Computer Vision

Computer Science Tripos Part II

Dr Christopher Town

7. Lambertian and specular surface properties. Reflectance maps.

UNIVERSITY OF CAMBRIDGE

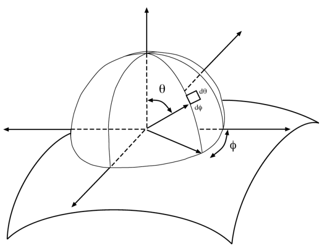
How can we infer information about the surface reflectance properties of objects from raw measurements of image brightness? This is a more recondite matter than it might first appear, because of the many complex factors which determine how (and where) objects scatter light.

- Surface albedo refers to the fraction of the illuminant that is re-emitted from the surface in all directions, in total. Thus, albedo corresponds more-or-less to "greyness."

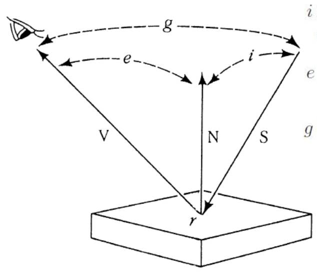
  The amount of light reflected is the product of two factors: the albedo of the surface, times a geometric factor that depends on angle.

Radiometry

- Questions:
  - how "bright" will surfaces be?
  - what is "brightness"?
    - measuring light
    - interactions between light and surfaces



Adapted from Computer Vision - A Modern Approach by D.A. Forsyth



i is the angle of the illuminant, relative to the surface normal N

e is the angle of a ray of light re-emitted from the surface

g is the angle between the emitted ray and the illuminant

Simplifying assumption: point light source

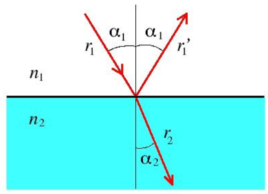
- A Lambertian surface is "pure matte." It reflects light equally well in all directions.

  Examples of Lambertian surfaces include snow, non-glossy paper, ping-pong balls, magnesium oxide, projection screens,...

  A Lambertian surface looks equally bright from all directions: the amount of light reflected depends only on the angle of incidence i of the illuminant, not on the angle of emission e.

- A specular surface is locally mirror-like. It obeys Snell's law (i.e. the angle of incidence of light is equal to the angle of reflection from the surface), and does not scatter light. Most metallic surfaces are specular.

Snell's law



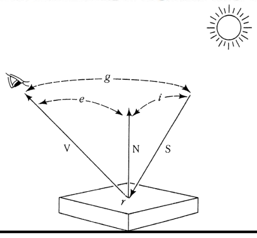
Forsyth & Ponce Figure 1.7

$$n_1 \sin\alpha_1 = n_2 \sin\alpha_2$$

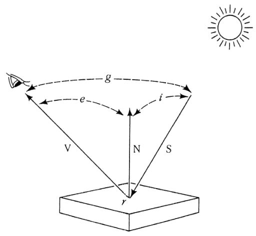
n_1 and n_2 are the indices of refraction of each material

- The reflectance map is a function $\phi(i, e, g)$ which relates intensities in the image to surface orientations of objects. It specifies the fraction of incident light reflected per unit surface area, per unit solid angle, in the direction of the camera; thus it has units of flux/steradian.

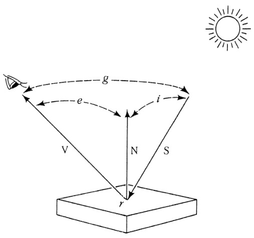
There are many types of reflectance functions, each of which is characteristic of certain surfaces and imaging environments. For a Lambertian surface, the reflectance function $\phi(i, e, g) = \cos(i)$. It looks equally bright viewed from all directions; the amount of reflected light depends only on angle of illumination.



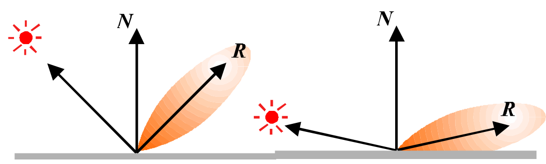
For surfaces such as the dusty surface of the moon, the reflectance function $\phi(i, e, g)$ depends only upon the ratio of the cosines of the angles of incidence and emission: $\cos(i)/\cos(e)$, but not upon their relative angle g nor upon the surface normal N. In case you ever wondered, this is why the moon looks like a penny (i.e. flat) rather than a sphere. Even though the moon is illuminated by a point source (the sun), it does not fade in brightness towards its limbs (as N varies). Surfaces with this property are called *lunar* surfaces.



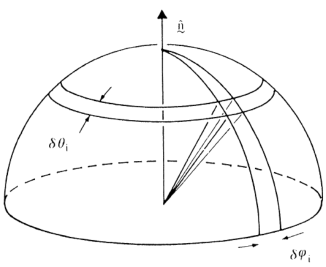
For a specular surface, the reflectance function $\phi(i, e, g)$ is especially simple: $\phi(i, e, g) = 1$ when $i = e$ and both are coplanar with the surface normal N, so $g = i + e$ (Snell's law for a pure mirror); and $\phi(i, e, g) = 0$ otherwise.



Specular Reflection

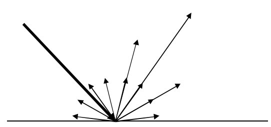


Typically there is not just one point source of illumination, but rather a multitude of sources (such as the extended light source provided by a bright overcast sky). In a cluttered scene, much of the light received by objects has been reflected from other objects (and coloured by them...) One needs almost to think of light not in terms of ray-tracing but in terms of thermodynamics: a "gas" of photons in equilibrium inside a room.



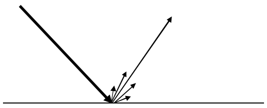
Diffuse Reflection

- Reflection uniformly in all directions
  - Matte (Non-shiny) appearance
  - Eg, chalk
- Most materials are not *ideally* diffuse



Specular Reflection

- Light reflects in a single direction
  - Shiny
  - Eg, silvered mirror
- Most materials are not *ideally* specular



Computer Vision

Computer Science Tripos Part II

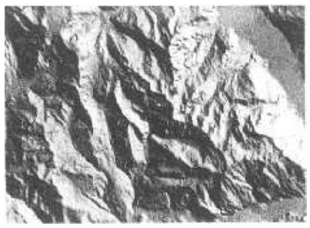
Dr Christopher Town

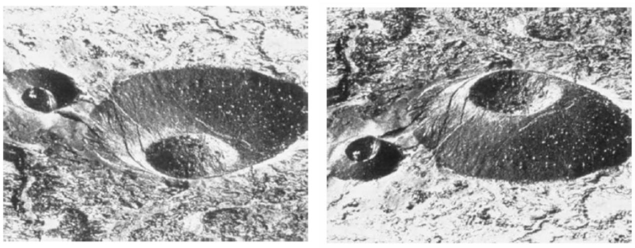
8. Shape description. Codons; superquadrics and surface geometry.

UNIVERSITY OF CAMBRIDGE

As with all of these problems, computing "shape-from-shading" requires the disambiguation of many confounding factors. These arise from the

1. geometry of the illuminant (e.g. is the light a point source or extended? If a point source, where is it?) Are there several light sources? How will these affect the shading and shadowing information?
2. reflectance properties of the surface. What kind of surface is it – e.g. Lambertian, or specular, or a combination of both?
3. geometry of the surface (its underlying shape). Are shadows cast?
4. rotations of the surface relative to perspective angle and illuminant.
5. variations in material and surface reflectance properties across space (e.g. variation from Lambertian to specular where skin becomes more oily).
6. variations in surface albedo ("greyness")

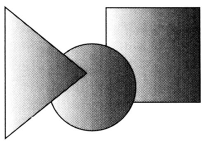




A. Pentland. Local shading analysis. *Trans. PAMI*, 6:170–187, 1984.

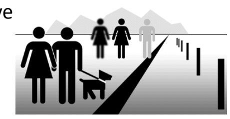
This picture is of an ash cone in the Hawaiian Islands (courtesy of W. Richards).

- Occlusion

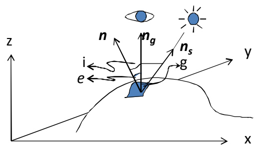


Lighting and colours

- Relative size

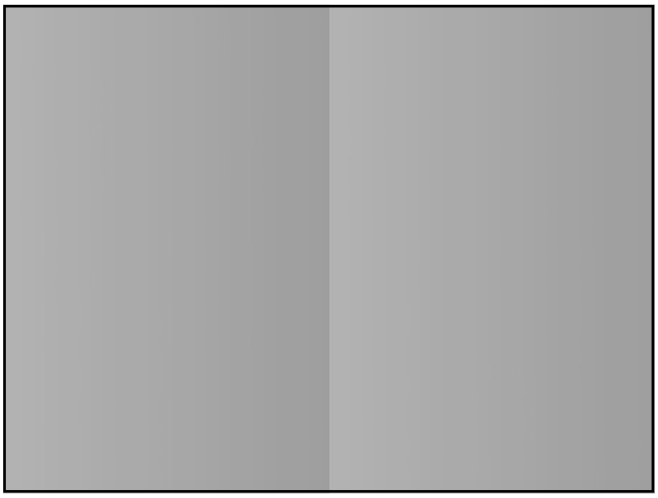


Surface Orientation from Reflectance Models

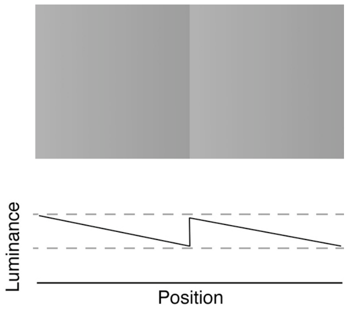


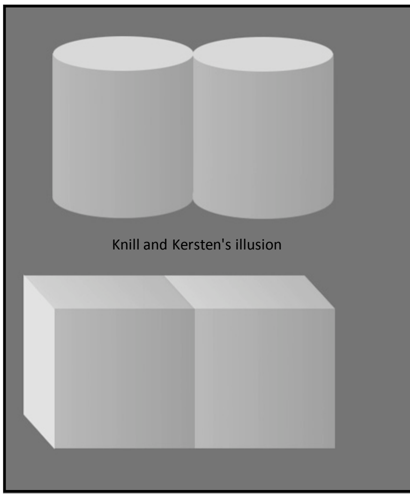
i = angle of incidence
e = angle of emittance
g = phase angle
n, $\mathbf{n_g}$, and $\mathbf{n_s}$ are unit vectors along the surface normal, view direction, and source direction, respectively.

L(*i, e, g*) = *s* C^n + (1 – *s*) cos (i) 0 <= *s* <= 1.
The larger the value of n, the sharper sharper the peaking in the specular direction.



Craik-O'Brien-Cornsweet effect





This illusion highlights the importance of scene interpretation.

← The effect is gone

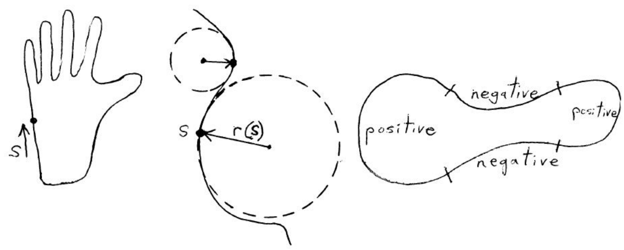
← and it comes back when the gradient is not explained by the shape.

How should shape be represented? Boundary descriptors; codons.

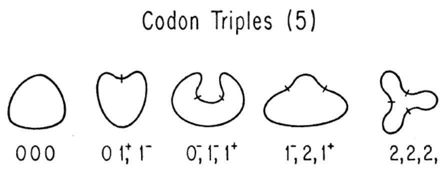
Closed boundary contours can be represented completely by their curvature map $\theta(s)$ (the reciprocal of the local radius of curvature $r(s)$ as a function of position s along the contour). This is the *Fundamental Theorem of Curves.* Local radius of curvature $r(s)$ is defined as the limiting radius of the circle that best "fits" the contour at position s, in the limit as the arc length Δs shrinks to 0, and the local curvature of the contour at that point is:

$$\theta(s) = \lim_{\Delta s \to 0} \frac{1}{r(s)}$$

$$\theta(s) = \lim_{\Delta s \to 0} \frac{1}{r(s)}$$



Closed boundary contours can be expanded with basis functions (such as "Fourier descriptors" of the radius of curvature) from their curvature map, in order to generate a shape description that is invariant to translation, rotation, and dilation. By cataloguing a list of all possible combinations of changes in sign of the curvature map relative to the zeroes of curvature, it is possible to generate a restricted "grammar" for the shapes of closed contours. A lexicon of all possible shapes having a certain number of zeroes-of-curvature generates a list of "codons," from which shapes can be classified and recognised.



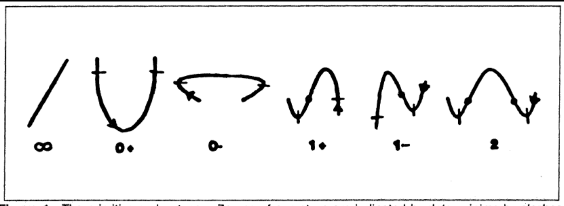


Figure 1 The primitive codon types. Zeroes of curvature are indicated by dots, minima by slashes. The straight line (∞) is a degenerate case included for completeness, although it is not treated in the text.

The boundaries between the codons are taken at the positive minima and negative maxima of curvature. The sign of curvature is determined by a clockwise (−) or counterclockwise (+) rotation of the tangent as one traverses the curve, keeping the "figure" to the left. The codon type is then specified by the presence or absence of an inflection (zero of curvature) and whether the positive maxima of curvature occurs before or after the inflection, when present. There are only five possible codon types, which are illustrated in Figure 1.

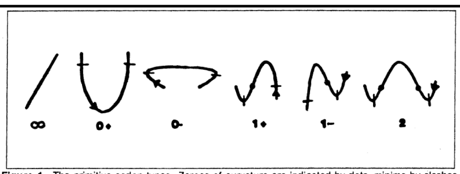


Figure 1 The primitive codon types. Zeroes of curvature are indicated by dots, minima by slashes. The straight line (∞) is a degenerate case included for completeness, although it is not treated in the text.

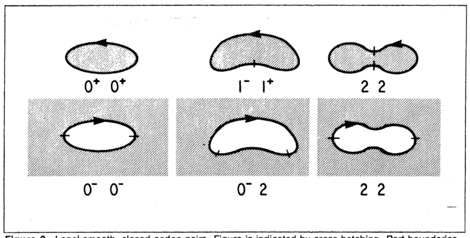


Figure 2 Legal smooth, closed codon pairs. Figure is indicated by cross hatching. Part boundaries are noted by the slashes.

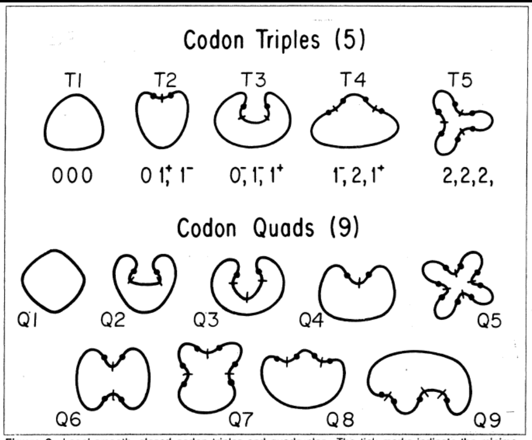
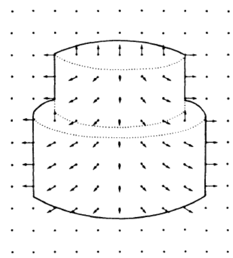


Figure 3 Legal smooth, closed codon triples and quadruples. The tick marks indicate the minima of negative curvature, which are the part boundaries, whereas the dots show the inflections.

The curvature map $\theta(s)$ together with a "starting point" tangent $t(s_o)$ specifies a shape fully. Some nice properties of curvature-map descriptions are:

1. The description is position-independent (i.e., object-centred).
2. The description is orientation-independent (rotating the shape in the plane does not affect its curvature map).
3. The description represents mirror-symmetric shapes simply by a change in sign:
$$\theta(s) \rightarrow \theta(-s)$$
4. Scaling property: Changing the size of a shape simply scales $\theta(s)$ by the same factor. The zero-crossings are unaffected.
$$\theta(s) \rightarrow K\theta(s)$$



The "2.5-dimensional" sketch

David Marr

Viewer centred			Object centred
Input Image	Primal Sketch	2 1/2-D Sketch	3-D Model Representation
Perceived intensities	Zero crossings, blobs,edges, bars, ends, virtual lines, groups, curves boundaries.	Local surface orientation and discontinuities in depth and in surface orientation	3-D models hierarchically organised in terms of surface and volumetric primitives

Shape/Object Representation and Recognition

Objects/Shapes

2-D (Planar)

Representation of closed boundaries (Recognition Sensitive to Noise and Occlusions) *e.g.,* Fourier Descriptors

Noise and Occlusion-tolerant Representation of parts and their interrelationships (e.g., Connectivity, adjacency). Parts could be regular shaped objects such as circles, squares, triangles recognizable by Hough transform, or curve segments represented using polynomial forms (B-Splines)

3-D

Volumetric/binary voxel representation (*e.g.* for *3-D* medical image constructed from 2D-slices)

Surface based representations(*e.g.* Coon's surface patches represented by 2-D polynomials, generalized cylinders)

Dr Chris Town

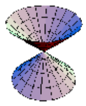
Implicit Surfaces

- Quadric surfaces
  – Implicit second-order polynomial equations
- Superquadric surfaces
  – A generalization of quadric surfaces
- Blobby objects (metaballs)
  – A collection of spherical density functions

Dr Chris Town

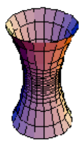
Quadric surfaces

- Double cones $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=0$

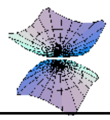


- Ellipsoids $\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$

- Hyperboloids of one sheet $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$



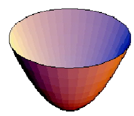
- Hyperboloids of two sheets $\frac{z^2}{c^2}-\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$



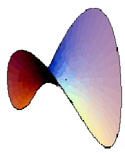
Dr Chris Town

Quadric surfaces

- Elliptic paraboloids $\frac{x^2}{a^2}+\frac{y^2}{b^2}=\frac{z}{c}$



- Hyperbolic paraboloids $\frac{y^2}{b^2}-\frac{x^2}{a^2}=\frac{z}{c}$

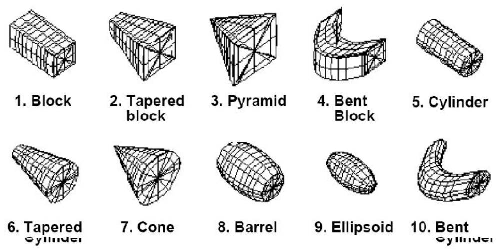


Dr Chris Town

3D Object-centred coordinates. Superquadrics.

Represent solids by the unions and intersections of generalised superquadric objects, defined by equations of the form:

$$Ax^\alpha + By^\beta + Cz^\gamma = R$$



A. Pentland, 1986.

Dr Chris Town

3D Object-centred coordinates. Superquadrics.

Matlab demo

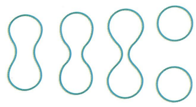
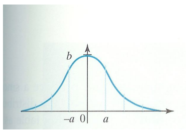
```
superquad(1, 1, 1, 1, 1, 100);

% SUPERQUAD - generates a superquadratic surface
%
% Usage:  [x,y,z] = superquad(xscale, yscale, zscale, e1, e2, n)
%
% Arguments:
%        xscale, yscale, zscale  - Scaling in the x, y and z directions.
%        e1, e2       - Exponents of the x and y coords.
%        n            - Number of subdivisions of logitude and latitude on
%                         the surface.
%
% Returns: x,y,z  - matrices defining paramteric surface of superquadratic
%
% If the result is not assigned to any output arguments the function
% plots the surface for you, otherwise the x, y and z parametric
% coordinates are returned for subsequent display using, say, SURFL.
%
% Examples:
%     superquad(1, 1, 1, 1, 1, 100)   -  sphere of radius 1 with 100 subdivisions
%     superquad(1, 1, 1, 2, 2, 100)   -  octahedron of radius 1
%     superquad(1, 1, 1, 3, 3, 100)   - 'pointy' octahedron
%     superquad(1, 1, 1, .1, .1, 100) -  cube (with rounded edges)
%     superquad(1, 1, .2, 1, .1, 100) - 'square cushion'
%     superquad(1, 1, .2, .1, 1, 100) -  cylinder
```

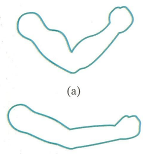
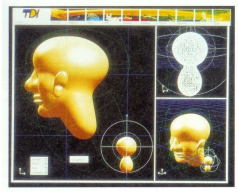
Dr Chris Town

Blobby Objects

- A collection of density functions

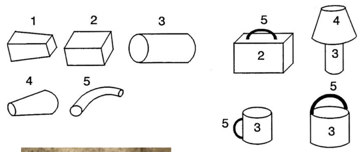
- Equi-density surfaces

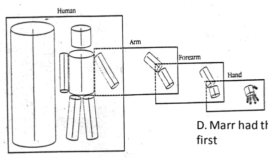
r Chris Town

High-level shape representations

Biederman: 36 geons and a compositional system
Compositionality
-- viewpoint-independent recognition



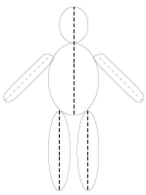




D. Marr had this idea first

Dr Chris Town

Generalised cylinders



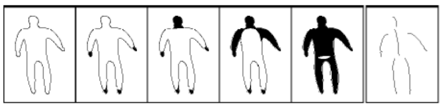
- component parts have axis
- cross section function describes variation along axis
- good for articulated objects, such as animals, tools
- can be extracted from intensity images with difficulty

Stockman CSE/MSU Fall 2008

Dr Chris Town

Extracting a model from a segmented image region

Rough generalized cylinder model of a person.



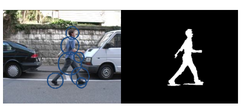
Constructing a generalized cylinder approximation from a 2D shape.

Courtesy of Chen and Medioni

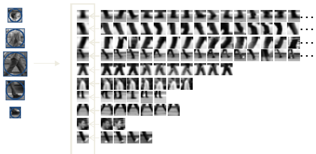
Stockman CSE/MSU Fall 2008

Dr Chris Town

Recap: Implicit Shape Model – Representation

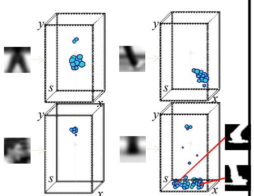


Training images
(+reference segmentation)



Appearance codebook

- Learn appearance codebook
  - Extract local features at interest points
  - Clustering ⇒ appearance codebook
- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object

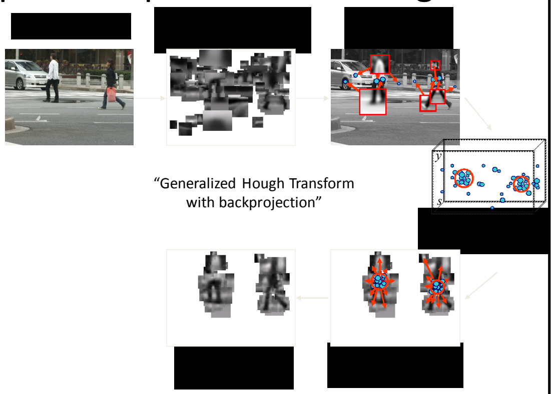


Spatial occurrence distributions
+ local figure-ground labels

B. Leibe

Dr Chris Town

Implicit Shape Model – Recognition



"Generalized Hough Transform with backprojection"

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Dr Chris Town

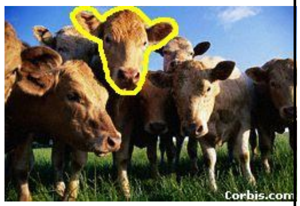
Objects = surfaces (regions) + boundaries (borders)

Surfaces (regions) have:
- brightness (mean luminance)
- colour (spectral composition)
- texture (energy at different orientations, scales)
- surface depth
- surface orientation (slant, tilt)
- surface curvature (bump, dent)
- transparency
- shadowing
- velocity
- velocity gradient

A given surface is perceived as part of an object or part of the background

Dr Chris Town

Inferences from images: figure-ground segmentation



Region analysis: surface attributes
- reflectance (colour, lightness)
- texture
- depth, surface orientation

Boundary analysis: shape
- orientation, curvature, angles
- border "ownership"

Bottom-up: feature analysis

Top-down: attention, prior knowledge

Dr Chris Town