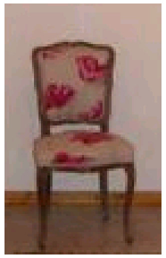
Find the chair in this image





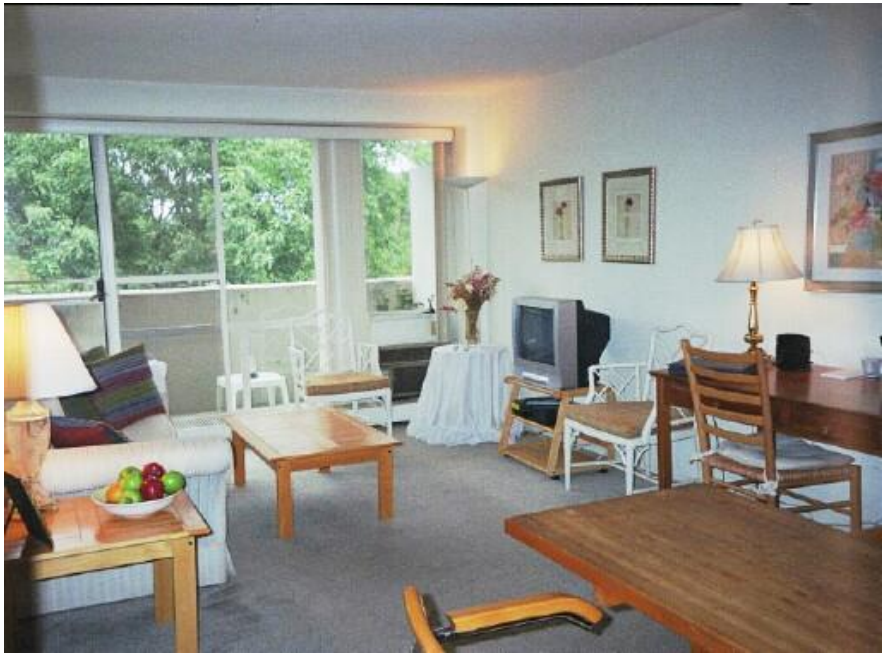
Pretty much garbage

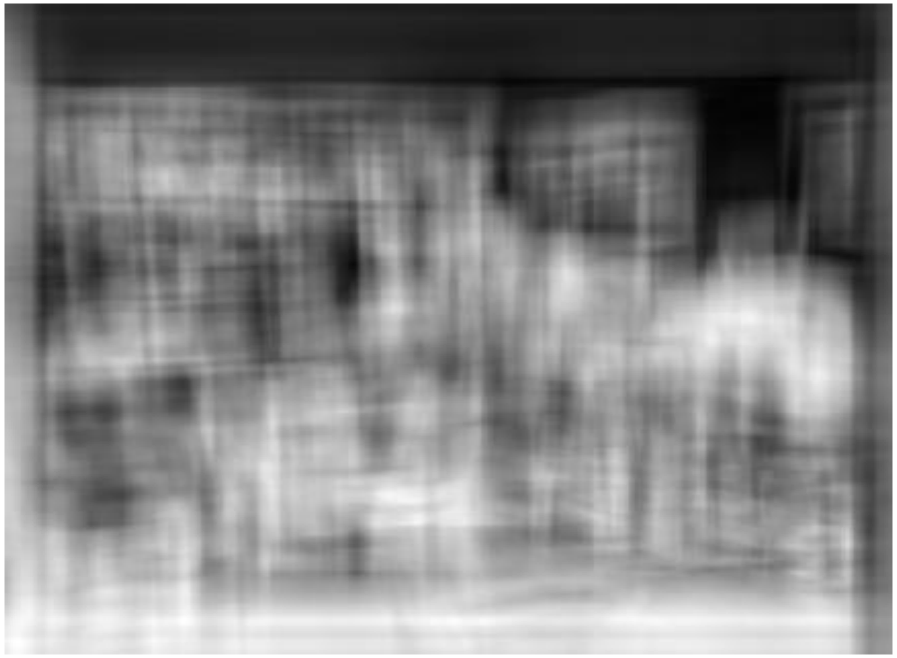
Simple template matching is not going to make it



Object recognition Is it really so hard?

Find the chair in this image





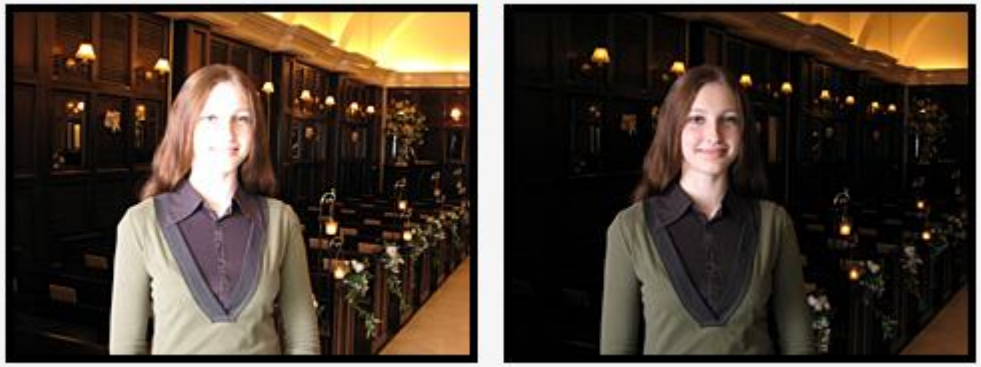
A “popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.” Nivatia & Binford, 1977.

Why not use SIFT matching for everything?

- Works well for object *instances*

- Not great for generic object *categories*

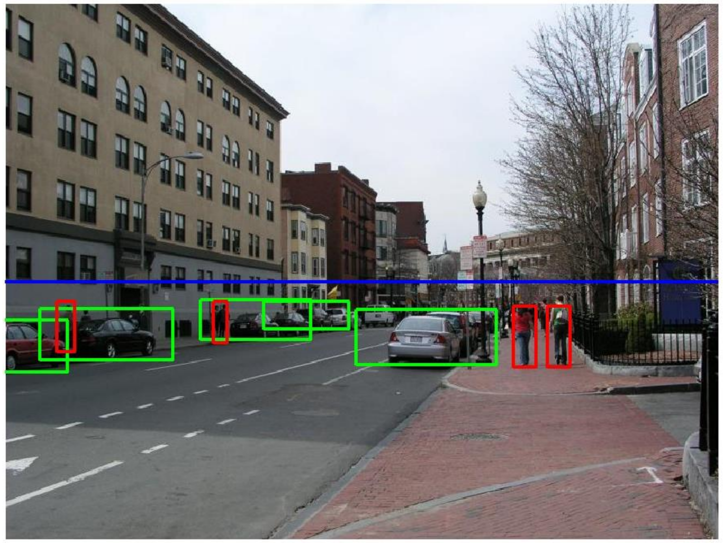
Applications: Computational photography

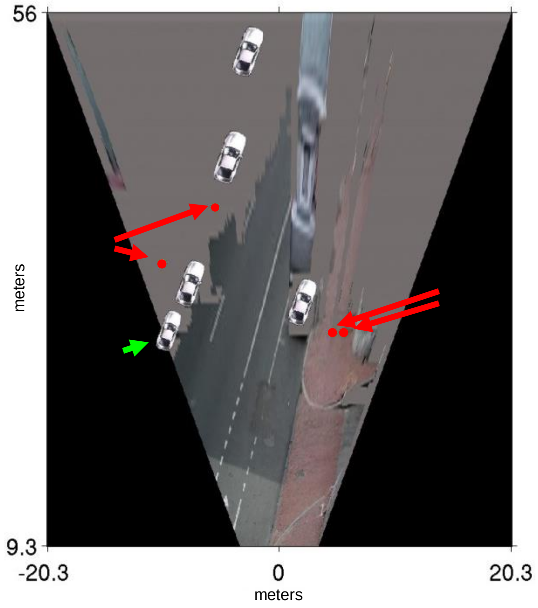


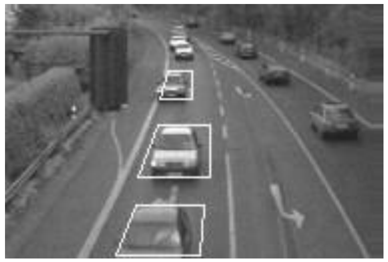
[Face priority AE] When a bright part of the face is too bright

Applications: Assisted driving

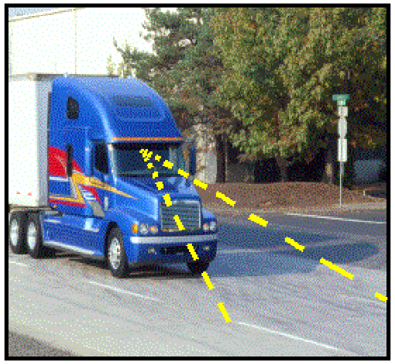
Pedestrian and car detection





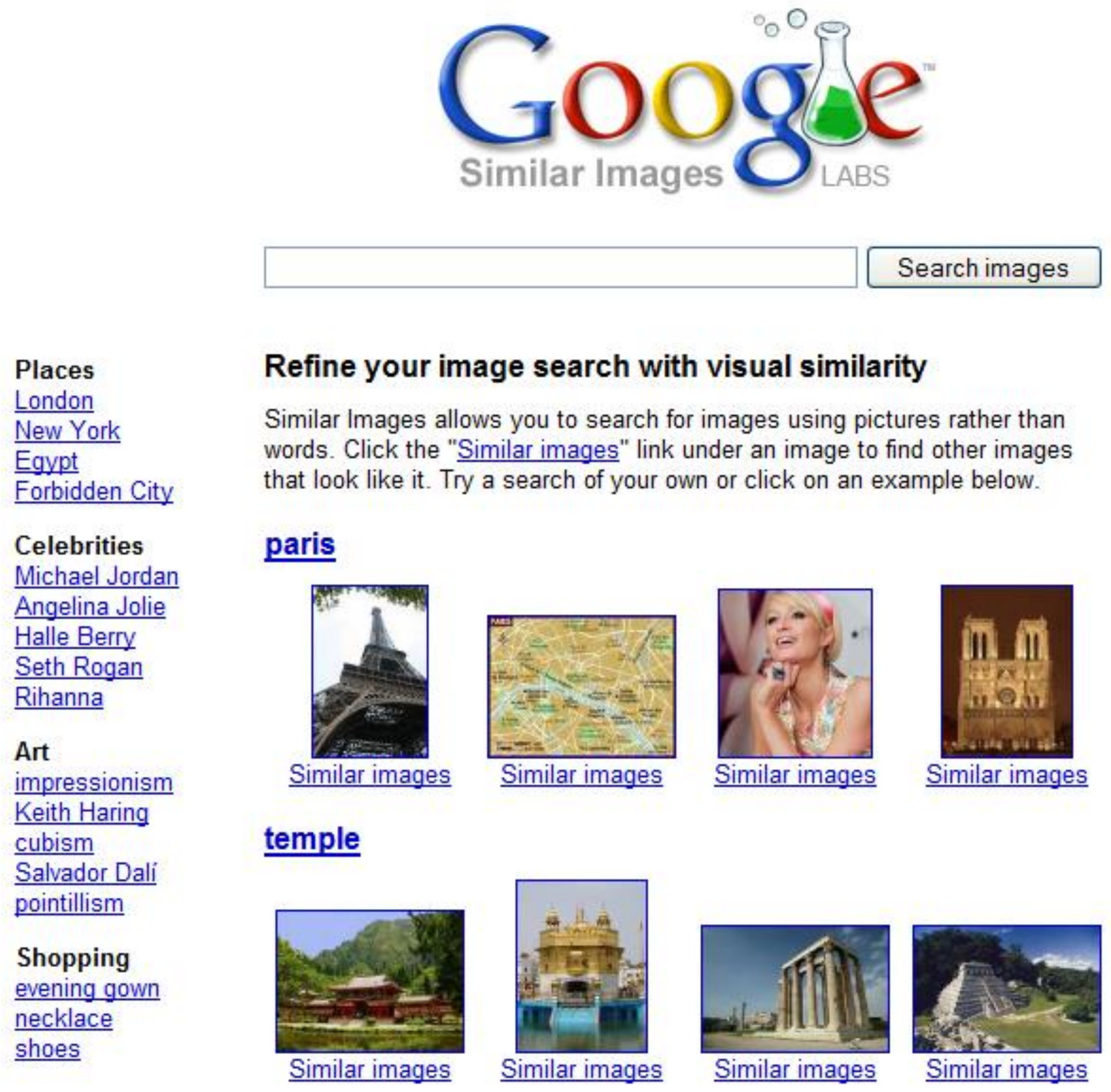


Lane detection



- Collision warning systems with adaptive cruise control,
- Lane departure warning systems,
- Rear object detection systems,

Applications: image search



Places
London
New York
Egypt
Forbidden City

Celebrities
Michael Jordan
Angelina Jolie
Halle Berry
Seth Rogan
Rihanna

Art
impressionism
Keith Haring
cubism
Salvador Dalí
pointillism

Shopping
evening gown
necklace
shoes

Refine your image search with visual similarity

Similar Images allows you to search for images using pictures rather than words. Click the "Similar images" link under an image to find other images that look like it. Try a search of your own or click on an example below.

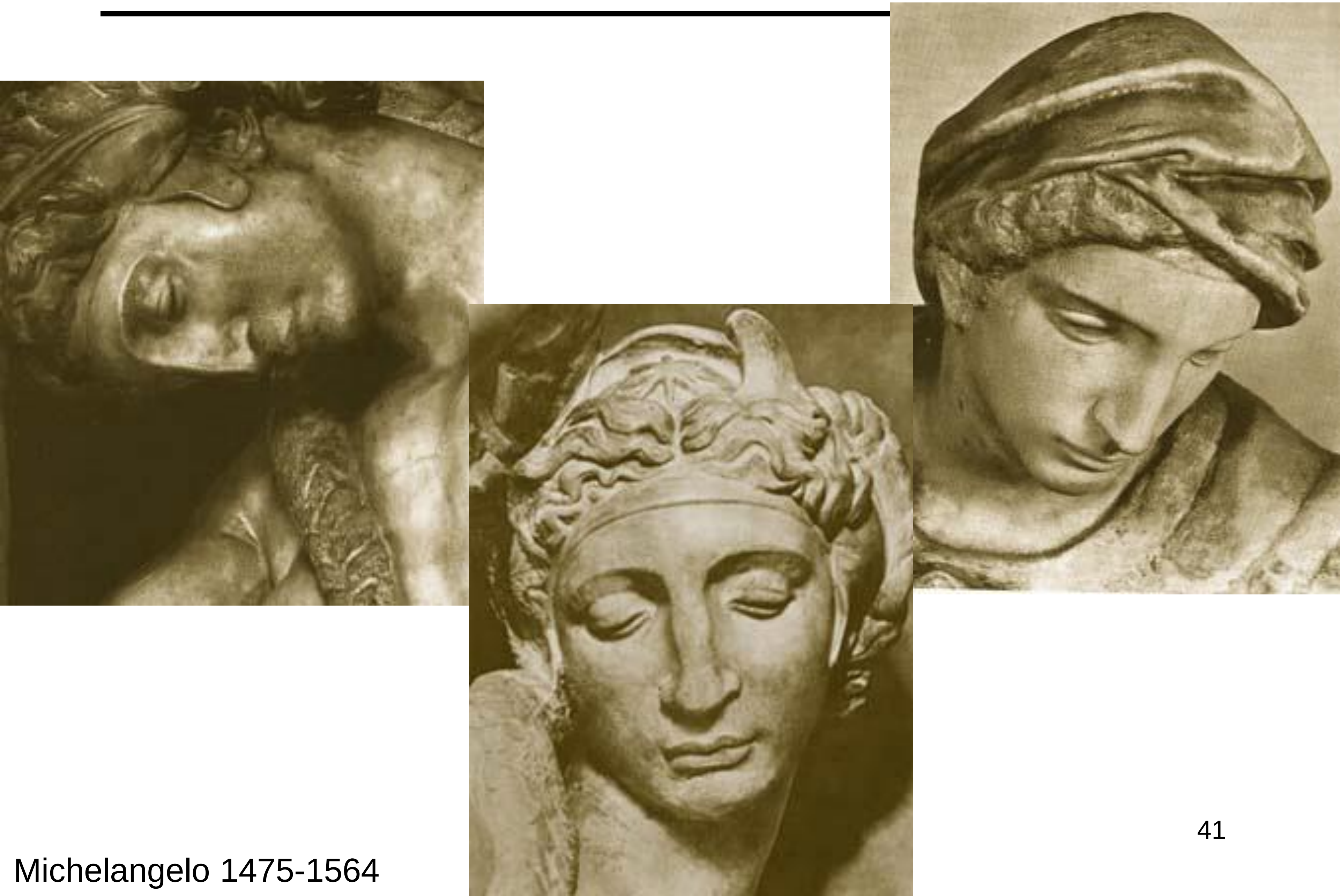
paris

Similar images Similar images Similar images Similar images

temple

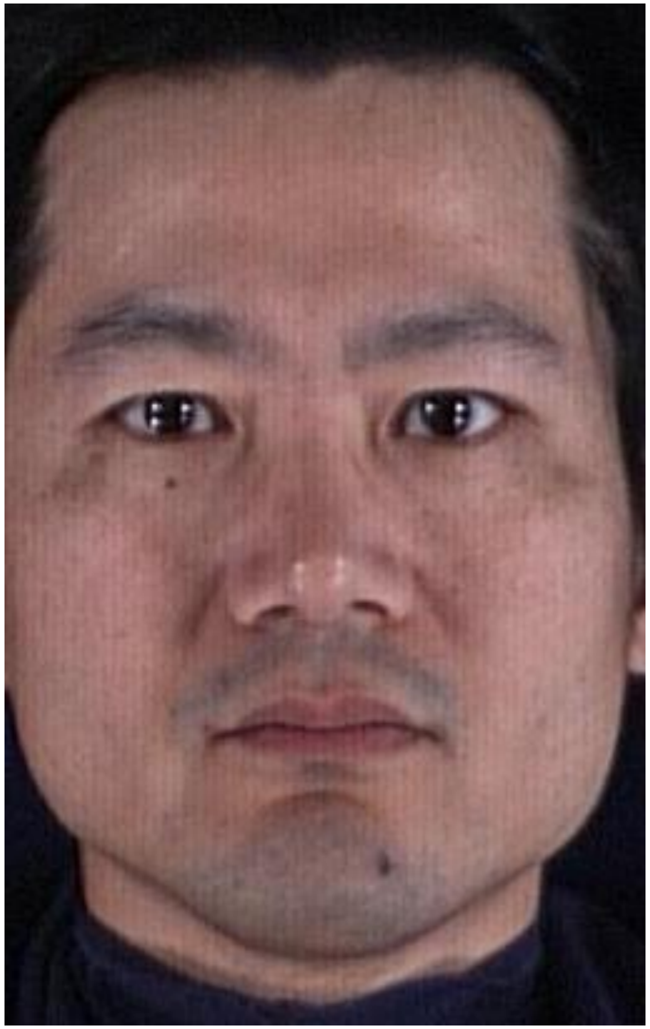
Similar images Similar images Similar images Similar images

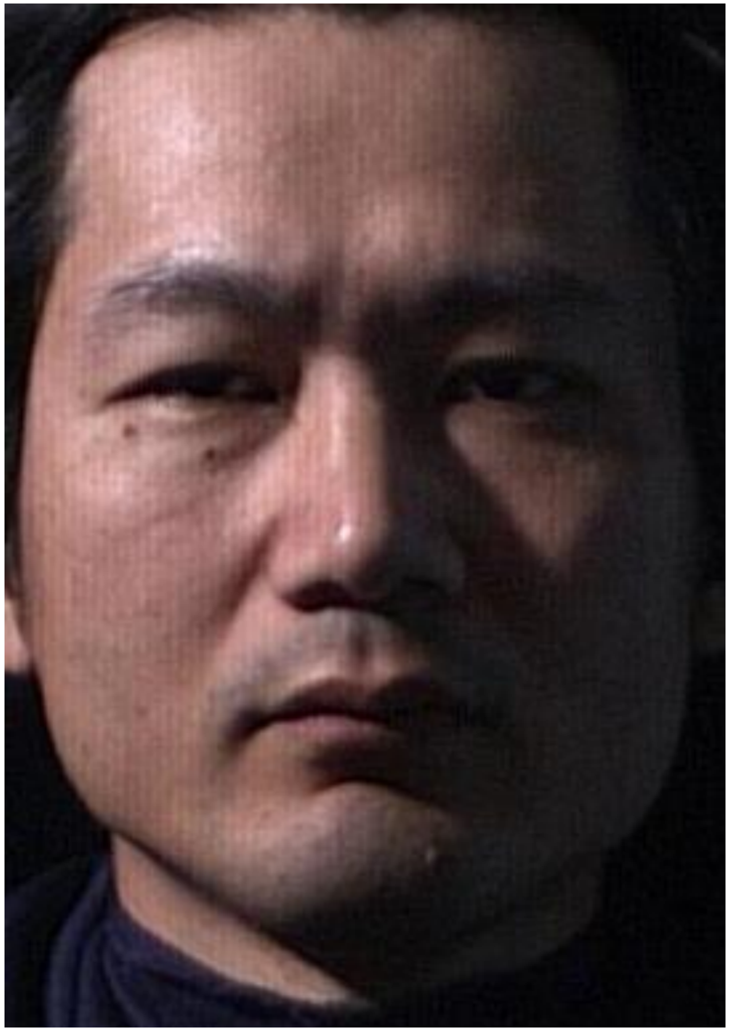
Challenges: viewpoint variation



Michelangelo 1475-1564

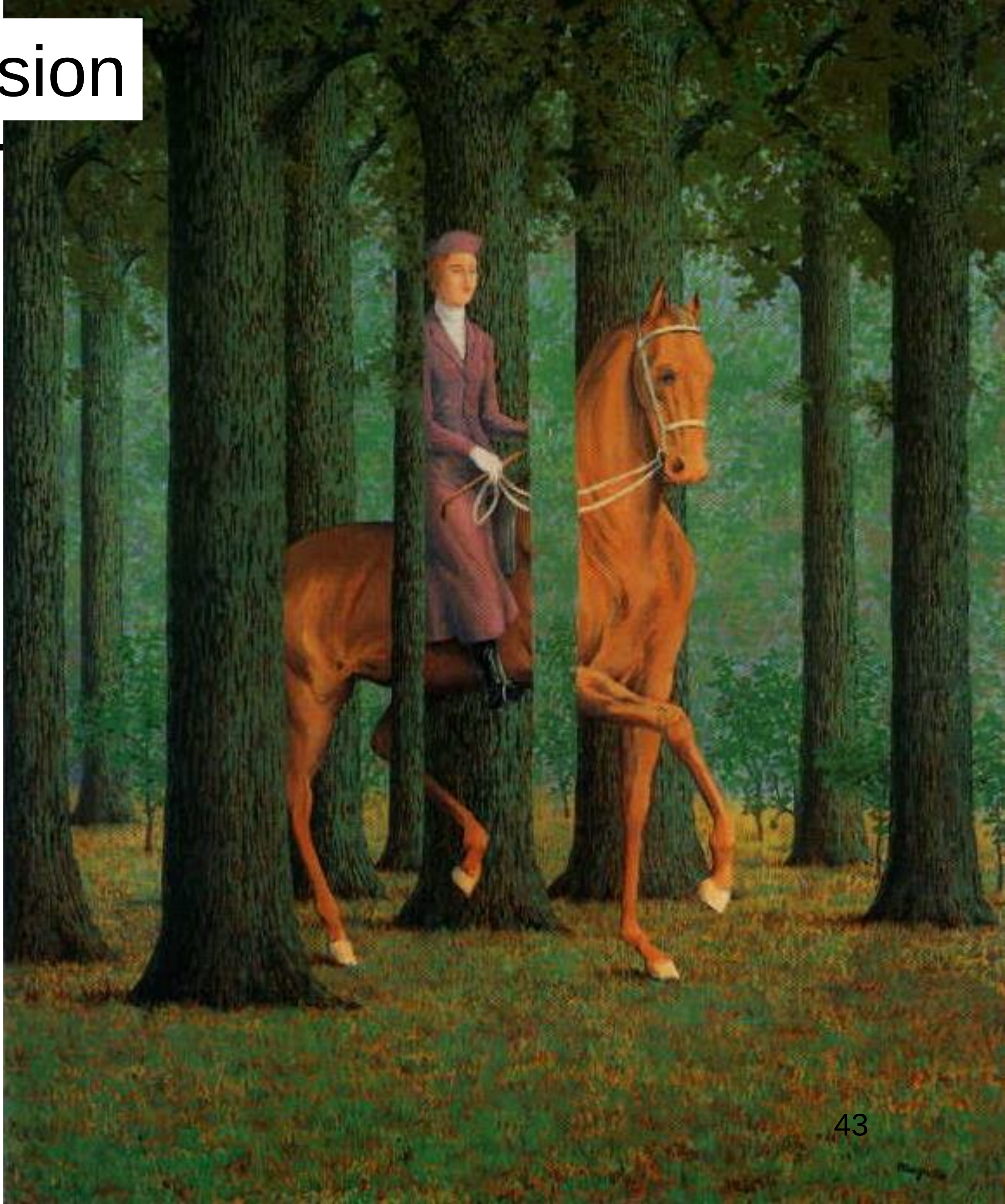
Challenges: illumination variation





slide credit: S. Ullman

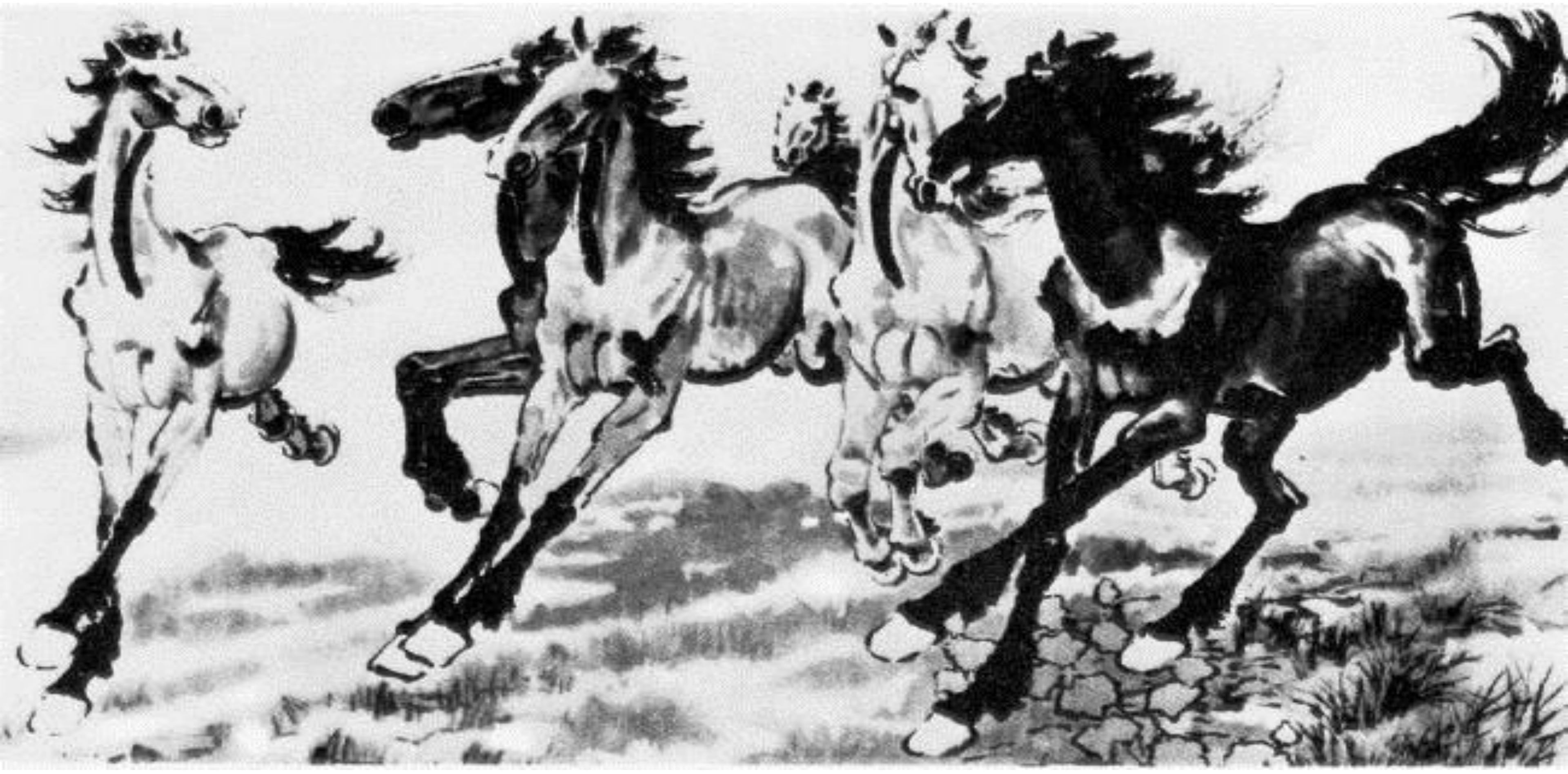
Challenges: occlusion



Magritte, 1957

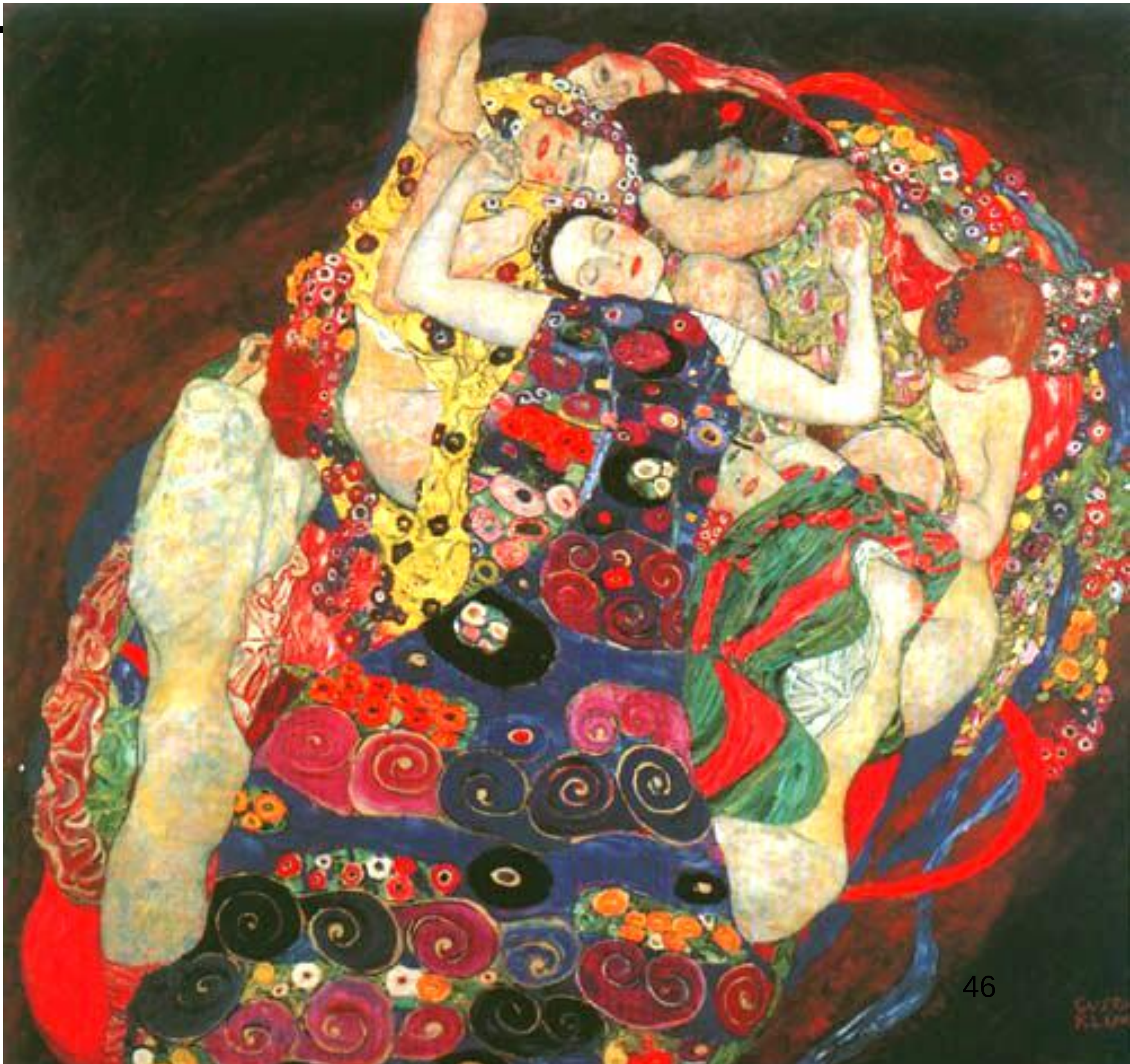
Challenges: scale

Challenges: deformation



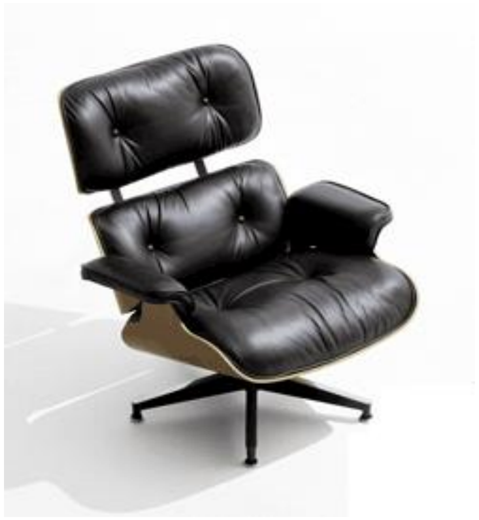
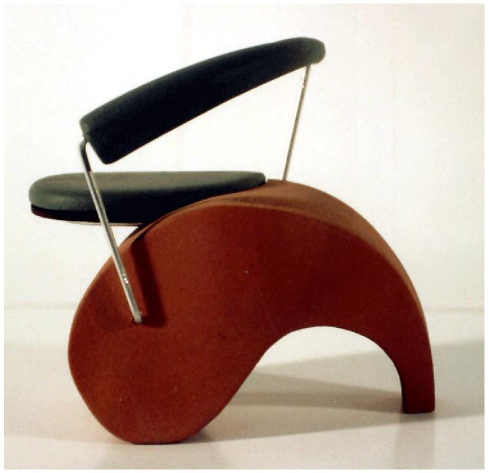
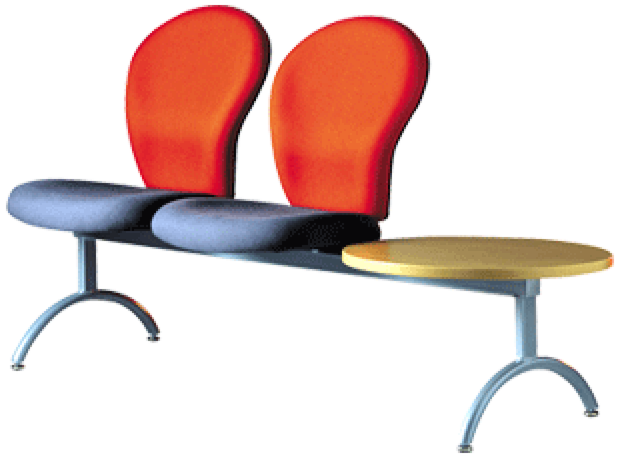
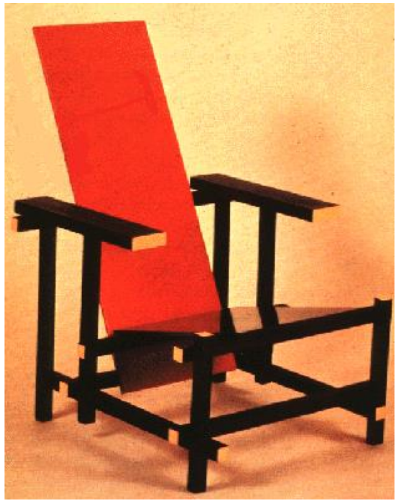
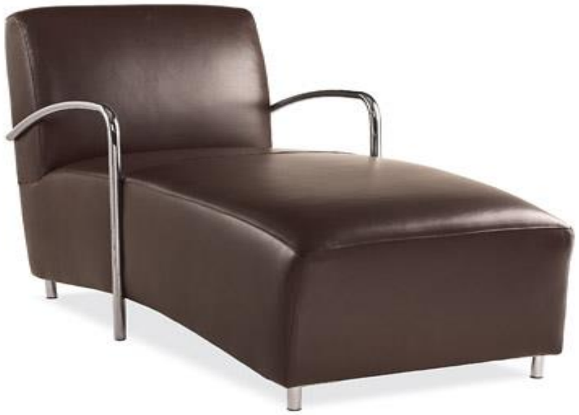
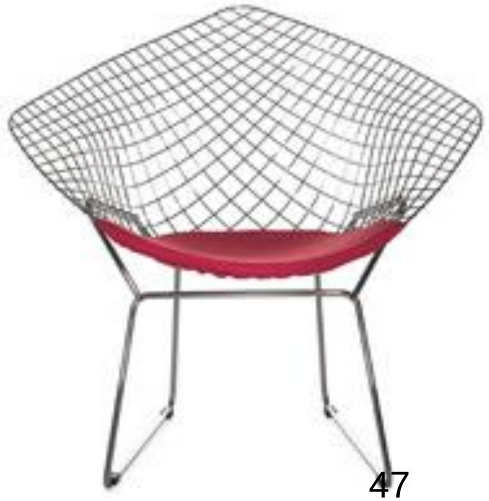
Xu, Beihong 1943

Challenges: background clutter



Klimt, 1913

Challenges: intra-class variation

Recognition problems

- What is it?
  - Object and scene recognition
- Who is it?
  - Identity recognition
- Where is it?
  - Object detection
- What are they doing?
  - Activities
- All of these are **classification** problems
  - Choose one class from a list of possible candidates

What is recognition?

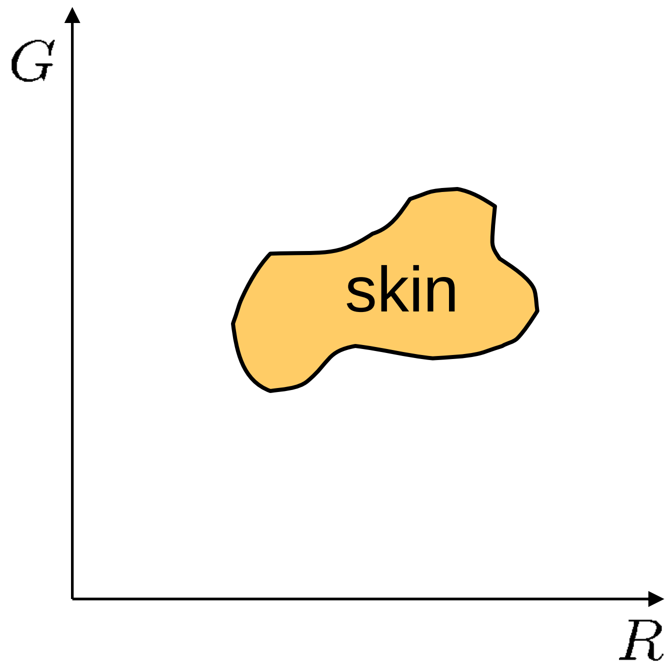
- A different taxonomy from [Csurka *et al.* 2006]:
- Recognition
  - Where is *this* particular object?
- Categorization
  - What *kind* of object(s) is(are) present?
- Content-based image retrieval
  - Find me something that looks similar
- Detection
  - Locate *all* instances of a given class

Steve Seitz

Face detection

How to tell if a face is present?

One simple method: skin detection



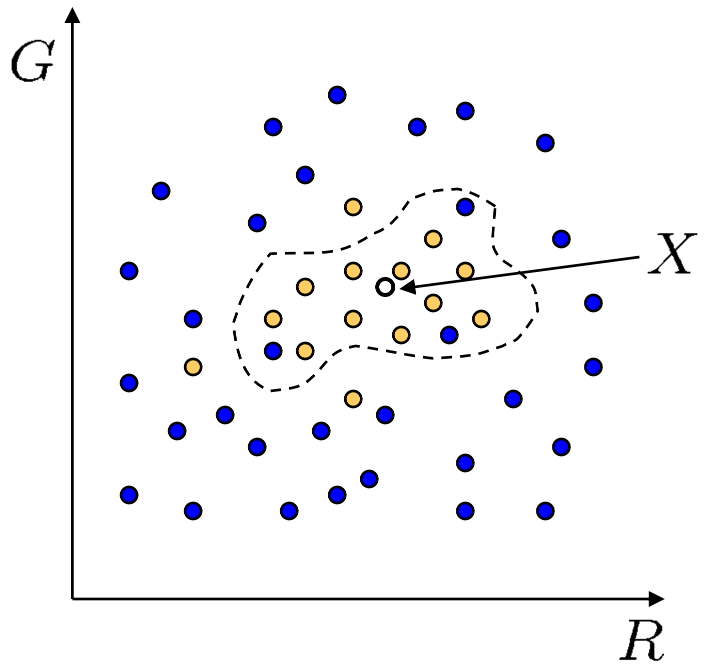
Skin pixels have a distinctive range of colors

- Corresponds to region(s) in RGB color space
  - for visualization, only R and G components are shown above

Skin classifier

- A pixel X = (R,G,B) is skin if it is in the skin region
- But how to find this region?

Skin detection



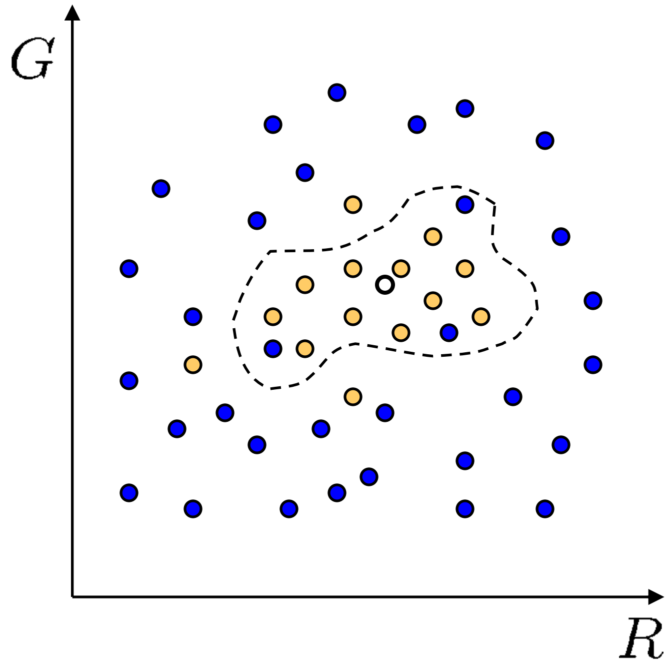
Learn the skin region from examples

- Manually label pixels in one or more “training images” as skin or not skin
- Plot the training data in RGB space
  - skin pixels shown in orange, non-skin pixels shown in blue
  - some skin pixels may be outside the region, non-skin pixels inside. Why?

Skin classifier

- Given X = (R,G,B): how to determine if it is skin or not?

Skin classification techniques



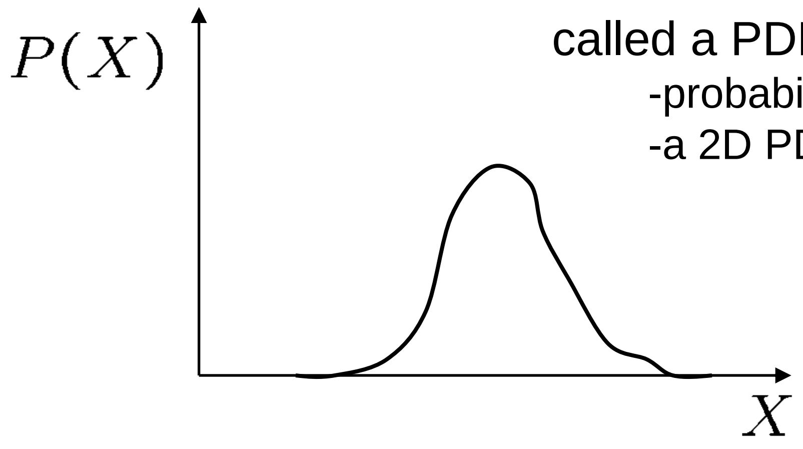
Skin classifier

- Given X = (R,G,B): how to determine if it is skin or not?
- Nearest neighbor
  - find labeled pixel closest to X
  - choose the label for that pixel
- Data modeling
  - fit a model (curve, surface, or volume) to each class
- Probabilistic data modeling
  - fit a probability model to each class

Probability

Basic probability

- X is a random variable
- P(X) is the probability that X achieves a certain value



called a PDF

-probability distribution/density function

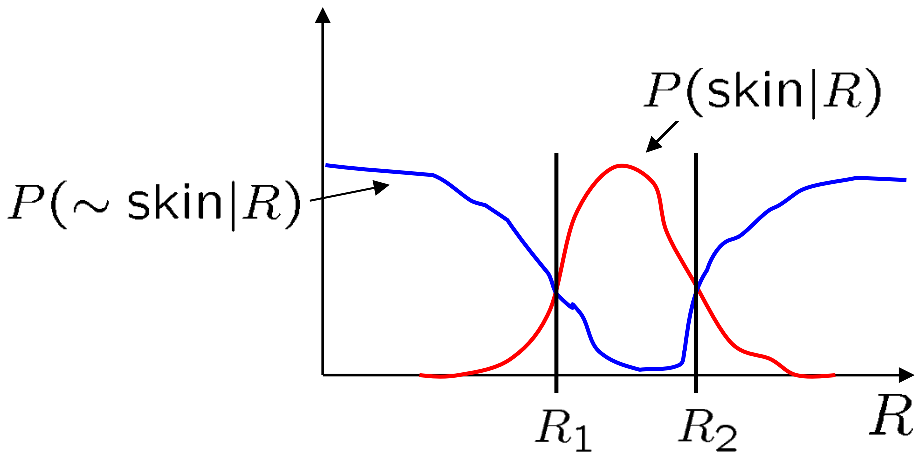
-a 2D PDF is a surface, 3D PDF is a volume

- $0 \leq P(X) \leq 1$
- $\int_{-\infty}^{\infty} P(X)dX = 1$ or $\sum P(X) = 1$

continuous X discrete X

- Conditional probability: P(X | Y)
  – probability of X given that we already know Y

Probabilistic skin classification



Now we can model uncertainty

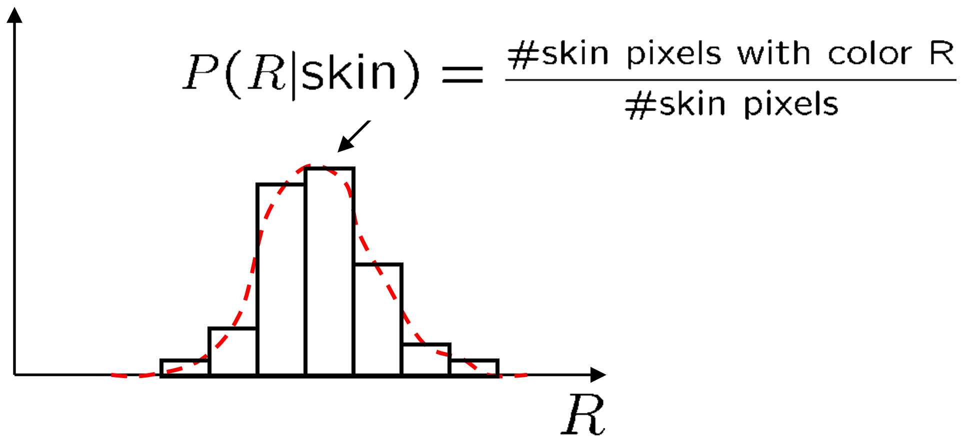
- Each pixel has a probability of being skin or not skin
  - $P(\sim \text{skin}|R) = 1 - P(\text{skin}|R)$

Skin classifier

- Given X = (R,G,B): how to determine if it is skin or not?
- Choose interpretation of highest probability
  - set X to be a skin pixel if and only if $R_1 < X \leq R_2$

Where do we get $P(\text{skin}|R)$ and $P(\sim \text{skin}|R)$?

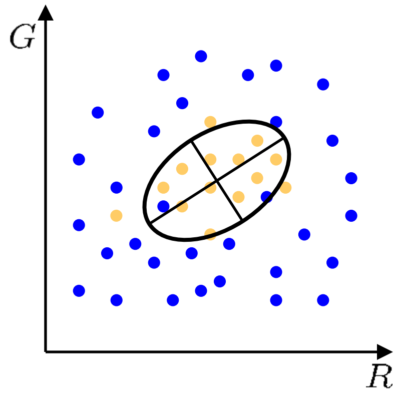
Learning conditional PDF's



We can calculate P(R | skin) from a set of training images

- It is simply a histogram over the pixels in the training images
  - each bin R_i contains the proportion of skin pixels with color R_i

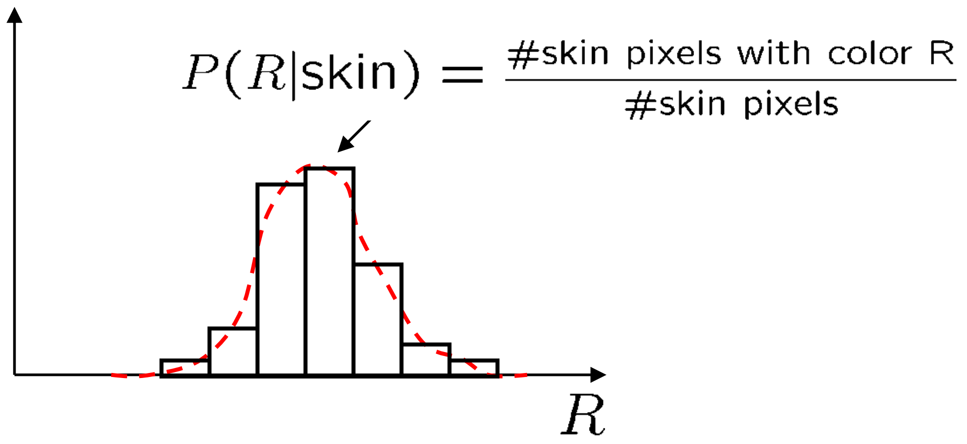
This doesn't work as well in higher-dimensional spaces. Why not?



Approach: fit parametric PDF functions

- common choice is rotated Gaussian
  - center $\mathbf{c} = \overline{X}$
  - covariance $\sum_X (X - \overline{X})(X - \overline{X})^T$
    - » orientation, size defined by eigenvecs, eigenvals

Learning conditional PDF's



We can calculate P(R | skin) from a set of training images

- It is simply a histogram over the pixels in the training images
  - each bin R_i contains the proportion of skin pixels with color R_i

But this isn't quite what we want

- Why not? How to determine if a pixel is skin?
- We want P(skin | R) not P(R | skin)
- How can we get it?

Bayes rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$$

In terms of our problem:

what we measure
(**likelihood**)

domain knowledge
(**prior**)

$$P(\text{skin}|R) = \frac{P(R|\text{skin})\ P(\text{skin})}{P(R)}$$

what we want
(**posterior**)

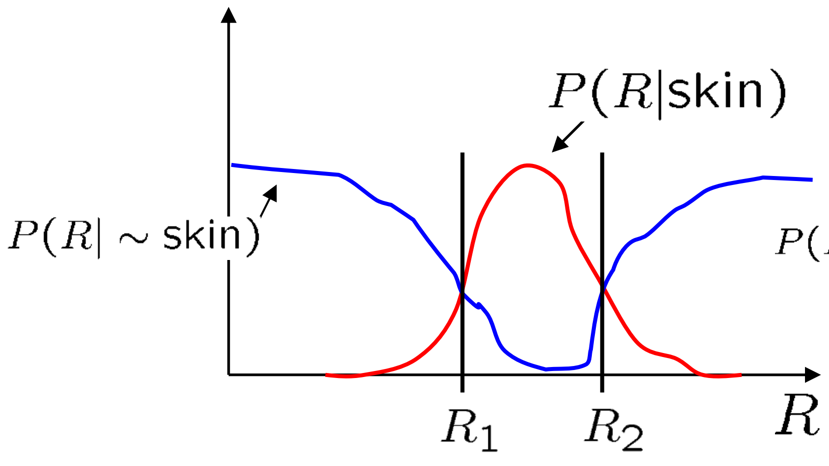
normalization term

$$P(R) = P(R|\text{skin})P(\text{skin}) + P(R|\sim\text{skin})P(\sim\text{skin})$$

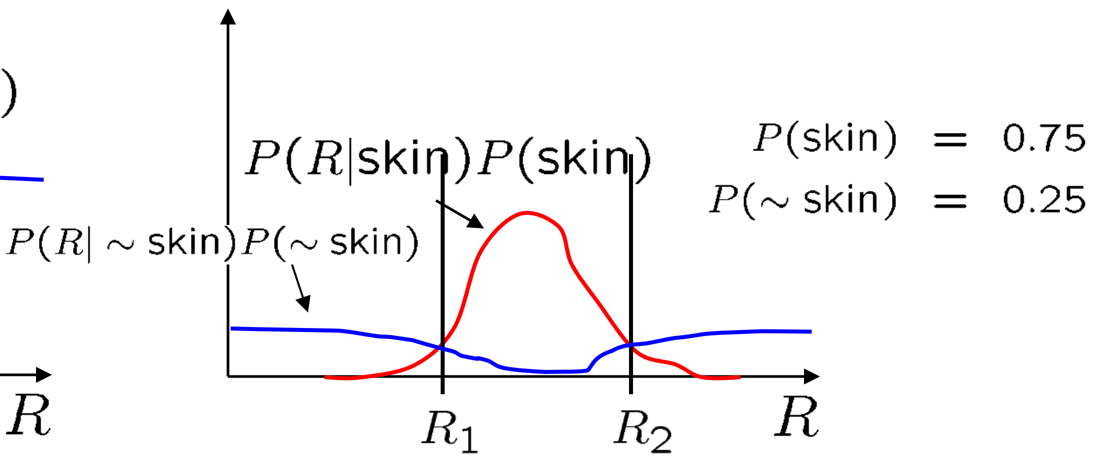
The prior: P(skin)

- Could use domain knowledge
  - P(skin) may be larger if we know the image contains a person
  - for a portrait, P(skin) may be higher for pixels in the center
- Could learn the prior from the training set. How?
  - P(skin) may be proportion of skin pixels in training set

Bayesian estimation



likelihood



posterior (unnormalized)

Bayesian estimation $\quad$ = minimize probability of misclassification

- Goal is to choose the label (skin or ~skin) that maximizes the posterior
  - this is called **Maximum A Posteriori (MAP) estimation**
- Suppose the prior is uniform: P(skin) = P(~skin) = 0.5
  - in this case $P(\text{skin}|R) = cP(R|\text{skin}), \quad P(\sim\text{skin}|R) = cP(R|\sim\text{skin})$
  - maximizing the posterior is equivalent to maximizing the likelihood
    - » $P(\text{skin}|R) > P(\sim\text{skin}|R)$ if and only if $P(R|\text{skin}) > P(R|\sim\text{skin})$
  - this is called **Maximum Likelihood (ML) estimation**

Skin detection results

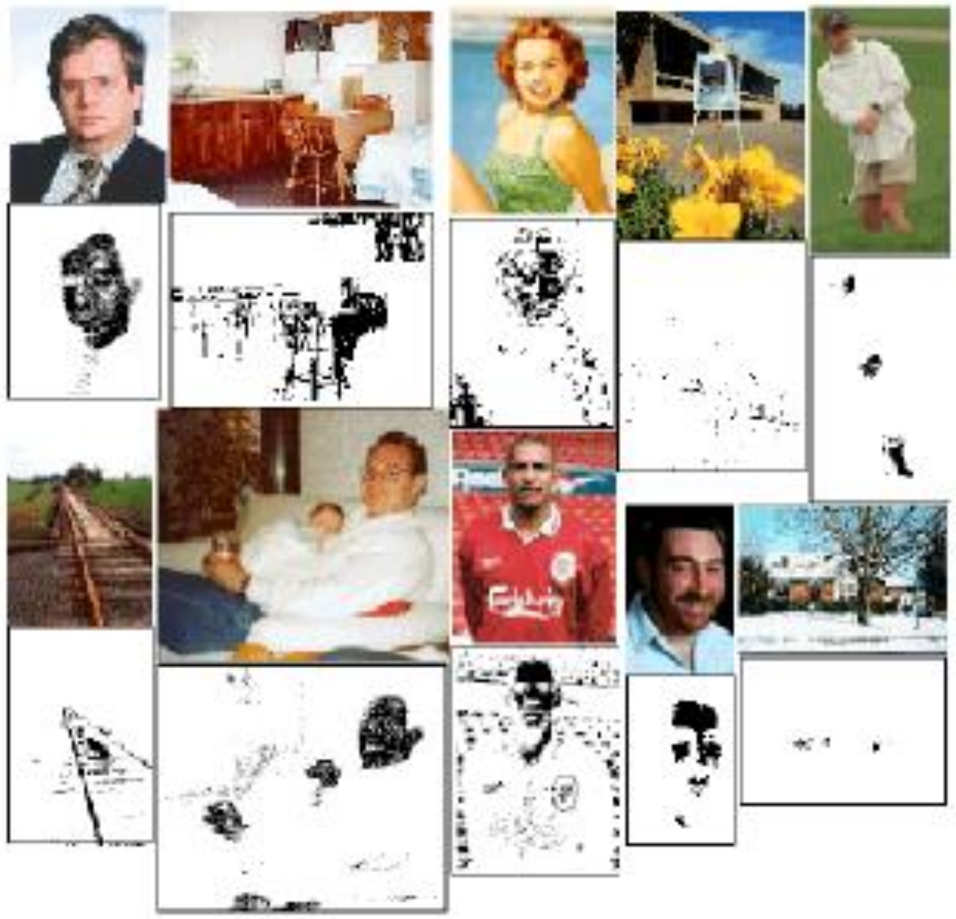
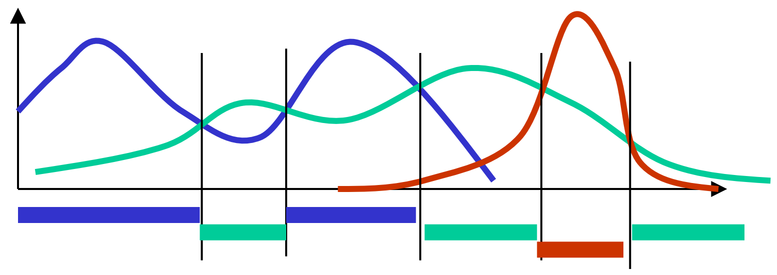


Figure 25.3. The figure shows a variety of images together with the output of the skin detector of Jones and Rehg applied to the image. Pixels marked black are skin pixels, and white are background. Notice that this process is relatively effective, and could certainly be used to focus attention on, say, faces and hands. *Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 © 1999, IEEE*

General classification

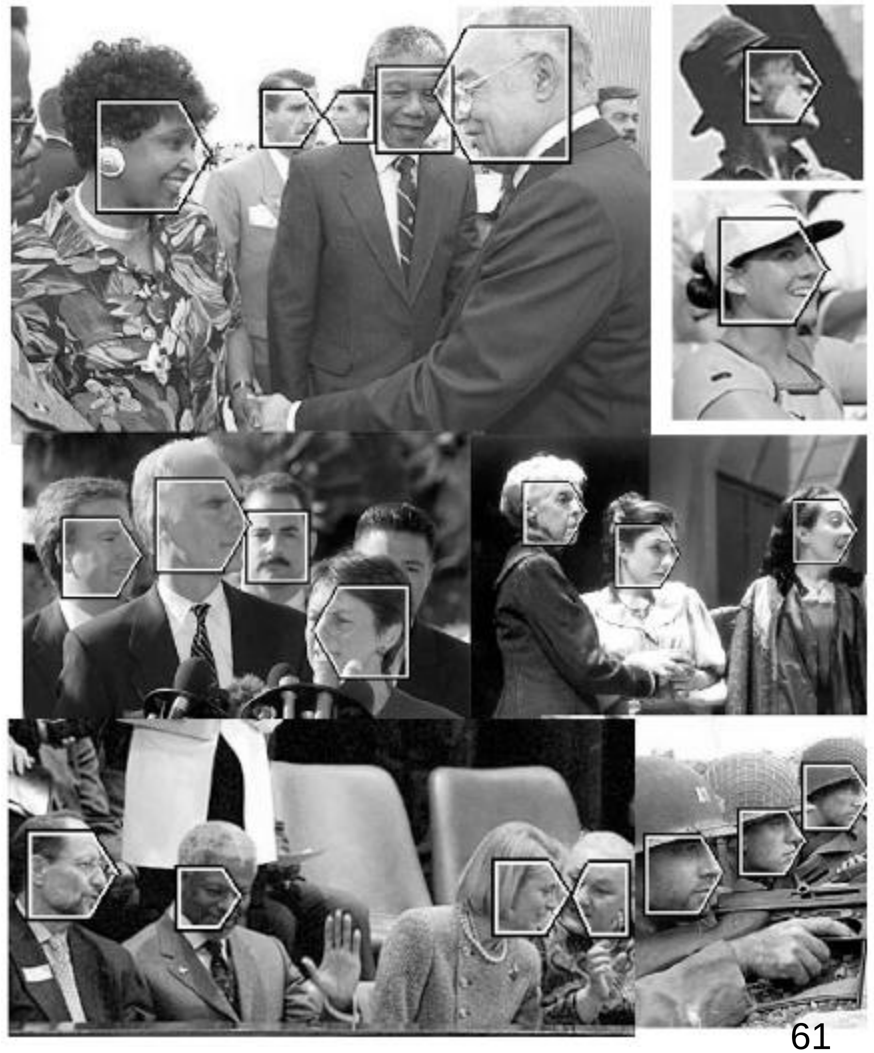
This same procedure applies in more general circumstances

- More than two classes
- More than one dimension



Example: face detection

- Here, X is an image region
  - dimension = # pixels
  - each face can be thought of as a point in a high dimensional space



H. Schneiderman and T.Kanade

H. Schneiderman, T. Kanade. "A Statistical Method for 3D Object Detection Applied to Faces and Cars". IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000) http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/hws/www/CVPR00.pdf