Università degli Studi di Ferrara

DOTTORATO DI RICERCA IN
Matematica e Informatica

CICLO XVIII

COORDINATORE Prof. Massimiliano Mella

Region-of-interest CT problem: effective forward operator and iterative shearlet-based regularization

Settore Scientifico Disciplinare MAT/08

Dottorando
Dott. Bubba Tatiana Alessandra

Tutore
Prof. Zanghirati Gaetano

Anni 2013-2015

Abstract

Region-of-interest Computed Tomography (ROI CT) is an X-ray based incomplete data imaging acquisition mode. Currently, it is among the “hot topics” in the field of X-ray tomographic imaging, due to its potential to lower exposure to X-ray radiation and to reduce the scanning time. This is particularly appealing, especially for the medical imaging community, due to a wide range of applications in biomedical imaging, such as contrast-enhanced cardiac imaging or positioning of intracranial stents.

However, this problem is rather challenging because of the truncation of projections, *i.e.*, of the acquired data. CT is in general an ill-posed problem, and the ill-posedness becomes severe due to incompleteness of the ROI CT dataset, even more as the ROI size gets smaller. Therefore, one of the major issues is that traditional or naive local reconstruction algorithms may be very unstable and might produce unreliable reconstructions, when applied directly to incomplete data, setting missing projections to zero. So far, both *ad hoc* analytic formulae and iterative numerical schemes have been proposed in the literature to address this problem, but typically rely on restrictive assumptions.

This thesis addresses the connection of the reconstruction problem and the incompleteness of the ROI CT data. The main goal is to obtain a stable and feasible reconstruction, possibly under reliable noise circumstances, and without any assumptions either on the size or on the location of the ROI. This would be impossible due to non-uniqueness of the interior problem.

To this end, we formulate the ROI CT reconstruction problem as a convex optimization problem with different regularization levels. A regularized functional based on *shearlets*, a new multiscale method whose main features are relevant in CT-like applications, is possibly coupled with a Total Variation regularization term. Both a smooth and a nonsmooth version of this convex problem are investigated.

An accurate and flexible forward projection technique, called *distance-driven*, is considered to study the problem, and an efficient implementation is developed to provide a faster digital reconstruction framework. This original contribution goes beyond the ROI CT problem and applies to more general CT problems.

Two iterative approaches are proposed and analyzed to face the numerical solution of the derived convex optimization problem. A scaled gradient projection method for the smooth approach and a variable metric inexact line-search algorithm for the nonsmooth case. Both methods have been proposed very recently, and, to the best of our knowledge, it is the first time that their performance is investigated in CT-like applications.

All experimental studies presented make use of simulated data, in the case of 2D fan-beam CT. The numerical tests illustrated in this thesis show that our approach is insensitive to the location of the ROI and remains very stable also when the ROI size is rather small.

The findings and conclusions of this work have a number of important implications for future research. Therefore, suggestions for further work will be given for each addressed topic.

Sommario

La tecnica di Tomografia Computerizzata ristretta a regioni-di-interesse (ROI CT) rientra tra le modalità di acquisizione di immagini tomografiche, mediante raggi X, da dati incompleti. Attualmente, è tra i "temi caldi" nel campo dell'imaging tomografico, poiché offre al contempo la possibiltà di diminuire l'esposizione a radiazioni derivanti dai raggi X e ridurre il tempo di scansione. Per la comunità medica, ROI CT riveste un ruolo di particolare interesse, grazie al gran numero di applicazioni in *imaging* biomedico, tra cui l'*imaging* cardiaco con intensificazione di contrasto e il posizionamento di stent intracranici.

Tuttavia, si tratta di un problema piuttosto difficile a causa del troncamento delle proiezioni, cioè dei dati acquisiti. CT è in generale un problema mal posto e, a causa dell'incompletezza dei dati, la mal posizione tende a peggiorare, sempre più con il diminuire della dimensione della regione-di-interesse. Perciò, uno dei principali problemi è che gli algoritmi classici o ricostruzioni locali *naive* possono rivelarsi estremamente instabili, restituendo ricostruzioni inaffidabili, quando applicati ai dati incompleti, ponendo a zero le proiezioni mancanti. A oggi, per affrontare questo problema, sono stati proposti in letteratura sia formule analitiche *ad hoc* sia schemi numerici iterativi, ma tipicamente si basano su ipotesi restrittive.

Questa tesi si propone di investigare la connessione tra il problema di ricostruzione e l'incompletezza dei dati derivanti dal problema ROI CT. L'obiettivo principale è di ottenere una ricostruzione stabile e ammissibile, possibilmente sotto ipotesi realistiche per il rumore e senza nessun tipo di ipotesi sulla dimensione o posizione della ROI. Ciò sarebbe impossibile a causa della non-unicità dell'*interior problem*.

A tal fine, ROI CT viene formulato come problema convesso di ottimizzazione, con diversi livelli di regolarizzazione. Si considera un funzionale di regolarizzazione basato sulle *shearlets*, un metodo multiscala introdotto di recente le cui caratteristiche sono rilevanti in applicazioni tomografiche, eventualmente combinato con un termine di tipo Variazione Totale. Di questo problema convesso vengono investigate sia una versione differenziabile sia una non differenziabile.

Per lo studio del problema, viene considerata una tecnica accurata e flessibile per la proiezione in avanti, chiamata *distance-driven*, per la quale è stata sviluppata un'implementazione efficiente in grado di fornire un ambiente di ricostruzione più veloce. Si tratta di un contributo originale che va al di là del solo problema ROI CT e si applica in generale a tutti i problemi di tipo CT.

Per la soluzione numerica del problema convesso di ottimizzazione vengono proposti e analizzati due approcci iterativi. Per la formulazione differenziabile viene considerato il metodo del gradiente scalato proiettato, mentre per la versione non differenziabile si considera l'algoritmo a metrica variabile inesatta con ricerca in linea. Entrambi i metodi sono stati proposti molto recentemente e, per quanto si è a conoscenza, è la prima volta che queste tecniche vengono investigate in applicazioni di tipo CT.

Tutti gli studi sperimentali che vengono presentati fanno uso di dati simulati nel caso della geometria 2D *a ventaglio*. I test numerici che vengono illustrati nella tesi mostrano che l'approccio presentato non è influenzato dalla posizione della ROI e rimane molto stabile anche quando la dimensione della ROI è piuttosto piccola.

I risultati e le conclusione di questo lavoro hanno importanti implicazioni per le ricerche future: per ciascun argomento trattato verranno dati spunti per il lavoro futuro.

Acronyms

1D	1-dimensional
2D	2-dimensional
3D	3-dimensional
ART	algebraic reconstruction technique
BB	Barzilai-Borwein (steplength updating rules)
CBGP	cyclic scaled gradient projection
CPU	central processing unit
CT	computed tomography
DM	data mismatch
DBP	differentiated back-projection
EM	expectation maximization
ES	early stopping
FBP	filtered back-projection
FDK	Feldkamp, Davis, and Kress algorithm
FOV	field of view
GB	gigabyte
GHz	gigahertz
GPU	graphics processing unit
IRR	iterative reconstruction-reprojection
LS	least square
LSCG	LS conjugate gradient
MAP	maximum a posteriori
MART	multiplicative ART
MBIR	model-based iterative reconstruction
ML	maximum likelihood
ML-EM	maximum likelihood expectation-maximization
MRA	multiresolution analysis
MRI	magnetic resonance imaging
NDT	nondestructive material testing
OS	ordered subsets
PET	positron emission tomography

PSNR	peak-signal-to-noise ratio
ROI	region-of-interest
ROI CT	region-of-interest tomography
SART	simultaneous ART
SIRT	simultaneous iterative reconstruction technique
SGP	scaled gradient projection (method)
SPECT	single photon emission computed tomography
SVD	singular values decomposition
TB	terabyte
TV	total variation
VMILA	variable metric inexact line-search algorithm

Symbols

$\otimes$	Kronecker product
$\langle \cdot, \cdot \rangle$	inner product in $\mathbb{R}^n$
$\lceil \cdot \rceil$	ceil function
$\lfloor \cdot \rfloor$	floor function
$\Vert \cdot \Vert_p$	p-norm in L^p
$\Vert \cdot \Vert_D$	norm induced by the symmetric positive definite matrix D
$\hat{g}$	Fourier transform of the function g
$\mathbf{0}_n$	n-dimensional zero vector
$\mathbb{O}_n$	n-dimensional zero matrix
$\mathbf{1}_n$	n-dimensional all-ones vector
$1_{\mathcal{A}}$	characteristic function of the set $\mathcal{A}$
$\mathbb{1}_n$	n-dimensional identity matrix
$\mathbb{R}^+$	set of positive real number (not including zero)
$\mathbb{R}_-$	set of negative real number (including zero)
$\mathbb{R}_0^+$	set of positive real number (including zero)
$\mathbb{R}^*$	set of real number (not including zero)
$\mathbb{Z}^n$	n-dimensional integer space
$\mathbb{R}^n$	n-dimensional euclidean space
$\mathbb{C}^n$	n-dimensional complex space
a	scaling parameter (unless otherwise specified)
$\boldsymbol{a} = (a_u, a_v)$	detector offset
A	generic point on the object – Chapter 3
$\mathcal{A}$	set of rotation angles
A_a	scaling matrix depending on the scaling parameter a
$\mathbb{A}_n$	n-dimensional full affine group of motions
α	steplength (unless otherwise specified)
α_k	steplength at iteration k
b	pixels or voxels basis function for object discretization
b_i	i-th (generic) basis function for object discretization
$B_2(0, r)$	2D euclidean ball centered in 0 with radius r
$B_\infty(0, r)$	infinity-norm centered in 0 with radius r

$\mathbf{B}$	linear operator for proximal inexact computation (VMILA)
β	elevation angle of a cone-beam system– Chapter 3
β	backtracking parameter – Chapters 5, 6
c_x, c_y, c_1	flags in distance-driven vectorized approach
$\boldsymbol{c}(\boldsymbol{n})$	vector of the center coordinates of the object voxel $\boldsymbol{n}$
C	isocenter – Chapter 3
$\mathcal{C}$	(generic) convex set
$\mathcal{C}^n$	unit cylinder in $\mathbb{R}^n$
$\mathscr{C}$	centered cube of the shearlets 3D frequency domain partition
$\mathscr{C}_i$	i-th cone of the shearlets 2D frequency domain partition
$\mathscr{C}^n(\cdot)$	class of functions differentiable with continuity up to order n
$\mathscr{C}^\infty(\cdot)$	class of infinitely differentiable functions
$\mathscr{C}_0^\infty(\cdot)$	$\mathscr{C}^\infty$ functions with bounded support
γ	aperture angle of a cone-beam system – Chapter 3
γ	backtracking parameter – Chapters 5, 6
Γ	(generic) objective function
Γ_0	smooth part of an objective function Γ
Γ_1	nonsmooth part of an objective function Γ
d_{AB}	distance $\\|B - A\\|_2$ between the points A and B
$d_S(\cdot)$	distance from the source – Chapter 3
$d_\sigma(\cdot)$	distance-like function
$\mathbf{d}$	descent direction
$\mathbf{d}^{(k)}$	descent direction at iteration k
D	scaling matrix in SGP and VMILA
D_k	scaling matrix at iteration k
D_M	dilation operator associated to the matrix M
$\mathcal{D}_L$	compact set of the symmetric positive definite matrices D with threshold L
$\mathscr{D}$	(discrete) derivative approximation for TV definition
δ	TV smooth parameter
$\delta(\cdot)$	Dirac δ function
$\boldsymbol{\delta}$	grid increment vector for detector grid
Δ	grid increment for "equilateral" pixel or voxel grid
$\boldsymbol{\Delta}$	grid increment vector for object grid
∇	discrete gradient operator
∇_i	i-block of the discrete gradient operator
$\boldsymbol{e}_n$	n-th versor of the canonical basis
$E\{\mathbf{Y}\}$	expected value of the random variable $\mathbf{Y}$

$\mathscr{E}$	cartoon-like images
η	parameter of η-approximation (VMILA)
θ	angle associated to the direction of the ray, *i.e.*, rotation angle
Θ	(generic) low pass operator
$\boldsymbol{\zeta}^{\mathrm{min}}, \boldsymbol{\zeta}^{\mathrm{max}}$	lowermost and the uppermost z coordinates of the projections of the endpoints of all the vertical edges of the voxels midway slices
f	object (attenuation or density) function
$\bar{f}$	ML or MAP reconstruction
f_J	best J-term approximation (shearlets)
$\mathbf{f}$	column vector representing the object to reconstruct (discretization of the attenuation or density function f)
$\bar{\mathbf{f}}$	ML or MAP estimate
$\mathbf{f}^*$	ML or MAP solution
$\mathbf{f}^{(k)}$	ML or MAP estimate at iteration k
F	analysis operator
F^*	synthesis operator
ϕ	scaling function
φ_i	element of a frame sequence
$\boldsymbol{\Phi}$	shearlet matrix in ROI CT objective function formulation
$\boldsymbol{\Phi}^T$	transpose of the matrix $\boldsymbol{\Phi}$
ψ	mother wavelet or shearlet
$\psi_{a,t}$	element of a 1D continuous wavelet system
$\psi_{j,m}$	element of a 1D discrete wavelet system
$\psi_{M,t}$	element of a nD wavelet system
ψ_{LM}	Lemariè-Meyer wavelet
$\psi_{a,s,t}$	element of a 2D shearlet system
$\psi_{j,k,m}$	element of a 2D discrete shearlet system
$\psi_{M_{a,s},t}$	element of a nD shearlet system
g	(generic) measured data
g_σ	(generic) noisy measured data
g^*	conjugate function of a (generic) function g
$\boldsymbol{g}_i^{\mathrm{LL}}, \boldsymbol{g}_i^{\mathrm{RL}}, \boldsymbol{g}_i^{\mathrm{LR}}, \boldsymbol{g}_i^{\mathrm{RR}}$	column vectors containing all the overlap differences
$h_\sigma(\cdot)$	function that defines the primal variable in VMILA
$\boldsymbol{h}^{\mathrm{H},i}$	column vector containing all the horizontal overlaps
$\boldsymbol{h}^{\mathrm{V},j}$	column vector containing all the vertical overlaps
$\boldsymbol{h}^{\mathrm{H},i,1}, \boldsymbol{h}^{\mathrm{H},i,2}, \boldsymbol{h}^{\mathrm{H},i,3}$	column vector containing the horizontal overlaps corresponding to $\boldsymbol{p}^{\mathrm{H},i,1}, \boldsymbol{p}^{\mathrm{H},i,2}, \boldsymbol{p}^{\mathrm{H},i,3}$, respectively
H_{obj}	scaling matrix for object reference system

H_{dtc}	scaling matrix for detector reference system
$H_\sigma(\cdot)$	dual formulation corresponding to $h_\sigma(\cdot)$ (VMILA)
$\mathcal{H}$	(generic) Hilbert space
$i_{\mathrm{start}}, i_{\mathrm{inc}}$	integers for handling "reversed ordering" in distance-driven vectorized approach
I	identity operator (unless otherwise specified)
$\mathcal{I}$	(generic) indices set
$\mathcal{I}_{xz}$	set of rotation angles for which $\pi_{\mathrm{prj}} = Oxz$
$\mathcal{I}_{yz}$	set of rotation angles for which $\pi_{\mathrm{prj}} = Oyz$
$\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl}}$	indices set for horizontal overlap computation
$\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl,btm}}$	elements of $\boldsymbol{p}^{\mathrm{H},i}$ identifying the voxels slices vertices projections, belonging to the lowest level
$\mathcal{I}^{\mathrm{H},i,1}_{\mathrm{vxl}}, \mathcal{I}^{\mathrm{H},i,2}_{\mathrm{vxl}}, \mathcal{I}^{\mathrm{H},i,3}_{\mathrm{vxl}}$	indices set for overlap computation, subsets of $\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl}}$
$\mathcal{I}^{\mathrm{prj}}_{\mathrm{vxl}}$	indices set of all voxel projections
$\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{btm},s}$	indices of rows in the s-th column of $Z_{\mathrm{H},i}$ whose corresponding z coordinate of midway slices voxel projection is greater than the z coordinate of the projection of the detector bottom edge
$\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{top},s}$	indices of rows in the s-th column of $Z_{\mathrm{H},i}$ whose corresponding z coordinate of midway slices voxel projection is not greater than the z coordinate of the projection of the detector top edge
$\mathcal{I}^\alpha f$	Riesz potential
$\iota_{\mathcal{A}}$	indicator function of the set $\mathcal{A}$
k	index for third dimension – Chapter 3
k	discrete (sampling for the) shearing parameter – Chapter 4
k	iteration – Chapters 5, 6
κ	memory for the nonmonotone line search
K	(generic) linear continuous operator between Hilbert spaces
$\mathbf{K}$	system matrix, *i.e.*, discretization of an operator K
l	dual iteration – Chapters 5, 6
ℓ	angle index – Chapter 3
$\ell(\theta,\tau)$	ray at angle θ and with signed distance τ
L	detector reference point – Chapter 3
L	threshold for the set $\mathcal{D}_L$ – Chapters 5, 6
$L^{\mathbf{Y}}_{\mathbf{y}}(\cdot)$	likelihood function
$L_{\max}$	maximum pixel value
$L^1(\mathbb{R}^n)$	space of Lebesgue integrable functions on $\mathbb{R}^n$
$L^2(\mathbb{R}^n)$	Hilbert space of square Lebesgue integrable functions on $\mathbb{R}^n$

$\boldsymbol{\lambda}^{\mathrm{min}}, \boldsymbol{\lambda}^{\mathrm{max}}$	lowermost and the uppermost z coordinates of the projections of the detector cells vertical edges endpoints
Λ	discrete sampling set for irregular shearlet systems
$\overline{\boldsymbol{m}} = (m_u, m_v)$	number of detector discretization pixels
M	mask identifying the ROI
$\mathbf{M}$	matrix discretization of the mask M
μ	regularization parameter
μ_k	regularization parameter at iteration k
μ_{opt}	optimal regularization parameter
n	$n \geq 1$ dimension of the space (unless otherwise specified)
$\overline{\boldsymbol{n}} = (n_x, n_y, n_z)$	number of object discretization voxels
N	width and height in pixels of the object to reconstruct
N_θ	number of projection angles
N_{dtc}	number of detector cells
$N_{\mathrm{vxl}}^{\mathrm{prj}}$	total number of voxels slices vertices to be projected
$N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}}$	total number of voxels slices vertices to be projected per layer
$\mathcal{N}'_{\mathrm{vxl}}$	extended set of voxels with uppermost virtual layer
$\boldsymbol{\nu}, \boldsymbol{\nu}^{\mathrm{min}}, \boldsymbol{\nu}^{\mathrm{max}}$	vector containing the detector cells projections for horizontal overlaps computation
O	main origin of the inertial system $Oxyz$
O'	origin of the roto-translated Cartesian system $O'utv$
$Oxyz$	main inertial Cartesian coordinate system
$O'utv$	roto-translated Cartesian coordinate system
$Ou't'v'$	rototed Cartesian coordinate system
$\mathbf{O}(\cdot)$	(implicit or explicit) objective function for ROI CT
$\Omega_{\mathbf{f}}$	feasible set for $\mathbf{f}$
p	norm subscript
p_{ijk}	linear index position corresponding to $\boldsymbol{v}_{\mathrm{vxl}}(i,j,k)$
$\widetilde{p}_k^{\mathrm{H}}$	displacement of the positions related to the k-th layer
$\boldsymbol{p}_k^{\mathrm{H,left}}, \boldsymbol{p}_k^{\mathrm{H,right}}$	column vectors of the indices of all the leftmost and rightmost endpoints of the horizontal edges of the midway voxel slices projections at layer k
$\boldsymbol{p}^{\mathrm{H,left}}, \boldsymbol{p}^{\mathrm{H,right}}$	column vectors of the indices of all the leftmost and rightmost endpoints of the horizontal edges of the midway voxel slices projections for all layer
$\widetilde{\boldsymbol{p}}^{\mathrm{H}}$	displacement vector for the assignment of vertical position to horizontal overlaps

$\boldsymbol{p}^{\mathrm{H},i}$	column vector of the voxel indices whose projections have a nonempty horizontal overlap with the projections of the i-th column of detector cells
$\boldsymbol{p}^{\mathrm{V},j}$	column vector of the voxel indices whose projections have a nonempty vertical overlap with the projections of the j-th row of detector cells
$\overline{\boldsymbol{p}}^{\mathrm{H},i}$	vector of "converted" linear indices (from voxels to voxel vertices)
$\boldsymbol{p}^{\mathrm{H},i}_{\mathrm{btm}}$	column vector corresponding to the indices set $\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl,btm}}$
$\boldsymbol{p}^{\mathrm{H},i,1}, \boldsymbol{p}^{\mathrm{H},i,2}, \boldsymbol{p}^{\mathrm{H},i,3}$	column vector corresponding to the indices sets $\mathcal{I}^{\mathrm{H},i,1}_{\mathrm{vxl}}, \mathcal{I}^{\mathrm{H},i,2}_{\mathrm{vxl}}, \mathcal{I}^{\mathrm{H},i,3}_{\mathrm{vxl}}$, respectively
$\left(p^{\mathrm{H,idx},i}_{\mathrm{min}}\right)_s, \left(p^{\mathrm{H,idx},i}_{\mathrm{max}}\right)_s$	minimum and maximum row index of the lower voxel in the s-th stack whose vertical projection reaches the projection of the first cell in the i-th detector column
$p^{\mathrm{H,col},i}_{\mathrm{btm},s}$	displacement from the global linear index of the voxel at the bottom of the stack
$p^{\mathrm{vxl},H}_{r}$	column index for $\mathbf{W}_{\mathrm{H}}$ overlap allocation
$\boldsymbol{p}^{\mathrm{V,low}}, \boldsymbol{p}^{\mathrm{V,up}}$	column vectors of the indices of all the lowermost and uppermost endpoints of the vertical edges of the midway voxels slices projections for all layer
$p_{\mathbf{Y}}(\cdot)$	probability density associated to the random variable $\mathbf{Y}$
$p_{\mathbf{FY}}$	joint probability density associated to the random variables $\mathbf{F}$ and $\mathbf{Y}$
$\mathbf{p}_{\mathrm{ROI}}$	ROI center
$\pi(\cdot)$	unitary representation
π_{dtc}	detector plane
π_{prj}	common projection plane
P	generic point on the detector – Chapter 3
$P_{\eta}(\cdot)$	set of η-approximations (VMILA)
$\mathcal{P}_{\mathcal{B}}$	euclidian projection onto the set $\mathcal{B}$
$\mathcal{P}_{\mathcal{B},D}$	scaled euclidian projection onto the set $\mathcal{B}$ according to the scaling matrix D
$\mathcal{P}(S)$	ROI-truncated projections
$\mathscr{P}_i$	i-th pyramidal region of the shearlets 3D frequency domain partition
q_{δ}	(smooth) approximation of the square root function for TV definition
$\widetilde{q}^{\mathrm{dtc}}_{i}$	rows offset for $\mathbf{W}_{\mathrm{H}}$ overlap allocation
$\overline{q}^{\mathrm{dtc}}_{\ell}$	rows displacement for matrix $\mathbf{W}$
$\boldsymbol{q}^{\mathrm{dtc},i}_{rs}$	row index for $\mathbf{W}_{\mathrm{H}}$ overlap allocation

$\widehat{\boldsymbol{q}}^{\mathrm{H,left}}, \widehat{\boldsymbol{q}}^{\mathrm{H,right}}$	column vectors of the indices of all the leftmost and rightmost endpoints of the detector cells projections for the lowest detector row
$\boldsymbol{q}^{\mathrm{H,left}}, \boldsymbol{q}^{\mathrm{H,right}}$	column vectors of the indices of all the leftmost and rightmost endpoints of the detector cells projections
$\boldsymbol{q}^{\mathrm{V,low}}, \boldsymbol{q}^{\mathrm{V,up}}$	column vectors of the indices of all the lowermost and uppermost endpoints of the vertical edges of the detector cells projections for all layer
$\mathbf{r}_{\mathrm{ROI}}$	ROI radius
R_ℓ	2D counter-clockwise rotation matrix of the Oxy
$\mathcal{R}_{\mathcal{C}}(\cdot)$	cone-beam circular rotation system
$\mathcal{R}_\mu$	family of operator defining a regularization algorithm
$\mathcal{R}f$	Radon transform of f
$\mathscr{R}$	centered square of the shearlets 2D frequency domain partition
ρ	TV regularization parameter
s	shearing parameter (unless otherwise specified)
S	region-of-interest (ROI) – Chapters 2, 6
S	X-ray source – Chapter 3
S	frame operator – Appendix A
S_s	shearing matrix depending on the shearing parameter s
$\mathcal{S}^{n-1}$	unit sphere in $\mathbb{R}^n$
$\mathscr{S}(\mathbb{R}^n)$	Schwartz space on $\mathbb{R}^n$
$\mathbb{S}$	full shearlet group
$\mathcal{SH}$	(continuous and discrete) shearlet system
$\mathscr{SH}_\psi$	(continuous and discrete) shearlet transform
$\sigma_{r,s}^{\mathrm{H,idx},i}$	auxiliary vector for horizontal overlap replication
σ	noise variance – Chapters 2, 6
σ	VMILA set of parameters – Chapter 5
Σ	domain of the VMILA set of parameters σ
t	translation parameter (usually a vector)
T_t	translation operator associated to the (vector) parameter t
$\mathcal{T}$	tangent space to the circle
$\mathcal{T}^n$	tangent bundle to $\mathcal{S}^{n-1}$
$\tau_i^{(\ell)}$	vertical gap for the i-th detector cells vertices column at angle θ_ℓ
τ_{it}	threshold for stopping iteration in SGP and VMILA
$U(\cdot)$	negative part of the gradient decomposition
$\mathcal{U}(\cdot)$	group of unitary operators
$\mathbf{v}$	primal variable in VMILA

$\tilde{\mathbf{v}}$	inexact computation of the primal variable $\mathbf{v}$
$\mathbf{v}^{(k)}$	primal variable in VMILA at iteration k
$\tilde{\mathbf{v}}^{(k,l)}$	inexact computation of $\mathbf{v}^{(k)}$
$\boldsymbol{v}_{\text{vxl}}(i,j,k)$	coordinates of a generic object voxel slice in $Oxyz$
$\boldsymbol{v}_{\text{vxl}}^{(\ell)}(i,j,k)$	coordinates of a generic object voxel slice in $Oxyz$ at angle θ_ℓ
$\boldsymbol{v}_{\text{dtc}}(i,j)$	coordinates of a generic detector cell vertice in $Oxyz$
$\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)$	coordinates of a generic detector cell vertice in $Oxyz$ at angle θ_ℓ
$V(\cdot)$	positive part of the gradient decomposition
$V_i(\cdot)$	i-th component of positive part of the gradient decomposition
$\mathcal{V}_{\text{vxl}}$	set of voxel slices vertices
$w_{m,n}$	distance-driven weights (elements of the matrix $\mathbf{W}$)
$\boldsymbol{w}$	versor of a generic ray
$\boldsymbol{w}_{\pi_{\text{prj}}}$	common projection plane versor
$\mathcal{W}_\psi$	(continuous and discrete) wavelet transform
$\mathbf{W}_{\text{H}}$	matrix containing horizontal overlaps
$\mathbf{W}_{\text{V}}$	matrix containing vertical overlaps
$\mathbf{W}$	matrix discretization of the projection geometry
$x_{u_{ij}}^{\text{prj}}, y_{u_{ij}}^{\text{prj}}, z_{u_{ij}}^{\text{prj}}$	projections coordinates of the detector cells vertices $\boldsymbol{v}_{\text{dtc}}$
$\boldsymbol{x}_{\text{dtc},0}^{\text{prj}}, \boldsymbol{y}_{\text{dtc},0}^{\text{prj}}, \boldsymbol{z}_{\text{dtc},0}^{\text{prj}}$	vectors of the projections of all the detector vertices at layer 0
$\boldsymbol{x}_{\text{dtc}}^{\text{prj}}, \boldsymbol{y}_{\text{dtc}}^{\text{prj}}, \boldsymbol{z}_{\text{dtc}}^{\text{prj}}$	vectors of the projections of all the detector vertices
$x_{x_{ijk}}^{\text{prj}}, y_{y_{ijk}}^{\text{prj}}, z_{z_{ijk}}^{\text{prj}}$	projections coordinates of a single vertex of any voxel slices
$\boldsymbol{x}_{\text{vxl},k}^{\text{prj}}, \boldsymbol{x}_{\text{vxl},k}^{\text{prj}}, \boldsymbol{x}_{\text{vxl},k}^{\text{prj}}$	vectors of the projections of all midway slices vertices at layer k
$\boldsymbol{x}_{\text{vxl}}^{\text{prj}}, \boldsymbol{x}_{\text{vxl}}^{\text{prj}}, \boldsymbol{x}_{\text{vxl}}^{\text{prj}}$	vectors of the projections of all midway slices vertices
$\mathbf{x}_i$	vector representing the i-th basis function center for object discretization
$\mathcal{X}f$	X-ray transform of f
$\mathcal{X}^*$	adjoint operator of $\mathcal{X}$
$\boldsymbol{\xi}, \boldsymbol{\xi}^{\min}, \boldsymbol{\xi}^{\max}$	vector containing the midway slices projections for horizontal overlaps computation
y	full sinogram
y_0	truncated sinogram
$\mathbf{y}$	column vector representing the full sinogram discretization
$\bar{\mathbf{y}}$	ML or MAP estimate of the full sinogram
$\mathbf{y}^{(k)}$	ML or MAP estimate at iteration k of the full sinogram

$\mathbf{y}_0$	column vector representing the discretization of the truncated sinogram
$\mathbf{Y}$	vector valued random variable corresponding to a vector $\mathbf{y}$ of measured data (*e.g.*, sinogram)
υ	generalized solution of the problem – Chapter 2
υ	dual variable in VMILA – Chapters 5, 6
$\upsilon^{(l)}$	dual variable at iteration l
Υ	(generic) objective function
Υ_p	Morozov formulation for the ROI CT objective function
Υ_{DM}	(generic) data mismatch functional
Υ_R	(generic) regularization functional
$z^{\mathrm{prj}}_{\mathrm{vxl,min}}, z^{\mathrm{prj}}_{\mathrm{vxl,max}}$	minimum and maximum over all z coordinates of the voxels slices projections onto π_{prj}
$z^{\mathrm{prj}}_{\mathrm{dtc,min}}, z^{\mathrm{prj}}_{\mathrm{dtc,max}}$	minimum and maximum over all z coordinates of the detector cells projections onto π_{prj}
$Z_{\mathrm{H},i}$	matrix whose columns contain the vertical coordinates of the voxels slices vertices projections with linear index in $\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl}}$
$Z^{\mathrm{idx}}_{\mathrm{H},i}$	matrix of integers containing the row indices for the i-th detector cells column, corresponding to the vertical coordinates in $Z_{\mathrm{H},i}$

Contents

Chapter 1

Introduction

Computed tomography (CT) was historically the first method to allow the possibility to acquire images of the inner structure of an object non-invasively, namely without penetrating or cutting into pieces the object. CT has been a major breakthrough in diagnostic medicine, where, thanks to a non-biased superposition of anatomical features, the internal structures of the human body could be seen distinctly. This opened new opportunities in the recognition and understanding of human diseases without exploratory surgery. Nowadays, CT is used routinely in medicine for diagnostic purpose, thanks to the clarity and accuracy of images produced by CT scanners. Although magnetic resonance imaging (MRI) and positron emission tomography (PET) have been widely installed in worldwide medicine departments, CT is, to date, the most widely used imaging diagnostic device in hospital departments and trauma clinic [23].

However, the revolution brought by CT does not end with radiology and nuclear medicine. In fact, the very first application of CT in 1956 took place in radio astronomy, and was due to Ronald Bracewell [20]. Later on, diverse technical, anthropomorphic, forensic, archeological, as well as paleontological, applications of CT have been developed [23]. The underneath leitmotif is to provide a generic industrial diagnostic tool for nondestructive material testing (NDT).

Up to date, it has been about 130 years, counting from the discovery of X-rays by Wilhelm C. Röntgen in 1895, of active research in the field of X-ray tomographic imaging. Nevertheless, research in the field of CT is still as exciting as at the beginning of its development during the 1960s and 1970s.

The focus of this thesis is region-of-interest tomography (ROI CT), a limited data tomographic acquisition mode, that will be described in Section 1.3. After a long period, from the late 1970s to the early 1990s, during which the ROI CT problem was believed to be insolvable, the intensive research in the last decade not only revealed the reconstruction from incomplete data possible but also counted ROI CT in the "hot topics" in the field of X-ray tomographic imaging [33]. The primary interest lays in the possibility to reduce the scanning time and lower the X-ray radiation dose, while maintaining the possibility to handle large object, in high resolution.

1.1 A brief historical overview

The history of X-ray imaging starts in 1895 with the discovery of a new kind of radiation called “X-rays” by its finder Wilhelm C. Röntgen. Between 1895-1896 Röntgen conducted a series of experiments to show that there existed, indeed, a technique able to take “X-ray photographs” of the internal features of a person without any surgical intervention [141, 140]. Quickly, his technique found justified recognition among doctors and spread worldwide: for this, he was awarded the Nobel prize in 1901.

By the 1930s, the design of X-ray equipment was refined. Simple radiographs of early attempts contained the 2D shadow deriving from the superposition of *all* 3D structures in the object. Thus, it was impossible to recover precise informations concerning any particular feature of interest. At the beginning of the 1920s, there were many attempts to erase superimposed shadows from X-ray images. The goal was to display sharply the plane in focus and to blur the out of focus planes. This resulted in a number of patent applications (*e.g.*, A.E.M. Bocage in 1922 [12] and E. Pohl in 1927 [133]) and a number of papers published by researchers from different countries that were rediscovering similar concepts: all those works dealt with the same geometry technique, in which the X-ray tube and the detector moved along two parallel planes. As a consequence, the technique became known under several names: it was called *stratigrafia* by A. Vallebona [153], *laminography* by J. Kieffer [93] and *planigraphy* by A.E.M. Bocage and B.G. Ziedses des Plantes [171].

The discovery of X-rays was a necessary but non-sufficient condition for the rise of CT. Indeed, modern imaging techniques rely on computers. The second part of the story take off, therefore, from the development of computational techniques. In the early 1960s, while observing the planning of radiotherapy treatments, Allan Cormack posited that, by taking X-ray images from multiple directions, one should be able to piece together the internal structure of the body. To give a proof of concept, he built a prototype scanner that it is perhaps the first CT scanner actually built [37, 38]. As a theoretical physicist, Cormack was slightly concerned about the practical application of his research. It was the work of the electrical engineering Godfrey N. Hounsfield, while employed at the Central Research Laboratories of EMI Ltd., which led to the construction of the first clinical CT scanner. The scanner was installed in 1971 at the Atkinson MorleyÕs Hospital in Wimbledon and in 1972 Hounsfield and J. Ambrose gave a talk on “Computerised Axial Tomography” at the 32nd Congress of the British Institute of Radiology [81]. By the end of 1973, the first commercial CT scanner was on the market and Hounsfield, along with Cormack, received the 1979 Nobel Prize in Medicine. In 1976 the first whole body *fan-beam* CT scanner appeared and Kalender published the first clinical *helical* CT (also referred to as *spiral* CT) in 1989. From the early 1990s, demonstrations of *cone-beam* and *multislice* CT come to appear. Nowadays, CT images are also used with other modalities, such as SPECT-CT and PET-CT. A more detailed overview of the generations of CT scanners is summarized in Chapter 3

Since the introduction of the first clinical scanner, tremendous advancements have been made in CT technology. Contemporary CT scanners can scan in a few hundred milliseconds and reconstruct a square image of 2048 pixels, which is incredible compared to performance of Hounsfield's first CT: a square image of 80 pixels required a scan time of about 4.5 minutes and 20 seconds for the reconstruction phase, with a cross-section of 13 mm thickness.

However, there are still many active research directions to investigate. Recent trends in CT scanner design look at three aspects: thinner slices for scanning; awareness of patient dose; increased use of 3D visualization devices as the primary diagnostic tools.

More information about the history of CT can be found, *e.g.*, in [23, 32, 155].

1.2 The mathematical viewpoint

The fundamental process at the basis of CT is the different levels of X-ray absorption by materials, in an object, or the human body. Basically, a CT apparatus consist of a single X-ray source and an array of X-ray detectors. The CT modality requires that the object is kept stationary while the tube and the detector array rotate together around it. By rotating, the CT system collects measurements, usually addressed to as *projections*, which represents the "shadow" of the object onto the detector, for a number of fixed positions of the source and detector, called *views*. The resulting collected data are referred to as *sinogram*.

From a mathematical point of view, the absorption is described by a function, generally called *linear attenuation function*, that represents the object to be imaged by the CT imaging system. Each intensity in the sinogram is proportional to the line integrals of the X-ray attenuation function between the corresponding source and detector positions. The problem, therefore, consist in reconstructing the linear attenuation function, with adequate accuracy, starting from the acquired sinogram. The described scenario belongs to class of *inverse problems* and the solution is generally complex and involves techniques in physics, mathematics, and computer science [9, 77].

The solution of the inverse problem in CT had already been published in 1917, many years prior to the advent of computers and CT. The mathematician Johann Radon published a paper entitled “On the determination of functions from their integrals along certain manifolds”, where he introduced the mathematical relation called *Radon transform* [139]. Due to the complexity and depth of the mathematical publication and the fact that was written in German, the important applications that were to come from his work were hardly foreseen until late mid-20th century [135, 23].

During the boom in the 1970s which brought to the development of classical CT, it was noticed that the reconstruction techniques independently produced by its early inventors were, indeed, equivalent to the work of J. Radon. Bracewell's approach used the method of *back projections* [20], based on the inversion of the Radon transform: nowadays, the Filtered Back-Projection (FBP) is probably the most famous among the analytic or transform-based methods. Roughly, FBP combines the back projection phase with a ramp filtering to denoise [76]. On the other hand, the original EMI machine prototyped by Hounsfield used an iterative algorithm universally known as Algebraic Iterative Technique (ART), among the class of iterative methods [82].

At the early 1970s, both iterative and analytic methods were successfully used in the first clinical CTs. Later on, evolution and innovation in CT reconstruction techniques were mainly driven by advances in CT system designs. Namely, analytical algorithms aim at formulating the solution in a closed-form equation, while iterative algorithms generates a sequence of improving approximate solutions. As a consequence, analytical reconstructions are considered computationally more efficient, while iterative reconstructions usually require much higher computational demands but result in an improvement of the image quality. For this reason, after the enthusiasm of early 1970s, when iterative reconstruction were employed since relatively small amounts per scan of measured data were generated, until the 1990s analytical methods had a prominent role. To this class belongs the algorithm published in 1984 by Feldkamp, Davis, and Kress [53], usually referred to as FDK, for circular cone-beam tomography, which is still widely used in state-of-the-art cone-beam scanning devices. It can be understood as an extension of exact 2D reconstruction algorithms for fan-beam projections to the 3D case by properly adapting the weighting factors. With the introduction of helical CT, the most exciting development was probably the extension of the FBP algorithm to the helical case, due to A. Katsevich [89, 90]. Beside this major contribution, there exist other approaches derived from the original FDK algorithm and adapted to the helical CT case.

Thanks to the ever-increasing growth of computer technology, in terms of computational capacities available in modern processors (central processing unit, CPU) and graphics adapter (graphics processing unit, GPU), the employment of iterative reconstruction methods for clinical workflow has become a realistic option. One main reason to prefer iterative techniques over analytical ones is the possibility to include various physical models and prior information as a constraint for reconstruction. The simplest form of iterative reconstruction is the above-mentioned ART [62], based on Kaczmarz method for solving linear systems of equations of the form $Ax = b$ [87]. One variant of ART is the simultaneous ART (SART) [2], which performs updates for complete raw data projections instead of processing a single pixel at the time. The simultaneous iterative reconstruction technique (SIRT) [60], with its variant ordered subsets SIRT (OS-SIRT) [161], and the multiplicative algebraic reconstruction technique methods (MART) [5]Êare other ART-based methods. They are non-statistical and, in general, are able to model the geometry of the acquisition process better than common analytical methods based on FBP since can better deal with sparse data and irregular samplings of acquisition positions. Among the iterative methods, the family of statistical methods comprehend two main groups: maximum likelihood (ML) principle based methods and least squares (LS) principle based methods. The maximum likelihood expectation-maximization (ML-EM) algorithm [109] is one of the most popular among the ML class. It consists of two alternating steps: the E-step computes the expected log-likelihood, and the M-step finds the next estimate by maximizing the finding at the previous step. Diverse variant of these methods exist, introduced to speed convergence and allows for easy parallelization: the convex algorithm, the OS-EM [122], and a combination of this two, the ordered subset convex algorithm (OSC) [88, 51, 7]. The class of LS methods counts in the LS conjugate gradient (LSCG) method [117, 118, 55], the iterative coordinate descent (ICD) [144, 18, 150] and its faster variant OS-ICD [112, 169]. These methods are usually harder to parallelize since single pixels or coordinates are iteratively updated to minimize a cost function. Finally, the family of iterative methods comprehends the model-based iterative reconstruction (MBIR) [150, 166], whose main concern is to model the acquisition process as accurately as possible, namely including both photon statistics and geometry modelling. A more consistent review on analytical and iterative reconstruction techniques, with a detailed comparison of pros and cons, can be found in [46, 8, 84].

To date, four major CT vendors have presented their iterative reconstruction products: two of these, VEO introduced by GE Healthcare in 2009 and SAFIRE (Sinogram affirmed iterative reconstruction) developed by Siemens in 2010, have received clearance released by the Food and Drug Administration of the USA, once certified the considerable X-ray dose reduction possibilities of these technologies [8]. Especially when dealing with clinical CT, this is a prime target, which argues in favor of iterative techniques. In fact, already shortly after the discovery of X-rays, the radiation injuries caused by the harmful effect of ionizing radiation were observed [152]. However, despite all the risks, the X-ray imaging still offered very attractive opportunities in medical imaging and NDT and that is why the research in this field never stopped.

At the turn of the century, classical 2D and 3D CT technologies are considered mature fields. Beside this, the above-mentioned concern in industrial CT testify that there is an ever-growing interest in CT techniques that allows a lower X-ray radiation dose. This motivates the renewed interest in problems of limited or incomplete data tomography. Intuitively, it is clear that when not all lines are measured or not all views are considered, there is a reduction of the X-ray dose. Moreover, in many important tomography problems complete data can not be obtained [98].

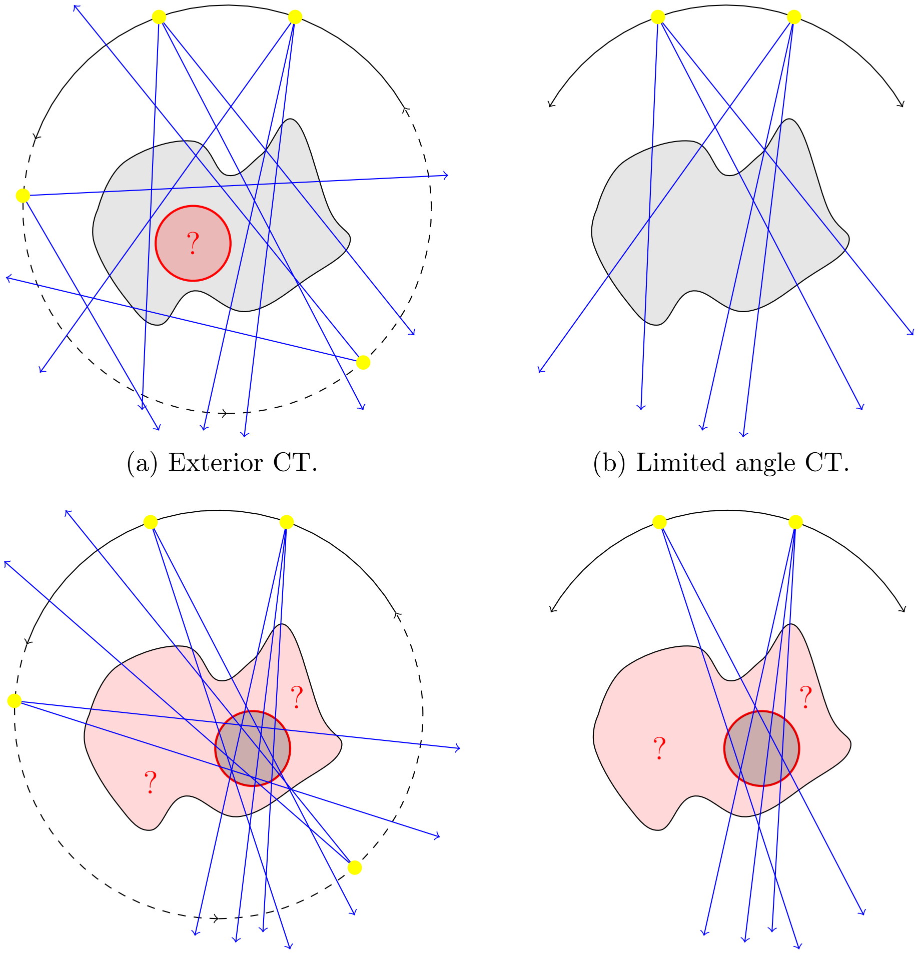
It is not a "new problem", though. At the end of the 1970s the research interest had already turned to the study of limited and incomplete data problems, to finally conclude, in the late 1980s, that no reliable solution could be ever be obtained. Discovering that accurate and robust reconstructions from incomplete data were, indeed, possible revived the interest in this topic, abetted also by the possibility to reduce the scanning time [33].

All this prompt us to focus on the resolution of ROI CT, an incomplete data problem, by employing more sophisticated algorithms, compared to the state-of-the-art methods presented in this Section. This required a sound understanding of the mathematical modeling of the problem and of its geometry. In the next Sections we present an overview of all classified limited and incomplete data problems. Next, a brief presentation of inverse problems as a general framework for ROI CT is given and the Chapter concludes with an outline of the thesis, pointing out the original contributions of this work.

1.3 Limited data tomography problems

We deal with *limited or incomplete data* problems when only a proper subset of all lines crossing the object is measured or a limited range of views is considered. According to T. Quinto [98], there exist four types of limited data problems.

(a) *Exterior CT*. In this case, only the data for lines outside an excluded region (usually a circle) are measured and the goal is to recover the object outside this region, as depicted in Figure 1.1a. This problem dates back to the early days of CT, when a single scan of a planar cross section required several minutes: the movements of a beating heart would create artifacts in the scan, unless a sufficient large region containing the moving heart could be consider, leaving outside a region that would not move. The advance in technology made this problem obsolete. Nowadays, exterior CT is fundamental for imaging large objects, *e.g.*, rocket shells, whose center is too



(a) Exterior CT.

(b) Limited angle CT.

(c) Region-of-interest CT.

(d) Limited angle ROI CT.

Figure 1.1: Types of limited data problems.

thick to penetrate. Finally, there exist a mathematical result proving that compactly supported functions can be uniquely reconstructed outside the excluded region from exterior data and several effective inversion methods have been developed [137, 98].

(b) *Limited angle CT.* Here, data are given only over all lines within a limited angular range, as in Figure 1.1b. Like exterior CT, it is a classical problem from the early days of CT and, currently, limited angle data are being employed in important applications, such as breast tomosynthesis, luggage scanners and dental X-ray scanning. Mathematically, a unique solution exists for compactly supported functions but it is, in general, very unstable. Many algorithms were developed to address this problem (see [113, 136] and the reference therein).

(c) *Region-of-interest CT.* In this modality measurements are taken only within a limited region-of-interest (generally, a circle) strictly inside the object support (interior ROI), as illustrated in Figure 1.1c. The problem consist in reconstructing the structure of the ROI only from these data. Notice that, because of the overlapping principle of the CT measurements, the contribution from the object outside the ROI is also included into the measured data. This problem comes up, *e.g.*, in biomedical applications of CT or micro-CT, where information is required only about some ROI, or in high-resolution tomography problems of small parts of objects, for which it is difficult or impossible to get complete high-resolution CT data.
Since this is the main topic of this thesis, more details will be discussed throughout the thesis (see Chapter 2).

(d) *Limited angle ROI CT.* This is a combination of the second and the third type, namely data are given over lines in a limited angular range with the additional constraint to pass through a given ROI, as shown in Figure 1.1d This problem is encountered, *e.g.*, in electron microscopy [137, 138].

1.4 Inverse problems

CT images provide the very first example of images obtained by solving a mathematical problem which belongs to the class of the so-called *inverse problems.* Nowadays, inverse problems are ubiquitous in physics (actually, science in general), mathematics engineering and industry, and, within the past decades, they received a great deal of attention by (applied) mathematicians, statisticians, and engineers. This is mainly due to the rise of large and powerful computers, the reliable numerical methods to face them and, first and foremost, the importance of its applications, that range from geophysics to biomedicine, to economy and finance, astronomy, and life science in general.

According to J.B. Keller, “we call two problems inverses of one another if the formulation of each involves all or part of the solution of the other”, and, he states further, “for historical reasons, one of the two problems has been studied extensively for some time, while the other has never been studied and is not so well understood. In such cases, the former is called the direct problem, while the latter is the inverse problem” [92]. Roughly, the duality that links a direct and the corresponding inverse problem is that one can be obtained from the other by switching the role of the data and the unknowns. In other words, a direct problem is a problem oriented along a cause-effect sequence while the inverse problem implies a reversal of the cause-effect sequence.

Thus, the mathematical viewpoint in inverse problems consist in using the results of actual observations or indirect measurements to infer the model or an estimation of certain values of the parameters characterizing the system under investigation. For instance, in radio astronomy one aims at determine the shape of celestial objects emitting radio waves, from the radio waves received by radio telescope on the surface of the Earth. In groundwater flow modeling, one estimates material parameters of an aquifer from measurements of pressure of a fluid that immerses the aquifer. In life sciences, inverse protein folding problems are examples of classical inverse problems, and, in finance, a central problem is the calibration of the volatility of the stock price, the main parameter which governs the models of derivates based on stock prices and can not directly read off of the market data.

In many instances, this estimation process is *ill-posed*, in the sense that noise (understood as "unwanted electrical fluctuations") in the data may give rise to significant errors in the estimate. More precisely, an ill-posed problem do not fulfill Hadamard's postulates of well-posedness, *i.e.*, a solution may not exists or the solution might not be unique and/or might not depend continuously on the data. On the other hand, many inverse problems can be formulated as linear problems, which frequently lead to integral equations of the first kind, even if many basic inverse problems are inherently nonlinear albeit the corresponding direct problem is linear. As we shall see in Chapter 2, this is also the case of CT, in general, and ROI CT, in particular: CT is a linear inverse problem and, when the projections are truncated, as in ROI CT, ill-posedness is even more severe.

How to deal with ill-posedness is still an open research field and it is generally problem-dependent. However, a new mathematical technique proposed in 1963 [151], called *regularization theory*, has dominated and is still dominating the scene of inverse problems. With this approach, an ill-posed problem is approximated by exploiting a family of well-posed ones. This technique relies on another well-established principle in inverse problems solution, namely, the relevance of additional information on the solution. As we shall see in Chapter 5, in regularization theory the additional information reappears in a statistical form.

To this framework belongs the ROI CT problem, for which will be investigated a regularization approach based on shearlets, a multiscale method whose optimally sparse approximation properties are potentially relevant in CT-like applications, as we will see in Chapter 4.

There is a vast literature on inverse and ill-posed problems, including books, reference proceedings and dedicated journals. Seminal works are, *e.g.*, [9, 50, 64, 73, 157] and more references can be found therein, while the recent [145] gives an overview on the solution of inverse problems from applied and industrial research applications.

1.5 Thesis outline and original contributions

This thesis consists of seven chapters and an appendix. This Chapter serves as an introduction to this doctoral dissertation. It includes an extensive summary on the historical development of CT and its reconstruction algorithms, an overview on the theoretical aspects of image reconstruction from limited or incomplete data, and a brief presentation of inverse problems as a general framework for the ROI CT problem.

The ROI CT problem formulation, both in the continuous and the discrete settings, is addressed in Chapter 2. Afterwards, the problem is stated as a convex objective function, which exploits different levels of regularization, and a nonsmooth version. Chapter 2 is also comprehensive of a literature review on reconstruction strategies for the ROI CT problem.

Chapter 3 is devoted to a brand new and effective vectorized approach to a state-of-the-art technique, called distance-driven, for a faster discretization of the forward projection operator. The proposed approach applies to both the 2D fan-beam and the 3D cone-beam CT geometries. A literature review on the different generations of CT scanners and on the existing forward models for the projection matrix is given, first.

The mathematical background on the theory, application and implementation of shearlets, a recent multiscale method for the representation of multivariate data, is reiterated in Chapter 4. In this thesis, shearlets are used as a regularization tool to address the ROI CT problem. Since this topic is self-contained and stand-alone, and belongs primarily to the field of harmonic analysis, some background material on frame theory and locally compact groups is summarized in Appendix A.

To face the numerical solution of the ROI CT problem, two iterative algorithms are proposed and analyzed in Chapter 5, within the framework of a statistical approach for image reconstruction. The former one is the *scaled gradient projection method* and belongs to the class of first-order descent methods. It mainly applies to smooth, possibly constrained, problems with “simple” feasible regions. The latter one is the *variable metric inexact line-search algorithm*, a proximal-gradient method suitable for minimizing the sum of a differentiable, possibly nonconvex, function plus a convex, possibly not differentiable, function. Their practical implementation to the ROI CT problem is thoroughly investigated.

Numerical experiments and results are given in Chapter 6, once the general setup for the simulated data has been described. The numerical simulations are performed by exploiting the different levels of regularization enabled by the objective function designed in Chapter 2. In the last Section, a final discussion compares the results achieved with the findings obtained with a traditional technique.

The last Chapter summarizes the conclusions and gives suggestions for further work and perspectives.

The main original contributions of this thesis are the following:

- the study and the proofs on the vectorized approaches to the distance-driven method for the forward CT projection operator;
- an efficient implementation of the forward projection matrix for the 2D fan-beam and the 3D cone-beam circular geometries;
- the arrangement of the scaled gradient projection method and the variable metric inexact line-search algorithm for the ROI CT problem setting, with a parameter optimization analysis;
- the use of shearlets as regularization tool in the ROI CT problem;
- a topical review which includes a thorough literature review on CT and ROI CT problems, shearlets and state-of-the-art techniques for the forward projection operator.

Chapter 2

Region-of-interest tomography

As outlined in the Introduction, the impact of CT has been enormous in diverse areas, including industrial NDT, security screening and forensic, archeological and medical diagnostics even though, in this last application, exposure to X-ray radiation comes with health hazards for patients. Luckily, in many biomedical situations, such as contrast-enhanced cardiac imaging or some surgical implant procedures, *e.g.*, the positioning of intracranial stents [164], one is interested in examining only a small ROI with high resolution. Hence, there is no need to irradiate the entire body but only a smaller region, with the advantage of reducing radiation exposure and shortening the scanning time.

In this Chapter we setup the ROI CT reconstruction problem, formulating it as a convex optimization problem with a regularized functional based on shearlets (introduced in Chapter 4). The numerical solution of this problem will be addressed in Chapter 5, using the iterative minimization techniques presented therein. Part of this Chapter is based on [22, 21].

2.1 State-of-the-art

The goal of ROI CT is to reconstruct an object only within a limited region of interest, starting from the data projection acquired only for those rays meeting the ROI. From the mathematical point of view, classical CT reconstruction is an inverse and ill-posed problem. We recall that, according to Hadamard's definition, in an ill-posed problem the solution does not exists, or it is not unique, or there is lack of continuous dependence of the solution on the data. When one attempts to solve the reconstruction problem from incomplete or truncated projections, as in the case of ROI CT, the ill-posedness is even more severe [127].

For over two decades, from the late 1970s to early 2000s, it has been widely believed that theoretically exact local reconstruction of ROIs can not be obtained, abetted by the construction of a counterexample showing the nonuniqueness of the solution to the interior problem. Unexpectedly, in 2002, the first examples of accurate partial reconstructions from incomplete data, in 2D, appeared. This contradicted the general understanding that incomplete or truncated projections must inevitably generate artifacts throughout the image: incomplete was no longer synonymous with insufficient, when talking about CT data [33].

Soon it was established that ROI CT requires *ad hoc* approaches to ensure reliable reconstruction. Indeed, traditional CT algorithms, such as FBP or the FDK algorithms, straightforwardly applied to the interior reconstruction problem may create unacceptable artifacts overlapping features of interest, being more and more unstable to noise as the size of the ROI decreases, rendering the image inaccurate or useless [128].

To address the problem of ROI reconstruction from truncated projections a number of analytic and iterative methods has been proposed, during the last decade. A first attempt was provided by Lambda tomography, a gradient-like nonquantitative technique that gives the values and locations of jumps but does not allow to reconstruct the attenuation function pointwise [52, 136]. Next, it was shown that it was possible to derive analytic ROI reconstruction formulae from truncated data, even though such formulae usually require restrictive assumptions on the location of the ROI and depend on the acquisition setting [131, 34, 173]. For example, initial Differentiated Back-Projection (DBP) methods could be applied only if there existed at least one projection view in which the complete (*i.e.*, non-truncated) projection data are available. Later on, many others DBP methods, also based on the projection onto convex sets, were proposed, in which the above restriction was substituted with the requirement of a known subregion inside the ROI, that is usually available when air gaps, water or blood landmarks are at disposal [130, 35, 167]. DBP methods relies on the inversion of the so-called truncated Hilbert transform. As an alternative, an SVD-based DBP approach was proposed by A. Katsevich *et al.*, where the inversion of the truncated Hilbert transform is performed by means of the singular value decomposition transform [85]. Finally, using the compressive sensing theory, it was proved that exact interior reconstruction can be theoretically achieved, provided that the attenuation function is piecewise polynomial on the ROI [165, 162]. This result was complemented by the one of A. Katsevich *et al.*, in [91], for function that are polynomial, rather than piecewise polynomial, on the ROI: this is viewed by the authors as a first step towards a general proof of the stability for the piecewise polynomial case. However, the impact of noise on the stability of the analytic ROI CT methods has not been fully examined, and such methods might indeed become unstable in the presence of noise, according to the literature available to date.

Next to analytic methods, some common iterative algorithms have been adapted to ROI tomography. It has already been pointed out that methods belonging to this class are generally more flexible, since they can be applied to essentially any type of acquisition mode, also including several physical processes in the modeling, like noise or scattering. Iterative algorithms are usually computationally more demanding, but advances in high-performance computing make them more and more competitive [70, 8]. The ML expectation-maximization (MLEM) algorithm [147], SIRT [78], and the least-squares Conjugate Gradient (LSCG) method [80] have been applied to ROI tomography by several authors. The idea of iterative reconstruction-reprojection (IRR), first introduced for traditional CT in [126], has also been adapted to the ROI CT problem. These approaches typically involve some form of prior knowledge on the object attenuation function, such as the pilot reconstruction of the full object in [71], or the transition zone between the ROI and the non-ROI in [172], or a regularization term to ensure a stable ROI reconstruction, as the Total Variation term in [163]. For all these methods, the performance is usually rather sensitive to the ROI size and the presence of noise. In [149], the algorithm by Chambolle and Pock [28] is applied to an optimization problem expressed by means of a data fidelity term, which compares a derivative of the estimated data with the available projection data. This approach is somehow related to Lambda tomography. A Bayesian multiresolution method for local tomography reconstruction in dental X-ray imaging is proposed in [129], using a wavelet basis for the representation of the dental structures, with a high resolution inside the ROI and coarser resolution outside the ROI. Their approach is closely related to the one recently proposed in [97]. A wavelet-based regularization algorithm based on IRR is proposed in [4], with a variant that employs a smoothing convolution operator for the re-projecting phase. Also the regularity-inducing convex optimization (RICO) algorithm employs a wavelet-based regularization. However, in their work the authors explicitly account for the presence of noise by leveraging data fidelity, data consistency and sparsity-based regularization [61].

The approach presented in this thesis basically relies on the setup in [61]. Anyhow, a slightly different objective function will be investigated, including a shearlet term, possibly coupled with Total Variation. Moreover, the numerical assessment, in place of a (split) augmented Lagrangian, exploits algorithms belonging to the class of gradient projected-type methods, and its generalization to the nonsmooth case.

2.2 From CT to ROI CT: continuous setup

The general framework introduced in this Section works even for dimensions higher than two and applies to different projection geometries, including fan-beam and cone-beam. However, since the numerical assessment presented in Chapter 6 is carried out with 2D objects, the ROI CT setup will be focused on the 2D case.

2.2.1 A unique solution for the ROI CT problem

As already pointed out in the Introduction, the CT reconstruction problem consists in reconstructing a density function from a set of projections, obtained by measuring attenuation over straight lines. Mathematically, this is understood as a “line-integral model" by means of the *X-ray transform* notion.

Let $\mathcal{S}^{n-1}$ represent the unit sphere in $\mathbb{R}^n$, with $n \geq 2$. A straight line $\ell(\omega, \xi)$ in $\mathbb{R}^n$ is represented by a direction $\omega \in \mathcal{S}^{n-1}$ and a point $\xi \in \omega^{\perp}$, that is:

$$\ell(\omega, \xi) = \{\xi + t\omega \, : \, t \in \mathbb{R}\}.$$

where $\omega^{\perp} = \{\xi \in \mathbb{R}^n \, : \, \langle \xi, \, \omega \rangle = 0\}$ denotes the subspace orthogonal to ω and $\langle \cdot, \cdot \rangle$ is the usual scalar product in $\mathbb{R}^n$.

Definition 2.1 ([128])**.** Let $f \in L^1(\mathbb{R}^n)$. The *X-ray transform* $\mathcal{X} : L^1(\mathbb{R}^n) \to \mathcal{T}^n$ of the function f is the line integral of f along the lines $\ell(\omega, \xi)$:

$$(\mathcal{X} f)(\omega, \xi) = \int_{\ell(\omega,\xi)} f(x)\, dx = \int_{\mathbb{R}} f(\xi + t\omega)\, dt,$$

where $\mathcal{T}^n = \{(\omega, \xi) : \omega \in \mathcal{S}^{n-1}, \, \xi \in \omega^{\perp}\}$ is the tangent bundle to $\mathcal{S}^{n-1}$.

There is another fundamental definition in traditional CT theory, the *Radon transform.*

Definition 2.2 ([128])**.** Let $f \in L^1(\mathbb{R}^n)$. The *Radon transform* $\mathcal{R} : L^1(\mathbb{R}^n) \to \mathcal{C}^n$ of the function f is the line integral of f over the hyperplanes $\mathscr{H}(\omega, \tau)$:

$$(\mathcal{R} f)(\omega, \tau) = \int_{\mathscr{H}(\omega,\tau)} f(x)\, dx = \int_{\omega^{\perp}} f(\xi + \tau\omega)\, d\xi,$$

where $\mathscr{H}(\omega, \tau) = \{x \in \mathbb{R}^n \, : \, \langle x, \omega \rangle = \tau\}$ is a hyperplane perpendicular to $\omega \in \mathcal{S}^{n-1}$ with (signed) distance $\tau \in \mathbb{R}$ from the origin and $\mathcal{C}^n = \{(\omega, \tau) : \omega \in \mathcal{S}^{n-1}, \, \tau \in \mathbb{R}\}$ is the unit cylinder in $\mathbb{R}^n$.

Therefore, in 2D, the Radon transform and the X-ray transform differ from each other only in the parameterization, *i.e.*, they are both defined as a line integral, while in 3D the Radon transform is an integral over a plane. That is why the X-ray transform is chosen over the Radon transform to model the tomographic acquisition when $n \geq 3$.

Hereafter, we will restrict ourselves to the 2D case, namely $n = 2$. Let ω be a function of a polar angle $\theta \in \mathbb{R}$, *i.e.*, $\omega_\theta = \big(\cos(\theta), \sin(\theta)\big)^T \in \mathcal{S}^1$, $f \in L^1(\mathbb{R}^2)$ and $x \in \mathbb{R}^2$. The X-ray transform (or, equivalently, the Radon transform) of f at (θ, τ) is the line integral of f over the lines (or rays) $\ell(\theta, \tau)$ perpendicular to ω_θ with signed distance $\tau \in \mathbb{R}$ from the origin:

$$\mathcal{X} f(\theta, \tau) = \int_{\ell(\theta,\tau)} f(x)\, dx = \int_{\mathbb{R}^2} \delta(\tau - \langle x, \omega_\theta \rangle)\, f(x)\, dx, \tag{2.1}$$

where $\ell(\theta, \tau) = \{x \in \mathbb{R}^2 : \langle x, \omega_\theta \rangle = \tau\}$. The underlying physics relation is the Beer's Law [84], stating that the decrease in intensity at x is proportional to the intensity $I(x)$, understood as "number of photons", with the opposite of the attenuation coefficient $-f(x)$ as proportionality constant:

$$\frac{dI}{dx} = -f(x)\, I(x).$$

Heuristically, the meaning is that the more dense the material is at x, the more the beam is attenuated and the greater is the decrease of I at x. By using the separation of variables integration technique, one gets exactly relation (2.1):

$$\log\left(\frac{I_0}{I_1}\right) = \int_{x_0}^{x_1} f(x)\, dx = \int_\ell f(x)\, dx,$$

where ℓ is a line along which X-rays travel, from the source point $x_0 \in \ell$, with X-ray emitter intensity I_0, to the detector cell point $x_1 \in \ell$, whose intensity is I_1.

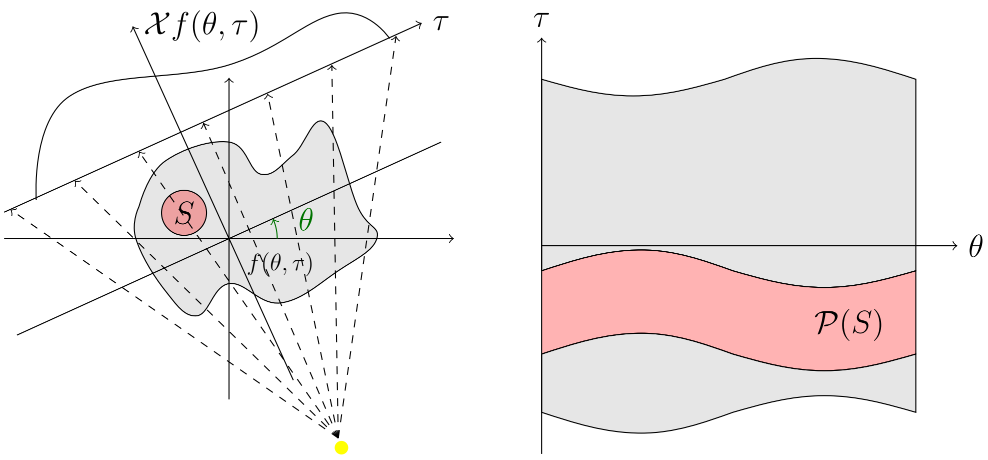
In what follows, we will refer to the X-ray projections as the *full sinogram* and we shall denote it by:

$$y(\theta, \tau) = \mathcal{X} f(\theta, \tau), \quad \theta \in [0, 2\pi),\ \tau \in \mathbb{R}. \tag{2.2}$$

Notice that $\mathcal{T} = \{(\theta, \tau) : \theta \in [0, 2\pi),\ \tau \in \mathbb{R}\}$ is the tangent space of the circle. Also, we will address the domain of (2.2) as the *projection domain* and the domain of the density or attenuation function as the *image domain.*

In the ROI tomography problem, *projections are collected only for those rays meeting a region of interest* inside the field of view. The goal is to recover the density function inside the ROI only, while the rest of the function is essentially ignored, even if the contribution from the object outside the ROI is included into the measured data, because of the overlapping principle of the CT measurements. By denoting with $S \subset \mathbb{R}^2$ the ROI, the set of ROI-truncated projections is identified to be the set

$$\mathcal{P}(S) = \{(\theta, \tau) \in \mathcal{T} : \ell(\theta, \tau) \cap S \neq \emptyset\} \subset \mathcal{T}.$$



(a) Image (object) domain.

(b) Projection domain.

Figure 2.1: Illustration of the ROI S in the image domain, on the left (a), and the corresponding ROI-truncated projections $\mathcal{P}(S)$, on the right (b).

Thus, the ROI reconstruction problem can be formulated as the problem of reconstructing the density function f restricted to the ROI S from the truncated X-ray projections. The geometrical set-up is illustrated in Figure 3.3. Mathematically, this reads as:

$$y_0(\theta,\tau) = M(\theta,\tau)\,\mathcal{X}f(\theta,\tau) \qquad \text{with} \quad M(\theta,\tau) = 1_{\mathcal{P}(S)}(\theta,\tau) \tag{2.3}$$

where M is the *mask function* on $\mathcal{T}$ and 1_A is the characteristic function of the set A, defined by

$$1_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}. \tag{2.4}$$

We will refer to $y_0(\theta,\tau)$ as the *truncated sinogram.* In the following, we will assume that the ROI S is a disk in $\mathbb{R}^2$, since it is the more natural choice, due to the circular trajectory of the X-ray source. If $\mathbf{p}_{\text{ROI}} \in \mathbb{R}^2$ denoted the center of the ROI and $\mathbf{r}_{\text{ROI}} \in \mathbb{R}$ its radius, it is clear that

$$\mathcal{P}(S) = \{(\theta,\tau) \in \mathcal{T} \,:\, |\tau - \mathbf{p}_{\text{ROI}} \cdot \omega_\theta| < \mathbf{r}_{\text{ROI}}\}.$$

More general convex ROIs can be taken into account by finding the minimal enclosing disk for such a ROI and reconstructing the image for this disk.

A natural approach for obtaining a stable reconstruction of f from equation (2.3) is by computing the least squares solution $\bar{f}$:

$$\bar{f} = \underset{f}{\operatorname{argmin}} \|M\mathcal{X}f - y_0\|_2^2. \tag{2.5}$$

The least squares approach is a state of the art method to solve linear inverse problems of the form (2.2), which can often be motivated for statistical reasons (see Chapter 5 for more details). It can be understood as a *data mismatch* term since, in general, it accounts for the discrepancy between the actual measurements and the ideal measurements.
However, it is known that the solution of the ROI problem, in general, is not guaranteed to be unique [127] and also the minimizer of problem (2.5) is not unique, since the set of solutions is the affine subspace

$$V = \{f \in L^2(\mathbb{R}^2) : y = \mathcal{X}f \quad \text{and} \quad My = y_0\},$$

given that the kernel of $M\mathcal{X}$ is not trivial, *i.e.*, $\ker(M\mathcal{X}) \neq \{0\}$, due to the truncation mask operator M. Moreover, even when the uniqueness is ensured, we already remarked that the inversion of the X-ray transform is an ill-posed problem, with the ill-posedness becoming more severe when the projections are truncated, as in the case of the ROI CT problem.

A classical approach to achieve uniqueness is to use the Tikhonov regularization method [151]. The basic idea of the Tikhonov method is to search for the minimum-norm solution, having incorporate a certain *a priori* assumptions about the smoothness of the solution by adding an additional norm condition. In details, one seek for the solution of the following minimization problem:

$$\underset{f}{\operatorname{argmin}} \left\{ \|Kf - g\|_2^2 + \mu \|f\|_2^2 \right\}, \tag{2.6}$$

where $K : \mathcal{H}_1 \to \mathcal{H}_2$ is a linear continuous operator between the Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, $g \in \mathcal{H}_2$ is the measured data and $\mu \in \mathbb{R}^+$, which is usually referred to as *penalty* or *regularization parameter*, controls the weight given to minimization of the additional norm term. Clearly, $K = M\mathcal{X}$ and $g = y_0$ according to the notation introduced in this Chapter. It can be shown that for each $\mu \in \mathbb{R}^+$ the minimum problem 2.6 is equivalent to the Euler equation

$$\big(K^*K + \mu I\big)f = K^*g$$

where K^* denotes the adjoint operator and I is the identity operator. The importance of the Tikhonov method relies in the following fundamental result.

Theorem 2.3 ([9, 50])**.** *The one-parameter family of operator $\{\mathcal{R}_\mu\}_{\mu\in\mathbb{R}^+}$ defined by*

$$\{\mathcal{R}_\mu\}_{\mu\in\mathbb{R}^+} = \big(K^*K + \mu I\big)^{-1}K^*$$

is a linear and regular regularization algorithm, in the sense specified by Definition 2.4.

Definition 2.4 ([9, 50])**.** Let $K : \mathcal{H}_1 \to \mathcal{H}_2$ be a linear continuous operator. A *regularization algorithm* is the one-parameter family of operator $\{\mathcal{R}_\mu\}_{\mu\in\mathbb{R}^+}$ such that

1. $\mathcal{R}_\mu : \mathcal{H}_1 \to \mathcal{H}_2$ is bounded for every $\mu \in \mathbb{R}^+$;
2. $\lim \|\mathcal{R}_\mu g - \upsilon\|_X = 0$ for every $g \in \mathcal{H}_2$ such that $Pg \in \text{range}(K)$, where P is the linear projection onto the closure $\overline{\text{range}(K)}$. Here, υ denotes the generalized solution of the problem.

The regularization algorithm $\{\mathcal{R}_\mu\}_{\mu\in\mathbb{R}^+}$ is said linear if $\mathcal{R}_\mu$ is linear for each value of the regularization parameter μ. The algorithm is also regular if for a given noise variance $\sigma \to 0$, the optimal choice of the regularization parameter $\mu_{\text{opt}}(\sigma) \to 0$ and $\mathcal{R}_{\mu_{\text{opt}}(\sigma)} g_\sigma \to \upsilon$, where g_σ is the noisy measurements.

Similarly to the least squares approach, the Tikhonov method can be motivated for statistical reasons (see Chapter 5), whence the definition of regular algorithm that accounts for noisy measurements.

In principle, the norm condition in the Tikhonov method can be applied in the image domain or in the projection domain, leading in general to different solutions. In details, this yields:

$$\underset{f}{\text{argmin}} \Big\{ \|Kf - g\|_2^2 + \mu \|\Theta(f)\|_2^2 \Big\}, \tag{2.7}$$

where Θ shall incorporate a low pass operators (*e.g.*, difference operator or wavelets-shearlets operator) to enforce smoothness or to induce a sparse solution. Finally, notice that, under mild condition [159], the Tikhonov regularization (2.6)-(2.7) is equivalent to the Morozov regularization:

$$\underset{f}{\text{argmin}} \, \|\Theta(f)\|_2^2 \qquad \text{s. t.} \quad Kf = g,$$

where the constraint equality can be relaxed by requiring $\|Kf - g\|_2^2 \leq \sigma$, with σ denoting the noise variance.
To define a suitable norm condition for the ROI CT problem in the projection domain, we recall the following definition.

Definition 2.5 ([128])**.** Let $h \in L^2(\mathcal{T})$. The Riesz potential operator $\mathcal{I}^{-\alpha}$ of the function h is defined by:

$$\widehat{(\mathcal{I}^\alpha h)}(\theta, \xi) = |\xi|^{-\alpha} \hat{h}(\theta, \xi), \qquad \alpha < 2, \tag{2.8}$$

where the hat symbol $\hat{\cdot}$ denotes the Fourier transform [58, 143].

Using (2.8), the L^2-norm of f can equivalently be computed in the projection domain. Indeed, the following formula, remarkable for its role in local tomography [98], holds true:

Lemma 2.6 ([128])**.** *Let $f \in \mathscr{S}(\mathbb{R}^n)$. Then for $\alpha < n$*

$$f = \frac{(2\pi)^{1-n}}{2} \mathcal{I}^{-\alpha}\, \mathcal{X}^*\, \mathcal{I}^{\alpha-n+1}\, \mathcal{X} f,$$

where $\mathcal{X}^*$ is the adjoint operator of $\mathcal{X}$, usually referred to as backprojection operator, and $\mathscr{S}(\mathbb{R}^n)$ denotes the Schwartz space on $\mathbb{R}^n$ [58, 143].

We are interested in the special case where $n = 2$ and $\alpha = 0$. This choice leads to:

$$f = \frac{1}{4\pi} \mathcal{X}^* \mathcal{I}^{-1} \mathcal{X} f, \tag{2.9}$$

Hence, we have:

$$\begin{aligned} \|f\|_2^2 &= \frac{1}{4\pi}\langle f, \mathcal{X}^*\mathcal{I}^{-1}\mathcal{X} f\rangle = \frac{1}{4\pi}\langle \mathcal{X} f, \mathcal{I}^{-1}\mathcal{X} f\rangle \\ &= \frac{1}{4\pi}\langle \mathcal{I}^{-\frac{1}{2}}\mathcal{X} f, \mathcal{I}^{-\frac{1}{2}}\mathcal{X} f\rangle = \frac{1}{4\pi}\left\| \mathcal{I}^{-\frac{1}{2}}\mathcal{X} f\right\|_2^2 \end{aligned} \tag{2.10}$$

Thus, we are lead to the following optimization problem:

$$\bar{f} = \operatorname*{argmin}_{f} \left\|\mathcal{I}^{-1/2}\mathcal{X} f\right\|_2^2 \qquad \text{s. t.} \quad y = \mathcal{X} f \quad \text{and} \quad My = y_0, \tag{2.11}$$

where the equivalence between Tikhonov and Morozov regularization has been used, with $\Theta = \mathcal{I}^{-1/2}\mathcal{X}$. By analogy, we set

$$\Upsilon_p(f) = \left\|\mathcal{I}^{-\frac{1}{2}}\mathcal{X} f\right\|_p^p \qquad \text{with } 1 \leq p \leq 2.$$

because in the following (see Chapter 5) we will not only consider the traditional case $p = 2$, but also a sparsity promoting modification of this minimization condition using the functional $\Upsilon_p(f)$ with $p = 1$. Notice that the use of the 1-norm in place of the 2-norm is a state-of-the-art approach, well investigated in literature in a number of applications, including compressive sensing, image deblurring and deconvolution, as convex approximation of the so-called ℓ_0-norm [49]. Roughly, we are looking for a solution which is sparse in the ℓ_1-sense in the transform domain.

Finally, the following remark will also be useful to design the objective function. It is easy to see that

$$y = y_0 + (I - M)\, y, \tag{2.12}$$

indicating that the ROI reconstruction problem can be viewed as an extrapolation problem where y_0, given on $\mathcal{P}(S)$, is to be extrapolated outside $\mathcal{P}(S)$. Clearly, not *any* extrapolated function y outside $\mathcal{P}(S)$ can be chosen, since, due to the overlapping principle of the CT measurements, the following global constraint is needed

$$\mathcal{X}f = y. \tag{2.13}$$

Equation (2.13) ensures that y belongs to range of the X-ray transform of the density function $f \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. By applying M and $I - M$ to the left-hand and right-hand sides of (2.13), respectively, we obtain the following equations:

$$M\mathcal{X}f = M\,y = y_0 \qquad \text{(data fidelity)} \tag{2.14}$$

$$(I - M)\mathcal{X}f = (I - M)\,y \qquad \text{(data consistency)} \tag{2.15}$$

The *data fidelity* equation defines a constraint inside the ROI while the *data consistency* equation enforces accurate reconstruction inside the ROI. These equations can be combined with regularization to obtain a suitable objective function for the ROI CT problem, as we shall see in section 2.3.

2.3 Discrete framework: explicit and implicit formulation of the objective function

To derive a discrete formulation for the ROI CT reconstruction framework, we need first to discretize equations (2.14) and (2.15). Given a projection geometry, the forward problem is discretized by means of a matrix $\mathbf{W}$ sized $N_\theta N_{\text{dtc}} \times N^2$, that represents the map from the image domain to the projection domain. Here, N_θ denotes the number of projection angles and N_{dtc} the number of detector cells (*i.e.*, samples along the detector array). The construction of the matrix $\mathbf{W}$ is carried out in Chapter 3. The mask identifying the ROI is the diagonal matrix $\mathbf{M}$ sized $N_\theta N_{\text{dtc}} \times N_\theta N_{\text{dtc}}$ whose entries are either 0 or 1. The unknown discrete density function f to be reconstructed is represented as a vector $\mathbf{f}$ of length N^2 in which the entries are stacked column by column. Here, N is both the width and the height in pixels (see Section 3.2) of the object to reconstruct. Similarly, the full sinogram y and the truncated sinogram y_0 are represented as vectors $\mathbf{y}$ and $\mathbf{y}_0$, respectively, of length $N_\theta N_{\text{dtc}}$, obtained by stacking the entries column by column. We recall that $\mathbf{y}$ and $\mathbf{y}_0$ are related to each other by the data fidelity equation

$$\mathbf{y}_0 = \mathbf{M}\mathbf{y} = \mathbf{M}\mathbf{W}\mathbf{f},$$

and the data consistency equation, that sets the extrapolation scheme outside the ROI, reads as

$$(\mathbb{1}_{N_\theta N_{\text{dtc}}} - \mathbf{M})\mathbf{W}\mathbf{f} = (\mathbb{1}_{N_\theta N_{\text{dtc}}} - \mathbf{M})\mathbf{y},$$

where $\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}$ is the $N_\theta N_{\mathrm{dtc}} \times N_\theta N_{\mathrm{dtc}}$ identity matrix. In particular, the data consistency equation yields

$$\mathbf{y} = (\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{W}\mathbf{f} + \mathbf{y}_0. \tag{2.16}$$

As indicated above, data fidelity and data consistency equations need to be coupled with regularization to yield a unique solution. Thus, similarly to (2.11), we obtain the following discrete optimization problem:

$$\bar{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{\Phi}\mathbf{W}\mathbf{f}\|_p^p \qquad \text{s. t.} \quad \mathbf{y} = \mathbf{W}\mathbf{f} \quad \text{and} \quad \mathbf{M}\mathbf{y} = \mathbf{y}_0 \tag{2.17}$$

where $p = 1, 2$ and the matrix $\mathbf{\Phi}$ sized $N_\theta N_{\mathrm{dtc}} \times N_\theta N_{\mathrm{dtc}}$ is a discrete filter corresponding to the Riesz potenzial operator $\mathcal{I}^{-\frac{1}{2}}$. Indeed, in place of considering a straightforward matrix discretization of the Riesz potential operator, we will approximate it by using the discrete shearlet transform, a multiscale method which refines the conventional wavelet framework by combining multiscale analysis and directional sensitivity [102]. Thanks to their properties, shearlets are potentially relevant in CT-like applications, since point-like structures in the image domain map onto sine-shaped curvilinear structures in the projection domain. A self-contained overview about shearlets main ideas is presented in Chapter 4. In particular, notice that when $p = 1$, the use of the 1-norm asks for the suppression of the smallest coefficients in favor of the largest ones, that should be the most significant. Instead, when $p = 2$ and $\mathbf{\Phi}$ is a frame or an orthonormal base, we are simply aiming at a Tikhonov-like regularization with the norm condition applying in the projection domain.

Now, rather than requiring the exact equalities stated by data fidelity and data consistency equations, we shall minimize the L^2-norm error associated to them, according to a maximum likelihood approach and the equivalence between the Tikhonov and Morozov minimization. By exploiting this idea, we can state two different optimization problems and objective functions. On the one hand, we can consider the L^2-norm error of the data fidelity equation only and incorporate the data consistency information in the regularization term. In this case, the only variable we are minimizing onto is the image $\mathbf{f}$ to be reconstructed. This approach yields:

$$\bar{\mathbf{f}} = \underset{\mathbf{f}\in\Omega_{\mathbf{f}}}{\operatorname{argmin}} \, \mathbf{O}(\mathbf{f})$$

with

$$\mathbf{O}(\mathbf{f}) = \frac{1}{2}\left\|\mathbf{M}\mathbf{W}\mathbf{f} - \mathbf{y}_0\right\|_2^2 + \mu \left\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{W}\mathbf{f} + \mathbf{y}_0)\right\|_p^p, \tag{2.18}$$

where the expression for the regularization term $\mu \|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{W}\mathbf{f} + \mathbf{y}_0)\|_p^p$ derives from (2.16)-(2.17). We will refer to (2.18) as the *implicit formulation*, since the full sinogram $\mathbf{y}$ does not appear explicitly.

On the other hand, we can consider both the data fidelity and the data consistency L^2-norm errors into the objective function to assess if, by explicitly incorporating the extrapolation scheme in the minimization problem, we achieve a more accurate reconstruction. In this case, the image $\mathbf{f}$ to be reconstructed and the full sinogram $\mathbf{y}$ are both unknowns. This second approach yields:

$$\left(\bar{\mathbf{f}},\, \bar{\mathbf{y}}\right) = \underset{\substack{\mathbf{f}\in\Omega_{\mathbf{f}} \\ \mathbf{y}\geq\mathbf{0}}}{\operatorname{argmin}}\, \mathbf{O}(\mathbf{f},\mathbf{y})$$

where

$$\begin{aligned}\mathbf{O}(\mathbf{f},\mathbf{y}) = \frac{1}{2}\left\|\mathbf{MWf}-\mathbf{y}_0\right\|_2^2 + \frac{1}{2}\left\|(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})(\mathbf{Wf}-\mathbf{y})\right\|_2^2 \\ + \mu\left\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})\mathbf{y}+\mathbf{y}_0)\right\|_p^p.\end{aligned} \tag{2.19}$$

In contrast with the above formulation, we will refer to (2.19) as the *explicit formulation*. In both cases, μ denotes the regularization parameter, $p = 1, 2$, and the feasible region $\Omega_{\mathbf{f}}$ is either defined as the non-negativity constraint $\mathbf{f} \geq 0$ or the box constraint $0 \leq \mathbf{f} \leq L_{\max}$, where $L_{\max}$ is the maximum pixel value of the image, and the inequalities are meant component-wise. In the explicit formulation, the full sinogram $\mathbf{y}$ is subject to a non-negativity constraint, too. We point out that the non-negativity constraint must be introduced in the formulation of the previous minimization problems. Indeed, the image and the full sinogram are non-negative given that the image $\mathbf{f}$ is understood as a measure of attenuation and, hence, measures the incoming "number of photons", and the full sinogram $\mathbf{y}$ should preserve the non-negativity under the action of the matrix $\mathbf{W}$. We also notice that, in both formulations, each term of the objective function is convex with respect to the unknowns (it is strictly convex for $p > 1$, and this is particularly relevant when $p = 2$).

Notice that the usefulness of two different objective functions relies on the image reconstruction formulation as a statistical problem (see Chapter 5), whose underlying principle is the presence of noise. Indeed, in the absence of noise, the solution $\bar{\mathbf{f}}$ is the same for both the implicit and the explicit formulations (up to a vector in the kernel of $\mathbf{W}$). However, in the presence of noise, the explicit formulation should benefit from the presence of the norm $\|(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{Wf} - \mathbf{y}\|_2^2$ that is in general non-zero (and can be controlled using an appropriate stopping criterion, typically depending on the noise variance), forcing a more accurate reconstruction *inside* the ROI.

Finally, we take into account a slightly different formulation of both the implicit and the explicit objective function, since it will be considered in the numerical experimentation presented in Chapter 6. To this end, we recall another regularization tool: the Total Variation (TV) functional, with its smoothed version (smooth TV) [142], that is widely used in image reconstruction [27, 108, 16]. In a continuous model, TV is given by the following Definition 2.7.

Definition 2.7 ([1])**.** Given a differentiable function f defined on a bounded open set $\mathcal{A} \subset \mathbb{R}^n$, with $n \geq 2$, the total variation of f has the following expression:

$$\mathrm{TV}(f) = \int_{\mathcal{A}} |\nabla f(x)|\, dx.$$

A more general definition of TV, valid even when f is nonsmooth, can be found, *e.g.*, in [1]. If we restrict ourselves to the $n = 2$ case, $\mathrm{TV}(f)$ can be interpreted geometrically as the lateral surface area of the graph of f. Clearly, if f has many large amplitude oscillations, then it has large lateral surface area, and hence $\mathrm{TV}(f)$ is large. As a consequence, with TV one can effectively reconstruct functions with jump discontinuities, by controlling oscillatory features.
According to the approach used by Vogel [157], TV can be discretize in the 2D case by means of the discrete gradient. Since we still meant the object $\mathbf{f}$ to be a vector, our approach combines Vogel's and the one proposed in [16]. The discrete gradient of $\mathbf{f}$ at the pixel $i = (j_{\mathrm{col}} - 1)N + j_{\mathrm{row}}$, with $j_{\mathrm{col}}, j_{\mathrm{row}} = 1, \ldots, N$ can be defined by the forward finite difference formula by:

$$(\nabla \mathbf{f})_i = \begin{pmatrix} f_{(j_{\mathrm{col}}-1)N + \mathrm{mod}(j_{\mathrm{row}}, N)+1} - f_i \\ f_{\mathrm{mod}(j_{\mathrm{col}}, N)N + j_{\mathrm{row}}} - f_i \end{pmatrix} = \nabla_i \mathbf{f}, \tag{2.20}$$

where $\nabla_i \in \mathbb{R}^{2\times N^2}$, with $i = 1, \ldots, N^2$, is a matrix with only two nonzero entries on each row, equal to -1 and 1. The smoothed version of TV is accomplished by taking into account a small quantity δ to to remove the singularity of the discrete gradient in pixels where $\nabla_i = 0$:

$$\mathrm{TV}_\delta(\mathbf{f}) = \frac{1}{2}\sum_{i=1}^{N^2} \left\| \begin{pmatrix} \nabla_i \mathbf{f} \\ \delta \end{pmatrix} \right\|_2 = \frac{1}{2} \sum_{j_{\mathrm{row}}, j_{\mathrm{col}}=1}^{N} q_\delta(\mathscr{D}^2_{j_{\mathrm{row}}, j_{\mathrm{col}}}) \tag{2.21}$$

where

$$q_\delta(x) = 2\sqrt{x + \delta^2}, \qquad q'_\delta(x) = \frac{1}{\sqrt{x+\delta^2}},$$

and

$$\mathscr{D}^2_{j_{\mathrm{row}}, j_{\mathrm{col}}} = (f_{j_{\mathrm{row}}+1, j_{\mathrm{col}}} - f_{j_{\mathrm{row}}, j_{\mathrm{col}}})^2 + (f_{j_{\mathrm{row}}, j_{\mathrm{col}}+1} - f_{j_{\mathrm{row}}, j_{\mathrm{col}}})^2.$$

Thus, when $\delta = 0$, we are dealing with the unsmoothed version of the TV. Notice that when ∇_i is (much) smaller then δ, the regularization provided by the TV functional approximates Tikhonov regularization in terms of the ℓ_2-norm of the modulus of the discrete gradient. Hence, smooth TV not only works as a edge preserving functional, but the presence of δ also incorporates some information about the smoothness of the object inner surface. Finally, observe that, the TV gradient operator can be understood as a sparsifying transformation, which transforms the original image into an edge map.

In conclusion, our objective functions reads as:

$$\mathbf{O}(\mathbf{f}) = \frac{1}{2}\left\|\mathbf{MWf} - \mathbf{y}_0\right\|_2^2 + \mu\left\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{Wf} + \mathbf{y}_0)\right\|_p^p + \rho\,\mathrm{TV}_\delta(\mathbf{f}) \tag{2.22}$$

for the implicit formulation and

$$\begin{aligned}\mathbf{O}(\mathbf{f},\mathbf{y}) = \frac{1}{2}\left\|\mathbf{MWf} - \mathbf{y}_0\right\|_2^2 + \frac{1}{2}\left\|(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})(\mathbf{Wf} - \mathbf{y})\right\|_2^2 \\ + \mu\left\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{y} + \mathbf{y}_0)\right\|_p^p + \rho\,\mathrm{TV}_\delta(\mathbf{f})\end{aligned} \tag{2.23}$$

for the explicit one. In both formulations, ρ is a regularization parameter.

In Chapter 5, we will investigate and analyze two different approaches (one for $p = 2$ and the other for the nonsmooth case corresponding to $p = 1$) for the solution of the minimization problems introduced in this Chapter.

Chapter 3

Effective distance-driven vectorized computation of the CT forward operator

As pointed out in Chapter 2, the mathematical model of the physical data acquisition process in CT can be discretized, yielding the so-called forward projection matrix. This matrix is a key element in the reconstruction algorithm, and a fast, accurate and memory efficient implementation is necessary since it comes into play at each iteration.

In this Chapter, we address how to give a discrete representation of the object, how to model the forward operator, by exploiting a state-of-the-art algorithm called *distance-driven* and, lastly, how to practically implement it in an efficient and accurate way. A new (and still unpublished) vectorized implementation will be proposed for the 3D cone-beam CT geometry, which applies also to the 2D fan-beam CT geometry, with the necessary arrangements. Before that, the Chapter opens with a literature review on the different generations of CT scanners and on the existing forward models for the forward projection matrix.

To the best of our knowledge, a description of a vectorized computation of the distance-driven technique is not available in the literature, nor a package providing it. Besides this, the few available non-vectorized presentations are in general poor in details and lack of a formal and rigorous mathematical description, which is instead the backbone of this Chapter.

3.1 Generations of CT scanners

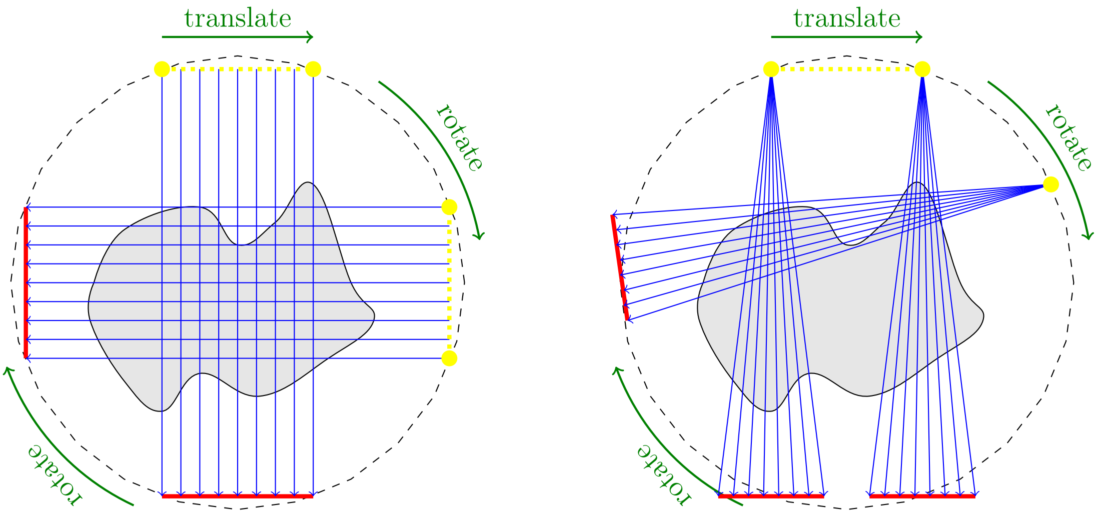
The aim of this section is a self-contained overview on the evolution of CT technology. The main reference is [83]. We will provide a general panel of the different generations of CT scanners, focusing mainly on fan-beam and cone-beam geometries, that are at the core of the numerical assessment in Chapter 6. Here, "generation" relies on both the order in which CT scanner design has been introduced and the arrangement of components and mechanical motion required to collect data.

In the original EMI scanner built by Hounsfield in 1971, only one pencil-like X-ray beam was measured at a time by a single detector cell. The X-ray source and the detector were linearly translated to acquire individual measurements and then rigidly rotated to the next angular position, to acquire the next set of measurements (see Figure 3.1a). In this very first generation of CT, the average duration to collect around 160 measurements across the scan field was about 25-30 minutes, thus producing very poor resolution images due to the patient motion during the acquisition.

The need to reduce the scanning time, led to a second generation of CT scanners. The X-ray source and the detector were still moved according to a translation-rotation dynamics, but the number of rotation steps was reduced by the use of multiple pencil beams, *i.e.*, by adding detectors angularly displaced to obtained several projections in a single translation (see Figure 3.1b). This is the first model of fan-beam geometry for a CT scanner, even if it was not able to cover the whole filed of view (FOV). Second generation CTs raged from the early versions with three detectors, each displaced by one degree, to the last ones, with up to 53 detectors, acquisition during a single breath hold and scans of the trunk.

In the third generation of CT scanner, many detector cells (from the initial 288 to the over 700 of the newer ones) are located on an arc concentric to the X-ray source. Thus, the X-ray source and the detector are kept stationary to each other while the entire apparatus rotates around the patient, and no linear motion is performed. Moreover, each detector FOV is sufficiently large to image the entire object at all times. This is a rotate-rotate dynamics with a fan-beam geometry (see Figure 3.2a). The main breakthrough of the third generation of CT scanner was the introduction of a slip ring technology for the gantry, that accounts for the power and data transmission process, in place of the traditional cables. The slip ring technology not only reduced to around 0.5 sec the scan time but was also a key step to the advent of helical CT. Nowadays, nearly all of the state of the art scanners on the market are third generation.

However, in the third generation, very high performance detectors are needed to avoid ring artifacts and the system is more sensitive to aliasing than the first or second generation scanners. To overcome this limitation, in the fourth generation the detector (with more than 2000 cells) is arranged in an outer ring and remains stationary during the entire scan, while the X-ray tube rotates around the patient (see Figure 3.2b). Unlike the third generation scanner, a projection is measured on a single detector as the X-ray beam sweeps across the object. Thus,



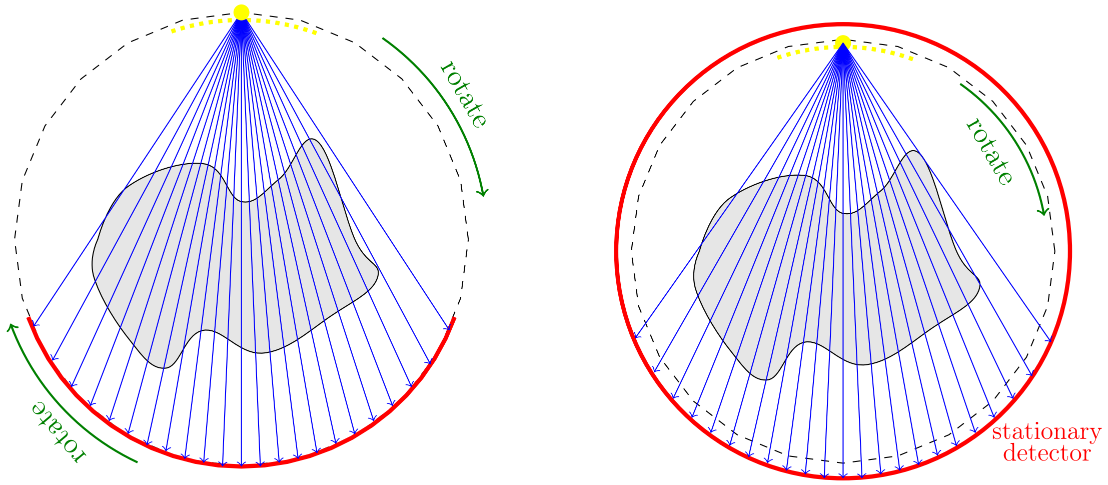
(a) First generation. Parallel-beam, translate-rotate.

(b) Second generation. Fan-beam, translate-rotate.

Figure 3.1: Generations of CT scanners. The yellow spot is the X-ray source, the red lines are the detector arrays.

the geometry is still fan-beam but the dynamics is rotate-fixed and the average duration of a full scan is few seconds. The main advantages of the fourth generation scanners are two. On one hand, potential aliasing artifacts are eliminated due to the spacing between adjacent samples in a projection determined only by the measurements rate, in contrast to the third generation in which the sample spacing is determined by the detector cell size. On the other hand, the detector can be recalibrated dynamically during the scan, since during every rotation, at some point, each detector cell is exposed directly to the X-ray source without any attenuation. Nevertheless, fourth generation scanners are almost unachievable due to economical and practical reasons. Indeed, the number of detectors required to form a complete ring, and the associated data acquisition electronics, is quite large and no effective scatter rejection can be performed. This drawbacks are likely to worsen with the introduction of multislice or volumetric CT.

There exists several other CT scanner geometries that do not precisely fit the above categories. The “cine CT” belongs to the fifth generation: here, instead of the mechanical motion of the X-ray tube, the source rotation is provided by the sweeping motion of the electron beam. The core idea is not allowing any mechanical moving parts to “freeze” cardiac motion. This way, a complete set of projections can be collected within $20ms$ to $50ms$. Even if multiple scans are often needed to average the produced final image, due to noise, cine CT systems



(a) Third generation. Fan-beam, rotate.

(b) Fourth generation. Fan-beam, rotate-stationary detector.

Figure 3.2: Generations of CT scanners. The yellow spot is the X-ray source, the red curvilinear paths are the detector arrays.

are ideal for some clinical application, including cardiac imaging with and without the use of contrast agents, lung imaging, and pediatric studies.

With the advent of sixth generation CT scanners from the 1990s, the advance in technology allows 3D image acquisition within a single breath hold: first and foremost, helical CT. In the sixth generation CTs the X-ray tube rotates as the patient is moved smoothly into X-ray scan field: thus, as the source rotates, the table translates, simultaneously to the data acquisition. The advent of helical CT improved speed, detections, contrast, manipulation and image reconstruction quality.

The seventh generation saw the advent of multiple detector array scanners, still preserving a single X-ray tube, but with a widened X-ray beam in the z-direction to acquire multiple slices simultaneously, with the slice thickness determined by detector size. The geometry is cone-beam: the X-ray source and the detector are mechanically joined and rotate around the object, while multiple sequential planar projection images of the FOV are acquired in a complete arc. While reducing the scanning time, this technique provides double or triple the volume per slice, and each rotation produces its own bank of raw data. Thus, the image quality can be improved, *e.g.*, reducing motion artifacts by averaging data of the multiple rotations. However, the whole machinery is quite expensive and suffer from a less scatter rejection compared to single slice.

3.2 Object discretization by means of series expansion

The first step towards the digital implementation of the forward operator is to choose how to approximate the real, or continuous, object. This is a crucial point because it influences the accuracy of the reconstructed object and, potentially, the speed and computational complexity of the algorithm.

A classical approach is to regard the object as a collection of pixels (voxels in 3D), by sampling the object at evenly spaced points. Generally, this corresponds to approximate the continuous object f with a, possibly linear, series expansion, *i.e.*, with a linear combination of a finite number of basis functions b_i, with $i = 1, \ldots, r$ [74]:

$$f(x) \approx \mathbf{f}(\mathbf{x}) = \sum_{i=1}^{r} f_i b_i(\mathbf{x} - \mathbf{x}_i),$$

where $\{f_i\}_{i=1}^{r}$ is a set of expansion coefficients, not necessary equal to the sampling values of the function f, depending this on the choice for the basis functions. The vectors $\mathbf{x}$ and $\mathbf{x}_i$ denote the current spatial position and the center of the i-th basis function, respectively.

Desirable properties of basis functions include linear independence, shift and rotational invariance, degree of sparsity and small or local support. Clearly, linear independence is necessary to identify uniquely the expansion coefficients while the shift and rotational invariance ensures the independence of the object from the orientation of the underlying grid. The last two desiderata are more subtle.The degree of sparsity concerns the number of basis functions required. Intuitively, a high number of coefficients results in a better approximation than a low number of basis functions, but requires a larger amount of computer storage and computation speed. In practice, one seeks for a tradeoff between accuracy and computational demands.The type of spatial support is closely related to the degree of sparsity. *Global* basis functions have an unlimited support: this is the case of Fourier series, used, *e.g.*, by A. Cormack [37] in his early work, and of the *natural pixels* introduced in [24]. *Local* basis functions, on the contrary, are zero-valued outside a limited support and, hence, are localized in space and typically give rise to sparser approximation than the non-local approach. For this reason local basis functions are usually preferable and different choices exist, including the above mentioned pixels and voxels [11, 146], generalized Kaiser-Bessel functions, also known as *blobs* [114] and B-splines [74]. Also, the local nature of the basis functions reduces the difficulties associated with applying possible local constraint and the computational effort can be minimized by using tabulated values.

In the following, we will only consider the pixels and the voxels basis function for the 2D and the 3D case, respectively. A pixel basis function has a value equal to one inside a local region, described by the corresponding spatial location, and zero outside. That is:

$$b(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} < \dfrac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases}$$

where Δ denotes a grid increment on a regular uniform Cartesian grid. Clearly, they are not permitted to overlap. The definition of voxel is given by analogy in the 3D regular uniform Cartesian grid. It is clear that a function approximated by pixels or voxels is piece-wise constant. Observe that, with this choice for the basis function, rotational invariance is not preserved, but it is quite unlikely that a single set of local basis functions can meet all the above mentioned desiderata. Also, notice that the choice of the basis function is always a compromise between simplicity, accuracy and complexity of practical implementation, since this affects also the choice of the model for the forward operator.

To date, pixels and voxels are still the easiest and most popular choice of basis functions for the object representation, notwithstanding the existence of other attractive alternatives, most notably blobs.

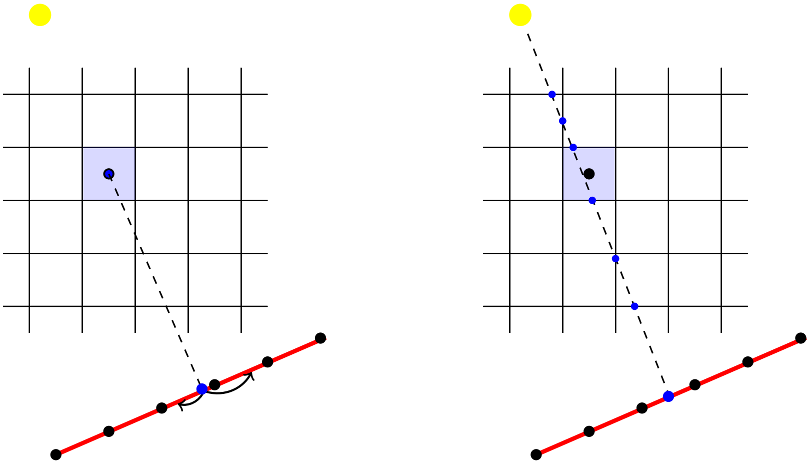
3.3 State-of-the-art approaches for the forward operator

The discretization of the mathematical model of the physical data acquisition process (the "line integral model") is a crucial point in the design of the digital counterpart of the ROI CT problem. There exist many methods for the implementation of the forward projection matrix, based on pixels and voxels basis functions, all of which provide some sort of tradeoff between accuracy and computational complexity.

Notice that, in all approaches, the real-world physical effects are neglected since, in in many practical cases of interest, the complex physical model can be approximated by taking the collections of line integrals and combining them in appropriate ways, thus reducing the problem to the calculation of line integral [45].

Most of method mentioned in the literature can be subdivided into two main categories, that date back to the early days of CT and remained at the basis of all subsequent versions. These are the *pixel-driven* and the *ray-driven* approaches.

(a) *Pixel-driven.* In this approach, the center of each pixel, visited in a loop, is projected onto the detector, and the contribution of each pixel is split between two neighboring detector cells by using interpolation, as shown in Figure 2.1a. Observe that, when the detector cell size is much smaller than

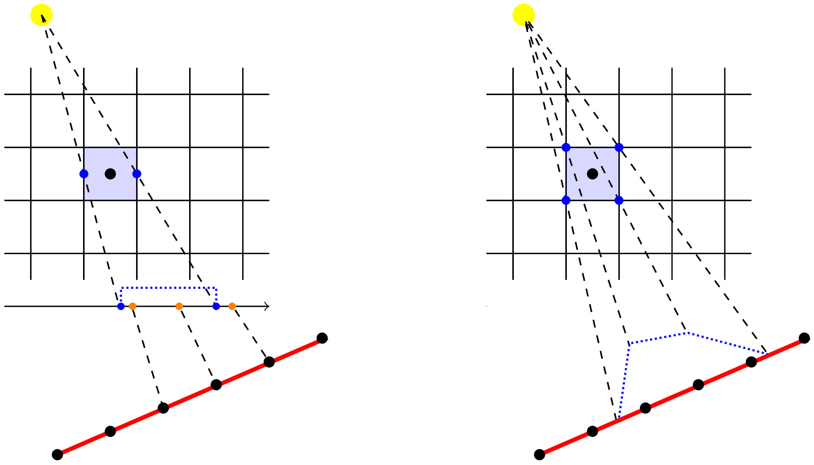


(a) Pixel-driven (b) Ray-driven

Figure 3.3: State-of-the-art approaches for the forward operator.

the pixel element size, it is highly likely that in some detector cells no value is written. This results in high-frequency artefacts, and this is the main reason why pixel-driven approaches are rarely used [45]. These artefacts can be prevented by using, *e.g.*, a smooth footprint for each pixel as in the *splatting* technique [124], or enabling higher accuracy by subdividing each pixel in smaller parts. However, all modifications introduced into the original algorithm are generally at the expense of an increased computational complexity, and even pre-calculating and storing the weights becomes impractical for large datasets [54]. Notice that the pixel-driven principle is the simplest approach and it is the one typically implemented for FBP reconstructions [76, 132, 170].

(b) *Ray-driven.* This method works by connecting the X-ray source to the center of each detector cell, as depicted in Figure 2.1b. The projection value is calculated essentially by discretizing the line integral directly, *i.e.*, each line integral is approximated with a weighted sum of all image pixels that lie close to the ideal line. Among the many different techniques to select and calculate these weights, there are two classical ray-driven approaches. In Joseph's algorithm [86], each weight results from the intersection length of the ray with the corresponding pixel, while Siddon's algorithm [148] performs linear interpolation between two pixels in a row (or column) that intersects the projection line. In contrast to pixel-driven methods, ray-driven approaches tend to introduce artefacts, the so-called *Moiré patterns*, in the backprojection



(a) Distance-driven (b) Separable footprint

Figure 3.4: State-of-the-art approaches for the forward operator.

operation. A variant of ray-driven approaches, in which the ray is assumed to have a width usually equal to the detector cell size, is called *strip-integral* method. In this technique, the contribution from a pixel to the detector cell corresponds essentially to the integral over the "true" strip area that results from the intersection between the ray and the pixel [116].

Many variants of pixel-driven and ray-driven approaches have been suggested, such as nearest-neighbour based methods or the shear-warp volume rendering algorithm [107]. Also, Fourier-based projection methods, that exploits the fast Fourier transform and then interpolates the Fourier samples onto a polar grid, has been proposed [123, 168]. All these approaches are conceptually straightforward to generalize to the 3D voxel case.

Recently, methods with a fundamentally different approach have been introduced, namely the *distance-driven* [44, 45] and the *separable trapezoid footprints* algorithms [119, 120]. In these methods, the pixel and detector boundaries are considered instead of the pixel and detector center, and the pixel footprint is approximated by a rectangle, in the distance-driven method, and by a trapezoid footprints, to shape a more accurate pixel footprint, in the separable footprint technique. This is illustrated in Figure 3.4. These approaches have a low arithmetic cost and avoid high-frequency artifacts in both the forward and backprojector operators.

Since the distance-driven method is the main topic of this Chapter and the numerical experiments reported in Chapter 6 are performed by using an efficient

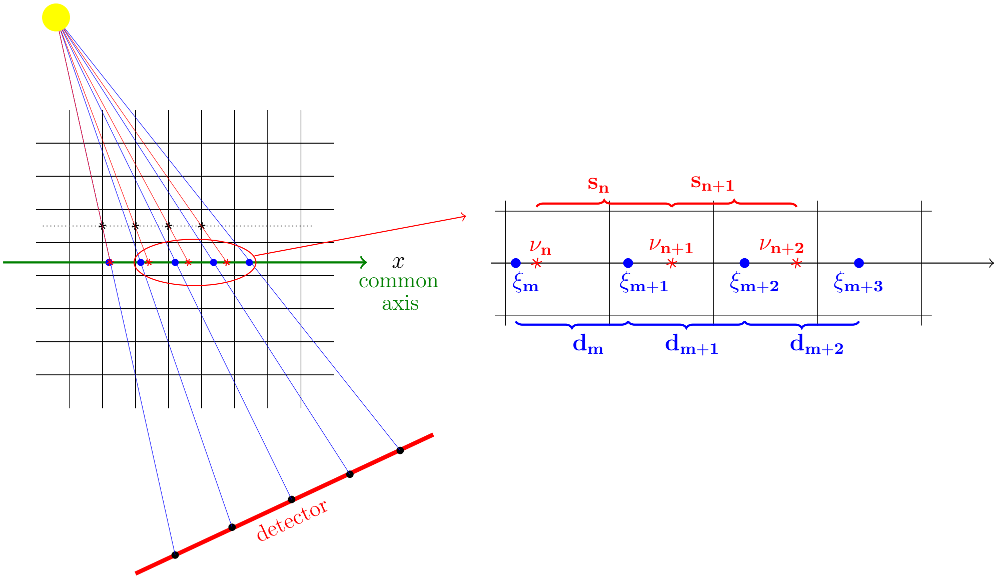


Figure 3.5: The distance-driven method as a 1D re-sampling problem.

implementation of the distance-driven approach for the forward projection matrix, we present in detail the main ideas of the method in the next Subsection.

3.3.1 The distance-driven technique

The distance-driven method is essentially based on converting the projection problem into a 1D re-sampling problem (see Figure 3.5). There are two main ingredients. The former one is the *kernel operation*:

$$\begin{gathered} d_m = \sum_n w_{mn} s_n \qquad \text{with} \\ w_{mn} = \frac{\left[\min(\xi_{m+1}, \nu_{n+1}) - \max(\xi_m, \nu_n)\right]_+}{\xi_{m+1} - \xi_m}\,, \quad [x]_+ = \max(x, 0)\,, \end{gathered} \tag{3.1}$$

which allows one to compute the destination signal values $\{d_j\}_j$ from the sample values $\{s_i\}_i$ of a source signal, the sample source locations $\{\xi_j\}_j$ and the sample destination locations $\{\nu_i\}_i$. The second element of the method is that there is a (possibly zero) length of overlap between each image pixel and each detector cell due to the bijection between the position on the detector and the position within an image row (or column). Thus, every point within an image row is *uniquely* mapped onto a point on the detector, and vice versa. In practice, to compute the overlap length, all pixel boundaries in an image row and all detector cell boundaries are mapped onto a common line, *e.g.*, a line parallel to a coordinate axis. That is, each length of overlap is the interval length between two adjacent intersections, obtained by connecting the boundaries midpoints of all detector cells and pixels in a image row to the X-ray source and by computing the intercepts of these lines with the common axis. The final weights are achieved by normalizing the overlap length by the detector cell width. This corresponds exactly to apply the kernel operation from equation (3.1).

For instance, in the ROI CT problem, d_j is the theoretical (unblurred and noiseless) value measured at the j-th detector cell and s_i is the estimate of the i-th pixel attenuation function. Hence, if the $(m+1)$-th detector cell is "shadowed" by only two pixels in a row, equation (3.1) reads as:

$$d_{m+1} = \frac{\nu_{n+1} - \xi_{m+1}}{\xi_{m+2} - \xi_{m+1}} s_n + \frac{\xi_{m+2} - \nu_{n+1}}{\xi_{m+2} - \xi_{m+1}} s_{n+1}.$$

The distance-driven technique can be extended quite naturally to the 3D case. The key element is still an overlap between a point in the object and a point on the detector, but the bijection is now between a position on the 2D detector and the position within each object *slab*. It still holds that every point within an object slab is mapped *uniquely* onto a point on the detector, and vice versa. In practice, all the horizontal and vertical midpoint boundaries of the object voxels and of the detector cells are mapped onto a common plane (*e.g.*, a plane parallel to a coordinate plane) and this determines an area of overlap. Then, one calculates the lengths of overlap along the directions in an object slab (as in the 2D case) and multiplies them to get the area of overlap.

In the next Section we shall see that an efficient implementation of the distance-driven method can be achieved by using a vectorized approach instead of processing one pixel or voxel at a time. The possibility to avoid (nested) loops makes the method faster.

Finally, notice that the distance-driven method is somehow a special case of separable footprint approach, that occurs when the generic trapezoids footprint is a rectangle. However, in the latter, the footprint functions are approximated as 2D separable functions and this simplifies the calculation of their integrals. This is a "hot topic" in the field of X-ray tomographic imaging and there is a lot of ongoing research [160, 94, 95].

3.4 Vectorized approach to the distance-driven technique

In this Section we describe how the forward projection matrix can be built up by means of a *vectorized* approach. Here, we describe the more demanding and burdening 3D case since the 2D case can be easily retrieved by simply discarding the third (second, respectively) dimension for the object (detector, respectively). The main result, which states that is possible to vectorize the distance-driven technique, is Theorem 5, reported at the end of this Section.

3.4.1 Cone-beam 3D circular geometry

In this Subsection we analyze the geometry of the 3D projection system. These geometrical preliminaries are fundamental to clarify the general setup, because the mathematical results on the vectorization of the distance driven technique we propose, rely on it. To this end, we start by introducing a sufficiently general formal *geometric* definition (see Figure 3.7).

Definition 3.1. Given two points $C, S \in \mathbb{R}^3$ and the real numbers d, $\gamma_{\min}$, $\gamma_{\max}$, $\beta_{\min}$ and $\beta_{\max}$ with $d \geq 0$, $-\pi/2 < \gamma_{\min} \leq 0 \leq \gamma_{\max} < \pi/2$ and $-\pi/2 < \beta_{\min} \leq 0 \leq \beta_{\max} < \pi/2$, a *cone-beam circular rotation system* $\mathcal{R}_{\mathcal{C}}\big(C, S, d, \gamma_{\min}, \gamma_{\max}, \beta_{\min}, \beta_{\max}\big)$ is a set such that:

a) S rotates on a circular trajectory with center C and radius $\|S - C\|_2$;

b) a plane π_{dtc}, orthogonal to the rotation plane of S and to the radius through S, rotates on another coplanar circular trajectory with the same center C and radius d, rigidly joined to S on the diametrically opposite position with respect to C;

c) a rectangular cone $\mathcal{C}$ is considered, having its vertex at S, its axis matching the diameter from S to π_{dtc}, the two faces orthogonal to the trajectory plane located at angles $\gamma_{\min}$ and $\gamma_{\max}$ from its axis, and the two faces parallel to the trajectory plane located at angles $\beta_{\min}$ and $\beta_{\max}$ from its axis.

The point S is called *beam source* (or simply *source*), the point C is the *rotation center* or *isocenter*, d is the *center-to-detector distance*, and π_{dtc} is the *detector plane*. The angles $\gamma_{\min}, \gamma_{\max}$ are the *minimum* and the *maximum aperture angle*, respectively, and $\beta_{\min}, \beta_{\max}$ are the *minimum* and the *maximum elevation angle*, respectively.

A cone-beam circular rotation system with $-\gamma_{\min} = \gamma_{\max} = \gamma$ is called an *aperture-omogeneous* system, and shall be denoted by $\mathcal{R}^{\text{ao}}_{\mathcal{C}}\big(C, S, d, \gamma, \beta_{\min}, \beta_{\max}\big) = \mathcal{R}_{\mathcal{C}}\big(C, S, d, \gamma, \gamma, \beta_{\min}, \beta_{\max}\big)$. Analogously, when $-\beta_{\min} = \beta_{\max} = \beta$ the cone-beam circular rotation system is called *elevation-omogeneous* and shall be denoted by $\mathcal{R}^{\text{eo}}_{\mathcal{C}}\big(C, S, d, \gamma_{\min}, \gamma_{\max}, \beta\big) = \mathcal{R}_{\mathcal{C}}\big(C, S, d, \gamma_{\min}, \gamma_{\max}, \beta, \beta\big)$. A cone-beam circular rotation system $\mathcal{R}^{\text{rct}}_{\mathcal{C}}\big(C, S, d, \gamma, \beta\big) = \mathcal{R}_{\mathcal{C}}\big(C, S, d, \gamma, \gamma, \beta, \beta\big)$ which is both aperture-omogeneous and elevation-omogeneous is called a *rectangular* system. Finally, a rectangular cone-beam circular rotation system with $\gamma = \beta$ is called a *square* system and shall be indicated with $\mathcal{R}_{\mathcal{C}}^{\mathrm{sqr}}\big(C, S, d, \gamma\big)$.

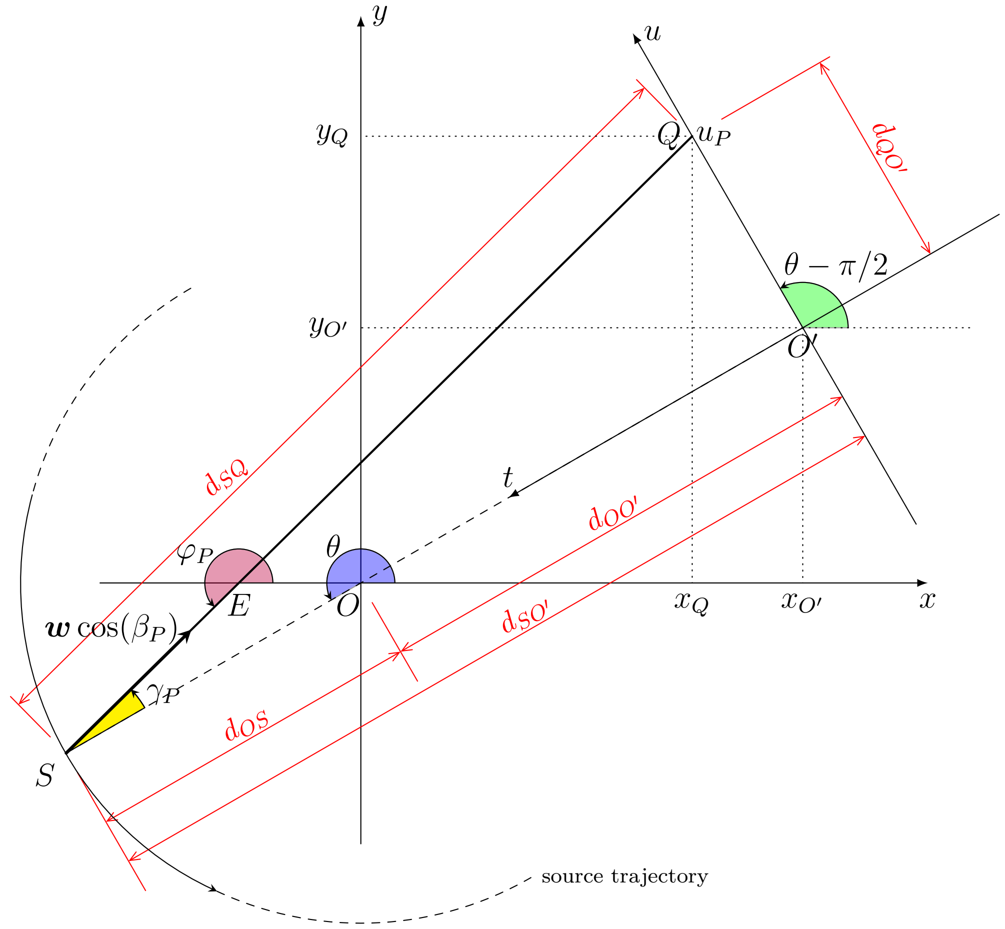


Figure 3.6: 2D geometric representation of the generic point Q on the detector, for a given position of the source S.

Starting from Definiton 3.1, we can set up the coordinate systems in $\mathcal{R}_{\mathcal{C}}$. We consider a general cone-beam circular rotation system with the standard Cartesian coordinate system $Oxyz$ as main system, the center located at the main origin O, and the source S running on a circle on the Oxy plane. In the following, this main system shall be also referred to as *inertial system*.

The *detector* is a 2D rectangular subset of a plane π_{dtc} orthogonal to the source rotation plane Oxy. By definition, this orthogonal plane rotates rigidly fixed to the source and it is always positioned at the opposite side with respect

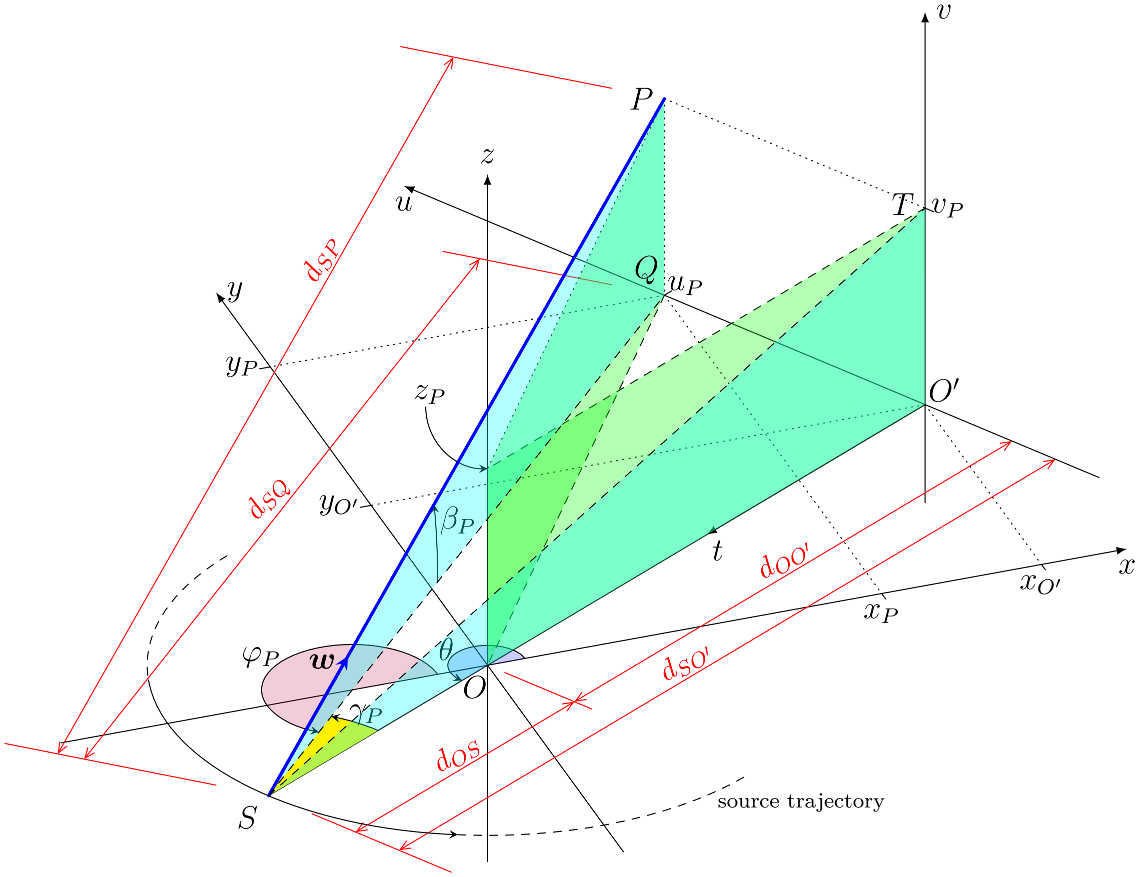


Figure 3.7: 3D geometric representation of the generic point P on the detector, for a given position of the source S.

to the z axis, at a constant distance from S. We consider a second 3D Cartesian coordinate system $O'utv$, rigidly joined to the detector plane, which shall be referred to as the *roto-translated coordinate system*. This coordinate system has its center O' positioned on the line going from S through O, at distance $d_{OO'} = \|O' - O\|_2$ from O. Hence, by using the notation introduced in Definiton 3.1, $\mathcal{R}_{\mathcal{C}} = \mathcal{R}_{\mathcal{C}}\big(O, S, d_{OO'}, \gamma_{\min}, \gamma_{\max}, \beta_{\min}, \beta_{\max}\big)$. Notice that the axes orientation of the roto-translated system are such that the v axis is parallel and concordant to the z axis, whilst the u and t axes belong to the source rotation plane, with t going from O' to O and u on the rotated plane π_{dtc}. The versus of u makes the roto-translated system a right-handed one, *i.e.*, when t overlaps y, u is concordant with x. For simplicity, in the following we consider on π_{dtc} the 2D (sub)system $O'uv$, as depicted in Figure 3.7, given that all the points on the detector have $t = 0$ in the roto-translated coordinate system. It is clear from above that the source position is the point S with coordinates $(x_S, y_S, 0)^T$ in the main coordinate system.

Its distance from O, which is also the rotation radius, is $d_{OS} = \|S - O\|_2$, under a rotation angle θ measured counter-clockwise from the x axis. In a standard polar coordinate system $O\rho\phi\psi$, where $\phi \in [0, 2\pi[$ denotes the azimuth angle (measured counter-clockwise from the positive x axis) and $\psi \in [0, \pi[$ the polar angle (measured starting from the positive z axis), the polar coordinates of the source are $(\rho_S, \phi_S, \psi_S)^T = (d_{OS}, \theta, \pi/2)^T$.

Now, consider a generic ray $\overline{SP}$ from the source S to a generic point $P = (u_P, v_P)^T$ on the detector plane π_{dtc} (see Figure 3.7). We aim at identifying the relations connecting the coordinates of P and S, by following an approach similar to the one proposed in [120]:

$$\begin{aligned}
\begin{pmatrix} x_S \\ y_S \end{pmatrix} &= d_{OS} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} & z_S &= 0, \\
\begin{pmatrix} x_{O'} \\ y_{O'} \end{pmatrix} &= d_{OO'} \begin{pmatrix} \cos(\theta + \pi) \\ \sin(\theta + \pi) \end{pmatrix} = d_{OO'} \begin{pmatrix} -\cos(\theta) \\ -\sin(\theta) \end{pmatrix} & z_{O'} &= 0, \\
\begin{pmatrix} x_P \\ y_P \end{pmatrix} &= \begin{pmatrix} x_{O'} \\ y_{O'} \end{pmatrix} + u_P \begin{pmatrix} \cos(\theta - \pi/2) \\ \sin(\theta - \pi/2) \end{pmatrix} = \begin{pmatrix} -\cos(\theta) & \sin(\theta) \\ -\sin(\theta) & -\cos(\theta) \end{pmatrix} \begin{pmatrix} d_{OO'} \\ u_P \end{pmatrix} & z_P &= v_P,
\end{aligned}$$

where we exploited the addition and subtraction rules for the sine and the cosine of a given angle. The angles of interest for the ray $\overline{SP}$ from S to the generic point P on the detector are the following:

$$\begin{aligned}
\gamma_P &= \gamma(u_P) = \arctan\left(\frac{u_P}{d_{SO'}}\right) \\
\varphi_P &= \varphi(u_P, \theta) = \theta + \gamma(u_P) \\
\beta_P &= \beta(u_P, v_P) = \arctan\left(\frac{v_P}{\sqrt{d_{SO'}^2 + u_P^2}}\right)
\end{aligned}$$

where $\gamma_P \in [\gamma_{\min}, \gamma_{\max}]$ is the *ray aperture angle*, namely the angle between the ray $\overline{SP}$ and the line $\overline{SO'}$ through S orthogonal to the detector (it is positive counter-clockwise); φ_P is the *ray rotation angle* in the Oxy plane, measured counter-clockwise starting from the x axis, as the angle θ of the source trajectory; β_P is the *ray elevation angle*, *i.e.*, the angle between the ray $\overline{SP}$ and its projection onto the Oxy plane, with $\beta_P \in [\beta_{\min}, \beta_{\max}]$, measured from the Oxy plane with positive direction matching the positive direction of the z axis. Observe that β_P is connected to the polar angle of P by the equivalence $\psi_P = \pi/2 - \beta_P$. The unit (direction) vector $\boldsymbol{w}$, from the source S to the generic point P on the detector, is given by:

$$\begin{aligned}\boldsymbol{w} = \frac{P-S}{\|P-S\|_2} &= \frac{1}{d_{SP}}\left\{\begin{pmatrix}\begin{pmatrix}-\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & -\cos(\theta)\end{pmatrix}\begin{pmatrix}d_{OO'}\\ u_P\end{pmatrix}\\ v_P\end{pmatrix} - d_{OS}\begin{pmatrix}\cos(\theta)\\ \sin(\theta)\\ 0\end{pmatrix}\right\}\\ &= \frac{1}{d_{SP}}\begin{pmatrix}-d_{SO'}\cos(\theta)+u_P\sin(\theta)\\ -d_{SO'}\sin(\theta)-u_P\cos(\theta)\\ v_P\end{pmatrix},\end{aligned}$$

where $d_{SO'} = d_{OS} + d_{OO'} = \|O' - S\|_2$ and $d_{SP} = \|P - S\|_2$. Now, observe that

$$\begin{aligned}\cos(\gamma_P) &= \frac{d_{SO'}}{d_{SQ}} = \frac{d_{SO'}}{d_{SP}\cos(\beta_P)} &&\Rightarrow \quad \frac{d_{SO'}}{d_{SP}} = \cos(\beta_P)\cos(\gamma_P)\\ \sin(\gamma_P) &= \frac{u_P}{d_{SQ}} = \frac{u_P}{d_{SP}\cos(\beta_P)} &&\Rightarrow \quad \frac{u_P}{d_{SP}} = \cos(\beta_P)\sin(\gamma_P)\\ \tan(\beta_P) &= \frac{v_P}{\sqrt{d_{SO'}^2+u_P^2}} &&\Rightarrow \quad \sin(\beta_P) = \frac{v_P\cos(\beta_P)}{\sqrt{d_{SO'}^2+u_P^2}} = \frac{v_P\cos(\beta_P)}{d_{SQ}} = \frac{v_P}{d_{SP}}\end{aligned}$$

where $d_{SQ} = \|Q - S\|_2 = d_{SP}\cos(\beta_P)$. This yields:

$$\frac{1}{d_{SP}}\begin{pmatrix}-d_{SO'}\cos(\theta)+u_P\sin(\theta)\\ -d_{SO'}\sin(\theta)-u_P\cos(\theta)\\ v_P\end{pmatrix} = \begin{pmatrix}-\cos(\beta_P)\cos(\gamma_P)\cos(\theta)+\cos(\beta_P)\sin(\gamma_P)\sin(\theta)\\ -\cos(\beta_P)\cos(\gamma_P)\sin(\theta)-\cos(\beta_P)\sin(\gamma_P)\cos(\theta)\\ \sin(\beta_P)\end{pmatrix}$$

By recalling that $\varphi_P = \theta + \gamma_P$, one has:

$$\begin{aligned}\cos(\theta)\cos(\gamma_P) - \sin(\theta)\sin(\gamma_P) &= \cos(\theta+\gamma_P) = \cos(\varphi_P)\\ \sin(\theta)\cos(\gamma_P) + \cos(\theta)\sin(\gamma_P) &= \sin(\theta+\gamma_P) = \sin(\varphi_P),\end{aligned}$$

that finally yields:

$$\boldsymbol{w} = \frac{P-S}{\|P-S\|_2} = \begin{pmatrix}-\cos(\beta_P)\cos(\varphi_P)\\ -\cos(\beta_P)\sin(\varphi_P)\\ \sin(\beta_P)\end{pmatrix}.$$

Notice that the above computations can be easily generalized to the case in which the source S lies on a plane parallel to Oxy. In this case, we simply have that $z_S \neq 0$, and, according to Definition 3.1, also $z_{O'} \neq 0$. Now, consider a 3D object located in the coordinate systems $\mathcal{R_C}$ between the source S and the detector plane π_{dtc} and a generic point A on this object. We aim at computing the coordinates of the generic point A with respect to a rotated system $Ou't'v'$, which is the detector coordinate system $O'utv$ shifted to the main origin O. By observing Figure 3.8,

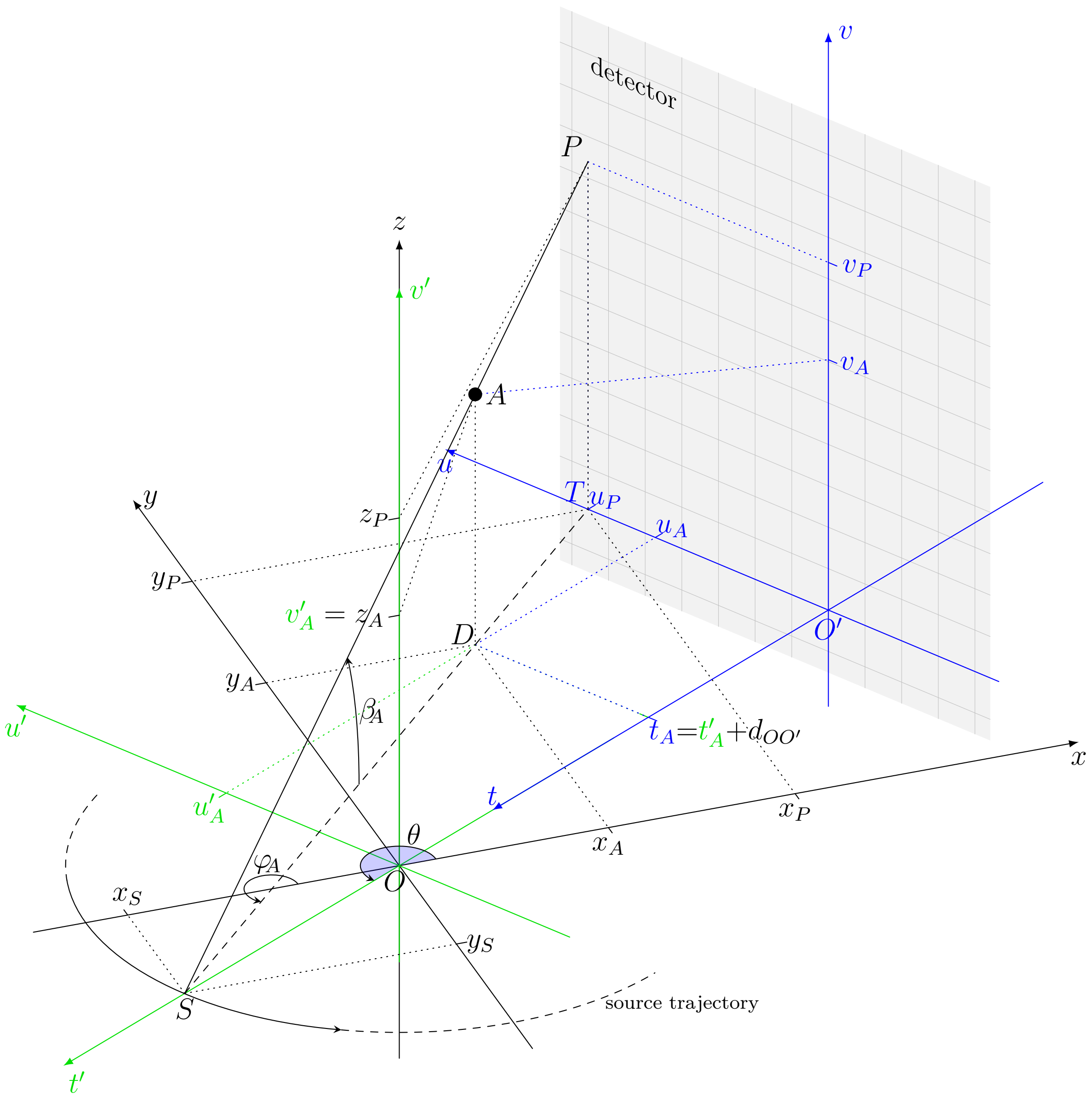


Figure 3.8: 3D geometric representation of the projection P on the detector of a generic object point A, for a given position of the source S.

it is clear that $u'_A = u_A$, $v'_A = v_A = z_A$ and $t'_A = d_{OO'} - t_A$ for every point A in the domain, where D is the projection of the point A onto the Oxy-$Ou't'$ plane. Hence, it is clear that the change of coordinates is the following:

$$\begin{pmatrix} u'_A \\ t'_A \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_A \\ y_A \end{pmatrix},$$

where the matrix is the usual (planar) coordinate system rotation matrix from the inertial system Oxy to the only rotated one $Ou't'$. If we now consider the similar rectangular triangles $SO'T$ and SFD in Figure 3.8, the following relations hold true:

$$\begin{aligned} \overline{O'T} : \overline{FD} = \overline{SO'} : \overline{SF} \quad &\Rightarrow \quad u_P(\theta, x_A, y_A) : u'_A = d_{SO'} : (d_{OS} - t'_A) \\ &\Rightarrow \quad u_P(\theta, x_A, y_A) = d_{SO'} \frac{u'_A}{d_{OS} - t'_A} = d_{SO'} \frac{u'_A}{d_S(\theta, x_A, y_A)} \end{aligned}$$

where $d_S(\theta, x_A, y_A) = d_{OS} - t'_A$ is the distance from the source S of the orthogonal projection F of the point E (and thus of A) onto the ray $\overline{SO'}$ from the source S through the rotation center C to the detector. By observing that $\overline{AD} = \overline{EF} = z_A$, $\overline{PT} = \overline{UO'} = v_P$ and that the two rectangular triangles SFE and $SO'U$ are similar, we can derive the coordinate of v_P:

$$\begin{aligned} \overline{UO'} : \overline{EF} = \overline{SO'} : \overline{SF} \quad &\Rightarrow \quad v_P(\theta, x_A, y_A, z_A) : z_A = d_{SO'} : d_S(\theta, x_A, y_A) \\ &\Rightarrow \quad v_P(\theta, x_A, y_A, z_A) = d_{SO'} \frac{z_A}{d_S(\theta, x_A, y_A)} \end{aligned} \tag{3.2}$$

Thus, with the source in position S, the generic point $A = (x_A, y_A, z_A)^T$ on the object is projected onto the point $P = (u_P, 0, v_P)^T = \big(u_P(\theta, x_A, y_A), 0, v_P(\theta, x_A, y_A, z_A)\big)^T$ of the detector, with respect to the coordinates system $O'utv$. The angles of interest for the ray $\overline{SP}$ going from the source to the detector through the object point A are the following:

$$\begin{aligned} \varphi_A &= \varphi(\theta, x_A, y_A) = \theta + \gamma_A = \theta + \arctan\left(\frac{\overline{FD}}{\overline{SF}}\right) = \theta + \arctan\left(\frac{u'_A}{d_S(\theta, x_A, y_A)}\right) \\ \beta_A &= \beta(\theta, x_A, y_A, z_A) \\ &= \arctan\left(\frac{\overline{AD}}{\overline{SD}}\right) = \arctan\left(\frac{z_A}{\sqrt{\overline{SF}^2 + \overline{FD}^2}}\right) = \arctan\left(\frac{z_A}{\sqrt{d_S^2(\theta, x_A, y_A) + (u'_A)^2}}\right) \end{aligned}$$

From the previous discussion, it is easy to show that the following result holds true.

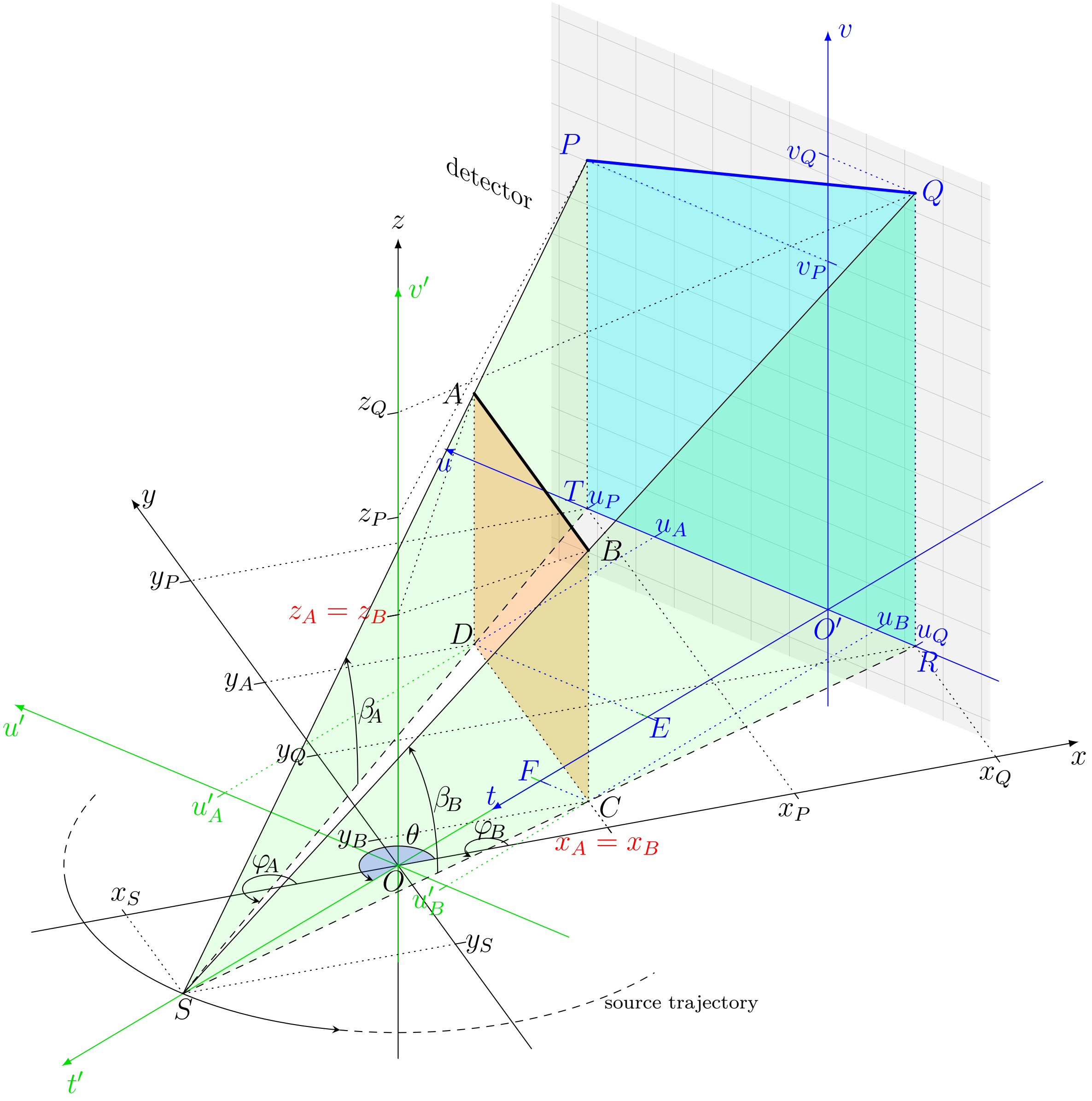


Figure 3.9: 3D geometric representation of the projected segment $\overline{PQ}$ of an object segment $\overline{AB}$, for a given position of the source S, when the detector plane is not parallel to $\overline{AB}$.

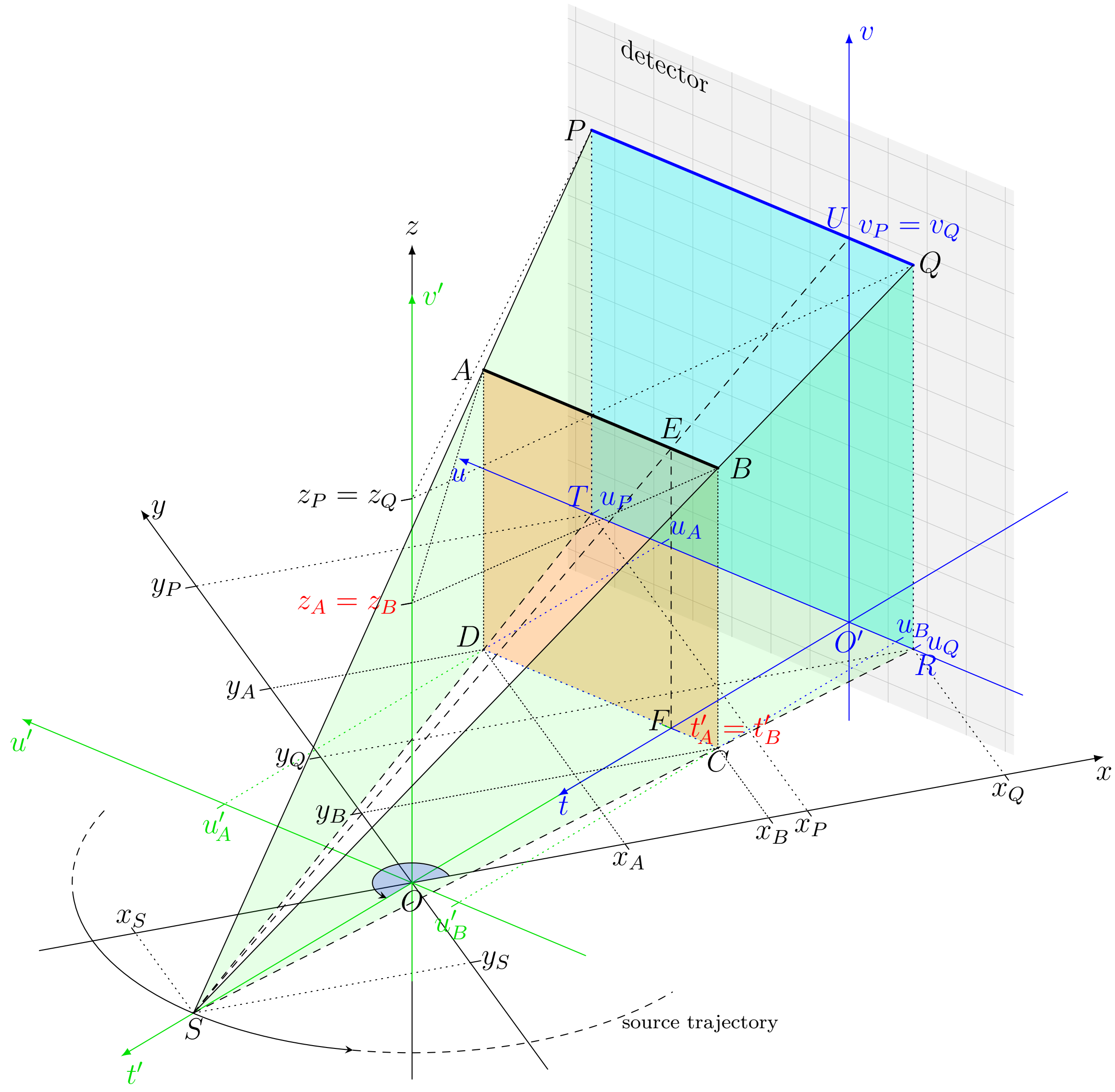


Figure 3.10: 3D geometric representation of the projected segment $\overline{PQ}$ of an object segment $\overline{AB}$, for a given position of the source S, when the detector plane is parallel to $\overline{AB}$.

Proposition 1. *Let $\mathcal{R}_{\mathcal{C}}\big(O, S, d_{OO'}, \gamma_{\min}, \gamma_{\max}, \beta_{\min}, \beta_{\max}\big)$ be a generic cone-beam circular rotation system, endowed with a main coordinate system $Oxyz$ and a rotating coordinate system $O'utv$. Then, the projection onto the detector plane π_{dtc} of a line segment $\overline{AB}$ parallel to the u axis is a line segment $\overline{PQ}$ in the $O'uv$ plane which is still parallel to the u axis.*

Proof. The thesis immediately follows from equation (3.2) for v_P and v_Q. Indeed, given that, by hypothesis, $t'_A = t'_B$, one has

$$d_S(\theta, x_A, y_A) = d_{OS} - t'_A = d_{OS} - t'_B = d_S(\theta, x_B, y_B),$$

that yields

$$v_P(\theta, x_A, y_A, z_A) = d_{SO'} \frac{z_A}{d_S(\theta, x_A, y_A)} = d_{SO'} \frac{z_B}{d_S(\theta, x_B, y_B)} = v_Q(\theta, x_B, y_B, z_B), \tag{3.3}$$

given that $z_A = z_B$ by hypothesis. □

The equivalence (3.3) from Proposition 1 does not hold true in general, *i.e.*, when the projection plane is not parallel to the projecting object segment, even if its endpoints have the same elevation. This can be easily observed by simply comparing Figure 3.9 and Figure 3.10.

Corollary 3.2. *Retaining the hypothesis of Proposition 1, let $\overline{AB}$ be a line segment parallel to the u axis. Then, its projection onto every projection plane parallel to the plane containing the object segment (and parallel to the detector plane π_{dtc}) is a line segment that is still parallel to the u axis.*

Proof. Observe that v_P and v_Q in equation (3.3) depend on z_A, z_B and t'_A, t'_B, but do not dependent on u'_A, u'_B. In addition, the distance $d_{SO'}$ remains constant for A and B, and, actually, for all the points on the line through them. Then, it is clear that the parallel condition of a segment and its projection holds true on every projection plane parallel to the plane containing the object segment. □

This final result will be crucial in the following. Indeed, Corollary 3.2 is the reason why projecting onto a plane parallel to the faces and the edges of the object discretization grid is preferable to projecting onto the detector plane, as it is usually the case.

3.4.2 3D vectorized generation of the distance-driven projection matrix

We now describe how the projection matrix $\mathbf{W}$ of the discretized volume enclosing the object can be built up using a *vectorized* approach, by means of Matlab vector instructions. As outlined in Subsection 3.3.1, each element of the CT forward projection matrix is proportional of the "overlap area" between the projections of a given detector cell (which provides the row index) and a given voxel in the object space (which provides the column index), at a given system rotation angle. As we shall see, by choosing a parallelepipedal voxel discretization for the object space, a rectangular grid discretization for the detector cells, and a coordinated plane as common projection plane, we can suitably approximate the voxel projection, as well as the detector cell projection, by means of rectangular areas. In this way, also the approximate projections overlaps are rectangular areas (possibly empty or degenerate), being intersections of rectangular areas. The baseline idea of our vectorized implementation is to construct, for each system rotation angle, two matrices with as many rows as the number of detector cells, and as many columns as the number of object space voxels: the matrix $\mathbf{W}_\mathrm{H}$ of the "horizontal" edges of the overlap areas, and the matrix $\mathbf{W}_\mathrm{V}$ of their "vertical" edges. Both matrices are efficiently built in a *vectorized fashion*. At the very end, each element of the CT forward projection matrix $\mathbf{W}$ is computed by a simple component-wise multiplication of $\mathbf{W}_\mathrm{H}$ and $\mathbf{W}_\mathrm{V}$, that, afterwards, it is positioned correspondingly to the current rotation angle. All these matrices are *sparse* arrays, and both $\mathbf{W}_\mathrm{H}$ and $\mathbf{W}_\mathrm{V}$ are overwritten at each angle step.

In this Subsection, sometimes arithmetic operations and elementary functions involving arrays operands are intended in the Hadamard sense, *i.e.*, component-wise: when it is not clear from the context, it is explicitly stated. To address subarrays, in place of a subscript index we use the symbol ":" to select all the indices in that dimension; the notation "$i_1 : i_2$", $i_1, i_2 \in \mathbb{N}$, shall be used to select all the contiguous, sorted indices $\{i_1,\ i_1{+}1,\ i_1{+}2,\ \dots, i_2{-}1,\ i_2\}$ in that dimension; finally, we use the subset notation $\mathcal{I} \subset \{1, \dots, n\}$ to select non-contiguous, and possibly unsorted, indices in that dimension, where n denotes the number of array elements in that dimension. Clearly, if $i_1 > i_2$ or $\mathcal{I} = \emptyset$ in all the previous cases, the resulting subarray is meant to be empty.

Remark 3.3. Notice that, to perform effective computations, we essentially need to "vectorize" the arrays containing points of interest in the object volume and the detector, as well as their projections onto π_{prj}. It is well known that, usually, there are two ways to do it, generally referred to as *column-major* and *row-major* reorderings, respectively[1]. If we consider a generic 3D array $A = \big(a_{ij\ell}\big)$ sized

[1] *Column-major* and *row-major* reorderings are historically connected with the way in which the elements of a multidimensional array A are stored in the computer memory, depending on the computer language used to write the program. All versions of Fortran use a column-major reordering, while all versions of C/C++ use the row-major reordering. For this reason, they are sometimes referred to as *Fortran-like* and *C-like* reorderings, repsectively. In the early days, Matlab was built on top of the traditional Fortran BLAS and LAPACK (LINPACK + EISPACK) routines packages, so it came with column-major array vectorization. In the latest versions of

$m \times n \times k$, for each ℓ-th layer, $\ell = 1, \ldots, k$, the column-major reordering stacks in the column vector $\boldsymbol{b}$ the $(j+1)$-th column under the j-th one, for $j = 1, \ldots, n$, while the row-major reordering stacks the $(i+1)$-th row, transposed, under the i-th row, transposed. Recall that the connections between the subscripts i, j and ℓ of the generic element $a_{ij\ell}$ and its "linear index" p in $\boldsymbol{b}$ are the following:

- column-major reordiering:

$$\begin{aligned} &p = (\ell-1)mn + (j-1)m + i \\ &\ell = [p/(mn)] + 1,\ j = [p/m] - (\ell-1)n + 1,\ i = \mathrm{rem}(p, m) \end{aligned} \tag{3.4}$$

- row-major reordiering:

$$\begin{aligned} &p = (\ell-1)mn + (i-1)n + j \\ &\ell = [p/(mn)] + 1,\ i = [p/n] - (\ell-1)m + 1,\ j = \mathrm{rem}(p, n) \end{aligned} \tag{3.5}$$

where $[x]$ is the integer part of x and the $\mathrm{rem}(q, d)$ function returns the remainder of the integer division of q by d. Given that Matlab uses the column-major reordering, in the following we assume the $\mathrm{vect}(\cdot)$ operator is compliant with (3.4).

General setup and initialization

As geometric environment, we consider a cone-beam circular rotation system where the object is assumed to be completely embedded in a rectangular parallelepipedal volume, whose barycenter is the rotation center (in the following also referred to as *isocenter*) C of the acquisition system. The source S, emitting X-rays, is assumed to move on a planar circular trajectory around the object volume, at a distance d_{SC} from the isocenter, with the rotational axis parallel to one of the three parallelepiped axes, usually the largest one, according to Definition 3.1. The X-ray source is located at the distance $d_{SO'}$ from a flat detector, where O' is the center of the 3D roto-translated coordinate detector system. The object inertial orthogonal 3D reference system $Oxyz$ is assumed to be fixed in such a way that the Oxy plane is parallel to the source rotational plane, the z axis is parallel to the rotational axis, the Oxz plane and the Oyz plane are parallel to two adjacent faces of the parallelepipedal volume, and the origin O is positioned in such a way that the isocenter C is located at $C = 0.5 + \overline{\boldsymbol{n}}/2$.

The object volume is discretized by means of a volumetric uniform rectangular grid of voxels, each sized $\boldsymbol{\Delta} = (\Delta_x, \Delta_y, \Delta_z)^T \in \mathbb{R}^3_+$ in some measure of length, *e.g.*, millimeters. The number of object discretization voxels is $\overline{\boldsymbol{n}} = (n_x, n_y, n_z)^T \in \mathbb{N}^3$.

Matlab the kernel has been completely re-written in C++ (including BLAS and LAPACK), but it still preserve the column-major array vectorization.

Analogously, we assume that the flat detector is discretized by means of a uniform rectangular grid of pixels, each sized $\boldsymbol{\delta} = (\delta_u, \delta_v)^T \in \mathbb{R}^2_+$ in the same measure of length as voxels. The number of detector discretization pixels is $\overline{\boldsymbol{m}} = (m_u, m_v)^T \in \mathbb{N}^2$. The rotational path of the source S is discretized by uniformly sampling $[0, 2\pi[$ with N_θ angles $\boldsymbol{\mathcal{A}} = \{\theta_1, \dots, \theta_{N_\theta}\}$.

The reference axes are scaled by the corresponding voxel size), so that the centers of all the voxels have positive integer coordinates, with the center $\boldsymbol{c}(\boldsymbol{n}) = (1,1,1)^T$ of the front-lower-left voxel $\boldsymbol{n} = (1,1,1)$ in the grid, and, in general, $\boldsymbol{c}(i_1, i_2, i_3) = (i_1, i_2, i_3)^T$, for all $i_1 \in \{1, \dots, n_x\}$, $i_2 \in \{1, \dots, n_y\}$, $i_3 \in \{1, \dots, n_z\}$. We also introduce the two following scaling matrices, which depends on the voxel and pixel pitch:

$$H_{\text{vxl}} = \begin{pmatrix} \Delta_x & & \\ & \Delta_y & \\ & & \Delta_z \end{pmatrix} \qquad \text{and} \qquad H_{\text{dtc}} = \begin{pmatrix} \delta_u & 0 \\ 0 & 0 \\ 0 & \delta_v \end{pmatrix}. \tag{3.6}$$

H_{obj} and H_{dtc} accounts for the conversion factor from the usual to the scaled coordinates system.

Finally, since in many real world CT systems offsets are usually considered to allow flexibility in handling objects, we additionally consider a possible offset $\boldsymbol{a} = (a_u, a_v)^T \in \mathbb{R}^2$ of the detector grid.

As outlined in Subsection 3.3.1, the distance-driven method allows some degree of freedom, mostly related to the choice of the common projection plane. For the vectorized approach we propose, the following choices have been made:

- the common projection plane π_{prj} is one of the coordinate planes Oxz or Oyz, selected by the condition

$$\boldsymbol{w}_{\pi_{\text{prj}}} = \underset{\boldsymbol{e}_1, \boldsymbol{e}_2}{\operatorname{argmax}} \{ |\boldsymbol{s}(\theta)^T \boldsymbol{e}_i| \} \tag{3.7}$$

  where $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ are the first and second versors of the canonical basis, that is the versors of the Oyz and Oxz coordinate planes, respectively, and $\boldsymbol{s}(\theta)$ is the versor of the ray going from the source S through the rotation center C;

- for each $\boldsymbol{n}$ in the object, we project onto the common projection plane the slab corresponding to the "midway slice" of the voxel, whose versor $\boldsymbol{e}(\boldsymbol{n}, \theta)$ satisfies the condition

$$\begin{aligned} \boldsymbol{e}(\boldsymbol{n}, \theta) &= \underset{\boldsymbol{e}_1, \boldsymbol{e}_2}{\operatorname{argmax}} \{ |\boldsymbol{w}_{xy}(\boldsymbol{n})^T \boldsymbol{e}_i| \} \\ \text{with} \qquad \boldsymbol{w}_{xy}(\boldsymbol{n}) &= \underset{\boldsymbol{w} \in \mathbb{R}^2, \|\boldsymbol{w}\|=1}{\operatorname{argmin}} \boldsymbol{w}(\boldsymbol{n})^T \boldsymbol{w} \end{aligned} \tag{3.8}$$

where $\boldsymbol{w}(\boldsymbol{n})$ is the versor of the ray from the source to the detector going through the center $\boldsymbol{c}(\boldsymbol{n})$ of the voxel $\boldsymbol{n}$ in the object volume and $\boldsymbol{w}_{xy}(\boldsymbol{n})$ is its projection onto the Oxy plane.

Condition (3.7) essentially means that the coordinated plane selected to be the common projection plane is the one having its versor making an angle with modulus not greater than $\pi/4$ with the source-through-rotation-center ray. In such a way, too "stretched" projections of (some part of) the object volume and/or of the detector cells shall be avoided. Condition (3.8) means that the voxel contribution to a given projection is approximated by computing the *shadow* onto the common projection plane of the middle (slice) parallel to π_{prj}, as the distance-driven technique requires. Clearly, the projection of this "midway slice" of the voxel is completely determined by the projections of its four vertices, which are the middle points of the four voxel edges orthogonal to π_{prj}. This fact is crucial for two reasons:

1) the relative positions of the projections of these middle points is sufficient to determine the *overlap areas* between the voxel projections and the detector projections onto π_{prj};

2) this approximation avoids the aliasing problem of the voxels belonging to the same layer.

Clearly, also the projection of each detector cell is completely determined by projecting its four vertices. Thus, analogously to the object voxel case, we project onto π_{prj} only the four vertices of each detector cell.

Conditions (3.7) and (3.8) imply that the selected common projection plane is the Oyz plane for source rotation angles $-\pi/4 < \theta < \pi/4$ or $3\pi/4 < \theta < 5\pi/4$, while for $\pi/4 \leq \theta \leq 3\pi/4$ or $5\pi/4 \leq \theta \leq 7\pi/4$ the selected common projection plane is the Oxz plane. Clearly, the choice of the common projection plane impacts on the choice of the midway slice to be projected for each voxel, since it follows immediately from conditions (3.7) and (3.8) that the midway slice selected is *always* parallel to the selected common plane π_{prj}. It is then easy to determine the four vertices of interest that identify such midway slices. Indeed, in the *scaled* main reference system, each voxel center is located at positive integer coordinates, and, obviously, inner adjacent voxels share one or two vertices. Thus, whatever common projection plane is selected between the Oxz and the Oyz planes, the vertices to be located are the middle points of all the voxel edges orthogonal to π_{prj}. Hence, when $\pi_{\mathrm{prj}} = Oxz$ the vertices to be projected have coordinates

$$\boldsymbol{v}_{\mathrm{vxl}}(i,j,k) = \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} = \begin{pmatrix} 0.5+i \\ j \\ 0.5+k \end{pmatrix}, \quad i = 0,\ldots,n_x{+}1,\ j = 1,\ldots,n_y,\ k = 0,\ldots,n_z{+}1. \tag{3.9}$$

On the other hand, when $\pi_{\text{prj}} = Oyz$ the vertices to be projected have coordinates

$$\boldsymbol{v}_{\text{vxl}}(i,j,k) = \begin{pmatrix} x_i \\ y_j \\ z_k \end{pmatrix} = \begin{pmatrix} i \\ 0.5+j \\ 0.5+k \end{pmatrix}, \quad i = 1,\dots,n_x,\ j = 0,\dots,n_y{+}1,\ k = 0,\dots,n_z{+}1. \tag{3.10}$$

Initially, *i.e.*, when $\theta = 0$, the system is assumed to be positioned with the source located at $S = C + (d_{SC}/\Delta_x)\boldsymbol{e}_1$, namely S is only displaced in the positive x direction, with respect to C. According to Definiton 3.1, the detector is located at the diametrically opposite position with respect to the object and it is orthogonal to the ray from S through C. We identify the lower-right corner L of the detector, when looking at it from the source, as its “reference point”. Thus, the numbering of the detector cells follows the same direction as the u axis, that is, it increases along the positive u axis. The position of the reference point L determines the detector position in the inertial coordinate system $Oxyz$. At the initial position $\theta = 0$, the roto-translated coordinate system $O'utv$ has the t axis oriented as the x axis and the u axis opposite to the y axis, whilst at $\theta = \pi/2$ both the u and the t axes are oriented exactly as the main inertial axes x and y, respectively. Hence, with respect to the scaled main coordinate system, the detector reference point at $\theta = 0$ is located at

$$L = C + \frac{d_{SC} - d_{SO'}}{\Delta_x}\boldsymbol{e}_1 + \frac{a_u - \delta_u m_u/2}{\Delta_y}\boldsymbol{e}_2 + \frac{a_v - \delta_v m_v/2}{\Delta_z}\boldsymbol{e}_3$$

where $\boldsymbol{a} = (a_u, a_v)^T \in \mathbb{R}^2$ accounts for a possible offset of the detector with respect to its reference point L. Given that the offset is only a translation inside π_{dtc}, for sake of simplicity, in the following we assume $\boldsymbol{a} = \boldsymbol{0}$. Additionally, for object voxels and detector cells, “equilateral” grids are in general considered, yielding $\Delta_x = \Delta_y = \Delta_z = \Delta$ and $\delta_u = \delta_v = \delta$.

In the 2D reference system Luv of the detector plane, we can easily identify the vertices of the detector cells by $(u_i, v_j)^T = (i\delta_u, j\delta_v)^T$, $i = 0,\dots,m_u$, $j = 0,\dots,m_v$. With respect to the scaled main reference system $Oxyz$, these vertices have coordinates $\boldsymbol{v}_{\text{dtc}}(i,j) = L + \begin{pmatrix} u_i/\Delta_x & 0 & v_j/\Delta_z \end{pmatrix}^T$.

For sake of computational convenience, it can be useful to take into account also some extra “virtual” detector cells, around the detector area. The vertices of these extra virtual cells are characterized by having subscripts index $i \in \mathbb{Z}_-$ or $i > m_u$, and $j \in \mathbb{Z}_-$ or $j > m_v$. Notice that the chosen positioning of the Luv scaled coordinate system implies that the “spatial”, *i.e.*, non-scaled, coordinates (u_i, v_j) of each detector cell vertex $\boldsymbol{v}_{\text{dtc}}(i,j)$ are stored in the same position (j,i) (actually, $(j+1, i+1)$ because Matlab array indices starts from 1 and not from 0) of two matrices U_u, for the u_i coordinates, and U_v, for the v_j coordinates. Indeed, the row index i increases with the v coordinate, while the column index j increases

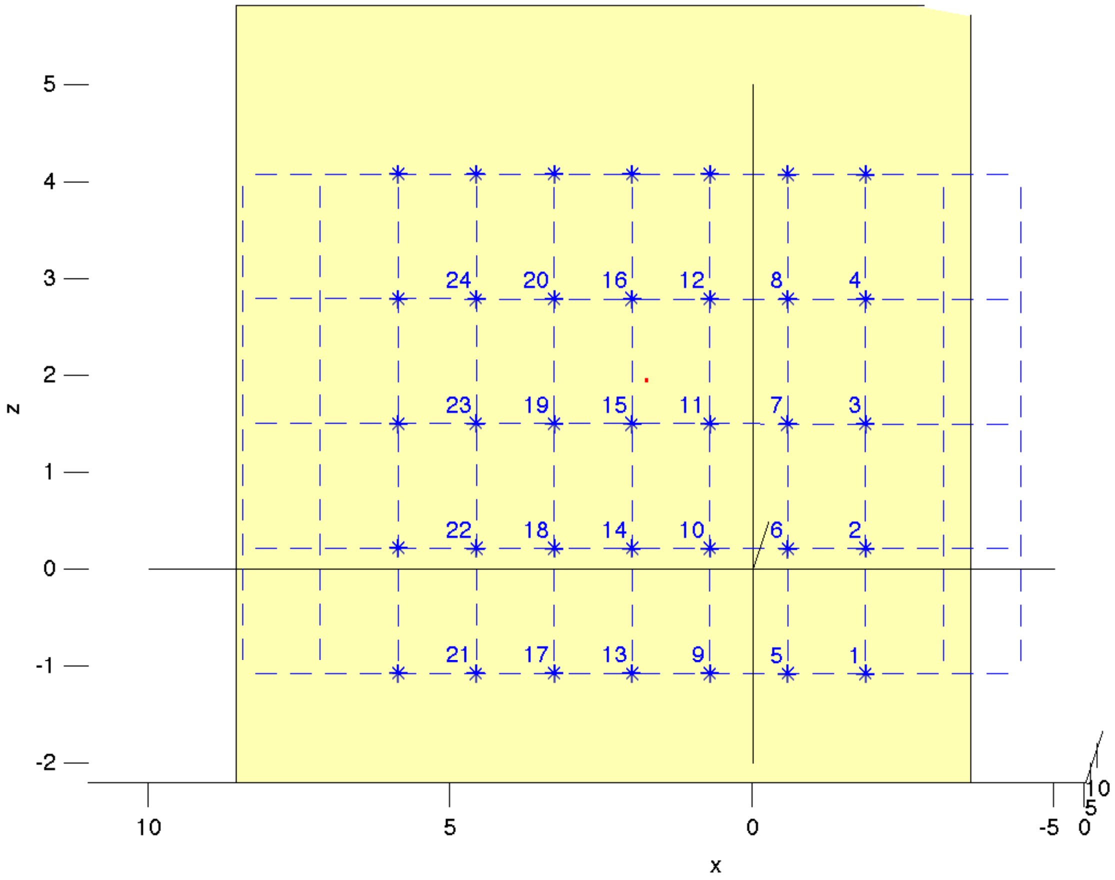


Figure 3.11: representation of the detector cell vertices numbering scheme. The red dot is the X-ray source, the yellow plane is the common projection plane.

with the u coordinate. In particular, this implies that the uppermost rows of the two matrices contain the coordinates of the bottom vertices (*i.e.*, those having the smallest v_j), whilst the lowermost matrices rows contain the coordinates of the top vertices (*i.e.*, those having the largest v_j). On the other hand, the leftmost columns of the two matrices always contain the coordinates of the vertices with the smallest u_i, while the rightmost columns contain the coordinates of the vertices with the largest u_i. This is clear bearing in mind how the Matlab `meshgrid` function works. The main advantage of this approach is that it makes the following matrix-vector computations consistent. In fact, before performing any computation, the two matrices U_u and U_v are vectorized into two column vectors $\overline{\boldsymbol{u}}_{\text{dtc}} = \text{vect}(U_u)$ and $\overline{\boldsymbol{v}}_{\text{dtc}} = \text{vect}(U_v)$, by column-major reordering. As a consequence, the linear index spanning $\overline{\boldsymbol{u}}_{\text{dtc}}$ and $\overline{\boldsymbol{v}}_{\text{dtc}}$ determines a consistent ordering of the cell vertices.

The first vertex, *i.e.*, $\boldsymbol{v}_{\rm dtc}(0,0)$, is the bottom-right corner of the detector grid (by looking at it from the source, as already specified), namely L when $\boldsymbol{a} = \mathbf{0}$. Afterwards, the numbering increases with the vertices $\boldsymbol{v}_{\rm dtc}(0,j)$, $j = 1,\dots,m_v$, aligned on top of $\boldsymbol{v}_{\rm dtc}(0,0)$ along the v direction. Next, the numbering proceeds with the second vertex $\boldsymbol{v}_{\rm dtc}(1,0)$ on the bottom edge of the detector and continues with the vertices $\boldsymbol{v}_{\rm dtc}(1,j)$, $j = 1,\dots,m_v$ on top of it. The last vertices are those on top of $\boldsymbol{v}_{\rm dtc}(m_u,0)$. The last vertex $\boldsymbol{v}_{\rm dtc}(m_u,m_v)$ have thus index $(m_v+1)(m_u+1)$. It follows that contiguous detector vertices from bottom to top on each line parallel to the v axis have consecutive linear indices, while contiguous detector vertices on each line parallel to u have linear indices with a stride of exactly $m_v + 1$. A clarifying representation of the detector cell vertices numbering scheme is depicted in Figure 3.11.

Remark 3.4. Notice that, so far, we made use of two kinds of coordinates in both the main inertial and the roto-translated reference systems: the usual coordinates are lengths, while the *scaled* coordinates have to be intended as (possibly noninteger) *multiples* of the voxel size, in the case of the object space, or of the detector cell size, in the case of the detector plane.

For instance, the generic point $(x,\, y,\, z)^T$ in the object space has scaled coordinates $(\widetilde{x},\, \widetilde{y},\, \widetilde{z})^T = (x/\Delta_x,\, y/\Delta_y,\, z/\Delta_z)^T$, and the generic point $(u,\, v)^T \in \pi_{\rm dtc}$ has scaled coordinates $(\widetilde{u},\, \widetilde{v})^T = (u/\delta_u,\, v/\delta_v)^T$ in the detector plane.

Clearly, when converting the coordinates of a point from, *e.g.*, the scaled detector system to the scaled object system, one has also to multiply by a *conversion factor*, which depends on the voxel and pixel sizes, in addition to the coordinates transformation. For instance, the conversion of the scaled detector plane coordinates $(\widetilde{u},\, \widetilde{v})^T$ to the scaled inertial reference system is

$$\boldsymbol{v}_{\rm dtc}(\widetilde{u},\widetilde{v}) = \begin{pmatrix} \widetilde{x} \\ \widetilde{y} \\ \widetilde{z} \end{pmatrix} = L + H_{\rm vxl}^{-1} \begin{pmatrix} R_\theta & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} H_{\rm dtc} \begin{pmatrix} \widetilde{u} \\ \widetilde{v} \end{pmatrix}$$

where $H_{\rm vxl}$, $H_{\rm dtc}$ are the scaling matrices defined in (5.9) and $R_\theta = R(\theta)$ is the rotation matrix associated to the source and the inertial system, as defined in (3.15). In particular, at the initial position $\theta = 0$, since the positive u axis is rotated by $-\pi/2$ with respect to the positive x axis, the coordinates of the detector cell vertices with respect to the scaled main reference system are

$$\boldsymbol{v}_{\rm dtc}^{(0)}(i,j) = \boldsymbol{v}_{\rm dtc}(i,j)\Big|_{\theta=0} = L + H_{\rm vxl}^{-1} \begin{pmatrix} 0 \\ -u_i \\ v_j \end{pmatrix}, \qquad (3.11)$$

with $i = 0,\dots,m_u$, $j = 0,\dots,m_v$.

Running on the source trajectory

Having stated the general setup and the necessary initialization, we can now describe the main loop over the projection angles $\theta_\ell = 2\pi(\ell - 1)/N_\theta$, $\ell = 1, \ldots, N_\theta$. As we shall see, we can reduce the computation cost by "subgrouping" the projection angles, once π_{prj} and the coordinates of the points to project onto it has been identified. Indeed, notice that:

(i) the vertices of the voxels slices remain the same if the common projection plane does not change;

(ii) the coefficients computed for different source rotation angles fill disjoint parts of the final CT projection matrix;

(iii) the projection computations for a given rotation angle does not impact on those of any other rotation angles.

As a consequence, it is easy to save some computation by changing the order in which the source rotation angles are considered. Indeed, we can first group together all the rotation angles for which $\pi_{\text{prj}} = Oxz$, that is

$$\mathcal{I}_{xz} = \Big\{\ell \in \{1, \ldots, N_\theta\} \,\Big|\, \pi/4 \le \theta_\ell \le 3\pi/4 \text{ or } 5\pi/4 \le \theta_\ell \le 7\pi/4\Big\} \tag{3.12}$$

and then we group together all the rotation angles for which $\pi_{\text{prj}} = Oyz$, namely

$$\begin{aligned} \mathcal{I}_{yz} &= \{1, \ldots, N_\theta\} \setminus \mathcal{I}_{xz} \\ &= \Big\{\ell \in \{1, \ldots, N_\theta\} \,\Big|\, 0 \le \theta_\ell < \pi/4 \text{ or } 3\pi/4 < \theta_\ell < 5\pi/4 \text{ or } 7\pi/4 < \theta_\ell < 2\pi\Big\}. \end{aligned} \tag{3.13}$$

In such a way, we need to compute the vertices of the voxels slices only twice: once for $\ell \in \mathcal{I}_{xz}$ and once for $\ell \in \mathcal{I}_{yz}$. Moreover, we can identify *in advance* the "switching indices", as the following proposition states.

Proposition 2. *Let $\mathcal{R}_\mathcal{C}\big(O, S, d_{CO'}, \gamma_{\min}, \gamma_{\max}, \beta_{\min}, \beta_{\max}\big)$ be a cone-beam circular rotation system endowed with the main coordinate systems $Oxyz$. Given a uniform partition $\mathcal{A}$ of $[0, 2\pi]$, with $\#\mathcal{A} = N_\theta$, the two sets $\mathcal{I}_{xz}$ and $\mathcal{I}_{yz}$ in* (3.12) *and* (3.13)*, respectively, are identified by exactly four integers $\ell_j \in \{1, \ldots, N_\theta\}$, $j = 1, \ldots, 4$.*

Proof. Let us consider the four odd multiples α_j of $\pi/4$, namely $\alpha_j = (2j-1)\pi/4$ for $j = 1, \ldots, 4$. Now, let $\ell_j = \ell_{\alpha_j}$ be the four *largest* indices such that $\theta_{\ell_j} < \alpha_j$

for $i = 1, 3$ and $\theta_{\ell_j} \leq \alpha_j$ for $j = 2, 4$. This yields the following four conditions:

$$
\begin{array}{llll}
j = 1,\, \alpha_1 = \pi/4: & \dfrac{2(\ell_1 - 1)\pi}{N_\theta} < \dfrac{\pi}{4} & \Rightarrow \quad \ell_1 < \dfrac{N_\theta}{8} + 1 & \Rightarrow \quad \ell_1 = \lceil N_\theta/8 \rceil\,, \\
j = 2,\, \alpha_2 = 3\pi/4: & \dfrac{2(\ell_2 - 1)\pi}{N_\theta} \leq \dfrac{3}{4}\pi & \Rightarrow \quad \ell_2 \leq \dfrac{3}{8} N_\theta + 1 & \Rightarrow \quad \ell_2 = \left\lfloor \dfrac{3}{8} N_\theta + 1 \right\rfloor, \\
j = 3,\, \alpha_1 = 5\pi/4: & \dfrac{2(\ell_3 - 1)\pi}{N_\theta} < \dfrac{5}{4}\pi & \Rightarrow \quad \ell_3 < \dfrac{5}{8} N_\theta + 1 & \Rightarrow \quad \ell_3 = \left\lceil \dfrac{5}{8} N_\theta \right\rceil, \\
j = 4,\, \alpha_4 = 7\pi/4: & \dfrac{2(\ell_4 - 1)\pi}{N_\theta} \leq \dfrac{7}{4}\pi & \Rightarrow \quad \ell_4 \leq \dfrac{7}{8} N_\theta + 1 & \Rightarrow \quad \ell_4 = \left\lfloor \dfrac{7}{8} N_\theta + 1 \right\rfloor,
\end{array}
$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceil and floor functions, respectively. As a consequence, one immediately has

$$
\begin{aligned}
\mathcal{I}_{xz} &= \{\ell_1 + 1, \ldots, \ell_2\} \cup \{\ell_3 + 1, \ldots, \ell_4\}\,, \\
\mathcal{I}_{yz} &= \{1, \ldots, \ell_1\} \cup \{\ell_2 + 1, \ldots, \ell_3\} \cup \{\ell_4 + 1, \ldots, N_\theta\}\,.
\end{aligned}
$$

□

Proposition 2 shows that identifying $\mathcal{I}_{xz}$ and $\mathcal{I}_{yz}$ is, on a computational level, very cheap. Indeed, it requires only one division and three multiplications, and four sums and four thresholding, which are both negligible operations. Thus, there is no need to compare θ against $\pi/4$ during the main loop on the angles. Even if this does not seem a great saving, it allows to know in advance the exact indices of the positions projecting onto Oxz and of those projecting onto Oyz, as well as their amount.

Projecting the detector cells

Once ℓ, and hence θ_ℓ, is fixed, we should handle how to compute the projections onto π_{prj} of the detector cells vertices. By construction, the unit vector $\boldsymbol{s}_\ell = \boldsymbol{s}(\theta_\ell) = \big(\cos(\theta_\ell), \sin(\theta_\ell), 0\big)^T$ is parallel to the Oxy plane and it is orthogonal to the detector plane π_{dtc}. The unit vector $\boldsymbol{u}_\ell = \boldsymbol{u}(\theta_\ell) = \big(\sin(\theta_\ell), -\cos(\theta_\ell), 0\big)^T$ belonging to π_{dtc} is always parallel to the u axis of the roto-translated reference system and it is oriented from the detector reference point towards the opposite side of the detector. Now, the coordinates of the scaled detector cells vertices in the $Oxyz$ scaled reference system can be easily computed starting from their "initial" positions $\boldsymbol{v}_{\text{dtc}}^{(0)}$ defined in (3.11):

$$
\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j) = \begin{pmatrix} R_\ell & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{pmatrix} \left(\boldsymbol{v}_{\text{dtc}}^{(0)}(i,j) - \begin{pmatrix} x_C \\ y_C \\ 0 \end{pmatrix} \right) + \begin{pmatrix} x_C \\ y_C \\ 0 \end{pmatrix} \tag{3.14}
$$

where R_ℓ is the counter-clockwise rotation matrix of the Oxy plane around the z axis, at the angle θ_ℓ:

$$R_\ell = R(\theta_\ell) = \begin{pmatrix} \cos(\theta_\ell) & -\sin(\theta_\ell) \\ \sin(\theta_\ell) & \cos(\theta_\ell) \end{pmatrix} \tag{3.15}$$

These operations can be easily vectorized by using the Matlab `repmat` function to replicate the rotation center coordinates:

```
DetPixBoundary_start = ...
    [ ([x_DetPixBoundary y_DetPixBoundary] ...
        - repmat(Isocenter(1:2),prod(detsize+1),1)) * Rot_src
          ...
        + repmat(Isocenter(1:2),prod(detsize+1),1), ...
      z_DetPixBoundary ];
```

where `x_DetPixBoundary` $= \operatorname{vect}\big((\boldsymbol{v}^{(0)}_{\text{dtc}})_x\big)$ is the column vector containing all the x coordinates of $\boldsymbol{v}^{(0)}_{\text{dtc}}(i,j)$ for all $i = 0,\ldots,m_u$ and $j = 0,\ldots,m_v$, and, analogously, `y_DetPixBoundary` $= \operatorname{vect}\big((\boldsymbol{v}^{(0)}_{\text{dtc}})_y\big)$ and `z_DetPixBoundary` $= \operatorname{vect}\big((\boldsymbol{v}^{(0)}_{\text{dtc}})_z\big)$. Also, `Rot_src` $= R_\ell^T$ and `detsize` is the row vector (m_u, m_v). Clearly, the same affine transformation (3.14) is applied to the extra virtual cells, if any, and even the source S is rotated according to (3.14).

Now, notice that, for certain rotation angles, the orientation of the u axis of the roto-translated reference system $O'utv$ is reversed with respect to the orientation of the axis which is the intersection of the common projection plane π_{prj} and the Oxy plane, namely the x axis for $\ell \in \mathcal{I}_{xz}$ and the y axis for $\ell \in \mathcal{I}_{yz}$. This impacts on the order of the distance-driven operations, which determine the *overlap* area on π_{prj} between the projections of the detector cells and the projections of the object voxels, approximated by the projection of their midway slice. To take this "reversed order" into account, we define two additional integers:

$$i_{\text{start}} = \begin{cases} 0 & \text{for } \ell \in \mathcal{I}^+_{xz} \cup \mathcal{I}^+_{yz} \\ N_{\text{dtc}} - m_u & \text{for } \ell \in \mathcal{I}^-_{xz} \cup \mathcal{I}^-_{yz} \end{cases} \qquad i_{\text{inc}} = \begin{cases} +1 & \text{for } \ell \in \mathcal{I}^+_{xz} \cup \mathcal{I}^+_{yz} \\ -1 & \text{for } \ell \in \mathcal{I}^-_{xz} \cup \mathcal{I}^-_{yz} \end{cases} \tag{3.16}$$

where $N_{\text{dtc}} = m_u m_v$ is the total number of detector cells and

$$\mathcal{I}^+_{xz} = \big\{\ell \in \mathcal{I}_{xz} \,\big|\, \ell_1 < \ell \leq \ell_2\big\}, \quad \mathcal{I}^-_{xz} = \mathcal{I}_{xz} \setminus \mathcal{I}^+_{xz} = \big\{\ell \in \mathcal{I}_{xz} \,\big|\, \ell_3 < \ell \leq \ell_4\big\}, \tag{3.17}$$

$$\mathcal{I}^+_{yz} = \big\{\ell \in \mathcal{I}_{yz} \,\big|\, \ell_2 < \ell \leq \ell_3\big\}, \quad \mathcal{I}^-_{yz} = \mathcal{I}_{yz} \setminus \mathcal{I}^+_{yz} = \big\{\ell \in \mathcal{I}_{yz} \,\big|\, 1 \leq \ell < \ell_1 \text{ or } \ell_4 < \ell \leq N_\theta\big\}. \tag{3.18}$$

Afterwards, the following two "flags" will come into play:

$$c_x = \begin{cases} 1 & \text{if } \pi_{\text{prj}} = Oxz \\ 0 & \text{if } \pi_{\text{prj}} = Oyz \end{cases} \qquad c_y = 1 - c_x\,. \tag{3.19}$$

The definitions in (3.16) can be summed up as follows:

$$\begin{cases} i_{\text{start}} = 0,\ i_{\text{inc}} = +1 & \text{for } \ell \text{ such that } \ell_1 < \ell \leq \ell_3 \\ i_{\text{start}} = N_{\text{dtc}} - m_u,\ i_{\text{inc}} = -1 & \text{for } \ell \in \{1, \ldots, \ell_1\} \cup \{\ell_3 + 1, \ldots, N_\theta\} \end{cases} \tag{3.20}$$

Additionally, we set $c_1 = 0$ in the first case of (3.20) and $c_1 = 1$ in the second one, where c_1 is another “flag” that takes memory of the reversed orientation. Indeed, given that the column index of the detector cells always increases with the positive direction of the u axis, the previous machinery handles the following two situations:

$c_1 = 0$: the column index of the detector cells (and hence that of their projections) increases in the same direction of the first axis of the common projection plane, *i.e.*, it is concordant with x if $\pi_{\text{prj}} = Oxz$, or concordant with y if $\pi_{\text{prj}} = Oyz$;

$c_1 = 1$: the column index of the detector cells (and hence that of their projections) increases in the opposite direction of the x axis, if $\pi_{\text{prj}} = Oxz$, or of the y axis, if $\pi_{\text{prj}} = Oyz$.

We are now ready to compute the projections of the detector cells vertices $\boldsymbol{v}_{\text{dtc}}^{(\ell)}$ and of the voxels midway slices vertices $\boldsymbol{v}_{\text{vxl}}^{(\ell)}$ onto the common projection plane π_{prj}. To this end, observe that for a given position θ_ℓ of the source S, the projection onto π_{prj} of a vertex $\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)$ of a detector cell is the intersection of the line going from S to $\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)$ with π_{prj}. As a consequence, one has:

- case $\pi_{\text{dtc}} = Oxz$ ($c_x = 1$):

$$\begin{aligned} x_{\text{dtc}}^{\text{prj}}(i,j) &= x_S - y_S \frac{x_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_x}{y_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_y}\,, \\ z_{\text{dtc}}^{\text{prj}}(i,j) &= z_S - y_S \frac{z_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_z}{y_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_y}\,; \end{aligned} \tag{3.21}$$

- case $\pi_{\text{dtc}} = Oyz$ ($c_y = 1$):

$$\begin{aligned} y_{\text{dtc}}^{\text{prj}}(i,j) &= y_S - x_S \frac{y_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_y}{x_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_x}\,, \\ z_{\text{dtc}}^{\text{prj}}(i,j) &= z_S - x_S \frac{z_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_z}{x_S - \big(\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j)\big)_x}\,. \end{aligned} \tag{3.22}$$

Notice that, to have the coordinates of the projections of all the detector vertices, we do not need to explicitly compute them all. Indeed:

1) consider one single column of detector cells and their vertices aligned on the same "vertical" side (*i.e.*, on the the same edge parallel to the z axis), that is, $\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i,:)$ for each fixed $i \in \{0,\dots,m_u\}$. Clearly, all these vertices have the same x and the same y coordinate:

$$\big(\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i,j)\big)_x = \big(\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i,0)\big)_x \quad \text{and} \quad \big(\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i,j)\big)_y = \big(\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i,0)\big)_y \quad \forall j = 1,\dots,m_v\,. \tag{3.23}$$

Hence, we only need to compute the projections onto π_{prj} of the vertices on the same "horizontal" line. We choose the very bottom detector edge and use (3.21) or (3.22), depending on which is the common projection plane. In either case, we define one column vector containing such projections:

$$\begin{aligned} \boldsymbol{x}^{\text{prj}}_{\text{dtc},0} &= \operatorname{vect}\Big(\big(x^{\text{prj}}_{\text{dtc}}(i,0)\big)_{i=0,\dots,m_u}\Big) \quad \text{if } \pi_{\text{dtc}} = Oxz\,, \qquad \text{or} \\ \boldsymbol{y}^{\text{prj}}_{\text{dtc},0} &= \operatorname{vect}\Big(\big(y^{\text{prj}}_{\text{dtc}}(i,0)\big)_{i=0,\dots,m_u}\Big) \quad \text{if } \pi_{\text{dtc}} = Oyz\,; \end{aligned} \tag{3.24}$$

2) consider one single row of detector cells and their vertices aligned on the same "horizontal" side (*i.e.*, on the the same edge parallel to the Oxy plane), namely $\boldsymbol{v}^{(\ell)}_{\text{dtc}}(:,j)$ for each fixed $j \in \{0,\dots,m_v\}$. All these vertices have the same z coordinate, that is,

$$\big(\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i,j)\big)_z = \big(\boldsymbol{v}^{(\ell)}_{\text{dtc}}(0,j)\big)_z \quad \forall i = 1,\dots,m_u\,. \tag{3.25}$$

Hence, we only need to compute the projections onto π_{prj} of the vertices on one "vertical" line. We choose the leftmost (or the rightmost) detector edge and use (3.21) or (3.22), depending on which is the common projection plane. We then define the following column vector containing such projections:

$$\boldsymbol{z}^{\text{prj}}_{\text{dtc},0} = \operatorname{vect}\Big(\big(z^{\text{prj}}_{\text{dtc}}(0,j)\big)_{j=0,\dots,m_v}\Big)\,. \tag{3.26}$$

It follows that, to have the coordinates in π_{prj} of the projections of all the detector vertices, we just need to appropriately replicate the vectors $\boldsymbol{x}^{\text{prj}}_{\text{dtc},0}$ (or $\boldsymbol{y}^{\text{prj}}_{\text{dtc},0}$) and $\boldsymbol{z}^{\text{prj}}_{\text{dtc},0}$ containing the projections. Because of the detector cells vertices ordering described above, two consecutive vertices horizontally aligned (that is, with constant v coordinate) have linear indices with stride $m_v + 1$, while two consecutive vertices vertically aligned (*i.e.*, with the same u coordinate) have consecutive linear indices. Hence, the two vectors $\boldsymbol{x}^{\text{prj}}_{\text{dtc}}$ (or $\boldsymbol{y}^{\text{prj}}_{\text{dtc}}$) and $\boldsymbol{z}^{\text{prj}}_{\text{dtc}}$ of the projections of all the detector vertices at the angle θ_ℓ can be expressed by means of the following Knonecker products:

$$\boldsymbol{x}_{\text{dtc}}^{\text{prj}} = \boldsymbol{x}_{\text{dtc},0}^{\text{prj}} \otimes \mathbf{1}_{(m_v+1)} \qquad \text{or} \qquad \boldsymbol{y}_{\text{dtc}}^{\text{prj}} = \boldsymbol{y}_{\text{dtc},0}^{\text{prj}} \otimes \mathbf{1}_{(m_v+1)} \tag{3.27}$$

and

$$\boldsymbol{z}_{\text{dtc}}^{\text{prj}} = \mathbf{1}_{(m_u+1)} \otimes \boldsymbol{z}_{\text{dtc},0}^{\text{prj}} \tag{3.28}$$

where $\mathbf{1}_{(m_v+1)}$ and $\mathbf{1}_{(m_u+1)}$ are the column vectors of all 1, with m_v+1 and m_u+1 elements, respectively. In particular, consecutive elements of $\boldsymbol{x}_{\text{dtc},0}^{\text{prj}}$ and $\boldsymbol{y}_{\text{dtc},0}^{\text{prj}}$ are mapped to positions with stride m_v+1 in $\boldsymbol{x}_{\text{dtc}}^{\text{prj}}$ and $\boldsymbol{y}_{\text{dtc}}^{\text{prj}}$, respectively. We can easily save computations by using, once more, the matrix replication Matlab function `repmat`:

```
u = repmat( uproj', detsize(1)+1, 1);
DetProj(:,1) = [ u(:)                              ];
DetProj(:,2) = [ repmat( vproj, detsize(2)+1, 1)];
```

where `uproj` $= \boldsymbol{x}_{\text{dtc},0}^{\text{prj}}$ or `uproj` $= \boldsymbol{y}_{\text{dtc},0}^{\text{prj}}$, and `vproj` $= \boldsymbol{z}_{\text{dtc},0}^{\text{prj}}$. The array `DetProj` contains in its first column the x (or the y) coordinates and in its second column the corresponding z coordinates of the projections of the vertices of all the detector cells, for the given rotation angle θ_ℓ. The array `DetProj` has exactly $N_{\text{dtc}}^{\text{prj}} = (m_u+1)(m_v+1)$ rows.

Finally, if on the detector plane π_{dtc} extra virtual cells are considered, the projections of their vertices are computed analogously, by using (3.21) or (3.22).

Notice that, since we choose a uniform rectangular grid for the detector cells, the horizontal and vertical coordinates of their vertices remains equally spaced for all rotation angles θ_ℓ, $\ell \in \{0, \dots, N_\theta\}$, as expected. Indeed, starting from (3.11) and (3.14):

$$\boldsymbol{v}_{\text{dtc}}^{(0)}(i,j) - \boldsymbol{v}_{\text{dtc}}^{(0)}(i,j-1) = \begin{pmatrix} 0 \\ 0 \\ \delta_v/\Delta_z \end{pmatrix}, \tag{3.29}$$

$$\boldsymbol{v}_{\text{dtc}}^{(0)}(i,j) - \boldsymbol{v}_{\text{dtc}}^{(0)}(i-1,j) = \begin{pmatrix} 0 \\ -\delta_u/\Delta_y \\ 0 \end{pmatrix}, \tag{3.30}$$

$$\begin{aligned} \boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j) - \boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,j-1) &= \begin{pmatrix} R_\ell & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \left(\boldsymbol{v}_{\text{dtc}}^{(0)}(i,j) - \boldsymbol{v}_{\text{dtc}}^{(0)}(i,j-1) \right) \\ &= \begin{pmatrix} 0 \\ 0 \\ \delta_v/\Delta_z \end{pmatrix}, \end{aligned} \tag{3.31}$$

$$
\begin{aligned}
\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j) - \boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i-1,j) &= \begin{pmatrix} R_\ell & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \Big(\boldsymbol{v}^{(0)}_{\mathrm{dtc}}(i,j) - \boldsymbol{v}^{(0)}_{\mathrm{dtc}}(i-1,j) \Big) \\
&= \frac{\delta_u}{\Delta_y} \begin{pmatrix} \sin(\theta_\ell) \\ -\cos(\theta_\ell) \\ 0 \end{pmatrix} \qquad (3.32)
\end{aligned}
$$

for all $i = 1, \ldots, m_u + 1$ and $j = 1, \ldots, m_v + 1$. In particular, relation (3.31) and (3.29) follows also from Proposition 3, while (3.32) and (3.30) do not. Moreover, we have

$$
\begin{aligned}
\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j) &= \begin{pmatrix} R_\ell & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \left(L + \begin{pmatrix} 0 \\ -i\delta_u/\Delta_y \\ j\delta_v/\Delta_z \end{pmatrix} - \begin{pmatrix} x_C \\ y_C \\ 0 \end{pmatrix} \right) + \begin{pmatrix} x_C \\ y_C \\ 0 \end{pmatrix} \\
&= L^{(\ell)} + \begin{pmatrix} R_\ell & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} -x_C \\ -(i\delta_u/\Delta_y + y_C) \\ j\delta_v/\Delta_z \end{pmatrix} + \begin{pmatrix} x_C \\ y_C \\ 0 \end{pmatrix} \qquad (3.33)
\end{aligned}
$$

$$\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_x = L^{(\ell)}_x + \big(1-\cos(\theta_\ell)\big)x_C + \big(i\delta_u/\Delta_y + y_C\big)\sin(\theta_\ell) \qquad (3.34)$$

$$\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_y = L^{(\ell)}_y - x_C\sin(\theta_\ell) - \big(i\delta_u/\Delta_y\big)\cos(\theta_\ell) + \big(1-\cos(\theta_\ell)\big)y_C \qquad (3.35)$$

$$\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_z = L^{(\ell)}_z + j\delta_v/\Delta_z \qquad (3.36)$$

where

$$L^{(\ell)} = \begin{pmatrix} R_\ell & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} L.$$

Equations (3.34) and (3.35) show that $\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_x$ and $\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_y$ depend on i, but not on j, while (3.36) shows that $\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_z$ depends on j, but not on i. The previous equations can be used to prove the above mentioned properties of the detector cells projections.

From (3.35) we have that $\big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_y = \big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j-1)\big)_y$, and from (3.21) (case $\pi_{\mathrm{dtc}} = Oxz$) we have:

$$
\begin{aligned}
z^{\mathrm{prj}}_{\mathrm{dtc}}(i,j) - z^{\mathrm{prj}}_{\mathrm{dtc}}(i,j-1) &= y_S \left(-\frac{z_S - \big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_z}{y_S - \big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_y} + \frac{z_S - \big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j-1)\big)_z}{y_S - \big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j-1)\big)_y} \right) \\
&= y_S \frac{-L^{(\ell)}_z + j\delta_v/\Delta_z + L^{(\ell)}_z - (j-1)\delta_v/\Delta_z}{y_S - \big(\boldsymbol{v}^{(\ell)}_{\mathrm{dtc}}(i,j)\big)_y} \\
&= \rho^{(\ell)}_{x,i} \frac{\delta_v}{\Delta_z} \qquad (3.37)
\end{aligned}
$$

with

$$\rho_{x,i}^{(\ell)} = \frac{y_S}{\Big(y_S - L_y^{(\ell)} + x_C \sin(\theta_\ell) + \big(i\delta_u/\Delta_y\big)\cos(\theta_\ell) - \big(1-\cos(\theta_\ell)\big)y_C\Big)}. \tag{3.38}$$

Analogously, from (3.34) we have that $\big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j)\big)_x = \big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j-1)\big)_x$, and from (3.22) (case $\pi_{\rm dtc} = Oyz$) we have:

$$\begin{aligned} z_{\rm dtc}^{\rm prj}(i,j) - z_{\rm dtc}^{\rm prj}(i,j-1) &= x_S\left(-\frac{z_S - \big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j)\big)_z}{x_S - \big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j)\big)_x} + \frac{z_S - \big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j-1)\big)_z}{x_S - \big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j-1)\big)_x}\right) \\ &= x_S \frac{-L_z^{(\ell)} + j\delta_v/\Delta_z + L_z^{(\ell)} - (j-1)\delta_v/\Delta_z}{x_S - \big(\boldsymbol{v}_{\rm dtc}^{(\ell)}(i,j)\big)_x} \\ &= \rho_{y,i}^{(\ell)} \frac{\delta_v}{\Delta_z} \end{aligned} \tag{3.39}$$

with

$$\rho_{y,i}^{(\ell)} = \frac{x_S}{\Big(x_S - L_x^{(\ell)} - \big(1-\cos(\theta_\ell)\big)x_C - \big(i\delta_u/\Delta_y + y_C\big)\sin(\theta_\ell)\Big)}. \tag{3.40}$$

All together, (3.37)–(3.40) show that, for each column of detector cells vertices, their projections are (vertically aligned and) are equally spaced.

Projecting the voxels midway slices

Once the detector cells vertices projections have been computed, we shall handle how to compute the projections of the object grid vertices, at the same angle ℓ. We assume to have the set $\mathcal{V}_{\rm vxl} = \{\boldsymbol{v}_{\rm vxl}(i,j,k)\}$ of the slices vertices to be projected onto $\pi_{\rm prj}$ already available form (3.9) or (3.10).

Analogously to the detector cell vertices case, we need to vectorize the arrays of the spatial components of the voxels slices vertices. This yields a "linear" index that allows to correctly browse the corresponding vectors. In this, we shall have three 3D arrays U_x, U_y and U_z containing the coordinates x_i, y_j and z_k, respectively, of all the $\boldsymbol{v}_{\rm vxl}(i,j,k)$ vertices, for $i = 1-c_x, \ldots, n_x$, $j = 1-c_y, \ldots, n_y$ and $k = 0, \ldots, n_z$.

The chosen positioning of the main inertial coordinate system $Oxyz$ implies that, in the three arrays, the coordinates of the voxels slices vertices on the bottom layer are stored first, followed by those of the other layers, with the layer index k increasing as z increases. Within each fixed k-th layer, the vertices $\boldsymbol{v}_{\rm vxl}(i,j,k)$ are stored in the same position (j,i,k) (actually, $(j+1,i+1,k+1)$ because of Matlab rules for array indices) of the three arrays U_x, U_y and U_z.

Indeed, the row index i increases with the y coordinate, while the column index j increases with the x coordinate. It follows that the uppermost row of each k-th page of the three arrays contains the coordinates of the vertices on the "front" face of the voxels slices grid, *i.e.*, the face nearest (and parallel) to the Oxz plane. Here, the coordinates of the bottom vertices (*i.e.*, those with the smallest z values) appear in the first row of the first page of each array, and the coordinates of the vertices at the top of that grid face (*i.e.*, those with the largest z) appear in the first row of the last page of the arrays. On the other hand, the last row of each page of the three arrays U_x, U_y and U_z contains the coordinates of the vertices belonging to the face of the voxels slices grid located further (and parallel) with respect to Oxz, with the coordinates of the vertices with the smallest z_k appearing in the first page and those of the vertices with the largest z_k appearing in the last page of each array. Similarly, the coordinates of the vertices on the face of the voxels slices grid nearer to the Oyz plane appear in the first column of each page of the three arrays, while the coordinates of the vertices on the voxels slices grid face further from the Oyz plane appear in the last column of each page of the three arrays.

This is clear bearing in mind how the Matlab `meshgrid` function works, and, once more, provides consistent computations. Indeed, before performing any computation, the three arrays are vectorized in three column vectors $\overline{\boldsymbol{x}}_{\mathrm{vxl}} = \mathrm{vect}(U_x)$, $\overline{\boldsymbol{y}}_{\mathrm{vxl}} = \mathrm{vect}(U_y)$ and $\overline{\boldsymbol{z}}_{\mathrm{vxl}} = \mathrm{vect}(U_z)$ by column-major reordering. As a consequence, the linear index $\boldsymbol{v}_{\mathrm{vxl}}(i,j,k)$ spanning these three vectors determines a consistent ordering of the voxels slices vertices. The first one is the vertex nearest to O, that is $\boldsymbol{v}_{\mathrm{vxl}}(0,1,0)$ or $\boldsymbol{v}_{\mathrm{vxl}}(1,0,0)$ according to (3.9) or (3.10), respectively. Afterwards, the numbering increases with the vertices $\boldsymbol{v}_{\mathrm{vxl}}(0,j,0)$ (or $\boldsymbol{v}_{\mathrm{vxl}}(1,j,0)$), for $j = 1,\ldots,n_y$, *i.e.*, the vertices in the lowest layer that are aligned along the leftmost edge of the slices grid (which is parallel to Oyz). Next, the numbering proceeds with the second vertex $\boldsymbol{v}_{\mathrm{vxl}}(1,1,0)$ (or $\boldsymbol{v}_{\mathrm{vxl}}(2,0,0)$) in the x direction, followed by $\boldsymbol{v}_{\mathrm{vxl}}(1,j,0)$ (or $\boldsymbol{v}_{\mathrm{vxl}}(2,j,0)$) for $j = 1,\ldots,n_y$, until the last vertex of the first layer, which is $\boldsymbol{v}_{\mathrm{vxl}}(n_x,n_y,0)$, in both cases. Notice that $\boldsymbol{v}_{\mathrm{vxl}}(n_x,n_y,0)$ has index $(n_x+1)n_y$, if $\pi_{\mathrm{prj}} = Oxz$, or $n_x(n_y+1)$, if $\pi_{\mathrm{prj}} = Oyz$. The same ordering is repeated in each next k-th layer, for $k = 1,\ldots,n_z$, as it is clear from Figure 3.12. As a consequence, the coordinates of contiguous vertices aligned in the y direction in a given layer have consecutive indices, while those of contiguous vertices on each line parallel to the x axis have indices with a stride of n_y or n_y+1, depending on whether $\pi_{\mathrm{prj}} = Oxz$ or $\pi_{\mathrm{prj}} = Oyz$, respectively. Lastly, the coordinates of contiguous vertices in adjacent layers in the z direction have indices with a stride of $(n_x+1)n_y$, if $\pi_{\mathrm{prj}} = Oxz$, or $n_x(n_y+1)$, if $\pi_{\mathrm{prj}} = Oyz$.

Now, the total number of slices vertices to be projected for each voxel layer depends on whether $\pi_{\mathrm{prj}} = Oxz$ or $\pi_{\mathrm{prj}} = Oyz$. Indeed, the midway slices are taken

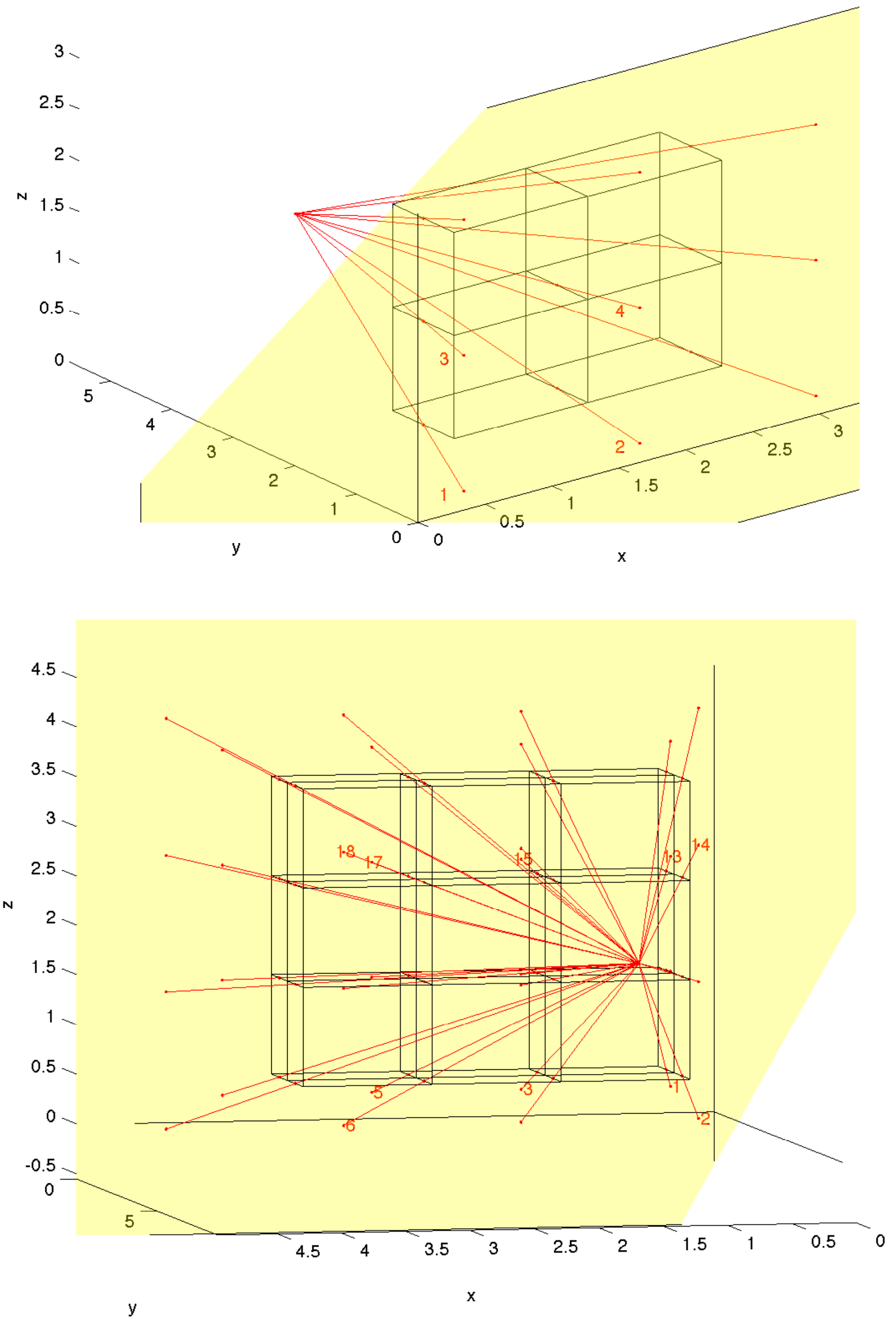


Figure 3.12: representation of the voxels slices vertices numbering scheme. The red dot is the X-ray source, the yellow plane is the common projection plane.

on planes that are parallel to the selected projection plane. These parallel planes contain the centers of the voxels belonging to a same object grid layer parallel to π_{prj}. By construction, when $c_y = 1$ these parallel planes have positive integer x coordinate, while, when $c_x = 1$, they have positive integer y coordinate:

$$\begin{cases} y = j & \forall j = 1, \ldots, n_y, \quad \text{if } \pi_{\mathrm{prj}} = Oxz\,, \\ x = i & \forall i = 1, \ldots, n_x, \quad \text{if } \pi_{\mathrm{prj}} = Oyz\,. \end{cases}$$

Hence, in the orthogonal direction with respect to the common projection plane, we have as many slice planes, and, consequently, aligned slices vertices, as the number of object voxels in that direction. On the other hand, in the other two coordinate directions, parallel to the common projection plane, in each layer we must take into account one more vertex than the number of voxels in each row and one more vertex than the number of voxels in each column. This results in the following total number of voxels slices vertices to be projected:

$$N_{\mathrm{vxl}}^{\mathrm{prj}} = (n_z + 1) N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} \quad \text{with} \quad N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} = \begin{cases} n_y(n_x + 1) & \text{if } \pi_{\mathrm{prj}} = Oxz \\ n_x(n_y + 1) & \text{if } \pi_{\mathrm{prj}} = Oyz \end{cases} \tag{3.41}$$

where $N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}}$ is the number of voxels slices vertices in each object grid horizontal (*i.e.*, parallel to Oxy) layer. The projection of a single vertex of any voxel slice can be easily computed similarly to the detector vertices case, hence analogously to (3.21) or (3.22):

- case $\pi_{\mathrm{dtc}} = Oxz$ ($c_x = 1$):

$$\begin{aligned} x_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k) &= x_S - y_S \frac{x_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_x}{y_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_y}\,, \\ z_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k) &= z_S - y_S \frac{z_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_z}{y_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_y}\,; \end{aligned} \tag{3.42}$$

- case $\pi_{\mathrm{dtc}} = Oyz$ ($c_y = 1$):

$$\begin{aligned} y_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k) &= y_S - x_S \frac{y_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_y}{x_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_x}\,, \\ z_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k) &= z_S - x_S \frac{z_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_z}{x_S - \left(\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(i,j,k)\right)_x}\,. \end{aligned} \tag{3.43}$$

Once more, these computations can be easily performed by using the vectorized syntax of Matlab. For instance, in the case $\pi_{\mathrm{prj}} = Oxz$, we can compute the projections of the voxels slices vertices as follows:

```
VoxProj(:,1) = [ Srcpos_start(1) - Srcpos_start(2) .* ...
                 (Srcpos_start(1) - x_ObjVoxelBoundary(:))...
                 ./ (Srcpos_start(2) - y_ObjVoxelBoundary(:))
                     ];
VoxProj(:,2) = [ Srcpos_start(3) - Srcpos_start(2) .* ...
                 (Srcpos_start(3) - z_ObjVoxelBoundary(:))...
                 ./ (Srcpos_start(2) - y_ObjVoxelBoundary(:))
                     ];
```

where the array VoxProj contains, in its first column, the x (or the y) coordinates and, in its second column, the corresponding z coordinates of the projections of the vertices of all the voxels slices, for the given rotation angle θ_ℓ. If, in analogy with (3.27) and (3.28), we define:

$$\boldsymbol{x}_{\mathrm{vxl},k}^{\mathrm{prj}} = \mathrm{vect}\Big(\big(x_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k)\big)_{i=(1-c_x),\ldots,n_x,j=(1-c_y),\ldots,n_y}\Big) \tag{3.44}$$

$$\boldsymbol{y}_{\mathrm{vxl},k}^{\mathrm{prj}} = \mathrm{vect}\Big(\big(y_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k)\big)_{i=(1-c_x),\ldots,n_x,j=(1-c_y),\ldots,n_y}\Big) \tag{3.45}$$

$$\boldsymbol{z}_{\mathrm{vxl},k}^{\mathrm{prj}} = \mathrm{vect}\Big(\big(z_{\mathrm{vxl}}^{\mathrm{prj}}(i,j,k)\big)_{i=(1-c_x),\ldots,n_x,j=(1-c_y),\ldots,n_y}\Big) \tag{3.46}$$

for all $k = 0, \ldots, n_z$, and

$$\boldsymbol{x}_{\mathrm{vxl}}^{\mathrm{prj}} = \mathrm{vect}\Big(\big(\boldsymbol{x}_{\mathrm{vxl},k}^{\mathrm{prj}}\big)_{k=0,\ldots,n_z}\Big) \tag{3.47}$$

$$\boldsymbol{y}_{\mathrm{vxl}}^{\mathrm{prj}} = \mathrm{vect}\Big(\big(\boldsymbol{y}_{\mathrm{vxl},k}^{\mathrm{prj}}\big)_{k=0,\ldots,n_z}\Big) \tag{3.48}$$

$$\boldsymbol{z}_{\mathrm{vxl}}^{\mathrm{prj}} = \mathrm{vect}\Big(\big(\boldsymbol{z}_{\mathrm{vxl},k}^{\mathrm{prj}}\big)_{k=0,\ldots,n_z}\Big). \tag{3.49}$$

we have precisely VoxProj(:,1) $= \boldsymbol{x}_{\mathrm{vxl}}^{\mathrm{prj}}$ (or VoxProj(:,1) $= \boldsymbol{y}_{\mathrm{vxl}}^{\mathrm{prj}}$) and VoxProj(:,2) $= \boldsymbol{z}_{\mathrm{vxl}}^{\mathrm{prj}}$. The array VoxProj has exactly $N_{\mathrm{vxl}}^{\mathrm{prj}}$ rows.

Computing projection overlaps

To compute the 3D distance-driven coefficients we need to handle how to compute the extension of the overlapping area of the projections onto π_{prj} of each voxel midway slice and each detector cell.

Notice that the chosen projection strategy implies that the projections of each voxel midway slice onto the common projection plane have always a rectangular shape. Moreover, also the projections of each detector cell have always a rectangular shape. Hence, the overlapping area of any voxel projection with any detector cell projection is either empty, or has a rectangular shape (possibly degenerate, *i.e.*, a line segment or a single point). This fact is crucial and, surely, gives reason for choosing one of the coordinated planes Oxz or Oyz as the common projection plane. For this reason, it is sufficient to determine the overlapping segments in the “horizontal” direction and in the “vertical” direction, and then multiply them *suitably.* Despite its apparent simplicity, this procedure is not trivial at all, and implies the following tasks:

A) efficiently compute the “horizontal” projections overlaps and associate them to the right areas;

B) efficiently compute the “vertical” projections overlaps and associate them to the right areas;

C) efficiently determine which are the projected segments to be multiplied and properly assign the result to the correct position of the CT projection matrix.

The designation “vertical” refers to (possible) overlaps of the segments belonging to π_{prj} which are all parallel to the z axis. By contrast, the designation “horizontal” refers to (possible) overlaps of the segments belonging to π_{prj} that are all orthogonal to the z axis, *i.e.*, parallel to the x or the y axis, depending on the selected projection plane.

We first describe how to manage task A, then we handle task B, and, at last, we deal with task C.

A. Computing horizontal projections overlaps

The idea for the computation of horizontal projections overlaps is essentially borrowed from the 2D fan-beam circular distance-driven projection strategy, and it is improved to make it effective.

To fix ideas, and without any loss of generality, consider the case $\pi_{\text{prj}} = Oxz$ and take any horizontal line of π_{prj}. Let us denote by $z^{\text{prj}}_{\text{vxl,min}}$ and $z^{\text{prj}}_{\text{vxl,max}}$ the minimum over all and the maximum over all z coordinates of the voxels slices projections onto π_{prj}, respectively, that is:

$$z^{\text{prj}}_{\text{vxl,min}} = \min_{i,j,k} z^{\text{prj}}_{\text{vxl}}(i,j,k) \quad \text{and} \quad z^{\text{prj}}_{\text{vxl,max}} = \max_{i,j,k} z^{\text{prj}}_{\text{vxl}}(i,j,k)\,.$$

Analogously, let us denote by $z^{\text{prj}}_{\text{dtc,min}}$ and $z^{\text{prj}}_{\text{dtc,max}}$ the minimum over all and the maximum over all z coordinates of the detector cells projections onto π_{prj}, respectively, namely:

$$z^{\text{prj}}_{\text{dtc,min}} = \min_{i,j} z^{\text{prj}}_{\text{dtc}}(i,j) \quad \text{and} \quad z^{\text{prj}}_{\text{dtc,max}} = \max_{i,j} z^{\text{prj}}_{\text{dtc}}(i,j)\,.$$

Lastly, let us define the two quantity $z^{\mathrm{prj}}_{\mathrm{min}}$ and $z^{\mathrm{prj}}_{\mathrm{max}}$:

$$z^{\mathrm{prj}}_{\mathrm{min}} = \min\bigl\{z^{\mathrm{prj}}_{\mathrm{dtc,min}}, z^{\mathrm{prj}}_{\mathrm{vxl,min}}\bigr\} \quad \text{and} \quad z^{\mathrm{prj}}_{\mathrm{max}} = \max\bigl\{z^{\mathrm{prj}}_{\mathrm{dtc,max}}, z^{\mathrm{prj}}_{\mathrm{vxl,max}}\bigr\}\,.$$

Evidently, every line on π_{prj} having equation $z = \zeta$ with $\zeta < z^{\mathrm{prj}}_{\mathrm{min}}$ or $\zeta > z^{\mathrm{prj}}_{\mathrm{max}}$ has no intersection at all neither with the whole object grid projection, nor with the whole detector grid projection. Hence, there is nothing to compute in this case. If either $z^{\mathrm{prj}}_{\mathrm{min}} \le \zeta < \max\bigl\{z^{\mathrm{prj}}_{\mathrm{dtc,min}}, z^{\mathrm{prj}}_{\mathrm{vxl,min}}\bigr\}$, or $\min\bigl\{z^{\mathrm{prj}}_{\mathrm{dtc,max}}, z^{\mathrm{prj}}_{\mathrm{vxl,max}}\bigr\} < \zeta \le z^{\mathrm{prj}}_{\mathrm{max}}$, then we have a special situation where the line intersects the projections of either the detector or the voxels slices, but not both of them. Clearly, there can not be any projections overlap on the line, so, even in this case, there is nothing to compute. As a consequence, we shall consider only horizontal lines of π_{prj} having intersections with both the detector cells projections and the voxels slices projections.

Then, let us consider one such line λ. The intersections are segments, whose endpoints are uniquely identified by the projections onto this line of the edges of the detector cells and the voxels midway slices. The set of endpoints related to the detector cells projections is clearly sorted as the corresponding detector cells, and the same holds true for the set of endpoints related to the voxels slices, even if the two orderings could be opposite to each other. The latter case appears when the u detector axis is *not* concordant with the x or the y axis of the common projection plane: this happens when $5\pi/4 \le \theta < 9\pi/4$, *i.e.*, when the projection angle index ℓ satisfies $\ell \in \mathcal{I}^-_{xz} \cup \mathcal{I}^-_{yz} = \{1,\dots,\ell_1\} \cup \{\ell_3+1,\dots,N_\theta\}$, according to (3.16). However, the endpoints in the two sets are usually interleaved, as it is sketched in Figure 3.5. As a consequence, regardless of their ordering in the two sets, the endpoints of a given i-th detector cell, temporarily addressed to as ν_{i-1} and ν_i, and those of a specific j-th voxel slice, temporarily denoted by ξ_{j-1} and ξ_j, can appear in only one out of four configurations:

1. the endpoints of the detector cell projection are both comprised between those of the voxel slice projection:
$$\xi_{j-1} \le \nu_{i-1} < \nu_i \le \xi_j\,;$$
2. the endpoints of the voxel slice projection and those of the detector cell projection are alternated:
$$(a)\quad \nu_{i-1} \le \xi_{j-1} < \nu_i \le \xi_j \qquad \text{or} \qquad (b)\quad \xi_{j-1} \le \nu_{i-1} < \xi_j \le \nu_i\,;$$
3. the endpoints of the voxel slice projection are both comprised between those of the detector cell projection:
$$\nu_{i-1} \le \xi_{j-1} < \xi_j \le \nu_i\,;$$

4. the two projections are disjoint or share at most one endpoint:

$$(a) \quad \nu_{i-1} < \nu_i \leq \xi_{j-1} < \xi_j \qquad \text{or} \qquad (b) \quad \xi_{j-1} < \xi_j \leq \nu_{i-1} < \nu_i \,.$$

The four configurations above identified correspond to the cases where the voxel slice projection segment completely overlaps the detector cell projection segment (case 1), or where there is only a partial overlap (case 2), or where the voxel slice projection segment is completely "inside" the detector cell projection segment (case 3), or where there is no overlap at all between the segments (case 4). All these four cases can be easily handled by the formula (3.1).

We remind that each coefficient of the CT forward operator measures, in some way, the contribution that a given voxel of the object grid provides to the signal received by a specific detector cell, by means of a percentage. Indeed, the weight w_{ij} in (3.1) satisfies $0 \leq w_{ij} \leq 1$. This is the *overlap* coefficient we need to compute for every voxel-detector pair.

Now, we show how the computation of such overlaps can be effectively vectorized. In practice, we need the *relative* positions of the endpoints of the projection segments for all the voxel slices and all the detector cells. These endpoints are the components of $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$, for the voxel grid, and the components of $\boldsymbol{x}_{\text{dtc}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{dtc}}^{\text{prj}}$ for the detector grid. Clearly, such endpoints only depends on the source rotation angle θ_ℓ. If we retain the notation of cases 1–4, ν_{i-1} and ν_i are generic contiguous components of $\boldsymbol{x}_{\text{dtc}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{dtc}}^{\text{prj}}$, while ξ_{i-1} and ξ_i are generic contiguous components of $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$.

The crucial point is that we only need to identify the *indices* of the components of these column vectors corresponding to horizontally adjacent vertices of the voxel grid and the detector grid. This can be easily done by using both the matrix replication and the Kronecker matrix product, and by exploiting the vertices numbering explained before.

To fix ideas, suppose $\pi_{\text{prj}} = Oxz$: in this case, the voxels midway slices are parallel to Oxz and their vertices are located as in (3.9). Remember that the column-major vectorization ordering implies that

- the components of the projections of all the $(n_x + 1)n_y$ vertices $\boldsymbol{v}_{\text{vxl}}(i, j, k)$ of the k-th horizontal layer are located in $N_{\text{vxl,vrt}}^{\text{lyr}} = (n_x + 1)n_y$ contiguous positions of $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ starting from $(k - 1)(n_x + 1)n_y + 1$;

- inside each k-th horizontal layer, the components corresponding to adjacent vertices along edges parallel to the y axis have consecutive indices in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$, whilst the components corresponding to adjacent vertices along edges parallel to the x direction have indices with a stride of n_y in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$:

$$\boldsymbol{v}_{\text{vxl}}(i,j,k) \to p_{ijk} = \widetilde{p}_k^{\text{H}} + in_y + j \;\Rightarrow\; \begin{cases} \boldsymbol{v}_{\text{vxl}}(i,j+1,k) \to p_{i,j+1,k} = p_{ijk} + 1 \\ \boldsymbol{v}_{\text{vxl}}(i+1,j,k) \to p_{i+1,j,k} = p_{ijk} + n_y \end{cases} \tag{3.50}$$

for all $i = 0, \ldots, n_x$ and $j = 1, \ldots, n_y$, where $\widetilde{p}_k^{\text{H}} = (k-1)(n_x+1)n_y$ is the *displacement* in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ of the positions related to the k-th layer.

Let us consider first the vertices belonging to the lowest layer (*i.e.*, $k = 0$). We need to identify the pair of endpoints of each segment to project. For sake of simplicity, we refer to the endpoints having the smallest x component in each pair as *leftmost* endpoints, and as *rightmost* endpoints to those having the largest x component. For instance, when we are projecting the lowest edge of the midway slice of voxel $\boldsymbol{n} = (i,j,0)$ parallel to Oxz, its endpoints are $\boldsymbol{v}_{\text{vxl}}(i,j,0)$ and $\boldsymbol{v}_{\text{vxl}}(i+1,j,0)$, so the leftmost endpoint is $\boldsymbol{v}_{\text{vxl}}(i,j,0)$ and the rightmost one is $\boldsymbol{v}_{\text{vxl}}(i+1,j,0)$. Then, it is clear that all the vertices with $i = 0, \ldots, n_x - 1$ and $j = 1, \ldots, n_y$ are leftmost endpoints, while all the vertices with $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_y$ are rightmost endpoints. The former are all the vertices of layer 0, except for those farthest from the Oyz plane (with the largest x components), while the latter are all the vertices of layer, 0 except for the nearest to the Oyz plane (with the smallest x components).

Now, let $\boldsymbol{p}_0^{\text{H,left}}$ and $\boldsymbol{p}_0^{\text{H,right}}$ be the column vectors of the indices of all the leftmost endpoints and the rightmost endpoints of layer $k = 0$, respectively, namely:

$$\begin{aligned} \boldsymbol{p}_0^{\text{H,left}} &= \big(1, 2, \ldots, n_x n_y\big)^T , \\ \boldsymbol{p}_0^{\text{H,right}} &= \big(n_y+1, n_y+2, \ldots, (n_x+1)n_y\big)^T = \boldsymbol{p}_0^{\text{H,left}} + n_y \,. \end{aligned} \tag{3.51}$$

Given that the vertices ordering in the other layers proceeds exactly in the same way as in layer 0, it immediately follows that the positions of the components belonging to the k-th layer are simply displaced by $\widetilde{p}_k^{\text{H}}$ from those of layer 0, that is:

$$\boldsymbol{p}_k^{\text{H,left}} = \widetilde{p}_k^{\text{H}} + \boldsymbol{p}_0^{\text{H,left}} \qquad \text{and} \qquad \boldsymbol{p}_k^{\text{H,right}} = \widetilde{p}_k^{\text{H}} + \boldsymbol{p}_0^{\text{H,right}} \qquad \forall\, k = 1, \ldots, n_z \,. \tag{3.52}$$

Notice that, for all $k = 0, \ldots, n_z$, $\boldsymbol{p}_k^{\text{H,left}}$ and $\boldsymbol{p}_k^{\text{H,right}}$ are column vectors with $N_{\text{vxl}}^{\text{lyr}} = n_x n_y$ elements.

Clearly, the same reasoning can be applied to the $\pi_{\text{prj}} = Oyz$ case, but it is not straightforward. Now, the slices edges are parallel to the y axis, the vertices are positioned according to (3.10), and the number of vertices in each k-th layer is $n_x(n_y+1)$, so that the layer displacement becomes $\widetilde{p}_k^{\text{H}} = (k-1)n_x(n_y+1)$. In this case, the *leftmost* endpoint of each segment to be projected is the one having the smallest y coordinate, while the *rightmost* endpoint has the largest component in the pair. While the definition is completely similar to the previous case, due to (3.50) the "neighboring" relations are quite different. Indeed, when we are projecting the lowest edge of the midway slice of voxel $\boldsymbol{n} = (i, j, 0)$ parallel to Oyz, its endpoints are $\boldsymbol{v}_{\mathrm{vxl}}(i, j, 0)$ and $\boldsymbol{v}_{\mathrm{vxl}}(i, j+1, 0)$, where the leftmost endpoint is $\boldsymbol{v}_{\mathrm{vxl}}(i, j, 0)$ and the rightmost one is $\boldsymbol{v}_{\mathrm{vxl}}(i, j+1, 0)$. It immediately follows that all the vertices with $i = 1, \dots, n_x$ and $j = 0, \dots, n_y - 1$ are leftmost endpoints, while all the vertices with $i = 1, \dots, n_x$ and $j = 1, \dots, n_y$ are rightmost endpoints. The former are all the vertices of layer 0 except for those farthest from the Oxz plane (with the largest y components), while the latter are all the vertices of layer 0 except for those nearest to the Oxz plane (with the smallest x components). We can still define the column vectors $\boldsymbol{p}_0^{\mathrm{H,left}}$ and $\boldsymbol{p}_0^{\mathrm{H,right}}$, but in this case the indices of the leftmost endpoints are grouped in n_x groups of exactly n_y consecutive indices, each group having its starting position with a stride of $n_y + 1$ from the previous,while the index of each rightmost endpoint is exactly the subsequent one of the corresponding leftmost endpoint. Hence, the analogue of (3.51) reads as:

$$\boldsymbol{p}_0^{\mathrm{H,left}} = \underbrace{\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}}_{n_x \text{ elements}} \otimes \begin{pmatrix}1\\2\\\vdots\\n_y\end{pmatrix} + (n_y+1)\begin{pmatrix}0\\1\\\vdots\\n_x-1\end{pmatrix} \otimes \underbrace{\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}}_{n_y \text{ elements}},$$
$$\boldsymbol{p}_0^{\mathrm{H,right}} = \boldsymbol{p}_0^{\mathrm{H,left}} + 1\,, \tag{3.53}$$

where "⊗" is the Kronecker product, while (3.52) remains the same. Again, for each k-th horizontal layer, $k = 0, \dots, n_z$, $\boldsymbol{p}_0^{\mathrm{H,left}}$ and $\boldsymbol{p}_0^{\mathrm{H,right}}$ are column vectors with $N_{\mathrm{vxl}}^{\mathrm{lyr}}$ elements.

Equations (3.51) and (3.53) may appear quite dissimilar from each other, but, actually, they are not so different. Indeed, notice that

$$\begin{pmatrix}1\\2\\\vdots\\n_xn_y\end{pmatrix} = \underbrace{\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}}_{n_x \text{ elements}} \otimes \begin{pmatrix}1\\2\\\vdots\\n_y\end{pmatrix} + n_y\begin{pmatrix}0\\1\\\vdots\\n_x-1\end{pmatrix} \otimes \underbrace{\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}}_{n_y \text{ elements}}. \tag{3.54}$$

and observe that (3.53) can be obtained from (3.54) by simply replacing n_y with $n_y + c_y$ as coefficient for the second Kronecker product. Moreover, even if the total number $N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}}$ of vertices in each layer changes, as well as the layer displacement

$\widetilde{p}_k^{\mathrm{H}}$, both can be easily obtained by using the following relations:

$$\begin{gathered} N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} = (n_y + 1 - c_x)(n_x + 1 - c_y) = (n_y + c_y)(n_x + c_x), \\ \widetilde{p}_k^{\mathrm{H}} = (k-1) N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} \,. \end{gathered} \tag{3.55}$$

Thus, we are able to provide a unified formula for the linear indices of all the leftmost and the rightmost endpoints, which holds true for both the $\pi_{\mathrm{prj}} = Oxz$ and the $\pi_{\mathrm{prj}} = Oyz$ cases:

$$\begin{gathered} \boldsymbol{p}_0^{\mathrm{H,left}} = \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{n_x \text{ elements}} \otimes \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n_y \end{pmatrix} + (n_y + c_y) \begin{pmatrix} 0 \\ 1 \\ \vdots \\ n_x - 1 \end{pmatrix} \otimes \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{n_y \text{ elements}}, \\ \boldsymbol{p}_0^{\mathrm{H,right}} = \boldsymbol{p}_0^{\mathrm{H,left}} + c_x n_y + c_y. \end{gathered} \tag{3.56}$$

In particular, the vectors $\boldsymbol{p}_0^{\mathrm{H,left}}$ and $\boldsymbol{p}_0^{\mathrm{H,right}}$ have as many elements as the number $N_{\mathrm{vxl}}^{\mathrm{lyr}}$ of *voxels* in the layer (not as the number of slices vertices). Finally, it is now easy to build the two column vectors $\boldsymbol{p}^{\mathrm{H,left}}$ and $\boldsymbol{p}^{\mathrm{H,right}}$ collecting the vectors $\boldsymbol{p}_k^{\mathrm{H,left}}$ and $\boldsymbol{p}_k^{\mathrm{H,right}}$ for all $k = 0, \ldots, n_z$:

$$\boldsymbol{p}^{\mathrm{H,left}} = \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{n_z + 1 \text{ elements}} \otimes \boldsymbol{p}_0^{\mathrm{H,left}} + N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ n_z \end{pmatrix} \otimes \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{N_{\mathrm{vxl}}^{\mathrm{lyr}} \text{ elements}} \tag{3.57}$$

and the same formula holds true for $\boldsymbol{p}^{\mathrm{H,right}}$, with $\boldsymbol{p}_0^{\mathrm{H,right}}$ replacing $\boldsymbol{p}_0^{\mathrm{H,left}}$. Both $\boldsymbol{p}^{\mathrm{H,left}}$ and $\boldsymbol{p}^{\mathrm{H,right}}$ have length

$$N'_{\mathrm{vxl}} = (n_z + 1) N_{\mathrm{vxl}}^{\mathrm{lyr}} = N_{\mathrm{vxl}} + N_{\mathrm{vxl}}^{\mathrm{lyr}}. \tag{3.58}$$

Thus, in their first N_{vxl} elements, $\boldsymbol{p}^{\mathrm{H,left}}$ and $\boldsymbol{p}^{\mathrm{H,right}}$ are indexed as the *voxels* (not as the slices vertices), and the last $N_{\mathrm{vxl}}^{\mathrm{lyr}}$ positions refers to a “virtual” layer of voxels on the top of the object volume (whose midway slices lower vertices $\boldsymbol{v}_{\mathrm{vxl}}^{(\ell)}(:, :, n_z)$ are the vertices of the uppermost layer). The need to consider these “virtual” voxels shall be clarified later (see page 82).

Remark 3.5. Notice that the Kronecker operations in (3.56) and (3.57) are formally correct, but they are unnecessary from a computational point of view: indeed, they are essentially vector replications, and, thus, can be efficiently implemented in Matlab via the `repmat` function, saving computational time.

Because of (3.42) and (3.43), each element in $\boldsymbol{v}_{\text{vxl}}^{(\ell)}$ is uniquely related to its projection onto π_{prj}, which is the homologous element in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$, depending on the common projection plane. Hence, we actually have the indices that address correctly in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$ the elements that are endpoints of the horizontal segment on π_{prj}, which is the projection of exactly one horizontal edge of each given voxel midway slice. Recall that for the rotation angles θ_ℓ with $\ell \in \mathcal{I}_{xz}^- \cup \mathcal{I}_{yz}^-$ the voxel index increases in the reversed direction with respect to any horizontal axis of π_{prj}. In practice, this implies that, for these source positions, the length of the horizontal projected segment we want to compute is given by the difference between the projection of the leftmost endpoint of the segment and the projection of its rightmost endpoint, instead of the other way round, since, for those ℓ, the former is larger than the latter. Hence, both situations can be handled at the same time by *always* taking the difference between the *largest projection* and the *smallest projection*, in magnitude, of the endpoints pair of each voxel slice horizontal edge. Bearing in mind this, we finally end up with the vectorized definition of the distance-driven voxels projections to be used in the formula (3.1):

$$\boldsymbol{\xi} = \begin{cases} \boldsymbol{x}_{\text{vxl}}^{\text{prj}} & \text{if } \pi_{\text{prj}} = Oxz \\ \boldsymbol{y}_{\text{vxl}}^{\text{prj}} & \text{if } \pi_{\text{prj}} = Oyz \end{cases} \quad \text{and} \quad \begin{cases} \boldsymbol{\xi}^{\min} = \min\Big\{\boldsymbol{\xi}\big(\boldsymbol{p}^{\text{H,left}}\big),\, \boldsymbol{\xi}\big(\boldsymbol{p}^{\text{H,right}}\big)\Big\} \\ \boldsymbol{\xi}^{\max} = \max\Big\{\boldsymbol{\xi}\big(\boldsymbol{p}^{\text{H,left}}\big),\, \boldsymbol{\xi}\big(\boldsymbol{p}^{\text{H,right}}\big)\Big\} \end{cases} \tag{3.59}$$

where "min" and "max" are Hadamard operators, *i.e.*, they are intended to be simultaneously applied to homologous elements when their arguments are arrays of the same size (we stress that this is *not* the usual behavior of `min` and `max` functions of Matlab). All vectors in (3.59) are column vectors. In particular, $\boldsymbol{\xi}$ has $N_{\text{vxl}}^{\text{prj}}$ elements and both $\boldsymbol{\xi}^{\min}$ and $\boldsymbol{\xi}^{\max}$ have N'_{vxl} components (*i.e.*, they are linearly indexed as the voxels, except for the last $N_{\text{vxl}}^{\text{lyr}}$ positions). Notice that $\boldsymbol{\xi}(\boldsymbol{p})$ is the column vector whose elements are the components of $\boldsymbol{\xi}$ having the indices specified in the index vector $\boldsymbol{p}$ (this is usually referred to as *indirect indexing*).

Let us now see how to compute the overlaps between the horizontal projections of the voxels slices and the detector cells.

First, suppose $\pi_{\text{prj}} = Oxz$ and $\ell \in \mathcal{I}_{xz}^+$, so that the horizontal u axis on the detector is concordant with the x axis of π_{prj}.

We start by considering what happens for the first row of detector cells vertices, *i.e.*, the detector vertices $\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,0)$ with linear index $q = i(m_v+1)+1$ for $i \in \{0,\dots,m_u\}$. Consider the lowest edge of the generic i-th horizontal cell of this row: its endpoints are $\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i-1,0)$ and $\boldsymbol{v}_{\text{dtc}}^{(\ell)}(i,0)$ and their projections are the corresponding elements of $\boldsymbol{x}_{\text{dtc},0}^{\text{prj}}$, *i.e.*, its i-th and $(i+1)$-th elements (since array indices start from 1). For any given voxel, we need to manage the four intersection cases 1–4 between the horizontal projections of its midway slice and the projection of the detector segment. The following result is crucial for what follows.

Proposition 3. *Let $\mathcal{R}_{\mathcal{C}}\big(O, S, d_{CO'}, \gamma_{\min}, \gamma_{\max}, \beta_{\min}, \beta_{\max}\big)$ be a cone-beam circular rotation system endowed with the main coordinate systems $Oxyz$, and consider a parallelepipedal voxel discretization of the object volume with all boundary faces parallel to the coordinate planes. Suppose the common projection plane is either $\pi_{prj} = Oxz$ or $\pi_{prj} = Oyz$. Then:*

1. *the horizontal coordinates of the projections of the midway slice vertices of all the voxels are independent of z and equal to the horizontal coordinate of the projections of the homologous vertices in the lowest horizontal layer (i.e., with minimum z);*

2. *the horizontal coordinates of the projections of all the detector cells vertices are independent of z and are equal to the horizontal coordinate of the projections of the homologous vertices in the lowest horizontal detector border (i.e., with minimum z);*

3. *the horizontal lengths of the possible overlaps of voxels slices projections and detector cells projections are independent of z.*

Proof. First, we prove item 1. Because of the choice of the common projection plane, each vertical layer of midway slices (all of them belonging to the same plane containing the centers $\boldsymbol{c}(\boldsymbol{n})$ of a single vertical layer of voxels, *i.e.*, the centers of the voxels $\boldsymbol{n}(:, j, :)$ for a fixed j when $\pi_{\text{prj}} = Oxz$, or the voxels $\boldsymbol{n}(i, :, :)$ for a fixed i when $\pi_{\text{prj}} = Oyz$) is always parallel to π_{prj}. Moreover, all the slices vertices aligned along the same line parallel to the z axis have the same x and y coordinates, *i.e.*, $\big(\boldsymbol{v}_{\text{vxl}}(i,j,k)\big)_x = \big(\boldsymbol{v}_{\text{vxl}}(i,j,0)\big)_x$ and $\big(\boldsymbol{v}_{\text{vxl}}(i,j,k)\big)_y = \big(\boldsymbol{v}_{\text{vxl}}(i,j,0)\big)_y$ for all $k = 1, \dots, n_x$. Hence, because of (3.42) and (3.43), also the horizontal coordinates of their projections, which are elements of $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ (or $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$), are all equal. It follows that, for each vertical layer of slices (parallel to π_{prj}) the horizontal lengths of the possible overlaps of voxels slices projections and detector cells projections can be computed for the vertices of the lowest horizontal voxels layer only. In all the other horizontal layers of voxels, the horizontal lengths of the overlapping areas remain the same.

To show item 2 we recall that, for any given angle index ℓ, we already observed in (3.23) that the x and y coordinates of the detector cell vertices aligned along the same vertical line on π_{dtc} are all the same. Then, it follows from (3.24) that also the horizontal coordinate on π_{prj} of their projections (which is an element of $\boldsymbol{x}_{\text{dtc}}^{\text{prj}}$ or $\boldsymbol{y}_{\text{dtc}}^{\text{prj}}$) is constant. Hence, because of (3.27), for each i-th column of detector cells the horizontal lengths of the overlapping areas between their projections and the projections of all the voxels can be computed only once, for just one detector row.

The last item follows immediately from items 1 and 2. □

Even if it is formally indifferent which row of the i-th column of detector cells is considered for the horizontal overlap lengths calculation, it is computationally convenient to consider the very first row of detector cells and, hence, the lowest row of cell vertices, *i.e.*, $\boldsymbol{v}_{\mathrm{dtc}}^{(\ell)}(:,0)$. Their projections are all the elements of $\boldsymbol{x}_{\mathrm{dtc},0}^{\mathrm{prj}}$ (or $\boldsymbol{y}_{\mathrm{dtc},0}^{\mathrm{prj}}$).

Notice that it will be *crucial* to be able to correctly associate to each voxel-cell pair the right horizontal length of projections overlap. The strategy proposed will be able to ensured this safely without any additional hypothesis, by simply using the defined indexing machinery.

To compute the horizontal lengths, we need to fix a positive direction: the most natural choice is the positive direction of the horizontal axis of π_{prj}, that is, the x axis, if $\pi_{\mathrm{prj}} = Oxz$, or the y axis, if $\pi_{\mathrm{prj}} = Oyz$.

Remark 3.6. Once more, our choice of the common projection plane gives the additional advantage that the horizontal lengths of voxels slices projections are always given by $\boldsymbol{\xi}^{\max} - \boldsymbol{\xi}^{\min}$, and this is independent of the source position.

On the contrary, at certain (known) positions of the source S, *i.e.*, $\ell \in \mathcal{I}_{xz}^{-} \cup \mathcal{I}_{yz}^{-}$, according to (3.17)–(3.18), the orientation of the u axis of the detector is reversed with respect to that of the horizontal axis of π_{prj}.

From Remark 3.6 it follows that, for some positions, we must take into account the reversed ordering of the detector cells. As a consequence, the detector projection $\nu^{\min}$ with minimal horizontal coordinate in π_{prj} is decided as follows:

(i) when $\ell \in \mathcal{I}_{xz}^{+} \cup \mathcal{I}_{yz}^{+}$, $\nu^{\min}$ is the projection of the leftmost lowest vertex of the very first detector cell, that is the projection of $\boldsymbol{v}_{\mathrm{dtc}}^{(\ell)}(0,0)$;

(ii) when $\ell \in \mathcal{I}_{xz}^{-} \cup \mathcal{I}_{yz}^{-}$, $\nu^{\min}$ is the projection of the rightmost lowest vertex of the last cell of the first detector row, *i.e.*, the projection of $\boldsymbol{v}_{\mathrm{dtc}}^{(\ell)}(m_u,0)$.

The situation is clearly reversed for the detector projection $\nu^{\max}$ with maximal horizontal coordinate in π_{prj}.

Both cases i–ii can be addressed exactly as in the case of voxel slices vertices projections, that is, by simply working with an indices vector $\boldsymbol{q}$ with

$$N_{\mathrm{dtc,vrt}} = (m_u + 1)(m_v + 1) \tag{3.60}$$

elements. However, as we already stressed, to compute the horizontal lengths of overlap we need to work only with the vertices of the lowest detector layer. Given that the horizontal coordinates of the projections of these vertices are the elements of $\boldsymbol{x}_{\mathrm{dtc},0}^{\mathrm{prj}}$ or $\boldsymbol{y}_{\mathrm{dtc},0}^{\mathrm{prj}}$, both column vectors sized $m_u + 1$, we shall consider an auxiliary indices vector $\widehat{\boldsymbol{q}}^{\mathrm{H}}$ such that

$$\widehat{q}_i^{\mathrm{H}} = q_j \quad \text{with} \quad j = (i-1)(m_v+1) + 1\,.$$

The horizontal overlap length of the projection of a given i-th detector cell in the first row ($i = 1, \dots, m_u$) is always given by the difference of the horizontal projections on π_{prj} of its lowest vertices $\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i-1, 0)$ and $\boldsymbol{v}^{(\ell)}_{\text{dtc}}(i, 0)$, by subtracting the projection with smaller horizontal coordinate from the projection with larger horizontal coordinate. As already pointed out, which one is the largest and which one is the smallest depends on the rotation angle θ_ℓ. Similarly to the voxels slices vertices case, we shall define a column vector $\widehat{\boldsymbol{q}}^{\,\text{H,left}}$ containing the indices of the "leftmost" projected endpoints, *i.e.*, the indices of the endpoints with smaller horizontal coordinate of the projections of the lowest horizontal edge of each cell in the first row of the detector, and a column vector $\widehat{\boldsymbol{q}}^{\,\text{H,right}}$ containing the indices of the "rightmost" projected endpoints, *i.e.*, the indices of the endpoints with larger horizontal coordinate of the same projected edge. Both column vectors $\widehat{\boldsymbol{q}}^{\,\text{H,left}}$ and $\widehat{\boldsymbol{q}}^{\,\text{H,right}}$ have m_u elements. According to (i)–(ii), we immediately have that

$$\text{for } \ell \in \mathcal{I}^+_{xz} \cup \mathcal{I}^+_{yz} : \qquad \widehat{\boldsymbol{q}}^{\,\text{H,left}} = (1, 2, \dots, m_u)^T, \qquad \widehat{\boldsymbol{q}}^{\,\text{H,right}} = \widehat{\boldsymbol{q}}^{\,\text{H,left}} + 1\,; \tag{3.61}$$

$$\text{for } \ell \in \mathcal{I}^-_{xz} \cup \mathcal{I}^-_{yz} : \qquad \widehat{\boldsymbol{q}}^{\,\text{H,left}} = (2, 3, \dots, m_u + 1)^T, \quad \widehat{\boldsymbol{q}}^{\,\text{H,right}} = \widehat{\boldsymbol{q}}^{\,\text{H,left}} - 1\,. \tag{3.62}$$

Notice that, to compute the horizontal overlaps, it does not matter the order in which we consider the projections of the cells of the first detector row. Indeed, in the latter case there is no need to reverse also the indices, in order to start again from the cell nearest to the detector reference point L, as it happens in the former case. The setting in (3.62) simply implies that we start the overlap computations always from the first cell of the detector, which in this case is the farthest from L. By using (3.17)–(3.18), (3.20) and the constant c_1 (defined right after (3.20)), we can express both (3.61) and (3.62) by means of a unified formula:

$$\widehat{\boldsymbol{q}}^{\,\text{H,left}} = (1, 2, \dots, m_u)^T + c_1 \quad \text{and} \quad \widehat{\boldsymbol{q}}^{\,\text{H,right}} = \widehat{\boldsymbol{q}}^{\,\text{H,left}} + i_{\text{inc}}\,. \tag{3.63}$$

Once more, because of (3.21) and (3.22), each element in $\boldsymbol{v}^{(\ell)}_{\text{dtc}}(:, 0)$ is uniquely related to its projection onto π_{prj}, which is the homologous element in $\boldsymbol{x}^{\text{prj}}_{\text{dtc},0}$ or $\boldsymbol{y}^{\text{prj}}_{\text{dtc},0}$. As a consequence, we know precisely the indices that correctly address in $\boldsymbol{x}^{\text{prj}}_{\text{dtc},0}$ or $\boldsymbol{y}^{\text{prj}}_{\text{dtc},0}$ the elements that are endpoints of the horizontal segment on π_{prj}, which is the projection of the lowest horizontal edge of each cell in the first detector row. Thanks to (3.63), and regardless of the rotation angle θ_ℓ, the length of the horizontal projected segment we want to compute is *always* given by the difference between the *largest* and the *smallest* coordinates of the projected endpoints pair of the lowest horizontal edge of the cell:

$$\boldsymbol{\nu}_0 = \begin{cases} \boldsymbol{x}^{\text{prj}}_{\text{dtc},0} & \text{if } \pi_{\text{prj}} = Oxz \\ \boldsymbol{y}^{\text{prj}}_{\text{dtc},0} & \text{if } \pi_{\text{prj}} = Oyz \end{cases} \qquad \text{and} \qquad \begin{cases} \boldsymbol{\nu}^{\text{min}}_0 = \boldsymbol{\nu}_0\big(\widehat{\boldsymbol{q}}^{\,\text{H,left}}\big)\,, \\ \boldsymbol{\nu}^{\text{max}}_0 = \boldsymbol{\nu}_0\big(\widehat{\boldsymbol{q}}^{\,\text{H,right}}\big)\,. \end{cases} \tag{3.64}$$

Notice that $\boldsymbol{\nu}_0^{\max}$ and $\boldsymbol{\nu}_0^{\min}$ have one less element than $\boldsymbol{\nu}_0$, *i.e.*, they have as many elements as the number m_u of cells in a detector row (not as the number of vertices). Clearly, we can extend the reasoning to the vertices $\boldsymbol{v}_{\mathrm{dtc}}^{(\ell)}(:,j)$ of each j-th detector vertices row, for all $j=1,\ldots,m_v$. By recalling the ordering of the detector cells vertices, we simply have:

$$\begin{aligned} \boldsymbol{q}^{\mathrm{H,left}} &= \big(1,2,\ldots,m_u(m_v+1)\big)^T + c_1(m_v+1), \\ \boldsymbol{q}^{\mathrm{H,right}} &= \boldsymbol{q}^{\mathrm{H,left}} + i_{\mathrm{inc}}(m_v+1). \end{aligned} \tag{3.65}$$

Both $\boldsymbol{q}^{\mathrm{H,left}}$ and $\boldsymbol{q}^{\mathrm{H,right}}$ are column vectors with

$$N'_{\mathrm{dtc}} = m_u(m_v+1) = N_{\mathrm{dtc}} + m_u \tag{3.66}$$

components. These last relations allow us to give the vectorized definition of the distance-driven detector cells projections which we shall use in the formula (3.1):

$$\boldsymbol{\nu} = \begin{cases} \boldsymbol{x}_{\mathrm{dtc}}^{\mathrm{prj}} & \text{if } \pi_{\mathrm{prj}} = Oxz \\ \boldsymbol{y}_{\mathrm{dtc}}^{\mathrm{prj}} & \text{if } \pi_{\mathrm{prj}} = Oyz \end{cases} \qquad \text{and} \qquad \begin{cases} \boldsymbol{\nu}^{\min} = \boldsymbol{\nu}\big(\boldsymbol{q}^{\mathrm{H,left}}\big)\,, \\ \boldsymbol{\nu}^{\max} = \boldsymbol{\nu}\big(\boldsymbol{q}^{\mathrm{H,right}}\big)\,. \end{cases} \tag{3.67}$$

Notice that $\boldsymbol{\nu}^{\max}$ and $\boldsymbol{\nu}^{\min}$ have exactly m_v+1 less elements than $\boldsymbol{\nu}$, *i.e.*, they are linearly indexed as the detector *cells*, not as their vertices (except for the last m_u positions, which refer to a "virtual" additional row of cells on top of the detector grid, whose lower vertices $\boldsymbol{v}_{\mathrm{dtc}}^{(\ell)}(:,m_v)$ are the upper vertices of the uppermost row). Analogously to the voxels slices projections case, the settings in (3.67) implies that the horizontal lengths of detector cells projections are always given by $\boldsymbol{\nu}^{\max}-\boldsymbol{\nu}^{\min}$, and this is independent of the source position.

Essentially, the previous machinery, which leads to the definition of $\boldsymbol{\xi}^{\max},\boldsymbol{\xi}^{\min}$ and $\boldsymbol{\nu}^{\max},\boldsymbol{\nu}^{\min}$, gives the proof of the following Proposition.

Proposition 4. *Retaining the hypothesis of Proposition 3, the horizontal lengths of the projections onto π_{prj} are given by:*

$$\textit{all object voxels: } \boldsymbol{\xi}^{\max}-\boldsymbol{\xi}^{\min}\,, \qquad \textit{all detector cells: } \boldsymbol{\nu}^{\max}-\boldsymbol{\nu}^{\min}\,,$$

where the vectors $\boldsymbol{\xi}^{\max},\boldsymbol{\xi}^{\min}$ are defined as in (3.59) *and the vectors $\boldsymbol{\nu}^{\max}$,$\boldsymbol{\nu}^{\min}$ are defined as in* (3.67)*. This is independent of the source rotation angle θ.*

Remark 3.7. Observe that the above machinery accounts automatically for the correct mutual orientation of voxels slices projections and detector cells projections, and there is *no need at all* to define any sort of "reference line", usually seen in some non-vectorized approaches to the distance-driven operator computation.

Despite Proposition 4, from a practical point of view, we only need (3.64) to compute the horizontal overlap lengths of voxels and detector cells *in a vectorized fashion*.

Now, consider the generic i-th cell of the first detector row and the projection of its lowest horizontal edge: the endpoints of the projected edge are always $\nu_{0,i}^{\min} < \nu_{0,i}^{\max}$. Consider also a generic p-th voxel slice: the projections of the endpoints of its lower horizontal edge are always $\xi_p^{\min} < \xi_p^{\max}$. Define also the following four differences:

$$g_{pi}^{\mathrm{LL}} = \xi_p^{\min} - \nu_{0,i}^{\min}\,, \qquad g_{pi}^{\mathrm{RL}} = \xi_p^{\max} - \nu_{0,i}^{\min}\,, \tag{3.68}$$

$$g_{pi}^{\mathrm{LR}} = \xi_p^{\min} - \nu_{0,i}^{\max}\,, \qquad g_{pi}^{\mathrm{RR}} = \xi_p^{\max} - \nu_{0,i}^{\max}\,. \tag{3.69}$$

Roughly, the previous four differences have the following meaning:

- g_{pi}^{LL}: difference between the leftmost horizontal projection of the p-th voxel slice and the leftmost horizontal projection of the j-th detector cell in the first row;
- g_{pi}^{RL}: difference between the rightmost horizontal projection of the p-th voxel slice and the leftmost horizontal projection of the j-th detector cell in the first row;
- g_{pi}^{LR}: difference between the leftmost horizontal projection of the p-th voxel slice and the rightmost horizontal projection of the j-th detector cell in the first row;
- g_{pi}^{RR}: difference between the rightmost horizontal projection of the p-th voxel slice and the rightmost horizontal projection of the j-th detector cell in the first row.

Here, as already pointed out, "leftmost" means "with the smallest horizontal coordinate" and "rightmost" means "with the largest horizontal coordinate". By using (3.68)–(3.69), we can quickly identify the overlap cases 1–4:

$$\left(g_{pi}^{\mathrm{LL}} < 0\right) \wedge \left(g_{pi}^{\mathrm{RR}} > 0\right) \Rightarrow \left[\xi_p^{\min}, \xi_p^{\max}\right] \supseteq \left[\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\right] \quad \text{full overlap (case 1)} \tag{3.70}$$

$$\left(g_{pi}^{\mathrm{LL}} \geq 0\right) \wedge \left(g_{pi}^{\mathrm{LR}} < 0\right) \Rightarrow \xi_p^{\min} \in \left[\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\right[\quad \text{partial overlap (cases 2a, 3)} \tag{3.71}$$

$$\left(g_{pi}^{\mathrm{RL}} > 0\right) \wedge \left(g_{pi}^{\mathrm{RR}} \leq 0\right) \Rightarrow \xi_p^{\max} \in \left]\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\right] \quad \text{partial overlap (cases 2b, 3)} \tag{3.72}$$

$$\left(g_{pi}^{\mathrm{LR}} > 0\right) \vee \left(g_{pi}^{\mathrm{RL}} < 0\right) \Rightarrow \left[\xi_p^{\min}, \xi_p^{\max}\right] \cap \left[\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\right] = \emptyset \quad \text{no overlap (case 4)} \tag{3.73}$$

Notice that case 3 is the "intersection" of cases 2a and 2b, *i.e.*, $\xi_p^{\min}, \xi_p^{\max} \in$

$\big[\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\big]$. By recalling that $\xi_p^{\min} < \xi_p^{\max}$ it immediately follows that:

$$\big(g_{pi}^{\mathrm{LL}} \geq 0\big) \wedge \big(g_{pi}^{\mathrm{RR}} \leq 0\big) \;\Rightarrow\; \big[\xi_p^{\min},\, \xi_p^{\max}\big] \subseteq \big[\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\big] \quad \text{partial overlap (case 3)} \tag{3.74}$$

that is, the horizontal voxel projection is completely comprised in the horizontal detector cell projection. This is a subcase of both 2a and 2b. Finally, relations (3.70)–(3.74) imply that the necessary and sufficient condition for the p-th voxel midway slice horizontal projection to have a (strictly) positive overlap length with the horizontal projection of the i-th cell of the detector first row is the following:

$$\big(g_{pi}^{\mathrm{LR}} < 0\big) \wedge \big(g_{pi}^{\mathrm{RL}} > 0\big) \;\Rightarrow\; \big]\xi_p^{\min},\, \xi_p^{\max}\big[\cap \big[\nu_{0,i}^{\min}, \nu_{0,i}^{\max}\big] \neq \emptyset\,. \tag{3.75}$$

Now, we see that, for every fixed $i \in \{1, \ldots, m_u\}$, all the differences (3.68)–(3.69) can be computed, and all the conditions (3.70)–(3.75) can be checked *for all voxels simultaneously.* To this end, we play once more with a suitable indexing. Let us define the following column vectors of differences, each vector having N'_{vxl} elements:

$$\boldsymbol{g}_i^{\mathrm{LL}} = \boldsymbol{\xi}^{\min} - \nu_{0,i}^{\min}\,, \qquad \boldsymbol{g}_i^{\mathrm{RL}} = \boldsymbol{\xi}^{\max} - \nu_{0,i}^{\min}\,, \tag{3.76}$$

$$\boldsymbol{g}_i^{\mathrm{LR}} = \boldsymbol{\xi}^{\min} - \nu_{0,i}^{\max}\,, \qquad \boldsymbol{g}_i^{\mathrm{RR}} = \boldsymbol{\xi}^{\max} - \nu_{0,i}^{\max}\,. \tag{3.77}$$

We define also the following subsets of the indices set $\mathcal{I}'_{\mathrm{vxl}} = \big\{1, 2, \ldots, N'_{\mathrm{vxl}}\big\}$ (also referred to as the *extended set of voxels indices*), and their corresponding column vectors:

$$\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i} = \Big\{p \in \mathcal{I}'_{\mathrm{vxl}} \,\Big|\, \big(g_{pi}^{\mathrm{LR}} < 0\big) \wedge \big(g_{pi}^{\mathrm{RL}} > 0\big)\Big\}\,, \qquad \boldsymbol{p}^{\mathrm{H},i} = \mathrm{vect}\big(\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i}\big)\,, \tag{3.78}$$

$$\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,1} = \Big\{p \in \mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i} \,\Big|\, \big(g_{pi}^{\mathrm{LL}} < 0\big) \wedge \big(g_{pi}^{\mathrm{RR}} > 0\big)\Big\}\,, \qquad \boldsymbol{p}^{\mathrm{H},i,1} = \mathrm{vect}\big(\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,1}\big)\,, \tag{3.79}$$

$$\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,3} = \Big\{p \in \mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i} \,\Big|\, \big(g_{pi}^{\mathrm{RL}} > 0\big) \wedge \big(g_{pi}^{\mathrm{RR}} \leq 0\big)\Big\}\,, \qquad \boldsymbol{p}^{\mathrm{H},i,3} = \mathrm{vect}\big(\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,3}\big)\,, \tag{3.80}$$

$$\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,2} = \Big\{p \in \mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i} \,\Big|\, \big(g_{pi}^{\mathrm{LL}} \geq 0\big) \wedge \big(g_{pi}^{\mathrm{LR}} < 0\big)\Big\} \setminus \mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,3}\,, \; \boldsymbol{p}^{\mathrm{H},i,2} = \mathrm{vect}\big(\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i,2}\big)\,, \tag{3.81}$$

where the vect$(\cdot)$ operator applied to a numeric set $\mathcal{B}$ means that we build the column vector by stacking the elements of the set. Usually, there is no no particular order for the elements in $\mathcal{B}$, but if the set is a set $\mathcal{I}$ of indices, then we *usually* suppose to stack them sorted in ascending order. We underline that this sorting is not at all relevant, neither in the following, nor for computational reasons. It just helps a little in following the reasoning. Finally, we define the following indices set:

$$\widehat{\mathcal{I}}_{\mathrm{vxl}}^{\mathrm{H},i} = \Big\{1, 2, \ldots, N_{\mathrm{vxl}}^{\mathrm{H},i}\Big\}\,, \qquad \text{where} \quad N_{\mathrm{vxl}}^{\mathrm{H},i} = \#\mathcal{I}_{\mathrm{vxl}}^{\mathrm{H},i}\,. \tag{3.82}$$

The following result, which has a straightforward proof, basically says that (3.78)–(3.81) are well defined.

Lemma 3.8. *For all $i \in \{1, \dots, m_u\}$, the three sets $\mathcal{I}_{vxl}^{H,i,1}$, $\mathcal{I}_{vxl}^{H,i,2}$, and $\mathcal{I}_{vxl}^{H,i,3}$ defined in* (3.79), (3.81), *and* (3.80), *respectively, are a partition of the set $\mathcal{I}_{vxl}^{H,i}$, i.e., they are mutually disjoint and satisfy*

$$\mathcal{I}_{vxl}^{H,i,1} \cup \mathcal{I}_{vxl}^{H,i,2} \cup \mathcal{I}_{vxl}^{H,i,3} = \mathcal{I}_{vxl}^{H,i} .$$

Proof. The second conditions in (3.79) and (3.80) are mutually exclusive, so $\mathcal{I}_{\text{vxl}}^{\text{H},i,1} \cap \mathcal{I}_{\text{vxl}}^{\text{H},i,3} = \emptyset$. The first conditions in (3.79) and (3.81) are mutually exclusive, thus $\mathcal{I}_{\text{vxl}}^{\text{H},i,1} \cap \mathcal{I}_{\text{vxl}}^{\text{H},i,2} = \emptyset$. From the definition in (3.81) it follows $\mathcal{I}_{\text{vxl}}^{\text{H},i,2} \cap \mathcal{I}_{\text{vxl}}^{\text{H},i,3} = \emptyset$. The last condition in the Lemma is an immediate consequence of the four definitions (3.79)-(3.81) and (3.78). □

We are now ready to piece together everything and compute *in a vectorized fashion* the horizontal overlaps between all voxels projections and the projection of the i-th detector cell:

$$\boldsymbol{h}^{\text{H},i,1} = \begin{cases} g_i^{\text{H}} \mathbf{1}_{N_{\text{vxl}}^{\text{H},i} \times 1} & \text{if } \mathcal{I}_{\text{vxl}}^{\text{H},i,1} \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases} \tag{3.83}$$

$$\boldsymbol{h}^{\text{H},i,2} = \begin{cases} \min\left\{ \boldsymbol{\xi}^{\max}\left(\boldsymbol{p}^{\text{H},i,2}\right), \nu_{0,i}^{\max} \right\} - \boldsymbol{\xi}^{\min}\left(\boldsymbol{p}^{\text{H},i,2}\right) & \text{if } \mathcal{I}_{\text{vxl}}^{\text{H},i,2} \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases} \tag{3.84}$$

$$\boldsymbol{h}^{\text{H},i,3} = \begin{cases} \boldsymbol{\xi}^{\max}\left(\boldsymbol{p}^{\text{H},i,3}\right) - \max\left\{ \boldsymbol{\xi}^{\min}\left(\boldsymbol{p}^{\text{H},i,3}\right), \nu_{0,i}^{\min} \right\} & \text{if } \mathcal{I}_{\text{vxl}}^{\text{H},i,3} \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases} \tag{3.85}$$

where

$$g_i^{\text{H}} = \nu_{0,i}^{\max} - \nu_{0,i}^{\min}$$

is the horizontal length of the projection onto π_{prj} of the i-th cell of the first detector row. Once again, the `min` and `max` operators applied to vector arguments are intended to work component-wise. Lastly, we reorder the components in $\boldsymbol{p}^{\text{H},i}$:

$$\boldsymbol{p}^{\text{H},i} \leftarrow \text{vect}\left(\boldsymbol{p}^{\text{H},i,1}, \boldsymbol{p}^{\text{H},i,2}, \boldsymbol{p}^{\text{H},i,3}\right) \tag{3.86}$$

and we finally set

$$\boldsymbol{h}^{\text{H},i} = \text{vect}\left(\boldsymbol{h}^{\text{H},i,1}, \boldsymbol{h}^{\text{H},i,2}, \boldsymbol{h}^{\text{H},i,3}\right) . \tag{3.87}$$

Once again, notice that the elements of $\boldsymbol{p}^{\text{H},i}$ span in $\mathcal{I}'_{\text{vxl}}$, so the indexing of $\boldsymbol{h}^{\text{H},i}$ follows the same indexing of the (extended) voxels set, not the one of the slices vertices.

This is a first milestone in the computation process: for the given i-th detector cell in the first row, the vector $\boldsymbol{p}^{\text{H},i}$ contains the indices of all the object voxels whose projections onto π_{prj} have a *horizontal* overlap of non null length with the projection of the i-th cell. However, as we already pointed out in (3.23), all the other m_v detector cells in the same i-th column have the same *horizontal* projection as the i-th cell in the first row, *i.e.*, $x^{\mathrm{prj}}_{\mathrm{dtc}}(i,j) = x^{\mathrm{prj}}_{\mathrm{dtc}}(i,0)$ (or $y^{\mathrm{prj}}_{\mathrm{dtc}}(i,j) = y^{\mathrm{prj}}_{\mathrm{dtc}}(i,0)$) for all $j = 1,\dots,m_v$. As a consequence, for a fixed i-th column of detector cells, in the column vector $\boldsymbol{h}^{\mathrm{H},i}$ we have at disposal the horizontal length of the overlaps of *all* and *only* those voxels whose projections overlap the projection of at least one cell of the i-th column of the detector.

Now, for every fixed i-th column of detector cells, we need to assign correctly the *vertical* position of each projected horizontal edge of voxels slices, in order to exactly identify to which projected detector cell it belongs. The idea is to compare, starting from the very bottom layer of voxels, the vertical coordinate of the voxels vertices projections against the vertical coordinates of the projections of the cells of the i-th detector column. For each voxel horizontal projection, this provides the *index* of the detector row whose projection contains the given voxel projection.

The key for efficiency is to perform, for each fixed i, the above described computations *simultaneously* for all the voxels with linear index in $\boldsymbol{p}^{\mathrm{H},i}$, *i.e.*, by accomplishing this task in a *vectorized fashion*. Once again, this can be done by working with a suitable indexing and using Proposition 3.

So, let us consider a given i-th column of detector cells, for a fixed $i \in \{1,\dots,m_u\}$. First of all, we define the index set $\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl,btm}}$ which contains the elements of $\boldsymbol{p}^{\mathrm{H},i}$ which identify the projections of the voxels slices vertices belonging to the lowest layer. Because of the numbering scheme, they are the elements of $\boldsymbol{p}^{\mathrm{H},i}$ which do not exceed $N^{\mathrm{lyr}}_{\mathrm{vxl}}$:

$$\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl,btm}} = \left\{p \in \mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl}} \,\middle|\, p \le N^{\mathrm{lyr}}_{\mathrm{vxl}}\right\} \qquad \boldsymbol{p}^{\mathrm{H},i}_{\mathrm{btm}} = \mathrm{vect}\big(\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl,btm}}\big)\,. \tag{3.88}$$

The next step is to identify the voxels slices vertices of the other layers. Due to item (1) in Proposition 3, we know that, if a vertex $\boldsymbol{v}^{(\ell)}_{\mathrm{vxl}}(i_1,i_2,0)$ in the lowest layer has the horizontal coordinate of its projection in a given position, also the projections of all the vertices $\boldsymbol{v}^{(\ell)}_{\mathrm{vxl}}(i_1,i_2,k)$, $k = 1,\dots,n_z$, of the other layers having the same horizontal position have the same horizontal coordinate. Here we use the notation “i_1, i_2” in place of “i, j” just to avoid possible misunderstandings with the rest of the discussion. Hence, once we know that the horizontal projection of a voxel slice vertex of the lowest layer falls inside the projection of the i-th detector cell, we only need to address the homologous voxels slices vertices aligned along the same line parallel to the z axis. As we have already discussed in (3.55), the linear indices of the projections of all these vertices have a stride of exactly $N^{\mathrm{lyr}}_{\mathrm{vxl,vrt}}$

positions. Hence, we define the column vector $\widetilde{\boldsymbol{p}}^{\mathrm{H}}$ with $n_z + 1$ elements:

$$\widetilde{\boldsymbol{p}}^{\mathrm{H}} = \begin{pmatrix} \widetilde{p}_0^{\mathrm{H}} \\ \widetilde{p}_1^{\mathrm{H}} \\ \vdots \\ \widetilde{p}_{n_z}^{\mathrm{H}} \end{pmatrix} = N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ n_z \end{pmatrix} . \tag{3.89}$$

Thus, for every $p \in \mathcal{I}_{\mathrm{vxl,btm}}^{\mathrm{H},i}$ the linear indices of all the contiguous voxels slices vertices "stacked" on top of it in the other voxels layers can be immediately obtained by $p + \widetilde{\boldsymbol{p}}^{\mathrm{H}}$. However, this is *not enough* to correctly address the right elements in $\boldsymbol{x}_{\mathrm{vxl}}^{\mathrm{prj}}$ (or $\boldsymbol{y}_{\mathrm{vxl}}^{\mathrm{prj}}$), as the following remark points out.

Remark 3.9. Even if the total number of slices vertices in each voxels layer is equal to $N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}}$, and $N_{\mathrm{vxl,vrt}}^{\mathrm{lyr}} > N_{\mathrm{vxl}}^{\mathrm{lyr}} = n_x n_y$ (*i.e.*, the total number of slices vertices in each voxels layer exceeds the number of voxels per layer) in both cases $\pi_{\mathrm{prj}} = Oxz$ and $\pi_{\mathrm{prj}} = Oyz$, in general the positions of the stored values differ in number and position, depending on whether $\pi_{\mathrm{prj}} = Oxz$ or $\pi_{\mathrm{prj}} = Oyz$:

- case $\pi_{\mathrm{prj}} = Oxz$ ($c_x = 1$):
  in this case, the midway slices of the voxels are parallel to Oxz, hence for each horizontal layer we have exactly n_y vertices in the y axis direction and $n_x + 1$ vertices in the x axis direction. Because of the vectorization ordering, for each horizontal layer, in $\boldsymbol{x}_{\mathrm{vxl}}^{\mathrm{prj}}$ and $\boldsymbol{z}_{\mathrm{vxl}}^{\mathrm{prj}}$ the projections of the first $N_{\mathrm{vxl}}^{\mathrm{lyr}}$ vertices have contiguous positions with consecutive linear indices, then additional n_y elements follows for the same layer, corresponding to the projections of the rightmost line of n_y vertices, that is, those vertices with larger x coordinate in the layer;

- case $\pi_{\mathrm{prj}} = Oyz$ ($c_y = 1$):
  in this case, the planes of the voxels slices are parallel to Oyz, hence for each horizontal layer we have exactly n_x vertices in the x axis direction and $n_y + 1$ vertices in the y axis direction. Because of the vectorization ordering, for each horizontal layer, in $\boldsymbol{y}_{\mathrm{vxl}}^{\mathrm{prj}}$ and $\boldsymbol{z}_{\mathrm{vxl}}^{\mathrm{prj}}$ there is one additional element after each column of n_y contiguous positions, for a total of n_x additional elements for each layer. These additional elements are the projections of the rightmost vertices with respect to the y axis, *i.e.*, the vertices with the largest y coordinate in the layer (those further from the Oxz plane).

In the following, to avoid confusion in the discussion, we shall use the term "stack" for a *vertical* column of contiguous voxels (or vertices), *i.e.*, contiguous elements along the z direction.

In practice, to correctly address the right elements in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$, $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$ and $\boldsymbol{z}_{\text{vxl}}^{\text{prj}}$, we need to *convert voxels linear indices into vertices linear indices.* This conversion can be easily understood by means of an *injective* map which associates each voxel to the lower-left vertex of its midway slice:

$$\begin{array}{rcl} \mathcal{N}'_{\text{vxl}} & \longrightarrow & \mathcal{V}_{\text{vxl}} \\ \boldsymbol{n} = (i,j,k) & \longmapsto & \boldsymbol{v}_{\text{vxl}}^{(\ell)}(i,j,k) \end{array} \tag{3.90}$$

for each $i \in \{1,\dots,n_x\}$, $j \in \{1,\dots,n_y\}$, $k \in \{1,\dots,n_z+1\}$; $\mathcal{N}'_{\text{vxl}}$ will be introduced shortly. The map (3.90) leaves the rightmost vertices[1] of each horizontal layer "unassociated", and this is consistent with the intention to avoid double associations of the rightmost voxels and explains why the map *is not surjective.* However, the uppermost level of vertices would not be mapped, and this would be wrong. For this reason, to make (3.90) consistent and well defined, we define the *extended* set $\mathcal{N}'_{\text{vxl}}$ of voxels by considering an additional layer of "virtual" voxels (the (n_z+1)-th *voxels* layer), whose lower-left corners of the midway slices are indeed the points of the uppermost *vertices* layer $\big($*i.e.*, $\boldsymbol{v}_{\text{vxl}}^{(\ell)}(:,:,n_z)\big)$.

Remark 3.9 suggests how to take into account these extra components, when computing the indices of contiguous vertices in a stack. If $\pi_{\text{prj}} = Oxz$, we need to add n_y to each voxel index $p \in \mathcal{I}_{\text{vxl}}^{\text{H},i}$ as many times as the number of horizontal voxels layers below the layer containing the voxel itself. On the other hand, when $\pi_{\text{prj}} = Oyz$, we need to add to each voxel index $p \in \mathcal{I}_{\text{vxl}}^{\text{H},i}$ the total number of voxels columns (counted over all horizontal layers) preceding the column containing the voxel itself. Once we define the indices set

$$\mathcal{I}_{\text{vxl}}^{\text{prj}} = \left\{1,2,\dots,N_{\text{vxl}}^{\text{prj}}\right\}, \tag{3.91}$$

we can set up the linear indices conversion. Given any voxel index $p \in \mathcal{I}_{\text{vxl}}^{\text{H},i}$, the linear index $\overline{p} \in \mathcal{I}_{\text{vxl}}^{\text{prj}}$ of the corresponding element in $\boldsymbol{x}_{\text{vxl}}^{\text{prj}}$ (or $\boldsymbol{y}_{\text{vxl}}^{\text{prj}}$) and $\boldsymbol{z}_{\text{vxl}}^{\text{prj}}$ is:

$$\overline{p} = p + \begin{cases} n_y \left\lfloor (p-1)/N_{\text{vxl}}^{\text{lyr}} \right\rfloor & \text{if } c_x = 1 \text{ (case } \pi_{\text{prj}} = Oxz) \\ \left\lfloor (p-1)/n_y \right\rfloor & \text{if } c_y = 1 \text{ (case } \pi_{\text{prj}} = Oyz) \end{cases} \tag{3.92}$$

By using the selection flags c_x and c_y, we are finally able to define the linear index conversion as the following index map associated to (3.90):

$$\begin{array}{rcl} \mathcal{I}'_{\text{vxl}} & \longrightarrow & \mathcal{I}_{\text{vxl}}^{\text{prj}} \\ p & \longmapsto & \overline{p} = p + c_x n_y \left\lfloor (p-1)/N_{\text{vxl}}^{\text{lyr}} \right\rfloor + c_y \left\lfloor (p-1)/n_y \right\rfloor \end{array} \tag{3.93}$$

[1] Just remember that these are $\boldsymbol{v}_{\text{vxl}}(n_x+1,:,:)$ when $\pi_{\text{prj}} = Oxz$, and $\boldsymbol{v}_{\text{vxl}}(:,n_y+1,:)$ when $\pi_{\text{prj}} = Oyz$.

It follows immediately that we can apply the map (3.93) simultaneously to all indices $p \in \mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl}}$, thus obtaining in a *vectorized fashion* the indices vector $\overline{\boldsymbol{p}}^{\,\mathrm{H},i}$ from the vector $\boldsymbol{p}^{\mathrm{H},i}$:

$$\overline{\boldsymbol{p}}^{\,\mathrm{H},i} = \boldsymbol{p}^{\mathrm{H},i} + c_x n_y \left\lfloor (p-1)/N^{\mathrm{lyr}}_{\mathrm{vxl}} \right\rfloor + c_y \left\lfloor (p-1)/n_y \right\rfloor . \tag{3.94}$$

The vector $\overline{\boldsymbol{p}}^{\,\mathrm{H},i}$ correctly addresses the components of $\boldsymbol{x}^{\mathrm{prj}}_{\mathrm{vxl}}$ (or $\boldsymbol{y}^{\mathrm{prj}}_{\mathrm{vxl}}$) and $\boldsymbol{z}^{\mathrm{prj}}_{\mathrm{vxl}}$ corresponding to all those voxels slices vertices whose projections are involved in the horizontal overlaps with the projection of the i-th column of detector cells. In practice, we have a “pointer” from the elements of $\boldsymbol{h}^{\mathrm{H},i}$ to the voxels projections.

Now, we shall see how to correctly map each selected horizontal overlap in $\boldsymbol{h}^{\mathrm{H},i}$ to the right projection of the cells of the i-th detector column. To this end, we need to consider the vertical coordinate of the projections.

We define an auxiliary matrix $Z_{\mathrm{H},i}$, sized $(n_z+1) \times N^{\mathrm{H},i}_{\mathrm{vxl,btm}}$, whose columns contain the vertical coordinates of the projections of the voxels slices vertices with linear index in $\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl}}$. These are exactly the $N^{\mathrm{H},i}_{\mathrm{vxl,btm}}$ vertical stacks of vertices whose horizontal coordinates $\boldsymbol{x}^{\mathrm{prj}}_{\mathrm{vxl}}(\overline{\boldsymbol{p}}^{\,\mathrm{H},i})$ (or $\boldsymbol{y}^{\mathrm{prj}}_{\mathrm{vxl}}(\overline{\boldsymbol{p}}^{\,\mathrm{H},i})$) of their projections falls inside the horizontal interval $\left[\boldsymbol{\nu}^{\min}_{0,i}, \boldsymbol{\nu}^{\max}_{0,i}\right]$, *i.e.*, inside the interval of the horizontal projection of the cells of the i-th detector column. Recall that contiguous vertices aligned along the same vertical line of voxels slices edges have their projection coordinates stored in elements with a stride of exactly $N^{\mathrm{lyr}}_{\mathrm{vxl,vrt}}$ positions. Hence, to build each column of $Z_{\mathrm{H},i}$, we only need to start from the vertical projection of a vertex in the bottom layer and then collect from $\boldsymbol{z}^{\mathrm{prj}}_{\mathrm{vxl}}$ all the other n_z projections with constant stride $N^{\mathrm{lyr}}_{\mathrm{vxl,vrt}}$. By using (3.89), this operations can be efficiently vectorized:

$$Z_{\mathrm{H},i}(:,s) = \boldsymbol{z}^{\mathrm{prj}}_{\mathrm{vxl}}\Big(\overline{\boldsymbol{p}}^{\,\mathrm{H},i}\big(\boldsymbol{p}^{\mathrm{H},i}_{\mathrm{btm}}(s)\big) + \widetilde{\boldsymbol{p}}^{\,\mathrm{H}}\Big) \qquad \forall s = 1,\dots,N^{\mathrm{H},i}_{\mathrm{vxl,btm}}\,, \tag{3.95}$$

where, for sake of clarity, we use the pair notation “(r,s)” in place of the subscripts “$_{rs}$” to address the elements of matrices and vectors.

Roughly, the matrix $Z_{\mathrm{H},i}$ collects the vertical coordinates of all the voxels projections with a nonzero horizontal overlap with the projection of the i-th detector column. From a computational point of view, it is useful to pre-allocate the matrix $Z_{\mathrm{H},i}$ before filling it up with the values.

Remark 3.10. Notice that the values stored in $\boldsymbol{z}^{\mathrm{prj}}_{\mathrm{vxl}}$, and hence in $Z_{\mathrm{H},i}$, can be smaller than $\min\{\boldsymbol{z}^{\mathrm{prj}}_{\mathrm{dtc}}\}$ or larger than $\max\{\boldsymbol{z}^{\mathrm{prj}}_{\mathrm{dtc}}\}$ (*i.e.*, the minimum and the maximum vertical coordinates of the detector cells projections, respectively), namely they can fall *outside* the projection of the detector. When this happens, we *must* consider also these “outside” values, because they *could* be endpoints of vertical edges of projected areas which overlap the detector projection. Whether this is the case or not, will be decided by the next computations.

Now, we need to switch from vertical coordinates to row indices for the i-th detector column of cells. Recall that, because of (3.37)–(3.40), for each column of detector cells vertices, their projections are equally spaced. Let us call $\tau_i^{(\ell)}$ the vertical gap:

$$\tau_i^{(\ell)} = z_{\text{dtc}}^{\text{prj}}(i,1) - z_{\text{dtc}}^{\text{prj}}(i,0)\,. \tag{3.96}$$

Then, the switch is easily accomplished by:

(i) subtracting the smaller vertical projection of the detector from all the elements of $Z_{\text{H},i}$,

(ii) dividing each r-th resulting row by $\tau_i^{(\ell)}$,

(iii) take the floor of the result and add 1.

We get:

$$Z_{\text{H},i}^{\text{idx}} = \left\lfloor \big(Z_{\text{H},i} - z_{\text{dtc}}^{\text{prj}}(i,0)\big) / \tau_i^{(\ell)} \right\rfloor + 1 \tag{3.97}$$

where all these operations are Hadamard operations, *i.e.*, they are performed component-wise. The elements of $Z_{\text{H},i}^{\text{idx}}$ are integer numbers, possibly negative or even zero. Non-positive elements refer to positions below the projected bottom edge of the detector, positive elements greater than m_v refer to positions above the projected top edge of the detector. In both cases, these voxels projections fall outside the detector.

Now, we use the indices in $Z_{\text{H},i}^{\text{idx}}$ to identify the voxels and the detector cells to which each horizontal overlap refers. In practice, we need to find the right position to which every given horizontal overlap contributes for the construction of the final CT forward operator. The matrix $Z_{\text{H},i}^{\text{idx}}$ has a key role for the identification of these positions, as we shall see shortly.

We have a bunch of possible situations similar to those described in 1–4 for the horizontal overlaps, but this time for the vertical direction. Indeed, the vertical height of the projection of a voxel onto π_{prj} can either:

5a) partially overlap the bottom row or the top row of detector cell projections;

5b) fall completely inside the projection of one single detector cell;

5c) have one of (or both) the horizontal edges overlapped to one of (or both) the horizontal edges of the projection of a single detector cell;

5d) span the projections of exactly two adjacent detector cells in the same column;

5e) span the projections of more than two vertically adjacent detector cells;

5f) fall completely outside the projection of the whole column of detector cells.

All these situations can be addressed by using the indices in $Z_{\text{H},i}^{\text{idx}}$. In the following, it is crucial to bear in mind that the length of the horizontal overlap of the projections of all the voxels in a stack is the same as the one of the bottom voxel in the stack. Thus, once we know the positions in which we shall store such a length, we only need to *replicate* the overlap length of the very bottom voxel in those positions.

Consider the s-th stack of voxels slices vertices projections whose bottom element has linear index belonging to $\mathcal{I}^{\mathrm{H},i}_{\mathrm{vxl,btm}}$. The vertical coordinates of all their projections constitute the s-th column of $Z_{\mathrm{H},i}$ (with the bottom value stored in the first row and the top value stored in the last row). Thus, consider the s-th column of $Z^{\mathrm{idx}}_{\mathrm{H},i}$ and define two sets of row indices:

$$\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{btm},s} = \left\{r \in \{1, \ldots, n_z + 1\} \,\middle|\, Z^{\mathrm{idx}}_{\mathrm{H},i}(r,s) > 0\right\}, \tag{3.98}$$

$$\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{top},s} = \left\{r \in \{1, \ldots, n_z + 1\} \,\middle|\, Z^{\mathrm{idx}}_{\mathrm{H},i}(r,s) \leq m_v\right\}. \tag{3.99}$$

The set $\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{btm},s}$ selects those rows in the s-th column of $Z^{\mathrm{idx}}_{\mathrm{H},i}$, and hence in the s-th column of $Z_{\mathrm{H},i}$, corresponding to voxels in the s-th stack whose midway slices have the z coordinate of the projection of their lower horizontal edge greater than the z coordinate of the projection of the bottom edge of the detector. The set $\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{top},s}$ selects the rows indices corresponding to voxels in the same s-th stack whose midway slices have the z coordinate of the projection of their upper horizontal edge not greater than the z coordinate of the projection of the detector top edge. The intersection of these two sets provides indices somehow related to those voxels in the s-th stack whose projection has a nonempty overlap with the projection of the i-th detector column. The next step is to precisely identify these voxels. We define two indices:

$$\left(p^{\mathrm{H,idx},i}_{\min}\right)_s = \max\left\{\min\left\{\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{btm},s}\right\} - 1,\ 1\right\} \tag{3.100}$$

$$\left(p^{\mathrm{H,idx},i}_{\max}\right)_s = \min\left\{\max\left\{\mathcal{I}^{\mathrm{H,idx,i}}_{\mathrm{btm},s}\right\},\ n_z\right\} \tag{3.101}$$

The index $\left(p^{\mathrm{H,idx},i}_{\min}\right)_s$ identifies, in the s-th column of $Z_{\mathrm{H},i}$, the row index of the lower voxel in the s-th stack, whose projection in vertical direction *reaches* (*i.e.*, overlaps from below, at least partially) the projection of the first cell in the i-th column of the detector. Similarly, the index $\left(p^{\mathrm{H,idx},i}_{\max}\right)_s$ identifies, in the same s-th column of $Z_{\mathrm{H},i}$, the row index of the higher voxel in the s-th stack, whose projection in vertical direction *reaches* (again, overlaps from below, at least partially) the projection of the last cell in the i-th column of the detector. Hence, the indices r such that

$$\left(p^{\mathrm{H,idx},i}_{\min}\right)_s \leq r \leq \left(p^{\mathrm{H,idx},i}_{\max}\right)_s + 1 \tag{3.102}$$

identify *all and only* those voxels[2] in the s-th stack whose projection overlaps the projection of the i-th column of the detector.

Remark 3.11. We stress that $Z^{\mathrm{idx}}_{\mathrm{H},i}$ has a key relevance: it embeds, at the same time, voxels and detector cells information, and connects them. Indeed, each s-th column identifies a single voxel stack, the row index in the column identifies

[2] We just remind that the indices in (3.102) are actually vertices indices (of voxels slices), but two adjacent such vertices identify a voxel.

the r-th voxel in the stack and the value of the integer number j in the (r, s)-th position identifies the (possibly "virtual") cell in the i-th column of the detector, whose projection has a nonempty intersection with the projection of the voxel. The horizontal length of this nonempty intersection is stored in the corresponding element of $\boldsymbol{h}^{\mathrm{H},i}$.

Now, we recover from $Z_{\mathrm{H},i}^{\mathrm{idx}}$ all the information we need to store the correct value in the right position of $\mathbf{W}_{\mathrm{H}}$. First, we shall take into account the cases 5a–5f: this is easily accomplished by computing the difference of consecutive elements of the same s-th column. The components of the resulting vector tell us how many consecutive projected cells in the i-th column of the detector projection are overlapped by the projection of the *same* voxel, *in addition* to the cell corresponding to the first overlap. Clearly, this is meaningful only for those components corresponding to nonempty overlaps, *i.e.*, whose index is in the range (3.102). If the number of additional cells spanned by the projection of the voxels is greater than zero, then we have to replicate the corresponding horizontal overlap in consecutive rows of the same column of $\mathbf{W}_{\mathrm{H}}$. To this end, we define the vector $\boldsymbol{\sigma}_s^{\mathrm{H,idx},i} = \left(\sigma_{r,s}^{\mathrm{H,idx},i}\right)$ with

$$\sigma_{r,s}^{\mathrm{H,idx},i} = Z_{\mathrm{H},i}^{\mathrm{idx}}(r+1, s) - Z_{\mathrm{H},i}^{\mathrm{idx}}(r, s) \qquad \forall r \in \left\{ \left(p_{\min}^{\mathrm{H,idx},i}\right)_s, \dots, \left(p_{\max}^{\mathrm{H,idx},i}\right)_s \right\}. \quad (3.103)$$

Remark 3.12. Notice that the explicit construction of the matrix $Z_{\mathrm{H},i}$ is not needed: all computations can be equally performed "in place" on the projection vectors, by suitably using the indices vector $\overline{\boldsymbol{p}}^{\mathrm{H},i}$. However, having at disposal the needed stacks of vertical coordinates grouped together in the columns of $Z_{\mathrm{H},i}$ allows the use of the effective Matlab function `diff`, in order to compute (3.103).

Next, we need to compute the right row and column indices to store the horizontal overlap in $\mathbf{W}_{\mathrm{H}}$. We remind that in the CT forward operator the column index is the voxel linear index of the column-major vectorization of the voxels grid (not that of the vertices), while the row index identifies the linear index of the column-major vectorization of the detector cells (not that of their vertices) at the given ℓ-th rotation angle.

So, we first have to map the stack row indices to the "global" linear indices. In the s-th stack, the global linear index $p_r^{\mathrm{vxl},H}$ of each r-th voxel can be easily obtained by computing the suitable displacement from the global linear index of the voxel at the bottom of the stack, which is

$$p_{\mathrm{btm},s}^{\mathrm{H,col},i} = \boldsymbol{p}^{\mathrm{H},i}\left(p_{\mathrm{btm},s}^{\mathrm{H},i}\right) \qquad \forall s \in \left\{1, \dots, N_{\mathrm{vxl,btm}}^{\mathrm{H},i}\right\}, \qquad (3.104)$$

as we shown in (3.88). Hence, given that the row index in the stack identifies the voxel horizontal layer, we have

$$p_r^{\mathrm{vxl},H} = p_{\mathrm{btm},s}^{\mathrm{H,col},i} + (r-1)N_{\mathrm{vxl}}^{\mathrm{lyr}} \qquad \forall r \in \left\{ \left(p_{\min}^{\mathrm{H,idx},i}\right)_s, \dots, \left(p_{\max}^{\mathrm{H,idx},i}\right)_s \right\} \qquad (3.105)$$

Next, we have to map the integer values in $Z^{\mathrm{idx}}_{\mathrm{H},i}$, and the corresponding replications in $\boldsymbol{\sigma}^{\mathrm{H,idx},i}$, to "global" row indices of $\mathbf{W}_{\mathrm{H}}$. We recall that, for the detector cells projections at the current ℓ-th rotation angle, we proceeded along the u axis, browsing in a vectorized fashion one i-th detector column at the time. Hence, all row indices j involved in the computations for this i-th column of the detector must be mapped to the row indices in $\mathbf{W}_{\mathrm{H}}$ with the usual offset given by

$$\widetilde{q}_i^{\mathrm{dtc}} = (i-1)m_v \qquad \forall i \in \{1, \dots, m_u\}\,. \tag{3.106}$$

As we already pointed out, the projection of each given r-th voxel in the fixed s-th stack overlaps (perhaps only partially) the projections of none, one or more than one contiguous cells in the i-th column of the detector. According to (3.103), the *extended* row indices (*i.e.*, possibly smaller than one or greater than m_v) of these cells are in the range from $Z^{\mathrm{idx}}_{\mathrm{H},i}(r,s)$ to $Z^{\mathrm{idx}}_{\mathrm{H},i}(r,s) + \sigma^{\mathrm{H,idx},i}_{r,s}$. However, to get the right indices of actual cells we must threshold this range to the set $\{1, \dots, m_v\}$:

$$q^{\mathrm{dtc},i}_{\mathrm{min},r,s} = \max\Big\{1,\ Z^{\mathrm{idx}}_{\mathrm{H},i}\big(r,s\big)\Big\}\,, \tag{3.107}$$

$$q^{\mathrm{dtc},i}_{\mathrm{max},r,s} = \min\Big\{m_v,\ Z^{\mathrm{idx}}_{\mathrm{H},i}\big(r,s\big) + \sigma^{\mathrm{H,idx},i}_{r,s}\Big\}\,. \tag{3.108}$$

It follows that the row indices of the column $p^{\mathrm{vxl},H}_r$ in $\mathbf{W}_{\mathrm{H}}$ in which *replicating* the horizontal length of overlap are

$$\boldsymbol{q}^{\mathrm{dtc},i}_{rs} = \begin{pmatrix} q^{\mathrm{dtc},i}_{\mathrm{min},r,s} \\ \vdots \\ q^{\mathrm{dtc},i}_{\mathrm{max},r,s} \end{pmatrix} + \widetilde{q}_i^{\mathrm{dtc}} \qquad \forall r \in \Big\{\big(p^{\mathrm{H,idx},i}_{\mathrm{min}}\big)_s, \dots, \big(p^{\mathrm{H,idx},i}_{\mathrm{max}}\big)_s\Big\}\,. \tag{3.109}$$

Finally, we have at disposal everything we need to correctly fill the selected positions of $\mathbf{W}_{\mathrm{H}}$ with the right horizontal length of overlap between voxels projections and detector cells projections for a fixed rotation angle index ℓ:

$$\mathbf{W}_{\mathrm{H}}\big(\boldsymbol{q}^{\mathrm{dtc},i}_{rs},\, p^{\mathrm{vxl},H}_r\big) = \boldsymbol{h}^{\mathrm{H},i}\big(p^{\mathrm{H},i}_{\mathrm{btm},s}\big) \tag{3.110}$$

for all $i \in \big\{1, \dots, m_u\big\}$, $s \in \Big\{1, \dots, N^{\mathrm{H},i}_{\mathrm{vxl,btm}}\Big\}$, $r \in \Big\{\big(p^{\mathrm{H,idx},i}_{\mathrm{min}}\big)_s, \dots, \big(p^{\mathrm{H,idx},i}_{\mathrm{max}}\big)_s\Big\}$. Notice that, by construction, $\boldsymbol{q}^{\mathrm{dtc},i}_{rs}$ has at least one element (which is $q^{\mathrm{dtc},i}_{\mathrm{min},r,s}$), and if it has more than one element, equation (3.110) means that the value on the right-hand side (which is a single real number) is assigned simultaneously to all those rows of the column $p^{\mathrm{vxl},H}_r$.

The complete discussion we end up in this Subsection gives the constructive proof of the following Proposition, which is the main result on the vectorized approach to the horizontal lengths computation of the overlaps of voxel and detector cells projections.

Proposition 3.13. *Let $\mathcal{R}_{\mathcal{C}}\big(O, S, d_{CO'}, \gamma_{\min}, \gamma_{\max}, \beta_{\min}, \beta_{\max}\big)$ be a cone-beam circular rotation system endowed with the main coordinate systems $Oxyz$ and with a flat detector with rectangular cells. Consider a parallelepipedal voxel discretization of the object volume with all boundary faces parallel to the coordinate planes. Let the common projection plane π_{prj} be the coordinated plane Oxz or Oyz, depending on the rules in Proposition 2. Then, the computation of the horizontal lengths of overlaps between the projections onto π_{prj} of voxels and detector cells can be carried out in a vectorized fashion, yielding a sparse matrix $\mathbf{W}_H$ that collects all the horizontal edges of the overlap areas.*

Remark 3.14. We just point out that the previous Proposition holds true under the hypothesis of Proposition 2, on which rely the entire construction in this Subsection. However, other rules can be considered, in principle, to select the common projection plane, but with the following restrictions:

(i) the common projection plane shall be parallel to one of the coordinated planes Oxz or Oyz, and

(ii) $O' - S \nparallel \pi_{\text{prj}}$, *i.e.*, there are not rays parallel to π_{prj}, so that no projection is lost.

For very small angles between $O' - S$ and π_{prj}, *i.e.*, for $\boldsymbol{s}(\theta) \approx \boldsymbol{w}^{\perp}_{\pi_{\text{prj}}}$ (see (3.7)), even if everything still works in theory, difficulties may arise from the computational point of view, due to very large values appearing in the vectors of the projections coordinates, which in turn could cause numerical instabilities (such as cancellation errors, or quotients with very small denominators, or bad approximation of trigonometric functions values).

B. Computing vertical projections overlaps

The same ideas described in the previous Subsection for vectorizing the computation of the horizontal lengths of overlap between voxel projections and detector cell projections, can be applied to the computation of the vertical lengths of the overlaps.

In this case, the idea is to proceed along the positive z axis, by considering the vertical projection of the first cell of each j-th detector row, $j = 1, \dots, m_v$, and then computing in a vectorized fashion the vertical lengths of overlap between all voxels projections and the projections of the cells in the same j-th detector row. So, everything is almost identical up to the computation of the "rightmost" and the "leftmost" projections of voxels and detector cells. In this case, given a vertical segment, we call "lowermost" and "uppermost" the segment endpoints having the minimal and the maximal z coordinate, respectively.

Unlike the horizontal projections case, with the vertical projections we do not have the problem of "reverse order", because the vertical increasing of the numbering of both the voxels and the detector cells is always concordant with the positive

z axis. In this case, we need to manage the analogous overlapping situations as 1–4. So, we name as $\boldsymbol{\zeta}^{\min}$, $\boldsymbol{\zeta}^{\max}$ the two vectors containing the lowermost and the uppermost z coordinates of the projections of the endpoints of all the vertical edges of the voxels midway slices, while we call $\boldsymbol{\lambda}^{\min}$ and $\boldsymbol{\lambda}^{\max}$ the similar vectors of the z coordinates of the projections of the vertical edges endpoints of all the detector cells.

Associated to these vectors, with a reasoning similar to the one described for the horizontal lengths, we introduce the indices vectors $\boldsymbol{p}^{\mathrm{V,low}}, \boldsymbol{p}^{\mathrm{V,up}}$ for the voxels projections (similar to $\boldsymbol{p}^{\mathrm{H,left}}, \boldsymbol{p}^{\mathrm{H,right}}$) and $\boldsymbol{q}^{\mathrm{V,low}}, \boldsymbol{q}^{\mathrm{V,up}}$ for the detector cells projections (similar to $\boldsymbol{q}^{\mathrm{H,left}}, \boldsymbol{q}^{\mathrm{H,right}}$).

So we get the analogous of Proposition 4, with $\boldsymbol{\zeta}^{\min}, \boldsymbol{\zeta}^{\max}$ replacing $\boldsymbol{\xi}^{\min}, \boldsymbol{\xi}^{\max}$ and $\boldsymbol{\lambda}^{\min}, \boldsymbol{\lambda}^{\max}$ replacing $\boldsymbol{\nu}^{\min}, \boldsymbol{\nu}^{\min}$, respectively.

We have then the analogous of Lemma 3.8 and, by using $\boldsymbol{z}_{\mathrm{dtc}}^{\mathrm{prj}}$ in place of $\boldsymbol{x}_{\mathrm{dtc}}^{\mathrm{prj}}$ or $\boldsymbol{y}_{\mathrm{dtc}}^{\mathrm{prj}}$, the corresponding vectors

$$\boldsymbol{p}^{\mathrm{V},j} \leftarrow \mathrm{vect}\big(\boldsymbol{p}^{\mathrm{V},j,1}, \boldsymbol{p}^{\mathrm{V},j,2}, \boldsymbol{p}^{\mathrm{V},j,3}\big) \tag{3.111}$$

and

$$\boldsymbol{h}^{\mathrm{V},j} = \mathrm{vect}\big(\boldsymbol{h}^{\mathrm{V},j,1}, \boldsymbol{h}^{\mathrm{V},j,2}, \boldsymbol{h}^{\mathrm{V},j,3}\big)\,. \tag{3.112}$$

With a similar reasoning to the one for the horizontal lengths, we define the u coordinates matrix $Z_{\mathrm{V},j}$ and the indices matrix $Z_{\mathrm{V},j}^{\mathrm{idx}}$. The latter identifies the (extended) *column index* of the projected cell of the j-th detector row whose overlap with the projection of a given voxel is nonempty. These are determined by simply looking at the projections of the rightmost column (or the leftmost, depending on which is the nearest to the z axis) of the detector cells. We call $N_{\mathrm{vxl,brd}}^{\mathrm{V},j}$ the number of voxels whose projections has intersection with the projection of the j-th cell of such a column and put in the column vector $\boldsymbol{p}_{\mathrm{brd}}^{\mathrm{V,row},j}$ the linear indices of those voxels (these are the analogue of $N_{\mathrm{vxl,btm}}^{\mathrm{H},i}$ and $\boldsymbol{p}_{\mathrm{btm}}^{\mathrm{H},i}$, which we defined for the horizontal lengths). $N_{\mathrm{vxl,brd}}^{\mathrm{V},j}$ is the number of columns of $Z_{\mathrm{V},j}$, while the number of its rows depends on which is the common projection plane, namely it is $n_x + 1$ if $\pi_{\mathrm{prj}} = Oyz$, or $n_y + 1$ if $\pi_{\mathrm{prj}} = Oxz$. By using the matrix $Z_{\mathrm{V},j}^{\mathrm{idx}}$, we define the column vector of column indices $\boldsymbol{p}_{\min}^{\mathrm{V,idx},j}$, $\boldsymbol{p}_{\max}^{\mathrm{V,idx},j}$, and the number of rows in $\mathbf{W}_{\mathrm{V}}$ in which the vertical length of overlap of a given voxel (whose linear index identifies the column) need to be replicated: these possible replications refer to adjacent cells in the j-th detector row, so their row indices in $\mathbf{W}_{\mathrm{V}}$ have stride m_v.

Lastly, for any fixed j-th row the detector, the s-th column of $Z_{\mathrm{V},j}^{\mathrm{idx}}$ provides the column indices of detector cells whose projections have a nonempty intersection with the projection of the voxels horizontally aligned to one specific voxel on the vertical layer nearest to π_{prj}. The vertical lengths of these intersection is stored in $\boldsymbol{h}^{\mathrm{V},j}\big(\boldsymbol{p}^{\mathrm{V},j}_{\mathrm{brd}}\big)$. With this information, we can then "convert" row and column indices to linear indices of voxels and detector cells. We are finally able to define the column vectors $\boldsymbol{q}^{\mathrm{dtc},j}_{rs}$ and $\boldsymbol{p}^{\mathrm{vxl},V}$ providing the right positions in $Z_{\mathrm{V},j}$ in which to store the overlaps vertical lengths:

$$\mathbf{W}_{\mathrm{V}}\big(\boldsymbol{q}^{\mathrm{dtc},j}_{rs},\, p^{\mathrm{vxl},V}_{s}\big) = \boldsymbol{h}^{\mathrm{V},j}\big(p^{\mathrm{V},j}_{\mathrm{brd},r}\big)\,. \tag{3.113}$$

for all $j \in \big\{1,\dots,m_v\big\}$, $s \in \Big\{1,\dots,N^{\mathrm{V},j}_{\mathrm{vxl,brd}}\Big\}$, $r \in \Big\{\big(p^{\mathrm{V,idx},j}_{\min}\big)_s,\dots,\big(p^{\mathrm{V,idx},j}_{\max}\big)_s\Big\}$.

Similarly to the horizontal case, we can state the following result.

Proposition 3.15. *Let $\mathcal{R}_{\mathcal{C}}\big(O,S,d_{CO'},\gamma_{\min},\gamma_{\max},\beta_{\min},\beta_{\max}\big)$ be a cone-beam circular rotation system endowed with the main coordinate systems Oxyzand with a flat detector with rectangular cells. Consider a parallelepipedal voxel discretization of the object volume with all boundary faces parallel to the coordinate planes. Let the common projection plane π_{prj} be the coordinated plane Oxz or Oyz, depending on the rules in Proposition 2. Then, the computation of the vertical lengths of overlaps between the projections onto π_{prj} of voxels and detector cells can be carried out in a vectorized fashion, yielding a sparse matrix $\mathbf{W}_V$ that collects all the vertical edges of the overlap areas.*

C. Computing the 3D CT distance-driven forward operator

Once the two matrices $\mathbf{W}_{\mathrm{H}}$ and $\mathbf{W}_{\mathrm{V}}$ have been computed for a given ℓ-th rotation angle, it is very easy to build up the corresponding section (*i.e.*, group of N_{dtc} contiguous rows) of the whole CT forward projection matrix. We recall that $\mathbf{W}$ is sized $N_\theta N_{\mathrm{dtc}} \times N_{\mathrm{vxl}}$, *i.e.*, it has as many rows as the number of rotation angles multiplied by the total number of detector cells, and as many columns as the total number N_{vxl} of voxels ($N_{\mathrm{vxl}} = N^3$ in the 3D setting).

Theorem 5. *Let $\mathcal{R}_{\mathcal{C}}\big(O,S,d_{CO'},\gamma_{\min},\gamma_{\max},\beta_{\min},\beta_{\max}\big)$ be a cone-beam circular rotation system endowed with the main coordinate systems Oxyzand with a flat detector with rectangular cells. Consider a parallelepipedal voxel discretization of the object volume with all boundary faces parallel to the coordinate planes. Let the common projection plane π_{prj} be the coordinated plane Oxz or Oyz, depending on the rules in Proposition 2. The 3D distance-driven approximation of the CT forward operator can be computed in a vectorized fashion.*

Proof. It is actually enough to simply multiply component-wise the matrices $\mathbf{W}_{\mathrm{H}}$ and $\mathbf{W}_{\mathrm{V}}$ of Propositions 3.13 and 3.15, respectively, and store the result in the right position of $\mathbf{W}$. We denote by $\overline{\boldsymbol{q}}^{\,\mathrm{dtc}}$ the rows displacement in the CT forward matrix at each rotation angle step, namely:

$$\overline{q}^{\,\mathrm{dtc}}_{\ell} = (\ell - 1)N_{\mathrm{dtc}}\,.$$

Hence, we have that, for any given $\ell \in \{1, \ldots, N_\theta\}$, the corresponding section of the CT forward projection matrix $\mathbf{W}$ is filled up by:

$$\mathbf{W}\big(\overline{q}_\ell^{\mathrm{dtc}} : (\overline{q}_\ell^{\mathrm{dtc}} + N_{\mathrm{dtc}}),\, :\big) = \mathbf{W}_{\mathrm{H}} \cdot \mathbf{W}_{\mathrm{V}} \,.$$

where the last product is meant component-wise, as already stated above. □

Theorem 5 piece everything together and it is the main result of this Section.

We conclude with the following remark, which will be useful for further considerations in the last Chapter of this thesis.

Remark 3.16. The computations of the horizontal and the vertical overlaps are completely independent on each other: hence, on a multiprocessor system they can be computed *simultaneously in parallel.*

Actually, other parts of the computations of both the horizontal and the vertical lengths of overlapping areas can be computed in parallel (and asynchronously from each other). However, this kind of parallelism is different form the previous because the number of “items” to compute simultaneously changes quite often during the main iteration, so it should be managed in a different way (by multithreading, for instance).

Chapter 4

Shearlets

In this Chapter we present a self-contained overview on the theory and applications of shearlets, a multiscale method emerged in the last decade to overcome some of the limitations of traditional multiscale methods, like wavelets. The interest in shearlets lies primarily in their employment in several numerical applications, including problems of denoising, deblurring, deconvolution and feature extraction, in a variety of fields, such as medical imaging, astronomy, seismology, meteorology, air traffic control, internet traffic and digital communications, to mention a few. In particular, the shearlet representation has been used to invert the Radon transform directly [36]. In this thesis, shearlets will be used as a regularization tool to address the ROI CT problem.

The presentation aims primarily at introducing the 2D and 3D construction, both in the continuous and the discrete setting. However, they are presented as a special case of the more general theory in $\mathbb{R}^n$, $n \geq 2$. Some background material from harmonic analysis is in Apprendix A.

4.1 From wavelets to shearlets

When *wavelets* emerged about 30 years ago, they literally "revolutionized" the image and signal processing world by immediately gaining a prominent role in the development of efficient encoding of piecewise regular signals. This was due not only to their ability to provide optimally sparse approximations of a large class of signals and to represent singularities much more efficiently than traditional Fourier methods, but also to their rich mathematical structure and the existence of fast algorithmic implementations.

However, wavelets lack of one key property, directional sensitivity, that it is crucial to deal with anisotropic features or distributed discontinuities, that frequently dominate multidimensional phenomena. To overcome this drawback, shearlets were derived by Guo, Kutyniok, Labate, Lim, and Weiss in [66, 106]. To address the directional sensitivity issues, several variations of the wavelet scheme had been proposed during the years, such as directional wavelets [3], complex wavelets [96], ridgelets [25], bandelets [111] and contourlets [47] and the most famous *curvelets* by Candès and Donoho [26], but, unlike shearlets, none of them was a truly multivariate extension of the wavelet framework. Indeed, as in classical wavelet theory, shearlets share a unified treatment of the continuum and discrete setting thanks to Multiresolution Analysis and the rich mathematical structure of affine system permits to design families of shearlets from a single or a finite set of generating functions. Fast algorithmic implementations are available and, unlike classical wavelet, shearlets are able to provide optimally sparse approximations of anisotropic features in multivariate data.

Since *shearlets* arise naturally from the general framework of wavelet analysis, a deep understanding of shearlets can only be derived through a full understanding of wavelets theory.

4.1.1 One-dimensional continuous wavelet transform

Let $\mathbb{A}_1$ be the *affine group* associated with $\mathbb{R}$, consisting of all pairs (a, t), a, $t \in \mathbb{R}, a > 0$, with group operation $(a, t)(a', t') = (aa',\, t+at')$. The *(continuous) affine systems* generated by $\psi \in L^2(\mathbb{R})$ are obtained from the action of the quasi-regular representation $\pi(a, t)$ of $\mathbb{A}_1$ on $L^2(\mathbb{R})$, that is

$$\left\{\psi_{a,t}(x) = \pi(a,t)\,\psi(x) = T_t D_a \psi(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-t}{a}\right) : (a,t) \in \mathbb{A}_1\right\},$$

where T_t is the translation operator and D_a is the dilation operator (see Appendix A). It was observed by Calderòn that, if ψ satisfies the admissibility condition

$$\int_0^\infty |\hat{\psi}(a\xi)|^2 \frac{da}{a} = 1 \qquad \text{for a.e. } \xi \in \mathbb{R}, \tag{4.1}$$

where $\hat{\psi}$ denotes the Fourier transform of ψ, then any $f \in L^2(\mathbb{R})$ can be recovered via the reproducing formula:

$$f = \int_{\mathbb{A}_1} \langle f, \psi_{a,t}\rangle\; \psi_{a,t}\; d\mu(a,t),$$

where $d\mu(a, t) = dt\frac{da}{a^2}$ is the left Haar measure of $\mathbb{A}_1$ [58]. If ψ satisfies (4.1), ψ is called a *continuous wavelet*, and the *continuous wavelet transform* of $f \in L^2(\mathbb{R})$ is defined to be the map:

$$f \;\longrightarrow\; \mathcal{W}_\psi f(a,t) = \langle f, \psi_{a,t}\rangle$$

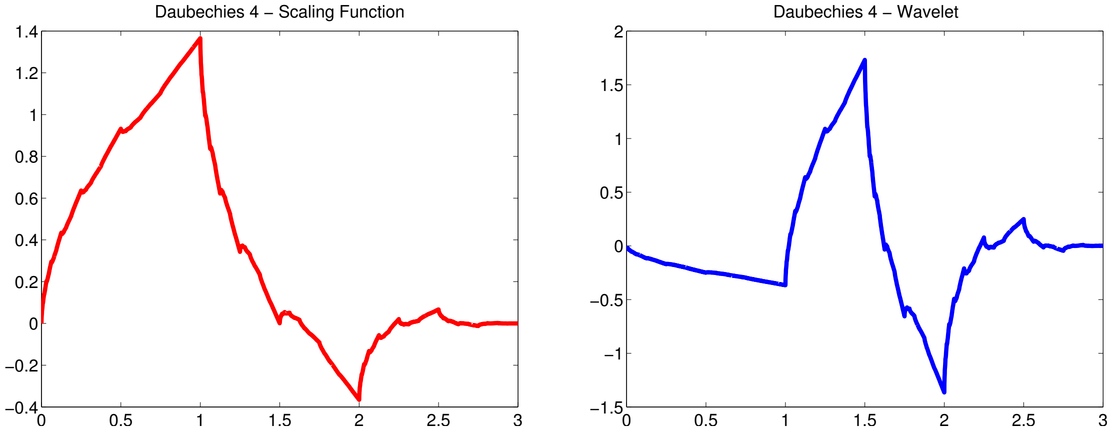


Figure 4.1: Daubechies scaling function (on the left) and Daubechies wavelet function (on the right) with 2 vanishing moments (and filter length 4).

with $(a, t) \in \mathbb{A}_1$. We refer to [158] for more details.

Discrete wavelets systems are derived by suitably "discretizing" the corresponding continuous systems, *i.e.*, by replacing the pairs $(a, t) \in \mathbb{A}_1$ with the discrete set $(2^j,\, 2^j m)$, with $j,\, m \in \mathbb{Z}$. Thus, the discrete dyadic affine system is:

$$\left\{ \psi_{j,m}(x) = T_{2^j m} D_2^j \psi(x) = D_2^j T_m \psi(x) = 2^{-j/2} \psi(2^{-j}x - m) : (j, m) \in \mathbb{Z} \right\}. \tag{4.2}$$

The function ψ is called a *wavelet* if (4.2) is an orthonormal basis or, more generally, a Parseval frame for $L^2(\mathbb{R})$ (see Appendix A). The associated *discrete wavelet transform* of $f \in L^2(\mathbb{R})$ is given by

$$f \;\longrightarrow\; \mathcal{W}_\psi f(j, m) = \langle f, \psi_{j,m} \rangle, \qquad j,\, m \in \mathbb{Z}.$$

It is possible to construct wavelets ψ which have compact support, with rapid decay in the frequency domain, or band-limited wavelets, with rapid decay in the spatial domain, or else *well localized* wavelets, in the sense that they have rapid decay in both spatial and frequency domains. Notice that the localization properties of wavelet bases play a fundamental role in their approximation properties and this is a major differences with respect to Fourier bases.

For instance, *Daubechies wavelets* (see Figure 4.1) have compact support and can be chosen to have high regularity, by prescribing an *a priori* number of continuous derivative, leading to good decay in the frequency domain. The simplest of this hierarchy of wavelets is the *Haar wavelet* ψ_{D2}, which is the only discontinuous one, defined by:

$$\psi_{\mathrm{D2}}(x) = \begin{cases} 1 & 0 \leq x < \dfrac{1}{2} \\ -1 & \dfrac{1}{2} \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

By going up the hierarchy, it is not possible to write them down in a closed form, but the interested reader can refer to [13, 79] for the construction.

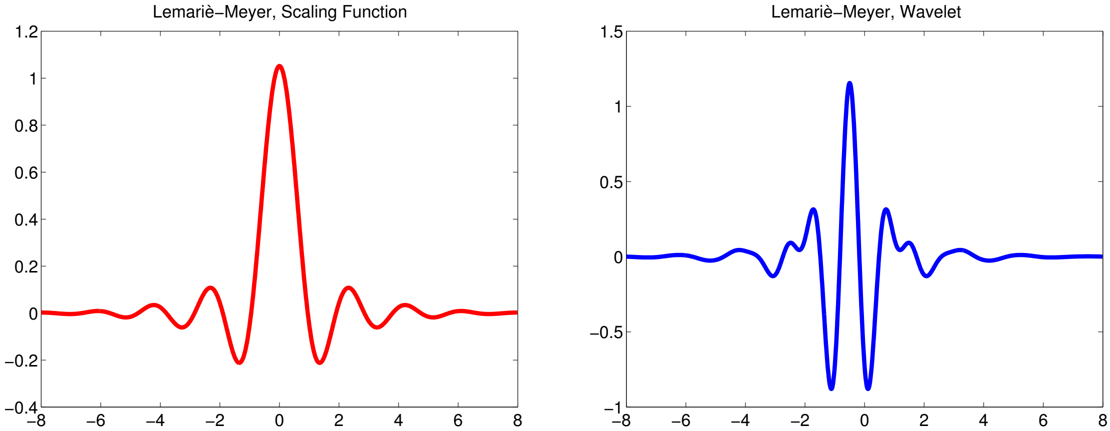


Figure 4.2: Lemariè-Meyer scaling function (on the left) and Lemariè-Meyer wavelet function (on the right).

On the contrary, the Lemariè-Meyer wavelets ψ_{LM} are band-limited and $\mathscr{C}^\infty$ in the frequency domain (see Figure 4.2), forcing rapid decay in the spatial domain. They are defined ([79]) by $\hat{\psi}_{\mathrm{LM}}(\omega) = \mathrm{e}^{i\pi\omega}\, b(\omega)$, where

$$b(\omega) = \begin{cases} \sin\left(\dfrac{\pi}{2}(3|\omega| - 1)\right) & \dfrac{1}{3} \leq |\omega| \leq \dfrac{2}{3} \\ \sin\left(\dfrac{3\pi}{4}\left(\dfrac{4}{3} - |\omega|\right)\right) & \dfrac{2}{3} \leq |\omega| \leq \dfrac{4}{3} \\ 0 & \text{otherwise} \end{cases}$$

Many other choices exist and, in general, being a wavelet is by no means very restrictive. In fact, there is a general machinery to construct orthonormal wavelet bases called *Multiresolution Analysis* (MRA), due to Stéphane Mallat [121]. We report the general definition in 1D, but it can be extended almost straightforward to higher dimension.

Definition 4.1 ([13])**.** Let V_j, $j = \dots, -2, -1, 0, 1, 2, \dots$ be a sequence of subspaces of functions in $L^2(\mathbb{R})$. The collection of spaces $\{V_j, j \in \mathbb{Z}\}$ is called a *multiresolution analysis with scaling function* ϕ if the following conditions hold:

1. Nested: $V_j \subset V_{j+1}$.
2. Density: $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$
3. Separation: $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$.
4. Scaling: the function $f(x)$ belongs to V_j if and only if the function $f(2^{-j}x)$ belongs to V_0.
5. Orthonormal basis: there exists a function $\phi \in L^2(\mathbb{R})$, called *scaling function*, such that it belongs to V_0 and the set $\{T_m\phi : m \in \mathbb{Z}\}$ is an orthonormal basis for V_0.

The spaces V_j are called *approximation spaces*. Provided that j is large enough, every $f \in L^2(\mathbb{R})$ can be approximated as closely as one likes by a function in a V_j. That is, a function $f_j \in V_j$ is decomposed as $f_j = f_{j-1} + g_{j-1} \in V_{j-1} \oplus W_{j-1}$, where W_j, $j \in \mathbb{Z}$, is a so-called *wavelet space*. These spaces are defined by considering the orthogonal complements

$$W_{j-1} = V_j \ominus V_{j-1}, \qquad j \in \mathbb{Z}.$$

Roughly, $f_{j-1} \in V_{j-1}$ contains the lower frequency component of f_j, and $g_{j-1} \in W_{j-1}$ its higher frequency component. It follows that $L^2(\mathbb{R})$ can be broken up as a direct sum of wavelet spaces.

Also, given an MRA with scaling function ϕ, there always exists a function $\psi \in L^2(\mathbb{R})$ such that $\{\psi_{j,m} : j, m \in \mathbb{Z}\}$ is an orthonormal basis for $L^2(\mathbb{R})$. In fact, the MRA approach allows to introduce an alternative orthonormal basis of the form

$$\{\phi_m = T_m\phi = \phi(\cdot - m) : m \in \mathbb{Z}\} \cup \{\psi_{j,m} : j \geq 0, m \in \mathbb{Z}\}$$

that involves both the wavelet and the scaling function. In this case, the translates of the scaling function take care of the low frequency region, namely the subspace $V_0 \subset L^2(\mathbb{R})$, and the wavelet terms account for the high frequency region, *i.e.*, the complementary space $L^2(\mathbb{R}) \ominus V_0$. For additional information about the theory of MRA see, *e.g.*, [121] and [31].

4.1.2 Higher-dimensional continuous wavelet transform

The extension of wavelet theory to higher dimensions requires to extend the theory of affine systems to higher dimensions. The natural way to do this is by replacing

$\mathbb{A}_1$ with the *full affine group of motions* on $\mathbb{R}^n$, $\mathbb{A}_n$, $n \geq 2$, consisting of the pairs $(M,t) \in GL_n(\mathbb{R}) \ltimes \mathbb{R}^n$ with group operation $(M,t)\cdot(M',t') = (MM', t+Mt')$. Similarly to the 1D case, the affine systems generated by $\psi \in L^2(\mathbb{R}^n)$ are given by

$$\left\{\psi_{M,t}(x) = T_t D_M \psi(x) = |\det M|^{-\frac{1}{2}}\psi(M^{-1}(x-t)) : (M,t) \in \mathbb{A}_n\right\}, \tag{4.3}$$

where the *dilation operator* D_M is defined by

$$D_M\psi(x) = |\det M|^{-\frac{1}{2}}\ \psi(M^{-1}x), \tag{4.4}$$

and the *translation operator* is given by

$$T_t\psi(x) = \psi(x-t). \tag{4.5}$$

Observe that, as in the 1D case, the mathematical structure of the affine systems can be expressed by means of the action of the unitary representation $\pi_{M,t}$ (see Definition A.6), defined by:

$$\begin{array}{rccl} \pi_{M,t}: & \mathbb{A}_n & \longrightarrow & \mathcal{U}(L^2(\mathbb{R}^n)) \\ & (M,t) & \longrightarrow & \pi(M,t) = T_t D_M. \end{array}$$

Thus, the elements of a wavelet system in (4.3) can be written as $\psi_{M,t} = \pi(M,t)\,\psi$. As far as reproducibility of functions in $L^2(\mathbb{R}^n)$ concerns, the following result holds true.

Theorem 4.2 ([101]). *Retaining the notations introduced in this Section, let $d\mu$ be a left-invariant Haar measure on $G \subset GL_n(\mathbb{R})$, and $d\lambda$ a left Haar measure on $\mathbb{A}_n$. Suppose also that $\psi \in L^2(\mathbb{R}^n)$ satisfies the admissibility condition*

$$\int_G |\hat{\psi}(M^T\xi)|^2\ |\det M|\ d\mu(M) = 1.$$

Then, any function $f \in L^2(\mathbb{R}^n)$ can be recovered via the following reproducing formula, interpreted weakly:

$$f = \int_{\mathbb{A}_n} \langle f, \psi_{M,t}\rangle\ \psi_{M,t}\ d\lambda(M,t).$$

When the conditions of Theorem 4.2 are satisfied, $\psi \in L^2(\mathbb{R}^n)$ is called a *continuous wavelet.* Given M, $t \in \mathbb{A}_n$, the associated *continuous wavelet transform* of $f \in L^2(\mathbb{R}^n)$ is defined to be the map

$$f \ \longrightarrow\ \mathcal{W}_\psi f(M,t) = \langle f, \psi_{M,t}\rangle.$$

There is a special case that is worth mentioning, namely when G is the group of *isotropic dilations* $G = \{a\mathbb{1}_n : a > 0\}$. In this case, the admissibility condition for ψ reads as

$$\int_0^\infty |\hat{\psi}(a\xi)|^2 \frac{da}{a} = 1.$$

and the *(isotropic) continuous wavelet transform* is the map of $f \in L^2(\mathbb{R}^n)$ into

$$\mathcal{W}_\psi f(a,t) = a^{-n/2} \int_{\mathbb{R}^n} f(x)\overline{\psi(a^{-1}(x-t))}dx, \qquad a > 0, t \in \mathbb{R}^n. \tag{4.6}$$

The adjective *isotropic* stems from the fact that the dilation factor a acts in the same way for each coordinate direction. Although the advantage of simplicity, the isotropic wavelet transform has a main drawback: it lacks of directional sensitivity and of some ability to detect the geometry of a function f (for more details, see [102, 100, 103]). It is reasonable to expect that, by choosing more general dilation groups G, one obtains wavelets with more interesting geometric properties, that shall be able to capture the anisotropic features which frequently dominate multidimensional phenomena. However, considering wavelets with anisotropic scaling is not enough to fix the situation. If a function f is understood as a curve, because of the lack of control on the direction of the elements, an ideal waveform system needs not only *anisotropic* elements but also a rotation parameter to align the elongated elements in the direction of the curve, along with a location parameter to locate the elements on the curve.

As we shall see in the next Section, this understanding was the real breakthrough to truly extend the wavelet approach to the multidimensional case, giving rise to *shearlets*.

4.2 Continuous shearlet systems

Before formally defining the (continuous) shearlet systems, it is convenient to introduce intuitively the main ideas of its construction. In this regard, we first restrict ourselves to the 2D case. Then, the fundamental results are presented in the greatest generality, exploiting the 2D and 3D cases as examples. These are indeed the two crucial situations, since all the others derive from the analysis of these two. Once it is known how to handle anisotropic features of different dimensions, the step from 3D to 4D can be dealt with in a similar way. The same happens for the extension to even higher dimensions.

As pointed out in the previous Section, an ideal waveform system able to detect anisotropic singularities, should consist of waveform elements ranging not only over several scales and locations but also at various *orientations*, with the ability to become increasingly elongated at finer scales. This requires a combination of a translation operator to displace these elements over the space, an appropriate scaling operator to generate elements at different scales and an orthogonal operator to change their orientations.

As can be easily imagined, for the translation operator one can use the standard operator T_t defined as in (4.5).

Next, a natural choice for a scaling operator that generates waveforms with anisotropic support is the family of dilation operators D_{A_a} , $a > 0$, based on *parabolic scaling matrices* A_a of the form

$$A_a = \begin{pmatrix} a & 0 \\ 0 & a^{1/2} \end{pmatrix},$$

where the dilation operator is defined as in (4.4). In particular, A_a produces *parabolic scaling*, that is $f(A_a x) = f\left(A_a \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right)$ leaves invariant the parabola $x_1 = x_2^2$. Notice that, rather than A_a, it can be used the more general matrices

$$\begin{pmatrix} a & 0 \\ 0 & a^{\alpha} \end{pmatrix},$$

where $\alpha \in (0, 1)$ controls the "degree of anisotropy". However, parabolic scaling is required to obtain optimally sparse approximations (see [103]) and, thus, has a dominant role in literature. For this reason, we will restrict ourself to the case $\alpha = 1/2$ when dealing with the 2D case.

Finally, as far as the orthogonal transformation concerns, the most obvious choice seems to be the rotation operator: this is indeed the core idea underneath *curvelets* by Candès and Donoho [26]. However, this is not the most "practicable" choice since, whenever the rotation angle is different from 0, $\pm\frac{\pi}{2}$, $\pm\pi$, $\pm\frac{3\pi}{2}$, rotations destroy the structure of the integer lattice $\mathbb{Z}^2$. This becomes a serious issue for the transition from the continuum to the discrete setting. An alternative choice is the shearing operator D_{S_s}, $s \in \mathbb{R}$, associated with the *shearing matrix* S_s given by

$$S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}.$$

The main advantage of shearing matrices is the parametrization of orientations by using the variable s associated with the slopes rather than the angles. In particular, provided that s is an integer, the shearing matrix leaves the integer lattice invariant.

Roughly, the operation of shearing is a translation along an axis (*e.g.*, the abscissa axis x) by an amount that increases linearly with another axis (the ordinate axis y). Thus, fixed a point (x, y), the shear transformation leaves the y coordinate unchanged, while the x coordinate is stretched in a linear way, based on the height of the point above the x axis, *i.e.*, on y. The result is a shape distortions as

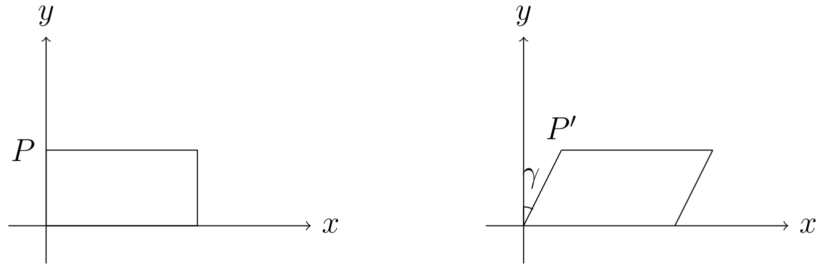


Figure 4.3: The shearing translation.

if objects were composed of layers that slide one over another. The change of coordinates has the form:

$$\begin{cases} x' = x + sy \\ y' = y \end{cases}$$

where s is the constant that measures the "degree of shearing". Clearly, if s is negative the shearing is in the opposite direction.

For example, in 2D a shear along the x direction changes a rectangle (with lower right corner at the origin) into a parallelogram, as it is shown in Figure 4.3. Here, the point $P(0, H)$ is taken into the point $P'(sH, H)$. It follows that the shearing angle γ (the angle through which the vertical edge is sheared) is given by:

$$\tan(\gamma) = \frac{sH}{H} = s.$$

So the parameter s is just the trigonometric tangent of the shearing angle.

In order to generalize these concepts to higher dimension, the first step is to identify a suitable shear matrix. Next, a generalization of the parabolic dilation matrix in $\mathbb{R}^n$, $n \geq 3$, is also needed. The approach we propose can be found in [41].

Given a n-dimensional vector space V and a k-dimensional subspace W of V, a reasonable model for the shear matrix is that the shear operation should fix the space W and translate all vectors parallel to W. Hence, for $v = w + w' \in V = W \bigoplus W'$, the shear operation S can be described as

$$S(v) = w + (w' + s(w')) = (w + s(w')) + w',$$

where s is a linear map from W' to W. Then, with respect to an appropriate basis of V, the shear operation S corresponds to a block matrix of the form

$$S = \begin{pmatrix} \mathbb{1}_k & s^T \\ \mathbb{O}_{(n-k)\times k} & \mathbb{1}_{n-k} \end{pmatrix}, \qquad s \in \mathbb{R}^{(n-k)\times k},$$

where $\mathbb{1}_n$ denote the n-dimensional identity matrix and $\mathbb{O}_n$ is n-dimensional zero matrix. Now, the crucial point is how to choose the block s. Indeed, the goal is to end up with a square integrable group representation, as in classical wavelet theory. Usually, the number of parameters has to fit together with the space dimension, otherwise the resulting group would be either too large or too small. Since we have n degrees of freedom related to translation and 1 degree of freedom related to dilation, $n-1$ degrees of freedom for the shear component would be optimal. Therefore, we take $s \in \mathbb{R}^{(n-1)\times 1}$, *i.e.*, $k = 1$:

$$S_s = \begin{pmatrix} 1 & s^T \\ \mathbf{0}_{n-1} & \mathbb{1}_{n-1} \end{pmatrix},$$

where $s \in \mathbb{R}^{n-1}$, and $\mathbf{0}_{n-1}$ is the vector with all the entries equal to zero.
As far as the dilation matrix concerns, we choose a diagonal matrix depending on the parameter $a \in \mathbb{R}^* = \mathbb{R} \setminus \{0\}$, namely:

$$A_a = \operatorname{diag}(a_1(a), \ldots, a_d(a)),$$

where $a_1(a) = a$ and $a_j(a) = a^{\alpha_j}$ with $\alpha_j \in (0,1)$, $j = 2, \ldots, n$. Since our aim is directional sensitivity, the dilation factors on the diagonal of A_a should be chosen in an anisotropic way, *i.e.*, for each $j = 2, \ldots, n$, $|a_j(a)|$ should increase less than linearly in a as $a \to \infty$. Clearly, as in the 2D case, $\alpha_j = 1/2$, $j = 2, \ldots, d$, plays a special role and, even in the 3D case, this will be our choice. Another possible choice, which still preserve the interpretation of the shearlet transform as square integrable group representation, is:

$$A_a = \begin{pmatrix} a & \mathbf{0}_{n-1}^T \\ \mathbf{0}_{n-1} & \operatorname{sgn}(a)\,|a|^{\frac{1}{n}}\,\mathbb{1}_{n-1} \end{pmatrix}.$$

The definition of *continuous shearlet systems* in $\mathbb{R}^n$, $n \geq 2$, stems from the combination of the three operators introduced, namely scaling, shearing and translation.

Definition 4.3 ([101]). For $\psi \in L^2(\mathbb{R}^n)$, $n \geq 2$, the *continuous shearlet system* $\mathcal{SH}(\psi)$ is defined by

$$\mathcal{SH}(\psi) = \left\{ \psi_{a,s,t}(x) = T_t D_{S_s} D_{A_a} \psi(x) : a > 0, s \in \mathbb{R}^{n-1}, t \in \mathbb{R}^n \right\}. \tag{4.7}$$

where

$$T_t D_{S_s} D_{A_a} \psi(x) = |\det A_a|^{-\frac{1}{2}}\ \psi(A_a^{-1} S_s^{-1}(x-t)) = |a|^{\frac{1}{2n}-1}\ \psi(A_a^{-1} S_s^{-1}(x-t)).$$

The associated *continuous shearlet transform* of $f \in L^2(\mathbb{R}^n)$ is given by

$$f \longrightarrow \mathscr{SH}_\psi f(a,s,t) = \langle f, \psi_{a,s,t} \rangle, \qquad (a,s,t) \in \mathbb{R}^+ \times \mathbb{R}^{n-1} \times \mathbb{R}^n.$$

In other words, the continuous shearlet transform projects the function f onto the functions $\psi_{a,s,t}$ at scale a, orientation s and location t. The anisotropic dilation A_a controls the shearlets scale, by applying a different dilation factor along the different directions. This ensures that the frequency support of shearlets becomes increasingly elongated at finer scales: indeed, as $a \to 0$ we obtain needlelike functions. The shear matrix S_s is non-expansive and determines the orientation of shearlets using the shear parameter s to detect different directions by slope. Finally, the location parameter t ensures position sensitivity.

4.2.1 The full shearlet group

One important structural property of the shearlet system (4.7) is that its elements can be generated by using a representation of a $(2n)$-parameter group, the so-called *full shearlet group.* Similarly to wavelets, the theory of continuous shearlet systems can be developed within the theory of unitary representations of the affine group $\mathbb{A}_n$ and its generalizations.

Proposition 4.4 ([41, 42]). *The set $\mathbb{R}^* \times \mathbb{R}^{n-1} \times \mathbb{R}^n$ endowed with the operation*

$$(a,\, s,\, t) \circ (a',\, s',\, t') = (aa',\, s + |a|^{1-\frac{1}{n}} s',\, t + S_s A_a t')$$

is a locally compact group, in the sense specified by Definition A.4. The left Haar measure on this group is given by

$$d\mu(a,\, s,\, t) = \frac{1}{|a|^{n+1}}\, da\, ds\, dt.$$

The previous proposition allows us to give the following definition.

Definition 4.5 ([41, 42]). The *full shearlet group* $\mathbb{S}$ is defined to be the set $\mathbb{R}^* \times \mathbb{R}^{n-1} \times \mathbb{R}^n$ along with the multiplication law given by

$$(a,\, s,\, t) \circ (a',\, s',\, t') = (aa',\, s + |a|^{1-\frac{1}{n}} s',\, t + S_s A_a t').$$

The following proposition states how the element of the shearlet system (4.7) can be generated by using a unitary representation.

Proposition 4.6 ([41, 42]). *Let $\pi : \mathbb{S} \to \mathcal{U}(L^2(\mathbb{R}^n))$ be a map into the group $\mathcal{U}(L^2(\mathbb{R}^n))$ of unitary operators on $L^2(\mathbb{R}^n)$ defined by*

$$\pi(a,\, s,\, t)\psi(x) = \psi_{a,s,t}(x) = |\det A_a|^{-\frac{1}{2}}\, \psi(A_a^{-1} S_s^{-1}(x-t)) = |a|^{\frac{1}{2n}-1}\, \psi(A_a^{-1} S_s^{-1}(x-t)).$$

Then π is a unitary representation of $\mathbb{S}$ on $L^2(\mathbb{R}^n)$.

For all $f \in L^2(\mathbb{R}^n)$, the unitary representation π can be related to the continuous shearlet transform likewise to the wavelets case, that is:

$$\mathscr{SH}_\psi f(a,s,t) = \langle f, \psi_{a,s,t}\rangle = \langle f, \pi(a,s,t)\psi\rangle.$$

Notice that, the continuous shearlet transform is nothing else but the continuous wavelet transform associated with a special subgroup $\mathbb{A}_n^S$ of the affine group $\mathbb{A}_n$. For a fixed $\alpha = (\alpha_1, \ldots, \alpha_{n-1})$, with $\alpha_j \in (0,1)$ and $j = 1, \ldots, n-1$, $\mathbb{A}_n^S$ consists of the elements $(M_{a,s}, t)$, where $t \in \mathbb{R}^n$ and

$$M_{a,s} = \begin{pmatrix} a & a^{\alpha_1}s_1 & \ldots & a^{\alpha_{n-1}}s_{n-1} \\ 0 & a^{\alpha_1} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & a^{\alpha_{n-1}} \end{pmatrix}$$

with $a > 0$ and $s = (s_1, \ldots, s_{n-1}) \in \mathbb{R}^{n-1}$. Clearly, each matrix $M_{a,s}$ is the product of the shear matrix S_s and the dilation matrix A_a:

$$S_s = \begin{pmatrix} 1 & s_1 & \ldots & s_{n-1} \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix} \qquad A_a = \begin{pmatrix} a & 0 & \ldots & 0 \\ 0 & a^{\alpha_1} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & a^{\alpha_{n-1}} \end{pmatrix}.$$

Obviously, given a pair $(M_{a,s}, t) \in \mathbb{A}_n^S$, the continuous shearlet transform of $f \in L^2(\mathbb{R}^n)$ is the map:

$$f \longrightarrow \langle f,\, \psi_{M_{a,s},t}\rangle,$$

and the analyzing elements $\psi_{M_{a,s},t}$ are the affine functions defined by:

$$\psi_{M_{a,s},t}(x) = |\det M_{a,s}|^{-\frac{1}{2}}\, \psi(M_{a,s}^{-1}(x-t)).$$

Now, recall that, in general, a nontrivial function $\psi \in L^2(\mathbb{R}^n)$ is called *admissible* if

$$\int_{\mathbb{S}} |\langle \psi, \pi(a,\, s,\, t)\psi\rangle|^2 \, d\mu(a,\, s,\, t) < \infty.$$

Further, if π is irreducible and there exits at least one admissible function $\psi \in L^2(\mathbb{R}^n)$, then π is called *square integrable.* The admissibility condition is really important, since it is automatically associated with a reconstruction formula (see Appendix A.2). The following result shows that the unitary representation π defined above is, indeed, square integrable.

Theorem 4.7 ([41, 42])**.** *A function $\psi \in L^2(\mathbb{R}^n)$ is admissible if and only if it fulfills the admissibility condition*

$$C_\psi = \int_{\mathbb{R}^n} \frac{|\hat{\psi}(\omega)|^2}{|\omega_1|^n}\, d\omega < \infty. \tag{4.8}$$

If ψ is admissible, then, for any $f \in L^2(\mathbb{R}^n)$, the following equality holds true:

$$\int_{\mathbb{S}} |\langle f, \psi_{a,s,t}\rangle|^2\, d\mu(a,\, s,\, t) = C_\psi\, \|f\|^2_{L^2(\mathbb{R}^n)}.$$

In particular, the unitary representation π is irreducible and hence square integrable.

Definition 4.8 ([41, 42])**.** A function $\psi \in L^2(\mathbb{R}^n)$ is called a *continuous shearlet* if it satisfies the admissibility condition (4.8).

Notice that examples of admissible shearlets, possibly well localized, are easy to construct. Essentially, any function ψ band-limited away from the origin is an admissible shearlet, as the following examples show.

Example 4.9 (2D Classical Shearlet, [66, 106])*.* Let $\psi \in L^2(\mathbb{R}^2)$ be defined by

$$\hat{\psi}(\xi) = \hat{\psi}(\xi_1, \xi_2) = \hat{\psi}_1(\xi_1)\hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right),$$

where $\psi_1 \in L^2(\mathbb{R}^2)$ is a discrete wavelet in the sense that it satisfies the discrete Calderòn condition, given by

$$\sum_{j\in\mathbb{Z}} |\hat{\psi}_1(2^{-j}\xi)|^2 = 1 \qquad \text{for a.e. } \ \xi \in \mathbb{R}, \tag{4.9}$$

with $\hat{\psi}_1 \in C^\infty(\mathbb{R})$ and $\operatorname{supp}(\hat{\psi}_1) \subseteq [-\frac{1}{2}, -\frac{1}{16}] \cup [\frac{1}{16}, \frac{1}{2}]$, and $\psi_2 \in L^2(\mathbb{R})$ is a bump function in the sense that

$$\sum_{k=-1}^{1} |\hat{\psi}_2(\xi + k)|^2 = 1 \qquad \text{for a.e. } \ \xi \in [-1, 1], \tag{4.10}$$

satisfying $\hat{\psi}_2 \in C^\infty(\mathbb{R})$ and $\operatorname{supp}(\hat{\psi}_2) \subseteq [-1, 1]$. Such a function ψ, which is wavelet-like along one axis and bump-like along another one, is called a *classical shearlet.*

As illustrated in Figure 4.4, each element $\psi_{a,s,t}$ of classical shearlet system has frequency support on a pair of trapezoids, symmetric with respect to the origin,

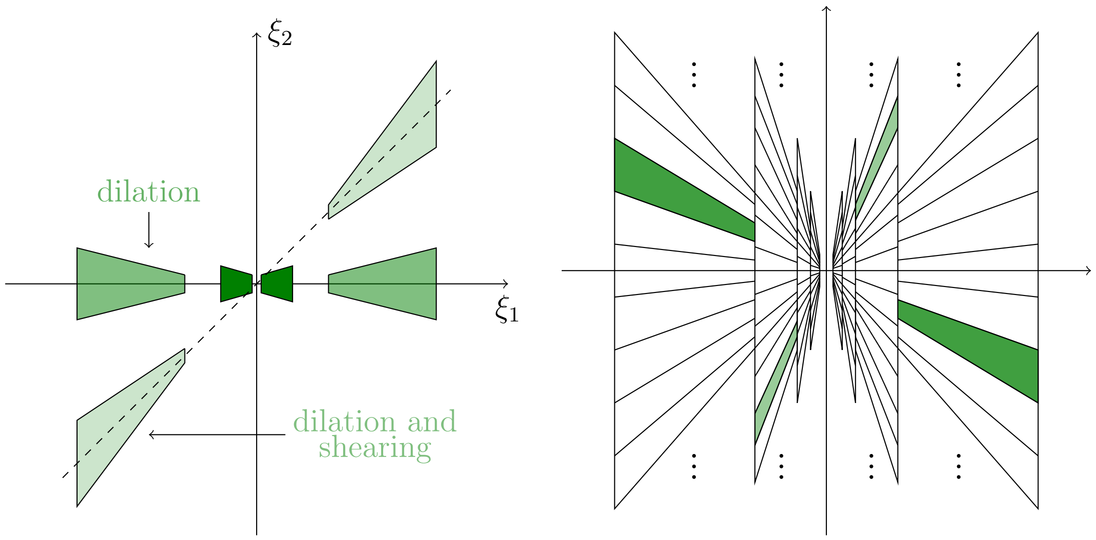


Figure 4.4: 2D Classical shearlets. Fourier domain support of several elements of the shearlet system, for different values of a (dilation) and s (shearing).

oriented along a line of slope s. The support becomes increasingly thin as $a \to 0$, given that

$$\operatorname{supp}(\hat{\psi}_{a,s,t}) \subset \left\{(\xi_1, \xi_2) : \xi_1 \in \left[-\frac{1}{2a}, -\frac{1}{16a}\right] \cup \left[\frac{1}{16a}, \frac{1}{2a}\right], \left|\frac{\xi_2}{\xi_1} + s\right| \leq \sqrt{a}\right\}.$$

The specific choices for the supports of the functions ψ_1, ψ_2 have no deeper meaning: all that is needed (for the detection of anisotropic structures) is that $\hat{\psi}_1$ is supported away from zero (*i.e.*, ψ_1 is a wavelet) and $\hat{\psi}_2$ is supported around zero [101].

Notice that there exist several choices of ψ_1 and ψ_2 satisfying conditions (4.9) and (4.10). One possible choice is to set ψ_1 to be a Lemariè–Meyer wavelet and ψ_2 to be a spline.

Finally, it is not difficult to prove that the shearlet transform associated with a classical shearlet $\psi \in L^2(\mathbb{R}^2)$ is an isometry.

Example 4.10 (3D and nD Classical shearlet, [41]). In 3D, and more generally in dimension n with $n \geq 3$, an example of a continuous shearlet can be exhibited using the same idea of the 2D case. Let ψ_1 be an admissible wavelet (*i.e.*, satisfying the Calderòn condition) with $\hat{\psi}_1 \in C^\infty(\mathbb{R})$ and $\operatorname{supp}(\hat{\psi}_1) \subseteq [-2, -1] \cup [1, 2]$, and let ψ_2 be a bump function such that $\hat{\psi}_2 \in C^\infty(\mathbb{R}^{n-1})$ and $\operatorname{supp}(\hat{\psi}_2) \subseteq [-1, 1]^{n-1}$. Then, the function $\psi \in L^2(\mathbb{R}^n)$ defined by

$$\hat{\psi}(\omega) = \hat{\psi}(\omega_1, \tilde{\omega}) = \hat{\psi}_1(\omega_1)\hat{\psi}_2\left(\frac{1}{\omega_1}\tilde{\omega}\right)$$

is a continuous shearlet.

As far as the 3D case concerns, the shearlet generator $\psi \in L^2(\mathbb{R}^3)$ reads as

$$\hat{\psi}(\xi) = \hat{\psi}_1(\xi_1)\,\hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right)\,\hat{\psi}_2\left(\frac{\xi_3}{\xi_1}\right),$$

where ψ_1 and ψ_2 satisfy the assumptions above with $n = 3$. Notice that, in the frequency domain, the band-limited function $\psi \in L^2(\mathbb{R}^3)$ is almost a tensor product of one wavelet with two bump functions. This implies that the support in frequency domain has a needle-like shape with the wavelet acting in radial direction ensuring high directional selectivity, as can be observed in Figure 4.7b.

4.2.2 Cone-adapted and pyramid-based continuous shearlet systems

Although the continuous shearlet systems defined above exhibit an elegant group structure, there exists a certain directional bias related to the shear parameter. By observing Figure 4.4, it is easy to see that the distribution of directions becomes infinitely dense as s grows. In general, if we consider a function f mostly concentrated along the ordinate axis in the frequency domain, it is clear that its energy is more and more concentrated in the shearlet components $\mathcal{SH}_\psi f(a, s, t)$ as $s \to \infty$. This can be a serious limitation for some practical applications.

In the 2D case, one way to address this problem is to partition the frequency domain into four cones $\mathscr{C}_i$, $i = 1, \ldots, 4$, and the square $\mathscr{R} = \{(\xi_1, \xi_2) : |\xi_1|, |\xi_2| \leq 1\}$ centered around the origin to separate the low-frequency region. This yields a partition of the frequency plane as illustrated in Figure 4.5, that leads to the definition of a variant of the continuous shearlet systems, usually referred to as *cone-adapted continuous shearlets.*

A similar construction can be convenient also in 3D, by following essentially the same idea as the 2D construction: the 3D frequency domain is partitioned into three pairs of pyramidal regions $\mathscr{P}_i$, $i = 1, 2, 3$, and a centered cube $\mathscr{C}$, as illustrated in Figure 4.6. This leads to the definition of *pyramid-based continuous shearlets.*

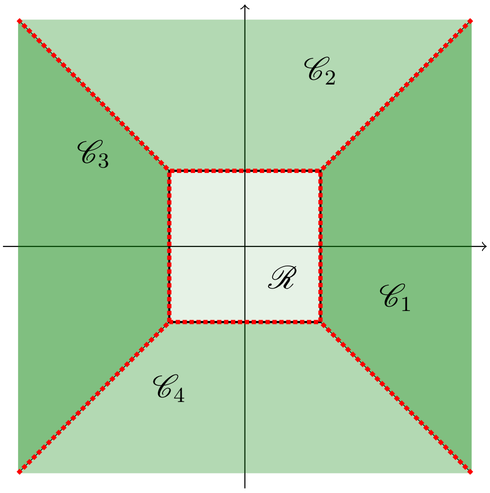


Figure 4.5: 2D partition of the frequency domain into four cones $\mathscr{C}_i$, $i = 1, \ldots, 4$, and a square $\mathscr{R}$ centered around the origin.

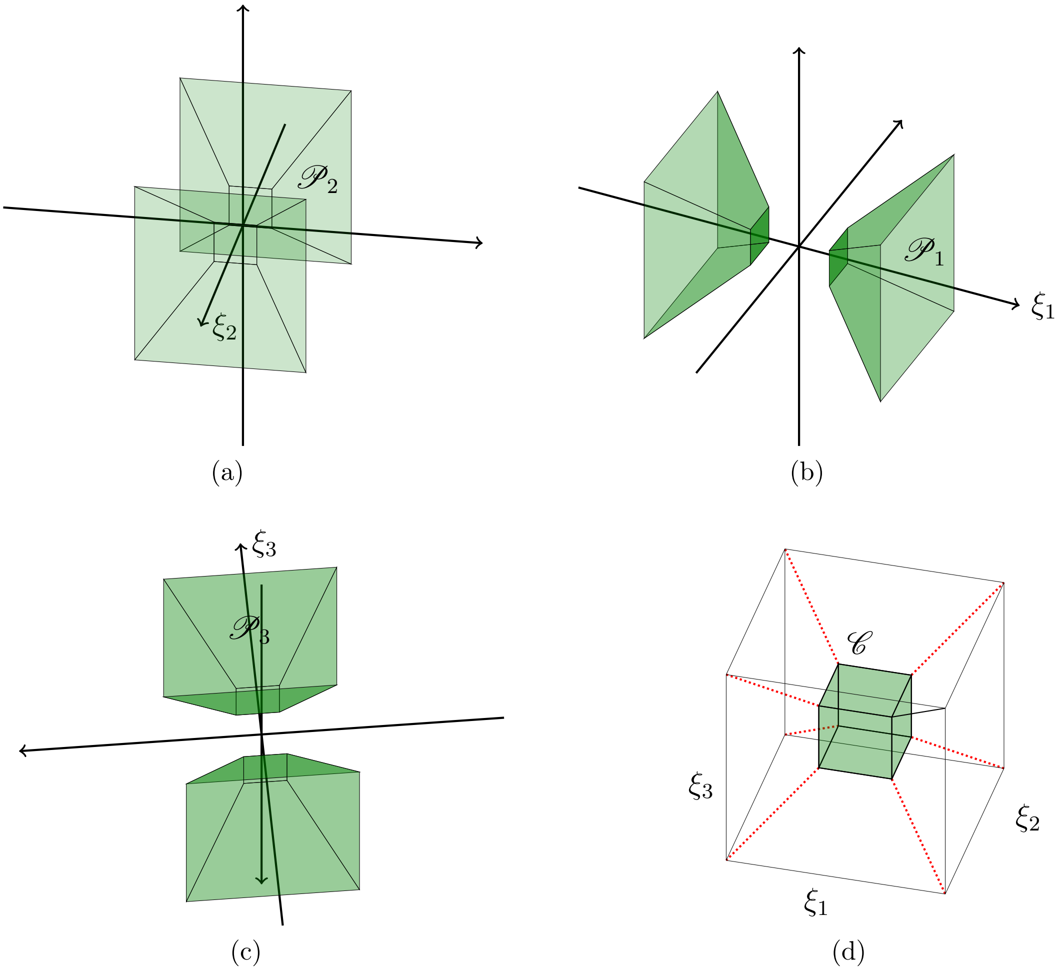


Figure 4.6: 3D partition of the frequency domain into three couple of pyramids $\mathscr{P}_i$, $i = 1, 2, 3$ and a centered cube surrounded by the three pairs of pyramids.

Definition 4.11 ([101])**.** For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, the *2D cone-adapted continuous shearlet system* $\mathcal{SH}(\phi, \psi, \tilde{\psi})$ is defined by

$$\mathcal{SH}(\phi, \psi, \tilde{\psi}) = \Phi(\phi) \cup \Psi(\psi) \cup \tilde{\Psi}(\tilde{\psi}),$$

where

$$\begin{aligned}
&\Phi(\phi) = \{\phi_t = \phi(\cdot - t) : t \in \mathbb{R}^2\},\\
&\Psi(\psi) = \{\psi_{a,s,t} = a^{-\frac{3}{4}}\psi(A_a^{-1}S_s^{-1}(\cdot - t)) : a \in (0,1], |s| \leq 1 + \sqrt{a}, t \in \mathbb{R}^2\},\\
&\tilde{\Psi}(\tilde{\psi}) = \{\tilde{\psi}_{a,s,t} = a^{-\frac{3}{4}}\psi(\tilde{A}_a^{-1}S_s^{-T}(\cdot - t)) : a \in (0,1], |s| \leq 1 + \sqrt{a}, t \in \mathbb{R}^2\},
\end{aligned}$$

and $\tilde{A}_a$ =diag($a^{1/2}$, a).

Notice that, within each cone, the shearing variable s is only allowed to vary over a finite range and consequently it is possible to detect only a certain subset of all possible directions. Also, to ensure that the system $\Phi(\phi)$ is associated with the low frequency region $\mathscr{R}$, the function ϕ will be chosen to have compact frequency support near the origin.

Example 4.12 (2D cone-adapted classical shearlet, [101]). If ψ is a 2D classical shearlet, the system $\Psi(\psi)$ is associated with the horizontal cones

$$\mathscr{C}_1 \cup \mathscr{C}_3 = \Big\{(\xi_1, \xi_2) : \Big|\frac{\xi_2}{\xi_1}\Big| \leq 1, |\xi_1| > 1\Big\}.$$

Analogously, the shearlet $\tilde{\psi}$ can be chosen with the roles of ξ_1 and ξ_2 reversed, *i.e.*, $\tilde{\psi}(\xi_1, \xi_2) = \psi(\xi_2, \xi_1)$. Then the system $\tilde{\Psi}(\tilde{\psi})$ is associated with the vertical cones

$$\mathscr{C}_2 \cup \mathscr{C}_4 = \Big\{(\xi_1, \xi_2) : \Big|\frac{\xi_2}{\xi_1}\Big| > 1, |\xi_2| > 1\Big\}.$$

In practice, the previous example suggest to imagine the function f splitted into $f = P^{(h)}f + P^{(v)}f$, where $P^{(h)}$ is the frequency projection onto the cone with slope $s \leq 1$, and $P^{(v)}$ is the frequency projection onto the cone with slope $\frac{1}{s} \leq 1$. Thus, fixed a shearlet $\psi^{(h)}$, we can analyze only $P^{(h)}f$ while $P^{(v)}f$ is analyzed by defining $\psi^{(v)}(\xi_1, \xi_2) = \psi^{(h)}(\xi_2, \xi_1)$.

Clearly, it is possible to extend the definition of 2D cone-adapted classical shearlet to the 3D case, as a special case of 3D classical shearlets from example 4.10. Since the formal definition of *3D pyramid-based continuous shearlet systems* is generally given in this special case, we start by introducing the 3D pyramid-based classical shearlet.

Example 4.13 (3D pyramid-based classical shearlet, [67]). Analogously to the 2D case, the 3D frequency domain is partitioned into the three pairs of pyramidal regions $\mathscr{P}_1$, $\mathscr{P}_2$, and $\mathscr{P}_3$ given by

$$\begin{aligned}
\mathscr{P}_1 &= \left\{(\xi_1,\xi_2,\xi_3)\in\mathbb{R}^3 : |\xi_1|\geq 1, \left|\frac{\xi_2}{\xi_1}\right|\leq 1 \text{ and } \left|\frac{\xi_3}{\xi_1}\right|\leq 1\right\},\\
\mathscr{P}_2 &= \left\{(\xi_1,\xi_2,\xi_3)\in\mathbb{R}^3 : |\xi_2|\geq 1, \left|\frac{\xi_1}{\xi_2}\right|\leq 1 \text{ and } \left|\frac{\xi_3}{\xi_2}\right|\leq 1\right\},\\
\mathscr{P}_3 &= \left\{(\xi_1,\xi_2,\xi_3)\in\mathbb{R}^3 : |\xi_3|\geq 1, \left|\frac{\xi_1}{\xi_3}\right|\leq 1 \text{ and } \left|\frac{\xi_2}{\xi_3}\right| > 1\right\},
\end{aligned}$$

and the centered cube

$$\mathscr{C} = \left\{(\xi_1,\xi_2,\xi_3)\in\mathbb{R}^3 : \|(\xi_1,\xi_2,\xi_3)\|_\infty < 1\right\}.$$

Given $\xi = (\xi_1,\xi_2,\xi_3)\in\mathbb{R}^3$ with $\xi_1\neq 0$, each $\psi^{(k)}$, $k = 1,2,3$, is defined by

$$\begin{aligned}
\hat{\psi}^{(1)}(\xi) &= \hat{\psi}^{(1)}(\xi_1,\xi_2,\xi_3) = \hat{\psi}_1(\xi_1)\,\hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right)\,\hat{\psi}_2\left(\frac{\xi_3}{\xi_1}\right),\\
\hat{\psi}^{(2)}(\xi) &= \hat{\psi}^{(2)}(\xi_1,\xi_2,\xi_3) = \hat{\psi}_1(\xi_2)\,\hat{\psi}_2\left(\frac{\xi_1}{\xi_2}\right)\,\hat{\psi}_2\left(\frac{\xi_3}{\xi_2}\right),\\
\hat{\psi}^{(3)}(\xi) &= \hat{\psi}^{(3)}(\xi_1,\xi_2,\xi_3) = \hat{\psi}_1(\xi_3)\,\hat{\psi}_2\left(\frac{\xi_1}{\xi_3}\right)\,\hat{\psi}_2\left(\frac{\xi_2}{\xi_3}\right),
\end{aligned}$$

where ψ_1, ψ_2 satisfy the same assumptions as in example 4.10.

Definition 4.14 ([103, 67])**.** For $d = 1,2,3$, let $\psi^{(d)}\in L^2(\mathbb{R}^3)$ and $\mathscr{P}_d$ be defined as in example 4.13. The *3D pyramid-based continuous shearlet systems* $\mathcal{SH}(\phi,\psi^{(1)},\psi^{(2)},\psi^{(3)})$ for $L^2(\mathscr{P}_d)^\vee$, generated by $\phi,\psi^{(1)},\psi^{(2)},\psi^{(3)}\in L^2(\mathbb{R}^3)$, is defined by

$$\mathcal{SH}(\phi,\psi^{(1)},\psi^{(2)},\psi^{(3)}) = \Phi(\phi)\,\cup\,\Psi^{(1)}(\psi^{(1)})\,\cup\,\Psi^{(2)}(\psi^{(2)})\,\cup\,\Psi^{(3)}(\psi^{(3)}),$$

where

$$\begin{aligned}
\Phi(\phi) &= \{\phi_t = \phi(\cdot - t) : t\in\mathbb{R}^3\},\\
\Psi^{(d)}(\psi^{(d)}) &= \left\{\psi^{(d)}_{a,s_1,s_2,t} : 0\leq a\leq\frac{1}{4},\ -\frac{3}{2}\leq s_1\leq\frac{3}{2},\ -\frac{3}{2}\leq s_2\leq\frac{3}{2},\ t\in\mathbb{R}^3\right\},
\end{aligned}$$

with

$$\psi^{(d)}_{a,s_1,s_2,t}(x) = |\det M^{(d)}_{a,s_1,s_2}|^{-\frac{1}{2}}\,\psi^{(d)}((M^{(d)}_{a,s_1,s_2})^{-1}(x-t)),$$

and

$$M^{(1)}_{a,s_1,s_2} = \begin{pmatrix} a & a^{1/2}s_1 & a^{1/2}s_2 \\ 0 & a^{1/2} & 0 \\ 0 & 0 & a^{1/2} \end{pmatrix}, \qquad M^{(2)}_{a,s_1,s_2} = \begin{pmatrix} a^{1/2} & 0 & 0 \\ a^{1/2}s_1 & a & a^{1/2}s_2 \\ 0 & 0 & a^{1/2} \end{pmatrix},$$

$$M^{(3)}_{a,s_1,s_2} = \begin{pmatrix} a^{1/2} & 0 & 0 \\ 0 & a^{1/2} & 0 \\ a^{1/2}s_1 & a^{1/2}s_2 & a \end{pmatrix}.$$

Notice that, similar to the 2D case, in each pyramidal region the shearing variables s_1, s_2 are only allowed to vary over a compact set. This approach is, indeed, what gives an almost uniform treatment of different directions, understood as good approximation to rotation.

Similar to the continuous shearlet systems, an associated transform can be defined for cone-adapted and pyramid-based continuous shearlet systems.

Definition 4.15 ([101, 67])**.** Set

$$\begin{aligned} \mathbb{S}_{\text{cone}} &= \{(a,s,t) : a \in (0,1],\ |s| \le 1+\sqrt{a},\ t \in \mathbb{R}^2\}, \\ \mathbb{S}_{\text{pyr}} &= \{(a,s_1,s_2,t) : a > 0,\ s_1, s_2 \in \mathbb{R},\ t \in \mathbb{R}^3\}. \end{aligned}$$

For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, the *2D cone-adapted continuous shearlet Transform* of $f{\in}L^2(\mathbb{R}^2)$ is the map

$$f \to \mathscr{SH}_{\phi,\psi,\tilde{\psi}} f(t', (a,s,t), (\tilde{a},\tilde{s},\tilde{t})) = (\langle f, \phi_t\rangle, \langle f, \psi_{a,s,t}\rangle, \langle f, \tilde{\psi}_{\tilde{a},\tilde{s},\tilde{t}}\rangle)$$

where $(t', (a,s,t), (\tilde{a},\tilde{s},\tilde{t})) \in \mathbb{R}^2 \times \mathbb{S}^2_{\text{cone}}$. In particular, for the 2D cone-adapted classical shearlet, the *2D cone-adapted continuous shearlet Transform* is the map:

$$f \longrightarrow \mathscr{SH}_{\psi} f(a,s,t) = \begin{cases} \langle f, \psi^{(h)}_{a,s,t}\rangle & \text{if } |s| \le 1 \\ \langle f, \psi^{(v)}_{a,\frac{1}{s},t}\rangle & \text{if } |s| > 1 \end{cases}$$

Similarly, for $\psi \in L^2(\mathbb{R}^3)$, the *3D pyramid-based continuous shearlet transform* of $f \in L^2(\mathbb{R}^3)$ is the map

$$f \longrightarrow \mathscr{SH}_{\psi} f(a,s_1,s_2,t) = \begin{cases} \langle f, \psi^{(1)}_{a,s_1,s_2,t}\rangle & \text{if } |s_1|, |s_2| \le 1 \\ \langle f, \psi^{(2)}_{a,\frac{1}{s_1},\frac{s_2}{s_1},t}\rangle & \text{if } |s_1| > 1, |s_2| \le |s_1| \\ \langle f, \psi^{(3)}_{a,\frac{s_1}{s_2},\frac{1}{s_2},t}\rangle & \text{if } |s_2| > 1, |s_2| > |s_1| \end{cases}$$

where $(a,s_1,s_2,t) \in \mathbb{S}_{\text{pyr}}$.

Notice that, depending on the values of the shearing variables, the 2D (respectively, 3D) continuous shearlet transform corresponds to one specific cone-adapted (respectively, pyramid-based) shearlet system.

It can be shown that a result similar to Theorem 4.7 holds true. As a consequence, under suitable conditions on the scaling function and the mother shearlet, the map $\mathscr{SH}_\psi$ is an isometry.

Finally, observe that the cone-adapted and pyramid-based continuous shearlet transform can be used to provide a precise characterization of edge discontinuities of a function f by analyzing the decay of its associated coefficients at fine scales. Roughly, edges are associated to the largest shearlet coefficients, since the continuous shearlet transform decays rapidly as $a \to 0$, namely satisfies

$$\mathscr{SH}_\psi f(a, s, t) = O(a^r), \qquad \text{for every } \ r \geq 0,$$

unless t is at the singularity and s describes the direction that is perpendicular to the discontinuity curve. This property is extremely useful in all those applications in which an analysis or detection of edge discontinuities is required. By way of example, the algorithm proposed in [36] to invert the Radon transform directly uses a shearlet-based edge detection.

A more detailed analysis of the cone-adapted and pyramid-based continuous shearlet transform can be found in [65, 67].

4.3 Discrete shearlet systems

Starting from continuous shearlet systems as defined in (4.7), several discrete versions of shearlet systems can be constructed by an appropriate sampling of the continuous parameter set $\mathbb{S}$, $\mathbb{S}_{\text{cone}}$ or $\mathbb{S}_{\text{pyr}}$. Usually, the aim is to derive discrete shearlet systems which preferably form an orthonormal basis or a tight frame (see Appendix A.1) for $L^2(\mathbb{R}^n)$, since this leads to orthonormal matrices that are fundamental, *e.g.*, to simplify the numerical complexity in many applications.

4.3.1 Regular discrete shearlet systems

Discrete shearlet systems are formally defined by sampling continuous shearlet systems on a discrete subset of the full shearlet group $\mathbb{S}$.

Definition 4.16 ([101]). Let $\psi \in L^2(\mathbb{R}^n)$, $n \geq 2$, and $\Lambda \subseteq \mathbb{S}$. An *irregular discrete shearlet system*, associated with ψ and Λ and denoted by $\mathcal{SH}(\psi, \Lambda)$, is defined by

$$\mathcal{SH}(\psi, \Lambda) = \Big\{ \psi_{a,s,t} = a^{\frac{1}{2n}-1} \psi(A_a^{-1} S_s^{-1}(\cdot - t)) : (a, s, t) \in \Lambda \Big\}.$$

A *(regular) discrete shearlet system*, associated with ψ and denoted by $\mathcal{SH}(\psi)$, is defined by

$$\mathcal{SH}(\psi) = \Big\{\psi_{j,k,m} = 2^{j\left(\frac{1}{2n}-1\right)}\psi(S_k A_{2^j} \cdot -m) : j \in \mathbb{Z}, k \in \mathbb{Z}^{n-1}, m \in \mathbb{Z}^n\Big\}.$$

Observe that the regular versions of discrete shearlet systems are derived from the irregular systems by choosing

$$\Lambda = \Big\{(2^{-j}, -k2^{-j/2}, S_{-k2^{-j/2}} A_{2^{-j}} m) : j \in \mathbb{Z}, k \in \mathbb{Z}^{n-1}, m \in \mathbb{Z}^n\Big\}.$$

Indeed, keeping the greatest generality, we can choose an arbitrary set of scales $\{a_j\}_{j\in\mathbb{Z}} \subset \mathbb{R}^+$. Next, the shear parameters $\{s_{j,k}\}_{k\in\mathbb{Z}^{n-1}} \subset \mathbb{R}^{n-1}$ can be picked dependent on j, so that the directionality of the representation is allowed to change with the scale. Then, in order to provide a "uniform covering", the location parameter is allowed to describe a different grid depending on j, and hence on k: a possible choice is $t_{j,k,m} = S_{s_{j,k}} A_{a_j} m$, where $m \in \mathbb{Z}^n$. Finally, the translation parameter can be chosen to belong to $c_1\mathbb{Z} \times \ldots \times c_n\mathbb{Z}$ for some $(c_1, \ldots, c_n) \in (\mathbb{R}^+)^n$. This provides some additional flexibility which is useful for some constructions.
As far as the previous definition concerns, we pick the dyadic sampling for the scaling parameter, *i.e.*, $a_j = 2^{-j},\ j \in \mathbb{Z}$. Next, we set $s_{j,k} = -k2^{-j/2},\ k \in \mathbb{Z}^{n-1}$, in order to get a larger number of directions as j is getting smaller. Finally, the location parameter is determined by adjusting the canonical grid $\mathbb{Z}^n$ to the particular scaling and shear parameter, *i.e.*, $t_{j,k,m} = S_{s_{j,k}} A_{a_j} m = S_{-k2^{-j/2}} A_{2^{-j}} m,\ m \in \mathbb{Z}^n$. Combining all this and observing that $A^{-1}_{2^{-j}} S^{-1}_{-k2^{-j/2}} = A_{2^j} S_{k2^{-j/2}} = S_k A_{2^j}$, the final result is the regular discrete shearlet system as defined above.

Similarly to the continuous case, the discrete shearlet transform, for the regular case, can be defined as follows.

Definition 4.17 ([101])**.** For $\psi \in L^2(\mathbb{R}^n)$ and $(j, k, m) \in \mathbb{Z} \times \mathbb{Z}^{n-1} \times \mathbb{Z}^n$, the *discrete shearlet transform* of $f \in L^2(\mathbb{R}^n)$ is the map defined by

$$f \to \mathscr{SH}_\psi f(j, k, m) = \langle f, \psi_{j,k,m}\rangle.$$

Thus, $\mathscr{SH}_\psi$ maps the function f to the coefficients $\mathscr{SH}_\psi f(j, k, m)$ associated with the scale index j, the orientation index k, and the position index m.
Clearly, the previous definition can be extended to the irregular shearlet systems in the most natural way.

Being able to derive the condition under which a discrete shearlet system $\mathcal{SH}(\psi)$ forms a basis or, more generally, a frame, is fundamental in order to apply shearlet systems as analysis and synthesis tools, as pointed out in Appendix A.1.

Notice that the analysis operator that associates a function to its frame coefficients can be expressed by means of a matrix. Even if, in general, an explicit expression for this matrix is useless, there exist some special cases in which it is needed: the digital setting we propose for ROI CT is, indeed, one of these. Additional details about this topic shall be given in Section 4.4.

Similarly to the classical wavelet case, we are interested in selecting a generator ψ with special properties, *e.g.*, regularity, well localization, vanishing moments, and compact support, so that the corresponding basis or frame of shearlets has satisfactory approximation properties. Toward this, the 2D classical shearlets from Example 4.9 are particularly useful examples since, as stated in the following result, they generate a Parseval frames for $L^2(\mathbb{R}^2)$. In Subsection 4.3.2, we shall see that a similar result holds true for 3D pyramid-adapted discrete shearlet system.

Proposition 4.18 ([99]). *Let $\psi \in L^2(\mathbb{R}^2)$ be a classical shearlet. Then $\mathcal{SH}(\psi)$ is a Parseval frame for $L^2(\mathbb{R}^2)$.*

In particular, the previous proposition implies that there exist Parseval frames $\mathcal{SH}(\psi)$ of well localized discrete shearlets, being the classical shearlet ψ a well localized function. As one can easily imagine, it is possible to remove the assumption of well localization for ψ, giving rise to other types of discrete shearlet systems, which might form not only tight frames but also orthonormal bases. So far, it seems that well localized shearlet orthonormal bases do not exist ([101]).
Finally, observe that the shearlet systems generated by classical shearlets are band-limited, *i.e.*, they have compact support in the frequency domain and, hence, cannot be compactly supported in the spatial domain. However, compactly supported discrete shearlet systems can be required, *e.g.*, to achieve spatial domain localization. Since this topic is far from the aim of this thesis, we refer [102] for more details.

4.3.2 Cone- and pyramid-adapted discrete shearlet systems

Given that (ir)regualr discrete shearlet systems are derived by appropriately sampling the continuous parameter set, they share the same biased treatment of the directions of continuous shearlet systems. To address this problem, the frequency domain can be partitioned into cones in 2D and pyramids in 3D.

In 2D, the definition is given for the regular parameter set, since it is much more frequently used. Clearly, it can be extended to the irregular shearlet systems in the most natural way. To allow more flexibility and enable changes to the density of the translation grid, a sampling factor $c = (c_1, c_2) \in (\mathbb{R}^+)^2$ in the translation index is introduced.

Definition 4.19 ([101])**.** For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ and $c = (c_1, c_2) \in (\mathbb{R}^+)^2$, the *(regular) cone-adapted discrete shearlet system* $\mathcal{SH}(\varphi, \psi, \tilde{\psi}; c)$ is defined by

$$\mathcal{SH}(\varphi, \psi, \tilde{\psi}; c) = \Phi(\phi; c_1) \cup \Psi(\psi; c) \cup \tilde{\Psi}(\tilde{\psi}; c),$$

where

$$\begin{aligned}
&\Phi(\phi; c_1) = \{\phi_m = \phi(\cdot - c_1 m) : m \in \mathbb{Z}^2\},\\
&\Psi(\psi; c) = \{\psi_{j,k,m} = 2^{\frac{3}{4}j}\, \psi(S_k A_{2^j} \cdot - M_c m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \mathbb{Z}^2\},\\
&\tilde{\Psi}(\tilde{\psi}; c) = \{\tilde{\psi}_{j,k,m} = 2^{\frac{3}{4}j}\, \tilde{\psi}(S_k^T \tilde{A}_{2^j} \cdot - \tilde{M}_c m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \mathbb{Z}^2\}.
\end{aligned}$$

with

$$M_c = \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \qquad \text{and} \qquad \tilde{M}_c = \begin{pmatrix} c_2 & 0 \\ 0 & c_1 \end{pmatrix}.$$

Clearly, if $c = (1, 1)$, the parameter c is omitted in the formulae above.

The generating functions ϕ will be referred to as *shearlet scaling functions* and the generating functions ψ, $\tilde{\psi}$ as *shearlet generators*. Analogously to the continuous case, the system $\Phi(\phi; c_1)$ is associated with the low frequency region, and the systems $\Psi(\psi; c)$ and $\tilde{\Psi}(\tilde{\psi}; c)$ are associated with the conic regions $\mathscr{C}_1 \cup \mathscr{C}_3$ and $\mathscr{C}_2 \cup \mathscr{C}_4$, respectively, as depicted in Figure 4.5.

The restrictions for the shear parameter prescribed above apply also in the 3D case. The following definition is a (discrete) special case of the general setup presented in Subsections 4.2.2.

Definition 4.20 ([103])**.** Let $c = (c_1, c_2) \in (\mathbb{R}^+)^2$ and $k = (k_1, k_2) \in \mathbb{Z}^2$. The *pyramid-adapted discrete shearlet system* $\mathcal{SH}(\phi, \psi^{(1)}, \psi^{(2)}, \psi^{(3)}; c)$ generated by ϕ, $\psi^{(1)}, \psi^{(2)}, \psi^{(3)} \in L^2(\mathbb{R}^3)$ is defined by

$$\mathcal{SH}(\phi, \psi^{(1)}, \psi^{(2)}, \psi^{(3)}; c) = \Phi(\phi; c_1) \cup \Psi^{(1)}(\psi^{(1)}; c) \cup \Psi^{(2)}(\psi^{(2)}; c) \cup \Psi^{(3)}(\psi^{(3)}; c),$$

where

$$\begin{aligned}
&\Phi(\phi; c_1) = \{\phi_m = \phi(\cdot - m) : m \in c_1 \mathbb{Z}^3\},\\
&\Psi^{(d)}(\psi^{(d)}; c) = \{\psi^{(d)}_{j,k,m} = 2^j\, \psi^{(d)}(S^{(d)}_k A^{(d)}_{2^j} \cdot - m) : j \geq 0,\ |k| \leq \lceil 2^{j/2} \rceil,\ m \in M^{(d)}_c \mathbb{Z}^3\},
\end{aligned}$$

where $d = 1, 2, 3$, $j \in \mathbb{N}_0$, $k \in \mathbb{Z}^2$ and the vector notation $|k| \leq K$ for $k = (k_1, k_2)$ and $K > 0$ stands for $|k_1| \leq K$ and $|k_2| \leq K$. Here, $A^{(d)}_{2^j}$, $d = 1, 2, 3$ are the paraboloidal scaling matrices defined by:

$$A^{(1)}_{2^j} = \begin{pmatrix} 2^j & 0 & 0 \\ 0 & 2^{j/2} & 0 \\ 0 & 0 & 2^{j/2} \end{pmatrix}, \quad A^{(2)}_{2^j} = \begin{pmatrix} 2^{j/2} & 0 & 0 \\ 0 & 2^j & 0 \\ 0 & 0 & 2^{j/2} \end{pmatrix}, \quad A^{(3)}_{2^j} = \begin{pmatrix} 2^{j/2} & 0 & 0 \\ 0 & 2^{j/2} & 0 \\ 0 & 0 & 2^j \end{pmatrix}.$$

The shear matrices $S_k^{(d)}$, $d = 1, 2, 3$, are given by:

$$S_k^{(1)} = \begin{pmatrix} 1 & k_1 & k_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad S_k^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ k_1 & 1 & k_2 \\ 0 & 0 & 1 \end{pmatrix}, \quad S_k^{(3)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ k_1 & k_2 & 1 \end{pmatrix}.$$

The matrices $M_c^{(1)} = \mathrm{diag}(c_1, c_2, c_2)$, $M_c^{(2)} = \mathrm{diag}(c_2, c_1, c_2)$, and $M_c^{(3)} = \mathrm{diag}(c_2, c_2, c_1)$ define the translation lattices.

Notice that, beside the paraboloidal scaling matrix considered in Definition 4.20, there is another possibility:

$$\begin{pmatrix} 2^j & 0 & 0 \\ 0 & 2^{j/2} & 0 \\ 0 & 0 & 2^j \end{pmatrix} \qquad \text{instead of} \qquad \begin{pmatrix} 2^j & 0 & 0 \\ 0 & 2^{j/2} & 0 \\ 0 & 0 & 2^{j/2} \end{pmatrix}.$$

The former choice leads to needle-like shearlets, while the latter one to plate-like shearlets. This distinction is essential to apply shearlets as a machinery to distinguish different types of singularities (1D or 2D). Usually, which one is the suitable choice is problem-dependent [101].

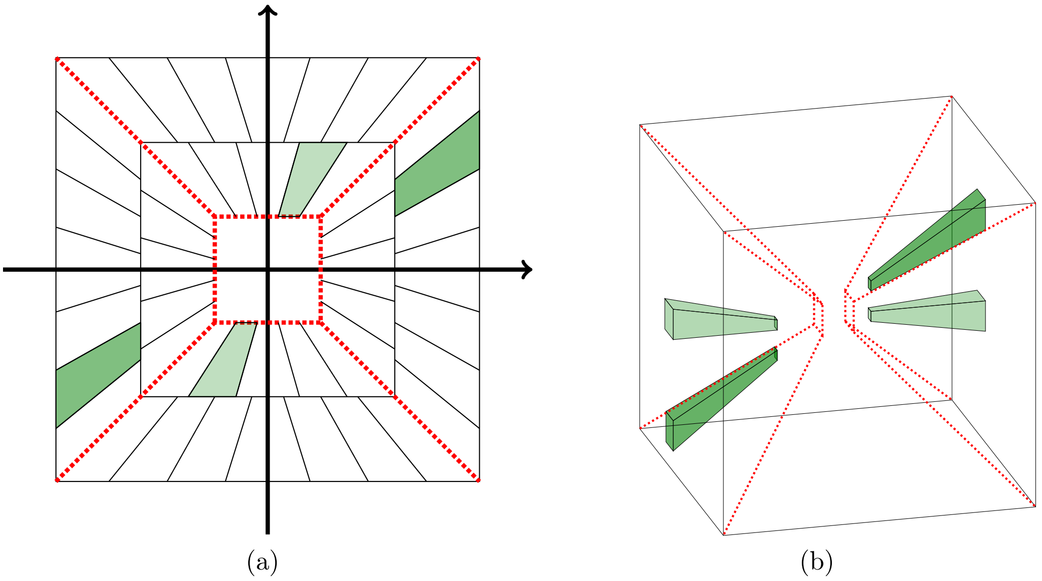


Figure 4.7: (a) Tiling of the frequency plane induced by a cone-adapted Parseval frame of shearlets in 2D. (b) Support of two shearlet elements $\psi_{j,k,m}$ in the frequency domain in 3D. The two shearlet elements have the same scale parameter j, but different shearing parameters $k = (k_1, k_2)$.

The discrete shearlet transform associated with cone and pyramid-adapted discrete shearlet systems is stated in the following definition.

Definition 4.21 ([66, 102])**.** Let $\Lambda = \mathbb{N}_0 \times \{\lceil 2^{j/2} \rceil, \dots, \lceil 2^{j/2} \rceil\} \times \mathbb{Z}^2$. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, the *cone-adapted discrete shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is the map defined by

$$f \to \mathscr{SH}_{\phi,\psi,\tilde{\psi}} f(m', (j,k,m), (\tilde{j}, \tilde{k}, \tilde{m})) = (\langle f, \phi_{m'} \rangle, \langle f, \psi_{j,k,m} \rangle, \langle f, \tilde{\psi}_{\tilde{j},\tilde{k},\tilde{m}} \rangle),$$

where $(m', (j,k,m), (\tilde{j}, \tilde{k}, \tilde{m})) \in \mathbb{Z}^2 \times \Lambda^2$. For $\psi \in L^2(\mathbb{R}^3)$, the *3D pyramid-adapted discrete shearlet transform* of $f \in L^2(\mathbb{R}^3)$ is the map

$$f \longrightarrow \mathscr{SH}_{\psi} f(j, k_1, k_2, m) = \begin{cases} \langle f, \psi^{(1)}_{j,k_1,k_2,m} \rangle & \text{if } |k_1|, |k_2| \leq 1 \\ \langle f, \psi^{(2)}_{j,\frac{1}{k_1},\frac{k_2}{k_1},m} \rangle & \text{if } |k_1| > 1, |k_2| \leq |k_1| \\ \langle f, \psi^{(3)}_{j,\frac{k_1}{k_2},\frac{1}{k_2},m} \rangle & \text{if } |k_2| > 1, |k_2| > |k_1| \end{cases}$$

where $j \in \mathbb{N}_0$, $k_1, k_2 \in \{\lceil 2^{j/2} \rceil, \dots, \lceil 2^{j/2} \rceil\}$ and $m \in \mathbb{Z}^3$.

Finally, our goal is to derive Parseval frames both in the 2D and in the 3D case. First of all, the next proposition states that a 2D classical shearlet is a shearlet generator of a Parseval frame for the subspace of $L^2(\mathbb{R}^2)$ of functions whose frequency support lies in the union $\mathscr{C}_1 \cup \mathscr{C}_3$ of the two cones $\mathscr{C}_1$ and $\mathscr{C}_3$.

Proposition 4.22 ([66])**.** *Let $\psi \in L^2(\mathbb{R}^2)$ be a classical shearlet. Then the shearlet system*

$$\Psi(\psi) = \{\psi_{j,k,m} = 2^{\frac{3}{4}j}\, \psi(S_k A_{2^j} \cdot -m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \mathbb{Z}^2\}$$

is a Parseval frame for $L^2(\mathscr{C}_1 \cup \mathscr{C}_3)^\vee = \{f \in L^2(\mathbb{R}^2) : \operatorname{supp}(\hat{f}) \subset \mathscr{C}_1 \cup \mathscr{C}_3\}$.

Clearly, by replacing ψ with $\tilde{\psi}$, a similar result to Proposition 4.22 holds true for the subspace $L^2(\mathscr{C}_2 \cup \mathscr{C}_4)^\vee$. Thus, a Parseval frame for the whole space $L^2(\mathbb{R}^2)$ can be build up by piecing together Parseval frames associated with different cones on the frequency domain along with a coarse scale system which takes care of the low frequency region. Hence, the following result holds true.

Theorem 4.23 ([66])**.** *Let $\psi \in L^2(\mathbb{R}^2)$ be a classical shearlet, and let $\phi \in L^2(\mathbb{R}^2)$ be chosen so that, for a.e. $\xi \in \mathbb{R}^2$,*

$$|\hat{\phi}(\xi)|^2 + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} |\hat{\psi}(S^T_{-k} A_{s^{-j}} \xi)|^2 1_C + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} |\hat{\psi}(S^T_{-k} \tilde{A}_{s^{-j}} \xi)|^2 1_{\tilde{C}} = 1,$$

where 1_C (respectively, $1_{\tilde{C}}$) is the characteristic function of the set C (respectively, $\tilde{C}$). Let $\mathcal{P}_C \Psi(\psi)$ denote the set of shearlet elements in $\Psi(\psi)$ after projecting their Fourier transforms onto $C = \left\{(\xi_1, \xi_2) \in \mathbb{R}^2 : \left|\frac{\xi_2}{\xi_1}\right| \leq 1\right\}$, and let $\mathcal{P}_{\tilde{C}} \tilde{\Psi}(\tilde{\psi})$, where $\tilde{C} = \mathbb{R}^2 \setminus C$, be defined analogously. Then, the modified cone-adapted discrete shearlet system $\Phi(\phi) \cup \mathcal{P}_C \Psi(\psi) \cup \mathcal{P}_{\tilde{C}} \tilde{\Psi}(\tilde{\psi})$ is a Parseval frame for $L^2(\mathbb{R}^2)$.

In Figure 4.7a is illustrated the tiling of the frequency plane induced by a cone-adapted Parseval frame of shearlets.
Notice that, when the cone-based shearlet systems are projected onto C and $\tilde{C}$, the shearlet elements overlapping the boundary lines $\xi_1 = \pm\xi_2$ in the frequency domain are cut so that the "boundary" shearlets lose their regularity properties. This main drawback can be avoided by redefining the "boundary" shearlets in such a way that their regularity is preserved [68].

As far as the 3D case concerns, the first step toward a Parseval frame result is the following Theorem.

Theorem 4.24 ([103])**.** *Let ψ be a band-limited shearlet as defined in Example 4.10. Then the family of functions $\mathcal{P}_{\mathscr{P}_1}\Psi(\psi)$ forms a tight frame for $L^2(\mathscr{P}_1)^\vee = \{f \in L^2(\mathbb{R}^3) : \operatorname{supp}(\hat{f}) \subset \mathscr{P}_1\}$, where $\mathcal{P}_{\mathscr{P}}$ denotes the orthogonal projection onto $L^2(\mathscr{P}_1)^\vee$ and*

$$\Psi(\psi) = \left\{\psi_{j,k,m} : j \geq 0,\ |k| \leq \lceil 2^{j/2} \rceil,\ m \in \frac{1}{8}\mathbb{Z}^3\right\}.$$

By Theorem 4.24 and a change of variables, it is possible to construct shearlet tight frames for $L^2(\mathscr{P}_1)^\vee$, $L^2(\mathscr{P}_2)^\vee$ and $L^2(\mathscr{P}_3)^\vee$, respectively. By denoting with $\mathcal{P}_C$ the orthogonal projection onto the closed subspace $L^2(C)^\vee$ for some measurable set $C \subset \mathbb{R}^3$, any function $f \in L^2(\mathbb{R}^3)$ can be expressed by

$$f = \mathcal{P}_{\mathscr{C}} f + \mathcal{P}_{\mathscr{P}_1} f + \mathcal{P}_{\mathscr{P}_2} f + \mathcal{P}_{\mathscr{P}_3} f$$

since $\mathbb{R}^3 = \mathscr{C} \cup \mathscr{P}_1 \cup \mathscr{P}_2 \cup \mathscr{P}_3$ as a disjoint union. Here, $\phi \in L^2(\mathbb{R}^3)$ is a wavelet such that $\Phi(\phi; \frac{1}{8})$ forms a tight frame for $L^2(\mathbb{R})$. Then, the projection $\mathcal{P}_{\mathscr{P}_1} f$ can be expanded in terms of the corresponding tight frame $\mathcal{P}_{\mathscr{P}_1}\Psi(\psi)$ and similarly for the other three projections. Hence, the representation of f is the sum of these four expansions.

In conclusion, notice that the discrete shearlet frame introduced in this subsection can be derived using a powerful methodology called *Coorbit Theory.* This approach is used to derive different discretizations while ensuring frame properties. More details about this topic can be found in [40, 41, 39].

4.3.3 Optimally sparse approximations

One of the main reason for the introduction of shearlets was the lack of directional sensitivity of traditional multiscale methods, like wavelets. Indeed, shearlets have been specially designed to account for the anisotropic features, like edges, that frequently dominate multidimensional phenomena. This anisotropic nature allows to obtain optimally sparse approximations, in the sense specified by the following Definition 4.26.

Definition 4.25 ([103])**.** For a fixed $c_{\mathscr{E}} > 0$, the class $\mathscr{E}^2(\mathbb{R}^n)$ of *cartoon-like images* is the set of functions $f : \mathbb{R}^n \to \mathbb{C}$ of the form

$$f = f_0 + f_1 \, 1_B,$$

where $B \subset [0,1]^n$ and $f_i \in \mathscr{C}^2(\mathbb{R}^n)$ are functions with $\operatorname{supp}(f_i) \subset [0,1]^n$ and $\|f_i\|_{\mathscr{C}^2} \leq c_{\mathscr{E}}$ for each $i = 0, 1$.

Definition 4.26 ([103])**.** Let $\Phi = (\phi_i)_{i \in \mathcal{I}}$ be a frame for $L^2(\mathbb{R}^n)$ with $n = 2$ or $n = 3$. We say that Φ *provides optimally sparse approximations* of cartoon-like images if, for each $f \in \mathscr{E}^2(\mathbb{R}^n)$, the associated J-term approximation f_J obtained by keeping the J largest coefficients of $(\langle f, \phi_i \rangle)_{i \in \mathcal{I}}$ satisfies

$$\|f - f_J\|_2^2 < J^{-\frac{2}{n-1}} \qquad \text{as } J \to \infty,$$

and

$$|(\langle f, \phi_i \rangle)_l^*| < l^{-\frac{n+1}{2(n-1)}} \qquad \text{as } l \to \infty,$$

up to a log-factor.

The class of cartoon-like images is generally used to derive results in approximation theory since it provides a simplified model of natural images, which emphasizes anisotropic features, in particular edges.

The following two results state that, by using cartoon-like images as model class in 2D and in 3D, the approximation properties can be measured by considering the decay rate of the L^2-error of the best J-term approximation. Notice that, to show the following results, a slightly different construction of shearlets tight frame are needed, which can be found in [68].

Theorem 4.27 ([102])**.** *Let $\Phi(\phi) \cup P_C \Psi(\psi) \cup P_{\tilde{C}} \tilde{\Psi}(\tilde{\psi})$ be a Parseval frame for $L^2(\mathbb{R}^2)$ as defined in Theorem 4.23, where $\psi \in L^2(\mathbb{R}^2)$ is a classical shearlet and $\hat{\psi} \in \mathscr{C}_0^\infty(\mathbb{R}^2)$. Let f be a cartoon-like image and f_J be its best J-term approximation. Then there exists a constant $C > 0$, independent of f and J, such that*

$$\|f - f_J\|^2 \leq C \, J^{-2} \log^3(J) \qquad \textit{as } J \to \infty.$$

Theorem 4.28 ([103])**.** *Assume that $\phi, \psi^{(1)}, \psi^{(2)}, \psi^{(3)} \in L^2(\mathbb{R}^3)$ are band-limited and $\mathscr{C}^\infty$ in the frequency domain and that the shearlet system $\mathcal{SH}(\phi, \psi^{(1)}, \psi^{(2)}, \psi^{(3)}; c)$ forms a frame for $L^2(\mathbb{R}^3)$. For any $c_{\mathscr{E}} > 0$, the shearlet frame $\mathcal{SH}(\phi, \psi^{(1)}, \psi^{(2)}, \psi^{(3)}; c)$ provides optimally sparse approximations of functions $f \in \mathscr{E}^2(\mathbb{R}^3)$ in the sense of Definition 4.26.*

These results are fundamental and will be useful in the discussion of Chapter 6 about the results of the numerical assessment of the ROI CT problem. So far, the only representation system able to provide optimally sparse approximations of the class of cartoon-like images in 2D, as well as in 3D, are shearlets.

4.4 Implementation and softwares

One main advantage of the shearlet approach is the unified treatment of the continuous and discrete settings that allow for numerical implementations which faithfully digitalize the discrete shearlet transform. So far, several numerical implementations of the discrete shearlet transform have been proposed. All the developed approaches can be grouped into two main categories: *Fourier-domain* based and *spatial-domain* based. The former relies on the fact that the cone-adapted and pyramid-based discrete shearlet transform provides a particular decomposition of the frequency plane. The Fourier-based approach aims to directly produce the same frequency tiling. The latter, the spatial-domain based, is a method where the filters associated with the transform are implemented by a convolution in the spatial domain. The details of the implementations can be found in [48, 115, 105]. Finally, in order to develop MRA-based implementations, a multiresolution analysis similar to the one associated with wavelets has been proposed [69, 104, 72].

To date, different packages providing numerical shearlet algorithms, prototyped in Matlab, are available. A first one is associated with [48] and can be downloaded from `www.math.uh.edu/~dlabate`; a second one can be found at `www.ShearLab.org` and is associated with [115, 105]; a last one is available at `www.mathematik.uni-kl.de/imagepro/software/ffst` and related details can be found in [75]. All packages enjoy 2D and 3D implementations, for both band-limited and compactly supported shearlets. Also a CUDA based parallel implementation is available and can be downloaded from `www.math.uh.edu/~dlabate`.

The Spot toolbox, an object-oriented Matlab package by Ewout van den Berg and Michael P. Friedlander implementing linear operators, can be combined with the above mentioned packages to provide an operator expression for the underlying matrix that gives the shearlet coefficients. We already pointed out that, in general, an explicit expression for this matrix is useless. However, one of the algorithms we propose for the solution of the ROI CT problem requires an expression for this matrix. The Spot toolbox is really useful in this sense, since a Spot operator represents a matrix, and can be treated in a similar way, but it does not rely on the matrix itself to implement most of the methods. This is helpful not only whenever the explicit transform operator is needed and not available, but also when Matlab provides an explicit expression for the matrix and it is not practical for some reason (*e.g.*, problems of large sizes). The latest release of Spot is available at the GitHub page `https://github.com/mpf/spot` and it is constantly updated.

So far, various routines are available in Spot to obtain Daubechies wavelets matrices. As far as the shearlets concern, Wang-Q Lim, co-author of the shearlets package at `www.math.uh.edu/~dlabate`, kindly provided a Spot-compliant routine to obtain the shearlet operator, that will be used in the numerical experimentation in Chapter 6.

Chapter 5

Iterative reconstruction techniques for ROI CT

Nowadays, iterative techniques to address CT-type problems are going through a revived interest, due to the technological advances in computer science [8, 84]. However, iterative methods, except in few special cases, have been an almost compelling choice in many medical imaging, microscopy and astronomy image reconstruction problems.

The aim of this Chapter is to put on two recently introduced iterative techniques, the scaled gradient projection (SGP) [17] method and the variable metric inexact line-search algorithm (VMILA) [14], with application to ROI CT data. To this end, basic facts on statical approach and general iterative reconstruction schemes will be retrieved, following the approach of [9, 10]. Then, SGP and VMILA main features will be investigated, discussing practical considerations for the implementation to the ROI CT problem. Since all the details regarding the general setup and convergence results are carried out in [17, 14], for simplicity the following presentation is heuristic and implementation-oriented.

5.1 Statistical approach for image reconstructions

The CT problem as defined in (2.1) can be understood as a particular case of the Fredholm integral equation of the first kind:

$$y(\theta,\tau)=\int_{\mathbb{R}^d} K(\mathbf{x},\theta,\tau)f(\mathbf{x})\;d\mathbf{x},$$

where $d=2$, 3 and the kernel $K(\mathbf{x},\theta,\tau)$ is given by

$$K(\mathbf{x},\theta,\tau)=\delta(\tau-\langle\mathbf{x},\omega_\theta\rangle).$$

It is well known that the solution of the Fredholm integral equation in the discrete domain leads to the following system of linear equations:

$$\mathbf{K}\,\mathbf{f} = \mathbf{y}, \tag{5.1}$$

where $\mathbf{f} = \{f_i\}_{i=1}^{n} \in \mathbb{R}^n$ is a discrete representation of the object to be imaged, $\mathbf{y} = \{y_j\}_{j=1}^{m} \in \mathbb{R}^m$ is a vector of the measured data and $\mathbf{K} \in \mathbb{R}^{m \times n}$ is the system matrix, namely it describes the measurement process.
Retaining the notation introduced in Chapter 2, $n = N^d$, where $d = 2, 3$ is the total number of pixels, resp. voxels; m is equal to the total number of detector elements N_{dtc} multiplied by the number of views N_θ. The system matrix $\mathbf{K}$ models the discrete forward operator as described in Chapter 3, *i.e.*, each element w_{ij} of $\mathbf{K} = \mathbf{W}$ represents the contribution of the i-th pixel (resp., voxel) f_i to the j-th ray position y_j. Notice that, in the ROI CT problem, the overall picture is slightly different: the acquired projection data are the ROI sinogram $\mathbf{y}_0 \in \mathbb{R}^m$ and a mask $\mathbf{M} \in \mathbb{R}^{m \times m}$ selects the rays meeting the ROI, hence premultiplies the system matrix $\mathbf{W}$. The goal is to recover $\mathbf{f}$ from $\mathbf{y}_0$.

In general, the matrix $\mathbf{K}$ is large and ill-conditioned: hence, the direct matrix inversion in equation (5.1) is practically infeasible. To overcome this problem, a statistical formulation of the image reconstruction problem can be profitable and it is, indeed, quite natural. According to this approach, data are seen as realizations of random variables: that is, each component y_j of the measured data is the realization of a random variable Y_j, with $\mathbf{Y} = \{Y_j\}_{j=1}^{m}$ denoting the vector valued random variable corresponding to the vector of measured data. In practice, a modeling of this type accounts for the noise introduced by the detection system. Reasonable assumptions for its probability density $p_{\mathbf{Y}}(\mathbf{y};\mathbf{f})$, that clearly depends on the object to be reconstructed, are the following:

(A1) The random variables Y_j and Y_k associated to different elements of the detector are statistically independent. Hence:

$$p_{\mathbf{Y}}(\mathbf{y};\mathbf{f}) = \prod_{j=1}^{m} p_{Y_j}(\mathbf{y}_j;\mathbf{f}).$$

(A2) The expected value of Y_j is given by the exact value of the incoming radiation, namely:

$$E\{\mathbf{Y}\} = \int \mathbf{y}\; p_{\mathbf{Y}}(\mathbf{y};\mathbf{f})\; d\mathbf{y} = \mathbf{K}\mathbf{f}.$$

Thus, the problem now consist in finding an estimate $\bar{\mathbf{f}}$ of the object to be reconstructed, that at this point appears as a set of unknown parameters, corresponding to the detected data $\mathbf{y}$, whose probability density $p_{\mathbf{Y}}(\mathbf{y};\mathbf{f})$ is known. The standard approach is to introduce the *likelihood function*, defined by

$$L_{\mathbf{y}}^{\mathbf{Y}}(\mathbf{f}) = p_{\mathbf{Y}}(\mathbf{y};\mathbf{f}),$$

which depends only on $\mathbf{f}$, and to look for its (possibly existing) maximum point:

$$\mathbf{f}^* = \operatorname*{argmax}_{\mathbf{f}\in\mathbb{R}^n} L_{\mathbf{y}}^{\mathbf{Y}}(\mathbf{f}).$$

This approach is called *maximum likelihood* (ML) estimation. In practice, it can be convenient to consider the negative logarithm (usually addressed to as *neglog*) of the likelihood function so that the maximization problem is transformed into a minimization one, given that the neglog is strictly convex. The problem now reads as:

$$\mathbf{f}^* = \operatorname*{argmin}_{\mathbf{f}\in\mathbb{R}^n} \Upsilon_{DM}(\mathbf{f};\mathbf{y}). \tag{5.2}$$

where

$$\Upsilon_{DM}(\mathbf{f};\mathbf{y}) = -A\log\big(L_{\mathbf{y}}^{\mathbf{Y}}(\mathbf{f})\big) + B$$

with A and B suitable constants, introduced to simplify the expression of the functional. The subscript index DM stands for *data mismatch* since, in general, this terms accounts for the discrepancy between the detected data and ideal measurements.

As an example, we consider in Examples 5.1 and 5.2 two classical types of noise, which will be investigated in the numerical experiments in Chapter 6.

Example 5.1 (additive white Gaussian noise). In this case, $\mathbf{Y}$ is given by

$$\mathbf{Y} = \mathbf{K}\mathbf{f} + E.$$

Here, E is a vector-valued random variable whose components are statistically independent, all having the same Gaussian distribution with expected value 0 and variance σ^2. Therefore, we have:

$$p_{\mathbf{Y}}(\mathbf{y};\mathbf{f}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \mathrm{e}^{-\dfrac{\|\mathbf{y}-\mathbf{K}\mathbf{f}\|_2^2}{\sqrt{2\sigma^2}}},$$

and

$$\Upsilon_{DM}(\mathbf{f};\mathbf{y}) = \|\mathbf{K}\mathbf{f}-\mathbf{y}\|_2^2$$

by taking $A = 2\sigma^2$ and $B = -2\sigma^2 m\log(\sqrt{2\pi\sigma^2})$. In this case the ML approach coincides with the well-known least squares approach.

Example 5.2 (Poisson noise). In such a case, each Y_j is a Poisson random variable with expected value given by $(\mathbf{Kf})_j$, yielding:

$$p_{\mathbf{Y}}(\mathbf{y};\mathbf{f}) = \prod_{j=1}^{m} \frac{\mathrm{e}^{-(\mathbf{Kf})_j}\,(\mathbf{Kf})_j^{y_j}}{(y_j)!},$$

and

$$\Upsilon_{DM}(\mathbf{f};\mathbf{y}) = \sum_{j=1}^{m}\left\{ y_j \log\left(\frac{y_j}{(\mathbf{Kf})_j} + (\mathbf{Kf})_j - y_j \right)\right\}$$

is the so-called Kullback-Leibler (KL) divergence, a convex (strictly convex if the equation $\mathbf{Kf} = 0$ has only the solution $\mathbf{f} = 0$), non-negative and locally bounded functional. The constants A and B can be explicitly determined by approximating the factorial $(y_j)!$ with Stirling's approximation:

$$\log((y_j)!) = y_j \log(y_j) - y_j + \log(\sqrt{2\pi y_j}).$$

Poisson noise is generally used to describe the noise affecting counting processes and, for this reason, it is sometimes referred to as "photon noise": the specific effect arising in this problem is known as *checkerboard effect*.

Two comments are in order. First, notice that both minimization problems are indeed ill-posed (motivations and clarifying examples can be found in [50, 125]). As a consequence, the minimum points $\mathbf{f}^*$ of the functionals corresponding to the different noise models do not provide, in general, sensible estimates $\bar{\mathbf{f}}$ of the unknown object and one is not interested in computing them.

Secondly, the minimization of the above functionals should be considered on the non-negative orthant since objects are non-negative in many interesting cases of object reconstruction, including ROI CT. This expresses the necessity of introducing some additional information on the object, *e.g.*, in the form of statistical properties of the object. This is a state-of-the-art approach, called *Bayesian approach*.

In other words, the unknown object $\mathbf{f}$ is also assumed to be a realization of a vector-valued random variable $\mathbf{F} = \{F_i\}_{i=1}^{n}$, whose probability density $p_{\mathbf{F}}(\mathbf{f})$ is given by

$$p_{\mathbf{F}}(\mathbf{f}) = \frac{1}{Z}\mathrm{e}^{-\mu\Theta(\mathbf{f})}.$$

Here, Z is a normalization constant, μ is a positive parameter, usually referred to as *regularization parameter*, and $\Theta(\mathbf{f})$ is a (possibly convex) functional. This is generally called *prior* and the one introduced belongs to the Gibbs type.

Then, the probability density $p_{\mathbf{Y}}(\mathbf{y};\mathbf{f})$ can be understood as the conditional probability density of $\mathbf{Y}$ when the random variable $\mathbf{F}$ assumes the value $\mathbf{f}$:

$$p_{\mathbf{Y}}(\mathbf{y}|\mathbf{f}) = p_{\mathbf{Y}}(\mathbf{y}|\mathbf{F} = \mathbf{f}),$$

and the joint probability density of the random variables $\mathbf{F}$ and $\mathbf{Y}$ is given by

$$p_{\mathbf{FY}}(\mathbf{f},\mathbf{y}) = p_{\mathbf{Y}}(\mathbf{y}|\mathbf{f})\, p_{\mathbf{F}}(\mathbf{f}).$$

By employing the Bayes formula for the conditional probability, we obtain the *a posteriori* probability density $P^{\mathbf{F}}_{\mathbf{y}}(\mathbf{f})$ of $\mathbf{F}$:

$$P^{\mathbf{F}}_{\mathbf{y}}(\mathbf{f}) = p_{\mathbf{F}}(\mathbf{f}|\mathbf{y}) = \frac{p_{\mathbf{Y}}(\mathbf{y}|\mathbf{f})\, p_{\mathbf{F}}(\mathbf{f})}{p_{\mathbf{Y}}(\mathbf{y})} = L^{\mathbf{Y}}_{\mathbf{y}}(\mathbf{f}) \frac{p_{\mathbf{F}}(\mathbf{f})}{p_{\mathbf{Y}}(\mathbf{y})},$$

where $p_{\mathbf{Y}}(\mathbf{y}) = \int p_{\mathbf{FY}}(\mathbf{f},\mathbf{y})\, d\mathbf{f}$ is the marginal probability density of $\mathbf{Y}$. Then, the problem of finding an estimate $\mathbf{f}^*$ of the unknown object reads as:

$$\mathbf{f}^* = \operatorname*{argmax}_{\mathbf{f}\in\mathbb{R}^n} \; P^{\mathbf{F}}_{\mathbf{y}}(\mathbf{f}),$$

that is, $\mathbf{f}^*$ is any object that maximizes the a posteriori probability density. This is called *maximum a posteriori* (MAP) estimate. Similarly to the likelihood case, it is convenient to consider the neglog function of $P^{\mathbf{F}}_{\mathbf{y}}(\mathbf{f})$. By assuming the Gibbs prior, this yields:

$$\mathbf{f}^* = \operatorname*{argmin}_{\mathbf{f}\in\mathbb{R}^n} \; \Upsilon(\mathbf{f};\mathbf{y})$$

where

$$\begin{aligned}\Upsilon(\mathbf{f};\mathbf{y}) &= -A\log P^{\mathbf{F}}_{\mathbf{y}}(\mathbf{f}) + B - A\log(Z) - A\log(p_{\mathbf{Y}}(\mathbf{y}))\\ &= \Upsilon_{DM}(\mathbf{f};\mathbf{y}) + \mu\,\Upsilon_R(\mathbf{f};\mathbf{y})\end{aligned}$$

with $\Upsilon_R(\mathbf{f};\mathbf{y}) = A\Theta(\mathbf{f})$, conceived as a regularization functional, hence the subscript index R. Observe that when $\mu = 0$ this is just the ML problem. Also, the minimum points satisfying the non-negativity constraint should be taken into account, leading to the general formulation:

$$\mathbf{f}^* = \operatorname*{argmax}_{\mathbf{f}\geq 0} \; \Upsilon_{DM}(\mathbf{f};\mathbf{y}) + \mu\,\Upsilon_R(\mathbf{f};\mathbf{y}). \tag{5.3}$$

Finally, notice that it is not obvious that a minimum point $\mathbf{f}^*$ of $\Upsilon(\mathbf{f};\mathbf{y})$ is a sensible estimate $\bar{\mathbf{f}}$ of the unknown object. In fact, the parameter μ is a "degree of freedom" and a wide literature exists on the problem of its optimal choice [50, 73].

5.2 Scaled gradient projection method

As already pointed out, in ML problems one generally does not want to reach the minimum, because of ill-posedness. Thus, ML problems are not treated as standard optimization problem. On the other hand, reaching the minimum is exactly the goal in the case of MAP problems. Therefore, this problems are usually addressed using different methods. In this section, we present an algorithm, the scaled gradient projection (SGP) method proposed in [17], that can provide a unified approach to both problems.

SGP is an iterative approach from the family of first-order descent methods that applies to convex (and non-convex), differentiable and constrained problems with "simple" feasible regions. This appealing features makes SGP perfectly suitable for the ROI CT formulation, given that the feasible region in (2.22) and (2.23) is usually either a box or a non-negativity constraint and, as already indicated, the involved functionals are convex and differentiable (when the L^2-norm is considered). First, a general description of SGP, whose main steps are stated in Algorithm 1, will be given and then some considerations for its practical implementation to the ROI CT problem will be discussed.

The main feature of SGP is the combination of non-expensive diagonally scaled gradient directions with steplength selection rules specially designed for these directions, exploiting a decomposition of the gradient. The $(k+1)$-th iteration, with $k = 0, 1, 2, \ldots$, is

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \mu_k \left[\mathcal{P}_{\mathcal{C}, D_k^{-1}} \Big(\mathbf{f}^{(k)} - \alpha_k D_k \nabla \Upsilon(\mathbf{f}^{(k)}; \mathbf{y}) \Big) - \mathbf{f}^{(k)} \right] \tag{5.4}$$

where μ_k, α_k are suitable steplengths, D_k is a scaling matrix and $\mathcal{P}_{\mathcal{C}, D_k^{-1}}$ denotes the projector onto a closed and convex set $\mathcal{C}$:

$$\mathcal{P}_{\mathcal{C},D}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{z} \in \mathcal{C}} \|\mathbf{z} - \mathbf{x}\|_D = \operatorname*{argmin}_{\mathbf{z} \in \mathcal{C}} \Big(\frac{1}{2} \mathbf{z}^T D \mathbf{z} - \mathbf{z}^T D \mathbf{x} \Big). \tag{5.5}$$

with $\|\mathbf{x}\|_D = \sqrt{\mathbf{x}^T D \mathbf{x}}$, *i.e.*, $\|\mathbf{x}\|_D$ is the norm induced by the symmetric positive definite matrix D. According to the formulation introduced in (5.3), $\mathcal{C} = \{\mathbf{f} \geq 0\}$.

In details, the following (possibly not unique) decomposition of the gradient [110] is considered:

$$-\nabla \Upsilon(\mathbf{f}; \mathbf{y}) = U(\mathbf{f}; \mathbf{y}) - V(\mathbf{f}; \mathbf{y}) \qquad U(\mathbf{f}; \mathbf{y}) \geq 0, \ V(\mathbf{f}; \mathbf{y}) > 0. \tag{5.6}$$

Clearly, such a decomposition always exists, since different choices of U and V can satisfy the above requirement. Also, the non-uniqueness of the decomposition is not a drawback: as a matter of fact, it reveals to be an advantage in some cases. We will refer to V as the positive part of the gradient decomposition; consequently, U will be addressed to as the negative part.

Next, any choice of the steplength α_k in a closed interval $[\alpha_{\min}, \alpha_{\max}] \subset \mathbb{R}^+$ and of the scaling matrix D_k in the compact set $\mathcal{D}_L$ is allowed, where $\mathcal{D}_L$ is the set of the symmetric positive definite matrices D such that $\|D\| \leq L$ and $\|D^{-1}\| \leq L$, for a given threshold $L > 1$. This is very important from a practical point of view because it makes their updating rules problem-related and performance-aware. In particular, SGP is equipped with an adaptive steplength selection based on the Barzilai-Borwein (BB) updating rules [6, 56]. In practice, by means of a variable threshold, one of the two different selection strategies

$$\alpha_k^{\mathrm{BB1}} = \underset{\alpha_k\in\mathbb{R}}{\operatorname{argmin}}\left\|B(\alpha_k)\,\mathbf{s}^{(k-1)} - \zeta^{(k-1)}\right\| \tag{5.7}$$

$$\alpha_k^{\mathrm{BB2}} = \underset{\alpha_k\in\mathbb{R}}{\operatorname{argmin}}\left\|\mathbf{s}^{(k-1)} - B(\alpha_k)^{-1}\zeta^{(k-1)}\right\| \tag{5.8}$$

is selected [59], where $B(\alpha_k) = (\alpha_k D_k)^{-1}$ approximates the Hessian matrix $\nabla^2\Upsilon(\mathbf{f}^{(k)};\mathbf{y})$, $\mathbf{s}^{(k-1)} = \mathbf{f}^{(k)} - \mathbf{f}^{(k-1)}$ and $\zeta^{(k-1)} = \nabla\Upsilon(\mathbf{f}^{(k)};\mathbf{y}) - \nabla\Upsilon(\mathbf{f}^{(k-1)};\mathbf{y})$.
As far as the scaling matrix concerns, the updating rule for each entry $d_i^{(k)}$ is:

$$d_i^{(k)} = \min\left\{L, \max\left\{\frac{1}{L}, \frac{\mathbf{f}_i^{(k)}}{V_i(\mathbf{f}^{(k)};\mathbf{y})}\right\}\right\} \qquad i = 1,\ \ldots, n \tag{5.9}$$

where L is an appropriate threshold and V_i is the i-th component of the positive part of the gradient decomposition V. The global convergence properties of SGP are ensured by exploiting a nonmonotone line-search strategy along the feasible direction [63, 43]. Such a strategy ensures that $\Upsilon(\mathbf{f}^{(k+1)};\mathbf{y})$ is lower than the maximum of the objective function on the last κ iterations by successive reductions of μ_k. Of course, if $\kappa = 1$ then the strategy reduces to the standard monotone Armijo rule.

For the sake of completeness, even if it is not the case of ROI CT, we observe that in some fields of application, including Microscopy and Astronomy, an additional constraint in the formulation (5.3) is often assumed, yielding $\mathcal{C} = \{\mathbf{f} \in \mathbb{R}^n : \mathbf{f} \geq 0,\ \sum_{i=1}^n f_i = c\}$, where the linear equation $\sum_{i=1}^n f_i = c$ is usually referred to as *flux condition*. In such a case, the following modification of the scaling matrix is considered:

$$D_k = \operatorname{diag}\left\{\max\left\{c_1, \frac{\mathbf{f}_i^{(k)}}{V_i(\mathbf{f}^{(k)};\mathbf{y})}\right\}\right\},$$

where $c_1 > 0$ is a prefixed threshold such that $c_1\|\mathbf{x}\|^2 \leq \mathbf{x}^T D\mathbf{x} \leq c_2\|\mathbf{x}\|^2$, for all $\mathbf{x} \in \mathbb{R}^n$. The threshold c_2 is chosen to be

$$c_2 = \frac{c}{\nu} \qquad \text{with} \quad \nu = \min_i\left\{\min_{\mathbf{f}\in\mathcal{C}}\{V_i(\mathbf{f};\mathbf{y})\}\right\}.$$

With this choice for the scaling matrix, the projection can obtained by solving a separable quadratic program for which efficient linear-time solvers can be used [17].

Algorithm 1 Scaled Gradient Projection Method

```
Choose the starting point $\mathbf{f}^{(0)} \in \mathcal{C}$, set the parameters $0 < \alpha_{\min} < \alpha_{\max}$ and $\beta, \gamma \in (0,1)$. Fix a positive integer $\kappa$.
for $k = 0, 1, 2, \dots$ do
    Step 1. Choose the parameter $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ and the scaling matrix $D_k \in \mathcal{D}_L$;
    Step 2. Projection: $\mathbf{z}^{(k)} = \mathcal{P}_{\mathcal{C},D_k^{-1}}(\mathbf{f}^{(k)} - \alpha_k D_k \nabla \Upsilon(\mathbf{f}^{(k)}; \mathbf{y}))$.
            if $\mathbf{z}^{(k)} = \mathbf{f}^{(k)}$ then stop: $\mathbf{f}^{(k)}$ is a stationary point; end if
    Step 3. Descent direction: $\mathbf{d}^{(k)} = \mathbf{z}^{(k)} - \mathbf{f}^{(k)}$;
    Step 4. Set $\mu_k = 1$ and $\Upsilon_{\max} = \max_{0 \le j \le \min(k, \kappa-1)} \Upsilon(\mathbf{f}^{(k-j)}; \mathbf{y})$;
    Step 5. Backtracking loop:
            if $\Upsilon(\mathbf{f}^{(k-j)}; \mathbf{y}) + \mu_k \mathbf{d}^{(k)}) \le \Upsilon_{\max} + \beta \mu_k \nabla \Upsilon(\mathbf{f}^{(k)}; \mathbf{y})^T \mathbf{d}^{(k)}$ then
                go to Step 6;
            else
                set $\mu_k = \gamma \mu_k$ and go to Step 5;
            end if
    Step 6. Set $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \mu_k \, \mathbf{d}^{(k)}$.
end for
```

Finally, notice that the SGP algorithm can be used as an iterative regularization method applied to the ML problem (un-regularized functional), by means of an early stopping technique. The above presentation ca be repeated almost *verbatim* by considering the functional $\Upsilon_{DM}(\mathbf{f}; \mathbf{y})$ in place of $\Upsilon(\mathbf{f}; \mathbf{y})$.

To conclude, we report the following important result, that ensures the convergence of the method when applied to (5.3). A convergence analysis of the SGP method for the general case of the minimization of differentiable functions on closed convex sets is carried out in [17].

Proposition 5.3 ([17]). *Let $\{\mathbf{f}^{(k)}\}$ be the sequence generated by applying algorithm SGP to problem* (5.3). *Every accumulation point $\mathbf{f}^*$ of $\{\mathbf{f}^{(k)}\}$ is a constrained stationary point, that is*

$$\nabla \Upsilon(\mathbf{f}^*; \mathbf{y})^T (\mathbf{f} - \mathbf{f}^*) \ge 0 \qquad \forall x \in \mathcal{C}.$$

If $\Upsilon(\mathbf{f}; \mathbf{y})$ is a convex function, then every accumulation point of $\{\mathbf{f}^{(k)}\}$ is a solution of problem (5.3).

5.2.1 Practical implementation of SGP to the ROI CT problem

When the objective function reads as in (2.18)-(2.22) with $p = 2$, the $(k+1)$-th iteration, with $k = 0, 1, 2, \ldots$, is given by:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \mu_k \left[\mathcal{P}_{\Omega_{\mathbf{f}}, D_k^{-1}} \Big(\mathbf{f}^{(k)} - \alpha_k D_k \nabla \mathbf{O}(\mathbf{f}^{(k)}) \Big) - \mathbf{f}^{(k)} \right] \tag{5.10}$$

where μ_k, α_k are the steplengths, D_k is the scaling matrix and $\mathcal{P}_{\Omega_{\mathbf{f}}}$ is the projector onto the feasible region $\Omega_{\mathbf{f}}$. In the remainder of this Chapter, we assume $p = 2$ with respect to the formulations (2.18)-(2.22) and (2.19)-(2.23).

The updating rules for the scaling matrix D_k depends on the formulation of the objective function and exploits the decomposition (5.6) of the gradient. We report in Table 5.1 the functions U and V for the different functionals that compose the implicit and the explicit objective function. Notice that the computation of the gradient takes advantage of the fact that we choose a shearlets tight frame, hence the equivalence $\mathbf{\Phi}^T \mathbf{\Phi} = \mathbb{1}_{N_\theta N_{\text{dtc}}}$ holds true.

In particular, when (2.18) is considered, we choose the following scheme:

$$D_k = \text{diag}\Bigg\{ \min \left\{ L, \max \left\{ \frac{1}{L}, \frac{\mathbf{f}^{(k)}}{\mathbf{W}^T(\mathbf{M} + \mu(\mathbb{1}_{N_\theta N_{\text{dtc}}} - \mathbf{M}))\mathbf{W}\mathbf{f}^{(k)}} \right\} \right\} \Bigg\},$$

where $V_{\text{rec}}(\mathbf{f}^{(k)}) = \mathbf{W}^T(\mathbf{M} + \mu(\mathbb{1}_{N_\theta N_{\text{dtc}}} - \mathbf{M}))\mathbf{W}\mathbf{f}^{(k)}$ is the positive part of the gradient decomposition (first plus third row in Table 5.1).
When, instead, the objective function reads as in (2.22), the updating rule for the scaling matrix becomes:

$$D_k = \text{diag}\Bigg\{ \min \left\{ L, \max \left\{ \frac{1}{L}, \frac{\mathbf{f}^{(k)}}{V_{\text{rec}}(\mathbf{f}^{(k)}) + \rho\, V_{\text{TV}}(\mathbf{f}^{(k)})} \right\} \right\} \Bigg\},$$

where $V_{\text{rec}}(\mathbf{f}^{(k)})$ is defined as above and

$$V_{\text{TV}}(\mathbf{f}) = \{2q'_\delta(\mathscr{D}^2_{j_{\text{row}}, j_{\text{col}}}) + q'_\delta(\mathscr{D}^2_{j_{\text{row}}, j_{\text{col}}-1}) + q'_\delta(\mathscr{D}^2_{j_{\text{row}}-1, j_{\text{col}}})\} f_{j_{\text{row}}, j_{\text{col}}}$$

is the positive part on the gradient decomposition of the TV term only, given that the negative part is:

$$\begin{aligned} U_{\text{TV}}(\mathbf{f}) = q'_\delta(\mathscr{D}^2_{j_{\text{row}}, j_{\text{col}}})(f_{j_{\text{row}}+1, j_{\text{col}}} + f_{j_{\text{row}}, j_{\text{col}}+1}) + q'_\delta(\mathscr{D}^2_{j_{\text{row}}, j_{\text{col}}-1}) f_{j_{\text{row}}, j_{\text{col}}-1} \\ + q'_\delta(\mathscr{D}^2_{j_{\text{row}}-1, j_{\text{col}}}) f_{j_{\text{row}}-1, j_{\text{col}}}. \end{aligned}$$

Notice that this term is calculated by exploiting the smooth discrete gradient operator as described by the relation (2.21) with $\delta \neq 0$.

	$V(\mathbf{f})$	$U(\mathbf{f})$
$\frac{1}{2}\|\mathbf{MWf}-\mathbf{y}_0\|_2^2$	$\mathbf{W}^T\mathbf{MWf}$	$\mathbf{W}^T\mathbf{My}_0$
$\frac{1}{2}\|(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})(\mathbf{Wf}-\mathbf{y})\|_2^2$	$\mathbf{W}^T(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})\mathbf{Wf}$	$\mathbf{W}^T(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})\mathbf{y}$
$\mu\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})\mathbf{Wf}+\mathbf{y}_0)\|_2^2$	$\mu\mathbf{W}^T(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})\mathbf{Wf}$	$\mu\mathbf{W}^T(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}}-\mathbf{M})\mathbf{y}_0$
$\mathrm{TV}_\delta(\mathbf{f})$	$V_{\mathrm{TV}}(\mathbf{f})$	$U_{\mathrm{TV}}(\mathbf{f})$

Table 5.1: Positive and negative part of the decomposition of the gradient (with respect to **f**) for the ROI CT functionals of the objective function.

Finally, when both the regularization term and the TV term are dropped in the implicit formulation, the updating rule for the scaling matrix is given by:

$$D_k = \mathrm{diag}\left\{\min\left\{L, \max\left\{\frac{1}{L}, \frac{\mathbf{f}^{(k)}}{\mathbf{W}^T\mathbf{MWf}^{(k)}}\right\}\right\}\right\},$$

where $V_{DM^{\mathrm{im}}}(\mathbf{f}^{(k)}) = \mathbf{W}^T\mathbf{MWf}^{(k)}$ is the positive part of the gradient decomposition for the data mismatch term (first row in Table 5.1). In all the formulae above reported, the notation is consistent with the one introduced in Chapter 2, L is a suitable threshold and the quotient of the two arrays is understood in the Hadamard sense, *i.e.*, component-wise.

As far as the steplength α_k concerns, the strategy proposed in [59] consists in an adaptive alternation between the values

$$\bar{\alpha}_k = \max\left\{\alpha_{\min}, \min\left\{\alpha_{\max}, \alpha_k^{\mathrm{BB1}}\right\}\right\}$$
$$\widetilde{\alpha}_k = \max\left\{\alpha_{\min}, \min\left\{\alpha_{\max}, \alpha_k^{\mathrm{BB2}}\right\}\right\}$$

where α_k^{BB1} and α_k^{BB2} are defined as in (5.7)-(5.8). The details of the steplength selection are given in Algorithm 2.

When the objective function reads as in (2.19)- (2.23), the SGP method needs to be setup for the case of the "enlarged" variable $\begin{pmatrix}\mathbf{f}\\ \mathbf{y}\end{pmatrix}$. This is accomplished by exploiting "enlarged" variables $\mathbf{s}^{(k-1)}$ and $\zeta^{(k-1)}$ to update the steplength α_k and by tacking a block-diagonal scaling matrix $D_k = \mathrm{diag}(D_k^{\mathbf{f}}, D_k^{\mathbf{y}})$. In details, analogously to the implicit case, the block $D_k^{\mathbf{f}}$ corresponding to the image **f** depends on the terms that appears in the objective function formulation, but the formulae

Algorithm 2 Steplength selection

```
if $k = 0$ then
    set $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$, $\tau_1 \in (0,1)$ and a nonnegative integer $\kappa_\alpha$;
else
    if $\mathbf{s}^{(k-1)^T} D_k^{-1} \zeta^{(k-1)} \le 0$ then
        $\alpha_k^{(1)} = \alpha_{\max}$;
    else
        $\alpha_k^{(1)} = \max\Big\{\alpha_{\min}, \min\Big\{\frac{\mathbf{s}^{(k-1)^T} D_k^{-1} D_k^{-1} \mathbf{s}^{(k-1)}}{\mathbf{s}^{(k-1)^T} D_k^{-1} \zeta^{(k-1)}}, \alpha_{\max}\Big\}\Big\}$;
    end if
    if $\mathbf{s}^{(k-1)^T} D_k^{-1} \zeta^{(k-1)} \le 0$ then
        $\alpha_k^{(2)} = \alpha_{\max}$;
    else
        $\alpha_k^{(2)} = \max\Big\{\alpha_{\min}, \min\Big\{\frac{\mathbf{s}^{(k-1)^T} D_k \zeta^{(k-1)}}{\zeta^{(k-1)^T} D_k D_k \zeta^{(k-1)}}, \alpha_{\max}\Big\}\Big\}$;
    end if
    if $\alpha_k^{(2)} / \alpha_k^{(1)} \le \tau_k$ then
        $\alpha_k = \min\Big\{\alpha_j^{(2)}, j = \max\{1, k - \kappa_\alpha\}, \dots, k\Big\}$;
        $\tau_{k+1} = \tau_k * 0.9$;
    else
        $\alpha_k = \alpha_k^{(1)}$;
        $\tau_{k+1} = \tau_k * 1.1$;
    end if
end if
```

are slightly different from the implicit case. When the objective function reduces to the data mismatch term, the updating rule for the scaling matrix is given by:

$$D_k^{\mathbf{f}} = \mathrm{diag}\Bigg\{\min\left\{L, \max\left\{\frac{1}{L}, \frac{\mathbf{f}^{(k)}}{\mathbf{W}^T\mathbf{W}\mathbf{f}^{(k)}}\right\}\right\}\Bigg\},$$

where $V_{DM^{\mathrm{ex}}}(\mathbf{f}^{(k)}) = \mathbf{W}^T\mathbf{W}\mathbf{f}^{(k)}$ is the positive part of the gradient decomposition for the explicit data mismatch term (first plus second row in Table 5.1). This applies also when the objective function reads as in (2.19), since the shearlets-based term now depends only on $\mathbf{y}$. When the objective function reads as in (2.23), the updating rule for the scaling matrix becomes:

$$D_k^{\mathbf{f}} = \mathrm{diag}\Bigg\{\min\left\{L, \max\left\{\frac{1}{L}, \frac{\mathbf{f}^{(k)}}{V_{DM^{\mathrm{ex}}}(\mathbf{f}^{(k)}) + \rho\, V_{\mathrm{TV}}(\mathbf{f}^{(k)})}\right\}\right\}\Bigg\},$$

where $V_{DM^{\mathrm{ex}}}(\mathbf{f}^{(k)})$ is defined as above and $V_{\mathrm{TV}}(\mathbf{f}^{(k)})$ is, again, the positive part on the TV gradient decomposition (fourth row in Table 5.1). Instead, for the block corresponding to the full sinogram $\mathbf{y}$, we take $D_k^{\mathbf{y}} = \mathbb{1}_{N_\theta N_{\text{dtc}}}$ for each $k = 0, 1, 2, \ldots$, and regardless of the formulation of the objective function. Hence, the $(k+1)$-th iteration, with $k = 0, 1, 2, \ldots$, is given by:

$$\begin{pmatrix} \mathbf{f}^{(k+1)} \\ \mathbf{y}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}^{(k)} \\ \mathbf{y}^{(k)} \end{pmatrix} + \mu_k \begin{pmatrix} \mathcal{P}_{\Omega_{\mathbf{f}}}\Big(\mathbf{f}^{(k)} - \alpha_k D_k^{\mathbf{f}} \nabla_{\mathbf{f}} \mathbf{O}(\mathbf{f}^{(k)}, \mathbf{y}^{(k)})\Big) - \mathbf{f}^{(k)} \\ \mathcal{P}_{\mathbf{y} \geq 0}\Big(\mathbf{y}^{(k)} - \alpha_k \nabla_{\mathbf{y}} \mathbf{O}(\mathbf{f}^{(k)}, \mathbf{y}^{(k)})\Big) - \mathbf{y}^{(k)} \end{pmatrix}.$$

The steplength selection is still performed as in Algorithm 2, but $\mathbf{s}^{(k-1)}$ and $\zeta^{(k-1)}$ are now the following "enlarged" variables:

$$\mathbf{s}^{(k-1)} = \begin{pmatrix} \mathbf{f}^{(k)} - \mathbf{f}^{(k-1)} \\ \mathbf{y}^{(k)} - \mathbf{y}^{(k-1)} \end{pmatrix}, \qquad \zeta^{(k-1)} = \begin{pmatrix} \nabla_{\mathbf{f}} \mathbf{O}(\mathbf{f}^{(k)}, \mathbf{y}^{(k)}) - \nabla_{\mathbf{f}} \mathbf{O}(\mathbf{f}^{(k-1)}, \mathbf{y}^{(k-1)}) \\ \nabla_{\mathbf{y}} \mathbf{O}(\mathbf{f}^{(k)}, \mathbf{y}^{(k)}) - \nabla_{\mathbf{y}} \mathbf{O}(\mathbf{f}^{(k-1)}, \mathbf{y}^{(k-1)}) \end{pmatrix}.$$

Notice that there exists also a "block" version of SGP, called cyclic scaled gradient projection (CBGP), whose convergence analysis is carried out in [134]. In this approach, each (outer) iteration consists of alternating an update of the object $\mathbf{f}$ and of the full sinogram $\mathbf{y}$ by means of fixed numbers of (inner) iterations of SGP, also allowing a different number of (inner) iterations in the two cases. We conjecture that this approach can be suitable to the ROI CT problem, but no numerical experiments has been performed, yet.

Finally, notice that since the feasible region in (2.22) and (2.23) is either a box or a non-negativity constraint, there is no need to solve a separable quadratic program. The projection step in both versions of SGP is performed by simply thresholding the values outside the feasible region.

5.3 Variable metric inexact line-search algorithm

In the previous section, a method belonging to the class of gradient projection methods, SGP, has been introduced. In 2014, the convergence results of SGP have been extended to the case of different metrics [15], in place of the scaled euclidean projection defined in (5.2).

An additional step is provided by the variable metric inexact line-search algorithm (VMILA), a proximal-gradient (or forward-backward) method, very recently proposed in [14]. VMILA is suitable for minimizing the sum of a differentiable, possibly nonconvex, function plus a convex, possibly non differentiable, function. The ROI CT nonsmooth formulation (2.18) with $p = 1$ clearly belongs to this problem class, making VMILA perfectly suitable.

Similarly to the SGP case, a general description of VMILA, whose main steps are stated in Algorithm 4, will be given and, then, some considerations for its practical implementation to the ROI CT problem will be discussed. First, we recall the fundamental definition of proximity operator, which will come into play in the following.

Definition 5.4 ([19])**.** Let $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$ be a convex function and $\mathcal{D}$ a symmetric positive definite matrix. The proximity operator associated to $\mathcal{F}$ in in the metric induced by $\mathcal{D}$ is defined to be:

$$\operatorname{prox}_{\mathcal{F}}^{\mathcal{D}}(x) = \operatorname*{argmin}_{z\in\mathbb{R}^n} \left\{ \mathcal{F}(z) + \frac{1}{2}\|z-x\|^2_{\mathcal{D}} \right\}, \qquad \forall x \in \mathbb{R}^n.$$

The general framework of VMILA applies to problems of the form

$$\min_{\mathbf{f}\in\mathbb{R}^n} \Gamma(\mathbf{f}) = \Gamma_0(\mathbf{f}) + \Gamma_1(\mathbf{f}), \tag{5.11}$$

where Γ_1 is a proper, convex, lower semicontinuous function and Γ_0 is smooth, *i.e.*, continuously differentiable, on an open subset $\mathcal{C}_0 \subset \mathbb{R}^n$ containing $\operatorname{dom}(\Gamma_1) = \{\mathbf{f} \in \mathbb{R}^n : \Gamma_1(\mathbf{f}) < +\infty\}$. Such a formulation includes also constrained problems over convex sets, by consedering the indicator function $\iota_{\mathcal{C}}$ of the feasible set $\mathcal{C}$ into Γ_1:

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ +\infty & \text{if } x \notin \mathcal{C} \end{cases}.$$

Notice that, by adopting the usual conventions

$$a + (+\infty) = +\infty, \quad a(+\infty) = +\infty, \quad \frac{1}{0} = +\infty \quad \text{and} \quad \frac{1}{+\infty} = 0,$$

with $a \in \mathbb{R} \cup \{+\infty\}$, the indicator function above defined and the characteristic function defined in (2.4) are related by the equations

$$1_{\mathcal{C}}(x) = \frac{1}{1+\iota_{\mathcal{C}}(x)} \qquad \text{and} \qquad \iota_{\mathcal{C}}(x) = (+\infty)(1 - 1_{\mathcal{C}}(x)).$$

Hence, one can freely convert between the two definitions, but the indicator function as defined above is better-suited to the methods of convex analysis.
The $(k+1)$-th iteration, with $k = 0, 1, 2, \ldots$, is given by:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \mu_k \left(\mathbf{v}^{(k)} - \mathbf{f}^{(k)}\right)$$

where the steplength μ_k is determined by means of a backtracking loop until a modified Armijo inequality is satisfied. The description of this rule is reported in Algorithm 3. The term $\mathbf{v}^{(k)} - \mathbf{f}^{(k)}$ is the descent direction, where

$$\begin{gathered} \mathbf{v}^{(k)} = \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n} h_{\sigma^{(k)}}(\mathbf{v}, \mathbf{f}^{(k)}) \qquad \text{with} \\ h_{\sigma^{(k)}}(\mathbf{v}, \mathbf{f}^{(k)}) = \nabla\Gamma_0(\mathbf{f}^{(k)})^T(\mathbf{v} - \mathbf{f}^{(k)}) + d_{\sigma^{(k)}}(\mathbf{v}, \mathbf{f}^{(k)}) + \Gamma_1(\mathbf{v}) - \Gamma_1(\mathbf{f}^{(k)}) \end{gathered} \tag{5.12}$$

Algorithm 3 Modified Armijo linesearch algorithm: computation of μ_k

```
Let $\{\mathbf{f}^{(k)}\}_{k\in\mathbb{N}}$, $\{\mathbf{v}^{(k)}\}_{k\in\mathbb{N}}$ be two sequences of points in $\mathcal{C}$, and $\{\sigma^{(k)}\}_{k\in\mathbb{N}}$ be a
sequence of parameters in $\Sigma$. Choose $\beta, \gamma \in (0,1)$.
for $k = 0, 1, 2, \dots$ do
   Step 1. Set $\mu_k = 1$ and $\mathbf{d}^{(k)} = \mathbf{v}^{(k)} - \mathbf{f}^{(k)}$.
   Step 2. if $\Gamma(\mathbf{f}^{(k)} + \mu_k \mathbf{d}^{(k)}) \leq \mathbf{f}^{(k)} + \beta\mu_k\Delta^{(k)}$, with $\Delta^{(k)} = h_{\sigma^{(k)}}(\mathbf{v}^{(k)}, \mathbf{f}^{(k)})$ then
             go to step 3.
           else
             set $\mu_k = \gamma\mu_k$ and go to Step 2.
   Step 3. end if
end for
```

and $d_\sigma : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ depends on the set of parameters $\sigma \in \Sigma$ and is a distance-like function, namely it is continuous, smooth, strongly convex and $d_\sigma(z, x) = 0$ if and only if $z = x$, where z, $x \in \mathbb{R}^n$.

When d_σ is the (scaled) euclidean distance, with $\sigma = (\alpha, D)$, where $\alpha > 0$ and D is a symmetric positive definite matrix:

$$d_\sigma(\mathbf{v}, \mathbf{f}) = \frac{1}{2\alpha}\|\mathbf{v} - \mathbf{f}\|_D^2, \tag{5.13}$$

an easy computation yields

$$\operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n} h_\sigma(\mathbf{v}, \mathbf{f}) = \operatorname{prox}^D_{\alpha\Gamma_1}(\mathbf{f} - \alpha D^{-1}\nabla\Gamma_0(\mathbf{f})) \tag{5.14}$$

where $\operatorname{prox}^D_{\alpha\Gamma_1}$ is the proximity or resolvent operator associated to the convex function $\alpha\Gamma_1$, in the metric induced by the matrix D. Indeed,

$$\begin{aligned}
\operatorname{prox}^D_{\alpha\Gamma_1}(\mathbf{f} - \alpha D^{-1}\nabla\Gamma_0(\mathbf{f})) &= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\Gamma_1(\mathbf{v}) + \frac{1}{2\alpha}\|\mathbf{v} - (\mathbf{f} - \alpha D^{-1}\nabla\Gamma_0(\mathbf{f}))\|_D^2\Big\} \\
&= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\Gamma_1(\mathbf{v}) + \frac{1}{2\alpha}\|(\mathbf{v} - \mathbf{f}) + \alpha D^{-1}\nabla\Gamma_0(\mathbf{f}))\|_D^2\Big\} \\
&= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\Gamma_1(\mathbf{v}) + \frac{1}{2\alpha}(\mathbf{v} - \mathbf{f})^T D(\mathbf{v} - \mathbf{f}) \\
&\qquad + \frac{\alpha^2}{2\alpha}\,(\nabla\Gamma_0(\mathbf{f}))^T D^{-1} D D^{-1}\nabla\Gamma_0(\mathbf{f}) \\
&\qquad + 2\,\frac{\alpha}{2\alpha}\,(\mathbf{v} - \mathbf{f})^T D D^{-1}\nabla\Gamma_0(\mathbf{f})\Big\} \\
&= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\Gamma_1(\mathbf{v}) + \frac{1}{2\alpha}(\mathbf{v} - \mathbf{f})^T D(\mathbf{v} - \mathbf{f}) + (\mathbf{v} - \mathbf{f})^T\nabla\Gamma_0(\mathbf{f})\Big\}
\end{aligned}$$

Algorithm 4 Variable Metric Inexact Line-search Algorithm

```
Choose $0 < \alpha_{\min} \le \alpha_{\max}$, set the parameters $\beta, \gamma \in (0,1)$, $\eta \in (0,1]$ and the starting point $\mathbf{f}^{(0)} \in \mathcal{C}$.
for $k = 0, 1, 2, \dots$ do
    Step 1. Choose the parameter $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ and the scaling matrix $D_k \in \mathcal{D}_L$;
    Step 2. Compute $\tilde{\mathbf{v}}^{(k)}$: compute a dual vector $v^{(l)} \in \mathbb{R}^{\tilde{m}}$ and the corresponding
            primal vector $\tilde{\mathbf{v}}^{(k,l)}$ such that (5.19) is satisfied, then set $\tilde{\mathbf{v}}^{(k)} = \tilde{\mathbf{v}}^{(k,l)}$.
    Step 3. Descent direction: $\mathbf{d}^{(k)} = \tilde{\mathbf{v}}^{(k)} - \mathbf{f}^{(k)}$;
    Step 4. Compute the steplength parameter $\mu_k$ with Algorithm 3;
    Step 5. Set $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \mu_k \mathbf{d}^{(k)}$.
end for
```

where, in the first equality, it has been used the general definition of the proximity operator associated to a convex function, in the metric induced by a symmetric positive definite matrix (see Definition 5.4). On the other hand, we have:

$$\begin{aligned}
\operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n} h_\sigma(\mathbf{v},\mathbf{f}) &= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\nabla\Gamma_0(\mathbf{f})^T(\mathbf{v}-\mathbf{f}) + \frac{1}{2\alpha}\,\|\mathbf{v}-\mathbf{f}\|_D^2 + \Gamma_1(\mathbf{v}) - \Gamma_1(\mathbf{f})\Big\} \\
&= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\nabla\Gamma_0(\mathbf{f})^T(\mathbf{v}-\mathbf{f}) + \frac{1}{2\alpha}\,(\mathbf{v}-\mathbf{f})^T D(\mathbf{v}-\mathbf{f}) + \Gamma_1(\mathbf{v}) - \Gamma_1(\mathbf{f})\Big\} \\
&= \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^n}\Big\{\Gamma_1(\mathbf{v}) + \frac{1}{2\alpha}(\mathbf{v}-\mathbf{f})^T D(\mathbf{v}-\mathbf{f}) + (\mathbf{v}-\mathbf{f})^T\nabla\Gamma_0(\mathbf{f})\Big\}
\end{aligned}$$

whence the equality (5.14).

Other examples of distance-like functions are the Bregman distances associated to a strongly convex function. In the following, it is assumed that d_σ is the scaled euclidean distance. Notice that α and D have exactly the same role as in SGP, *i.e.*, α is a steplength and D a scaling matrix. In particular, α_k shall be chosen in the closed interval $[\alpha_{\min}, \alpha_{\max}] \subset \mathbb{R}^+$, and the scaling matrix D_k in the compact set $\mathcal{D}_L$, where $\mathcal{D}_L$ is the set of the symmetric positive definite matrices D such that $\|D\| \le L$ and $\|D^{-1}\| \le L$, for a given threshold $L > 1$. Therefore, the same strategy used in SGP can be exploited (*i.e.*, Algorithm 2 for α and relation (5.9) for D).

It can be shown that $\mathbf{f}$ is a stationary point for (5.11) if and only if $\mathbf{f} = \operatorname{argmin}_{\mathbf{v}\in\mathbb{R}^n} h_\sigma(\mathbf{v},\mathbf{f})$. Hence, the operator $p : \mathcal{C}_0 \to \mathcal{C}$ associated to the function h_σ, defined by

$$p(\mathbf{f}; h_\sigma) = \operatorname*{argmin}_{z\in\mathbb{R}^n} h_\sigma(z,\mathbf{f}),$$

plays exactly the same role that the projection operator $\mathcal{P}_{\mathcal{C},D}$, defined in (5.2), plays in SGP. Moreover, the general SGP formulation is a particular case of the VMILA setup, which can be recovered by taking $\Gamma_1 = \iota_{\mathcal{C}}$, and $d_\sigma(\mathbf{v}, \mathbf{f})$ equal to the scaled euclidean distance, as defined in (5.13). Thus, VMILA can be understood as a nonsmooth version of SGP.

The following proposition resumes a convergence result of VMILA when applied to (5.11). A more general convergence analysis, even for the nonconvex case, also accounting for the convergence rate under the assumption of Lipschitz continuity for the smooth part of the objective function, is carried out in [14].

Proposition 5.5 ([14]). *Let $\{\mathbf{f}^{(k)}\}_{k\in\mathbb{N}}$, $\{\tilde{\mathbf{v}}^{(k)}\}_{k\in\mathbb{N}}$ be two sequences of points in $\mathcal{C}$, and $\{\sigma^{(k)}\}_{k\in\mathbb{N}} \subset \Sigma$. Assume that there exists a limit point $\mathbf{f}^*$ of $\{\mathbf{f}^{(k)}\}_{k\in\mathbb{N}}$ and let $K' \subseteq \mathbb{N}$ be a subset of indices such that $\lim_{k\in K', k\to\infty} \mathbf{f}^{(k)} = \mathbf{f} \in \mathcal{C}$. Assume that, for any $k \in \mathbb{N}$ we have*

$$\Gamma(\mathbf{f}^{(k+1)}) \leq \Gamma(\mathbf{f}^{(k)} + \mu_k \mathbf{d}^{(k)}), \qquad \mathbf{d}^{(k)} = \tilde{\mathbf{v}}^{(k)} - \mathbf{f}^{(k)},$$

where μ_k is computed by Algorithm 3, $\tilde{\mathbf{v}}^{(k)}$ satisfies $h_{\sigma^{(k)}}(\tilde{\mathbf{v}}^{(k)}, \mathbf{f}^{(k)}) < 0$ and there exists $K'' \subseteq K'$ such that

$$\lim_{k\in K'', k\to\infty} \left\{ h_{\sigma^{(k)}}(\tilde{\mathbf{v}}^{(k)}, \mathbf{f}^{(k)}) - h_{\sigma^{(k)}}(\mathbf{v}^{(k)}, \mathbf{f}^{(k)}) \right\} = 0, \qquad \text{with} \quad \mathbf{v}^{(k)} = p(\mathbf{f}^{(k)}; h_{\sigma^{(k)}}).$$

Then $\mathbf{f}^*$ is a stationary point for problem (5.11).

In many common situation, including ROI CT, the proximity operator arising from the minimization problem in (5.12) can not be computed in a closed-form and therefore one aims for an approximation. In [14] two types of admissible approximations are devised, but we will consider and introduce only the so-called η-approximation. This approximation is based on the following definition:

$$P_\eta(\mathbf{f}; h_\sigma) = \{\tilde{\mathbf{v}} \in \mathcal{C} : h_\sigma(\tilde{\mathbf{v}}, \mathbf{f}) \leq \eta\, h_\sigma(\mathbf{v}, \mathbf{f}), \text{ where } \mathbf{v} = p(\mathbf{f}; h_\sigma)\}$$

for some $\eta \in (0, 1]$. It can be shown that, if

$$\tilde{\mathbf{v}} \in P_\eta(\mathbf{f}; h_\sigma),$$

then $h_\sigma(\tilde{\mathbf{v}}, \mathbf{f}) \leq 0$ and $h_\sigma(\tilde{\mathbf{v}}, \mathbf{f}) = 0$ if and only if $h_\sigma(\mathbf{v}, \mathbf{f}) = 0$, which implies $\tilde{\mathbf{v}} = \mathbf{v}$. Under the condition $\tilde{\mathbf{v}}^{(k)} \in P_\eta(\mathbf{f}^{(k)}; h_\sigma)$, the following convergence result holds true.

Proposition 5.6 ([14]). *Let $\eta \in (0, 1]$, $\{\sigma^{(k)}\}_{k\in\mathbb{N}} \subset \Sigma$ and $\{\mathbf{f}^{(k)}\}_{k\in\mathbb{N}} \subset \mathcal{C}$ satisfying the following condition:*

$$\Gamma(\mathbf{f}^{(k+1)}) \leq \Gamma(\mathbf{f}^{(k)} + \mu_k \mathbf{d}^{(k)}), \qquad \text{with} \quad \mathbf{d}^{(k)} = \tilde{\mathbf{v}}^{(k)} - \mathbf{f}^{(k)}$$

where μ_k is computed by Algorithm 3, with $\tilde{\mathbf{v}}^{(k)} \in P_\eta(\mathbf{f}^{(k)}; h_\sigma)$. Then, either for some k the iterate $\mathbf{f}^{(k)}$ is stationary for problem (5.11)*, or any limit point $\mathbf{f}^*$ of $\{\mathbf{f}^{(k)}\}_{k\in\mathbb{N}}$ is stationary for problem* (5.11)*.*

The η-approximation can be practically computed when

$$\Gamma_1(\mathbf{f}) = g(\mathbf{B}\mathbf{f}), \tag{5.15}$$

where $\mathbf{B} \in \mathbb{R}^{\widetilde{m}\times n}$ and $g : \mathbb{R}^{\widetilde{m}} \to \mathbb{R}$ is a proper, convex, lower semicontinuous function with an easy-to-compute resolvent operator. In this case, VMILA consists in a double loop method, where the inner loop, designed to compute an approximation of the proximity operator, is devised with a suitable stopping criterion.
In the η-approximation case, the minimization problem in (5.12) is written in an equivalent dual form, namely:

$$\min_{\mathbf{v}\in\mathbb{R}^n} h_{\sigma^{(k)}}(\mathbf{v}, \mathbf{f}^{(k)}) = \max_{\upsilon\in\mathbb{R}^{\widetilde{m}}} H_{\sigma^{(k)}}(\upsilon, \mathbf{f}^{(k)})$$

with

$$\begin{aligned} H_{\sigma^{(k)}}(\upsilon, \mathbf{f}^{(k)}) = &-\frac{1}{2\alpha_k}\left\|\alpha_k D_k^{-1}\mathbf{B}^T\upsilon - \mathbf{z}^{(k)}\right\|_{D_k}^2 - g^*(\upsilon) - \Gamma_1(\mathbf{f}^{(k)}) \\ &- \frac{\alpha_k}{2}\left\|\nabla\Gamma_0(\mathbf{f}^{(k)})\right\|_{D_k^{-1}}^2 + \frac{1}{2\alpha_k}\|\mathbf{z}^{(k)}\|_{D_k}^2 \end{aligned} \tag{5.16}$$

where $\mathbf{z}^{(k)} = \mathbf{f}^{(k)} - \alpha_k D_k^{-1}\nabla\Gamma_0(\mathbf{f}^{(k)})$, $\upsilon \in \mathbb{R}^{\widetilde{m}}$ is the dual variable whose primal is $\mathbf{v} \in \mathbb{R}^n$ and g^* is the conjugate function of g defined by [19]:

$$g^*(\xi) = \sup_{x\in\mathbb{R}^{\widetilde{m}}}\left\{x^T\xi - g(x)\right\} \qquad \forall\xi \in \mathbb{R}^{\widetilde{m}}. \tag{5.17}$$

Indeed, relation (5.17), that is $g(\mathbf{B}\mathbf{v}) = \max_{\upsilon\in\mathbb{R}^{\widetilde{m}}} \upsilon^T\mathbf{B}\mathbf{v} - g^*(\upsilon)$, yields:

$$\begin{aligned} \min_{\mathbf{v}\in\mathbb{R}^n} h_{\sigma^{(k)}}(\mathbf{v}, \mathbf{f}^{(k)}) &= \min_{\mathbf{v}\in\mathbb{R}^n}\left\{(\nabla\Gamma_0(\mathbf{f}^{(k)}))^T(\mathbf{v} - \mathbf{f}^{(k)}) + \frac{1}{2\alpha_k}\|\mathbf{v} - \mathbf{f}^{(k)}\|_{D_k}^2 + \Gamma_1(\mathbf{v}) - \Gamma_1(\mathbf{f}^{(k)})\right\} \\ &= \min_{\mathbf{v}\in\mathbb{R}^n}\left\{(\nabla\Gamma_0(\mathbf{f}^{(k)}))^T(\mathbf{v} - \mathbf{f}^{(k)}) + \frac{1}{2\alpha_k}\|\mathbf{v} - \mathbf{f}^{(k)}\|_{D_k}^2 + g(\mathbf{B}\mathbf{v}) - \Gamma_1(\mathbf{f}^{(k)})\right\} \\ &= \min_{\mathbf{v}\in\mathbb{R}^n}\max_{\upsilon\in\mathbb{R}^{\widetilde{m}}}\left\{\frac{1}{2\alpha_k}\|\mathbf{v} - \mathbf{f}^{(k)}\|_{D_k}^2 + (\nabla\Gamma_0(\mathbf{f}^{(k)}))^T(\mathbf{v} - \mathbf{f}^{(k)}) - \Gamma_1(\mathbf{f}^{(k)}) \right. \\ &\qquad\qquad \left. + \mathbf{v}^T\mathbf{B}^T\upsilon - g^*(\upsilon)\right\} \end{aligned}$$

By adding and subtracting the term $\frac{1}{2\alpha_k}\|\alpha_k\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k^{-1}}$, we get:

$$\begin{aligned}
\min_{\mathbf{v}\in\mathbb{R}^n} h_{\sigma^{(k)}}(\mathbf{v},\mathbf{f}^{(k)}) = \min_{\mathbf{v}\in\mathbb{R}^n}\max_{\upsilon\in\mathbb{R}^{\tilde{m}}} & \Big\{\mathbf{v}^T\mathbf{B}^T\upsilon - g^*(\upsilon) - \Gamma_1(\mathbf{f}^{(k)}) + \frac{1}{2\alpha_k}\,\|\mathbf{v}-\mathbf{f}^{(k)}\|^2_{D_k} \\
& + (\nabla\Gamma_0(\mathbf{f}^{(k)}))^T D_k D_k^{-1}(\mathbf{v}-\mathbf{f}^{(k)}) + \frac{1}{2\alpha_k}\|\alpha_k\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k^{-1}} \\
& - \frac{1}{2\alpha_k}\|\alpha_k\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k^{-1}}\Big\} \\
= \min_{\mathbf{v}\in\mathbb{R}^n}\max_{\upsilon\in\mathbb{R}^{\tilde{m}}} & \Big\{\mathbf{v}^T\mathbf{B}^T\upsilon - g^*(\upsilon) - \Gamma_1(\mathbf{f}^{(k)}) - \frac{\alpha_k}{2}\|\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k^{-1}} \\
& + \frac{1}{2\alpha_k}\Big[\|\mathbf{v}\|^2_{D_k} - 2\mathbf{v}^T D_k\big(\mathbf{f}^{(k)} - \alpha_k D_k^{-1}\nabla\Gamma_0(\mathbf{f}^{(k)})\big)\Big] \\
& + \frac{1}{2\alpha_k}\,\|\mathbf{f}^{(k)} - \alpha_k D_k^{-1}\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k}\Big\} \\
= \min_{\mathbf{v}\in\mathbb{R}^n}\max_{\upsilon\in\mathbb{R}^{\tilde{m}}} & \Big\{\mathbf{v}^T\mathbf{B}^T\upsilon - g^*(\upsilon) - \Gamma_1(\mathbf{f}^{(k)}) - \frac{\alpha_k}{2}\|\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k^{-1}} \\
& + \frac{1}{2\alpha_k}\Big[\|\mathbf{v}\|^2_{D_k} - 2\mathbf{v}^T D_k\mathbf{z}^{(k)}\Big] + \frac{1}{2\alpha_k}\,\big\|\mathbf{z}^{(k)}\big\|^2_{D_k}\Big\}
\end{aligned}$$

where we used the equality $\mathbf{z}^{(k)} = \mathbf{f}^{(k)} - \alpha_k D_k^{-1}\nabla\Gamma_0(\mathbf{f}^{(k)})$. By minimizing with respect to $\mathbf{v}\in\mathbb{R}^n$ (*i.e.*, by posing the gradient with respect to $\mathbf{v}\in\mathbb{R}^n$ equal to 0), we get:

$$\mathbf{v} = \mathbf{z}^{(k)} - \alpha_k D_k^{-1}\mathbf{B}^T\upsilon \tag{5.18}$$

that expresses the relation between the primal and the dual variable. Observe that:

$$-\frac{1}{2\alpha_k}\left\|\alpha_k D_k^{-1}\mathbf{B}^T\upsilon - \mathbf{z}^{(k)}\right\|^2_{D_k} = -\frac{\alpha_k}{2}\left\|\mathbf{B}^T\upsilon\right\|^2_{D_k^{-1}} - \frac{1}{2\alpha_k}\left\|\mathbf{z}^{(k)}\right\|^2_{D_k} + \upsilon^T\mathbf{B}\mathbf{z}^{(k)},$$

By using (5.18) and the previous relation we are led to:

$$\begin{aligned}
\min_{\mathbf{v}\in\mathbb{R}^n} h_{\sigma^{(k)}}(\mathbf{v},\mathbf{f}^{(k)}) = \max_{\upsilon\in\mathbb{R}^{\tilde{m}}} & \Big\{(\mathbf{z}^{(k)} - \alpha_k D_k^{-1}\mathbf{B}^T\upsilon)^T\mathbf{B}^T\upsilon - g^*(\upsilon) - \Gamma_1(\mathbf{f}^{(k)}) + \frac{1}{2\alpha_k}\,\big\|\mathbf{z}^{(k)}\big\|^2_{D_k} \\
& - \frac{\alpha_k}{2}\,\|\nabla\Gamma_0(\mathbf{f}^{(k)})\|^2_{D_k^{-1}} + \frac{1}{2\alpha_k}\|\mathbf{z}^{(k)} - \alpha_k D_k^{-1}\mathbf{B}^T\upsilon\|^2_{D_k} \\
& - \frac{1}{\alpha_k}\Big[(\mathbf{z}^{(k)} - \alpha_k D_k^{-1}\mathbf{B}^T\upsilon)^T D_k\mathbf{z}^{(k)}\Big]\Big\} \\
= \max_{\upsilon\in\mathbb{R}^{\tilde{m}}} & \Big\{-\frac{1}{2\alpha_k}\left\|\alpha_k D_k^{-1}\mathbf{B}^T\upsilon - \mathbf{z}^{(k)}\right\|^2_{D_k} - g^*(\upsilon) - \Gamma_1(\mathbf{f}^{(k)}) \\
& - \frac{\alpha_k}{2}\left\|\nabla\Gamma_0(\mathbf{f}^{(k)})\right\|^2_{D_k^{-1}} + \frac{1}{2\alpha_k}\|\mathbf{z}^{(k)}\|^2_{D_k}\Big\} \\
= \max_{\upsilon\in\mathbb{R}^{\tilde{m}}} & \; H_{\sigma^{(k)}}(\upsilon,\mathbf{f}^{(k)})
\end{aligned}$$

Observe that formulation (5.15) includes also the case where $\Gamma_1(\mathbf{f})$ is defined as

$$\Gamma_1(\mathbf{f}) = \sum_{j=1}^{\tilde{q}} g_j(\mathbf{B}_j\mathbf{f})$$

where $\mathbf{B}_j \in \mathbb{R}^{\tilde{m}_j \times n}$ and $g_j : \mathbb{R}^{\tilde{m}_j} \to \mathbb{R}$. In this case, formulation (5.15) is recovered by setting

$$\mathbf{B} = [\mathbf{B}_1^T \; \mathbf{B}_2^T \; \dots \; \mathbf{B}_{\tilde{q}}^T]^T \quad \in \; \mathbb{R}^{\tilde{m} \times n}$$

with $\sum_{j=1}^{\tilde{q}} \tilde{m}_j = \tilde{m}$. By analogy, the dual variable υ is partitioned ad $\upsilon = [\upsilon_1^T \; \upsilon_2^T \; \dots \; \upsilon_{\tilde{q}}^T]$, where $\upsilon_j \in \mathbb{R}^{\tilde{m}_j}$ and $g^*(\upsilon) = \sum_{j=1}^{\tilde{q}} g_i^*(\upsilon_i)$.

An approximation $\tilde{\mathbf{v}}^{(k)} \in \mathbb{R}^n$ of $\mathbf{v}^{(k)} \in \mathbb{R}^n$ can be computed by applying a suitable method for the solution of optimization problems to the dual problem

$$\max_{\upsilon \in \mathbb{R}^{\tilde{m}}} H_{\sigma^{(k)}}(\upsilon, \mathbf{f}^{(k)}),$$

generating a sequence $\{\upsilon^{(l)}\}_{l \in \mathbb{N}}$ such that $H_{\sigma^{(k)}}(\upsilon^{(l)}, \mathbf{f}^{(k)})$ converges to the maximum $\tilde{\mathbf{v}}^{(k,l)}$ of the dual function $H_{\sigma^{(k)}}(\cdot, \mathbf{f}^{(k)})$. For instance, if the dual problem is differentiable, as in the ROI CT case, SGP is a suitable choice. By exploiting (5.18), namely

$$\tilde{\mathbf{v}}^{(k,l)} = \mathbf{z}^{(k)} - \alpha_k D_k^{-1} \mathbf{B}^T \upsilon^{(l)},$$

we set $\tilde{\mathbf{v}}^{(k)} = \tilde{\mathbf{v}}^{(k,l)}$ (*i.e.*, we stop the dual iterations) when the following inequality is satisfied:

$$h_{\sigma^{(k)}}(\tilde{\mathbf{v}}^{(k,l)}, \mathbf{f}^{(k)}) \le \eta \; H_{\sigma^{(k)}}(\upsilon^{(l)}, \mathbf{f}^{(k)}), \tag{5.19}$$

for a given $\eta \in (0, 1]$. Notice that any point $\tilde{\mathbf{v}}^{(k,l)}$ which satisfies (5.19) belongs to the domain $\mathcal{C}$ of $h_{\sigma^{(k)}}(\cdot, \mathbf{f}^{(k)})$. However, a very large number of inner iterations l may be required to satisfy (5.19) and the primal sequence points $\tilde{\mathbf{v}}^{(k,l)}$ might be feasible only in the limit. For this reason, it may be convenient to consider also the sequence generated by projecting $\tilde{\mathbf{v}}^{(k,l)}$ onto the domain $\mathcal{C}$ of $h_{\sigma^{(k)}}(\cdot, \mathbf{f}^{(k)})$, *i.e.*, the sequence $P_{\mathcal{C}}(\tilde{\mathbf{v}}^{(k,l)})$, where $P_{\mathcal{C}}$ denotes as usual the euclidian projection onto the set $\mathcal{C}$. Hence, to preserve feasibility, we can set $\tilde{\mathbf{v}}^{(k)} = P_{\mathcal{C}}(\tilde{\mathbf{v}}^{(k,l)})$ if at some iteration l of the inner loop the following inequality is satisfied:

$$h_{\sigma^{(k)}}(P_{\mathcal{C}}(\tilde{\mathbf{v}}^{(k,l)}), \, \mathbf{f}^{(k)}) \le \eta \; H_{\sigma^{(k)}}(\upsilon^{(l)}, \mathbf{f}^{(k)}).$$

Indeed, when $\tilde{\mathbf{v}}^{(k,l)}$ converges to $\mathbf{v}^{(k)}$ as l diverges, also the sequence $P_{\mathcal{C}}(\tilde{\mathbf{v}}^{(k,l)})$ converges to $\mathbf{v}^{(k)}$, whence the well-posedness of the above stopping criterion.

Finally, observe that, even if the presence of two nested loops might increase the computational cost, in many practical case the inner loop requires only a few number of iteration to satisfy (5.19). Also, the possibility to use at each iterate an approximation of the proximity operator makes the method well suited for the solution of a wide variety of structured problems.

5.3.1 Practical implementation of VMILA to the ROI CT problem

When the objective function reads as in (2.18), with $p = 1$, it is clear that VMILA can be applied by posing:

$$\Gamma_0(\mathbf{f}) = \frac{1}{2}\Big\|\mathbf{M}\mathbf{W}\mathbf{f} - \mathbf{y}_0\Big\|_2^2 \quad \text{and} \quad \Gamma_1(\mathbf{f}) = \mu\Big\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{W}\mathbf{f} + \mathbf{y}_0)\Big\|_1 + \iota_{\Omega_\mathbf{f}}(\mathbf{f}),$$

where $\iota_{\Omega_\mathbf{f}}$ is the indicator function of the feasible region $\Omega_\mathbf{f}$. In the remainder of this chapter, we assume $p = 1$ with the respect to formulations (2.18)-(2.22). The $(k+1)$-th iteration, with $k = 0, 1, 2, \ldots$, is given by:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \mu_k\left[\mathrm{prox}_{\alpha_k\Gamma_1}^{D_k^{-1}}(\mathbf{f}^{(k)} - \alpha_k D_k \nabla\Gamma_0(\mathbf{f}^{(k)})) - \mathbf{f}^{(k)}\right]$$

where μ_k, α_k are the steplengths and D_k is the scaling matrix. The updating rule for α_k is the adaptive steplength selection given by Algorithm 2, while, for each $k = 0, 1, 2, \ldots$, we take

$$D_k = \mathrm{diag}\left\{\min\left\{L, \max\left\{\frac{1}{L}, \frac{\mathbf{f}^{(k)}}{\mathbf{W}^T\mathbf{M}\mathbf{W}\mathbf{f}^{(k)}}\right\}\right\}\right\},$$

where $V_{DM}(\mathbf{f}^{(k)}) = \mathbf{W}^T\mathbf{M}\mathbf{W}\mathbf{f}^{(k)}$ is the positive part of the gradient decomposition for the smooth term of the the objective function (first row in Table 5.1). Here, the proximity operator is computed by means of the η-approximation, by applying SGP with the scaling matrix equal to the identity matrix and the steplength adaptively selected as described in Algorithm 2. Notice that, in this case, the steplengths α^{BB1} and α^{BB2} reduce to the standard BB rules. The matrix $\mathbf{B}$ is expressed by means of blocks, *i.e.*, $\mathbf{B} = [\mathbf{B}_1^T\ \mathbf{B}_2^T]^T$. The first block $\mathbf{B}_1 \in \mathbb{R}^{N_\theta N_{\mathrm{dtc}} \times N^2}$ corresponds to the 1-norm term while the second one, $\mathbf{B}_2 \in \mathbb{R}^{N^2 \times N^2}$ accounts for the indicator function. In details,

$$\mathbf{B} = \begin{pmatrix} \mathbf{\Phi}(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{W} \\ \mathbb{1}_{N^2} \end{pmatrix} \quad \in \mathbb{R}^{(N_\theta N_{\mathrm{dtc}} + N^2) \times N^2},$$

where, according to the notation introduced in Chapter 2, $\mathbf{\Phi}$ is the shearlet matrix, $\mathbf{M}$ is the mask and $\mathbf{W}$ is the forward projection matrix. Moreover, the dual variable $v = [v_1^T\ v_2^T]^T$ belongs to $\mathbb{R}^{N_\theta N_{\mathrm{dtc}} + N^2}$, with $v_1 \in \mathbb{R}^{N_\theta N_{\mathrm{dtc}}}$ and $v_2 \in \mathbb{R}^{N^2}$. Next, $g^* = g_1^*(v_1) + g_2^*(v_2)$, where g_1^* is the indicator function of the set $B_\infty(0,1) \times \ldots \times B_\infty(0,1)$, being $B_\infty(0,1) \subset \mathbb{R}$ the ball in the infinity-norm centered in 0 with radius 1, and g_2^* is the indicator function of the set $\mathbb{R}_-^{N^2}$. Notice that $\mathbb{R}_-^{N^2}$ accounts only for the projection onto the nonnegative portion of the plane, by exploiting the dual sequence $v^{(l)}$ with l denoting the inner loop iteration. To preserve the feasibility of the box constraint $\Omega_{\mathbf{f}}$, we consider the sequence generated by projecting the corresponding primal sequence $\tilde{\mathbf{v}}^{(k,l)}$ onto $\Omega_{\mathbf{f}}$.

As in the smooth case, a total variation term can be added to formulation (2.18), yielding (2.22). This can be done in two ways:

- Add smooth TV. In this case, since we are dealing with the smoothed version of TV, we simply add it to Γ_0:

  $$\begin{aligned}\Gamma_0(\mathbf{f}) &= \frac{1}{2}\left\|\mathbf{MWf} - \mathbf{y}_0\right\|_2^2 + \rho\,\mathrm{TV}_\delta(\mathbf{f})\\ \Gamma_1(\mathbf{f}) &= \mu\left\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{Wf} + \mathbf{y}_0)\right\|_1 + \iota_{\Omega_{\mathbf{f}}}(\mathbf{f}).\end{aligned}$$

  The gradient of the smoothed TV term is computed by exploiting the smooth discrete gradient operator as defined in (2.21), exactly as in SGP. In this case, for each $k = 0, 1, 2, \ldots$, the scaling matrix is given by:

  $$D_k = \mathrm{diag}\left\{\min\left\{L, \max\left\{\frac{1}{L}, \frac{\mathbf{f}^{(k)}}{\mathbf{W}^T\mathbf{MWf}^{(k)} + \rho\,V_{\mathrm{TV}}(\mathbf{f}^{(k)})}\right\}\right\}\right\},$$

  where $V_{\mathrm{TV}}(\mathbf{f}^{(k)})$ is the positive part on the gradient decomposition of the TV term only.

- Add nonsmooth TV. In this case, the TV term must be added to Γ_1, yielding:

  $$\begin{aligned}\Gamma_0(\mathbf{f}) &= \frac{1}{2}\left\|\mathbf{MWf} - \mathbf{y}_0\right\|_2^2\\ \Gamma_1(\mathbf{f}) &= \mu\left\|\mathbf{\Phi}((\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{Wf} + \mathbf{y}_0)\right\|_1 + \iota_{\Omega_{\mathbf{f}}}(\mathbf{f}) + \rho\,\mathrm{TV}(\mathbf{f}).\end{aligned}$$

  As a consequence, the matrix $\mathbf{B}$ “gains” another block corresponding to the TV term. By exploiting the discrete gradient operator $\nabla_i \in \mathbb{R}^{2\times N^2}$ at the pixel i as defined in (2.20), $\mathbf{B} = [\mathbf{B}_1^T\ \mathbf{B}_2^T\ \mathbf{B}_3^T]^T$ becomes

  $$\mathbf{B} = \begin{pmatrix}\mathbf{\Phi}(\mathbb{1}_{N_\theta N_{\mathrm{dtc}}} - \mathbf{M})\mathbf{W}\\ \mathbb{1}_{N^2}\\ \nabla\end{pmatrix} \qquad \in \mathbb{R}^{(N_\theta N_{\mathrm{dtc}}+N^2+2N^2)\times N^2},$$

  where $\mathbf{B}_1 \in \mathbb{R}^{N_\theta N_{\mathrm{dtc}}\times N^2}$, $\mathbf{B}_2 \in \mathbb{R}^{N^2\times N^2}$, and $\mathbf{B}_3 = \nabla = (\nabla_1^T \ldots \nabla_{N^2}^T)^T \in \mathbb{R}^{2N^2\times N^2}$. The dual variable $\upsilon = [\upsilon_1^T\ \upsilon_2^T\ \upsilon_3^T]^T$ belongs to $\mathbb{R}^{N_\theta N_{\mathrm{dtc}}+N^2+2N^2}$ and $g^* = g_1^*(\upsilon_1) + g_2^*(\upsilon_2) + g_3^*(\upsilon_3)$, where g_1^* is the indicator function of the set $B_\infty(0,1)\times\ldots\times B_\infty(0,1)$, g_2^* is the indicator function of $\mathbb{R}_-^{N^2}$ and, finally, g_3^* is the indicator function of the set $B_2(0,\rho)\times\ldots\times B_2(0,\rho)$, being $B_2(0,\rho)\subset\mathbb{R}^2$ the 2D euclidean ball centered in 0 with radius ρ.

The nonsmooth approach for total variation is still a work in progress, due to the increased numerical difficulties and no numerical experiment has been performed, yet. A version of VMILA for the "enlarged" variable $\begin{pmatrix} \mathbf{f} \\ \mathbf{y} \end{pmatrix}$ is also a work in progress: the idea is to implement a block-cyclic version of VMILA, similar to CBGP [134].

Chapter 6

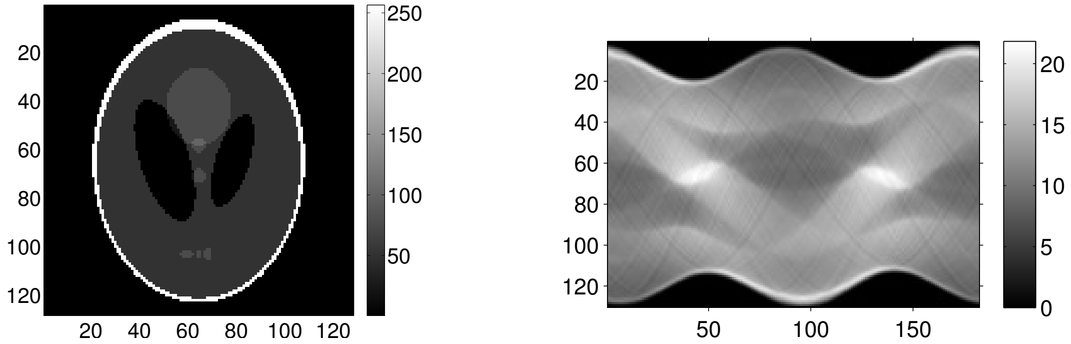
Numerical experiments

The goal of this Chapter is to validate by syntectic simulated data the *ad hoc* algorithms developed and analyzed in Chapter 5. To this end, we present extensive numerical results for both smooth and nonsmooth formulation of the ROI CT reconstruction problem in the framework of 2D fan-beam geometry. Then, as a benchmark comparison, we consider a traditional technique, that is the least squares conjugate gradient (LSCG) method [80]. Discussions and remarks on the results obtained conclude the Chapter.

6.1 Data simulation

To demonstrate and validate our approach, we use a synthetic data set known as *modified Shepp-Logan phantom* (see Figure 6.1a), sized $N \times N$ pixels with $N = 128$. It is available, for instance, in the Matlab Image Processing toolbox. All phantom data are simulated using the geometry of a micro-CT scanner used for real measurements (see Figure 6.1b). The 2D fan-beam data are simulated over $N_\theta = 182$ uniformly spaced angles over 2π. The detector consists of $N_{\mathrm{dtc}} = 130$ elements with a pixel pitch of $0.8\,mm$. The distance between tube and detector is set to $291.20\,mm$ and the radius of rotation is $115.84\,mm$. The detector is offset by 1.5 pixels. The matrix $\mathbf{W}$ that represents a discretization of the 2D fan-beam geometry, according to the above specifications, is implemented by using the efficient vectorized implementation of the distance-driven technique presented in Chapter 3.

The results reported in the following covers concentric ROI disks that, according to the hypothesis of the interior tomography problem, are fully inside the object being imaged, with decreasing ROI radius $\mathbf{r}_{\mathrm{ROI}}$, placed off-center with respect to the FOV, namely, in pixels, $\mathbf{p}_{\mathrm{ROI}} = (64, 80)$. Here, the Shepp-Logan phantom is assumed to be placed in the first quadrant of the Cartesian coordinate system with



(a) Modified Shepp-Logan phantom.

(b) Corresponding sinogram.

Figure 6.1: The modified Shepp-Logan phantom, sized $N \times N$ pixels with $N = 128$, and the corresponding noise free sinogram obtained according to a 2D fan-beam geometry of a micro-CT scanner.

the image lower left corner at the origin. Also, a larger ROI radius that exceeds the object along one coordinate axis, *i.e.*, $\mathbf{r}_{\mathrm{ROI}} = 0.5N$, has been considered for comparison. We report in Figure 6.2 the ROI with decreasing radii considered for the numerical simulation: the ROI is identified with a dashed white circle. Truncated projection data were obtained by discarding the samples outside the ROI projection $\mathcal{P}(S)$ (see Figure 6.3). This, indeed, corresponds to a ROI disk in the image domain, as outlined in Section 2.2.1. Here, we assume that the noise that corrupts projection data is described by a white Gaussian process (see Example 5.1), with zero mean and a 5% variance (see Figure 6.4).

All numerical tests were compared against two state of the art figures of merit, namely the peak-signal-to-noise ratio (PSNR) and the relative error. We recall that the PSNR, measured in dB, is defined as follows:

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{\mathrm{MPV}^2}{\mathbf{e}_{\mathrm{MSE}}}\right)$$

where MPV is the maximum pixel value and $\mathbf{e}_{\mathrm{MSE}}$ is the mean squared error. We stress that both PSNR and relative error are evaluated inside the ROI S only. This is consistent with the motivation of ROI CT, that aims to recover the image inside the ROI only.

Notice that the measure of the relative error evaluated inside the ROI has been used as a suitable criterion for stopping the iterations of the proposed algorithms, namely, we stop the iterations when the difference between the values of the ROI relative error of two consecutive iterations is smaller that a certain threshold τ_{it}. Roughly, this means that the reconstructed solution does not change appreciably after the threshold has been reached. A rule of thumb for choosing τ_{it} does not

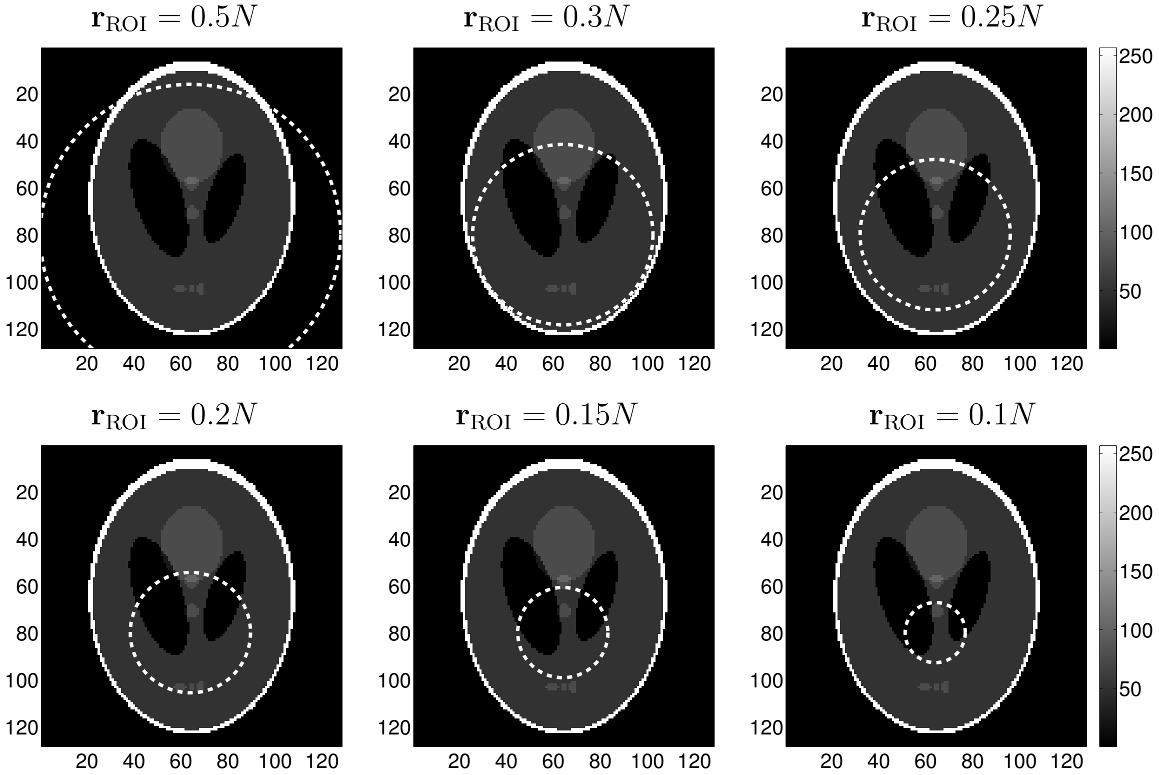

$\mathbf{r}_{\text{ROI}} = 0.5N$
$\mathbf{r}_{\text{ROI}} = 0.3N$
$\mathbf{r}_{\text{ROI}} = 0.25N$
$\mathbf{r}_{\text{ROI}} = 0.2N$
$\mathbf{r}_{\text{ROI}} = 0.15N$
$\mathbf{r}_{\text{ROI}} = 0.1N$
20
40
60
80
100
120
250
200
150
100
50

Figure 6.2: Decreasing ROI disk, identified with the dashed white circle, superimposed to the modified Shepp-Logan phantom. The aim is to recover the image inside the ROI only.

exist, and its choice follows from our experience. In this thesis, we used $\tau_{\text{it}} = 10^{-7}$ for obtaining satisfactory reconstructions with SGP, and $\tau_{\text{it}} = 10^{-4}$ as a reliable value for VMILA. The different threshold between the two approaches only aims at obtaining more accurate reconstructions. Also, VMILA requires to set an additional parameter $\eta \in (0, 1]$, from Equation (5.19), that controls the stopping criterion for the inner loop: this clearly needs to be tuned with the threshold for the outer loop. In this thesis, we set $\eta = 10^{-5}$. Further, a maximum number of 200 inner iterations is also imposed. This is just a precaution, since, in the results reported in Section 6.3, the criterion (5.19) is always met before reaching the prefixed maximum number of iterations. We set a precautionary maximum number of 7000 (outer) iterations also for SGP and VMILA, but in the results reported throughout the Chapter it was never attained (*i.e.*, iterations are stopped on the threshold τ_{it} criterion).

The choice of the regularization parameters μ, ρ is also crucial for a good restoration result. As we already stressed in Chapter 5, a general analytical method

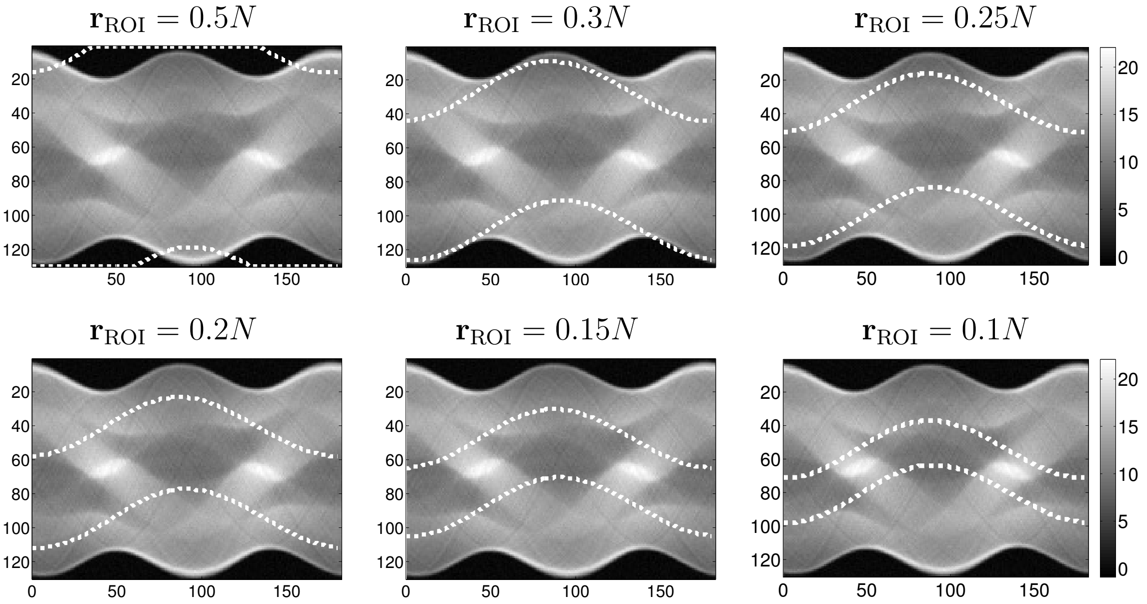
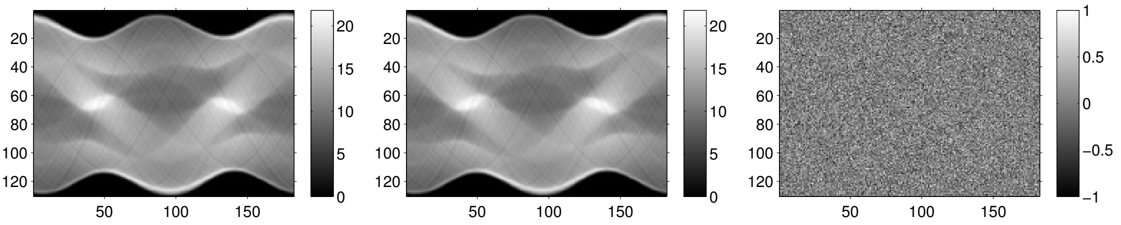


Figure 6.3: Simulated sinogram corrupted by white Gauss noise, with zero mean and a 5% variance. The two sinusoidal dashed white lines identify the truncated sinogram, according to the above specified R_{ROI}.

that applies to all the different formulations of the objective function (explicit and implicit, smooth and nonsmooth), is not available, and it is even more difficult when there is more than one parameter to assess. Since the original object was available, we choose the values of μ and ρ providing the minimum ROI relative error (and maximum ROI PSNR), by sampling the values 10^{ℓ}, with $\ell = -4, -3, \ldots, 4$, for the regularization parameter μ, and the values 10^{-2}, 10^{-1}, 1, 10 for the TV parameter ρ. The TV smooth parameter δ is set equal to 10^{-4}. Recall that SGP works not only as a minimization method for a regularized objective functional but also as an iterative regularization method applied to the not-regularized functional by means of an early stopping technique. In the latter case the semi-convergent behavior of the algorithm can be exploited to stop the iteration, and the number of iteration required serves as regularization parameter for the method. The same holds true for VMILA, but in the ROI CT case the not-regularized term, *i.e.*, the discrepancy term, of the objective function is smooth: in this case, VMILA is nothing but SGP.

As far as the Tikhonov-like regularization term concerns, the shearlet operator was obtained by using the Spot–A Linear-Operator Toolbox with the routine kindly provided by Wang-Q Lim, co-author of one of available shearlets packages (see Section 4.4 for more details about shearlets implementation and available softwares). In details, the number of scales for the shearlet transform has been set



(a) Noise-free sinogram. (b) Noisy sinogram. (c) Difference: noise amount.

Figure 6.4: The noise-free full sinogram (left), the noisy sinogram (center) and their difference (right), that reveals the amount of noise.

equal to 4 and the number of directions across the scales is set to $(8, 8, 16, 16)$.

The threshold L for the scaling matrix D_k has been set equal to 10^5, and, concerning the steplength α_k defined by Algorithm 2, we set $\alpha_{\min} = 10^{-5}$, $\alpha_{\max} = 10^5$ and $\alpha_0 = 1.3$ (see Section 5.2.1). For the line search parameters we used $\gamma = 0.4$, $\beta = 10^{-4}$, and $\kappa = 10$ for the nonmonotone version. These choices for the scaling matrix, the steplength rule and the line search parameters apply to both SGP and VMILA.

Finally, the proposed methods imply the use of an iterative algorithm with an initial guess $\mathbf{f}^{(0)}$. We choose to use the vector of all zeros, as first estimate for $\mathbf{f}^{(0)}$. There is a vast literature about the influence of the choice of the initial guess for diverse classes of algorithms, for instance [154]. However, as far as SGP and VMILA concern, the line search strategy ensures global convergence. In VIMILA, the initial guess of the inner loop at the first outer iterate is, likewise, the vector of all zeros, while at all successive iterates a “warm start” is exploited, namely the inner solver is initialized with the dual solution computed at the previous iterate.

All the algorithms were implemented in Matlab 8.1.0 and the experiments performed on a dual CPU server, equipped with two 6-cores Intel Xeon X5690 at 3.46GHz, 188 GB DDR3 central RAM memory and up to 12 TB of disk storage.

6.2 SGP: smooth objective function

In this Section we present the results obtained using SGP on the modified Shepp-Logan phantom previously described. For sake of convenience, the results are organized in two different Subsections, depending on the implicit or explicit formulation. At the end of the Section, pros and cons of the two approaches are compared to assess the best approach with respect to both the objective function and the regularization parameters.

	$\mathbf{r}_{\text{ROI}} = 0.5N$			$\mathbf{r}_{\text{ROI}} = 0.3N$			$\mathbf{r}_{\text{ROI}} = 0.25N$		
	iter	value	param	iter	value	param	iter	value	param
				PSNR					
Sh+TV	686	32.34	$\mu = 10^{-4}$	931	38.98	$\mu = 10^{-4}$	2986	41.35	$\mu = 10^{-4}$
			$\rho = 0.1$			$\rho = 1$			$\rho = 10$
Sh	1308	23.95	$\mu = 10^{-2}$	1186	26.63	$\mu = 10^{-4}$	1544	26.09	$\mu = 10^{-4}$
TV	1341	32.12	$\rho = 0.1$	1845	40.22	$\rho = 1$	1661	44.09	$\rho = 1$
ES	156	31.12		45	34.68		36	38.37	
				Relative error					
Sh+TV	686	0.12	$\mu = 10^{-4}$	931	0.12	$\mu = 10^{-4}$	2986	0.11	$\mu = 10^{-4}$
			$\rho = 0.1$			$\rho = 1$			$\rho = 10$
Sh	2986	0.30	$\mu = 10^{-2}$	1186	0.50	$\mu = 10^{-4}$	1544	0.66	$\mu = 10^{-4}$
TV	1341	0.12	$\rho = 0.1$	1845	0.10	$\rho = 1$	1661	0.08	$\rho = 1$
ES	156	0.13		45	0.20		36	0.16	

Table 6.1: Optimal results for all the formulation of the implicit objective function. The corresponding reconstructed images are reported in Figure 6.6.
Sh+TV = shealerts and TV, (2.22). Sh = just shearlets, (2.18). TV = pure TV. ES = early stopping.

6.2.1 Implicit formulation

Different versions of the SGP method have been designed to apply to the different formulations of the implicit objective function, where "different" indicates the variable level of regularization considered (see Section 5.2.1). Indeed, beside formulations (2.18)-(2.22) with $p = 2$, we considered a pure TV approach and the early stopping technique on the discrepancy term. In Tables 6.1 and 6.2 are reported the optimal results for all the possible implicit formulation, and for decreasing ROI radii, compared against the two state-of-the-art figures of merit specified in the previous Section, namely the PSNR and the relative error, both evaluated inside the ROI only. In both Tables, in each row, and for a fixed column (*i.e.*, for a fixed ROI radius $\mathbf{r}_{\text{ROI}}$), it is reported the best value of the figure of merit (value), the number of iteration at which was achieved (iter) and the corresponding optimal regularization parameters (param). As already pointed out, for the early stopping approach the number of iterations serves as regularization parameter.

The corresponding reconstructed images are reported in Figure 6.6 and 6.7, respectively. In all the Figures, each row contains the ROI image reconstruction obtained with a different formulations of the implicit objective function, while each column contains the ROI image reconstructions obtained for a different ROI radius. In all the Figures, the ROI is identified with a dashed white circle.

The results reported in Table 6.1, corresponding to the largest ROI radii $\mathbf{r}_{\text{ROI}} =$

	$\mathbf{r}_{\text{ROI}} = 0.2N$			$\mathbf{r}_{\text{ROI}} = 0.15N$			$\mathbf{r}_{\text{ROI}} = 0.1N$		
	iter	value	param	iter	value	param	iter	value	param
				PSNR					
Sh+TV	3071	39.89	$\mu = 10^{-4}$	2707	37.86	$\mu = 10^{-4}$	3296	36.51	$\mu = 10^{-4}$
			$\rho = 10$			$\rho = 10$			$\rho = 10$
Sh	463	28.58	$\mu = 10^{-4}$	315	28.97	$\mu = 10^{-4}$	1347	28.10	$\mu = 10^{-4}$
TV	2257	45.86	$\rho = 1$	2236	47.87	$\rho = 1$	2038	51.28	$\rho = 1$
ES	46	40.61		36	42.47		46	45.58	
				Relative error					
Sh+TV	3071	0.19	$\mu = 10^{-4}$	2707	0.35	$\mu = 10^{-4}$	3296	0.67	$\mu = 10^{-4}$
			$\rho = 10$			$\rho = 10$			$\rho = 10$
Sh	463	0.69	$\mu = 10^{-4}$	315	0.97	$\mu = 10^{-4}$	1347	1.78	$\mu = 10^{-4}$
TV	2257	0.09	$\rho = 1$	2236	0.11	$\rho = 1$	2038	0.12	$\rho = 1$
ES	46	0.17		36	0.21		46	0.24	

Table 6.2: Optimal results for all the formulation of the implicit objective function. The corresponding reconstructed images are reported in Figure 6.7.
Sh+TV = shealerts and TV, (2.22). Sh = just shearlets, (2.18). TV = pure TV. ES = early stopping.

$0.5N$, $0.3N$, $0.25N$, show that the best approaches are the ones that incorporate the TV term in the formulation of the objective function (first and third rows). The performance of the early stopping technique seems to be comparable, but with a considerably lower number of iterations to converge, while the pure shearlets approach shows increasing values for the ROI relative error and, correspondingly, decreasing values for the ROI PSNR, as the ROI radius decreases. For the smallest ROI radii, $\mathbf{r}_{\text{ROI}} = 0.2N$, $0.15N$, $0.1N$, the situation is slightly different. As shown in Table 6.2, the best approaches are pure TV and early stopping (third and fourth rows). The shearlets plus TV approach exhibits worse values for both figures of merit, but still acceptable, while the pure shearlets approach reaches unacceptable values (the relative error is even ten times worse and the ROI PSNR loses out up to 20 dB) as the ROI radius becomes smaller.

Except for the largest ROI radius $\mathbf{r}_{\text{ROI}} = 0.5N$, the optimal value for the regularization parameter μ is always equal to 10^{-4}, the smallest one of the investigated range, while the optimal value for the TV regularization parameter ρ is equal to 1, for the pure TV approach, and to 10, the largest one of the investigated range, for the shearlets plus TV formulation.

The findings in Table 6.1 are confirmed by the corresponding reconstructed images reported in Figure 6.6. Indeed, on a visual basis, the best reconstruction are those whose corresponding approach includes the TV term. In these images

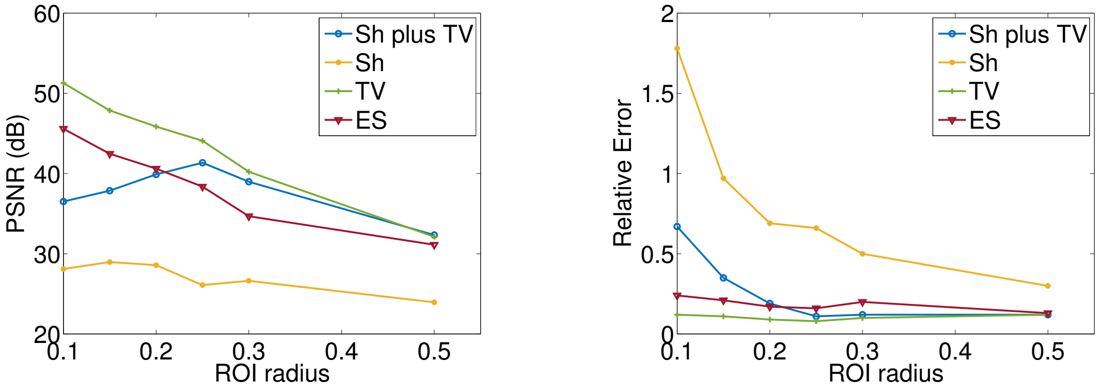


Figure 6.5: Implicit formulation. Trend of the ROI PSNR (left) and the ROI relative error (right) for decreasing ROI radii.

the contrast of the features inside the ROI is high and the transition between the inside and the exterior of the ROI seems smooth (first and third rows in Figure 6.6). The early stopping approach suffers from a mild checkerboard effect, but all the fundamental features are still detected accurately (Figure 6.6, fourth row). In the pure shearlets approach, the checkerboard effect is predominant, namely many component of the ROI solution are zero, compromising an accurate reconstruction of the smallest features on the bottom (Figure 6.6, second row). For the smallest ROI radii $\mathbf{r}_{\mathrm{ROI}} = 0.2N$, $0.15N$, $0.1N$, the results in Table 6.2 are only partially confirmed by the corresponding reconstructed images reported in Figure 6.7. Indeed, on a visual basis, the best approaches seem to be the ones that incorporate the TV term in the objective function (first and third rows in Figure 6.7), while the results reported in Table 6.2 select pure TV and early stopping (third and fourth rows in 6.7). As a matter of fact, the shearlets plus TV and the pure TV approach exhibits more accurate reconstructions: no artifacts of any type are visible and the features are detected at high contrast. On the contrary, the early stopping approach suffers from a mild checkerboard effect that, anyway, does not compromise the detection of the main features. As for the largest ROI radii, the pure shearlets approach (Figure 6.7, second row) suffers from a prominent checkerboard effect and a ring artifact is evident on the boundary of the ROI. However, even if panel (f) in Figure 6.7 corresponds to a definitely large relative error (100%), the reconstruction is good enough to identify, on a visual basis, the fundamental features. Such a large relative error is probably due to the checkerboard effect, that results in a sub- or under-estimation of the pixel intensity.

In Figure 6.5 we report the trend of the ROI PSNR and the ROI relative error.

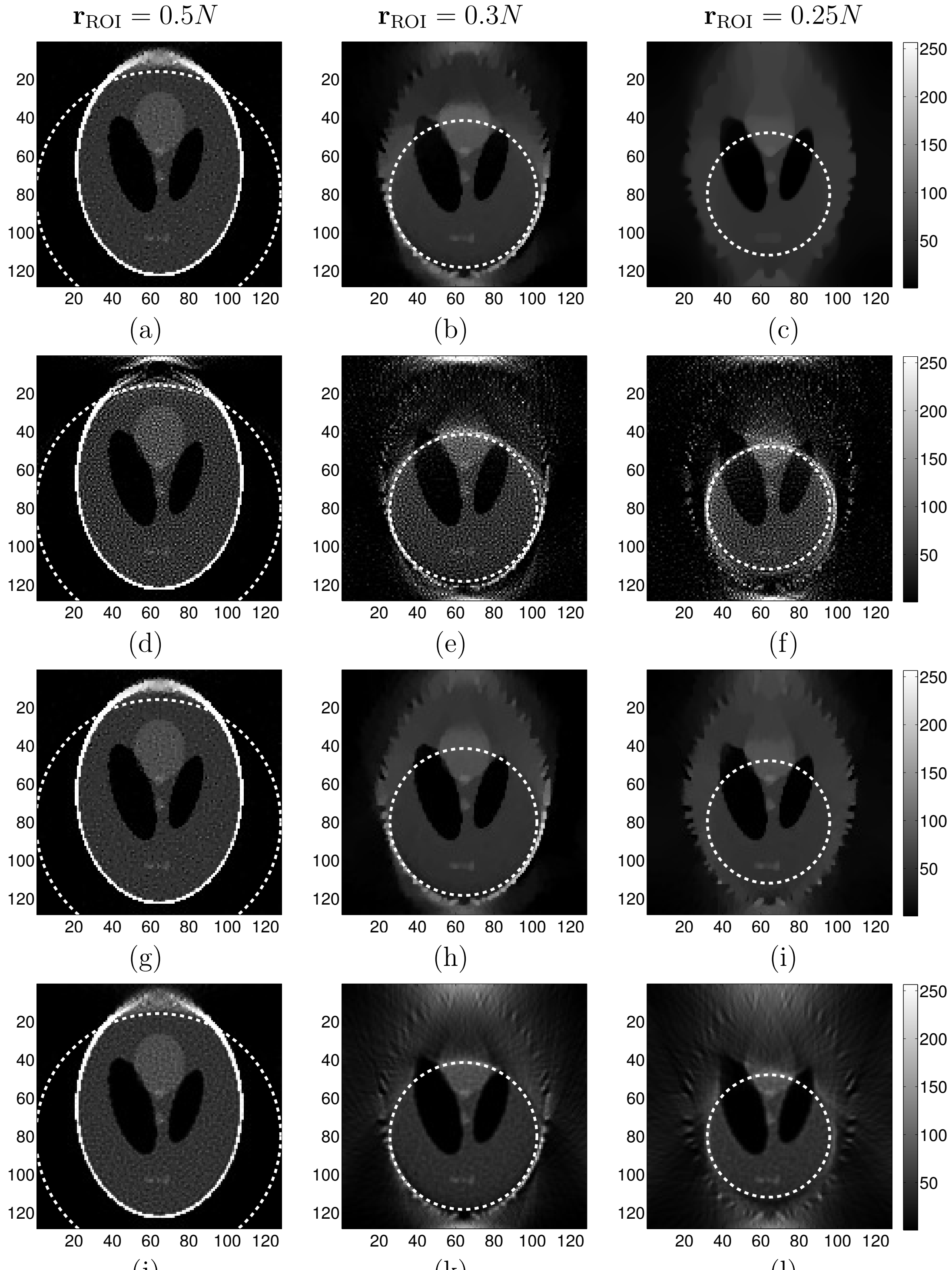
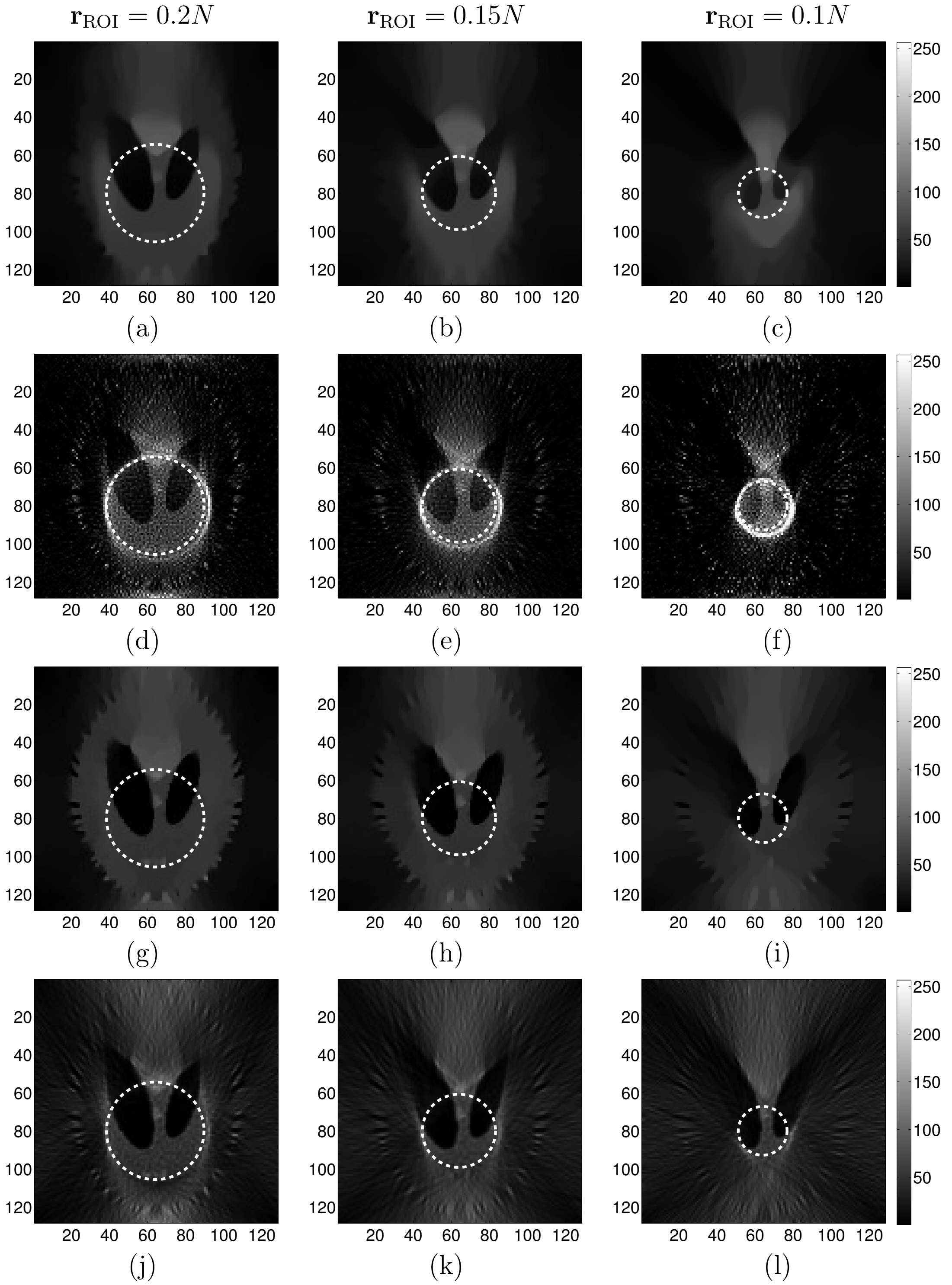

$\mathbf{r}_{\mathrm{ROI}} = 0.5N$
$\mathbf{r}_{\mathrm{ROI}} = 0.3N$
$\mathbf{r}_{\mathrm{ROI}} = 0.25N$
20
40
60
80
100
120
250
200
150
100
50
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)

Figure 6.6: Optimal reconstructions of the Shepp-Logan phantom for the implicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.5N$ for (a), (d), (g) and (j), $\mathbf{r}_{\mathrm{ROI}} = 0.3N$ for (b), (e), (h) and (k), $\mathbf{r}_{\mathrm{ROI}} = 0.25N$ for (c), (f), (i) and (l). First row: shealerts and TV. Second row: just shearlets. Third row: pure TV. Fourth row: early stopping.


$\mathbf{r}_{\mathrm{ROI}} = 0.2N$
$\mathbf{r}_{\mathrm{ROI}} = 0.15N$
$\mathbf{r}_{\mathrm{ROI}} = 0.1N$
20
40
60
80
100
120
250
200
150
100
50

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.7: Optimal reconstructions of the Shepp-Logan phantom for the implicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.2N$ for (a), (d), (g) and (j), $\mathbf{r}_{\mathrm{ROI}} = 0.15N$ for (b), (e), (h) and (k), $\mathbf{r}_{\mathrm{ROI}} = 0.1N$ for (c), (f), (i) and (l). First row: shealerts and TV. Second row: just shearlets. Third row: pure TV. Fourth row: early stopping.

	$\mathbf{r}_{\text{ROI}} = 0.5N$			$\mathbf{r}_{\text{ROI}} = 0.3N$			$\mathbf{r}_{\text{ROI}} = 0.25N$		
	iter	value	param	iter	value	param	iter	value	param
	PSNR								
Sh+TV	249	32.48	$\mu = 10^{-3}$	2223	38.92	$\mu = 10^{-4}$	4366	40.23	$\mu = 10^{-4}$
			$\rho = 0.1$			$\rho = 1$			$\rho = 10$
Sh	136	29.58	$\mu = 10^{-2}$	251	30.60	$\mu = 10^{-4}$	383	24.84	$\mu = 10^{-2}$
TV	1107	32.19	$\rho = 0.1$	912	38.73	$\rho = 1$	1283	42.22	$\rho = 1$
ES	105	30.81		71	34.67		45	35.84	
	Relative error								
Sh+TV	249	0.11	$\mu = 10^{-3}$	2223	0.12	$\mu = 10^{-4}$	4366	0.13	$\mu = 10^{-4}$
			$\rho = 0.1$			$\rho = 1$			$\rho = 10$
Sh	136	0.16	$\mu = 10^{-2}$	251	0.31	$\mu = 10^{-4}$	383	0.76	$\mu = 10^{-2}$
TV	1107	0.12	$\rho = 0.1$	912	0.12	$\rho = 1$	1283	0.10	$\rho = 1$
ES	105	0.14		71	0.20		45	0.22	

Table 6.3: Optimal results for all the formulation of the explicit objective function. The corresponding reconstructed images are reported in Figure 6.9.
Sh+TV = shealerts and TV, (2.23). Sh = just shearlets, (2.19). TV = pure TV. ES = early stopping.

Overall these plots show that, for all ROI radii, the TV-based approaches (green and blue lines) performs better, while the pure shearlets formulation (yellow line) seems to perform significantly worse.

6.2.2 Explicit formulation

Similarly to the implicit case, different versions of the "enlarged" variable version of SGP have been designed to apply to the different formulations of the explicit objective function (see Section 5.2.1). Tables 6.3 and 6.4 summarize the optimal results for all the possible explicit formulation (namely, formulations (2.23)-(2.19) with $p = 2$, pure TV and early stopping), and for decreasing ROI radii, compared against the ROI PSNR and the ROI relative error.

The corresponding reconstructed images are reported in Figure 6.9 and 6.10, respectively. In all the Figures, a dashed white circle identifies the ROI. Figures 6.11, and 6.12 collects the corresponding reconstructed sinograms obtained for the explicit formulation. Here, the truncated projections, that corresponds to the circular ROI in the image domain, are marked by two sinusoidal dashed white lines.

The results reported in Table 6.3 and 6.4 show that, for all ROI radii, the best approaches are those coupled with the TV term in the formulation of the

	$\mathbf{r}_{\text{ROI}} = 0.2N$			$\mathbf{r}_{\text{ROI}} = 0.15N$			$\mathbf{r}_{\text{ROI}} = 0.1N$		
	iter	value	param	iter	value	param	iter	value	param
	PSNR								
Sh+TV	1056	37.51	$\mu = 10^{-4}$	2474	36.40	$\mu = 10^{-4}$	626	30.86	$\mu = 10^{-4}$
			$\rho = 10$			$\rho = 10$			$\rho = 1$
Sh	177	25.85	$\mu = 10^{-2}$	132	25.03	$\mu = 10^{-4}$	197	25.61	$\mu = 10^{-4}$
TV	1110	40.56	$\rho = 1$	1160	38.30	$\rho = 1$	2069	34.51	$\rho = 1$
ES	64	34.77		67	32.89		6	36.02	
	Relative error								
Sh+TV	1056	0.25	$\mu = 10^{-4}$	2474	0.41	$\mu = 10^{-4}$	626	1.29	$\mu = 10^{-4}$
			$\rho = 10$			$\rho = 10$			$\rho = 1$
Sh	177	0.95	$\mu = 10^{-2}$	132	1.53	$\mu = 10^{-2}$	197	2.37	$\mu = 10^{-2}$
TV	1110	0.17	$\rho = 1$	1160	0.33	$\rho = 1$	2069	0.85	$\rho = 1$
ES	64	0.34		67	0.62		6	0.71	

Table 6.4: Optimal results for all the formulation of the explicit objective function. The corresponding reconstructed images are reported in Figure 6.10.
Sh+TV = shealerts and TV, (2.23). Sh = just shearlets, (2.19). TV = pure TV. ES = early stopping.

objective function (first and third rows). For the largest ROI radii (namely, $\mathbf{r}_{\text{ROI}} = 0.5N$, $0.3N$, $0.25N$), the shearlets plus TV and the pure TV approaches exhibit approximatively the same values, for both figures of merit, while for the smallest ROI radii (*i.e.*, $\mathbf{r}_{\text{ROI}} = 0.2N$, $0.15N$, $0.1N$) the different performance between the two approaches increases as the ROI radius decreases. The outcome of the early stopping technique, that requires a considerably lower number of iterations to converge, seems to be comparable, except for the smallest ROI radius. Indeed, the early stopping technique seems to outperform all the other approaches when $\mathbf{r}_{\text{ROI}} = 0.1N$. However, this is not confirmed on a visual basis, since the corresponding reconstructed image (panel (f), Figure 6.10) is completely blurred and the features of interest are poorly visible. Finally, the pure shearlets approach seems to perform considerably worse than the others, reaching unacceptable values (the ROI relative error is even ten times worse and the ROI PSNR loses out up to 15 dB), as the ROI radius becomes smaller.

As far as the optimal values for the regularization parameter concern, the optimal choice for μ, except for the largest ROI radius $\mathbf{r}_{\text{ROI}} = 0.5N$, is always 10^{-4} in the shearlets plus TV approach, and μ equal to 10^{-2} for the pure shearlets approach, except for $\mathbf{r}_{\text{ROI}} = 0.3N$. In any case, the optimal values for μ are among the smallest ones of the investigated range. On the contrary, the optimal value for the TV regularization parameter ρ is either equal to 1, for the pure TV approach and for $\mathbf{r}_{\mathrm{ROI}} = 0.3N$, $0.1N$ with the shearlets plus TV formulation, or to 10, the largest one of the investigated range, for $\mathbf{r}_{\mathrm{ROI}} = 0.25N$, $0.2N$, $0.15N$ with the shearlets plus TV approach. Lastly, the largest ROI radius $\mathbf{r}_{\mathrm{ROI}} = 0.5N$, requires a smaller TV regularization parameter, namely $\rho = 0.1$, and a slightly bigger regularization parameter μ equal to either 10^{-3} or 10^{-2}, depending on the approach.

The results in Tables 6.1 and 6.2 are basically confirmed by the corresponding reconstructed images reported in Figures 6.9 and 6.10, respectively. On a visual basis, the best reconstruction for all ROI radii are those associated with a formulation of the objective function that includes the TV term (first and third rows in Figures 6.9 and 6.10). Indeed, the contrast of the features inside the ROI is high and the transition between the inside and the exterior of the ROI appears smooth. The early stopping approach suffers from a mild checkerboard effect, but all the fundamental features are still detected accurately (Figures 6.9 and 6.10, fourth row), except for the smallest ROI radius $\mathbf{r}_{\mathrm{ROI}} = 0.1N$, as already pointed out. In the pure shearlets approach, the checkerboard effect is prominent, and, for the largest ROI radii, an accurate reconstruction of the smallest features on the bottom is compromised (Figures 6.9 and 6.10, second row). Also, the boundary of the ROI contains a quite evident ring artifact. However, similarly to the implicit case, even if panel (e) and (f) in Figure 6.10 corresponds to a definitely large relative error (100%), the reconstruction is good enough to identify, on a visual basis, the fundamental features. This is probably due to quantitatively inexact reconstructed density values, because of the checkerboard effect.

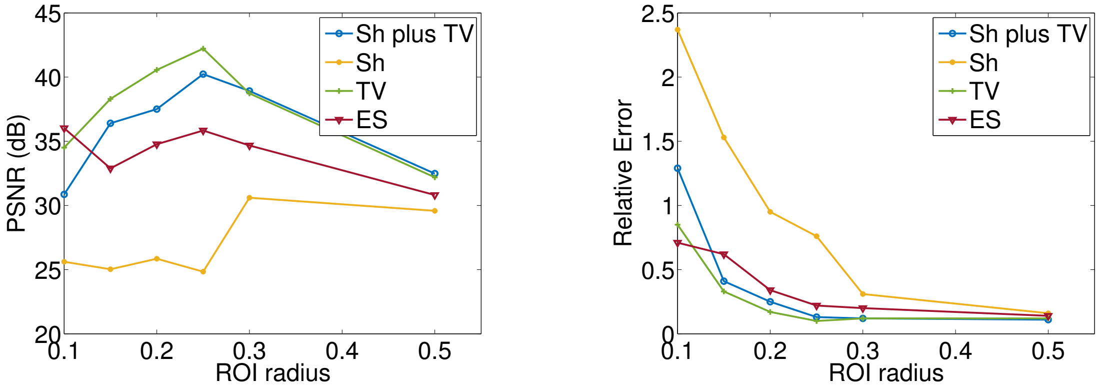
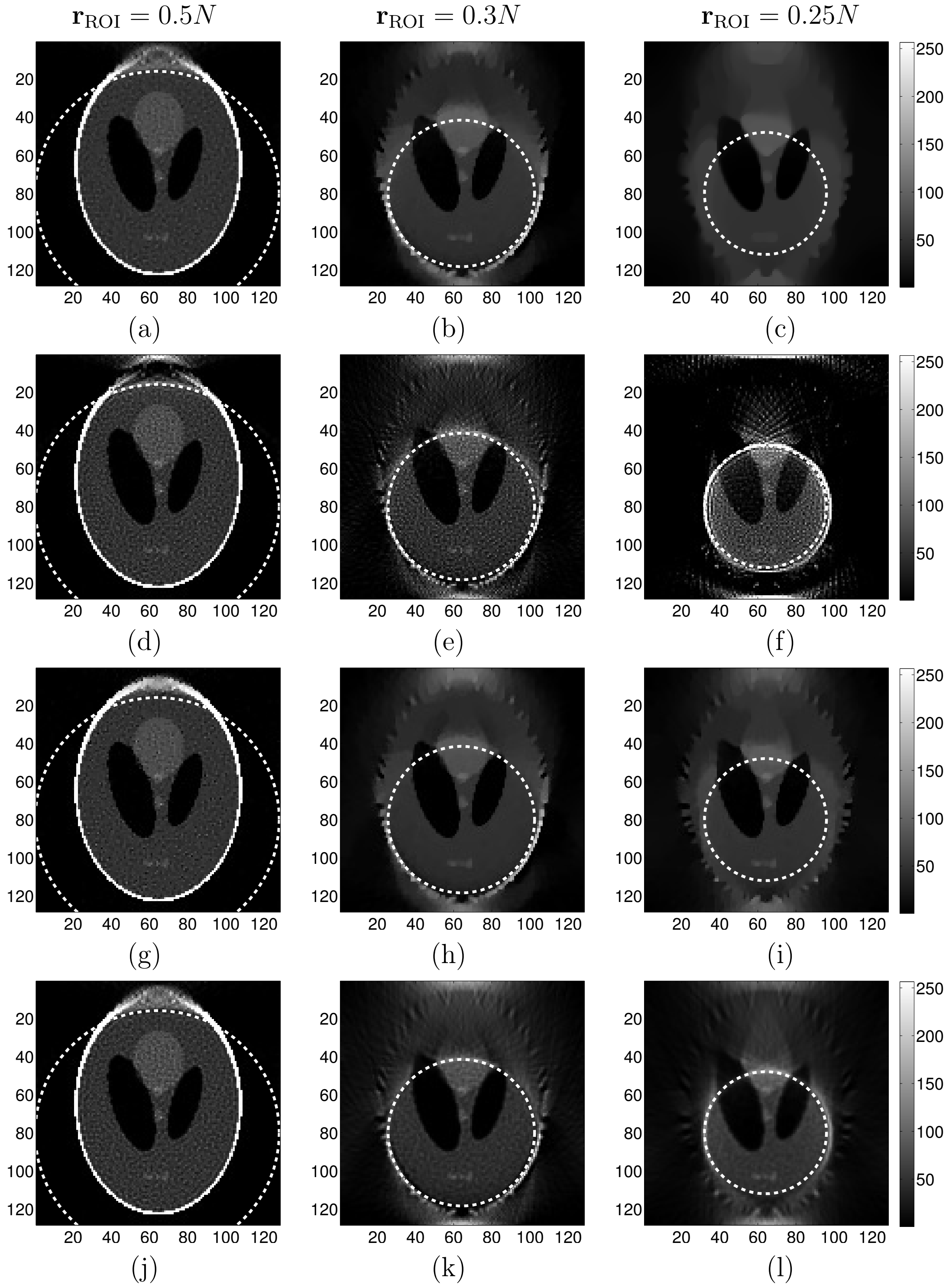


Figure 6.8: Explicit formulation. Trend of the ROI PSNR (left) and the ROI relative error (right) for decreasing ROI radii.

In Figure 6.8 we report the trend of the ROI PSNR and the ROI relative error. Similarly to the implicit case, these plots show that, for all ROI radii, the TV-based approaches (green and blue lines) performs better, while the pure shearlets formulation (yellow line) seems to perform significantly worse.


$\mathbf{r}_{\text{ROI}} = 0.5N$
$\mathbf{r}_{\text{ROI}} = 0.3N$
$\mathbf{r}_{\text{ROI}} = 0.25N$
20
40
60
80
100
120
20 40 60 80 100 120
250
200
150
100
50

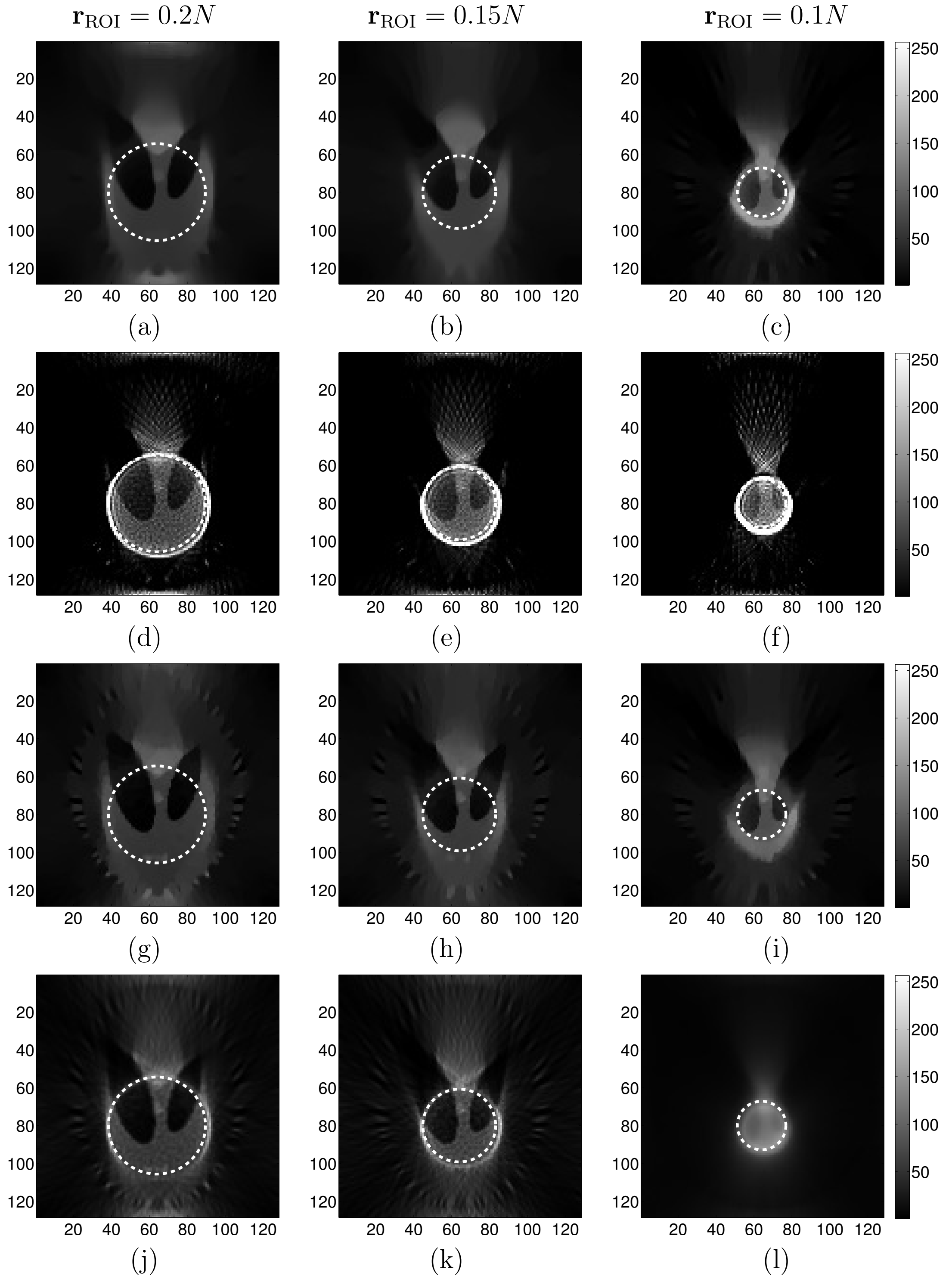
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.9: Optimal reconstructions of the Shepp-Logan phantom for the explicit formulation. $\mathbf{r}_{\text{ROI}} = 0.5N$ for (a), (d), (g) and (j), $\mathbf{r}_{\text{ROI}} = 0.3N$ for (b), (e), (h) and (k), $\mathbf{r}_{\text{ROI}} = 0.25N$ for (c), (f), (i) and (l). First row: shealerts and TV. Second row: just shearlets. Third row: pure TV. Fourth row: early stopping.


$\mathbf{r}_{\mathrm{ROI}} = 0.2N$
$\mathbf{r}_{\mathrm{ROI}} = 0.15N$
$\mathbf{r}_{\mathrm{ROI}} = 0.1N$
20
40
60
80
100
120
250
200
150
100
50

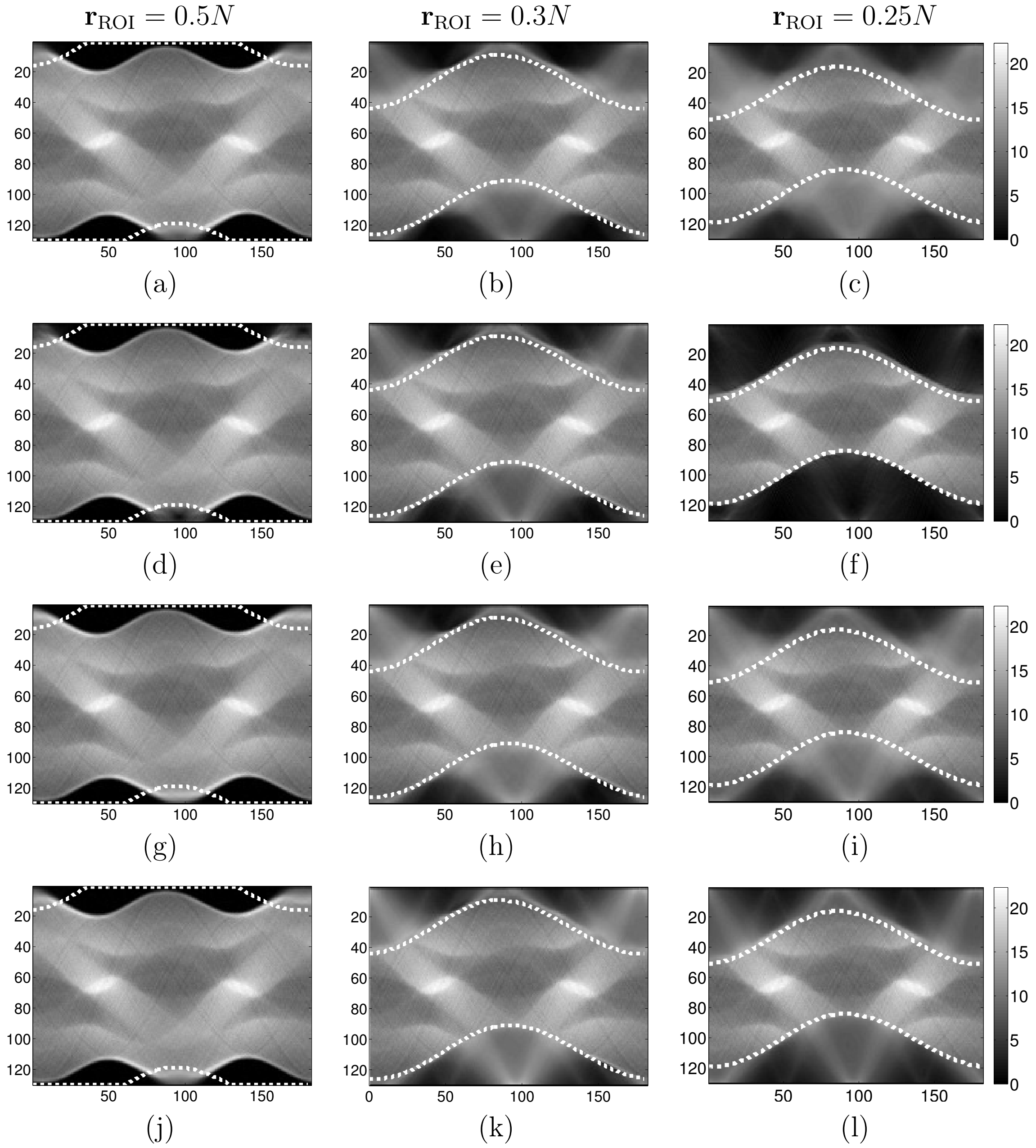
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.10: Optimal reconstructions of the Shepp-Logan phantom for the explicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.2N$ for (a), (d), (g) and (j), $\mathbf{r}_{\mathrm{ROI}} = 0.15N$ for (b), (e), (h) and (k), $\mathbf{r}_{\mathrm{ROI}} = 0.1N$ for (c), (f), (i) and (l). First row: shealerts and TV. Second row: just shearlets. Third row: pure TV. Fourth row: early stopping.



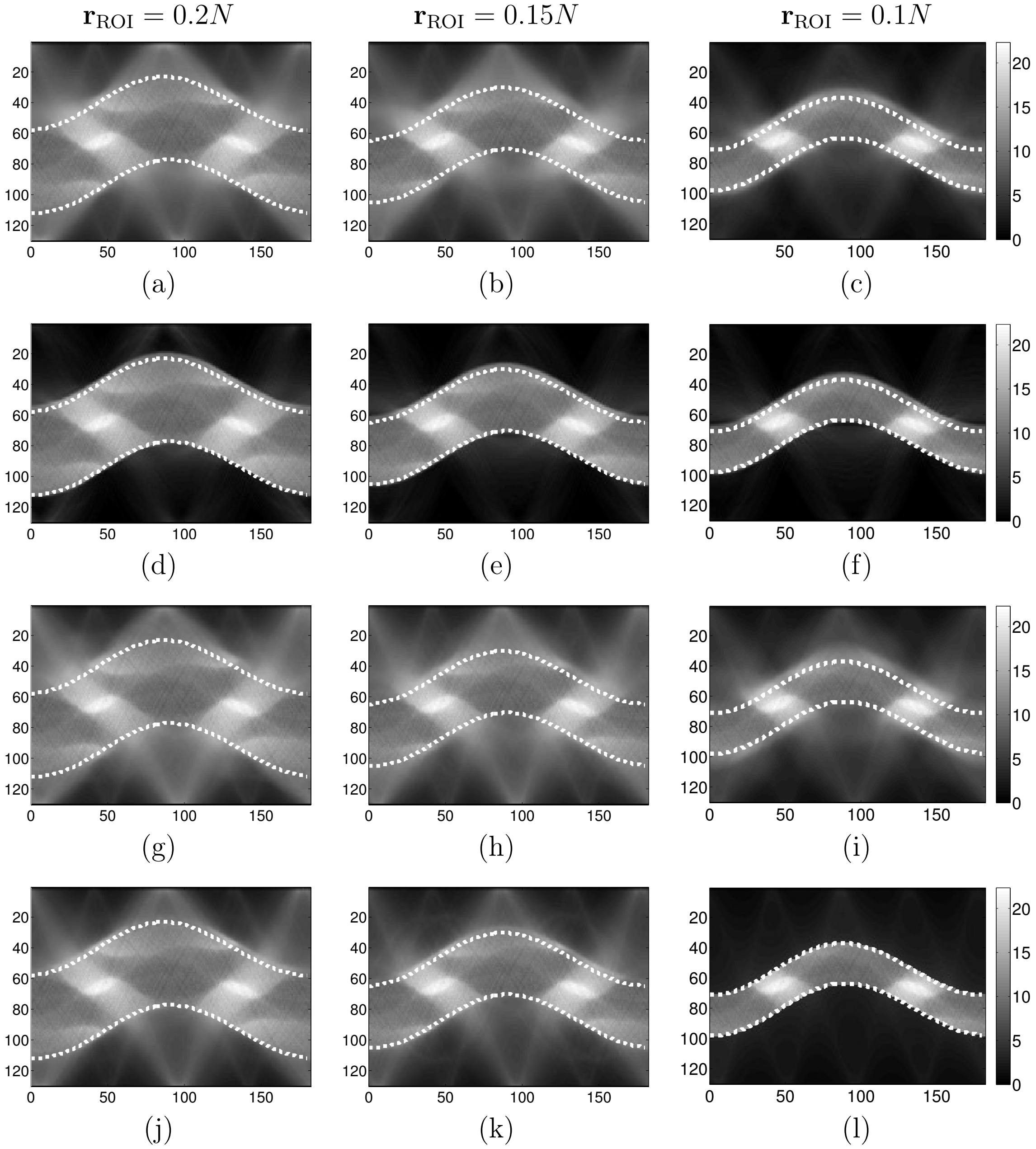
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.11: Optimal reconstructions of the sinogram of the Shepp-Logan phantom for explicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.5N$ for (a), (d), (g) and (j), $\mathbf{r}_{\mathrm{ROI}} = 0.3N$ for (b), (e), (h) and (k), $\mathbf{r}_{\mathrm{ROI}} = 0.25N$ for (c), (f), (i) and (l). First row: shealerts and TV. Second row: just shearlets. Third row: pure TV. Fourth row: early stopping.


$\mathbf{r}_{\mathrm{ROI}} = 0.2N$
$\mathbf{r}_{\mathrm{ROI}} = 0.15N$
$\mathbf{r}_{\mathrm{ROI}} = 0.1N$
20
40
60
80
100
120
0
50
100
150
0
5
10
15
20

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.12: Optimal reconstructions of the sinogram of the Shepp-Logan phantom for explicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.2N$ for (a), (d), (g) and (j), $\mathbf{r}_{\mathrm{ROI}} = 0.15N$ for (b), (e), (h) and (k), $\mathbf{r}_{\mathrm{ROI}} = 0.1N$ for (c), (f), (i) and (l). First row: shealerts and TV. Second row: just shearlets. Third row: pure TV. Fourth row: early stopping.

6.2.3 Final Remarks

The numerical experiments reported in this Section show that all SGP-based algorithms are satisfactory for all versions of the objective function and regardless of the size and the location of the ROI. Except for the pure shearlets approach, in all the other formulations the main structures are recovered very accurately, with no ring artifacts and a sufficiently smooth transition from between the inside and the exterior of the ROI, also for rather small ROI sizes. The implicit purely TV-based approach performs slightly better than the shearlets plus TV approach, at least for smaller ROI radii (*i.e.*, when the ROI is fully inside the field of view). We conjecture that this behavior might be dependent on the phantom features (which is piecewise constant) and may not hold for more general data. We expect that, using more realistic sinograms, the contribution of the shearlet term will become more relevant for the regularization. Evidence of this can be found in [156].

6.3 VMILA: nonsmooth objective function

In this Section we present the results obtained using VMILA on the modified Shepp-Logan phantom described in Section 6.1. So far, VMILA has been tested only on the implicit formulation (2.18)-(2.22) with $p = 1$. A suitable version of VMILA for the explicit formulation is a work in progress, as outlined in Chapter 5.

6.3.1 Implicit formulation

As already pointed out, when we drop the regularization term from the implicit formulation, VMILA is nothing but SGP. For this reason, VMILA has been designed to apply only to the two objective functions (2.18) and (2.22), with $p = 1$. The optimal results for all these formulation are reported in Tables 6.5 and 6.6, compared against the PSNR and the relative error, both evaluated inside the ROI only, and for decreasing ROI radii. In both Tables, for a fixed ROI radius $\mathbf{r}_{\text{ROI}}$ (*i.e.*, for a fixed column), each row contains the best value for the figure of merit (value), the number of iteration at which was achieved (iter) and the corresponding optimal regularization parameters (param).

The corresponding reconstructed images are reported in Figure 6.14 and 6.15, respectively. In both Figures, each row contains the ROI image reconstruction obtained by exploiting a different formulations of the implicit objective function, while each column contains the ROI image reconstructions obtained for different ROI radii. In all the Figures, the ROI is identified with a dashed white circle.

The results reported in Table 6.5 and 6.6 show that the performance of the two approaches is comparable. The shearlets plus TV approach performs slightly

	$\mathbf{r}_{\text{ROI}} = 0.5N$			$\mathbf{r}_{\text{ROI}} = 0.3N$			$\mathbf{r}_{\text{ROI}} = 0.25N$		
	iter	value	param	iter	value	param	iter	value	param
	PSNR								
Sh+TV	145	31.43	$\mu = 10^{-4}$ $\rho = 1$	162	37.41	$\mu = 10^{-3}$ $\rho = 1$	159	39.95	$\mu = 10^{-2}$ $\rho = 1$
Sh	68	29.36	$\mu = 10^{-3}$	91	33.91	$\mu = 10^{-4}$	83	36.94	$\mu = 10^{-4}$
	Relative error								
Sh+TV	145	0.13	$\mu = 10^{-4}$ $\rho = 1$	162	0.14	$\mu = 10^{-3}$ $\rho = 1$	159	0.13	$\mu = 10^{-2}$ $\rho = 1$
Sh	68	0.16	$\mu = 10^{-3}$	91	0.22	$\mu = 10^{-4}$	83	0.19	$\mu = 10^{-4}$

Table 6.5: Optimal results for all the formulation of the implicit objective function. The corresponding reconstructed images are reported in Figure 6.14.
Sh+TV = shealerts and TV, (2.22). Sh = just shearlets, (2.18).

better than the purely shealets-based approach, and the quantitative difference between the two is remarkable only for $\mathbf{r}_{\text{ROI}} = 0.3N$ and $\mathbf{r}_{\text{ROI}} = 0.1N$. This is also evident in Figure 6.13, where the trend of the ROI PSNR and the ROI relative error is reported. Indeed, the shearlet plus TV approach (blue line) performs better than the pure shearlets formulation (yellow line).

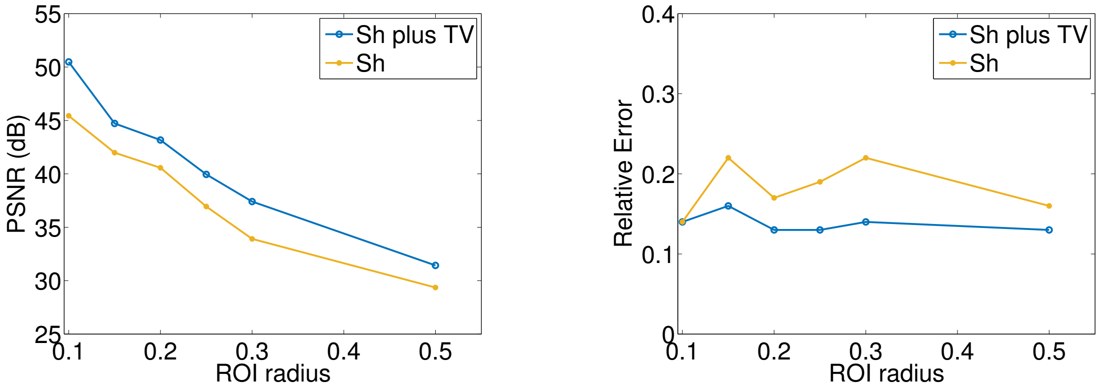
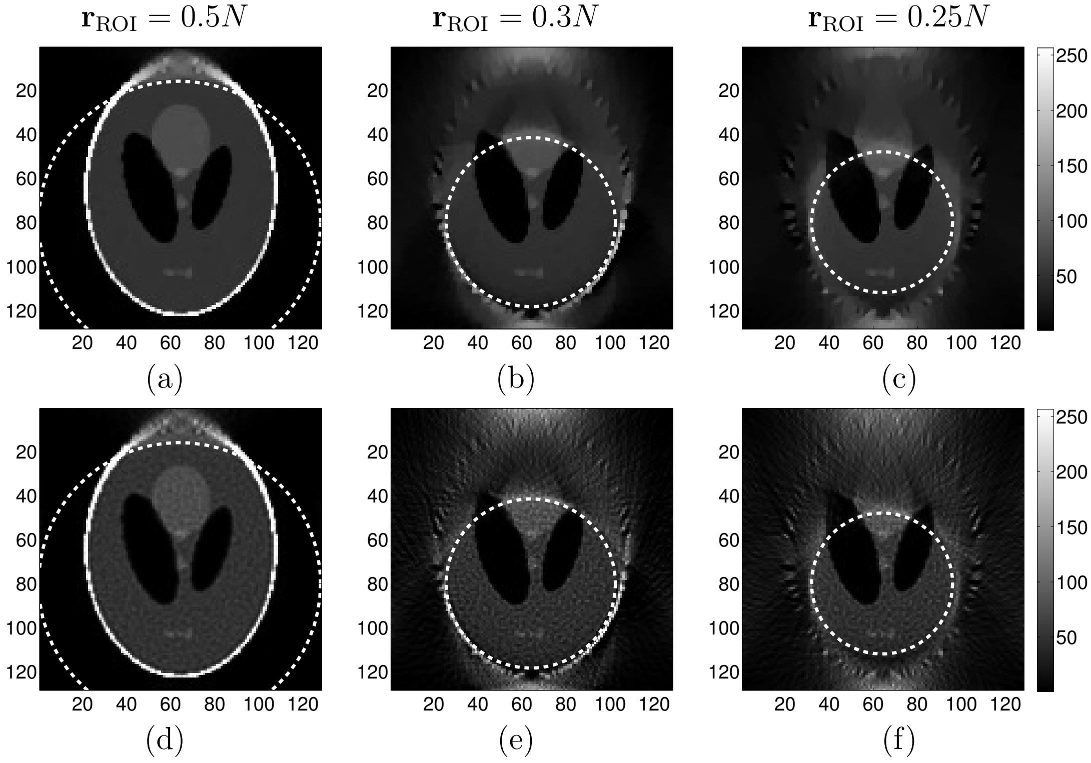


Figure 6.13: Implicit formulation. Trend of the PSNR (left) and the relative error (right) for decreasing ROI radii.

	$\mathbf{r}_{\mathrm{ROI}} = 0.2N$			$\mathbf{r}_{\mathrm{ROI}} = 0.15N$			$\mathbf{r}_{\mathrm{ROI}} = 0.1N$		
	iter	value	param	iter	value	param	iter	value	param
	PSNR								
Sh+TV	143	43.17	$\mu = 10^{-3}$ $\rho = 1$	131	44.72	$\mu = 10^{-4}$ $\rho = 1$	213	50.48	$\mu = 10^{-4}$ $\rho = 1$
Sh	55	40.57	$\mu = 10^{-4}$	78	41.98	$\mu = 10^{-4}$	48	45.43	$\mu = 10^{-4}$
	Relative error								
Sh+TV	143	0.13	$\mu = 10^{-3}$ $\rho = 1$	131	0.16	$\mu = 10^{-4}$ $\rho = 1$	213	0.14	$\mu = 10^{-4}$ $\rho = 1$
Sh	55	0.17	$\mu = 10^{-4}$	78	0.22	$\mu = 10^{-4}$	48	0.24	$\mu = 10^{-4}$

Table 6.6: Optimal results for all the formulation of the implicit objective function. The corresponding reconstructed images are reported in Figure 6.15.
Sh+TV = shealerts and TV, (2.22). Sh = just shearlets, (2.18).


$\mathbf{r}_{\mathrm{ROI}} = 0.5N$ $\mathbf{r}_{\mathrm{ROI}} = 0.3N$ $\mathbf{r}_{\mathrm{ROI}} = 0.25N$

(a) (b) (c)

(d) (e) (f)

Figure 6.14: Optimal reconstructions of the Shepp-Logan phantom for the implicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.5N$ for (a) and (d), $\mathbf{r}_{\mathrm{ROI}} = 0.3N$ for (b) and (e), $\mathbf{r}_{\mathrm{ROI}} = 0.25N$ for (c) and (f). First row: shealerts and TV. Second row: just shearlets.

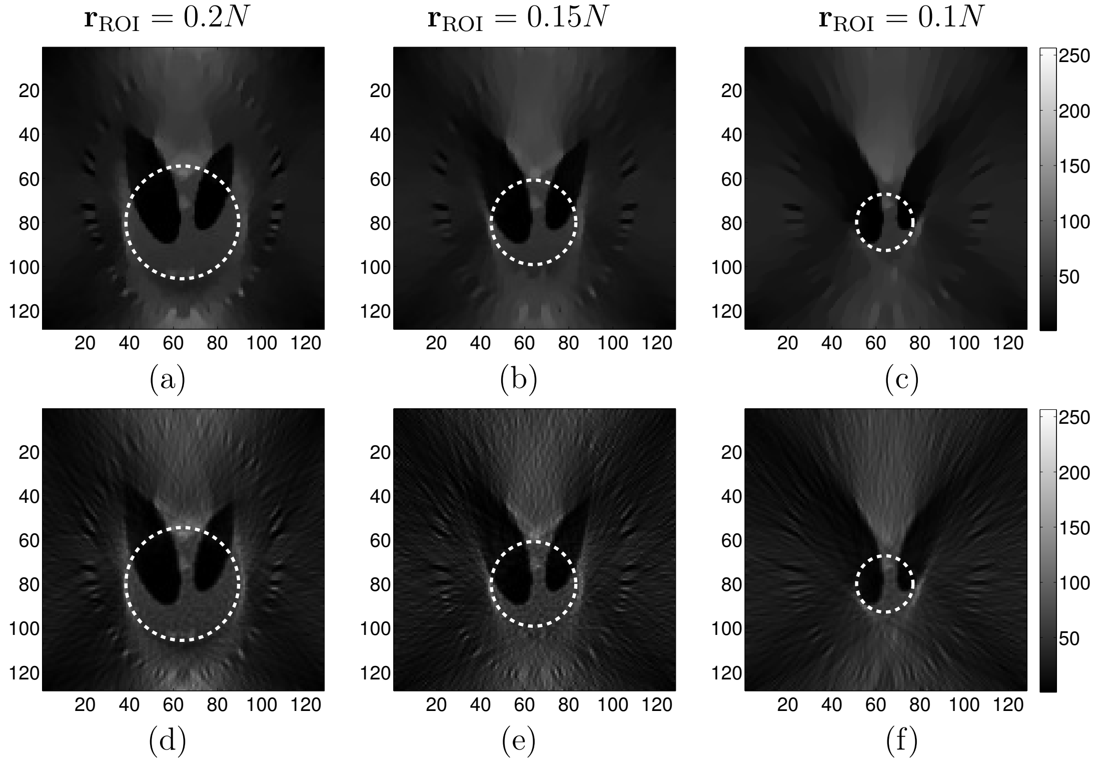

$\mathbf{r}_{\mathrm{ROI}} = 0.2N$
$\mathbf{r}_{\mathrm{ROI}} = 0.15N$
$\mathbf{r}_{\mathrm{ROI}} = 0.1N$
20
40
60
80
100
120
250
200
150
100
50
(a)
(b)
(c)
(d)
(e)
(f)

Figure 6.15: Optimal reconstructions of the Shepp-Logan phantom for the implicit formulation. $\mathbf{r}_{\mathrm{ROI}} = 0.2N$ for (a) and (d), $\mathbf{r}_{\mathrm{ROI}} = 0.15N$ for (b) and (e), $\mathbf{r}_{\mathrm{ROI}} = 0.1N$ for (c) and (f). First row: shealerts and TV. Second row: just shearlets.

The optimal value for the TV regularization parameter ρ is always equal to 1, while for the regularization parameter μ optimal values are 10^{-2}, 10^{-3} and 10^{-4} depending on the ROI radius and the approach. In details, the pure shearlet approach selects $\mu = 10^{-4}$ for all the ROI radii except $\mathbf{r}_{\mathrm{ROI}} = 0.5N$, whose optimal value is $\mu = 10^{-3}$. The shearlets plus TV, instead, needs $\mu = 10^{-4}$ for $\mathbf{r}_{\mathrm{ROI}} = 0.5N$, $0.15N$, $0.1N$, $\mu = 10^{-3}$ for $\mathbf{r}_{\mathrm{ROI}} = 0.3N$, $0.2N$ and $\mu = 10^{-2}$ for $\mathbf{r}_{\mathrm{ROI}} = 0.25N$. In any case, the optimal values for μ are among the smallest ones of the investigated range, while for ρ among the biggest one.

The results in Tables 6.5 and 6.6 are essentially confirmed by the corresponding reconstructed images reported in Figure 6.14 and 6.15. Indeed, on a visual basis, the best reconstruction for all ROI radii is provided by the shearlets plus TV approach: the transition between the inside and the exterior of the ROI is smooth and the features inside the ROI are detected at high contrast (Figures 6.14 and 6.15, first rows). The pure shearlets approach suffers from a mild checkerboard effect, that is more evident for $\mathbf{r}_{\mathrm{ROI}} = 0.3N$, $0.25N$, that corresponds to a higher regularization parameter μ. Anyhow, this does not compromise the accurate re-

$\mathbf{r}_{\text{ROI}} = 0.5N$			$\mathbf{r}_{\text{ROI}} = 0.3N$			$\mathbf{r}_{\text{ROI}} = 0.25N$		
iter	value	maxit	iter	value	maxit	iter	value	maxit
PSNR								
20	25.24	20	20	33.66	20	20	37.73	20
144	15.66	7000	155	21.34	7000	162	23.54	7000
Relative error								
20	0.26	20	20	0.19	20	20	0.17	20
144	0.79	7000	155	0.91	7000	162	0.89	7000

$\mathbf{r}_{\text{ROI}} = 0.2N$			$\mathbf{r}_{\text{ROI}} = 0.15N$			$\mathbf{r}_{\text{ROI}} = 0.1N$		
iter	value	maxit	iter	value	maxit	iter	value	maxit
PSNR								
20	40.41	20	19	42.93	20	19	45.97	20
168	26.85	7000	160	31.32	7000	177	36.68	7000
Relative error								
20	0.18	20	19	0.19	20	19	0.23	20
168	0.85	7000	160	0.74	7000	177	0.66	7000

Table 6.7: Optimal results with the LSCG method. The corresponding reconstructed images are reported in Figure 6.16.

construction of the smaller feature on the bottom (Figure 6.14, second row).

Notice that the reconstruction is qualitatively good for rather small ROI sizes, with no ring artifacts.

6.4 Discussion

The numerical experiments summarized in this Chapter show that both SGP-based and VMILA-based algorithms provide near-exact reconstructions inside the ROI. Roughly, the fine-scale features of the image are the same as in case of reconstruction from non-truncated data, even when the ROI size is rather small compared to the size of the entire object. As a matter of fact, all the objective functions considered are convex with respect to each variable, hence SGP and VMILA are guaranteed to converge to the minimum of the objective function. This occurs regardless of the size and the location of the ROI.

We recall that there is no interest in the reconstructed images outside the ROI. Indeed, these are not reliable since, for every point outside the ROI, only a small fraction of the rays meeting that point are in general available.

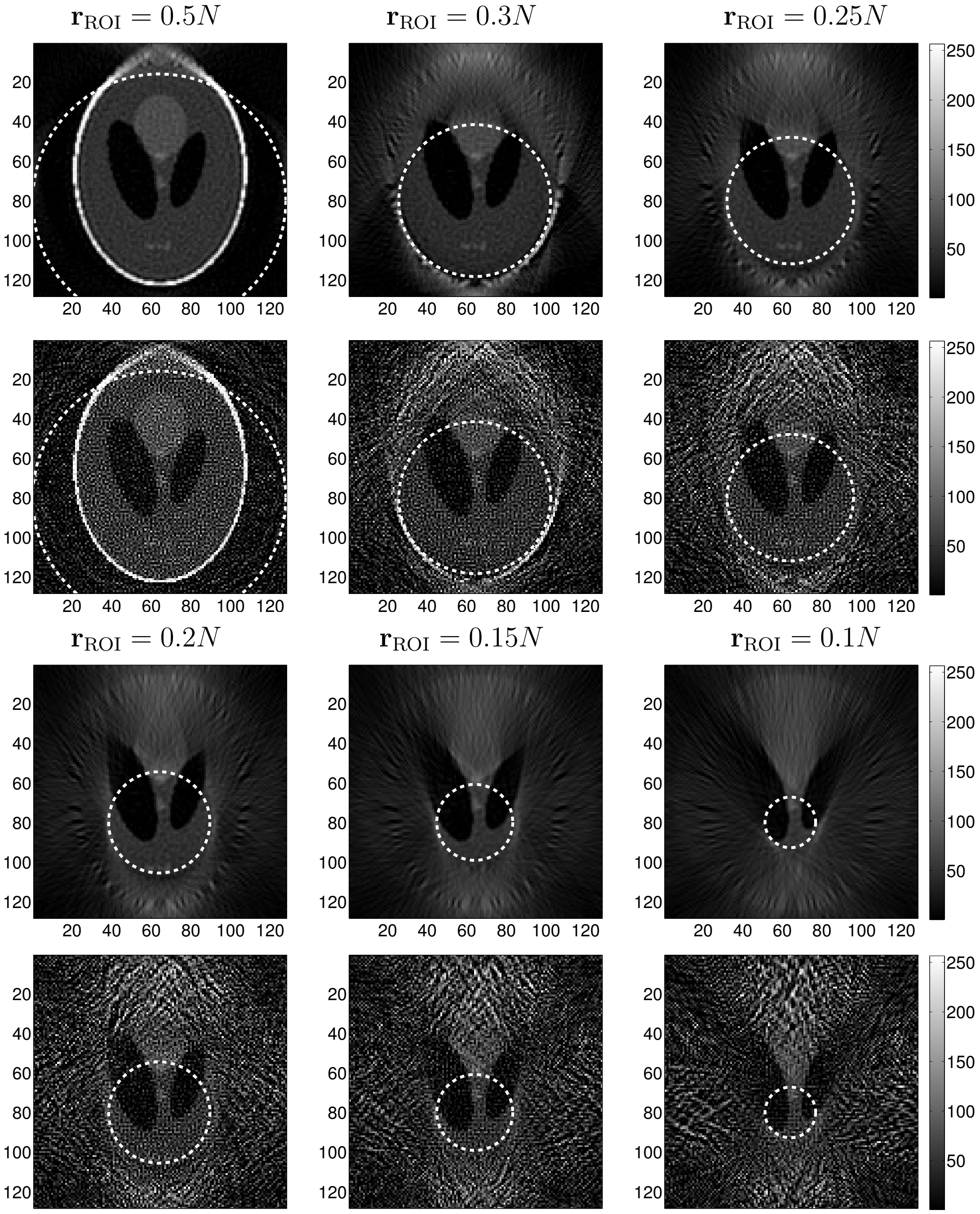


Figure 6.16: Reconstruction of the Shepp-Logan phantom by using the LSCG method. First and third row: maximum number of iteration set to 20. Second and fourth row: maximum number of iteration set to 7000.

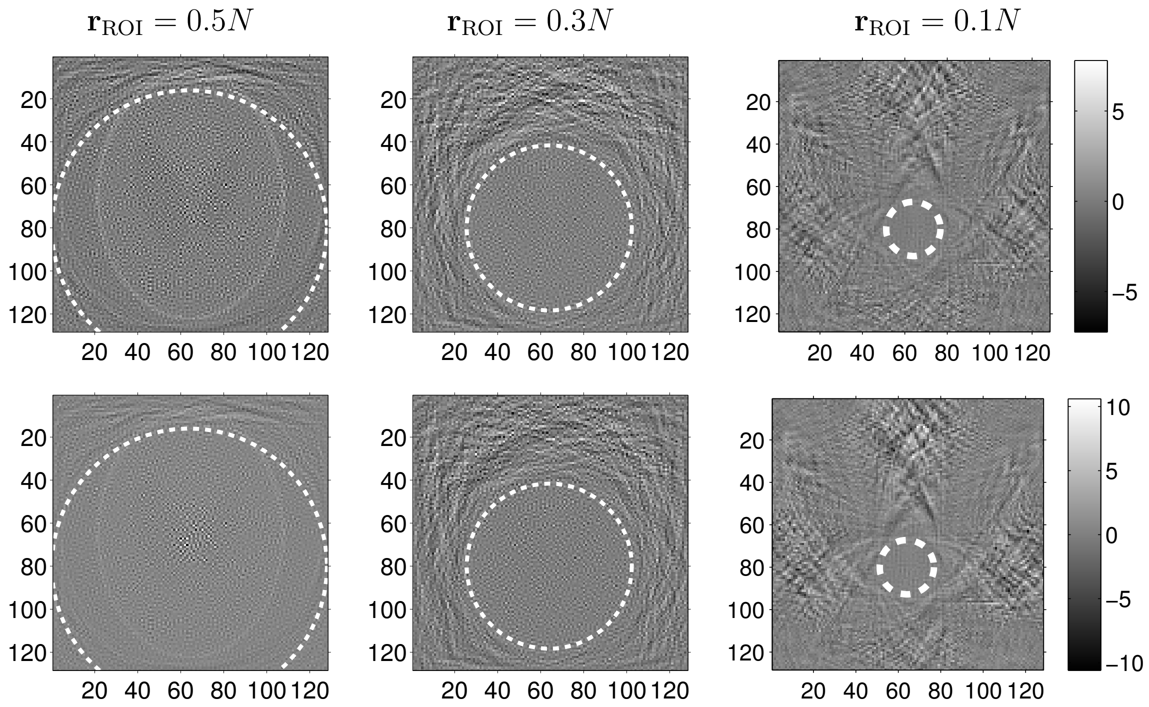

$\mathbf{r}_{\text{ROI}} = 0.5N$
$\mathbf{r}_{\text{ROI}} = 0.3N$
$\mathbf{r}_{\text{ROI}} = 0.1N$
20
40
60
80
100
120
20 40 60 80 100 120
5
0
−5
10
5
0
−5
−10

Figure 6.17: Reconstruction of the Shepp-Logan phantom by using the LSCG method. First row: threshold set to 10^{-5}. Second row: threshold set to 10^{-7}.

The goodness of some of the ROI reconstructions is highlighted by the difference images collected in Figure 6.18. Here, black corresponds to a zero difference, while over- and under-estimated intensities corresponds to warm colours and cool colours, respectively.

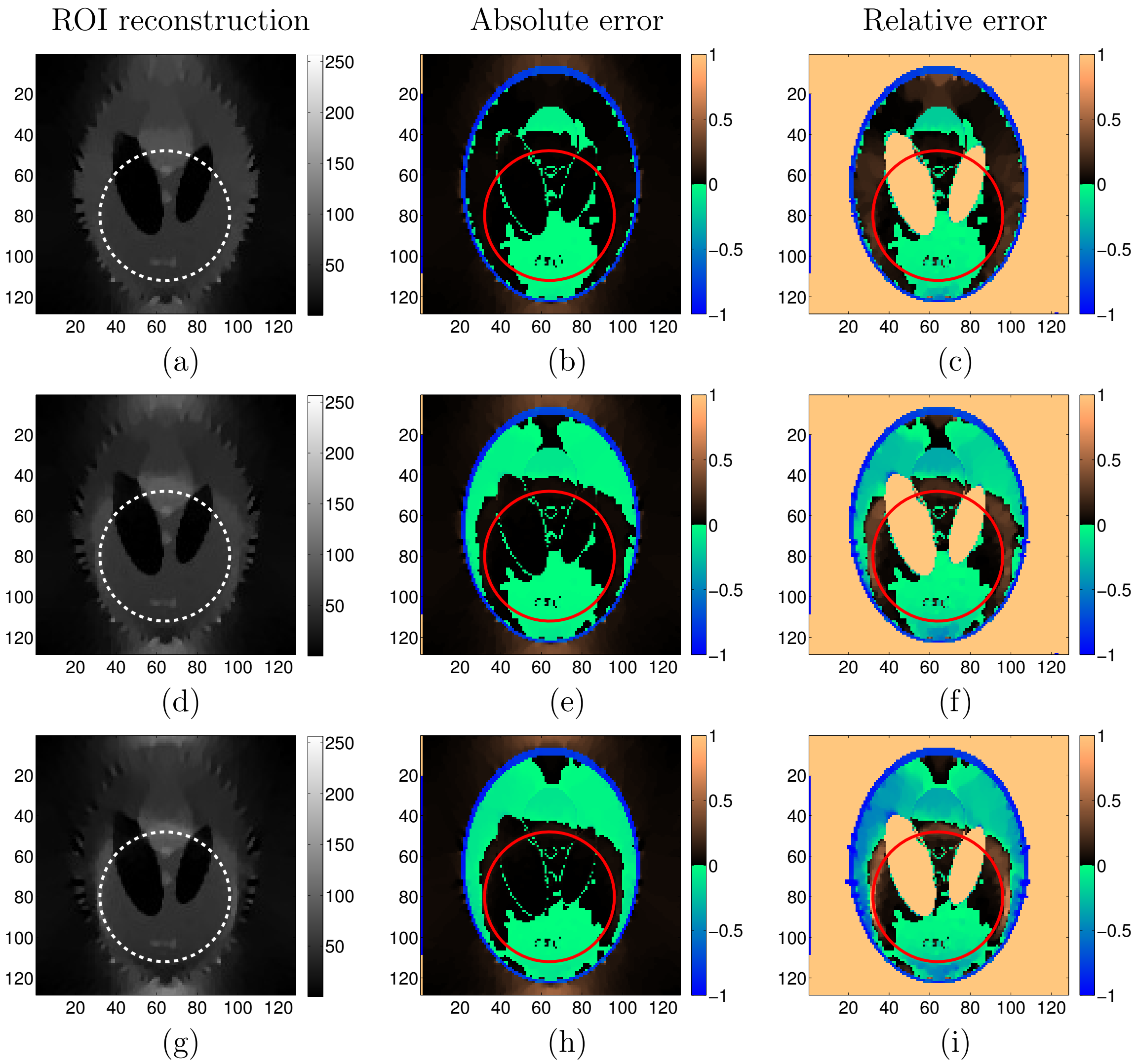
For baseline comparison, we report in Figure 6.16 and in Table 6.7 the ROI reconstruction, and the corresponding values for the figures of merit, obtained with the LSCG approach applied to the system of the normal equation $\mathbf{W}^T\mathbf{MWf} = \mathbf{W}^T\mathbf{y}_0$ corresponding to the data mismatch functional $\|\mathbf{MWf} - \mathbf{y}_0\|_2^2$. All the experiments were performed by exploiting the Matlab default pcg function. We investigated different threshold levels of the residual norm for stopping the iteration, namely 10^{-4}, 10^{-5} and 10^{-7}, and both 20 and 7000 as maximum number of iterations. We used the inverse of the diagonal matrix obtained by selecting the diagonal of the system matrix as preconditioner. In Table 6.7 and Figure 6.16, we report the results obtained with 10^{-4} as threshold level for the residual norm, for both 20 and 7000 as maximum number of iterations. Notice that 10^{-4} is a threshold on the residual norm, namely iterations are stopped when $\|\mathbf{res}^{(k)}\|_2 < 10^{-4}$ is met, where $\mathbf{res}^{(k)} = \mathbf{W}^T\mathbf{MWf}^{(\mathbf{k})} - \mathbf{W}^T\mathbf{y}_0$, for each iteration $k = 0, 1, \ldots$, and the returned solution is the one with minimum residual, according to the early stopping technique for not-regularized functionals. This threshold criterion is different

$\mathbf{r}_{\text{ROI}}$	iter	best PSNR	best Rel. Err.	approach	param
	Smooth implicit - SGP				
0.5	686	32.34	0.12	Sh+TV	$\mu = 10^{-4}, \rho = 0.1$
0.3	1845	40.22	0.10	TV	$\rho = 1$
0.25	1661	44.09	0.08	TV	$\rho = 1$
0.2	2257	45.86	0.09	TV	$\rho = 1$
0.15	2236	47.87	0.11	TV	$\rho = 1$
0.1	2038	51.28	0.12	TV	$\rho = 1$
	Smooth explicit - SGP				
0.5	249	32.48	0.11	Sh+TV	$\mu = 10^{-3}, \rho = 0.1$
0.3	2223	38.92	0.12	Sh+TV	$\mu = 10^{-4}, \rho = 1$
0.25	1283	42.22	0.10	TV	$\rho = 1$
0.2	1110	40.56	0.17	TV	$\rho = 1$
0.15	1160	38.30	0.33	TV	$\rho = 1$
0.1	6	36.02	0.71	ES	
	Nonsmooth implicit - VMILA				
0.5	145	31.43	0.13	Sh+TV	$\mu = 10^{-4}, \rho = 1$
0.3	162	37.41	0.14	Sh+TV	$\mu = 10^{-3}, \rho = 1$
0.25	159	39.95	0.13	Sh+TV	$\mu = 10^{-2}, \rho = 1$
0.2	143	43.17	0.13	Sh+TV	$\mu = 10^{-3}, \rho = 1$
0.15	131	44.72	0.16	Sh+TV	$\mu = 10^{-4}, \rho = 1$
0.1	213	50.48	0.14	Sh+TV	$\mu = 10^{-4}, \rho = 1$

Table 6.8: Overall best results for both implicit and explicit, smooth and non smooth, approaches.

from the one used for SGP and VMILA, where the threshold is on the ROI relative error. The results reported in Table 6.7 show that the ROI reconstructions obtained with 20 as maximum number of iterations exhibit up to 20 dB improvement for the ROI PSNR and a ROI relative error even five times better, especially for the smaller radii, with respect to the ones obtained with 7000. This is confirmed by the corresponding reconstructed images reported in Figure 6.16. However, even if the values of ROI PSNR and ROI relative error could be considered acceptable for the 20 iterations based reconstructions (clearly not for the 7000-based one), there are evidences of checkerboard effect, that slightly compromise the sharp detection of the smaller phantom features. The overall appearance is definitely not as good as the SGP and VMILA TV-based reconstructions.

The 7000-based reconstructions suffer from such a prominent checkerboard effect that the smaller features are indistinguishable. Even if ring artifacts are not evident, the transition from the ROI to the non-ROI is not smooth.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.18: Best reconstruction of the Shepp-Logan phantom (left), absolute error (center) and relative error (right), for the ROI radius $\mathbf{r}_{\text{ROI}} = 0.25N$, with respect to the original object. First row: smooth implicit (SGP). Second row: smooth explicit (SGP). Third row: nonsmooth implicit (VMILA).

The results obtained with 10^{-5} and 10^{-7} as threshold level for the residual norm are utterly worse: the ROI PSNR values are negative and the ROI relative error is 100% for all radii. We report in Figure 6.17 some of these images just to give the idea. Notice that, since no projection onto the feasible set is performed, the range for the reconstructed intensity of the images exceeds the proper one, and this clearly influences the figures of merit values.

Overall, the performance of the LSCG approach is considerably worse than

	$\mathbf{r}_{\text{ROI}} = 0.5N$			$\mathbf{r}_{\text{ROI}} = 0.3N$			$\mathbf{r}_{\text{ROI}} = 0.25N$		
	iter	value	param	iter	value	param	iter	value	param
					PSNR				
Implicit	99	30.90	$\mu = 10^{-2}$	55	34.90	$\mu = 10^{-4}$	83	35.86	$\mu = 10^{-4}$
Explicit	88	30.98	$\mu = 10^{-4}$	2	26.05	$\mu = 10^{-4}$	127	32.64	$\mu = 10^{-4}$
					Relative error				
Implicit	99	0.14	$\mu = 10^{-2}$	55	0.19	$\mu = 10^{-4}$	83	0.21	$\mu = 10^{-4}$
Explicit	88	0.14	$\mu = 10^{-4}$	2	0.53	$\mu = 10^{-4}$	127	0.31	$\mu = 10^{-4}$

	$\mathbf{r}_{\text{ROI}} = 0.2N$			$\mathbf{r}_{\text{ROI}} = 0.15N$			$\mathbf{r}_{\text{ROI}} = 0.1N$		
	iter	value	param	iter	value	param	iter	value	param
					PSNR				
Implicit	90	37.49	$\mu = 10^{-4}$	109	33.61	$\mu = 10^{-4}$	106	32.57	$\mu = 10^{-4}$
Explicit	49	34.04	$\mu = 10^{-4}$	92	32.07	$\mu = 10^{-4}$	71	36.02	$\mu = 10^{-4}$
					Relative error				
Implicit	90	0.25	$\mu = 10^{-4}$	109	0.57	$\mu = 10^{-4}$	106	1.06	$\mu = 10^{-4}$
Explicit	49	0.37	$\mu = 10^{-4}$	92	0.68	$\mu = 10^{-4}$	71	1.31	$\mu = 10^{-4}$

Table 6.9: Optimal results by exploiting the pure shearlet-based regularization, with $\tau_{\text{it}} = 10^{-4}$. The corresponding reconstructed images are reported in Figure 6.19.

SGP and VMILA approaches. This is probably due to the lack of regularization in the objective function.

In fact, the figures of merit from the numerical assessment presented in this Chapter show that TV-based approaches yield the best results, among both implicit and explicit, smooth and non smooth formulations, even if their performance is only slightly better than some of the other approaches. We already pointed out that this may depend on the piecewise constant nature of the phantom, which consists of constant regions with sharp boundaries, and may not hold for more general data. A comparison with a phantom composed by smoothing regions and complex textures, or, even better, with real data, might lead to different results: it is well known that high-resolution medical images are not of bounded variation. A confirmation of this is provided by the analysis carried out in [156]. For the reader's convenience, the overall best results are resumed in Table 6.8.

Beside this, it is remarkable that VMILA applied to the implicit nonsmooth shearlets-based approach (equation (2.18) with $p = 1$) outperforms the smooth approach addressed with SGP (equation (2.18) with $p = 2$), and this is even more evident for the smallest ROI radii $\mathbf{r}_{\text{ROI}} = 0.15N$, $0.1N$. Indeed, both the implicit

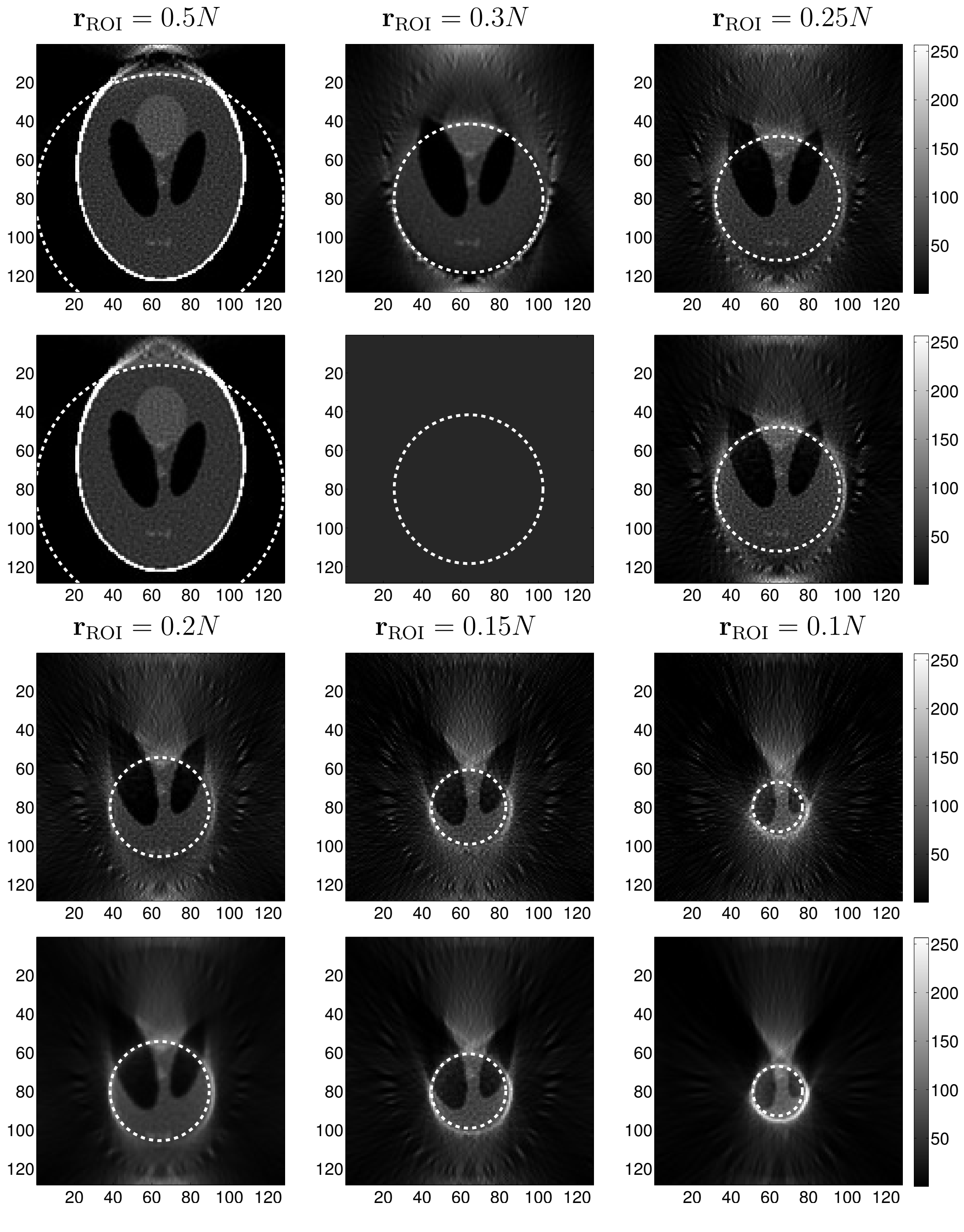

$\mathbf{r}_{\mathrm{ROI}} = 0.5N$
$\mathbf{r}_{\mathrm{ROI}} = 0.3N$
$\mathbf{r}_{\mathrm{ROI}} = 0.25N$
$\mathbf{r}_{\mathrm{ROI}} = 0.2N$
$\mathbf{r}_{\mathrm{ROI}} = 0.15N$
$\mathbf{r}_{\mathrm{ROI}} = 0.1N$
20
40
60
80
100
120
250
200
150
100
50

Figure 6.19: Reconstruction of the Shepp-Logan phantom with pure shearlet-based regularization with $\tau_{\mathrm{it}} = 10^{-4}$. First and third row: smooth implicit. Second and fourth row: smooth explicit.

and the explicit smooth cases have a 100% ROI relative error (Tables 6.2 and 6.4), while the nonsmooth approach investigated with VMILA achieve 22% and 24%, respectively (Table 6.6). This is definitively confirmed by the corresponding images, where no artifacts or even mild checkerboard effect is visible for the nonsmooth reconstruction, while the smooth ones suffer from ring artifacts, low contrast between features and prominent checkerboard effect. The significantly worse performance of the pure shearlets-based approach for the smooth case, both implicit and explicit, may depend on a too tight threshold τ_{it}, since we observed that the relative error decreases up to a certain number of iterations and then goes up steeply. In the results reported in Section 6.2 we set $\tau_{\mathrm{it}} = 10^{-7}$ for the smooth case, but we found out that by posing $\tau_{\mathrm{it}} = 10^{-4}$ (as in VMILA) SGP is able to stop at the right iteration. This provides better results for both the figures of merit, and the image appearance is also improved, except for one case (smooth explicit approach with $\mathbf{r}_{\mathrm{ROI}} = 0.3N$). These results are summarized in Table 6.9 and Figure 6.19.

A possible explanation for the superior performance of the nonsmooth approach might be that the use of the 1-norm leads to the suppression of many small shearlet coefficients in favor of few large shearlet coefficients, which are associated to edges. This allows to separate the structural components of the image from the noise, and this roughly corresponds to denoising. Overall, the results obtained for the nonsmooth formulation, yielded by VMILA, are of great importance because the primary goal of the ROI CT problem is to reconstruct ROIs as small as possible.

Notice also that, in general, VMILA requires a considerably lower number of iterations to converge, especially with respect to the TV-based approaches. This also may depend on the different threshold τ_{it} chosen for stopping the iterations, but, by posing $\tau_{\mathrm{it}} = 10^{-7}$ for VMILA, we get worse results for the pure shearlets-based approach, and slightly better results for the shearlets plus TV approach, at the cost of many more iterations.

Moreover, observe that the safeguard number of 200 inner iterations for VMILA is unnecessary, since criterion (5.19) is always met in only one iteration. Roughly, this means that the computation of the proximity approximation is nearly perfect and the requirement of an inner solver does not compromise neither the convergence speed nor the computational cost.

Finally, observe that the choice of the optimal values for both regularization parameters is consistent to both approach. We found that, by increasing μ, the reconstruction error inside the ROI increases, and, on a visual basis, the transition between the inside and the exterior of the ROI appears less smooth and the features of interest are barely distinguishable. On the contrary, by decreasing ρ, the checkeborad effect overwhelms the visual appearance of the reconstruction.

Chapter 7

Conclusions

This thesis addresses the various aspects involved in the solution of the ROI CT problem, an X-ray based incomplete data imaging acquisition modality. What makes rather challenging this problem is the truncation of the acquired data, that yields a severely ill-posed inverse problem that can not be treated by simply applying CT state-of-the-art techniques.

We proposed two different objective functions, each one with a variable level of regularization, and possibly with a nonsmooth variant to promote sparsity (Chapter 2). Both formulations aim at ensuring a stable reconstruction from truncated data. In particular, the (possibly nonsmooth) regularization term exploits shearlets, a multiscale method emerged in the last decade, for the representation of multivariate data (Chapter 4). Indeed, shearlets enjoy many appealing features that are relevant in CT-like applications, including well localization, accounting for anisotropic structures (*e.g.*, elongated structures such as (soft) edges), and high directional sensitivity, that provide optimal sparsity for images which have discontinuities (edges) along a $\mathscr{C}^2$ curve.

The solution has been addressed within the framework of the statistical approaches for image reconstruction. A first-order iterative minimization method, called SGP, and a proximal-gradient method, called VMILA, have been investigated for the smooth and the nonsmooth formulation, respectively. Both approaches use only the projections meeting the ROI, do not require any prior knowledge of the values of the solution within the ROI or make any assumptions on either the size or the location of the ROI (Chapter 5).

The numerical experiments show that our reconstruction algorithms are very satisfactory for all versions of the objective function, provided an accurate setting of the regularization parameters, notwithstanding the presence of noise. Also, the main structures are recovered accurately, with minimal ring artifacts (Chapter 6). In details, in terms of ROI PSNR and ROI relative error, the best quantitative results are provided by SGP applied to the pure TV-based smooth implicit formulation. We already stressed that this behavior might be dependent on the piecewise constant nature of the phantom and may not hold for more general data. However, from a qualitative viewpoint, the most impressive results are those provided by VMILA applied to the pure shearlet-based nonsmooth implicit formulation. This latter approach is able to consistently outperform the corresponding smooth formulation, both in the implicit and the explicit cases, especially for the smallest radii. In particular, the ROI relative error is even ten times better and the ROI PSNR achieves an improvement up to 15 dB. This is remarkable since the aim of ROI CT is to provide stable and reliable reconstructions for as small as possible radii, in order to consistently reduce the X-ray exposition and the scanning time.

A key element in the design of the digital setting is the forward projection matrix that discretizes the mathematical model of the physical data acquisition process in CT. The goal is a tradeoff between accuracy and computational complexity to achieve a fast, accurate and memory efficient implementation. We proposed a brand new vectorized approach of the state-of-the-art technique called distance-driven, that applies to both the 2D fan-beam and the 3D cone-beam circular CT geometries (Chapter 3). This original contribution goes beyond the ROI CT problem: roughly, every CT-based application can benefit from this approach.

The findings of this doctoral dissertation have a number of important implications for future research, most notably the following.

- An extension of the forward projection matrix implementation to the cone-beam 3D helical geometry is an almost completed work in progress. Consequently, all the machinery developed for ROI CT shall be applied and tested on helical ROI CT problems, matching the possibility to reduce scanning time and exposition to X-rays with the possibility to handle large objects.

- A Spot-compliant object-oriented implementation of the forward projection matrix codes should be profitable to assess CT-like problems in all those situations where the explicit transform operator is needed. Also, a parallel implementation shall provide an effective tool for large scale problems, particularly in the 3D setting.

- The smooth explicit formulation can be tested against the CBGP method. This can be the first step towards the understanding of a new version of VMILA, suitable for the nonsmooth explicit formulation: likely, a block-cyclic nonsmooth version of CBGP. This is necessary to conclude the numerical assessment of the ROI CT problem. The findings of these extensive numerical experiments should enlighten if the 1-norm shearlet-based approach always outperforms the corresponding smooth counterpart, and, possibly, clarify why.

- The case of data corrupted by Poisson noise can be investigated by simply extending the discrepancy term to the Poisson context by means of the Kullback-Leibler divergence. This should be a more suitable model for CT-like applications and both SGP and VMILA have already been extensively tested on the Kullback-Leibler divergence.

- The numerical assessment presented in this work should be carried over to 3D cone-beam data with no further adjustments, since the optimization algorithms used can be applied irrespectively from dimensionality. The only subtle point is the increased computational complexity, for which the parallel implementation of both the forward operator and the shearlet framework shall provide an effective solution. This is a topic of ongoing research.

- In the forthcoming future, it might be interesting investigating whether and how the machinery developed in this thesis works in the case of limited angle ROI CT. This requires a deeper understanding on how the theoretical aspects of the image reconstruction from the limited angle data combine with the truncated data from ROI CT.

Appendix A

Background material on harmonic analysis

In this Appendix, we retrieve some background material from harmonic analysis, including frame theory and representation of locally compact groups, for the reader's convenience. We will only give a glimpse of the general theory, recalling the basic definitions. Most of the material presented here comes from [29, 30, 39, 57] and the reader is referred to them for a deeper discussion on these topics.

A.1 Frame Theory

The first definition of *frame* draws back to 1952 and is due to Duffin and Schaeffer. However, it took several years before the potential of this new theory was realized: in the 1980s, Mallat, Daubechies and Mayer used frames, in place of the traditional notion of orthonormal bases, to analyze wavelets. Indeed, a frame roughly allow a redundant representation of a system of functions that guarantees stability while allowing non-unique decompositions. Since then, frame theory is associated with wavelets, signal and image processing, and data compression.

Definition A.1 ([30])**.** A sequence $(\varphi_i)_{i\in\mathcal{I}}$ in $\mathcal{H}$ is called a *frame* for $\mathcal{H}$, if there exist constants $0 < A \leq B < \infty$ such that

$$A\|x\|^2 \leq \sum_{i\in\mathcal{I}} |\langle x, \varphi_i\rangle|^2 \leq B\|x\|^2 \qquad \forall\, x \in \mathcal{H}.$$

The frame constants A and B are called *lower* and *upper frame bound*, respectively. They are not unique. The *optimal upper frame bound* is the infimum over all upper frame bounds, and the *optimal lower frame bound* is the supremum over all lower frame bounds.

Frames for which the optimal frame bounds coincide play a special role in wavelet and shearlets theory:

Definition A.2 ([30])**.** A sequence $(\varphi_i)_{i\in\mathcal{I}}$ in $\mathcal{H}$ is a *tight frame* if there exists a constant $A > 0$ such that

$$\sum_{i\in\mathcal{I}} |\langle x, \varphi_i\rangle|^2 = A\|x\|^2, \qquad \forall\, x \in \mathcal{H}.$$

If $A = 1$, then $(\varphi_i)_{i\in\mathcal{I}}$ is called *Parseval frame.* A frame is called *equal-norm* if there exists some $c > 0$ such that $\|\varphi_i\| = c$ for all $i \in \mathcal{I}$, and it is *unit-norm* if $c = 1$.

The most simple example of Parseval frame is the sequcence composed by three vectors of the same length in $\mathbb{R}^2$ forming a Mercedes-Benz star, whence the name "Mercedes-Benz frame".

One of the main reasons for applying the frame concept to the wavelet and shearlet theory is that frames serve as an analysis tool. That is, if $(\varphi_i)_{i\in\mathcal{I}}$ in $\mathcal{H}$ is a frame for $\mathcal{H}$, then the so-called *analysis operator* F, defined by

$$\begin{array}{rcl} F\,:\,\mathcal{H} & \longrightarrow & \ell^2(\mathcal{I}) \\ x & \longrightarrow & (\langle x, \varphi_i\rangle)_{i\in\mathcal{I}} \end{array}$$

allows the analysis of data through the study of the associated $(\langle x, \varphi_i\rangle)_{i\in\mathcal{I}}$, usually referred to as *frame coefficients.* The adjoint F^* of the analysis operator is called *synthesis operator* and it is given by:

$$\begin{array}{rcl} F^* : \ell^2(\mathcal{I}) & \longrightarrow & \mathcal{H} \\ (c_i)_{i\in\mathcal{I}} & \longrightarrow & \displaystyle\sum_{i\in\mathcal{I}} c_i\varphi_i. \end{array}$$

By composing F and F^* we obtain the main operator associated with a frame, the so-called *frame operator* S, that basically provides a stable reconstruction process:

$$\begin{array}{rcl} S = F^*F : \mathcal{H} & \longrightarrow & \mathcal{H} \\ x & \longrightarrow & \displaystyle\sum_{i\in\mathcal{I}} \langle x, \varphi_i\rangle\varphi_i. \end{array}$$

Because $(\varphi_i)_{i\in\mathcal{I}}$ is a Bessel sequence, namely there exists a constant $B > 0$ such that $\sum_{i\in\mathcal{I}} |\langle x, \varphi_i\rangle|^2 \leq B\|x\|^2$ for all $x \in \mathcal{H}$, the series that defines S converges unconditionally for all $x \in \mathcal{H}$ (see corollary 2.4 in [30]). Moreover, S is a positive, self-adjoint invertible operator on $\mathcal{H}$ and, in particular,

$$A \cdot I_{\mathcal{H}} \leq S \leq B \cdot I_{\mathcal{H}},$$

where A and B are the frame constants and $I_{\mathcal{H}}$ denotes the identity operator on $\mathcal{H}$. Clearly, this yields $S = I_{\mathcal{H}}$, in the case of a Parseval frame.

The notion of frame operator allow to recover an element $x \in \mathcal{H}$ from its frame coefficients through a so-called *reconstruction formula*, that is:

$$x = \sum_{i \in \mathcal{I}} \langle x, \varphi_i \rangle \, S^{-1} \varphi_i. \tag{A.1}$$

Notice that one of the cases where this decomposition is very useful is when $x \in \mathcal{H}$ represents a signal.
It can be shown that the *canonical dual frame* sequence $(S^{-1}\varphi_i)_{i \in \mathcal{I}}$ forms a frame itself. Hence, regarding a frame as a means for expansion in the system $(\varphi_i)_{i \in \mathcal{I}}$, we are led to:

$$x = \sum_{i \in \mathcal{I}} \langle x, S^{-1}\varphi_i \rangle \, \varphi_i \qquad \forall \, x \in \mathcal{H}. \tag{A.2}$$

The *frame decomposition* (A.2) shows that if $(\varphi_i)_{i \in \mathcal{I}}$ is a frame for $\mathcal{H}$, then every element in $\mathcal{H}$ has a representation as an infinite linear combination of the frame elements. Thus, frames are somehow a "generalized basis". In particular, when the frame $(\varphi_i)_{i \in \mathcal{I}}$ does not constitute a basis, *i.e.*, it is *redundant*, the coefficient sequence $(\langle x, S^{-1}\varphi_i \rangle)_{i \in \mathcal{I}}$ of this expansion can not be unique. It is actually this property that allows for much sparser expansions and, thus, represent a benchmark in many signal processing applications.

Despite this, (A.1) and (A.2) are not practically useful, since this would require either to find the operator S^{-1}, or to compute its action on the whole $(\varphi_i)_{i \in \mathcal{I}}$. One way to bypass this problem is to work only with tight frames, for which the following result holds true:

Proposition A.3 ([30]). *If $(\varphi_i)_{i \in \mathcal{I}}$ is a tight frame with frame bound A, then the canonical dual frame is $(A^{-1}\varphi_i)_{i \in \mathcal{I}}$, and for all $x \in \mathcal{H}$ the following reconstruction formula holds true:*

$$x = \frac{1}{A} \sum_{i \in \mathcal{I}} \langle x, \varphi_i \rangle \, \varphi_i.$$

Tight frames have additional advantages. First and foremost, the canonical dual frame of a tight frame shares the same structure as the frame itself. In particular, if the frame has a shealets (resp., wavelet) structure, the same holds for the canonical dual frame, while the canonical dual frame of a nontight shealets (resp., wavelet) frame might not have the shealets (resp., wavelet) structure. In practice, the tight frame structure allow to control the behavior of the associated canonical dual frame, whereas the complicated structure of both the frame operator and its inverse makes it difficult in the more general nontight case.

A.2 Representation of locally compact groups

Representation theory of groups in harmonic analysis is a matter of interest due to its application to wavelet, and hence shearlet, theory. Many of the aspects of this theory can be studied within the class of locally compact topological groups, even if the most interesting examples belongs to another class of groups, the Lie groups class, that will not be accounted here.
We start by refreshing the definition of locally compact topological group.

Definition A.4 ([57]). A *topological group* is a group G endowed with a topology relative to which the group operations

$$(g, h) \longrightarrow gh, \qquad g \longrightarrow g^{-1}$$

are continuous as maps $G \times G \to G$ and $G \to G$, respectively. G is said *locally compact* if every point has a compact neighborhood. We shall also assume our groups to be Hausdorff.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert spaces and consider a linear and bounded function $\mathcal{T} : \mathcal{H}_1 \to \mathcal{H}_2$. Recall that $\mathcal{T}$ is an isometry if

$$\|\mathcal{T}u\| = \|u\| \quad \forall u \in \mathcal{H}_1.$$

Equivalently, $\mathcal{T}$ is an isometry if and only if $\mathcal{T}^*\mathcal{T} = I_{\mathcal{H}_1}$, given that $\|\mathcal{T}u\|^2 = \langle \mathcal{T}u, \mathcal{T}u\rangle = \langle \mathcal{T}^*\mathcal{T}u, u\rangle$ and $\|u\|^2 = \langle u, u\rangle$. In particular, isometries are injective, but they are not necessarily surjective.

Definition A.5 ([39]). A linear and bounded function$\mathcal{T} : \mathcal{H}_1 \to \mathcal{H}_2$ is called a *unitary map* if it is a bijective isometry.

Notice that if $\mathcal{T}$ is unitary, such is also $\mathcal{T}^{-1}$ and in this case $\mathcal{T}\mathcal{T}^* = I_{\mathcal{H}_2}$. In particular if $\mathcal{H}_1 = \mathcal{H}_2 = \mathcal{H}$, the following set forms a group:

$$\mathcal{U}(\mathcal{H}) = \{\mathcal{T} \text{ bounded and linear } : \mathcal{T} \text{ is unitary}\}.$$

Let now G be a locally compact Hausdorff topological group.

Definition A.6 ([39]). A *unitary representation* of G on the Hilbert space $\mathcal{H}$ is a group homomorphism $\pi : G \longrightarrow \mathcal{U}(\mathcal{H})$ continuous in the strong operator topology, namely for every $g, h \in G$ and $u \in \mathcal{H}$:

(i) $\pi(gh) = \pi(g)\pi(h)$;

(ii) $\pi(g^{-1}) = \pi(g)^{-1} = \pi(g)^*$;

(iii) $g \to \pi(g)u$ is continuous from G to $\mathcal{H}$.

The affine group "$ax + b$", defined on $G = \mathbb{R}^+ \times \mathbb{R}$ endowed with the group operation

$$(a', b')(a, b) = (a'a, a'b + b'),$$

is particularly interesting since it is at the basis for the wavelet representation construction.

Example A.7 (Wavelet representation,[39]). Let G be the "$ax+b$" group and $\mathcal{H} = L^2(\mathbb{R})$. The so-called *wavelet representation* is define to be:

$$\pi(a, b)\psi(x) = \frac{1}{\sqrt{a}}\, \psi\left(\frac{x-b}{a}\right).$$

Notice that $\pi(a, b)$ it is given by the composition of two important unitary maps: the translation operator T_b and the dilation operator D_a, defined by

$$T_b\psi(x) = \psi(x - b), \qquad D_a\psi(x) = \frac{1}{\sqrt{a}}\, \psi\left(\frac{x}{a}\right).$$

Indeed

$$T_b D_a\psi(x) = T_b(D_a\psi)(x) = D_a\psi(x - b) = \frac{1}{\sqrt{a}}\, \psi\left(\frac{x-b}{a}\right).$$

Observe that $T_b T_{b'} = T_{b+b'}$ and $D_a D_{a'} = D_{aa'}$. Hence, in general, T_b and D_a do not commute (*i.e.*, $T_b D_a \neq D_a T_b$):

$$\begin{aligned} D_a T_b\psi(x) = \frac{1}{\sqrt{a}}(T_b\psi)\left(\frac{x}{a}\right) &= \frac{1}{\sqrt{a}}\psi\left(\frac{x}{a} - b\right) \\ &= \frac{1}{\sqrt{a}}\psi\left(\frac{x-ab}{a}\right) = D_a\psi(x - ab) = T_{ab}D_a\psi(x). \end{aligned}$$

Definition A.8 ([39])**.** Let $\mathcal{M}$ be a closed subspace of $\mathcal{H}$. $\mathcal{M}$ is called an *invariant subspace* for the unitary representation π if $\pi(g)\mathcal{M} \subseteq \mathcal{M}$ for all $g \in G$. If there exists a nontrivial invariant subspace for π, then π is called reducible, otherwise π is said *irreducible.*

From the point of view of applications, a unitary representation π of a locally compact group G is particularly useful if it yields a *reproducing formula*, that is, a (weak) reconstruction of a function f in the representation space $\mathcal{H}$ of the form

$$f = \int_G \langle f, \pi(g)\psi\rangle\, \pi(g)\psi\, dg \tag{A.3}$$

where dg is a left Haar measure and $\langle f, \pi(g)\psi\rangle$ originates from the so-called *voice transform*

$$\begin{array}{rcl} V_\psi : \mathcal{H} & \longrightarrow & L^2(G) \\ g & \longrightarrow & V_\psi(f)(g) := \langle f, \pi(g)\psi\rangle. \end{array}$$

In other words, a reproducing formula is a weak reconstruction that holds true for every $f \in \mathcal{H}$, for some admissible $\psi \in \mathcal{H}$, *i.e.*, a $\psi \in \mathcal{H}$ satisfying

$$\int_G |\langle \psi, \pi(g)\psi\rangle|^2 \, dg < \infty,$$

for a given unitary representation π of G on $\mathcal{H}$. In this case, (G, π, ψ) is called a *reproducing system*; otherwise, G is simply said to be a reproducing group. If π is irreducible, this is nothing else but the traditional concept of square integrable representation.
The most famous example holds when $\mathcal{H} = L^2(\mathbb{R}^n)$, with $n \geq 1$ and the ψ, usually called generating or *mother* function, is a *wavelet*, as illustrated in Chapter 4.

Finally, notice that all this can be translated in the discrete framework. Indeed, if a countable collection $(\psi_i)_{i\in\mathcal{I}}$ in a Hilbert space $\mathcal{H}$ is a Parseval frame, then this is equivalent to the reproducing formula $f = \sum_{i\in\mathcal{I}} \langle f, \psi_i\rangle \, \psi_i$ for all $f \in \mathcal{H}$, where the series converges in the norm of $\mathcal{H}$.

Bibliography

[1] Robert Acar and Curtis R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10(6):1217–1229, 1994.

[2] Anders H. Andersen and Avinash C. Kak. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. *Ultrason Imaging*, 6:81–94, 1984.

[3] Jean-Pierre Antoine, Romain Murenzi, and Pierre Vandergheynst. Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns. *Appl. Comput. Harmon. Anal.*, 6(3):314–345, 1999.

[4] Robert Azencott, Bernhard G. Bodmann, Demetrio Labate, Anando Sen, and Daniel Vera. Region-of-interest reconstructions from truncated 3D x-ray projections. *submitted*, 2015.

[5] Cristian Badea and Richard Gordon. Experiments with the nonlinear and chaotic behaviour of the multiplicative algebraic reconstruction technique (MART) algorithm for computed tomography. *Phys. Med. Biol.*, 49:1455–1474, 2004.

[6] Jonathan Barzilai and Jonathan M. Borwein. Two point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[7] Freek J. Beekman and Chris Kamphuis. Ordered subset reconstruction for X-ray CT. *Phys. Med. Biol.*, 46:1835–1844, 2001.

[8] Marcel Beister, Daniel Kolditz, and Willi A. Kalender. Iterative reconstruction methods in X-ray CT. *Physica Medica*, 28:94–108, 2012.

[9] Mario Bertero and Patrizia Boccacci. *Introduction to inverse problems in imaging.* Bristol: IoP, Institute of Physics Publishing, 1998.

[10] Mario Bertero, Henri Lantéri, and Luca Zanni. Iterative image reconstruction: a point of view. In *Mathematical Methods in Biomedical Imaging and*

Intensity-Modulated Radiation Therapy (IMRT), pages 37–63. Edizioni della Normale, Pisa, Italy, Birkhäuser-Verlag, 2008.

[11] Fred C. Billingsley. Digital video processing at JPL. In *Electronic Imaging Techniques I*, pages XV–1–19. Proceedings of SPIE, Eugene B. Turner (ed.), 1965.

[12] André E. M. Bocage. Procede et dispositif de radiographie sur plaque en mouvement. *French Paten FR 236464*, 1922.

[13] Albert Boggess and Francis J. Narcowich. *A first course in wavelets with Fourier analysis.* John Wiley & Sons Inc., Hoboken, NJ, second edition, 2009.

[14] Silvia Bonettini, Ignace Loris, Federica Porta, and Marco Prato. Variable metric inexact line-search based methods for nonsmooth optimization. *SIAM Journal on Optimization (in press)*, 2015.

[15] Silvia Bonettini and Marco Prato. A new general framework for gradient projection method. *arXiv*, page 1406.6601v1, 2014.

[16] Silvia Bonettini and Valeria Ruggiero. An alternating extragradient method for total variation based image restoration from Poisson data. *Inverse Problems*, 27(6):095001, 2011.

[17] Silvia Bonettini, Riccardo Zanella, and Luca Zanni. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, 25:015002, 2009.

[18] Charles Bouman and Ken Sauer. A unified approach to statistical tomography using coordinate descent optimization. *IEEE Trans. Image Process.*, 5:480–492, 1996.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[20] Ronald N. Bracewell. Strip integration in radio astronomy. *Australian Journal of Physics*, 9:198–216, 1956.

[21] Tatiana A. Bubba, Demetrio Labate, Gaetano Zanghirati, and Silvia Bonettini. Numerical assessment of shearlet-based regularization in ROI tomography. *ArXiv e-prints*, nov 2015.

[22] Tatiana A. Bubba, Demetrio Labate, Gaetano Zanghirati, Silvia Bonettini, and Bart Goossens. Shearlet-based regularized ROI reconstruction in fan beam computed tomography. In *SPIE Optics & Photonics, Wavelets And Applications XVI*, volume 9597, page 95970K, San Diego, CA, USA, Aug 10-12 2015.

[23] Thorsten M. Buzug. *Computed Tomography. Principles, design, artifacts, and recent advances. 2nd edition.* Springer, 2008.

[24] Michael H. Byonocore, William R. Brody, and Albert Macovski. A natural pixel decomposition for two-dimensional image reconstruction. *IEEE Trans. Biomed. Eng.*, 28(2):69–78, 1981.

[25] Emmanuel J. Candès and David L. Donoho. Ridgelets: a key to higher-dimensional intermittency? *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, 357(1760):2495–2509, 1999.

[26] Emmanuel J. Candès and David L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Comm. Pure Appl. Math.*, 57(2):219–266, 2004.

[27] Antoine Chambolle. An algorithm for Total Variation minimization and applications. *J. Math. Imag. Vis.*, 20:89–97, 2004.

[28] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40:120–145, 2011.

[29] Ole Christensen. *An introduction to frames and Riesz bases.* Boston, MA: Birkhäuser, 2003.

[30] Ole Christensen. B-spline generated frames. In *Four short courses on harmonic analysis. Wavelets, frames, time-frequency methods, and applications to signal and image analysis*, pages 51–86. Basel: Birkhäuser, 2010.

[31] Charles K. Chui. *An introduction to wavelets.* Boston, MA: Academic Press, 1992.

[32] Robert Cierniak. *X-Ray Computed Tomography in Biomedical Engineering.* Springer, 2011.

[33] Rolf Clackdoyle and Michel Defrise. Tomographic reconstruction in the 21st century. Region-of-interest reconstruction from incomplete data. *IEEE Signal Processing*, 60:60–80, 2010.

[34] Rolf Clackdoyle and Frederic Noo. A large class of inversion formulae for the 2-D Radon transform of functions of compact support. *Inverse Problems*, 20:1281–1291, 2004.

[35] Rolf Clackdoyle, Frederic Noo, Junyu Guo, and John Roberts. Quantitative reconstruction from truncated projections in classical tomography. *IEEE Trans. Nuclear Science*, 51:2570–2578, 2004.

[36] Flavia Colonna, Glenn Easley, Kanghui Guo, and Demetrio Labate. Radon transform inversion using the shearlet representation. *Appl. Comput. Harmon. Anal.*, 29(2):232–250, 2010.

[37] Allan M. Cormack. Representation of a function by its line integrals, with some radiological applications. *J. Appl. Phys.*, 34(9):2722–2727, 1963.

[38] Allan M. Cormack. Representation of a function by its line integrals, with some radiological applications, II. *J. Appl. Phys.*, 35(9):2908–2913, 1964.

[39] Stephan Dahlke, Filippo De Mari, Philipp Grohs, and Demetrio Labate. *Harmonic and Applied Analysis: From Groups to Signals.* Boston, MA: Birkhäuser, 2015.

[40] Stephan Dahlke, Gitta Kutyniok, Gabriele Steidl, and Gerd Teschke. Shearlet coorbit spaces and associated banach frames. *Appl. Comput. Harmon. Anal.*, 27(2):195–214, 2009.

[41] Stephan Dahlke, Gabriele Steidl, and Gerd Teschke. The continuous shearlet transform in arbitrary space dimensions. *J. Fourier Anal. Appl.*, 16(3):340–364, 2010.

[42] Stephan Dahlke, Gabriele Steidl, and Gerd Teschke. Multivariate shearlet transform, shearlet coorbit spaces and their structural properties. In *Shearlets. Multiscale analysis for multivariate data.*, pages 105–144. Boston, MA: Birkhäuser, 2012.

[43] Yu-Hong Dai and Roger Fletcher. On the asymptotic behaviour of some new gradient methods. *Math. Programming*, 103:541–559, 2005.

[44] Bruno De Man and Samit Basu. Distance-driven projection and backprojection. *Nuclear Science Symposium Conference Record, IEEE*, 3:1477–1480, 2002.

[45] Bruno De Man and Samit Basu. Distance-driven projection and backprojection in three dimensions. *Physics in Medicine and Biology*, 7:2463–2475, 2004.

[46] Bruno De Man and Jeffrey A. Fessler. Statistical iterative reconstruction for X-ray computed tomography. In *Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems*, pages 113–140. Medical Physics Publishing, 2010.

[47] Minh N. Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. Image Process.*, 14(4):2091–2106, 2005.

[48] Glenn Easley, Demetrio Labate, and Wang-Q Lim. Sparse directional image representations using the discrete shearlet transform. *Appl. Comput. Harmon. Anal.*, 25:25–46, 2008.

[49] Michael Elad. *Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing.* Springer, New York, 2010.

[50] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems.* Kluwer Academic Publishers, Dordrecht, 1996.

[51] Hakan Erdogan and Jeffrey A. Fessler. Ordered subsets algorithms for transmission tomography. *Phys. Med. Biol.*, 44:2835–2851, 1999.

[52] Adel Faridani, David V. Finch, Erik L. Ritman, and Kennan T. Smith. Local tomography II. *SIAM J. App. Math.*, 57:1095–1127, 1997.

[53] Lee A. Feldkamp, Lloyd C. Davis, and James W. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America*, pages 612–619, 1984.

[54] Jeffrey A. Fessler. Penalized weighted least-squares image reconstruction for positron emission tomography. *IEEE Trans. Med. Imaging*, 13:290–300, 1994.

[55] Jeffrey A. Fessler and Scott D. Booth. Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction. *IEEE Trans. Med. Imag.*, 8(5):688–699, 1999.

[56] Roger Fletcher. On the Barzilai-Borwein method. Technical Report NA/207, Technical Report, Department of Mathematics, University of Dundee, Dundee, UK, 2001.

[57] Gerald B. Folland. *A course in abstract harmonic analysis.* Boca Raton, FL: CRC Press, 1995.

[58] Gerald B. Folland. *Real analysis. Modern techniques and their applications. 2nd edition.* New York, NY: Wiley, 2nd ed. edition, 1999.

[59] Giacomo Frassoldati, Gaetano Zanghirati, and Luca Zanni. New adaptive stepsize selections in gradient methods. *J. Industrial and Management Optim*, 4:299–312, 2008.

[60] Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *J. Theor. Biol.*, 36:105–117, 1972.

[61] Bart Goossens, Demetrio Labate, and Bernhard G. Bodmann. Region-of-interest computed tomography by regularity-inducing convex optimization. *submitted*, 2016.

[62] Richard Gordon, Robert Bender, and Gabor T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. theor. Biol.*, 29:471–481, 1970.

[63] Luigi Grippo, Francesco Lampariello, and Stefano Lucidi. A nonmonotone line-search technique for NewtonÕs method. *SIAM J. Numer. Anal.*, 23:707–716, 1986.

[64] Charles W. Groetsch. *Inverse problems in the mathematical sciences.* Springer Vieweg, Braunschweig, 1993.

[65] Philipp Grohs. Shearlets and microlocal analysis. In *Shearlets. Multiscale analysis for multivariate data*, pages 39–67. Boston, MA: Birkhäuser, 2012.

[66] Kanghui Guo, Gitta Kutyniok, and Demetrio Labate. Sparse multidimensional representations using anisotropic dilation and shear operators. In *Wavelets and splines: Athens 2005. Papers based on lectures presented at the international conference on interactions between wavelets and splines, Athens, GA, USA, May 16–19, 2005. In honor of Professor Charles K. Chui on the occasion of his 65th birthday*, pages 189–201. Brentwood: Nashboro Press, 2006.

[67] Kanghui Guo and Demetrio Labate. Analysis and identification of multidimensional singularities using the continuous shearlet transform. In *Shearlets. Multiscale analysis for multivariate data.*, pages 69–103. Boston, MA: Birkhäuser, 2012.

[68] Kanghui Guo and Demetrio Labate. The construction of smooth Parseval frames of shearlets. *Math. Model. Nat. Phenom.*, 8(1):82–105, 2013.

[69] Kanghui Guo, Demetrio Labate, Wang-Q Lim, Guido Weiss, and Edward Wilson. Wavelets with composite dilations and their MRA properties. *Appl. Comput. Harmon. Anal.*, 20(2):202–236, 2006.

[70] Yan Guorui, Jie Tian, Zhu Shouping, Chenghu Qin, Yakang Dai, Yang Fei, Dong Di, and Wu Ping. Fast Katsevich algorithm based on GPU for helical cone-beam computed tomography. *IEEE Transactions on Information Technology in Biomedicine*, 14(4):1053–1061, 2010.

[71] Benoit Hamelin, Yves Goussard, Jean-Pierre Dussault, Guy Cloutier, Gilles Beaudoin, and Gilles Soulez. Design of iterative ROI transmission tomography reconstruction procedures and image quality analysis. *Medical Physics*, 37(9):4577–4589, 2010.

[72] Bin Han, Gitta Kutyniok, and Zuowei Shen. Adaptive multiresolution analysis structures and shearlet systems. *SIAM J. Numer. Anal.*, 49(5):1921–1946, 2011.

[73] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion.* SIAM. Monographs on mathematical modeling and computation, 1998.

[74] Kenneth M. Hanson and George W. Wecksung. Local basis-function approach to computed tomography. *Applied Optics*, 24(23):4028–4039, 1985.

[75] Sören Häuser and Gabriele Steidl. Fast finite shearlet transform: a tutorial. *ArXiv*, (1202.1773), 2014.

[76] Gabor T. Herman. *Image reconstruction from projections: the fundamentals of computerized tomography.* Academic Press, New York, 1980.

[77] Gabor T. Herman. Tomography. In *Handbook of Mathematical Methods in Imaging. 2nd edition*, pages 801–845. Springer New York, 2015.

[78] Gabor T. Herman and Arnold Lent. Iterative reconstruction algorithms. *Computers in Biology and Medicine*, 6:273–294, 1976.

[79] Eugenio Hernández and Guido Weiss. *A first course on wavelets.* Boca Raton, FL: CRC Press, 1996.

[80] Magnus R. Hestenes and Eduard Stiefel. Methods of Conjugate Gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 6:409–436, 1952.

[81] Godfrey N. Hounsfield. A method of and apparatus for examination of a body by radiation such as X or gamma radiation. *Patent Specification n. 1283915, The Patent Office, London*, 1972.

[82] Godfrey N. Hounsfield. Computerised transverse axial scanning (tomography): Part 1. description of system. *Br. J. Radiol.*, 46:1016–1022, 1973.

[83] Jiang Hsieh. *Computed Tomography. From Photon Statistics to Modern Cone-Beam CT.* SPIE Press. Wiley-Interscience, 2009.

[84] Jiang Hsieh, Brian Nett, Zhou Yu, Ken Sauer, Jean-Baptiste Thibault, and Charles A. Bouman. Recent advances in CT image reconstruction. *Curr. Radiol. Rep.*, 1(1):39–51, 2013.

[85] Xin Jin, Alexander Katsevich, Hengyong Yu, Ge Wang, Liang Li, and Zhiqiang Chen. Interior tomography with continuous singular value decomposition. *IEEE Trans Med Imaging.*, 31:2108–2119, 2012.

[86] Peter M. Joseph. An improved algorithm for reprojecting rays through pixel images. *IEEE Trans. Med. Im.*, 1(3):192–196, 1982.

[87] Stefan Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bull. Internat. Acad. PolonSci Lettres A*, pages 335–357, 1937.

[88] Chris Kamphuis and Freek J. Beekman. Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm. *IEEE Trans. Med. Imaging*, 17:1101–1105, 1998.

[89] Alexander Katsevich. Theoretically exact filtered backprojection-type inversion algorithm for spiral CT. *SIAM Journal on Applied Mathematics*, 62(6):2012–2026, jan 2002.

[90] Alexander Katsevich. An improved exact filtered backprojection algorithm for spiral computed tomography. *Advances in Applied Mathematics*, 32:681–697, 2004.

[91] Eugene Katsevich, Alexander Katsevich, and Ge Wang. Stability of the interior problem for polynomial region of interest. *Inverse Problems*, 28:065022, 2012.

[92] Joseph B. Keller. Inverse problems. *The American Mathematical Monthly*, 83(2):107–118, 1976.

[93] Jean Kieffer. X-ray device and method of technique. *US Patent n. 1954321*, 1929.

[94] Jung K. Kim, Zhengya Zhang, and Jeffrey A. Fessler. Hardware acceleration of iterative image reconstruction for X-ray computed tomography. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12158307:1697–1700, 2011.

[95] Kyeong Y. Kim, Soo M. Kim, Dong S. Lee, and Jae S. Lee. Comparative evaluation of separable footprint and distance-driven methods for PET reconstruction. *J. Nucl. Med.*, 54(2):2130, 2013.

[96] Nick Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Appl. Comput. Harmon. Anal.*, 10(3):234–253, 2001.

[97] Esther Klann, Eric Todd Quinto, and Ronny Ramlau. Wavelet methods for a weighted sparsity penalty for region of interest tomography. *Inverse Problems*, 31:025001, 2015.

[98] Venkateswaran P. Krishnan and Eric Todd Quinto. Microlocal analysis in tomography. In *Handbook of Mathematical Methods in Imaging. 2nd edition*, pages 847–902. Springer New York, 2015.

[99] Gitta Kutyniok and Demetrio Labate. Construction of regular and irregular shearlet frames. *J. Wavelet Theory and Appl.*, 1:1–10, 2007.

[100] Gitta Kutyniok and Demetrio Labate. Resolution of the wavefront set using continuous shearlets. *Trans. Am. Math. Soc.*, 361(5):2719–2754, 2009.

[101] Gitta Kutyniok and Demetrio Labate. Introduction to shearlets. In *Shearlets. Multiscale analysis for multivariate data.*, pages 1–38. Boston, MA: Birkhäuser, 2012.

[102] Gitta Kutyniok and Demetrio Labate, editors. *Shearlets*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2012. Multiscale analysis for multivariate data.

[103] Gitta Kutyniok, Jakob Lemvig, and Wang-Q Lim. Shearlets and optimally sparse approximations. In *Shearlets. Multiscale analysis for multivariate data.*, pages 145–197. Boston, MA: Birkhäuser, 2012.

[104] Gitta Kutyniok and Tomas Sauer. Adaptive directional subdivision schemes and shearlet multiresolution analysis. *SIAM J. Math. Anal.*, 41(4):1436–1471, 2009.

[105] Gitta Kutyniok, Morteza Shahram, and Xiaosheng Zhuang. ShearLab: a rational design of a digital parabolic scaling algorithm. *SIAM J. Imaging Sci.*, 5(4):1291–1332, 2012.

[106] Demetrio Labate, Wang-Q. Lim, Gitta Kutyniok, and Guido Weiss. Sparse multidimensional representation using shearlets. In San Diego CA, editor, *SPIE Proc.*, volume 5914 of *SPIE*, pages 254–262, Bellingham, 2005.

[107] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Proc. SIGGRAPH 1994 (Orlando)*, pages 451–458, 1994.

[108] Germana Landi and Elena Loli Piccolomini. A Total Variation regularization strategy in dynamic MRI sequences. *Optimization Methods and Software*, 20(4-5):545–558, 2005.

[109] Kenneth Lange and Richard Carson. EM reconstruction algorithms for emission and transmission tomography. *J. Comput. Assist. Tomogr.*, 8:306–316, 1984.

[110] Henri Lantéri, Muriel Roche, Olga Cuevas, and Claude Aime. A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints. *Signal Process.*, 81:945–974, 2001.

[111] Erwan Le Pennec and Stéphane Mallat. Sparse geometric image representations with bandelets. *IEEE Trans. Image Process.*, 14(4):423–438, 2005.

[112] Soo-Jin Lee. Accelerated coordinate descent methods for bayesian reconstruction using ordered subsets of projection data. *Proc. SPIE Conf. Math. Model.*, 4121:170–181, 2000.

[113] Yulia Levakhina. *Three-Dimensional Digital Tomosynthesis Iterative Reconstruction, Artifact Reduction and Alternative Acquisition Geometry.* Springer, 2014.

[114] R. M. Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Physics in Medicine and Biology*, 37(3):705–716, 1992.

[115] Wang-Q Lim. The discrete shearlet transform: a new directional transform and compactly supported shearlet frames. *IEEE Trans. Image Process.*, 19:1166–1180, 2010.

[116] Shih-Chung B. Lo. Strip and line path-integrals with a square pixel matrix - a unified theory for computational CT projections. *IEEE Trans. Med. Imaging*, 7(4):355–363, 1988.

[117] Elena Loli Piccolomini and Fabiana Zama. Regularization algorithms for image reconstruction from projections. In F. Pavese D. Richter eds. P. Ciarlini, M.G. Cox, editor, *Series on Advances for Applied Science*, volume 45 of *World Scientific*, 1996.

[118] Elena Loli Piccolomini and Fabiana Zama. The conjugate gradient regularization method in Computed Tomography problems. *Applied Mathematics and Computation*, 102(1):87–99, 1999.

[119] Yong Long and Jeffrey A. Fessler. 3D forward and back-projection for X-ray CT using separable footprints with trapezoid functions. *Proceedings of The First International Conference on Image Formation in X-Ray Computed Tomography*, pages 216–219, 2010.

[120] Yong Long, Jeffrey A. Fessler, and James M. Balter. 3D forward and back-projection for X-ray CT using separable footprints. *IEEE Trans. Med. Imag.*, 29(11):1839–1850, 2010.

[121] Stéphane Mallat. *A wavelet tour of signal processing. The sparse way. 3rd edition.* Amsterdam: Elsevier/Academic Press, 3rd ed. edition, 2009.

[122] Stephen H. Manglos, George M. Gagne, Andrzej Krol, F. Deaver Thomas, and Rammohan Narayanaswamy. Transmission maximum-likelihood reconstruction with ordered subsets for cone beam CT. *Phys. Med. Biol.*, 40:1225–1241, 1995.

[123] Samuel Matej, Jeffrey A. Fessler, and Ivan G. Kazantsev. Iterative tomographic image reconstruction using Fourier-based forward and back-projectors. *IEEE Trans. Med. Imaging*, 23(4):401–412, 2004.

[124] Klaus Mueller, Roni Yagel, and John J. Wheller. A fast and accurate projection algorithm for 3D cone-beam reconstruction with the algebraic reconstruction technique (ART). *Proc. SPIE*, 3336:1–9, 1998.

[125] Heinrich N. Mülthei. Iterative continuous maximum likelihood reconstruction methods. *Math. Methods Appl. Sci.*, 15:275–286, 1993.

[126] Menahem Nassi, William R. Brody, Barry P. Medoff, and A. Macovski. Iterative reconstruction-reprojection: An algorithm for limited data cardiac-computed tomography. *IEEE Transactions on Biomedical Engineering*, BME-29(5):333–341, 1982.

[127] Frank Natterer. *The Mathematics of Computerized Tomography.* SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA (USA), 2001.

[128] Frank Natterer and Frank Wübbeling. *Mathematical methods in image reconstruction.* SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA (USA), 2001.

[129] Kati Niinimaki, Samuli Siltanen, and Ville Kolehmainen. Bayesian multiresolution method for local tomography in dental X-ray imaging. *Physics in Medicine and Biology*, 52:6663–6678, 2007.

[130] Frederic Noo, Rolf Clackdoyle, and Jed Pack. A two-step Hilbert transform method for 2D image reconstruction. *Physics in Medicine and Biology*, 49:3903–3923, 2004.

[131] Frederic Noo, Michel Defrise, Rolf Clackdoyle, and Hiroyuki Kudo. Image reconstruction from fan-beam projections on less than a short scan. *Physics in Medicine and Biology*, 47(14):2525–2546, 2002.

[132] Terry Peters. Algorithms for fast back- and re-projection in computed tomography. *IEEE Trans. Nucl. Sci.*, 28:3641–3647, 1981.

[133] Ernst Pohl. Method and apparatus for making Roentgen projections. *EU patent n. 332496*, 1927.

[134] Marco Prato, Andrea La Camera, Silvia Bonettini, and Mario Bertero. A convergent blind deconvolution method for post-adaptive-optics astronomical imaging. *Inverse Problems*, 29(6):065017, 2013.

[135] Elena Prestini. *The evolution of applied harmonic analysis. Models of the real world.* Birkhäuser Boston, 2004.

[136] Eric Todd Quinto. An antroduction to X-ray tomography and Radon transforms. *Proc. of Symposia in Appl. Math.*, 63:1–23, 2006.

[137] Eric Todd Quinto. Local algorithms in exterior tomography. *J. Comput. Appl. Math.*, 199(1):141–148, 2007.

[138] Eric Todd Quinto and Ozan Öktem. Local tomography in electron microscopy. *SIAM J. Appl. Math.*, 68(5):1282–1303, 2008.

[139] Johann Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Ber. vor Sächs. Akad. Wiss., Leipzig Math. Phys.*, 69:262–277, 1917.

[140] Wilhelm C. Röentgen. Eine neue art von strahlen (II. mittheilung). 1895.

[141] Wilhelm C. Röentgen. Ueber eine neue art von strahlen (vorläufige mittheilung). 1895.

[142] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[143] Walter Rudin. *Real and complex analysis. 3rd edition.* New York, NY: McGraw-Hill, 3rd ed. edition, 1987.

[144] Ken Sauer and Charles Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Trans. Sig. Proc.*, 41:534–548, 1993.

[145] Otmar Scherzer. *Handbook of Mathematical Methods in Imaging. 2nd edition.* Springer New York, 2015.

[146] William F. Schreiber. Picture coding. *Proc. IEEE*, 55, 1967.

[147] Lawrence A. Shepp and Yuval Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans Med Imaging*, 1:113–122, 1982.

[148] Robert L. Siddon. Fast calculation of the exact radiological path length for a three-dimensional CT array. *Med. Phys.*, 12(2):252–255, 1985.

[149] Emil Sidky, David N. Kraemer, Erin G. Roth, Christer Ullberg, Ingrid S. Reiser, and Xiaochuan Pan. Analysis of iterative region-of-interest image reconstruction for X-ray computed tomography. *Journal of Medical Imaging*, 1:031007, 2014.

[150] Jean-Baptiste Thibault, Ken Sauer, Charles Bouman, and Jiang Hsieh. Three-dimensional statistical approach to improved image quality for multislice helical CT. *Med. Phys.*, 34:4526–4544, 2007.

[151] Andrey N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Sov. Math. Dokl.*, 4:1035–1038, 1963.

[152] Arthur C. Upton. The first hundred years of radiation research: what have they taught us? *Environ. Res.*, 59(1):36–48, 1992.

[153] Alessandro Vallebona. Una modalità di tecnica per la dissociazione radiografica delle ombre applicata allo studio del cranio. *Radiol. Med.*, 17:1090–1097, 1932.

[154] G.M.P. Van Kempen and L.J. Van Vliet. The influence of the regularization parameter and the first estimate on the performance of Tikhonov regularized non-linear image restoration algorithms. *J. Microsc.*, 198(Pt. 1):63–75, 2000.

[155] René Van Tiggelen. In search for the third dimension: from radiostereoscopy to three-dimensional imaging. *Belgian Journal of Radiology*, 85(5):266–270, 2002.

[156] Bert Vandeghinste, Bart Goossens, Roel Van Holen, Christian Vanhove, Aleksandra Pižurica, Stefaan Vandenberghe, and Steven Staelens. Iterative CT reconstruction using shearlet-based regularization. *IEEE Trans. Nuclear Science*, 5:3305–3317, 2013.

[157] Curtis R. Vogel. *Computational methods for inverse problems.* SIAM Philadelphia, 2002.

[158] Guido Weiss and Edward N. Wilson. The mathematical theory of wavelets. In *Twentieth century harmonic analysis–A celebration. Proceedings of the NATO Advanced Study Institute, Il Ciocco, Italy, July 2–15, 2000*, pages 329–366. Dordrecht: Kluwer Academic Publishers, 2001.

[159] Stephen Wright, Robert D. Nowak, and Màrio Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Proc.*, 57:2479–2493, 2009.

[160] Meng Wu and Jeffrey A. Fessler. GPU acceleration of 3D forward and backward projection using separable footprints for X-ray CT image reconstruction. *The 11th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2011.

[161] Fang Xu, Wei Xu, Mel Jones, Bettina Keszthelyi, John Sedat, David Agard, and Klaus Mueller. On the efficiency of iterative ordered subset reconstruction algorithms for acceleration on GPUs. *Comput. Methods Programs. Biomed.*, 98:261–270, 2010.

[162] Jiansheng Yang, Ming Jiang, and Ge Wang. Supplemental analysis on compressed sensing based interior tomography. *Physics in Medicine and Biology*, 54:N425–N432, 2009.

[163] Jiansheng Yang, Hengyong Yu, Ming Jiang, and Ge Wang. High-order total variation minimization for interior tomography. *Inverse Problems*, 26(3):035013, 2010.

[164] Pengfei Yang, Azam Ahmed, Sebastian Schafer, David Niemann, Beverly Aagaard-Kienitz, Kevin Royalty, and Charles Strother. Low-dose volume-of-interest C-Arm CT imaging of intracranial stents and flow diverters. *AJNR Am. J. Neuroradiol.*, 2015.

[165] Hengyong Yu and Ge Wang. Compressed sensing based interior tomography. *Physics in Medicine and Biology*, 54:2791–2805, 2009.

[166] Zhou Yu, Jean-Baptiste Thibault, Charles Bouman, Ken Sauer, and Hsieh Jiang. Fast model-based X-ray CT reconstruction using spatially nonhomogeneous ICD optimization. *IEEE Trans. Image Process.*, 20:161–175, 2011.

[167] Bin Zhang and G. Lawrence Zeng. Two dimensional iterative region of iterest reconstruction from truncated projection data. *Medical Physics*, 34(3):935–944, 2007.

[168] Yingying Zhang-O'Connor and Jeffrey A. Fessler. Fourier-based forward and back-projectors in iterative fan-beam tomographic image reconstruction. *IEEE Trans. Med. Imaging*, 25(5):582–589, 2006.

[169] Hongqing Zhu, Huazhong Shu, Jian Zhou, and Limin Luo. A weighted least squares PET image reconstruction method using iterative coordinate descent algorithms. *Proc. IEEE Nuclear Science Symp. Conf. Record*, 6:3380–3384, 2004.

[170] Weihua Zhuang, Sanjay S. Gopal, and T. J. Hebert. Numerical evaluation of methods for computing tomographic projections. *IEEE Trans. Nucl. Sci.*, 41(4):1660–1665, 1994.

[171] Bernard G. Ziedses des Plantes. Eine neue methode zur differenzierung in der röntgenographie (planigraphie). *Acta Radiol.*, 13:182–192, 1932.

[172] Andy Ziegler, Tim Nielsen, and Michael Grass. Iterative reconstruction of a region of interest for transmission tomography. *Medical Physics*, 35(4):1317–1327, 2008.

[173] Yu Zou, Xiaochuan Pan, and Emil Sidky. Image reconstruction in regions-of-interest from truncated projections in a reduced fan-beam scan. *Phys. Med. Biol.*, 50:13–28, 2005.

Ringraziamenti

A conclusione di questo percorso desidero ringraziare tutte le persone che mi hanno aiutato e mi sono state vicine durante questa avventura. Ringrazio innanzitutto il mio relatore, prof. Gaetano Zanghirati, che mi ha dato la possibilità di lavorare su una tematica interessante e stimolante, facendomi conoscere e apprezzare il mondo della ricerca universitaria. Lo ringrazio ancor di più per l'incoraggiamento costante e la fiducia dimostrami, per essersi sempre dimostrato disponibile a fornirmi utili consigli e spiegazioni per superare i problemi che man mano ho incontrato, ma soprattutto per avermi permesso di lavorare in un clima amichevole e informale. Grazie Gaetano, anche se non sarai più il mio tutor, di sicuro resterai la mia guida per sempre.

Un ringraziamento speciale lo devo al prof. Demetrio Labate, che mi ha accolto non solo nel suo dipartimento, ma anche nella sua casa, facendomi da "mediatore culturale" per la mia prima esperienza americana. Grazie Demetrio, per avermi fatto sentire a casa nonostante l'oceano, e per avermi dato la possibilità di imparare tanto. Non tutti hanno la possibilità di imparare a conoscere qualcosa di nuovo proprio da chi ha aggiunto quel tassello di conoscenza.

Un ringraziamento sincero anche alla prof. Valeria Ruggiero, che è stata ed è fonte continua di ispirazione, un esempio come ricercatore e come donna.

Ringrazio Federica con la quale è stato entusiasmante condividere problemi e interessi, e tutti gli altri componenti del gruppo di ricerca, Silvia, Riccardo, Marco, Alessandro e Luca, che in vario modo mi hanno aiutato nel mio percorso. Un grazie sentito anche agli altri coinquilini del corridoio, Giacomo e Elisa, per tutte le risate, ma soprattutto i cappuccini. Un grazie speciale a Michele, per tutte le volte che ha salvato il mio computer dalla fine alla quale lo avevo condannato.

Un ringraziamento di cuore va alla mia famiglia che mi ha sempre sostenuto e incoraggiato, affacciandosi con curiosità sul mondo di cui avevo scelto di far parte. Un grazie sentito ai miei zii, Ciro e Maria, che per me ci sono sempre e fanno di tutto per rendermi felice.

Grazie di cuore anche alla mia famiglia ferrarese: ci siamo trovate per caso, ma se avessi potuto scegliere con chi condividere questa avventura, non avrei saputo scegliere meglio. Grazie amiche, per avermi "addomesticato".

Un ringraziamento sincero va a tutti gli amici e compagni di Università di questi anni: se sono arrivata qui, è anche merito vostro. Come affrontare questa materia lo abbiamo imparato insieme e, se gli anni dell'Università sono stati i più belli, lo devo solo a voi. Grazie soprattutto a te, con cui ho studiato tutti i pomeriggi, preparato tutti gli esami, ma che soprattutto mi hai insegnato come si festeggia.

Infine, il ringraziamento più sentito va a Mariella e Francesco. Grazie infinite per la pazienza smisurata e il sostegno continuo, per l'incoraggiamento costante e l'amore incondizionato. Grazie, perchè nessuno più di voi ha contribuito a rendermi la persona che sono.