Why is there a need to set a problem?

Most problems are initially expressed using natural language. Engineers need to **translate** them so that they can use engineering tools (computers, circuits, etc...) to "solve" them.

Problems often suffer from

- Lack of clarity
- Multiplicity (or indefinitess) of goals
- Complexity (too many degrees of freedom, interrelations)
- Dynamics (impredictability with time)

Researchers usually believe that **their** problem is well-posed... but this is not so

Examples

An ill-posed problem: image segmentation

Given an image, segment it into objects.

Why is it ill-posed? Because

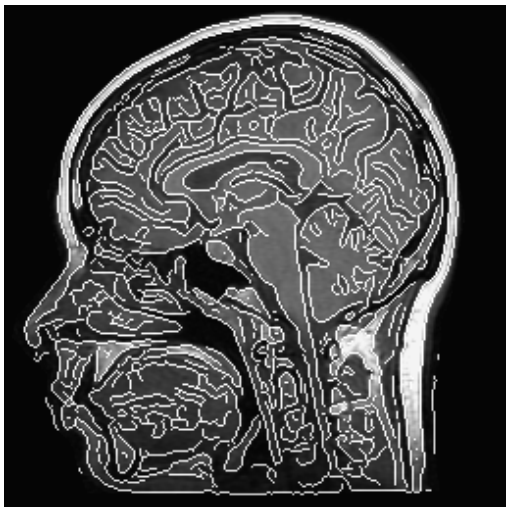
- There is no unambiguous definition of what *is* an object;
- There is no unambiguous definition of what *is not* an object;
- There is no objective measure of the segmentation quality;

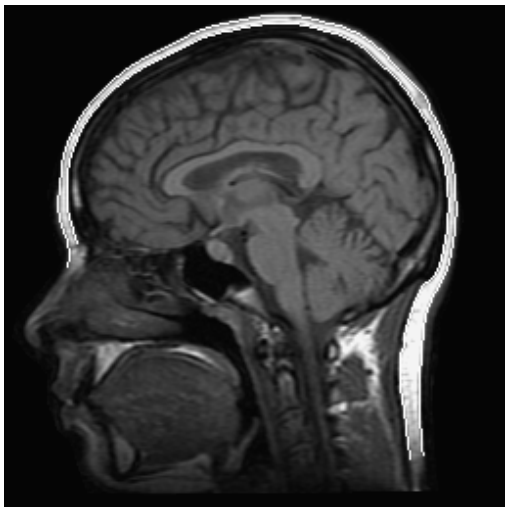
Consequence: segmentation is more an *art* than a *science*, and always requires double checking with a human eye.

In practice: segmentation algorithms concentrate on one (non-exclusive) characteristic of objects (edges, relative uniformity) to provide segmentation results.

Examples

Edge detection making *assumptions* on what may characterize an edge (large gradient, continuity)





Edge maps obtained with various thresholds

Examples

Another ill-posed problem: signal interpolation

Given samples of a signal, find samples at a higher resolution.

Why is it ill-posed? Because

- There is an infinity of ways to generate these missing samples;
- The term "sample" may be ambiguous (multiple definitions);
- No objective measure of the interpolation quality;

In practice: interpolation algorithms constrain the signal into a small signal subclass and the problem becomes well-posed.

Problem setting and solving

In his famous book *How to solve it* for teaching mathematics, G. Pólya (1887–1985) suggests four steps each of which consists in *asking* questions.
The four steps:

1. Understanding the problem.
2. Devising a plan
3. Carrying out the plan.
4. Looking (looping?) back.

Problem setting and solving

Understanding the problem

- What is the problem?
  Is it possible to express it with a very short sentence?
- What are the unknowns?
  Are they quantified? If not, how to quantify them?
- What are the (known) data?
  Are they quantified? If not, how to quantify them?
- What are the conditions on the unknowns?

The goal is to get better acquainted with the problem, and *feel* how well it is posed.

Tips: Draw a *figure*; introduce adequate *notations*.

Problem setting and solving

Devising a plan

- Is there an analogy with another problem?
- Can the problem be split into smaller, simpler problems?
- Does the problem look simpler by generalizing it?
- Is there a particular case where this problem can be solved?
- Test the hypothesis and its influence on the well-posedness?

The goal is to learn know how to approach the problem.

A temporary failure is *normal*, even beneficial.

Problem setting and solving

Carrying out the plan: checking each step

- Does each step look obviously correct?
- Is each step provably correct?
- Have all the data been used?
- Have all the hypotheses been used? How many times?

Check for circular reasoning!

Problem setting and solving

Looking back: examine the solution

- Can the solution be checked?
- Would it be possible to solve the problem differently?
- Could the methodology used be applied to solve another problem?
- Is there a contradiction?

The goal is to learn the characteristics of the solution to the problem, and possibly to use this new knowledge to improve over the solution.

A solution is *never* definitive.