Computational Foundations of Cognitive Science

Lecture 16: Models of Object Recognition

Frank Keller

School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

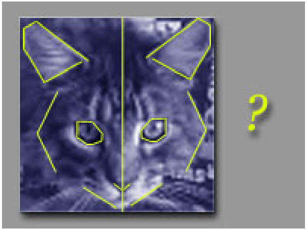
February 23, 2010

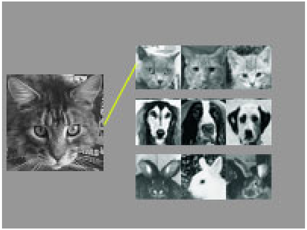
1. Motivation

2. Representations
   - Primal Sketch
   - Mid-lecture Problem
   - $2\frac{1}{2}$D Sketch
   - 3D Model

Motivation

Object recognition is the task of determining which of a given set of objects appears in an image.





An influential model of human object recognition was proposed by Marr (1982). It relies on convolutions and other matrix operations.

Motivation

Why is object recognition hard?

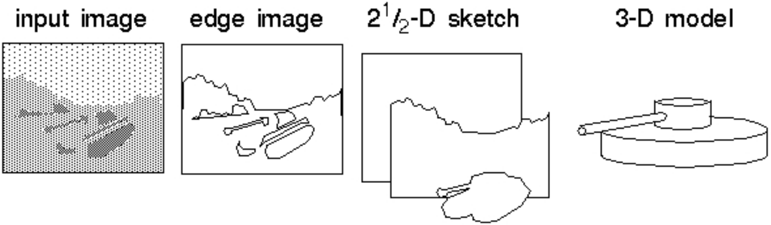
- a given object can cast an infinite number of different 2D images onto the retina;
- the object's position, pose, lighting, and background vary relative to the viewer;
- the object can be occluded by other objects;
- often the different objects in a given object category vary widely in their visual properties.

Marr's model breaks down the recognition problem by assuming that a sequence of increasingly complex *representations* of the image are constructed.

Representations

The input image (the perceived intensities) are successively converted:

- *primal sketch:* representation of the main light intensity changes in the visual input (edges, bars, blobs);
- *$2\frac{1}{2}D$ sketch:* representation of the orientation and depth of visible surfaces (motion, shading, shape, texture);
- *3D model:* hierarchically representation of volumetric primitives.



Primal Sketch

The primal sketch is a 2D representation of the main light intensity changes in the visual input.

It is computed by detecting the following *place tokens* (low level features):

- edges
- blobs
- bars
- terminations

Each of these is represented by a 5-tuple: type, position, orientation, scale, contrast.

Zero Crossings

The detection of *zero crossings* plays an important role in computing the primal sketch. This involves:

- blurring the image by convolving it with a *Gaussian kernel:* limits the intensity changes and facilitates edge detection;
- edge detection by convolving it with a *differential operator:* measures the difference between two adjacent pixels;
- identification of *zero crossings:* points where the difference changes from positive to negative or vice versa.

The zero crossings can then be used to identify edges, blobs, bars, terminations, as well as the orientations of these elements.

Zero Crossings

Marr uses the a two-dimensional Gaussian G kernel for blurring:

$$G(x, y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x^2+y^2)}$$

This works like the Gaussian filter discussed in the last lecture, but in two dimensions.

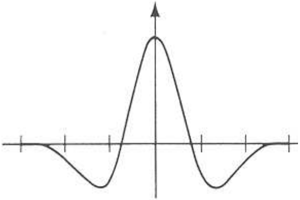
This is combined this with a Laplacian operator $\bigtriangledown^2$ for edge detection:

$$\bigtriangledown^2(x, y) = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

This operator computes the second partial derivatives in two dimensions.

Zero Crossings

The resulting function $G\bigtriangledown^2$ looks like this ("Mexican hat"):



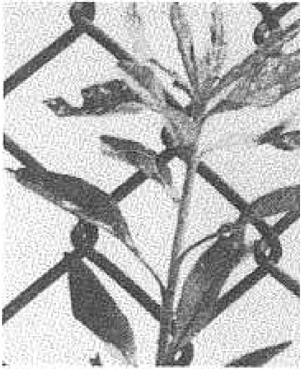
For edge detection, we then compute $G\bigtriangledown^2 * I$, where $I(x, y)$ is the intensity function of the image.

The $G\bigtriangledown^2$ kernel is *orientation independent,* in contrast to kernels such as $K = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ (vertical Sobel edge detector).

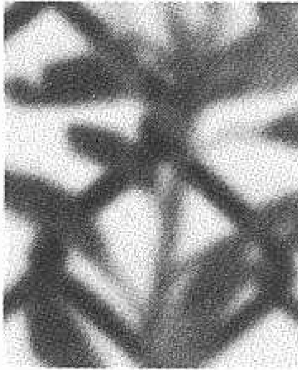
This models neurons in the visual cortex that respond to edges.

Blurring

Original image:

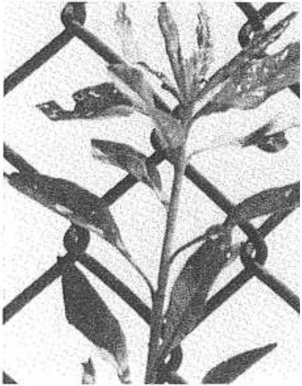


After convolution with G:

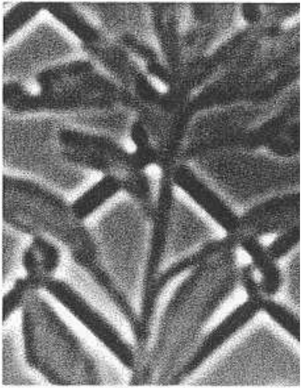


Edge Detection

Original image:



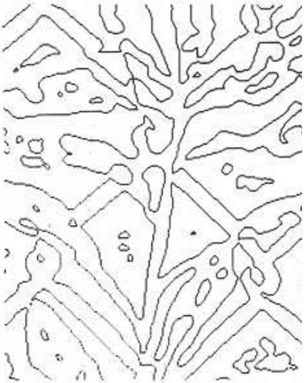
After convolution $G\bigtriangledown^2$:



Edge Detection

Positive values in black and negative values in white:

Zero crossings:



Mid-lecture Problem

The $G\bigtriangledown^2$ function can be applied using its exact mathematical definition, but this is quite expensive computationally. Often, it is approximated using a discrete kernel.

Define a discrete kernel K (a 3×3 matrix) that approximates the Laplacian $\bigtriangledown^2$.

Can you also define a discrete approximation for the Gaussian kernel?

How can the two kernels be combined to implement the $G\bigtriangledown^2$ function?

$2\frac{1}{2}$D Sketch

The primal sketch only represents intensity changes, it does not explicitly represent the properties of the objects in the image.

Example: suppose a primal sketch identifies a contour. This could be a marking on an object, or the boundary of the object.

The $2\frac{1}{2}$D sketch solves this problem by representing the *distance and orientation of visible surfaces.* Discontinuities in surface orientation and depth are explicitly encoded.

But: surfaces are not grouped into objects.

$2\frac{1}{2}$D Sketch

The $2\frac{1}{2}$D sketch assigns portions of the image to surfaces in the world and specifies distance and orientation relative to the viewer.

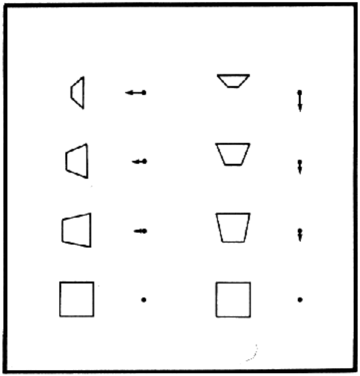
It is computed based on the following information:

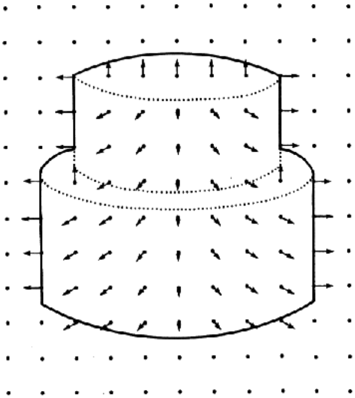
- motion
- shading
- color
- texture
- binocular disparity

Both the primal sketch and the $2\frac{1}{2}$D sketch are *observer centered,* i.e., viewpoint dependent.

Surface Primitives

Example: representation of a wedding cake in front of a wall.





Left: surface primitives; right: representation of the cake.

Surface Primitives

The $2\frac{1}{2}$D sketch on the previous slide:

- encodes the orientation relative to the viewer of each small patch of surface;
- represents orientation by an arrow perpendicular to each patch;
- represent arrows pointing directly at the viewer as dots (e.g., wall);
- indicates discontinuities in surface orientation as dotted lines;
- represents discontinuities in depth as black lines.

The next step is now to assemble the surface representations into object representations which can then be used to recognize objects.

3D Model

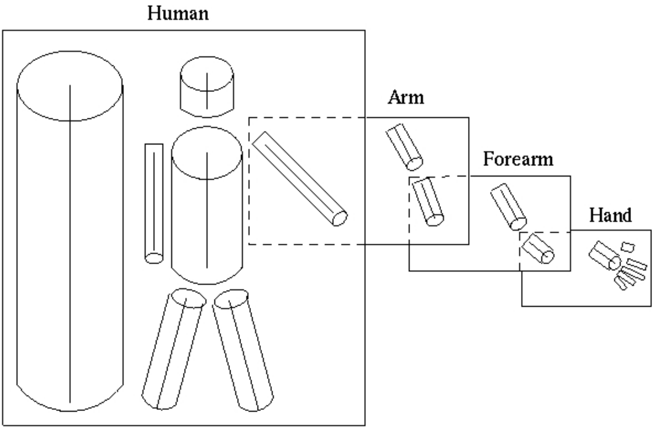
The 3D model represents shapes and their organization using a modular, hierarchical set of volumetric primitives. This representation is *model centered,* i.e., viewpoint independent.

The recognition process compares this representation against a stored catalog of 3D models.

3D models are organized in a hierarchy according to the specificity of information they carry. The decomposition happens along model's principal axis.

Recognition relies on three indices in the model catalog: specificity index, adjunct index, and parent index.

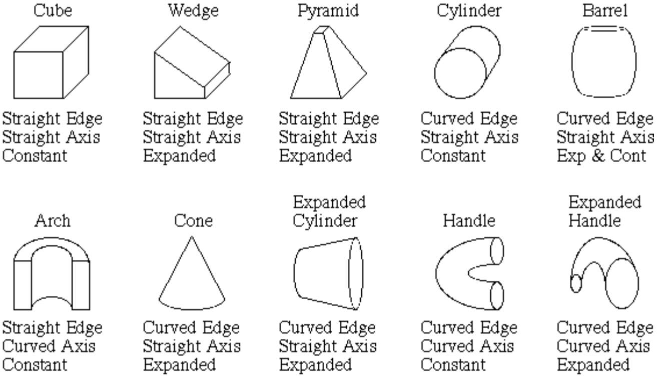
Volumetric Primitives



Each box corresponds to a 3D model. Left side of a box: model axis; right side: decomposition of the model's component axes.

Volumetric Primitives

This has been develop further, e.g., by replacing Marr's volumetric primitives with *geons,* a more complex set of primitives, which can also include curved edges and axes.



Summary

- Object recognition: determining which of a given set of objects appear in an image;
- difficult because images of objects are viewpoint dependent, subject to occlusion; objects of the same category vary;
- Marr's influential theory of object recognition assumes:
  - primal sketch representing main light intensity changes;
  - $2\frac{1}{2}$D sketch representing orientation and depth of surfaces;
  - 3D model representing objects hierarchically based on volumetric primitives;
- convolutions are crucial for computing the primal sketch (e.g., blurring, edge detection).

References

Marr, David. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, New York.