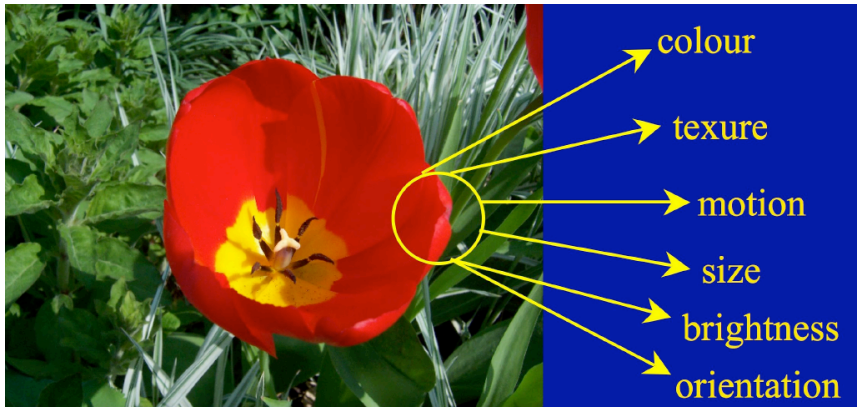
Why Vision is Hard (for the brain)

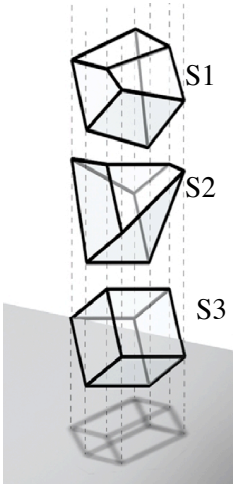


Structure

- Two reasons why is vision hard
- Reason 1: The problem of vision is ill-posed
- Reason 2: The problem of vision scales exponentially
- Parameter Subspaces
- The Binding Problem
- Reasons To Be Cheerful?
- Conclusion

Reason 1: The problem of vision is ill-posed

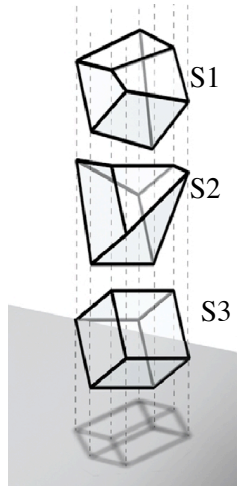
- Any image could be generated by an infinite number of objects (e.g. S1, S2 or S3).
- Vision is an ***ill-posed*** problem.

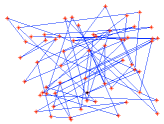


Reason 1 in a nutshell

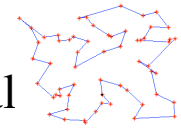
Shigeo Fukuda's Piano

Mirror



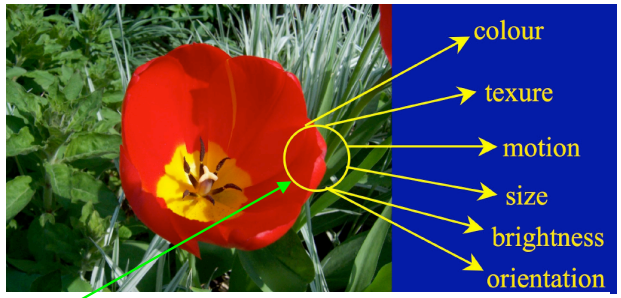


Reason 2: Vision is exponential



- All problems have a variety of solutions, some of which are better than others.
- The number (*M*) of *all possible* solutions (including really bad solutions) to a problem is defined by the *size* of the problem.
- The precise manner in which the number of possible solutions increases with problem size is described by ***complexity theory***.
- If the number of solutions increases ***exponentially*** as the problem size increases then the fastest computer could not find the ***optimal*** solution ***before the universe ended***.
- ***Vision is in this class***.

Reason 2



Consider a small image region that contains the edge of a flower petal.
Within that region, each property or ***parameter*** (e.g. colour) can adopt any one of *N=10* different ***parameter values*** (e.g. red, blue, green).
If there are *k=6* parameters (e.g. brightness, colour, size etc, as above) and each parameter can adopt one of *N* values then there are ***M=10^6***, (**one million**) possible ***combinations*** of parameter values (i.e. *M* possible images in that region).
If we used one neuron to recognise each possible image then we would need 1 million neurons *per image region*.

Reason 2

In the remainder of this lecture I will make two points

1) The number of possible images per image region increases *exponentially*.

2) If each neuron is responsible for detecting one possible image then we need an exponential number of neurons, but there aren't enough neurons to go around.

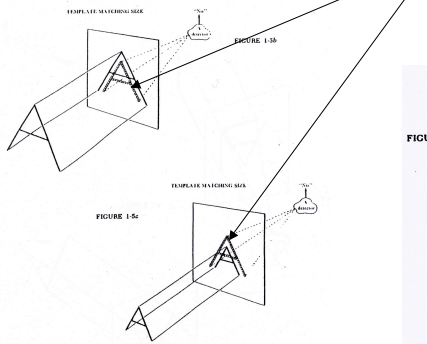
This is a ***counting argument*** that template matching is not a viable strategy for vision.

We can regard these *hypothetical* neurons as ***template-neurons…***

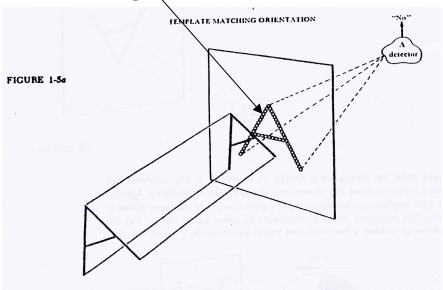
Parameters and Template Matching

Letter could be wrong size …

Template



or wrong orientation …

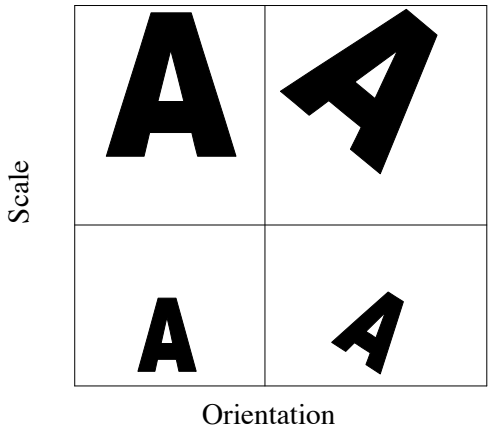


Vision Scales Exponentially

- Suppose the letter **A** could appear at one of two sizes and at one of two orientations …
- … and that we had **one neuron to detect A at each size and orientation**.
- How many neurons would be required?

NB: Diagrams of A detectors are schematic, no 'A' detectors have been found. Have used letter A for sake of concrete example. Could have used simple line instead of edge, with line varying in orientation and size because the fundamental problem of recognising lines and A's is the same.

4 Combinations (neurons)

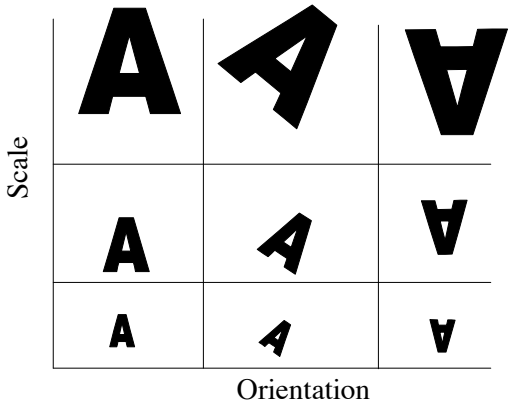


Two sizes and two orientations =>
2 x 2 = 2^2 = 4 possible views

('=>' means 'implies that')

N = 2 possible values for each of k = 2 parameters (scale and orientation)
$N^k = 2^2 = 4$.

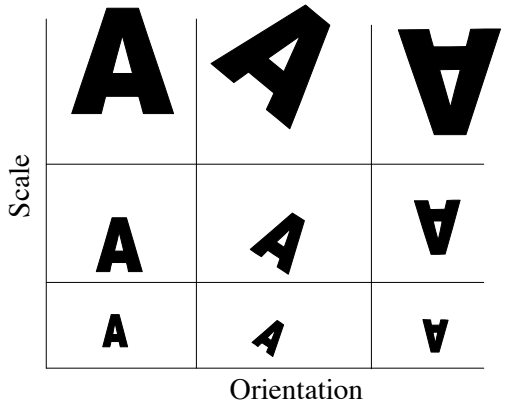
9 Combinations (neurons)



Three scales and three orientations =>
3 x 3 = 3^2 = 9 possible views

N = 3 possible values for each of k = 2 parameters (scale and orientation)
$N^k = 3^2 = 9$

Interlude: Parameter Spaces



A ***parameter space*** is the space of all possible parameter values.

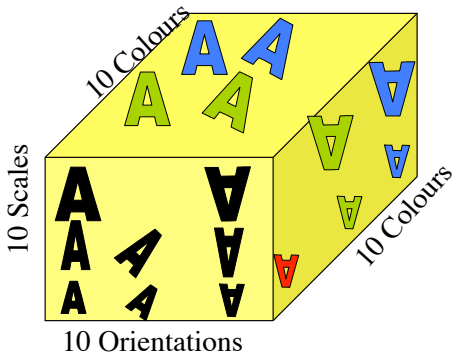
If there are 2 parameters, such as scale and orientation, then the parameter space is two dimensional (2D), as here.

A parameter space does not *necessarily* correspond to any brain region (but see V4 and V5 below).

100 Combinations

- Rather than $N=3$ possible values of each parameter, assume more plausible value of $N=10$.
- For $N=10$ possible values of size and orientation, this implies $M = N^2 = 10^2 = 10\text{x}10 = 100$ neurons.

3 Parameters: Scale, orientation and colour



If have $N=10$ orientations,
$N=10$ sizes, and
$N=10$ colours
then there are

$10\text{x}10\text{x}10=1000$

possible combinations of orientation, size, and colour.

Note that every time we *add* a new parameter (i.e. add one to k), the number of possible combinations is *multiplied* by a factor of N.

1000 Combinations

- If include **colour**, and assume letter occurs in one of 10 colours then require

$$M = 10\text{x}10\text{x}10 = 10^3 = 1000$$

neurons to represent the letter 'A' at any size, orientation or colour

(i.e. there are 1000 possible combinations of size, orientation and colour).

Number of template-neurons increases exponentially with number of parameters

If the number of parameters is k, and each parameter can adopt N distinct values and if each possible image in an image region is detected by one neuron-template then the number M of required neuron-templates is

$$M = N^k$$

k>3 parameters

What if now consider different fonts:

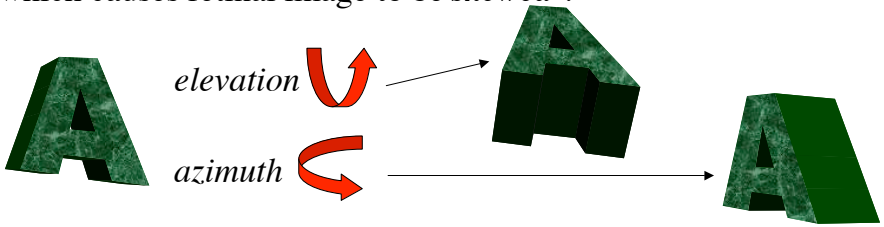
Each font can have one of 10 sizes, orientations and colours, so each font requires its own 3D block (see "3 parameters" slide), and each block required 1000 neurons.

So if have 10 fonts then would need

$N^k = 10^4 = 10\text{x}1000 = 10{,}000$ neurons per type of letter.

k>4 parameters

What if now consider letter viewed from different 3D angles which causes retinal image to be ***skewed*** ?



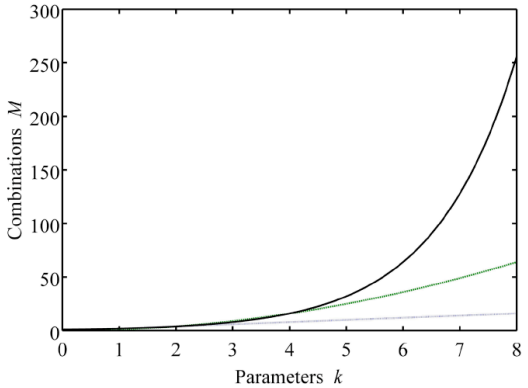
If have 10 views of *elevation (q)* and 10 views of *azimuth (p)* then require 100 neurons to represent all views, and $100\text{x}10{,}000 = 10^6$ neurons per type of letter.

k >> 4

- Above applies to single letter, for which need 10^6 neurons.
- If can recognise 1 million objects (e.g. faces) then would require $N^k = 10^6 \text{ x } 10^6 = 10^{12}$ neurons.
- Above analysis ignores other 6 parameters such as location (X,Y), speed, direction, stereo disparity, and depth, which (including above considerations) would require a total of $N^k = 10^6 \text{ x } 10^{12} = 10^{18}$ neurons.
- **There are about 20 billion ($20\text{x}10^9$) neurons in human neocortex.**

Not enough to solve the problem of vision using brute-force computing power.

Exponential Functions of *k*



Exponential $M = N^k$

Polynomial $M = k^N$

Linear $M = kN$

If the image of an object is defined in terms of k parameters, and each parameter can adopt N values then the number M of possible images increases as an ***exponential*** function N^k of k ($N=2$ in graph). Reducing N has little impact on requirements; it is k that dominates how the number of possible images increases.

Would a faster computer help?

Consider the problem of recognising an image of the letter 'A' described in terms of k parameters. If each of the $M = N^k$ possible 'A' images is detected by one *template*, and **if we could compare the image with 1 million templates per second** then the time required to search through all templates (for $N=10$, and $k=10$ to 40) is:

k	Linear (Nxk)	Polynomial (k^N)	Exponential (N^k)
10	1.1×10^{-10} days	10^{10}=2.8 hours	10^{10}=2.8 hours
20	2.3×10^{-10} days	20^{10}=118.5 days	$10^{20}=3 \times 10^6$ **(3 million years**
30	3.5×10^{-10} days	30^{10}=18.7 years	$10^{30}=3.2 \times 10^{16}$ years
40	4.6×10^{-10} days	40^{10}=332.6 years	$10^{40}=3.17 \times 10^{26}$ years

No, a faster computer would not help very much! (10^6 = 1,000,000)
Because every time we **add** a new parameter we **multiply** the processing time (number of combinations) by a factor of N(e.g. N=10).
(Note that $10^{-10} = 0.0000000001$)

Would a bigger brain help?

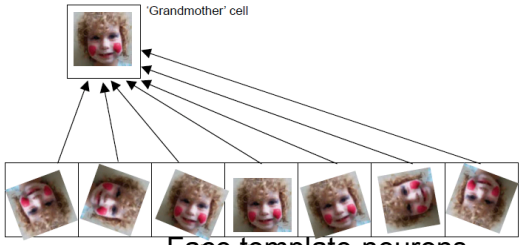
Again, consider the problem of recognising an image of the letter 'A' described in terms of k parameters. If each of the $M = N^k$ possible 'A' images is detected by one neuron then the number of neurons required (for $N=10$, and $k=10$ to 40) is $M = N^k$

No, a bigger brain would not help very much! Because every time we **add** a new parameter we **multiply** the number of neurons required by a factor of N (e.g. N=10).

Grandmother cells

If we feed the output of 'high order' neuron-templates (each of which responds to *one* view of a face) into one cell then such a cell would respond to *any* view of that face. This is a hypothetical ***grandmother cell*** (demonstrated as a *daughter* cell here).

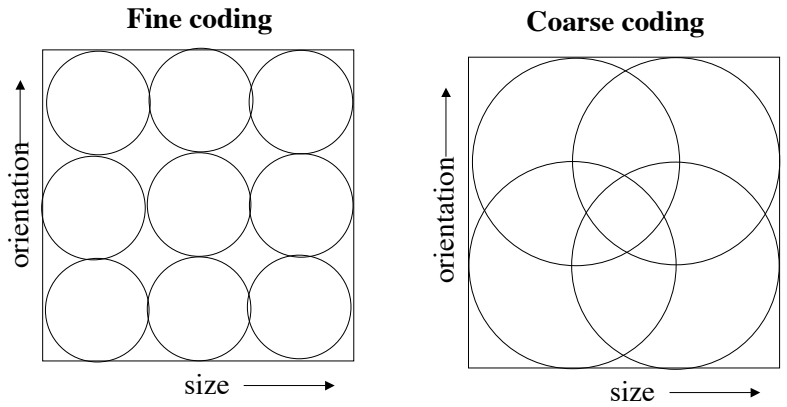
But if a face is defined in terms of k parameters then we need $M = N^k$ template-neurons to support the grandmother cell. And a grandmother cell does not convey information about size, orientation etc.



Interim Summary

- We have pushed argument to its logical conclusion to make a point: *there aren't enough neurons to go round*.
- Because the number $M=N^k$ of combinations of parameter values increases ***exponentially*** with the number k of encoded parameters (colour, orientation, etc). Thus the number of possible **images** of 'A' increases exponentially as k increases.
- If we use one neuron per image of 'A' then this implies that the number of neurons increases with *something* (the base) raised to the power of *something else* (the exponent k). That something is N, but its *exact* value has little impact on the essential problem …

Possible Solution 1: Coarse Coding



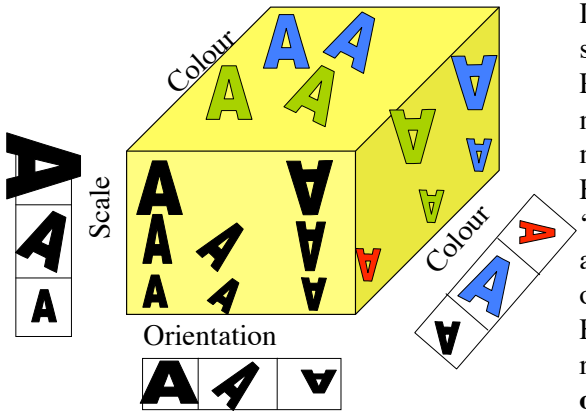
Each circle represents the range of parameter values encoded by a single template-neuron. Each circle is effectively a receptive field in parameter space, a ***parameter field***.

Possible Solution 1: Coarse Coding

- Up to now have used small, **non-overlapping** fields to span parameter space, which requires $\boldsymbol{M = N^k}$ neurons. This is called ***fine coding***.
- If use large, **overlapping** fields with **diameter** D then this is called ***coarse coding***, also as ***population coding***.
- We can span parameter space with same decoding accuracy using fewer neurons:

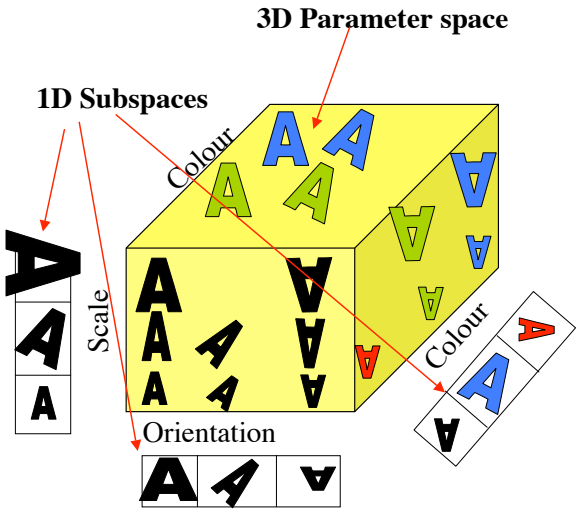
  $M_{coarse} = DC^k$ (compare $M_{fine} = N^k$)
- Coarse coding only reduces the value of the constant (base) from N (number of *cells* per parameter) to $C = N/D$, this has little impact on the problem ..
- So we still have something C raised to the power of k … and so the problems is **exponential** using both fine *and* coarse coding.

Possible Solution 2: Parameter Subspaces



Decompose 3D parameter space into 3 lots of 1D spaces. Represent the 3D block of neurons as 3 1D rows of neurons.
Each of the 3 neurons in the 'scale' row responds to letter A at **one size** at **any** orientation or colour.
Each neuron in the colour row responds to letter A at **one colour** at **any** orientation or size, and so on.

Parameter spaces and subspaces 1



3D parameter space requires N^k neurons.
Three 1D subspaces require N*k neurons.

If N=10 then …

OLD
Require N^3 = **1000 neurons** if use 3D parameter space.

NEW
Require
3N = 3x10=**30 neurons** if use 3 sets of 1D parameter subspaces.

Parameter Subspaces 2

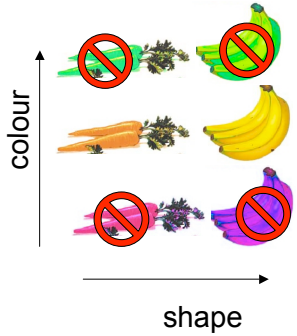
- Can decompose any *k*D parameter space into *k* separate 1D subspaces.
- For example, if neurons in one brain region respond to colour irrespective of scale/orientation/motion/location then that brain region represents a 1D subspace.
- **Area V4 contains neurons like this.**

Subspaces: The Binding Problem

- But using subspaces may only permit a *temporary* escape from the exponential problem.
- This is because the small number of parameters encoded in each subspace must be *recombined* at some stage.
- For example, neurons in V4 encode only one parameter, colour (V4 represents a 1D parameter subspace).
- If colour information is not recombined with (or ***bound*** to) parameters such as object type, location, speed, and size then the brain would not be able to tell which object is which colour, size, etc.
- For example, does this image contain a *red spider* and *blue bird* or a *red bird* and a *blue spider*? Without ***binding*** colour to object, the brain could not tell.
- This is an example of the ***binding problem***.

Reasons to be cheerful?

Reason 1: Some parameter ***values*** (e.g viewpoints) are very uncommon (e.g. assume generic viewpoint). Implies do not have to code for *every possible value* of every parameter.



Reason 2: Some ***combinations*** of parameter values are uncommon (e.g. bananas are not purple, children do not have two heads). Implies do not have to code for *every possible combination* of parameter all values.

Reasons to be cheerful?

- **Reason 3**: Some parameters can be 'engineered away' (e.g.preclude ***position invariant recognition*** by foveating on object).
- **Reason 4**: Do not usually require ***optimal*** solution, can find good but sub-optimal solution.
- Together the above 'tricks' may mean that the vision system does not need to span all parameter values in parameter space.
- But the problem is still **exponential**. Most exponential problems can only be solved approximately. This *may* account for why vision is so fallible …

Summary

Tiling Parameter Space: If each neuron prefers a specific value for each of k parameters, and if there is one neuron for every *combination* of N values of each parameter, then we would require $M = N^k$ neurons to *tile* the k-dimensional parameter space.

As we do have that many neurons, it is unlikely that *pure* template-matching is used by the brain.

References

Essential

- Frisby JP and Stone JV, Chapter 11, Seeing and Complexity.

Background

- Ballard, DH, "Cortical connections and parallel processing: Structure and function", in Vision, Brain and cooperative computation, pp 563-621, 1987, Arbib, MA and Hanson AR (Eds). Download (2.8Mb) from http://cognet.mit.edu/library/books/view?isbn=0262010941
- This paper provides a good account of the problems covered in this lecture, plus other material not covered in the lecture.