Chapter 12: Early Vision

Alan Yuille and Dan Kersten

May 6, 2015

Lecture 12.1

- This lecture gives an overview of early vision and basic concepts.
- We discuss how vision can be divided into low-, mid-, and high-level vision.
- We discuss the visual areas and their properties.

Visual phenomena

- These lectures cover many visual phenomena.
- We recommend Prof. Michael Bach's website, `http://michaelbach.de/ot/` for many fascinating demonstrations of these phenomena (with explanations).
- We suggest (1) Hidden Figures, (2) Rotating Face Masks, (3) Ames Window, (4) Neon Color Spreading, (5) Dress Code Enigma, (6) Adelson's "Checker Shadow" Illusion, and (7) Biological Motion.
- In addition, we encourage you to familiarize yourself with IPython Notebook in preparation for the interactive demos in later sections by going to website: `http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261`.

What is vision?

- Vision is the process of extracting information from light arriving at the retina.
- Humans can estimate a rough approximation of the three-dimensional scene that has generated the image.
- Humans can rapidly attend to different regions of the scene and ignore the rest.
- Vision is also used to enable actions, such as grasping objects or determining where to put your feet while hiking.
- In summary, vision performs a range of *visual tasks* that extract information from the scene in order to achieve goals.

Extracting information from images

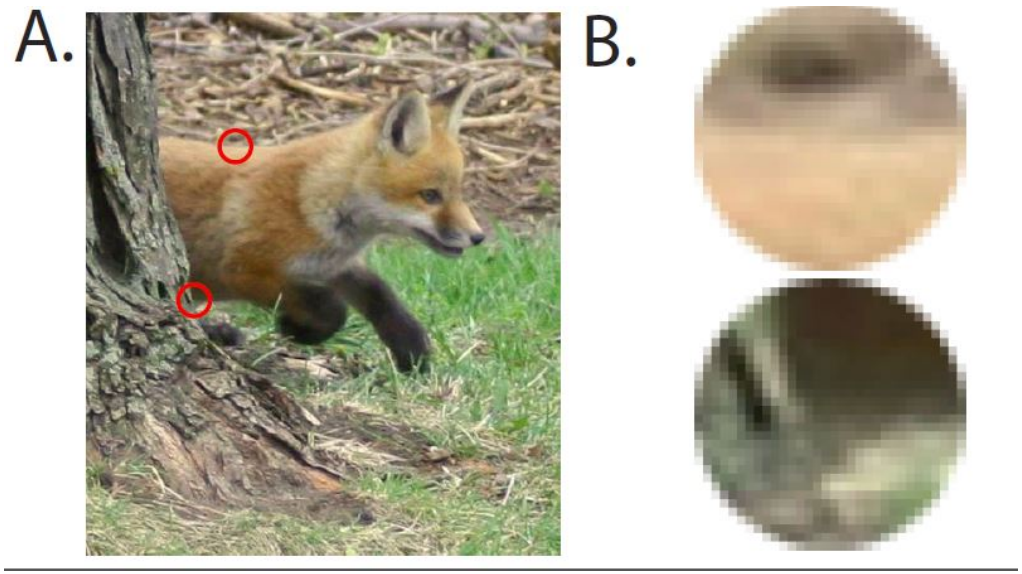


Figure 1 : **(A) Humans can extract a lot of information from a single image. For example, "There is a young fox emerging from behind the base of a tree not far from the viewpoint; it is heading right, stepping through short grass and moving quickly. Its body fur is fluffy, reddish-brown, light in color, but with some variation. It has darker colored front legs and a dark patch above the mouth. Most of the body hairs flow from front to back." (B.) Images are locally ambiguous. These two patches correspond to small parts of the fox's back and the side of the tree, see red circles in (A), but are highly ambiguous without context.**

Vision is extremely difficult

- This is perhaps surprising because humans find it very easy.
- But this is only possible because a very large part of your brain is involved in doing vision. It is estimated that roughly 40% of neurons in the cortex are involved in visual processing.
- Despite decades of work, current machine vision systems perform significantly worse than humans, except for a few highly specialized tasks.
- But probably most other animals, except monkeys and our other close relatives, get far less information from vision, judging by the much smaller numbers of neurons they devote to vision.

Why is vision hard?

- The input to a visual system is the intensity patterns caused by the number of photons, or magnitude of light rays, that are imaged at different positions in the retina.
- The human visual system must decode these intensity patterns and determine that they are caused, for example, by a fox emerging from behind a tree.
- These tasks are particularly complex because the intensity patterns will change significantly if we make small changes to the visual scene. The patterns will vary greatly if we alter the pose of the fox, the lighting conditions, the viewpoint of the observer, and how much the fox is occluded by the tree.
- It is hard to perform *visual tasks*, such as *segmenting the image* into regions corresponding to different objects, performing *object recognition* to determine that a region of the image corresponds to a fox and another region to a tree, or performing *depth estimation* to determine the positions of the objects in the visual scene.

The image and the raw input

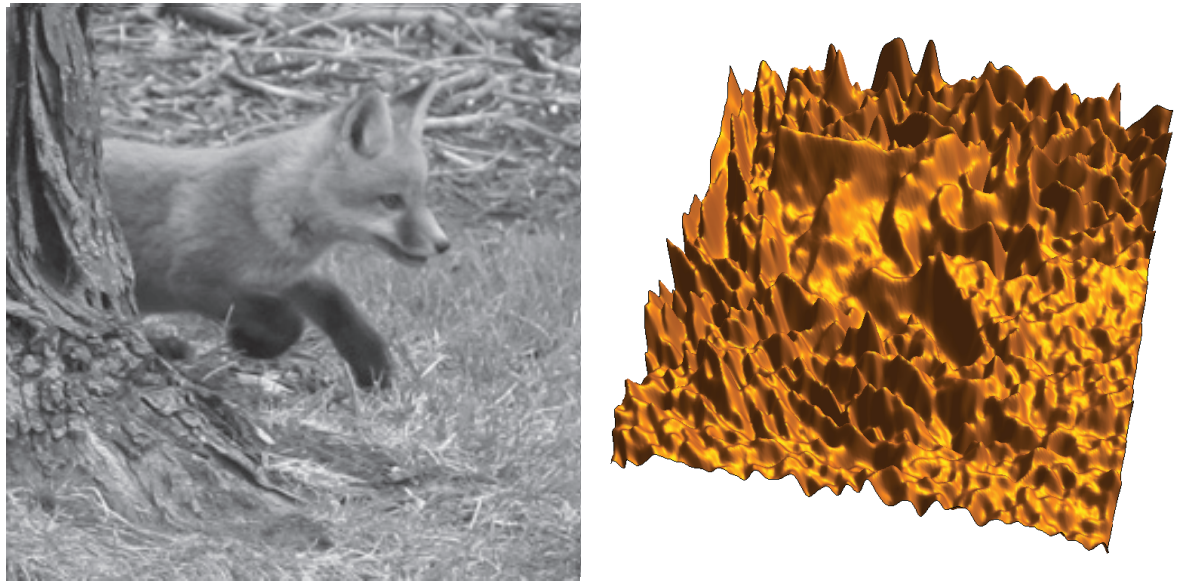


Figure 2 : Why is vision hard? The raw input to the fox image (left panel) is the intensity values plotted as a function of spatial position (right panel). These intensity patterns vary depending on the pose of the fox, the lighting conditions, and other factors. The human visual system must decode this raw input, which is extremely difficult.

The complexity problem

- The main challenge of vision is the enormous complexity of natural images, and their local ambiguities.
- The number of possible images, or intensity patterns, that can be described by a small image array with 100×100 positions, or *image pixels*, is $(256)^{10,000}$ which is astronomically large (Kersten, 1987).
- These images are caused by the very large number of possible objects, which can be arranged in a scene and illuminated in an enormous number of different ways.
- Vision performs the *inverse inference* task of determining the scene from the image.
- It is almost miraculous that humans can simply open their eyes and recognize objects and visual scenes within a few hundred milliseconds (the time it takes to blink an eye).

Natural/ecological constraints

- To perform visual tasks, the visual system must be able to detect and exploit regularities in image patterns.
- These regularities include the assumption that surfaces are generally spatially smooth, that objects tend to move rigidly, that most scenes contain a ground plane, and that objects touch the ground plane at contact points. These assumptions have been called ecological, or natural, constraints (Gibson, 1986; Marr, 1982).
- It is speculated that humans have learned to exploit the structure of natural images and the world through evolution (Glenn-Northcutt & Kaas,1995), early development (Kellman & Arterberry, 2000), or by learning later in life (Green & Bavelier, 2008).
- Vision science researchers can learn these image regularities by applying machine learning methods to image data sets.

Natural constraints and flying carpets

Figure 3 : This image gives the illusion of a flying carpet, where the woman on the towel is perceived to be floating above the beach. The illusion shows that humans use constraints – about ground planes, shadows, and contact points – to interpret images. But in this case, the constraints are violated, because we incorrectly think that the shadow is being cast by the towel, rather than by a flag outside the image.

How is vision studied?

- Vision is studied in three related ways:
  1. at the "behavioral" level, by studying how well humans, and other animals, can perform visual tasks
  2. at the "neural" level, to understand the neural mechanisms (by electrode recording or by non-invasive methods like fMRI)
  3. at the "computational" level, by designing mathematical models and computer vision algorithms that can perform visual tasks
- Some mathematical models of vision attempt to describe how humans or other animals see and account for behavioral or neural data. By contrast, the goal of computer vision is to perform visual tasks without attempting to model how humans or other animals perform them. But both must address similar visual tasks and deal with the complexity of image patterns.

Simplifications

- The visual system is so complex that vision scientists must make simplifications to break it down into manageable pieces. We will question these throughout the lectures.
- They include:
  1. studying visual tasks in isolation instead of addressing the complete visual system
  2. simplifying the visual stimuli
  3. simplifying the models of neurons and neural circuits
  4. simplifying the overall structure of the visual areas of the brain and how they interact with each other
- Vision researchers break vision down into different visual tasks that can be studied separately. These tasks include image segmentation, depth estimation, and object recognition. They are performed by *modules* that output *representations*. Modules, however, might not be localized to distinct parts of the brain.

Marr's framework for vision

- Marr's framework (Marr, 1982) illustrates how visual tasks can be studied in isolation. He proposed that the visual system uses modules to compute a sequence of representations of the image.
- This starts with a *primal sketch* of the image; proceeds to a $2 - 1/2$-*D sketch* which represents the three-dimensional structure of the scene; and concludes with a *3-D* representation of objects.
- The modules interact by outputting representations that are used as inputs to other modules. Marr's framework captures important aspects of the visual system. It also classifies visual tasks into:
  1. *low-level* vision, which processes the image (e.g., produces the primal sketch)
  2. *mid-level* vision, which estimate the structure and properties of geometric surfaces (e.g., produces the 21/2-D sketch)
  3. *high-level* vision, which recognizes objects and analyzes scenes.

Marr's framework illustration

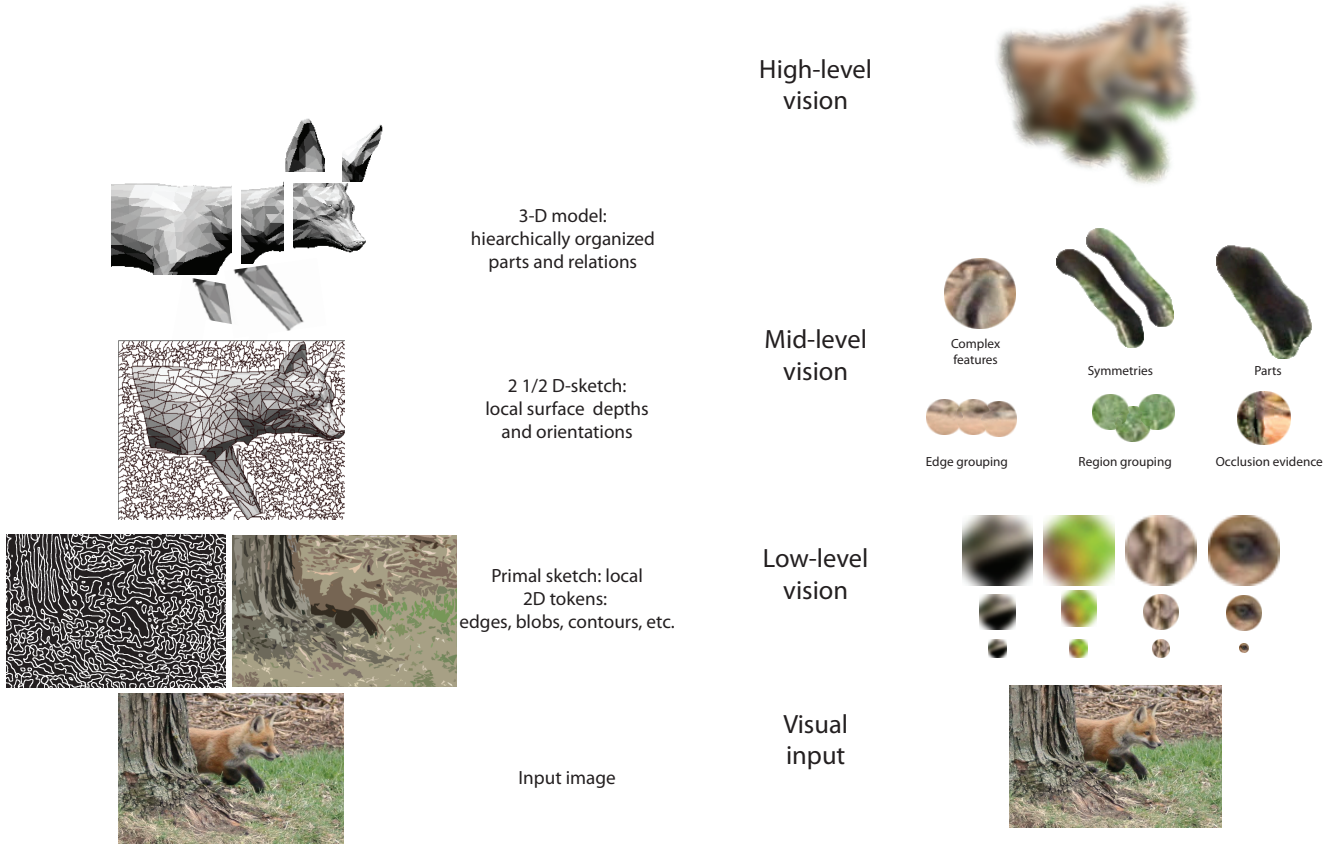


Figure 4 : Marr's framework for vision (left panel) consists of a series of representations. Visual tasks can be classified into low-, mid-, and high-level tasks (right panel), which roughly corresponds to Marr's framework.

Simplification of stimuli

- The set of visual stimuli is so large that it is impractical to study visual systems behaviorally by their response to all stimuli.
- When studying specific visual tasks, it is sensible to use only stimuli that contain information, or *visual cues* specific for these tasks. Good experimental design requires controlled stimuli so that the difficulty of performing a specific task can be quantified in terms of varying a small number of variables.
- For these reasons the study of vision is often simplified by using synthetic stimuli. For example, the ability of humans to perceive depth from Julesz's random dot stereograms (Julesz, 1971) demonstrates that humans can perceive depth when objects are not present, see figure (5).
- But too much reliance on synthetic stimuli can be misleading, and there is concern that experimental findings on synthetic stimuli may not generalize to human, or mammalian, abilities in more natural situations (Carandini, 2005; Yuille & Kersten, 2006).

Example of simplified stimuli

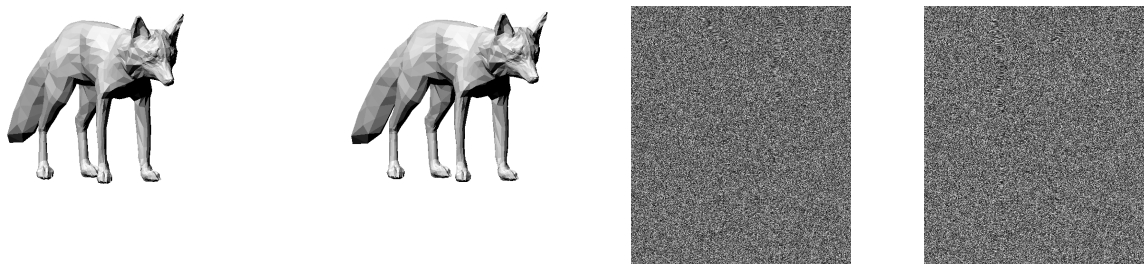


Figure 5 : Binocular stereo of the fox for real images and for Julesz's random dot stereograms. The left two images are a stereo pair (the left and right images) of a fox; when fused (e.g., by a stereo viewer presenting the left and right images to the right and left eyes, respectively), the images yield the three-dimensional shape of the fox. The right two images are stereo pairs of random dot images of a fox. When fused, they also give the three-dimensional shape of the fox.

Simplified models of neurons and neural circuits

- Simplifications must be made when modeling neurons and neural circuits. The *integrate-and-fire* model is standard, but real neurons are more complicated. They may signal information by a sophisticated "neural code" involving the precise timing of action potentials.
- There are wide varieties of neurons that differ in their anatomy and function. There is also evidence that neural circuits can behave differently in different situations.
- The numbers of neurons involved in visual perception is extremely large. This means that simplifications need to be done when studying the overall structure of visual areas of the brain and how they relate to each other. We do not yet have wiring diagrams describing the connections between neurons within each visual area.

Low-, mid- and high-level vision

- Low-level visual tasks estimate local properties of the image. They include finding the boundary of an object (without deciding what the object is) and estimating the motion flow.
- Mid-level visual tasks estimate properties of geometrical surfaces, the shape and position of surfaces in the visual scene, and their depth ordering.
- High-level visual tasks estimate properties of objects, scene structures, relationships between objects, and actions of objects.
- In addition, each level provides information that is passed on to the next level, as illustrated by Marr's theory.
- This organization can be thought of in terms of the knowledge available at each level. Low-level vision knows only about image patterns. Mid-level knows about geometric surfaces. High-level knows about objects. The flow of information from low- to high-level vision is from generic to specific.

Low-level vision

- Low-level vision includes tasks such as detecting edges, performing segmentation, and extracting representations of image patterns that can be used for higher-level processing.
- Low-level vision can exploit statistical properties of images that are true for most images (e.g., that images tend to be piecewise smooth).
- Low-level vision also includes estimating the local motion of images by finding the correspondence between points in images taken at different times. This is done by matching regions that have similar intensity properties.

Low-level vision: Edge detection

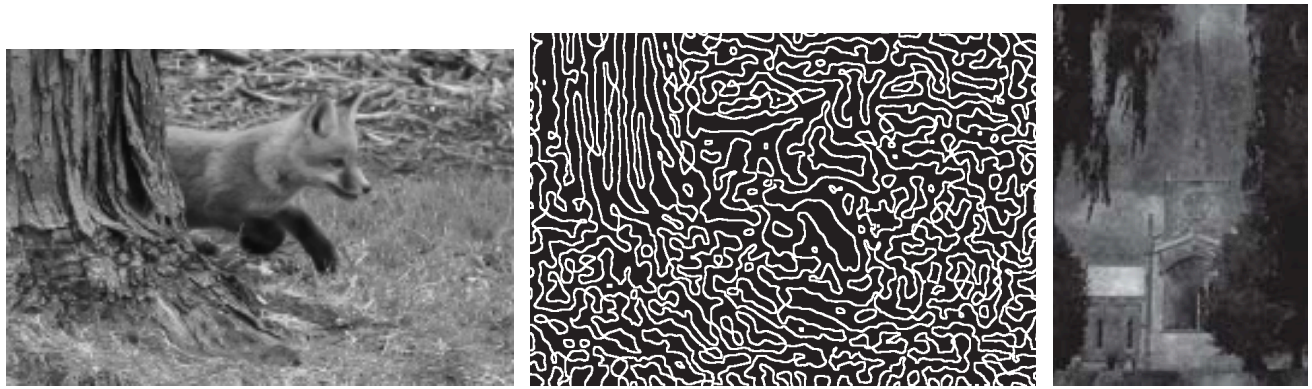


Figure 6 : The edges of the fox image (left panel) detected by low-level processing (center panel). Some edges lie on the boundary of objects, like the fox and the tree, while others are due to properties of the textures (e.g., the grass or the bark of the tree). It is difficult to distinguish between these different types of edges. The church steeple (right panel), and the position of its edges, is obvious if you view the whole image, but almost impossible to see locally because there is no strong local evidence for the edges of the steeple. This shows that sometimes edge detection is impossible except when done in conjunction with object detection, as when low-level vision proposes many possible edges that are validated, or rejected, by object models.

Low-level vision: Aperture problem

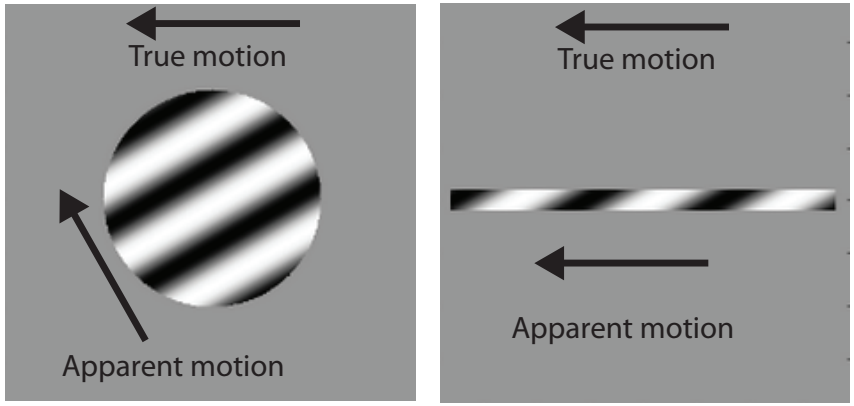


Figure 7 : These images show black and white bars, whose *true motion* is leftward, viewed through two apertures (circular and rectangular). But the motion is locally ambiguous because we can directly observe only the motion component normal to the bars (we cannot detect any motion tangential to the bars), and so the observed stimuli is consistent with many possible motions. The human visual system uses constraints to resolve these ambiguities. For these stimuli, humans assume that the motion is as slow as possible and hence is perpendicular to the bars (assuming that the unobservable tangential component is zero), as indicated by the *apparent motion*. More generally, humans tend to assume that motion is slow and smooth.

Interactions between low and higher level vision

- Although low-level vision can be studied in isolation there is evidence that it interacts strongly with higher-level vision.
- For example, using low-level processing alone, it is usually impossible to detect the edges of the objects in an image without making mistakes.
- The dalmatian dog illusion demonstrates how the extreme ambiguity of low-level cues for edges make it very hard to detect the dalmatian (see `http://michaelbach.de/ot/cog_dalmatian/`).

Mid-level vision

- Mid-level visual processing involves geometry, materials, and lighting but not specific objects or scene structures. For example, mid-level vision "knows" about surfaces of red metal, but not about red cars.
- Mid-level vision includes inferences about depth ordering of surfaces and reasoning about how surfaces can partially occlude each other. The Kanizsa triangle, on the next slide, shows that humans can perceive occluding surfaces even if there is little local evidence for them.
- The Kanizsa triangle is an example of *Gestalt* grouping phenomena, many of which can be explained in terms of a human tendency to interpret images as simple geometric structures (Kanizsa, 1979).

Mid-level vision: Kanizsa triangle

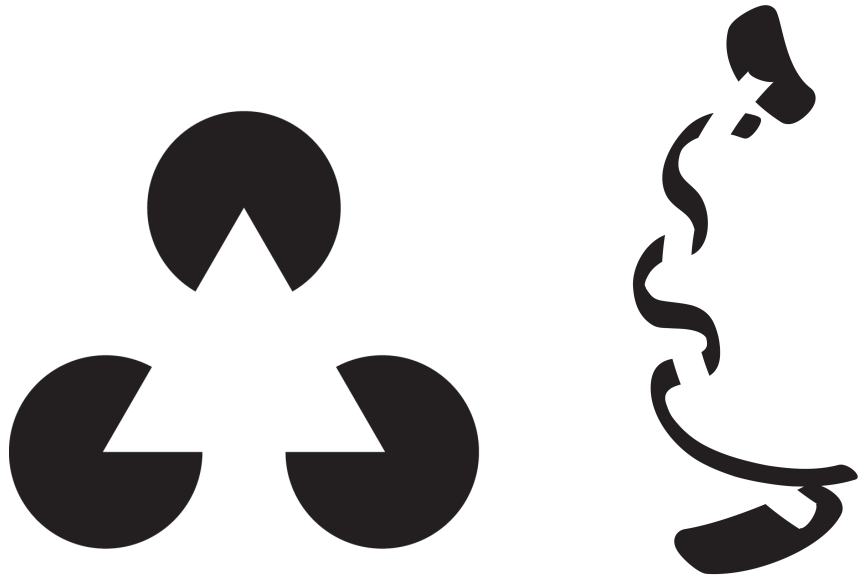


Figure 8 : The Kanizsa triangle (left panel) is perceived as a white triangular surface that partially occludes three black disks. It shows the tendency of humans to interpret images in terms of geometric structures. Another example of this (right panel) shows that human tend to "explain away" gaps in the black band by positing a white band lying sometimes above and sometimes below the black band. For an interesting variant, see `http://www.michaelbach.de/ot/cog-kanizsa/`, which shows how the effect can disappear if other cues are present.

Mid-level vision: Flying carpet

- In this interpretation, the visual system assumes that most images contain a ground plane with objects standing on it (e.g., a man standing on a lawn). The contact points of the objects with the plane specify the positions of the object in the scene. This is based on assumptions about geometry alone without requiring any knowledge of specific objects.
- This relates to shape from perspective. By the laws of *perspective projection*, if there are parallel lines in the image (such as the tracks of a railway line), then the projection of these lines in the image will converge at a *vanishing point.*
- Humans can use vanishing points to estimate the orientation of the ground plane. More information about the scene can be extracted if the image contains several vanishing points corresponding to surfaces that are orthogonal in space.

Mid-level vision: Binocular stereo

- Binocular stereo is another vision module, which estimates the depth and orientation of surfaces. Humans have the ability to get depth from two eyes – hence the popularity of so-called 3D movies.
- This requires solving a *correspondence problem* between features in the two eyes which are caused/imaged by the same point in space. If correspondence can be performed, then the depth can be estimated by trigonometry. The correspondence problem is made easier byf the *epipolar line constraint*, which means that corresponding points require only searching in a one-dimensional direction.
- But knowledge of the epipolar lines requires knowing the direction of gaze of the cameras (maybe done by feedback from muscles controlling the eyes, or by *calibration*). Note that partial occlusion can happen, when part of the scene is visible to one only eye. Da Vinci was the first to point out that this was a useful visual cue.

Mid-level vision: Stereo figure

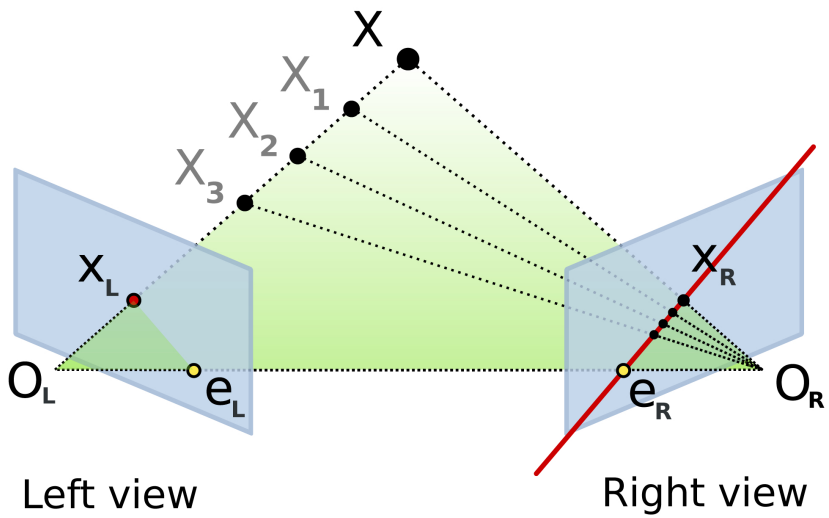


Figure 9 : Stereopsis and epipolar lines. A point $\vec{x}$ in three-dimensional space gets projected onto positions x_L and x_R in the left and right eyes. This uses a pinhole camera model of each eye, where the eye is specified as a plane (in grey), and O_L and O_R represent the centers of projection. All points on the plane defined by O_L, O_R, and $\vec{x}$ get projected onto straight lines $\vec{e}_L$ and $\vec{e}_R$, the corresponding *epipolar lines*, in the two eyes, as shown by the projections of x_1, x_2, x_3 onto the right eye. If we alter the position of the point $\vec{x}$ in space, then we will get a family of corresponding epipolar lines. The *epipolar line constraint* states that points on an epipolar line in one eye can only be matched to a point in the other eye on the corresponding epipolar line.

Mid-level vision: Other visual cues

- Humans can also get three-dimensional shape information about surfaces from shading, texture, and even contours.
- Shape from shading assumes a model, usually Lambert's law (Basri, 2003), for how the image is formed in terms of the lighting and the shape of the viewed object. In some conditions, this can be inverted to estimate the shape of the object from the intensity patterns, which is called *shape from shading* (Basri, 2003).
- Shape from texture arises if a surface has a regular pattern of texture. This pattern will be distorted by the shape of the object, which enables the shape of the surface to be estimated from the intensity patterns by *shape from texture* (Knill, 2001).
- Finally, certain contours naturally suggest shapes, which is called *shape from contour* (Knill, 2001).

Mid-level vision: Shape and shading

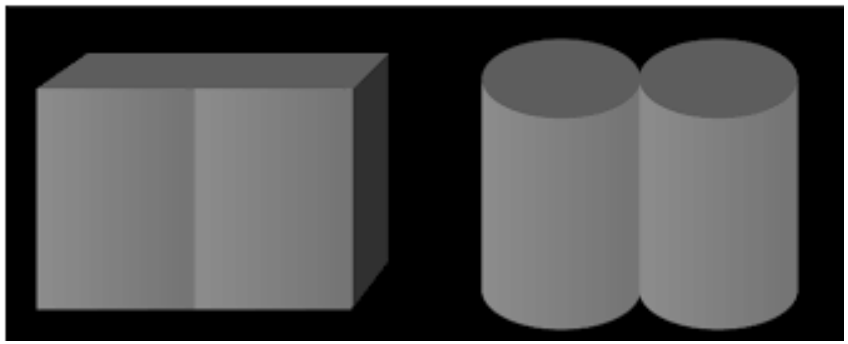


Figure 10 : Shape from from and shape from shading are coupled. The shapes of the contours affect the perception of the shading patterns. The intensity patterns are the same in both stimuli but the shape of the boundaries makes the perception of shading different in the two cases (Knill & Kersten, 1991).

Visual cues

- These vision modules depend on specific properties of images called *cues*, for example, the shadow on the sand in figure (3).
- These modules, and the visual cues they rely on, can be modeled and studied separately, but they are often coupled. For example, Knill and Kersten (1991) illustrate that shape from contour can alter the perception of shading patterns and material properties, as we saw in the previous slide.
- The complexity of image formation often makes it hard to decouple visual cues. Under certain lighting conditions, it is even difficult to tell the color of an object being viewed (see `http://michaelbach.de/ot/col-dress/`).
- There are also interactions between mid-level and high-level vision. For example, humans perceive an inverted (i.e., concave) face mask to be a normal convex face even when binocular cues are present (see `http://michaelbach.de/ot/fcs_hollow-face/`).

The visual system and the brain

- This section is a brief overview of what we know about how the brain does vision.
- It reviews the areas of the brain that perform visual processing, the relationships between them, the structures of these areas, and the visual tasks they perform.
- Because of the complexity of the visual system our knowledge of these issues, though considerable, is limited.
- It is based on a combination of anatomical studies, electrophysiological recordings, and noninvasive imaging methods such as functional magnetic resonance imaging (fMRI).

The retina

- Visual neural processing begins with the retina, which transmits information to the visual cortex via the lateral geniculate nucleus (LGN). The anatomy and electro-physiology of the retina and LGN have been studied in detail (Merigan & Maunsell, 1993; Gollisch & Meister, 2010; Briggs & Usrey, 2011).
- The retina converts intensity patterns – the light rays that reach the retina – into patterns of neural activity. This starts with *photoreceptors* which are directly activated by light and are efficient at "capturing" photons (Rieke et al., 1997).
- The remaining set of neurons in the retina, in particular *ganglion cells*, process the photoreceptors output and encode it for transmission via the *optic nerve* to the rest of the brain.
- The retina functions as a sophisticated camera that captures the information in the incoming images and encodes it so that it can be transmitted to the visual cortex, but eye movements means that the retina is not a passive device and instead actively searches for information, see Zhaoping (2014).

The challenges of the retina

- The retina faces two challenges:
  1. the enormous variability of intensity in natural images
  2. the ability to encode the images so that they can be transmitted efficiently and robustly
- Neural models of the retina are largely motivated by these challenges.

The dynamic range of images

- The intensity of natural light varies enormously from faint starlight to bright sunlight, with intensity magnitudes ranging from 1 to 10^9. Moreover, the changes of intensity within specific images can also vary hugely (Demb, 2002).
- But neurons have limited ranges of response, and hence they cannot encode these huge ranges of intensity. Hence many theories of the retina propose that the ganglion cells perform *gain control* and filter the images so that they capture only the *local contrast* – the differences of intensity between nearby parts of the retina – and hence reduce the need to represent the entire intensity range.
- Observe that digital cameras perform similar functions, since they convert incoming light patterns into digital representations where the intensity only takes 256 values, from 0 to 255 (in each color channel).

Encoding information

- The retina must encode the image information so that it can be transmitted through the optic nerve to the rest of the brain for further processing. The image information is transmitted through a relatively small number of fibers in the optic nerve (compared with the number of photo-receptor cells).
- Information theory offers guidelines for how information can be encoded efficiently based on statistical knowledge of the stimuli. Researchers have applied this theory to predict retinal properties with some success, but this work is out of scope of this lecture (Zhaoping, 2014).

The complexity of the retina

- Theories of the retina illustrate the "simplification issues" which re-occur throughout the chapter and these lectures. At the computational level, the theories for describing how the retina deals with intensity are simpler than the engineering methods used by computer vision and image processing researchers who deal with the same challenges.
- At the experimental level, many of the findings about retinal neurons are based on simplified models of neurons obtained from studying their responses to synthetic stimuli. Moreover, despite considerable knowledge of the anatomy, only recently have studies of detailed wiring diagrams and characterization of fifty or more anatomical types of neurons (Masland & Martin, 2007).
- It is also unclear why so many neurons are required to overcome the two challenges. Indeed it has been argued that the retina is considerably "smarter" than current theories suggest (Meister & Berry, 1999; Gollisch & Meister, 2010) and may require detecting motion, expansion, extrapolation, and more generally adapting to the complexity of image patterns.

The LGN

- The output from the retina is transmitted to the LGN and then to the visual cortex, where it arrives in *visual area V1*.
- The LGN is generally believed to have limited function as a way station on the route to the visual cortex. Hence current models of LGN neurons are fairly simple.
- But there is reason to believe that LGN is more complex. For example, there is substantial feedback from V1 to LGN (Briggs & Usrey, 2011) as well as connections between LGN and other areas aside from V1 (Sherman & Guillery, 2002; Nassi et al., 2006).

Cortical visual areas and the relationships between them

- The visual cortex can be decomposed into a number of visual areas based on anatomical and electrophysiological measurements (Van Essen et al., 1992). The visual areas V1, V2, V4, medial temporal (MT), medial superior temporal (MST), and the inferior temporal cortex (IT) are illustrated in the figure on the following slide. It is common to concentrate on two *hierarchical streams*:
  1. The *ventral stream* consists of V1, V2, V4 (the functional organization of V3 has been under some debate), and the infero-temporal (IT) areas of extrastriate cortex. This pathway is believed to perform object detection and scene understanding.
  2. The *dorsal stream* goes from V1, MT, to the parietal cortex. It is believed to be used for analysis of the movements and positions of objects as they relate to navigation and actions (Milner & Goodale, 2006).
- Although the distinction between ventral and dorsal pathways is well established (Lennie, 1998), this is probably a simplification (Schenk & McIntosh, 2010).

Schematic of the visual cortex

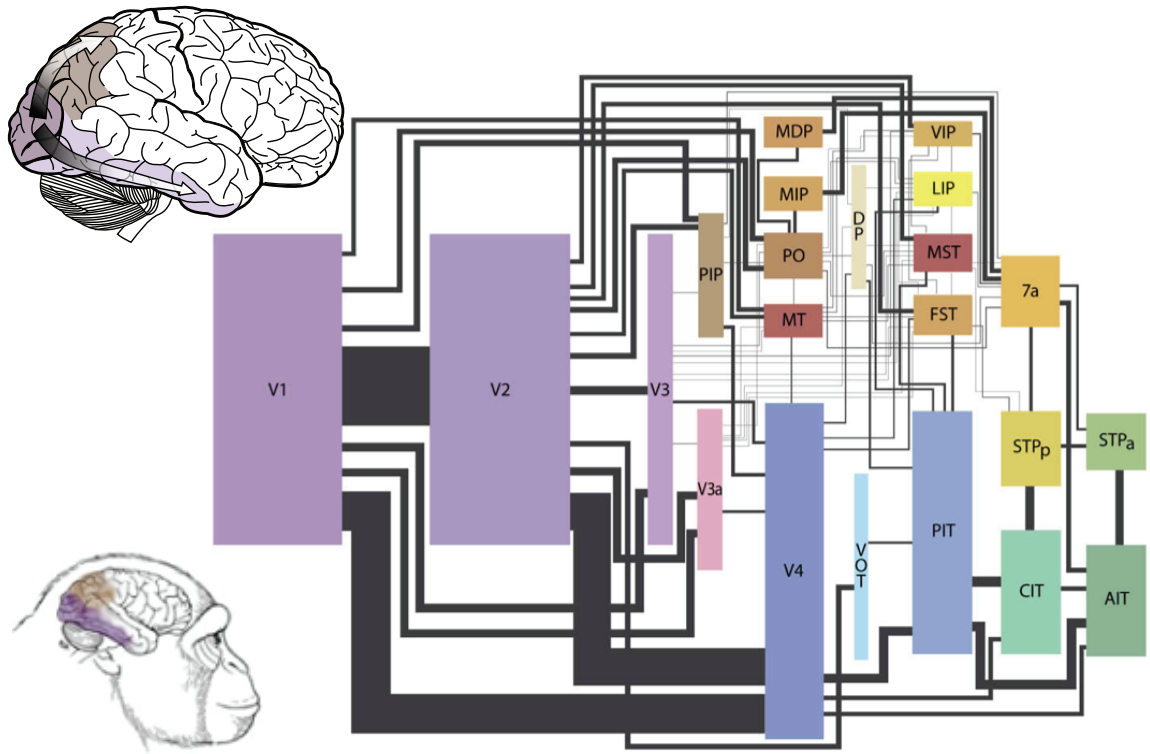


Figure 11 : Left panel (top and bottom) illustrate the monkey visual cortex. The right panel is a schematic of connections between visual cortical areas in the macaque monkey brain. The colored rectangles represent visual areas (see Felleman & van Essen, 1991). The black lines show the connections between areas, with the thickness proportional to the number of feedforward fibers. Areas in cool and warm tones belong to the ventral and dorsal streams, respectively. (Wallisch & Movshon, 2008; Lennie, 1998).

Sizes of visual areas

- The size of the visual areas varies greatly.
- The first two areas, V1 and V2, are enormous and together account for roughly 70% of the number of neurons in the visual cortex (hence 30% of the neurons in the entire cortex). The number of neurons in V1 is much higher, by a factor of at least two hundred, than the number of fibers that leave the eye.
- Indeed it has been estimated that this is more, by a factor of several hundred, than the amount needed to represent the information conveyed by the LGN (Lennie, 1998), consistent with the idea that the purpose of V1 is to start interpreting the image instead of simply encoding it.
- Another major feature of the hierarchy of visual areas is that their size gets progressively smaller as they rise in the hierarchy. For example, V4 is much smaller than V2, and visual areas within IT are considerably smaller than V4.

Structural organization: Retinotopy

- Electrophysiology studies the response of neurons to synthetic stimuli with different *perceptual dimensions*, such as position, orientation, color, texture, shape, sensitivity to input from both eyes, and motion.
- Neighboring neurons in early visual areas usually respond to similar regions of the image. These areas are roughly *retinotopic* in the sense that their spatial organization is similar to that of the image at the retina, with a spatial transformation (Schwartz, 1980).
- This retinotopic structure is strongest in V1 and V2 and gets weaker at high visual areas. Neurons are often classified by how they are *tuned* to specific perceptual dimensions. But neurons in V1 respond to several dimensions (Lennie, 1998), and classification is challenging in higher areas (Roe et al., 2009; Roe et al., 2012).
- Mapping with optical techniques (Lu & Roe, 2007; Kinoshita et al., 2009) has shown that most early visual areas are organized retinotopically, although this is strongest in V1 and V2.

Other salient structures

- Other salient structures of V1 include *hypercolumns* (∼1-2 mm), consisting of:
  1. a regular array of orientation columns, perpendicular to the cortical surface, in which orientation selectivity of neurons is approximately the same and varies slowly parallel to the cortical surface
  2. ocular dominance columns (where the proportion of input from both eyes is constant within each column, but varies smoothly between columns)
  3. a lattice of cytochrome oxidase blobs – sensitive to color (Hubel, 1982; Livingstone & Hubel, 1984)
- From a more abstract perspective, the organizational structures of hypercolumns can be partly explained by the need to map stimulus dimensions (e.g., retinal position, orientation, etc.) onto two-dimensional cortical surface while attempting to make the map as smooth as possible (this is not possible, on topological grounds, so discontinuities occur) (Durbin & Mitchison,1990).

Hierarchical organization

- A notable property of these visual areas is their hierarchical organization, which relates to the distinction between low, mid, and high levels.
- Broadly speaking, V1 and MT seem to be involved in low-level processing; V2, V4, and MST in mid-level vision; and IT in high-level vision. Hence early vision is believed to be mostly performed in V1, V2, V4, MT, and MST.
- There is a strong tendency for receptive fields to be larger as they ascend the visual hierarchy. Compared to those in V1, the receptive fields are 2-3 times bigger in V2, 4-5 times larger in V3/VP, and 7-10 times larger in MT. But, conversely, the receptive fields become increasing specific to stimuli, and stimuli of greater complexity, as we move up the ventral stream. In summary, the receptive fields become more invariant to position and more specific to structure as we proceed up the ventral stream from V1 to V2 to IT (Rust & DiCarlo, 2010; Logothetis & Sheinberg, 1996).

Experimental methods

- Many of the findings are based on electrophysiological studies of monkeys and non-invasive studies of monkeys and humans. Researchers have found close relationships in early visual areas V1, V2, V3 (Wandell et al., 2007), but not always at higher areas (Wandell & Winawer, 2011).
- Noninvasive studies like fMRI suffer from limited spatial and temporal resolution and currently can observe only coarse properties of the visual system.
- Electrophysiology is restricted to recording from a small number of neurons in response to a limited range of stimuli. See Carandini (2005) for the problems of interpreting these results in the early visual cortex. It is not easy to predict the response of neurons in V1 to natural stimuli.
- There is considerable progress in developing experimental methods that can probe the properties of neural circuits in much greater detail, such as optogenetics, which may revolutionize our understanding pf the early visual system.

Lecture 12.2

- This lecture introduces linear models of neurons, describing how they are used to model the receptive fields of neurons in the retina, the LGN, and the *simple cells* in V1. We also describe complex cells in V1.
- Then we provide a different perspective of these cells as representing images and introduce overcomplete bases and sparse encoding.
- This lecture includes two exercises involving interactive demos: (12.2.1) Linear filters and convolution, and (12.2.2) Gabor filters.

Linear models of simplified cells

- This section introduces a model of a simplified cell.
- The cell receives inputs $\mathbf{I} = (I_1, I_2, ..., I_N)$ from *dendrites* that are weighted by *synaptic strengths* $\mathbf{w} = (w_1, w_2, ..., w_N)$.
- These are summed at the *soma* (cell body) to obtain:

$$\mathbf{w} \cdot \mathbf{I} = \sum_{i=1}^{N} w_i I_i$$

.

- The cells outputs a response $f(\mathbf{w} \cdot \mathbf{I})$ along its *axon*, indicated by the firing rate of the neuron. $f(.)$ is a monotonic function (see next slide) but in this lecture we use a linear approximation:

$$S = \mathbf{w} \cdot \mathbf{I} = \sum_{i=1}^{N} w_i I_i$$

The nonlinear function $f(.)$

- $f(.)$ is monotonic nonlinear function, which takes value 0 if the input is small, then increases linearly in the *linear regime* until it saturates at a maximum value.
- A typical choice of $f(.)$ is the sigmoid function $f(\mathbf{w} \cdot \mathbf{I}) = \sigma(\mathbf{w} \cdot \mathbf{I} - T)$, where T is a threshold and $\sigma(.)$ is a soft threshold.
- In this lecture, we ignore $f(.)$ and study the behavior of the model in the linear regime.
- Cells in the retina and the LGN are often modeled without the nonlinear function $f(.)$, adding instead a constant C to the output, to account for spontaneous firing of the cell, and yielding an output $\mathbf{w} \cdot \mathbf{I} + C$, see (Zhaoping, 2014).

Linear filter figure

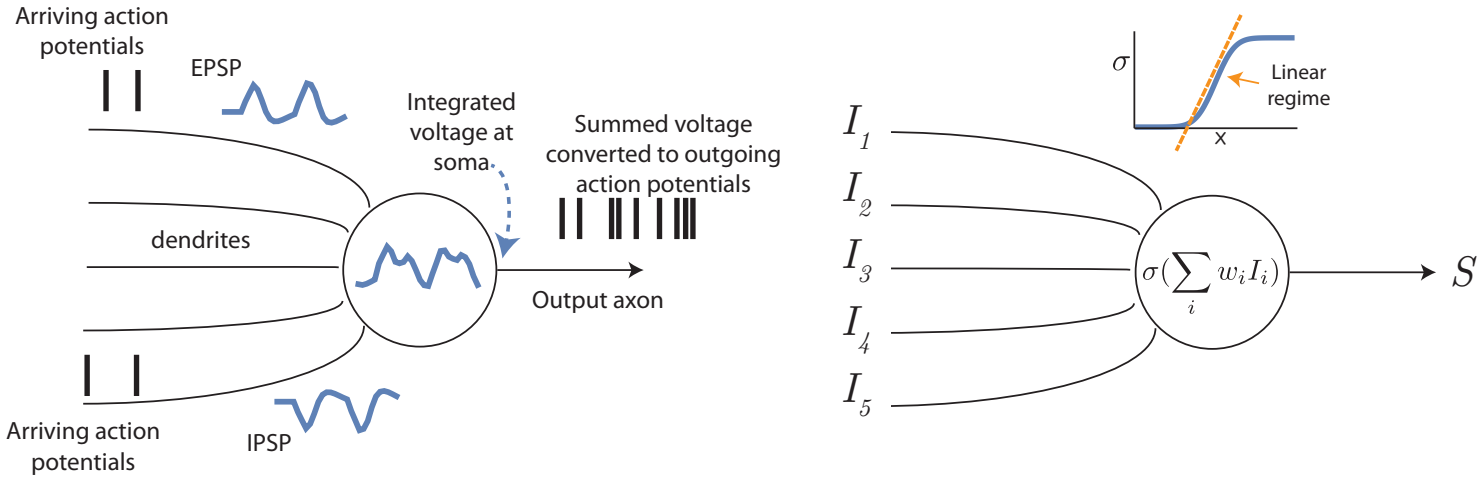


Figure 12 : Left: A neuron receives input – action potentials from other neurons – at its dendrites, which generate excitatory and inhibitory postsynaptic potentials (EPSPs and IPSPs respectively), whose voltages are integrated at the soma and converted to outgoing action potentials. Right: A simplified model of a neuron. There are inputs $(I_1, ..., I_5)$ at the dendrites, with synaptic strengths $w_1, ..., w_5$. These are summed at the soma, $\sum_i w_i I_i$, and the output S is given by a sigmoid function $\sigma(\sum_i w_i I_i)$. The sigmoid function $\sigma()$ (top right) has a linear regime (brown line) and low and high thresholds.

Linearity and superposition

- This model $S = \mathbf{w} \cdot \mathbf{I}$ is linear in two respects.
- First, it is linear in the input $\mathbf{I}$ so that if we double the input $\mathbf{I} \mapsto 2\mathbf{I}$, then the output doubles also $S \mapsto 2S$. Second, it is linear in the weights $\mathbf{w}$.
- Most importantly, it obeys the *principle of superposition*, so that if S^1, S^2 are the outputs to input $\mathbf{I}^1, \mathbf{I}^2$ respectively, then the output to input $\lambda_1 \mathbf{I}^1 + \lambda_2 \mathbf{I}^2$ is $\lambda_1 S_1 + \lambda_2 S_2$.
- This result is important for characterizing the response of simple neural cells, since it implies that we can determine the output of the cell to any stimulus by observing its response to a limited set of input stimuli $\mathbf{I}$.
- Note that this property still remains if we re-introduce the nonlinear function $f(.)$, provided the function is known.

Retinotopy (I)

- The retinotopic organization of the early visual system has two implications for these cells.
- *First*, the weights of the cell depend on its retinotopic position $\vec{x} = (x_1, x_2)$ and the positions $\vec{y} = (y_1, y_2)$ of its dendrites.
- We replace the input I_i by $I(\vec{y})$ and the weights w_i by $w(\vec{x} - \vec{y})$. The *receptive field* $w(\vec{x} - \vec{y})$ will typically be zero unless $|\vec{x} - \vec{y}|$ is small.
- The neuron is modeled by:

$$S(\vec{x}) = \sum_{\vec{y}} w(\vec{x} - \vec{y}) I(\vec{y}) = \mathbf{w} * I$$

Retinotopy (II)

- *Second*, retinotopy implies that there are cells with similar properties (e.g., the same weights $\vec{w}$) arranged roughly evenly in spatial position (apart from the log-polar transformations (Schwartz, 1980)).
- This can be thought of as having "copies" of the same cell at all positions in space. In terms of linear filter theory, these sets of cells are *convolving* the image $\vec{I}$ by a filter $\vec{w}$.

Receptive fields in retina and LGN.

- The receptive fields of the ganglion cells in the retina and the cells in the LGN can be determined by measuring the firing rate of the neurons in terms of their response to different input stimuli $\vec{I}$ and estimating a model for the response.
- The experimental findings are that many simple cells have a characteristic receptive field called *center-surround*. But these findings are the result of using synthetic stimuli, and cells' response may be more complex if they are studied using natural stimuli.
- Photoreceptors have different properties, see (Rieke et al., 1997).

On-center and off-center receptive fields

- There are two different types: on-center and off-center. The receptive field weights $w(\vec{x} - \vec{y})$ are radially symmetric and take the form of a "Mexican hat" or inverted Mexican hat, for on-center and off-center cells, respectively (Marr, 1982).
- These cell responses are usually thresholded, e.g., by the sigmoid function, so that they usually give only positive responses.
- The weights $w(\vec{x} - \vec{y})$ can be approximated by the *Laplacian of a Gaussian* (LOG) or by its negative:

$$w_{LOG}(\vec{x}) = -\{\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\} G(\vec{x} : \vec{0}, \sigma^2)$$

where $G(\vec{x} : \vec{0}, \sigma^2) = \frac{1}{2\pi\sigma} \exp\{-(x_1^2 + x_2^2)/(2\sigma^2)\}$.

Illustration of center-surround cells

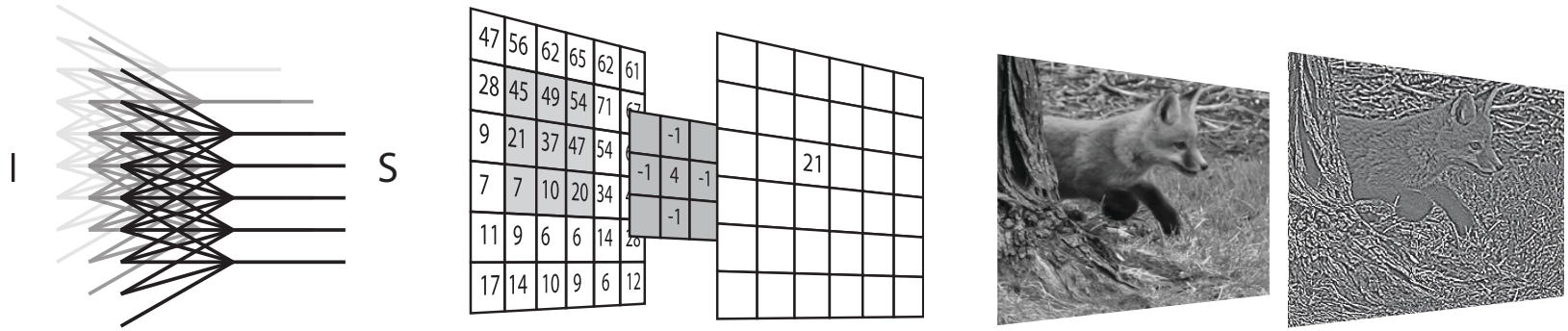


Figure 13 : This figure shows the input-output of a center surround cell (e.g., Laplacian of a Gaussian) in three different ways. First, in terms of the inputs and outputs of neurons (left). Second, in terms of the digitized input image, the filter, and the digitized output (center). The output at each pixel is given by the product of the filter to the appropriate intensity values in the input image, e.g., $4 \times 37 - 1 \times 49 - 1 \times 47 - 1 \times 10 - 1 \times 21 = 21$. Third, in terms of the input and output images (right).

Figure of Gaussians and derivatives of Gaussians

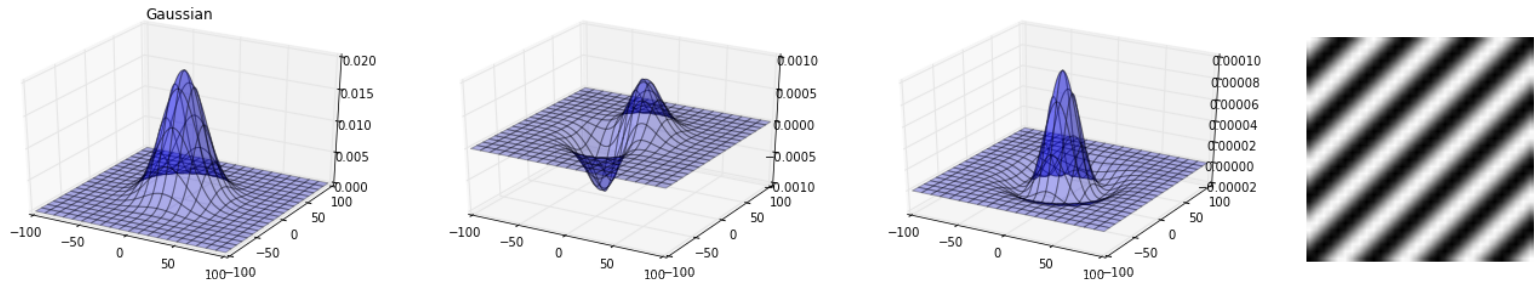


Figure 14 : A Gaussian filter (far left). The first derivative of a Gaussian (left). The Laplacian of a Gaussian, or Mexican hat (right). A sinusoid (far right).

Symmetry and properties of receptive fields

- These cells have two important properties:
  1. They are radially symmetric in the sense that $w_{LOG}(.)$ is invariant to rotation; e.g., suppose we express position $\vec{x}$ in terms of radial components: $x_1 = r\cos\theta, x_2 = r\sin\theta$, then $w_{LOG}(r\cos\theta, r\sin\theta)$ is independent of θ.
  2. The receptive field weights $w(.)$ sum up to zero. More precisely,

$$\sum_{\vec{x}} w_{LOG}(\vec{x}) = 0.$$

- Note that center-surround cells are often modelled as the *differences of two Gaussians*: $w_{DOG}(\vec{x}) = A_1 G(\vec{x} : \vec{0}, \sigma_1^2) - A_2 G(\vec{x} : \vec{0}, \sigma_2^2)$, where σ_1, σ_2 take different values (Zhaoping, 2014). This gives a similar model, if $|\sigma_1 - \sigma_2|$ and $|A_1 - A_2|$ are small.

Purpose of center surround cells: Dynamic range

- These center-surround cells are believed to help deal with the large dynamic range of images.
- Suppose we can express the image locally as $I(\vec{x}) = C(\vec{x}) + B$ where $C(\vec{x})$ is the *contrast*, which describes the local details of the image, and B is the *background*. Then filtering an image by a center-surround cell, whose receptive field sums to 0, removes the background term and preserves part of the contrast.
- More precisely:

$$S(\vec{x}) = \sum_{\vec{y}} w_{LOG}(\vec{x} - \vec{y}) I(\vec{y}) = \sum_{\vec{y}} w_{LOG}(\vec{x} - \vec{y})(C(\vec{y}) + B)$$

$$= \sum_{\vec{y}} w_{LOG}(\vec{x} - \vec{y}) C(\vec{y})$$

Encoding information for transmission

- Receptive fields of this type can also help efficiently encode the information at the retina in order to transmit it efficiently to the visual cortex.
- This can be studied using information theory and the statistics of natural images to predict properties of receptive fields and how they change in different environments (Atick & Redlich, 1992).
- This theory is beyond the scope of this chapter and we refer to the detailed exposition in (Zhaoping, 2014).

Is the retina more complex?

- These models of cells in both the retina and the LGN are well studied. Although many of their properties were estimated using synthetic input data, it has been shown that in some cases, the input image can be estimated from the response of cells in either the retina or the LGN using these types of models Warland et al., 1997; Dan et al., 1996; Carandini:2005).
- But others (Gollisch & Meister, 2010) argue that the retina is more complex, and that, in particular, the neurons may act more as *feature detectors* than as spatial-temporal filters.
- In particular, Gollisch & Meister (2010) describe many findings suggesting that the retina is more complex than the linear filtering model described above. It is known, for example, that if the light levels go down, then the receptive field size becomes larger (Zhaoping, 2014).

Temporal and color properties

- A more realistic model of the output is

$$S(\vec{x}, t) = \sum_{\vec{y},\tau} w(\vec{x} - \vec{y}, t - \tau) I(\vec{y}, \tau)$$

  where $w(\vec{x} - \vec{y}, t - \tau)$ is a space-time filter.

- There are two types of cells with different temporal properties:
  1. M-cells, whose receptive fields are spatially large but temporally small (faster), project to the dorsal stream.
  2. P-cells, whose receptive fields are spatially smaller but temporally larger (slower), project to the ventral stream.

- We can also model the dependence of the cells on the wavelength of the input light by

$$S(\vec{x}) = \int d\lambda w(\vec{x} - \vec{y}) w_c(\lambda) I(\vec{x}, \lambda),$$

  where λ denotes the wavelength and $w_c(\lambda)$ specifies the sensitivity of the cell to color, see (Zhaoping, 2014).

Tuning of receptive fields to sinusoids

- To determine the receptive field of a neuron, we study its response to a class of stimuli while varying the stimulus parameters (i.e., the perceptual dimensions). To find how well the neuron is *tuned* to particular stimulus parameters, see (Hubel, 1982).
- In this section, we analyze tuning when the stimuli are sinusoid gratings.
- We stimulate the receptive field of a neuron by a sinusoid grating

$$I(\vec{x}) = A\cos(\vec{\omega} \cdot \vec{x} + \rho) + I_0,$$

  where A is the *amplitude*, ρ is the *phase*, $\vec{\omega}$ is the *frequency*, and I_0 is the mean light level.
- The frequency specifies the orientation of the stimulus by the unit vector $\hat{\vec{\omega}} = \vec{\omega}/|\vec{\omega}|$, and the period of the oscillation by $|\vec{\omega}|$. The phase ρ shifts the center of the sinusoid. To see this, re-express $A\cos(\vec{\omega} \cdot \vec{x} + \rho) = A\cos(\vec{\omega} \cdot (\vec{x} - \vec{x}_0))$, where $\vec{x}_0 = -\rho\vec{\omega}/|\vec{\omega}|^2$ is the shift in position. If $\rho = 0$, the center occurs at $\vec{x} = 0$.

The response of a center-surround cell to sinusoids

- We assume that the neuron is a center-surround cell and its receptive field is a Laplacian of a Gaussian $w_{LOG}(\vec{x})$.
- The predicted response is:

$$\int d\vec{x} w_{LOG}(\vec{x}) A \cos(\vec{\omega} \cdot \vec{x} + \rho) = A(\cos \rho)(\vec{\omega} \cdot \vec{\omega}) \exp\{-(\sigma^2 \vec{\omega} \cdot \vec{\omega})/2\}.$$

- We deduce three properties:
  1. The response is biggest if the center of the sinusoid is aligned to the center of the cell, i.e., $\rho = 0$, falling to zero at $\rho = \pi/2$
  2. The cell responds best to frequencies with $|\vec{\omega} \cdot \vec{\omega}| = 2\sigma^{-2}$ (by maximizing the response with respect to $|\vec{\omega}|$)
  3. The cell is insensitive to the orientation of the stimuli.
- We can characterize a neuron by measuring its firing rate when it is stimulated with sinusoids. We can use these properties to determine if it is center-surround or not, and if it is, to estimate its parameter σ^2.

Simple cell receptive fields in V1

- The receptive field properties of *simple cells* in V1 were studied by Hubel and Wiesel (1962, 1968) who showed that many cells were *tuned* to the orientation of edges and to the size of bars of light.
- They also showed that these cells were spatially organized with hypercolumns and retinotopic organization. Further electrophysiological studies by Roner and Pollen (1981) and Jones and Palmer (1987) showed that the receptive field properties of these cells could be approximately modelled by *Gabor filters* (Daugman, 1985), which are the product of Gaussians and sinusoids. Derivative of Gaussian filters give an alternative model (Young et al., 2001).
- It was also reported that the receptive fields occur in quadrature pairs (Pollen & Roner, 1981), so that neighboring cells are 90 degrees out of phase (e.g., a cosine Gabor is paired with a sine Gabor).

Gabor filters

- Gabor functions are the product of a Gaussian
  $G(\vec{x}; \vec{0}, \Sigma) = \frac{1}{2\pi|\Sigma|} \exp\{-(1/2)\vec{x}^T \Sigma^{-1} \vec{x}\}$

  with covariance Σ times a sinusoid:
  $\exp\{i\vec{\omega} \cdot \vec{x}\} = \cos \vec{\omega} \cdot \vec{x} + i \sin \vec{\omega} \cdot \vec{x}$.
- This gives two basic types of Gabors:
  1. cosine-Gabors
     $$G_{\cos}(\vec{x}) = G(\vec{x}; \vec{0}, \Sigma) \cos \vec{\omega} \cdot \vec{x}$$
  2. sine-Gabors
     $$G_{sin}(\vec{x}) = G(\vec{x}; \vec{0}, \Sigma) \sin \vec{\omega} \cdot \vec{x}.$$
- These form a *quadrature pair*, because sin(.) and cos(.) are 90 degrees out of phase.

Properties of Gabor filters

- Gabor filters give a good trade-off between *localization* in position and in frequency.
- The Gaussian has good localization in position, in the sense that its response is very small if $|\vec{x}| > 2\sigma$. The sinusoid has perfect localization in frequency (due to the orthogonality of sinusoids) but is unable to localize in position (because a sinusoid does not tend to zero for large $\vec{x}$).
- Gabor derived the Gabor function by optimizing a criterion that balanced optimality in frequency with optimality in position (Daugman, 1985).

Illustration of Gabor filters

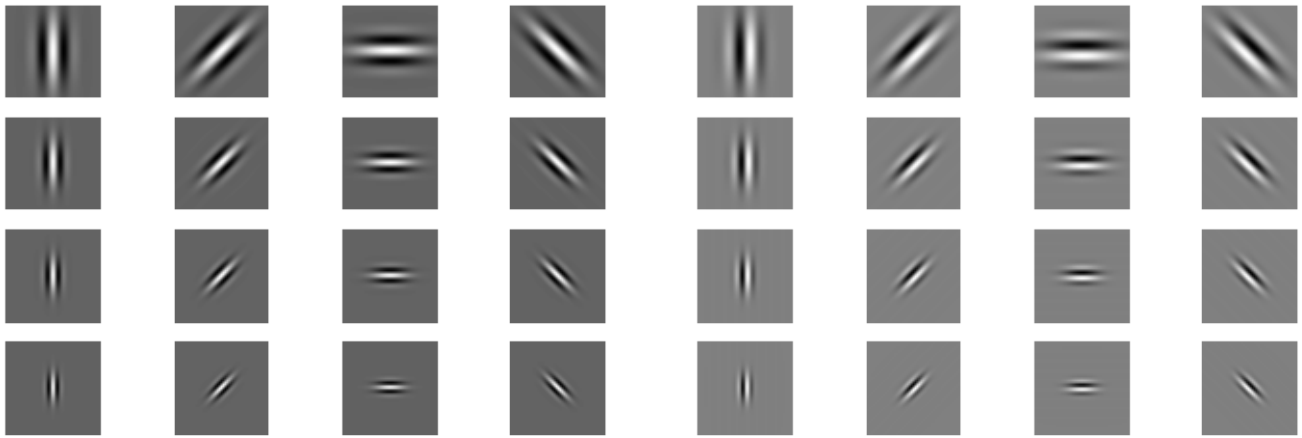


Figure 15 : A family of Gabor receptive fields. The panels show cosine-Gabors (left) and sine-Gabors (right) at different orientations (rows) and different scales (columns). Observe that the cosine-Gabors have biggest responses at their centers (because $\cos 0 = 1$), while the sine-Gabors have small responses there (because $\sin 0 = 0$).

The response of Gabor filters

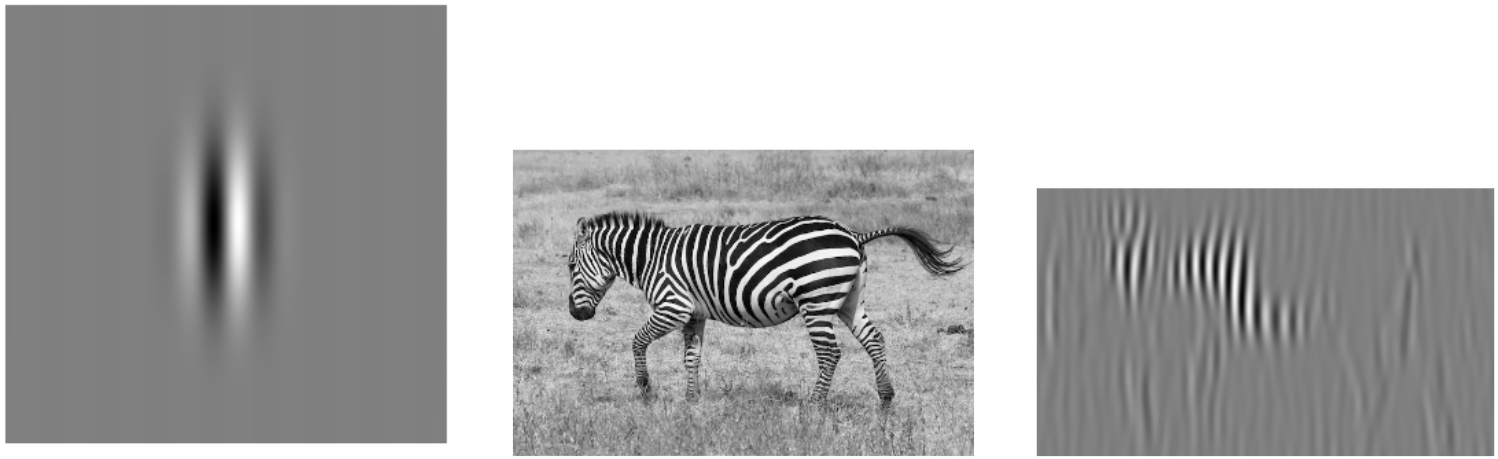


Figure 16 : A Gabor functions aligned to the vertical axis (left). The image of a zebra (center). The response of the vertical Gabor filter on the zebra image (right).

Modelling V1 neurons with Gabor filters

- It has been argued (Lee, 1996) that many simple cells in V1 could be modeled by a family of Gabor filters with specific relationships between the parameters of the Gaussian and the sinusoid, Σ and $\vec{\omega}$. The orientations of the Gaussian and the sinusoid are aligned, and the aspect ratio between the major and minor axes of the Gaussian is 4.
- In more detail, express the frequency of the sinusoid by $\vec{\omega} = \omega(\cos\theta, \sin\theta)$, where θ is its orientation and ω is the frequency. Then the covariance Σ of the Gaussian is proportional to $(1/4)(\cos\theta, \sin\theta)(\cos\theta, \sin\theta)^T + (-\sin\theta, \cos\theta)(-\sin\theta, \cos\theta)^T$ (T denotes vector transform).
- The sinusoid $\exp(i\vec{x} \cdot \vec{\omega})$ has its "propagating direction" along the shorter axis of the Gaussian, so the Gaussian smooths more in the direction perpendicular to the propagating direction, by a factor of $1/2 = \sqrt{1/4}$.

A family of Gabor filters

- This family is specified by:

$$\psi(\vec{x};\omega,\theta,K) = \frac{\omega^2}{4\pi K^2}$$
$$\times \exp\{-(\omega^2/8K^2)\{4(\vec{x}\cdot(\cos\theta,\sin\theta))^2 + (\vec{x}\cdot(-\sin\theta,\cos\theta))^2\}$$
$$\times \exp\{i\omega\vec{x}\cdot(\cos\theta,\sin\theta)\}\exp\{(K^2/2)\}\}.$$

- The variance is proportional to K^2. This is normalized so that $\int d\vec{x}\{\psi(\vec{x};\omega,\theta,K)\}^2 = 1$. $K \approx \pi$ for a frequency bandwidth of one octave, $K \approx 2.5$ for a frequency bandwidth of 1.5 octaves ("octaves" are the log ratio of the frequency – see Zhaoping, 2014).
- This family can also be scaled to give a form:

$$\psi_a(\vec{x};\omega,\theta,K) = \frac{1}{a}\psi_a(\vec{x}/a;\omega,\theta,K)$$

The tuning of Gabor filters (I)

- We study the tuning of Gabor cells by stimulating them with a family of stimuli of form $A\cos(\vec{\omega} \cdot \vec{x} + \rho)$ and varying $\vec{\omega}$ and ρ.
- We define $\omega_x = \vec{\omega} \cdot (\cos\theta, \sin\theta)$ and $\omega_y = \vec{\omega} \cdot (-\sin\theta, \cos\theta)$ to be the projections of the input sinusoid in the favored direction of the cell (i.e., $\vec{\omega}$) and in the orthogonal direction (i.e., $\omega_y = 0$ if the input sinusoid aligns perfectly with the orientation of the cell).

The tuning of Gabor filters (II)

- The responses of the cosine-Gabor G_{cos} and the sine-Gabor G_{sin} are given by:

$$\frac{A}{2}\cos\rho\exp\{-2K^2\omega_y^2/\omega^2\}$$
$$\times\{\exp\{-(K^2/2\omega^2)(\omega+\omega_x)^2\}+\exp\{-(K^2/2\omega^2)(\omega-\omega_x)^2\}\}\exp\{K^2/2\}$$

$$\frac{A}{2}\sin\rho\exp\{-2K^2\omega_y^2/\omega^2\}$$
$$\times\{\exp\{-(K^2/2\omega^2)(\omega+\omega_x)^2\}-\exp\{-(K^2/2\omega^2)(\omega-\omega_x)^2\}\}\exp\{K^2/2\}.$$

- The cosine-Gabor cell is tuned to $\rho = 0$, and the tuning falls off as $\cos\rho$. The cell also favors sinusoid stimuli, which are aligned to it (i.e., $\omega_y = 0$), and whose frequency $\omega_x = \pm\omega$.
- The sine-Gabor prefers stimuli with $\rho = \pi/2$ and has similar tuning to the frequency with $\omega_y = 0$ and $\omega_x = \pm\omega$.

Complex cells

- Complex cells are sensitive to orientation, but they are less sensitive than simple cells to the spatial position of the stimuli. This illustrates the standard theory of the ventral stream: visual processing proceeds up this stream using receptive fields, similar to simple and complex cells, which are increasingly tuned to more complex structures and are less sensitive to the precise positions of the stimuli.
- From this perspective, complex cells are the second stage after simple cells, forming a simple-complex cell module that gets repeated up the hierarchy.

Complex cells energy model

- We describe here the *energy model* where the complex cell receives input from two simple cells that are 90 degrees out of phase (i.e., cosine-Gabors and sine-Gabors). This is partly motivated by quadrature cells (Jones & Palmer, 1987) and partly by these cells being less sensitive than simple cells to the specific position of the stimuli.
- More precisely, the energy model of a complex cell gives response:

$$S(\vec{x}) = \{\psi_{\text{sin}} * I(\vec{x})\}^2 + \{\psi_{\text{cos}} * I(\vec{x})\}^2$$

where $*$ indicates convolution.

Tuning of complex cells

- We study the tuning of complex cells by measuring their response to sinusoid stimuli. The findings show that these cells are, like simple cells, tuned to orientation, frequency, and phase. But their tuning, particularly to phase, is less precise. Hence complex cells are less sensitive to the precise position of the stimuli. The response is given by:

$$\frac{A^2}{4}\exp\{K^2\}\exp\{-4K^2\omega_y^2/\omega^2\}$$
$$\{\exp\{-(K^2/\omega^2)(\omega+\omega_x)^2\}+\exp\{-(K^2/\omega^2)(\omega-\omega_x)^2\}$$
$$+2\cos 2\rho\exp\{-(K^2/\omega^2)(\omega+\omega_x)^2\}\exp\{-(K^2/\omega^2)(\omega-\omega_x)^2\}\}.$$

- Observe that the dependence on the phase ρ is much smaller (the dominant term in the second line is independent of ρ).

Illustration of complex cells

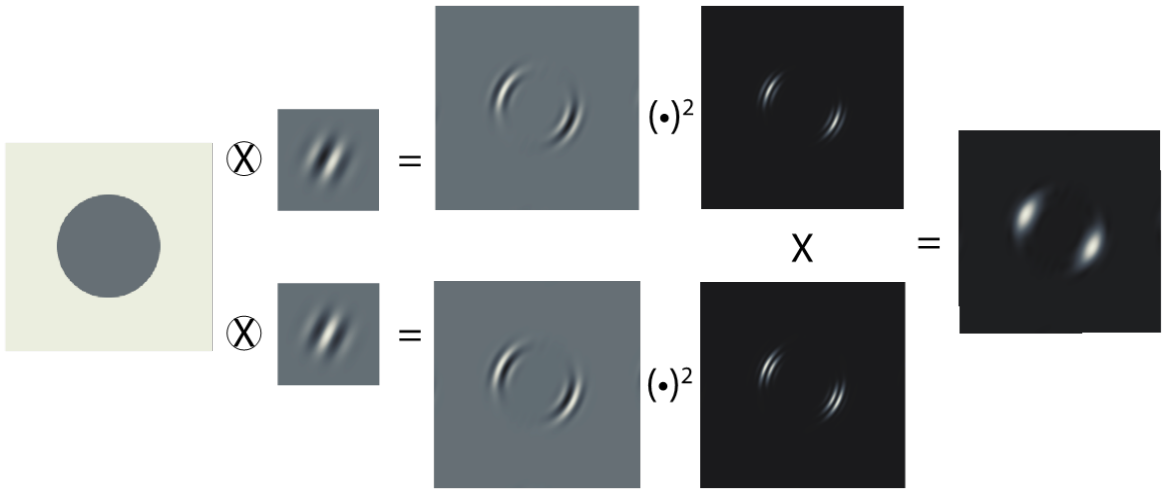


Figure 17 : A complex cell can be modeled as a quadrature pair of Gabor filters. The stimulus is a grey circle on a white background (far left). A quadrature pair of Gabor filters is applied to the stimulus, giving the largest responses when the orientation of the Gabors matches the orientation of the edge of the circle. The responses of the Gabors are squared and then summed to yield the final output (far right).

Complex cells: Complications

- In other models, complex cells are built from simple cells in alternative ways, but the complex cells retain their basic property of being tuned to orientation and frequency but being less sensitive to the position of the stimuli.
- But some researchers question whether complex cells receive input from single cells arguing that the computations could be done by nonlinear neurons that exploit the complexity of the dendritic tree (Mel et al., 1998).
- Other researchers argue (Mechler & Ringach, 2002) that there is no sharp dichotomy between simple and complex cells, but instead there is an continuum of cells with variable sensitivity to position.

Linear filtering and basis functions (I)

- We put these models into the context of the literature on linear filtering and Fourier analysis. This is an advanced section that gives greater understanding but is not required for a basic introduction.
- As discussed earlier, simple cell models apply *linear filters* to images and cells at different spatial locations, performing *convolution* $*$ by applying the same filter $\vec{w}$ across the image:

$$S(\vec{x}) = \vec{w} * I(\vec{x}) = \sum_{\vec{y}} w(\vec{x} - \vec{y}) I(\vec{y}).$$

- It is also convenient to approximate this (take the continuum limit) and express it as an integral:

$$S(\vec{x}) = \int_{\vec{y}} w(\vec{x} - \vec{y}) I(\vec{y}) d\vec{y}.$$

- This continuum limit is a good approximation, if the summation $\sum_{\vec{y}}$ is over a dense set of positions $\vec{y}$, and enables certain type of analysis (e.g., showing that a center-surround cell model sums, approximately, to zero).

Convolving by a Gaussian and derivatives of a Gaussian

Convolving an image by a linear filter produces an output image $S(\vec{x})$ whose form depends on the type of filter $\vec{w}$. For example, if $w(\vec{x})$ is a Gaussian function $G(\vec{x};\sigma) = \frac{1}{2\pi\sigma^2}\exp\{-(x_1^2 + x_2^2)/(2\sigma^2)\}$, then convolution effectively just smooths the image by taking a linear weighted average. If $\vec{w}$ is a derivative of the Gaussian in the x_1 direction, $w(\vec{x}) = \frac{d}{dx_1}G(\vec{x};\sigma)$, then this filter gives a large response to *edges*, positions $\vec{y}$ where the intensity $I(\vec{y})$ changes abruptly, and has small responses in places where the image intensity changes slowly.

Linear filtering, basis functions: Fourier analysis (I)

We can better understand images, and linear filtering, by using *functional analysis*. This states that an image, or any signal, can be expressed uniquely as a weighted sum of *basis functions*:

$$I(\vec{x}) = \sum_i \alpha_i b_i(\vec{x}), \tag{1}$$

where the $b_i(\vec{x})$ are basis functions and the $\{\alpha_i\}$ are *coefficients*. These basis functions are usually chosen to be *orthonormal*, so that $\sum_{\vec{x}} b_i(\vec{x}) b_j(\vec{x}) = \delta_{ij}$ ($= 1$ if $i = j$ and $= 0$ if $i \neq j$). If the basis functions are orthogonal, then the coefficients α can be obtained by:

$$\alpha_i = \sum_{\vec{x}} I(\vec{x}) b_i(\vec{x}). \tag{2}$$

Superposition

- The principle of superposition states that we can determine the output S as a weighted combination of the outputs of the basis functions:

$$S(\vec{x}) = \sum_i \alpha_i S_i(\vec{x}), \quad \text{where } S_i(\vec{x}) = \sum_{\vec{y}} w(\vec{x} - \vec{y}) b_i(\vec{y}). \tag{3}$$

- This implies that if we know the response $S_i(.)$ to each basis function $b_i(.)$, then we can predict the response to any input. This is an attractive property that if it holds, enables us to measure the receptive field of a linear neuron, or a thresholded linear neuron, from a limited set of stimuli.

Linear filtering, basis functions: Fourier analysis (II)

Fourier analysis deals with a special class of basis functions. These are sinusoids, i.e., of form $\sin \omega x, \cos \omega x$. The α's are the *fourier transform* of the image. If we restrict ourselves to an image defined on a lattice (i.e., so that x_1, x_2 each take a finite number of values, as on a digital camera), then this is the *discrete fourier transform*. But if we allow x_1, x_2 to take continuous values, then we get the fourier transform:

$$I(\vec{x}) = \frac{1}{2\pi} \int \hat{I}(\vec{\omega}) \exp\{-i\vec{\omega} \cdot \vec{x}\} d\vec{\omega} \tag{4}$$

$$\hat{I}(\vec{\omega}) = \frac{1}{2\pi} \int I(\vec{x}) \exp\{i\vec{\omega} \cdot \vec{x}\} d\vec{x} \tag{5}$$

Here $\exp\{i\vec{\omega} \cdot \vec{x}\} = \cos(\vec{\omega} \cdot \vec{x}) + i \sin(\vec{\omega} \cdot \vec{x})$. Note that if $I(.)$ is symmetric, $I(\vec{x}) = I(-\vec{x})$, then $\hat{I}(\vec{\omega})$ is also symmetric, $\hat{I}(-\vec{\omega}) = \hat{I}(\vec{\omega})$. Observe that equations (4, 5) correspond to equations (1, 2) for special choices of the basis functions (and changing from discrete to continuous $\vec{x}$).

Linear filtering, basis functions: Fourier analysis (III)

Fourier analysis is particularly important because it gives us a way to represent nonlocal structure of images in terms of *frequencies* ω. The high frequencies (large $|\vec{\omega}|$) represent image patterns that change rapidly, while the lower frequencies (small $|\vec{\omega}|$) represent slowly changing patterns. In particular, if an image pattern is *periodic*, like the stripes on a zebra, then it can be expressed in form:

$$I(\vec{x}) = \sum_n A_n \cos(2\pi n \vec{\omega}_0 \cdot \vec{x}),$$

where $\vec{\omega}_0$ is the basic frequency and n denotes integers. Then the Fourier transform is only nonzero at integer multiples of the basic frequency $\vec{\omega} = \vec{\omega}_0$. Hence periodic image patterns, such as *textures*, have very simple descriptions in Fourier space.

Linear filtering, basis functions: Fourier analysis (IV)

- If we blur the image, by convolving with a Gaussian $G(\vec{x};\sigma)$, to obtain $G * I(\vec{x})$, then the high frequencies of the image $\vec{I}$ will be smoothed out. By the *convolution theorem*, the Fourier transform of $G * I(\vec{x})$ is the product of the Fourier transforms of G and $\vec{I}$. The F.T. of a Gaussian is also a Gaussian $\exp\{-|\vec{\omega}|^2(\sigma^2/2)\}$. Hence we can express the convolved image as a weighted combination of sinusoids, where the high-frequency weights are decreased by $\exp\{-|\vec{\omega}|^2(\sigma^2/2)\}$:

$$\vec{I}(\vec{x}) = \frac{1}{2\pi}\int \hat{I}(\vec{\omega})\exp\{-i\vec{\omega}\cdot\vec{x}\}\exp\{-|\vec{\omega}|^2(\sigma^2/2)\}d\vec{\omega}.$$

- If we increase the blurring, by increasing the variance σ^2, we will make the high-frequency coefficients small. Blurring the image can be obtained by defocusing your eyes so that the image is seen out of focus. The receptive fields of cells occurs at a range of different scales, corresponding to convolving with Gaussians of different variances.

Linear filtering, basis functions: Fourier analysis (V)

The superposition principle, combined with the use of basis functions, shows that we can determine the receptive fields of linear neurons by stimulating them with sinusoids. Sinusoids can be used as basis functions, and superposition can be used to predict the response to stimuli that have not been seen yet (i.e., as superpositions of those stimuli to which the response is known). This, however, is rarely done.

Sparsity, matched filters, and natural images

- Next, we consider receptive field models from different perspectives. This includes the use of *sparsity* to suggest receptive field properties based on the statistics of natural images as well as the idea of *matched filters*, which revert to an older idea of receptive fields as feature detectors (Lettvin et al., 1959). Sparsity was proposed by Barlow (1961) as a general principle for modeling the brain based on the observation that typically only a small number of neurons are active. It was developed as a way to predict receptive field properties by Olshausen and Field (1996). It is natural to ask whether the receptive fields of cells encode basis functions that somehow capture the typical structure of images and represent it in a form that is suitable for later processing.
- Our starting point is the idea that images, and particularly local regions of images, can be represented as a linear combination of basis functions $I(\vec{x}) = \sum_i \alpha_i \vec{b}_i(\vec{x})$, as we saw in equation (1).

Sparsity and overcomplete bases

- Consider an image consisting of regions where the intensity varies spatially smoothly and regions where the intensity consists of a number of bright spots, or *impulses*. The smoothly varying regions of the image can be efficiently represented by Fourier analysis, in the sense that we can approximate the intensity by only a small number of weighted sinusoids
- By contrast, the impulses are much better represented in terms of a basis of impulse functions. It would be inefficient to represent them in terms of sinusoids.
- In short, different types of basis functions are suitable for different regions of the image.
- This suggests a strategy of seeking a representation in terms of an overcomplete set of basis functions, in this case sinusoids and impulse functions, and a criterion that selects an efficient representation so that only a small number of basis functions are activated for each image. This requirement is called ℓ_1 *sparsity*.

ℓ_1 sparsity

- More formally, we represent an image, or local image region, by:

$$I(\vec{x}) = \sum_{i=1}^{N} \alpha_i b_i(\vec{x}),$$

  where the $\{b_i\}$ are the basis functions and the $\{\alpha_i\}$ are the coefficients.
- The number N of bases is bigger than the dimension of the image, and hence the bases are *overcomplete*. Overcompleteness implies that there are many ways to represent the image in terms of these basis functions (by different choices of the α's) and that we need an additional criterion to select the α's. The ℓ_1 sparsity criterion proposes that we favor representations that make $\sum_{i=1}^{N} |\alpha_i|$ small, penalize the weights of the basis functions, and encourage most coefficients to be 0.

ℓ_1 sparsity criterion

- We represent an image $\vec{I}$ by the approximation $\sum_{i=1}^{N} \hat{\alpha}_i \vec{b}_i$, where the $\{\hat{\alpha}_i\}$ are chosen to minimize the function:

$$E(\alpha) = \sum_{\vec{x}} (I(\vec{x}) - \sum_{i=1}^{N} \alpha_i b_i(\vec{x}))^2 + \lambda \sum_{i=1}^{N} |\alpha_i|. \tag{6}$$

- The first term penalizes the error of the approximation, and the second term, whose strength is weighted by a parameter λ, penalizes the coefficients $\{\alpha_i\}$. The solution $\hat{\alpha} = \arg\min_\alpha E(\alpha)$ cannot be specified in closed form, but $E(\alpha)$ is a *convex* function of α, and efficient algorithms exist for minimizing it to estimate $\hat{\alpha}$. The results of these algorithms can, for example, decompose an image into a sum of sinusoids and a sum of impulse functions.

Sparsity and receptive fields (I)

These ideas give an alternative way to think about the receptive fields of cells in V1. First, observe that V1 has far more cells than the retina or the LGN, and so it has enough neural machinery to implement overcomplete bases. Second, overcomplete bases can be designed for specific image structures of interest (e.g., impulse functions or edges), which enables us to start interpreting the image instead of simply representing it. Third, it relates to the observation that cells in V1 fire *sparsely*, which suggests (Barlow, 1961) that they are tuned to specific stimuli and may relate to metabolic processes (firing a neuron takes energy, which needs to be replenished). Hence the idea that the visual cortex seeks to obtain sparse, and hence presumably more easily interpretable, representations has intuitive appeal.

Sparsity and receptive fields (II)

- Families of Gabor filters give an overcomplete basis, so they do not specify a unique representation of an image. These issues, and the relations of Gabors to wavelets, are discussed in more detail in (Lee, 1996).
- Sparsity can be used to derive the properties of receptive fields of cells in V1 from natural images (Olshausen, 1996), see figure (18)(Left). Hence instead of hypothesizing models of receptive fields (e.g., Gabor filters), we can try to predict these receptive fields from studying images. These predictions do give some justification for Gabor functions, but they also suggest other receptive field models that have been experimentally observed.

Learning receptive fields using ℓ_1 sparsity

- To learn the basis functions $\{\vec{b}_i\}$ from a set of natural images $\{\vec{I}^{\mu} : \mu \in \Lambda\}$, we extend equation (6) to obtain a criteria $E(b, \alpha)$ for fitting basis functions b and coefficients α to the set of images:

$$E(b, \alpha) = \sum_{\mu \in \Lambda} (I^{\mu}(\vec{x}) - \sum_{i=1}^{N} \alpha_i^{\mu} b_i(\vec{x}))^2 + \lambda \sum_{\mu \in \Lambda} \sum_{i=1}^{N} |\alpha_i|.$$

- We estimate the basis functions $\hat{b}$ and the coefficients $\hat{\alpha}$ by minimizing $E(b, \alpha)$ to obtain:

$$(\hat{b}, \hat{\alpha}) = \arg \min_{(b,\alpha)} E(b, \alpha).$$

- This criterion has been applied to natural images (where the $\vec{I}$ represent small image regions), and the resulting basis functions, see figure (18)(left), include filters that look like Gabor functions but they also include other types of filters observed in experiments (Olshausen, 1996).

Alternatives: ICA

- Other methods can predict receptive field properties from natural images using a similar image model, $I(\vec{x}) = \sum_{i=1}^{N} \alpha_i b_i(\vec{x})$, but imposing different assumptions on the form of the bases. In particular, independent component analysis (ICA) gives similar receptive field models (van Hateren & Ruderman, 1998). Hyvarinen (2010) explains this by showing that both types of models – L1 sparsity and ICA – encourage the α_i to be strongly peaked at 0, but can occasionally have large nonzero values.
- What happens if we remove the sparsity requirement and instead find the basis functions that minimize $\sum_{\mu \in \Lambda} (I^{\mu}(\vec{x}) - \sum_{i=1}^{N} \alpha_i^{\mu} b_i(\vec{x}))^2$? The basis functions will be the eigenvectors of the correlation matrix of the images and can be found by principal component analysis (PCA).

ℓ_1 sparsity figure

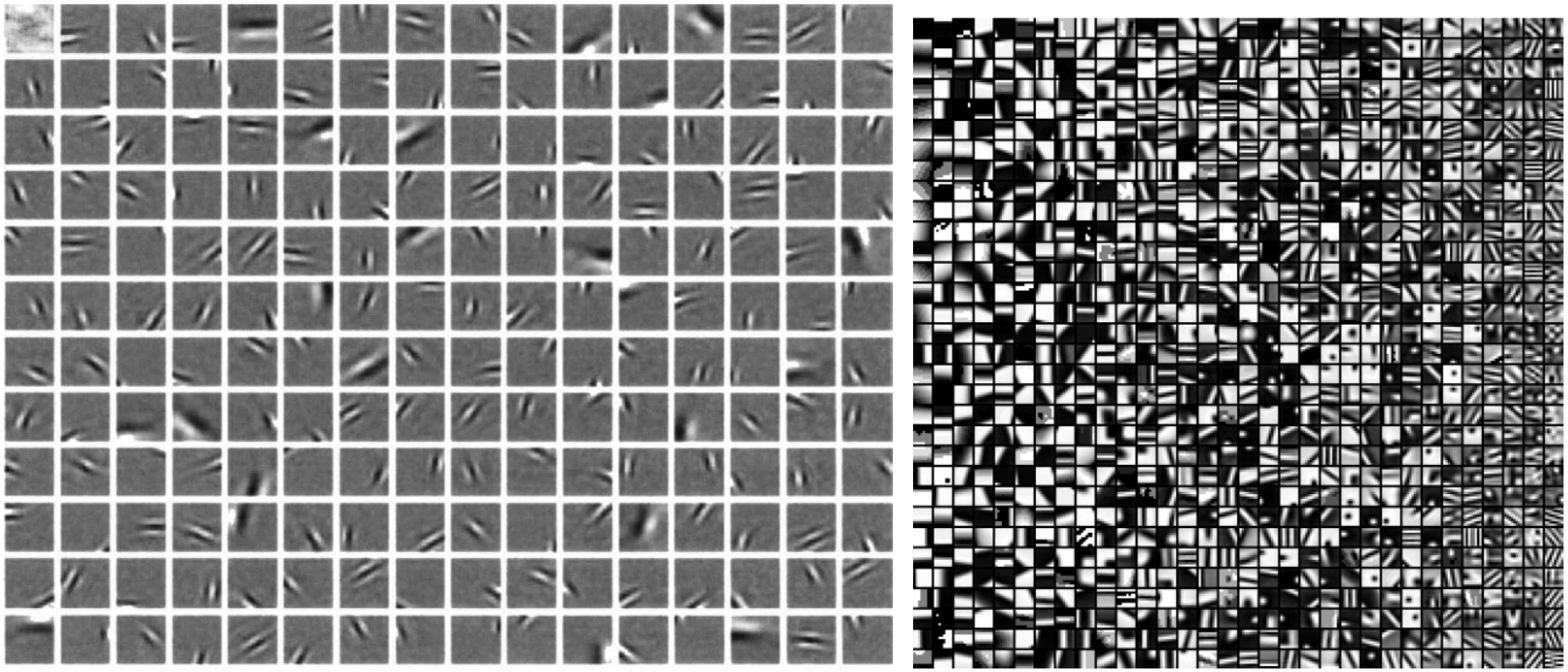


Figure 18 : Left: The receptive fields learned using sparsity (Olshausen, 1996). Right: receptive fields learned by matched filters.

Matched filter interpretation

- An alternative idea is that cells are feature detectors (Lettvin et al., 1959). This can be modelled by a set of matched filters, which is an extreme form of sparsity, because any image patch can be represented by a single filter.
- Examples of matched filters are shown in the previous figure (right).
- Suppose we have a filter $\vec{W}$ and an input image patch $\vec{I}_p$. We want to find the best fit of the filter to the image by allowing us to transform the filter by $\vec{W} \mapsto a\vec{W} + b\vec{e}$, where $\vec{e} = (1/\sqrt{N})(1, ..., 1)$. This corresponds to scaling the filter by a and adding a constant vector b. If $\vec{W}$ is a derivative filter, then by definition, $\vec{W} \cdot \vec{e} = 0$. We normalize $\vec{W}$ and $\vec{e}$ so that $\vec{W} \cdot \vec{W} = \vec{e} \cdot \vec{e} = 1$.

Matched filters

- The goal is to find the best scaling/contrast a and background b to minimize the match:

$$E(a, b) = |\vec{I}_p - a\vec{W} - b\vec{e}|^2.$$

- The solutions $\hat{a}, \hat{b}$ are given by (take derivatives of E with respect to a and b, recalling that $\vec{W}$ and $\vec{e}$ are normalized):

$$\hat{a} = \vec{W} \cdot I_p, \quad \hat{b} = \vec{e} \cdot \vec{I}_p.$$

- The filter response is just the best estimate of the contrast a. The estimate of the background b is just the mean value of the image. Finally, the energy $E(\hat{a}, \hat{b})$ is a measure of how well the filter "matches" the input image.

Matched filter dictionaries

The idea of a matched filter leads naturally to the idea of having a "dictionary" of filters $\{\vec{W}^{\mu} : \mu \in \Lambda\}$, in which different filters $\vec{W}^{\mu}$ are tuned to different types of image patches. In other words, the input image patch is encoded by the filter that best matches it. The dictionary of matched filters could be implemented by a set of cells (e.g., orientation columns). In this interpretation, the magnitude of the dot product $\vec{W} \cdot \vec{I}$ is less important than deciding which filter best matches the input $\vec{I}_p$. Matched filters can be thought of as an extreme case of sparsity. In the previous slides, an image was represented by a linear combination of basis functions whose weights were penalized by the ℓ_1, $\sum_i |\alpha_i|$. By comparison, matched filters represent an image by a single basis function. This gives an ever sparser representation of the image, but at the possible cost of a much larger image dictionary. Matched filters can be thought of as *feature detectors* because they respond only to very specific inputs.

Lecture 12.3

- This lecture describes how linear filters can be learned from images by unsupervised algorithms or estimated from neural data by regression. We describe how these receptive field models can be used for binocular stereo and for motion estimation.
- Then we introduce probabilities and decision theory. We motivate this by discussing how cues can be combined to detect edges in images.
- This lecture includes exercises involving interactive demos: (12.3.1) Oja's Rule and Principal Component Analysis, (12.3.2) Natural Image Statistics, and (12.3.3) Statistical Edge Detection.

Unsupervised learning of the receptive fields.

- We now introduce unsupervised neural network algorithms for learning receptive fields. This section is based on computational studies performed in the 1980's (Linsker, 1986a,b; Yuille et al., 1989), see (Zhaoping, 2014) for other references. These studies are based on modifications of the Hebb learning rule, which has some experimental support. Exercise demo (12.3.1) illustrates principal component analysis and Oja's rule (Oja, 1982).
- The basic findings are that center-surround, orientation selective, quadrature pairs, and disparity sensitive cells (precursors to cells that can estimate depth from binocular stereo) could all be obtained by variants of the same learning rule. Analysis of these findings suggest that this is partly due to the shift invariance of images.

Unsupervised learning by Hebb's rule (I)

- We first describe a simple unsupervised learning model for a single cell (Oja, 1982). The output $S(t)$ of the cell is a function of time t and is a weighted sum of the inputs $I_i(t)$, where the weights $\omega_i(t)$ are functions of time and are updated by Oja's rule (Oja, 1982):

$$S(t) = \sum_j w_j(t) I_j(t),$$

$$\frac{dw_i(t)}{dt} = S(t)\{I_i(t) - S(t)w_i(t)\}. \tag{7}$$

- The first term (Hebbs) increases the strength of a weight w_i if its input $I_i(t)$ is positively correlated with the output $S(t)$ (i.e., $< S(t)I_i(t) >> 0$), while the second term decreases the value of all weights by an amount proportional to their strength.
- This can be expressed as a single update equation:

$$\frac{dw_i(t)}{dt} = \sum_j w_j I_i(t) I_j(t) - \sum_{jk} w_i w_j w_k I_j(t) I_k(t). \tag{8}$$

Unsupervised learning by Hebb's rule: Analysis (I)

- Next we assume that the weights w_i change at a slower rate than the input images. This enables us to replace the terms $I_i(t)I_j(t)$ with their expectation $K_{ij} =< I_i(t)I_j(t) >$, which is the correlation function of the input. This gives:

$$\frac{dw_i(t)}{dt} = \sum_j w_j K_{ij} - \sum_{jk} w_i w_j w_k K_{jk}. \tag{9}$$

- The fixed points of this equation, the values of w such that $\frac{dw_i(t)}{dt} = 0$, can be shown to be eigenvectors of the correlation function K_{ij}. A slight modification gives an update rule (Yuille et al., 1989) that converges to the global minimum of the cost function:

$$E(\vec{w}) = -(1/2)\sum_{i,j} K_{ij} w_i w_j + (k/4)(\sum_i w_i^2)^2$$

Unsupervised learning by Hebb's rule: Analysis (II)

- The global minimum corresponds to the biggest eigenvalue of K_{ij}. If the correlation function K_{ij} decreases with distance, then the biggest eigenvalue is at frequency 0, so the cell is not tuned to any frequency. But if the correlation function has the shape of a Mexican hat, then the biggest eigenvalue has a nonzero frequency, which implies that the cell is orientated (Yuille et al., 1989).
- The correlation function of natural images does decrease spatially, but Linsker (1986a,b) showed that correlation functions similar to the Mexican hat arise if this learning procedure is applied to a sequence of layers.
- This analysis yields receptive fields that are sinusoids, and hence have no spatial fall-off, which is unrealistic. But receptive fields of neurons are limited by the geometrical positions of the dendrites. If these constraints are included, then the algorithms converge to receptive fields that are similar to Gabor functions.

How to empirically estimate receptive field models by regression.

- We can estimate the receptive field properties of cells from electrical recordings of neurons by estimating the best model using *regression*. This makes few assumptions about the form of the receptive field.
- Recall that the receptive field properties of neurons are traditionally found by probing their response to different perceptual dimensions, such as orientations and frequency. This gives a classification of the type of the receptive field but does not specify its receptive field weights $\vec{w}$ unless strong assumptions are made (e.g., that the receptive field is a Gabor function).

Estimating receptive field models by regression.

- The regression method makes few assumptions about the forms of the receptive field, but it does require more data. It requires a stimulus data set of $\mathcal{S} = \{(S^\mu, \vec{I}^\mu) : \mu = 1, ..., N\}$ of inputs $\vec{I}^\mu$ and outputs S^μ (e.g., the firing rates). It requires a model, such as $g(\vec{I} : \vec{w}) = \sigma(\vec{w} \cdot \vec{I})$, where $\sigma(.)$ is a sigmoid function.
- Regression requires minimizing a cost function like:

$$F(\vec{w}) = \frac{1}{|\mathcal{S}|} \sum_{\mu \in \mathcal{S}}^{N} E(S^\mu - g(I^\mu; \vec{w}))$$

  where $E(.)$ is a penalty function, e.g.,$(S^\mu - g(I^\mu; T))^2$.
- This minimization can be done by standard computer packages. It outputs an estimate of the model parameters $\vec{w}^*$ and an error measure $F(\vec{w}^*) = \frac{1}{|\mathcal{S}|} \sum_{\mu \in \mathcal{S}} E(S^\mu - g(I^\mu; \vec{w}^*))$.

Complications (I)

In practice, there are several complications. It is unrealistic to show the neuron all possible stimuli because there are so many possible image stimuli. Hence researchers have to choose a restricted set of stimuli. If neurons are linear, or a nonlinear function of a linear filter, then this should not matter because we can exploit the superposition principle and estimate the receptive field from a limited number of stimuli. But in reality, linearity is only an approximation, and in practice, the choice of stimuli can matter considerably. One concern is that the stimulus set does not contain the types of stimuli that the neuron is most sensitive to, in which case regression will output unreliable estimates. Also, if the linear assumption is only partially correct, then there is no guarantee that the receptive field learned on one set of stimuli will predict the behavior well on another set of stimuli.

Complications (II)

The complications are illustrated by recent findings (Talebi & Baker, 2012) that estimates of the receptive fields of neurons can depend heavily on the set of stimuli. The authors used three different stimulus sets: (1) white noise (WN), (2) oriented bars (B), and (3) natural images (NI). This gives three estimates for the receptive fields $\vec{w}_{WN}, \vec{w}_B, \vec{w}_{NI}$ by using stimulus sets $\mathcal{S}_{WN}, \mathcal{S}_B, \mathcal{S}_{NI}$. For each data set, they compute the prediction errors F_{WN}, F_B, F_{NI} which are the errors for that data set, e.g., $F_{WN}(\vec{w}^*_{WN}) = \frac{1}{|\mathcal{S}_{WN}|} \sum_{\mu \in \mathcal{S}_{WN}} E(S^\mu - g(I^\mu; \vec{w}^*_{WN}))$. These quantities show how well the models can fit each stimulus set. They can also enable us to study how well the estimated receptive field from one stimulus set can predict the other data sets. This involves computing quantities such as $F_{WN}(\vec{w}^*_B)$, $F_{WN}(\vec{w}^*_{NI})$, $F_B(\vec{w}^*_{WN})$, $F_{WN}(\vec{w}^*_{NI})$, $F_{NI}(\vec{w}^*_{WN})$, $F_{WN}(\vec{w}^*_B)$. They show that the receptive fields estimated on the natural image stimulus set were much better at predicting the responses on the other two stimulus sets.

Local models for binocular stereo (I)

- Linear filter models of receptive fields can also be used to perform local estimates of binocular stereo and motion. These models involve having filterbanks, or populations of filters, that are tuned to different properties of the stimuli, so that estimates of depth and motion can be extracted from the population (Zhaoping, 2014).
- Recall that we introduced binocular stereo earlier. Depth is estimated by triangulation provided we can solve the *correspondence problem* by finding which points in the left and right eyes correspond to the same point in three-dimensional space. This reduces to estimating the displacement, or *disparity*, between the images in the left and right eyes. In this section, we introduce the disparity energy model, which estimates disparity based on local properties of the image. Later we will discuss how nonlocal context can be used to improve disparity estimation.

Local models for binocular stereo (II)

- The disparity energy model is formulated using Gabor filters and has some claim to biological plausibility (Ohzawa et al., 1990; Qian, 1994). The model assumes that we have a large set of cells, receiving input from both images and tuned to different image frequencies and spatial phases.
- We give the presentation in one dimension, exploiting the epipolar line constraint. It assumes that the cell receives input from both left and right eyes with receptive fields $f_l(x) = \exp\{-x^2/(2\sigma^2)\}\cos(\omega x + \rho_l)$ and $f_r(x) = \exp\{-x^2/(2\sigma^2)\}\cos(\omega x + \rho_r)$. These are Gabors where the Gaussian has variance σ^2, tuned to frequency ω and with phases ρ_l, ρ_r. The linear response is:

$$r = \int dx\{f_l(x)I_l(x) + f_r(x)I_r(x)\} \tag{10}$$

- This filter is tuned to spatial frequency ω. The filter is most sensitive to the image component at this frequency. Hence we can represent the image (approximately) by $I(\vec{x}) = \rho\cos(\omega x + \theta)$.

Local models for binocular stereo (III)

- Suppose that the right image is a displaced version of the left image $I_r(x) = I_l(x + D(x))$, where $D(x)$ is the disparity. We assume that the disparity varies slowly so that we can approximate it locally as a constant D (over the size of the Gaussian, 2σ). To analyze the model, ignore the Gaussian when calculating r. This gives:

$$r_1 = \rho\{\cos(\theta - \rho_l) + \cos(\theta - \rho_r - \omega D)\} \tag{11}$$

  which can be re-expressed (using trigonometry identities):

$$r_1 = 2\rho \cos(\theta - \frac{\rho_l + \rho_r}{2} - \frac{\omega D}{2}) \cos(\frac{\rho_l - \rho_r}{2} - \omega \frac{D}{2}) \tag{12}$$

- The response of the cell depends on the disparity but also on image properties (e.g., image phase θ). So we need a population of cells to detect disparity.

Lcoal models for binocular stereo (IV)

- To see this, suppose that we consider quadrature pairs of the two cells tuned to the same ω. Where one cell has phases ρ_l, ρ_r, and the other has phases ρ'_l, ρ'_r, where $(\rho_l - \rho_r) = (\rho'_l - \rho'_r)$ and $\rho'_l + \rho'_r = \rho_l + \rho_r + \frac{\pi}{2}$. Then the second cell has response
$r_2 = 2\rho\cos(\theta - \frac{\rho_l+\rho_r}{2} - \frac{\omega D}{2})\cos(\frac{\rho_l-\rho_r}{2} - \omega\frac{D}{2}) =$
$2\rho\sin(\theta - \frac{\rho_l+\rho_r}{2} - \frac{\omega D}{2})\cos(\frac{\rho_l-\rho_r}{2})$. Hence if we square and add the responses of the two cells, we obtain:

$$r_1^2 + r_2^2 = \cos^2(\frac{\rho_l - \rho_r}{2} - \omega\frac{D}{2}) \tag{13}$$

- This response depends only on the disparity D and the image frequency ω. It takes largest values when $\rho_l - \rho_r = \omega D$. Hence we can estimate D from a population of quadrature cells tuned to different phases ρ_l, ρ_r and frequencies ω.

Local models for binocular stereo (V)

- A neural network for estimating D using a population of neurons consists of two steps. In step (1) we define a set of disparity cells tuned to disparities $\{D_i : i = 1, ..., N\}$. The disparity cell tuned to disparity D_i receives input $\cos^2(\frac{\rho_l - \rho_r}{2} - \omega\frac{D_i}{2})$ from each quadrature pair (ρ_l, ρ_r, ω) and sums these inputs together to compute a vote $v(D_i)$:

$$v(D_i) = \sum_{\rho_l, \rho_r, \omega} \cos^2(\frac{\rho_l - \rho_r}{2} - \omega\frac{D_i}{2}). \tag{14}$$

  Step (2) uses a winner-take-all network (Maass, 2000) to compute the disparity with the biggest vote by solving $\hat{D} = \arg\max_{i=1,...,N} v(D_i)$, so that $v(\hat{D}) \geq v(D_i)$ for $i = 1, ..., N$.

- There is plenty of evidence that the brain represents information by neural populations (Georgopoulos et al., 1983; McIlwain, 1991). There have also been several theoretical studies of how populations of neurons could encode knowledge and perform computations (Pouget et al., 2003; Ma et al., 2006).

Illustration of local model of binocular stereo

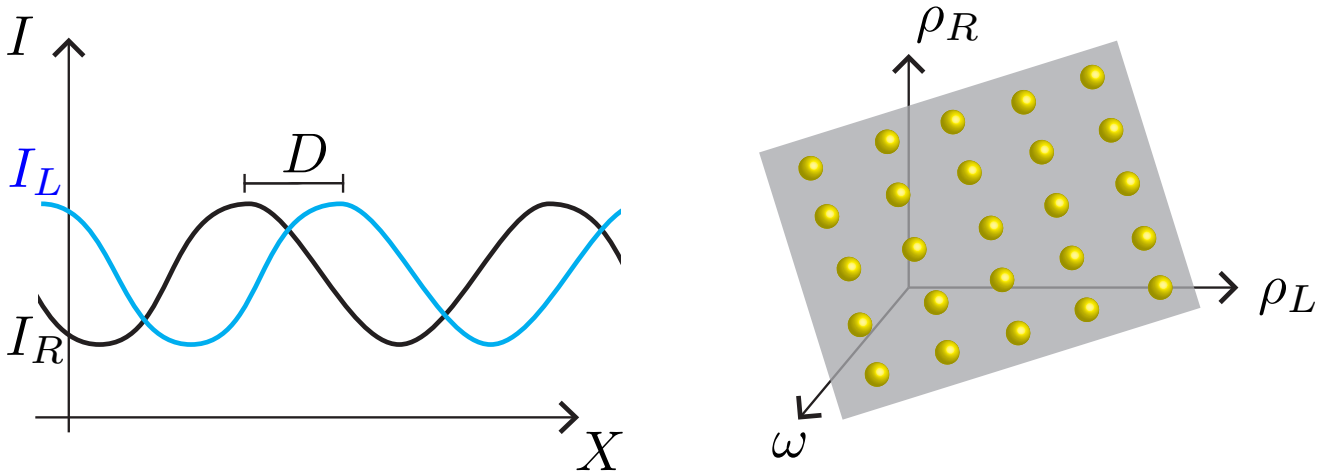


Figure 19 : Left: The disparity D between the images in the two eyes corresponds to a change of phase if we approximate the intensities by sinusoids. Right: The local disparity D is encoded by the feature response of cells tuned to frequencies that obey $\rho_l - \rho_r = \omega D$.

Motion measurement: Spatio-temporal filters.

We now discuss how related models can be used to estimate motion for sequences of images. Spatiotemporal filters are biologically plausible ways to measure motion that agree with properties of cells in the visual cortex. The standard model suggests two classes of cells: the first comprises spatiotemporal filters that are sensitive to the directions of motion, while the second class combines outputs of these filters to estimate the motion itself (Adelson & Bergen, 1985; Grzywacz & Yuille, 1990; Schrater et al., 2000).

Motion measurement: Figures

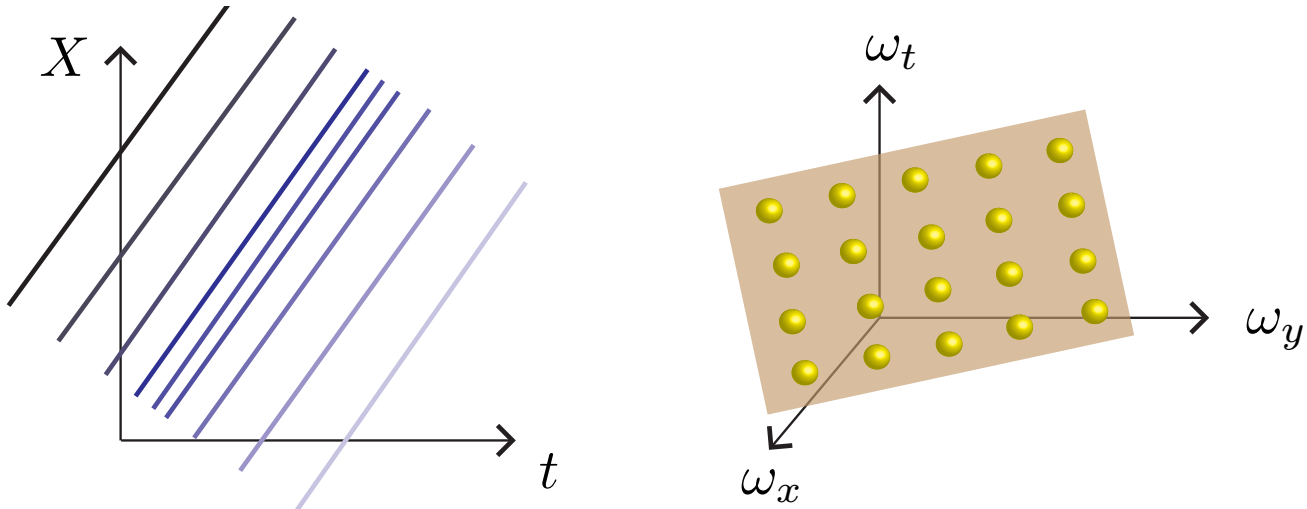


Figure 20 : Left: This figure shows the space-time illustration of a signal traveling with constant velocity $I(X,t) = F(X - tv)$. This means that the intensity $I(X,t)$ is constant on the lines $X - tv = constant$. Right: A stimuli moving with velocity $\vec{v}$ will activate spatiotemporal filters $\vec{\omega}, \omega_t$, which lie on the plane $\vec{v} \cdot \vec{\omega} + \omega_t = 0$. Hence the velocity can be estimated from the population of activity of the filters.

Motion measurement (I)

- Measuring the motion velocity assumes that locally, the intensity can be modeled as a linear translating pattern:

$$I(\vec{x}, t) = F(\vec{x} - \vec{v}t). \tag{15}$$

- Differentiating with respect to $\vec{x}$ and t (using $\vec{\nabla} I = \vec{\nabla} F$ and $\frac{\partial I}{\partial t} = -\vec{v} \cdot \vec{\nabla} F$) gives the *optical flow equation*:

$$\vec{v} \cdot \vec{\nabla} I + \frac{\partial I}{\partial t} = 0. \tag{16}$$

- This enables us to estimate one component of the motion $\vec{v}$ but suffers from the aperture problem and so is ambiguous.

Motion measurement (II)

- The ambiguity can be resolved by a population of filters $\{G^{\mu}(\vec{x}, t) : \mu = 1, ..., M\}$ indexed by μ (e.g., Gaussians). These filters introduce local context:

$$G^{\mu} * I(\vec{x}, t) = \int G^{\mu}(\vec{x} - \vec{y}, t - s) I(\vec{y}, s) ds d\vec{y}. \tag{17}$$

  Each filter gives a constraint on the velocity:

$$\vec{v} \cdot \vec{\nabla} G^{\mu} * I + \frac{\partial G^{\mu} * I}{\partial t} = 0. \tag{18}$$

- We get an estimate of the velocity $\vec{v}$ by minimizing the cost:

$$E(\vec{v}) = \sum_{\mu=1}^{M} (\vec{v} \cdot \vec{\nabla} G^{\mu} * I + \frac{\partial G^{\mu} * I}{\partial t})^2.$$

- This minimization can be done using a similar neural network to that used for estimating disparity for stereo in the previous section.

Motion measurement (III)

We have a set of cells tuned to different velocities $\{\vec{v}_i : i = 1, ..., N\}$. The cell tuned to velocity $\vec{v}_i$ receives input $(\vec{v} \cdot \vec{\nabla} G^\mu * I + \frac{\partial G^\mu * I}{\partial t})^2$ from each filter μ and sums the responses to obtain $E(\vec{v}_i)$. Then we use a variant of winner-take-all to compute $\hat{\vec{v}} = \arg\min_{i=1,...,N} E(\vec{v}_i)$.

Motion measurement: The need for spatial and temporal context

This approach assumes that there is enough local information to resolve the motion ambiguity which may not be the case. For example, for the stimuli in figure 12.7 in the chapter, we can only locally estimate one component of the motion because of the aperture problem. To resolve this ambiguity, we need to use more spatial or temporal context.

Motion measurement: Spatial and temporal context (I)

An alternative way to analyze this problem is by applying Ffourier analysis to equation (15):

$$\hat{I}(\vec{\omega}, \omega_t) = \frac{1}{2\pi} \int \int \int \exp\{i(\vec{\omega} \cdot \vec{x} + \omega_t t)\} I(\vec{x}, t) d\vec{x} dt$$

$$\hat{I}(\vec{\omega}, \omega_t) = \frac{1}{2\pi} \int \int \int \exp\{i(\vec{v} \cdot \vec{x} + \omega_t) t\} \exp\{i\vec{\omega} \cdot (\vec{x} - \vec{v}t)\} F(\vec{x} - \vec{v}t) d\vec{x} dt$$

$$\hat{I}(\vec{\omega}, \omega_t) = \frac{1}{2\pi} \int \exp\{i(\vec{v} \cdot \vec{\omega} + \omega_t) t\} dt \int \int \exp\{i\vec{\omega} \cdot \vec{\bar{x}}\} F(\vec{\bar{x}}) d\vec{\bar{x}}$$

$$\hat{I}(\vec{\omega}, \omega_t) = \delta(\vec{v} \cdot \vec{\omega} + \omega_t) \hat{F}(\vec{\omega})$$

where $\vec{\bar{x}} = \vec{x} - \vec{v}t$ is a change of variables in the integral.

Motion measurement: Spatial and temporal context (II)

This shows that if we have filters $\exp\{i(\vec{x}\vec{\omega} + \omega_t t)\}$ tuned to spatiotemporal frequencies $\vec{\omega}, \omega_t$, then the only filters that respond are those whose frequencies obey the equation $\vec{v} \cdot \vec{\omega} + \omega_t = 0$ and hence lie on a plane in frequency space. Hence we can determine $\vec{v}$ from a population of filters by observing which filters are activated and finding the best fit plane.

Motion measurement – Non-Fourier

- In practice, we cannot use filters tuned to frequency because these are not bounded in space and time. But it can be shown (Grzywacz & Yuille, 1990) that if the filters are spatio-temporal Gabors, then the most active filters are those whose spatiotemporal tuning is centered on the plane $\vec{v} \cdot \vec{\omega} + \omega_t = 0$. Hence the plane in frequency space can be estimated from a population of spatiotemporal filters and the velocity locally estimated.
- This gives a two stage model of motion estimation, in which the first population of neurons (i.e., filters) are each sensitive to the spatiotemporal frequency of the input image but not directly to the motion. The second population of neurons extract the motion information from the first population, and hence these neurons are tuned directly to motion. This is consistent with experimental findings (Adelson & Bergen, 1985),(Grzywacz & Yuille, 1990), (Schrater et al., 2000). Similar models arise in related work on the fly and beetle visual systems (Hassenstein & Reichardt. 1956; Borst & Euler, 2011).

Probabilities and decision theory

- We now describe a principled approach for combining the response of many features/filters to perform tasks like stereo or motion estimation. This approach is based on decision theory. This section also illustrates the importance of knowing whether filter responses, hence visual cues for the task, are dependent or independent.
- We introduce the probabilities of filter responses by describing a classical experimental finding about natural image statistics. Intuitively, the intensities of neighboring pixels tend to be similar. This intuition can be captured by taking derivative filters of the image, i.e., $\frac{dI}{dx}$ or $\frac{d^2I}{dx^2}$, and plotting their probability distribution, or histogram. Surprisingly these probability distributions are very similar from image to image (Simoncelli & Olshausen, 2001).

Edge detectors/ texture detectors and decisions

- Consider the tasks of deciding whether an *image patch* at position x contains an *edge* by which we mean the boundary of an object or a strong texture boundary (e.g., the writing on a t-shirt). The previous section showed that some Gabor filters are tuned (i.e., respond strongly) to edges at specific orientations. But such filters will also respond to other stimuli, such as texture patterns, so how can we decide if their response is due to an edge? The simplest way is to *threshold* the response so that an edge, at a specific orientation, is signalled if the filter response is larger than a certain threshold value. But what should that threshold be? How do we do a trade-off to balance *false negative* errors, when we fail to detect a true edge in the image, with *false positive* errors when we incorrectly label a pixel as an edge?
- Also each filter in a filterbank contains some evidence about the presence of an edge, so how can we combine that evidence in an optimal manner? How can we formulate the intuition that some filters give *independent* evidence, while others do not?

Decision theory

Decision theory gives a way to address these issues. The theory was developed as a way to make decisions in the presence of uncertainty. In this section we develop the key ideas of decision theory by addressing the specific task of edge detection. In the next section we give a more general treatment. We only treat the case when we are detecting edges based on local evidence in the image. Later we extend to when we can use nonlocal, or contextual, information.

Filters

To start with, we consider the evidence for the presence of an edge using a single filter $f(.)$ only. We assume we have a benchmarked data set so that at each pixel, we have intensity $I(x)$ and a variable $y(x) \in \{\pm 1\}$ (where $y = 1$ indicates an edge, and $y = -1$ does the opposite). We apply the filter to the image to get a set of filter responses $f(I(x))$. If the filter is tuned to edges, then the response $f(I(x))$ is likely to be higher if an edge is present than if not. This requires selecting a filter $f(x)$, such as the modulus of the gradient of intensity $|\vec{\nabla} I(x)| = \sqrt{\frac{dI}{dx}^2 + \frac{dI}{dy}^2}$ (since $|\vec{\nabla} I(x)|$ is likely to be large on edges and small off edges).

Conditional probability distributions

- To quantify this, we use the benchmarked data set to learn *conditional probability distributions* for the filter response $f(I)$ conditioned on whether there is an edge or not:

$$P(f(I)|y=1), P(f(I)|y=-1).$$

- Each distribution is estimated by computing the *histogram* of the filter response by counting the number of times the response occurs within one of N equally spaced bins and normalizing by dividing by the total number of responses. The histograms for $P(f(I)|y=1)$ and $P(f(I)|y=-1)$ are computed from the filter responses on the points labeled as edges $\{f(I(x)) : y(x)=1\}$ and not-edges $\{f(I(x)) : y(x)=1\}$ respectively. Typical conditional distributions are shown in the figure on the next slide.

Figure for conditional distributions

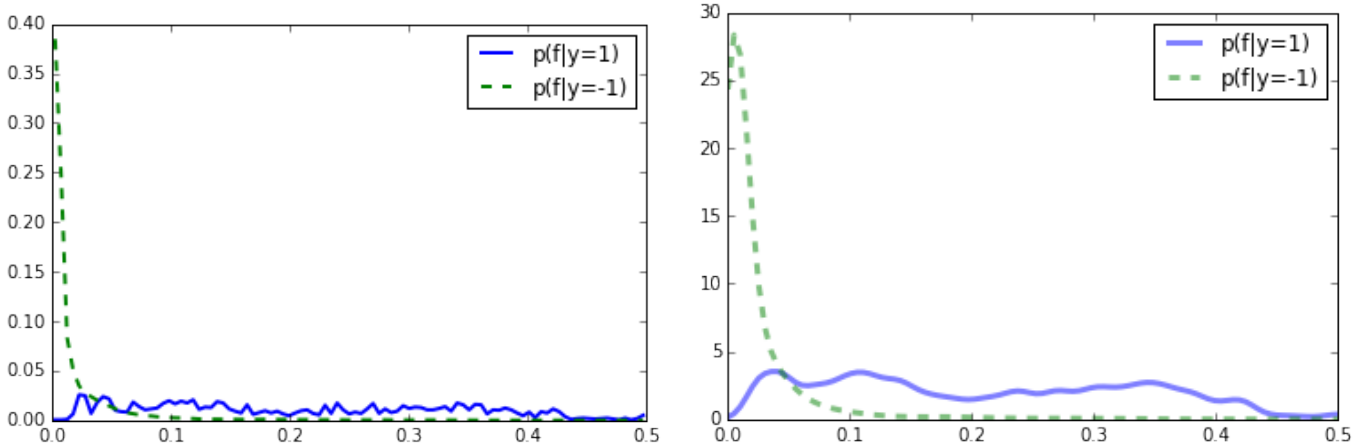


Figure 21 : The probability of filter responses conditioned on whether the filter is *on* or *off* an edge – $P(f|y=1), P(f|y=-1)$, where $f(x) = |\vec{\nabla} I(x)|$. Left: The probability distributions learned from a data set of images. Right: The smoothed distributions after fitting the data to a parametric model.

Statistical edge detection

We can now perform edge detection on an image. At each pixel x we compute $f(I(x))$ and calculate the conditional distributions $P(f(I(x))|y = 1)$ and $P(f(I(x))|y = -1)$. These distributions give local evidence for the presence of edges at each pixel. Note, however, that local evidence for edges is often highly ambiguous. Spatial context can supply additional information to help improve edge detection, and so can high-level knowledge (e.g., by recognizing the objects in the image).

Log-likelihood ratio

The log-likelihood ratio $\log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)}$ gives evidence for the presence of an edge in image I at position x. This ratio takes large positive values if $P(f(I(x))|y=1) > P(f(I(x))|y=-1)$ (i.e., if the probability of the filter response is higher given an edge is present) and large negative values if $P(f(I(x))|y=-1) > P(f(I(x))|y=1)$. So a natural decision criterion is to decide that an edge is present if the log-likelihood ratio is greater than zero and that otherwise there is no edge. This can be formulated as a *decision rule* $\alpha(x)$:

$$\alpha(x) = 1, \text{ if } \log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)} > 0, \quad \alpha(x) = -1, \text{ if } \log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)} < 0.$$

This can expressed, more compactly, as

$$\alpha(x) = \arg \max_{y \in \{\pm 1\}} y \log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)}.$$

Statistical edge detection figure

Figure 22 : The input image and its groundtruth edges (far left and left). The derivative dI/dx of the image in the x direction (center). The probabilities of the local filter responses $P(\vec{f}(I(x))|y = 1)$ (right) and $P(\vec{f}(I(x))|y = -1)$ (far right) have their biggest responses on the boundaries and off the boundaries, respectively, hence the log-likelihood ratio $\log \frac{P(\vec{f}(I(x))|y=1)}{P(\vec{f}(I(x))|y=-1)}$ gives evidence for the presence of edges.

Ambiguities in edge detection

- Note that this rule gives perfect results (i.e., is 100% correct) if the two distributions do not overlap, i.e., if $P(f(I(x))|y=1)P(f(I(x))|y=-1)=0$ for all I. In this case it is impossible to confuse the filter responses to the different types of stimuli. But this situation is very unlikely to happen. Now consider a more general *log-likelihood ratio test* that depends on a threshold T; this gives a rule:

$$\alpha_T(x) = \arg \max_{y \in \{\pm 1\}} y\{\log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)} - T\}.$$

- By varying T we get different types of mistakes. We can distinguish between the *false positives*, which are non-edge stimuli that the decision rule mistakenly decides are edges, and *false negatives*, which are edge stimuli that are mistakenly classified as not being edges. Increasing the threshold T reduces the number of false positives but at the cost of increasing the number of false negatives, while decreasing T has the opposite effect.

Ambiguity of edges figure

Figure 23 : The local ambiguity of edges. An observer has no difficulty in detecting all of the boundary of the horse if the full image is available (left). But it is much more difficult to detect edges locally (other panels).

Decision theory and trade-offs

Making a decision requires a trade-off between these two types of errors. Bayes decision theory says this trade-off should depend on two issues: first, the *prior* probability that the image patch is an edge. Statistically most image patches do not contain edges, so we would get a small number of total errors (false positives and false negatives) by simply deciding that every image patch is non-edge. This would encourage us to increase the threshold T (to $-\infty$ so that every image patch would be classified as non-edge). Second, we need to consider the *loss* if we make a mistake. If our goal is to detect edges, then we may be willing to tolerate many false positives provided we keep the number of false negatives small. This means we choose a decision rule, by reducing the threshold T, so that we detect all the real edges but also output "false edges," which we hope to remove later by using contextual cues. Later we show how this approach can be justified using the framework of decision theory.

Combining multiple cues for edge detection

- Now we consider combining several different filters $\{f_i(.)|i = 1, ..., M\}$ to detect an edge by estimating the *joint* response of all the filters $P(f_1, f_2, \ldots|y) = P(\{f_i(I(x))\}|y)$ *conditioned* on whether the image patch I at x is an edge $y = 1$ or not an edge $y = -1$. This leads to a decision rule:

$$\alpha_T(I(x)) = \arg \max_{y \in \{\pm 1\}} y\{\log \frac{P(\{f_i(I(x))\}|y = 1)}{P(\{f_i(I(x))\}|y = -1)} - T\}.$$

- This approach has two related drawbacks. First, the joint distributions require a large amount of data to learn, particularly if we represent the distributions by histograms. Second, the joint distributions are "black boxes" and give no insight into how the decision is made. So it is better to try to get a deeper understanding of how the different filters contribute to making this decision by studying whether they are *statistically independent*.

Combining cues with statistical independence

- The response of the filters is statistically independent if:

$$P(\{f_i(I(x))\}|y) = \prod_i P(f_i(I(x))|y) \text{ for each y}$$

- This implies that the distributions $P(f_i(I(x))|y)$ can be learned separately (which decreases the amount of data) and also implies that the log-likelihood test can be expressed in the following form:

$$\alpha_T(x) = \arg \max_{y \in \{\pm 1\}} y\{\sum_i \log \frac{P(f_i(I(x))|y=1)}{P(f_i(I(x))|y=-1)} - T\}$$

- Hence the decision rule corresponds to summing the evidence (the log-likelihood ratio) for all the filters to determine whether the sum is above or below the threshold T. This means that each filter gives a "vote," which can be positive or negative, and the decision is based on the sum of these votes. This process is very simple, so it is easy to see which filters are responsible for the decision.

Combining cues with conditional independence

- But very few filters are statistically independent. For example, the response of each filter will depend on the total brightness of the image patch, so all of them will respond more to a "strong" edge than to a "weak" edge. This suggests a weaker independence condition known as *conditional independence*. Suppose we add an additional filter $f_0(I(x))$ that, for example, measures the overall brightness. Then it is possible that the other filters are statistically independent conditioned on the value of $f_0(I(x))$:

$$P(\{f_i(I(x))\}, f_0(I(x))|y) = P(f_0(I(x))|y) \prod_i P(f_i(I(x))|f_0(I(x)), y)$$

- This requires only representing (learning) the distributions $P(f_i(I(x))|f_0(I(x)), y)$ and $P(f_0(I(x))|y)$.

Combining cues with conditional independence

- It also leads to a simple decision rule:

$$\alpha_T(x) = \arg \max_{y \in \{\pm 1\}} y\{\log \frac{P(f_0(I(x))|y=1)}{P(f_0(I(x))|y=-1)} + \sum_i \log \frac{P(f_i(I(x))|f_0(I(x)), y=1)}{P(f_i(I(x))|f_0(I(x)), y=-1)} - T\} \quad (19)$$

- It has been argued (Ramachandra & Mel, 2013) that methods of this type can be implemented by neurons and may be responsible for edge detection. Note that the arguments here are general and do not depend on the type of filters $f_i(.)$ or whether they are linear or nonlinear. It has, for example, been suggested that edge detection is performed using the energy model of complex cells (Morrone & Burr, 1988).

Classification for other visual tasks

- The same approach can be applied to other visual tasks. For example, consider using local filter responses to classify whether the local image patch at x is "sky," "vegetation," "water," "road," or "other"). We denote these by a variable $y \in \mathcal{Y}$ (e.g., where $\mathcal{Y} = \{"sky", "vegetation", "water", "road", or "other"\}$. We choose a set of filters $\{f_i(I(x))\}$ that are sensitive to texture and color properties of image patches. Then, as before, we learn distributions $P(\{f_i(I(x))\}|y)$ for $y \in \mathcal{Y}$. We select a decision rule of form:

$$\alpha(I(x)) = \arg\max_{y \in \mathcal{Y}} P(\{f_i(I(x))\}|y) T_y,$$

  where T_y is a set of thresholds (which can be derived from decision theory).
- Experiments on images show that this method can locally estimate the local image class with reasonable error rates for these types of classes (Konishi & Yuille, 2000) and computer vision researchers have improved these kinds of results using more sophisticated filters.

Classifying other image classes

Figure 24 : Classifying local image patches. The images show the groundtruth (Mottaghi et al., 2014). Certain classes – sky, grass, water – can be classified approximately from small image patches.

Context

We stress that the theories described in this section model edge detection *without context*. There are two types of context we will consider in this lecture. The first uses spatial context and is low or mid level since it depends only on *generic* properties of images and surfaces. It exploits the idea that edges in natural images are often geometrically regular and co-linear. The second type of context, is high level and object specific. For example, if we detect a face in an image, then our knowledge about faces enables us to detect the boundaries of a face better than if we relied only on local edge cues. This second type of context is out of the scope of this chapter but is briefly discussed at the end of these lectures.

Lecture 12.4

- This lecture discusses Bayes decision theory.
- We describe divisive normalization and context.
- Then we discuss the role of context and specify stochastic and deterministic models of groups of neurons.
- This lecture includes two excercises involving interactive demos: (12.4.1) Gibbs sampling, and (12.4.2) Mean Field Theory.

Bayes decision theory and ideal observers

- Bayes decision theory is a framework for making optimal decisions in the presence of uncertainty. We represent the input by $x \in \mathcal{X}$ and the output by $y \in \mathcal{Y}$ (e.g., for edge detection x is the filter response $f(I)$, and $y \in \{\pm 1\}$ indicates if an edge is present or not).
- We assume that there is a probability distribution $P(x, y)$ that generates the input and output. This can be expressed in terms of a *prior* $P(y)$ and a *likelihood* $P(x|y)$ by the identity $P(x, y) = P(x|y)P(y)$. A decision rule is expressed as $\hat{y} = \alpha(x)$. We specify a *loss function* $L(\alpha(x); y)$, which is the cost of making decision $\alpha(x)$ if the real decision should be y.
- The *risk* is specified by $R(\alpha) = \sum_{x,y} P(x, y)L(\alpha(x), y)$. The *Bayes rule* is $\hat{\alpha} = \arg\min_\alpha R(\alpha)$. The *Bayes risk* is $\min_\alpha R(\alpha) = R(\hat{\alpha})$.

Bayes rule (I)

The Bayes rule is the best decision rule you can make (subject to this criterion) and the Bayes risk is the best performance. Hence Bayes decision theory can specify the optimal way to estimate y from input x. There are several important special cases. If the loss function penalizes all errors by the same amount, i.e., $L(\alpha(x), y) = K_1$ if $\alpha(x) \neq y$ and $L(\alpha(x), y) = K_2$ if $\alpha(x) = y$ (with $K_1 > K_2$), then the Bayes rule corresponds to the *maximum a posteriori* estimator $\alpha(x) = \arg\max P(y|x)$, where $P(y|x) = \frac{P(x|y)P(y)}{P(x)}$ is the *posterior* distribution of y conditioned on y. If, in addition, the prior is a uniform distribution, i.e., $P(y) = \textit{constant}$, then Bayes rule reduces to the *maximum likelihood* estimate $\alpha(x) = \arg\max P(x|y)$.

Bayes rule (II)

For binary decision problems $y \in \{\pm 1\}$, the loss function is usually chosen to pay no penalty if the correct decision is made (i.e., $\alpha(x) = y$ -1) but has a penalty F_p for *false positives*, where $y = -1$ but $\alpha(x) = 1$, and F_n for *false negatives*, where $y = 1$ but $\alpha(x) = -$ (it is assumed here that the *target* is $y = 1$ and the *distracter* is $y = -1$, so a false positive occurs if we decide that a distracter is a target, and a false negative if we decide that a target is a distracter). It follows that we can express the Bayes rule in terms of a log-likelihood ratio test $\log \frac{P(x|y=1)}{P(x|y=-1)} > T$, where T depends on the prior $p(y)$ and the loss function $L(\alpha(x), y)$.

Bayes rule (III)

- More specifically, the Bayes risk is $R(\alpha) = \sum_x p(x) \sum_y L(\alpha(x), y) p(Y|x)$. Then we divide the data (x, y) into four sets: (1) the *true positives* $\{(x, y) : \text{ s.t. } \alpha(x) = y = 1\}$; (2) the *true negatives* $\{(x, y) : \text{ s.t. } \alpha(x) = y = -1\}$; (3) the *false positives* $\{(x, y) : \text{ s.t. } \alpha(x) = 1, y = -1\}$; and (4) the *false negatives* $\{(x, y) : \text{ s.t. } \alpha(x) == -1, y = 1\}$. These four cases correspond to loss function values $L(\alpha(x) = 1, y = 1) = T_p$, $L(\alpha(x) = -1, y = -1) = T_n$, $L(\alpha(x) = 1, y = -1) = F_p$, $L(\alpha(x) = -1, y = 1) = F_n$ respectively. Then the decision rule $\alpha_T(.)$ reduces to:

$$\log \frac{P(x|y=1)}{P(x|y=-1)} > \log \frac{T_n - F_p}{T_p - F_n} + \log \frac{P(y=-1)}{P(y=1)}.$$

- The intuition is that the evidence in the log-likelihood must be bigger than our prior biases while taking into account the penalties paid for different types of mistakes.

Bayes rule (IV)

The results in the previous section on edge detection and texture classification can be derived from decision theory. The priors $P(y)$ specify the probability that an image patch contains an edge (empirically $P(y = 1) \approx 0.05$ and $P(y = -1) \approx 0.95$). The loss function should be chosen to specify the cost of making different types of mistakes. For texture classification, the variable y takes values in a set $\mathcal{Y}$, which is called a multiclass decision. The same theory applies to tasks for which we need to make a set of related but nonlocal decisions.

Signal detection theory (I)

We now show that an important special case of *signal detection theory* (Green & Swets, 1966) – often used as a framework to model how humans make decisions when performing visual, auditory, and other tasks – can be obtained as a special case of Bayes decision theory. We consider the two class case, where $y \in \{\pm 1\}$, and suppose that the likelihood functions are specified by Gaussian distributions, $P(x|y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\{-(x-\mu_y)^2/(2\sigma_y^2)\}$, which differ by their means (μ_1, μ_{-1}) and their variances $(\sigma_1^2, \sigma_{-1}^2)$. The Bayes rule can be expressed in terms of the log-likelihood ratio test:

$$\hat{\alpha}(x) = \arg\max_y y\{-(x-\mu_1)^2/(2\sigma_1^2) - \log\sigma_1 + (x-\mu_{-1})^2/(2\sigma_{-1}^2) + \log\sigma_2 - T\}.$$

Signal detection theory (II)

- This decision rule requires determining whether the data point x is above or below a quadratic polynomial curve in x. In the special case when the standard deviations are identical $\sigma_1^2 = \sigma_2^2$ (so we drop the subscripts $_{1,-1}$), the decision is based only on whether the data point x satisfies:

$$2x(\mu_1 - \mu_{-1}) + (\mu_1^2 - \mu_2^2) < 2T\sigma^2$$

- This special case, with $\sigma_1^2 = \sigma_{-1}^2$, is much studied in signal detection theory (Green & Swets, 1966). It means that the decision is based on a single function $d' = \frac{\mu_1 - \mu_{-1}}{\sigma}$. This quantity is used to quantify human performance for psychophysical tasks.

Ideal observer (I)

This motivates the idea of an *ideal observer*. An observer like this has optimal performance which requires exploiting the statistical properties of the distribution $P(x, y)$ of the data. A classic example of ideal observer theory shows that under certain conditions, photoreceptors in the retina are almost *optimal* at detecting the photons that reach them (Barlow, 1962; Pelli, 1990). This takes into account the probability of the photoreceptors *firing* x if it receives a photon, $P(x|y = 1)$, and the probability that the photoreceptor fires spontaneously, $P(x|y = -1)$.

Ideal observer (II)

Ideal observers can also be defined for other vision tasks (Tjan et al., 1995; Gold et al,. 2012; Trenti et al., 2010; Geisler, 2011). The difficulty, however, is judging whether humans are adapted to doing the task. It is possible to define ideal observers when human performance is much worse than the ideal observers (Watson et al., 1983). Why can this happen? The task may provide information for which humans are not adapted (e.g., visual inspection of circuit boards to find deficits). Also, the ideal observers know the distributions $P(x, y)$ that, for synthetic stimuli, are those chosen by the scientist performing the experiment and may have little similarity to the natural statistics of stimuli of the world, which human vision has probably adapted to.

Receiver operating characteristic curve

- Another important concept is the receiver operating characteristic (ROC) curve. This allows us to study decisions when we do not want to restrict ourselves to specific priors and loss functions. Instead, we plot the *true positive rate* as a function of the *false positive rate* by allowing the decision threshold T to vary. For each value T of the threshold, we have a decision rule $\alpha_T(.)$, which results in a fraction of *true positives* $\sum_{x:\alpha_T(x)=1} P(x|y=1)$ and *false positives* $\sum_{x:\alpha_T(x)=1} P(x|y=-1)$. This gives a single point on the ROC curve. We plot the curve by allowing T to vary. Observe that for very large T (as $T \mapsto \infty$), the true positive and false positive rates will tend to 0. While as T gets very small ($T \mapsto -\infty$), both rates will tend to 1. Hence the ROC illustrates the trade-off between the two rates.
- Bayes decision theory can be extended in a straightforward manner if the output y takes multiple values. In particular, it applies when we have a set of decision variables defined on each lattice site of an image.

Divisive normalization

- An important example is the use of probabilistic models (Wainwright & Simoncelli, 2000) to account for divisive normalization. This is a mechanism whereby cells mutually inhibit one another, effectively normalizing their responses with respect to stimulus inputs. Originally developed to explain nonlinear responses to contrast in V1 (Heeger, 1992), divisive normalization has been proposed as a basic cortical computation that underlies various effects of context, as well as higher-level processe,s such as attention (Carandini & Heeger, 2011) .
- The probabilistic approach gives a theoretical justification for divisive normalization in V1. The main idea is that filters with similar preferences for orientation representing nearby spatial locations in a scene have striking statistical dependencies, which can be removed by divisive normalization. Specifically, if we plot the statistics of two linear filters f_c, f_s (center and surround), then the magnitudes of f_c, f_s are coordinated in a straightforward way, which has a characteristic shape of a bow tie.

Modeling divisive normalization using hidden variables

This can be modeled by assuming there are hidden variables ν that affect both responses and hence induces correlation between the responses. For example, ν could represent the local average image intensity, which could affect the response of both filters, but after the filter response, it could be made independent by conditioning on the average intensity. Suppose ν has a prior distribution $P(\nu) = \nu \exp\{-\nu^2/2\}$ for $\nu \geq 0$. We have a pair of filters $\{I_i : i = 1, 2\}$ that are related to Gaussian models $\{g_1 : i = 1, 2\}$. Then we can model the activation of the set of filter responses:

$$P(I_1, I_2) = \int d\nu P(\nu) \prod_{i=1}^{2} P(I_i|\nu, g_i))P(g_i), \tag{20}$$

where $P(I_i|\nu, g_i) = \delta(I_i - \nu g_i)$. In this model the filter responses are generated by independent processes, g_1, g_2, but then are multiplied by the common factor ν. This is illustrated in the next figure.

Figure for divisive normalization model

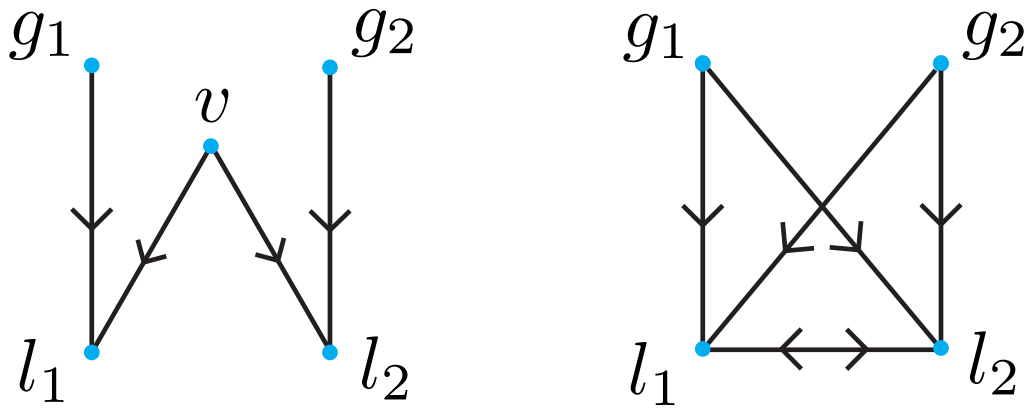


Figure 25 : Left: The graphical structure of the divisive normalization model. The filter responses l_1, l_2 are generated from stimuli g_1, g_2 and by the common factor ν. The distributions of l_1, l_2 are factorized if we condition on ν. Right: But if we integrate out ν, then almost all the variables become dependent, as reflected by the complexity of the graph structure.

Divisive normalization model

- In particular, for each filter we can compute $P(g_i|I_1, I_2)$. After some algebra, this is computed to be:

$$P(g_1|I_1, I_2) = \frac{g_1^{-1} \exp\{-\frac{g_1^2 I^2}{2\sigma^2 I_1^2} - \frac{I_1^2}{2g_1^2}\}}{B(0, I/\sigma)}, \tag{21}$$

  where $I = \sqrt{I_1^2 + I_2^2}$, and $B(.,.)$ is a Bessel function. To get intuition, note that $g_1 = I_1/\nu$ and $g_1 = I_1/\nu$. So if ν is small, then $|I_1|$ and $|I_2|$ are likely to be small together, while if ν is large, then $|I_1|$ and $|I_2|$ are both likely to be large.

- Assume that the goal of a model unit is to estimate the g_i from the observed filter responses $\{I_i : i = 1, 2\}$, which gives the nonlinear response of the cell. It follows, from analysis above, that

$$E(g_1|I_1, I_2) \propto \text{sign}\{I_1\}\sqrt{|I_1|}\sqrt{\frac{|I_1|}{\sqrt{I_1^2 + I_2^2 + k}}}. \tag{22}$$

  The $\sqrt{I_1^2 + I_2^2 + k}$ term sets the gain and performs the divisive normalization.

Application to the tilt illusion

- The model has also been applied to explain the classic tilt illusion in perception (Schwartz et al., 2009; Qiu et al., 2013). In the "simultaneous" tilt illusion, a set of vertically oriented lines appears to tilt right when surrounded by an annulus of lines tilted left–an effect called "repulsion." But for large differences between the center orientation and the surround (tilted left), the center vertical lines can appear to tilt left–an effect called "attraction." In the model, the population of neurons responding to the surround tilted lines contributes to divisive normalizing of the neurons responding to the center stimulus. This results in a change of their neural tuning curves, which, together with the degree of coupling between center and surrounds, accounts for repulsion and attraction.
- The suppressive effect of surround contrast on a central region is an example of local spatial context.

Context and spatial interactions between neurons

- There is considerable evidence that low-level vision involves long-range spatial interaction,s so that human perception of local regions of an image can be strongly influenced by their spatial context. Psychophysicists have discovered many perceptual phenomena demonstrating spatial interactions.
- For example, local image regions that differ from their neighbors tend to "pop out" and attract attention, while, conversely, similar image features that form spatially smooth structures tend to get "grouped" together to form a coherent percept, see chapter figure 12.26 (left panel). Image properties such as color tend to spread out, or fill in regions, until they hit a boundary (Grossberg & Mingolla, 1985; Sasaki et al., 2004) as shown in chapter figure 12.26 (right panel).

Context and spatial Interactions between neurons

- In general, there is a tendency for low-level vision to group similar image features and make breaks at places where the features change significantly. These perceptual phenomena are not surprising from a theoretical perspective since they correspond to low-level visual tasks, such as segmentation and the detection of salient features. Segmenting an image into different regions is one of the first stages of object recognition (in the ventral stream) and a precursor to estimating the three-dimensional structure of objects, or surfaces, in order to grasp them or avoid them (dorsal stream).
- Detection of salient features has many uses, including bottom-up attention (Itti & Koch, 2001). It has been suggested that many of these processes are performed in V1 (Zhaoping, 2014), although this involves possibly feedback and interactions between V1 and V2 (Shushruth et al., 2013).

Context figures

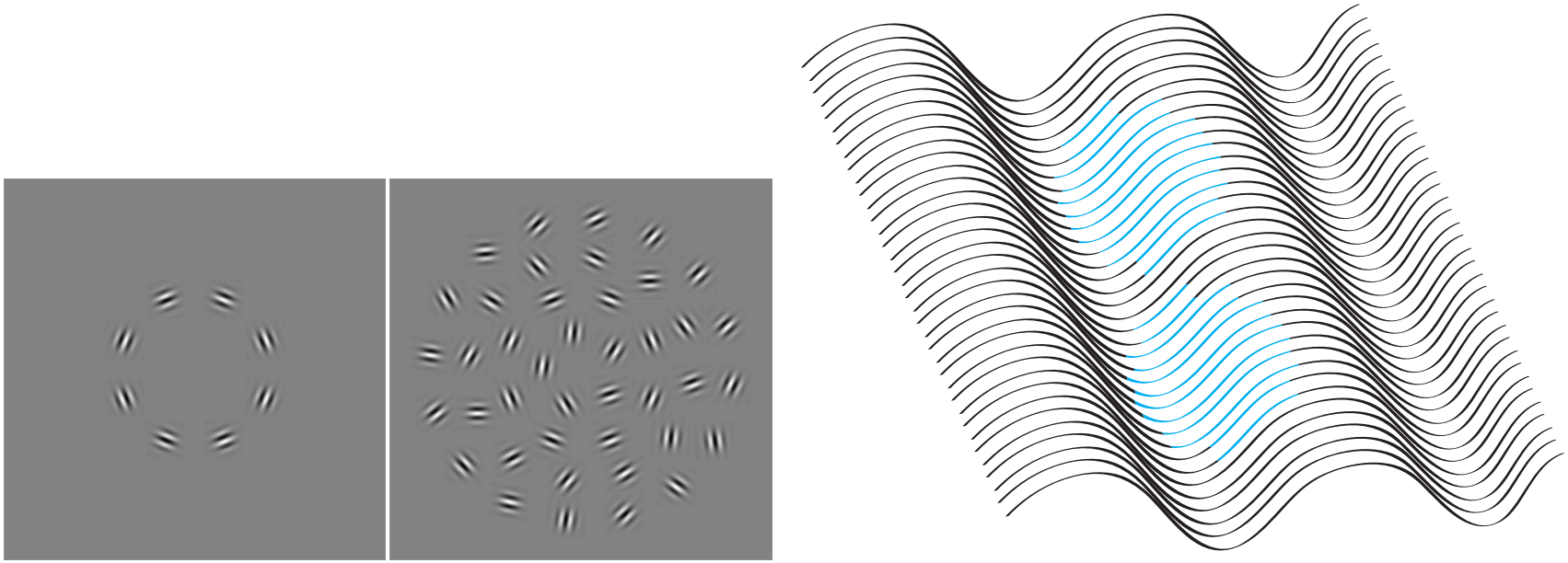


Figure 26 : Left: Association fields. The circular alignment of Gabor patches (left) make it easier to see the circular form in the presence of clutter (right). Right: The neon color illusion. A bluish color appears to fill in the white regions between the blue lines, creating the appearance of blue transparent disks.

d

Context electrophysiology

The psychophysical and theoretical studies discussed so far are supported by single-electrode studies (Lamme, 1995; Lee & Yuille, 2006), which show that the activities of neurons on monkey area V1 appear to involve spatial interactions with other neurons. When monkeys are shown stimuli consisting of a textured square surrounded by a background with a different texture, their responses over the first 60 msec are similar to those predicted by classic models (e.g., previous sections), but their later activity spreads in from the boundaries, roughly similar to predictions of computational models (Yuille, 2006). There is also a considerable literature on the related topic of *nonclassical receptive fields* (Kapadia et al., 2000).

Neural network models

This section discusses neural network models that address these phenomena. Although the models capture the essence of the phenomena, they are simplifications in three respects. First, they use simple models of neurons, and it is currently not possible to compare them directly to real neural circuits. Second, these models are formulated in terms of lateral, or horizontal, connections. Third, the performance of these models on natural images is significantly worse than a human's. There are more advanced computer vision models, built on similar principles, whose performance starts to approach human vision (unless high-level cues are present, which humans can exploit).

Probability distributions on graphs

We formulate these models in terms of probability distributions defined over graphs, where the nodes of the graph represent neurons. This differs from some of the standard "neural network" models for these types of phenomena, see (Grossberg & Mingolla, 1985). but our approach has several advantages. First, this enables us to use a coherent framework that unifies the models in this section with those we will discuss in later sections. Second, it puts the models in a form that can be directly related to a class of computer vision models. Third, this probabilistic formulation is of increasing use in models of artificial intelligence, cognitive science, and the machine learning and statistical techniques used to analyze experimental neuroscience data. Fourth, it is possible to derive many of these neural network models as approximations to the probability models.

Probabilistic models of neurons

- We first introduce probabilistic models of neurons and show how our previous linear filter models can be derived as approximations. Next we introduce neural network models and show their relationship to probability models. Then we use this material to derive some specific models for a range of visual tasks.

Single neurons: Probabilistic model and integrate and fire (I)

We have described neurons as linear filters and briefly mentioned thresholds and nonlinearities. In this section, we provide a more realistic model of a *stochastic neuron*, where the neuron has a probability of firing an action potential. We will show how linear filters, thresholds, and nonlinearities can be obtained as approximations to this stochastic model. This stochastic model is, in turn, an approximation, and we refer to the literature for more realistic models, such as assuming that the probability of firing is specified by a Poisson process (Rieke et al., 1997). For simplicity, we restrict ourselves to the simpler stochastic *integrate-and-fire* model, which is easier to analyze and to relate to computational models.

Single neurons: Probabilistic model and integrate and fire (II)

In the integrate-and-fire model, a neuron i receives input I_j at each dendrite j. These inputs are weighted by the synaptic strengths w_{ij} and sent along the dendrites to the soma. At the soma, these weighted inputs are summed linearly to yield $\sum_j w_{ij} I_j$. The probability of firing $s_i = 1$, or not firing $s_i = 0$, is given by:

$$P(s_i|\vec{I}) = \frac{\exp\{s_i(\sum_j w_{ij} I_j - T_i)\}}{1 + \exp\{\sum_j w_{ij} I_j - T_i\}}, \tag{23}$$

where T_i is a threshold.

Relations to the stochastic model (I)

- To relate this stochastic model to our earlier linear models, we calculate the probability that the neuron will fire. This is given by a sigmoid function:

$$\sum_{s_i=0}^{1} s_i P(s_i|\vec{I}) = \frac{1}{1+\exp\{\sum_j w_{ij} I_j - T_i\}} = \sigma(\sum_j w_{ij} I_j - T_i). \tag{24}$$

- Observe that this is also the *expected firing rate* $\sum_{s_i=0,1} s_i P(s_i|\vec{I})$ because

$$\sum_{s_i=0,1} s_i P(s_i|\vec{I}) = P(s_i = 1|\vec{I}) = \sigma(\sum_j w_{ij} I_j - T_i). \tag{25}$$

Relations to the stochastic model (II)

- By computing the expected firing rate, we obtain a deterministic approximation to a stochastic neuron. This is a sigmoid function of a linear weighted sum of the input (minus a threshold).
- The sigmoid function is approximately linear for small inputs, saturates at value 1 for large positive inputs, and suppresses large negative inputs to 0. Hence there is a linear regime where the probability of firing is $\sum_j w_{ij} I_j - T_i$. This enables us to recover the linear models used in the previous section as an approximation.
- Next we modify the model so that it deals with nonlinear image features. This allows us to relate it to the types of computational models described in the previous section and will enable us to construct richer models of this type that can deal with spatial context.

Enhancing the model to allow complex input

- Consider detecting if there is an edge at pixel x. Formulate the problem as Bayes estimation with conditional distributions $P(f(I(x))|s)$ and priors $P(s)$ for $s \in \{0, 1\}$. The posterior distribution $P(s|f(I(x)))$ can be expressed in the form:

$$P(s|f(I(x))) = \frac{1}{Z} \exp\{s(\log \frac{P(f(I(x))|s=1)}{P(f(I(x))|s=0)} + \log \frac{P(s=1)}{P(s=0)})\},$$

  where Z is a normalization constant (chosen so that $\sum_{s=0}^{1} P(s|f(I(x))) = 1$).
- This shows that the posterior distribution for the presence of an edge can be expressed in the same form. The only difference is that the input is a nonlinear function of the image instead of the image itself.
- This claim can be justified by expressing $P(f(I(x))|s) = \{P(f(I(x))|s=1)\}^s \{P(f(I(x))|s=0)\}^{1-s}$, $P(s) = \{P(s=1)\}^s \{P(s=0)\}^{1-s}$, then substituting these into the posterior $P(s|f(I(x))) = P(f(I(x))|s)P(s)/P(f(I(x)))$.

Probability models with context

- Now apply the model to foreground/background classification and modify it to include spatial context. Intuitively, neighboring pixels in the image are likely to be either all background or all foreground. This is a form of prior knowledge that can be learned by analyzing natural images.
- We specify neurons by spatial position $\vec{x}$ instead of index i. As above, we have distributions $P(f(I(\vec{x}))|s)$ for the features $f(I(\vec{x}))$ at position $\vec{x}$ conditioned on whether this is part of the foreground object $s(\vec{x}) = 1$, or not, $s(\vec{x}) = 0$. We use the notation $\vec{S}$ to be the set of the states of all neurons $\{s(\vec{x})\}$. We also specify a prior distribution:

$$P(\vec{S}) = \frac{1}{Z} \exp\{-\gamma \sum_{\vec{x}} \sum_{\vec{y} \in N(\vec{x})} \{s(\vec{x}) - s(\vec{y})\}^2\},$$

where γ is a constant. This prior uses a neighborhood $N(\vec{x})$, which specifies those spatial positions that directly interact with $\vec{x}$ in the model. In graphical terms, the positions $\vec{x}$ are the nodes $\mathcal{V}$ of a graph $\mathcal{G}$, and the edges $\mathcal{E}$ specify which nodes are connected.

Markov structure (I)

- Formally, the edges of the graph define the *Markov structure* of the probability distribution $P(\vec{S})$. It can be shown that the conditional distribution of the state $s(\vec{x})$ at one position depends *only* on the states of positions in its neighborhood $N(\vec{x})$. This is the *Markov condition*:

$$P(s(\vec{x})|\vec{S}/s(\vec{x})) = P(s(\vec{x})|\{s(\vec{y}) : \vec{y} \in N(\vec{x})\}),$$

  where $\vec{S}/s(\vec{x})$ denotes all states in $\vec{S}$ except $s(\vec{x})$.
- In real vision applications, this type of prior, including the size of the neighborhoods, can be estimated from the statistics of natural images.

Markov structure (II)

- Next, we define a probability model for the observed image features at positions $\vec{x}$ in the image. We use the same models as before, at each position $\vec{x}$:

$$P(f(I(\vec{x}))|s) = \{P(f(I(\vec{x}))|s=1)\}^{s}\{P(f(I(\vec{x}))|s=0)\}^{1-s}.$$

- We combine these, using independence assumptions, to get a distribution:

$$P(f(\vec{I})|\vec{S}) = \prod_{\vec{x}} P(f(I(\vec{x}))|s) = \frac{1}{Z_I}\exp\{\sum_{\vec{x}} s(\vec{x})(\log \frac{P(f(I(\vec{x}))|s=1)}{P(f(I(\vec{x}))|s=0)}\},$$

where Z_I is a normalization term (which can be calculated directly).

Posterior distribution (I)

- These distributions $P(f(\vec{I})|\vec{S})$ and $P(\vec{S})$ can be combined to get the posterior distribution $P(\vec{S}|f(\vec{I}))$, which is of form:

$$P(\vec{S}|f(\vec{I})) = \frac{1}{Z_p} \exp\{-E(\vec{S})\},$$

where

$$E(\vec{S}) = -\sum_{\vec{x}} s(\vec{x}) \log \frac{P(f(I(\vec{x}))|s=1)}{P(f(I(\vec{x}))|s=0)} + \sum_{\vec{x}} \sum_{\vec{y} \in N(\vec{x})} \gamma\{s(\vec{x}) - s(\vec{y})\}^2.$$

- The first term of $E(\vec{S})$ gives the local cues for foreground or background (the log-likelihood ratios of the features), while the second term adds the local context. This context encourages neighboring positions to be either all foreground or all background. Note that this method of specifying a distribution $P(\vec{S})$ in terms of a function $E(\vec{S})$ will keep reoccurring throughout this section.

Posterior distribution (II)

- This model specifies the posterior distribution for foreground-background classification using spatial context, and as we will show, similar methods can be applied to other visual tasks. But there remains the issue of how to estimate the most probable states, i.e., computing the Bayes estimator.

$$\hat{\vec{S}} = \arg\max P(\vec{S}|f(\vec{I})).$$

- In the next two sections we discuss neurally plausible algorithms that can do this. There are two types: (1) stochastic models that are natural extensions of the probabilistic neural models discussed earlier, which in the statistics literature are called *Gibbs* samplers (Liu, 2008), and (2) neural network models that are based on simplified biophysics of neurons but that can also, in certain cases, be related to *mean field approximations* to the stochastic models.

Graphical model figures

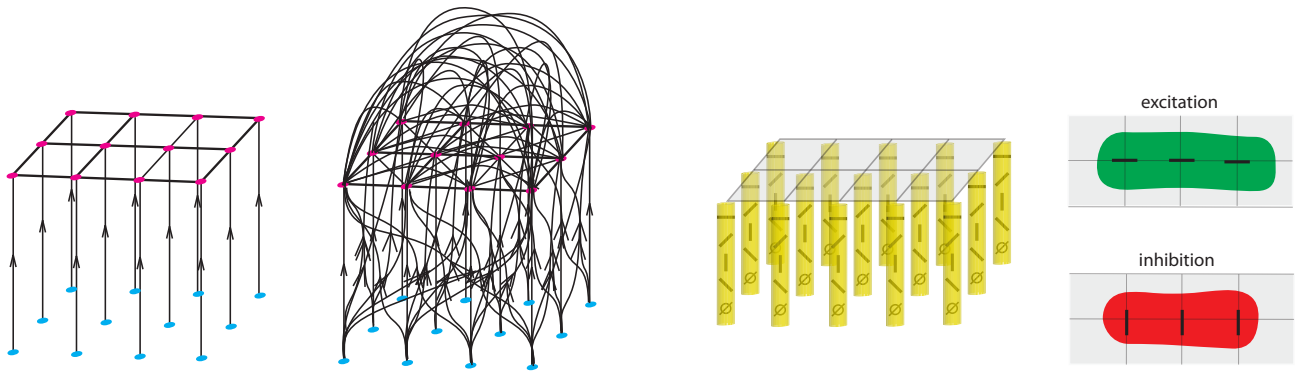


Figure 27 : Far left: The graphical structure of the Markov model with nearest neighbor connections. Left: A fully connected graphical model. Right: A hyper-column structure in which neurons within each column are tuned to different orientations and inhibit each other. Far right: Edges have excitation (green) along the direction of the edge and inhibition (red) perpendicular to the edge.

Probabilistic models of groups of neurons. (I)

- Here, we introduce a more general probability distribution. It is also specified by a model defined over a graph where the nodes correspond to neurons and the edges to connections between them. But we will not make any Markov restrictions on the edges, so this model can be fully connected.
- More specifically, we have set of M neurons with states $\vec{S} = (s_1, ..., s_M)$ and with input $\vec{I} = (I_1, ..., I_N)$. We specify a *Gibbs probability distribution* over the set of activity of all neurons $\vec{S} = (s_1, ..., s_n)$ as follows. First we define an energy function:

$$E(\vec{S}, \vec{I} : \vec{W}, \vec{\theta}) = -\sum_{ij} W_{ij} s_i I_j - (1/2) \sum_{kl} \theta_{kl} s_k s_l.$$

Probabilistic models of groups of neurons. (II)

- This energy contains two types of terms: (1) those of form $s_i I_j$, which give the interactions between the states of the neurons $\vec{S}$ and the input $\vec{I}$, and (2) those that specify interactions between the neurons. This energy is used to specify a Gibbs distribution:

$$P(\vec{S}, \vec{I}) = \frac{1}{Z} \exp\{-E(\vec{S}, \vec{I} : \vec{W}, \vec{\theta})\}. \tag{26}$$

- Here Z is a normalization constant chosen to ensure that $\sum_{\vec{S}} P(\vec{S}|\vec{I}) = 1$. Note that Gibbs distribution originally arose in statistical physics, to specify the probability distribution of a physical system in thermal equilibrium. Here the physical energy of the system is E, and the distribution can be derived using the maximum entropy principle.

Probabilistic models of groups of neurons. (III)

- The weights $\{w_{ij}\}, \{\theta_{kl}\}$ specify the strength of the interactions between the neuron and the inputs, and between the neurons and each other. In particular, the *interaction term* $\sum_{kl} \theta_{kl} s_k s_l$ specifies the interactions between the neurons. If this term is not present, then the distribution simplifies and can be expressed as a product of independent distributions:

$$P(\vec{S}|\vec{I}) = \frac{1}{Z} \exp\{\sum_{ij} w_{ij} s_i I_j\} = \prod_{i=1}^{n} P(s_i|\vec{I}). \tag{27}$$

- Hence in this special case, the neurons act independently and are driven purely by the input (i.e., there is no context). As a technical point, the normalization factor in this case can be computed directly as $Z = \prod_i Z_i$, where $Z_i = \sum_{s_i=0}^{1} \exp\{\sum_j w_{ij} s_i I_j\}$.

Stochastic dynamics (I)

- Now we specify stochastic dynamics on this model. These dynamics have two purposes: first, to describe the activities of sets of neurons interacting with each other; second, to provide algorithms for estimating properties, such as the most probable configurations of the states $\vec{S}$, which can be used for visual tasks and for making decisions.
- To specify stochastic dynamics, we generalize the stochastic neural model, equation (23), to deal with a set of neurons. A neuron received input $\vec{S}$ from other neurons in addition to direct input from the stimulus $\vec{I}$. Consider only the activity of this neuron, fixing the states of all the others. Then the neurons will have total input of $\sum_j w_{ij} I_j$ plus input $\sum_k \theta_{ik} s_k$ from the other neurons.

Stochastic dynamics (II)

- Then, extending equation (23), the probability that the cell i fires is:

$$P(s_i|\vec{I}, \vec{S}_{/i}) = \frac{1}{Z_i} \exp\{s_i(\sum_j w_{ij} I_j + \sum_{k \neq i} \theta_{ik} s_k)\} \tag{28}$$

  where the notation $\vec{S}/i$ means the states $\{s_j : j \neq i\}$ of all the neurons except the neuron we are considering. The term Z_i is defined so that the distribution is normalized, so it is given by $Z_i = 1 + \exp\{\sum_j w_{ij} I_j + \sum_{k \neq i} \theta_{ik} s_k\}$.
- This gives the following dynamics for a group of neurons. At each time, a neuron is selected at random and fires with a probability specified by equation (28). This model assumes that no neurons ever fire at the same time and ignores the time for a spike fired from one neuron to reach other neurons.

Relations to Gibbs distribution?

How does this stochastic dynamics relate to the Gibbs distribution specified above? From the statistical perspective, this is an example of *Markov Chain Monte Carlo* (MCMC) sampling (Liu, 2008). MCMC refers to a class of algorithms that explore the state space of $\vec{S}$ stochastically so that it will gradually move to configurations that have high probability $P(\vec{S}|\vec{I})$. More precisely, MCMC algorithms are guaranteed to give samples from the Gibbs distribution — $\vec{S}_1, ..., \vec{S}_M \; P(\vec{S}|\vec{I})$. The stochastic update rule in equation (28) is a special type of MCMC algorithm known as a *Gibbs sampler*, because it samples from the conditional distribution $P(s_i|\vec{I}, \vec{S}_{/i})$. These samples enable us to estimate the most probable state of the system $\hat{\vec{S}} = \arg\max P(\vec{S}|\vec{I})$, hence they can estimate the MAP estimator of $\vec{S}$ and make optimal decisions for visual tasks.

Learning and Boltzmann machines

To apply these models to visual tasks, we need to specify the weights. One strategy is purely data driven and consists of learning the weights from training examples. This is the *Boltzmann machine* (Ackley et al., 1985) which is out of scope for this chapter. Another strategy is to specify distributions for specific visual tasks, and we will give examples in the next few sections.

Dynamical system models of neurons (I)

There is an alternative way to model sets of neurons using *dynamical systems* based on simplified models of their biophysics (Rieke et al., 1997; Dayan & Abbbott, 2001). Pioneering work on this topic was done by Wilson and Cowan (1972), Grossberg and Mingolla (1968, 1985), Hopfield and Tank (1986), Abbott and Kepler (1990), and others. There is no space to cover the richness of these models, and in any case, these lectures concentrate on the probabilistic formulation. But we will discuss an important subclass of dynamical models (Hopfield & Tank, 1986) that, as we will show, has very close relations to the probabilistic approach.

Dynamical system models of neurons (II)

- These dynamical systems are described as follows (Hopfield & Tank, 1986). A neuron is described by two (related) variables: (1) a continuous valued variable $u_i \in \{-\infty, \infty\}$, and (2) a continuous variable $q_i \in \{0, 1\}$. Roughly speaking, u_i represents the input to the cell body (soma), both direct input and input from other neurons and q_i describes the probability that the cell will fire an action potential. These variables are related by the equations $u_i = \log(q_i/(1 - q_i))$ or, equivalently, by $q_i = \sigma(u_i)$ (where $\sigma(.)$ is the sigmoid function).
- The dynamics of the neuron is given by:

$$\frac{du_i}{dt} = -u_i + \sum_j w_{ij} I_j + \sum_k \theta_{ik} q_k. \tag{29}$$

- Here, as before, $\sum_j w_{ij} I_j + \sum_k \theta_{ik} q_k$ represent the direct input and the input from the other neurons.

Dynamical system models of neurons (III)

This dynamic system continually decreases a function $F(\vec{q})$, so that $(dF)/dt \leq 0$. The function F acts as a *Lyapunov function* for the system in the sense that it decreases monotonically as time t increases and is bounded below. The existence of a Lyapunov function for the dynamics guarantees that the system will converge to a state that minimizes $F(\vec{q})$ (note that $F(\vec{q})$ will typically have many minimums, and the system may converge to any one of them).

Relations between probabilistic models and dynamical system models (I)

- Perhaps surprisingly, there is a very close relationship between the dynamic systems in equation (29) and the stochastic update in equation (23). More specifically, the dynamic system is a mean field approximation to the stochastic dynamics. *Mean field theory* (MFT) was developed by physicists as a way to approximate stochastic systems.
- To explain this relationship, we first define *the mean field free energy* $F(\vec{q})$:

$$F(\vec{q}) = -\sum_{ij} W_{ij} I_j q_i - (1/2) \sum_{ij} \theta_{ij} q_i q_j + \sum_i \{q_i \log q_i + (1-q_i) \log(1-q_i)\}. \tag{30}$$

- Next we specify dynamics by performing steepest descent on the free energy (multiplies by a positive factor):

$$\frac{dq_i}{dt} = -q_i(1-q_i)\frac{\partial F(\vec{q})}{\partial q_i}. \tag{31}$$

Relations between probabilistic models and dynamical system models (II)

- Interestingly, these are identical to the dynamical system in equation (29). This can be seen by introducing a new variable $u_i = \log q_i/(1-q_i)$, which implies that $q_i = \sigma(u_i)$. Note that $\partial F/\partial q_i = -\sum_j W_{ij} I_j - \sum_j \theta_{ij} q_j + \log q_i/(1-q_i)$, $u_i = \log q_i/(1-q_i)$, and $dq_i/q_i(1-q_i) = du_i$.
- Equation (31) implies that the dynamical system decreases the free energy $F(\vec{q})$ monotonically with time t. This is because $dF/dt = -\sum_i (\partial F/\partial q_i)(\partial q_i/\partial t) = -\sum_i q_i(1-q_i)(\partial F/\partial q_i)^2$. Hence $F(\vec{q})$ is a Lyapunov function for equations (29, 31), and so the dynamics converges to a fixed point.

Relations between probabilistic models and dynamical system models (III)

This shows that there is a close connection between the neural dynamical system and minimizing the mean field free energy. In turn, the mean field free energy is related to deterministic approximations to stochastic update methods like Gibbs sampling (Amit, 1992; Hertz, 1991). This connection is technically advanced and is not needed to understand the rest of this chapter. Briefly, the mean field free energy $F(\vec{q})$ is the *Kullback-Leibler divergence* $F(Q) = \sum_{\vec{S}} Q(\vec{S}) \log \frac{Q(\vec{S})}{P(\vec{S}|\vec{I})}$ between the distribution $P(\vec{S}|\vec{I})$ and a factorized distribution $Q(\vec{S}) = \prod_i q_i^{S_i}(1-q_i)^{1-S_i}$ (plus an additive constant). Hence the dynamical system seeks to find the factorized distribution $\hat{Q}(\vec{S})$ that best approximates $P(\vec{S}|\vec{I})$ by minimizing the Kullback-Leibler divergence. In this approximation the response q_i is an approximation to the expected response $\sum_{S_1} S_i P(\vec{S}|\vec{I})$. The connections between mean field theory and neural models was described in Yuille, 1987). For technical discussions about mean field theory and Gibbs sampling see (Yuille, 2011).

Lecture 12.5

- This lecture describes how groups of neurons can perform edge detection, edge grouping, stereo, and motion.
- We also introduce weak methods for cue combination.
- This lecture includes the exercises: (12.5.1) Hopfield network for binocular stereo, and (12.5.2) Cue combination.

The line process model (I)

- Our first example is the classic *line process* model (Geman & Geman, 1984; Blake & Zisserman, 2003; Mumford & Shah, 1989), which was developed as a way to segment images. It has explicit *line process* variables that "break" images into regions where the intensity is piecewise smooth. Our presentation follows the work of Koch et al. (1986), who translated it into neural circuits.
- The model takes intensity values $\vec{I}$ as input, and outputs smoothed intensity values. But this smoothness is broken at places where the intensity changes are too high. The model has continuous variables $\vec{J}$ representing the intensity, and binary-valued variables $\vec{l}$ for the line processes (or edges). The model is formulated as performing *maximum a posteriori* (MAP) estimation. The algorithm for estimating MAP is a neural network model that can be derived from the original Markov model (Geman & Geman, 1984) by mean field theory (Geiger & Yuille, 1991). Note that in this model, the variables do not have to represent intensity. Instead they can represent texture, depth, or any other property that is spatially smooth except at sharp discontinuities.

The line process model (II)

- For simplicity we present the weak membrane model in one dimension. The input is $\vec{I} = \{I(x) : x \in \mathcal{D}\}$; the estimated, or smoothed, image is $\vec{J} = \{J(x) : x \in \mathcal{D}\}$; and the line processes are denoted by $\vec{l} = \{l(x) : x \in \mathcal{D}\}$, where $l(x) \in \{0, 1\}$.
- The model is specified by a posterior probability distribution:

$$P(\vec{J}, \vec{l} | \vec{I}) = \frac{1}{Z} \exp\{-E[\vec{J}, \vec{l} : \vec{I}]/T\},$$

where

$$E[\vec{J}, \vec{l} : \vec{I}] = \sum_x (I(x) - J(x))^2 + A \sum_x (J(x+1) - J(x))^2 (1 - l(x)) + B \sum_x l(x).$$

The line process model (III)

The first term ensures that the estimated intensity $J(x)$ is close to the input intensity $I(x)$. The second encourages the estimated intensity $J(x)$ to be spatially smooth (e.g., $J(x) \approx J(x+1)$), unless a line process is activated by setting $l(x) = 1$. The third pays a penalty for activating a line process. The result encourages the estimated intensity to be piecewise smooth unless the input $I(x)$ changes significantly, in which case a line process is switched on and the smoothness is broken. The parameter T is the variance of the probability distribution and has a default value $T = 1$.

The line process model illustration

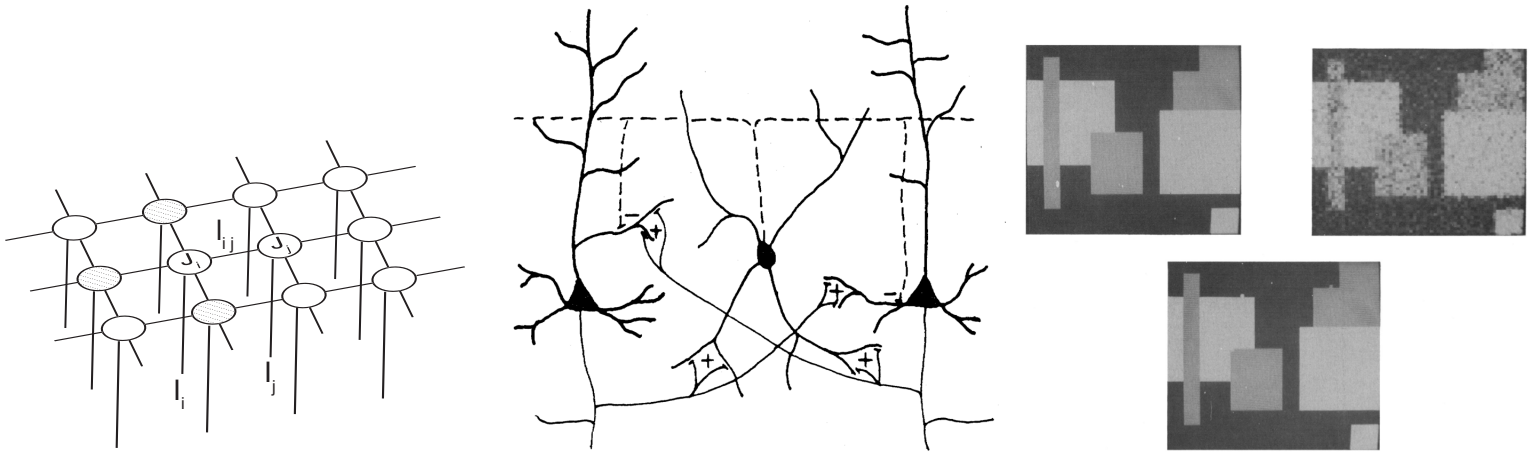


Figure 28 : A representation of the line process model (left) compared to a real neural network (center). On the right, the original image (upper left), the image corrupted with noise (upper right), and the image estimated using the line process model (bottom).

The line process model and neural circuits (I)

- This model can be implemented by a neural circuit (Koch et al., 1986). The connections between these neurons is shown in the previous figure. To implement this model Koch et al., (1986) proposed a neural net model that is equivalent to doing mean field theory on the weak membrane MRF (as discussed earlier) by replacing the binary-valued line process variables $l(x)$ by continuous variables $q(x) \in [0, 1]$ (corresponding roughly to the probability that the line process is switched on).
- This gives an algorithm that updates the regional variables $\vec{J}$ and the line variables $\vec{q}$ in a coupled manner. It is helpful, as before, to introduce a new variable $\vec{u}$ which relates by $q(x) = \frac{1}{1+\exp\{-u(x)/T\}}$ and $u(x) = T \log \frac{q(x)}{1-q(x)}$.

The line process model and neural circuits (II)

$$\frac{dJ(x)}{dt} = -2(J(x) - I(x))$$

$$= -2A\{(1 - q(x))(J(x) - J(x+1)) + (1 - q(x-1))(J(x) - J(x-1))\}, \quad (32)$$

$$\frac{dq(x)}{dt} = \frac{1}{T} q(x)(1 - q(x))\{A(J(x+1) - J(x))^2 - B - T \log \frac{q(x)}{1 - q(x)}\}, \quad (33)$$

$$\frac{du(x)}{dt} = -u(x) + A(J(x+1) - J(x))^2 - B. \quad (34)$$

The update rule for the estimated intensity $\vec{J}$ behaves like nonlinear diffusion, which smooths the intensity while keeping it similar to input $\vec{I}$. The diffusion is modulated by the strength of the edges $\vec{q}$. The update for the lines $\vec{q}$ is driven by the differences between the estimated intensity; if this is small, then the lines are not activated.

The line process model and neural circuits (III)

This algorithm has a Lyapunov function $L(\vec{J}, \vec{q})$ (derived using mean field theory methods) and so will converge to a fixed point, with

$$L(\vec{J}, \vec{q}) = \sum_x (I(x) - J(x))^2 + A \sum_x (J(x+1) - J(x))^2 (1 - q(x)) + B \sum_x q(x) + T \sum_x \{q(x) \log q(x) + (1 - q(x)) \log(1 - q(x))\}. \quad (35)$$

Relations to electrophysiology (I)

- There is some evidence that a generalization of this models roughly matches the electrophysiological findings for those types of stimuli. The generalization is performed by replacing the intensity variables $I(x), J(x)$ by a filterbank of Gabor filters so that the weak membrane model enforces edges at places where the texture properties change (Lee et al., 1992). The experiments, and their relation to the weak membrane models are reviewed in (Lee & Yuille, 2006). The initial responses of the neurons, for the first 80 msec, are consistent with the linear filter models described earlier. But after 80 msec, the activity of the neurons changes and appears to take spatial context into account.
- While the weak membrane model is broadly consistent with the perceptual phenomena of segmentation and "filling in," the types of filling in, their dynamics, and the neural representations of contours and surface are complicated (von der Heydt, 2002; Komatsu, 2006). Exactly how contour and surface information is represented and processed in cortex is an active topic of research (Grossberg & Hong, 2006; Roe et al., 2012).

Relations to electrophysiology (II)

- The findings of the electrophysiological experiments are summarized as follows:
  (1) There are two sets of neurons, with one set encoding regional properties (such as average brightness), and the other set coding boundary location (in agreement with J and I variable in the model, respectively).
  (2) The processes for computing the region and the boundary representations are tightly coupled, with both processes interacting with and constraining each other (as in the dynamical equations above).
  (3) During the iterative process, the regional properties diffuse within each region and tend to become constant, but these regional properties do not cross the region (in agreement with the model).
  (4) The interruption of the spreading of regional information by boundaries results in sharp discontinuities in the responses across two different regions (in agreement with the model). The development of abrupt changes in regional responses also results in a gradual sharpening of the boundary response, reflecting increased confidence in the precise location of the boundary.
- These findings are roughly consistent with neural network implementations of the weak membrane model. But other explanations are possible. For example, the weak membrane model requires lateral (sideways) interaction, and it is possible that the computations are done hierarchically using feedback from V2 to V1.

Relations to electrophysiology illustration

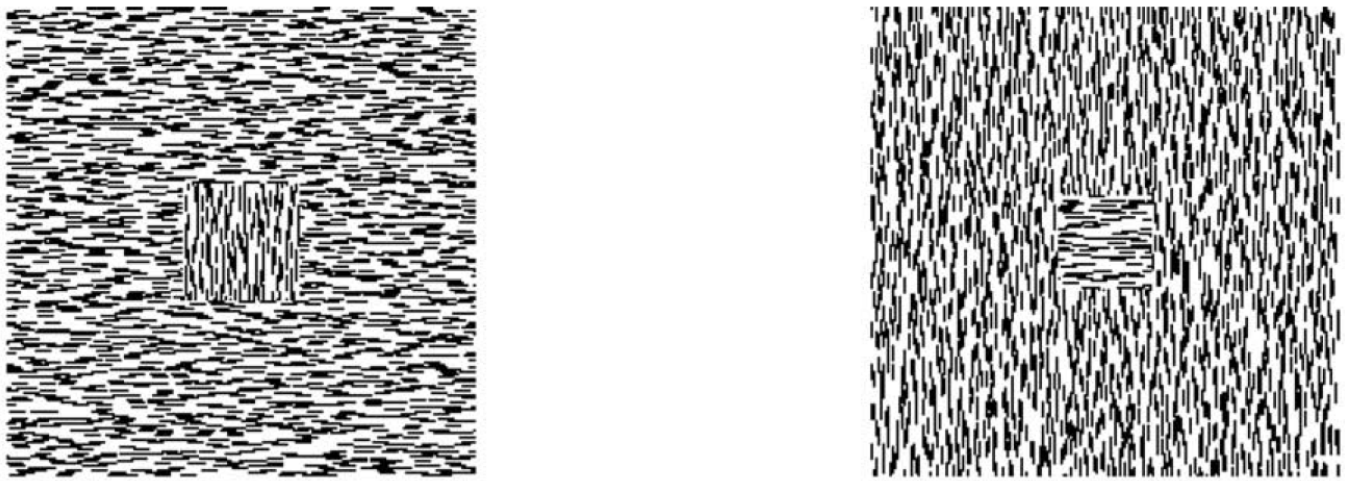


Figure 29 : The stimuli for the experiments by TS Lee and his collaborators (Lee & Yuille, 2006).

Edge detection with spatial context (I)

- Our second example is to develop a model for detecting edges using spatial context. This relates to the phenomena known as association fields, see chapter figure 12.26 (left panel), where Gabor filters that are spatially aligned (in orientation and direction) get grouped into a coherent form.
- For this model, we have a set of neurons at every spatial position x, each tuned to a different angle $\theta_i : i = 1, ..., 8$, and a default cell at angle θ_0. The first cells are designed to detect edges at each orientation – i.e., they can be driven by the log-likelihood ratio of an edge detector at orientation θ_i at this position. The default cell is a dummy that is intended to fire if there is no edge present at this position. This organization forms a population of cells arrayed according to orientation (similar to a hypercolumn in V1).

Edge detection with spatial context (II)

We define a Gibbs distribution for the activity s_{x,θ_i} of the cells. The energy function $E(\vec{s})$ contains four types of terms: The first term, $\sum_x \sum_{i=0}^{8} s_{x,i}\phi(f_1, ..., f_M)$, represents the local evidence for an edge at each point and for its orientation. The second term $\sum_x (\sum_{i=0}^{8} s_{x,i} - 1)^2$., is intended to ensure that only one cell is active at any spatial position. This corresponds to an inhibitory interaction between cells in the same hypercolumn. The cells in the hypercolumn give alternative, and inconsistent, interpretations of the input – hence only one of them can be correct. The third term encourages edges to be continuous and change their directions smoothly. To define this term, we let $\vec{\theta_i} = (\cos\theta_i, \sin\theta_i)$ and $\vec{\theta_i}^T = (-\sin\theta_i, \cos\theta_i)$ denote the tangent to the edge and the normal. This term encourages there to be edges in the tangent direction, while the next term discourages them in the normal direction. This term is motivated by the intuition that curves are spatially smooth and can be justified by the statistics of natural images (Geisler & Perry, 2009; Elder & Goldberg, 2002).

Edge detection with spatial context (III)

We write it as $\sum_{x,y} \sum_{i,j=1}^{8} W^T_{(x,\theta_i),(y,\theta_j)} s_{x,i} s_{y,j}$, where

$$W^T_{(x,\theta_i),(y,\theta_j)} = -\exp\{-|\vec{\theta}_i - \vec{\theta}_j|/K_1\} \exp\{-|x-y|/K_2\} \exp\{-|\hat{x}y - \vec{\theta}_i|/K_3\} \quad (36)$$

and $\hat{x}y$ is the unit vector in direction $x - y$. This term encourages edges that are in similar directions (first term) and nearby in position (second term), where the edge orientation is similar to the difference $x - y$ between the two points. This term is excitatory. The fourth and final term is inhibitory and discourages edges from being parallel to each other (if they are nearby). It is written as $\sum_{x,y} \sum_{i,j=1}^{8} W^N_{(x,\theta_i),(y,\theta_j)} s_{x,i} s_{y,j}$. Here,

$$W^N_{(x,\theta_i),(y,\theta_j)} = \exp\{-|x-y|/K_4\} \exp\{-|\hat{x}y - \vec{\theta}^T_i|\} \quad (37)$$

Edge detection with spatial context (IV)

- The first term says this interaction decreases with distance. The second term discourages edges which are parallel to each other.
- This gives an overall energy:

$$E(\vec{s}) = \sum_x \sum_{i=0}^{8} s_{x,i}\phi(f_1, ..., f_M) + \hat{K}_0 \sum_x (\sum_{i=0}^{8} s_{x,i} - 1)^2$$

$$+\hat{K}_1 \sum_{x,y} \sum_{i,j=1}^{8} W^T_{(x,\theta_i),(y,\theta_j)} s_{x,i} s_{y,j} + \hat{K}_2 + \sum_{x,y} \sum_{i,j=1}^{8} W^N_{(x,\theta_i),(y,\theta_j)} s_{x,i} s_{y,j}. \quad (38)$$

- This yields a probability:

$$P(\vec{s}|\vec{f}) = \frac{1}{Z} \exp\{-E(\vec{s})\}.$$

- This model can be implemented in neural networks by defining either stochastic or deterministic neural dynamics (i.e., either Gibbs sampling or mean field theory). The resulting update equations are more complex than those defined for our earlier examples but have the same basic ingredients. Models of this type can qualitatively account for associative field phenomena.

Stereo models

This section introduces computational models for estimating depth by binocular stereo. The key problem to solve is the *correspondence problem* between the inputs in the two eyes to determine the *disparity*. Then the depth of the points in space can be estimated by trigonometry. (This presupposes that the eyes are *calibrated*, meaning that the distance between the eyes and the direction of gaze are known, which is beyond the scope of this chapter.) Julesz (1971) showed that humans could perceive depth from stereo if the images consisted of random dot stereograms, which minimize the effect of feature similarity cues, suggesting that human vision can solve this task by relying mainly on geometric regularities (assumed about the structure of the world). Other researchers (Bulthoff & Mallot, 1988) have studied human estimation of surface shape quantitatively and showed, among other things, bias toward fronto-parallel surfaces.

Stereo: The correspondence problem

Most stereo algorithm address the correspondence problem by assuming that (1) image features in the two eyes are more likely to correspond if they have similar appearance, and (2) the surface being viewed obeys prior knowledge, such as being piecewise smooth (e.g., like the weak membrane model). The first assumption depends on local properties of the images, while the second assumption uses nonlocal context. In an earlier lecture, we discussed how a population of Gabor filters could be used to match local image features. Here we describe how context can be used to impose prior knowledge about the geometry of the scene. We will study classic models, that assume that the surface is piecewise smooth. This leads to a Markov field model that includes excitatory connections, imposing the geometric constraints, with inhibitory connections that prevent points from one eye having more than one match in the second eye. This yields an algorithm that involves cooperation to implement the excitatory constraints, and competition to deal with the inhibitory constraints. This is consistent with findings from recent electrophysiological experiments (Samonds et al., 2009),(Samonds et al., 2012), which complement experiments (Ohzawa et al., 1990) that tested the local stereo models described earlier.

A cooperative stereo model (I)

- We now specify a computational model for stereo that for simplicity, we formulate in one dimension. There is a long history of this type of model, starting with the cooperative stereo algorithm (Dev, 1975; Marr & Poggio, 1976), and current computer vision stereo algorithms are mostly designed on similar principles.
- We specify the left and right images by $\vec{I}_L, \vec{I}_R$ and denote features extracted from them by $\vec{f}(\vec{I}_L) = \{f(x_L) : x_L \in \mathcal{D}_L\}$, $\vec{f}(\vec{I}_R) = \{f(x_R) : x_R \in \mathcal{D}_R\}$. We define a discrete-valued correspondence variable $V(x_L, x_R)$ so that if $V(x_L, x_R) = 1$, the features at x_L, x_R in the two images correspond, and hence the disparity is $x_L - x_R$. If the features do not match, then we set $V(x_L, x_R) = 0$. We encourage all data points to match one other data point, but allow some data points to be unmatched and others to match more than once (by paying a penalty).

A cooperative stereo model (II)

We specify a distribution $P(\vec{V}|\vec{f}(\vec{I}_L), \vec{f}(\vec{I}_R)) = \frac{1}{Z}\exp\{-E(\vec{V}; \vec{f}(\vec{I}_L), \vec{f}(\vec{I}_R))/T\}$, where the energy $E(\vec{V}; \vec{f}(\vec{I}_L), \vec{f}(\vec{I}_R))$ is given by:

$$E(\vec{V}; \vec{f}(\vec{I}_L), \vec{f}(\vec{I}_R)) = \sum_{x_L, x_R} V(x_L, x_R) M(f(x_L), f(x_R))$$

$$+A\sum_{x_L}(\sum_{x_R} V(x_L, x_R) - 1)^2 + A\sum_{x_R}(\sum_{x_L} V(x_L, x_R) - 1)^2$$

$$+C\sum_{x_L, x_R}\sum_{y_L \in N(x_L)}\sum_{y_R \in N(x_R)} V(x_L, x_R)V(y_L, y_R)\{(x_R - x_L) - (y_R - y_L)\}^2. \quad (39)$$

A cooperative stereo model (III)

The first term imposes matches between image points with similar features; here $M(.,.)$ is a measure that takes small values if $f(x_L), f(x_R)$ are similar and large values if they are different. We will discuss at the end of this section how $M(f(x_L), f(x_R))$ relates to the model for local stereo discussed earlier. The second two terms penalize image points that are either unmatched or matched more than once. The third term encourages the disparities, $x_L - x_R$, to be similar for neighboring points (here $N(.)$ defines a spatial neighborhood as before). These models can be applied to two-dimensional images by solving the correspondence problem for each epipolar line separately (by maximizing $P(\vec{V}|\vec{f}(\vec{I}_L), \vec{f}(\vec{I}_R))$). This is shown in the figure that follows. The parameter T is the variance of the model, as for the line process model, and has default value $T = 1$.

A cooperative stereo model illustration

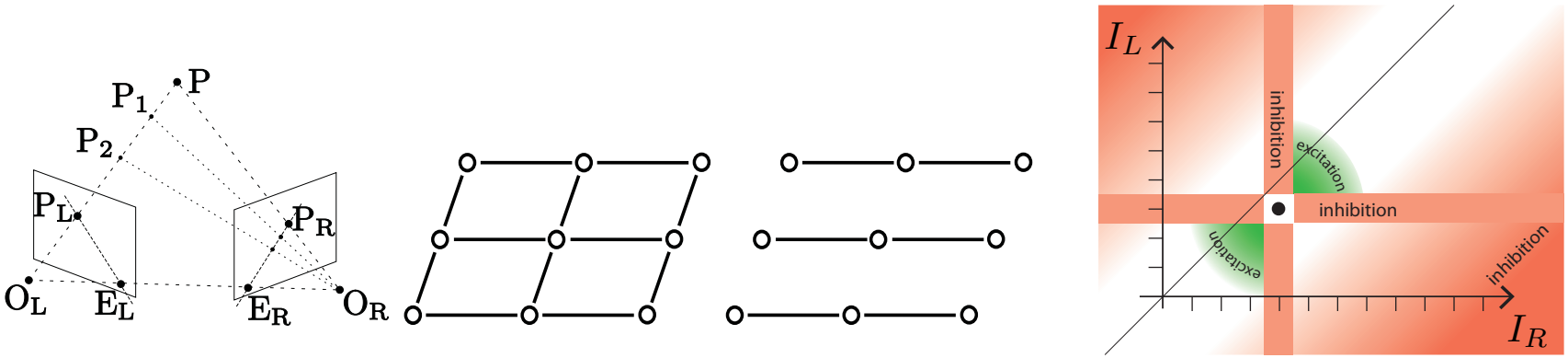


Figure 30 : Far left and center: The geometry of stereo. A point P in 3-D space is projected onto points PL and PR. The projection is specified by the focal points OL, OR, and the directions of the cameras' gaze (the camera geometry). The geometry of stereo enforces that points in the plane specified by P, OR, OL must be projected onto corresponding lines EL, ER (the epipolar line constraint). If we can find the correspondence between the points on epipolar lines, then we can use trigonometry to estimate their depth, which is (roughly) inversely proportional to the disparity, which is the relative displacement of the two images. Far right: Binocular stereo requires solving the correspondence problem, which involves excitation (to encourage matches with similar depths/disparities) and inhibition (to prevent points from having multiple matches).

A cooperative stereo model (IV)

- We obtain a neural circuit model by performing mean field theory on $P(\vec{V}|\vec{f}(\vec{I}_L), \vec{f}(\vec{I}_R))$. This replaces $V(x_L, x_R) \in \{0, 1\}$ by continuous-valued $q(x_L, x_R) \in [0, 1]$ and an associated variable $u(x_L.x_R) = T \log \frac{q(x_L, x_R)}{1-q(x_L, x_R)}$ with $q(x_l, x_R) = \frac{1}{1+\exp\{-u(x_L, x_R)\}}$.
- The update equation is:

$$\frac{du(x_L, x_R)}{dt} = -u(x_L, x_R) - M(f(x_L), f(x_R))$$
$$-2A(\sum_{y_R \neq x_R} q(x_L, y_R) - 1) - 2A(\sum_{y_L \neq x_L} q(y_L, x_R) - 1),$$
$$-2C \sum_{y_L \in N(x_L)} \sum_{y_R \in N(x_R)} q(y_L, y_R)\{(x_R - x_L) - (y_R - y_L)\}^2. \quad (40)$$

- This update includes the standard integration term (first term), and the second term encourages matches where the features agree. There is also inhibition between competing matches (the third and fourth term), and excitation for matches that are consistent with a smooth surface (last term).

A cooperative stereo model: Interactive demo

There is a variant of this algorithm that is a discrete Hopfield network which attempts to minimize the energy $E(\vec{V}; \vec{f}(\vec{I_L}), \vec{f}(\vec{I_R}))$ in equation (39). The algorithm starts by assigning initial values, 0 or 1, to each state variable $V(x_L, x_R)$. The algorithm proceeds by selecting a state variable, changing its value (e.g., changing $V(x_L, x_R) = 1$ to $V(x_L, x_R) = 0$), calculating if this change reduces the energy $E(\vec{V}; \vec{f}(\vec{I_L}), \vec{f}(\vec{I_R}))$, and keeping the change if it does. This process repeats until the algorithm converges (i.e., all possible changes raise the value of the energy).

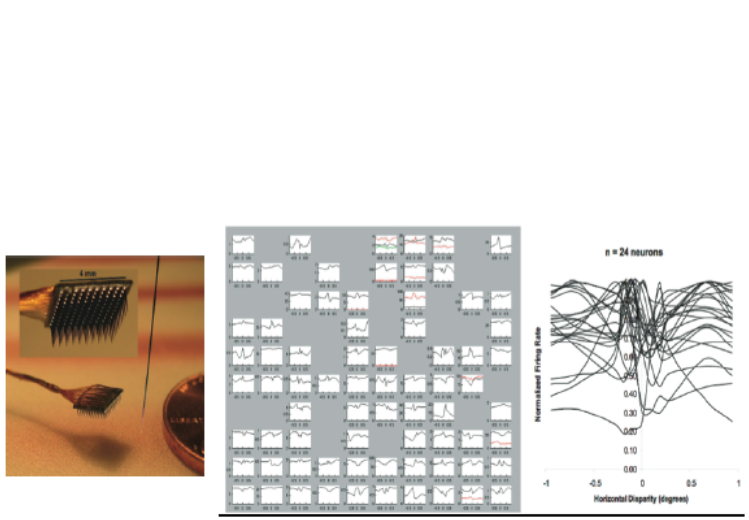
A cooperative stereo model and the local model

How does the cooperative stereo algorithm relate to our earlier algorithm for computing stereo disparity locally? Recall that the algorithm estimated the disparity at a single point by having a set of neurons tuned to different disparities $\{D_i : i = 1, ..., N\}$, summing the votes $v(D_i)$ for each disparity by equation (14), and selecting the disparity with the most votes. Using the cyclopean coordinate system (Jules, 1971), we express the disparity by $D(x) = \frac{1}{2}(x_R - x_L)$, where $x = \frac{1}{2}(x_R + x_L)$. At each point x we specify a population of neurons that encodes the votes $v(D(x))$ for the different disparities. Then, instead of using winner-take-all to make a local decision, we feed the responses $v(D(x))$ back into cooperative stereo algorithm by defining $M(f(x_L), f(x_R)) = \exp\{-v(\frac{1}{2}(x_R - x_L))\}$ (the negative exponential $\exp\{-\}$ is required so the $M(f(x_L), f(x_R))$ is small if the vote for disparity $D(x) = \frac{1}{2}(x_R - x_L)$ is large).

A cooperative stereo model and electrophysiology

Analyses of electrophysiological studies (Samonds et al., 2009),(Samonds et al., 2012) were in general agreement with the predictions of this type of stereo algorithm. In particular, studies showed that neural population responses included excitation between cells tuned to similar disparities at neighboring spatial positions as well as inhibition between cells tuned to different disparities at the same position. In addition, Samonds et al. (2013) implemented a variant of the stereo algorithm described above and showed that it could account for additional phenomena, such as sharper tuning to the disparity for larger stimuli and performance on anticorrelated stimuli (where the left and right images have opposite polarity).

A cooperative stereo model and electrophysiology illustration



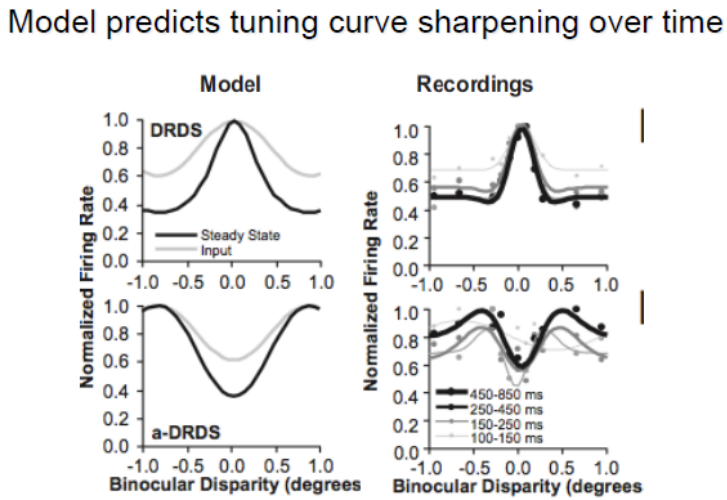


Figure 31 : Experiments for testing stereo algorithms (Samonds et al., 2009, 2012). Left: The experimental setup. Right: The experiments give evidence for excitation between similar disparity and inhibition to prevent multiple matches.

Motion

Similar models have been applied to a range of motion phenomena. Early computational studies (Ullman, 1979) showed that several perceptual phenomena of long-range motion could be described by a "minimal mapping" theory that uses a slowness prior. Subsequent work showed that smoothness priors accounted for findings on short-range motion (Hildreth, 1984), including the surprising fact that an ellipse rotating in the image plane is perceived to move non-rigidly. Yuille et al. (1998) qualitatively showed that a slow-and-smooth prior could account for a large range of motion perceptual phenomena – including motion capture and motion cooperation – for short- and long-range motion. Weiss and his collaborators showed that slow (Weiss & Adelson, 1998) and slow-and-smooth priors (Weiss et al., 2002) could explain other short-range motion phenomena, such as how percepts can change dramatically as we alter the balance between the likelihood and prior terms (i.e., for some stimuli the prior dominates the likelihood and vice versa).

Motion

All these models combine local estimates of the motion, such as those described in the previous section, with contextual cues implementing slow-and-smooth priors. They can be formulated using the same mathematical techniques. See `http://www.michaelbach.de/ot/mot-motionBinding/` to see how spatial context can be affected by other cues such as occlusion. It is also possible to perceive three-dimensional structure by observing a motion sequence (somewhat similar to binocular stereo) as can be seen in `http://michaelbach.de/ot/mot-ske/`.

Motion and time

The perception of motion can be strongly influenced by its history and not merely by the change of image from frame to frame. For example, Anstis and Ramachandran(1987) demonstrated perceptual phenomena where motion perception seems to require a temporal coherence prior in addition to the slow and smoothness priors described earlier in this section. Similarly, Watamaniuk et al. (1995) demonstrated that humans could detect a coherently moving dot despite the presence of many incoherently moving dots. These classes of phenomena can be addressed by models that make prior assumptions about how motion changes over time. These can be performed (Yuille et al., 1998) by adapting the Bayes-Kalman filter (Kalman, 1960; Ho & Lee, 1964) filter which gives an optimal way to combine information over time.

Bayes-Kalman filter (I)

- The task of the Bayes-Kalman filter is to estimate the state x_t of a system at time t dependent on a set of observations $y_t, ..., y_1$ (e.g., x_t could be the position of an airplane and y_t a noisy measurement of the airplane's position at time t). The model assumes a probability distribution $P(x_{t+1}|x_y)$ for how the state changes over time and a likelihood function $P(y_t|x_t)$ for the observation.
- The task is to estimate the state x_t of a system at time t dependent on a set of observations $y_t, ..., y_1$ (e.g., x_t could be the position of an object and y_t a noisy measurement of the object position at time t). The model assumes a probability distribution $P(x_{t+1}|x_y)$ for how the state changes over time and a likelihood function $P(y_t|x_t)$ for the observation. This can be formulated by a Markov model, where the observations $y_t, ..., y_1$ and states $x_t, ..., x_1$ are represented by the blue and red dots, respectively (the lower and upper dots if viewed in black and white).

Bayes-Kalman filter (II)

- The purpose of Bayes-Kalman is to estimate the distribution $P(x_t|Y_t)$ of the state x_t conditioned on the measurements $Y_t = \{y_t, ..., y_1\}$ up to time t. It performs this by repeatedly performing the following two steps, which are called prediction and correction. The prediction uses the prior $P(x_{t+1}|x_t)$ to predict distribution $P(x_{t+1}|Y_t)$ of the state at $t+1$:

$$P(x_{t+1}|Y_t) = \int dx_t P(x_{t+1}|x_t)P(x_t|Y_t). \tag{41}$$

- The correction step integrates the new observation y_{t+1} to estimate $P(x_{t+1}|Y_{t+1})$ by:

$$P(x_{t+1}|Y_{t+1}) = \frac{P(y_{t+1}|x_{t+1})P(x_{t+1}|Y_t)}{P(y_{t+1}|Y_t)}. \tag{42}$$

- Bayes-Kalman is initialized by setting $P(x_1|y_1) = P(y_1|x_1)P(x_1)/P(y_1)$ where $P(x_1)$ is the prior for the original position of the object at the start of the sequence. Then equations (41, 42) are run repeatedly. The effect of prediction is to introduce uncertainly about the state x_t, while correction reduces uncertainty by providing a new measurement.

Bayes-Kalman filter: Figures

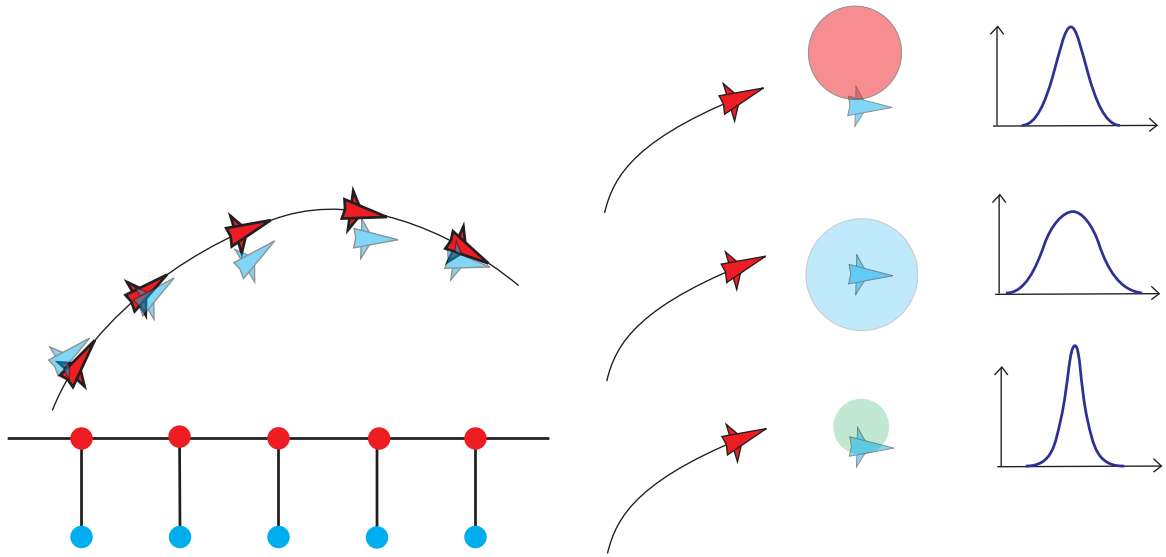


Figure 32 : Left: Graph illustrating the unobserved states (red) and the observed states (blue) as a function of time. The airplanes true positions are shown in red, and their observations (biased) are shown in blue. The Bayes-Kalman filter integrates observations to make estimate the true state using prior probabilities. Right: Bayes-Kalman updates a probability distribution for the estimated position of the target. The variance of the distribution is illustrated by the one-dimensional figure (on the right) and the size of the circle (red, blue, or green). In the prediction stage (middle) the variance becomes large, and after the measurement, the variance becomes smaller.

Summary of models with context

This section illustrated how neural networks and Markov models could be used to apply context to visual tasks. We concentrated on edge detection, segmentation, and binocular stereo. We stressed how context can include excitatory and inhibitory interactions. And how inference can be performed using stochastic neurons (e.g., Gibbs sampling) or dynamic neural networks (e.g., mean field approximations). These models have some relations to psychophysics and electrophysiology. But we stress that detailed biological evidence in favor of these models remains preliminary due to the current limitations of experimental techniques. We note that current computer vision algorithms that address similar visual tasks are more complex although based on similar principles (Blake et al., 2011).

Cue coupling

- This section describes models for coupling different visual cues.
- The ideas in this section are logical extensions of the ideas in the earlier sections. But we are now addressing more complex aspects of vision, so the techniques and the tools become more complex and more abstract as we begin to reason about surfaces, objects, and their relations.

Vision modules and cue combination

- Quantifiable psychophysics experiments for individual cues are roughly consistent with the predictions of the types of models discussed in the previous two sections– see (Bulthoff & Mallot, 1988; Cumming et al., 1993) – but with some exceptions (Todd et al., 2001).
- But how are different visual cues combined?
- The most straightforward manner is to use a separate module for each cue to compute different estimates of the properties of interest, e.g., the surface geometry, and then merge these estimates into a single representation. This was proposed by Marr (Marr, 1982) who justified this strategy by invoking the principle of modular design.
- Marr proposed that surfaces should be represented by a *2 1/2D sketch* that specifies the shape of a surface by the distance of the surface points from the viewer. A related representation, *intrinsic images*, also represents surface shape together with the material properties of the surface.

Cue coupling from a probabilistic perspective

- We consider the problem of cue combination from a probabilistic perspective (Clark & Yuille, 1990).
- This suggests that we need to distinguish between situations when the cues are statistically independent of each other and situations when they are not. We also need to determine whether cues are using similar, and hence redundant, prior information.
- These considerations lead to a distinction between *weak* and *strong* coupling, where weak coupling corresponds to the traditional view of modules, while strong coupling considers more complex interactions. To understand strong coupling, it is helpful to consider the *causal factors* that generate the image.
- Note that there is strong evidence that high-level recognition can affect the estimation of three-dimensional shape, e.g., a rigidly rotating inverted face mask is perceived as nonrigidly deforming face, while most rigidly rotating objects are perceived to be rigid.

Combining cues with uncertainty

- We first consider simple models that assume the cues compute representations independently, and then we combine their outputs by taking linear weighted combinations.
- Suppose there are two cues for depth that separately give estimates $\vec{S}_1^*, \vec{S}_2^*$. One strategy to combine these cues is by linear weighted combination yielding a combined estimate $\vec{S}^*$:

$$\vec{S}^* = \omega_1 \vec{S}_1^* + \omega_2 \vec{S}_2^*,$$

  where ω_1, ω_2 are positive weights such that $\omega_1 + \omega_2 = 1$.
- Landy et al. (1995) reviewed many early studies on cue combination and argued that they could be qualitatively explained by this type of model. They also discussed situations when the individual cues did not combine as well as "gating mechanisms" that require one cue to be switched off.

Case where weights are derived from uncertainties

- An important special case of this model is when the weights are measures of the uncertainty of the two cues. This approach is optimal under certain conditions and yields detailed experimental predictions, which have been successfully tested for some types of cue coupling (Jacobs, 1999; Ernst & Banks, 2002), see (Cheng et al., 2007; Gori et al., 2008) for exceptions.
- If the cues have uncertainties σ_1^2, σ_2^2, we set the weights to be $w_1 = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}$ and $w_2 = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$.
- The cue with lowest uncertainly has highest weight.
- This gives the linear combination rule:

$$\vec{S}^* = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\vec{S}_1^* + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\vec{S}_2^*.$$

Optimality of the linear combination rule (I)

The linear combination is optimal for the following conditions:
1. The two cues have inputs $\{\vec{C}_i : i = 1, 2\}$ and outputs $\vec{S}$ related by conditional distributions $\{P(\vec{C}_i|\vec{S}) : i = 1, 2\}$.
2. These cues are *conditionally independent* so that $P(\vec{C}_1, \vec{C}_2|S) = P(\vec{C}_1|\vec{S})P(\vec{C}_2|\vec{S})$ and both distributions are Gaussians:

$$P(\vec{C}_1|\vec{S}) = \frac{1}{Z_1} \exp\{-\frac{|\vec{C}_1 - \vec{S}|^2}{2\sigma_1^2}\},$$

$$P(\vec{C}_2|\vec{S}) = \frac{1}{Z_2} \exp\{-\frac{|\vec{C}_2 - \vec{S}|^2}{2\sigma_2^2}\}.$$

3. The prior distribution for the outputs is uniform.

Optimality of the linear combination rule (II)

- In this case, the optimal estimates of the output $\vec{S}$, for each cue independently, are given by the maximum likelihood estimates:

$$\vec{S}_1^* = \arg\max_S P(\vec{C}_1|\vec{S}) = \vec{C}_1, \quad \vec{S}_2^* = \arg\max_S P(\vec{C}_2|\vec{S}) = \vec{C}_2.$$

- If both cues are available, then the optimal estimate is given by:

$$\vec{S}^* = \arg\max_{\vec{S}} P(\vec{C}_1, \vec{C}_2|\vec{S}) = \arg\max_{\vec{S}} P(\vec{C}_1|\vec{S})P(\vec{C}_2|\vec{S})$$

$$= \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\vec{C}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\vec{C}_2,$$

which is the linear combination rule by setting $\vec{S}_1^* = \vec{C}_1$ and $\vec{S}_2^* = \vec{C}_2$.

Optimality of the linear combination rule: Illustration

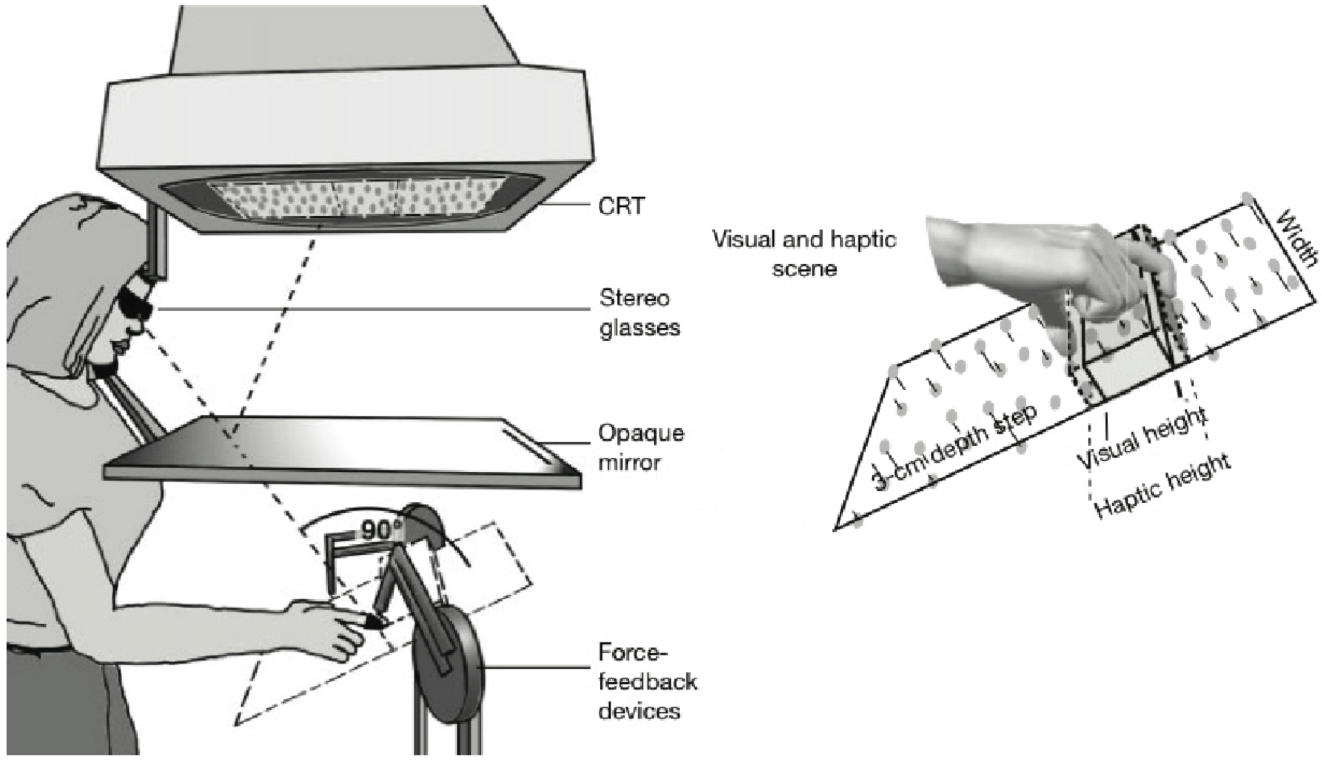


Figure 33 : The work of Ernst and Banks shows that cues are sometimes combined by weighted least squares, where the weights depend on the variance of the cues. Figure adapted from Ernst & Banks (2002).

Bayesian analysis: Weak and strong coupling

- We now describe more complex models for coupling cues from a Bayesian perspective (Clark & Yuille, 1990; Yuille & Bulthoff, 1996), which emphasizes that the uncertainties of the cues are taken into account and the statistical dependencies between the cues are made explicit.
- Examples of cue coupling, where the cues are independent, are called "weak coupling" in this framework. In the likelihood functions are independent Gaussians, and if the priors are uniform, then this reduces to the linear combination rule.
- By contrast, "strong coupling" is required if the cues are dependent on each other.

The priors: Avoiding double counting

- Models of individual cues typically include prior probabilities about $\vec{S}$. For example, cues for estimating shape or depth assume that the viewed scene is piecewise smooth. Hence it is typically unrealistic to assume that the priors $P(\vec{S})$ are uniform.
- Suppose we have two cues for estimating the shape of a surface, and both use the prior that the surface is spatially smooth. Taking a linear weighted sum of the cues would not be optimal, because the prior would be used twice. Priors introduce a bias to perception, so we want to avoid doubling this bias.
- This is supported by experimental findings (Bulthoff & Mallot, 1988) in which subjects were asked to estimate the orientation of surfaces using shading cues, texture cues, or both. If only one cue, shading or texture, was available, subjects underestimated the surface orientation. But human estimates were much more accurate if both cues were present, which is inconsistent with double counting priors (Yuille & Bulthoff, 1996).

Avoiding double counting: Experiments

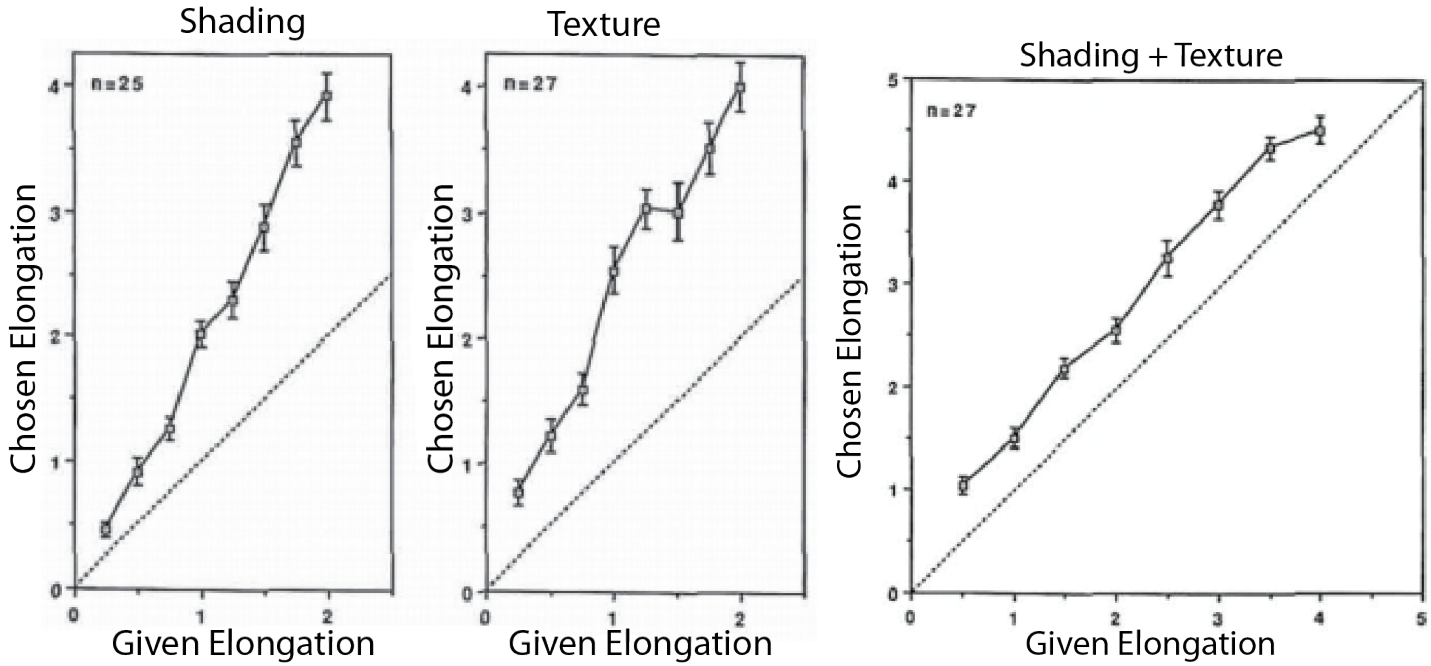


Figure 34 : Cue coupling results that are inconsistent with linear weighted average (Bulthoff et al., 1990). Left: If depth is estimated using shading cues only, then humans underestimate the perceived orientation (i.e., they see a flatter surface). Center: Humans also underestimate the orientation if only texture cues are present. Right: But if both shading and texture cues are available, then humans perceive the orientation correctly. This is inconsistent with taking the linear weighted average of the results for each cue separately. Figure adapted from Bulthoff et al. (1990).

Avoiding double counting: Probabilistic analysis (I)

- We model the two cues separately by likelihoods $P(\vec{C}_1|\vec{S}), P(\vec{C}_2|\vec{S})$ and a prior $P(\vec{S})$. For simplicity we assume that the priors are the same for each cue.
- This gives posterior distributions for each visual cue:

$$P(\vec{S}|\vec{C}_1) = \frac{P(\vec{C}_1|\vec{S})P(\vec{S})}{P(\vec{C}_1)}, \quad P(\vec{S}|\vec{C}_2) = \frac{P(\vec{C}_2|\vec{S})P(\vec{S})}{P(\vec{C}_2)}.$$

- This yields estimates of surface shape to be $\vec{S}_1^* = \arg\max_{\vec{S}_1} P(\vec{S}|\vec{C}_1)$ and $\vec{S}_2^* = \arg\max_{\vec{S}_2} P(\vec{S}|\vec{C}_2)$.

Avoiding double counting: Probabilistic analysis (II)

- The optimal way to combine the cues is to estimate $\vec{S}$ from the posterior probability $P(\vec{S}|\vec{C}_1, \vec{C}_2)$:

$$P(\vec{S}|\vec{C}_1, \vec{C}_2) = \frac{P(\vec{C}_1, \vec{C}_2|\vec{S})P(\vec{S})}{P(\vec{C}_1, \vec{C}_2)}.$$

- If the cues are *conditionally independent*, $P(\vec{C}|\vec{S}) = P(\vec{C}_1|\vec{S})P(\vec{C}_2)|\vec{S})$, then this simplifies to:

$$P(\vec{S}|\vec{C}_1, \vec{C}_2) = \frac{P(\vec{C}_1|\vec{S})P(\vec{C}_2|\vec{S})P(\vec{S})}{P(\vec{C}_1, \vec{C}_2)}.$$

Avoiding double counting: Probabilistic analysis (III)

- Coupling the cues, using the model in the previous slide, cannot correspond to a linear weighted sum, which would essentially be using the prior twice (once for each cue).
- To understand this, suppose the prior is $P(\vec{S}) = \frac{1}{Z_p} \exp\{-\frac{|\vec{S}-\vec{S}_p|^2}{2\sigma_p^2}\}$. Then, setting $t_1 = 1/\sigma_1^2, t_2 = 1/\sigma_2^2, t_p = 1/\sigma_p^2$, the optimal combination is $\vec{S}^* = \frac{t_1\vec{C}_1+t_2\vec{C}_2+t_p\vec{S}_p}{t_1+t_2+t_p}$, hence the best estimate is a linear weighted combination of the two cues $\vec{C}_1, \vec{C}_2$ and the mean $\vec{S}_p$ of the prior.
- By contrast, the estimate using each cue individually is given by $\vec{S}_1^* = \frac{t_1\vec{C}_1+t_p\vec{S}_p}{t_1+t_2+t_p}$ and $\vec{S}_2^* = \frac{t_2\vec{C}_2+t_p\vec{S}_p}{t_1+t_2+t_p}$.

Lecture 12.6

- This lecture discusses the dependencies between visual cues, and how these can be modeled by graphical models, often with causal structure.
- We also briefly discuss how the models in these lectures can fit with theories of high-level vision.

Cue dependence and causal structure (I)

- Visual cues are rarely independent.
- In the flying carpet example, the perception of depth is due to perspective, segmentation, and shadow cues interacting in a complex way. The perspective and segmentation cues determine that the beach is a flat ground plane. Segmentation cues must isolate the person, the towel, and the shadow. Then the visual system must decide that the shadow is cast by the towel and hence presumably must lie above the ground plane. These complex interactions are impossible to model using the simple conditional independent model described above.

Cue dependence and causal structure (II)

- The conditional independent model is also problematic when coupling shading and texture cues (Bulthoff & Mallot, 1988). This model for describing these experiments presupposes that it is possible to extract cues $\vec{C}_1, \vec{C}_2$ directly from the image $\mathbf{I}$ by a preprocessing step that computes $\vec{C}_1(\mathbf{I})$ and $\vec{C}_2(\mathbf{I})$.
- This requires decomposing the image $\mathbf{I}$ into texture and shading components. This decomposition is practical for the simple stimuli used in (Bulthoff & Mallot, 1988). But in most natural images, it is extremely difficult, and detailed modeling of it lies beyond the scope of this chapter.

Causal structure: Ball-in-a-box

- The "ball-in-a-box" experiments (Kersten et al., 1997) suggest that visual perception does seek to find causal relations underlying the visual cues.
- In these experiments, an observer perceives the ball as rising off the floor of the box only if this is consistent with a cast shadow.
- To solve this task, the visual system must detect the surface and the orientation of the floor of the box (and decide it is flat), detect the ball, and estimate the light source direction, and the motion of the shadow.
- It seems plausible that in this case, the visual system is unconsciously doing inverse graphics to determine the most likely three-dimensional scene that generated the image sequence.

Causal structure: Ball-in-a-box figure

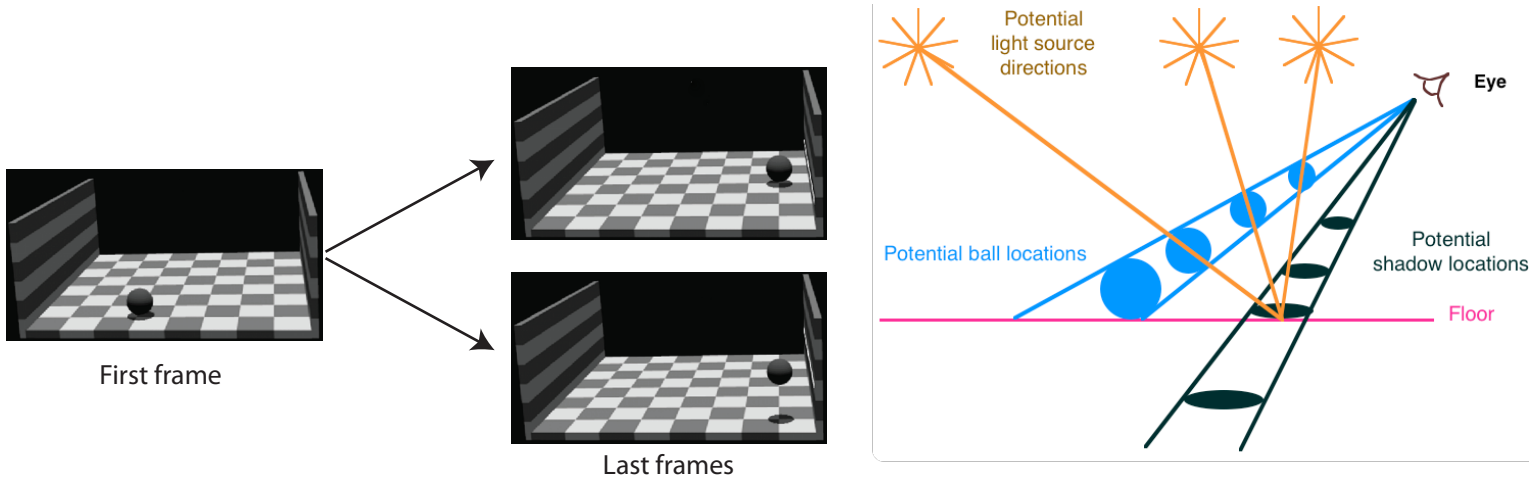


Figure 35 : In the "ball-in-a-box" experiments, the motion of the shadow affects the perceived motion of the ball. The ball is perceived to rise from the ground if the shadow follows a horizontal trajectory in the image; but it is perceived to move towards the back of the box if the shadow follows a diagonal trajectory. See `http://youtu.be/hdFCJepvJXU`. Left: The first frame and the last frames for the two movies. Right: The explanation is that the observer resolves the ambiguities in the projection of a three-dimensional scene to perceive the 3D trajectory of the ball (Kersten et al., 1997).

Directed graphical models

- Directed, or causal, graphical models (Pearl, 1988) offer a mathematical language to describe these phenomena. These are similar to the "undirected" graphical models used earlier, because the graphical structure makes the conditional dependencies between variables explicit, but the causal models differ in that the edges between nodes are directed.
- See Griffiths & Yuille (2006) for an introduction to undirected and directed graphical models from the perspective of cognitive science.

Formal directed graphical models

- *Directed graphical models* are formally specified as follows. The random variables X_μ are defined at the nodes $\mu \in \mathcal{V}$ of a graph.
- The edges $\mathcal{E}$ specify which variables directly influence each other. For any node $\mu \in \mathcal{V}$, the set of parent nodes $pa(\mu)$ are the set of all nodes $\nu \in \mathcal{V}$ such that $(\mu, \nu) \in \mathcal{E}$, where (μ, ν) means that there is an edge between nodes μ and ν pointing to node μ. We denote the state of the parent node by $\vec{X}_{pa(\mu)}$.
- This gives a local *Markov property* – the conditional distribution $P(X_\mu|\vec{X}_{/\mu}) = P(X_\mu|\vec{X}_{pa(\mu)})$, so the state of X_μ is directly influenced only by the state of its parents (note $\vec{X}_{/\mu}$ denotes the states of all nodes except for node μ). Then the full distribution for all the variables can be expressed as:

$$P(\{X_\mu : \mu \in \mathcal{V}\}) = \prod_{\mu \in \mathcal{V}} P(X_\mu|\vec{X}_{pa(\mu)}). \tag{43}$$

Directed graphical models: Divisive normalization and Bayes-Kalman

- We have already seen two examples of directed graphical models in this chapter:
- First, when we studied divisive normalization used to represent the dependencies between the stimuli, the filter responses, and the common factor.
- Second, when exploring the Bayes-Kalman filter, where the hidden state x_t at time t "causes" the hidden state x_{t+1} at time t and the observation y_t.
- Note that in some situations, the directions of the edges indicate physical causation between variables, but in others, the arrows merely represent statistical dependence. The relationship between graphical models and causality is complex and is clarified in (Pearl, 2000).

Causal structure: Taxonomy of cue interactions

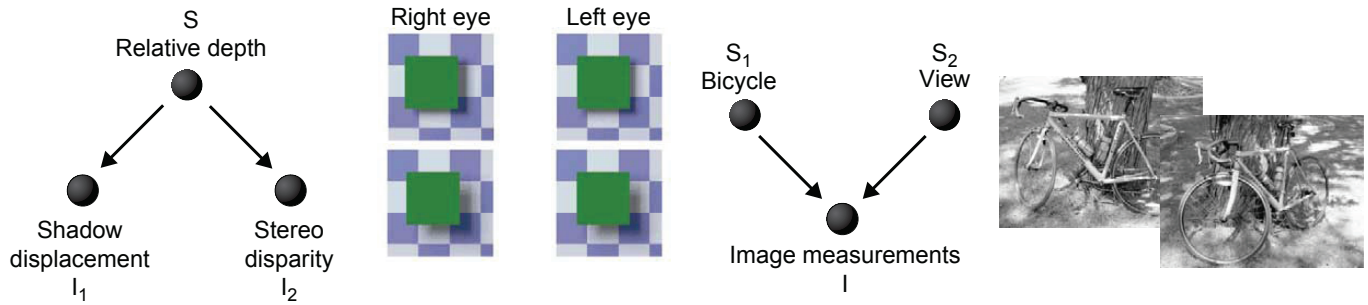


Figure 36 : Graphical models give a taxonomy of different ways in which visual cues can be combined. Left: An example of common cause. The shadow and binocular stereo cues are caused by the same event – two surfaces with one partially occluding the other. Right: The image of the bicycle is caused by the pose of the bicycle, the viewpoint of the camera, and the lighting conditions.

Graphical models and explaining away (I)

- Graphical models can be used (Pearl, 1988) to illustrate the phenomena of *explaining away*. This describes how our interpretations of events can change suddenly as new information becomes available.
- For example, suppose a house alarm A can be activated by either a burglary B or by an earthquake E. This can be modeled by $P(A|B,E)$ and priors $P(B), P(E)$ for a burglary and an earthquake. In general, the prior probability of a burglary is much higher than the prior probability of an earthquake. So if an alarm goes off, then it is much more probable to be caused by a burglary, formally $P(B|A) >> P(E|A)$. But suppose, after the alarm has sounded, you are worried about your house and check the Internet only to discover that there has been an earthquake. In this case, this new information "explains away" the alarm, so you stop worrying about a burglary.

Graphical models and explaining away (II)

- Variants of this phenomena arise in vision. Suppose you see the "partly occluded T" where a large part of the letter T is missing. In this case there is no obvious reason that part of the T is missing, so the perception may be only of two isolated segments. On the other hand, if there is a grey smudge over the missing part of the T, then most observers perceive the T directly. The presence of the smudge "explains away" why part of the T is missing.
- The Kanizsa triangle can also be thought of in these terms. The perception is of three circles partly occluded by the triangle. Hence the triangle explains why the circles are not complete. We will give a closely related explanation when we discuss model selection.

Directed graphical models and visual tasks (I)

- The human visual system performs a range of visual tasks, and the way cues are combined can depend on the tasks being performed.
- For example, consider determining the shape of a shaded surface. In most cases we need only shape from shading to estimate the shape of the surface. But occasionally we may want to estimate the light source direction.
- This can be formulated by a model $P(I|S,L)P(S),P(L)$, where I is the observed image, S is the surface shape, and L is the light source direction. $P(I|S,L)$ is the probability of generating an image I from shape S with lighting L, and $P(S),P(L)$ are prior probabilities on the surface shape and the lighting.

Directed graphical models and visual tasks (II)

- If we only want to estimate the surface shape S, then we do not care about the lighting L. The optimal Bayesian procedure is to integrate it out to obtain a likelihood $P(I|S) = \int dL P(I|S,L)P(L)$, which is combined with a prior $P(S)$ to estimate S.
- Conversely, if we only want to estimate the lighting, then we should integrate out the surface shape to obtain a likelihood $P(I|L) = \int dS P(I|S,L)P(S)$ and combine it with a prior $P(L)$.
- If we want to estimate both the surface shape and the lighting, then we should estimate them using the full model $P(I|S,L)$ with priors $P(S)$ and $P(L)$.
- "Integrating out" nuisance, or generic, variables relates to the *generic viewpoint assumption* (Freeman, 1994) which states that the estimation of one variable, such as the surface shape, should be insensitive to small changes in another variable (e.g., the lighting).

Model selection.

- Certain types of cue coupling require *model selection*.
- While some cues, such as binocular stereo and motion, are usually valid in most places of the image, other cues are only valid for subparts of each image. For example, the lighting and geometry in most images are too complex to make shape from shading a reliable cue. Also shape from texture is only valid in restricted situations.
- Similarly, the visual system can use *perspective cues* to exploit the regular geometrical structure in the ball-in-a-box experiments. But such cues are only present in restricted classes of scenes, which obey the "Manhattan world" assumption. These cues will not work in the jungle. These considerations show that cue combination often requires *model selection* in order to determine in what parts of the image, if any, the cues are valid.

Model selection illustration

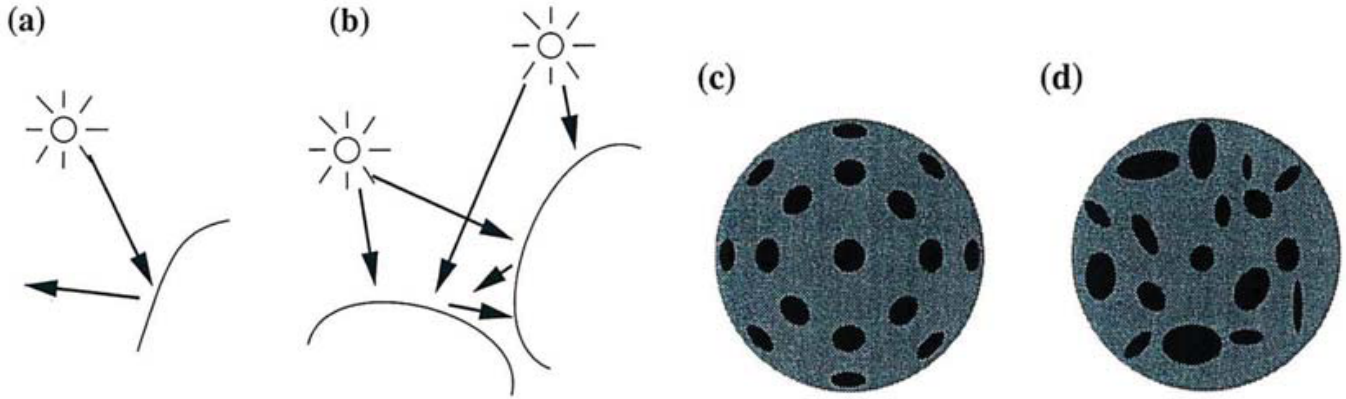


Figure 37 : Model selection may need to be applied to decide if a cue can be used. Shape from shading cues will work for case (a) because the shading pattern is simply due to a smooth convex surface illuminated by a single source. But for case (b) the shading pattern is complex – due to mutual reflection between the two surfaces – and so shape from shading cues will be almost impossible to use. Similarly, shape from texture is possible for case (c), because the surface contains a regular texture pattern, but is much harder for case (d), because the texture is irregular.

Model selection examples

- Model selection also arises when there are several alternatives ways to generate the image.
- By careful experimental design, it is possible to adjust the image so that small changes shift the balance between one interpretation and another.
- Examples include the experiments with two rotating planes that can be arranged to have two competing explanations (Kersten et al., 1992). With slight variations to the transparency cues, the two surfaces can be seen to move rigidly together or to move independently (see `http://youtu.be/gSrUBpovQdU`).

Model selection: shadows and specularity

- A classic experiment (Blake & Bulthoff, 1990) studies human percption using a sphere with a Lambertian (diffuse) reflection function, which is viewed binocularly.
- A specular component is adjusted so that it can lie in front of the sphere, between the center and the sphere, or at the center of the sphere.
- If the specularity lies at the center, then it is perceived to be a transparent light bulb.
- If the specularity is placed between the center and the sphere, then the sphere is perceived to be shiny and specular.
- If the specularity lies in front of the sphere, then it is perceived as a cloud floating in front of a matte (Lambertian sphere).
- This is interpreted as strong coupling using model selection (Yuille & Bulthoff, 1996).

Model selection examples: Illustration

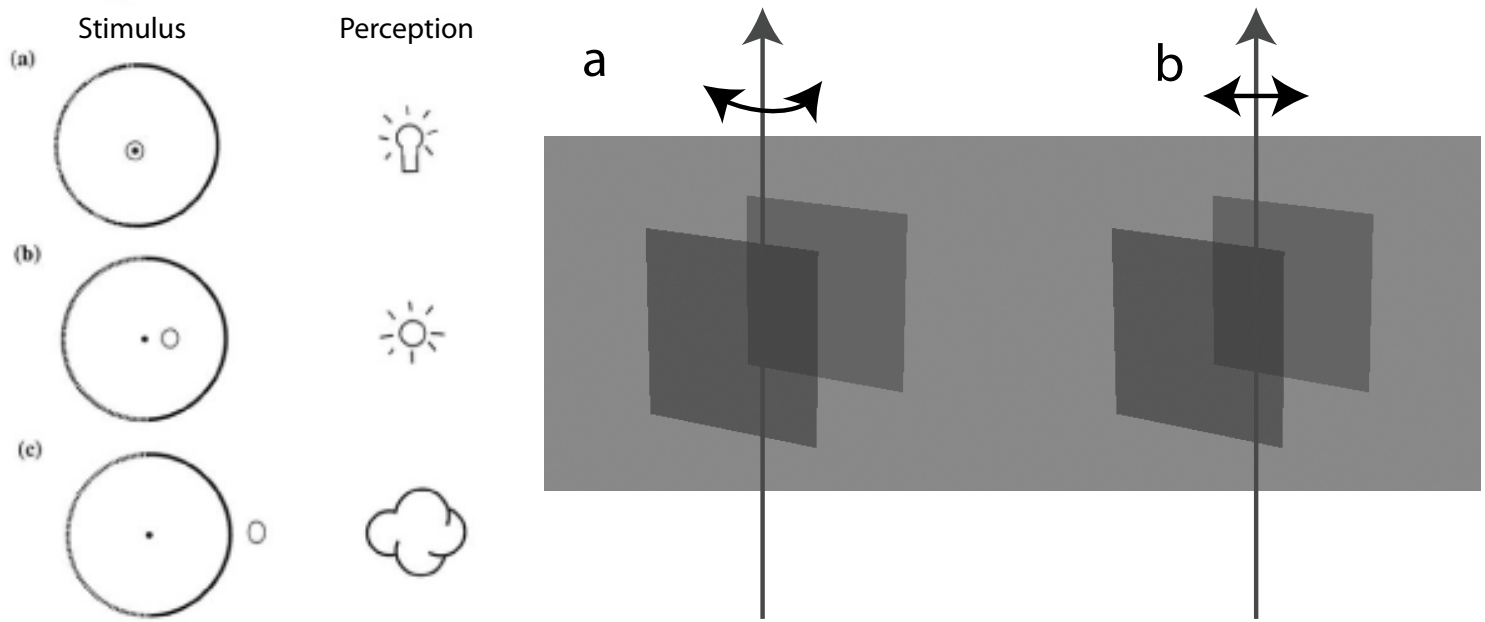


Figure 38 : Examples of strong coupling with model selection. Left: A sphere is viewed binocularly, and small changes in the position of the specularity lead to very different percepts (Blake and Bülthoff, 1990). Right: Similarly altering, the transparency of the moving surfaces can make the two surfaces appear to rotate either rigidly together or independently.

Model selection and explaining away

- Model selection can also give an alternative explanation for "explaining away"
- For example, consider two alternative models for partially occluded T
- The first model is of two individual segments plus a smudge region. The second is a T that is partially hidden by a smudge. The second model is more plausible since it would be very unlikely, an accidental viewpoint (or alignment), for the smudge to happen to cover the missing part of the T, unless it really did occlude it.
- A similar argument can be applied to the Kanizsa triangle. One interpretation is three circles partly occluded by a triangle. The other is three partial circles arranged so that the missing parts of the circles are aligned. The first interpretation is judged to be most probable.

Flying carpet revisited

- Like Kersten's ball-in-a-box experiments, the flying carpet illusion requires estimating the depth and orientation of the ground plane (i.e., the beach), segmenting and recognizing the woman and the towel she is standing on, detecting the shadow, and then using the shadow cues, which requires making some assumptions about the lighting, to estimate that the towel is hovering above the ground.
- This is a very complex way to combine all the cues in this image. Observe that it relies on the generic viewpoint assumption, in the sense that it is unlikely for there to be a shadow of that shape in that particular part of the image unless if was cast by some object. The real object that cast the shadow (the flag) is outside the image, so the visual system "attaches" the shadow to the towel, which then implies that the towel must be hovering off the ground.

Examples of strong coupling

We now give two examples of strong coupling. The first example deals with coupling different modalities, while the second example concerns the perception of texture.

Multisensory cue coupling

- Human observers are sensitive to both visual and auditory cues.
- Sometimes these cues have a common cause, e.g., you see a barking dog. But in other situations, the auditory and visual cues have different causes, e.g., a nearby cat moves and a dog barks in the distance.
- Ventriloquists are able to make the audience think that a puppet is talking by making it seem that visual cues (the movement of the puppet's head) and auditory cues (words spoken by the ventriloquist) are related. The ventriloquism effect occurs when visual and auditory cues have different causes – and so are in conflict – but the audience perceives them as having the same cause.

Multisensory cue coupling: The model (I)

- We describe an ideal observer for determining whether two cues have a common cause or not (Kording et al., 2007), which gives a good fit to experimental findings.
- The model is formulated using a meta-variable C, where $C = 1$ means that the cues x_A, x_V are coupled.
- More precisely, they are generated by the same process S by a distribution $P(x_A, x_V|S) = P(x_A|S)P(x_V|S)$.
  $P(x_A|S)$ and $P(x_V|S)$ are normal distributions $N(x_A|S, \sigma_A^2)$, $N(x_V|S, \sigma_V^2)$ – with the same mean S and variances σ_A^2, σ_V^2.
- It is assumed that the visual cues are more precise than the auditory cues, so that $\sigma_A^2 > \sigma_V^2$. The true position S is drawn from a probability distribution $P(S)$, which is assumed to be a normal distribution $N(0, \sigma_p^2)$.

Multisensory cue coupling: The model (II)

- $C = 2$ means that the cues are generated by two different processes S_A and S_B.
- In this case, the cues x_A and x_V are generated respectively by $P(x_A|S_A)$ and $P(x_V|S_V)$, which are both Gaussian $N(S_A, \sigma_A^2)$ and $N(S_V, \sigma_V^2)$. We assume that S_A and S_V are independent samples from the normal distribution $N(0, \sigma_p^2)$.
- Note that this model involves model selection, between $C = 1$ and $C = 2$, and so, in vision terminology, it is a form of strong coupling with model selection (Yuille & Bulthoff, 1996).

Multisensory cue coupling: Illustration

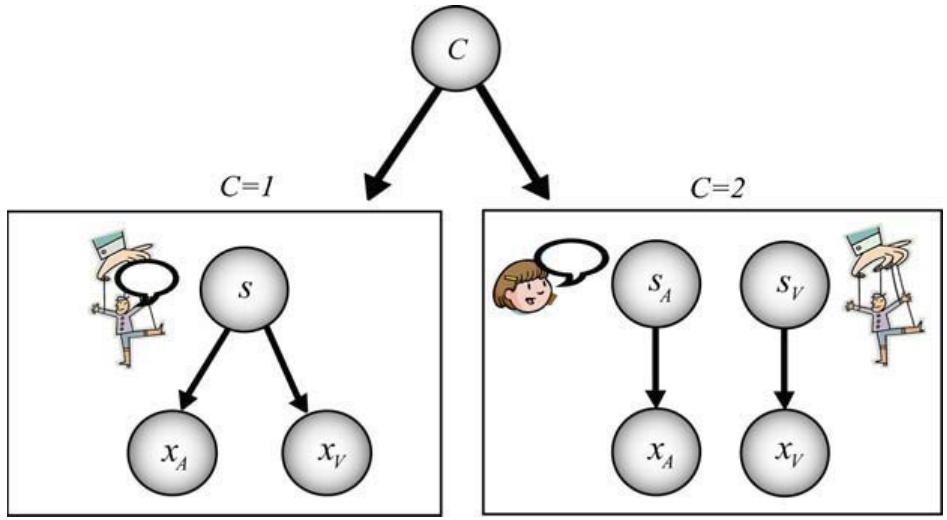


Figure 39 : The subject is asked to estimate the position of the cues and to judge whether the cues are from a common cause – i.e., at the same location – or not. In Bayesian terms, the task of judging whether the cause is common can be formulated as model selection: are the auditory and visual cues more likely generated by a single cause (left) or by two independent causes (right)? Figure adapted from Kording et al. (2007).

Multisensory cue coupling: Comparison with experiments (I)

- This model was compared to experiments in which brief auditory and visual stimuli were presented simultaneously, with varying amounts of spatial disparity.
- Subjects were asked to identify the spatial location of the cue and/or whether they perceived a common cause (Wallace et al., 2004).
- The closer the visual stimulus was to the audio stimulus, the more likely subjects would perceive a common cause.
- In this case subjects' estimate of the stimuli's position was strongly biased by the visual stimulus (because it is considered more precise with $\sigma_V^2 > \sigma_A^2$).
- But if subjects perceived distinct causes, then their estimate was pushed away from the visual stimulus, and exhibited *negative bias*.

Multisensory cue coupling: Comparison with experiments (II)

- Körding et al. (2007) argue that this negative bias is a selection bias stemming from restricting to trials in which causes are perceived as being distinct.
- For example, if the auditory stimulus is at the center and the visual stimulus at 5 degrees to right of center, then sometimes the (very noisy) auditory cue will be close to the visual cue and hence judged to have a common cause, while in other cases, the auditory cause is farther away (more than 5 degrees).
- Hence the auditory cue will have a truncated Gaussian (if judged to be distinct) and will yield negative bias.

Multisensory cue coupling: Results and figure

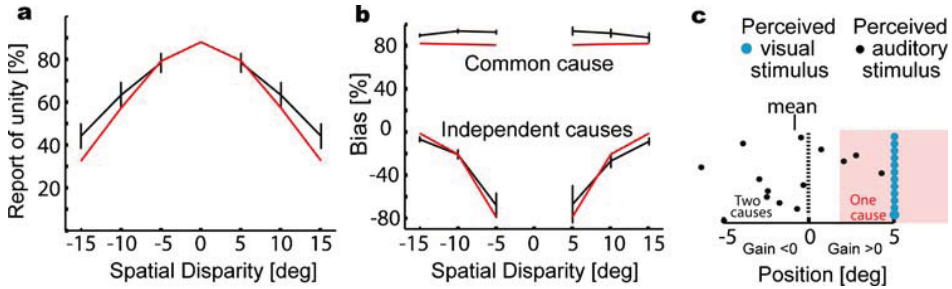


Figure 40 : Reports of causal inference. (a) The relative frequency of subjects reporting one cause (black) is shown, with the prediction of the causal inference model (red). (b) The bias, i.e., the influence of vision on the perceivedauditory position, is shown (gray and black). The predictions of the model are shown in red. (c) A schematic illustration explaining the finding of negative biases. Blue and black dots represent the perceived visual and auditory stimuli, respectively. In the pink area, people perceive a common cause. Reprinted with permission from Kording et al. (2007)

Multisensory cue coupling: The mathematics (I)

More formally, the beliefs $P(C|x_A, x_V)$ in these two hypotheses $C = 1, 2$ are obtained by summing out the estimated positions s_A, s_B of the two cues as follows:

$$P(C|x_A, x_V) = \frac{P(x_A, x_V|C)P(C)}{P(x_A, x_V)}$$

$$= \frac{\int dS P(x_A|S)P(x_V|S)P(S)}{P(x_A, x_V)}, \quad \text{if } C = 1,$$

$$= \frac{\int\int dS_A dS_V P(x_A|S_A)P(x_V|S_V)P(S_A)P(S_V)}{P(x_A, x_V)}, \quad \text{if } C = 2.$$

Multisensory cue coupling: The mathematics (II)

- There are two ways to combine the cues. The first is model selection. This estimates the most probable model $C^* = \arg\max P(C|x_V, x_A)$ from the input x_A, x_V and then uses this model to estimate the most likely positions s_A, s_V of the cues from the posterior distribution:

$$P(s_V, s_A) \approx P(s_V, s_A|x_V, x_A, C^*) = \frac{P(x_V, x_A|s_V, s_A, C^*)P(s_V, s_A|C^*)}{P(x_V, x_A|C^*)}.$$

- The second way to combine the cues is by *model averaging*. This does not commit itself to choosing C^* but instead averages over both models:

$$P(s_V, s_A|x_V, x_A) = \sum_C P(s_V, s_A|x_V, x_A, C)P(C|x_V, x_A)$$

$$= \sum_C \frac{P(x_V, x_A|s_V, s_A, C)P(s_V, s_A|C)P(C|x_V, x_A)}{P(x_V, x_A|C)},$$

where $P(C = 1|x_V, x_A) = \pi_C$ (the posterior mixing proportion).

Multisensory cue coupling: Extension

- Natarajan et al. (2008) showed that a variant of the model could fit the experiments even better.
- They replaced the Gaussian distributions with alternative distributions that are less sensitive to rare events. Gaussian distributions are non-robust because the tails of their distributions fall off rapidly, which gives very low probability to rare events.
- More precisely Natarajan et al. (2008) assumed that the data is distributed by a mixture of a Gaussian distribution, as above, and a uniform distribution (yielding longer tails).
- More formally, they assume $x_A \ \pi N(x_A : s_A, \sigma_A^2) + \frac{(1-\pi)}{r_l}$ and $x_V \ \pi N(x_V : s_V, \sigma_V^2) + \frac{(1-\pi)}{r_l}$, where π is a mixing proportion, and $U(x) = 1/r_1$ is a uniform distribution defined over the range r_1.

Homogeneous and isotropic texture

- The second example is by Knill and concerns the estimating of orientation in depth (slant) from texture cues (Knill, 2003).
- There are alternative models for generating the image, and the human observer must infer which is most likely. In this example, the data could be generated by isotropic homogeneous texture or by homogeneous texture only.
- Knill's finding is that human vision is biased to interpret image texture as isotropic, but if enough data are available, the system turns off the isotropy assumption and interprets texture using the homogeneity assumption only.

Homogeneous and isotropic texture: Illustration

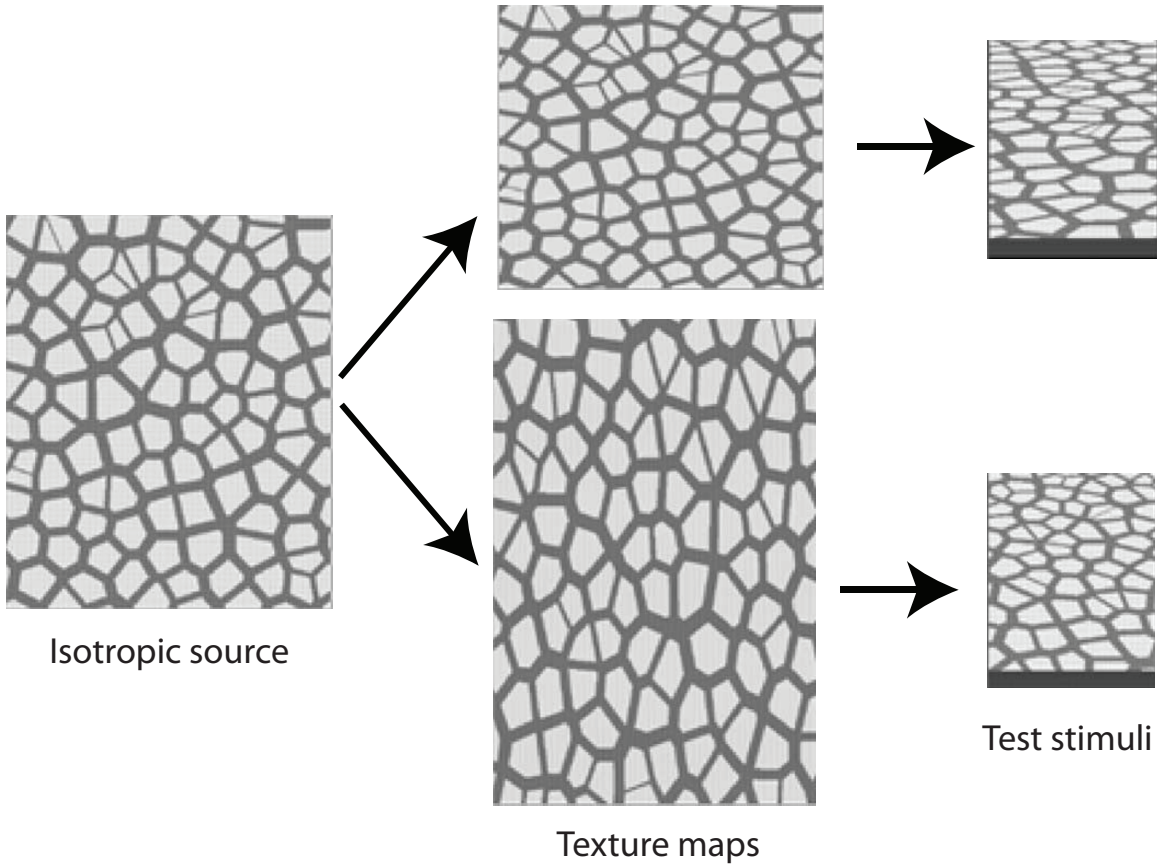


Figure 41 : Generating textures that violate isotropy. An isotropic source image is either stretched (top middle) or compressed (bottom middle), producing texture maps that get applied to slanted surfaces shown on the right. A person that assumes surface textures are isotropic would overestimate the slant of the top stimulus and underestimate the slant of the bottom one. Figure adapted from from Knill (2003).

Homogeneous and isotropic texture: Theory (I)

- The posterior probability distribution for S is given by:

$$P(S|I) = \frac{P(I|S)P(S)}{P(I)}, \quad P(I|S) = \sum_{i=1}^{n} \phi_i P_i(I|S),$$

where ϕ_i is prior probability of model i, and $p_i(I|S)$ is corresponding likelihood function.

- More specifically, texture features T can be generated by either an isotropic surface or a homogeneous surface. The surface is parameterized by tilt and slant σ, τ. Homogenous texture is described by two parameters α, θ, and isotropic texture is a special case where $\alpha = 1$. This gives two likelihood models for generating the data:

$$P_h(T|(\sigma,\tau),\alpha,\theta), \quad P_i(T|(\sigma,\tau),\theta)$$

Here, $P_i(T|(\sigma,\tau),\theta) = P_h(T|(\sigma,\tau),\alpha = 1,\theta)$.

Homogeneous and isotropic texture: Theory (II)

- Isotropic textures are a special case of homogenous textures.
- The homogeneous model has more free parameters and hence has more flexibility to fit the data, which suggests that human observers should always prefer it. But the Occam factor (MacKay, 2003) means that this advantage will disappear if we put priors $P(\alpha)P(\theta)$ on the model parameters and integrate them out. This gives:

$$P_h(T|(\sigma,\tau)) = \int\int d\alpha d\theta P_h(T|(\sigma,\tau),\alpha,\theta),$$

$$P_i(T|(\sigma,\tau)) = \int d\theta P_h(T|(\sigma,\tau),\theta).$$

- Integrating over the model priors smooths out the models. The more flexible model, P_h, has only a fixed amount of probability to cover a large range of data (e.g., all homogeneous textures) and hence has lower probability for any specific data (e.g., isotropic textures).

Homogeneous and isotropic texture: The mathematics

- Knill describes how to combine these models using model averaging. The combined likelihood function is obtained by taking a weighted average:

$$P(T|(\sigma,\tau)) = p_h P_h(T|(\sigma,\tau)) + p_i P_i(T|(\sigma,\tau)), \tag{44}$$

where (p_h, p_i) are prior probabilities that the texture is homogeneous or isotropic. We use a prior $P(\sigma,\tau)$ on the surface and finally achieve a posterior:

$$P(\sigma,\tau|I) = \frac{P(I|(\sigma,\tau))P(\sigma,\tau)}{P(I)}. \tag{45}$$

- This model has a rich interpretation. If the data are consistent with an isotropic texture, then this model dominates the likelihood and strongly influences the perception. Alternatively, if the data are consistent only with homogeneous texture, then this model dominates. This gives a good fit to human performance (Knill, 2003).

Summary and the relations of early and high-level vision

- These lectures have given a rapid tour of early vision. We have provided a modern perspective and conceptualization of early vision in terms of probabilistic graphical models. In this final section, we briefly mention how early vision relates to high-level vision.
- In particular, we will sketch the relations to three of the dominant frameworks for vision:
  1. Marr's theory of vision (Marr, 1982)
  2. Hierarchical theories of vision such as HMax (Riesenhuber & Poggio, 1999)
  3. The "analysis by synthesis" framework (Mumford, 1992; Lee & Mumford, 2003).
- The first two of these frameworks are "feedforward" in the sense that visual processing proceeds bottom-up from low-level to mid-level and ultimately to high-level. By contrast "analysis by synthesis" emphasizes the role of top-down, or "feedback," processing. Other researchers, e.g., Ullman (1995), Epshtein et al. (2008), have proposed theories that include bottom-up and top-down processing.

Relationship to Marr's framework

- Marr's framework (Marr, 1982) for vision is feedforward. He proposed that visual processing constructs a series of representations: (1) the primal sketch; (2) the 2-1/2 D sketch; and (3) a 3D model. Roughly speaking, the first two representations involve low-level and mid-level vision, while the third corresponds to high-level vision.
- Many of the models described in this chapter would fit nicely as components of Marr's framework. Edge detection and the weak membrane model are both ways to construct the primal sketch. Modules, like binocular stereo, could be used to construct the 2-1/2D sketch.
- There are, however, several differences. Marr's framework was not formulated in probabilistic terms and does not address issues such as strong coupling between visual cues. It pays little attention to top-down processing and concentrates largely on the "computational level," rather than on neural implementations.

Relationship to theories of the ventral stream and HMax

- This class of theory (cf. Riesenhuber & Poggio, 1999) models the ventral stream (visual areas V1, V2, IT) by a hierarchical neural network in which as we ascend the hierarchy, the receptive fields of neurons are tuned to increasingly complex visual structures but are increasingly less sensitive to the precise positions of the input features.
- This theory focuses on object detection and recognition. It starts from the models of simple and complex cells and extends this idea to build a hierarchy of cells.
- This theory is strongly motivated by properties of the visual system. At the higher levels it emphasizes the importance of learning. The theory is predominantly feedforward.

Analysis by synthesis theories of vision

- By contrast with the first two frameworks, analysis by synthesis emphasizes both feedforward and feedback visual processing. It relates to the idea that visual is inverse computer graphics and that visual processing should discover the causal factors of images. The framework for analysis by synthesis is based on pattern theory (Grenander, 1976, 1978) which is formulated probabilistically.
- Mumford (Mumford, 1992) argued for the importance of top-down processing in vision, citing the large number of backprojections in the cortex that are reminiscent of the "analysis by synthesis" approach. This class of theories has been developed in (Lee & Mumford, 2003) and related ideas appear in Ullman (1995), and Epshtein et al. (2008). The theory of Rao and Ballard (1999), who suggest that top-down processing can be used to implement predictive processing, somewhat similar to the Bayes-Kalman models briefly discussed in these lectures.

Early and high-Level vision: Strong coupling

- The situation is more complicated for the third type of framework, which combines bottom-up and top-down processing. But this can also be formulated by extending the graphical model theories we have dicussed so that they are hierarchical. In these models the low-level nodes represent elementary features, such as edges, and the intermediate-level nodes represent compositions of the lower-level features, such as the grouping of edges to form longer segments, or the grouping of parallel line segments. These intermediate-level structures are combined to form larger structures, such as objects and object parts.
- These theories are sometimes called compositional (Geman et al., 2002; Zhu et al., 2011) because they build objects by composition and they are closely related to stochastic grammars (Zhu & Mumford, 2007; Mumford & Desolneux, 2010). For these classes of theories, the early and high levels of vision are strongly coupled (similar to strong coupling of cues). Inference can be performed either bottom-up, where it is driven directly by the input image, and low-level hypotheses are combined to make hypotheses for more complex structures, or top-down, where high-level hypotheses drive the computation.