MACHINE VISION

- A fancy term for interpreting pictures by a computer.
- **Image Analysis** is a more modest-sounding synonym.
- The difficulty of the problem has been grossly underestimated.

To interpret a picture we need to find what is in there.

- **HOW DO WE FIND OBJECTS IN A PICTURE?**
  - By looking for their outlines (BOTTOM UP)
  - By matching characteristics of a known object to parts of a picture (TOP DOWN)

Edge Detection

- We assume that the outline of an object will consist of pixels where the luminance $I(x,y)$ varies a lot, in other words the size of the **gradient** of the luminance is large.
- This process is called **edge detection**.

Edge Detection

- Edge detection relies on calculating the luminance gradient.
- At pixel location (x,y) the mathematical expression for the size of the gradient is

$$\|g\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

Sobel Edge Detector

- Derivatives amplify noise so we like to do some local smoothing before taking the gradient. We compute discrete approximations to the gradient as
- $HG(x, y) = I(x+1,y-1) - I(x-1, y-1) + 2*(I(x+1, y) - I(x-1, y)) + I(x+1, y+1) - I(x-1,y+1)$
- $VG(x, y) = I(x-1,y+1) - I(x-1, y-1) + 2*(I(x, y+1) - I(x, y-1)) + I(x+1, y+1) - I(x+1,y-1)$
- Define as **edgels** all pixels where the sum of the squares of HG and VG exceeds some given threshold.

The Sobel coefficients in matrix form

-1	0	1
-2	**0**	2
-1	0	1

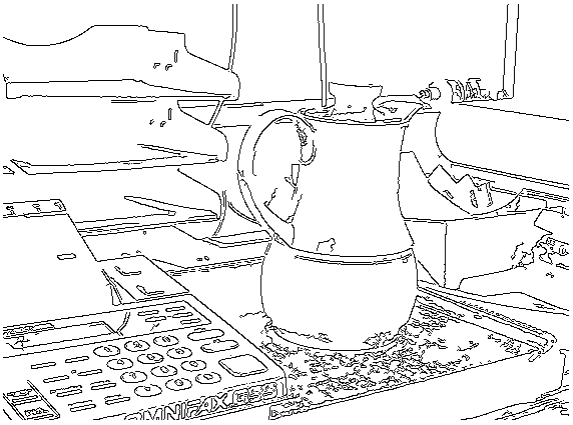
For HG

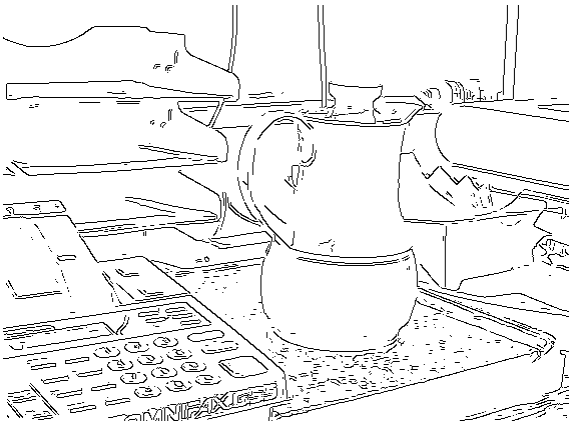
1	2	1
0	**0**	0
-1	-2	-1

For VG

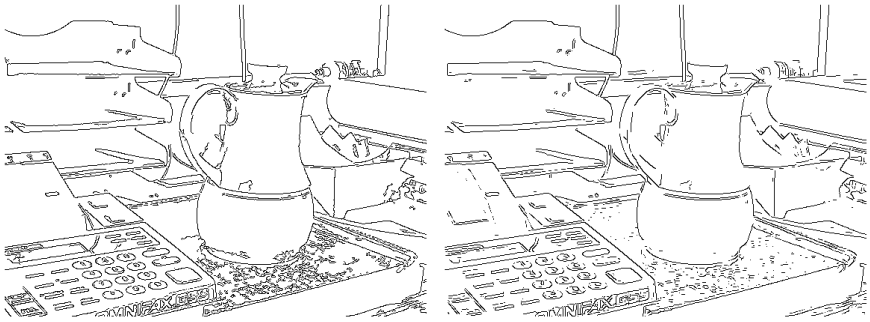
More Edge Detectors

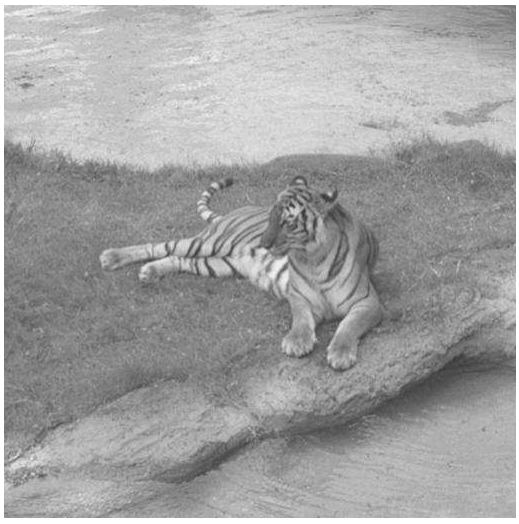
- We can use different kinds of smoothing and pay more attention to the direction of the gradient (rather than just add up the squares of the two differences).
- Such edge detectors do a better job than Sobel's edge detector but not by much!

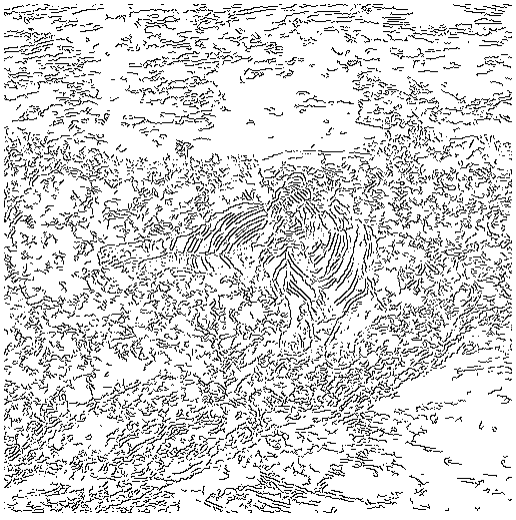


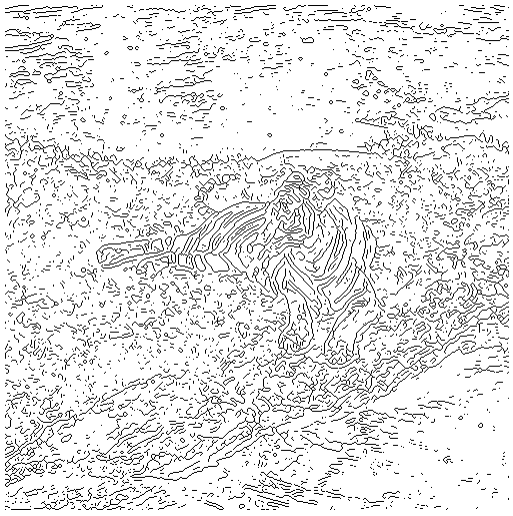


Results of two edge detectors

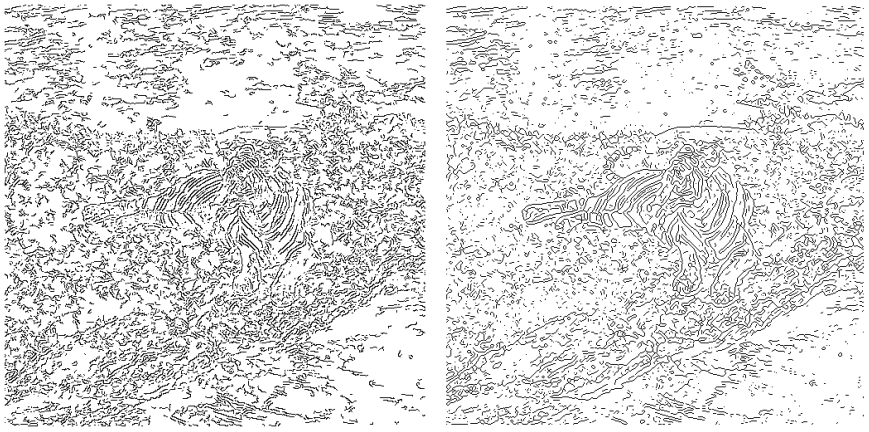








Results of two edge detectors



Why is so easy for people to see the tiger while the computer has a tough time?

We expect that dealing with images to be much harder than dealing with text.

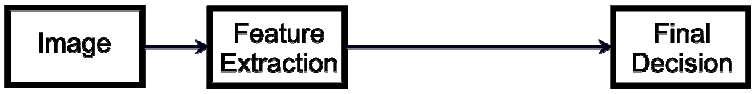
- The human visual system has evolved from animal visual systems over a period of more than **200 million** years.

- Speech is barely over **100 thousand** years old.
- Written text is about **5 thousand** years old.

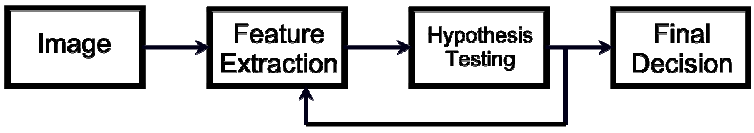
In humans the visual system occupies 1/3 of the brain, a much larger portion than the auditory system. 85% of human sensorial information is the result of visual inputs.

A Big Difference

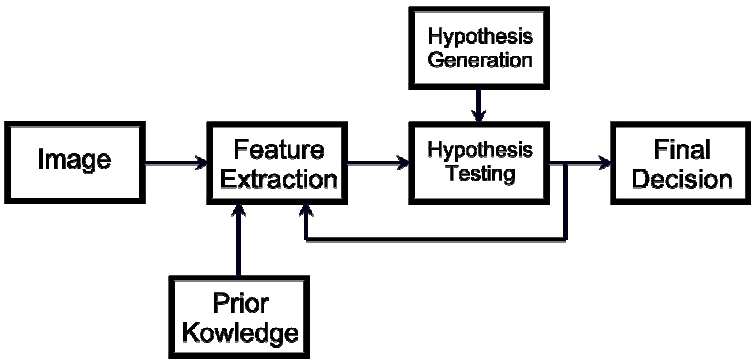
- People do not look at the pixels "from scratch" but they have an anticipation of what is there to see.
- Inferring from pixels what is in a picture it is called a "Bottom Up" process.
- Inferring from expectations what is in a picture it is called a "Top Down" process.

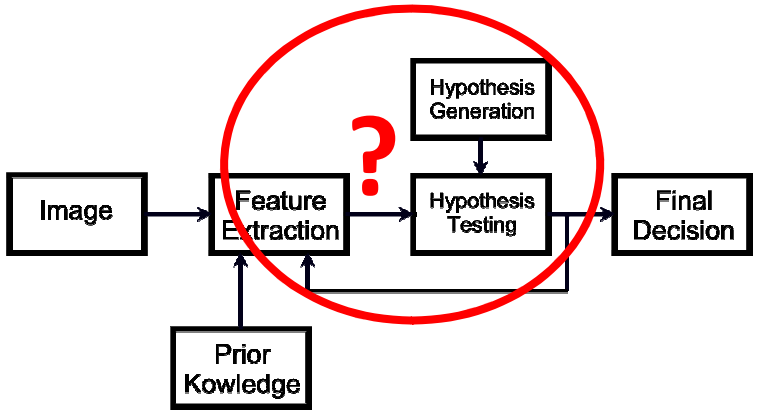


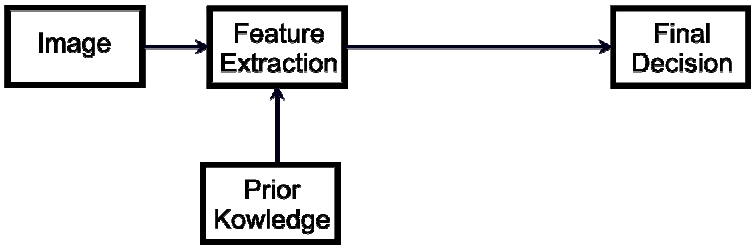
Typical computer vision process



Human (and animal) vision tries to guess what is seen.


Hypothesis Generation
Image
Feature Extraction
Hypothesis Testing
Final Decision
Prior Kowledge

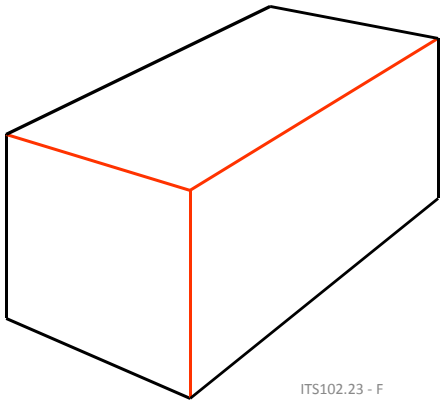

Hypothesis Generation
?
Image
Feature Extraction
Hypothesis Testing
Final Decision
Prior Kowledge



An Example

Find the outline of the box in the image below (ultimate goal is to find the dimensions of the box):

Basic Idea: Because the three edges meeting at a vertex are mutually perpendicular we can compute their relative size from one view.



A paradox

- Human viewers have no trouble identifying the box and its outline.
- Application of Edge Detection or Segmentation produces a "mess:"
  - Contrast inside the box may be higher than contrast between the box and the background.

An Inspiration from Nature

- In a classical paper J. Letvin *et al* showed that the frog's visual system responds to only two kinds of stimuli:
  - fast moving, high contrast small shapes (food) or
  - decrease in the ambient illumination (danger).

  [*Proceedings of IRE*, 1959]

Implications for the box dimension problem

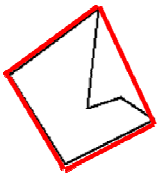
- The system should look only for hexagonal shapes occupying most of the image.
- This means that the only edges of interest should be lines of length comparable to the dimensions of the field of view.
- Such lines should form a convex set.
- The convex set should be a hexagon.

Mathematical Definitions

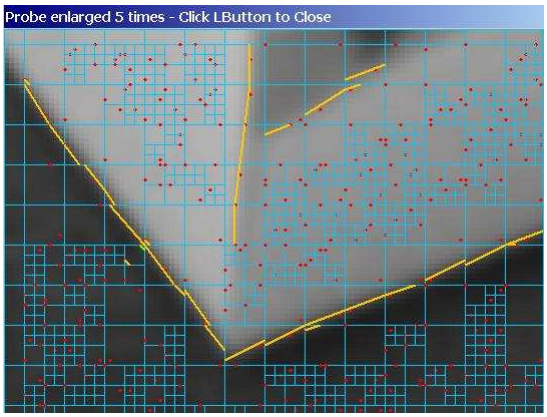
- An object is **convex** if for every pair of points within the object, every point on the straight line segment that joins them is also within the object.
- For example, a solid cube is convex, but anything that is hollow or has a dent in it, for example, a crescent shape, is not convex.

Mathematical Definitions (cont)

- The convex hull of a set of points is the smallest convex set containing all the points.



Use **(Long) Line Detection** as the first step (rather than segmentation or edge detection)



Summary of the Box Outlining Method

- After the long lines have been found we look for "proximity clusters."
- Then the convex hull of the lines is determined.
- If the convex hull is a hexagon (outline of a rectangular object) occupying most the image, we are done.

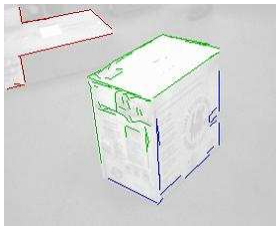
Summary of the Box Outlining Method (cont)

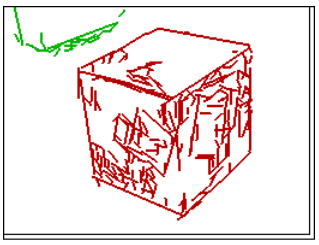
- If we cannot find a hexagon, certain lines, marked as “noise” are removed and we try again until we succeed in finding such a hexagon or determine that such a shape does not exist.

Proximity Clusters

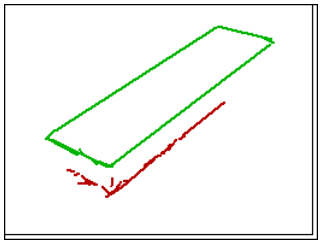
- The line segments found are merged to find long lines (we look at co-linearity for that).
- The lines found are then clustered into proximity clusters.
- A **proximity cluster** is defined as a set of line segments $\mathcal{L}$ with the property that for each s in $\mathcal{L}$, there is a t in $\mathcal{L}$, such that t and s have at least a pair of endpoints near each other.

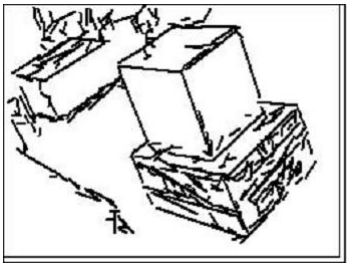
Examples of Proximity Clusters





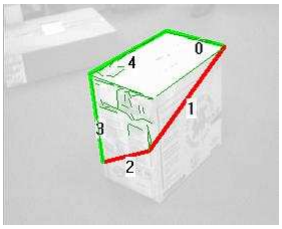
!!!

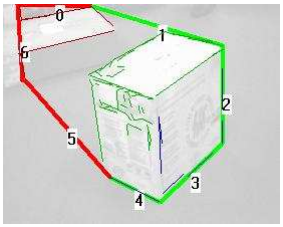




Convex Hull

- Next we find the convex hull of each cluster as well as that of groups of clusters. (We use a standard algorithm for the process.)

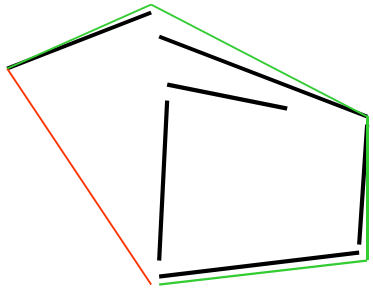




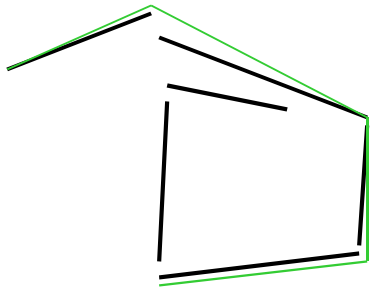
Editing the Convex Hull (Main Heuristic)

- Line segments of the convex hull are assigned a confidence level that is high if they are nearly collinear to a line segment of the cluster.
- Line segments with low confidence (red in figures) are removed together with all line segments that contributed to them.

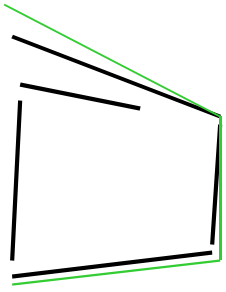
Editing Example



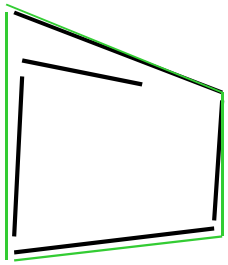
Editing Example



Editing Example



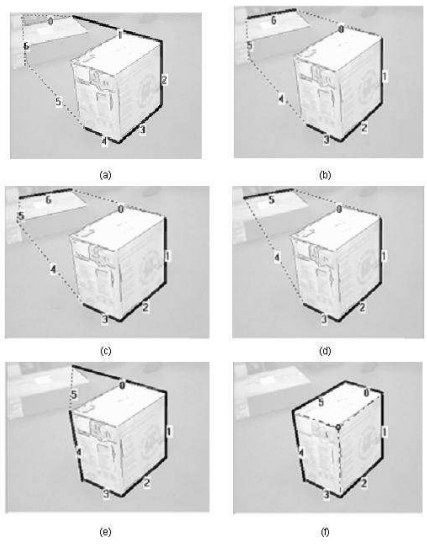
Editing Example



Editing Continued

- We also check how closely the convex hull resembles a hexagon (the projection of a rectangular object) and remove edges that reduce the quality.

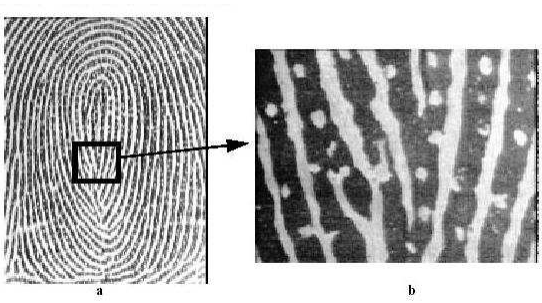
Sequence of Editing Operations



The Key to Success

- If you know what you are looking for, then you can find it! Areas where this is true:
  - Industrial Inspection: Check if a chip has the right wiring.
  - Co-operative face recognition: Compare a face shot to a data base and confirm that some is who says he is.

Fingerprint Identification



Detect lines and look for certain shape configurations.
We are interested in matching only identical fingerprints!

Looking for Flesh

- Here is a code fragment that tells us whether a pixel has "flesh" color (we assume flesh of a person of European origin).
- BOOL Flesh = FALSE;
  if((bchr+rchr > 250) &&
  (bchr > 60 && bchr < 130) &&
  (rchr > 138 && rchr < 172)) Flesh = TRUE;
  if(lum < 90 || lum > 200) Flesh = FALSE;

Blocking “Porn”

- If an image contains a lot of pixels with flesh color then it is blocked. This also blocks a picture of a light yellow-brown cat and lets through many objectionable images.
- While it is relatively easy to block objectionable text based on the frequency of the occurrence of certain words, the situation is much harder for images. In addition to what is objectionable in an image is very much in the mind of the viewer.

Finding Faces - 1

- Find all pixels with flesh color.
- For each group of adjacent pixels of that color compute the bounding rectangle **rw** by **rh**. Let **A** stand for the number of flesh color pixels inside the rectangle.
- areaRatio = (rh*rw)/A;
  aspectRatio = rh/rw;
  if(aspectRatio >= 0.6 && aspectRatio <= 2.1 && areaRatio<=3) { // accept as face }

Finding Faces - 2

- You can add code to look for “eyes” (two dark round spots), “eyebrows”, and “mouth”.
- Many modern cameras use such face detection code for autofocus.
- But the method is far from foolproof!

Your Tax Dollars at Work

- In the aftermath of 9/11, the U. S. government spent millions of dollars for face detection software to find terrorists in a crowd.

More Politics - 1

- An ACLU press release (2002) stated that "interim results of a test of face- recognition surveillance technology … from Palm Beach International Airport confirm previous results showing that the technology is ineffective.“

More Politics - 2

- "Even with recent, high quality photographs and subjects who were not trying to fool the system, the face-recognition technology was less accurate than a coin toss. Under real world conditions, Osama Bin Laden himself could easily evade a face recognition system. … It hardly takes a genius of disguise to trick this system. All a terrorist would have to do, it seems, is put on eyeglasses or turn his head a little to the side.”

More Politics - 3

- Boston Globe (2002) quoted the director of security consulting firm saying that the "technology was not ready for prime time yet.'' He added that the " systems produced a high level of false positives, requiring an airport worker to visually examine each passenger and clear him for boarding.“

More Politics - 4

- The article goes on to say: "One of the biggest deployments of the technology has occurred in England, in the London borough of Newham. Officials there claim that the installation of 300 facial-recognition cameras in public areas has led to a reduction in crime. However, they admit that the system has yet to result in a single arrest.”

More Politics - 5

- The suppliers of the face recognition systems insist that the testers need to prove beyond reasonable doubt that their systems are faulty, instead of themselves having to prove that they are selling a valid product.

Find the Terrorist!

Results of the Robotics Institute, CMU program. A green rectangle is overlaid on any face detected. A major miss is evident.

Do you need more to be convinced?

Can you tell how these two pictures differ?

How about these two?