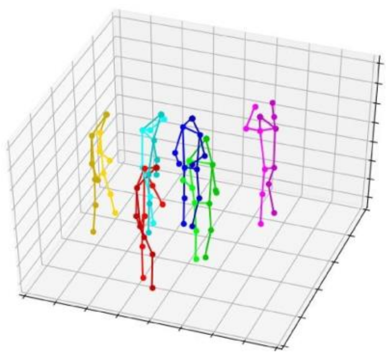
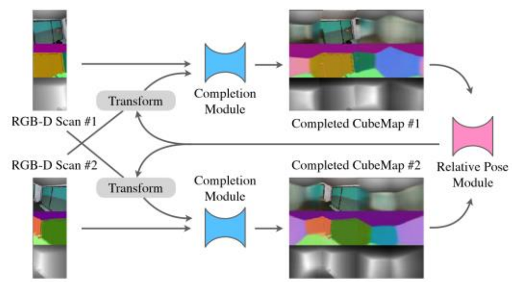
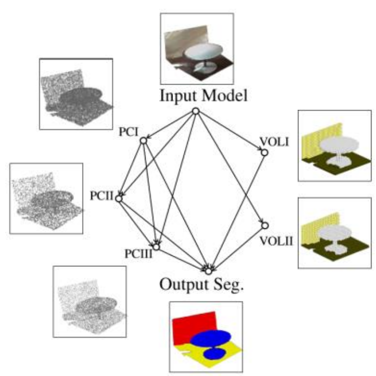
CS376 Computer Vision
Lecture 4: Binary Image Analysis





Qixing Huang

Feb. 4th 2019



Last Lecture

- Image gradients
- Seam Carving
- Edge detector

Image gradients (smoothing + gradient)

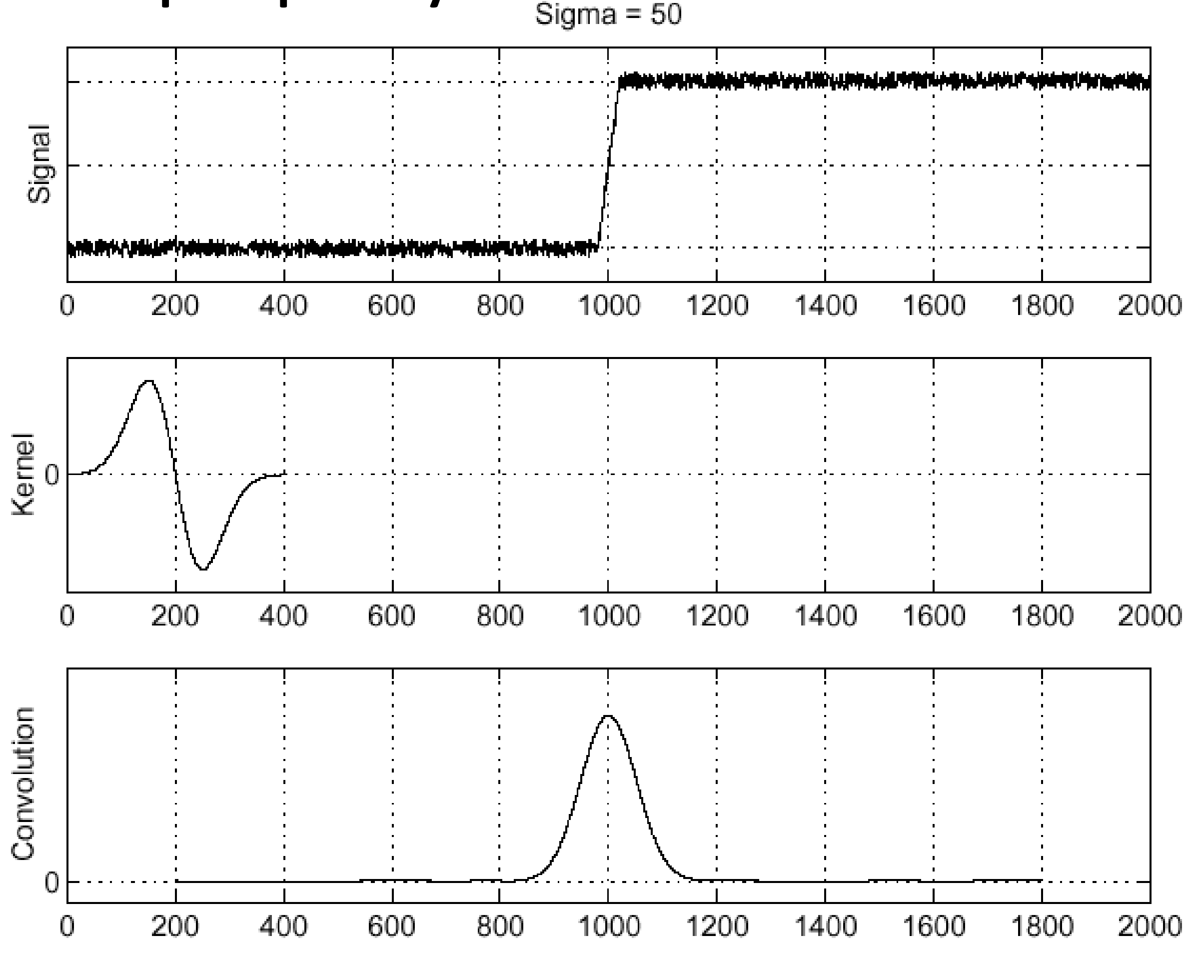
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

Differentiation property of convolution.

f

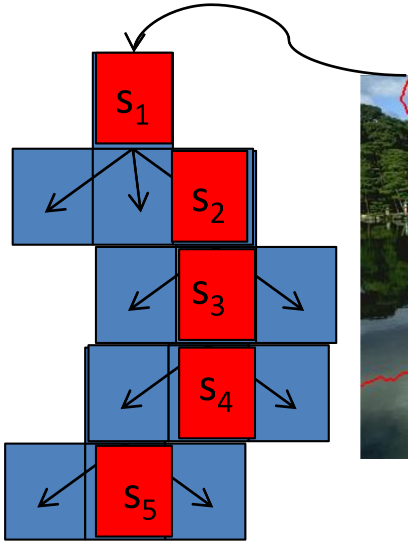
$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



Slide credit Steve Seitz

Seam carving: algorithm



$$Energy(f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Let a vertical seam **s** consist of *h* positions that form an 8-connected path.

Let the cost of a seam be:

Optimal seam minimizes this cost:

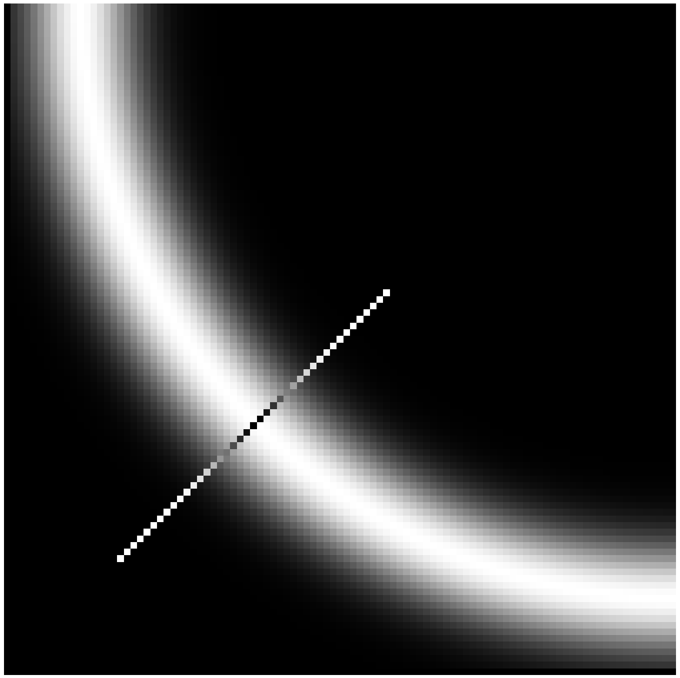
Compute it efficiently with dynamic programming.

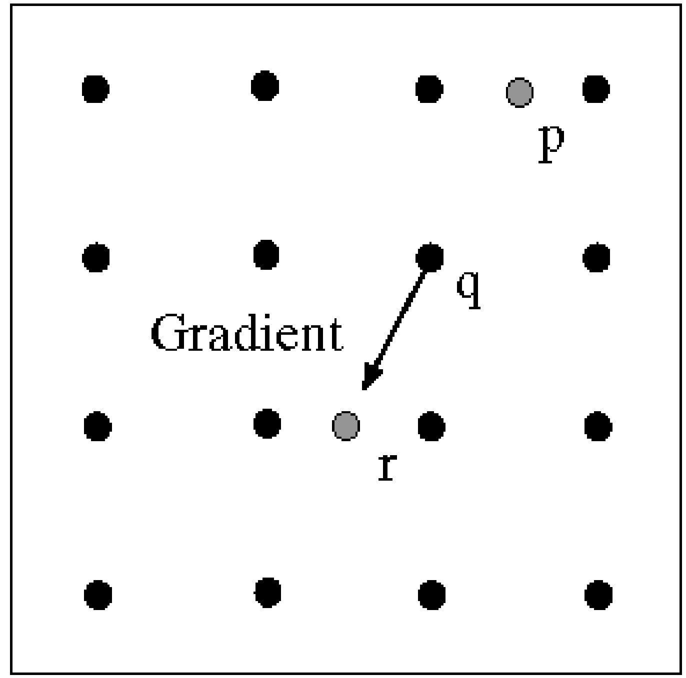
$$Cost(\mathbf{s}) = \sum_{i=1}^{h} Energy(f(s_i))$$

$$\mathbf{s}^* = \min_{\mathbf{s}} Cost(\mathbf{s})$$

Slide credit:
Kristen Grauman

Non-maximum suppression



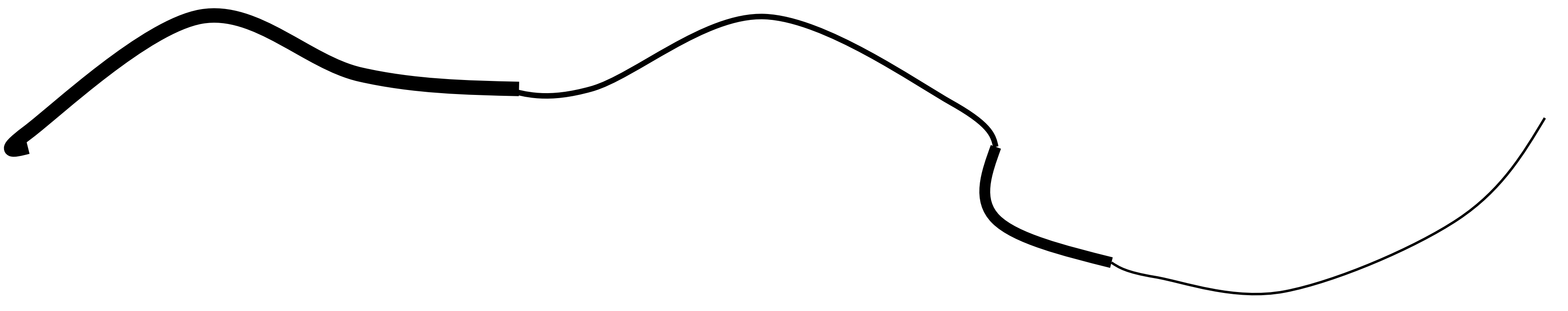


Check if pixel is local maximum along gradient direction, select single max across width of the edge

- requires checking interpolated pixels p and r

Hysteresis thresholding

- Use a high threshold to start edge curves, and a low threshold to continue them.



Source: Steve Seitz

This Lecture

- Template Matching

- Comparing contours

- Binary Image Analysis
  - Inflation
  - Erosion

Template Matching

Another Application: Template Matching

- A building block of neural networks is called a filter
  - Map raw pixels to an intermediate (feature) representation
  - Neural networks utilize filters in a hierarchical manner

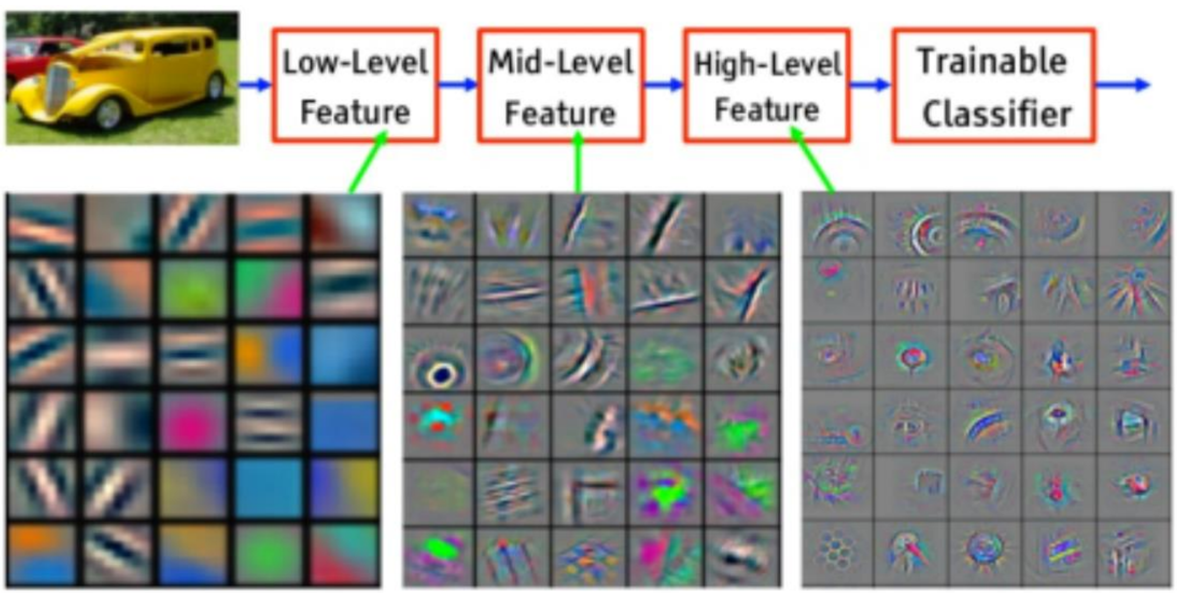
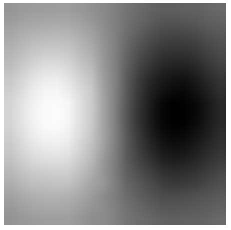
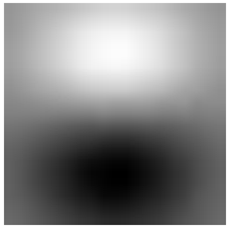


Image credit: Yann Lecun

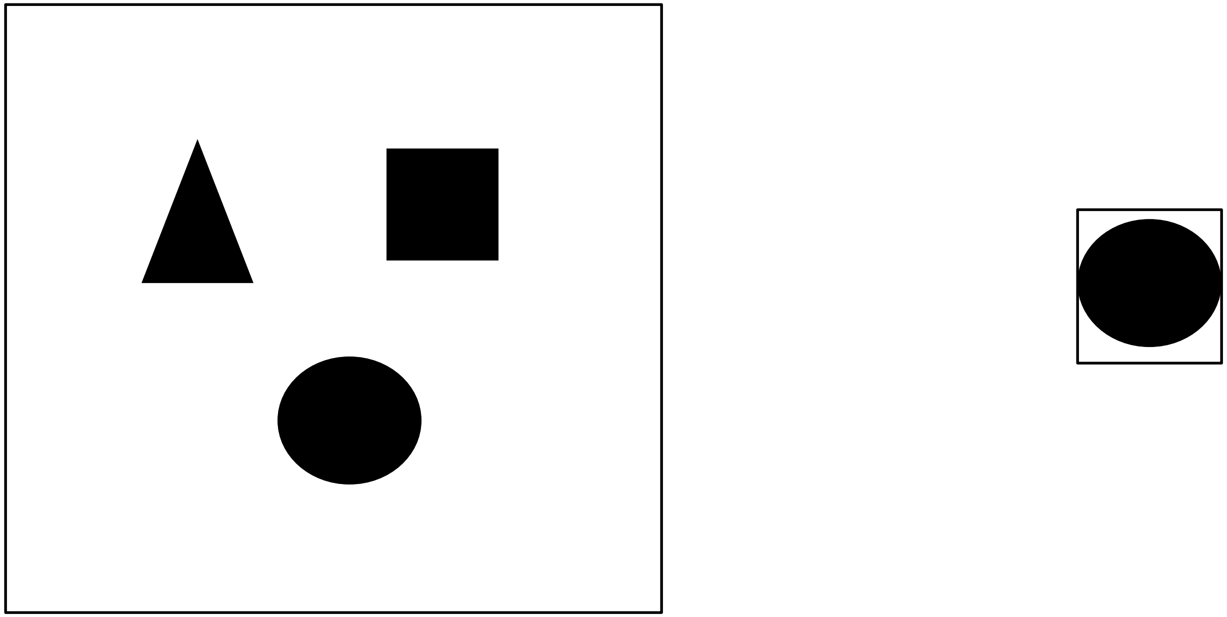
Template matching

- Filters as templates
- Note that filters look like the effects they are intended to find --- “matched filters”

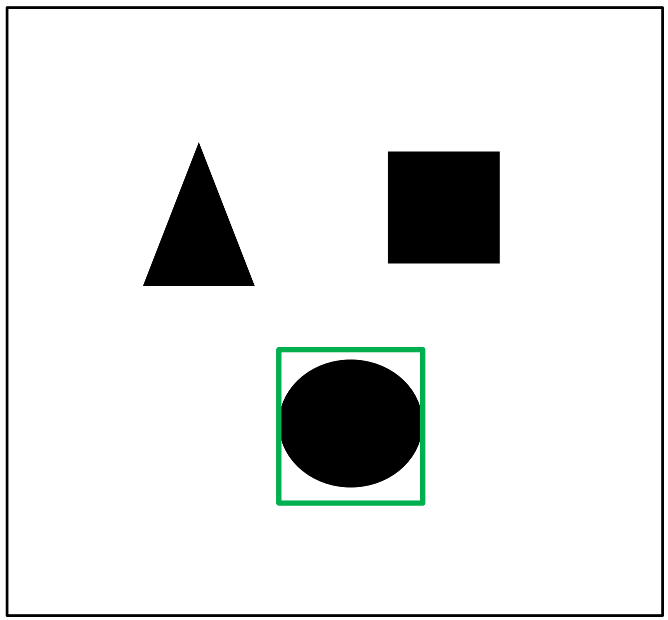
- Use normalized cross-correlation score to find a given pattern (template) in the image
- Normalization needed to control for relative brightnesses

Template matching

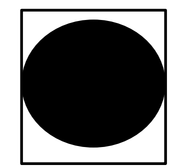


A toy example

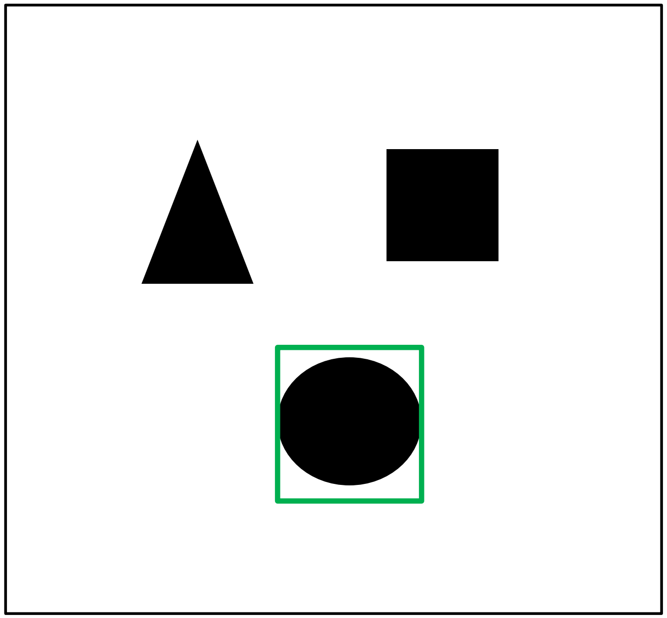
Template matching



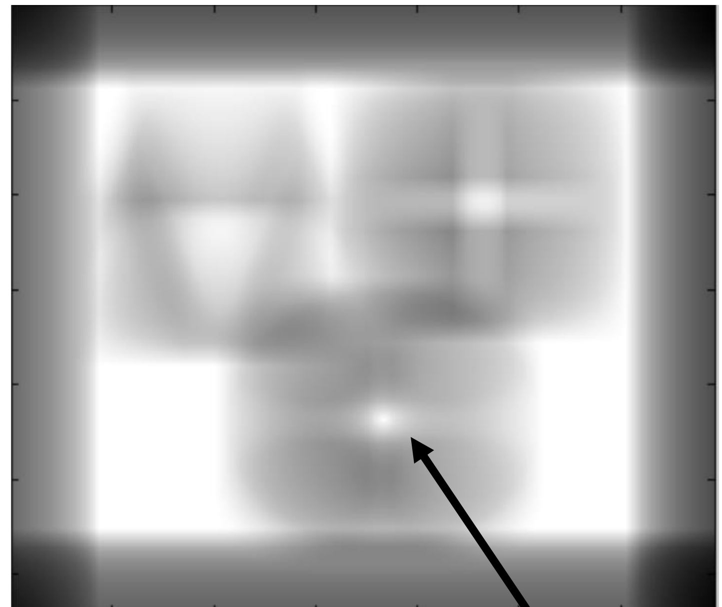
Detected template



Template matching



Detected template



Correlation map

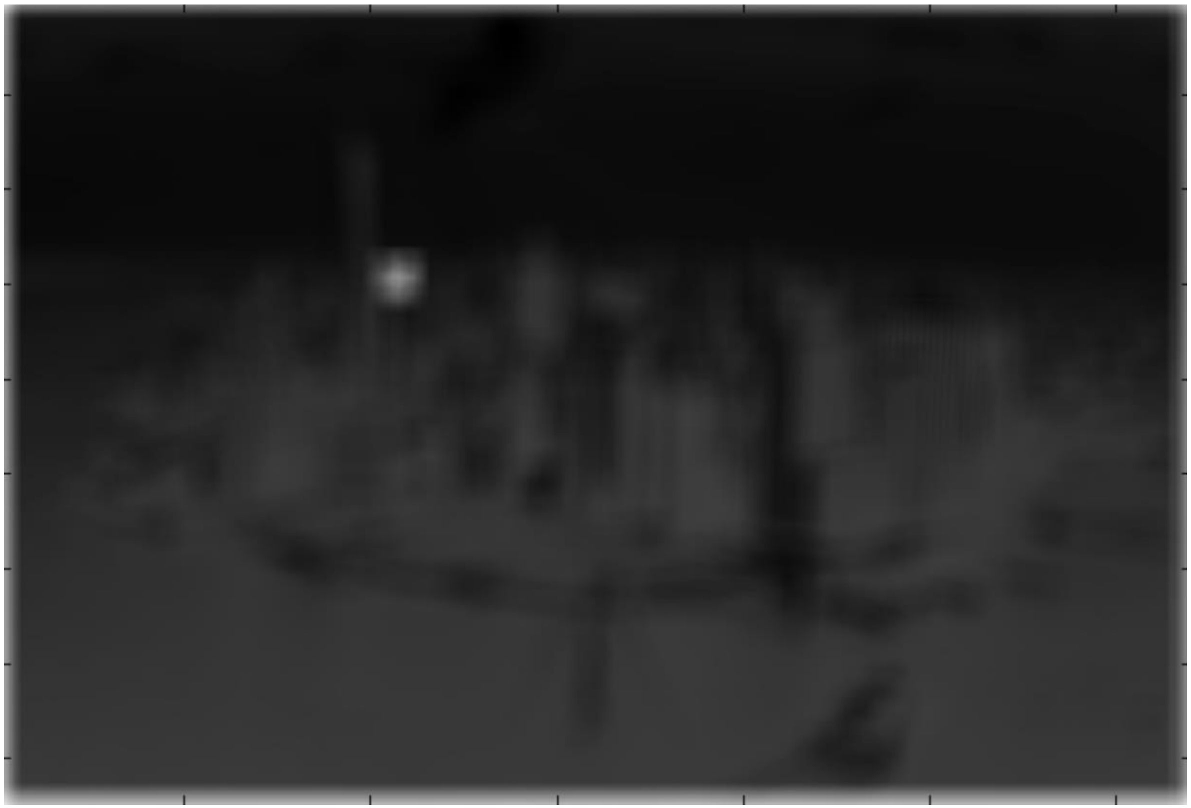
Peak

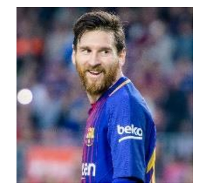
Where’s Lion Messi

Scene

Template

Where’s Lion Messi





Template

Try multiple scales

Correlation Map

Where's Lion Messi

Template

Scene

Template matching

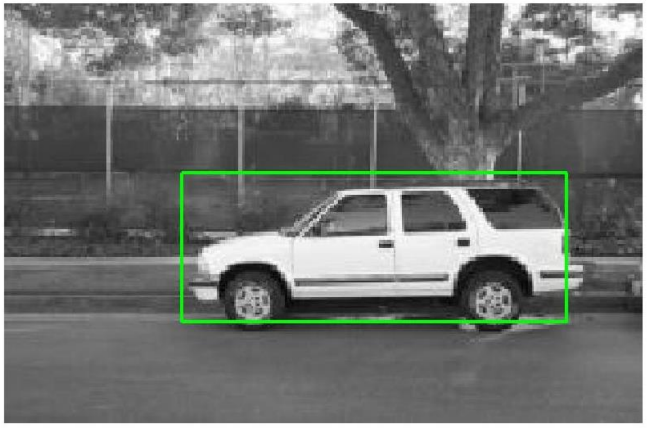
Scene

Template

What if the template is not identical to some subimage in the scene?

Slide credit:
Kristen Grauman

Template matching



Detected template

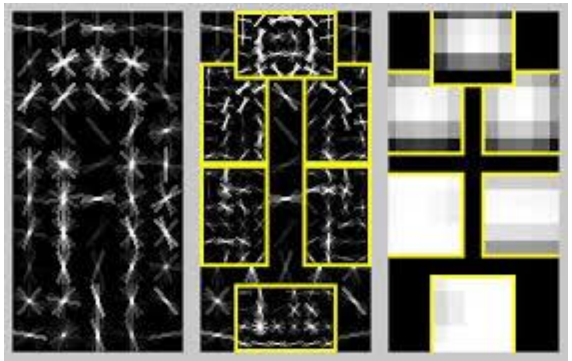
Template

Match can be meaningful, if scale, orientation, and general appearance is right.

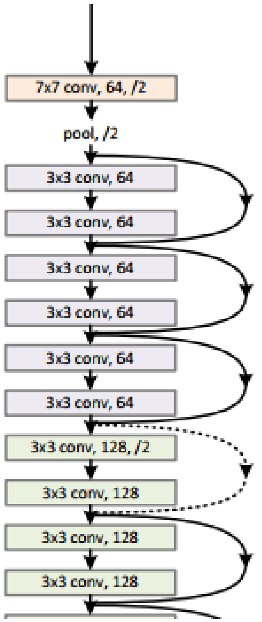
Slide credit:
Kristen Grauman

How about human?

- Deformable part model (deforming template) [Felzenszwalb et al. 10]



- Multilayer-neural network [He et al. 16]
  - The deformation in each layer is close to identity



Recap: Linear Filters

- Smoothing
  - Values positive
  - Sum to 1 → constant regions same as input
  - Amount of smoothing proportional to mask size
  - Remove “high-frequency” components; “low-pass” filter
- Derivatives
  - Opposite signs used to get high response in regions of high contrast
  - Sum to 0 → no response in constant regions
  - High absolute value at points of high contrast
- Filters act as templates
  - Highest response for regions that “look the most like the filter”
  - Dot product as correlation

Summary

- Image gradients
- Seam carving -> gradients as “energy”
- Gradients -> edges and contours
- Template matching
  - Image patch as a filter

Comparing Contours

Motivation

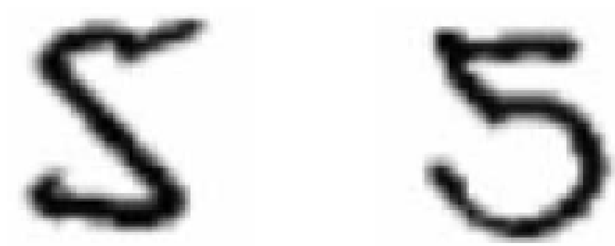


Fig. 1. Examples of two handwritten digits. In terms of pixel-to-pixel comparisons, these two images are quite different, but to the human observer, the shapes appear to be similar.

Other similar problems such as comparing shapes

[Belongie et al. 02]

Chamfer distance

- Average distance to nearest feature

$$D_{chamfer}(T,I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

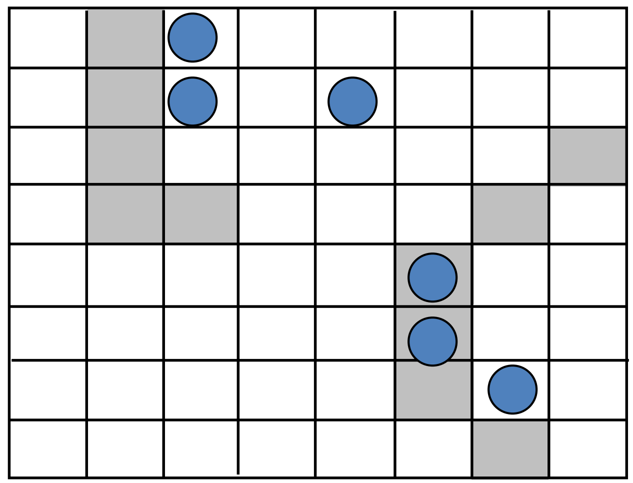
$I =$ Set of points in image

$T =$ Set of points on (shifted) template

$d_I(t) =$ Minimum distance between point t and some point in I

Slide Credit: Kristen Grauman

Chamfer distance



$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

Slide Credit: Kristen Grauman

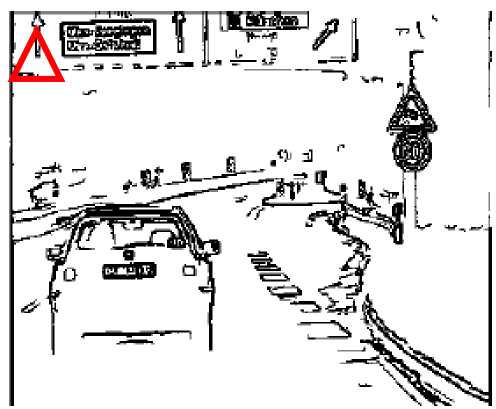
Chamfer distance

- Average distance to nearest feature

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

How is the measure different than just filtering with a mask having the shape points?

How expensive is a naïve implementation?

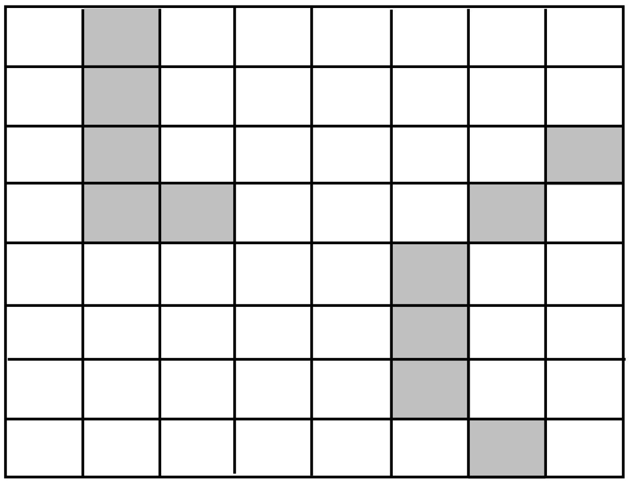


Edge image

Slide Credit: Kristen Grauman

Distance transform

Image features (2D)



Distance Transform

1	0	1	2	3	4	3	2
1	0	1	2	3	3	2	1
1	0	1	2	3	2	1	0
1	0	0	1	2	1	0	1
2	1	1	2	1	0	1	2
3	2	2	2	1	0	1	2
4	3	3	2	1	0	1	2
5	4	4	3	2	1	0	1

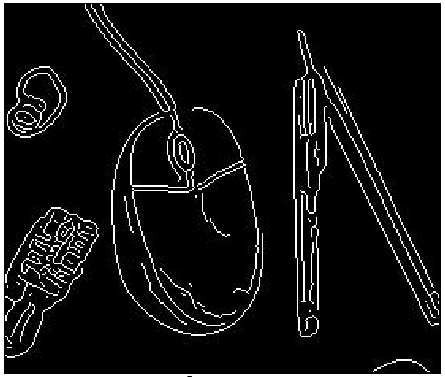
Distance Transform is a function $D(\cdot)$ that for each image pixel p assigns a non-negative number $D(p)$ corresponding to distance from p to the nearest feature in the image I

Features could be edge points, foreground points,…

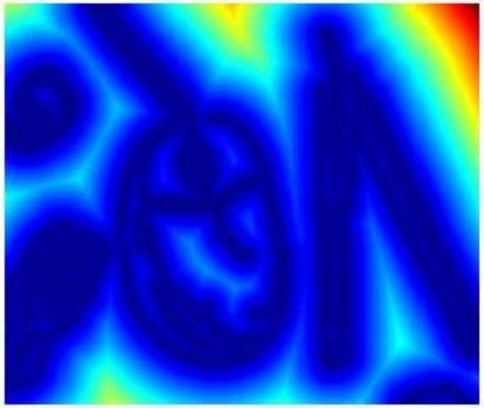
Source: Yuri Boykov

Distance transform

original



edges



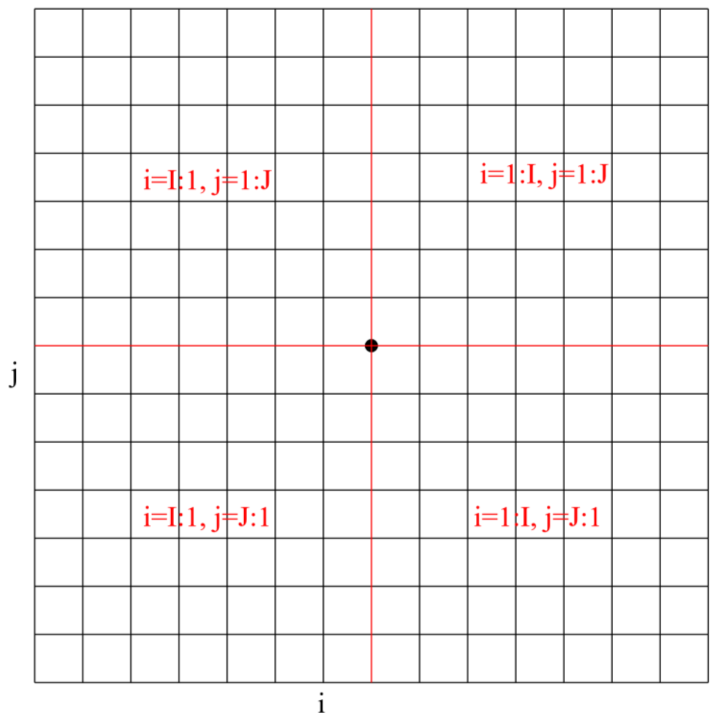
distance transform

Value at (x,y) tells how far that position is from the nearest edge point (or other binary mage structure)

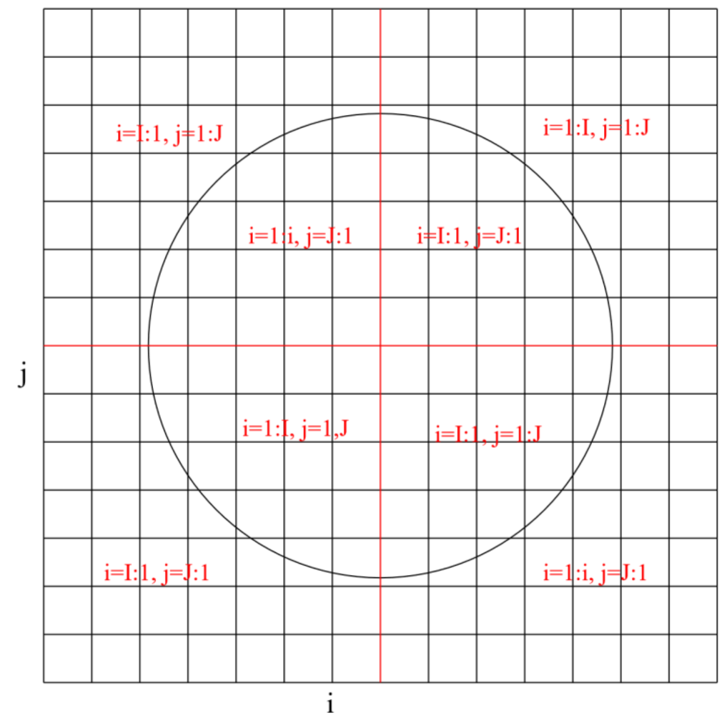
```
>> help bwdist
```

Slide credit: Kristen Grauman

Fast Sweeping for Distance Computation



a) the fast sweeping algorithm for a single data point



(b) the fast sweeping algorithm for a circle

A fast sweeping method for eikonal equations. Hongkai Zhao. 2001

Fast Sweeping

$$\begin{array}{ll} |\nabla u(\boldsymbol{x})| = f(\boldsymbol{x}) & \boldsymbol{x} \in R^n \\ u(\boldsymbol{x}) = 0 & \boldsymbol{x} \in \Gamma \subset R^n, \end{array}$$

Initialization: To enforce the boundary condition, $u(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \Gamma \subset R^n$, assign exact values or interpolated values at grid points in or near Γ. These values are fixed in later calculations. Assign large positive values at all other grid points. These values will be updated later.

Gauss-Seidel iterations with alternating sweeping orderings: At each grid $\boldsymbol{x}_{i,j}$ whose value is not fixed during the initialization, compute the solution, denoted by $\overline{u}$, of (2.2) from the current values of its neighbors $u^h_{i\pm1,j}, u^h_{i,j\pm1}$ and then update $u^h_{i,j}$ to be the smaller one between $\overline{u}$ and its current value, i.e., $u^{new}_{i,j} = \min(u^{old}_{i,j}, \overline{u})$. We sweep the whole domain with four alternating orderings repeatedly,

$$\begin{array}{ll} (1)\ i = 1:I, j = 1:J & (2)\ i = I:1, j = 1:J \\ (3)\ i = I:1, j = J:1 & (4)\ i = 1:I, j = J:1 \end{array}$$

The unique solution to the equation

$$[(x-a)^+]^2 + [(x-b)^+]^2 = f^2_{i,j}h^2, \tag{2.3}$$

where $a = u^h_{xmin}, b = u^h_{ymin}$, is

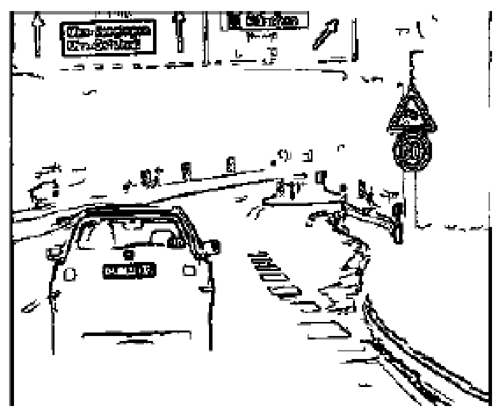
$$\overline{x} = \begin{cases} \min(a,b) + f_{i,j}h & |a-b| \ge f_{i,j}h \\ \frac{a+b+\sqrt{2f^2_{i,j}h^2-(a-b)^2}}{2} & |a-b| < f_{i,j}h \end{cases} \tag{2.4}$$

A fast sweeping method for eikonal equations. Hongkai Zhao. 2001

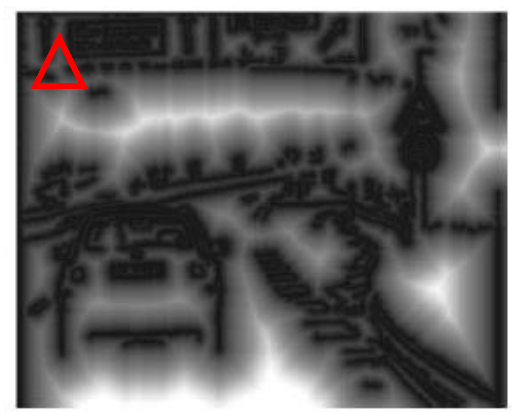
Chamfer distance

- Average distance to nearest feature

$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$



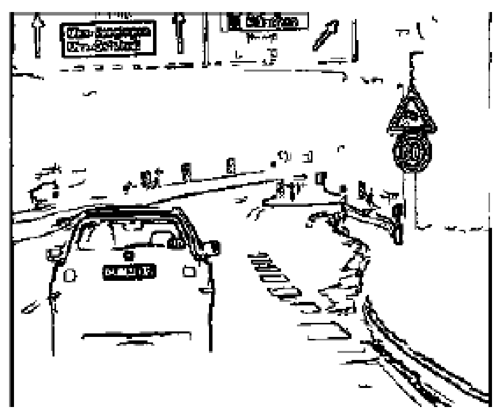
Edge image



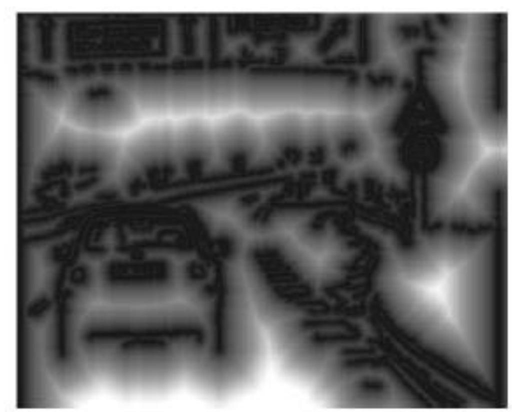
Distance transform image

Slide credit: Kristen Grauman

Chamfer distance



Edge image



Distance transform image

Slide credit: Kristen Grauman

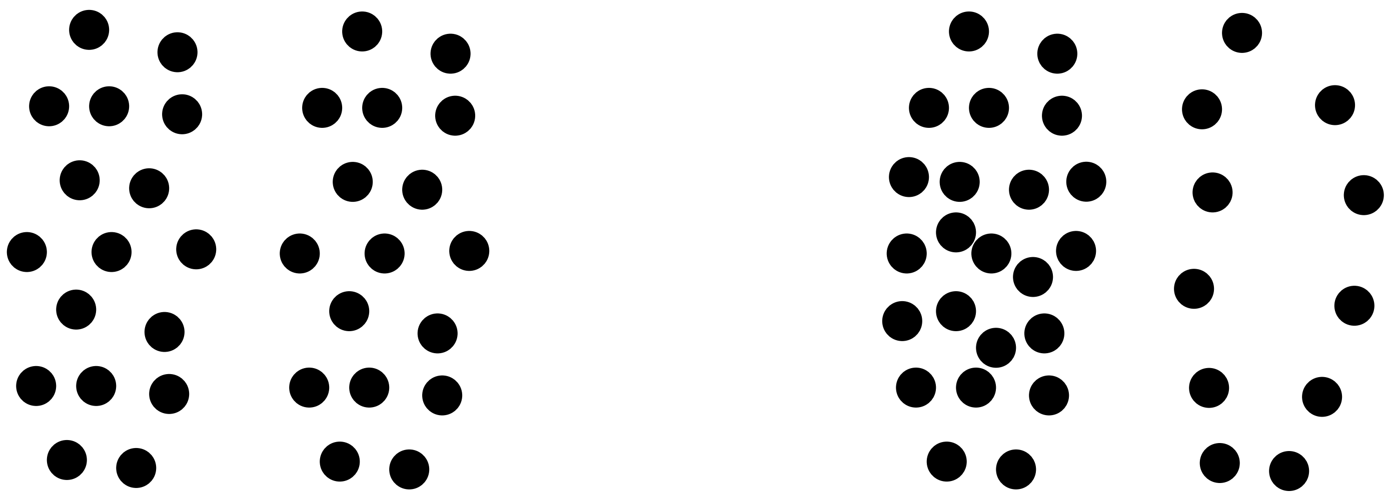
Fig from D. Gavrila, DAGM 1999

Chamfer distance: properties

- Sensitive to scale and rotation
- Tolerant of small shape changes, clutter
- Need large number of template shapes
- Inexpensive way to match shapes

Slide credit: Kristen Grauman

Earth-mover distance (or EMD)

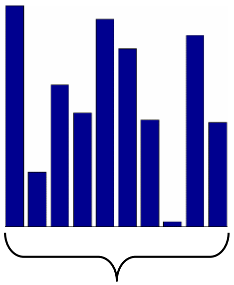


Chamber distance is small

EMD distance is large

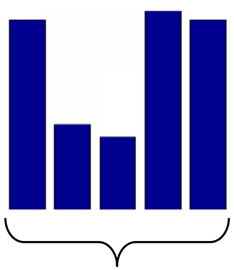
Earth-mover distance

P



m clusters

Q



n clusters

$$\sum_{\text{All movements}} \text{(distance moved) * (amount moved)}$$

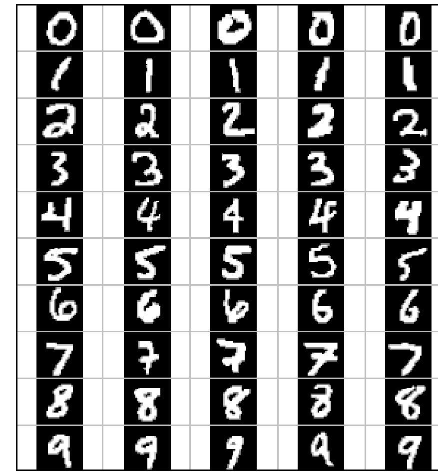
$$\sum_{i=1}^{m} \sum_{j=1}^{n} \text{(distance moved) * (amount moved)}$$

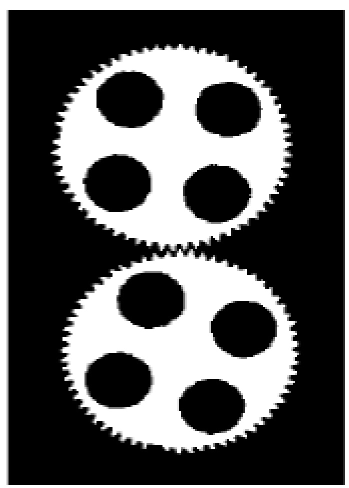
$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} \text{ * (amount moved)}$$

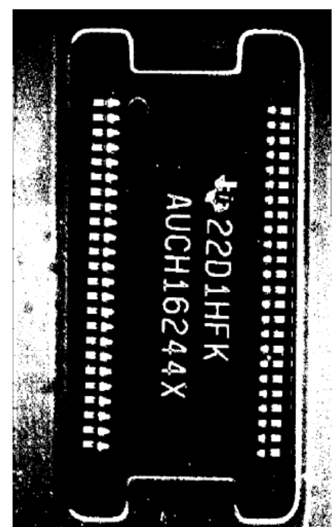
$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} \, f_{ij} = \text{WORK}$$

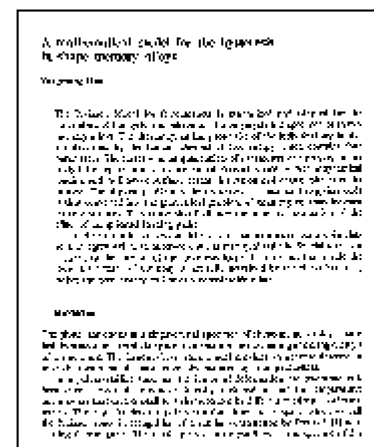
Binary Image Analysis

Binary images









Slide credit: Kristen Grauman

Binary image analysis: basic steps

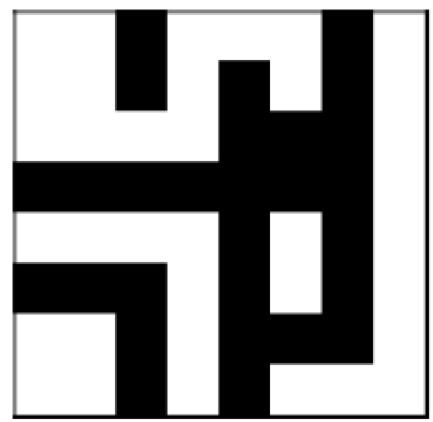
- Convert the image into binary form
  - Thresholding
- Clean up the thresholded image
  - Morphological operators
- Extract separate blobs
  - Connected components
- Describe the blobs with region properties

Slide credit: Kristen Grauman

Binary images

- Two pixel values
  - Foreground and background
  - Mark region(s) of interest

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1



Slide credit: Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: edge detection

Gradient magnitude

`fg_pix = find(gradient_mag > t);`

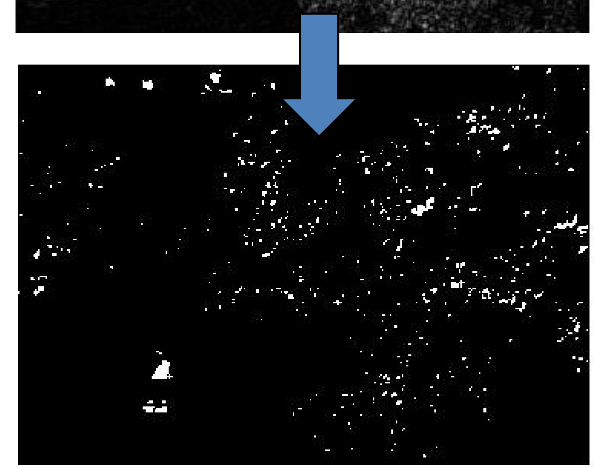
Looking for pixels where gradient is strong.

Slide credit: Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: background subtraction



Looking for pixels that differ significantly from the "empty" background.

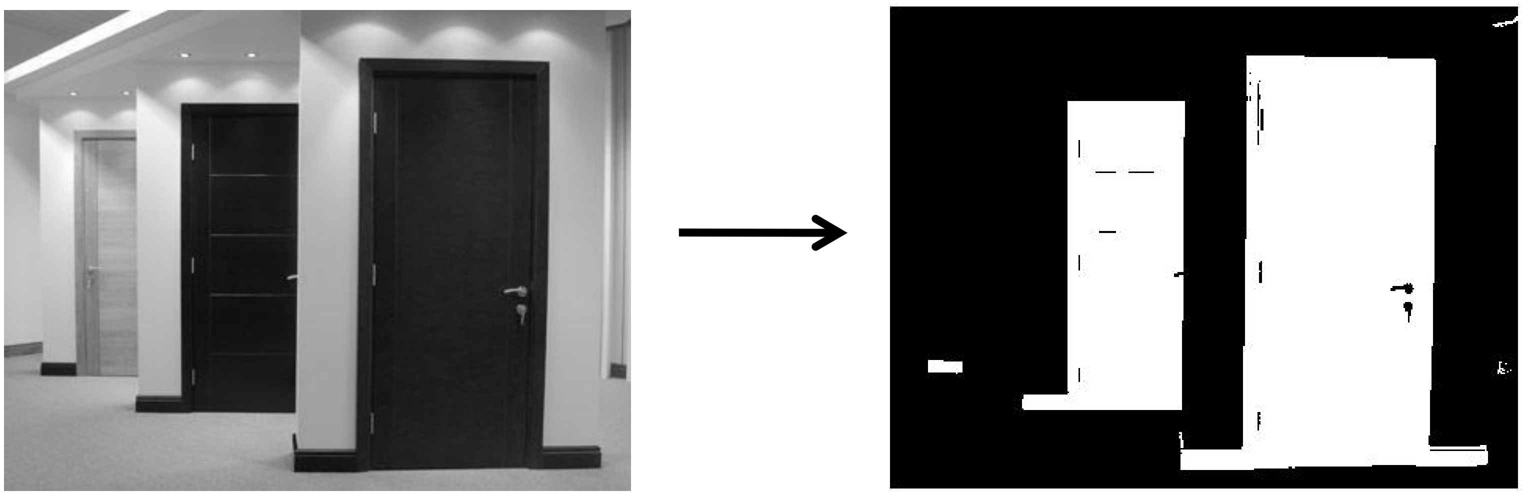
Slide credit: Kristen Grauman

```
fg_pix = find(diff > t);
```

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: intensity-based detection



```
fg_pix = find(im < 65);
```

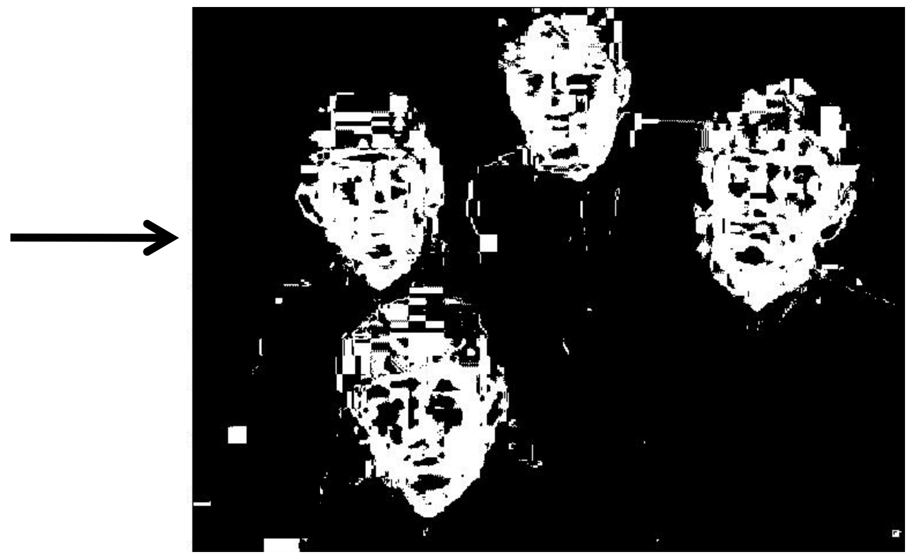
Looking for dark pixels

Slide credit: Kristen Grauman

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: color-based detection



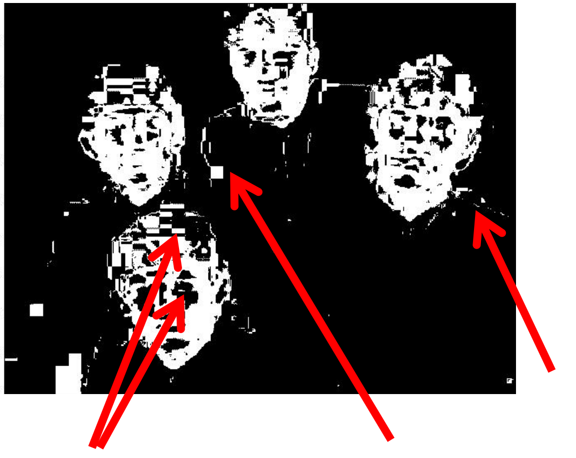
```
fg_pix = find(hue > t1 & hue < t2);
```

Looking for pixels within a certain hue range.

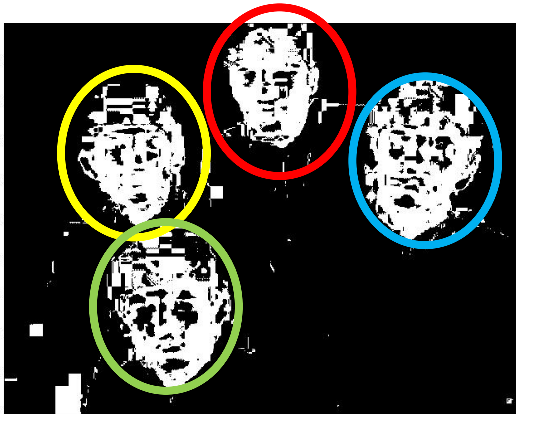
Slide credit: Kristen Grauman

Issues

- What to do with “noisy” binary outputs?
  - Holes
  - Extra small fragments



- How to demarcate multiple regions of interest?
  - Count objects
  - Compute further features per object



Slide credit: Kristen Grauman

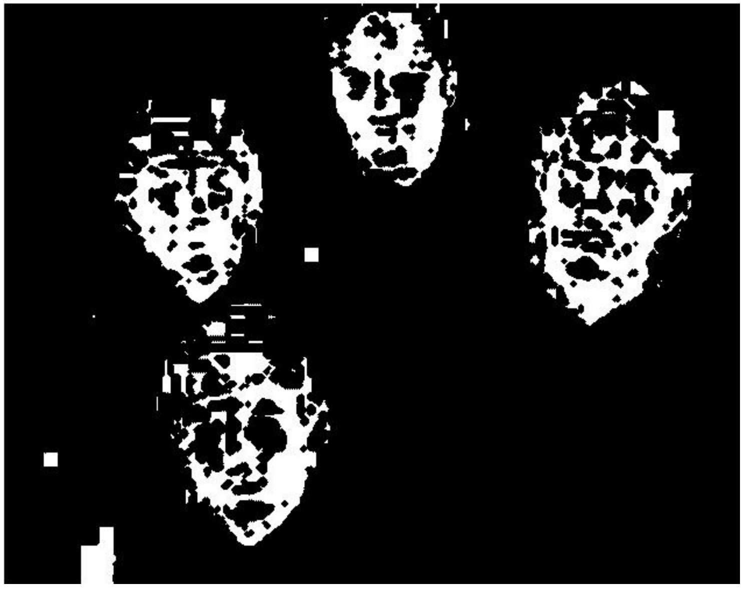
Morphological operators

- Change the shape of the foreground regions via intersection/union operations between a scanning structuring element and binary image.
- Useful to clean up result from thresholding
- Basic operators are:
  - Dilation
  - Erosion

Slide credit: Kristen Grauman

Dilation

- Expands connected components
- Grow features
- Fill holes



Before dilation

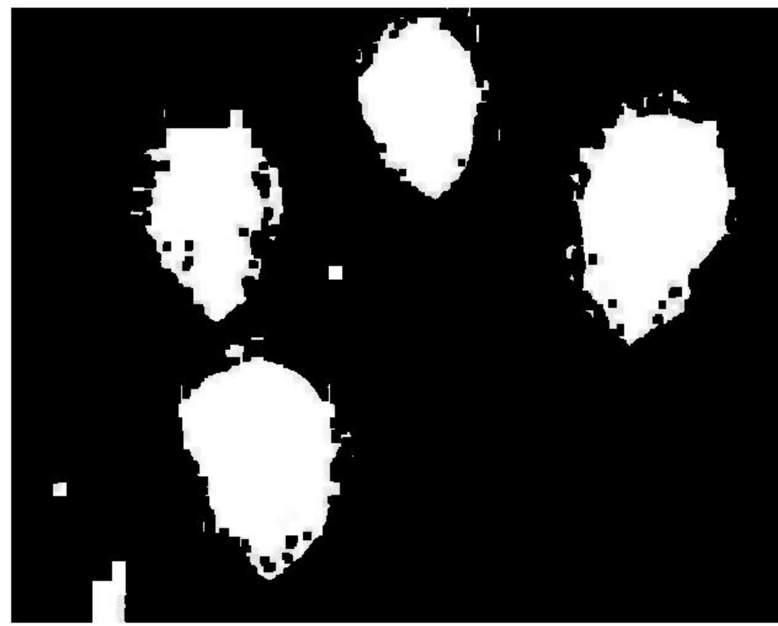
After dilation

Slide credit: Kristen Grauman

Erosion

- Erode connected components
- Shrink features
- Remove bridges, branches, noise

Before erosion

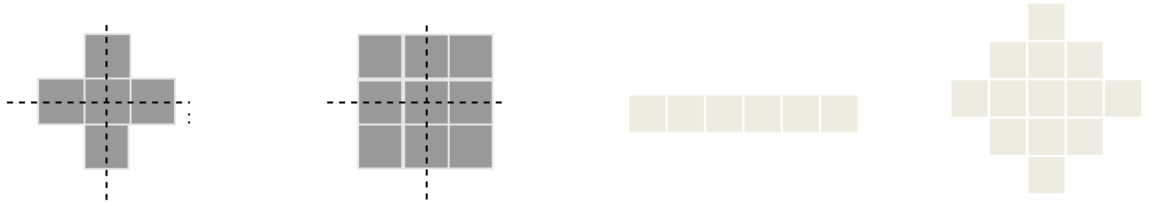


After erosion

Slide credit: Kristen Grauman

Structuring elements

- **Masks** of varying shapes and sizes used to perform morphology, for example:



- Scan mask across foreground pixels to transform the binary image

```
>> help strel
```

Slide credit: Kristen Grauman

Dilation vs. Erosion

At each position:

- **Dilation**: if current pixel is foreground, OR the structuring element with the input image.

Slide credit: Kristen Grauman

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

1	1	0	1	1	1	1	1	1	1

Note that the object gets bigger and holes are filled.

```
>> help imdilate
```

Slide credit: Kristen Grauman

Dilation vs. Erosion

At each position:

- **Dilation**: if **current pixel** is foreground, OR the structuring element with the input image.
- **Erosion**: if **every pixel** under the structuring element's nonzero entries is foreground, OR the current pixel with S.

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	**1**	**1**

Output Image

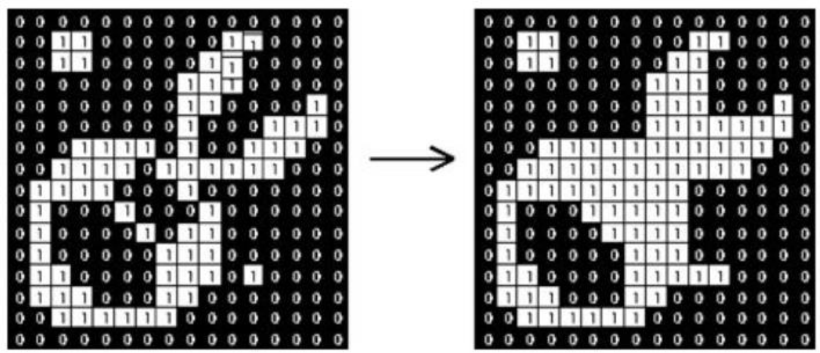
0	0	0	0	0	1	0	0	0	1

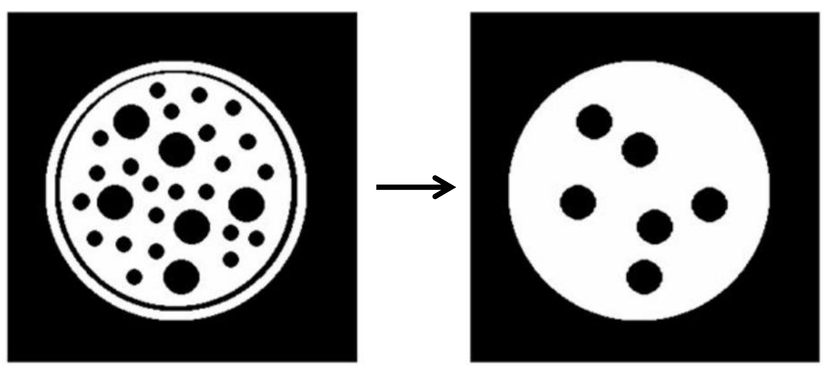
Note that the object gets smaller

```
>> help imerode
```

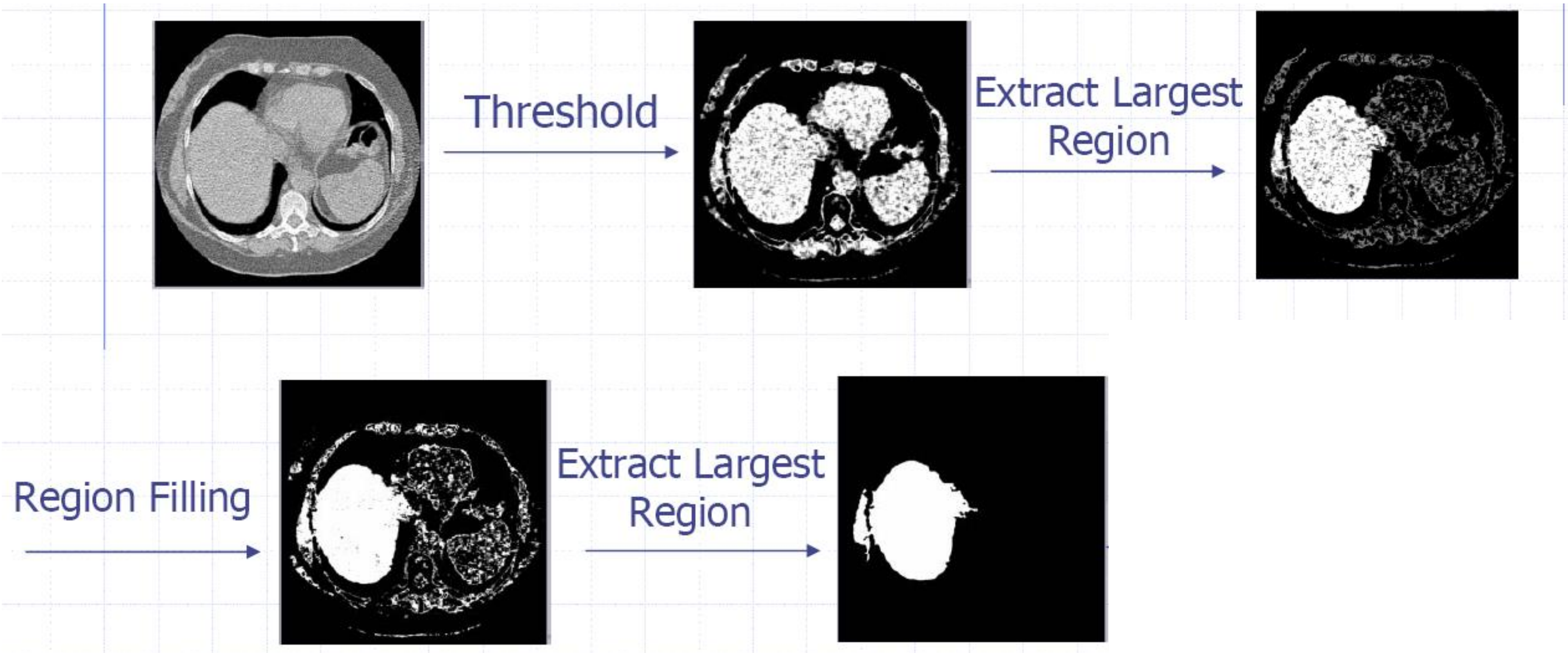
Typical operation

- Erode, then dilate
- Remove small objects, keep original shape





Example using binary image analysis: segmentation of a liver

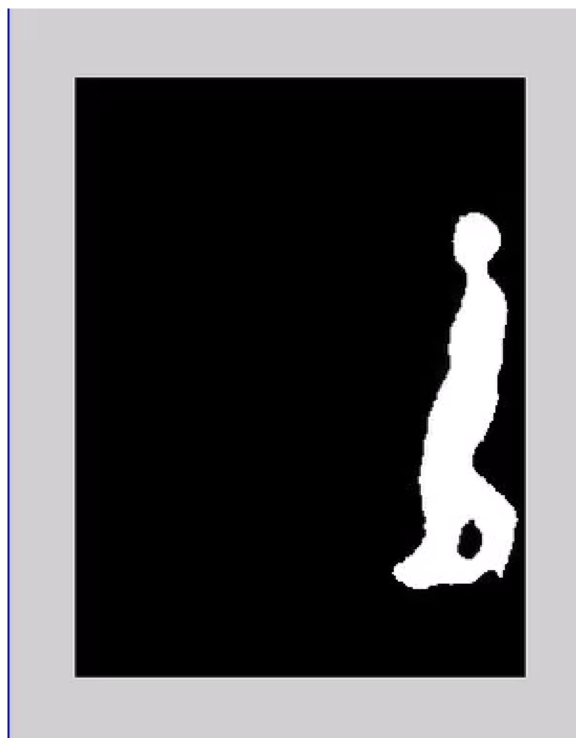


Slide credit: Li Shen

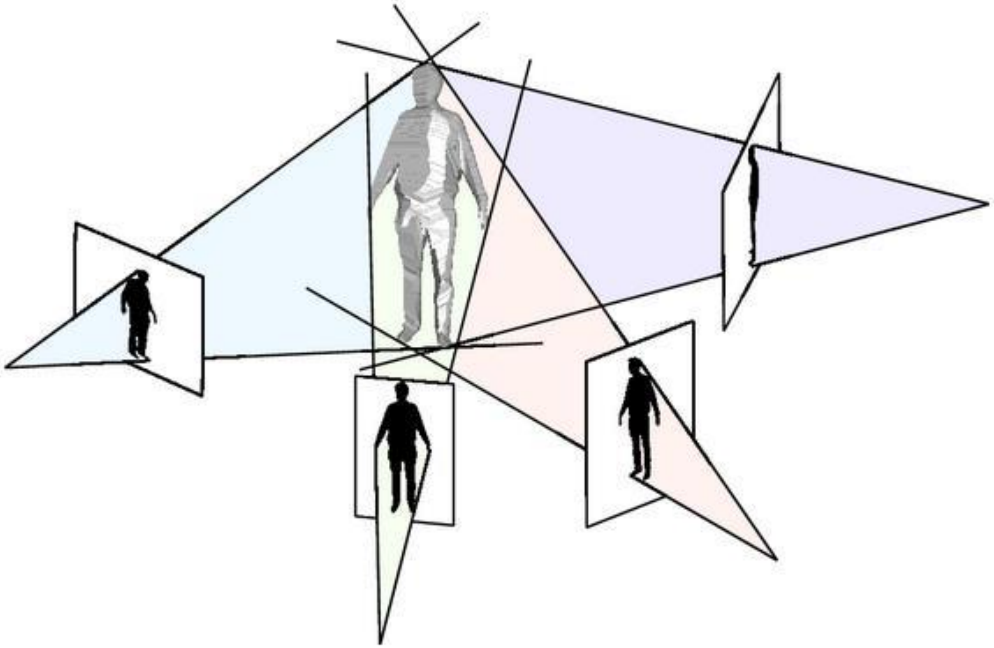
Application by Jie Zhu, Cornell University

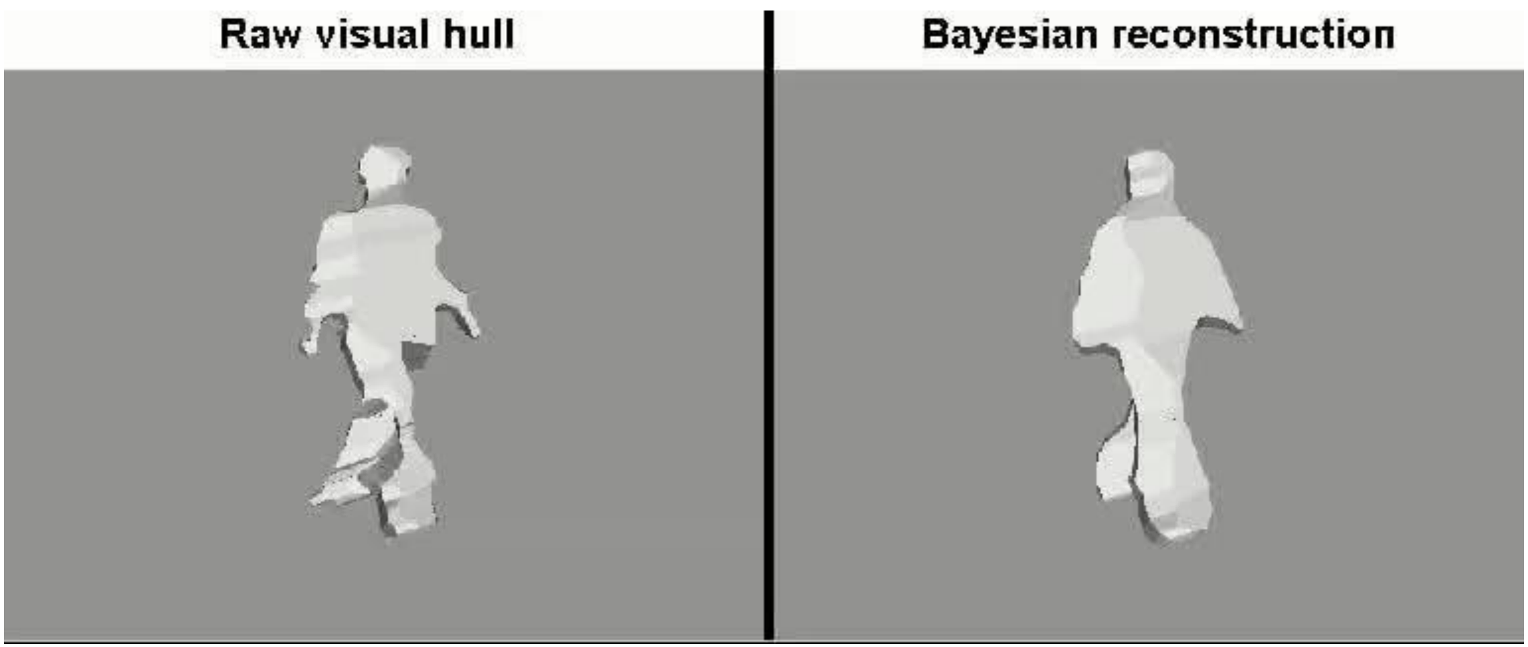
Example using binary image analysis: Bg subtraction + blob detection

…



Visual hulls





Slide credit: Kristen Grauman

Next lecture

- Texture analysis
- Texture synthesis