2. Building a space map

2.1 Ordinary space

The word "ordinary" may be misleading as it often appears in a demeaning way, to describe something as trivial or uninteresting. In that sense, there is nothing ordinary about ordinary space. Philosophers have argued for over two thousand years on the nature of space. Some of the debate centered on whether space, as we think of it, really exists at all, or is it just a product of our minds. Perhaps a more approachable question is whether space exists independent of what fills it. Does "empty space" have any meaning at all?

From grade school we are exposed to notions like forces acting across a distance. We have learned from Isaac Newton that the motion of the earth around the sun can be accounted for by assuming that earth and sun pull on each other in direct proportion to their masses and in inverse proportion to the square of their distance. What is most intriguing is that the force that planets and stars exert on each other supposedly acts across vast regions of empty space. On a smaller scale, we all have experienced the force that a magnet exerts on another magnet across a distance. We are so used to these concepts that we do not question them. They seem natural and reasonable. Yet, these concepts were foreign to the very scientists that developed the foundations of modern physics. The concept of empty space was particularly hard to accept. So hard that Gottfried Leibniz, the mathematician who laid the foundations of infinitesimal calculus in the late 17^{th} century, constructed the theory of a universe filled with special entities, the monads, without any space between them. The idea of a wave is so connected to the undulatory motion of a body of water or air, that until recently physicists were convinced that light and other electromagnetic waves propagated within an invisible and mysterious substance called "ether". It took a long time and some crucial experiments to accept that a wave of pure energy may indeed travel across empty space.

Another critical concept is that of absolute space. Is there a point of view in which space can be considered as standing still? Modern physics teaches otherwise. When we sit on an airplane, the video screen mounted on the ceiling of the economy class section is fixed. The space around us in the cabin has many fixed points. However, to an observer on earth, these points are rapidly translating with the airplane. Over any interval, this observer sees these points as forming line segments. You can see one of the most compelling effects of changing viewpoint by placing a

fixed camera on long pole so it looks down at a merry-go-round. Long ago, we (RS and SMI) left the lab that we shared at Cambridge, Massachusetts and walked down the Charles river to a park that happened to have a merry-go-round. We brought a few tennis balls and sat across each other on the merry-go-round. As the carousel spinned, we threw balls at each other. Instead of moving straight, we saw the ball curving opposite to the motion of the carousel. And yet, if we had a video camera, it would show that the ball moved along a perfectly rectilinear path. The reader is encouraged to see examples of these movies and animations by searching the web for "Coriolis effect".

While we are interested in presenting some of the mathematical foundations of the concept of space, we will not dwell on the rich philosophical debate on the topic. The interested reader can find a concise summary of this debate, in relation to Neuroscience, in the first chapter of "The Hippocampus as a cognitive map" by John O'Keefe and Lynn Nadel (1978). Instead, we will take a rather pragmatic approach by developing a detailed mathematical model of a simplified space and of a similarly simplified visual system[i].

We want to represent some of the computational tasks from the perspective of a hypothetical organism moving inside an environment and receiving incomplete and distorted images of its surroundings. This is a simplification of the gerbil's viewpoint of the previous chapter. How can this organism's brain develop a sense of space? Or, in more modern terms, how does it develop an internal model of the space by combining sensor data with movement commands? We will address this question from the viewpoint of a Mongolian gerbils that we will call "G".

2.2 A simple model

The first challenge for G's brain as it moves inside an environment populated by various landmarks is to construct a representation of space from information captured by its eyes. The eye is a complex organ, where images are projected on a curved surface with an uneven distribution of neural elements that transform photons into electrical impulses. We do not want to develop a realistic model of this wonderful neural and optical machinery. We only wish to capture the idea that the information about space comes from projections on a curved element. So, let us begin by simplifying the dimensionality of the problem. Ordinary space is three-dimensional and the surfaces of the eyes are two-dimensional. The math becomes manageable if we assume that the space is two dimensional – G is a flat gerbil - and, accordingly, that G's idealized eye is a

circular, 1-dimensional line. This geometrical expedient, which we will shamelessly call "eye", is depicted in Fig. 2.1.

Figure 2.1. A simple eye. This is a projection model that maps points in the plane (L) into images over a one-dimensional "retina" in the shape of a circle centered at O. The heading direction is indicated by the arrow attached to N. The image of L on this retina is the arc NL'. Assuming that the radius ($\overline{OL'}$) has unit length, this arc measures in radians the angle NOL.

In our model, the two-dimensional space is populated by "landmarks", that is, by points of particular significance. As G moves around, the landmarks are projected on the circumference of its eye. Now, we need to introduce something that makes this picture less "even", less symmetric. Ordinary space is isotropic: all points are identical and all directions are equivalent. However, our view is oriented because we have a body and we face the direction in which we move. That is, to us and to our eye all directions are not equivalent. G's heading direction is indicated in the figure by an arrow that intersects the eye at a point N, which stands for "North". If L is a landmark in the external space, the projection of the landmark on the simplified 1-dimensional "retina" (the circle) is obtained by tracing the segment $\overline{LO}$ joining the landmark to the center of the circle and by taking the intersection, *L'*, of the segment with the circle. The point *L'* is for our purposes a perspective image of the landmark.

2.3 Points and lines

So far, we have not introduced a metric notion, such as a measure or distance. At a very basic level, geometry, and particularly projective geometry, does not use distances. Projective geometry is only about objects, such as points, lines and surfaces. Much of its original raison d'être is the need to represent three-dimensional reality within the confines of two-dimensional paintings. Here, we consider perspective in the opposite (or "inverse") sense. We want to regenerate the reality of space from lower-dimensional pictures in our eyes. And we want to see this reconstruction as a combination of senses and motion. This is indeed the way in which the neurons in the hippocampus and in its main input structure, the entorhinal cortex behave: they combine visual memory and self-motion information for generating a neural activity that code for the position of the body in the environment. But before developing a quantitative theory we may ask what information can be extracted about the environment, without recourse to metric concepts. Is it possible for G's brain to understand that three or more distinct points are on the

same straight line? The task would be easy if G could measure the distances between these points. G could take advantage of the well known fact that the shortest distance between two points is measured along the straight segment that joins them. But what if one does not know how to measure distances? Then, G can make use of a simpler notion, the notion of "order". This is the intuitive idea of a point sitting "in between" two others. The order relation was formalized first by Moritz Pasch and subsequently by David Hilbert in a set of axioms known as Hilbert axioms. Can G's brain exploit the order of images on the sensor circle to infer something about the structure of the external space?

G can use the order relation, because it has a motor system that allows him to move around its environment. Let us start by accepting that given two points A and B, we can find a third point C such that C is between A and B. Then, the segment $\overline{AB}$ is simply the collection of all points that are between the two extremities, A and B. Points that belong to the same segment are said to be *collinear*. Consider now the situation depicted in Fig. 2.2. There are four collinear landmarks that are ordered as A, B, C, D, or equivalently as D, C, B, A. We say that A, B, C, D and D, C, B, A are equivalent orders because they only differ by the "reading direction" that is by a reflection. In this sense A, B, C, D and, say, A, C, B, D are not equivalent orders. The same order relation is consistently present in the projected images A', B', C', D'. This is true for almost any position and heading of the eye. We say "almost" because one must only exclude the "singular" sensor-landmarks configurations at which A, D and O (the center of projections) are collinear. There, all landmarks project onto the same image. However, if the landmarks lie on a straight line, the order of their projections is never altered. Collinearity is preserved as G moves around. Conversely, if collinearity is preserved as G's position changes, G's brain can conclude, without need for measures of length or distance, that the landmarks lie on the same straight segment. This is of fundamental importance, because we have now derived the notion of straightness of a line without using any metric concept of length. As we shall see later, the collinearity relation gives us information on the *affine* structure of the external space.

Figure 2.2. Recognizing straightness. If the four points are on a common straight line, the projections preserve the order relation as the observer moves in the environment. There is only a complete reversal, a "mirror symmetry" when the projections cross the midline.

In our example, the eye is a 1-dimensional circle while the external space is 2-dimensional. There is an imbalance of dimensions and this imbalance is reflected by order relations of the projected

images, which change with the position of the eye in space. In Fig. 2.3, the four landmarks are not collinear. We may place them in different orders over different curved lines passing through all the landmarks. As a consequence, the relative order of their images changes for different positions of G relative to the landmarks. This is a cue that G may use to establish that the external world has more than one dimension and that the landmarks are not placed along a straight line. Summing up, it is possible to extract important information about space based only on relations of order between projected points, without recourse to metric operations. However, as we will see next, if one can measure lengths and distances one can learn more about the structure of external space.

Figure 2.3. Recognizing straightness. The images of the four non collinear landmarks are ordered in a way that depends on the observer's location and orientation.

2.4 Distance and coordinates

Let us look again at G's simplified eye in Fig. 2.1. It has three particularly important points: the center, O, the "North pole", N, and the "South pole", S. The two poles break the symmetry of the circle. One may say that the poles exist to signify that all directions are not equivalent.

Animals, as well as most human-made vehicles, have a front and a back. Front is sometimes, but not always, defined by the location of the eyes. With some exceptions the orientation of the eyes corresponds to the preferred direction of motion: it is safer to advance where one can see. This forward or frontal direction is what we call "heading". It defines the point N and its opposite, S. Both the anatomical structure and the behavioral preference to move forward contribute to establish a set of particular points on the optical sensor.

Consider two additional stipulations. First, as G moves forward without changing direction, stationary landmarks that project on one side of the $\overline{NS}$ axis will continue to do so. This is simply because a stationary point in a flat environment moves with respect to us parallel to the $\overline{NS}$ axis. If we see a point crossing this axis as we move, G may safely conclude that the corresponding landmark is not at rest.

Second, our movements do not affect the state of the objects around us, unless we come in physical contact with them. Things do not get smaller or bigger because we move. While this is

entirely obvious, it has some profound consequences. The perceptual separation between "us" and "environment" is essential for our understanding that the space around us is Euclidean. As G moves, it performs two operations on the positions of the objects relative to itself: translations and rotations. Or - better said - objects that are at rest in the environment rotate and translate relative to G. And G's brain can safely assume that the objects do not change in size or shape. Two objects with the same shape and size are said to be *congruent*, and a transformation that preserves shape and size is called an *isometry*. A related observation is that an object that G is not contacting and is at rest before G starts moving will likely remain at rest. Therefore, in constructing an internal model of the environment, G's brain can take these simple facts (or axioms) into account for extracting spatial information from sensory-motor data.

As G moves forward, the projections of the external landmarks change their position on the eye. G knows that the space around it is more than one-dimensional because it observed that the order of fixed landmark projections over the eye may change as it moves. Now G needs to construct a representation of the landmarks as they are located in the external space. Like the gerbils in Collett's experiments in the previous chapter, G wants to form an extrinsic representation: a representation obtained from its own motion but one that remains invariant as G moves. To this end, G's brain carries out two concurrent operations: a) keep track of G's location and b) estimate the distances between G and the landmarks. By combining these two operations the brain, as a sailor tracking along a coast line, builds and maintains a stable representation of the world.

We begin by establishing a measure of distance that will place the scale of the environment in relation to G's own scale. We associate each landmark projection L' (Fig. 2.1) with a number expressing the length of the arc NL' in units of radius length. This corresponds to measuring the angle $L'ON$ in radians. Call this angle ξ. We now construct a Cartesian coordinate system $Ox^o y^o$ centered on the origin of the circle (Figure 2.4 - Left). The origin, *O*, is the center of the eye, the axis y^O points in the north direction and the axis x^O points to the right (toward the local "east"). The superscripts refer to the particular origin to which the axes are attached. The concept of Cartesian coordinates is a familiar one and does not need to be discussed here in more detail. Its critical importance lies on the possibility to calculate distances between points using Pythagoras' theorem. If two points $P_1 = [x_1, y_1]^T$ and $P_2 = [x_2, y_2]^T$ are given in terms their Cartesian coordinates[ii], their distance, according to Pythagoras is

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.1}$$

Let us begin by assuming that we have an estimate of where we are with respect to a fixed point M in our internal model of the space (Figure 2.4 Right). We also assume that our internal representation of space is constructed as a 2-dimensional Cartesian coordinate system $Mx^M y^M$. These assumptions will be discussed further below. Our estimated position in this coordinate system is $\mathbf{r}_0^M = \left[x_O^M, y_O^M \right]^T$. Figure 2.4A illustrates the view in the sensor's reference frame. At all times we know the value ξ_L associated with the landmark L. This value is $O \le \xi_L < 2\pi$. The projection of the landmark over the sensor is captured by a function that maps the coordinates of the landmark $\left[x_L^O, y_L^O \right]^T$ to the angle/arc-length ξ_L:

$$\xi_L = \arctan\left(-\frac{x_L^O}{y_L^O} \right) = -\arctan\left(\frac{x_L^O}{y_L^O} \right) \tag{2.2}$$

Figure 2.4. Space representation based on metric information. (Left) A Cartesian coordinate system, centered on the eye (O) yields ordered pairs of numbers to identify the location of the landmark relative to the observer. (Right) A second coordinate frame, centered on a fixed point in the environment provides an allocentric reference, the observer can recover the location of the landmark in this stationary framework by combining the egocentric representation with a record of motions from M to its current location.

In more concise and general (but less informative) terms we see that this is a function mapping two spatial coordinates into a single sensor coordinate:

$$\xi = f(x_L^O, y_L^O) \tag{2.3}$$

This function is a non-linear coordinate transformation, with multiple points mapping to the same projection. Because it maps two variables into one, it does not have a unique inverse. This is another way to state that all points that are collinear with the segment $\overline{OL}$ map to the same sensor coordinate ξ_L. Points on $\overline{OL}$ are equivalent[iii] with respect to the coordinate transformation. We obtain a linear transformation by taking the temporal derivative of Eq. (2.3), that is by computing how the velocity of a point in space translates into the velocity of its projection on the "retina":

$$\dot{\xi} \equiv \frac{d\xi}{dt} = \frac{\partial \xi}{\partial x_L^0}\dot{x}_L^0 + \frac{\partial \xi}{\partial y_L^0}\dot{y}_L^0 = \begin{bmatrix} \frac{\partial \xi}{\partial x_L^0} & \frac{\partial \xi}{\partial y_L^0} \end{bmatrix} \begin{bmatrix} \dot{x}_L^0 \\ \dot{y}_L^0 \end{bmatrix}. \tag{2.4}$$

The 1x2 matrix $\mathrm{J}(x_L^0, y_L^0) = \begin{bmatrix} \frac{\partial \xi}{\partial x_L^0} & \frac{\partial \xi}{\partial y_L^0} \end{bmatrix}$ is the *Jacobian* of the coordinate transformation, which results in:

$$\dot{\xi} = \begin{bmatrix} -\frac{y_L^0}{\left(x_L^0\right)^2 + \left(y_L^0\right)^2} & \frac{x_L^0}{\left(x_L^0\right)^2 + \left(y_L^0\right)^2} \end{bmatrix} \begin{bmatrix} \dot{x}_L^0 \\ \dot{y}_L^0 \end{bmatrix} \tag{2.5}$$

Note that this transformation contains nonlinear terms inside the Jacobian. However, unlike Eq. (2.2), the transformation in Eq. (2.5) is linear in the velocity coordinates of the landmark, $\left[\dot{x}_L^O, \dot{y}_L^O\right]^T$. In general, with a non-linear coordinate transformation for the representation of a point, one obtains a linear local transformation for the velocity vector representing the motion of the point. The dependence of the Jacobian upon the point at which it is calculated indicates that linearity is achieved locally. As the position of the landmark changes, so does the Jacobian.

2.5 Deriving the environment from noise-free sensor data

Now, we wish to use Eq. (2.5) to obtain the position of the landmark from the observed motion of its projection on the sensor. This task falls in the broader class of "inverse optics" problems. First, we simplify our problem by making some assumptions:

1. That we move along the heading direction, i.e. the y-axis of the local frame $Ox^o y^o$.
2. That the landmark is fixed in the environment. This corresponds to the concept that our own motion does not affect the state of the external world. As a consequence, if we move along the heading direction with a speed *v* the relative velocity of the landmark in the sensor frame of reference is *-v*.
3. That our dead-reckoning system is accurate. That is we know the position and orientation of the frame $Ox^o y^o$ within the environment frame $Mx^M y^M$.

The first and last hypotheses will later be relaxed to consider translation and rotations and to allow for errors caused by uncertainty about our own state of motion. We take advantage of the first two assumptions to simplify Eq. (2.5) as

$$\dot{\xi}_L = \begin{bmatrix} -\frac{y_L^0}{\rho_L^2} & \frac{x_L^0}{\rho_L^2} \end{bmatrix} \begin{bmatrix} 0 \\ -v \end{bmatrix} = -\frac{x_L^0}{\rho_L^2} v. \tag{2.6}$$

Here, we have also taken advantage of the fact that $\rho_L = \sqrt{\left(x_L^0\right)^2 + \left(y_L^0\right)^2}$ is the distance of the landmark from the center of the sensor along the projecting direction. We know our speed, *v*, the

projection angle of the landmark, ξ_L, and the rate of change of the projection angle $\dot{\xi}_L$. In alternative to the information on the temporal derivatives of the landmark and of its projection angle, we may use the changes of these two variables over some small but finite time interval. In this case, however, the greater these changes, the greater the approximation error associated with Eq. (2.6) expressed in terms of finite differences, $\Delta\xi$ and $\Delta y_L^O = v\Delta t$. We substitute the numerator on the right side of Eq. (2.6) with its expression in terms of the projection angle, $x_L^O = -\rho_L \sin(\xi_L)$. Then, the unknown distance of the landmark from the eye center, *O* is

$$\rho_L = \frac{\sin(\xi_L)}{\dot{\xi}_L} v \tag{2.7}$$

A finite approximation for ρ_L is

$$\rho_L \approx \sin(\xi_L) \cdot \frac{\Delta y_L^0}{\Delta \xi_L} \tag{2.8}$$

Note that there are two critical situations in which Eq. (2.7) cannot be used, both related to the vanishing of the image speed, $\dot{\xi}_L$. One is when the landmark is in the heading direction. Then, $\xi_L = \dot{\xi}_L = 0$ and the landmark position can be anywhere along the heading line, either in the N or in the S direction. The other condition ($\xi_L \neq 0$, $\dot{\xi}_L = 0$) corresponds to the landmark being very far away, ideally at infinity along the ray at ξ_L radians from the heading direction. If either condition occurs, it is impossible to form a model of the landmark location. Otherwise, the local landmark coordinates are

$$\begin{cases} x_L^0 = -\rho_L \sin(\xi_L) \\ y_L^0 = \rho_L \cos(\xi_L) \end{cases} \tag{2.9}$$

These coordinates are combined with the dead-reckoning information about the position and heading direction of the sensor to form a stable representation of the landmarks in the external space. This representation does not depend upon G's state of motion with respect to the landmarks. The position of the sensor center is a vector $\mathbf{r}_O^M = \begin{bmatrix} x_O^M & y_O^M \end{bmatrix}^T$. The heading direction is the angle η of the oriented $\overrightarrow{ON}$ line with respect to the north direction of the model space. This is also expressed as a unit vector $\begin{bmatrix} -\sin(\eta) & \cos(\eta) \end{bmatrix}^T$. The unit vector describing the sensor x-axis in terms of the model axes is $\begin{bmatrix} \cos(\eta) & \sin(\eta) \end{bmatrix}^T$. Combining this information

with the local coordinates of the landmark, we obtain the landmark coordinates in the external space model:

$$\begin{cases} x_L^M = x_O^M + x_L^O \cos(\eta) - y_L^O \sin(\eta) = x_O^M - \rho_L \sin(\xi_L)\cos(\eta) - \rho_L \cos(\xi_L)\sin(\eta) \\ y_L^M = y_O^M + x_L^O \sin(\eta) + y_L^O \cos(\eta) = x_O^M - \rho_L \sin(\xi_L)\sin(\eta) + \rho_L \cos(\xi_L)\cos(\eta) \end{cases}$$

(2.10)

This expression can be written in a more compact form, using a vector/matrix notation:

$$\mathbf{r}_L^M = \mathbf{r}_O^M + R(\eta)\mathbf{r}_L^O \qquad (2.11)$$

where we introduced the rotation matrix

$$R(\eta) = \begin{bmatrix} \cos(\eta) & -\sin(\eta) \\ \sin(\eta) & \cos(\eta) \end{bmatrix}. \qquad (2.12)$$

This is a special type of matrix, as will be further discussed, which describes rotations over a plane. The behavior of a rat's place cell is consistent with this operation, as the cell fires when the rat passes at a spatial location that is referred to a fixed frame of reference.

2.6 Rigid motions and homogeneous coordinates

Rigid motions are combination of translations and rotations. These motions are called "rigid" because they do not affect the distances between points in space. In mechanics, a rigid body is a solid object whose points remain always at a fixed distance with respect to each other. Such an object can only undergo translations and rotations, which are therefore called rigid transformations. When we move around a room, everything else remains at rest – assuming that we are not colliding with any object. Therefore, if we look at things from our perspective, we see the stationary environment moving with respect to us as a big "rigid body". But how can we take advantage of this basic element of knowledge to derive our own motion from what we observe?

To approach this issue it is useful to introduce an algebraic tool that was first conceived by August Ferdinand Moebius, a mathematician known to many for the homonymous Moebius strip, a two-dimensional surface in which up and down cannot be distinguished. Less broadly known is the fact that Moebius introduced homogeneous coordinates to simplify problems of projective geometry. Homogeneous coordinates also provide us with a single framework to describe in matrix form both rotations and translations. It is a nice trick. Consider a point P in a 2-dimensional Cartesian space, with coordinates x and y. We can do two types of operations in

the Cartesian framework that will change the coordinates. We can apply a transformation such as a stretch, a shear or a rotation, while the origin of the coordinate system remains fixed. These transformations are represented by 2x2 matrices, so that in the new system, the new coordinates of P, $\bar{x}$ and $\bar{y}$ are linear transformations of the old coordinates:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2.13}$$

Transformations of this kind form an important group, called the general linear group, GL. We obtain a second type of transformation simply by moving, or translating, the origin of the reference frame. If we displace the origin by a vector $-b$ with coordinates $-b_x$ and $-b_y$, then every point in the plane will have new coordinates

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \end{bmatrix}. \tag{2.14}$$

Combining a transformation of GL with a translation of the origin and using a more compact notation, one obtains a general affine transformation:

$$\bar{\mathbf{r}} = M\mathbf{r} + b \tag{2.15}$$

with $\mathbf{r} = [x, y]^T$ and $\bar{\mathbf{r}} = [\bar{x}, \bar{y}]^T$. While this expression looks quite simple, Moebius managed to make it simpler by introducing homogeneous coordinates. In homogeneous coordinates, the general affine transformation is reduced to a single matrix operation.

Figure 2.5. Affine space. (Top) In affine geometry a vector is a translation that brings a point into another; (Bottom) If the landmarks (L1 and L2) are stationary, the motion of the observer is equal and opposite to the motion of each landmark relative to the observer.

To obtain this result we need to change the representation of the points by adding one component to each of them. Here, we will not go into much detail about the significance of this extra component in projective geometry. However, to understand the concept of affine geometry, we need to make a distinction between points and vectors (Fig. 2.5). A space (or a plane) is a collection of points. A vector is the transformation that leads from a point to another. Therefore, a pair of points, A and B, defines the vector $\overrightarrow{AB}$ that transforms A into B. The combined notions of points and vectors constitute what is known as the affine space. Homogeneous coordinates represent points on a plane by three coordinates, which we place into a column vector for

performing algebraic operations. The three coordinates are graphically obtained by considering a family of parallel planes intersecting the *z*-axis at different distances, *w*, from a center of projection, P. Consider the plane at $w=1$. Over this plane, a point with Cartesian coordinates (x, y) has homogeneous coordinates $\begin{bmatrix} x, & y, & 1 \end{bmatrix}^T$. In projective geometry this point is equivalent to the points on other planes, along the same ray from P. These equivalent points have coordinates $\begin{bmatrix} wx, & wy, & w \end{bmatrix}^T$ where $w \neq 0$ is an arbitrary positive number. Thus, for example, we can represent the point $(3x, 3y)$ as $\begin{bmatrix} 3x, & 3y, & 1 \end{bmatrix}^T$ or, equivalently, as $\begin{bmatrix} x, & y, & \frac{1}{3} \end{bmatrix}^T$.

Thus, the third component of the homogeneous vector is a scaling factor for the coordinates of the point lying on the plane at $w=1$. On this plane, the point at infinity along the direction of $\begin{bmatrix} x, y \end{bmatrix}^T$ has homogeneous coordinates $\begin{bmatrix} x, y, 0 \end{bmatrix}^T$. To represent all points in the Euclidean plane at finite distance from the origin, we set $w=1$. Consider a point P_1 with coordinates $[x_1, y_1, 1]^T$ and a point P_2 with coordinates $[x_2, y_2, 1]^T$. The vector $\mathbf{d}$ that joins them is

$$\mathbf{d} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \tag{2.16}$$

and the distance between the two point is simply the Euclidean norm of $\mathbf{d}$, namely

$$\|\mathbf{d}\| = \sqrt{\mathbf{d}^T \mathbf{d}}\,. \tag{2.17}$$

Starting from the original representation of a point in two dimension as $\mathbf{r} = \begin{bmatrix} x, y \end{bmatrix}^T$, we write the representation in homogeneous coordinates as $\begin{bmatrix} x, & y, & 1 \end{bmatrix}^T = \begin{bmatrix} \mathbf{r}^T & 1 \end{bmatrix}^T$. We then derive the general affine transformation in 2 dimensions by building the matrix

$$H = \begin{bmatrix} m_{1,1} & m_{1,2} & b_x \\ m_{2,1} & m_{2,2} & b_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} M & \mathbf{b} \\ 0 & 1 \end{bmatrix} \tag{2.18}$$

Applying *H* to the point in homogeneous coordinates we obtain:

$$H \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix} = \begin{bmatrix} M & \mathbf{b} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix} = \begin{bmatrix} M\mathbf{r} + \mathbf{b} \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{r}} \\ 1 \end{bmatrix} \tag{2.19}$$

which is analogous to Eq. (2.15). Thus, we are now able to express all affine transformations as matrix operations on points in homogeneous coordinates.

Rigid transformations are particular affine transformations that do not affect the distance between points. Consider, again, the points P_1 and P_2 with their distance $\mathbf{d}$ as in Eq. (2.17) . This distance should not change after a rigid transformation. The coordinates of the two points, after an affine transformation become

$$\begin{bmatrix} M\mathbf{r}_1 + \mathbf{b} \\ 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} M\mathbf{r}_2 + \mathbf{b} \\ 1 \end{bmatrix} \tag{2.20}$$

with $\mathbf{r}_i = \begin{bmatrix} x_i, & y_i \end{bmatrix}^T \quad (i = 1, 2)$. Therefore the new difference vector is

$$\bar{\mathbf{d}} = M(\mathbf{r}_2 - \mathbf{r}_1) = M\mathbf{d} \tag{2.21}$$

and the distance is

$$\left\| \bar{\mathbf{d}} \right\| = \sqrt{\mathbf{d}^{\mathrm{T}} M^{\mathrm{T}} M \mathbf{d}} \tag{2.22}$$

The requirement that $\|\mathbf{d}\| = \|\bar{\mathbf{d}}\|$ is evidently satisfied by any translation, since the vector **b** does not appear in Eq. (2.21). As for the matrix *M*, the invariance of distances corresponds to requiring that

$$M^T = M^{-1} \tag{2.23}$$

This is the definition of an orthogonal matrix and is satisfied by rotation matrices, as in Eq. (2.12). As a result, any rigid motion combines a rotation and a translation and is represented by a single matrix in homogeneous coordinates: the product of a translation matrix and a rotation matrix:

$$\begin{matrix} \textit{Rotation} \\ \begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \end{matrix} \begin{matrix} \textit{Translation} \\ \begin{bmatrix} I & \mathbf{b} \\ 0 & 1 \end{bmatrix} \end{matrix} = \begin{matrix} \textit{Rigid Motion} \\ \begin{bmatrix} M & \mathbf{b} \\ 0 & 1 \end{bmatrix} \end{matrix} \tag{2.24}$$

What insight do we derive from this? Consider how the order of two rigid motions affects the final result. Take a step forward and then turn to the left by 90 degrees. Make a note of your position and orientation and start again. Turn left by 90 degrees and then take a step forward. It is evident that we are now in a position that is quite different from the previous one. The two combinations of rotation and step only differ by their order. This effect of the order is typical of matrix multiplications. In general, the product of two matrices is not commutative, i.e., $AB \neq BA$ with the exception of some particular cases.

Let us now consider what happens when a generic rotation *R* and a translation *T* , both in the plane, are described by the two homogeneous matrices:

$$R = \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.25}$$

and

$$T = \begin{bmatrix} I & \mathbf{b}_1 \\ 0 & 1 \end{bmatrix} \tag{2.26}$$

In Eq. (2.25), R_1 is a 2x2 matrix of the form in Eq. (2.12), and $\mathbf{b}_1$ is a 2x1 vector. To derive the effect on a vector of a translation followed by a rotation, we write a cascade:

$$RT = \begin{bmatrix} R_1 & \mathbf{0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & \mathbf{b}_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 & R_1\mathbf{b}_1 \\ 0 & 1 \end{bmatrix} \tag{2.27}$$

The reverse sequence – rotation followed by translation is:

$$TR = \begin{bmatrix} I & \mathbf{b} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 & \mathbf{b} \\ 0 & 1 \end{bmatrix}. \tag{2.28}$$

This illustrates that in the step-turn/turn-step example, we end up with the same orientation but in different locations. This lack of commutativity creates an ambiguity that is resolved in a continuous movement where small rotations and small translations along the heading directions are repeated in time. In fact, if we consider very small motions, we see that rotations and translations commute. With a small angle, $\delta\theta$, the homogeneous rotation is approximated by

$$\delta R = \begin{bmatrix} 1 & -\delta\theta & 0 \\ \delta\theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.29}$$

and with a small translation in the heading direction, δy, the homogeneous translation is

$$\delta T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.30}$$

Then, combining the two we obtain

$$\delta R \cdot \delta T = \begin{bmatrix} 1 & -\delta\theta & -\delta\theta \cdot \delta y \\ \delta\theta & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\delta\theta & 0 \\ \delta\theta & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix} = \delta T \cdot \delta R \tag{2.31}$$

The approximation corresponds to neglecting second order terms.

2.7 Updating the space model.

As we move, we can safely assume that most objects around us remain stationary with respect to each other. Thus they collectively form a frame of reference that we can use to derive and update a model of space. Suppose that our gerbil now takes a small step, δl, in the heading direction and that it also rotates by a small angle $\delta\theta$. How would this added rotation affect G's estimate of the landmark locations? To derive the coordinates of the landmarks in Eq. (2.9) we assumed a pure translational motion. Now we want to allow for both rotations and translations, under the hypothesis that all landmarks in sight are stationary. Therefore, in G's field of view, the landmarks will all move by the same amount, equal and opposite to G's motion. Using Eq. (2.31), we derive where G expects to see the landmark at time *t*:

$$\begin{bmatrix} x_i^O \\ y_i^O \\ 1 \end{bmatrix}(t) = \begin{bmatrix} 1 & \delta\theta & 0 \\ -\delta\theta & 1 & -\delta l \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_i^O \\ y_i^O \\ 1 \end{bmatrix}(t-1) = \begin{bmatrix} x_i^O \\ y_i^O \\ 1 \end{bmatrix}(t-1) + \begin{bmatrix} y_i^O(t-1)\cdot\delta\theta \\ -x_i^O(t-1)\cdot\delta\theta - \delta l \\ 1 \end{bmatrix} \quad (2.32)$$

The change in each landmark's location relative to G is

$$\begin{bmatrix} \delta x_i^O \\ \delta y_i^O \\ 1 \end{bmatrix}(t) = \begin{bmatrix} y_i^O(t-1)\cdot\delta\theta \\ -x_i^O(t-1)\cdot\delta\theta - \delta l \\ 1 \end{bmatrix} \quad (2.33)$$

We can now abandon the homogeneous coordinate notation, which has fulfilled its role of combining rigid motions. We place the movement commands, δl and $\delta\theta$, in a single command, or "input" array: $\mathbf{u} = \begin{bmatrix} \delta l & \delta\theta \end{bmatrix}^T$. With this, the relative motions of the landmarks become

$$\mathbf{r}_i^O(t) = \mathbf{r}_i^O(t-1) + M(\mathbf{r}_i^O(t-1))\mathbf{u}(t) \quad (2.34)$$

where

$$\mathrm{M} = \begin{bmatrix} 0 & y_i^O(t-1) \\ -1 & -x_i^O(t-1) \end{bmatrix}$$

Let us go back to the expression in Eq. (2.5) for the Jacobian of the landmark projections. We use it now to derive the expected change in the projection of landmark *i* caused by the motion command:

$$\delta\xi_i \approx \begin{bmatrix} -\frac{y_i^O}{\rho_i^2} & \frac{x_i^O}{\rho_i^2} \end{bmatrix}\begin{bmatrix} y_i^O\delta\theta \\ -x_i^O\delta\theta - \delta l \end{bmatrix} = \frac{-\left(y_i^O\right)^2 - \left(x_i^O\right)^2}{\rho_i^2}\delta\theta - \frac{x_i^O}{\rho_i^2}\delta l = -\delta\theta + \frac{\sin\xi_i}{\rho_i}\delta l$$

From this we obtain the new expression for the distance of the landmark:

$$\rho_i \approx \frac{\delta l}{\delta \xi_i + \delta \theta} \sin \xi_i \tag{2.35}$$

This simply says that in deriving the distance of each landmark one should subtract the projection change $-\delta\theta$ associated to our own rotation. Indeed our own rotation cannot carry any information about the distance of an object, since the effect is the same for all objects on the same projective line!

Once we have an initial model of the space and the landmarks, we can ask how G can maintain this model by collecting additional information. The problem of deriving a map of space and to localize oneself in this map is an important problem in robotics (Dissanayake, Newman, Clark, Durrant-Whyte, & Csorba, 2001; Thrun, Fox, Burgard, & Dellaert, 2001). Practical applications include the development of autonomous vehicles capable of moving unmanned in a mine, inside a harbor or other dangerous environments to collect and transport items. The environment is populated by various objects and by people moving around. It may be possible, however, to place beacons or other fixtures at a variety of locations. Then the task for the vehicle becomes quite similar to the task faced by the gerbils when looking for seeds in relation to fixed landmarks. Often, robotic engineers have explored these problems from a biomimetic perspective. "Simultaneous localization and map building" or SLAM (Dissanayake, et al., 2001) is a term to describe how the problem of navigation is dealt in the mathematical framework of optimal state estimation (we will get to this topic in chapter 4). Here, we merely introduce the general issues encountered in forming and maintaining a map of space.

We build and update G's model of space in the reference frame of the fixed landmarks. This reflects the observation that space coding cells in the hippocampus and in the entorhinal cortex respond to moving into locations of space that are fixed in some external reference frame. Without getting into the modeling style of artificial neural networks, here we wish only to present some mathematical problems associated with the formation of a spatial map. We begin by establishing an external frame of reference, centered at some point that may either be one of the landmarks or any element of the scene that is stationary with respect to the landmarks. In what follows, we make the assumption that all coordinates are referred to this fixed frame. Thus, the extrinsic description of the space model has a state vector

$$\mathbf{s} = \left[x_O, \quad y_O, \quad \eta, \quad x_1, \quad y_1, \quad \cdots, \quad x_N, \quad y_N \right]^T = \left[\mathbf{r}_O^T, \eta, \mathbf{r}_1^T, \cdots, \mathbf{r}_N^T \right]^T \tag{2.36}$$

The state vector includes our position and heading direction together with the position of the N fixed landmarks. In this extrinsic frame, when G makes a movement, $[\delta l, \delta\theta]^T$ its position changes by a translation along the heading direction η:

$$\mathbf{r}_O(t) = \mathbf{r}_O(t-1) + \begin{bmatrix} -\delta l \cdot \sin\eta(t-1) \\ \delta l \cdot \cos\eta(t-1) \end{bmatrix} \tag{2.37}$$

and then the heading direction is updated:

$$\eta(t) = \eta(t-1) + \delta\theta \tag{2.38}$$

This can be re-written in a compact matrix form as

$$\begin{bmatrix} x_O \\ y_O \\ \eta \end{bmatrix}(t) = \begin{bmatrix} x_O \\ y_O \\ \eta \end{bmatrix}(t-1) + \begin{bmatrix} -\sin\eta(t-1) & 0 \\ \cos\eta(t-1) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta l \\ \delta\theta \end{bmatrix} \tag{2.39}$$

By definition, in the extrinsic reference the landmarks do not move. Therefore, the state of the environment is governed by the following Eq.:

$$\mathbf{s}(t) = \mathbf{s}(t-1) + B(s(t-1))\mathbf{u} \tag{2.40}$$

where we have introduced the $(N+3)\times 2$ matrix

$$B = \begin{bmatrix} -\sin\eta(t-1) & 0 \\ \cos\eta(t-1) & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \tag{2.41}$$

Eq. (2.40) has a deceptive linear appearance. However, it is not a linear equation because the "control matrix" B depends upon one of the state variables, the heading direction. This limits the possibility to apply known linear methods even in this very simple case.

We consider two elements that contribute to the updating of the internal model of the environment. We have described how G expects its position with respect to the fixed landmarks to change in time based on how it thinks it is moving. This is called the "process". The other is a model of the expected sensation caused by G's motion. This is called the "observation". The observation may come in two flavors. One may assume to know how objects generate projected images in the eye and have a model of how such *sensations* are formed. Alternatively, we have a model like the one described here earlier, which generates images of the objects based on

sensations from the eye. This is a model of *perception*, and is the kind of observation model that we consider here. The perception model has the useful geometrical property of generating mathematical objects of the same type as the mathematical objects produced by the process model. Both perception and process models generate hypotheses about the state of navigation. Thus, we can compare their results.

The observation model provides an estimate of the current locations of the landmarks relative to us, in our own frame of reference, $\mathbf{r}_i^O(t)$. We may readily transform these data into an estimate of the positions of the landmarks in the external frame, M :

$$\hat{\mathbf{r}}_i(t) = H(\eta, \xi_i, \delta l(t), \delta\xi_i(t)) \tag{2.42}$$

The hat superscript indicates the data obtained from the observation. To derive this expression more explicitly, we need to know the heading direction and the step δs along this direction. We can apply Eq. (2.11). Using the homogeneous coordinate notation:

$$H(\eta, \xi_i, \delta l(t), \delta\xi_i(t)) = \begin{bmatrix} R_{-\eta} & -R_{-\eta} r_M^O(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_i^O(\xi_i, \delta l, \delta\xi_i) \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos(\eta) & -\sin(\eta) & -\cos(\eta)x_M^O + \sin(\eta)y_M^O \\ \sin(\eta) & \cos(\eta) & -\sin(\eta)x_M^O - \cos(\eta)y_M^O \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\dfrac{\delta l}{\delta\xi_i}\sin^2(\xi_i) \\ \dfrac{\delta l}{\delta\xi_i}\sin(\xi_i)\cos(\xi_i) \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} -\dfrac{\delta l}{\delta\xi_i}\left(\sin^2(\xi_i)\cos(\eta) + \sin(\xi_i)\cos(\xi_i)\sin(\eta)\right) - \cos(\eta)x_M^O + \sin(\eta)y_M^O \\ \dfrac{\delta l}{\delta\xi_i}\left(\sin(\xi_i)\cos(\xi_i)\cos(\eta) - \sin^2(\xi_i)\sin(\eta)\right) - \sin(\eta)x_M^O - \cos(\eta)y_M^O \\ 1 \end{bmatrix} \tag{2.43}$$

Note that the displacement term, $r_M^O = \left[x_M^O, y_M^O \right]$ is the location of the fixed reference landmark in G's moving frame of reference. Therefore, G's own location is derived by transforming its origin from its own frame (i.e. the point $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$) to the external frame (Fig 2.4):

$$\mathrm{r}_0 = \begin{bmatrix} R_{-\eta} & -R_{-\eta} r_M^O(t) \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 1 \end{bmatrix} = \begin{bmatrix} -R_{-\eta} r_M^O(t) \\ 1 \end{bmatrix} = \begin{bmatrix} -\cos(\eta) x_M^O + \sin(\eta) y_M^O \\ -\sin(\eta) x_M^O - \cos(\eta) y_M^O \\ 1 \end{bmatrix} \quad (2.44)$$

Eq. (2.40) and Eq. (2.42) are two ways for computing the same thing: the structure of the space around G in terms of the landmarks and G's location at different instants of time. The first method is based on predicting how things will look when G moves. The second uses G's observation of the landmarks and of the reference point. Both methods use some form of prior knowledge about the landmarks being stationary and about self motion. But they do so in different ways. If everything is working then the predictions from the process model and the actual observation must coincide, as shown in Fig. 2.6. The leftmost panel displays the paths as we are moving. The environment has three fixed landmarks, plus a reference point, indicated by a small square. The reference direction (North) is shown by the arrow in Fig. 2.6. As we move in this space we form two models, one based on the observation model of Eq. (2.42) and the other based on the predictive model of Eq. (2.40). The outcomes of the prediction models are shown on the top panels of Fig. 2.6. The outcomes of the observation models are shown in the lower panels.

Figure 6. A simple navigation. Top Left: The external space. The simulated gerbil (G) moves within a planar environment populated by landmarks. These are indicated as points with letters A, B, C . A small square (R) and an arrow indicate a reference point in the environment and the actual "North" direction. The point R can be thought of as a particular landmark or as point in space with some particular salience. Top Middle: Space model obtained from the projected images of the landmarks and of the reference on the eye model (Top Right). The scales of the model and of the space are deliberately different in this figure. The locations of the landmarks and G are derived from the motions of their projections on the eye after a small translation of the agent in the heading direction. In the model all locations are referred to the image of the reference, which becomes the origin of the model's coordinates. Bottom: Noise effects as G moves in the environment (Bottom-Left). The four "process" panels represent four internal models of the environment obtained from the iteration of the process model. The models are corrupted by Gaussian noise of increasing amplitude from left to right. The four "observation" panels are model reconstructions based only on the observation of the landmarks. These models are also corrupted by Gaussian noise of increasing amplitude from left to right. The leftmost observation and process models have zero noise. Note that, without noise, the observation model has a minimal amount of error, revealed by the slightly larger images of the landmarks. This is because the observation model has a nonlinear inverse perspective transformation that is affected by the discrete approximation of the positional increments.

Error sources

Why can there be a discrepancy between what G expects to observe and what it actually sees? The answer is deceivingly simple: because of noise. But what is noise? This is a more complex

question. We can call noise whatever causes unexpected behaviors. If we know exactly the structure of the process, the command that we are issuing and their effect on our movement, and if the images have no blur or unexpected distortion, there would be no question about the fidelity of our internal representation of the environment. Unfortunately, things are not so simple. Any model is likely to have some structural errors. For example, one normally integrates small but finite movements instead of infinitesimal displacements. Figure 2.6 illustrates the effect of increasing amounts of uncertainty on G's position and on the placement of the landmarks within a model of space. The top row demonstrates what would happen if G's model were built only based on what G knows a priori about its movement. Here, G starts from an initial estimate of the landmarks and its own location. It assumes that this initial estimate is correct. Then, each time G takes a step it calculates a new position for itself and for the landmarks, using Eq. (2.40). However, now there is an unexpected term $\boldsymbol{\varepsilon}_s(t)$. A random variable representing the uncertainty of the predictions:

$$\mathbf{s}(t) = \mathbf{s}(t-1) + B(\mathbf{s}(t-1))\mathbf{u} + \boldsymbol{\varepsilon}_s(t) \tag{2.45}$$

The random variable may follow some unknown distribution. However, most analyses assume that noise is drawn from a normal distribution with known variance, Q. When the state variable is a vector quantity, the variance is replaced by a covariance matrix of the same dimension. The process noise that was used in the examples of Fig. 2.6 had two components, one for position uncertainty and one for heading uncertainty. In practical situations, process uncertainty derives not only from a limited knowledge of the actual value of the commands, but also and more importantly from the unexpected external factors that may affect the outcome of each command. Factors like rough terrain and wind gusts would cause variable degrees of uncertainty on G's predicted position. We are safe to assume that the position of the landmarks is constant. However, knowledge of this position could also be affected by some degree of uncertainty. In generating the trajectories of Fig. 2.8, the process model assumed that the uncertainty about each landmark position was the same as the uncertainty on G's position. Alternatively, one may assume a lower amount of uncertainty for elements that are known to be stationary. But the really critical issue here is the shape of the expected error distribution. One approach to random variables of uncertain origin is to consider them as normally distributed, with zero mean and variance Q:

$$\varepsilon_s \sim N(0, Q) \tag{2.46}$$

This choice has its main rationale in the central limit theorem of probability theory, which states that the mean of any random variable tends to be normally distributed, as the number of observations grows larger. However, one can be sure that the effect of friction on the movement

of a vehicle is always opposite to the direction of motion. Therefore, when looking at this effect as a random variable, it could hardly be described as a noise with zero mean with symmetric distribution. Nevertheless, the standard assumption of Eq. (2.46) has significant computational advantages and it is common practice in mathematics to make simplifications of this type. In such cases it remains important to keep in mind what elements of reality are being ignored. Back to Fig. 2.6, the variance Q for the positional noise was varied from .25 to 4 units. This being a simulation, the entity of a unit is somewhat arbitrary. To give an idea of the dimensions at play, the distances between pairs of landmarks was about 90 units. The variance for the heading direction varied between 1 and 9 degrees.

Noise is also present in the observation process, starting from the signals originating from sense organs. In our case, G observes the locations of the landmarks and infers its own position by looking at the projections of the landmarks and knowing that the landmarks are fixed in space. In a perception model, one observes variables that are related at all times to one's own state of motion. We have derived a particular expression, Eq. (2.43), for the observation process that gives the location of the landmarks as a function of G's state of motion. As it was the case for the process model, we now add a random variable ε_r to the deterministic component of the observation model:

$$\hat{\mathbf{r}}_i(t) = H(\eta, \xi_i + \varepsilon_\xi, \delta l(t), \delta\xi_i(t)) + \varepsilon_r(t) \tag{2.47}$$

An additional noise term, ε_ξ appears inside the function H. This represents the uncertainty on the "retinal" signals, ξ, which results into an uncertainty on the reconstructed landmark locations. The observed data about the positions of the landmarks relative to G are then reflected into the uncertainty on G's position. The effects of observation noise are illustrated in the lowest portion of Fig. 2.6. We generated these examples with the retinal noise varying from .05 to .2 degrees , the position noise from .5 to 1.5 and the heading noise from .5 to 1.5 degrees.

2.8 Combining Process and Observation models

We have considered two models. One generates a prediction about the next state and the other makes an observation of the same state. The two models include some amount of randomness that causes uncertainty on their outcome. Methods of optimal estimation (more details in Chapter 4) are based on the idea of combining the outcomes of these two models and a particular way to do so is to require that the combination be convex. A convex combination of two points is a third

point that lies between them. The most general form for a convex combination of a point P1 and P2 is a point P3 that lies on the segment joining them:

$$P_3 = \alpha_1 \cdot P_1 + \alpha_2 \cdot P_2 \quad \text{with} \quad \alpha_1 + \alpha_2 = 1 \tag{2.48}$$

Note that this rule applies to points in any number of dimensions. Let us begin by considering a simple one-dimensional case, in which the process model generates the estimate $s(t)$ and the observation generates another estimate $\hat{s}(t)$. Both are one-dimensional real numbers. It seems plausible that the true unknown value may likely fall between these two estimates. However this is not always the case. So, instead of considering individual trials, we should consider collections of "equivalent" trials. This is easy to do in a simulation, although it is time consuming. All one needs is to repeat each prediction and each observation multiple times and collect some statistics of the outcomes. In doing so, one implicitly assumes that the process under study is *ergodic*. By this, we mean that one can infer the statistical properties of the process from a large number of samples at each point of time. In our case, we repeat each step of the process a number of times and calculate the mean and covariance of the predicted and observed states. Suppose that the observed state has very little variability compared to the predicted state. Because we have assumed that the noise has a Gaussian distribution with zero mean, we could safely conclude that the true state is likely closer to the observed state. Conversely, if the observations are more variable than the predictions, the true state is likely closer to the predicted state. Therefore, variance appears to be a reasonable criterion to establish the position of the final estimate between the observed and the predicted state. One simple way to do so is to give the more weight to the process with smaller variance. That is, let $\beta_O = \frac{1}{\sigma_O^2}$ be the inverse of the variance of the observation $\hat{s}(t)$ and $\beta_P = \frac{1}{\sigma_P^2}$ be the inverse of the variance of the prediction model, $s(t)$. Then, we generate the normalized coefficients

$$\begin{cases} \dfrac{\beta_O}{\beta_O + \beta_P} = \dfrac{{\sigma_P}^2}{{\sigma_O}^2 + {\sigma_P}^2} \\ \dfrac{\beta_P}{\beta_O + \beta_P} = \dfrac{{\sigma_O}^2}{{\sigma_O}^2 + {\sigma_P}^2} \end{cases}$$

and we derive the state estimate from the convex combination

$$s_E(t) = \frac{{\sigma_O}^2}{{\sigma_O}^2 + {\sigma_P}^2} \cdot s(t) + \frac{{\sigma_P}^2}{{\sigma_O}^2 + {\sigma_P}^2} \cdot \hat{s}(t). \tag{2.49}$$

A simple algebraic manipulation leads to an expression for the estimated state that is a correction of the predicted state based on the difference between observed and predicted state:

$$s_E = \frac{{\sigma_O}^2}{{\sigma_O}^2 + {\sigma_P}^2} \cdot s(t) + \frac{{\sigma_P}^2}{{\sigma_O}^2 + {\sigma_P}^2} \cdot s(t) + \frac{{\sigma_P}^2}{{\sigma_O}^2 + {\sigma_P}^2} \cdot \hat{s}(t) - \frac{{\sigma_P}^2}{{\sigma_O}^2 + {\sigma_P}^2} \cdot s(t) = \\ = s(t) + K \cdot \left(\hat{s}(t) - s(t) \right) \tag{2.50}$$

with

$$K = \frac{{\sigma_P}^2}{{\sigma_O}^2 + {\sigma_P}^2} \tag{2.51}$$

We see that the estimated state is obtained from the predicted state with a correction proportional to the difference between predicted and observed states. If the variability of the prediction is much larger than the variability of the observation, then the gain K will be close to 1. In the opposite case, K will be close to zero.

Figure 2.7. Combining predictions and observations. As G moves among landmarks, it forms a map of the landmarks and localizes itself in this map. This figure illustrates the effects of two sources of noise: noise in the internal model of the process and noise in the observation. The top panels correspond to a condition in which the observation noise is large compared to the process noise. The bottom panels describe a situation in which both noise terms are relatively large and similar. The panels on the left show estimated locations of landmarks and G, obtained by the observation system alone. At each time step, the model used five samples of the landmark locations and G's position. These are shown in light grey. The dark colored markers are averages of these individual samples. The second panels from the left show estimated positions obtained from the process model alone. Again, each dark colored marker is an average over 5 points. The third panels from the left show the same scenario derived from a convex combination of observed and predicted locations. The combination is based on the relative variance of predicted and observed points, as described in the text. The graphs on the right display the overall reconstruction errors obtained for each of the two noise distributions and for each reconstruction model. Note that form most times the combination of observation and prediction models provides a better estimate that either method.

This approach is illustrated by the examples in Fig. 2.7. In our case, the state is not a scalar, but the array (2.36) with heterogeneous positional and angular components. To preserve the flavor of a convex combination, we estimate independently each element of the state vector – $\mathbf{r}_O^T, \eta, \mathbf{r}_1^T, \cdots, \mathbf{r}_N^T$ -by taking the trace of the respective covariance matrices for the observed and the predicted values. The more rigorous approach to this type of estimate is presented in Chapter 4. Here, we consider two different cases, with different values for the relative variances of the observation and of the prediction processes. The data shown on the top row were obtained with

high observation noise and low prediction noise. The trajectories and landmarks derived from the observation and prediction processes are shown together with the trajectory obtained from their combination, using Eq. (2.49). The graph on the top rightmost panel illustrates the net "space error", that is the net positional error for the three landmark and for G's position. As expected, the observation model generates larger errors than the prediction model and the estimated combination has similar performance to the prediction model. The situation in the bottom row is characterized by a similar amount of variance in the prediction and observation noise. Here, the lower rightmost panel shows that the performance of the combined model is superior to both component models.

As we pointed out earlier, an obvious drawback of this approach stems from the need to calculate means and variances from multiple data, when a process may only be allowed to generate one sample per time interval. In real life, we would not take multiple small steps, back and forth along any given trajectory to generate multiple samples of the landmarks and of our own position. Therefore, a great deal of attention has been devoted by signal and control theorists to the problem of estimating the statistics "on the go", one sample at a time. The Kalman filter that will be discussed in Chapter 4 is a successful and fundamental algorithm that solves this problem for linear systems with normally distributed zero mean noise. The algorithm of the Kalman filter uses an update expression that is very similar in form and substance to Eq.s (2.50) and (2.51) and its most important and critical part is in the update of the process covariance as the data keep coming in.

2.9 Back to the gerbils

We have described how a simple combination of geometrical and probabilistic rules is sufficient to reconstruct the spatial distribution of the landmarks around G together with G's own location. Of course, G is only a fictional character based on an oversimplified model of the visuomotor apparatus. Can this model account for real data? Let us refresh our memory of the experiment by Collett and collaborators that we described in the previous chapter. They placed their Mongolian gerbils inside a circular arena with two distinct landmarks. They hid a seed under the gravel at a fixed location with respect to the landmarks. After some explorations the gerbil found the seed. The gerbil engaged in this search several times, with the seed at the same location. Eventually, the gerbil remembered the location of the seed and went straight to get it. The interesting part of the experiment came after this initial practice. Once the gerbil had learned to find the seed, Collett and colleagues played a revealing trick. They displaced the landmarks and removed the seed. So now, the gerbil went in the changed environment looking for food while the scientists recorded

where the gerbil would spend time searching. In one case, the distance between the landmarks was doubled. The gerbil responded to that perturbation by searching at two locations, each location corresponding to the learned location of the seed relative to each marker (Figure 1.3C). This was seen as evidence that "they treat each landmark independently when planning a path to the goal and formulate a separate trajectory for each landmark". However, this finding appears to be at odds with the result of another experiment, where the landmarks had different appearance and, after training, were rotated by 180 degrees. Then, instead of searching at two locations, the gerbils concentrated their effort at a single location that was also rotated by the same amount (Figure 1,3F). As we pointed out in the previous chapter, this result implies that the gerbil's internal model of space was Euclidean. We have included already this assumption in some of the operation of G's rudimentary visual system. Let us now construct a single mathematical function capable of accounting for both of Collet's findings.

G's navigation system has a state that combines G's own location, r_0, and the locations of the markers, $\mathrm{r}_1,\ldots,\mathrm{r}_N$ in a single vector. In the navigation paradigm that we have described so far, we did not address an important issue: how does G decide where to move? We simply assumed that it moves somewhere. If G were a real gerbil looking for food, we may assume that not knowing the food location, it would engage in some kind of random walk, stopping occasionally to check under the gravel. Things would change when G finds the seed. The seed is a reward that would cause G to store the state vector for future use. As this experience is repeated, this memory is likely to become stronger and more stable, and perhaps also a little blurred, because the state would not be identical from trial to trial. Mathematically, we can represent this memory as a "value" function encoding the probability of finding the reward at one or more locations of the space map. Now, we can formulate a rule to build the value map, given that we found a nutritious seed at a state

$$\hat{\mathbf{s}} = \left[\mathbf{r}_{SEED}^T, \hat{\mathbf{r}}_1^T, \hat{\mathbf{r}}_2^T \right]^T . \tag{2.52}$$

Here, $\mathbf{r}_{SEED}$, $\mathbf{r}_1$, $\mathbf{r}_2$ are the positions of the seed and of the two landmarks. We dropped the heading direction η since it is not relevant to the problem at hand.

Figure 2.8. *Representations of the seed in the reference frames of the landmark. The state vector contains the position of G and of the two landmarks at the time the seed is*

discovered. Each pair of landmarks defines a reference vector upon which the seed vector is projected by inner-product. The outer product operation yields the projection of the seed vector on the orthogonal direction

We use the three elements of $\hat{\mathbf{s}}$ to build two independent representations of the seed, as shown in Fig 2.8. These are:

$$\mathbf{s}_1^0 = \left[\frac{(\mathbf{r}_0 - \mathbf{r}_1)^T (\mathbf{r}_2 - \mathbf{r}_1)}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \quad \frac{(\mathbf{r}_0 - \mathbf{r}_1) \times (\mathbf{r}_2 - \mathbf{r}_1) \cdot \hat{\mathbf{k}}}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \right]^T$$
$$\mathbf{s}_2^0 = \left[\frac{(\mathbf{r}_0 - \mathbf{r}_2)^T (\mathbf{r}_1 - \mathbf{r}_2)}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \quad \frac{(\mathbf{r}_0 - \mathbf{r}_2) \times (\mathbf{r}_1 - \mathbf{r}_2) \cdot \hat{\mathbf{k}}}{\|\mathbf{r}_2 - \mathbf{r}_1\|} \right]^T \tag{2.53}$$

The only element here that is in a way extraneous to the state vector (2.52) is the term $\hat{\mathbf{k}}$, which appears on the second term of $\mathbf{s}_0^1$ and $\mathbf{s}_0^2$. This is simply the unit vector perpendicular to the plane in which G moves and pointing upward. The "cross product" indicated by the symbol $\times$ is a standard operator of vector calculus. Given a system of unit axes in three orthogonal directions, $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$, and two vectors, $\mathbf{v} = v_1\hat{\mathbf{i}} + v_2\hat{\mathbf{j}} + v_3\hat{\mathbf{k}}$ and $\mathbf{w} = v_1\hat{\mathbf{i}} + v_2\hat{\mathbf{j}} + v_3\hat{\mathbf{k}}$, the cross product of v and w is

$$\mathbf{v} \times \mathbf{w} = \begin{bmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix} = (v_2 w_3 - v_3 w_2)\hat{\mathbf{i}} + (v_3 w_1 - v_1 w_3)\hat{\mathbf{j}} + (v_1 w_2 - v_2 w_1)\hat{\mathbf{k}} \tag{2.54}$$

We need also to remember that the cross product of two vectors is proportional to the sine of the angle between them:

$$\mathbf{v} \times \mathbf{w} = \|\mathbf{v}\| \times \|\mathbf{w}\| \mathbf{sin(vw)} \tag{2.55}$$

From both Equations (2.54) and (2.55) it follows that the cross product is anti-commutative:

$$\mathbf{v} \times \mathbf{w} = \mathbf{-w} \times \mathbf{v}\,. \tag{2.56}$$

Therefore, the sign of the second components of the "seed vectors" depends on the relative orientation of the landmarks. This is critical for reproducing both the apparently conflicting results of Collett and coworkers, when they increased the distance between the landmarks and when they rotated the landmark pattern.

In our model, when G finds a seed it constructs a value function over the navigation space, expressing the probability to find the hidden reward in relation to each visible landmark. An example of such function with the above two-landmark scenario is:

$$V\left(\mathbf{s}\,|\,\mathbf{s}_1^0,\mathbf{s}_2^0\right) \propto \exp\left(-\frac{\left\|\mathbf{s}_1-\mathbf{s}_1^0\right\|^2}{\sigma^2}\right)+\exp\left(-\frac{\left\|\mathbf{s}_2-\mathbf{s}_2^0\right\|^2}{\sigma^2}\right) \tag{2.57}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are representations of the current position of G with respect to the current landmarks, derived using the same transformation- Eq. (2.53) -that generated the two "seed vectors", $\mathbf{s}_1^0$ and $\mathbf{s}_2^0$. Each landmark, with its neighbors is an independent reference frame that generates an additive component of the value function. On the next navigation, G will move toward the places that promise the highest reward. In the landmark arrangement of Fig. 2.9A, the value function has two coincident "hills" (Fig. 9B). Therefore, G will search in the same place where it found the seed in previous trials. If the landmarks are placed at a greater distance, then the two exponentials in the value function will separate – the degree of separation being a function of the uncertainty, σ^2 (Fig. 2.9C). If instead the landmarks are rotated by 180 degrees (or any other angle), the Gaussian contributions to the value function will also rotate accordingly and will map to the same location on the navigation map (Figure 2.9D).

Figure 2.9 Simulation of the experiment by Collett et al. (1986). Compare with Fig. 2.3 of the previous chapter. A: Two landmarks and a hidden seed are arranged in a triangular configuration. B: As G navigates in the environment it forms a state representation of itself and the landmark, as in Figs. 2.6 and 2.7. Since the seed is always found at a fixed location with respect to each marker, the value function has a single "hill" centered at the seed's location. C. If the landmarks are placed at a greater distance, the two representations of the seed separate by an equivalent amount. D: If the landmark array is rotated by 180 degrees , the value function is also rotated and its true components are still fully overlapped.

The map of space built during navigation by combining the process and the observation models provides a spatial domain that offers a support for storing and retrieving memories of rewarding events, such as the discovery of food, as well as of adverse and dangerous situations. It is therefore not surprising that episodic memory and spatial information processing share common territories in the mammalian brain and that damage to this territory impairs both our ability to remember recent facts and to orient ourselves in space.

SUMMARY

How can we construct a mathematical map of space, starting from sensory and motor information? We consider the simplified model of a gerbil, with a single 1-dimensional "eye", moving over a two dimensional plane. The first problem that we encounter is to extract geometrical information from the projections of the objects on the eye. Important geometrical properties, such as the straightness of a line and the distance between two points on the navigation plane can be reconstructed from the knowledge of our own motion and from a basic assumption that we are looking at objects that are fixed in space. This assumption constrains the relative motions of the objects on the world to be in the class of rigid motions.

Homogeneous coordinates provide us with a compact representation for rigid motions by combining rotations and translations into a single linear operation. With this operation, we may update the state of the navigation environment, which includes the position of the moving gerbil and of the surrounding landmarks. This update constitutes the "process model", yielding a prediction of the future state, given the current state and knowledge of the movement intention. Sensory information from the visual system also provides an evolving representation of the state of the navigation environment. This is the "observation model". Observation and process model are both affected by uncertainty, caused by different forms of noise. Uncertainties result in variability of the corresponding models. An intuitive way to obtain the best estimate of the state of navigation is by forming a convex combination of the state estimates generated by the observation model and the process model. In this combination, each process contributes in inverse proportion of its own uncertainty.

The state of navigation is effectively a map of the environment in terms of its fixed landmarks and the gerbil's location. When a salient event occurs – such as finding food - this map provides a spatial domain upon which memory can take the form of a reward function. The reward function is a photograph of the location at which the event occurred. It provides a goal for the navigation when the same environment is encountered. From the state of the navigation environment we derive multiple representations of the location of the gerbil with respect to each landmark, at the time the salient event occurred. By a simple additive mechanism it is possible to construct a reward function that reproduces some of the experimental findings described in the previous chapter. This provides a computational rationale for the interaction of spatial and episodic memory.

References

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation, 17*(3), 229-241.

O'Keefe, J., & Nadel, L. (1978). *The Hippocampus as a Cognitive Map.* Oxford: Clarendon Press.

Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence, 128*(1-2), 99-141.

Index

NOTES

[i] The reader may be familiar with one of the many version of a joke about extreme simplification. We found this one in the Wikipedia entry for "spherical cow":
Milk production at a dairy farm was low so the farmer wrote to the local university, asking help from academia. A multidisciplinary team of professors was assembled, headed by a theoretical physicist, and two weeks of intensive on-site investigation took place. The scholars then returned to the university, notebooks crammed with data, where the task of writing the report was left to the team leader. Shortly thereafter the physicist returns to the farm, saying to the farmer "I have the solution, but it only works for spherical cows in a vacuum."
The joke reflects a common process in science, in which elements of reality are removed from a problem so as to render it tractable with the available mathematical means. Obviously a spherical cow model would not help much with understanding milk production. But it would not be too bad if you where to calculate the energy at impact of a cow falling from a cliff. Our planar gerbil with a one-dimensional circular retina is as extreme a simplification as the spherical cow. So, the reader should not think of it as a simplified model of the complex behavior of this marvelous little animal. However, this model highlights in accessible terms some of the extremely complex mathematical issues that the brain must deal to navigate within and localize itself within the environment.

[ii] In this book we adopt the convention, from Linear Algebra, to indicate the components of vectors as 1-dimensional column arrays. This is useful to represent linear coordinate transformations as matrix-vector products and is readily extended to any number of dimensions. Thus we have

$$x = \left[x_1, x_2, \ldots, x_n\right]^T \equiv \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \quad \text{and} \quad Ax = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,n} \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} .$$

[iii] This is also true in formal terms. An equivalence relation between elements of a set (indicated by the symbol ~) is a relation with three defining properties:

1. reflexive ($a \sim a$)
2. symmetric ($a \sim b \Leftrightarrow b \sim a$), and
3. transitive ($\text{if } a \sim b \text{ and } b \sim c \text{ then } a \sim c$)

The elements of a set that are equivalent to a given element define an *equivalence class*. The elements of a set are partitioned by an equivalence relation into a collection of non-overlapping equivalence classes. One ca easily see that he points (x, y) that map to the same projection ξ form an equivalence class and that all points of the 2D space external to the sensor circle are partitioned into such equivalence classes. The equivalence class constructed in this way from a function are also called a *fiber* of *f* at ξ .