ANALYSIS OF 3D SCENES

Contrast:

- Analysis of 2-dimensional images
  (of 2D or 3D scenes)

- Analysis of 3-dimensional scenes

3-dimensional scene analysis still uses, as input, 2-dimensional images.

The nature of information which we want to extract from the images is different, e.g.

- depth information for 3D description
- motion
- temporal changes

Depth computation

If the depth for each point on a surface is known, this surface can be reconstructed in 3D.

Depth can be either measured or derived.

Direct measurements

- Range detectors (e.g. laser light, ultrasound)
- In range images the pixel intensity is a measure of distance between a sensor and a point in the image.
- Edges (discontinuities) in the range image indicate sharp distance changes, and hence object edges in 3D space
- Very helpful for 2D and 3D analysis if combined with intensity images.

Depth measured indirectly from grey level images

The main schemes are:

- depth from stereo images
- shape (and depth) from shading
- depth from texture
- depth from optical flow

Stereo images

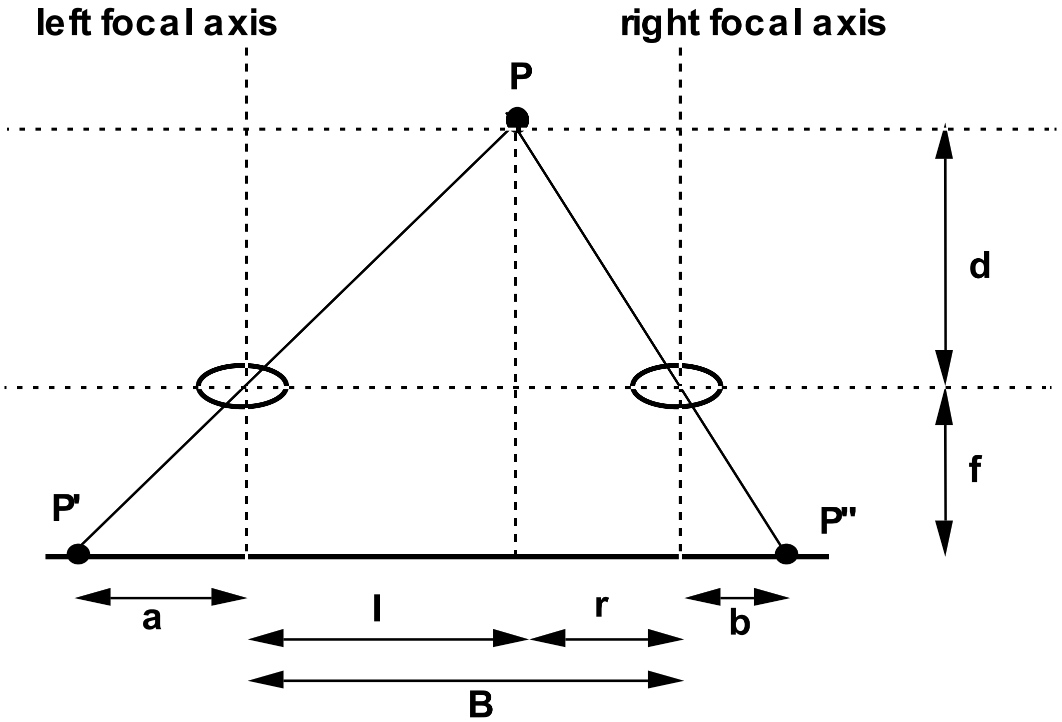
The principle of stereo vision (binocular imaging)

- based on geometrical model
- two images taken from two different positions can give clues for relative depth
- baseline and focal length must be known
- disparity can be found from the image

Stereo vision in humans

The outline of the technique

- take two images separated by a baseline
- match points between two images
- derive two lines (linking each of the "eyes" with a point in space)
- find the lines intersection
- calculate depth



$$\frac{d}{l} = \frac{d + f}{l + a} \qquad \frac{d}{r} = \frac{d + f}{r + b}$$

$$\boxed{d = \frac{fB}{(a + b)}}$$

Shape from shading

Under uniform illumination it is possible to calculate a surface orientation (and hence the surface shape) from surface shading.

- Light is reflected from a surface in a direction of the viewer
- Amount of light depends on various angles and the surface properties

- Brightness of a *Lambertian surface* (ideally matt) depends only on the direction of a light source (the angle it makes with a surface normal) and the surface property

- Isobrightness lines are the lines along which brightness is the same

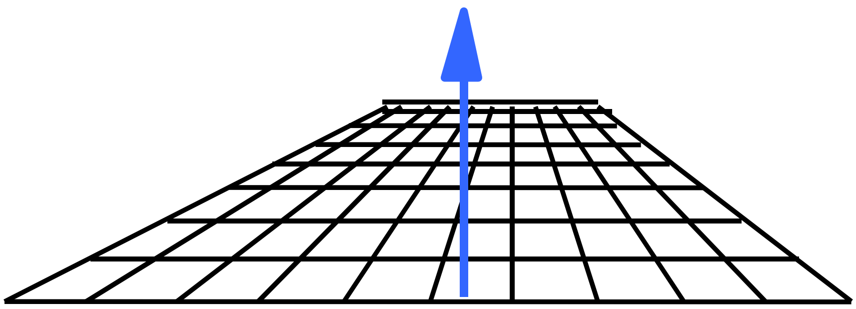
- Isobrightness lines projected onto a flat surface constitute *reflectance maps* (like maps in cartography)

- Surface direction is recovered from shading

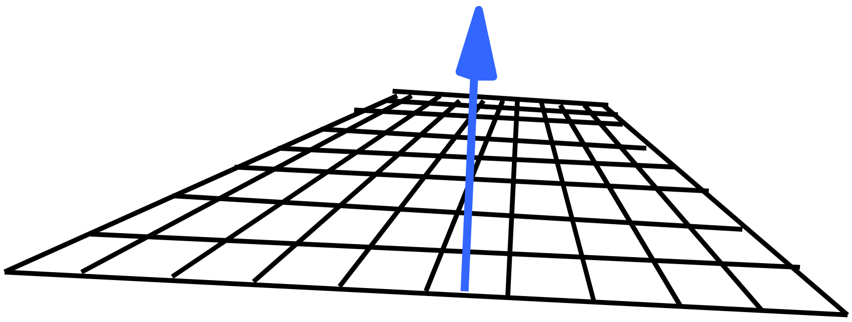
Texture gradient

Surface orientation can be derived from images of textured surfaces

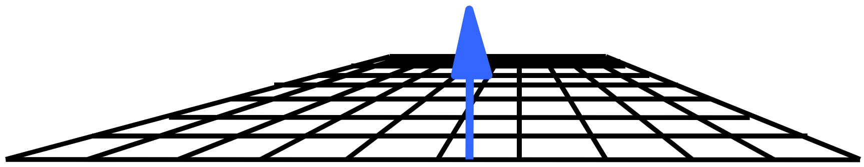
- Texture is first segmented into primitives
- The maximum rate of change of projected primitive sizes is the direction of the texture gradient

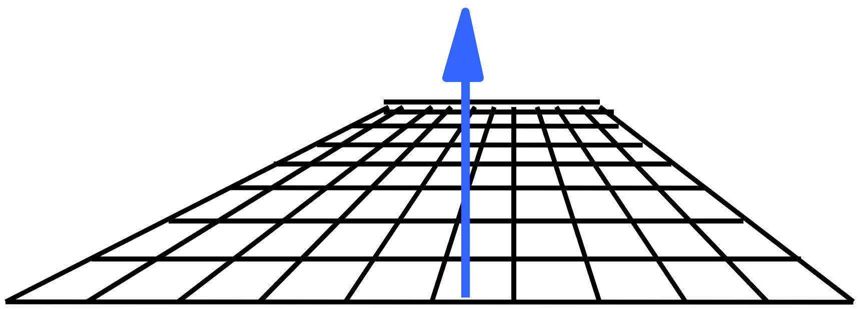


- The orientation of the direction determines a rotation of the plane about the camera line of sight



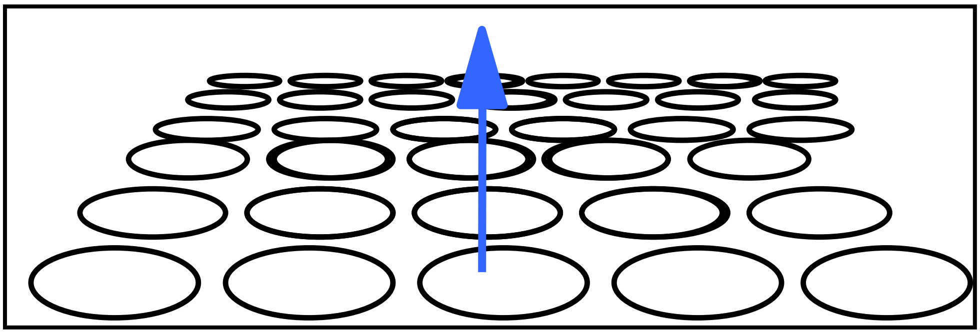
- The magnitude of gradient determines how much the plane is tilted with respect to the camera





- Knowing a real shape of a texture primitive, orientation and tilt can be evaluated

- For textures in the form of regular grids a vanishing point can be calculated



- Regular pattern projected on a smooth surface makes it textured

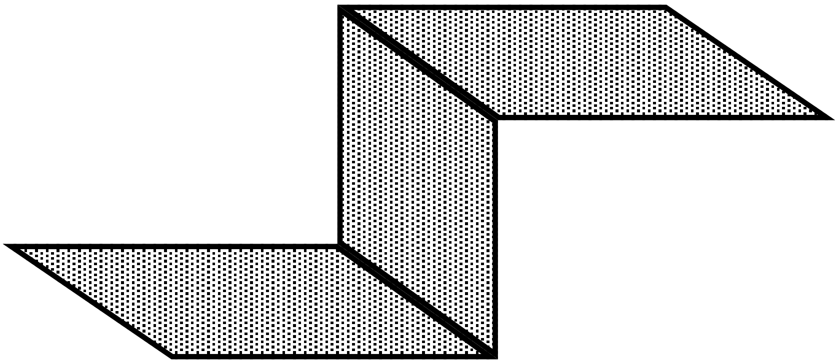
- Surface shape can be accurately evaluated from this texture

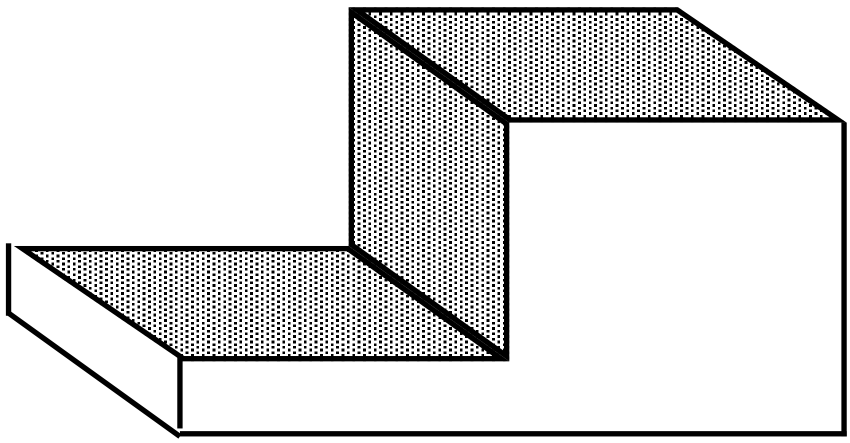
- If shape of a texture primitive is not known or its distribution is irregular, a stereo-pair of textured images can be used for depth reconstruction.

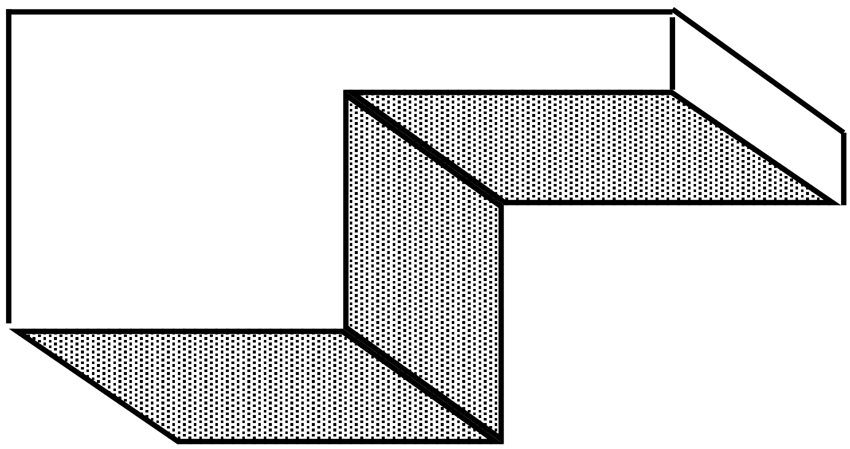
SURFACE RECOVERY THROUGH NATURAL CONSTRAINTS

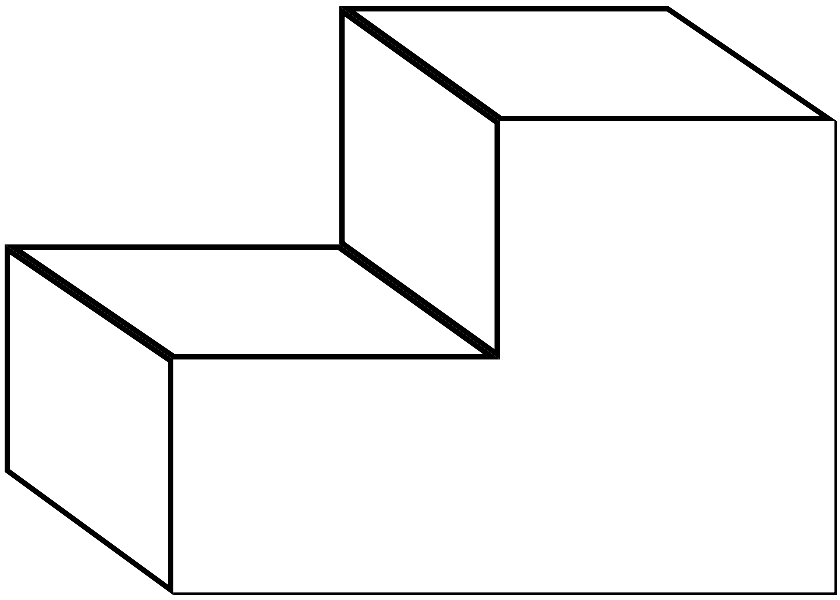
A model can be based on a set of <u>constraints</u> (e.g. on the properties or relationships among the parts pixels, regions, etc).

The pictures whose descriptions satisfy these constraints constitute the class defined by this <u>declarative</u> model.





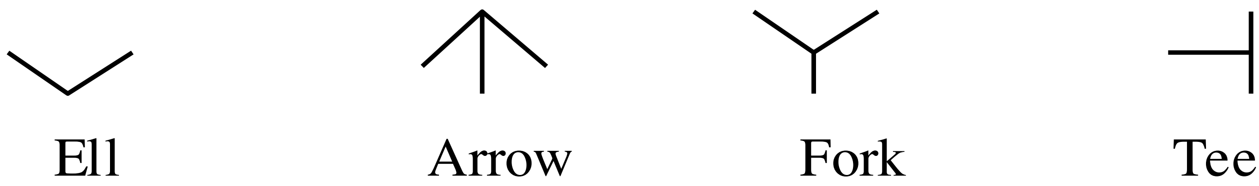


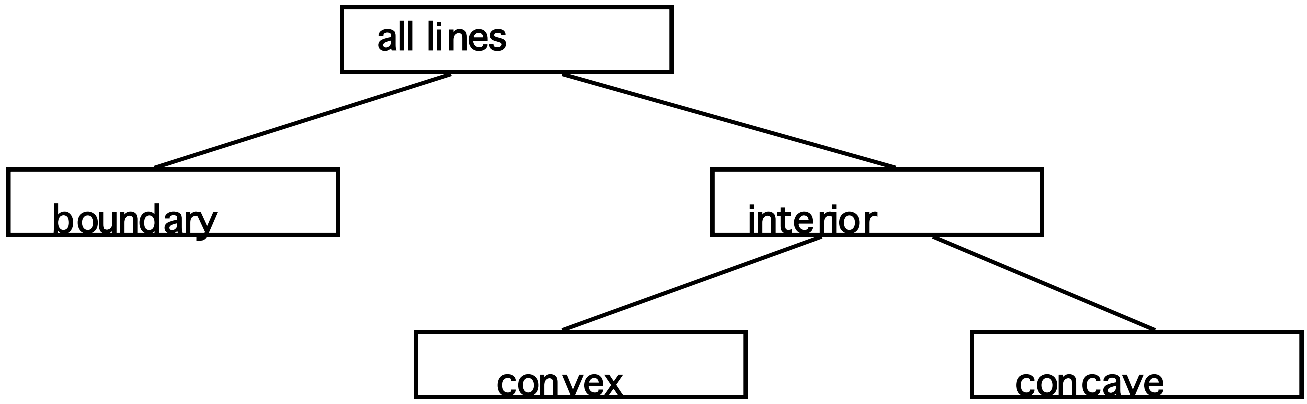


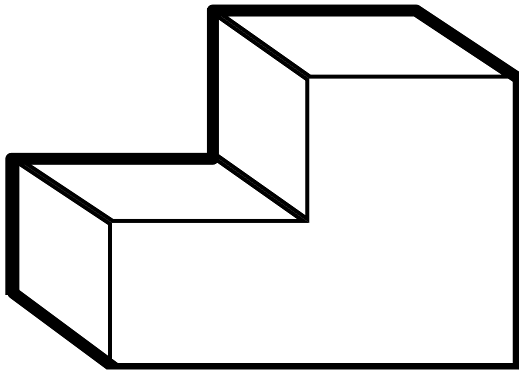
Polyhedral scene labelling

Vertices in three-dimensional polyhedral scenes generate junctions in line drawings of the scenes.

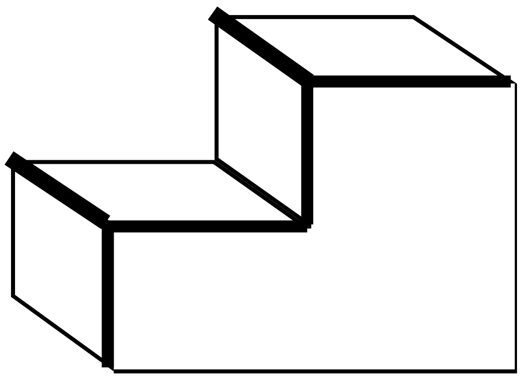
If there are no shadows or cracks and all vertices are three-faced then only four types of junctions can occur in the drawing:



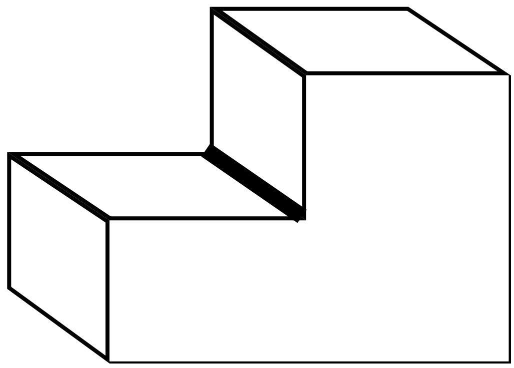

all lines
boundary
interior
convex
concave



Boundary



Interior convex



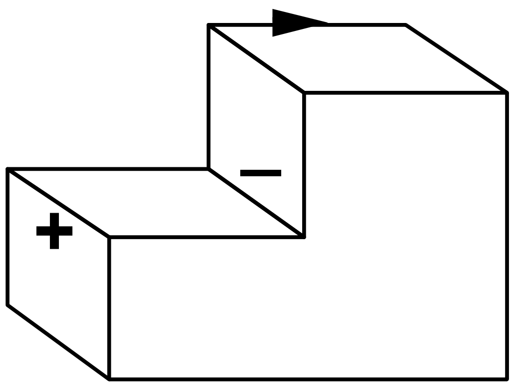
Interior concave

Line types are identified on drawings by line labels:

+ convex lines

- concave lines

> boundary lines

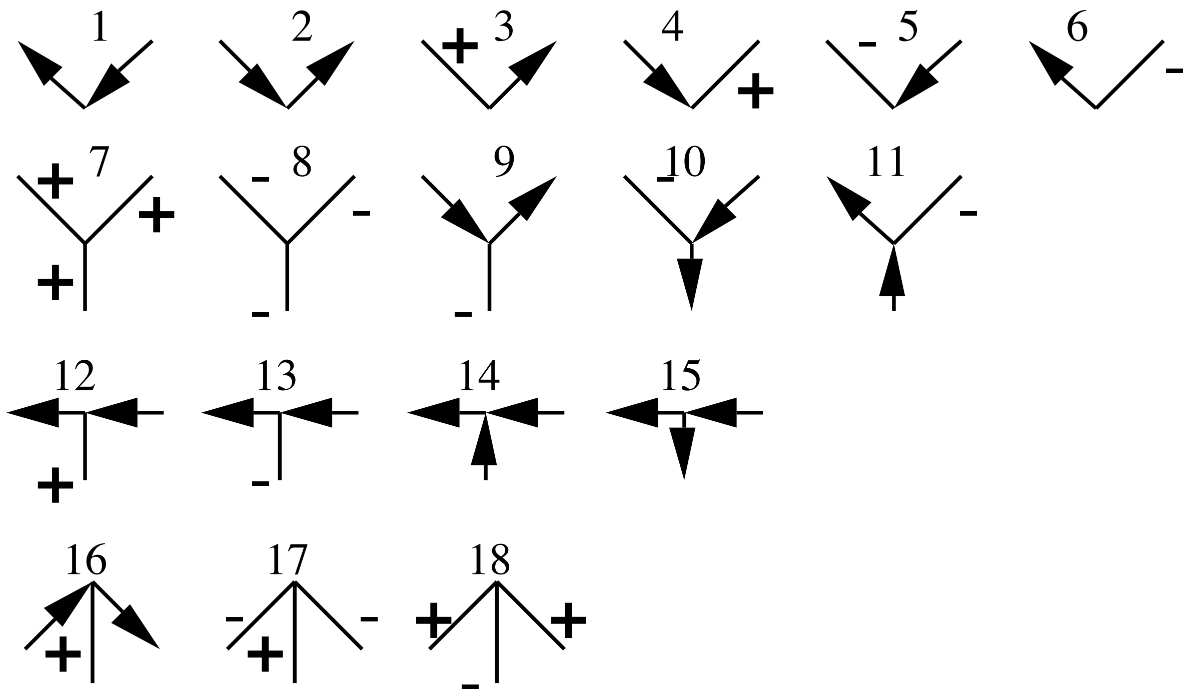


Combinations of line labels surrounding junctions are called junction labels.

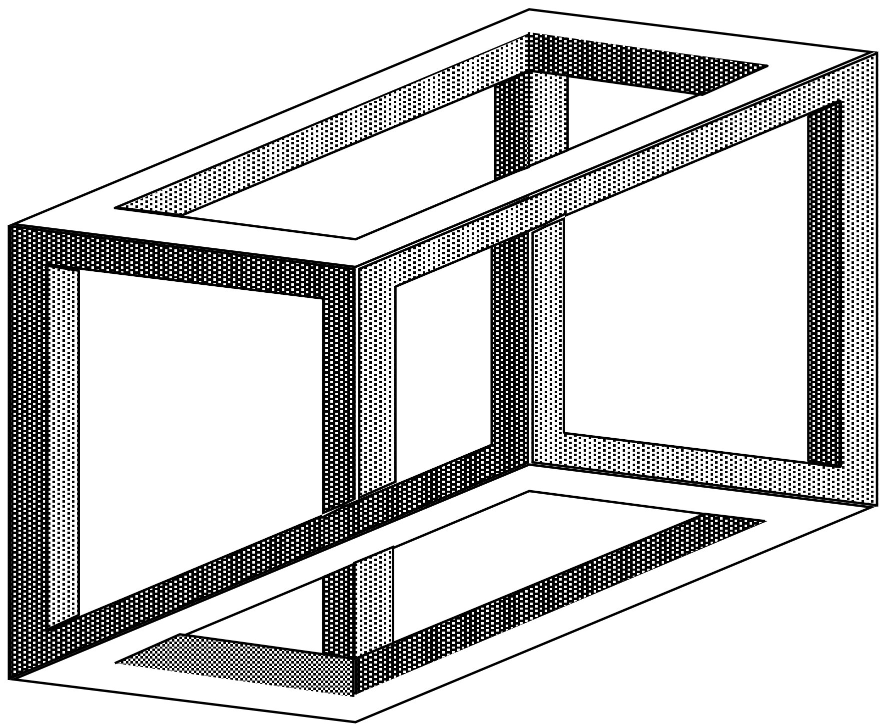
There are 208 all possible junction labels.

Only 18 occur in reality:

Junction “catalogue”



By correctly labelling junctions in line drawings it is possible to derive three-dimensional interpretation of the polyhedral scene and detect “impossible” polyhedra.



Overall scheme

- Create the catalogue of junctions
- Find correct labels by constraint propagation

Creating the catalogue of junctions

“Look” at every possible three-faced vertex from every possible direction.

Finding correct labels

Three-dimensional objects are projected onto a two-dimensional surface.

Three-dimensional information can be recovered from this two-dimensional image by imposing geometrical constraints:

- law of projection
- law of 3D world

- Start with labelling external boundary lines

  (they are 'hit' when traversing an image starting from the image edge)

- Use knowledge of the <u>possible junctions</u>
  (as in the catalogue)

  and the knowledge that line labels must be <u>consistent</u> between the junctions.

It could be a tedious process, especially for complex drawings.

The constraint propagation is a possible technique.

Constraint propagation

Constraint propagation techniques use constraints of a particular domain to achieve

global consistency via **local** computation.

Labelling through constraint propagation:

In general

- Given a finite input set, a set of labels, and a set of constraints,

  find a consistent labelling, i.e. assign labels to objects without violating the constraints.

- Initially each object is given all the legal labels according to unary constraints (junction type).

- Constraints involving more and more objects are incorporated successively as each object is checked to see if it is consistent with the new constraint.

- At each step the new object's constraint is propagated, i.e. the structure is checked if it is consistent with the new constraint.

Labelling applied to polyhedral junctions:

- To solve the problem of consistent junction labelling, each junction is initially assigned all the labels it can legally have.
- All the junctions are examined in turn and consistency of labelling between the current junction and all its neighbours is examined.
- If any of the labels is illegal in this context, it is removed and all the neighbouring junctions are checked themselves for consistency.
- This continues until no more changes are occurring.

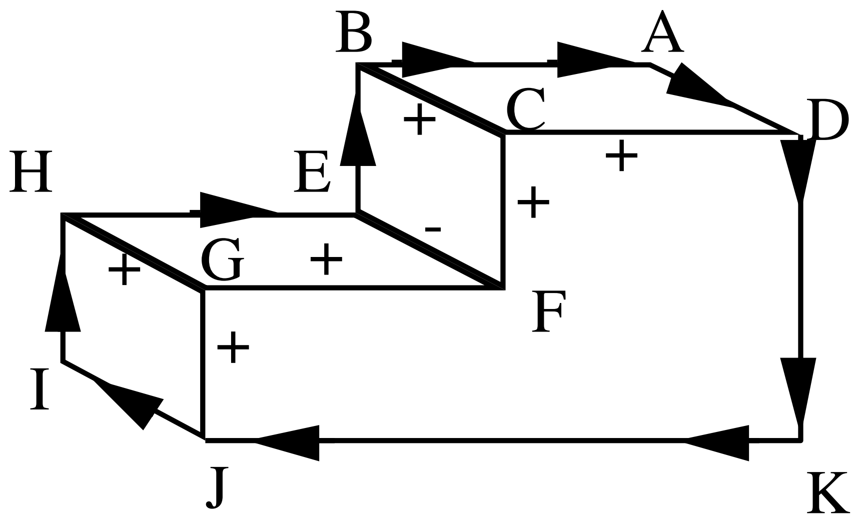
Waltz algorithm for propagating symbolic constraints.

1 Form a queue consisting of all junctions

2 Until queue is empty repeat:

- **2.1** Remove the first element from the queue; call it a current junction.
  - **2.1.1** If the current junction has never been visited, create a stack of junction labels for it consisting of all possible junction labels for the junction type involved. Note that the stack change has occurred.
  - **2.1.2** If any junction label from the current junction's stack is incompatible with all the junction labels in any neighbouring junction's stack, eliminate that incompatible label from the current junction's stack. Note that the stack change has occurred.
- **2.2** If a stack change has occurred, for each neighbouring junction with a stack which is not on the queue, add that junction to the front of the queue.

The Waltz algorithm to label the junctions in the line drawing



The labelled polyhedron