

Tomographic reconstruction

Lecture 1: Radon transform, filtered backprojection and inverse problems

Camille Pouchol¹
pouchol@kth.se

Department of Mathematics
KTH, Stockholm

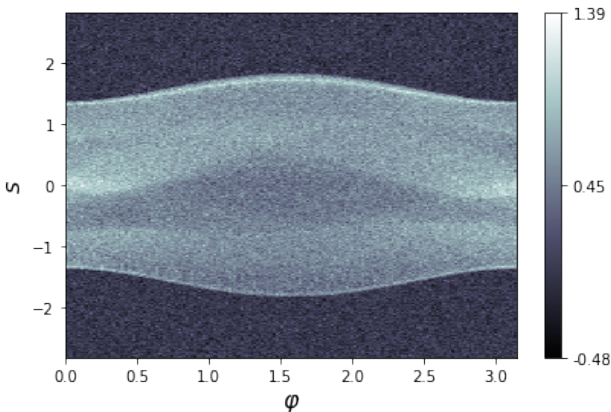
HL2027: 3D Image Reconstruction and Analysis in Medicine
January/February 2019

¹based on a previous lecture by Sebastian Banert, Ozan Öktem

Overview of the tomographic reconstruction module (1)

Main goal: understanding how one gets an **image** out of a **scanner**.

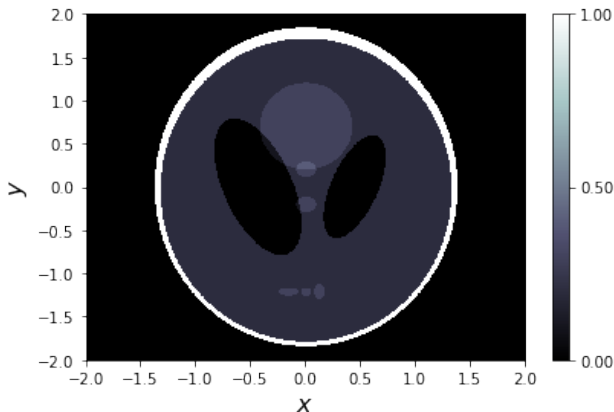
Here is what you would (typically) get out of CT-scanner...



Overview of the tomographic reconstruction module (1)

Main goal: understanding how one gets an **image out of a scanner**.

... would you have guessed it comes from the image below?



Overview of the tomographic reconstruction module (1)

Main goal: understanding how one gets an **image out of a scanner**.

- X-rays of incidental energy I_0 are attenuated when passing through the body/object, recovered energy is

$$I_1 = I_0 \exp(-\text{"matter encountered by the ray"}).$$

- Scanner measures I_1 for many different incident rays, so you have many values "matter encountered along some ray".

From all this information, want to recover the matter density everywhere: that's the image you are looking for.

Overview of the tomographic reconstruction module (2)

Main goal: understanding how one gets an **image out of a scanner**.

This requires

- defining what we mean by an image: mathematical viewpoint of an image as a function f ,
- then "matter encountered by the ray" $= \int_L f(z) dz$ where L is some line,
- becomes the problem of *recovering a function knowing its integral on many lines*.

For this and because it covers most instances of medical imaging, we will build a general abstract setting, that of **inverse problems**, which covers CT-scan **but not only**.

Overview of the tomographic reconstruction module (3)

Main goal: understanding how one gets an **image out of a scanner**.

Problem: one would want to find the value of f at each point in space. This is in practice impossible and one only wants the averages of f over some small subsets of space (pixels/voxels). Requires the notion of **discretisation** of a space. In the end, the problem boils down to a set of (linear) equations.

You will

- get notions about why solving the equations is difficult, this is the notion of **ill-posedness**,
- learn some algorithms that tackle the problem, thanks to the theory of **regularisation**,
- implement some of them.

Formalising the notion of an image

What is an image?

- **User's viewpoint:** a quantity spatially distributed in 2D/3D.
- **Mathematical viewpoint:** an image is a function in 2D/3D:
 - **Grey-scale images:** real valued function in 2D/3D, *i.e.*
 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ where $n = 2$ for 2D images and $n = 3$ for 3D images.
 - **Colour images:** vector valued function in 2D/3D, *i.e.*
 $f: \mathbb{R}^n \rightarrow \mathbb{R}^k$ where $n = 2$ or 3 and k is the number of colour channels (e.g. $k = 3$ for red, green & blue).

We will only work with grey-scale images.

- **Support of an image:** the set $\Omega \subset \mathbb{R}^n$ where the image is defined, usually a rectangular region.
- **Dynamic range of an image:** the range of values in \mathbb{R}^k that an image $f: \Omega \rightarrow \mathbb{R}^k$ can attain.

Image reconstruction

In many imaging applications, the image we seek is only indirectly observed. This is especially the case when we seek to observe a 3D image.

- 3D/2D tomography in medical imaging
- Geophysical exploration
- 3D microscopy
- Radar imaging

Image reconstruction

Image reconstruction as an inverse problem

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

- $f_{\text{true}}: \Omega \rightarrow \mathbb{R}$ is the image that is to be recovered and X (reconstruction space) is the set of feasible images.
- $g \in \mathbb{R}^m$ is the measured data, i.e., the m numeric quantities recorded by the imaging device.
- $\mathcal{A}: X \rightarrow Y$ is the forward operator and Y (data space) is the set of possible continuum data. \mathcal{A} models the imaging device for continuum data in absence of noise and measurement errors.
- $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ models how continuum data is digitized (sampling operator) during the imaging.
- $g_{\text{noise}} \in \mathbb{R}^m$ is the noise component of measured data.

Image reconstruction

Image reconstruction as an inverse problem

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

- $f_{\text{true}}: \Omega \rightarrow \mathbb{R}$ is the image that is to be recovered and X (reconstruction space) is the set of feasible images.
- $g \in \mathbb{R}^m$ is the **measured data**, i.e., the m numeric quantities recorded by the imaging device.
- $\mathcal{A}: X \rightarrow Y$ is the forward operator and Y (data space) is the set of possible continuum data. \mathcal{A} models the imaging device for continuum data in absence of noise and measurement errors.
- $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ models how continuum data is digitized (sampling operator) during the imaging.
- $g_{\text{noise}} \in \mathbb{R}^m$ is the noise component of measured data.

Image reconstruction

Image reconstruction as an inverse problem

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

- $f_{\text{true}}: \Omega \rightarrow \mathbb{R}$ is the image that is to be recovered and X (reconstruction space) is the set of feasible images.
- $g \in \mathbb{R}^m$ is the measured data, i.e., the m numeric quantities recorded by the imaging device.
- $\mathcal{A}: X \rightarrow Y$ is the **forward operator** and Y (**data space**) is the set of possible continuum data. \mathcal{A} models the imaging device for continuum data in absence of noise and measurement errors.
- $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ models how continuum data is digitized (sampling operator) during the imaging.
- $g_{\text{noise}} \in \mathbb{R}^m$ is the noise component of measured data.

Image reconstruction

Image reconstruction as an inverse problem

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

- $f_{\text{true}}: \Omega \rightarrow \mathbb{R}$ is the image that is to be recovered and X (reconstruction space) is the set of feasible images.
- $g \in \mathbb{R}^m$ is the measured data, *i.e.*, the m numeric quantities recorded by the imaging device.
- $\mathcal{A}: X \rightarrow Y$ is the forward operator and Y (data space) is the set of possible continuum data. \mathcal{A} models the imaging device for continuum data in absence of noise and measurement errors.
- $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ models how continuum data is digitized (**sampling operator**) during the imaging.
- $g_{\text{noise}} \in \mathbb{R}^m$ is the noise component of measured data.

Image reconstruction

Image reconstruction as an inverse problem

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

- $f_{\text{true}}: \Omega \rightarrow \mathbb{R}$ is the image that is to be recovered and X (reconstruction space) is the set of feasible images.
- $g \in \mathbb{R}^m$ is the measured data, *i.e.*, the m numeric quantities recorded by the imaging device.
- $\mathcal{A}: X \rightarrow Y$ is the forward operator and Y (data space) is the set of possible continuum data. \mathcal{A} models the imaging device for continuum data in absence of noise and measurement errors.
- $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ models how continuum data is digitized (sampling operator) during the imaging.
- $g_{\text{noise}} \in \mathbb{R}^m$ is the **noise component** of measured data.

Images

The infinite dimensional setting

Reconstruction space X

- Set of possible functions $f : \Omega \rightarrow \mathbb{R}$ (images)
- Usually some (infinite dimensional) vector space over \mathbb{R} , i.e.,

$$f + h \in X \text{ and } \alpha f \in X \quad \text{for any } f, h \in X \text{ and } \alpha \in \mathbb{R}.$$

Square integrable functions: $L^2(\Omega)$ is infinite dimensional and

$$\int_{\Omega} |f(x)|^2 dx < \infty \quad \text{whenever } f \in L^2(\Omega).$$

$L^2(\Omega)$ has inner-product and norm (Hilbert space):

$$\begin{aligned} \langle f, h \rangle_{L^2(\Omega)} &:= \int_{\Omega} f(x)h(x) dx \quad \text{for } f, h \in L^2(\Omega) \\ \|f\|_2 &:= \sqrt{\langle f, f \rangle_{L^2(\Omega)}} := \left(\int_{\Omega} f(x)^2 dx \right)^{1/2}. \end{aligned}$$

Hilbert spaces: generalises the notion of an inner product to infinite dimensional vector spaces.

Images

The infinite dimensional setting

Reconstruction space X

- Set of possible functions $f : \Omega \rightarrow \mathbb{R}$ (images)
- Usually some (infinite dimensional) vector space over \mathbb{R} , i.e.,

$$f + h \in X \text{ and } \alpha f \in X \quad \text{for any } f, h \in X \text{ and } \alpha \in \mathbb{R}.$$

Square integrable functions: $L^2(\Omega)$ is infinite dimensional and

$$\int_{\Omega} |f(x)|^2 dx < \infty \quad \text{whenever } f \in L^2(\Omega).$$

$L^2(\Omega)$ has inner-product and norm (Hilbert space):

$$\langle f, h \rangle_{L^2(\Omega)} := \int_{\Omega} f(x)h(x) dx \quad \text{for } f, h \in L^2(\Omega)$$

$$\|f\|_2 := \sqrt{\langle f, f \rangle_{L^2(\Omega)}} := \left(\int_{\Omega} f(x)^2 dx \right)^{1/2}.$$

Hilbert spaces: generalises the notion of an inner product to infinite dimensional vector spaces.

Discretisation: computers cannot handle elements in infinite dimensional vector spaces. Need to replace the infinite dimensional vector space with a finite dimensional counterpart.

Basis expansion

Assume X is a Hilbert space with norm $\|\cdot\|_X$ and $\{\phi_j\}_j \subset X$ is a fixed set (dictionary/frame). Define $X_n \subset X$ as the linear span of $\{\phi_1, \dots, \phi_n\}$, so for any element $f \in X_n$ then there exist real numbers $\alpha_j \in \mathbb{R}$ (that depend on f) such that

$$f(x) = \sum_{j=1}^n \alpha_j \phi_j(x) \quad \text{for all } x \in \Omega.$$

If $\{\phi_j\}_j$ are a basis (linearly independent), then X_n is an n -dimensional vector space over \mathbb{R} .

Discretisation: computers cannot handle elements in infinite dimensional vector spaces. Need to replace the infinite dimensional vector space with a finite dimensional counterpart.

Basis expansion

Assume X is a Hilbert space with norm $\| \cdot \|_X$ and $\{\phi_j\}_j \subset X$ is a fixed set (dictionary/frame). Define $X_n \subset X$ as the linear span of $\{\phi_1, \dots, \phi_n\}$, so for any element $f \in X_n$ then there exist real numbers $\alpha_j \in \mathbb{R}$ (that depend on f) such that

$$f(x) = \sum_{j=1}^n \alpha_j \phi_j(x) \quad \text{for all } x \in \Omega.$$

If $\{\phi_j\}_j$ are a basis (linearly independent), then X_n is an n -dimensional vector space over \mathbb{R} .

How to construct α_j 's from a given f

Given $f \in X$, it is natural to seek an element $\tilde{f} \in X_n$ that is the “best approximation” to f .

Stated equivalently, given f we seek real numbers $\alpha_j \in \mathbb{R}$ (that depend on f) that minimises the “difference” between f and

$$\tilde{f}(x) := \sum_{j=1}^n \alpha_j \phi_j(x) \quad \text{for } x \in \Omega.$$

Solution: choose \tilde{f} that minimises $\|f - \tilde{f}\|_X$. If $\{\phi_j\}_j$ is an orthonormal basis ($\langle \phi_i, \phi_j \rangle_X = 0$ if $i \neq j$ and $= 1$ if $i = j$), then this corresponds to choosing $\tilde{f} \in X_n$ as the orthogonal projection of f onto X_n , so

$$\alpha_j := \langle f, \phi_j \rangle_X = \int_{\Omega} f(x) \phi_j(x) \, dx.$$

How to construct α_j 's from a given f

Given $f \in X$, it is natural to seek an element $\tilde{f} \in X_n$ that is the “best approximation” to f .

Stated equivalently, given f we seek real numbers $\alpha_j \in \mathbb{R}$ (that depend on f) that minimises the “difference” between f and

$$\tilde{f}(x) := \sum_{j=1}^n \alpha_j \phi_j(x) \quad \text{for } x \in \Omega.$$

Solution: choose \tilde{f} that minimises $\|f - \tilde{f}\|_X$. If $\{\phi_j\}_j$ is an orthonormal basis ($\langle \phi_i, \phi_j \rangle_X = 0$ if $i \neq j$ and $= 1$ if $i = j$), then this corresponds to choosing $\tilde{f} \in X_n$ as the orthogonal projection of f onto X_n , so

$$\alpha_j := \langle f, \phi_j \rangle_X = \int_{\Omega} f(x) \phi_j(x) \, dx.$$

Images

Discretisation – Voxel basis

Sampling (uniform): sub-divide (the rectangular) region $\Omega \subset \mathbb{R}^3$ into n non-overlapping (rectangular) sub-regions Ω_j , voxels, of equal size that cover Ω .

Voxel basis: given a sampling, define the voxel basis as

$$\phi_j(x) := \frac{1}{\text{Vol}(\Omega_j)} \begin{cases} 1 & \text{if } x \text{ is in the } j\text{:th voxel } \Omega_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then $\{\phi_j\}_j$ is an orthonormal basis and the orthogonal projection \tilde{f} of f onto X_n is given by

$$\tilde{f}(x) := \sum_{j=1}^n \alpha_j \phi_j(x) \quad \text{for } x \in \Omega$$

where

$$\alpha_j := \langle f, \phi_j \rangle_X = \frac{1}{\text{Vol}(\Omega_j)} \int_{\Omega_j} f(x) \, dx$$

is the average of f in the j :th voxel.

Images

Discretisation – Voxel basis

Sampling (uniform): sub-divide (the rectangular) region $\Omega \subset \mathbb{R}^3$ into n non-overlapping (rectangular) sub-regions Ω_j , voxels, of equal size that cover Ω .

Voxel basis: given a sampling, define the voxel basis as

$$\phi_j(x) := \frac{1}{\text{Vol}(\Omega_j)} \begin{cases} 1 & \text{if } x \text{ is in the } j\text{:th voxel } \Omega_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then $\{\phi_j\}_j$ is an orthonormal basis and the orthogonal projection \tilde{f} of f onto X_n is given by

$$\tilde{f}(x) := \sum_{j=1}^n \alpha_j \phi_j(x) \quad \text{for } x \in \Omega$$

where

$$\alpha_j := \langle f, \phi_j \rangle_X = \frac{1}{\text{Vol}(\Omega_j)} \int_{\Omega_j} f(x) dx$$

is the average of f in the j :th voxel.

Images

Discretisation – Digital images

Basic assumptions

- $X \subset L^2(\Omega)$ is a sub-space
- A sampling of the domain Ω into voxels
- $\{\phi_j\}_j$ is a voxel basis based on the sampling

Discretisation: to each image $f \in X$ we can associate a unique vector (a finite dimensional representation) $\alpha \in \mathbb{R}^n$ as follows:

$$\alpha := (\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n \quad \text{where} \quad \alpha_j := \langle f, \phi_j \rangle_X.$$

Digital image

- A vector representing grey-scale intensities corresponding to a voxel.
- Grey-scale values are usually mapped to integers $\{0, \dots, 2^p - 1\}$ from white to black (p -bit image). We usually scale it back to, say, $[0, 1]$ from 0 (black) to 1 (white).
 - 2-bit image (binary images) corresponds to two values.
 - 8-bit corresponds to 256 different values.
 - 32-bit image corresponds to 4 294 967 296 different values.

Images

Discretisation – Digital images

Basic assumptions

- $X \subset L^2(\Omega)$ is a sub-space
- A sampling of the domain Ω into voxels
- $\{\phi_j\}_j$ is a voxel basis based on the sampling

Discretisation: to each image $f \in X$ we can associate a unique vector (a finite dimensional representation) $\alpha \in \mathbb{R}^n$ as follows:

$$\alpha := (\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n \quad \text{where} \quad \alpha_j := \langle f, \phi_j \rangle_X.$$

Digital image

- A vector representing grey-scale intensities corresponding to a voxel.
- Grey-scale values are usually mapped to integers $\{0, \dots, 2^p - 1\}$ from white to black (p -bit image). We usually scale it back to, say, $[0, 1]$ from 0 (black) to 1 (white).
 - 2-bit image (binary images) corresponds to two values.
 - 8-bit corresponds to 256 different values.
 - 32-bit image corresponds to 4 294 967 296 different values.

The reconstruction problem

Discretisation – Reformulating the reconstruction problem

- **Accuracy:** for inverse problems, the discretisation must be accurate in both reconstruction and data spaces, *i.e.*,

$$f \approx \sum_{j=1}^n \alpha_j \phi_j \quad \text{and} \quad \mathcal{A}(f) \approx \mathcal{A}\left(\sum_{j=1}^n \alpha_j \phi_j\right)$$

for any $f \in X$ where $\alpha_j := \langle f, \phi_j \rangle_X$.

- **Discretised forward operator:** define $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ as

$$A(\alpha) := \mathcal{S}\left(\mathcal{A}\left(\sum_{j=1}^n \alpha_j \phi_j\right)\right) \quad \text{for } \alpha \in \mathbb{R}^n.$$

- **Linearity & measurement matrix:** assume $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is linear (e.g., both $\mathcal{A}: X \rightarrow Y$ and $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ are linear). There exists an $(m \times n)$ -matrix \mathbf{A} (measurement matrix) so that $A(\alpha) = \mathbf{A} \cdot \alpha$.

The reconstruction problem

Discretisation – Reformulating the reconstruction problem

- **Accuracy:** for inverse problems, the discretisation must be accurate in both reconstruction and data spaces, *i.e.*,

$$f \approx \sum_{j=1}^n \alpha_j \phi_j \quad \text{and} \quad \mathcal{A}(f) \approx \mathcal{A}\left(\sum_{j=1}^n \alpha_j \phi_j\right)$$

for any $f \in X$ where $\alpha_j := \langle f, \phi_j \rangle_X$.

- **Discretised forward operator:** define $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ as

$$A(\alpha) := \mathcal{S}\left(\mathcal{A}\left(\sum_{j=1}^n \alpha_j \phi_j\right)\right) \quad \text{for } \alpha \in \mathbb{R}^n.$$

- **Linearity & measurement matrix:** assume $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is linear (e.g., both $\mathcal{A}: X \rightarrow Y$ and $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ are linear). There exists an $(m \times n)$ -matrix \mathbf{A} (measurement matrix) so that $A(\alpha) = \mathbf{A} \cdot \alpha$.

The reconstruction problem

Discretisation – Reformulating the reconstruction problem

- **Accuracy:** for inverse problems, the discretisation must be accurate in both reconstruction and data spaces, *i.e.*,

$$f \approx \sum_{j=1}^n \alpha_j \phi_j \quad \text{and} \quad \mathcal{A}(f) \approx \mathcal{A}\left(\sum_{j=1}^n \alpha_j \phi_j\right)$$

for any $f \in X$ where $\alpha_j := \langle f, \phi_j \rangle_X$.

- **Discretised forward operator:** define $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ as

$$A(\alpha) := \mathcal{S}\left(\mathcal{A}\left(\sum_{j=1}^n \alpha_j \phi_j\right)\right) \quad \text{for } \alpha \in \mathbb{R}^n.$$

- **Linearity & measurement matrix:** assume $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is linear (e.g., both $\mathcal{A}: X \rightarrow Y$ and $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ are linear). There exists an $(m \times n)$ -matrix \mathbf{A} (measurement matrix) so that $A(\alpha) = \mathbf{A} \cdot \alpha$.

The reconstruction problem

Discretisation – Reformulating the reconstruction problem

- **Elements of the measurement matrix:** when $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is linear, then

$$A(\alpha) = \sum_{j=1}^n \alpha_j \mathcal{S}(\mathcal{A}(\phi_j)).$$

Hence, the measurement matrix \mathbf{A} is given as

$$\mathbf{A} := \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

where

$$a_{i,j} := i\text{:th component of } \mathcal{S}(\mathcal{A}(\phi_j)) = \mathcal{S}(\mathcal{A}(\phi_j))_i \in \mathbb{R}$$

for $i = 1, \dots, m$ and $j = 1, \dots, n$.

Remember: n is number of voxels and m is number of data points.

The reconstruction problem

Different formulations of the reconstruction problem

Original formulation (sampled noisy data)

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

Here, $\mathcal{A}: X \rightarrow Y$ is the forward operator and $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ is the sampling operator.

Fully discretised formulation

Recover the digital image $\alpha_{\text{true}} \in \mathbb{R}^n$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}.$$

Here, \mathbf{A} is the $(m \times n)$ -measurement matrix.

Discretising has allowed us to go from trying to recover $f_{\text{true}} \in X$ (infinite-dimensional object) to trying to recover only its average value α_j in the voxel Ω_j , for $j = 1, \dots, n$.

The reconstruction problem

Different formulations of the reconstruction problem

Original formulation (sampled noisy data)

Recover the image $f_{\text{true}} \in X$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathcal{S}(\mathcal{A}(f_{\text{true}})) + g_{\text{noise}}.$$

Here, $\mathcal{A}: X \rightarrow Y$ is the forward operator and $\mathcal{S}: Y \rightarrow \mathbb{R}^m$ is the sampling operator.

Fully discretised formulation

Recover the digital image $\alpha_{\text{true}} \in \mathbb{R}^n$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}.$$

Here, \mathbf{A} is the $(m \times n)$ -measurement matrix.

Discretising has allowed us to go from trying to recover $f_{\text{true}} \in X$ (infinite-dimensional object) to trying to recover only its average value α_j in the voxel Ω_j , for $j = 1, \dots, n$.

The reconstruction problem

Tomography

- **The data space:** Y is an infinite dimensional vector space of real-valued functions defined on a manifold M of lines in \mathbb{R}^3 , so continuum data $g \in Y$ is a function

$$g: M \rightarrow \mathbb{R}.$$

- **The manifold of lines M :** a line ℓ in \mathbb{R}^3 is uniquely determined by its direction ω (directional vector) and a point $x \in \omega^\perp$ that lies on ℓ (ω^\perp is the 2D plane orthogonal to ω):

$$\ell: t \mapsto x + t\omega.$$

Hence, the pair (ω, x) corresponds to a unique line ℓ and vice versa, so M can be considered as a vector space of such pairs.

Note: each $x \in \omega^\perp$ corresponds to a unique point on the detector surface.

- **Data acquisition geometry:** the arrangement of the m lines in M that correspond to the measurements, i.e., the sampling of ω and $x \in \omega^\perp$.

The reconstruction problem

Tomography

- **The data space:** Y is an infinite dimensional vector space of real-valued functions defined on a manifold M of lines in \mathbb{R}^3 , so continuum data $g \in Y$ is a function

$$g: M \rightarrow \mathbb{R}.$$

- **The manifold of lines M :** a line ℓ in \mathbb{R}^3 is uniquely determined by its direction ω (directional vector) and a point $x \in \omega^\perp$ that lies on ℓ (ω^\perp is the 2D plane orthogonal to ω):

$$\ell: t \mapsto x + t\omega.$$

Hence, the pair (ω, x) corresponds to a unique line ℓ and vice versa, so M can be considered as a vector space of such pairs.

Note: each $x \in \omega^\perp$ corresponds to a unique point on the detector surface.

- **Data acquisition geometry:** the arrangement of the m lines in M that correspond to the measurements, i.e., the sampling of ω and $x \in \omega^\perp$.

The reconstruction problem

Tomography

- **The data space:** Y is an infinite dimensional vector space of real-valued functions defined on a manifold M of lines in \mathbb{R}^3 , so continuum data $g \in Y$ is a function

$$g: M \rightarrow \mathbb{R}.$$

- **The manifold of lines M :** a line ℓ in \mathbb{R}^3 is uniquely determined by its direction ω (directional vector) and a point $x \in \omega^\perp$ that lies on ℓ (ω^\perp is the 2D plane orthogonal to ω):

$$\ell: t \mapsto x + t\omega.$$

Hence, the pair (ω, x) corresponds to a unique line ℓ and vice versa, so M can be considered as a vector space of such pairs.

Note: each $x \in \omega^\perp$ corresponds to a unique point on the detector surface.

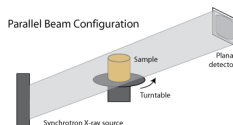
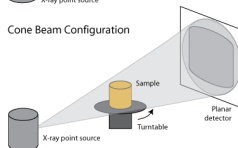
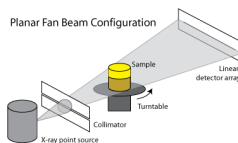
- **Data acquisition geometry:** the arrangement of the m lines in M that correspond to the measurements, i.e., the sampling of ω and $x \in \omega^\perp$.

The reconstruction problem

Tomography

Common data acquisition geometries

- Parallel beam tomography
- Fan-beam
- 3D cone-beam tomography
- 3D helical/spiral tomography

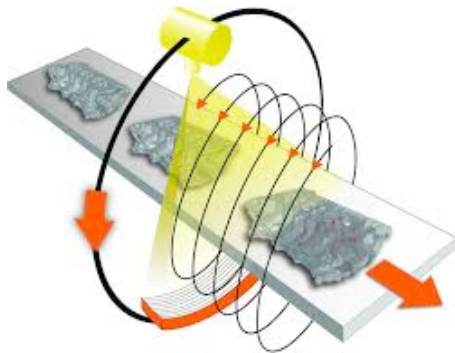


The reconstruction problem

Tomography

Common data acquisition geometries

- Parallel beam tomography
- Fan-beam
- 3D cone-beam tomography
- 3D helical/spiral tomography



The reconstruction problem

Tomography

- **Transmission tomography:** forward operator given by the ray transform:

$$\mathcal{A}(f)(\omega, x) = \int_{-\infty}^{\infty} f(x + t\omega) dt \quad \text{for a line } (\omega, x) \in M.$$

Note that $g := \mathcal{A}(f)$ is a real-valued function defined on M a set of lines in \mathbb{R}^3 .

- **Exact reconstruction in 2D:** write x and ω in polar coordinates:

$$x = \begin{pmatrix} t \cos \theta \\ t \sin \theta \end{pmatrix}$$
$$\omega = \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}$$

Filtered backprojection

Ingredients

- Forward operator in polar coordinates:

$$\mathcal{A}(f)(t, \theta) = \int_{-\infty}^{\infty} f \left[t \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} + s \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix} \right] ds$$

for $t \in \mathbb{R}$ and $0 < \theta < \pi$

- Backprojection = average over all lines through one point:

$$\mathcal{B}(g)(x, y) = \frac{1}{\pi} \int_0^\pi g(x \cos \theta + y \sin \theta, \theta) d\theta.$$

The backprojection takes a sinogram (=image on M) and maps it back to an image in \mathbb{R}^2 . But it is **not** the inverse mapping of \mathcal{A} .

Filtered backprojection

Ingredients

- **Fourier transform:** we transform only with respect to the first variable

$$\mathcal{F}(g)(t, \theta) = \int_{-\infty}^{\infty} g(s, \theta) \exp(-ist) \, ds$$

$$\mathcal{F}^{-1}(g)(t, \theta) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g(s, \theta) \exp(+ist) \, ds$$

Each of these transformations maps a function on M to another function on M , and \mathcal{F}^{-1} is the inverse of \mathcal{F} . There exist fast implementations for both CPU (e.g. (py)FFTW) and GPU (cuFFT).

Filtered backprojection

The formula

$$f(x, y) = \frac{1}{2} \mathcal{B}(\mathcal{F}^{-1}(|t| \mathcal{F}(\mathcal{A}f)(t, \theta)))(x, y)$$

Step by step

- 1 Start with the sinogram $\mathcal{A}f$, which you obtain from the scanner.
- 2 Calculate the Fourier transform $\mathcal{F}(\mathcal{A}f)$.
- 3 Multiply by the absolute value of the first (radial) component. (This is the filtering).
- 4 Calculate the inverse Fourier transform. You still have a sinogram.
- 5 Apply the (nonfiltered) backprojection. Here you obtain an image.
- 6 Divide by 2.

Filtered backprojection

Challenges

- Requires full knowledge of all line integrals to give an exact reconstruction.
- The multiplication with the absolute value amplifies errors, in particular high frequencies. Solution: use a different filter, which does not grow so rapidly.
- But it is fast in comparison with iterative and variational methods.

The reconstruction problem

Tomography

- **Sampling operator:** given m data sampling points $(\omega_1, x_1), \dots, (\omega_m, x_m)$ (which are lines) that correspond to actual measured data, define

$$\mathcal{S}(g) = (g(\omega_1, x_1), \dots, g(\omega_m, x_m)) \in \mathbb{R}^m \quad \text{for } g \in Y.$$

In particular, the i :th measurement corresponding to the i :th line (ω_i, x_i) is modeled by

$$\mathcal{S}(\mathcal{A}(f))_i = \mathcal{A}(f)(\omega_i, x_i) = \int_{-\infty}^{\infty} f(x_i + t\omega_i) dt.$$

The reconstruction problem

Tomography

- **Measurement matrix:** fix a basis $\{\phi_i\}_i$ of X and a data acquisition geometry $(\omega_1, x_1), \dots, (\omega_m, x_m)$. Then, the elements in the $(m \times n)$ -measurement matrix

$$\mathbf{A} := \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

are given as

$$a_{i,j} = \mathcal{A}(\phi_i)(\omega_j, x_j) = \int_{-\infty}^{\infty} \phi_i(x_j + t\omega_j) dt$$

for $i = 1, \dots, n$ and $j = 1, \dots, m$.

- **Voxel basis:** the ray transform assumes the x-ray beam is one-dimensional (pencil beam), so

$$a_{i,j} = \text{length of part of } i\text{:th line in } j\text{:th voxel.}$$

The reconstruction problem

Tomography

- **Voxel basis and x-ray with 2D-dimensional cross section:** the lines are replaced by 3D strips, so

$a_{i,j}$ = volume of part of i :th strip in j :th voxel.

More complicated beam profiles: can be included by weighting different parts of the strips differently. The calculation of $a_{i,j}$ requires a lot of work and much of the literature on iterative methods discuss effects of using various schemes to approximate the measurement matrix.

Historical note: in the early approaches for 2D tomography by Hounsfield, one used

$$a_{i,j} = \begin{cases} 1 & \text{if center of } j\text{:th pixel is in } i\text{:th strip,} \\ 0 & \text{otherwise.} \end{cases}$$

Then, values of $a_{i,j}$ can be computed at run time even with slow computers and do not have to be stored.

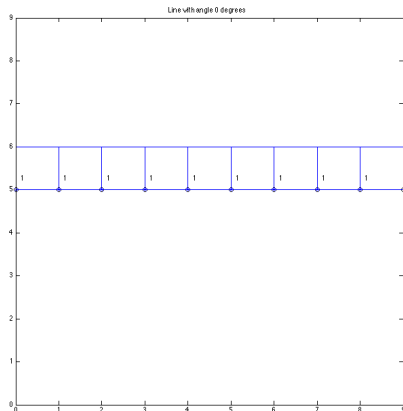
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 0° with
horizontal axis



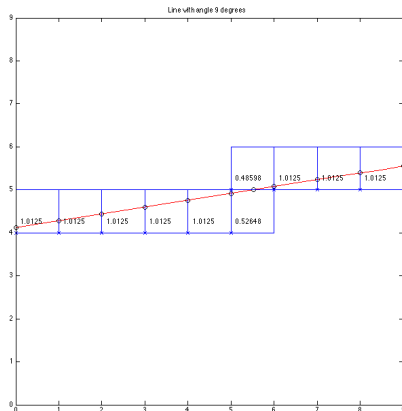
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 9° with
horizontal axis



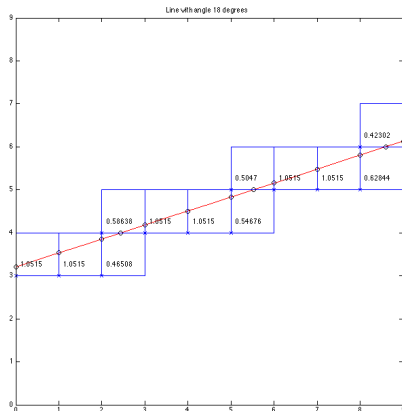
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 18° with
horizontal axis



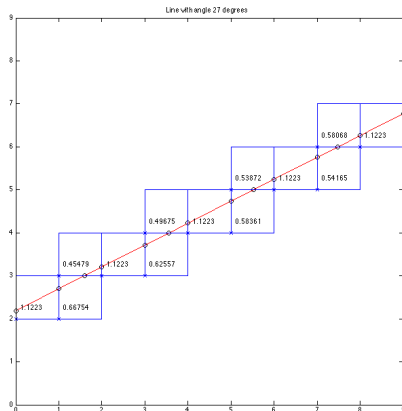
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size = 1.

Angle = 27° with horizontal axis



Tomography: computing the measurement matrix for lines in 2D

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size = 1.

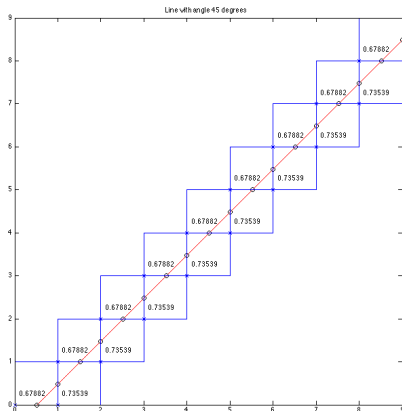
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 45° with horizontal axis



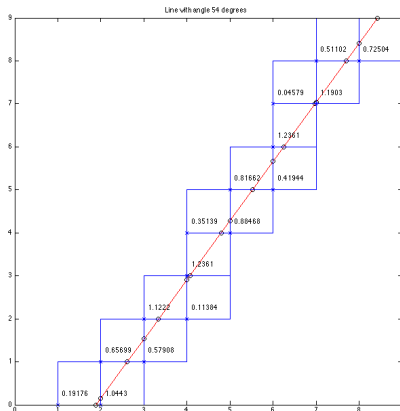
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 54° with horizontal axis



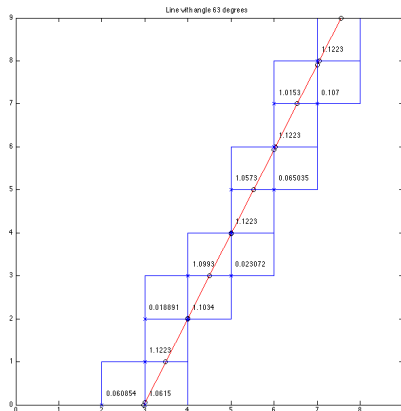
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size = 1.

Angle = 63° with
horizontal axis



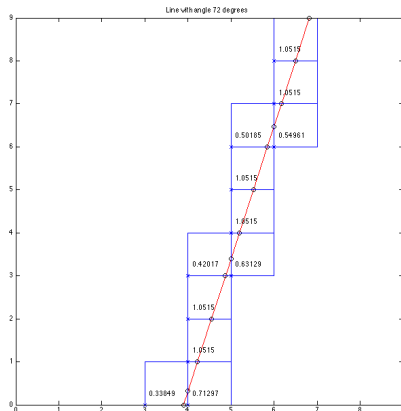
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 72° with horizontal axis



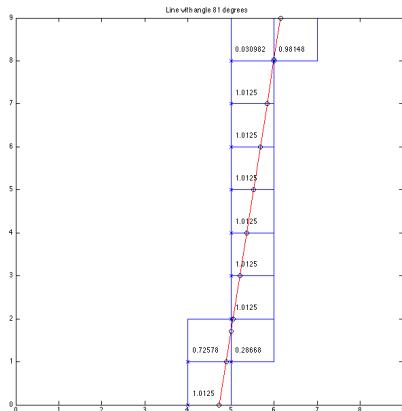
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 81° with
horizontal axis



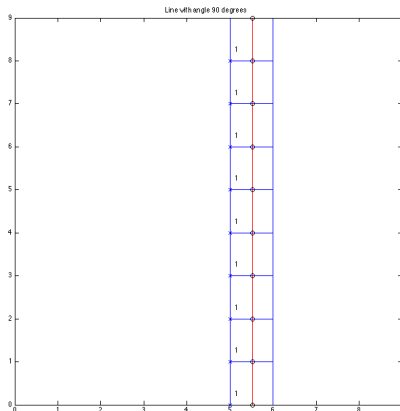
The reconstruction problem

Tomography: computing the measurement matrix for lines in 2D

Matrix element $a_{i,j}$ is the length of i :th line in j :th pixel.

Example: calculating i :th row a_i for a line through a 9×9 pixel 2D grid where pixel size =1.

Angle = 90° with
horizontal axis

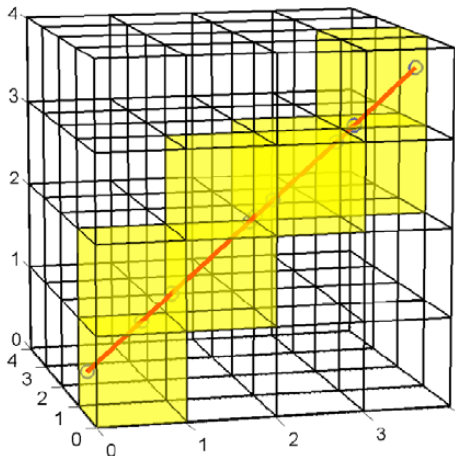


The reconstruction problem

Tomography: computing the measurement matrix for lines in 3D

Matrix element $a_{i,j}$ is the length of i :th line in j :th voxel.

Calculate the intersection of i :th line ℓ_i with j :th voxel.



The reconstruction problem

Tomography: computing the measurement matrix for lines in 3D

Matrix element $a_{i,j}$ is the length of i :th line in j :th voxel.

Calculate the intersection of i :th line ℓ_i with j :th voxel.

- **Intersection with x_1 -planes:** consider the intersection point in \mathbb{R}^3 between a line ℓ and a hyperplane parallel to the x_1 -axis:

$$(x_1, x_2, x_3) \mapsto (p, x_2, x_3) \quad \text{where } p \text{ is fixed.}$$

Assume ℓ is given by (ω, x^0) , i.e.,

$$\ell_i : t \mapsto x^0 + t\omega.$$

Let $x^* = (x_1^*, x_2^*, x_3^*)$ denote this intersection, so $x_1^* = p$ by definition and

$$x_2^* = x_2^0 + \frac{p - x_1^0}{\omega_1} \omega_2 \quad \text{and} \quad x_3^* = x_3^0 + \frac{p - x_1^0}{\omega_1} \omega_3.$$

The reconstruction problem

Tomography: computing the measurement matrix for lines in 3D

Matrix element $a_{i,j}$ is the length of i :th line in j :th voxel.

Calculate the intersection of i :th line ℓ_i with j :th voxel.

- **Intersection with x_2 - and x_3 -planes:** similar equations for the planes $x_2 = q$ and $x_3 = r$.
- From these intersections it is easy to compute the ray length in voxel j and there are fast methods for these computations:
 - R. L. Siddon. *Fast calculation of the exact radiological path for a 3-dimensional CT array*. Medical Physics, vol. 12, no. 2, pages 252–255, 1985.
 - H. Gao. *Fast parallel algorithms for the X-ray transform and its adjoint*. Medical Physics, vol. 39, no. 11, pages 7110–7120, 2012.

The reconstruction problem

Tomography: properties of the measurement matrix

- For most tomography problems, the measurement matrix **A** is very large and it cannot be stored in computer memory.

Problem size in a modern 64-slice helical CT:

- a single 2D cross-section
 - about 1 000 detector elements arranged along 64 (detector) slices, each roughly corresponding to a 2D slice through the object. Radiation source attains 800 different positions each second (two rotations/second)
 $\implies 1000 \cdot 800 \approx 800\,000$ data points.
 - 2D cross-section is 512×512 pixels in size $\implies 260\,000$ pixels.

Total: 260 000 unknowns and 800 000 equations.

- a single 2D cross-section corresponds to a 1 mm thick slice in 3D, so 1 000 cross sections needed to cover 1 m:
 $\implies n \sim 260 \cdot 10^6$ and $m \sim 800 \cdot 10^6$.
- **A** is sparse since each ray only intersects a few voxels.

The reconstruction problem

Tomography: properties of the measurement matrix

- For most tomography problems, the measurement matrix **A** is very large and it cannot be stored in computer memory.

Problem size in a modern 64-slice helical CT:

- a single 2D cross-section
 - about 1 000 detector elements arranged along 64 (detector) slices, each roughly corresponding to a 2D slice through the object. Radiation source attains 800 different positions each second (two rotations/second)
 $\implies 1000 \cdot 800 \approx 800\,000$ data points.
 - 2D cross-section is 512×512 pixels in size $\implies 260\,000$ pixels.

Total: 260 000 unknowns and 800 000 equations.

- a single 2D cross-section corresponds to a 1 mm thick slice in 3D, so 1 000 cross sections needed to cover 1 m:
 $\implies n \sim 260 \cdot 10^6$ and $m \sim 800 \cdot 10^6$.
- **A** is sparse since each ray only intersects a few voxels.

The reconstruction problem

Naive inversion and concept of ill-posedness

Recover the digital image $\alpha_{\text{true}} \in \mathbb{R}^n$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}. \quad (1)$$

Here, \mathbf{A} is the $(m \times n)$ -measurement matrix.

Existence: naive inversion $\alpha_{\text{naive}} := \mathbf{A}^{-1} \cdot g$ is not possible when \mathbf{A} is not invertible (possibly no solutions to (1)).

Generalised solution: relax the notion of a solution to (1) to enforce existence. Define a data error function $E: \mathbb{R}^m \rightarrow \mathbb{R}_+$ and look for a vector $\alpha^* \in \mathbb{R}^n$ that solves

$$\min_{\alpha} E(\mathbf{A} \cdot \alpha - g).$$

The reconstruction problem

Naive inversion and concept of ill-posedness

Recover the digital image $\alpha_{\text{true}} \in \mathbb{R}^n$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}. \quad (1)$$

Here, \mathbf{A} is the $(m \times n)$ -measurement matrix.

- **Least-squares solutions:** choose E as the 2-norm of the residual, *i.e.*, solve

$$\min_{\alpha} \|\mathbf{A} \cdot \alpha - g\|_2^2.$$

- **Maximum likelihood solution:** choosing E as the neglog of the data likelihood gives maximum likelihood solutions to (1).

The reconstruction problem

Naive inversion and concept of ill-posedness

Recover the digital image $\alpha_{\text{true}} \in \mathbb{R}^n$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}. \quad (1)$$

Here, \mathbf{A} is the $(m \times n)$ -measurement matrix.

Ill-posedness: generalised solutions handle the issue of existence (there is always a generalised solution to (1)). Two serious problems remain:

- **Non-uniqueness:** infinitely many generalised solutions.
- **Instability:** generalised solutions do not depend continuously on the data g .

The inverse problem (1) is **ill-posed** if any of these issues occurs.

The reconstruction problem

Naive inversion and concept of ill-posedness

Recover the digital image $\alpha_{\text{true}} \in \mathbb{R}^n$ from measured data $g \in \mathbb{R}^m$ assuming

$$g = \mathbf{A} \cdot \alpha_{\text{true}} + g_{\text{noise}}. \quad (1)$$

Here, \mathbf{A} is the $(m \times n)$ -measurement matrix.

Condition number and ill-posedness:

assuming that A is invertible ($n = m$), what happens if we try $\alpha_{\text{naive}} := \mathbf{A}^{-1} \cdot g$.

Classical perturbation theory leads to the bound

$$\frac{\|\alpha_{\text{true}} - \alpha_{\text{naive}}\|_2}{\|\alpha_{\text{true}}\|_2} \leq \text{cond}(\mathbf{A}) \frac{\|g - g_{\text{exact}}\|_2}{\|g_{\text{exact}}\|_2}$$

where $g_{\text{exact}} := \mathbf{A} \cdot \alpha_{\text{true}}$ and $\text{cond}(\mathbf{A})$ is the condition number (ratio of the maximal and minimal singular values).

$\text{cond}(\mathbf{A})$ large \implies naive inversion not useful since α_{naive} can be very far from α_{true} .

The reconstruction problem

Regularization

Regularisation: mathematical theory for handling ill-posed problems. Stability and uniqueness is enforced by accounting for **a priori knowledge** about the true (unknown) image.

The main idea of regularisation

Replace the original ill-posed reconstruction problem by a well-posed reconstruction problem (*i.e.*, it has a unique solution that depends continuously on the data) that is convergent as the data error goes to zero.

Well-posedness: guarantees stability

Convergence: the reconstructions obtained converge to a least squares solution when the data error approaches zero and the parameters in the reconstruction method (regularisation parameters) are chosen appropriately.

The reconstruction problem

Regularization

Regularisation: mathematical theory for handling ill-posed problems. Stability and uniqueness is enforced by accounting for **a priori knowledge** about the true (unknown) image.

The main idea of regularisation

Replace the original ill-posed reconstruction problem by a well-posed reconstruction problem (*i.e.*, it has a unique solution that depends continuously on the data) that is convergent as the data error goes to zero.

Well-posedness: guarantees stability

Convergence: the reconstructions obtained converge to a least squares solution when the data error approaches zero and the parameters in the reconstruction method (regularisation parameters) are chosen appropriately.

The reconstruction problem

Regularization

Regularisation: mathematical theory for handling ill-posed problems. Stability and uniqueness is enforced by accounting for **a priori knowledge** about the true (unknown) image.

The main idea of regularisation

Replace the original ill-posed reconstruction problem by a **well-posed** reconstruction problem (*i.e.*, it has a unique solution that depends continuously on the data) that is **convergent** as the data error goes to zero.

Well-posedness: guarantees stability

Convergence: the reconstructions obtained converge to a least squares solution when the data error approaches zero and the parameters in the reconstruction method (regularisation parameters) are chosen appropriately.