
Introduction to Artificial Intelligence

Vision

Bernhard Beckert



UNIVERSITÄT KOBLENZ-LANDAU

Winter Term 2003/2004

Outline

- **Perception generally**
- **Image formation**
- **Early vision**
- **2D / 3D**
- **Object recognition**

Perception Generally

Stimulus / perception depends on the world

S: **Stimulus**

W: **World**

g: **“graphics”**

$$S = g(W)$$

Perception Generally

Stimulus / perception depends on the world

S : **Stimulus**

W : **World**

g : **“graphics”**

$$S = g(W)$$

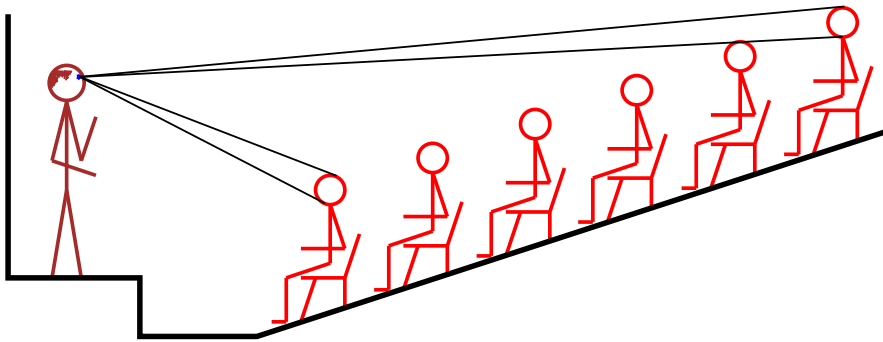
Needed for vision

The world as a function of the stimulus

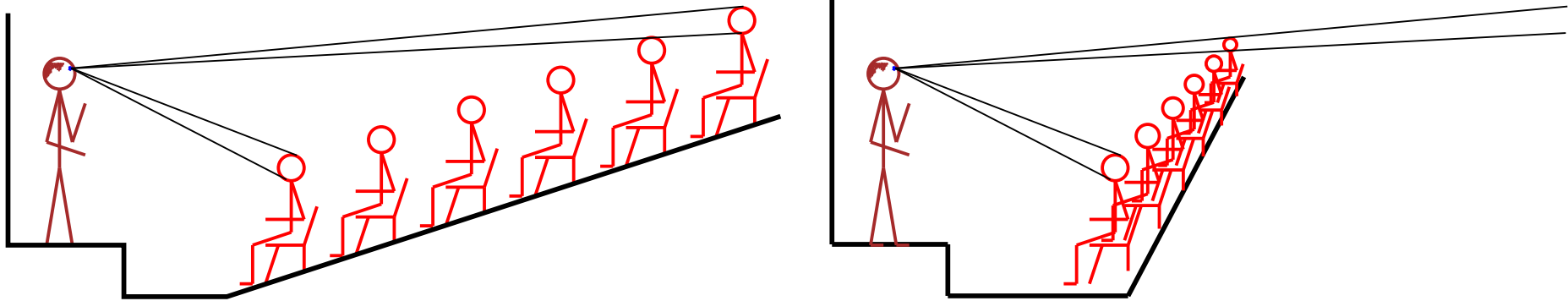
$$W = g^{-1}(S)$$

Problem: Massive ambiguity!

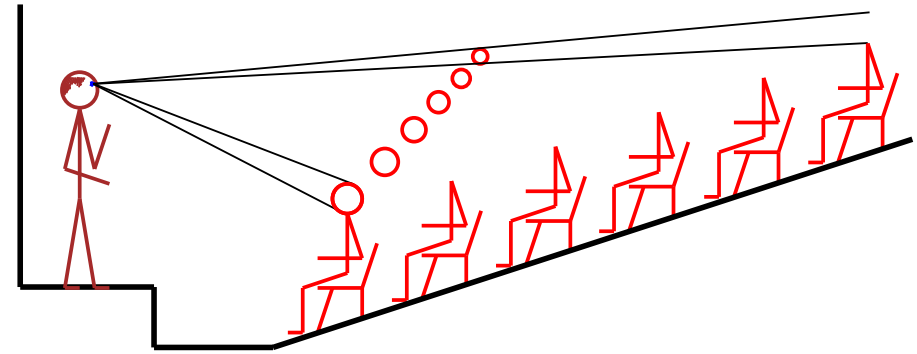
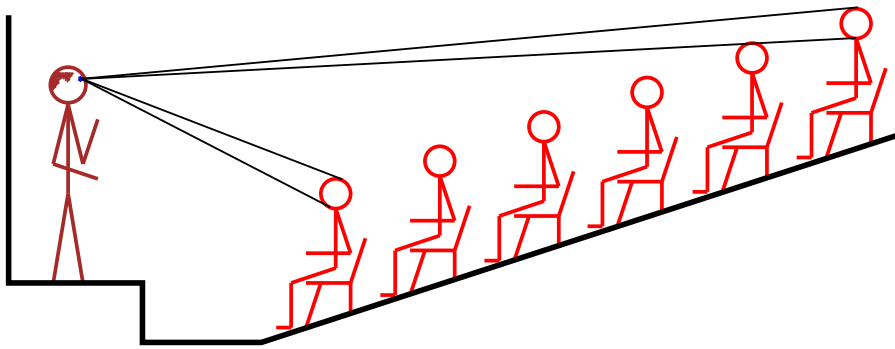
Perception: Ambiguity



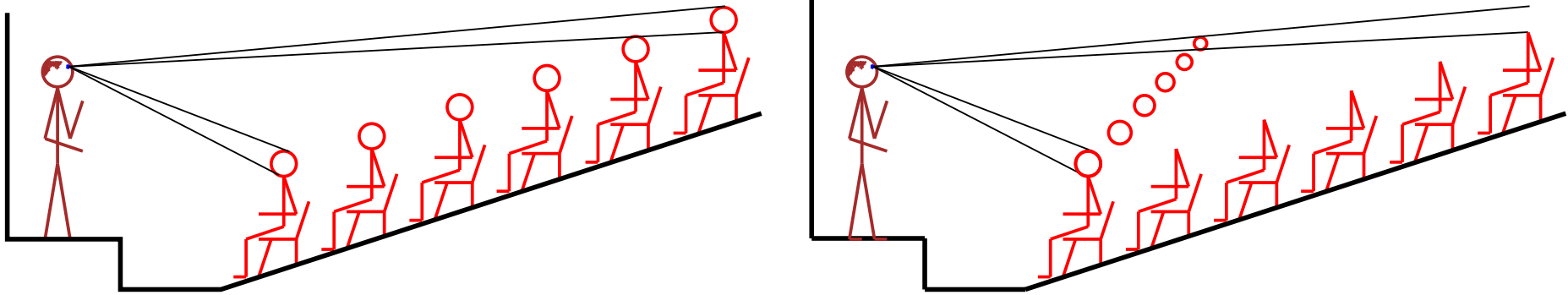
Perception: Ambiguity



Perception: Ambiguity



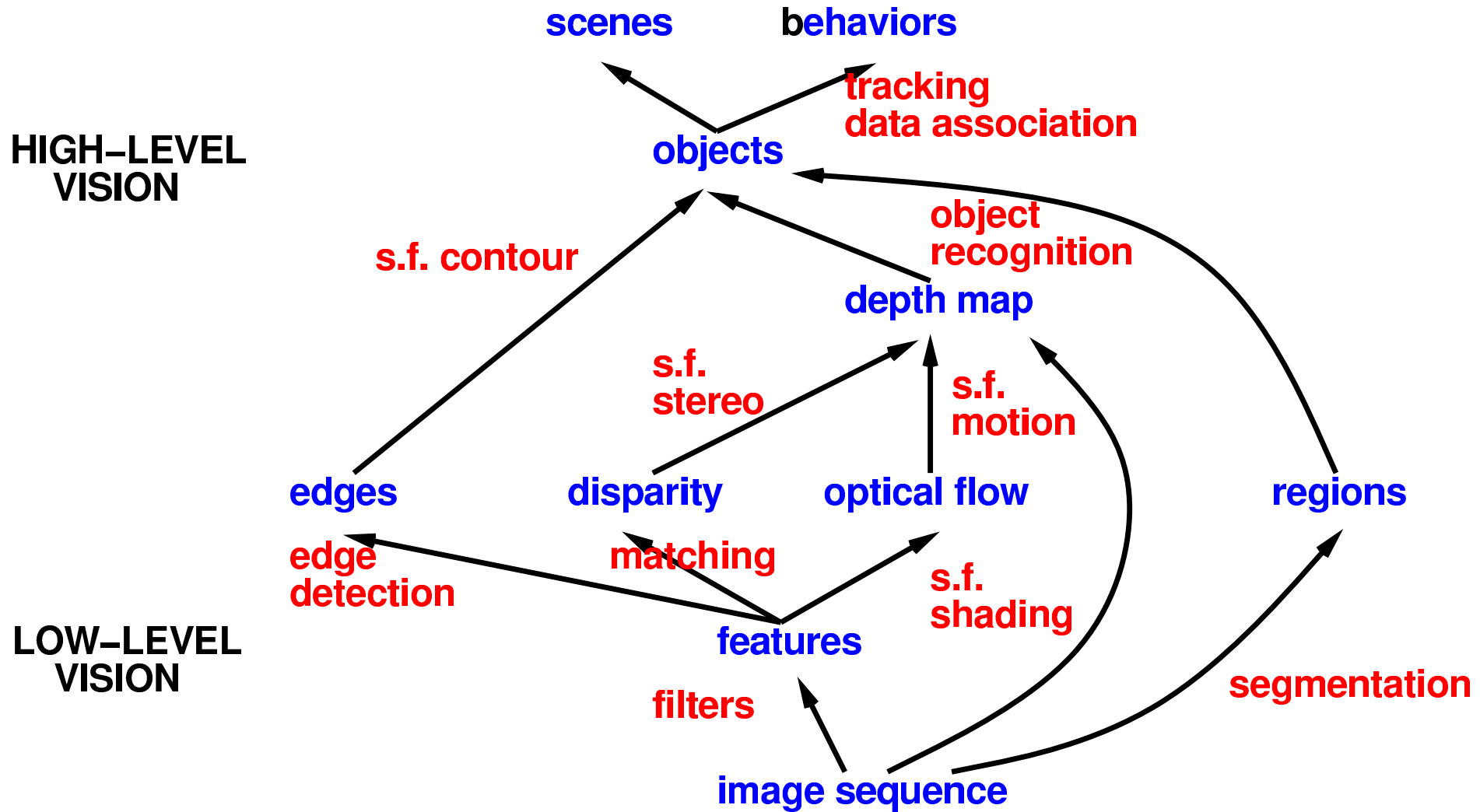
Perception: Ambiguity



Therefore vision requires

- to combine multiple cues
- to use cues from a-priori knowledge about the environment

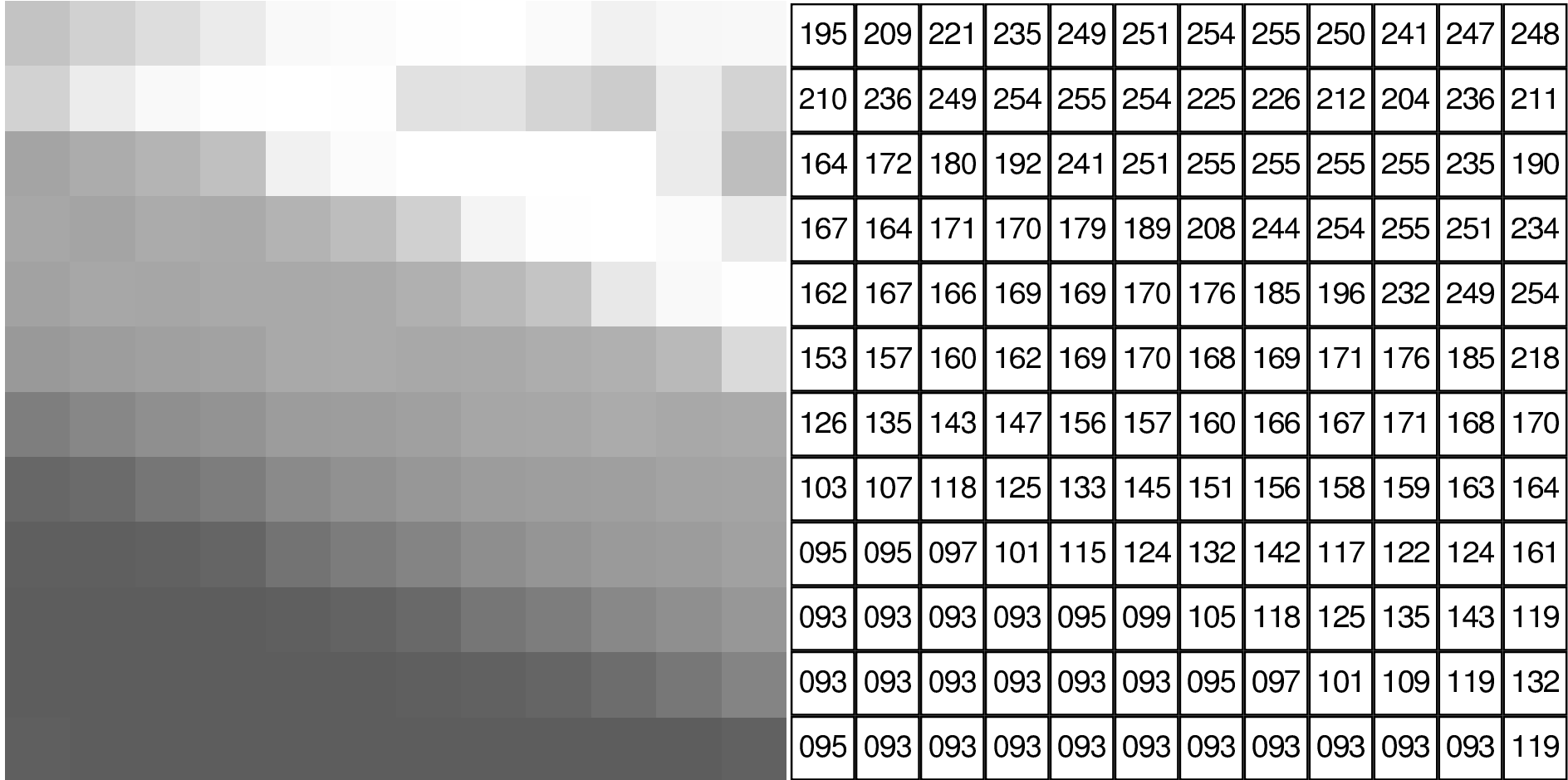
Vision "Subsystems"



Images



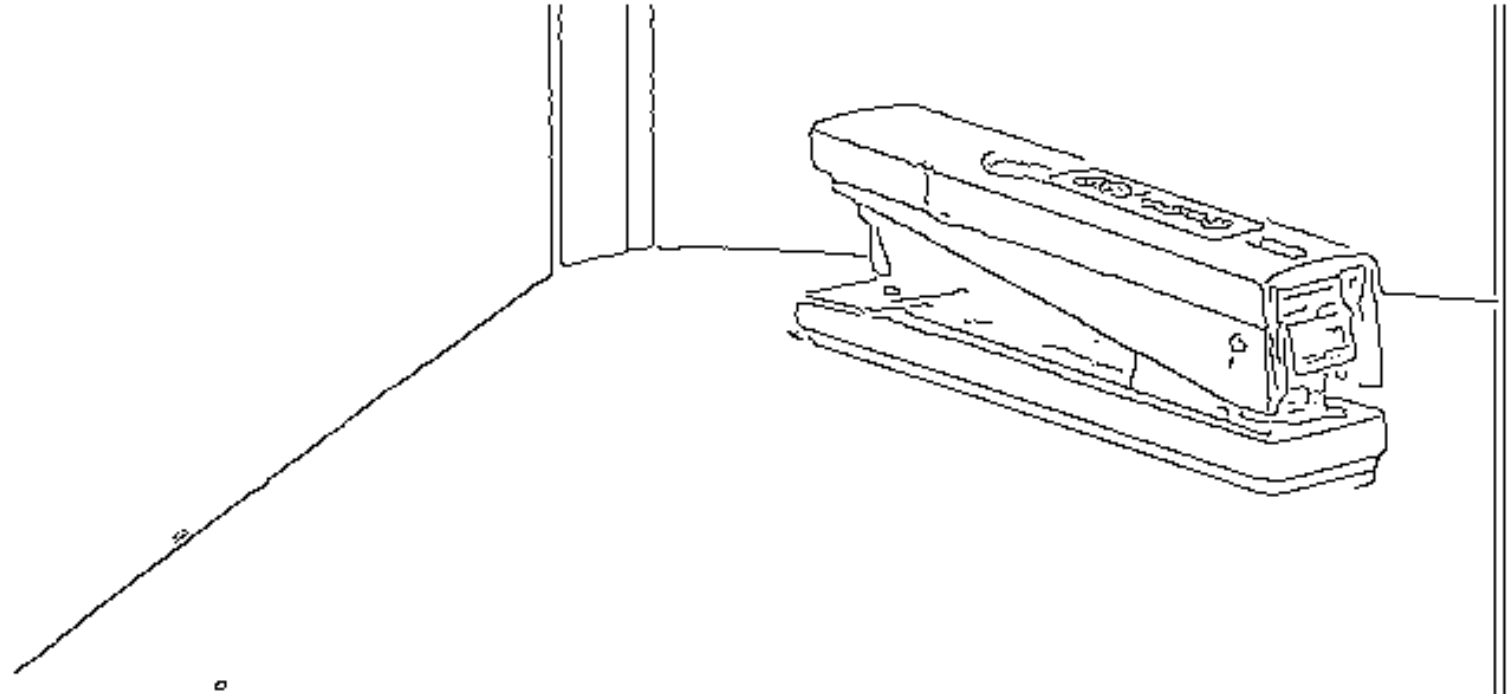
Images



CCD camera \approx **4,000,000 pixels**

human eyes \approx **240,000,000 pixels** **(0.25 terabit/sec)**

Edge Detection



Edges in image = discontinuities in scene

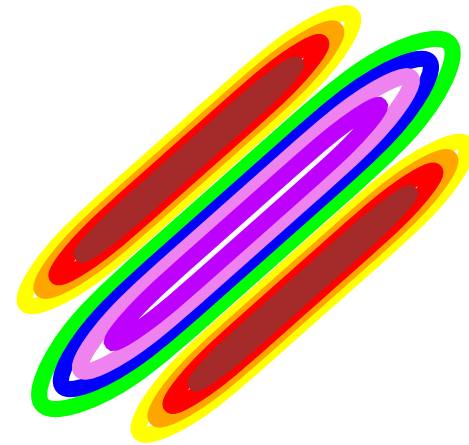
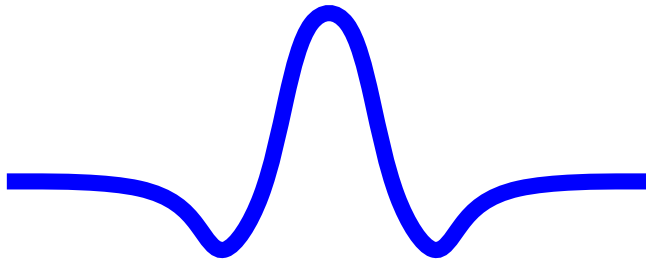
- depth
- surface orientation
- reflectance (surface markings)
- illumination (shadows, etc.)

Edge Detection

First step

Convolve image with spatially oriented filters

$$E_{\theta}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{\theta}(u, v) I(x + u, y + v) du dv$$



Edge Detection

Second step

Label above-threshold pixels with edge orientation

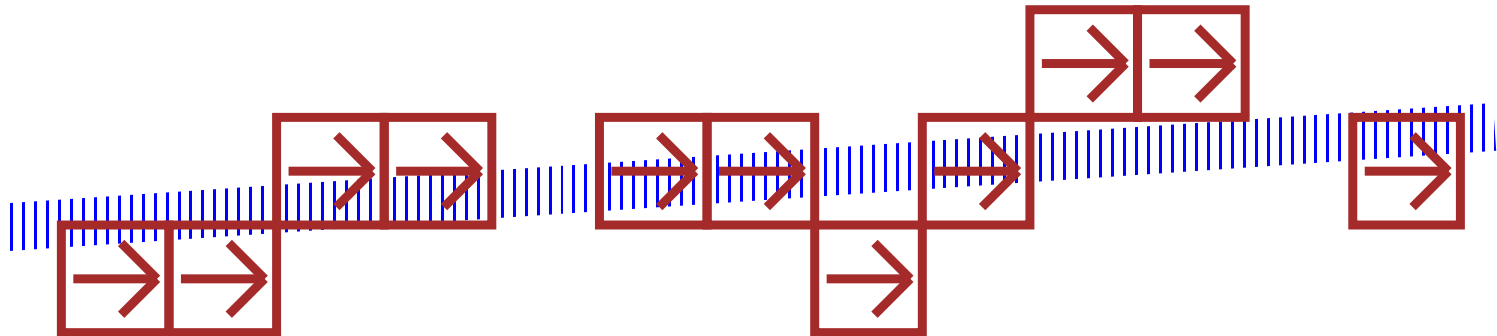
Edge Detection

Second step

Label above-threshold pixels with edge orientation

Third step

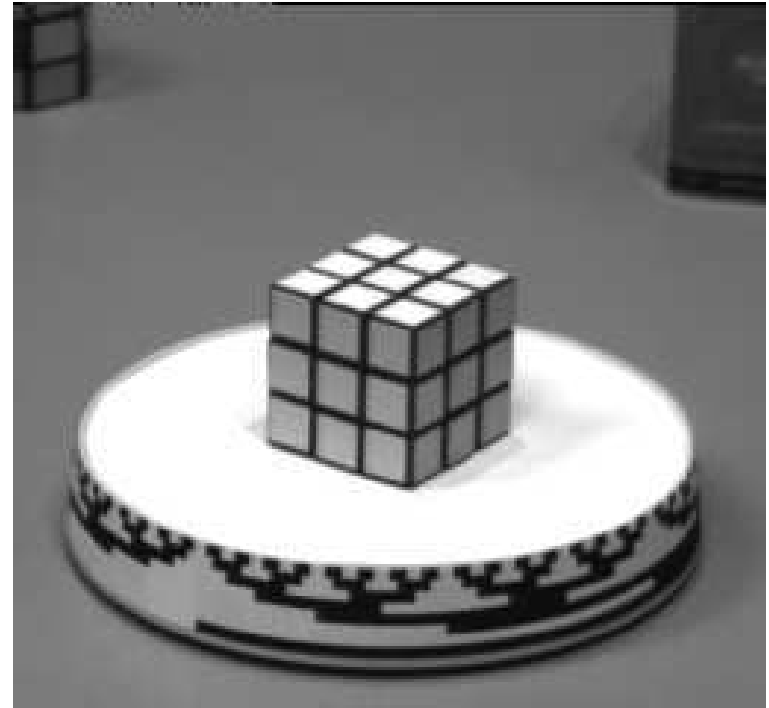
Infer “clean” line segments by combining edge pixels with same orientation



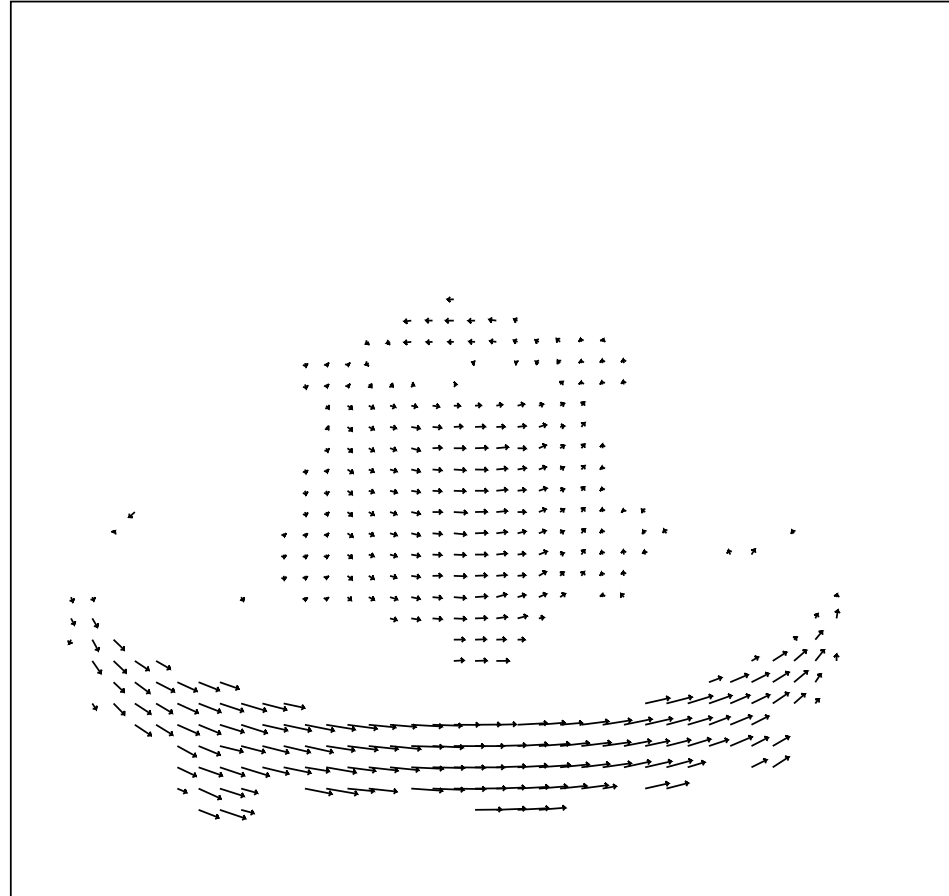
Cues from A-priori Knowledge

Cue...	Assumes
motion	rigid bodies, continuous motion
stereo	solid, non-repeating bodies
texture	uniform texture
shading	uniform reflectance
contour	minimum curvature

Motion



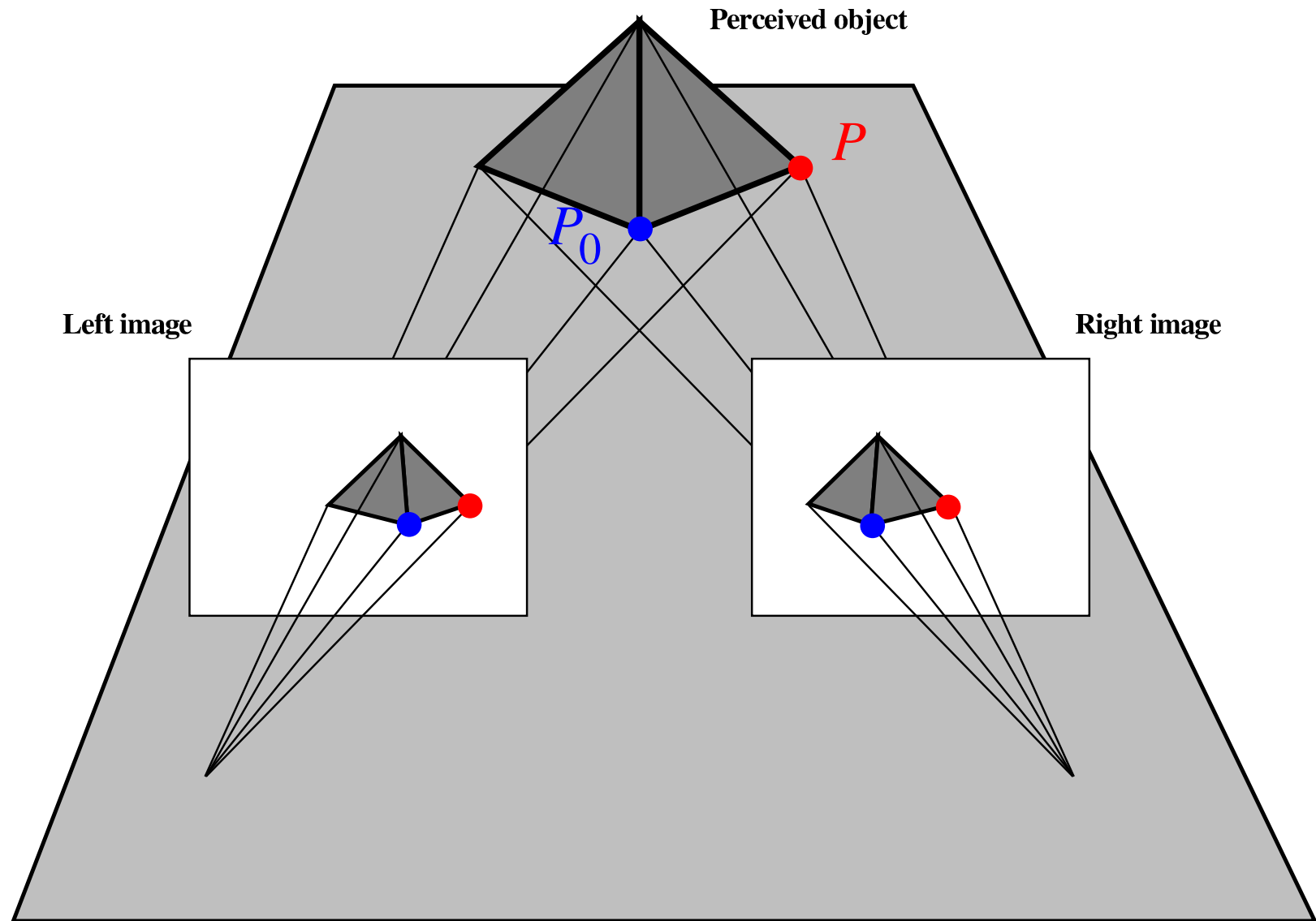
Motion



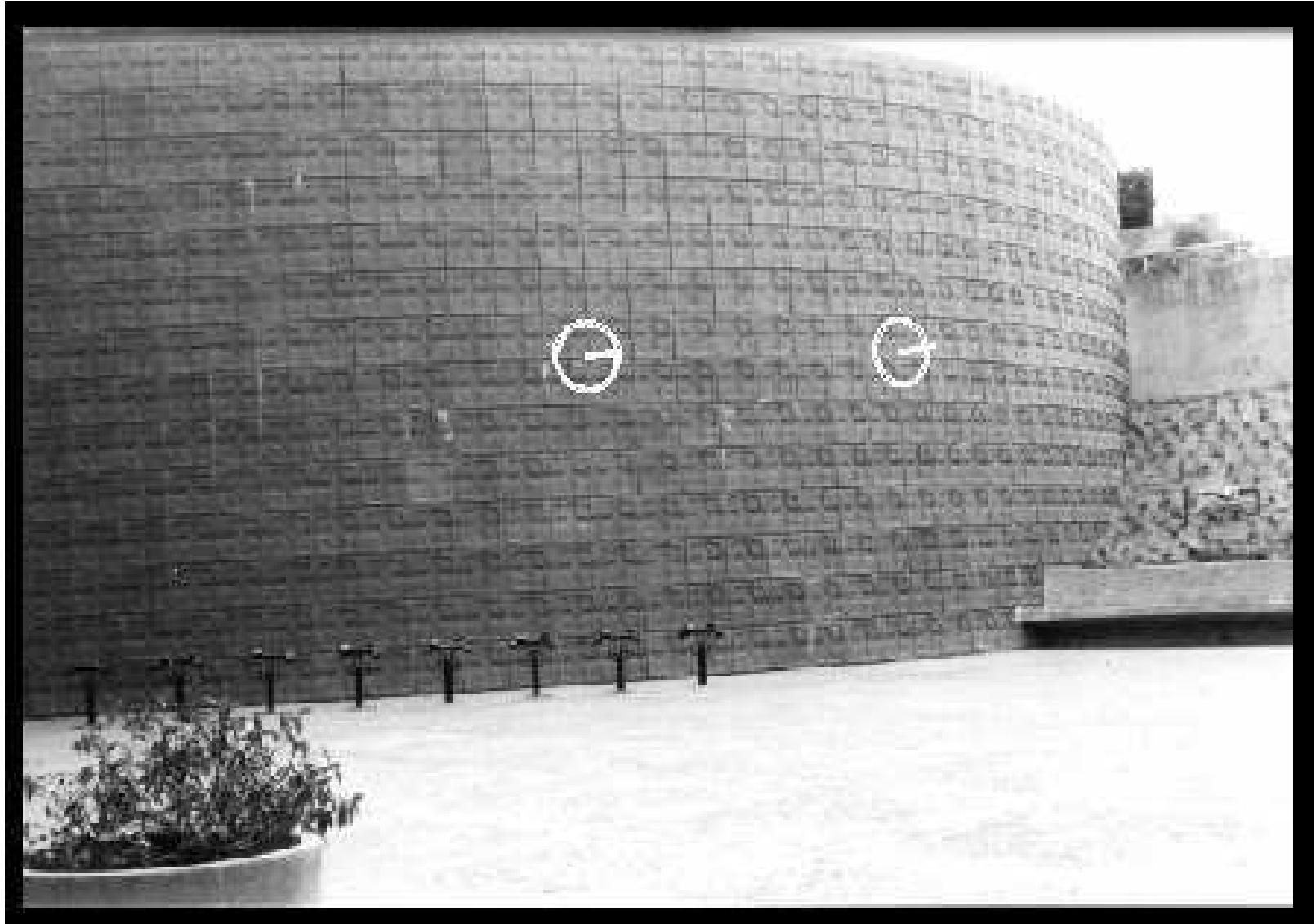
Note

Shape determines optical flow due to a moving body

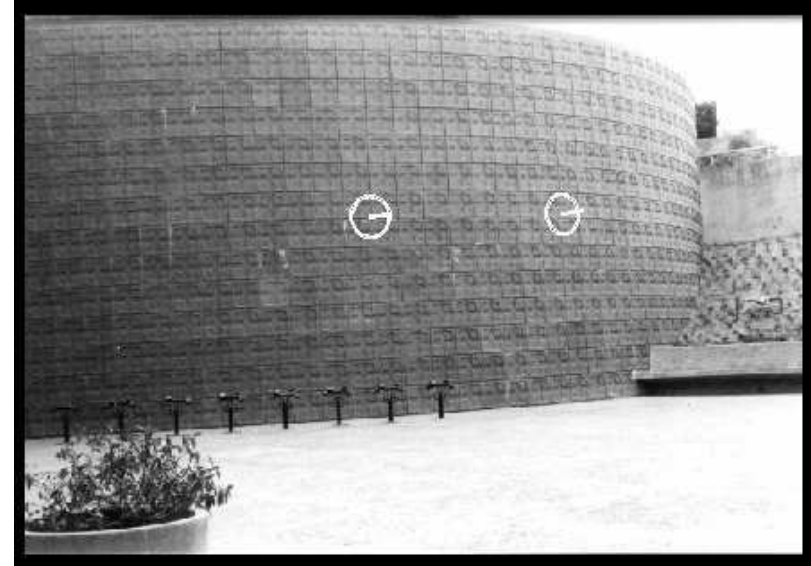
Stereo



Texture



Texture



Idea

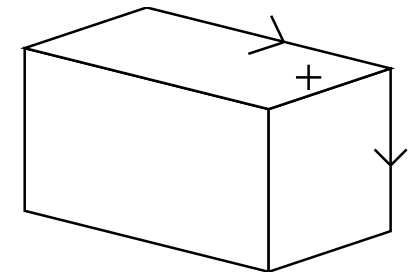
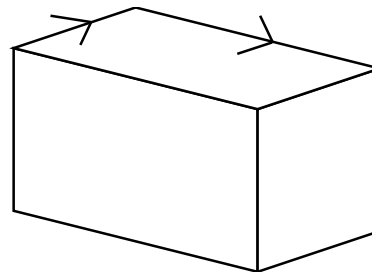
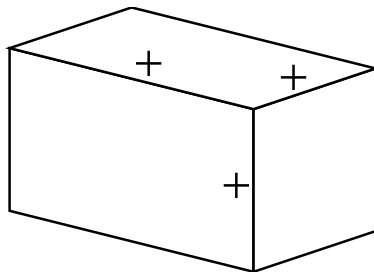
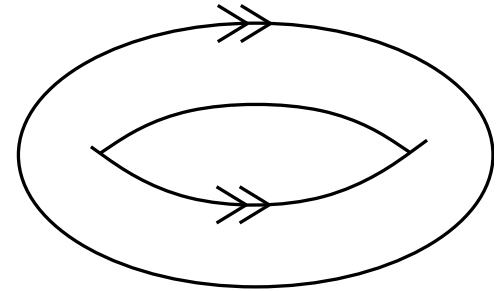
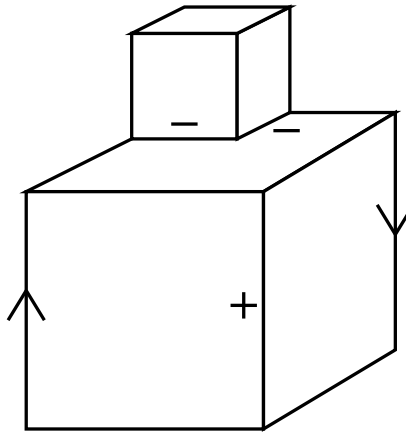
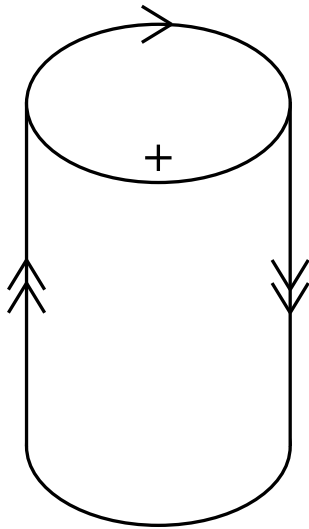
Assume actual texture is uniform

Compute surface shape that would produce this distortion

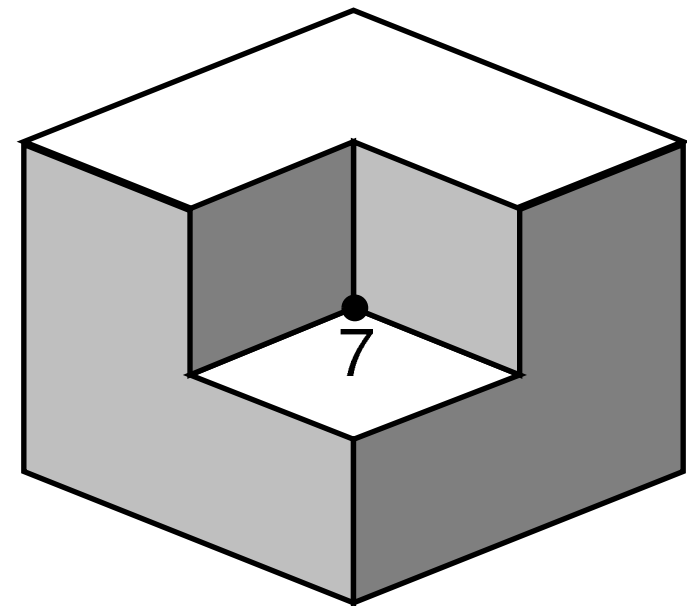
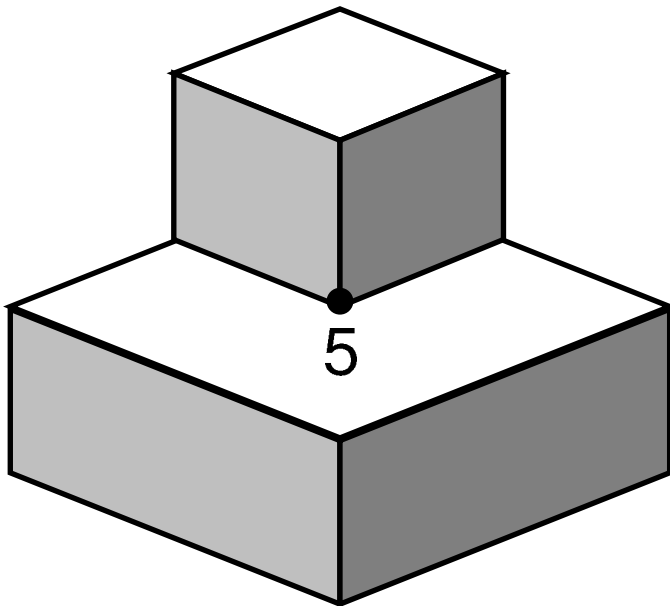
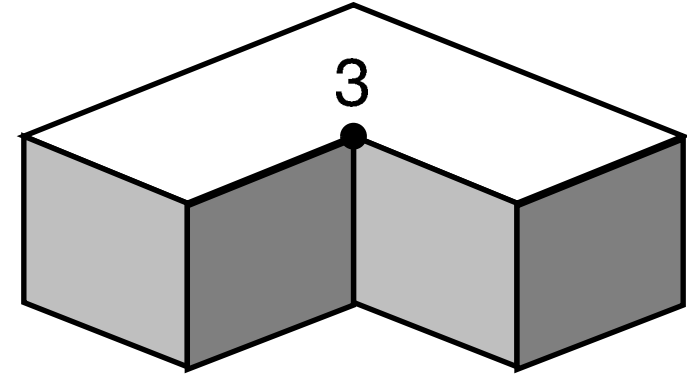
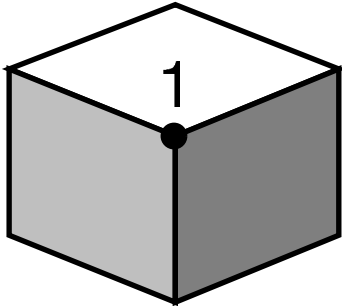
(Similar idea works for shading)

Edge Types

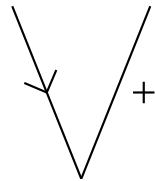
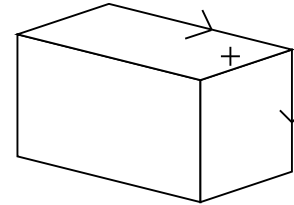
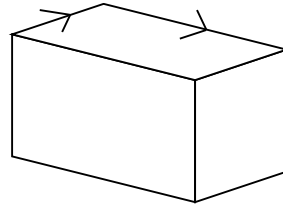
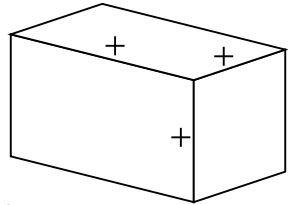
Assume world of solid polyhedral objects with trihedral vertices



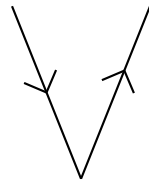
Vertex Types



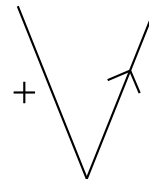
Vertex/Edge Labels



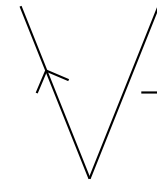
3



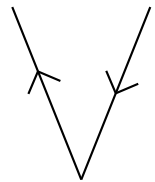
3



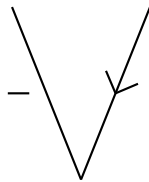
3



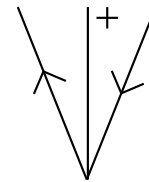
5



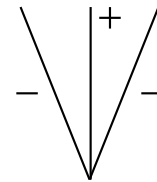
1



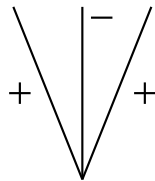
5



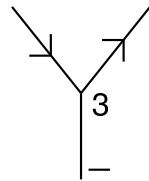
1



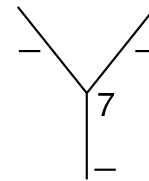
5



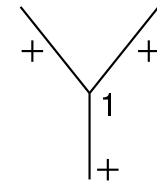
3



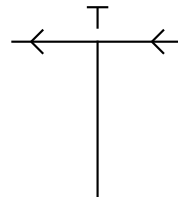
3



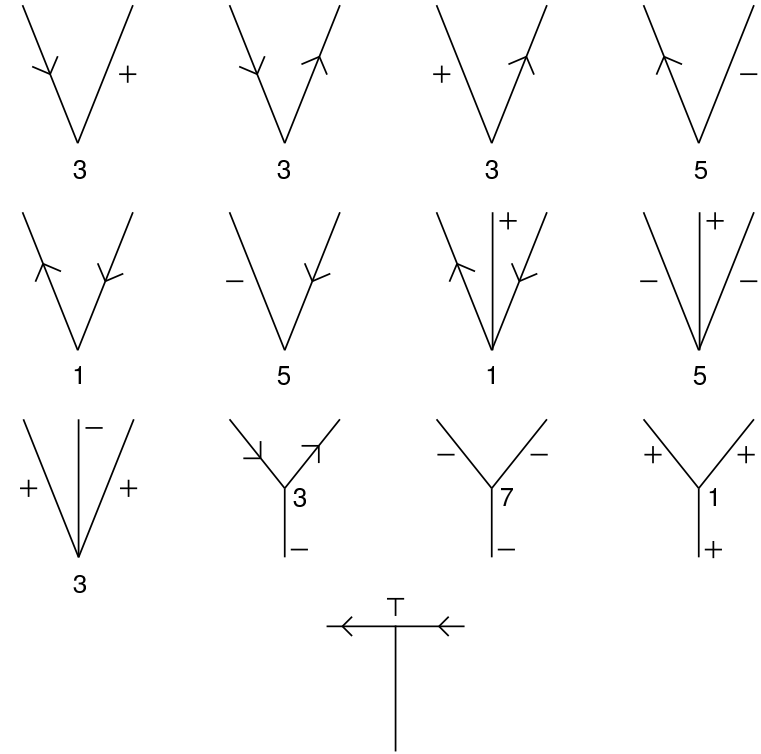
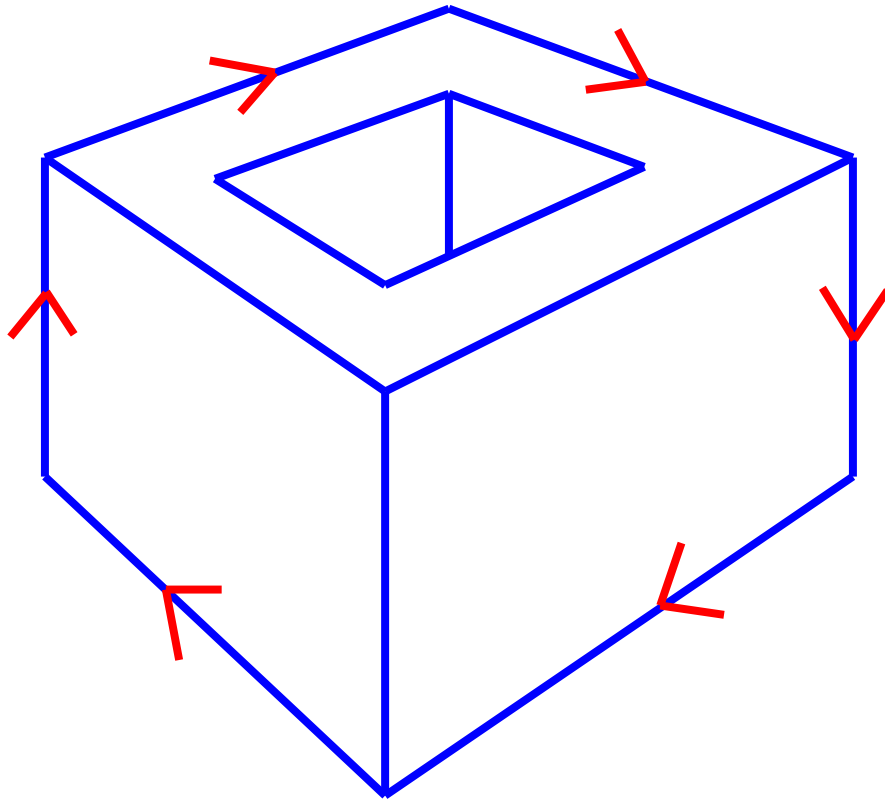
7



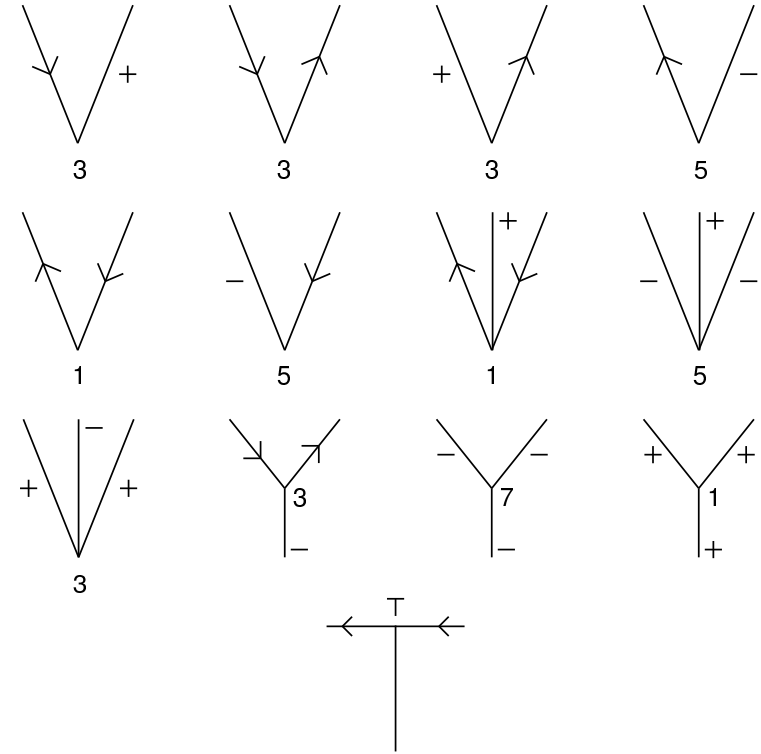
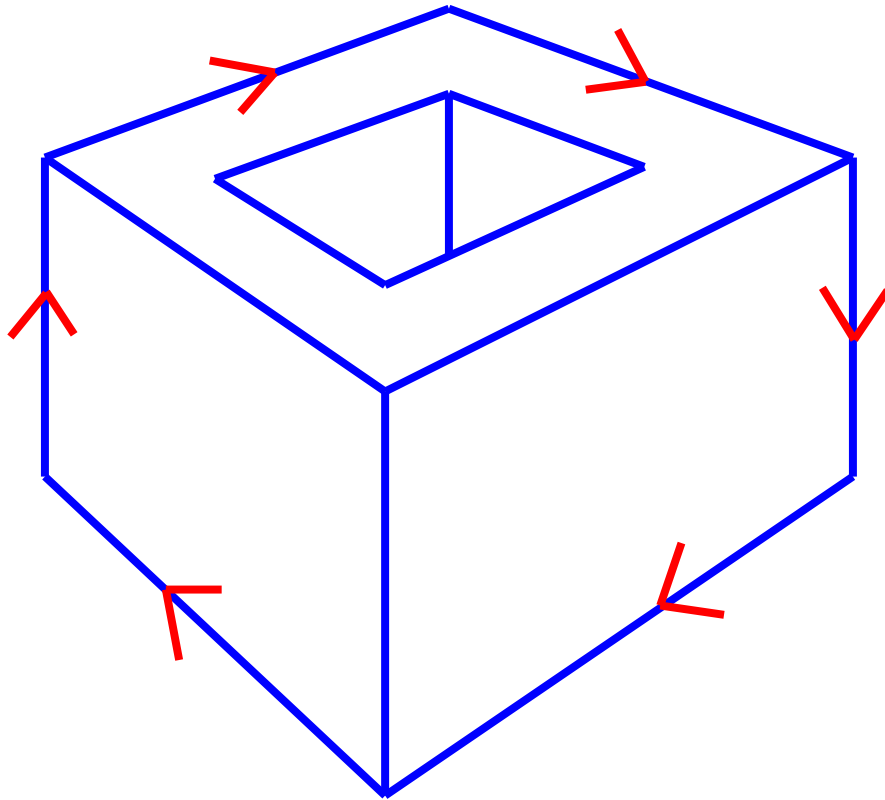
1



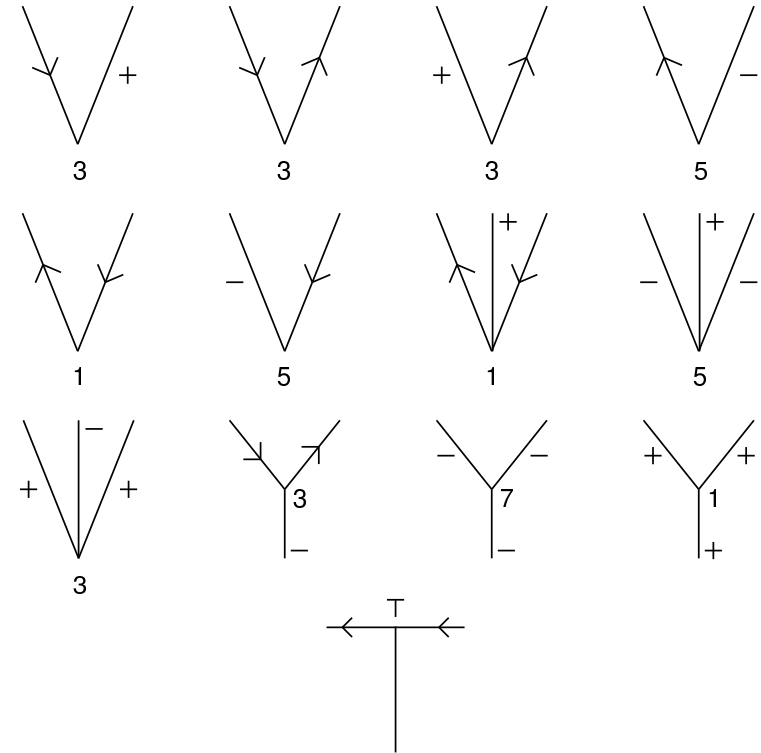
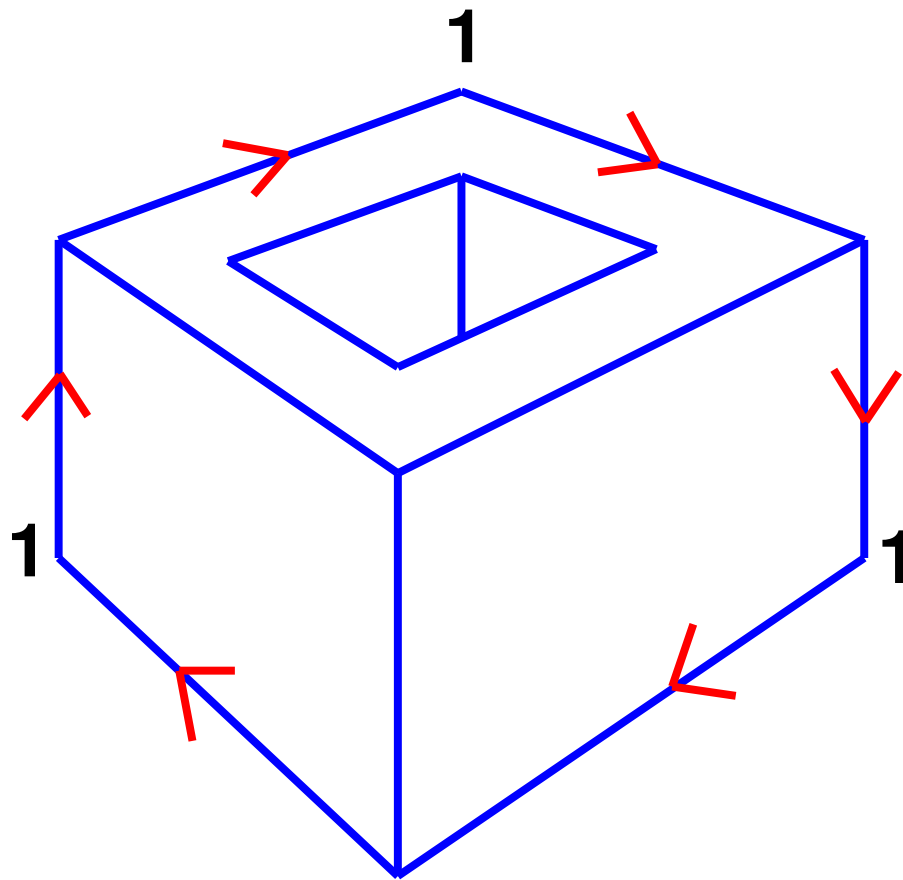
Vertex/Edge Labelling Example



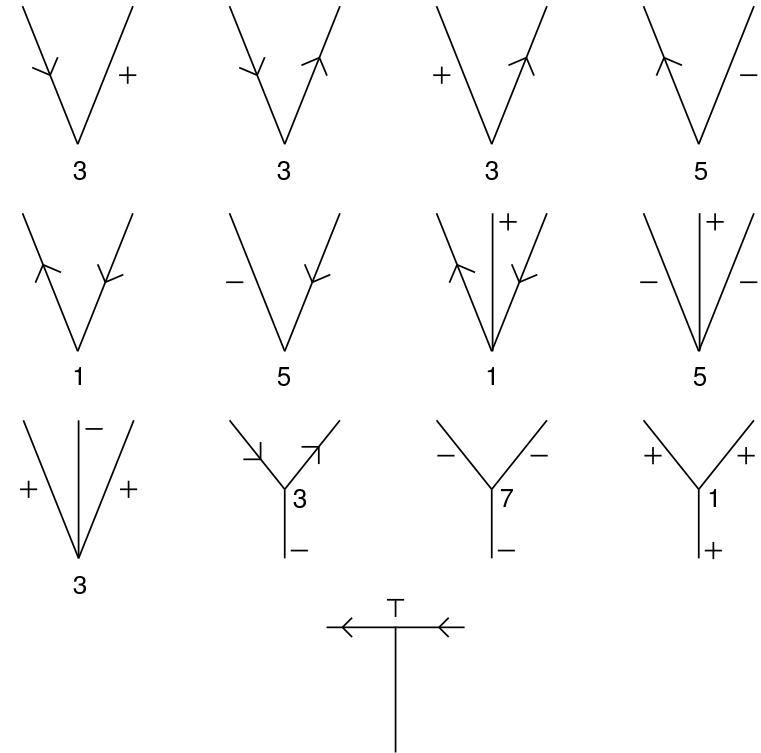
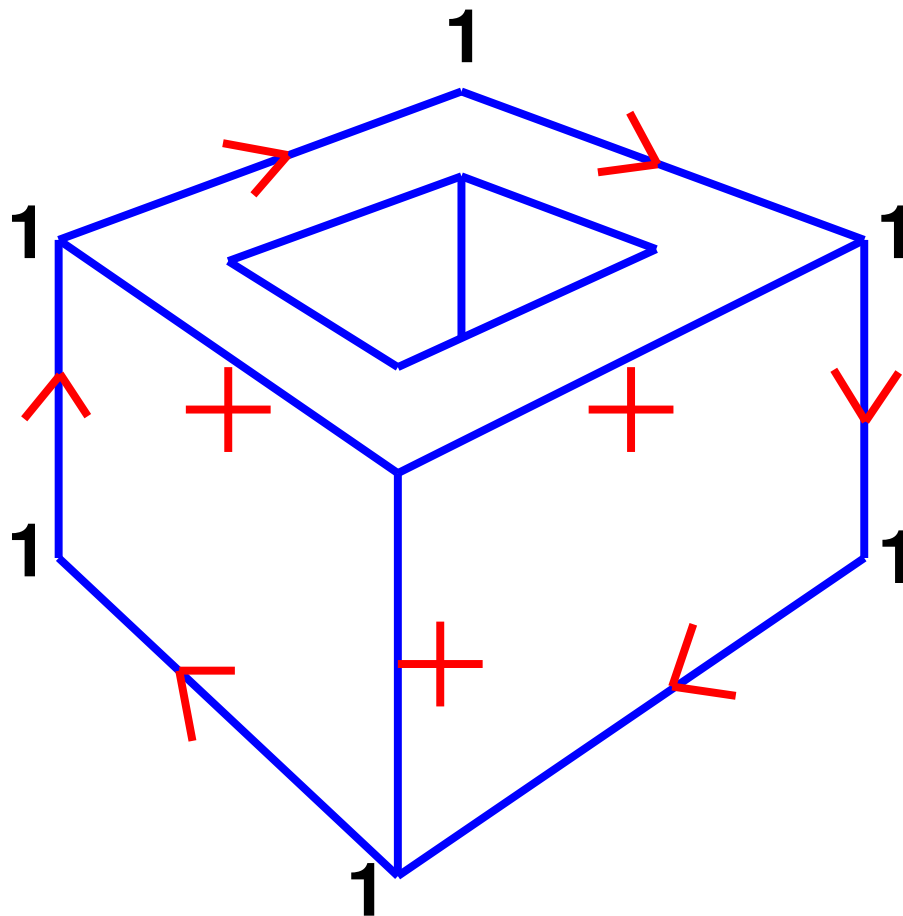
Vertex/Edge Labelling Example



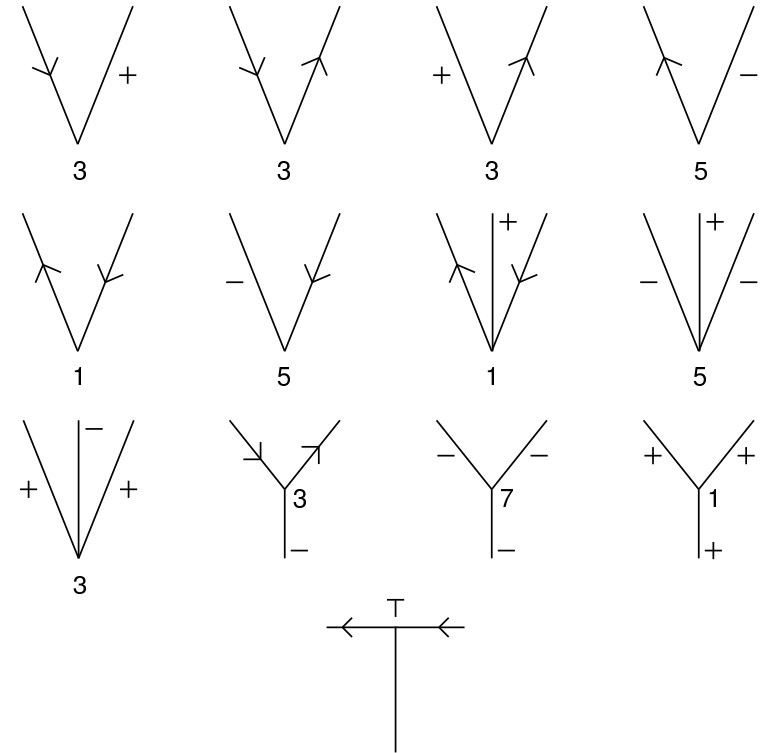
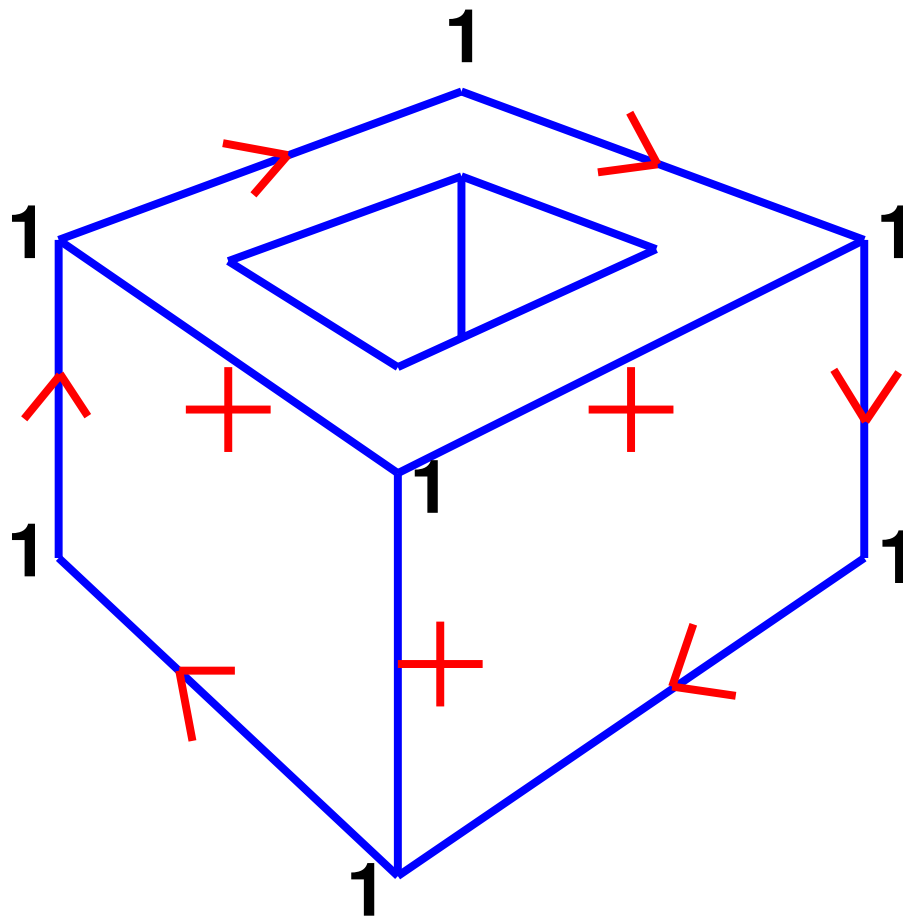
Vertex/Edge Labelling Example



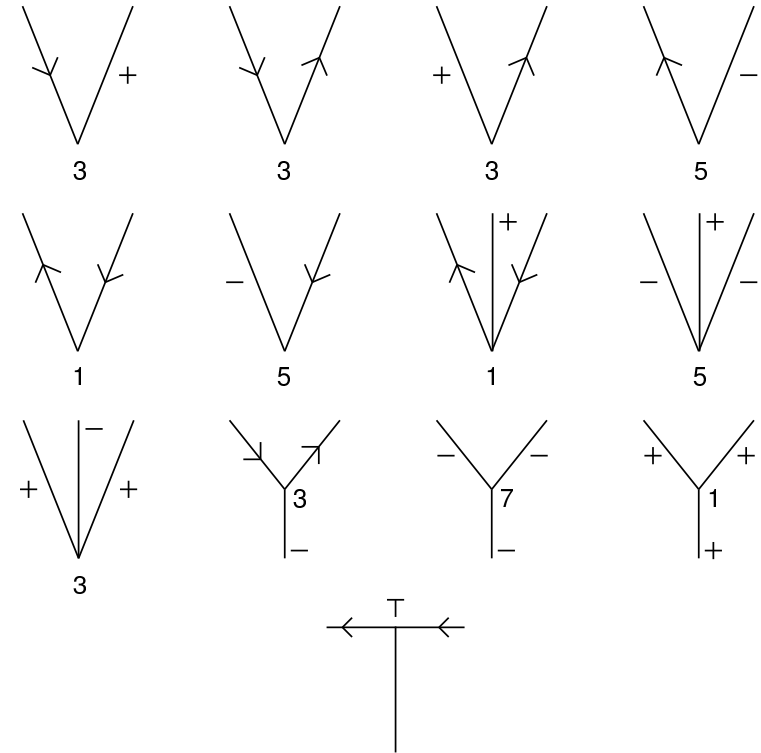
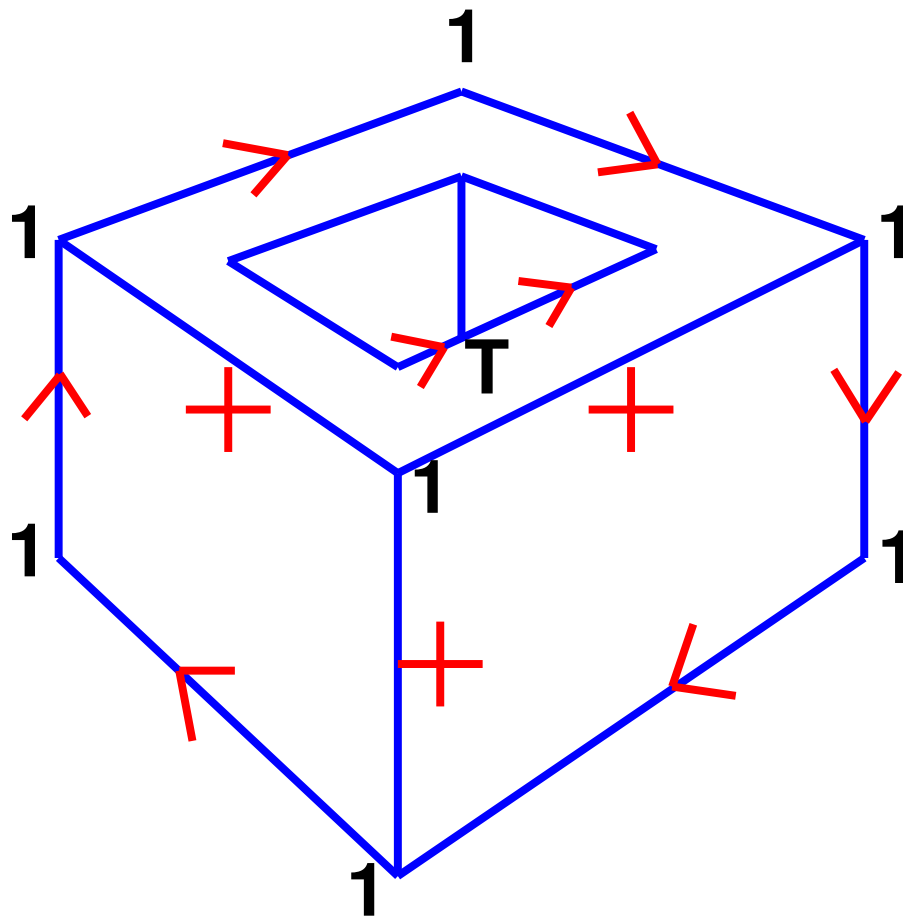
Vertex/Edge Labelling Example



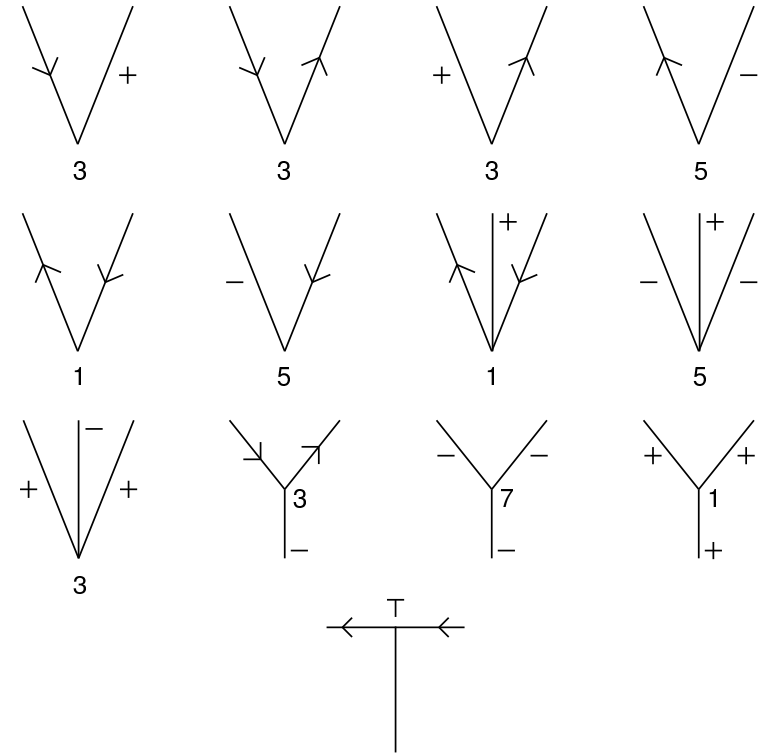
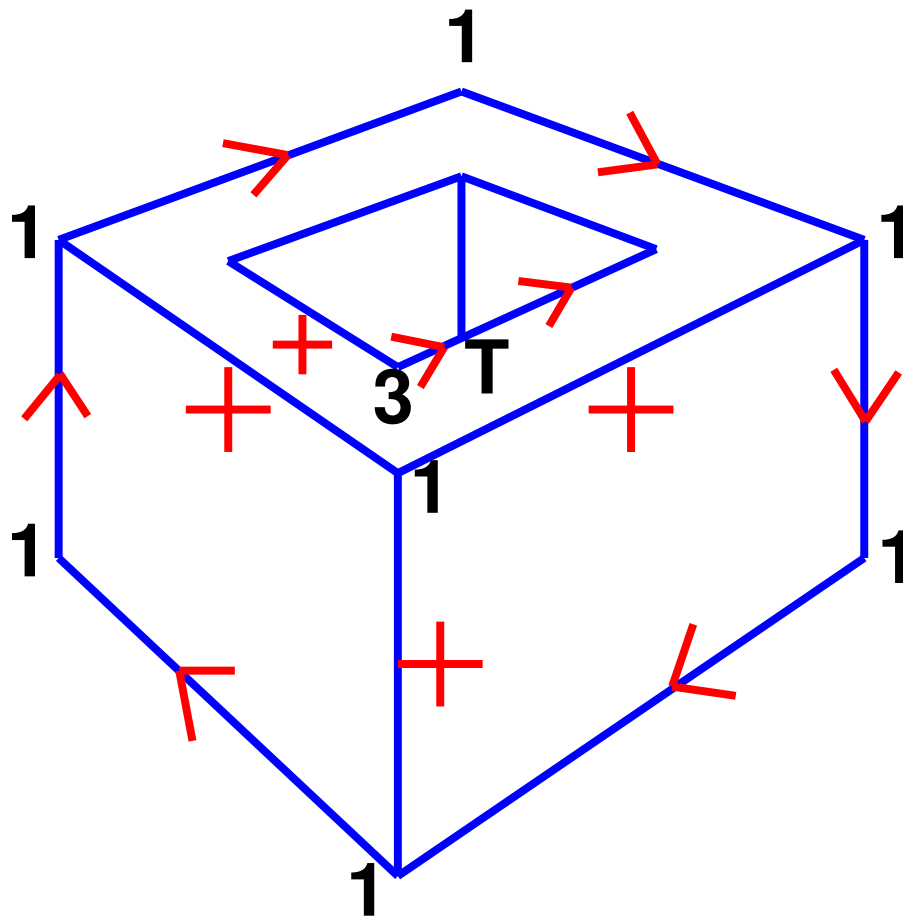
Vertex/Edge Labelling Example



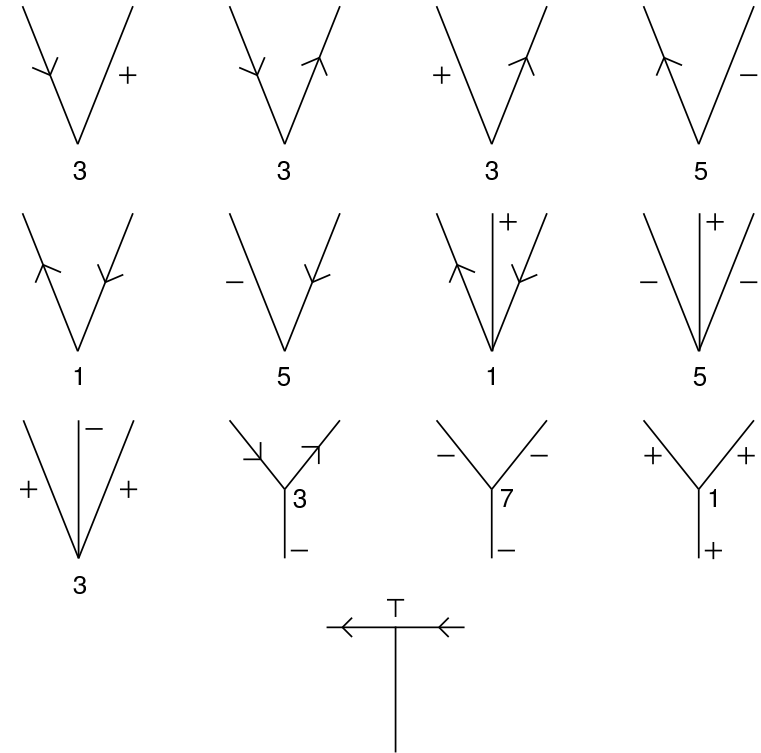
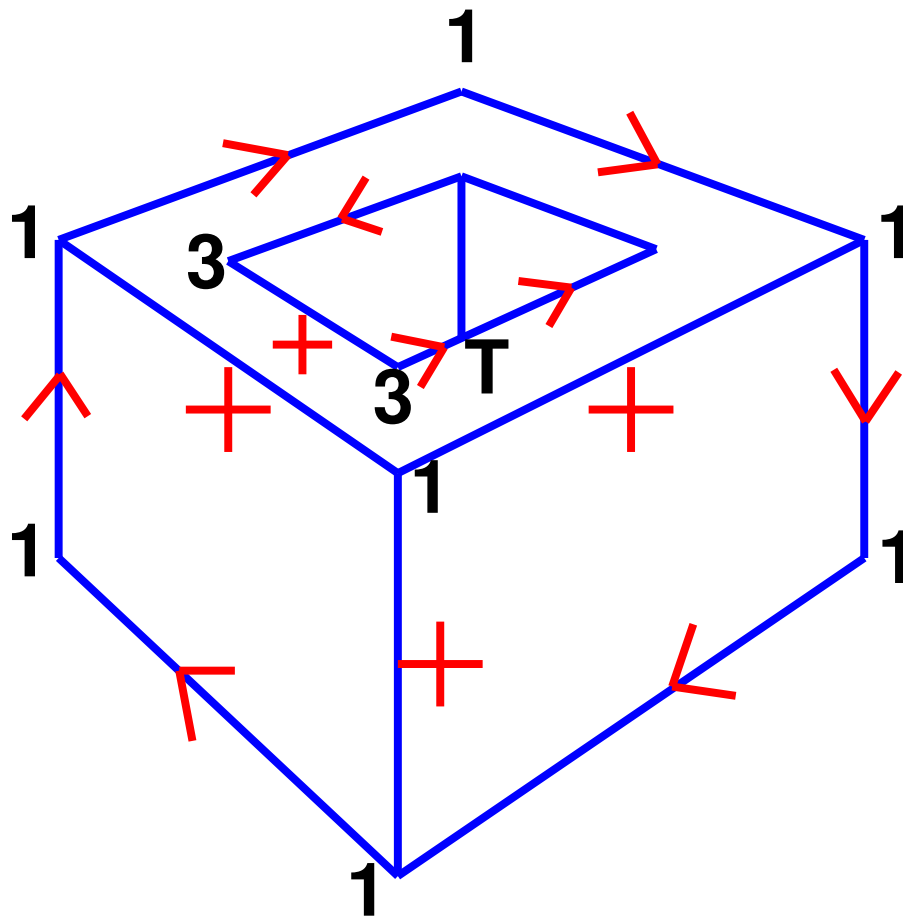
Vertex/Edge Labelling Example



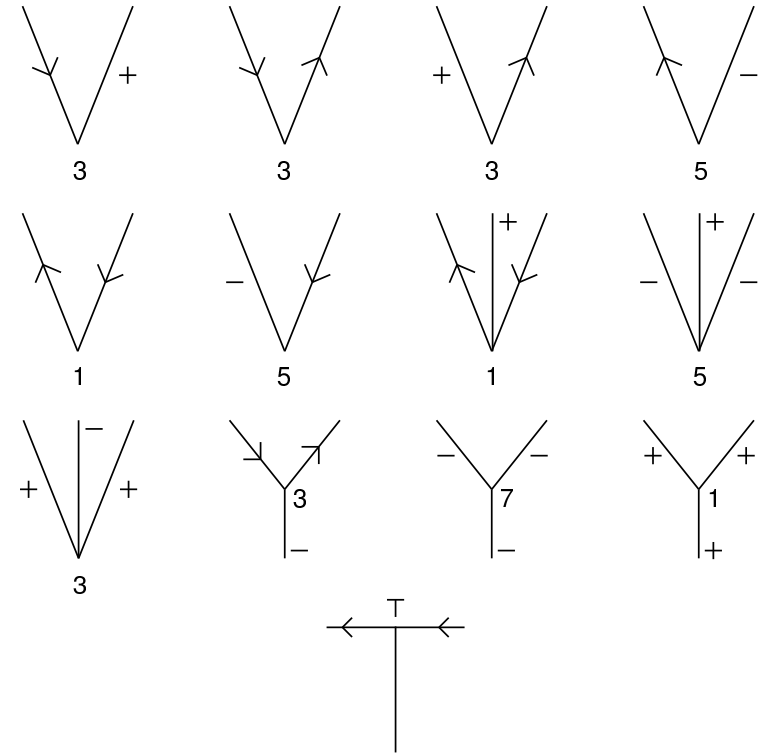
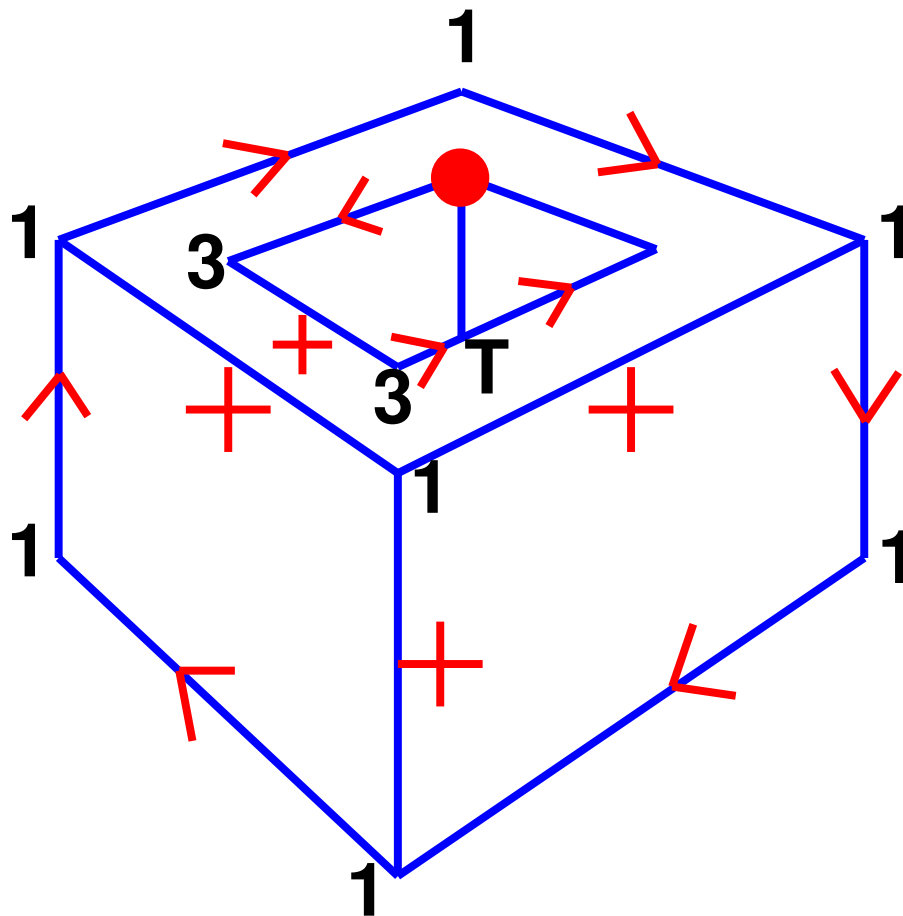
Vertex/Edge Labelling Example



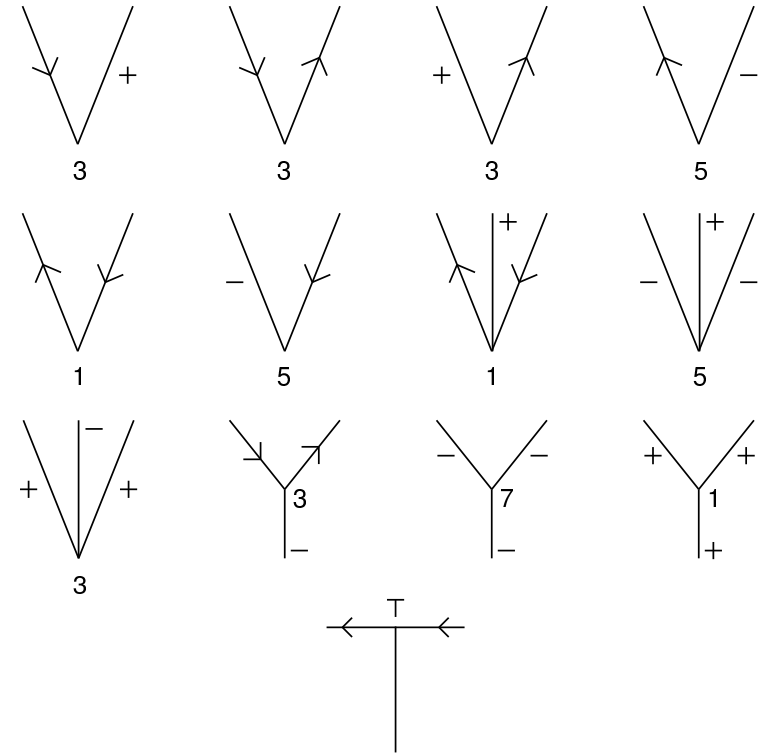
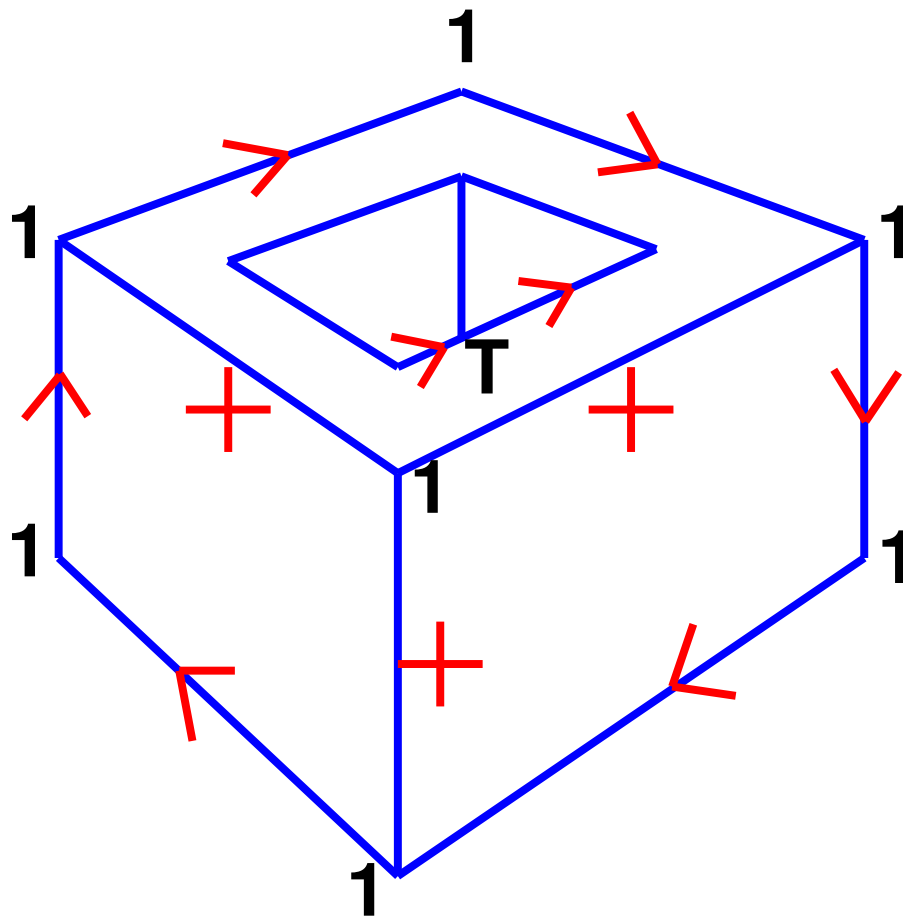
Vertex/Edge Labelling Example



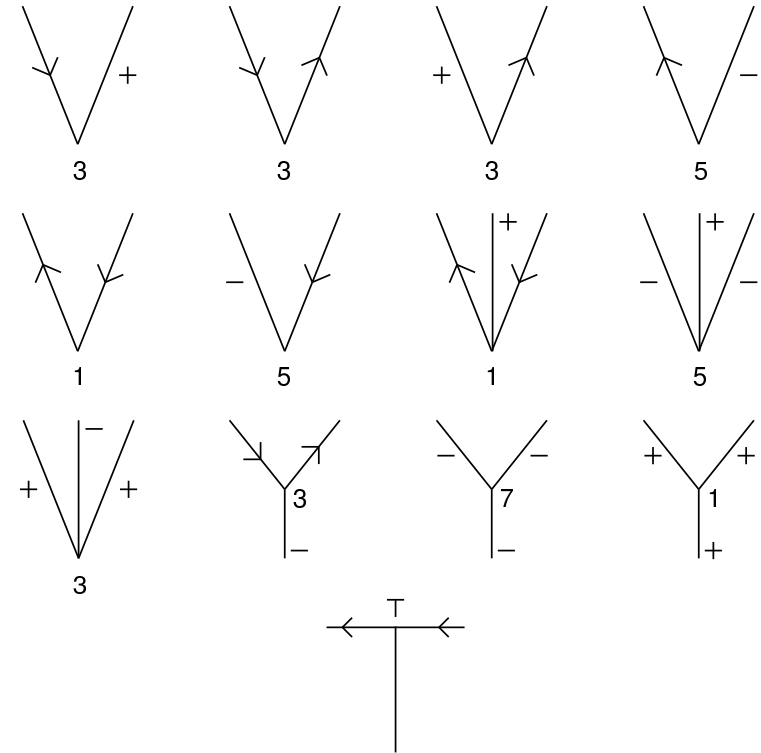
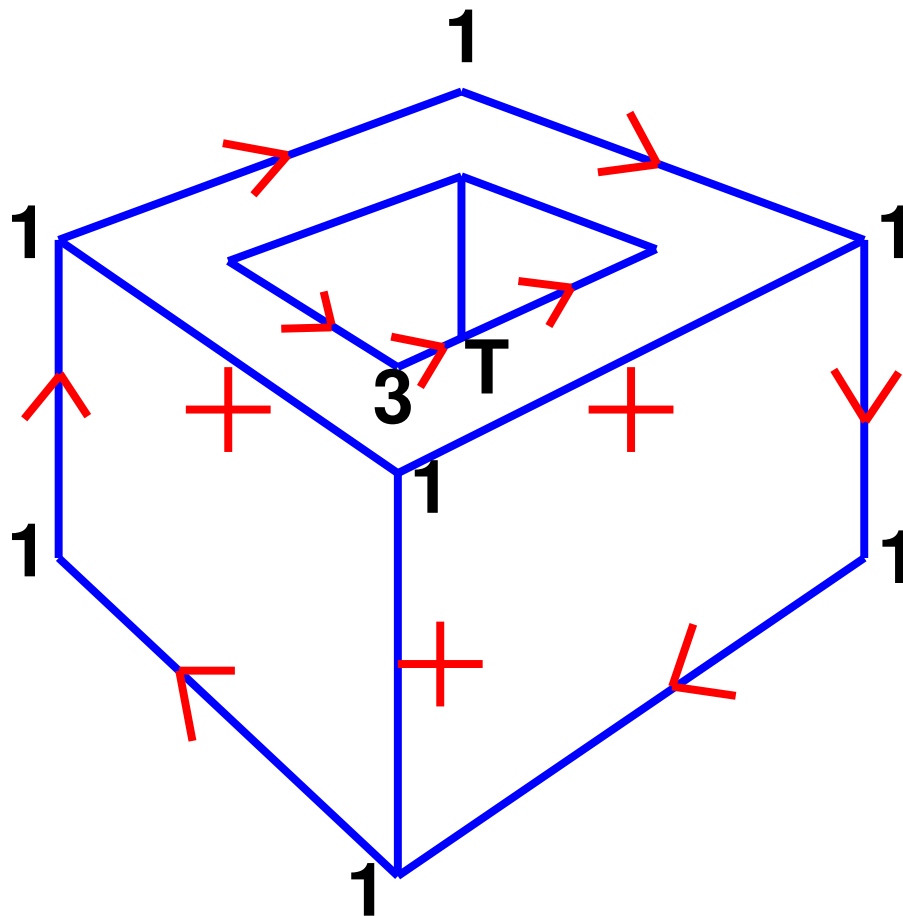
Vertex/Edge Labelling Example



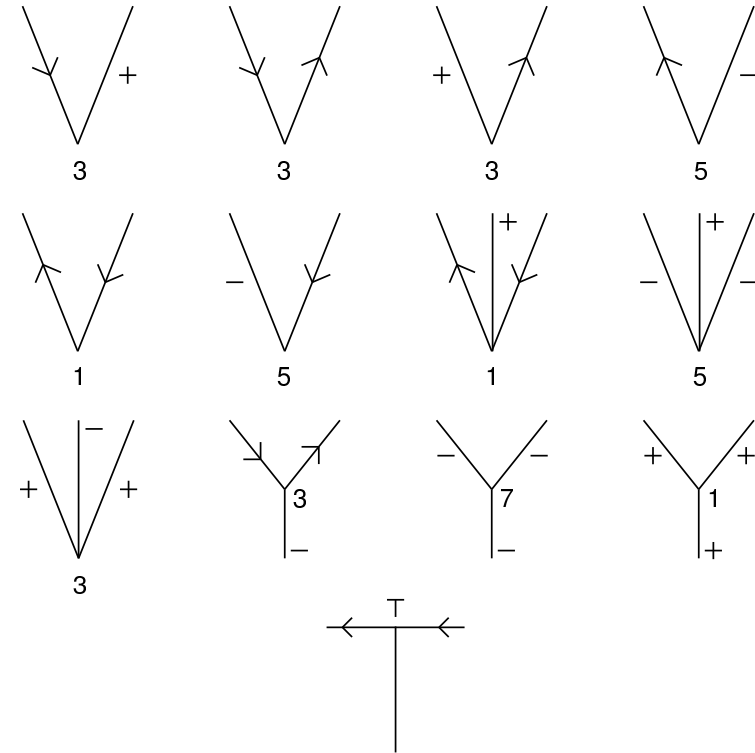
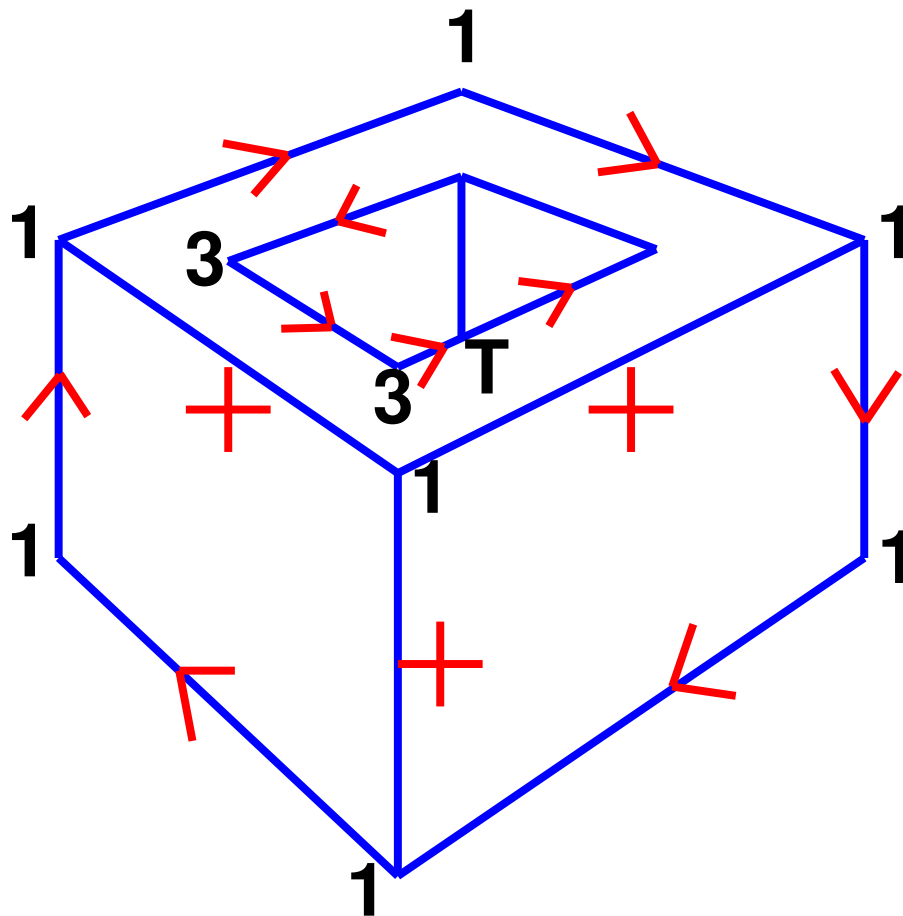
Vertex/Edge Labelling Example



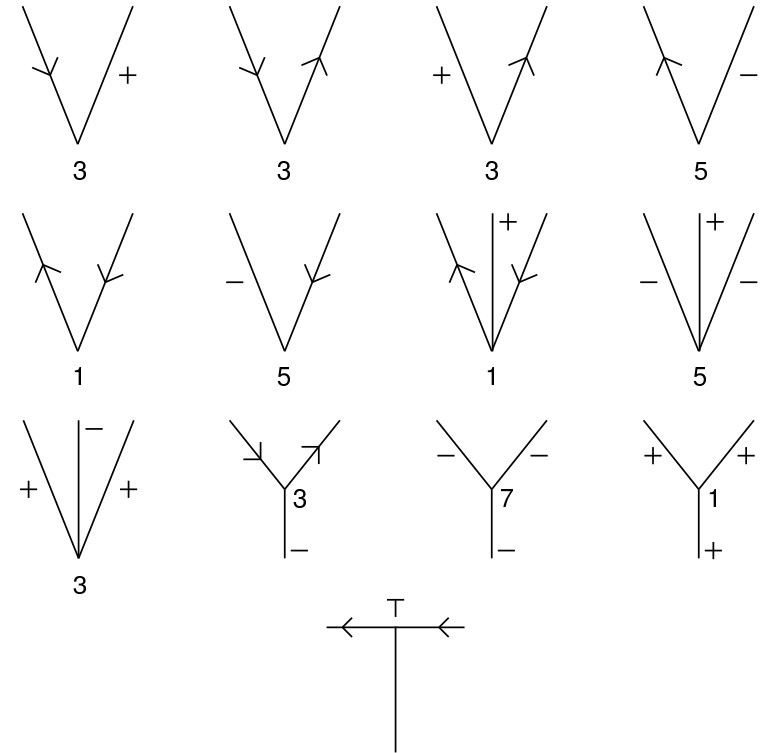
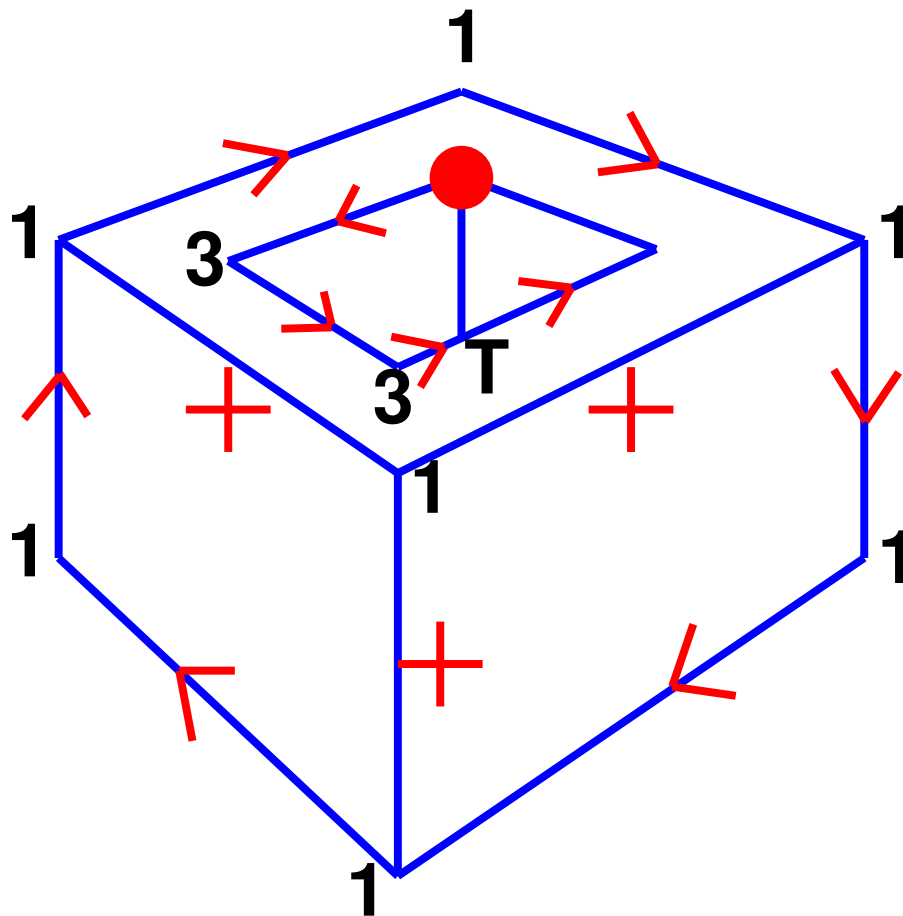
Vertex/Edge Labelling Example



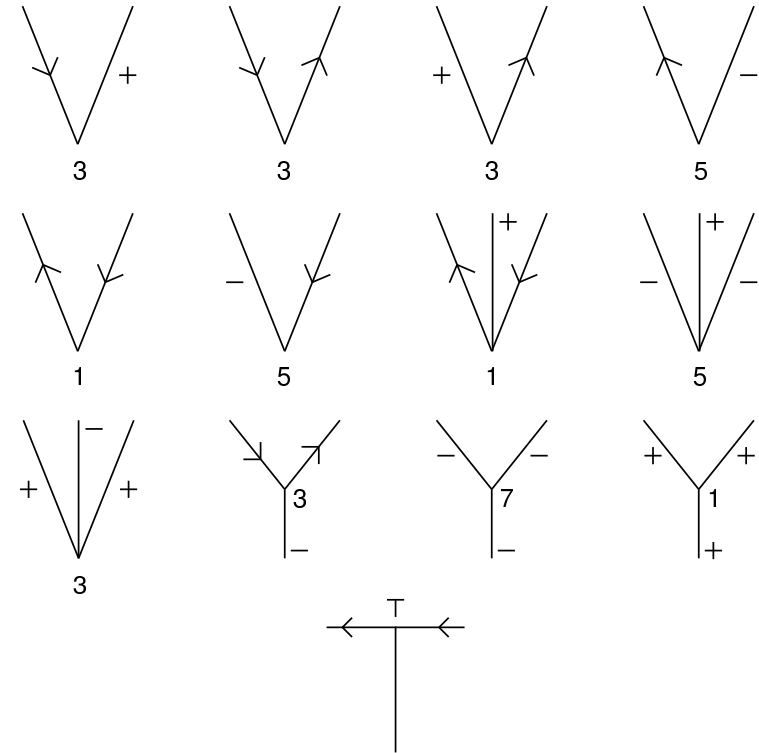
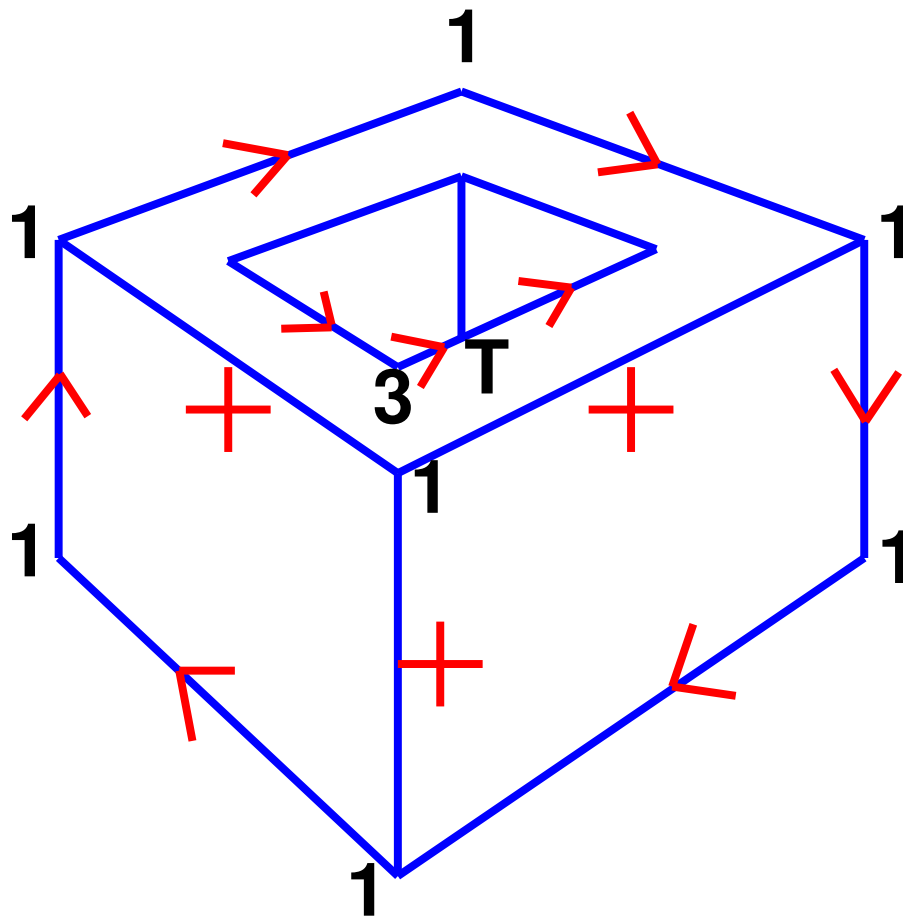
Vertex/Edge Labelling Example



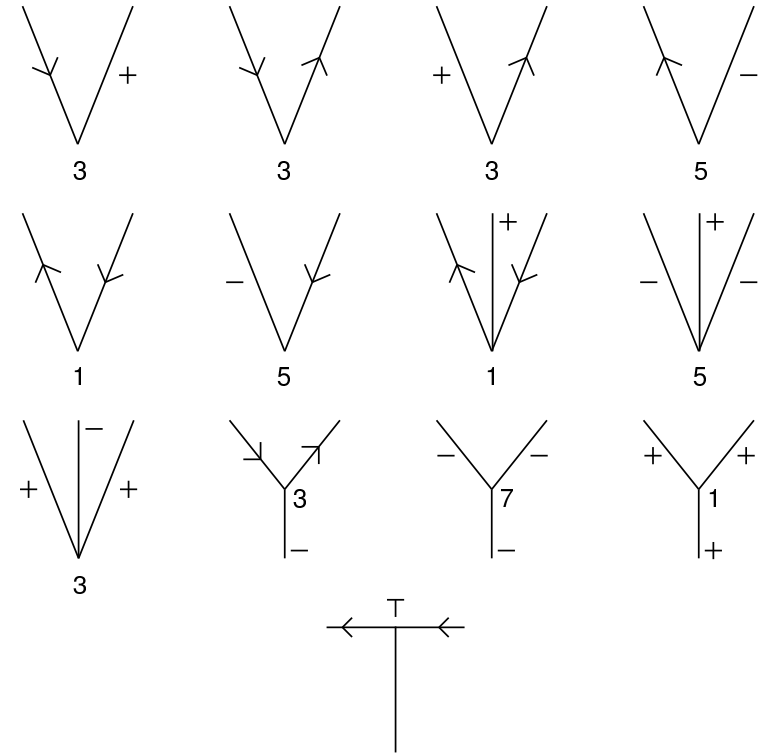
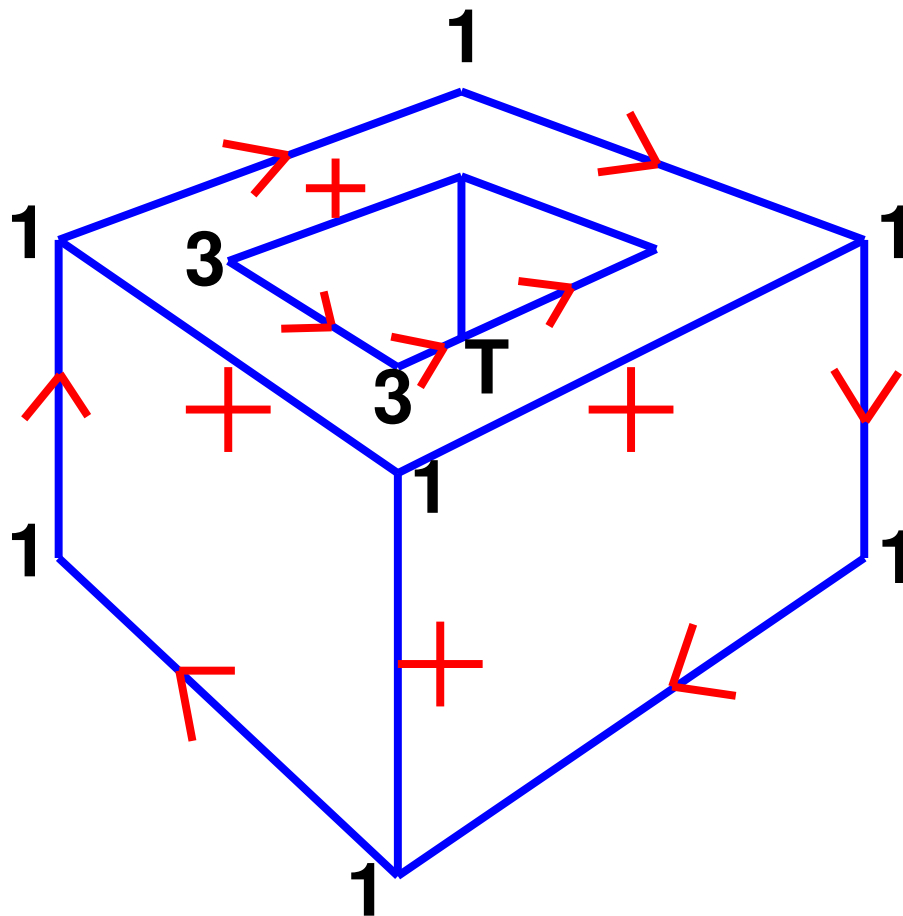
Vertex/Edge Labelling Example



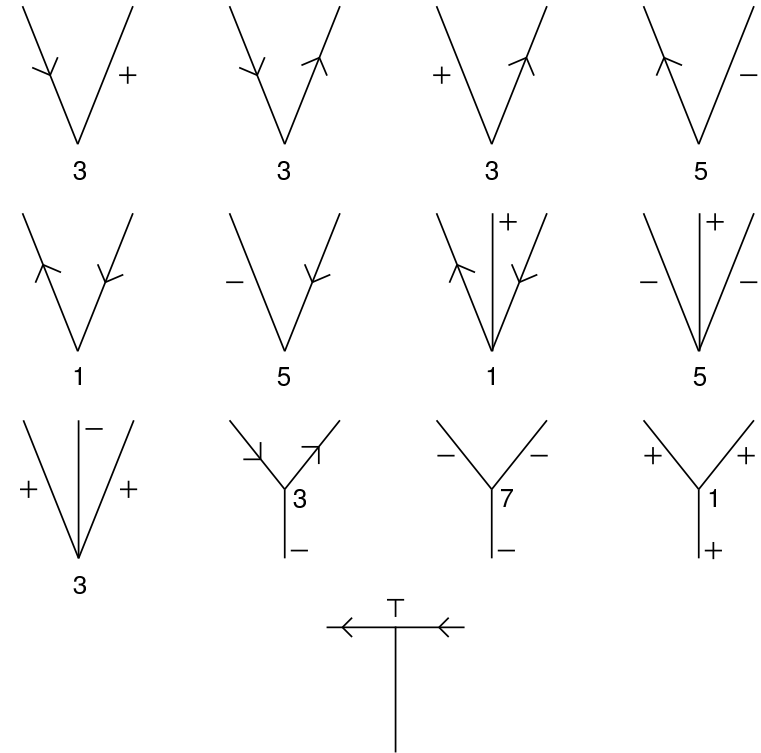
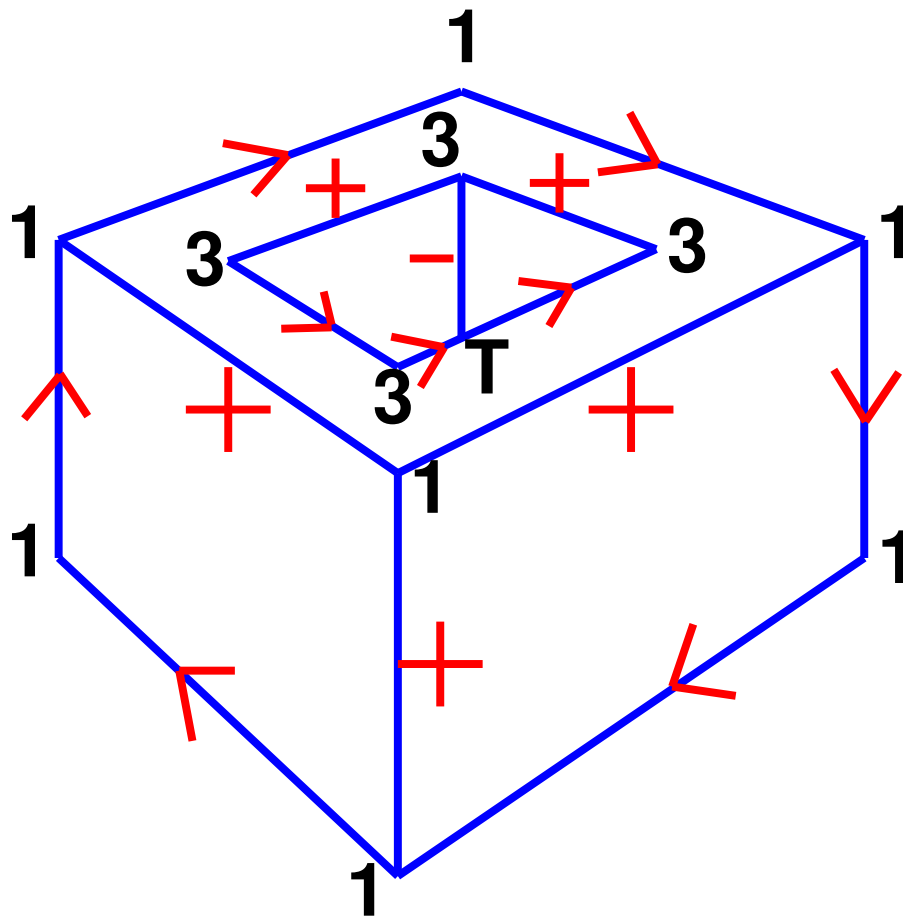
Vertex/Edge Labelling Example



Vertex/Edge Labelling Example



Vertex/Edge Labelling Example



Object Recognition

Simple idea

- Extract 3-D shapes from image
- Match against “shape library”

Object Recognition

Simple idea

- Extract 3-D shapes from image
- Match against “shape library”

Problems

- Curved surfaces
- Improper segmentation, occlusion
- Unknown illumination, shadows, markings, noise, complexity, etc.
- Representing shape of extracted object
- Representing shape and variability of library object classes

Object Recognition

Simple idea

- Extract 3-D shapes from image
- Match against “shape library”

Problems

- Curved surfaces
- Improper segmentation, occlusion
- Unknown illumination, shadows, markings, noise, complexity, etc.
- Representing shape of extracted object
- Representing shape and variability of library object classes

Approaches

- Index into library by measuring invariant properties of objects
- Match image against multiple stored views (*aspects*) of library object

Summary

- **Vision is hard: noise, ambiguity, complexity**
- **Prior knowledge is essential to constrain the problem**
- **Need to combine multiple cues: motion, contour, shading, texture, stereo**
- **Image/object matching: features, lines, regions, etc.**