MODELING IN APPLIED MATHEMATICS

Bengt Fornberg and Ben Herbst

Department of Applied Mathematics
University of Colorado
Boulder, CO 80309
USA
Email: bengt.fornberg@colorado.edu

Applied Mathematics
University of Stellenbosch
Stellenbosch 7601
South Africa
Email: herbst@sun.ac.za

Contents

NOTE. When this manuscript grows up it wants to be a book. In the mean time you will have to live with its growing pains. We do not take any responsibility for any injury, real or imaginary, that may result from using it. If you like it, please let us know what you like about it. If you don't like it, please tell us why, and what we can change to make it better. And if you know of good examples that might be useful, tell us about it.

This manuscript is somewhat unusual in the sense that it might not be possible to read it front-to-back. When we discuss the Applications in the first part for example, we sometimes refer to material that is covered in detail in later parts of the manuscript. It means that the reader may find it necessary to return to topics for a full understanding, after excursions to other parts of the manuscript. We don't really want to apologize for this. It is how things work in practice, at least in our experience. When first presented with an interesting problem, most of the time we have only a vague idea (or none at all) of how to solve it. It is only after repeatedly returning to the problem after numerous excursions, that we sometimes come up with something useful.

In earlier versions we had a section with Matlab code. It has been taken out of this version but it still exists. If you want to play with the code (highly recommended), we plan to make it available on some or other website, probably close to where you found this manuscript. Otherwise, please contact one of the authors for details.

Part 1

APPLICATIONS

CHAPTER 1

TOMOGRAPHIC IMAGE RECONSTRUCTION

1.1. Introduction.

In medicine as well as in many other situations, it is invaluable to be able to look inside objects without actually needing to slice them open, or to do something else that is grossly invasive (we regard here taking an X-ray image as 'non-invasive'). The problem with conventional X-ray imaging is that all objects along the path of the X-rays appear to be superposed on top of each other (a bit like taking many exposures without winding the film in a camera). A 3-D object has been *projected* to 2-D (with a big loss of information content, especially since one is often interested in localized and subtle changes in soft tissues that are near-transparent to X-rays). The methods we will discuss in this chapter allow full spatial reconstruction throughout a 3-D object or throughout a 2-D *slice* (in Greek $\tau\omega\mu\omega\sigma$—hence the name tomography) of the object.

In Section 1.2 we discuss very briefly six different approaches to non-invasive imaging techniques. The remaining Sections 1.3—1.8 are focused on Computational Tomography (CT)—a means of getting full reconstructions in one or—simultaneously or sequentially—many slices through the object. The input data is numerous X-ray images captured on 2-D 'film-like' or 1-D line-type electronic detectors. This data is then computationally processed to create the full, spatially true reconstruction. Of the several possible computational approaches to this reconstruction, we will focus on three: least squares (LS), filtered back projection (FBP) and Fourier reconstruction (FR).

The applications of CT extend far beyond medicine. To mention one example: In the 1980s, Exxon developed micro-tomography, mainly in order to explore the detailed pore structures of coal and of oil-saturated sand stones. One might think that non-invasiveness would not be particularly important for such objects, but,

understanding for example how oil flows to wells requires knowledge about how microscopic pores and channels in sand stone are connected. With the grains extremely hard compared to the pores, invasive procedures would inevitably destroy the pore structures before they could be recorded (a little bit like trying to feel the fine structure of a snow flake with bare fingers—the evidence would just 'melt away'). In contrast to medical tomography which achieves mm-size 2-D resolution across a slice of body-sized objects, micro-tomography achieves μm size 3-D resolution throughout cubic mm sized samples. The X-ray source is typically synchrotron radiation from an accelerator. The 10^9pixel ($1000 \times 1000 \times 1000$) volume image sets require the fastest computational inversion (FR), whereas medical imaging traditionally is based on the much slower FBP method. Here the data sets (typically 2-D) are much smaller, and throughput is limited more by patient handling than by equipment speed.

1.2. Non-invasive medical imaging techniques.

In the last few decades, several non-invasive imaging techniques have been discovered which are capable of providing full 3-D information of an object. Earlier methods had either

- loss of spatial information (e.g. standard X-rays, failing to discriminate between overlapping structures), or
- been highly invasive (for example 'serial-section microscopy', typically requiring the object to be frozen and then sliced up).

Important non-invasive imaging methods include

(1) *Ultrasound*. Very high sound frequencies (several MHz) allow beams to remain very narrow. These beams are typically sent / received by a small transducer which is held in contact with the body. It can rapidly change the direction of the beam, making it 'sweep' an angular domain. Echoes from interfaces between different soft tissue types are recorded, with time delays corresponding to depths. The technique is used for a large number of organs, including fetal monitoring during pregnancy. A potential future application - still requiring further developments - is to replace X-rays for mammography. Strong sound absorption by bones somewhat limits its use, for ex. in brain studies. The method gives images in real-time, and the equipment is

inexpensive. It used to be considered entirely safe, but some doubts about this emerged after a recent Swedish study showed that, after two scans, the likelihood of left handedness in babies increased by 32%. Although this is small compared to the 5 times increase that has been reported in cases of premature birth, the fact that it has any effect at all gives rise to concerns. However, present opinion seems to be that the benefits well outweigh the possible dangers.

(2) *CT - Computerized Tomography.* A parallel sheet of X-rays is sent through the object, and recorded by a 1-D row of detectors. From the accumulated data when source and receiver (or object) are rotated 180°, cross-sectional images can be computed. In medical application, the resolution is normally about 0.3 mm. With the use of much more intense X-rays (which would destroy living tissues; such X-rays can be obtained from accelerators in the form of synchrotron radiation), resolutions around 0.001 mm (= 1 μm) are achieved. This is comparable to the best resolution that is possible with optical microscopes, used on sliced samples. Some drawbacks with medical use of X-ray tomography include possible tissue damage from ionization (X-ray absorption depends on the target's electron density), and low contrasts between different types of soft tissues, for example between malignant and healthy tissues.

Mathematical tools needed for successful CT imaging were discovered more than once, not recognized for their potential and then forgotten before successful experimental realizations (employing less effective algorithms) were achieved. For independent pioneering work in experimentally realizing CT and bringing it to medical use, the 1979 Nobel Prize in Physiology and Medicine was awarded jointly to G. Hounsfield and A.M. Cormack. The history of CT and other applications of it are described in more detail in Section 1.3.

(3) *MRI - Magnetic Resonance Imaging* (earlier called NMR - Nuclear Magnetic Resonance). The object that is to be imaged is placed in a very strong, highly uniform magnetic field (e.g. inside a large superconducting magnet). Two different, relatively weak magnetic gradients are introduced — one

stationary and orthogonal to it, one that is stepped in time. When subjected to accurately tuned high frequency radio pulses, many light nuclei (with an odd number of nucleons, such as hydrogen) start to spin. While returning to a state of magnetic alignment, they re-radiate these waves. The frequency is proportional to the local magnetic field, i.e. it carries information about the positions of the different atoms. The numerical techniques needed to create images are similar to those used in CT. However, Fourier inversion is nowadays preferred over back projection-type algorithms.

Advantages of MRI over CT in medical applications include

- high contrast between many different soft tissues,
  - possibility (although little used) to 'tune in' on different atoms with very distinct biological functions (e.g. H^1, Na^{23}, and P^{31} resonate at 42.57, 11.26 and 17.24 MHz respectively in a field of 1 Tesla), and
    - far safer radiation (the frequencies are about 11 orders of magnitude lower than those of X-rays - the associated electromagnetic quanta carry correspondingly less energy, and cannot alter molecules of living tissues). In spite of using wavelengths in the 5—25 meter range, 1—2 mm resolution is obtained.

Disadvantages compared to CT include slightly less resolution and higher cost of equipment. In practical usage, the big risk factor turns out to be that inadvertently present metallic objects can become dangerous projectiles due to the extreme magnetic fields.

The Nobel Prize in Physics for 1952 was awarded to E. Purcell and F. Bloch (at Harvard and Stanford Universities) for their discovery of the NMR phenomenon. The problem of obtaining spatial information from NMR data was considered already in the early 50's and solved in different ways in the mid-70's. Routine medical use began in the mid-80's. Technology improvements have reduced recording times from hours to, in some cases, 30-100 ms for instance, when using echo-planar imaging (EPI)—a high-speed recording technique that permits a full image to be obtained in a single nuclear

excitation cycle, as opposed to a few hundred cycles; cf. [**?**]. A summary of the principles of MRI is given, for example, in [**?**].

(1) *PET - Positron Emission Tomography.* A radioactively labeled substance is injected and follows the blood stream, while emitting positrons. After traveling a very short distance, a positron will encounter an electron, and annihilate it. The energy gets transferred into two gamma rays that are sent off in nearly perfectly opposite directions of each other. When two detectors (out of a big array surrounding the body part—typically the head) detect signals at the same instant, the emission is assumed to have occurred along the straight line between them. This procedure generates data on the accumulated concentrations of the tracer substance along a large number of different lines through the body, thus allowing its distribution to be reconstructed through 3-D generalizations of the 2-D CT algorithms that are described in Sections 1.4—1.8.

When using radioactively labeled glucose, brain activities can be followed in 'real time', since the blood flow (and glucose usage) very quickly responds to areas of activity (however, EPI-type MRI, in connection with the use of contrast agents in the blood, offer competition to PET in this field). Another usage is based on the fact that certain substances tend to concentrate in different tissues, e.g. Cu^{64} can be used to spot some brain abnormalities. Disadvantages include quite low resolution, very high cost, and possibly dangerous radiation levels which are somewhat minimized by the use of radio-isotopes with short half-times. However, this requires the availability of a nearby reactor or accelerator.

The last two methods to be mentioned here are entirely non-invasive also as far as waves and radiation are concerned. However, their ability to provide true imaging is very limited. Both can record brain signals in cases when thousands of neighboring neurons fire in a synchronized manner.

(2) *EEG - Electroencephalography*; electric potentials on the scalp are recorded at tens (or more) locations with time resolutions in milliseconds. The low spatial resolution and the mathematically ill-posed inversion problem makes

the technique more important in studying neural firing patterns than for imaging.

(3) *MEG - Magnetoencephalography*; the very weak magnetic fields from neural activities are picked up outside the scull by SQUIDs (superconducting quantum interference devices), possibly the most sensitive recording devices of any kind. Using low temperature, liquid helium cooled, superconductors, individual flux quanta can be recorded. This can in turn be utilized for a variety of measurement tasks, giving astounding precisions, e.g.

magnetic field	10^{-15}T	(= 1 fT; femto-Tesla); signals from the brain reach about 10-100 fT (measured outside the skull); from the heart 50,000 fT; the earth's field is about 10^{11}fT = 10^{-4}T.
voltage	$10^{-14}V$	about 5 orders of magnitude better than semiconductor voltmeters,
motion	10^{-18}m	about 1/1,000 of the diameter of an atomic nucleus; 1/1,000,000 of the typical diameter of an atom.

'High temperature' (using liquid nitrogen at 77K) superconducting SQUIDs are much cheaper than liquid He-ones; however their ability to detect detect fields of around 25 fT is only barely sufficient for brain studies.

Although SQUIDs operate much faster than neurons, acceptable signal-to-noise ratios when applied to brain imaging require recording times in tens of seconds. Already in 1853, it was shown by Helmholtz that the inversion problem, determining internal currents from external magnetic fields, was not uniquely solvable. Like for X-ray crystallography (Section 30.2), additional data needs to be supplied. Possibilities for MEG include the use of

- simultaneous EEG-data for potentials. This offers the best signal when currents are orthogonal to the scull—the magnetically least visible case, and
  - MRI-provided structural information. This will pinpoint folds in the cortex. As it happens, primary sensor areas tend to be located in such

folds, with the consequence that the key currents become relatively parallel to the skull, i.e. well oriented for a good magnetic signal.

MEG is described in Hämäläinen (1993). One of the inventors of MEG (D. Cohen, MIT) has raised serious questions about the utility of the approach [**?**].

1.3. Additional Background on Computerized Tomography.

From a mathematical point of view, the main challenge in CT lies in the inversion technique. In a purely mathematical form (prompted by an issue relating to gravitational field equations), this problem was solved by the Austrian mathematician Johann Radon [**?**]. His solution assumes that all variables are continuous functions defined on an infinite domain. In practice, one has to work with a finite number of rays at a finite number of angles to produce a reconstruction at a finite number of grid points. In the exercise section of this book, we will see how Radon's inversion method connects to two of the presently most used techniques—*filtered back projection* and *Fourier inversion*.

The paper by Radon was just the first of several which gave solutions to the inversion problem which were not noted by later pioneers. Another case is a paper by R. Bracewell [**?**] (in the context of obtaining images of the sun from microwave data) which describes a Fourier-based reconstruction method. With the use of the FFT (fast Fourier transform) algorithm, Fourier based reconstruction methods are now the fastest ones available. Although surprisingly little used in medical contexts (where filtered back projection dominates), they are preferred in the even more data intensive application of *micro-tomography*.

Allan MacLeod Cormack (1924-1998) was working as a physicist (at University of Cape Town) and assisting a local hospital with routine radiological tasks, when it occurred to him that if enough X-ray projections were taken in a variety of directions, there would be enough data for a full reconstruction. He realized that CT could revolutionize medical imaging and his tests on simple wood and aluminum objects in the late 1950s and early 1960s, showed the concept to be practical. In two seminal papers [**?, ?**], Cormack very clearly outlined the medical implications, and presented still another numerical procedure for the reconstruction. However, his efforts at the time to interest the medical community were not successful.

About a decade after Cormack's pioneering work, Goodfrey Hounsfield (1919-) independently developed the idea of medical tomography (while doing pattern recognition studies at the electronics company EMI Ltd. in Britain). His first apparatus was similar to Cormack's, but used an americum radiation source and a crystal detector. Following very successful preliminary tests, the radionuclide source was replaced with an X-ray tube, reducing data gathering times from well over a week to about 9 hours. Following many further improvements, his work led to the first clinical machine, installed in a hospital in Wimbledon in 1971. By this time the technique had advanced to the point that 180 projections (at 1° separation) could be collected in just under 5 minutes, followed by about 20 minutes for the image reconstruction. These specifications improved even more and current machines provide about 0.3 mm resolution throughout full body slices. The major limiting factor in further improvements comes from the need to keep X-ray doses within safety limits.

There are many applications other than medical ones, of tomography. Below are just a few examples:

astronomy	Marsh and Horne [1988] (binary stars),Gies et al [1994](accretion discs) Hurlburt et.al.[1994] (coronal studies)
oceanography	Worcester and Spindel [1990],Worchester et.al. [1991] Munk et.al. [1995] (acoustic probing of ocean conditions
geophysics	Anderson and Dziewonski [1984] (mantle flows), Frey et al [1996] (aurora) Gorbunow [1996] (atmosphere)
porous media	Hal [1987]

1.4. Model Problem.

Figures 1.4.1 and 1.4.2 show a test object, defined on a 63×63 grid. This object was generated by the code logo.m. The X-ray absorption levels at different locations are displayed as darkness and elevation respectively. Figure 1.4.3 shows how X-ray data can be collected for a sequence of angles $\theta_i = \frac{\pi\, i}{64}, i = 0, 1, ..., 63$. Figure 1.4.4 shows what the scan data would look like in the case of the test object in Figures 1.4.1 and 1.4.2. The scan lines are shown as successive lines (in the r-direction)

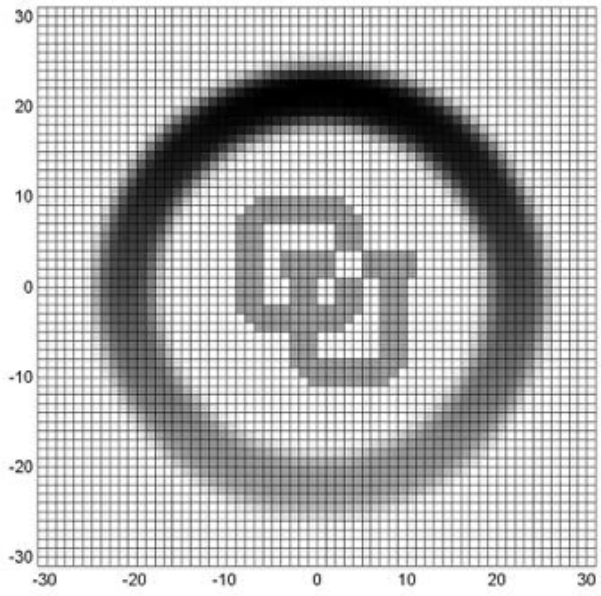


FIGURE 1.4.1. Test object.

from the front left, $\theta = 0$, to the back right, $\theta = \pi$; this last line being an up-down reflection of the first one.

Given the density function $f(x, y)$ of the 2-D object, the scan data can be written as

$$g(r, \theta) = \int_{-\infty}^{\infty} f(x, y)\, ds \tag{1.1}$$

where the coordinate axes are as defined in Figure 1.4.5. Since the (s, r) axes differ from the (x, y) axes by a pure rotation, they are related by

$$\begin{cases} s = x\cos\theta + y\sin\theta \\ r = -x\sin\theta + y\cos\theta \end{cases} \qquad \begin{cases} x = s\cos\theta - r\sin\theta \\ y = s\sin\theta + r\cos\theta \end{cases} . \tag{1.2}$$

This means that we sum all the contributions along lines parallel to the s axis in Figure 1.4.5.

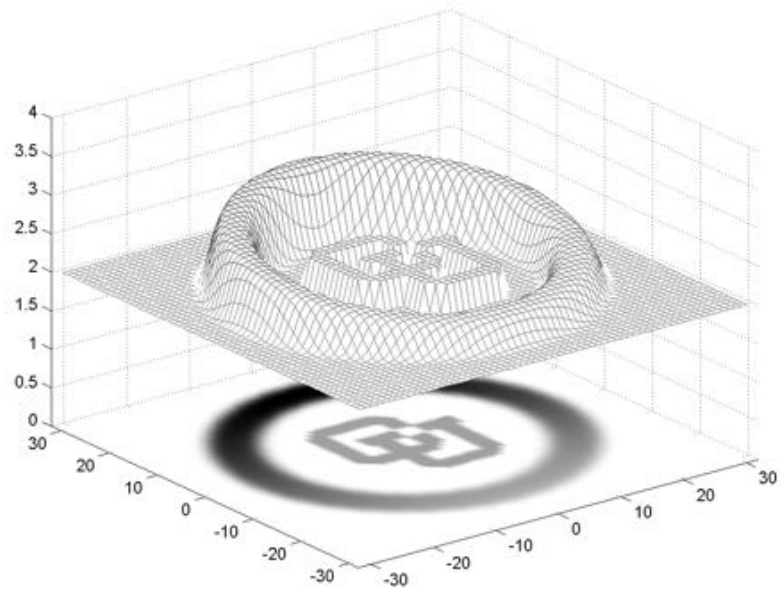


FIGURE 1.4.2. Another view of test object.

Equation (1.1) is known as the *Radon transform*. The computational issue in CT is to invert this transform, i.e. to recover $f(x, y)$ from $g(r, \theta)$.

1.5. Least Squares Approach.

To understand the idea behind the back projection method (how to achieve the reconstruction, how much data is needed etc.) we consider first a very small object—a 3×3 structure of 9 square elements, having unknown densities $x_1,\ x_2, \ldots, x_9$ respectively:

(1.1)
$$\begin{array}{|c|c|c|}\hline x_1 & x_4 & x_7 \\ \hline x_2 & x_5 & x_8 \\ \hline x_3 & x_6 & x_9 \\ \hline \end{array}\ .$$

Suppose that we have knowledge only of the row sums r_1, r_2, r_3 and of column sums s_1, s_2, s_3 (like having done X-ray recordings only horizontally and vertically). This

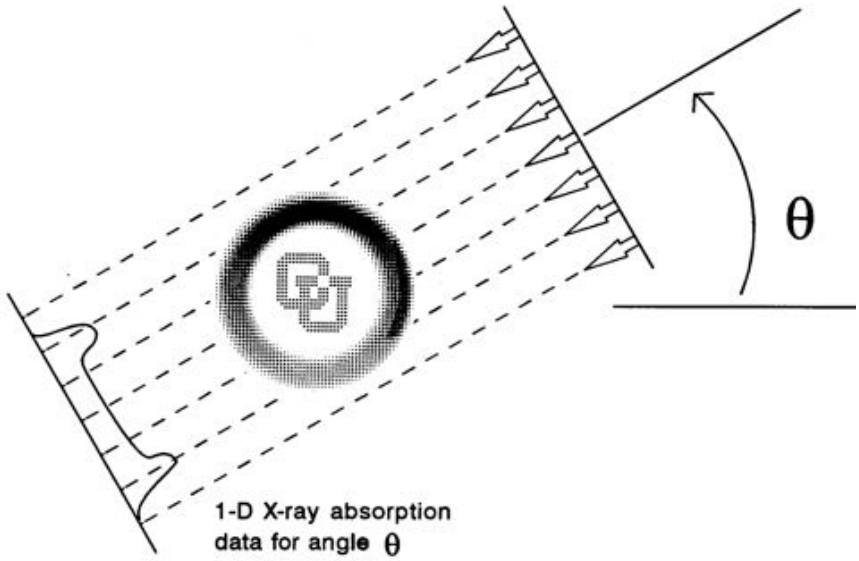


FIGURE 1.4.3. Principle for generation of 1-D scan data from a 2-D object.

gives rise to the linear system of equations

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} \tag{1.2}$$

The first question that we need to ask ourself is whether it is possible to obtain the densities $x_1,\ x_2, \ldots, x_9$ of the elements from these row- and column sums only. Considering that all the six density patterns shown in Table 1 have the same row- and column sums, it is clear that the problem will not always have a unique answer. Next, we might ask if (1.2) will always have at least one solution, no matter what

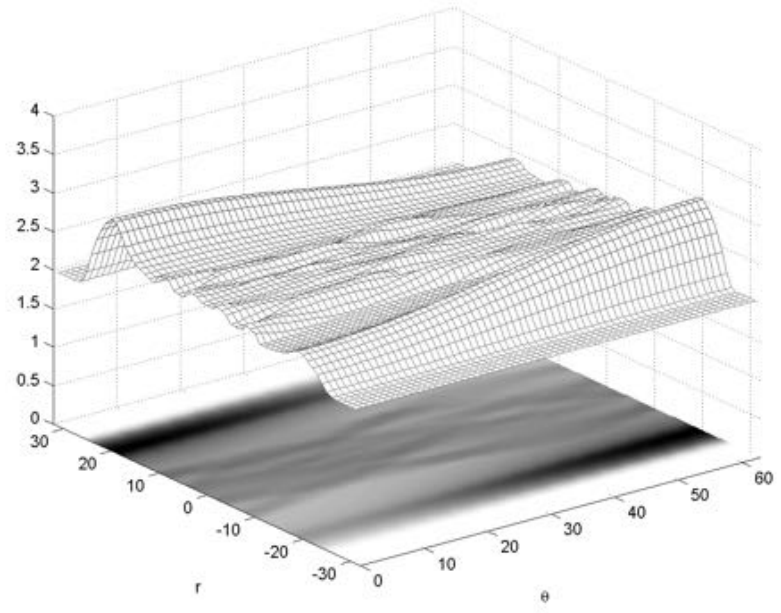


FIGURE 1.4.4. Scan data of the CU-object.

the values are in the right hand side. It is easy to see this can't be the case (in spite of the fact that this system $Ax = b$ has fewer equations than unknowns). If we add rows 1, 2, and 3, we get

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = r_1 + r_2 + r_3$$

while adding rows 4, 5, and 6 gives

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = s_1 + s_2 + s_3$$

Unless

$$r_1 + r_2 + r_3 = s_1 + s_2 + s_3 \tag{1.3}$$

holds, there is no possibility for a solution to exist. We might argue that (1.3) should hold if our data came from any object, such as the one indicated in (1.1). However, all actual data contains errors of some size, and one can never rely on data having to be completely error free. For much bigger linear systems than (1.2), there is no

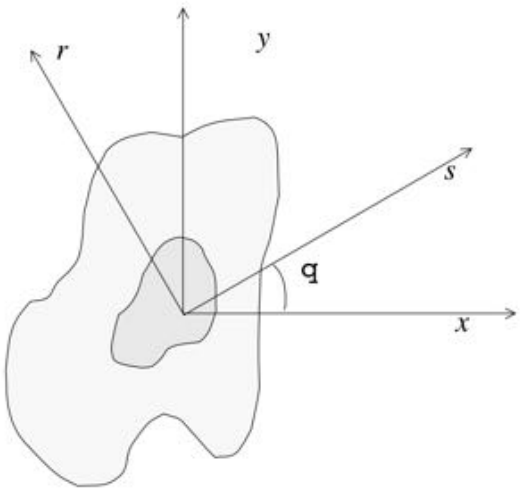


FIGURE 1.4.5. Relation between the (x, y) and (s, r) coordinate systems.

1	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1
0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	0
0	0	1	0	1	0	0	0	1	1	0	0	0	1	0	1	0	0

TABLE 1. The six possible permutation matrices of size 3x3

chance to spot multiple solutions or situations with non-existence of solutions in the way we have just dome. What is needed are general results telling just when systems have solutions (and, if so, how many) or do not have any. If there are solutions, how does one effectively find them? Maybe surprisingly, systems with more equations than unknowns—almost invariably lacking exact solutions altogether—is the most important case in applications. And we will soon see that our first approach to tomographic inversion is an illustration of this.

1.5.1. SVD analysis of (1.2). Usually the best way to explore the solvability of any specific linear system starts with performing an SVD factorization of the coefficient matrix A (cf. Section ....; in particular note Figure .... illustrating how the decomposition related to the four fundamental subspaces of a matrix) Writing this decomposition as $A = U\,\Sigma\,V^*$, we get in the particular case of the coefficient matrix A in (1.2)

$$U = \begin{bmatrix} -0.4082 & 0 & 0 & 0.8165 & 0 & 0.4082 \\ -0.4082 & 0 & -0.2599 & -0.4083 & 0.6576 & 0.4082 \\ -0.4082 & 0 & 0.2599 & -0.4083 & -0.6576 & 0.4082 \\ -0.4082 & 0.5393 & -0.5701 & 0 & -0.2253 & -0.4082 \\ -0.4082 & -0.8006 & -0.1493 & 0 & -0.0590 & -0.4082 \\ -0.4082 & 0.2612 & 0.7194 & 0 & 0.2843 & -0.4082 \end{bmatrix},$$

$$\Sigma = \begin{bmatrix} 2.4495 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.7321 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.7321 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.7321 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.7321 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$V^* = \begin{bmatrix} -0.3333 & 0.3114 & -0.3292 & 0.4714 & -0.1301 & 0.6327 & 0.1464 & 0.1503 & -0.0096 \\ -0.3333 & -0.4622 & -0.0862 & 0.4714 & -0.0341 & -0.4331 & 0.2297 & 0.1479 & 0.4269 \\ -0.3333 & 0.1508 & 0.4154 & 0.4714 & 0.1642 & -0.1997 & -0.3762 & -0.2981 & -0.4174 \\ -0.3333 & 0.3114 & -0.4792 & -0.2357 & 0.2496 & -0.2662 & -0.5409 & 0.1831 & 0.2179 \\ -0.3333 & -0.4622 & -0.2363 & -0.2357 & 0.3456 & 0.3042 & 0.0479 & -0.5903 & -0.0332 \\ -0.3333 & 0.1508 & 0.2653 & -0.2357 & 0.5438 & -0.0380 & 0.4930 & 0.4073 & -0.1846 \\ -0.3333 & 0.3114 & -0.1791 & -0.2357 & -0.5098 & -0.3665 & 0.3945 & -0.3333 & -0.2083 \\ -0.3333 & -0.4622 & 0.0638 & -0.2357 & -0.4137 & 0.1288 & -0.2777 & 0.4425 & -0.3937 \\ -0.3333 & 0.1508 & 0.5654 & -0.2357 & -0.2155 & 0.2377 & -0.1168 & -0.1091 & 0.6020 \end{bmatrix}.$$

The last entry (zero) in the main diagonal of Σ tells that the rank of the system is 5 (rather then 6, as might have been expected). The first 5 columns of U form then an orthogonal basis for the column space of A, i.e. the system (1.2) is solvable if and only if the right hand side b of (1.2) lies in this space. The columns of U are orthogonal - so another way to say this is that b needs to be orthogonal to the last

column of U, giving the condition for solvability

$$r_1 + r_2 + r_3 - s_1 - s_2 - s_3 = 0\ . \tag{1.4}$$

We arrived, mainly by chance, earlier at this very same requirement (1.3). The big difference is that we now have derived it in a manner that works for absolutely any system. And we also now see that this is the only requirement that is needed for solvability. Physically, this condition is natural: $r_1 + r_2 + r_3$ and $s_1 + s_2 + s_3$ both express the same quantity, viz. the sum of all the nine unknowns. If (1.4) holds, the general solution is any one particular solution to which we can add any combination of vectors from the null space of A. By the same theory about SVD and the fundamental subspaces, this space will be found in the last 4 rows of V^*.

Once we have the SVD of A, we can alternatively extract all the information above - plus find a particular solution—without referring to the fundamental subspaces. The system $Ax = b$ can be written $U\Sigma V^* x = b$, i.e.

$$\Sigma y = U^* b \tag{1.5}$$

with

$$x = Vy. \tag{1.6}$$

In order for (7.1) not to contain a contradiction in the last row, we need the last element of U^*b to be zero, leading once again to the condition (1.4). With that being the case, (7.1) gives uniquely the first 5 entries of y, but leaves the last 4 free. The most general solution then follows from (7.4). We note again that the solution is undetermined precisely with respect to any combination of the last 4 rows of V^*. This last result is very much more complete than our earlier observation that the six particular solutions represented by the matrices in Table 1 all corresponded to the same RHS for (1.2).

1.5.2. A least-square formulation of the CT inversion problem. The most obvious shortcoming with using only row- and column sums for the model (1.1) is that we do not get enough data for the number of unknowns. Increasing the resolution from 3×3 to $n \times n$ elements does not help; we will then only have $2n$ equations for n^2 unknowns. The 'obvious' remedy to this is to do the scans in many

more directions than just two. There is no limit to how many directions we can use. With m directions and, for each of these, sending n side-by-side rays through an object made up of $n \times n$ elements, we will instead of the $n = 3$, $m = 2$ case represented by (1.2), obtain a linear system of the type

$$\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & A & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}_{mn,n^2} \begin{bmatrix} \cdot \\ x \\ \cdot \end{bmatrix}_{n^2} = \begin{bmatrix} \cdot \\ \cdot \\ b \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}_{mn} . \tag{1.7}$$

If we have more equations than unknowns, the system becomes overdetermined, in which case there is typically no exact solution. However, least squares solutions can be readily found (cf. Section 11.4), making the difference between left hand– and right hand sides of the system as small as possible. Typically, using $m >> n$ will not be disadvantageous (as one might think, based on introducing more possibilities of conflict) but instead advantageous (by making the solution more stable against data noise). The entries in the coefficient matrix are now not only 0's or 1's (as in (1.2)), but instead real numbers which reflect how visible each element is to each ray. When we only used horizontal and vertical rays, they either went through the full extent of an element, or they missed the element altogether. With rays going thorough the block-structured object at arbitrary angles, the length of the intersection between an element and a ray can take any value between zero and the length of its diagonal. The coefficient matrix A will be quite sparse, and with a complicated sparsity structure.

With the object consisting of $n \times n$ elements (whose densities are to be determined), the detailed structure of the system (1.7) becomes as shown in Figure 1.5.1. There are n^2 densities that need to be calculated. We rearrange the $n \times n$ set of unknowns as before into a column vector $\mathbf{x}$ of length n^2. The first n equations correspond to the n rays when scanning at the first angle; the next n equations corresponding to the second angle, etc. It is natural to think of A as made up of an $m \times n$ layout of $n \times n$-sized blocks, and x and b of n and m end-to-end vectors,

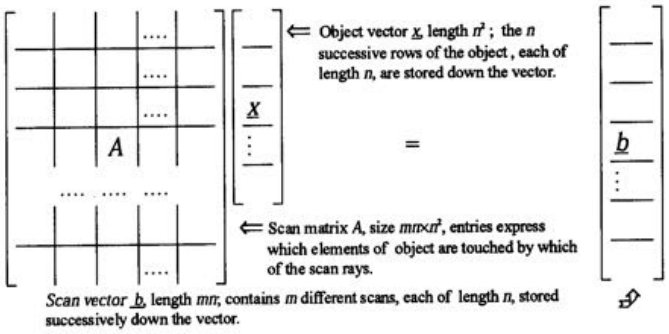


FIGURE 1.5.1. Structure of the overdetermined linear system of the least-square method.

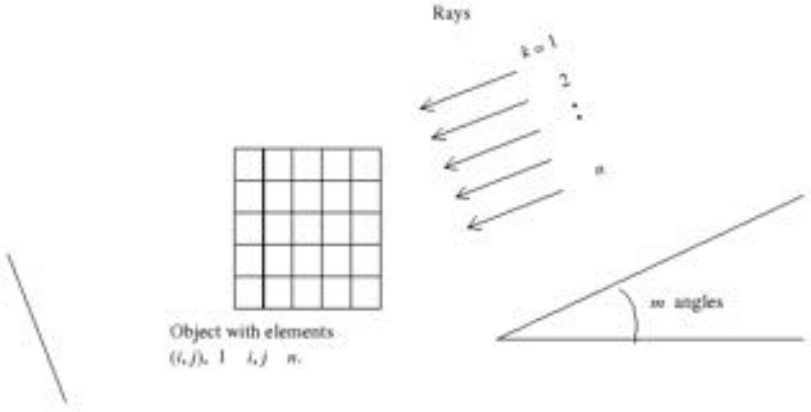


FIGURE 1.5.2. Illustration of how the A-matrix entries are formed.

respectively, each of length n. Figure 1.5.2 illustrates how the linear system is formed (with one block row of A for each scan angle).

1.5.3. Least square solution. We now need to solve the overdetermined system (1.7) in a least squares sense. The best approach will be to use an iterative method that is capable of fully utilizing the sparsity structure of A. One such algorithm is Matlab's *lsqr*—a conjugate gradient implementation of the normal equations. In Section 11.3 it is described how a direct solver can be implemented in a stable

way using QR factorization. Codes for both of these methods are included in the computer code section .... For a simple operation count for a direct solver, we consider the *normal equation* approach (roughly the same count as for the QR method, but less stable numerically, so not recommended). The normal equations of a linear $mn \times n^2$ system $Ax = b$ is a square $n^2 \times n^2$ system

$$A^T Ax = A^T b \ .$$

Forming $A^T A$ will cost $O(mn^5)$ operations, $A^T b$ costs $O(mn^3)$; Cholesky decomposition of $A^T A$ into $L\ L^T$ will add another $O(n^6)$. The final step to get x through two back substitutions, using L and L^T respectively, adds a further $O(n^4)$ operations. Assuming m is just slightly larger than n, the total cost becomes $O(n^6)$ operations.

A very large saving can be realized by noting that the A-matrix is independent of the object we are studying—that will only affect the b- and x- vectors. The L-matrix can therefore be determined once and for all. Each image will then cost 'only' $O(n^4)$ operations. We will soon see that even this is not competitive—the next two methods, filtered back projection and the Fourier method will cost only $O(n^3)$ and $O(n^2 \log n)$ respectively. Table 2 compares the computational time needed by the different approaches.

Hounsfield used the least squares approach in his first successful experimental inversion. Already with a sample as coarse as 8×8, the necessary computing took hours on EMI's then state-of-the-art ICL machine. The code least_sq is a direct implementation of this approach. Figure 1.5.3 shows the result if our test object is scanned with 31 rays and 40 angles, followed by reconstruction with this method to a 31×31 grid. This level of resolution is too low for a good reconstruction, but we can still see a rough version of the ring and the central lettering. With the iterative solver least_sq_iter we can easily afford a 63×63 inversion on a standard PC. The result of this inversion is shown in Figure 1.5.3

Current medical imaging uses the filtered BP-method, to be described next.

1.6. Back Projection method.

The idea of back projection is conceptually very straightforward, it is easy to implement, and the computational cost is moderate. In its most direct form, the

Comparisons of times for algorithms of different complexity on a system performing 10^8 floating point operations per second

Size $n \times n$ pixels	n^6 (least squares)	n^4 (least squares)	n^3 (filtered BP)	$n^2 \log n$ (Fourier)
$n = 100$	3 h	1 s	0.01 s	0.2 ms
$n = 1000$	320 y	3 h	10 s	0.03 s

TABLE 2. Comparison of computational efficiencies for some non-iterative inversion methods.

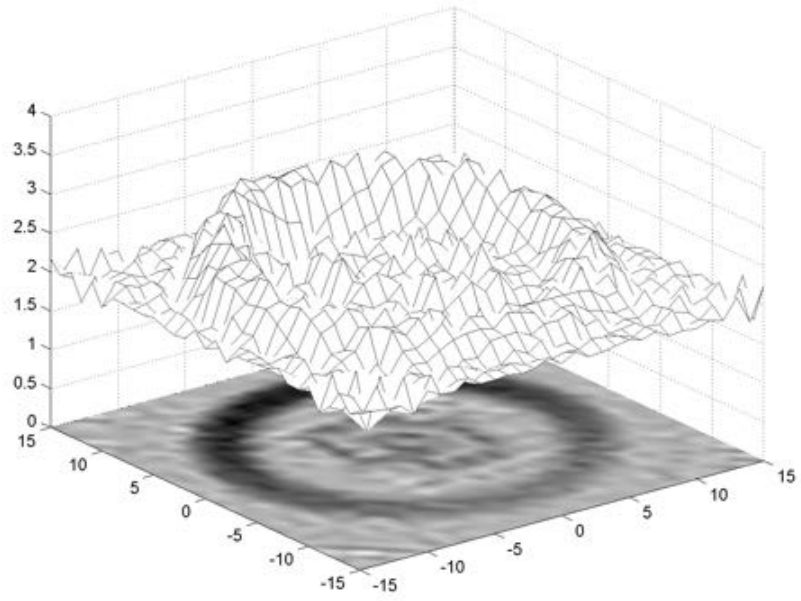


FIGURE 1.5.3. Reconstruction with the least squares method when the logo was scanned with 31 rays at 40 angles, followed by a reconstruction to a 30×30 grid.

reconstruction comes out 'smeared'. However, the addition of a simple filter all but resolves this. It is of little surprise that Filtered Back Projection (FBP) has become the most widely used reconstruction process in the medical community where it very well meets the requirements placed on it.

For an $n \times n$ image reconstruction, the FBP method will cost $O(n^3)$ operations. In many contexts, this cost is acceptable or rather, it became accepted at a time when the major alternatives were even more costly. The computing time using FBP is quite fast in comparison to the other tasks involved such as patient handling etc. However, in an application such as micro-tomography, the situation is very different. There the resolution can be as high as 1000 simultaneously recorded slices, each to be imaged on a 1000×1000 grid, giving a 3-D 1μm resolution throughout a cubic millimeter sample. Each full inversion for such a cube of using back projections would cost on the order of 10^{12} operations. This is likely to become a slow process in comparison with the rapid one of automated sample handling where data for the 1000 slices are collected simultaneously by using a 2-D rather that a 1-D array of X-ray sensors. We describe in Section 1.7 an inversion algorithm which cuts this cost by some orders of magnitude—in this case to about 10^{10} operations.

1.6.1. Immediate back projection. Figure 1.6.1(a) shows a point-type object, and 1.6.1(b) its scan data. Let us remind ourselves how the scan data is obtained. At each angle, the total absorption of each ray is recorded; we do not have any information about the contribution from any specific location. The most obvious thing to do is to assume that all locations along the ray contributes an equal amount. As illustrated in Figure 1.6.2 the simplest form of back projection consists of drawing parallel bands across the image area, with the darkness of the band corresponding to the absorption that was recorded for each ray. Mathematically back projection is described by

$$h(x, y) = \int_0^\pi g(r, \theta) d\theta. \tag{1.1}$$

The right part of Figure 1.6.1 shows the result of this process in this case of the point object. The only error is that the point has turned into a 'smeared out' cone-type mound. The sharp edge of the original point object has been lost, as areas near the point are also covered by some of the bands. However, the position of the recovered mound is precisely the same as that of the original point-object. Also, the amplitude and shape of the mound is position invariant—it takes the same values wherever the original point object was located.

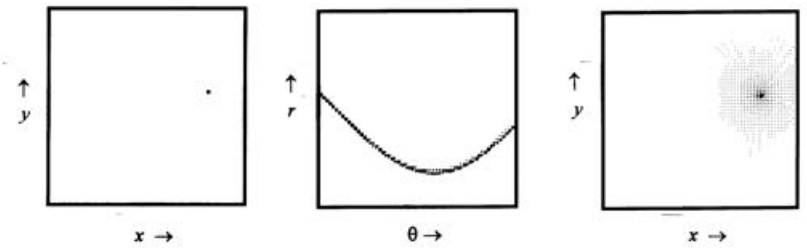


FIGURE 1.6.1. Point-type object, its scan data, and the image recovered through immediate back projection.

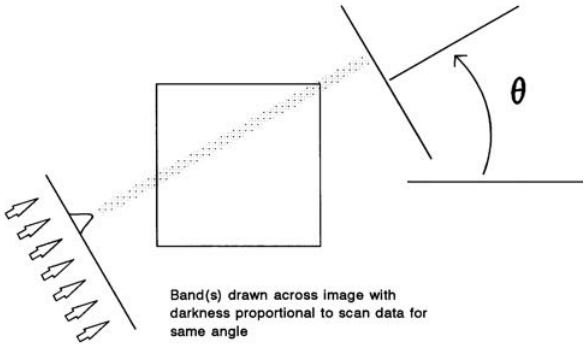


FIGURE 1.6.2. Principle behind back projection (when applied immediately to scan data - no filtering) shown here in the case of a point object.

To appreciate the significance of this example with a point object, we need to note that both the scanning and the back projection phases are *linear*. If there had been two point objects, the scan data would just have been the *sum* of the scan data for the two objects if recorded independently. Similarly, the back projection produces in that case a result which is the sum of the back projections of the two objects, if they were treated separately. Finally, the darkness of the back projected result of each object by itself is proportional to the original darkness. The process of scanning followed by back projection satisfies the two criteria for a function (here

a matrix-valued function with a matrix input) to be *linear*:

$$\begin{aligned} f(x+y) &= f(x)+f(y) \\ f(\alpha x) &= \alpha\, f(x). \end{aligned}$$

From this linearity follows the important conclusion that the reconstruction must also work for full images and not just point objects.

In summary, wherever the imaged object x had a gray pixel, the image f will feature a smeared one centered at the same location, and with a darkness proportional to the darkness of the one in the original. From the linearity follows that if x was a sum of two images with a different gray pixel in each, f will become the sum of the two corresponding images. Continuing this observation: Since every image is a combination of pixels, this linearity implies that f must become a (smeared) representation of the original object. Immediate back projection using the scan data shown in Figures 1.6.1 leads to the reconstruction seen in Figure 1.6.3. The original object is recovered but smeared out. In the next section we discuss a simple method of improving the situation.

1.6.2. Filtered back projection. Comparing the original object in Figure 1.6.1 with Figure 1.6.3 (both featuring the same grid density of 63×63 points), we clearly see a loss in sharpness. Hence, we look for some way to enhance the output from direct back projection in order to reduce the smearing. The step from original object to scan data is essentially outside our control (dictated by X-rays and physics). Options remaining include

- Based on the smeared image, apply some filter which sharpens all gradients (such filters can for example be based on FFTs), and
- Since the cause of the smearing is understood (for the point image), try to alter the scan data in a way that off-sets the back projection smearing.

Both options above are viable; filtered back projection pursues the second one. Two key questions become:

(1) Does there exist any special type of (simulated) scan data for which the back projection method will give a nearly point like result?

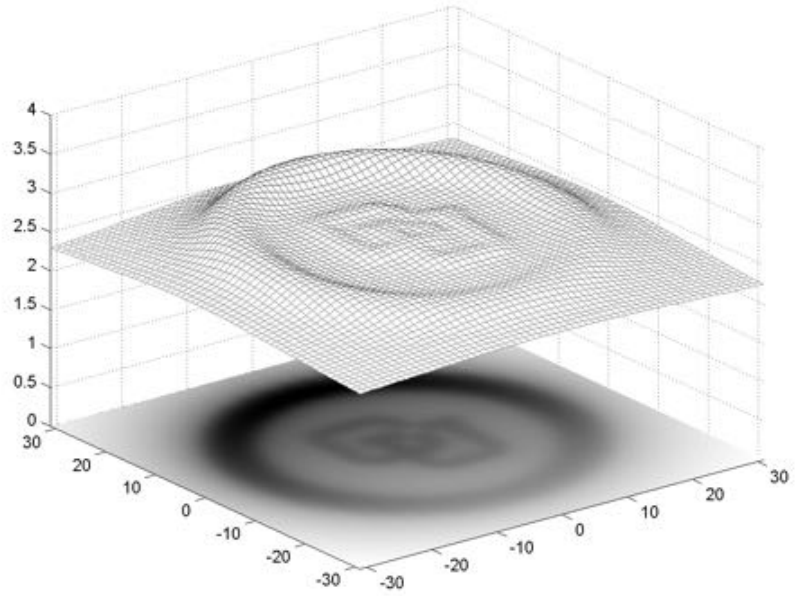


FIGURE 1.6.3. Immediate back projection of the scan data for the test object.

(2) Is there any operator—linear, location preserving, and not altering the darkness at the point itself—that we can apply to turn the actual scan data for the point-type test object into the form that we looked for in point 1 above?

Addressing the first issue, we note that we can replace the 'single-hump' data by a 'hump' at the same place, but with a bright band on each side of it. The contributions for all the angles will still superimpose to an equally dark spot at precisely the desired location, but at nearby locations, the bright sidebands might just cancel some of the undesired darkness, as the contributions for different angles θ are superimposed.

To turn the scan data for a fixed angle, $(0,0,...,0,1,0,...,0)^T$, into a vector with a bright (negative) entry on each side of the 'one'—wherever it is located—can be achieved by multiplying the scan data from the left by a symmetric $n \times n$ tri-diagonal

Toeplitz band matrix

$$E = \begin{bmatrix} 1 & -\beta & & & & & & \\ -\beta & 1 & -\beta & & & & & \\ & -\beta & 1 & -\beta & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & -\beta & 1 & -\beta & \\ & & & & & -\beta & 1 & -\beta \\ & & & & & & -\beta & 1 \end{bmatrix} \tag{1.2}$$

Applying this idea to the test object, Figure 1.4.1, the results are shown in Figure 1.6.4. To get these figures, we multiplied every single scan data vector with this matrix E, before back projecting where of course, we exploit the sparse structure of E.

This modification preserves

- location and strength of images of point objects—they will appear less smeared, and
- linearity, meaning that a general image will be as good as is the treatment of point objects. Thus linearity also implies that an optimal value of β should be obtainable from considering a point object, a question to be returned to in Section 1.8.

Choosing β= 0.3, 0.4, 0.5 and 0.6 respectively in (1.2) gives, with the scan data of our test object, the reconstructions that are shown in Figures 1.6.4. In spite of the filter being narrow (tri-diagonal; wider filters could be 'tuned' better) and the optimization of the filter coefficient being crude—we simply pick the best-looking of the four cases, we get excellent inversions.

1.7. Fourier transform method.

The Fourier transform (FT) method requires considerably more mathematical background than did the previous two methods. Computationally, it is the fastest known method. In the exercises we shall see how one can mathematically derive from this method both the filtered back projection method and Radon's original inversion

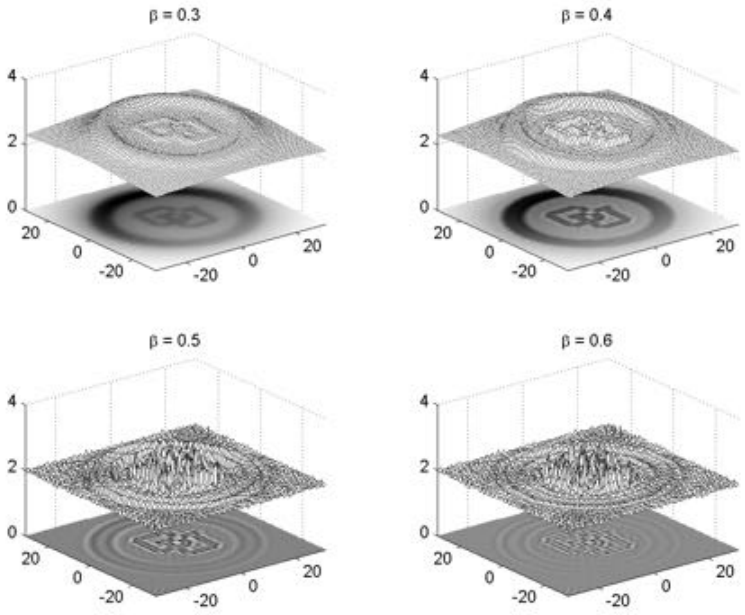


FIGURE 1.6.4. Filtered back projection with some choices of simple tri-diagonal filters.

formula (which, as we noted before, is mathematically compact but numerically impractical).

1.7.1. Analytical description. We assume as before that the density of the object is represented by a density function $f(x, y)$ where x and y denote the two spatial directions. The 2-D Fourier transform of the density function is given by

$$\hat{f}(\omega_x, \omega_y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, e^{-i\omega_x x}\, e^{-i\omega_y y}\, dx\, dy. \tag{1.1}$$

The density function $f(x, y)$ can then be recovered by

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(\omega_x, \omega_y)\, e^{i\omega_x x}\, e^{i\omega_y y}\, d\omega_x\, d\omega_y \tag{1.2}$$

We will give two descriptions of how one can arrive at the FT method. The most heurisic one follows below. A more concise but less intuitive approach is given in the exercise section.

The key step is to turn the scan data into $\hat{f}(\omega_x, \omega_y)$. This rests on two observations:

- Noting what happens if we send in the X-rays in a direction parallel to the x-axis: The left part of Figure 1.7.1 illustrates how we obtain the scan data $g(y) = \int_{-\infty}^{\infty} f(x,y)dx$. Its 1-D Fourier transform is

$$\begin{aligned}
\hat{g}(\omega_y) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} g(y)\, e^{-i\omega_y y}dy \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x,y)dx\right] e^{-i\omega_y y}dy \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i0x}e^{-i\omega_y y}dxdy \\
&= 2\pi\hat{f}(0,\omega_y),
\end{aligned}$$

  i.e. we have obtained $\hat{f}(\omega_x, \omega_y)$ along a vertical line through origin in the (ω_x, ω_y)-plane (cf. Figure 1.7.1(b)).
- Considering the difference if the X-rays would have entered from another direction. If the X-rays had entered from an angle θ (Figure 1.7.3(a)), the collected scan data will be exactly the same as if instead, the object had been turned an angle of $-\theta$ (Figure 1.7.3(b)). As is shown in Section 7.3 on Fourier transforms, turning an object turns its Fourier transform through exactly the same angle. Therefore, we obtain in this case values for $\hat{f}(\omega_x, \omega_y)$ along the line through the origin in the (ω_x, ω_y)-plane orthogonal to the X-ray direction in the physical plane (Figure 1.7.3(c), (d)). When we have taken X-ray images for $0 \le \theta < \pi$ (and applied the 1-D Fourier transform to them), we have actually obtained $\hat{f}(\omega_x, \omega_y)$ along all radial lines through the origin, i.e. throughout the complete (ω_x, ω_y)-plane. Thus we find that

$$\hat{f}(\omega_r, \theta) = \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty} g(r,\theta)e^{-i\omega_r r}dr. \tag{1.3}$$

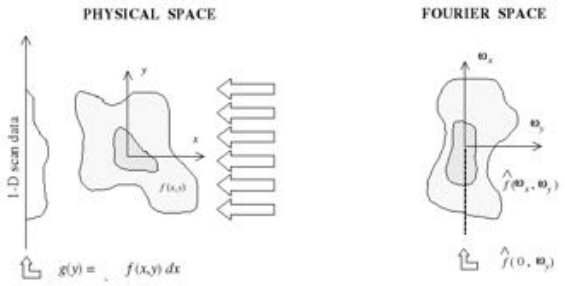


FIGURE 1.7.1. Recording with rays parallel to the x-axis, and the corresponding data set in Fourier space.

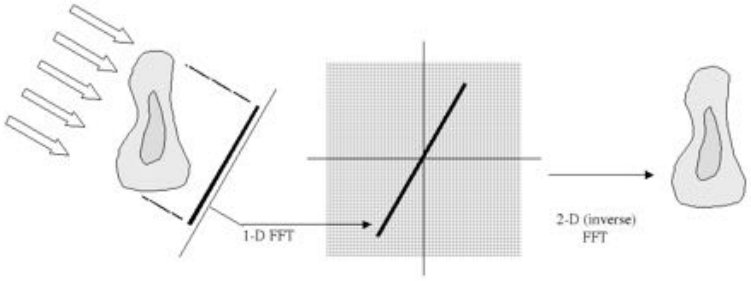


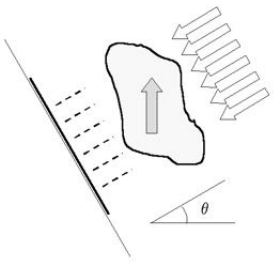
FIGURE 1.7.2. Schematic illustration of the steps in the Fourier reconstruction method.

The density function $f(x, y)$ is then recovered by the 2-D Fourier transform (1.2).

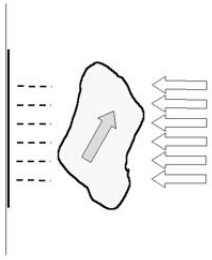
1.7.2. Numerical results with FT method. Following the description above gives a reconstruction as shown in Figure 1.7.4.

Although the details in the center is near-perfect, we see a quite disturbing wobble in the base level of the reconstruction. The FT method that is implemented in the Matlab code therefore contains one more refinement. Each scan vector is 'padded' to double length by just adding zeros at each end of it. Once brought to Fourier space, it is laid out on a correspondingly enlarged 2-D (ω_x, ω_y)-plane. After returning (by

a. Scan with incoming X-rays at an angle θ. The density of the object is $f(x, y)$.



b. Equivalent recording as in a. The image is now rotated through an angle θ, and the X-rays enter horizontally.



c. The $1D$ Fourier Transform of the recording of part b provides a line of data along the ω-axis of the $2D$ Fourier Transform.

d. Since the FT rotates through an angle θ when the image rotates through the same angle, we have now obtained a function $\widehat{f}(\omega_x, \omega_y)$ along a line in the (ω_x, ω_y) plane, sloping the same way as the scan line of part a.

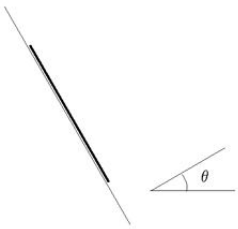


FIGURE 1.7.3. Fundamental principle behind the FT method for tomographic reconstruction.

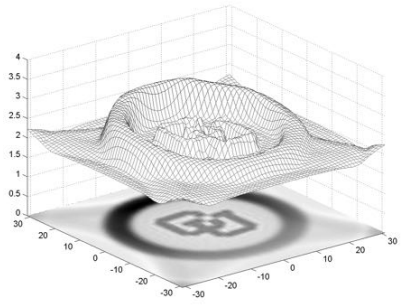


FIGURE 1.7.4. Reconstruction by direct FT method from 63 ray, 64 angle scan data to a 64×64 grid.

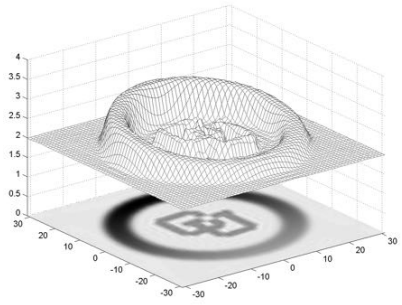


FIGURE 1.7.5. Same Fourier reconstruction as in the previous figure, but with the spatial domain 'padded' by a factor of two within the FT algorithm.

the inverse 2-D FFT) to the (x,y)-plane, we keep only the central square; i.e. disregard the borders (containing 3/4 of the total reconstructed area—these borders only contain an image of the padding areas). The resulting picture is seen in Figure 1.7.5.

There are a couple of ways to understand why this padding idea helps:

- Very heuristically: We are using periodic FFTs instead of infinite-domain transforms. Periodic images of the object are then present near the boundaries of the shown domain. Discrepancies between the concept of periodicity in polar- and in Cartesian coordinates cause a difficult-to-analyze error pattern. From this loose argument, one might expect that padding would improve the image by increasing the distances to unphysical ghost images.

- More theoretically: Extending the spatial domain by a factor of two means that in each direction a twice as *dense* set of Fourier modes become available. For example, in a 1-D spatial domain of [-π, π], modes

$$\dots e^{-3ix},\ e^{-2ix},\ e^{-ix},\ e^{0ix},\ e^{1ix},\ e^{2ix},\ e^{3ix}, \dots$$

are available. If the domain is extended to $[-2\pi, 2\pi]$, also the intermediate modes

$$\dots e^{-\frac{5}{2}ix},\ e^{-\frac{3}{2}ix},\ e^{-\frac{1}{2}ix},\ e^{\frac{1}{2}ix},\ e^{\frac{3}{2}ix},\ e^{\frac{5}{2}ix} \dots$$

become present. The interpolation from polar to Cartesian grids occurs in Fourier space and with a denser grid in that space, interpolation becomes more accurate.

It is important to use better than-linear-interpolation and our code uses cubic interpolation. We can illustrate this in 1-D by trying to represent a half-integer mode $e^{i(n+\frac{1}{2})x}$ on a grid in Fourier space that only has integer modes e^{ikx}, $k = \dots, -3, -2, -1, 0, 1, 2, 3, \dots$ available. Linear (second order) interpolation gives

$$e^{i(n+\frac{1}{2})x} \approx \frac{1}{2}e^{inx} + \frac{1}{2}e^{i(n+1)x} = e^{i(n+\frac{1}{2})x}\ \left(\frac{e^{-i\frac{1}{2}x} + e^{i\frac{1}{2}x}}{2}\right) = e^{i(n+\frac{1}{2})x}\ \cos\frac{x}{2}$$

and fourth order interpolation (cf. Table 2 of Section 12.5)

$$e^{i(n+\frac{1}{2})x} \approx -\frac{1}{16}e^{i(n-1)x} + \frac{9}{16}e^{inx} + \frac{9}{16}e^{i(n+1)x} - \frac{1}{16}e^{i(n+2)x} = e^{i(n+\frac{1}{2})x}\ (\frac{9}{8}\cos\frac{x}{2} - \frac{1}{8}\cos\frac{3x}{2})$$

The interpolation would have been perfect, had the factors (referred to as the damping factors) multiplying $e^{i(n+\frac{1}{2})x}$ in the RHSs in the two equations above been equal to one. Figure 1.7.6 displays the actual factor for interpolation of different orders. Note that this factor depends on x only, i.e. not on n. Interpolation by order 4 and above achieves excellent results in the center of the domain. Hence, we can expect good reconstruction where, in the padded case, the object is located.

Although a factor two padding leads internally to a larger grid, the operation count still remains $O(n^2 \log n)$, only the proportionality constant has increased around a factor of four.

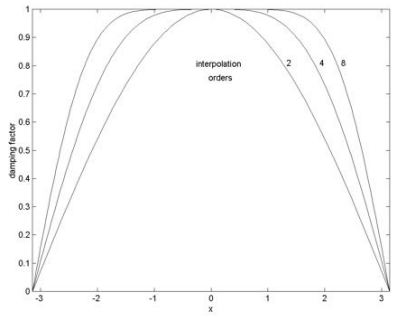


FIGURE 1.7.6. The damping factor at different physical locations across the domain when a half-integer Fourier mode is interpolated *in Fourier space* to integer frequencies.

1.8. Filtered BP method derived from the FT method.

The Fourier inversion method is exact—assuming continuous functions and a doubly infinite domain. The 'raw' BP method gave a quite smeared reconstruction, but it became very good after applying an empirically found filter to each of the scan data vectors. Although we arrived at the two methods—BP and FT—by quite different arguments, they ought to be somehow related.

A little bit of notation to start with: With the coordinate axes as shown in Figure 1.4.5, we can write the scan data function as

$$g(r,\theta) = \int_{-\infty}^{\infty} f(x,y)ds \tag{1.1}$$

where $x = x(s,r,\theta),\ y = y(s,r,\theta)$. Immediate back projection, as shown in Figure 1.6.2, gives a reconstruction

$$h(x,y) = \int_0^{\pi} g(r,\theta)\, d\theta \tag{1.2}$$

with $r = r(x,y,\theta)$. The result of the immediate back projection was shown in Figure 1.6.3. Although it is a reasonably good recovery, it is clear that $h(x,y) \neq f(x,y)$. In the next section, we will show that if we replace $g(r,\theta)$, as produced by (1.1), with

$$g(r,\theta) \;\rightarrow\; \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} g(\rho,\theta)\, e^{-i\omega_r\rho}d\rho\right]\, |\omega_r|\; e^{i\omega_r r}d\omega_r \tag{1.3}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{g}(\omega_r,\theta)|\omega_r|e^{i\omega_r r}dr \tag{1.4}$$

before substituting into (1.2), we get $h(x,y) = f(x,y)$—i.e. exact reconstruction.

Note that (1.4) is the Fourier transform of the product $\hat{g}(\omega_r,\theta)|\omega_r|e^{i\omega_r r}$. Writing $|\omega_r|$ as the Fourier transform of a function that is to be determined at the end of this section, we can interpret (1.4) as a particular filter applied to the function $g(r,\theta)$.

There is another way to express (1.4) that is mathematically equivalent:

$$g(r,\theta) \;\rightarrow\; \frac{1}{2\pi^2}\int_{-\infty}^{\infty}\frac{\partial g(\rho,\theta)/\partial\rho}{r-\rho}\,d\rho. \tag{1.5}$$

This expression was found by J. Radon in 1917. It is mathematically very elegant, shorter than (1.4), and superficially looks simpler (just a single integral), but turns out to be far less practical for computational use. Derivatives are usually more difficult to approximate well than integrals. Also, the integral has a 'principal value' singularity at $\rho = r$ which adds computational difficulty.

1.8.1. Derivation of replacement formula for $g(r,\theta)$. Both the FT method as well as (1.4) are exact. It is therefore natural to suppose that (1.4) can be derived from the FT method. Starting from

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\hat{f}(\omega_x,\omega_y)e^{i\omega_x x+i\omega_y y}d\omega_x d\omega_y$$

we change to the scan data coordinates (1.2) (see also Figure 1.4.5),

$$\omega_x = -\omega_r\sin\theta,\;\; \omega_y = \omega_r\cos\theta,\;\; \frac{d(\omega_x,\omega_y)}{d(\omega_r,\theta)} = \omega_r,$$

to obtain

$$f(x,y) = \int_0^{2\pi}\int_0^{\infty}\hat{f}(\omega_r,\theta)e^{-i\omega_r\sin\theta x+i\omega_r\cos\theta\, y}\omega_r\; d\omega_r d\theta.$$

If we now alter the description of the standard polar domain $0 \le \omega_r < \infty,\ 0 \le \theta \le 2\pi$ to $-\infty < \omega_r < \infty,\ 0 \le \theta \le \pi$ and recalling from Section 1.4 that $-x\sin\theta + y\cos\theta = r$, it follows that,

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} \hat{f}(\omega_r, \theta)\ e^{i\omega_r r}\ |\omega_r|\ d\omega_r d\theta.$$

Finally, the key result of the FT method (1.3) gives

$$f(x,y) = \frac{1}{(2\pi)^2} \int_0^{\pi} \left\{ \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} g(r,\theta) e^{-i\omega_r r} dr \right] |\omega_r|\, e^{i\omega_r r} d\omega_r \right\} d\theta.$$

Comparing this with (1.2) and (1.4), we see that (1.4) is established.

1.8.2. Interpretation of the replacement formula as a filter. We have just shown that the back projection method would be exact if each scan vector was modified according to (1.4) before being actually used for back projecting. The formula (1.4) amounts, independently for each θ, to a *convolution* of g with some yet undetermined function whose Fourier transform in the r-direction is $|\omega_r|$. Simplifying our notation a bit ($r \to x,\ \omega_r \to \omega$), we need to ask ourselves what function $e(x)$ has the Fourier transform $|\omega|$, i.e.

$$|\omega| = \frac{1}{2\pi} \int_{-\infty}^{\infty} e(x)\ e^{-i\omega x}\ dx$$

or

$$e(x) = \int_{-\infty}^{\infty} |\omega|\ \ e^{i\omega x}\ d\omega \tag{1.6}$$

This first attempt runs into a roadblock—the integral (1.6) is divergent.

The integral is divergent because it is defined on an infinite interval. In practical work, we only consider finite intervals. So one way to gain insight into what (1.6) is 'trying to tell us' is to discretize it and look at the DFT of the function $|\omega|$. We can then see if there is any clear pattern emerging as $N \to \infty$. We choose N even and

for frequency:	$-\frac{N}{2}+1$	$-\frac{N}{2}+2$	$\cdots$	-2	-1	0	1	2	$\cdots$	$\frac{N}{2}-2$	$\frac{N}{2}-1$	$\pm\frac{N}{2}$
enter value:	$\frac{N}{2}-1$	$\frac{N}{2}-2$	$\cdots$	2	1	0	1	2	$\cdots$	$\frac{N}{2}-2$	$\frac{N}{2}-1$	$\frac{N}{2}$

After having normalized the FFT output by multiplying it with $\frac{4}{N^2}$ we get

$N = 32$	...	$-.0176$	0.0	$-.0464$	0.0	$-.4066$	1.0000	$-.4066$	0.0	$-.0464$	0.0	$-.0176$	...
$N = 64$	...	$-.0165$	0.0	$-.0454$	0.0	$-.4056$	1.0000	$-.4056$	0.0	$-.0454$	0.0	$-.0165$	...
$N = 128$	...	$-.0163$	0.0	$-.0451$	0.0	$-.4054$	1.0000	$-.4054$	0.0	$-.0451$	0.0	$-.0163$	...
.....													
$\lim_{N\to\infty}$	...	$-.0162$	0.0	$-.0450$	0.0	$-.4053$	1.0000	$-.4053$	0.0	$-.0450$	0.0	$-.0162$	...
$=$	...	$-\left(\frac{2}{5\pi}\right)^2$	0	$-\left(\frac{2}{3\pi}\right)^2$	0	$-\left(\frac{2}{1\pi}\right)^2$	1	$-\left(\frac{2}{1\pi}\right)^2$	0.0	$-\left(\frac{2}{3\pi}\right)^2$	0.0	$-\left(\frac{2}{5\pi}\right)^2$	...

The three entries

$$-0.4053 \quad 1.0000 \quad -0.4053$$

very much dominate the other entries. We have arrived theoretically at just the same type of filter as was empirically proposed in Section 1.6. The value $\beta \approx 0.4$ we used there is indeed nearly optimal.

1.9. Exercises.

We recall from Section 1.4 that given the density function $f(x, y)$ of a 2-D object, the scan data can be written

$$g(r, \theta) = \int_{-\infty}^{\infty} f(x, y)\, ds \tag{1.1}$$

where the rotated coordinate axes are as defined through

$$\begin{cases} s = x\cos\theta + y\sin\theta \\ r = -x\sin\theta + y\cos\theta \end{cases} \qquad \begin{cases} x = s\cos\theta - r\sin\theta \\ y = s\sin\theta + r\cos\theta \end{cases} \quad \cdots$$

- Determine the scan data produced by the function $f(x, y) = \begin{cases} 1 & x^2 + y^2 \le 1 \\ 0 & \text{otherwise} \end{cases}$

  Answer: $g(r, \theta) = \begin{cases} 2\sqrt{1 - r^2} & |r| \le 1 \\ 0 & \text{otherwise} \end{cases}$.

- Determine the scan data produced by the function $f(x, y) = \begin{cases} 1 & \max(x, y) \le 1 \\ 0 & \text{otherwise} \end{cases}$

  Answer: Define $\phi = \text{mod}(\theta + \frac{\pi}{4}, \frac{\pi}{2}) - \frac{\pi}{4}$ (suffices to solve for $0 \le \theta \le \frac{\pi}{4}$; remaining θ-intervals reflections of this one). Then

$$g(r,\theta) = \begin{cases} \frac{2}{\cos\phi} & |r| \le \cos\phi - \sin\phi \\ \frac{\cos\phi + \sin\phi - r}{\cos\phi\ \sin\phi} & \cos\phi - \sin\phi < |r| \le \cos\phi + \sin\phi \\ 0 & \text{otherwise} \end{cases} .$$

- Verify from the definition (1.1) that $g(r,\theta)$ is a *linear* function of $f(x,y)$.

  Hint: Write down the two requirements for linearity, and test these on $g(r,\theta)$.

- We let R denote the Radon transform operator, i.e. $Rf(x,y) = g(r,\theta)$.

  a. Show that $R\,f(\alpha x, \alpha y) = |\alpha|\ g(\alpha r, \theta)$.

  b. Show that if we change independent variable $\begin{pmatrix} \xi \\ \eta \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix}$ where A is a 2×2 matrix; $B = A^{-1}$, then $R\,f(\xi,\eta) = |\det B|\ g(...,...)$.

- Show that for $g(r,\theta) = \begin{cases} 1 & |r| \le 1 \\ 1 - \frac{|r|}{\sqrt{r^2-1}} & \text{otherwise} \end{cases}$ immediate back projection leads to a reconstruction $h(x,y) = \begin{cases} 1 & x^2 + y^2 \le 1 \\ 0 & \text{otherwise} \end{cases}$.

CHAPTER 2

Facial Recognition

2.1. Introduction

How can one tell whether the criminal who has just been convicted of a crime, is a first– or a habitual offender? This is a most serious question since the sentence depends on the answer. The habitual offender can of course expect a harsher sentence and will do everything possible to hide his/her real identity. This was exactly the situation in Europe during the second half of the nineteenth century. There was no reliable system in place to identify individuals and the police had to rely almost entirely on personal recognition. People were often misidentified—with sometimes disastrous consequences. A case in point, as late as 1896, Adolf Beck was misidentified as John Smith, a conman and repeat offender, and sentenced to seven years in prison. The fact that according to descriptions, John Smith had brown eyes while Beck had blue eyes, that Smith was circumcised and Beck not, made no difference—it was ascribed to administrative error. Just too many witnesses were willing to swear that Beck was indeed the perpetrator. It was only after Beck's release that John Smith was arrested on a charge of hoaxing two actresses out of their rings that the full sorry tale was revealed, see [**?**].

A reliable personal identification system was clearly long overdue.

At that time two rival personal identification systems were being developed. William James Herschel, not to be confused with his grandfather, the eminent astronomer, also William Herschel, was experimenting with hand prints, and a little later with fingerprints in India. But the most influential individual in the development of fingerprints as a personal identification system was Henry Faulds. His discovery was accidental. Because of his anthropological interests, he was in the habit of collecting fingerprints of his students and friends, his collection soon numbered in the thousands. At about this time he noticed that the supply of medical

alcohol at his hospital started to run inexplicably low. When he then discovered a cocktail glass in the form of a laboratory beaker with an almost complete set of fingerprints, the culprit was promptly identified. This was exactly the spark that was needed, although it took at least another 20 years before fingerprints were widely accepted as forensic evidence.

The second system was developed in France during the 1870's by one Alphonse Bertillon. His system was the result of frustration and bitterness over a mindlessly boring and futile job. He was required to write police descriptions into the five million police files gathering dust in their massive archive. A typical description would say, 'Stature: average', 'Face: ordinary'. These descriptions were obviously totally useless. He then hit on the idea of measuring an individual. Maybe if one takes enough measurements that total set would be unique for an individual. His system employed eleven separate measurements: height, length and breadth of head, length and breadth of ear, length form elbow to end of middle finger, lengths of middle and ring fingers, length of left foot, length of the trunk, and length of outstretched arms from middle fingertip to middle fingertip. Apart from being to able distinguish between different individuals, it also allowed a classification system that enabled Bertillion to quickly locate the file of a criminal, given only the measurements. The system was so successful that France was one of the last countries to adopt fingerprints for personal identification, see [**?**].

Our modern technological society relies heavily on personal identity verification, be it to gain access to bank accounts or secure areas, or just to login on a computer. Automated identity verification is a complex problem and many different options have been pursued. Some old favorites, including

- Personal Identification Numbers (PIN)
- Passwords
- Identity documents, etc

are easy to copy and encourage fraud. The problem is that these systems do not contain any personal information about the individual. Passwords and PIN's are of course totally divorced from the individual using it—it is impossible to verify that the person offering the password is in fact authorized to use it. It is not surprising that

fraudulent transactions based on the misuse of identification systems have become a most serious problem.

In recent years one therefore finds an increasing move away from systems relying on something the individual *own or know* as a means of identification, to systems recognizing something the individual *is*. Thus much effort has gone into the development of automated personal identification/verification systems based on the characteristics unique to each individual, the so-called Biometric Personal Identification Systems. Ideally these system are based on personal characteristics that even the individual is not able to alter (disguise). A number of biometric identification systems are already commercially available. It is possible to login onto your computer using your fingerprint or go through immigration after a retina scan has establish you identity. Dynamic signature verification is totally dependent on the participation of the individual and is ideal in situations where one has to endorse a transaction; more of this in Chapter 22.

Facial recognition on the other hand, is a passive system requiring no participation from the individual. No wonder that it is becoming increasingly popular for surveillance systems. For humans it is also the most natural identification system available. It is indeed difficult to imagine a world without the human face as we know it: flat (i.e. no muzzle), hairless and with its characteristic features, eyes, protruding nose, mouth and chin. Yet the true human face appeared only about 180 000 years ago with Homo sapiens in Africa. The human face is a most remarkable object. It houses four of the five senses (sight, smell, taste and touch) and we learn from the earliest infancy to rely on it for identifying each other. Equally important is its use for communication. It might be argued that the human face has evolved to its present form for no other reason than to improve communication. Indeed, facial hair hides expression and the fact that the facial muscles are directly attached to skin, allows for an infinity of facial expressions, reflecting an infinitude of emotional subtleties. No wonder that the human face has held such a fascination for artists through all the centuries. Some of the greatest works of art convey such a complex of emotions that it defies description. For some the smile of the Mona Lisa by Leonardo da Vinci is ‘divinely pleasing’, someone else believes she is flirting and yet another senses a ‘hateful haughtiness’. See [**?**] for a fascinating discussion of the human face.

Serious scientific studies of the visual qualities of the human face date back at least to the same Leonardo da Vinci who made detailed studies of the interaction of light and face and recently of course, it has become the object of intensive scientific study. Although the scientist may have very different goals, its fascination with the geometric structure, the interaction of this structure with light, its moods and expressions, is no less keen than that of the artist. For it is exactly these very human characteristics that provide a person his/her visual individuality—one of the major concerns of the biometric scientist.

A number of different ideas have been developed for automated facial recognition , see for example [**?**]. In this chapter we concentrate on systems based on the so-called eigenface technique. The basic idea, first introduced by Sirovich and Kirby [**?**], has subsequently been developed into some of the most reliable facial recognition systems available. In particular, the eigenface-based system developed at the Media Laboratory at MIT ([**?**],[**?**] and [**?**]) has consistently performed among the best in the comprehensive FERET tests [**?**].

Eigenfaces are derived from a carefully constructed set of facial images, the training set. The training set is a substitute for all the facial images the system is expected to encounter and should therefore represent the characteristics of all relevant facial images. The aim of the eigenface approach is to distill these characteristics in the form of the eigenfaces. The idea is simple: find an orthonormal basis for the subspace spanned by the images in the training set, easily achieved through the Singular Value Decomposition (SVD). The power of this approach lies in the fact that the facial images in the training set lie inside a low dimensional subspace of the general image space. This subspace is identified through the nonzero, or very small, singular values. The eigenfaces are the orthonormal basis elements associated with the remaining nonzero singular values. Assuming that the training set is really representative of all faces, the eigenfaces therefore form a low dimensional, orthonormal basis for the linear subspace containing all facial images (note that we try to formulate this very carefully—faces themselves do not form a linear subspace). Any facial image can be orthogonally projected onto the eigenfaces with the result that any particular face is represented by its projection coefficients. In a facial recognition system the

similarity of the projection coefficients of different faces is used to decide whether two images are from the same individual or not.

In this very basic description of a facial recognition system based on eigenfaces important practical issues have been ignored. For example, experiments by Pentland and co-workers, see [**?, ?**], show that the efficiency of the system is improved significantly if a comparison of global, eigenface expansions is augmented by a local eigenfeature expansion, consisting for example of the eyes, noses and mouths. Since these 'local' features are also compared by expanding in their 'eigenfeatures', the ideas described in this chapter also apply to these situations.

2.2. An Overview of Eigenfaces.

The idea behind the eigenface technique is to extract the relevant information contained in a facial image and represent it as efficiently as possible. Rather than manipulating and comparing faces directly, one manipulates and compares their representations.

Assume that the facial images are represented as 2D arrays of size $m = p \times q$. Obviously, m can be quite large, even for a coarse resolution such as 100×100, $m = 10,000$. By 'stacking' the columns, we can rewrite any $m = p \times q$ image as a vector of length m. Thus, we need to specify m values in order to describe the image completely. Therefore, all $p \times q$ sized images can be viewed as occupying an $m = pq$-dimensional vector space. Do facial images occupy some lower dimensional subspace? If so, how is this subspace calculated?

Consider n vectors with m components, each constructed from a facial image by stacking the columns. This is the training set and the individual vectors are denoted by $\mathbf{f}_j$ where $j = 1, \ldots, n$. Obviously, it is impossible to study every single face on earth, so the training set is chosen to be representative of all the faces our system might encounter. This does not mean that all *faces* one might encounter are included in the training set, we merely require that all faces are adequately *represented* by the faces in the training set. For example, it is necessary to restrict the deviation of any individual face from the average face. Some individuals may be so unique that our system simply cannot cope. See [**?, ?**] for a detailed analysis of issues relating

to training sets. Obviously, the training set must be developed with care. Also, typically $n \ll m$.

As mentioned above, one should ensure that the faces are normalized with respect to position, size, orientation and intensity. All faces must have the same size, be at the same angle (upright is most appropriate) and have the same lighting intensity, etc, requiring some nontrivial image processing (see [**?**]). Assume it is all done.

Since all the values for the faces are nonnegative and the facial images are well removed from the origin, the average face can be treated as a uniform bias of all the faces. Thus we subtract it from the images as will be explained in more detail in Section 11.4. The average and the deviations, also referred as the caricatures, are

$$\mathbf{a} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{f}_j, \tag{2.1}$$

$$\mathbf{x}_j = \mathbf{f}_j - \mathbf{a}. \tag{2.2}$$

We illustrate the procedure using the Surrey database [**?**]. This consists of a large number of RGB color images taken against a blue background. Using color separation it is easy to remove the background and the images were then converted to gray-scale by averaging the RGB values. A training set consisting of 600 images (3 images of each of 200 individuals) was constructed according to the Lausuanne Protocol Configuration 1 [**?**]. We should point out that the normalization was done by hand and is not particularly accurate as will become evident in the experiments. Figure 2.2.1 shows three different images of two different persons—the first two images are of the same person where the first image is part of the training set. The second image is not inside the training set, taken on a different day and one should note the different in pose as well as facial expression. The third image is of a person not in the training set.

We need a basis for the space spanned by the $\mathbf{x}_j$. These basis vectors will become the building blocks for reconstructing any face in the future, whether or not the face is in training set. In order to construct an orthonormal basis for the subspace spanned by the faces in the training set, we define the $m \times n$ matrix X with columns

FIGURE 2.2.1. Sample faces in the database.

corresponding to the faces in the training set,

$$X = \frac{1}{\sqrt{n}} \left[\begin{array}{cccc} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \end{array} \right], \tag{2.3}$$

where the constant $\frac{1}{\sqrt{n}}$ is introduced for convenience. The easiest way of finding an orthonormal basis for X is to calculate its Singular Value Decomposition (SVD), as explained in detail in Section **??**. Thus we write

$$X = U\Sigma V^T.$$

For an $m \times n$ dimensional matrix X, U and V orthonormal matrices with dimensions $m \times m$ and $n \times n$, respectively. Σ is an $m \times n$ diagonal matrix with the non-negative singular values, $\sigma_j,\ j = 1, \dots, \min(m, n)$ arranged in non-decreasing order on its diagonal. If there are r nonzero singular values then the first r columns of U form an orthonormal basis for the column space of X. In practice one can also discard those columns associated with very small singular values. Let us say we keep the first ν columns of U where $\nu \leq r$ and $\sigma_{\nu+1}$ is regarded to be sufficiency small so that it can be discarded.

In Figure 2.2.2, we plot the normalized singular values—normalized so that the largest singular value is one—for our training set of facial images. Typically these eigenvalues decrease rapidly. In this case the 150th singular values is already quite small. One would therefore expect a rather good representation using the first 150 columns of U, i.e. $\nu \approx 150$. These basis vectors, $\mathbf{u}_j,\ j = 1, \dots, \nu$, are the eigenfaces. Some of them are shown in Figure 2.2.3. Note how the higher eigenfaces have more detail added to them. This is in general the case—the lowest eigenfaces contain

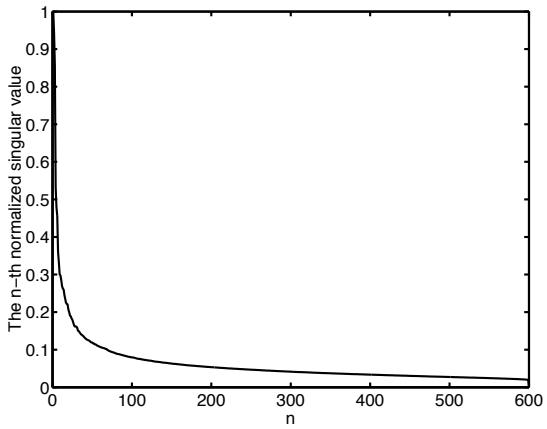


FIGURE 2.2.2. The normalized singular values.

general face-like information with the detail required to distinguish an individual provided by the higher order eigenfaces. For a more detailed explanation see [**?**, **?**]. In this case one should point out that the almost complete lack of structure of the first eigenface is an indication that our training set is not well normalized.

If our training set were representative, all facial images are in a 150 dimensional linear subspace. However, the faces in our training set were selected arbitrarily and not optimized to be as widely representative as possible. More importantly, our normalization is crude and we'll describe a simple experiment indicating just how sensitive the system is to the normalization. One should also realize that for a automated facial recognition system, visually perfect reconstruction is not required. Reconstructions that include sufficient information to distinguish between different individuals suffice. Estimates in the literature range from 40 to about 100 eigenfaces based on experiments with a heterogeneous population (see [**?**]), indicating that faces are described very efficiently in terms of eigenfaces—the effective dimension of the linear vector space containing the facial images is of order 100 (rather than $m = pq$) for a general $p \times q$ image.

One might ask how important it is to subtract the average $\mathbf{a}$. Imagine that the facial images are clustered around the average face, far from the origin. The first eigenface indeed points in the direction of the average face. The problem is that the rest of the eigenfaces are constrained to be orthogonal to this rather arbitrary

(a) Eigenface 1

(b) Eigenface 10

(c) Eigenface 50

(d) Eigenface 100

FIGURE 2.2.3. Eigenfaces number 1, 10, 50, and 100.

direction, reducing their ability to describe the distribution of the faces clustered around the average face. Experiments show that if the average is not removed, the first eigenface is indeed closely related to the average, but the singular values decrease slightly slower.

2.3. Calculating the eigenfaces.

One striking feature of the calculation of the SVD is the relative sizes of m and n. Because $m = pq$ it can easily become very large, the number n of images in the training set is typically one or two orders of magnitude smaller. The question arises whether one can exploit this fact for numerical purposes. This turns out to be a little tricky. The main problem is that one has to be very careful to ensure that numerical errors do not destroy the orthogonality of the orthogonal matrices. For example one might be tempted to follow the procedure prescribed by Fukunaga [**?**, p39] and first calculate the reduced V_+ from the $n \times n$ symmetric eigenvalue problem (11.2), using any of the efficient methods available for calculating the eigensystem of symmetric matrices, including the QR algorithm, Rayleigh quotient iteration, divide and conquer methods, etc. See [**?, ?**] for more details. Although there is already a loss in accuracy by computing $X^T X$ (see Section 11.4 for more detail), it is the next step where one can go seriously wrong. For example, it is tempting to calculate U_+ from

$$XV_+ = U_+\Sigma_+, \tag{2.1}$$

i.e. $U_+ = XV_+\Sigma_+^{-1}$. This is not a good idea. Note, for example that round-off error destroys the orthogonality of the columns of U_+.

In order to exploit the fact the $n << m$ in a numerically stable way, we suggest the following procedure. Given an $m \times n$ matrix X with $n << m$, we use of another of the great matrix factorizations, namely the QR factorization with column permutations (the basis of the QR algorithm mentioned above), to calculate $XP = QR$ where Q is an $m \times m$ matrix with orthonormal columns and R is an $m \times n$ upper triangular matrix. P is a permutation matrix rearranging the columns of X such that the diagonal entries of R are in non-increasing order of magnitude. Since the last $m - n$

rows of R consist of zeroes, we can form the reduced QR factorization by keeping the first n columns of Q and the first n rows of R. Thus we obtain $X = Q_+R_+$ where R_+ is an $n \times n$ upper triangular matrix. The next step is to calculate the SVD of $R_+ = U_R\Sigma V^T$. Thus we get, $XP = (Q_+U_R)\Sigma V^T$, or $XP = U\Sigma V^T$ with U the product of two orthonormal matrices, $U = Q_+U_R$.

2.4. Using Eigenfaces

In the previous section we discussed the reasons why faces can be efficiently represented in terms of eigenfaces. To obtain the actual representation is quite straightforward. The idea is to project any given face orthogonally onto the space spanned by the eigenfaces. More precisely, given an face $\mathbf{f}$ we need to project $\mathbf{f} - \mathbf{a}$ onto the column space of $U_\nu := [\mathbf{u}_1 \cdots \mathbf{u}_\nu]$. This means that we wish to solve

$$U_\nu \mathbf{y} = \mathbf{f} - \mathbf{a} \tag{2.1}$$

in a least squares sense. Since the columns of U_ν are orthonormal, it follows that the eigenface *representation* is given by

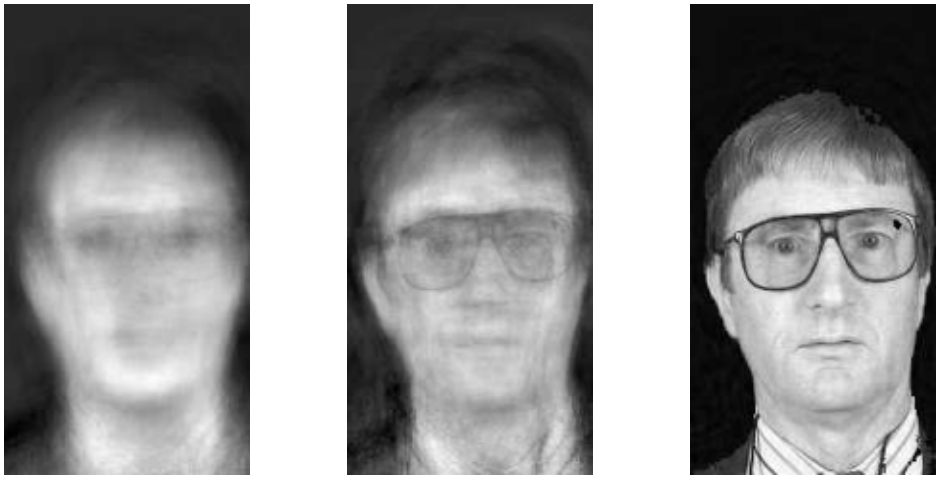
$$\mathbf{y} = U_\nu^T(\mathbf{f} - \mathbf{a}). \tag{2.2}$$

This representation captures all the features associated with the face $\mathbf{f}$ and instead of comparing faces directly, we rather compare features.

The eigenface *reconstruction* of $\mathbf{f}$ is given by

$$\widetilde{\mathbf{f}} = U_\nu \mathbf{y} + \mathbf{a}. \tag{2.3}$$

The eigenface reconstruction of the first face in Figure 2.2.1 (the face in the training set) is shown in Figure 2.4.1, using 40, 100 and 450 eigenfaces. The root mean square (rms) errors of the three reconstructions—the 2-norm of the difference between the original and the reconstruction, or equivalently, the magnitude of the first neglected singular value—are given by 6742, 4979 and 753. Certainly up to 100 eigenfaces, the reconstruction is visually not particularly good, despite the fact the



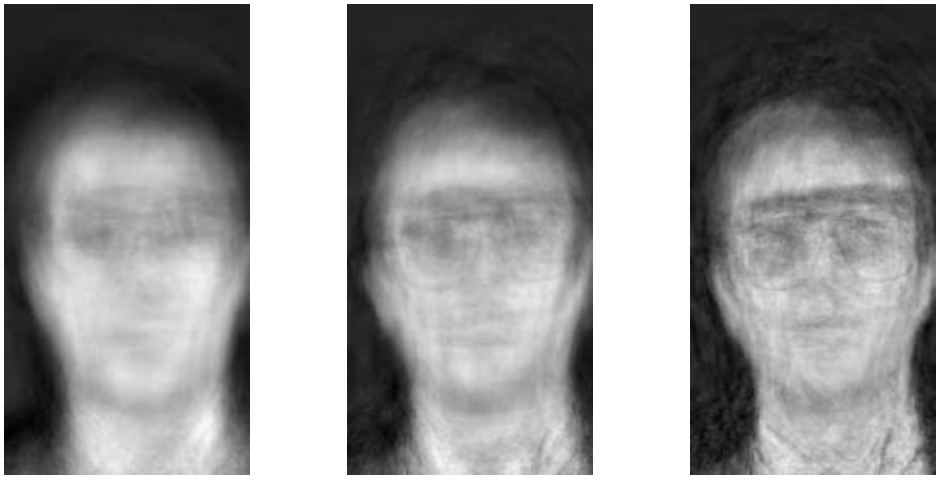
(a) 40 Eigenfaces (b) 100 Eigenfaces (c) 450 Eigenfaces

FIGURE 2.4.1. The reconstruction of a face in the training set.

the singular values (rms values) are already relatively small. The 2-norm is clearly not a good norm in which to measure visual correspondence. It is only for rms values less than about 1000 that one finds good visual reconstruction.

We do the same for the second image of the first person, not in the training set (the second image of Figure 2.2.1), and the results are shown in Figure 2.4.2. Although the result is not visually pleasing, the important question is whether enough information is available to recognize the individual.

In Figure 2.4.3 we show the reconstruction of the third face of Figure 2.2.1, the face not in the training set, again using 40, 100 and 450 eigenfaces. Although the reconstruction is visually not particularly good, the individual is already recognizable using 100 eigenfaces. In fact this reconstruction is better than that of Figure 2.4.2. The reason is that most of the images in the training set are facing the camera directly. Thus, although not in the training set, the individual with a similar pose are better reconstructed in Figure 2.4.3 than the individual in the training set but with a pose not represented in the training set.



(a) 40 Eigenfaces (b) 100 Eigenfaces (c) 450 Eigenfaces

FIGURE 2.4.2. The reconstruction of a face not in the training set.

(a) 40 Eigenfaces (b) 100 Eigenfaces (c) 450 Eigenfaces

FIGURE 2.4.3. The reconstruction of a person not in the training set.

The final experiment demonstrates the sensitivity of the system to the normalization. In this experiment we shifted the first image of Figure 2.2.1 two pixels down from its original position. Figure 2.4.4 shows its reconstruction using 450 eigenfaces. The result is clearly a significant deterioration of the visually good reconstruction we obtained in Figure 2.4.1.

FIGURE 2.4.4. Reconstruction of a non-normalized face.

Up to this point the gray-scale images have been 2D representations of the interaction of light with a 3D structure. This poses several difficulties. For example, since no 3D information is available it is hard to correct for essentially 3D distortions such as out-of-plane rotations. We had more problems reconstructing the image with out-of-plane rotation of a person inside the training set, than of a person not in the training set, but directly facing the camera. Of course facial expression also plays a important role and is not easily corrected for. Gray-scale images are also heavily dependent on the strength and direction of the light source and it is not easy to compensate for different lighting conditions [**?**]. Working with gray-scale images where the different shades of gray represent 3D structure directly remove these problems. This idea is pursued in Chapter 32. Provided a simple camera model is used, the reconstruction again relies on the SVD.

CHAPTER 3

Global Positioning Systems

3.1. Introduction.

The history of navigation is one of the longest and most important quests in the evolution of civilization. With GPS, technology has at last provided a near-perfect solution, now freely available for everybody to use. GPS receivers have quickly become indispensable, not only by professionals such as seamen or pilots, but also to hikers and basically to anybody enjoying the outdoors. The accuracy of present small hand-held units is of the order of 10—15 meters after *selective availability* — degrading the precision for non-military users—was eliminated in 2000. One can the for example use these to record the position where one leaves a car in a big outdoor car park. Used simply as clocks, they providing local time to better than 10^{-6} seconds—far higher accuracy of course than is ever needed in everyday life. Together with *differential correction* and a *phase locking* technique, accuracies can be as high as one millimeter, potentially making all other surveying techniques obsolete, both for global (e.g. piloting) and local (e.g. construction site) usage.

Section 3.2 recalls very briefly the history of navigational devices, from ancient days until the breakthrough of GPS. Section 3.3 summarizes the principles of GPS. We formulate in Section 3.4 a test problem and then solve it in two different ways. This is followed by some error analysis in Section 3.5 and a discussion of *pseudo-random sequences* in Section 3.6.

3.2. A Brief History of Navigation.

Early man observed the sun and the stars, and presumably used these for navigation long before leaving any written records about it. As late as the Viking age (800-1100 AD), little further help was available for navigation on open seas. Rough estimates of the Polar star's height over the horizon, together with 'dead reckoning'

(a phrase originating from 'deduced reckoning'; estimating distances based on course, currents, winds and speeds), did not always suffice to find the intended destinations. Innumerable marine disasters have been caused by navigational errors.

> One particularly gruesome incident occurred on a foggy night in October, 1707 when a group of four British warships with about 2,000 men on board ran aground just off the English SW coast. Only two men reached shore. One happened to be the fleet commander, who was promptly murdered upon reaching shore by a local woman for a ring he was wearing. Maybe there was some justice in this. The fleet officers knew full well before the accident that their navigation had been faulty; nevertheless, a seaman who had kept a perfect log, and dared to very carefully and respectfully offer this to an officer the day before—knowing the risk but hoping to help avoid a disaster—was immediately hanged for insubordination. This particular incident was a contributing factor to a British competition that about half a century later finally led to a successful means to determine longitude at sea—latitude is much easier—and hence to much safer sea travels.

The concept of longitudes and latitudes goes back at least to Ptolemy. All 27 sheets of his world atlas from 150 AD have such lines drawn, together with a separate list of coordinates for all its named locations. The equator was on his atlas marked as the zeroth parallel (latitude) and the Canary Islands defined the zero meridian (longitude). This latter choice was quite arbitrary, and indicative of the coming difficulties in determining the longitude at sea. Before settling at Greenwich, 'prime meridians' were at times placed at the Azores, Cape Verde Islands, Rome, Copenhagen, Jerusalem, St. Petersburg, Pisa, Paris, Philadelphia and many other places as well.

The big advances in navigation from the days of the Vikings have been

- discovery of the compass,
- finding the longitude,
  (latitude can be read off easily from the height of stars—e.g. the pole star—over the horizon),
- navigation by radio beacons (LORAN), and

- GPS.

The first compasses were simple chunks of loadstone (magnetite, a common iron ore) which tend to orient themselves in a fixed direction, when suspended freely (by a string, or floated in a container of water). Their first documented use for navigation occurred in the Mediterranean during the 12th century. Magnetic compasses remain to this date indispensable on all ships, at the very least as a navigational back-up device. Gyro-compasses work by a completely different principle—a suitably suspended rapidly rotating disc will keep its axis aligned with that of the earth. These compasses will always point to true north, and are insensitive to variations in the magnetic field (which can be due to geological anomalies or electrical storms on the sun). Although far more complicated than magnetic compasses, they are nowadays used in most larger ships and aircraft, often in connection with 'inertial guidance' devices that compute changes in positions from sensed accelerations.

The lack of any reliable means for determining the longitude at sea caused great hazards for sea travels until the chronometer was developed in the second half of the 18th century. If it was not possible to simply follow coastlines which can be dangerous, especially at night and in bad weather, it was common practice to try to reach ports by first finding the appropriate latitude, and then follow it until the destination appeared in sight. This procedure was not very satisfactory for several reasons

- it works less well for coastlines facing north or south, as opposed to east or west,
- when aiming for a small island, the approach to the desired latitude had to be quite far off to the east or to the west in order to leave no ambiguity about the direction to finally proceed in,
- it forced sailing ships to follow paths that might not be suitable with regard to shoals, winds and currents, and
- it offered opportunities for pirates to lie in wait out at sea at the latitudes of main harbors.

The main competing approaches for finding the longitude all required that the local time which is easily available by the position of the sun, be compared to the simultaneous time at some fixed reference location such as Greenwich. Ideas to determine this reference time included

- Observing the position of our moon relative to the sun and the stars. Newton's law of gravity was discovered first in 1684, and the complicated orbit of the moon—a quite non-circular path influenced by both the earth and the sun—could not be predicted with enough precision until well after the whole approach had been made obsolete by the chronometer.
- Observations of Jupiter's moons. Since their orbits could be tabulated accurately, the moons can serve as an accurate clock in the sky. Eclipses when a moon disappears in the shadow of the planet, happens roughly every two days for each of the inner moons and are near-instantaneous events, allowing very accurate time readings. Although this worked very well on land (for example to determine the location of islands), even in good weather it proved to be utterly impractical at sea.
- The chronometer—basically an accurate clock, designed to be insensitive to motions and changes in temperature, humidity and gravity. This became the winner in the longitude competition. John Harrison's produced a series of increasingly accurate chronometers, culminating in 1760 with the pocket-sized 'H-4'. On its first sea trial—UK to Jamaica, arriving in January 1762—it lost only 5 seconds. This corresponds to an error of only 2 km after 81 days at sea. However, this was somewhat lucky—an error of about one minute or 24 km could have been expected; even that a vast improvement over other methods. By 1780, chronometers were starting to come in wide use throughout the British and other navies. These were often privately purchased by the Captains, as official navy channels still were slow in providing them.

More exotic ideas at the time included

- Placing light-ships at known strategic locations. These would then every-so-often send up a rocket that exploded brightly—deemed to be visible at

night for up to 60 to 100 miles, providing travelers within that range with a time signal, and

- Mapping the vertical inclination of the earth's magnetic field. Lines of equal inclination would generally intersect the lines of constant latitude (or the angle could be mapped), thus together with the latitude providing complete positional information. However, not only does the earth's magnetic field change slowly with time, it can also fluctuate dramatically with solar activities, up to about 10 degrees—enough to cause positional uncertainties as wide as an ocean.

The history of how the longitude problem got resolved though the chronometer recounted in many books; a recent one being 'Longitude' by Sobel [**?**]. Even then, navigation was not always easy. After the loss of his ship, the Endurance, in the pack-ice of the Antarctic, the famous British explorer, Ernst Shackleton, and his whole crew ended up on Elephant island, one of the most remote spots on earth. This was April 1916, Europe was at war and no one in any case had the faintest idea of the critical situation of the Shackleton expedition. The nearest help was at the South Georgia islands, about 800 miles across some of the stormiest oceans imaginable. Their most seaworthy vessel was the James Caird, an open lifeboat, totally unsuitable for the task ahead. With no other options left, Shackleton and a small crew, including the navigator and captain of the Endurance, Frank Worseley, sailed from Elephant island on April 24, 1916 on their 800 mile journey. Due to bad weather Worseley was forced to navigate mainly through dead-reconing. As they approached the South Georgia islands, the situation became critical. In the words of Worseley [**?**]:

On the thirteenth day we were getting nearer to our destination. If we made the tragic mistake of passing it we could never retrace our way on account of the winds and the currents, it therefore became essential that I should get observations. But the morning was foggy, and if you cannot see the horizon it is impossible to measure the altitude of the sun to establish your position. Now, the nearer your eye is to the surface of the sea, the nearer is the horizon. So I adopted the expedient of kneeling on the stones in the bottom of the boat, and by this means succeeded in taking a rough

> observation. It would have been a bold assumption to say that it was a correct one; but is was the best I could do, and we had to trust it. Two observations are necessary, however, to fix your position, and my troubles were far from over; for at noon, when I wanted to observe our latitude, I found conditions equally difficult. The fog, which before had been on a level with us and therefore did not altogether obscure the sun, had now risen above us and was hovering between the sun and ourselves, so that all I could see was a dim blur. I measured to the centre of this ten times, using the mean of these observations as the sun's altitude.
>
> With serious misgiving I worked out our position and set my course by it to sight South Georgia, near King Haakon Sound, the next day.
>
> Thus, against all odds, the whole expedition was saves without the loss of a single man.

The first radio-based navigation technique amounted to determining the direction to a known transmitter by rotating a direction-sensitive antenna. Much higher precision was offered by a series of systems known as OMEGA, DECCA, GEE and LORAN (long range navigation). These were developed around the time of World War II. By the timing difference in arrivals of radio signals from a 'master' and a 'slave' transmitter which re-transmitted the master signal the moment it received it, a ship could locate itself along a specific curve, In the 2-D plane case, it is a hyperbola which is the curve with constant difference from two points. By also receiving signals from another transmitter pair, the ship could determine its location from the intersections of the two curves. This system gave a typical accuracy of around 1 km and a useful range of about 1000 km at daytime, and about double that at night time. Radio navigation systems were the first ones that could give positional fixes in any weather conditions.

The GPS idea is to have a number of satellites in orbit, each transmitting both its orbital data and very accurate time pulses. A receiver can then time the arrivals of the incoming time pulses. Knowing the speed of light, the distances to the satellites can be found (c = 299,792,458 m/s in vacuum). This is exact and serves since 1983 as the definition of the meter based on an existing definition of the second. From knowing their orbits, the receiver's position can be found. With the high velocity

of the satellites (and the high speed of light!), the demands on the precision of the equipment are extreme.

> The cesium or rubidium clocks in the GPS satellites operate at 10.22999999545 MHz rather than the nominal 10.23 MHz to compensate for both the special relativity effect of a moving source and the general relativity effect of operating from a point of higher gravitational potential. The master clock at the GPS control center near Colorado Springs is set to run 16 ns a day fast to compensate for its location 1830 m above sea level.

The military's need for the system was also extreme—it was developed towards the end of the cold war as a means of accurately guiding ICBMs. Hence, it is hardly surprising that there are today two parallel fully operational systems in place, one created by the US Department of Defense and one by its Soviet counterpart. The cost for getting the GPS systems operational was staggering—at least 12 B$ (i.e. $12{\cdot}10^9$$) for the US system. The fact that both systems now are available to the general public, without any charge, is almost as impressive as their technical capabilities. With low cost handheld receivers (around 100 dollars), anyone can now determine his/her position to better than 100 meters at any time, in any weather, at any point on earth. With the best, and much more expensive, receiving equipment available, that can be improved to an amazing 1 mm in both horizontal and vertical coordinates. Surprisingly, GPS is still not used routinely in aviation (in 2000)—possibly because neither of the two signal providers is officially committed to providing uninterrupted public service. For most civil and private usage, this concern is far outweighed by its practical advantages.

3.3. Principles of GPS.

Table 1 summarizes some technical specifications for GPS and GLONASS. These American and Russian systems are very similar in most respects. Before concentrating on GPS, let us note one difference: All the GPS satellites broadcast on exactly the same frequency (in order to save bandwidth); their transmission of separate pseudo-random (PR) sequences—described in Section 3.6—allows this without causing any signal confusion. Two GLONASS satellites exactly opposite each other use

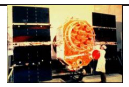
	GPS	GLONASS
		
Operated by	US DOD (Department of Defense)	Russia
Control center	Falcon Air Force Base (near Colorado Springs, CO)	?
First satellite launched	1978	1982
System operational	1993	1993
Satellite constellation		
Number of satellites	$24 + 3$ spares	$24 + 2$or 3 spares
Satellite distribution	4 spaced 90^o apart in 6 planes	8 spaced 45^oapart in 3 planes
Orbital inclination to equatorial plane	55^o(limited by possible orbits of the space shuttle for launching and servicing)	64.8^o
Average elevation (from center of earth)	27,560 km (about 3.0 earth radii above its surface—at the	25,510 km outer edge of upper van Allen belt)
Orbital period	11 h 58 min (one half sidereal day)	11 h 15 m 45 s
Frequency of orbital information update from ground	every hour	every half hour
Radio frequencies (civilian)	1575.42 MHz (Navigational information)	$1602 + n \cdot 0.5625$MHz, $n = 0, 1, \ldots, 12$
Length of pseudo-random code	$1023 = 2^{10} - 1$ bits	$511 = 2^9 - 1$ bits
Chip rate; repeat time of pseudor. code	1.023 MHz; 1.0 ms	0.511 MHz; 1.0 Ms
Data package; rate, length	50 bits/s; 30 s	50 bits/s ; 30 s

TABLE 1. Some technical specifications of GPS and GLONASS

the same frequency—the 24 satellites require therefore 12 separate frequencies. The GLONASS satellites also use a PR sequence, but the same sequence is used by all the satellites.

In this section, we will briefly describe how a position is determined and why we need to receive signals from four satellites for this.

We start with the simplest possible situation, assuming that

(1) The satellites and the receiver are constrained to lie in a 2-D plane, and
(2) Both the satellites and the receiver have perfect clocks.

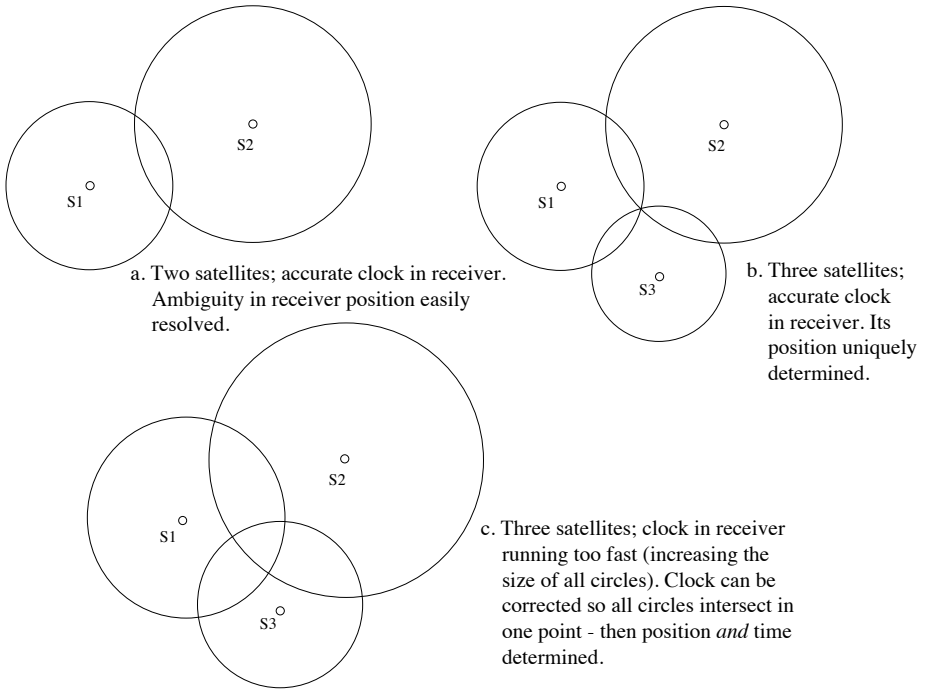


FIGURE 3.3.1. Principle behind how receiver position is calculated in case of 2-D satellite and receiver configuration.

The satellites send out a data package that tells their precise orbits, so their position at any time can be assumed to be perfectly known. They send out their time pulses at exactly known times, and the receiver records accurately when these arrive. Also knowing the speed of light, the receiver can therefore calculate how far it is away from the (known positions) of the satellites. In a 2-D model world, we get a picture as shown in Figure 3.3.1 where we listen in to two satellites. The receiver can be at either of the two places where the circles intersect. In Figure 3.3.1(b), a third satellite is added, and the position becomes uniquely determined.

The assumption of perfect clocks is quite true for the satellites; although their cesium or rubidium clocks have the phenomenal accuracy of up to one part in 10^{13}, they are still corrected from the ground several times a day. The cost and bulk of similar clocks in the receivers would be prohibitive. In reality, the receivers have built-in clocks not much better than a typical wrist watch—the errors can be in the

order of seconds or even minutes. Figure 3.3.1(c) shows what happens if the receiver clock has gone a bit too fast—the receiver would think that all signals had been traveling for a longer time than the actually have. Hence, all the circles will have their radii too large, but all are increased with the same amount. The three circles that should have intersected in one point don't do that any longer. The receiver uses this to calculate a *clock correction*—it determines how much its clock needs to be corrected so the three circles again intersect in one point. After that, it is accurate both in position and in time.

In 3-D, the situation is very similar—only that we need four satellites to determine both position and time, using exactly the same ideas. Two spheres intersect along a circle; a third sphere selects out two possible positions. It takes a fourth satellite to get us a discrepancy allowing the clock to be corrected. So when receiving from four satellites, we can determine both position and time in a 3-D space.

3.4. Test Problem with Numerical Solutions.

With 24 GPS satellites in the sky (not counting spares), as many as 10-12 might be above the horizon at the same time. The orbits are designed so that at least 4 will be in fairly good positions at all times and from any point on earth. Therefore, one is always assured of being able to get a GPS positional fix. To get the best accuracy, it makes sense to utilize information from all satellites that are available. Finding a position usually becomes an over-determined problem ; we have more data that what is minimally needed to get a unique solution.

We will next formulate a numerical test problem, and then discuss two different methods of solving it.

3.4.1. Test problem. We assume that we have six satellites (S1 - S6) and a receiver (R) located as seen in Figure 3.4.1 and described in Table 2.

This data set is 'rigged' so that our answer, receiver position and clock error, all will be integers. This has no significance for any of the algorithms—it just makes the equations shorter to write, and also makes it easier to follow how the convergence in the algorithms is progressing.

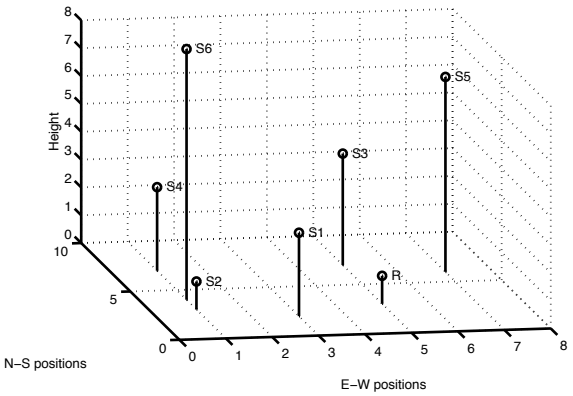


FIGURE 3.4.1. Location of satellites and receiver in the test problem (distance units 1,000 km).

Satellite and receiver position in test problem

GIVEN DATA		
Transmitters (satellites)		Recorded delay (ms) between accurate transmission time and the receive time
Nr	x,y,z - locations (in units of 1,000 km)	according to inaccurate receiver clock
S1	3, 2, 3	10010.00692286
S2	1, 3, 1	10013.34256381
S3	5, 7, 4	10016.67820476
S4	1, 7, 3	10020.01384571
S5	7, 6, 7	10023.34948666
S6	1, 4, 9	10030.02076857
TO BE DETERMINED (4 quantities)		
Receiver location		Clock error
R	5, 3, 1	t = 10,000

TABLE 2. Satellite and receiver locations in test problem

3.4.2. Numerical Solutions. With (x, y, z, t) denoting the unknowns, receiver position and clock error, the nonlinear system to be solved can be written as

$$
\begin{aligned}
(x-3)^2+(y-2)^2+(z-3)^2-[(10010.00692286-t)\cdot c]^2 &= 0\\
(x-1)^2+(y-3)^2+(z-1)^2-[(10013.34256381-t)\cdot c]^2 &= 0\\
(x-5)^2+(y-7)^2+(z-4)^2-[(10016.67820476-t)\cdot c]^2 &= 0\\
(x-1)^2+(y-7)^2+(z-3)^2-[(10020.01384571-t)\cdot c]^2 &= 0\\
(x-7)^2+(y-6)^2+(z-7)^2-[(10023.34948666-t)\cdot c]^2 &= 0\\
(x-1)^2+(y-4)^2+(z-9)^2-[(10030.02076857-t)\cdot c]^2 &= 0
\end{aligned}
\tag{3.1}
$$

where $c = 0.299792458$ (in the unit of 1,000 km/ms). The two numerical methods we will describe below are linearization and Newton's method.

Linearization

The equations (3.1) are nonlinear, but if we expand all the squares, each equation will take the form $x^2+y^2+z^2+t^2c^2+\{\text{linear terms}\}=0$, i.e. if we subtract one of the equations say, the last one, from the rest, all nonlinearities vanish, and we are left with the linear system

$$
\begin{bmatrix}
4 & -4 & -12 & 3.59751\\
0 & -2 & -16 & 2.99792\\
8 & 6 & -10 & 2.39834\\
0 & 6 & -12 & 1.79875\\
12 & 4 & -4 & 1.19917
\end{bmatrix}
\begin{bmatrix} x\\ y\\ z\\ t \end{bmatrix}
=
\begin{bmatrix}
35971.1\\
29957.2\\
24031.4\\
17993.5\\
12059.7
\end{bmatrix}
$$

This overdetermined system can be solved in the least square sense with the methods in Section 11.5, giving

$$x = 5.0000,\quad y = 3.0000,\quad z = 1.0000,\quad t = 10,000\quad .$$

If we have only four satellites visible, the elimination of the nonlinear terms would give us three linear equations in four unknowns. The *echelon form* (Section **??**) allows us to express three of the unknowns in terms of the fourth one. Substituting these expressions into the last one would give us a quadratic equation in the remaining unknown. This quadratic will typically have two solutions, only one of which will correspond to a 'reasonable' position. We can then immediately find the remaining unknowns.

Newton's method

It was a very unusual circumstance that allowed the nonlinear terms in all but one of the equations (3.1) to be eliminated. Linearization—as it occurs in Newton's method—is much more general. It does not rely on any particular coincidences in the structure of the equations, nor do we end up with one equation less to work with. With numerical linearization, the solution process becomes iterative, and we need to provide a starting guess. How close such a guess has to be varies from problem to problem. As we shall see, this is not a difficulty in the present case.

Applying Newton's method, as described in (*to be written*) to (3.1) gives the iteration

$$x_{n+1} = x_n + \Delta x_n,\ y_{n+1} = y_n + \Delta y_n,\ z_{n+1} = z_n + \Delta z_n,\ t_{n+1} = t_n + \Delta t_n,$$

where the updates are obtained from solving the (overdetermined) linear system

$$\begin{bmatrix} 2(x_n-3) & 2(y_n-2) & 2(z_n-3) & 2c^2(10010.00692286 - t_n) \\ 2(x_n-1) & 2(y_n-3) & 2(z_n-1) & 2c^2(10013.34256381 - t_n) \\ 2(x_n-5) & 2(y_n-7) & 2(z_n-4) & 2c^2(10016.67820476 - t_n) \\ 2(x_n-1) & 2(y_n-7) & 2(z_n-3) & 2c^2(10020.01384571 - t_n) \\ 2(x_n-7) & 2(y_n-6) & 2(z_n-7) & 2c^2(10023.34948666 - t_n) \\ 2(x_n-1) & 2(y_n-4) & 2(z_n-9) & 2c^2(10030.02076857 - t_n) \end{bmatrix} \begin{bmatrix} \Delta x_n \\ \Delta y_n \\ \Delta z_n \\ \Delta t_n \end{bmatrix} =$$

$$= - \begin{bmatrix} (x_n-3)^2+(y_n-2)^2+(z_n-3)^2-[(10010.00692286 - t_n)\cdot c]^2 \\ (x_n-1)^2+(y_n-3)^2+(z_n-1)^2-[(10013.34256381 - t_n)\cdot c]^2 \\ (x_n-5)^2+(y_n-7)^2+(z_n-4)^2-[(10016.67820476 - t_n)\cdot c]^2 \\ (x_n-1)^2+(y_n-7)^2+(z_n-3)^2-[(10020.01384571 - t_n)\cdot c]^2 \\ (x_n-7)^2+(y_n-6)^2+(z_n-7)^2-[(10023.34948666 - t_n)\cdot c]^2 \\ (x_n-1)^2+(y_n-4)^2+(z_n-9)^2-[(10030.02076857 - t_n)\cdot c]^2 \end{bmatrix}$$

The next issue is to find a start guess (x_0, y_0, z_0, t_0). Knowing that the travel times for signals cannot be negative, one can for example choose t_0 as the shortest time recorded, i.e. $t_0 = 10010.00692286$. Let us also guess that we are at the location

$x = y = z = 0$. This is an extremely coarse guess; the errors of 5000 and 3000 km in the x- and y-directions are the size of a continent.

The iterations proceed as follows:

n	x	y	z	t
0	0.0	0.0	0.0	10010.007
1	6.368727	0.374601	-2.403971	9985.218
2	4.984063	3.018241	1.046303	10000.266
3	5.000160	2.999808	0.999585	9999.998
4	5.000000	3.000000	1.000000	10000.000

We see the typical signs of quadratic convergence—a doubling of correct digits once the iterations have 'settled in'. Here, convergence to all the precision we want is obtained after just 4 iterations—a common situation when a reasonable guess is available. The numerical errors are at this point reduced to better than 1 m in distance and 1 μs in time.

3.5. Error Analysis.

A recording of a position is not of much use if it is not accompanied by some form of error estimate. There are many sources of errors encountered in the GPS process. Table 3 is a very schematic summary of how much different sources typically contribute to the error in the readings.

We will here carry out one example of error analysis to illustrate the process of tracing how different sources of errors in input data can carry through to errors in the computed position. One key feature this will illustrate is that error analysis generally is *linear*, provided of course that the errors remain small enough; the final effect of different error sources can be studied separately, and effects can be added together for a 'worst case' estimate.

> If there are many measurements available for the same quantity, it often happens that the errors will fluctuate randomly and partly cancel when averaging. Statistical tools should then be applied so as not to get unduly

Typical errors (in meters) in computed distance to each satellite due to different error sources

Source	**Standard GPS**
Satellite clocks[1]	1.5
Orbit errors	2.5
Ionospheric delays[2]	5.0
Tropospheric delays	0.5
Receiver noise	0.3
Multipath	0.6
Typical resulting positional accuracy	
Horizontal	10
Vertical	40

[1] Clock stability: Rubidium 10^{-11}–10^{-12}, cesium $10^{-12} - -10^{-13}$. (Cf. hydrogen maser 10^{-16} and typical watch 10^{-6})

[2] Can be reduced to 0.5 - 1.0 m if receiving on two frequencies; delay proportional to (number of electrons)$/f^2$where f is the signal frequency.

TABLE 3. Summary of error sources

pessimistic 'worst case' errors only. For example, with n estimates, the expected error often decreases like $1/\sqrt{n}$.

We suppose here that we have only the top four equations of the set (3.1) available. We write these in the form

$$
\begin{aligned}
f_1 \equiv\ & (x-3)^2 + (y-2)^2 + (z-3)^2 - [(T_1 + t_1 - t)\cdot c]^2 &= 0 \\
f_2 \equiv\ & (x-1)^2 + (y-3)^2 + (z-1)^2 - [(T_2 + t_2 - t)\cdot c]^2 &= 0 \\
f_3 \equiv\ & (x-5)^2 + (y-7)^2 + (z-4)^2 - [(T_3 + t_3 - t)\cdot c]^2 &= 0 \\
f_4 \equiv\ & (x-1)^2 + (y-7)^2 + (z-3)^2 - [(T_4 + t_4 - t)\cdot c]^2 &= 0
\end{aligned}
\tag{3.1}
$$

where the recorded time delays according to the receiver's very inaccurate clock are

$$
\begin{aligned}
T_1 &= 10010.00692286 \\
T_2 &= 10013.34256381 \\
T_3 &= 10016.67820476 \\
T_4 &= 10020.01384571
\end{aligned}
$$

We have also introduced additional variables t_1, t_2, t_3, t_4 which represent further errors in the timing signals from each of the four satellites. Causes for these errors could for example be ionospheric delays. We want to estimate the uncertainty is in the position x, y, z and corrected time t, as functions of variations in t_1, t_2, t_3, and t_4 .

Simplified one variable / one equation situation:

Had our system of equations been just one scalar equation in one variable

$$f_1 \equiv (x-3)^2 - [(T_1 + t_1 - t)\cdot c]^2 = 0 \tag{3.2}$$

we would first set $t_1 = 0$ i.e. assume this extra error was not there, and solve for x. Next, we would re-introduce t_1 and ask how the variations in t_1 will influence x. Hence, we view x as a function of t_1 : $x = x(t_1)$. Differentiating (13.1) with respect to t_1 gives

$$\frac{df_1}{dt_1} = 2(x-3)\frac{dx}{dt_1} - 2c^2(T_1 + t_1) = 0.$$

Now we again set $t_1 = 0$ and solve for $\frac{dx}{dt_1}$. That derivative is precisely what we want—a measure of how much x will change for small changes in t_1.

Original four variable / four equation situation:

In (3.1), we similarly have $x = x(t_1, t_2, t_3, t_4)$, $y = y(t_1, t_2, t_3, t_4)$, $z = z(t_1, t_2, t_3, t_4)$, $t = t(t_1, t_2, t_3, t_4)$. Differentiating f_i with respect to t_j becomes an exercise in using the chain rule:

$$\frac{df_i}{dt_j} = \frac{\partial f_i}{\partial x}\frac{\partial x}{\partial t_j} + \frac{\partial f_i}{\partial y}\frac{\partial y}{\partial t_j} + \frac{\partial f_i}{\partial z}\frac{\partial z}{\partial t_j} + \frac{\partial f_i}{\partial t}\frac{\partial t}{\partial t_j} + \frac{\partial f_i}{\partial t_j} = 0, \qquad i, j = 1, ..., 4.$$

This is most clearly written in matrix form

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial t} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial t} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial t} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial t} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial t_1} & \frac{\partial x}{\partial t_2} & \frac{\partial x}{\partial t_3} & \frac{\partial x}{\partial t_4} \\ \frac{\partial y}{\partial t_1} & \frac{\partial y}{\partial t_2} & \frac{\partial y}{\partial t_3} & \frac{\partial y}{\partial t_4} \\ \frac{\partial z}{\partial t_1} & \frac{\partial z}{\partial t_2} & \frac{\partial z}{\partial t_3} & \frac{\partial z}{\partial t_4} \\ \frac{\partial t}{\partial t_1} & \frac{\partial t}{\partial t_2} & \frac{\partial t}{\partial t_3} & \frac{\partial t}{\partial t_4} \end{bmatrix} = - \begin{bmatrix} \frac{\partial f_1}{\partial t_1} & \frac{\partial f_1}{\partial t_2} & \frac{\partial f_1}{\partial t_3} & \frac{\partial f_1}{\partial t_4} \\ \frac{\partial f_2}{\partial t_1} & \frac{\partial f_2}{\partial t_2} & \frac{\partial f_2}{\partial t_3} & \frac{\partial f_2}{\partial t_4} \\ \frac{\partial f_3}{\partial t_1} & \frac{\partial f_3}{\partial t_2} & \frac{\partial f_3}{\partial t_3} & \frac{\partial f_3}{\partial t_4} \\ \frac{\partial f_4}{\partial t_1} & \frac{\partial f_4}{\partial t_2} & \frac{\partial f_4}{\partial t_3} & \frac{\partial f_4}{\partial t_4} \end{bmatrix}$$

Taking partial derivatives of (3.1) gives all the entries of the first and last matrices above

$$2\begin{bmatrix} x-3 & y-2 & z-3 & c^2(T_1+t_1-t) \\ x-1 & y-3 & z-1 & c^2(T_2+t_2-t) \\ x-5 & y-7 & z-4 & c^2(T_3+t_3-t) \\ x-1 & y-7 & z-3 & c^2(T_4+t_4-t) \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial t_1} & \frac{\partial x}{\partial t_2} & \frac{\partial x}{\partial t_3} & \frac{\partial x}{\partial t_4} \\ \frac{\partial y}{\partial t_1} & \frac{\partial y}{\partial t_2} & \frac{\partial y}{\partial t_3} & \frac{\partial y}{\partial t_4} \\ \frac{\partial z}{\partial t_1} & \frac{\partial z}{\partial t_2} & \frac{\partial z}{\partial t_3} & \frac{\partial z}{\partial t_4} \\ \frac{\partial t}{\partial t_1} & \frac{\partial t}{\partial t_2} & \frac{\partial t}{\partial t_3} & \frac{\partial t}{\partial t_4} \end{bmatrix} =$$

$$= 2\begin{bmatrix} c^2(T_1+t_1-t) & 0 & 0 & 0 \\ 0 & c^2(T_2+t_2-t) & 0 & 0 \\ 0 & 0 & c^2(T_3+t_3-t) & 0 \\ 0 & 0 & 0 & c^2(T_4+t_4-t) \end{bmatrix}.$$

Writing this as $A\,X = B$, we can solve for X by simply multiplying by A^{-1} from the left, or better still—view this as a linear system of equations with four RHSs and four side-by-side solution vectors. Using the known values for T_i, setting $t_i = 0$ and using our numerical solution $x = 5,\ y = 3,\ z = 1$ and $t = 10,000$ gives

(3.3)

$$\begin{bmatrix} \frac{\partial x}{\partial t_1} & \frac{\partial x}{\partial t_2} & \frac{\partial x}{\partial t_3} & \frac{\partial x}{\partial t_4} \\ \frac{\partial y}{\partial t_1} & \frac{\partial y}{\partial t_2} & \frac{\partial y}{\partial t_3} & \frac{\partial y}{\partial t_4} \\ \frac{\partial z}{\partial t_1} & \frac{\partial z}{\partial t_2} & \frac{\partial z}{\partial t_3} & \frac{\partial z}{\partial t_4} \\ \frac{\partial t}{\partial t_1} & \frac{\partial t}{\partial t_2} & \frac{\partial t}{\partial t_3} & \frac{\partial t}{\partial t_4} \end{bmatrix} = \begin{bmatrix} 0.149896 & -0.349758 & -0.624568 & 0.824429 \\ 0.149896 & 0.249827 & 0.124914 & -0.524637 \\ -0.449689 & 0.749481 & 0.374741 & -0.674533 \\ -0.500000 & 2.166667 & 2.083333 & -2.750000 \end{bmatrix}$$

This tells how sensitive each variable x, y, z, t is to the small errors in the timings t_1, t_2, t_3, t_4 for the signals from the four satellites.

If the timings are all accurate to within 0.1 μs = 0.0001 ms, the worst case errors in the results can be calculated using the formula,

$$\Delta x \approx \frac{\partial x}{\partial t_1}\Delta t_1 + \frac{\partial x}{\partial t_2}\Delta t_2 + \frac{\partial x}{\partial t_3}\Delta t_3 + \frac{\partial x}{\partial t_4}\Delta t_4,$$

to obtain

$$\begin{array}{lllll} x\text{-dir} & (0.1499 + 0.3498 + 0.6246 + 0.8244) & \cdot\, 0.1 \text{ km} & \approx 195 \text{ m} \\ y\text{-dir} & (0.1499 + 0.2498 + 0.1249 + 0.5246) & \cdot\, 0.1 \text{ km} & \approx 105 \text{ m} \\ z\text{-dir} & (0.4497 + 0.7495 + 0.3747 + 0.6745) & \cdot\, 0.1 \text{ km} & \approx 225 \text{ m} \\ t\text{-err} & (0.5000 + 2.5000 + 2.0833 + 2.7500) & \cdot\, 0.1\ \mu\text{s} & \approx 0.75\ \mu\text{s} \end{array}$$

In this case, the positional error turns out to be largest in the z-direction. The best time the receiver can calculate is about 7.5 times less accurate than the precision of the incoming signals.

The analysis above was based on reception from only the first four of our six satellites, giving us a 4×4 matrix in (3.3) with sensitivity information. In a similar way, we could have analyzed the full 6-satellite case to arrive at

$$\begin{bmatrix} \frac{\partial x}{\partial t_1} & \frac{\partial x}{\partial t_2} & \frac{\partial x}{\partial t_3} & \frac{\partial x}{\partial t_4} & \frac{\partial x}{\partial t_5} & \frac{\partial x}{\partial t_6} \\ \frac{\partial y}{\partial t_1} & \frac{\partial y}{\partial t_2} & \frac{\partial y}{\partial t_3} & \frac{\partial y}{\partial t_4} & \frac{\partial y}{\partial t_5} & \frac{\partial y}{\partial t_6} \\ \frac{\partial z}{\partial t_1} & \frac{\partial z}{\partial t_2} & \frac{\partial z}{\partial t_3} & \frac{\partial z}{\partial t_4} & \frac{\partial z}{\partial t_5} & \frac{\partial z}{\partial t_6} \\ \frac{\partial t}{\partial t_1} & \frac{\partial t}{\partial t_2} & \frac{\partial t}{\partial t_3} & \frac{\partial t}{\partial t_4} & \frac{\partial t}{\partial t_5} & \frac{\partial t}{\partial t_6} \end{bmatrix} =$$

$$= \begin{bmatrix} -0.0362 & -0.1061 & -0.0267 & 0.2221 & -0.4185 & 0.3655 \\ 0.1240 & 0.2097 & -0.1619 & -0.3125 & 0.2007 & -0.0602 \\ -0.0643 & 0.3353 & 0.0169 & -0.0456 & 0.2505 & -0.6213 \\ -0.3616 & 1.1765 & 0.0047 & -0.5127 & 1.4550 & -1.4852 \end{bmatrix}$$

This time, the worst-case errors are notably smaller even though none of the inputs are any more accurate:

$$\begin{array}{ll} x\text{-dir} & \approx 118 \text{ m} \\ y\text{-dir} & \approx 107 \text{ m} \\ z\text{-dir} & \approx 133 \text{ m} \\ t\text{-err} & \approx 0.50\,\mu\text{s} \end{array}$$

The improvement in expected errors is better still—the probability that the sign and size of all errors conspire to create a maximum error situation is far less likely the more independent input variables that enter, i.e. cancellation of errors becomes increasingly likely.

3.6. Pseudorandom Sequences.

Two of the problems that arise in connection with transmitting timing pulses from the satellites are

(1) how to send a very sharp pulse, so that its arrival can be very accurately timed, without needing to use a wide bandwidth (this will be explained in more detail in a moment), and
(2) how to allow all the satellites to transmit on exactly the same frequency without their signals interfering with each other.

One mathematical construct—pseudo-random (PR) sequences—resolves both these problems very nicely. Once we have seen how they resolve the first problem, it will be clear how the solution to the second one follows.

To appreciate the dilemma posed in point 1, we recall the function-Fourier transform pair $f(x) = e^{-\alpha x^2}$, $\widehat{f}(\omega) = \frac{1}{\sqrt{4\pi\alpha}} e^{-\omega^2/(4\alpha)}$(c.f. Table 2 in Section 9.3). If the parameter α is large, the function $f(x)$ will be a sharp spike, allowing its position to be accurately pinpointed (think here of x as time). However, $\widehat{f}(\omega)$ will then have a very broad maximum, i.e. occupy a lot of frequency space (bandwidth). On the other hand, making the parameter α small will make the pulse broad and difficult to accurately determine its position (e.g. center of its peak).

The answer to this dilemma turns out to be that after all one does not need a sharp pulse in order to achieve a fine time resolution—a suitably structured signal of long duration can also achieve a fine time resolution. Each satellite sends its own PR signal as illustrated in Figure 3.6.1; very small up/down-variations in frequency according to a pattern that looks random, but is fixed for each satellite, and repeats periodically after 1023 chip times of 1 μs each. Thus, the whole pattern repeats roughly each ms. During each 1 μs chip time, the carrier (at 1575.42 MHz) goes through about 1,500 cycles—enough to detect which one of the two very nearby frequency levels that is used. A receiver knows the pseudo-random sequence (for each satellite), and slides its copies relative to the received signal until the match gets perfect (c.f. again Figure 3.6.1).

The correlation function $f(x)$ of two functions, $g(x)$ and $h(x)$, is defined as,

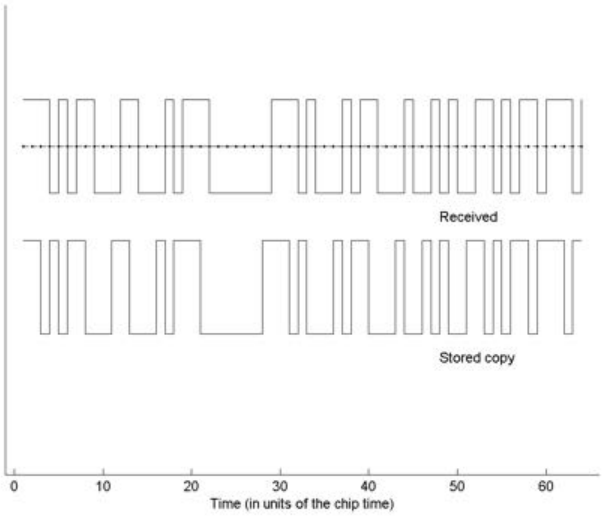


FIGURE 3.6.1. Pseudo-random (PR) received code compared with a copy stored in the receiver (in this picture shifted one chip time).

$$f(x) = \int g(s)h(s+x)ds,$$

i.e. it measures the 'similarity' between two functions as they are shifted aver each other. Figure 3.6.2 shows the correlation function of the two sequences of Figure 3.6.1 (where summation replaces the integral), displayed as a function of the sideways shift in a case of a periodic PR sequence of length 128.

Whenever the shift is a multiple of 128, we get here a very sharp spike of perfectly triangular shape with a base width of 2 chip times. Even a small misalignment in time—a fraction of the chip time—will notably bring down the correlation. A timing error of about 1/20th of a chip time would lead to a distance error of about 15 m; typical of the hardware capabilities of present low cost GPS receivers.

Of course, with 24 satellites in orbit, each receiver has to test the received signal against 24 copies.

The whole PR sequence repeats about each 1 ms. Given the speed of light, this corresponds to about 300 km. All distances to satellites become undetermined with respect to multiples of this distance. The easiest way to avoid this ambiguity

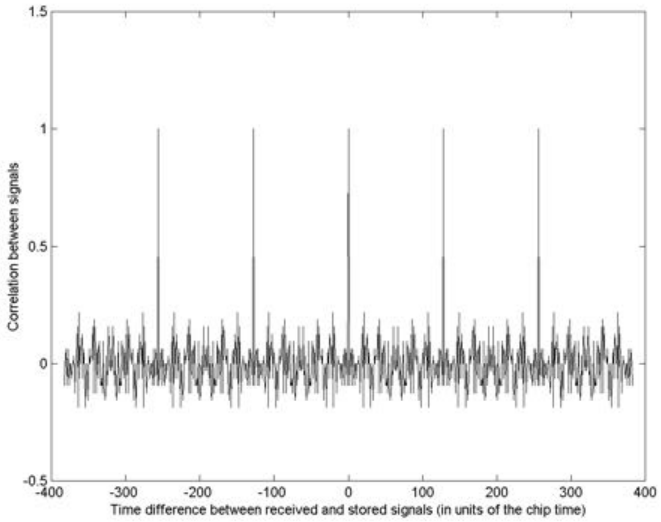


FIGURE 3.6.2. Correlation between a periodic PR code of length 128 and a translated copy of itself. The sharp spikes appear here every 128 chip times.

is to require that the receiver has to be initialized—the user has first to enter an approximate position that is accurate to better than 150 km, thus removing the arbitrariness. There are (at least) two other ways to resolve this problem:

i.: Guessing wrong by 300 km (when receiving signals from the minimal number of satellites required to get a position) is very likely to produce a position hundreds of kilometers under ground or above ground—easily rejected for all receivers not used in space crafts, and

ii.: When receiving signals from more satellites, a wrong guess will quite certainly produce a contradiction—no single point will satisfy all the distance requirements.

This second observation allows us to state the problem of finding our position as one of locating a point so that the distance to every observed satellite is equal to an unknown integer plus a known (measured) fractional part of this 300 km distance. Determining these integers would be an example of *integer programming*—the task of

finding best approximations to a problem for which several unknowns are constrained to integer values only.

> Integer programming is actually of great importance for GPS, but in a slightly different way. To get the highest possible accuracy, we apply the idea not to the 300 km PR sequence repeat distance, but to the approximately 0.2 m wavelength of the 1745.42 MHz carrier wave. Trying to lock on to the phase angle of the carrier oscillations, we get distances to within an unknown multiple of 0.2 m. With a good positional guess, say from differentially corrected GPS, we may have only about 50—100 multiples to be concerned with. Finding a position that gives a correct carrier phase for all available satellites can pinpoint just which carrier multiple we are locked onto for each of the satellites in view, i.e. an error better than 0.2 m. Finally, being locked onto exactly the right integer multiple of the carrier oscillation, the phase angle can be reconciled to maybe one part in 200. The accuracy is now down to about 1 mm—not bad considering that the radio signals are of quite narrow bandwidth and how fast these satellites fly far out in space! For more discussion on this issue of locking onto an individual carrier wave and its phase angle—and on integer programming in the context of GPS—see Strang and Borre [**?**].