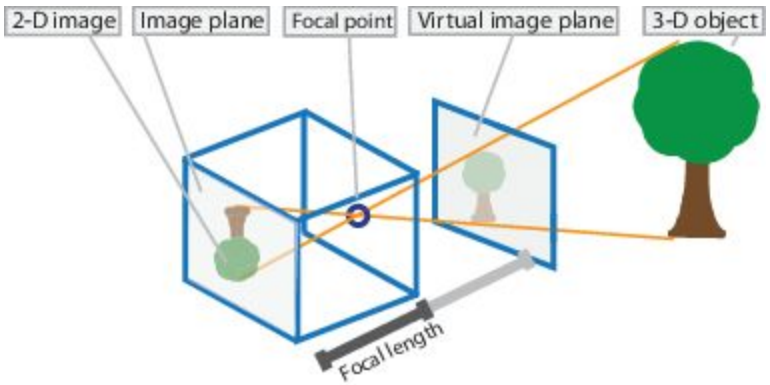
Cognitive Problems in Computational Terms

One of the great advances that has been made in understanding the mind because of the comparisons between computational methods and the problems that human beings have to face: **what makes a problem hard or difficult**

Examples:

1. **Optical Vision**

   When you open your eyes, you seem to automatically know where all the objects are in the world. You look around and you see an object there and an object over there and you know where you are, it seems a kind of automatic thing. It doesn't seem like it ought to be that hard to do at least as far as our intuition is concerned. It turns out that in some sense seeing is an impossible problem. We do it and yet in a mathematical sense the most formal statement of the problem is impossible.



   The structure of the eye is a pinhole camera where light comes in through the pupil and hits the retina, and from that information alone, that is from the information of patterns of light hitting the retina, we are going to determine where objects are in the world. It is mathematically impossible to phrase all this. The reason being that you cannot recover three-dimensions from two-dimensions or to put it another way, an infinite number of patterns of three-dimensional objects or an infinite number of three-dimensional structures could give rise to the same pattern of light on the retina.

2. Human Language

   Learning language seems quite easy for us which is actually difficult to implement. There are a number of different theories and hypotheses and explanations. One thing that is

exceptionally interesting is not just problems like learning how to speak or learning how to see, but all of the **common sense knowledge** that we seem to have about the world in the course of our conversations.

These are four consecutive sentences taken from apparently a real children's book. We can think about what it would take to program a machine to understand these sentences. What kind of knowledge it would take.

Sentence 1: “Jane was invited to Jack's birthday party.”
Sentence 2: "She wondered if he would like a kite."
Sentence 3: “She went to her roo and shook her piggy bank”
Sentence 4: “It made no sound”

So in effect, by trying to mimic minds with machines, we have learned a great deal about the nuance of what makes a problem easier or difficult.

Ill-defined Problems:

We have no idea really or very little idea about how even to begin to program an answer to these problems. They feel like very profound problems.

Examples:

1. Write a program that will pass the Turing test. It's hard to know where to start. We might have some ideas, but it's awfully hard to know where to start.

2. Plato's dialogue The Meno, starts with somebody, a young man, going up to Plato's mentor Socrates in the streets of Athens and saying, "Socrates, can you teach virtue? Is virtue teachable?" From that they immediately get to the question of what is virtue. If we know what virtue is, maybe we know whether it can be taught. Well, in some sense, that's a very important problem. Every parent wants to be able to teach their child, if possible, to be a good and virtuous person. To some it's an unimportant problem but it's hard to know where to start to begin to define virtue and to think about what it would mean to teach it.

Impossible Problems:

It can't be done. In human affairs, this is actually a tremendous knowledge, usually. When you know that something, a well-defined problem is impossible to solve, you know a great deal and often these impossibility proofs represent major landmarks of human thinking.

Examples:

1. The angle try-section problem using a straightedge and compass, and given an angle theta, construct an angle of magnitudes theta over three. Take an arbitrary angle and create an angle one-third that size, an arbitrary angle. Now, it turns out that given a straightedge and

compass, this can't be done. But it was an extremely provocative problem to the ancient geometers and they did not know that it was impossible. In fact, the proof that this problem is impossible wasn't established until the **19th century**.

2. To write a computer program which given any other program P and input N will determine whether the program P halts on N. What we want is a program that will take as its input on another computer program P and an input value N and will run on its own and then come back and tell us whether the program P ever halts the input N. It turns out that that is impossible. That's called the **halting problem**, and **Alan Turing** showed that that was impossible in a marvelous and much discussed and anthologized paper written in **1936-1937**.

Knowing that it's impossible is again a huge achievement in human thought. The perpetual motion problem in Physics and so forth, knowing that something is impossible is a very big deal.