University of Szeged
Department of Image Processing and Computer Graphics

Application of Discrete Tomographic Methods in Nondestructive Testing and Materials Science

PhD Thesis
by
Lajos Rodek

Academic Supervisor
Attila Kuba, DSc

University of Szeged
PhD School in Computer Science

Szeged
2015

Dedicated to my Mom and Grandma...
Édesanyámnak és nagymamámnak...

Acknowledgments

First of all, the author would like to thank his supervisor, professor Attila Kuba, for introducing the author into the wonderful world of discrete tomography, and for his limitless patience and support until his unfortunate death. The author is equally grateful to professors Gabor T. Herman and Henning Friis Poulsen for letting the author have a glimpse into the challenging fields of materials science, crystallography and polycrystals during the author's scholarship and internship, and for providing countless help along the way.

Many thanks are due to the following colleagues for all the conversations, ideas and suggestions: Andreas Alpers, Péter Balázs, Zoltán Kiss, Erik Bergbäck Knudsen, Arun K. Kulshreshth, Hstau Y. Liao, Antal Nagy, Wolfgang Pantleon, Stuart W. Rowland, László Ruskó, Søren Schmidt.

The author is also thankful for Márton Balaskó[1], Jacob R. Bowen[2], and Markus Strobl and Wolfgang Treimer[3] for supplying physical measurements.

For careful proofreading, I thank Andreas Alpers, Péter Balázs, Gabor T. Herman, Erik Bergbäck Knudsen, Antal Nagy, and Henning Friis Poulsen.

Last but not least, no words can express all the gratefulness the author owes his family for their endless support over the years.

The author conducted his research at the following institutions:

- Department of Image Processing and Computer Graphics, University of Szeged, Szeged, Hungary
  http://www.inf.u-szeged.hu/ipcg/starten.xml

- Discrete Imaging and Graphics Group, Department of Computer Science, The Graduate Center, City University of New York, New York, NY, USA
  http://www.dig.cs.gc.cuny.edu/

- Center for Fundamental Research: "Metal Structures in Four Dimensions", Risø National Laboratory, Roskilde, Denmark
  http://www.risoecampus.dtu.dk/DTU_Fysik/Projects_Neutrons/metals4D.aspx

The work had been supported by COST–Action 524 (Neutron Imaging Techniques for the Detection of Defects in Materials), Hungarian Scientific Research Fund (OTKA) under grants T032241 and T048476 (Theory and Application of Discrete Tomography; New Aspects and Applications of Discrete Tomography in Neutron Radiography), National Science Foundation (NSF) grant DMS0306215 (Aspects of Discrete Tomography), National Institutes of Health (NIH) grant

[1] KFKI Atomic Energy Research Institute, Budapest, Hungary

[2] Center for Fundamental Research: "Metal Structures in Four Dimensions", Risø National Laboratory, Roskilde, Denmark

[3] Hahn Meitner Institut, Berlin, Germany

HL070472 (Image Processing in Biological 3D Electron Microscopy), by the EU 6^{th} Framework Programme TotalCryst, by the Danish National Research Foundation, and by the Danish Natural Science Research Council (via Dansync). The European Synchrotron Radiation Facility is acknowledged for provision of beamtime.

Contents

Glossary and Notations

This chapter summarizes the meaning of some commonly employed abbreviations and acronyms, as well as the most important notational conventions used throughout the thesis. For convenience, each entry contains a pointer to the place of definition or first occurrence.

Acronyms

3DXRD	three-dimensional X-ray diffraction; see page 83
ANSI	American National Standards Institute; see page 53
ART	algebraic reconstruction technique; see page 6
CCD	charge-coupled device; see page 29
CT	computed / computerized tomography; see page 6
CUNY	City University of New York; see page 83
DIRECT	DIscrete REConstruction Techniques (framework); see page 54
DT	discrete tomography; see page 7
EBSD	electron backscattering diffraction; see page 101
ECAE	equal channel angular extrusion; see page 102
EM	electron microscopy; see page 14
ESRF	European Synchrotron Radiation Facility; see page 83
FBP	filtered backprojection; see page 6
FCC	face-centered cubic (crystalline lattice); see page 104
FOM	figure-of-merit; see page 62
GRAINDEX	name of program for "tracking" associated diffraction spots back to sample grains; see page 83
GRAINSWEEPER	name of program for determining the basic orientations, orientation spreads, centers-of-mass, and approximate morphologies of grains; see page 90

HMI	Hahn–Meitner Institut, Berlin, Germany; see page 36
KFKI	Atomic Energy Research Institute (Központi Fizikai Kutatóintézet), Budapest, Hungary; see page 36
LAC	linear attenuation coefficient; see page 37
MCMC	Markov chain Monte Carlo; see page 10
MRF	Markov random field; see page 9
MRI	magnetic resonance imaging; see page 6
NDT	nondestructive testing; see page 35
PET	positron emission tomography; see page 6
Risø	Risø National Laboratory; see page 83
RT	room temperature; see page 102
SA	simulated annealing; see page 11
SEM	scanning electron microscope; see page 101
SO(3)	special orthogonal group in 3D; see page 16
SPECT	single photon emission computed tomography; see page 6
VRML	Virtual Reality Modeling Language; see page 56

General

$\mathbf{A}$	matrix $\mathbf{A}$
$\lvert\cdot\rvert$	absolute value
$\mathfrak{c}$	cardinality of the continuum, $\mathfrak{c} = \lvert\mathbb{R}\rvert$
$\mathbb{C}$	field of complex numbers
$\lvert S\rvert$	cardinality of set S
$S \times T$	Cartesian product of sets S and T
$\mathbf{a} \times \mathbf{b}$	cross (vectorial) product of vectors $\mathbf{a}$ and $\mathbf{b}$
$\mathbf{a} \cdot \mathbf{b}$	inner (scalar) product of vectors $\mathbf{a}$ and $\mathbf{b}$
e	base of the natural logarithm (Euler's number)
i	complex imaginary unit
$\overline{AB}$	length of the line segment with end points A and B
$\mathbb{N}$	commutative semiring of natural numbers

$\|\cdot\|_1$	ℓ_1 norm (i. e. Manhattan metric)
$\|\cdot\|_2$	ℓ_2 norm (i. e. Euclidean metric)
$\Pr(\cdot)$	probability measure
$\mathbb{R}$	field of reals
$\operatorname{Tr}\mathbf{A}$	trace of square matrix $\mathbf{A}$
$\mathbf{v}$	vector $\mathbf{v}$
$\mathbb{Z}$	commutative ring of integers

Tomography

α	weight of $\Phi(f)$ in $\gamma_\Phi(f)$, $\alpha \in \mathbb{R}^+$; see Equation (2.14), page 13
$[\mathfrak{B}g](x,y)$	backprojection transform of g, $[\mathfrak{B}g](x,y) \in \mathbb{R}$; see Equation (2.5), page 6
β	inverse of the temperature, $\beta = \frac{1}{T}$, $\beta \in \mathbb{R}^+$; see Equation (2.12), page 13
$\mathcal{D}$	domain of f, $\mathcal{D} \subset \mathbb{R}^2$; see Equation (2.1), page 3
E	energy level of the incident radiation, $E \in \mathbb{R}^+$; see Equation (3.1), page 27
$E(X)$	energy of X, $E(X) \in \mathbb{R}$; see page 11
f	function to be reconstructed, $f\colon \mathcal{D} \to \mathcal{R}$; see Equation (2.1), page 3
$[\mathfrak{F}_1 g](v)$	1D Fourier transform of $g(x)$, $[\mathfrak{F}_1 g](v) \in \mathbb{C}$; see Equation (2.3), page 5
$[\mathfrak{F}_2 h](v,w)$	2D Fourier transform of $h(x,y)$, $[\mathfrak{F}_2 h](v,w) \in \mathbb{C}$; see Equation (2.3), page 5
$\hat{f}$	optimal approximation to f_{ideal}, $\hat{f}\colon \mathcal{D} \to \mathcal{R}$; see page 9
f_{ideal}	the unknown ideal solution being sought, $f_{\text{ideal}}\colon \mathcal{D} \to \mathcal{R}$; see page 9
$\gamma_{\text{B}}(f)$	objective function associated with f defined as a Boltzmann distribution; see Equation (2.15), page 14
$\gamma(f)$	(simple) objective function associated with f; see Equation (2.13), page 13
$\gamma_{\text{G}}(f)$	objective function associated with f defined as a Gibbs distribution; see Equation (2.16), page 14
$\gamma_\Phi(f)$	objective function associated with f and enhanced with $\Phi(f)$; see Equation (2.14), page 13

$H(X)$	(Hamiltonian) energy of X in joules, $H(X) \in \mathbb{R}$; see Equation (2.9), page 12
k_{B}	Boltzmann constant, $k_{\mathrm{B}} \approx 1.380\,648\,8 \times 10^{-23}\ \frac{\mathrm{J}}{\mathrm{K}}$; see Equation (2.9), page 12
μ	attenuation map (with implicit assumption of E), $\mu \colon \mathcal{D} \to \{0\} \cup \mathbb{R}^+$; see Equation (3.3), page 27
μ_E^t	LAC associated with material t and energy level E, $\mu_E^t \in \{0\} \cup \mathbb{R}^+$; see Equation (3.1), page 27
Ω	probability space; see page 10
$\mathbf{P}$	collection of input projections; see page 4
p	Metropolis ratio of acceptance, $p \in \mathbb{R}^+$; see Equation (2.7), page 11
$\mathbf{P}_{f,\vartheta}$	simulated discrete projection given f at direction ϑ; see Equation (2.13), page 13
$\mathbf{P}_{\vartheta}$	input discrete projection at direction ϑ; see Equation (2.13), page 13
$P(X)$	probability distribution of X; see page 10
$\Phi(f)$	fitness of f with respect to *a priori* data; see Equation (2.14), page 13
$\pi_{\mathrm{B}}(X)$	Boltzmann distribution; see Equation (2.9), page 12
$\pi_{\mathrm{G}}(X)$	Gibbs distribution; see Equation (2.12), page 13
$Q(X\|Y)$	conditional probability distribution of X given the random variable Y; see page 10
$\mathcal{R}$	range of f, $\mathcal{R} \subseteq \mathbb{R}$; see Equation (2.1), page 3
$[\mathfrak{R}f](s,\vartheta)$	Radon transform of f, $[\mathfrak{R}f](s,\vartheta) \in \mathbb{R}$; see Equation (2.2), page 3
(s,u)	the right-handed (x,y) coordinate system rotated about the origin by an angle ϑ counterclockwise; see Equation (2.2), page 3
T	temperature in kelvins, $T \in \mathbb{R}^+$; see page 11
T_0	initial temperature, $T_0 \in \mathbb{R}^+$, $T_0 \geq T_{\min}$; see page 11
$T_{\min}$	minimal allowed temperature, $T_{\min} \in \mathbb{R}^+$, $T_0 \geq T_{\min}$; see Equation (2.8), page 12
ϑ	projection angle, $\vartheta \in [0, 2\pi[$; see Equation (2.2), page 3
X	random variable, $X \colon \Omega \to \mathbb{R}$; see page 10
$Z_{T;X}$	T-dependent partition function over all possible X, $Z_{T;X} \in \mathbb{R}^+$; see Equation (2.10), page 13

Parametrized Objects

$\mathcal{A}$	finite set of permissible LAC values, $\mathcal{A} \subset \{0\} \cup \mathbb{R}^+$; see Equation (4.1), page 37
$\mathcal{C}$	configuration space; see Equation (4.5), page 39
$\mathbf{c}$	configuration, $\mathbf{c} \in \mathcal{C}$; see Equation (4.4), page 39
$\mathbf{c}_0$	initial configuration, $\mathbf{c}_0 \in \mathcal{C}$; see page 43
$\mathcal{D}$	finite 2D or 3D image domain of interest, $\mathcal{D} \subset \mathbb{Z}^2$ or $\mathcal{D} \subset \mathbb{Z}^3$; see Equation (4.1), page 37
$\mathcal{D}_\mathbf{P}$	finite 1D or 2D projection domain of interest, $\mathcal{D}_\mathbf{P} \subset \mathbb{Z}$ or $\mathcal{D}_\mathbf{P} \subset \mathbb{Z}^2$; see page 41
δ	precision to use when estimating the maximal feasible radius of a new randomly generated disc, sphere or cylinder, $\delta \in \mathbb{R}^+$; see page 46
ε	tolerance used when making comparisons with reals, $\varepsilon \in \mathbb{R}^+$; see page 41
ε_1	minimal allowed magnitude of a parameter alteration during the optimization of $\gamma(f_\mathbf{c})$, $\varepsilon_1 \in \mathbb{R}^+$; see page 42
f	digital image (attenuation map) to be reconstructed, $f\colon \mathcal{D} \to \mathcal{A}$; see Equation (4.1), page 37
$f_\mathbf{c}$	digital image (attenuation map) determined by $\mathbf{c}$, $f_\mathbf{c}\colon \mathcal{D} \to \mathcal{A}$; see Equation (4.7), page 39
$f_\mathbf{c}^{\text{bin}}$	binary digital image determined by $\mathbf{c}$, $f_\mathbf{c}^{\text{bin}}\colon \mathcal{D} \to \{0,1\}$; see Equation (5.7), page 56
$f_\mathbf{c}^{\text{gray}}$	gray-scale digital image determined by $\mathbf{c}$, $f_\mathbf{c}^{\text{gray}}\colon \mathcal{D} \to \{0, \frac{1}{2}I_{\max}, \frac{3}{4}I_{\max}, I_{\max}\}$; see Equation (5.6), page 55
$f_{\mathbf{c}_\text{o},\mathbf{c}_\text{r}}^{\text{gray}}$	gray-scale digital difference image determined by $\mathbf{c}_\text{o}$ (original, i. e. ideal target) and $\mathbf{c}_\text{r}$ (reconstructed), $f_{\mathbf{c}_\text{o},\mathbf{c}_\text{r}}^{\text{gray}}\colon \mathcal{D} \to \{0, \frac{1}{2}I_{\max}, I_{\max}\}$; see Equation (5.9), page 56
FOM_{CD}	figure-of-merit for the "configuration distance"; see Equation (5.19), page 63
FOM_{RME}	figure-of-merit for the "relative mean error"; see Equation (5.16), page 62
FOM_{SE}	figure-of-merit for the "shape error"; see Equation (5.17), page 62
FOM_{VE}	figure-of-merit for the "volume / area error"; see Equation (5.18), page 62
$\gamma(f_\mathbf{c})$	objective function associated with $f_\mathbf{c}$; see Equation (4.10), page 42

$H(f_{\mathbf{c}})$	(Hamiltonian) energy of $f_{\mathbf{c}}$, $H(f_{\mathbf{c}}) \in \mathbb{R}$; see Equation (4.9), page 41
h_i $(1 \leq i \leq N)$	height of the i^{th} cylinder, $h_i \in \mathbb{R}^+$, $h_i \geq h_{\min}$; see Equation (4.16), page 44
$h_{\max}$	height of the reconstruction area for a 3D $\mathcal{D}$, $h_{\max} \in \mathbb{R}^+$, $h_{\max} \geq h_{\min}$; see Equation (4.12), page 43
$h_{\min}$	minimal allowed height for cylinders, $h_{\min} \in \mathbb{R}^+$, $h_{\max} \geq h_{\min}$; see page 44
h_{TE}	height of the external cylinder of the tube, $h_{\text{TE}} \in \mathbb{R}^+$, $h_{\text{TE}} \geq h_{\min}$; see Equation (4.15), page 44
h_{TI}	height of the internal cylinder of the tube, $h_{\text{TI}} \in \mathbb{R}^+$, $h_{\text{TI}} \geq h_{\min}$; see Equation (4.15), page 44
HEIGHT	number of cross-sections in the 3D f, HEIGHT $\in \mathbb{Z}^+$; see Equation (4.11), page 43
$I_{\max}$	maximal gray-scale intensity, $I_{\max} \in \mathbb{Z}^+$; see Equation (5.6), page 55
λ	factor of adjustment for configuration parameters when violating geometrical constraints, $\lambda \in \mathbb{R}^+$, $0 < \lambda < 1$; see page 51
λ_1	factor of adjustment for the magnitude of a parameter alteration during the optimization of $\gamma(f_{\mathbf{c}})$, used when violating geometrical constraints, $\lambda_1 \in \mathbb{R}^+$, $0 < \lambda_1 < 1$; see page 42
MERGE_O	upper bound for the center distance when merging discs found in nearby 2D cross-sections into a single 3D geometrical primitive, $\text{MERGE}_O \in \mathbb{R}^+$; see page 51
MERGE_r	upper bound for the radius difference when merging discs found in nearby 2D cross-sections into a single 3D geometrical primitive, $\text{MERGE}_r \in \mathbb{R}^+$; see page 51
μ_0	LAC of the vacuum / air surrounding the object ($\frac{1}{\text{pixel}}$), $\mu_0 \in \mathcal{A}$; see page 38
μ_{D}	LAC of the discs ($\frac{1}{\text{pixel}}$), $\mu_{\text{D}} \in \mathcal{A}$; see page 38
$\hat{\mu}_{\text{D}}$	estimation for μ_{D}, $\hat{\mu}_{\text{D}} \in \{0\} \cup \mathbb{R}^+$; see page 51
$\hat{\mu}_{\text{I}}$	estimation for μ_{I}, $\hat{\mu}_{\text{I}} \in \{0\} \cup \mathbb{R}^+$; see page 51
$\hat{\mu}_{\text{R}}$	estimation for μ_{R}, $\hat{\mu}_{\text{R}} \in \{0\} \cup \mathbb{R}^+$; see page 51
$\hat{\mu}_{\text{SC}}$	estimation for μ_{SC}, $\hat{\mu}_{\text{SC}} \in \{0\} \cup \mathbb{R}^+$; see page 51
$\hat{\mu}_{\text{T}}$	estimation for μ_{T}, $\hat{\mu}_{\text{T}} \in \{0\} \cup \mathbb{R}^+$; see page 51
μ_{I}	LAC of the interior ($\frac{1}{\text{pixel}}$), $\mu_{\text{I}} \in \mathcal{A}$; see page 38
μ_{R}	LAC of the ring ($\frac{1}{\text{pixel}}$), $\mu_{\text{R}} \in \mathcal{A}$; see page 38

μ_{SC}	LAC of the spheres and cylinders ($\frac{1}{\text{pixel}}$), $\mu_{\text{SC}} \in \mathcal{A}$; see page 44
μ_{T}	LAC of the tube ($\frac{1}{\text{pixel}}$), $\mu_{\text{T}} \in \mathcal{A}$; see page 44
N	number of discs, spheres or cylinders, $N \in \mathbb{N}$; see Equation (4.4), page 39
n	number of projections, $n \in \mathbb{N}$, $n \geq 2$; see page 59
O_{C}	center of the reconstruction area, $O_{\text{C}} \in \mathbb{R}^2$ or $O_{\text{C}} \in \mathbb{R}^3$; see Equation (4.8), page 40
O_i $(1 \leq i \leq N)$	center of the i^{th} disc, sphere or cylinder, $O_i \in \mathbb{R}^2$ or $O_i \in \mathbb{R}^3$; see Equation (4.4), page 39
(O, r)	circle, disc or sphere with center O and radius r, $O \in \mathbb{R}^2$ or $O \in \mathbb{R}^3$, $r \in \mathbb{R}^+$, $r \geq r_{\text{min}}$; see page 38
(O, r, h)	cylinder with center O, radius r and height h, $O \in \mathbb{R}^3$, $r \in \mathbb{R}^+$, $r \geq r_{\text{min}}$, $h \in \mathbb{R}^+$, $h \geq h_{\text{min}}$; see page 44
O_{RE}	center of the external circle of the ring, $O_{\text{RE}} \in \mathbb{R}^2$; see Equation (4.4), page 39
O_{RI}	center of the internal circle of the ring, $O_{\text{RI}} \in \mathbb{R}^2$; see Equation (4.4), page 39
O_{TE}	center of the external cylinder of the tube, $O_{\text{TE}} \in \mathbb{R}^3$; see Equation (4.15), page 44
O_{TI}	center of the internal cylinder of the tube, $O_{\text{TI}} \in \mathbb{R}^3$; see Equation (4.15), page 44
$\mathbf{P}_{f_{\mathbf{c}},\vartheta}$	1D or 2D simulated discrete projection of $f_{\mathbf{c}}$ at direction ϑ, $\mathbf{P}_{f_{\mathbf{c}},\vartheta} \colon \mathcal{D}_{\mathbf{P}} \to \{0\} \cup \mathbb{R}^+$; see Equation (4.9), page 41
$\mathbf{P}_{\vartheta}$	1D or 2D input discrete projection at direction ϑ, $\mathbf{P}_{\vartheta} \colon \mathcal{D}_{\mathbf{P}} \to \{0\} \cup \mathbb{R}^+$; see Equation (4.9), page 41
r_i $(1 \leq i \leq N)$	radius of the i^{th} disc, sphere or cylinder, $r_i \in \mathbb{R}^+$, $r_i \geq r_{\text{min}}$; see Equation (4.4), page 39
r_{max}	radius of the reconstruction area, $r_{\text{max}} \in \mathbb{R}^+$, $r_{\text{max}} \geq r_{\text{min}}$; see Equation (4.8), page 40
r_{maxS}	maximal allowed radius for spheres, $r_{\text{maxS}} \in \mathbb{R}^+$, $r_{\text{max}} \geq r_{\text{maxS}} \geq r_{\text{min}}$; see Equation (4.13), page 44
r_{min}	minimal allowed radius for geometric primitives, $r_{\text{min}} \in \mathbb{R}^+$, $r_{\text{max}} \geq r_{\text{min}}$; see page 38
r_{RE}	radius of the external circle of the ring, $r_{\text{RE}} \in \mathbb{R}^+$, $r_{\text{RE}} \geq r_{\text{min}}$; see Equation (4.4), page 39
r_{RI}	radius of the internal circle of the ring, $r_{\text{RI}} \in \mathbb{R}^+$, $r_{\text{RI}} \geq r_{\text{min}}$; see Equation (4.4), page 39

r_{TE}	radius of the external cylinder of the tube, $r_{\mathrm{TE}} \in \mathbb{R}^+$, $r_{\mathrm{TE}} \geq r_{\min}$; see Equation (4.15), page 44
r_{TI}	radius of the internal cylinder of the tube, $r_{\mathrm{TI}} \in \mathbb{R}^+$, $r_{\mathrm{TI}} \geq r_{\min}$; see Equation (4.15), page 44
SIZE	number of rows and columns in the 2D f, SIZE $\in \mathbb{Z}^+$; see Equation (4.2), page 37
T	temperature, $T \in \mathbb{R}^+$; see Equation (4.10), page 42
T_0	initial temperature, $T_0 \in \mathbb{R}^+$, $T_0 \geq T_{\min}$; see page 43
$T_{\min}$	minimal allowed temperature, $T_{\min} \in \mathbb{R}^+$, $T_0 \geq T_{\min}$; see page 43
τ	expected noise level in $\mathbf{P}_\vartheta$, $\tau \in \mathbb{R}^+$; see page 51
$Z_{T;f_{\mathbf{c}}}$	T-dependent partition function over all possible $f_{\mathbf{c}}$, $Z_{T;f_{\mathbf{c}}} \in \mathbb{R}^+$; see Equation (4.10), page 42

Quaternions, Rotations

$\mathbf{0}$	the vector $(0,0,0)$, representing the vectorial (imaginary) part of a quaternion $(a, \mathbf{0})$; see page 20
Δ_Q	sampling unit of quaternion components (b,c,d) in $\mathbb{H}^Q_{1\mathrm{c}}$, $\Delta_Q \in \mathbb{R}^+$; see Equation (8.3), page 99
$\mathbb{H}$	skew field of quaternions
$\mathbb{H}_1$	non-commutative multiplicative group of unit quaternions; see page 21
$\mathbb{H}_{1\mathrm{c}}$	set of unit quaternions in the canonical form; see page 23
$\mathbb{H}^Q_{1\mathrm{c}}$	set of quantized unit quaternions in the canonical form, using sampling resolution Q; see page 99
i, j, k	imaginary units; see Equation (2.18), page 19
$\mathcal{N}_Q(\boldsymbol{q})$	set of neighbors of $\boldsymbol{q}$ in $\mathbb{H}^Q_{1\mathrm{c}}$; see Equation (8.2), page 99
$(\mathbf{n}, \theta)$	rotation about axis $\mathbf{n}$ by an angle θ (measured counterclockwise), $\mathbf{n}$ is a unit vector in $\mathbb{R}^3$, $\theta \in [0, \pi]$; see page 16
$(\psi, \varphi_1, \varphi_2)$	rotation defined by Euler angles ψ, φ_1 and φ_2, $\psi \in [0, 2\pi)$, $\varphi_1 \in [0, \pi]$, $\varphi_2 \in [0, 2\pi)$; see page 17
Q	sampling resolution for the interval $[-1, 1]$ of unit quaternion components (b,c,d), $Q \in \mathbb{Z}^+$ odd, $Q \geq 3$; see page 98
$\boldsymbol{q}^{-1}$	multiplicative inverse of quaternion $\boldsymbol{q}$, $\boldsymbol{q} \neq (0,0,0,0)$; see Equation (2.27), page 21
$\boldsymbol{q}_1\boldsymbol{q}_2 = \boldsymbol{q}_1 \cdot \boldsymbol{q}_2$	product of quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$; see Equation (2.23), page 20

$\boldsymbol{q}_1 + \boldsymbol{q}_2$	sum of quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$; see Equation (2.21), page 20
$\boldsymbol{q}_{1\to 2} = (a_{1\to 2}, \mathbf{q}_{1\to 2})$	transition quaternion expressing the rotation from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$, $\mathbf{q}_{1\to 2} \equiv (b_{1\to 2}, c_{1\to 2}, d_{1\to 2})$, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are unit quaternions; see Equation (7.21), page 93
$\boldsymbol{q} = (a, b, c, d)$	quaternion with scalar (real) component a and vectorial (imaginary) part (b, c, d), $a, b, c, d \in \mathbb{R}$; see Equation (2.17), page 19
$\boldsymbol{q} = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$	quaternion with scalar (real) component a and vectorial (imaginary) part $b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$, $a, b, c, d \in \mathbb{R}$; see Equation (2.18), page 19
$\boldsymbol{q} = (a, \mathbf{q})$	quaternion with scalar (real) component a and vectorial (imaginary) part $\mathbf{q} \equiv (b, c, d)$, $a, b, c, d \in \mathbb{R}$; see Equation (2.22), page 20
$\bar{\boldsymbol{q}}$	conjugate of quaternion $\boldsymbol{q}$; see Equation (2.20), page 19
$\boldsymbol{q} = \left(\cos\left(\frac{1}{2}\theta\right), \mathbf{n}\sin\left(\frac{1}{2}\theta\right)\right)$	unit quaternion representing the rotation $(\mathbf{n}, \theta)$, $\mathbf{n}$ is a unit vector in $\mathbb{R}^3$, $\theta \in [0, \pi]$; see Equation (2.29), page 21
$\lvert\boldsymbol{q}\rvert$	norm (magnitude) of quaternion $\boldsymbol{q}$; see Equation (2.25), page 21
$\mathbf{r} = \mathbf{n}\tan\left(\frac{1}{2}\theta\right)$	Rodrigues vector representing the rotation $(\mathbf{n}, \theta)$, $\mathbf{n}$ is a unit vector in $\mathbb{R}^3$, $\theta \in [0, \pi]$; see page 18
$r(\boldsymbol{q}_1, \boldsymbol{q}_2)$	distance of rotations (unit quaternions) $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, $0 \leq r(\cdot, \cdot) \leq 1$; see Equation (7.22), page 94

Polycrystals

α	weight of projection error term in $L(\mathbf{P}\lvert o)$, $\alpha \in \mathbb{R}^+$; see Equation (7.9), page 87
C	a clique (of any kind) in $\mathcal{D}$; see Equation (7.4), page 85
$\mathcal{C}_+$	set of all horizontal and vertical pair cliques in $\mathcal{D}$; see Equation (7.4), page 85
$\mathcal{C}_{+,\ell}$	set of all horizontal and vertical pair cliques in $\mathcal{D}$ restricted to grain g_ℓ; see Equation (7.16), page 90
$\mathcal{C}_k$ $(0 \leq k \leq 6)$	sets of all cliques in $\mathcal{D}$ with 3×3 blocks of pixels modeling grain morphologies in g; see page 90
$\mathcal{C}_{3\times 3}$	set of all cliques in $\mathcal{D}$ with 3×3 blocks of pixels; see Equation (7.6), page 86
$\mathcal{C}_\times$	set of all diagonal pair cliques in $\mathcal{D}$; see Equation (7.4), page 85
$\mathcal{C}_{\times,\ell}$	set of all diagonal pair cliques in $\mathcal{D}$ restricted to grain g_ℓ; see Equation (7.16), page 90

$\mathcal{D}$	finite 2D sample domain of interest, $\mathcal{D} \subset \mathbb{Z}^2$; see Equation (7.1), page 85
$d\left(o(i), o(j)\right)$	distance of orientations $o(i)$ and $o(j)$ (i. e. disorientation), $0 \leq d(\cdot, \cdot) \leq 1$; see Equation (7.5), page 85
Δ	minimal degree of separation of neighboring grains (i. e. misorientation), $\Delta \in \mathbb{R}^+$, $\Delta \geq \delta$; see Equation (7.7), page 86
δ	maximal expected orientation spread within any grain, $\delta \in \mathbb{R}^+$; see Equation (7.5), page 85
FOM_g	figure-of-merit for g; see Equation (8.5), page 106
FOM_o	figure-of-merit for o; see Equation (8.6), page 107
$\mathcal{G}$	set of grain labels; see Equation (7.12), page 89
G	number of grains, $G \in \mathbb{Z}^+$; see page 89
g	2D grain map to be reconstructed, $g\colon \mathcal{D} \to \mathcal{G}$; see Equation (7.11), page 89
g_0	initial grain map; see page 92
g_ℓ	subset of g associated with the grain labeled ℓ; see page 89
$\gamma(g, o)$	objective function associated with the pair (g, o); see Equation (7.19), page 92
$\gamma(o)$	objective function associated with o; see Equation (7.10), page 87
$H_1(g, o)$	homogeneity of o given g in $H(g, o)$; see Equation (7.16), page 90
$H_1(o)$	homogeneity of o in $H(o)$; see Equation (7.4), page 85
$H_2(g)$	grain borders of g in $H(g, o)$; see Equation (7.17), page 91
$H_2(o)$	grain borders of o in $H(o)$; see Equation (7.6), page 86
$H(g, o)$	(Hamiltonian) energy of the pair (g, o); see Equation (7.15), page 90
$H(o)$	(Hamiltonian) energy of o; see Equation (7.3), page 85
$\{hkl\}$	Miller indexes of the set of all lattice planes that are equivalent to (hkl) due to the symmetries of the lattice, $h, k, l \in \mathbb{Z}$ with greatest common divisor equal to 1; see page 136
(hkl)	Miller indexes corresponding to a set of lattice planes whose plane normal is defined by coefficients (h, k, l) in terms of the primitive vectors of the reciprocal lattice, $h, k, l \in \mathbb{Z}$ with greatest common divisor equal to 1; see page 136
i	a pixel of the sample domain, $i \in \mathcal{D}$; see Equation (7.1), page 85
$I^1_C(o)$	binary indicator function for horizontal and vertical borders in $\mathcal{C}_{3\times3}$; see Equation (7.6), page 86

$I^2_C(o)$	binary indicator function for diagonal borders in $\mathcal{C}_{3\times3}$; see Equation (7.6), page 86
$I^3_C(o)$	binary indicator function for intermediate borders in $\mathcal{C}_{3\times3}$; see Equation (7.6), page 86
$\{i,j\}$	a pair clique in $\mathcal{D}$ formed by pixels i and j; see Equation (7.5), page 85
$\kappa_{k;g}$ $(0 \leq k \leq 6)$	weight (potential) of $\mathcal{C}_k$ cliques in $H_2(g)$, $\kappa_{k;g} \in \mathbb{R}^+$; see Equation (7.17), page 91
κ_k $(k \in \{1,2,3\})$	weight (potential) of $I^k_C(o)$ cliques in $H_2(o)$, $\kappa_k \in \mathbb{R}^+$; see Equation (7.6), page 86
ℓ	a grain label, $\ell \in \mathcal{G}$; see page 89
L	sample–detector distance; see page 31
$L(\cdot)$	likelihood; see Equation (7.8), page 87
λ_1	weight (potential) of $\mathcal{C}_+$ cliques in $H_1(o)$, $\lambda_1 \in \mathbb{R}^+$; see Equation (7.4), page 85
$\lambda_{1;g}$	weight (potential) of $\mathcal{C}_{+,\ell}$ cliques in $H_1(g,o)$, $\lambda_{1;g} \in \mathbb{R}^+$; see Equation (7.16), page 90
λ_2	weight (potential) of $\mathcal{C}_\times$ cliques in $H_1(o)$, $\lambda_2 \in \mathbb{R}^+$; see Equation (7.4), page 85
$\lambda_{2;g}$	weight (potential) of $\mathcal{C}_{\times,\ell}$ cliques in $H_1(g,o)$, $\lambda_{2;g} \in \mathbb{R}^+$; see Equation (7.16), page 90
$N(\mathcal{C}_k, g)$	number of occurrences of $\mathcal{C}_k$ cliques in g, $N(\mathcal{C}_k,g) \in \mathbb{N}$; see Equation (7.17), page 91
$\mathcal{N}\big(o(i)\big)$	set of neighbors of $o(i)$ in $\mathcal{O}$; see page 88
$\mathcal{O}$	set of crystalline orientations; see Equation (7.1), page 85
o	2D orientation map to be reconstructed, $o\colon \mathcal{D} \to \mathcal{O}$; see Equation (7.1), page 85
o_0	initial orientation map; see page 88
ω	angle of rotation applied to the specimen with respect to the laboratory frame of reference (i. e. projection angle), $\omega \in [-\pi, \pi[$; see page 31
$\mathbf{P}$	set of input diffraction patterns; see Equation (7.8), page 87
$\mathbf{P}_o$	set of simulated diffraction patterns given o; see Equation (7.9), page 87
$\Phi_{\{i,j\}}(o)$	similarity measure of orientations for clique $\{i,j\}$; see Equation (7.5), page 85

$\mathcal{S}$	set of all (proper) symmetry rotations of the lattice; see page 94
θ	Bragg angle associated with some reflection event; see Equation (C.2), page 137

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Discrete tomography (DT), a relatively new field of image processing, deals with the reconstruction of images—or functions in the general mathematical setting—from their projections, given the assumption that the range of the images / functions is a known, finite set. The latter constraint is, actually, rather easy to satisfy in many real-life problems, thus enabling the usage of DT in these cases.

The present thesis discusses two very different applications of DT: The first one deals with the reconstruction of images of objects composed of some geometrical primitives like tubes, cylinders and spheres. The specific scenario considered here arose in industrial nondestructive testing of objects using radiographic measurements. The second application, on the other hand, involves the reconstruction of orientation maps and grain maps of deformed polycrystalline material samples from X-ray diffraction patterns. These tasks can be very challenging but they are also crucial for several materials scientific concepts.

The outline of the dissertation is as follows: A short introduction and overview of topics is given in this chapter. Then, Chapter 2 and Chapter 3 summarize the necessary mathematical and physical foundations, respectively. The next 3 chapters cover the first application of DT: Chapter 4 presents the problem and the implemented algorithm, Chapter 5 portrays simulation results, while Chapter 6 shows results obtained with physical phantoms. The second application of DT is dealt with in the 2 subsequent chapters: the problem statement and the implemented approach are described in Chapter 7, while simulation results are revealed in Chapter 8. Apart from the aforementioned main chapters, further reconstruction results, algorithm details as well as theoretic background are given in numerous appendices.

For the reader's convenience, the thesis also includes a glossary and list of notations, lists of figures, tables and algorithms, an extensive bibliography, and an index of the most important terms.

Chapter 2

Mathematical Background

Tomography is a tool of image processing for determining (reconstructing) an image (or a function, in general) from a set of measurements over it (called projections). Whereas (classical) tomography considers functions whose range is infinite and possibly continuous, discrete tomography deals with the special case when the range is a known, finite set. This chapter first gives an overview of the fundamental mathematical constructs involved, as well as some typical use cases where these techniques can be employed.

The subject discussed in Chapters 7 and 8 relies on the notion of orientation of 3D crystalline lattices. Since orientations are fundamentally defined in terms of rotations, the rest of this chapter provides a survey about the various ways for representing rotations in 3D, especially focusing on the usage of unit quaternions.

2.1 Tomography

2.1.0.1 Mathematical Foundations

Let

$$f\colon \mathcal{D} \to \mathcal{R}, \quad (x, y) \mapsto f(x, y) \tag{2.1}$$

be a bivariate, compactly supported continuous function with domain $\mathcal{D} \subset \mathbb{R}^2$ and range $\mathcal{R} \subseteq \mathbb{R}$. Furthermore, let (s, u) denote the right-handed (x, y) coordinate system rotated about the origin by an angle ϑ counterclockwise ($\vartheta \in [0, 2\pi[$). Then the **Radon transform** $[\mathfrak{R}f](s, \vartheta)$ of f is given by

$$\begin{aligned}[\mathfrak{R}f](s, \vartheta) &= \int_{-\infty}^{\infty} f(x, y)\,\mathrm{d}u = \\ &= \int_{-\infty}^{\infty} f(s \cdot \cos\vartheta - u \cdot \sin\vartheta, s \cdot \sin\vartheta + u \cdot \cos\vartheta)\,\mathrm{d}u\,.\end{aligned} \tag{2.2}$$

This functional is illustrated by Figure 2.1. In fact, the formula above computes line integrals of f along a set of parallel lines (all being parallel to u), as depicted in Figure 2.2(a). For a fixed ϑ_0, the univariate function $[\mathfrak{R}f](s, \vartheta_0)$ is called the **projection** of f at direction ϑ_0, whereas the collection $(\ldots, [\mathfrak{R}f](s, \vartheta_i), \ldots)$, $\vartheta_i \in [0, 2\pi[$, is dubbed the **sinogram** of f. (That is, the sinogram is the collection of all the projections taken at every possible direction ϑ_i.) Analogously, the lines parallel to u are called **projection lines**. By definition, projections corresponding to direction $\vartheta_0 + \pi$ are "mirror" images of those taken at ϑ_0, obtained by flipping $[\mathfrak{R}f](s, \vartheta_0)$ about the u axis.

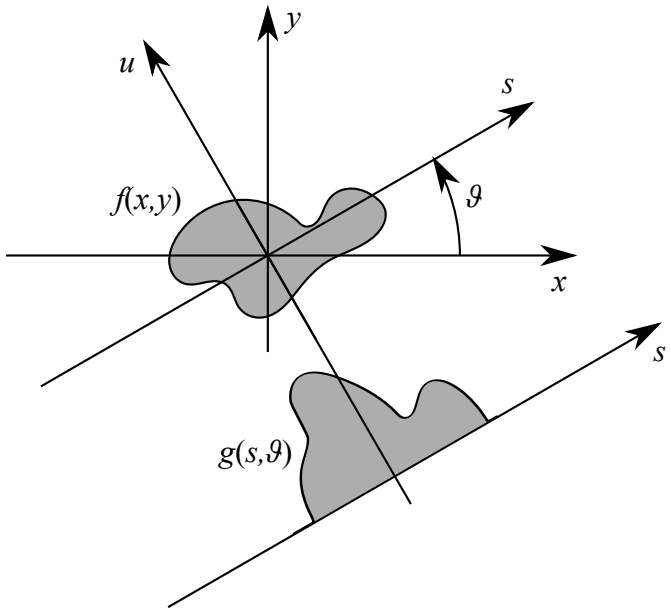


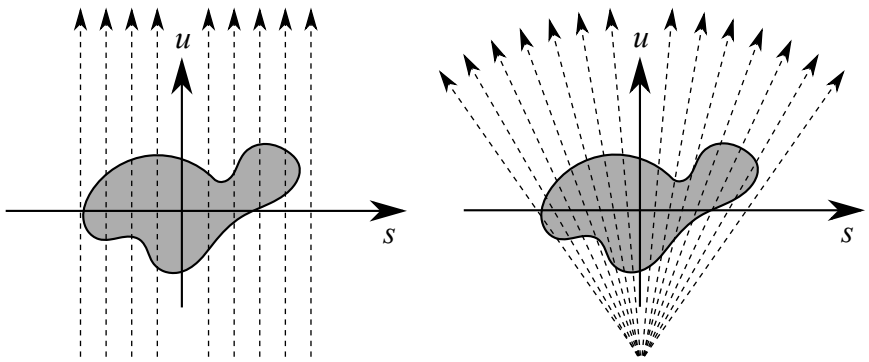
Figure 2.1. 2D Radon transform $g(s, \vartheta) \coloneqq [\mathfrak{R}f](s, \vartheta)$ of function $f(x, y)$.

The aforementioned transform was introduced in 1917 by Johann Karl August Radon, an Austrian mathematician [160, 161]. As can be seen, Equation (2.2) defines the 2D version of the transform (producing 1D projections) in the Euclidean space, but it can also be generalized to higher dimensions and other geometries [83, 127, 140].

The setup shown in Figure 2.2(a) uses **parallel geometry** (referring to the direction of projection lines for any fixed ϑ_0). Another widespread configuration is depicted in Figure 2.2(b), the so-called **divergent** or **fan beam geometry**, where all projection lines pass through a fixed point in the (s, u) coordinate system. (The analogous setup in 3D is called **cone beam geometry**. See Section 3.1.1 for the physical motivation behind these projection geometries.) It can be shown that the two setups yield equivalent results, as the projection lines of a fan-beam configuration can be reordered—or re-interpolated—into groups of parallel projection lines. (This reordering in known as **rebinning**.) More on this and on the practical implementation of divergent geometry can be found e. g. in [149, 150]. In even more advanced settings, integrals may be taken over more general subsets of f, e. g. along hyperbolas, as is the case in certain applications of the wave phenomenon diffraction; such an application is discussed in Chapters 7–8 of the thesis.

Tomography is a field of **image processing** that deals with the following problem: Given a collection $\mathbf{P} \coloneqq (\dots, p_i, \dots)$, $p_i \in \mathbb{R}$, of integrals of an unknown function f over certain subsets (manifolds) of its domain, determine f. (For practical reasons, $\mathbf{P}$ is often organized as a vector or matrix.) This is called the **reconstruction** of f from its projections $\mathbf{P}$. (It is evident that tomographic reconstruction is a kind of inverse problem.) As a matter of fact, the word "tomography" originates from the composition of the Ancient Greek $\tau\acute{o}\mu o\varsigma$ ("tomos"; slice or section) and $\gamma\rho\acute{a}\varphi\varepsilon\iota\nu$ ("graphein"; to write / draw).[1] The **reconstruction area**—encompassing the support of f—is determined as the intersection of projection lines for all possible values of ϑ.

[1] Due to the author's laziness, the Greek words are typeset using mathematical symbols rather than the appropriate text-mode Greek script.



(a) (b)

Figure 2.2. Projection geometries for the 2D Radon transform. **(a)** Parallel geometry. **(b)** Divergent (fan beam) geometry.

This shape is a disc (i. e. a solid circle) when using parallel or divergent projection geometries.

In certain industrial and medical applications of tomography, the values $f(x, y)$ can be identified with the emission intensity of a radioactive substance. In such a setup, projections need to be computed by taking into account the propagation and weakening (attenuation) of emitted radiation, thus resulting in projections quite different from those acquired via Equation (2.2). (In fact, this difference in computation also implies that projections corresponding to ϑ_0 and $\vartheta_0 + \pi$ are not simply mirror images of each other.) This technique is known as **emission tomography**, whereas the case described earlier is called **transmission tomography**. For more details on the physical notions involved, see Section 3.1.1.

In practice, the Radon transform—and tomography—is often used to compute the projections of a digital image ("**image function**", with function values representing gray-scale intensities or colors). This means that the domain $\mathcal{D}$ of f is a grid of some kind (e. g. the rectangular lattice $\mathbb{Z}^2$ or the regular hexagonal grid). Even though f is not continuous in such cases (in fact, it is a discrete function), it is possible to define its **discrete Radon transform** in analogy with Equation (2.2), producing discrete projections.

The Radon transform has an interesting relation to the **Fourier transform**. In particular, let $[\mathfrak{F}_1 g](v)$ and $[\mathfrak{F}_2 h](v, w)$ denote the 1D and 2D Fourier transforms of functions $g(x)$ and $h(x, y)$, respectively:

$$\begin{aligned} [\mathfrak{F}_1 g](v) &:= \int_{-\infty}^{\infty} g(x) \mathrm{e}^{-2\pi \mathrm{i} x v} \,\mathrm{d}x \\ [\mathfrak{F}_2 h](v, w) &:= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) \mathrm{e}^{-2\pi \mathrm{i}(xv + yw)} \,\mathrm{d}x \,\mathrm{d}y \end{aligned} , \tag{2.3}$$

where i is the complex imaginary unit. Moreover, let $[\mathfrak{R} f]_{\vartheta_0}(s)$ denote the 1D projection at direction ϑ_0. Then, the famous **Fourier slice theorem** (or **projection-slice theorem**) states that

$$\left[\mathfrak{F}_1 [\mathfrak{R} f]_{\vartheta_0}\right](v) = [\mathfrak{F}_2 f](v \cdot \cos\vartheta_0, v \cdot \sin\vartheta_0) \,. \tag{2.4}$$

In other words, the 2D Fourier transform of f can be obtained by "overlaying" the 1D Fourier transforms of $[\mathfrak{R} f]_{\vartheta_0}(s)$ (for all ϑ_0) in the 2D Fourier space at appropriate directions.

2.1.1 Reconstruction Methods

It should be noted that the Radon transform is not injective, hence it cannot be uniquely inverted in general (unless employing infinitely many projections). Instead, one can define the following adjoint (dual transform) to $g(s,\vartheta) \coloneqq [\mathfrak{R}f](s,\vartheta)$:

$$\begin{aligned}[\mathfrak{B}g](x,y) &= \frac{1}{2\pi}\int_0^{2\pi} g(s,\vartheta)\,\mathrm{d}\vartheta = \\ &= \frac{1}{2\pi}\int_0^{2\pi} g(x\cdot\cos\vartheta + y\cdot\sin\vartheta,\vartheta)\,\mathrm{d}\vartheta\end{aligned} \quad . \tag{2.5}$$

$[\mathfrak{B}g](x,y)$ is called the **backprojection transform** of $g(s,\vartheta)$. Although this transform can be also used by itself for sake of reconstruction, it is usually employed in combination with convolutional filtering. Specifically, higher quality results can be obtained when the projections are first filtered with some convolutional kernel (e. g. sinc, cosine, Ram–Lak, Shepp–Logan or Hamming) before being passed over to backprojection. The modified approach is known as **filtered backprojection** (FBP), and is one of the first tomographic reconstruction techniques devised.

The Fourier slice theorem provides another approach for reconstructing f. Namely, one can take the 1D Fourier transforms of each projection $[\mathfrak{R}f](s,\vartheta_0)$, and store the resulting complex vector into a common 2D matrix (or, in the continuous case, into $\mathbb{C}^2$) along a line incident with the origin and oriented in the appropriate direction as per Equation (2.4). Finally, the reconstruction result is obtained by applying the 2D inverse Fourier transform to the aforementioned matrix of coefficients.

The aforementioned two approaches—namely backprojection (with or without filtering) and Fourier reconstruction—are collectively known as **transform methods**. Besides, there are numerous further techniques for tomographic reconstruction; a few are listed below without completeness:

- Iterative algebraic techniques: algebraic reconstruction technique (ART) [75, 92], simultaneous iterative reconstruction technique (SIRT) [72], simultaneous algebraic reconstruction technique (SART) [19].
- Probabilistic optimization: maximum a posteriori expectation maximization (MAP EM) [86, 131], expectation maximization (EM) [55], maximum likelihood (ML) [178].
- Other optimization: total variation (TV) minimization [87].

A detailed overview of the mathematical background of tomography, along with a thorough discussion of related reconstruction algorithms and their usage in medical applications, can be found in [85, 95, 151].

2.1.2 Applications

Probably the most well-known application of tomography is its usage in medical imaging. This includes X-ray computed / computerized tomography (CT) [85, 95, 151], single photon emission computed tomography (SPECT), positron emission tomography (PET), magnetic resonance imaging (MRI), and various other uses (e. g. angiography) [20, 87, 131, 178]. Further important applications include: industrial radiography and nondestructive testing [172], optical tomography [20, 70], and acoustic tomography [56].

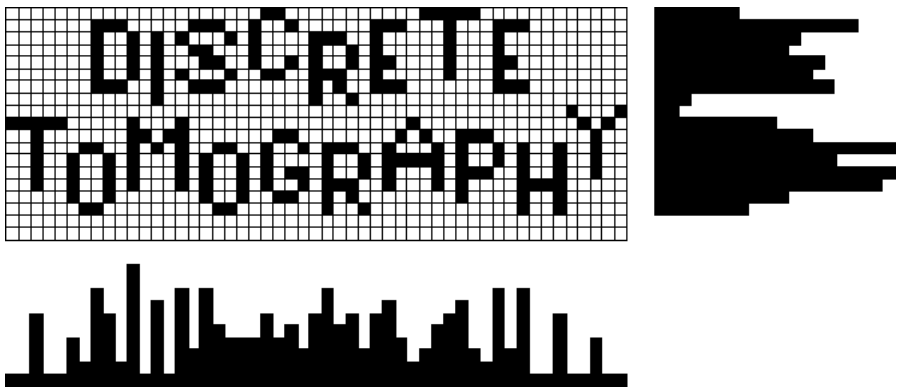


Figure 2.3. A binary image and its 2 orthogonal projections (column and row sums). Pixel intensities correspond to 0 / background (white) and 1 / foreground (black). (Image is courtesy of Gabor T. Herman and Attila Kuba.)

2.2 Discrete Tomography

2.2.1 Mathematical Foundations

As a special case of the notions introduced in Section 2.1, one speaks of **discrete tomography** (DT) when the range $\mathcal{R}$ of f is a known, finite set [88, 90]. In practical applications, this corresponds to the scenario when f represents 2D cross-sections of an object, with each pixel expressing some physical property of the material at that location (e. g. density, radiation attenuation / absorption capability, intensity of emitted radiation, or even more exotic features like the orientation of the underlying crystalline lattice). In many everyday situations, the object under investigation is composed of only a few materials of known properties, therefore complete—or at least an approximate—knowledge of $\mathcal{R}$ is available. An example is shown in Figure 2.3 for a binary f (i. e. when $|\mathcal{R}| = 2$). Though not an explicit requirement, most applications of DT also assume that the number of projections accessible is rather low (2–10) for various reasons; see Section 3.1.1 for a discussion.

DT is a relatively young field of image processing—dating back to the middle of the 20^{th} century—, having connections with discrete mathematics [12], combinatorics [43], number theory [17], complexity theory [68], and graph / network theory [63]. Similarly to the subjects discussed in Section 2.1—which we shall call **continuous** or **classical tomography** from now on—, the central objective of DT is to reconstruct f from its projections. This problem is accompanied by several additional matters that are investigated by DT:

Existence Given a collection $\mathbf{P}$ of measurements, determine if any solutions f exist at all having projections $\mathbf{P}$. In fact, this requires that the projections be **consistent** ("compatible") with each other, expressing the natural assumption that measurements shall not be contradictory. This topic was first considered for measurable functions [134], and later for binary matrices and 2 orthogonal projections [168, 169] as well as for flow networks [63].

Uniqueness Given a collection $\mathbf{P}$ of consistent measurements, determine if there is exactly one f having projections $\mathbf{P}$. As it turned out, a necessary and sufficient condition for this is that f does not contain so-called **switching components** (or **switching operations**). For binary images over the rectangular grid and 2 orthogonal projections, these are represented by the (sub)matrices $\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right)$ [168, 169]. (As is apparent, these matrices have

identical horizontal and vertical row sums, thus the reconstruction problem has at least two solutions.) This idea was later generalized to measurable sets and 2 orthogonal projections [124], to more than 2 projections [65], and to non-rectangular grids and higher dimensions [109]. Further uniqueness issues are investigated e. g. in [67, 113, 114, 119, 120].

Solution space Given a collection $\mathbf{P}$ of consistent measurements, how many solutions f exist having projections $\mathbf{P}$? How are they related? What is the structure of the solution space (also known as the **search space**)? How can one derive another solution f' from a known f? In general, is there a systematic means for obtaining *all* solutions?

Complexity Given a class $\mathcal{C}$ of images / functions with some specific property, what is the best achievable computational complexity of obtaining a reconstruction of any $f \in \mathcal{C}$? While this problem is NP-complete or even NP-hard for most of the cases [12, 54, 57, 67, 68, 97], there are also a few polynomial results [24, 25, 28, 54, 118, 119].

Stability Given some uncertainties or imperfections in the measurements $\mathbf{P}$, how different are reconstructions from the ideal solution? Some results can be found e. g. in [8, 10, 12, 13, 44, 96].

Convergence Given a particular reconstructions technique, determine under what conditions—and how fast—it converges to a solution. Some results include [71][2] and [108, 176].

For a detailed summary of the history of DT, along with a precise description of the aforementioned topics of interest, see [115, 116].

All the notions mentioned earlier for continuous tomography can be re-formulated within the frames of DT too. Thus one can speak of transmission and emission projections, parallel and divergent geometries, continuous and discrete Radon transforms etc. As a matter of fact, a related field of image processing dubbed **geometric tomography** deals with the reconstruction of planar and higher-dimensional shapes [65].

2.2.2 Reconstruction Methods

Due to the low number of projections and the special restrictions on $\mathcal{R}$, most classical (continuous) reconstruction methods are of limited effectiveness in DT. (Exceptions do exist; for instance ART can be tailored to the reconstruction of binary images, see below.) DT reconstruction techniques roughly fall into either of two categories: those that produce *exact* reconstructions and those seeking *approximate* ones.

Techniques for obtaining exact reconstructions These approaches are applicable when projections are perfect (i. e. in the absence of noise or other distortions). They may further assume that f possesses some special structure or property. Such methods usually rely on logical, number-theoretic or combinatorial tools.

- Binary matrices from 2 orthogonal projections: $(0, 1)$-valued [124, 168, 169], general [114].
- Convex / connected binary objects, polyominos: 2-SAT, [24, 25, 27, 44, 112, 118, 119].
- Graph theoretic techniques: [54].

[2] This paper discusses the convergence properties of the Gibbs Sampler when used in conjunction with simulated annealing (SA) [101, 185].

Techniques for obtaining approximate reconstructions These methods assume that measurements cannot be trusted, hence no exact solution can be expected in general. (Remember: By definition, no exact solution exists if projections are inconsistent.) Instead, they aim for producing an $\hat{f}$ that is as close to the ideal f_{ideal} as possible. This is usually attained by formulating the reconstruction process as an optimization task, searching for a *global* optimum of an objective function $\gamma(f)$ defined over the set of all possible f. (See Section 2.2.4 for a discussion of how such objective functions can be formulated.)

- Iterative algebraic techniques: BART [84], discrete algebraic reconstruction technique (DART) [30, 31].
- Probabilistic optimization: Markov random field (MRF) [40, 71, 192, 195], maximum a posteriori estimation (MAP) [32, 71, 143], multiscale models [61].
- Deterministic optimization: iterated conditional modes (ICM) [36, 143], greedy algorithms [5, 59, 78, 126], tabu search [73, 126], graduated non-convexity (GNC) [38], expectation maximization (EM) [191], linear programming (LP) [44, 137], gradient descent / steepest descent [159, Section 10.6] [32, 48], convex programming [195], difference-of-convex-functions (D.C.) [176, 191], convex–concave regularization [176, 190], mean field annealing (MFA) [37, 195].
- Stochastic optimization (Monte Carlo, heuristics): simulated annealing (SA) [159, Section 10.9] [101, 103, 143, 185, 188, 190], Metropolis algorithm [40, 81, 141, 192], modified Metropolis dynamics (MMD) [98, 99][3], Gibbs Sampler [71]. This class of techniques will be presented in more details in Section 2.2.3.
- Artificial intelligence: genetic algorithm [26, 74], machine learning [64], dynamic programming [41].

An evaluation / comparison of some of the aforementioned approaches can be found in [27, 46, 187, 190]. See also [88, 90] for a summary of these techniques, along with their possible applications.

It should be noted that, apart from trivial or very special cases, it is not viable to traverse the solution space $\mathcal{S}$ in order to locate solutions. This is so because $\mathcal{S}$ is simply too enormous and the associated solution graph too complex, so that classical graph search algorithms like A*—but also backtracking or brute-force—cannot possibly deduce solutions in an efficient way. (This is in perfect agreement with the complexity results mentioned earlier.)

The execution of certain DT algorithms can, sometimes, take a long time due to a slow convergence rate, a huge solution space, or the stochastic nature of the approach. There are various tricks for a speed-up, such as using look-up tables of precomputed values [15, 186], employing graphics processing unit (GPU) acceleration [187], or initializing the algorithm with an f_0 that is as close to the desired f_{ideal} as possible. (Examples for the latter can be found in the rest of the thesis.)

Reconstruction techniques can be also classified depending on the nature of the solution space. A majority of methods directly operate on digital images or their equivalents (discrete functions or matrices); these shall be called **pixel-based methods** from now on. On the other hand, some techniques, called **parametric methods**, define f with a vector $\mathbf{p}$ of parameters (the number of which is typically much smaller than the number of pixels in f). The purpose of such parameters is to encode the abstract structure of the object represented by f; see Section 4.1.2 for some specific examples.

[3] These papers actually present applications in computer vision and image segmentation.

2.2.3 Stochastic Optimization

Stochastic optimization is a form of optimization directly relying on random variables, random objective functions, random constraints, or iterative steps guided by random decisions. One specific application where such an approach has proven useful is the search for *global* optima of functions (of multiple variables in most cases). Below we shortly overview some of these methods that are essential to understand the reconstruction algorithms presented in the rest of the thesis. It should be remarked that all the examples discussed here are iterative techniques: they operate by starting from an initial guess X_0 of the random variable X over some probability space Ω, then construct a new approximation X_{i+1} in each iterative step based on the current latest approximation X_i.

Metropolis Algorithm

Though not an optimization technique fundamentally, the **Metropolis algorithm** [40, 81, 141, 192] (also called **Metropolis–Hastings algorithm**) lies at the core of—or had motivated—several stochastic global optimization methods. It is part of a larger family of algorithms called **Markov chain Monte Carlo** (MCMC); these repeatedly sample a probability distribution $P(X)$ in such a way that the sequence X_i of samples forms a Markov chain. MCMC approaches are especially beneficial when the direct sampling of $P(X)$ is difficult, e. g. because it depends on multiple variables or its exact value is not known.

In order to apply the Metropolis algorithm, one needs a function $h(X)$ that can be evaluated for all X and it satisfies $h(X) \propto P(X)$. It also requires a probability distribution $Q(X|Y)$ that suggests a new approximation for X based on the previous value Y. ($Q(X|Y)$ is usually chosen to favor values "close" to Y.) The outline of the process is given in Algorithm 2.1.

Algorithm 2.1. Metropolis algorithm

```
1  Choose X_0 arbitrarily (or based on some a priori information)
2  Fix Q(X|Y) as desired
3  i := 0
4  while termination criteria are not satisfied do
5      Draw X' randomly from Q(X'|X_i)
6      Calculate the ratio p := h(X')/h(X_i)
7      Accept X' with probability min{1, p}
8      if X' has been accepted then
9          X_{i+1} := X'
10     else
11         X_{i+1} := X_i
12     i := i + 1
13 X_i is a sample from P(X)
```

Depending on the value of p (dubbed the **Metropolis ratio of acceptance**), the proposed X' is either instantly accepted (when $p \geq 1$), or accepted with probability p (otherwise); see line 7. In practice, the latter action is implemented by taking a uniform random number $r \in [0, 1)$ and accepting X' if $r < p$.

As can be seen in the definition of p in line 6, the algorithm does not even need the exact value of $h(X)$; it suffices that the ratio $\frac{h(X_1)}{h(X_2)}$ be computable for all (X_1, X_2). This can be a big advantage if, for instance, the definition of $h(X)$ contains some constant factor c (independent of X) that is very time-consuming to deduce:

$$h(X) := c \cdot h_1(X). \tag{2.6}$$

Given the evaluation of $h_1(X)$ is not costly, the value of p becomes

$$p = \frac{h(X')}{h(X_i)} = \frac{c \cdot h_1(X')}{c \cdot h_1(X_i)} = \frac{h_1(X')}{h_1(X_i)},$$

that is c cancels out, so this ratio does not require its knowledge. Because $h(X)$ is proportional to $P(X)$,

$$p = \frac{h(X')}{h(X_i)} = \frac{P(X')}{P(X_i)}, \tag{2.7}$$

that is the distribution of X_i indeed converges to $P(X)$.

Iterations are carried on until some **termination criteria** are satisfied; these can be also chosen as most appropriate for the specific case. For example, the loop can be stopped when i or $h(X_i)$ exceeds some preset threshold.

Besides returning samples X_i that follow $P(X)$, the execution of the Metropolis algorithm can be interpreted as the maximization (if $\forall X\colon h(X) > 0$) or minimization (if $\forall X\colon h(X) < 0$) of the objective function $h(X)$. This optimization is a stochastic one, however, since X' may get accepted even when $|h(X')| < |h(X_i)|$ (i. e., when $h(X)$ worsens). This feature of the Metropolis algorithm makes it possible to reach a global optimum of $h(X)$, in spite of the existence of local optima that otherwise would be traps for the reconstruction process. The output of the algorithm is the X for which $h(X)$ is maximal or minimal among all the values visited during the execution of the algorithm.

The optimization procedure can, in theory, be started from an arbitrary X_0, but the closer the initial X_0 is to the expected global optimum, the faster the convergence of the algorithm will be (and the less the chance will be to get stuck in local optima).

Simulated Annealing

Simulated annealing (SA) [159, Section 10.9] [101, 185, 188] is a technique inspired by the annealing (cooling) of metals in metallurgy. The idea is that, by carefully controlling the rate of cooling, (poly)crystalline materials can achieve a significantly lower energy[4] level—as well as a reduced amount of internal defects—than what could be obtained by an abrupt or irregular cooling. In terms of applying these concepts in the realms of optimization, if $E(X)$ denotes the "**energy**[5]" associated with the random variable X and T the "**temperature**" of the system, the objective function $h(X) \equiv h_{E,T}(X)$ is interpreted as a function of $E(X)$ and T, while the slow annealing translates to a gradually decreasing probability for accepting a proposed X' that results in a worse $h_{E,T}(X)$ value. (Note that, by optimizing $h_{E,T}(X)$, one is simultaneously optimizing $E(X)$ as well. Depending on how $h_{E,T}(X)$ is constructed, it is also possible that the X that is a global maximum of $h_{E,T}(X)$ is actually a global minimum of $E(X)$, or vice versa.)

As can be seen in Algorithm 2.2, SA is an adaptation of the Metropolis algorithm.

As shown in line 6, X' is picked from the set of "neighbors" of X_i, that is by applying some small perturbation to X_i. (This notion corresponds to the analogously defined $Q(X|Y)$ in Algorithm 2.1.)

The decision about the fate of X' in line 7 uses the same p ratio and logic introduced in the Metropolis algorithm. There is one striking difference, though: here the evaluation of $h_{E,T}(X)$ depends on the current temperature T_i. Specifically, T_i controls the acceptance ratio as follows: a higher temperature increases the probability of accepting X' even when the value of the objective

[4] Specifically, this is the so-called thermodynamic free energy.

[5] It should be remarked that, contrary to its physical counterpart, $E(X)$ is permitted to take on 0 or negative values as well.

Algorithm 2.2. Simulated annealing

1 Choose X_0 arbitrarily (or based on some *a priori* information)
2 Set T_0 to some high value
3 Fix the annealing schedule to be used
4 $i \coloneqq 0$
5 **while** termination criteria are not satisfied **do**
6 Let X' be a randomly chosen neighbor of X_i
7 Determine the acceptance of X' with respect to X_i using the Metropolis ratio $p \coloneqq \frac{h_{E,T_i}(X')}{h_{E,T_i}(X_i)}$
8 **if** X' has been accepted **then**
9 $X_{i+1} \coloneqq X'$
10 **else**
11 $X_{i+1} \coloneqq X_i$
12 Let T_{i+1} be the new temperature based on T_i and the annealing schedule
13 $i \coloneqq i + 1$
14 X_i is the global optimum of $h_{E,T}(X)$, hence it is also a global optimum for $E(X)$

function $h_{E,T}(X)$ declines, while a lower temperature "freezes" the system into accepting only those X' that result in an improvement of $h_{E,T}(X)$.

The algorithm is started with a sufficiently high temperature T_0 that then gradually gets decreased according to a preset **annealing schedule** (see line 12). Some commonly employed schedules are shown below:

$$\begin{aligned} T_{i+1} &\coloneqq c \cdot T_i \\ T_{i+1} &\coloneqq \frac{T_0}{\ln(i+2)} \\ T_{i+1} &\coloneqq \max\{T_{\min}, T_i - \Delta T\} \end{aligned} \quad . \tag{2.8}$$

The first one essentially yields $T_{i+1} = c^{i+1} \cdot T_0$ where $0 < c < 1$ is some real close to 1, hence realizing an exponential decrease. The second one, on the other hand, follows a much slower logarithmic plan. Finally, the third one defines a simple linear model where T_i is decreased by some $\Delta T > 0$, while not letting it fall below a given $T_{\min} > 0$. As a special case, the temperature can be also kept constant (either all the time or starting from some i); this expresses a trade-off between adequate running speed and the possibility of getting stuck in unwanted local optima.

It is a common practice to define $h_{E,T}(X)$ so that it follows some probability distribution. One widespread choice is the **Boltzmann distribution** [139]:

$$\pi_{\mathrm{B}}(X) \coloneqq \frac{1}{Z_{T;X}} \mathrm{e}^{-\frac{H(X)}{k_{\mathrm{B}} T}} , \tag{2.9}$$

named after Ludwig Boltzmann, an Austrian physicist and philosopher. Originally used in statistical mechanics and thermodynamics, $\pi_{\mathrm{B}}(X)$ gives the probability that a system is in a certain state with energy $H(X)$ when the temperature of the system is T (given in kelvins). Moreover, k_{B} is the so-called **Boltzmann constant**, $Z_{T;X}$ denotes the **partition function**[6] (essentially, a T-dependent normalization constant that is defined in terms of all possible solutions X), and the **energy function** $H(X)$ (also called the **Hamiltonian** operator in quantum mechanics) gives the energy of X (originally expressed in joules). With the agreement $E(X) \coloneqq H(X)$, the setting $h_{E,T}(X) \coloneqq \pi_{\mathrm{B}}(X)$ can then be taken.[7] The maximization of $\pi_{\mathrm{B}}(X)$ is, in fact, equivalent with the minimization of $H(X)$.

[6] To be precise, the $Z_{T;X}$ shown here is the so-called canonical partition function of a discrete system in the context of classical mechanics.

[7] For all practical purposes, $H(X)$ can readily replace $E(X)$; the difference between them is purely notational and kept for historical reasons.

$Z_{T;X}$ is computed by summing the exponential term of Equation (2.9) over all possible solutions X:

$$Z_{T;X} \coloneqq \sum_{X} \mathrm{e}^{-\frac{H(X)}{k_\mathrm{B}T}} \,. \tag{2.10}$$

Sometimes it is very time-consuming—or even outright impossible—to calculate $Z_{T;X}$ like above; this would naturally imply that $h_{E,T}(X)$ cannot be evaluated precisely. Thanks to the remarks made about the Metropolis ratio p before, however, this is not an issue in practice: the expression (called the **Boltzmann factor**)

$$\frac{h_{E,T}(X_1)}{h_{E,T}(X_2)} = \frac{\mathrm{e}^{-\frac{H(X_1)}{k_\mathrm{B}T}}}{\mathrm{e}^{-\frac{H(X_2)}{k_\mathrm{B}T}}} = \mathrm{e}^{-\frac{H(X_1)-H(X_2)}{k_\mathrm{B}T}} \tag{2.11}$$

does not depend on $Z_{T;X}$ anymore.

Another common option for $h_{E,T}(X)$ is the **Gibbs distribution** [40, 71, 192] (also called **Gibbs measure**):

$$\pi_\mathrm{G}(X) \coloneqq \frac{1}{Z_{T;X}} \mathrm{e}^{-\frac{H(X)}{T}} = \frac{1}{Z_{T;X}} \mathrm{e}^{-\beta H(X)} \,, \tag{2.12}$$

where the positive real β corresponds to the inverse of the temperature. Named after Josiah Willard Gibbs (an American mathematician, physicist and chemist), $\pi_\mathrm{G}(X)$ provides a generalization of $\pi_\mathrm{B}(X)$ to infinite systems. Here, the relaxation parameter β is used to control the acceptance ratio: a lower β yields higher acceptance ratios, while increasing β "freezes" the system into accepting only improvements.

Gibbs Sampler

The **Gibbs Sampler** [71] can be considered a special case of the Metropolis algorithm. Instead of needing to evaluate the joint distribution of multiple random variables, it draws samples from a conditional distribution based on a local "neighborhood" of the current approximation X_i, thus requiring smaller computational efforts. It can be used by itself as well as in conjunction with SA.

2.2.4 Constructing the Objective Function over f

As mentioned in Section 2.2.2, the majority of reconstruction techniques for DT formulate the reconstruction problem as a global optimization task. In particular, the aim is to find an approximate solution $\hat{f}$ that is a *global* extremum of a suitably chosen objective function $\gamma(f)$ defined over the set of all possible f.

The simplest $\gamma(f)$ evaluates the distance of the simulated projections of f from the input projections:

$$\gamma(f) \coloneqq \sum_{\vartheta} \left\| \mathbf{P}_{f,\vartheta} - \mathbf{P}_\vartheta \right\| \,, \tag{2.13}$$

where $\mathbf{P}_{f,\vartheta}$ denotes the simulated discrete projection of f at direction ϑ, $\mathbf{P}_\vartheta$ the input—i. e. measured—discrete projection at direction ϑ, and $\|\cdot\|$ is some vector or matrix norm. Thus the equation above takes into account the **projection error** only, so that the optimal $\hat{f}$ is retrieved by *minimizing* $\gamma(f)$:

$$\hat{f} \coloneqq \arg\min_{f} \gamma(f) \,.$$

A more sophisticated objective function can be obtained from Equation (2.13) as follows:

$$\gamma_\Phi(f) \coloneqq \sum_\vartheta \|\mathbf{P}_{f,\vartheta} - \mathbf{P}_\vartheta\| + \alpha \cdot \Phi(f)\,. \tag{2.14}$$

Here, $\Phi(f)$ is a function measuring the **fitness** of f with respect to some ***a priori* information** (see below), while the scalar $\alpha > 0$ controls the relative strength of the **regularization** term $\Phi(f)$ to the projection error. Essentially, $\Phi(f)$ acts as a penalty / cost term [49], favoring candidates f with a better fitness and penalizing those having a worse fitness. With the new formulation, the optimum is still found by minimizing $\gamma_\Phi(f)$.[8]

The aforementioned *a priori* information expresses some additional knowledge about the optimum $\hat{f}$ (which itself is unknown, of course). Such prior data may include (without completeness): convexity, connectedness, homogeneity / smoothness, geometrical / topological constraints, ideal / reference model f_{ideal} (a "blueprint"), prototype object f_{proto}, or a probability distribution. (f_{ideal} may be a fixed model known beforehand, the result of a neighboring cross-section in a 3D set-up etc. f_{proto} can be chosen as an abstraction of f_{ideal}, or it may be acquired via some external method.) Specific examples will be given later in the thesis.

As mentioned in Section 2.2.3, objective functions can be also defined as probability distributions. In particular, taking the examples shown in Equations (2.9) and (2.12) brings us to

$$\gamma_{\text{B}}(f) \coloneqq \pi_{\text{B}}(f) = \frac{1}{Z_{T;f}} \mathrm{e}^{-\frac{H(f)}{k_{\text{B}}T}}\,, \tag{2.15}$$

based on the Boltzmann distribution, and

$$\gamma_{\text{G}}(f) \coloneqq \pi_{\text{G}}(f) = \frac{1}{Z_{T;f}} \mathrm{e}^{-\beta H(f)}\,, \tag{2.16}$$

relying on the Gibbs distribution. In both cases, $H(f)$ is constructed like Equations (2.13) or (2.14), which is still to be minimized, so that the optimum is acquired by *maximizing* $\gamma_{\text{B}}(f)$ or $\gamma_{\text{G}}(f)$:

$$\begin{aligned} \hat{f} &\coloneqq \arg\max_f \gamma_{\text{B}}(f) \\ \hat{f} &\coloneqq \arg\max_f \gamma_{\text{G}}(f) \end{aligned}\,.$$

2.2.5 Applications

DT has many applications in engineering, industry, and natural sciences. A listing of a few sample uses follows: electron microscopy (EM) [91,102], medicine [28,32,47,89,100,117], radiography and nondestructive testing [23, 49, 111, 143, 172, 180, 187], reconstruction from limited view angles [62,100,111,191], electron tomography [11], materials science [14,15,31,126,136,148,155], geometric tomography [28,66,67,69,108], emission tomography [28,118–120], geotomography [195]. Further examples for using DT in nondestructive testing and materials science are presented in Chapters 4–6 and Chapters 7–8 of the thesis, respectively.

DT has also found an interesting use for leisure in the form of nonograms, a kind of graphical puzzle [184].

[8] This aim does not contradict the phrase "fitness"; allowing negative values for $\Phi(f)$, the fittest $\hat{f}$ is the one where $\Phi(\hat{f})$ takes on the most negative value.

2.3 Representing Rotations in 3D

From a geometrical point of view, **rotation** is an affine transformation—specifically, an isometry—that leaves a single point fixed (viz. the origin of the coordinate system). In linear algebra, rotations form a subset of general linear transformations: they are realized as orthogonal $n \times n$ matrices over the field $\mathbb{R}$, forming a group under matrix multiplication. From now onwards, we will be focusing only on the $n = 3$ case, i. e. on the properties of rotations in 3D. Unlike their 2D companions, rotations in 3D are generally not commutative, which agrees with the well-known fact that matrix multiplication is not commutative either in general. (It can be shown that the multiplication of 2×2 orthogonal matrices does commute, however.[9])

A related and important concept is the **orientation** of a rigid body, what is usually understood as the rotation needed to move the body to its current placement with respect to some fixed frame of reference. As a special case, it is also possible to define the orientation of a crystalline lattice as the rotation that results in the current placement of atoms etc. forming the lattice. (It should be remarked that this rotation R is not unique due to the presence of rotational symmetries of the lattice: the same arrangement of lattice atoms etc. can be achieved by applying *any* of the rotations $R' := R \circ S$, where $\circ$ denotes composition and S is some rotational symmetry operation [76, 77, 82]. This issue will be dealt with in Section 7.4.) In fact, we are, in the current case, interested in how individual grains (as defined in Section 7.1) are oriented with respect to the reference frame of the acquisition system, that is in the transformation that maps the crystal lattice of a particular grain to the lattice of a hypothetical crystal aligned with the reference frame. (Even though such a transformation may prescribe a translation as well, only the rotation part matters from the orientation's point of view.)

When one speaks of rotations in 3D as a whole, what is usually meant is the set of **proper rotations**: transformations consisting of a sole rotation. On the contrary, **improper rotations** (also called **rotoinversions**) are a combination of a rotation and an inversion in a point. (As a consequence, rotoinversions reverse the handedness of the local coordinate system of the rotation target.) While the latter are indeed interesting when describing the rotational symmetry group of a crystalline lattice (dubbed the crystallographic **point group**), only proper rotations are relevant for establishing—and preserving—the crystalline orientation and thus those containing an inversion can be safely ignored [82].

There are numerous ways to represent a rotation in 3D; some of them are summarized below. While each of the approaches is powerful enough to describe all proper rotations, they nevertheless have vastly differing properties that may make them well-suited for certain applications but possibly less convenient for others. In order to choose the method of representation that best suits our goals (namely, to define crystalline orientations and orientation maps), the following guidelines are to be satisfied:

- Easy sampling of the space of parameters describing a rotation; in particular:
  - Minimal redundancy with respect to the number of parameters (i. e. compact representation).
  - Bounded parameter space.
- Unique representation of rotations.

[9] This comes from the following properties of complex numbers: According to Euler's formula, $e^{\theta i} = \cos\theta + (\sin\theta)i$; any such unit complex number defines a (proper) rotation in the complex plane about the origin by the counterclockwise angle θ. Moreover, any complex number $a+bi$ can be also represented as the real matrix $\left(\begin{smallmatrix} a & -b \\ b & a \end{smallmatrix}\right)$; for $e^{\theta i}$ this yields the orthogonal matrix $\left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$ with determinant $+1$, which is the familiar rotation matrix in 2D. The commutativity of multiplication for such matrices then follows from the commutativity of multiplication for complex numbers. See [53, Section 2.1] [135, Chapter 2] for more details.

- No **gimbal lock** in parameter space.[10]
- Easy re-normalization of parameters (for sake of counteracting numeric computation errors).
- Easy **composition** of successive rotations (in order to calculate the resultant transformation $R' := R_n \circ \cdots \circ R_2 \circ R_1$ representing a sequence of two or more rotations R_i $(1 \leq i \leq n)$).
- Easy calculation of inverse rotations R^{-1}.
- Having a "distance" metric (for measuring the "similarity" of two rotations).[11]

Though not essential, the following factors should also be considered:

- Intuitive representation.
- Easy visualization (especially using colors).

The subsequent list gives a quick overview of some representation schemes in widespread use. (See also [7, 9] for a comparison of some further approaches.) It should be stressed that the "disadvantages" indicated here are purely subjective and reflect the aforementioned requirements.

Axis-angle pairs: As stated by **Euler's rotation theorem** [53, Section 3.1], every motion of a rigid body with a fixed center is a rotation about an axis that runs through the center, therefore any sequence of rotations can be expressed as a single resultant rotation. It is also apparent that any sort of rotation leaves its axis invariant. That is why it appears to be the most ordinary method to represent an orientation as the pair $(\mathbf{n}, \theta)$, defining a rotation about the directed **axis** (unit vector) $\mathbf{n}$ by an **angle** θ (usually measured counterclockwise assuming a right-handed coordinate system). That makes a total of 4 parameters (3 for $\mathbf{n}$, 1 for θ), so this representation is quite compact. Observing that $(\mathbf{n}, \theta + \pi)$ and $(\mathbf{n}, \theta - \pi)$ establish the very same transformation, and the rotations determined by $(\mathbf{n}, \theta)$, $(\mathbf{n}, \theta + 2k\pi)$ and $(-\mathbf{n}, -\theta)$ are identical $(k \in \mathbb{Z})$, the angle can be taken such that $0 \leq \theta \leq \pi$ (in radians). The representation of rotations thus becomes unique with this convention.

Cons:

- Composition of rotations is messy, basically reducing to a form of quaternion multiplication based on vectorial scalar and cross products.
- Visualization is a bit troublesome as 4 parameters need to be mapped to 3 color primaries.

Rotation matrices: The **rotation matrix R** of a proper rotation is a 3×3 real orthogonal matrix with determinant $+1$. (On the contrary, improper rotations are represented by 3×3 real orthogonal matrices having determinant -1.) They form the multiplicative group called **special orthogonal group** (abbreviated as SO(3) for rotations in 3D). A minor inconvenience of this representation is that it is quite redundant: a total of 9 parameters need to be given (which are, of course, not totally independent of each other). On the other

[10] Gimbal lock is a sort of singularity or loss of degree of freedom occurring in some representation schemes when using certain combinations of rotation parameters. Basically, it refers to the undesirable phenomenon that not every change in the space of rotations can be realized by a change in the parameter space. See [18, 125] for details.

[11] As it turns out, this criterion can be solved rather easily irrespective of the actual representation, provided rotation inversion and composition are simple to devise. More details will be given in Section 7.4.

hand, the composition of rotations and rotation inverse can be calculated trivially using matrix multiplication and transposition, respectively. It is possible to extract the axis-angle pair $(\mathbf{n}, \theta)$ associated with the rotation expressed by a rotation matrix $\mathbf{R}$, as well as to directly determine the rotation matrix corresponding to $(\mathbf{n}, \theta)$ (see e. g. [18, Sections 3.2–3.3] or [125, Sections 3.4–3.5 and 7.5]). In particular, $\mathbf{n}$ is the eigenvector of $\mathbf{R}$ associated with the eigenvalue $+1$, while $\cos\theta = \frac{1}{2}(\operatorname{Tr}\mathbf{R} - 1)$ where $\operatorname{Tr}\cdot$ is the trace operator.

Cons:

- Redundant representation (9 parameters), hence problematic sampling in parameter space.
- Re-normalization is very costly, requiring to find the nearest orthonormal representation (using e. g. singular value decomposition or the matrix square root).
- Visualization is not trivial; the easiest is to extract the underlying axis-angle pair, but that then leads to the issue mentioned above (4 vs. 3 values).

Euler angles, Cardan angles: The **Euler angles** $(\psi, \varphi_1, \varphi_2)$ determine a sequence of 3 rotations: the first is by an angle $\psi \in [0, 2\pi)$ about axis_1, the second is by an angle $\varphi_1 \in [0, \pi]$ about $\text{axis}_2 \neq \text{axis}_1$, and the third is by an angle $\varphi_2 \in [0, 2\pi)$ about $\text{axis}_3 = \text{axis}_1$. A common choice is to have $\text{axis}_1 = \text{axis}_3 = z$-axis and $\text{axis}_2 = x$-axis (the so-called zxz-convention; see [14, 153]), though many other possibilities exist. The **Cardan angles** (also called **Tait–Bryan angles**) are different in that axis_3 has to be chosen as the leftover third coordinate axis, so that each rotation is about a different coordinate axis. The major problems with these representations are their order dependence (a different result is obtained when the order is altered) and non-uniqueness (given a fixed order, the same result can be acquired using two different—and unrelated—sets of angles). Furthermore, both schemes suffer from the presence of gimbal locks, irrespective of the choice of axes. (For the zxz-convention, a gimbal lock happens when φ_1 equals 0 or π. In those cases, only $\psi + \varphi_2$ or $\psi - \varphi_2$ are uniquely determined, not the individual values of these angles.) Still, they do have some appealing advantages: they possess the most compact representation so far (requiring just 3 parameters—a value matched only by Rodrigues vectors as shown below), their visualization is straightforward (each of the parameters being mapped to a color component), and the orientation is described in terms of anatomical motion (hence yielding a very intuitive representation). It is due to the latter property that Euler angles are extensively used for navigation [125] (sailing, aviation, aerospace, satellites etc.), and are also the representation of choice in crystallography [45].

Cons:

- Non-unique representation.
- At least one gimbal lock for any selection of axes.
- Composition of rotations is quite complicated to determine.

Unit quaternions: In a very simplified view, **quaternions** can be considered as vectors of $\mathbb{R}^4$ obeying certain algebraic rules that allow the definition of multiplication, inversion, exponentiation etc. of these objects. When restricting the length (Euclidean norm) of such vectors to unit length, the resulting **unit quaternions** turn out to have a nice geometrical interpretation. Specifically, any unit quaternion $\boldsymbol{q} \coloneqq (a, b, c, d) = \left(\cos\left(\frac{1}{2}\theta\right), \mathbf{n}\sin\left(\frac{1}{2}\theta\right)\right)$ can be shown to establish a rotation parametrized with the axis-angle pair $(\mathbf{n}, \theta)$, where

a is dubbed the scalar (real) component and (b, c, d) the vectorial (imaginary) part. (The components of the unit quaternion formulated so are mostly called **Euler–Rodrigues parameters**, in honor to their "inventors" Leonhard Euler and Benjamin Olinde Rodrigues.) The composition of rotations and rotation inverse can be then easily calculated using quaternion multiplication and conjugation, respectively. This representation scheme conforms to most aforementioned guidelines except for two points: First, visualization is not trivial (4 parameters are to be mapped to 3 color components), but this can be overcome with a smart choice of the mapping function as demonstrated later (see Section 8.2.1). Second, although $\boldsymbol{q}$ and $-\boldsymbol{q}$ determine the same rotation, it is possible to make a deterministic choice between the elements of each pair $\{\boldsymbol{q}, -\boldsymbol{q}\}$, thus resulting in a truly unique representation. (A more rigorous and detailed description of quaternions, including the discussion of how to resolve the recently mentioned two issues, will be given in Section 2.4.)

Cons:

- Achieving uniqueness needs some elaboration.
- Visualization is tricky.

Rodrigues vectors: There are several ways to combine the 4 parameters of an axis-angle pair $(\mathbf{n}, \theta)$ into just 3 values. Among them, the object formulated as $\mathbf{r} := \mathbf{n} \tan\left(\frac{1}{2}\theta\right)$ is known as the **Rodrigues vector** [167] (also called **Gibbs** or **Gibbs–Rodrigues vector**). As can be seen, Rodrigues vectors can be obtained from unit quaternions as $\frac{1}{a}(b, c, d)$, that is by multiplying the vectorial part by the inverse of the scalar component. It is, hence, evident that the length of such vectors $\mathbf{r}_\infty$ tends to infinity as $\theta \longrightarrow \pm\pi$, making the parameter space unbounded. Moreover, as rotations about any fixed axis $\mathbf{n}$ by π and $-\pi$ are equivalent, the opposing vectors $\mathbf{r}_\infty$ and $-\mathbf{r}_\infty$ must be identified with each other, therefore making computations prone to gimbal locks near $\theta = \pm\pi$. On the other hand, it is rather easy to devise the composition of rotations and rotation inverse [60]. Despite the inconveniences mentioned before, the use of Rodrigues vectors—along with unit quaternions—is rather popular among materials scientists [60, 145, 146, 152, 153].[12]

Cons:

- Unbounded parameter space—also making visualization troublesome.
- Gimbal lock / singularity for rotations by $\theta = \pm\pi$.

As can be seen, no representation scheme is without flaws. Having weighed the associated advantages and complications, unit quaternions were finally chosen for representing rotations and orientations. The next section is devoted to the overview of the basic properties of quaternions and their usage for representing rotations.

2.4 Quaternions and Rotations

Since orientations are inherently related to rotations, this section aims to review a mathematically elegant and useful representation of rotations in 3D. Namely, this will be done using "quaternions" that will be defined instantly.

[12] It should be noted that the problems mentioned here can be alleviated by restricting rotations to a bounded subspace of the full Rodrigues space, which might be possible in the presence of rotational crystal symmetries. On the other hand, doing so makes the notion of "rotation distance" more difficult to handle.

What is the "distance" between the orientations defined by $(\mathbf{n}_1, \theta_1)$ and $(\mathbf{n}_2, \theta_2)$? Heuristically, we may argue that it should be (a monotonic function of) the smallest nonnegative angle θ_3 such that the rotation $(\mathbf{n}_1, \theta_1)$ followed by the rotation $(\mathbf{n}_3, \theta_3)$, for some unit vector $\mathbf{n}_3$, will result in the orientation defined by $(\mathbf{n}_2, \theta_2)$. At this stage it is far from obvious how such a distance should be computed. Quaternions will provide us with a natural way of achieving that. (The exact answer will be given in Section 7.4; see Equations (7.24) and (7.25).)

Quaternions and their applications to rotations and orientations have a well-developed theory [18, 53, 60, 125, 147]. This section gives a summary of their most important aspects that can be helpful for achieving our goals.

Quaternions are a particular example of a more general class of **hypercomplex numbers**, that is they are a sort of generalization of complex numbers, first invented by Irish physicist, astronomer and mathematician Sir William Rowan Hamilton. A **quaternion** $\boldsymbol{q}$ is a 4-tuple

$$\boldsymbol{q} \coloneqq (a, b, c, d)\,, \tag{2.17}$$

often written as a formal linear combination

$$\boldsymbol{q} \coloneqq a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}\,, \tag{2.18}$$

where a, b, c and d are real. By analogy with complex numbers, the symbols i, j and k are called **imaginary units**, a the **scalar component** (or real component), and (b, c, d) the components forming the **vectorial part** (also dubbed the imaginary part). The imaginary units satisfy the conditions

$$\begin{aligned} \mathrm{i}^2 &\coloneqq -1\,, \\ \mathrm{j}^2 &\coloneqq -1\,, \\ \mathrm{k}^2 &\coloneqq -1\,, \\ \mathrm{ijk} &\coloneqq -1\,, \end{aligned} \tag{2.19}$$

which also imply

$$\mathrm{ij} = -\mathrm{ji} = \mathrm{k}\,, \quad \mathrm{jk} = -\mathrm{kj} = \mathrm{i}\,, \quad \mathrm{ki} = -\mathrm{ik} = \mathrm{j}\,.$$

These rules are summarized in Table 2.1. As is apparent, the table is not symmetric; it can be thus expected that the multiplication of quaternions cannot be commutative. Indeed, quaternions form a non-commutative division algebra $\mathbb{H}$ (also called division ring or skew field), an algebraic structure obeying every field axiom except for that of commutativity of multiplication.[13]

In analogy with the complex conjugate, the **quaternion conjugate** $\bar{\boldsymbol{q}}$ of $\boldsymbol{q}$ is defined as

$$\bar{\boldsymbol{q}} \coloneqq (a, -b, -c, -d)\,. \tag{2.20}$$

Furthermore, double conjugation yields the original quaternion: $\bar{\bar{\boldsymbol{q}}} \equiv \boldsymbol{q}$.

The **addition** of two quaternions is simply realized as the component-wise sum

$$\boldsymbol{q}_1 + \boldsymbol{q}_2 \coloneqq (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2)\,. \tag{2.21}$$

It can be checked that the zero quaternion $(0, 0, 0, 0)$ is the **additive identity element**, and $-\boldsymbol{q} = (-a, -b, -c, -d)$ is the additive inverse of $\boldsymbol{q}$:

$$\begin{aligned} (0, 0, 0, 0) + \boldsymbol{q} = \boldsymbol{q} = \boldsymbol{q} + (0, 0, 0, 0)\,, \\ \boldsymbol{q} + (-\boldsymbol{q}) = (0, 0, 0, 0)\,. \end{aligned}$$

[13] It can be actually shown that such a construction of $\mathbb{H}$ is unique (see Hurwitz's theorem [53, Section 6.4]): With the definition of the quaternion norm and its multiplicative nature (see Equations (2.25) and (2.26)), there exist—up to isomorphism—exactly 4 *normed* division algebras (also called composition algebras) over $\mathbb{R}$: $\mathbb{R}$ itself, $\mathbb{C}$, $\mathbb{H}$, and $\mathbb{O}$ (the algebra of octonions). These are of dimension 1, 2, 4, and 8, respectively.

Table 2.1. Multiplication rules for quaternion imaginary units. Entries show the product e_1e_2 of units e_1 (left column) and e_2 (top row).

e_1e_2 \ e_2 / e_1	1	i	j	k
1	1	i	j	k
i	i	−1	k	−j
j	j	−k	−1	i
k	k	j	−i	−1

Although **multiplication** can also be carried out in the form introduced in Equation (2.18), a more compact form can be achieved by interpreting quaternions as pairs consisting of a scalar and a 3D "vector"[14]:

$$\boldsymbol{q} \coloneqq (a, b, c, d) = (a, \mathbf{q})\,, \tag{2.22}$$

where $\mathbf{q} \equiv (b, c, d)$. In this notation, the product of two quaternions becomes

$$\boldsymbol{q}_1\boldsymbol{q}_2 \coloneqq (a_1, \mathbf{q}_1)\cdot(a_2, \mathbf{q}_2) = (a_1a_2 - \mathbf{q}_1\cdot\mathbf{q}_2, a_1\mathbf{q}_2 + a_2\mathbf{q}_1 + \mathbf{q}_1\times\mathbf{q}_2)\,, \tag{2.23}$$

with $\mathbf{q}_1\cdot\mathbf{q}_2$ and $\mathbf{q}_1\times\mathbf{q}_2$ denoting the inner (scalar) and cross (vectorial) product of vectors $\mathbf{q}_1$ and $\mathbf{q}_2$, respectively. As expected, multiplication is not commutative but can be shown to satisfy associativity, and so one can use unambiguously the notation $\boldsymbol{pqr}$ for the product of the three quaternions $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{r}$. Moreover, multiplication is distributive with respect to addition. It can be also verified that

$$\overline{\boldsymbol{q}_1\boldsymbol{q}_2} = \bar{\boldsymbol{q}}_2\bar{\boldsymbol{q}}_1\,.$$

For completeness, the components of the product $\boldsymbol{q}_3 \coloneqq \boldsymbol{q}_1\boldsymbol{q}_2$ can be expanded as

$$\begin{aligned} a_3 &= a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2\,,\\ b_3 &= a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2\,,\\ c_3 &= a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2\,,\\ d_3 &= a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2\,. \end{aligned} \tag{2.24}$$

It is then easy to see that the quaternion $(1, 0, 0, 0)$ becomes the **multiplicative identity element**: for any $\boldsymbol{q}$ the following holds

$$(1, \mathbf{0})\boldsymbol{q} = \boldsymbol{q} = \boldsymbol{q}(1, \mathbf{0})\,,$$

where $\mathbf{0}$ denotes the vector $(0, 0, 0)$.

It is a straightforward task to check that

$$\begin{aligned} (a_1, \mathbf{0}) + (a_2, \mathbf{0}) &= (a_1 + a_2, \mathbf{0})\,,\\ (a_1, \mathbf{0})\cdot(a_2, \mathbf{0}) &= (a_1a_2, \mathbf{0})\,,\\ (a_1, \mathbf{0})\cdot(a_2, \mathbf{q}_2) &= (a_1a_2, a_1\mathbf{q}_2)\,. \end{aligned}$$

[14] Strictly speaking, the imaginary part of a quaternion is a *pseudovector* [18, Section 12.3]: an object left invariant under inversion in a point; whereas *vectors* change sign under inversion. There is an analogous difference between *scalars* and *pseudoscalars*, the former ones being invariant under inversion, while the latter ones changing sign. Though these are important distinctions in certain circumstances, the author will be deliberately ignoring it from here onwards for simplicity—especially because only proper rotations are of interest.

Hence the quaternion $(a, \mathbf{0})$ can be identified with the scalar a. In addition,

$$\begin{aligned}(a_1, b_1, 0, 0) + (a_2, b_2, 0, 0) &= (a_1 + a_2, b_1 + b_2, 0, 0)\,,\\ (a_1, b_1, 0, 0) \cdot (a_2, b_2, 0, 0) &= (a_1 a_2 - b_1 b_2, a_1 b_2 + a_2 b_1, 0, 0)\,,\end{aligned}$$

thus the quaternion $(a, b, 0, 0)$ can be treated as the complex number $a + b\mathrm{i}$. It is also evident, though quaternion multiplication in general is not commutative due to the cross product in Equation (2.23), that the multiplication of quaternions having the form $(a, b, 0, 0)$ is indeed always commutative. In other words, both $\mathbb{R}$ and $\mathbb{C}$ are embedded in $\mathbb{H}$.

The **norm** (or **magnitude**) $|\boldsymbol{q}|$ of $\boldsymbol{q}$ is given by

$$|\boldsymbol{q}| \coloneqq \sqrt{\boldsymbol{q}\bar{\boldsymbol{q}}} = \sqrt{\bar{\boldsymbol{q}}\boldsymbol{q}} = \sqrt{a^2 + b^2 + c^2 + d^2}\,. \tag{2.25}$$

(This definition actually matches the Euclidean norm of (a, b, c, d) when considered as a vector in $\mathbb{R}^4$.) It can be proven that the norm is multiplicative, that is

$$|\boldsymbol{q}_1 \boldsymbol{q}_2| = |\boldsymbol{q}_1| \cdot |\boldsymbol{q}_2| = |\boldsymbol{q}_2 \boldsymbol{q}_1|\,. \tag{2.26}$$

The (multiplicative) **inverse** $\boldsymbol{q}^{-1}$ of $\boldsymbol{q} \neq (0, \mathbf{0})$ is defined as

$$\boldsymbol{q}^{-1} \coloneqq \frac{\bar{\boldsymbol{q}}}{|\boldsymbol{q}|^2} = \frac{\bar{\boldsymbol{q}}}{\boldsymbol{q}\bar{\boldsymbol{q}}}\,. \tag{2.27}$$

The inverse is uniquely determined for every quaternion with a nonzero norm, that is except for the zero quaternion. It is clear that the multiplication of a quaternion by its inverse is commutative, and the product equals the identity element:

$$\boldsymbol{q}\boldsymbol{q}^{-1} = (1, \mathbf{0}) = \boldsymbol{q}^{-1}\boldsymbol{q}\,.$$

Quaternions of norm 1 are called **unit quaternions**. It is obvious, for every nonzero quaternion $\boldsymbol{q}$, that $\boldsymbol{q}/|\boldsymbol{q}|$ is always a unit quaternion. The inverse $\boldsymbol{q}^{-1}$ of a unit quaternion $\boldsymbol{q}$ happens to equal its conjugate $\bar{\boldsymbol{q}}$ (cf. Equation (2.27)), which has again a unit norm. Furthermore, due to Equation (2.26), the product of two or more unit quaternions is again a unit quaternion. This means that the set $\mathbb{H}_1$ of unit quaternions is closed under multiplication and inversion, multiplication is associative and has an identity element, therefore $\mathbb{H}_1$ forms a non-commutative group. It is also apparent that $\mathbb{H}_1$ can be mapped bijectively onto the surface of the 4D unit hypersphere.

Another useful property is that, if $\boldsymbol{q}_1$ is a unit quaternion and $\boldsymbol{q}_2$ is any quaternion, the following holds:

$$\text{if } \boldsymbol{q}_3 \coloneqq \boldsymbol{q}_1 \boldsymbol{q}_2 \bar{\boldsymbol{q}}_1 \text{ or } \boldsymbol{q}_3 \coloneqq \bar{\boldsymbol{q}}_1 \boldsymbol{q}_2 \boldsymbol{q}_1 \Longrightarrow a_3 = a_2\,, \tag{2.28}$$

that is the first component of $\boldsymbol{q}_3$ is the same as that of $\boldsymbol{q}_2$.

A quaternion $\boldsymbol{q}$ is called **pure** if it has a zero scalar component, that is it is of the form $\boldsymbol{q} = (0, \mathbf{q})$. If, in addition, $\boldsymbol{q}$ is of norm 1, one speaks of a **pure unit quaternion**. Pure quaternions can be thought of as embeddings of $\mathbb{R}^3$ "vectors" in $\mathbb{H}$.

There is a strong relation between the aforementioned multiplicative group of unit quaternions and SO(3). In particular, let $\mathbf{n}$ denote an arbitrary unit vector in 3D. Every unit quaternion $\boldsymbol{q}$ can then be written as

$$\begin{aligned}\boldsymbol{q} &\coloneqq \left(\cos\left(\tfrac{1}{2}\theta\right), n_x \sin\left(\tfrac{1}{2}\theta\right), n_y \sin\left(\tfrac{1}{2}\theta\right), n_z \sin\left(\tfrac{1}{2}\theta\right)\right)\\ &= \left(\cos\left(\tfrac{1}{2}\theta\right), \mathbf{n} \sin\left(\tfrac{1}{2}\theta\right)\right)\end{aligned} \tag{2.29}$$

for some $\theta \in [0, \pi]$ and $\mathbf{n}$. The unit quaternion given by this equation corresponds to a (proper) **rotation** about the directed axis $\mathbf{n}$ by an angle θ (measured counterclockwise assuming a right-handed coordinate system). (How such quaternions can be applied to perform rotations is described below.) As a special case, setting θ to π results in a quaternion having a zero scalar component. Hence every pure unit quaternion $\boldsymbol{q} = (0, \mathbf{n})$ defines a **binary rotation** about $\mathbf{n}$, including the imaginary units i, j and k as well.

Let $\boldsymbol{q}$ be a unit quaternion in the form Equation (2.29), and $\boldsymbol{p} = (0, \mathbf{p})$ the pure quaternion corresponding to the 3D "vector" $\mathbf{p}$. The most natural way to rotate $\boldsymbol{p}$ would be to multiply it by $\boldsymbol{q}$: $\boldsymbol{p}' \coloneqq \boldsymbol{q}\boldsymbol{p}$. Nevertheless, this product would not always be a pure quaternion. The solution is to apply the inverse $\boldsymbol{q}^{-1} = \bar{\boldsymbol{q}}$ of $\boldsymbol{q}$ as well:

$$\boldsymbol{p}' \coloneqq \boldsymbol{q}\boldsymbol{p}\bar{\boldsymbol{q}}\,, \tag{2.30}$$

known as the **Hamilton–Cayley rotation formula** [18, 53, 125]. It can be verified that $\boldsymbol{p}'$ is indeed a pure quaternion $(0, \mathbf{p}')$, where $\mathbf{p}'$ is the vector obtained from $\mathbf{p}$ by the rotation about the directed axis $\mathbf{n}$ by an angle θ. (According to Equation (2.28), the first component of $\boldsymbol{p}'$ matches that of $\boldsymbol{p}$, so the former turns out to be also a pure quaternion.) Two important consequences of the non-commutativity of multiplication are presented here. First, $\boldsymbol{q}$ and $\bar{\boldsymbol{q}}$ do not cancel out one another. Second, swapping $\boldsymbol{q}$ and $\bar{\boldsymbol{q}}$ (i. e. as $\boldsymbol{p}' \coloneqq \bar{\boldsymbol{q}}\boldsymbol{p}\boldsymbol{q}$) establishes the inverse transformation that can be obtained from Equation (2.30) by substituting θ by $-\theta$.

Expanding Equation (2.30) gives

$$\begin{aligned} \boldsymbol{p}' \coloneqq \boldsymbol{q}\boldsymbol{p}\bar{\boldsymbol{q}} &= (0, \mathbf{p}') \\ &= \big(0, \mathbf{p} + (\sin\theta)(\mathbf{n} \times \mathbf{p}) + 2\left(\sin^2\left(\tfrac{1}{2}\theta\right)\right)\mathbf{n} \times (\mathbf{n} \times \mathbf{p})\big) \\ &= \big(0, (\cos\theta)\mathbf{p} + (\sin\theta)(\mathbf{n} \times \mathbf{p}) + (1 - \cos\theta)(\mathbf{n} \cdot \mathbf{p})\mathbf{n}\big) \end{aligned} \tag{2.31}$$

using the identities

$$\begin{aligned} \sin^2\left(\tfrac{1}{2}\theta\right) &= \frac{1 - \cos\theta}{2}\,, \\ \mathbf{a} \times (\mathbf{b} \times \mathbf{c}) &= (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}\,. \end{aligned}$$

This operation is called the **conical transformation** of $\boldsymbol{p}$ by $\boldsymbol{q}$, since the transformed vector $\mathbf{p}'$ moves on a cone around the rotation axis $\mathbf{n}$ (see [18, Section 12.8] and [167]). Moreover, $|\mathbf{p}| = |\mathbf{p}'|$ because of the multiplicativity of the quaternion norm.

It can be shown that there is a 2-to-1 homomorphism of $\mathbb{H}_1$ onto SO(3) (also called a "double cover"), namely the unit quaternions $\boldsymbol{q}$ and $-\boldsymbol{q}$ define the same rotation (see [53]). It should be remarked that the parameters $(\mathbf{n}, \theta)$ and $(-\mathbf{n}, -\theta)$ determine the very same quaternion, what is in agreement with their usual interpretation. Even though this notation of a rotation was first introduced by the French banker, mathematician and social reformer Benjamin Olinde Rodrigues as discussed in [18], the components of the quaternion in Equation (2.29) are mostly called **Euler–Rodrigues parameters** (or simply Euler parameters). For an in-depth discussion of the geometry of rotations represented using $\mathbb{H}_1$, see [140].

According to Euler's rotation theorem, the most general motion of a sphere with fixed center is a rotation, implying also that two or more successive rotations can be expressed as a single resultant rotation. For the product of two or more unit quaternions is again a unit quaternion, the **composition** of rotations—formulated as unit quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$—is calculated as

$$\boldsymbol{p}' \coloneqq \boldsymbol{q}_2(\boldsymbol{q}_1\boldsymbol{p}\bar{\boldsymbol{q}}_1)\bar{\boldsymbol{q}}_2 = (\boldsymbol{q}_2\boldsymbol{q}_1)\boldsymbol{p}(\bar{\boldsymbol{q}}_1\bar{\boldsymbol{q}}_2) = (\boldsymbol{q}_2\boldsymbol{q}_1)\boldsymbol{p}(\overline{\boldsymbol{q}_2\boldsymbol{q}_1}) = \boldsymbol{q}_3\boldsymbol{p}\bar{\boldsymbol{q}}_3\,, \tag{2.32}$$

where $\boldsymbol{q}_3 \coloneqq \boldsymbol{q}_2\boldsymbol{q}_1$ defines a sequence of rotations, namely the rotation by $\boldsymbol{q}_1$ followed by that by $\boldsymbol{q}_2$. The net effect of these rotations is the rotation given by $\boldsymbol{q}_3$. When expressed in Euler–Rodrigues parameters, $\boldsymbol{q}_3$ yields an effective approach for computing the composition of rotations given as axis-angle pairs.

The composition of two binary rotations about a common axis $\mathbf{n}$—that is the square of the pure unit quaternion $(0, \mathbf{n})$—establishes a rotation by 2π, what is normally associated with the identity. Thus it seems a bit surprising that

$$(0, \mathbf{n}) \cdot (0, \mathbf{n}) = (-1, \mathbf{0}) \equiv -1\,.$$

This is, actually, in conformity with the definition of imaginary units in Equation (2.19). The equivalent unit quaternions $\boldsymbol{q}$ and $-\boldsymbol{q}$ are, therefore, related as

$$\begin{aligned} -\boldsymbol{q} = (-1, \mathbf{0}) \cdot \boldsymbol{q} &= \left(\cos\left(\tfrac{1}{2}2\pi\right), \mathbf{n}\sin\left(\tfrac{1}{2}2\pi\right)\right) \cdot \left(\cos\left(\tfrac{1}{2}\theta\right), \mathbf{n}\sin\left(\tfrac{1}{2}\theta\right)\right) \\ &= \left(\cos\left(\tfrac{1}{2}(\theta + 2\pi)\right), \mathbf{n}\sin\left(\tfrac{1}{2}(\theta + 2\pi)\right)\right), \end{aligned}$$

that is $-\boldsymbol{q}$ is associated with the rotation $(\mathbf{n}, \theta + 2\pi)$. Along a similar reasoning, $-(-\boldsymbol{q}) \equiv \boldsymbol{q}$ corresponds to the rotation $(\mathbf{n}, \theta + 4\pi)$.

Even though there are infinitely many $(\mathbf{n}, \theta)$ pairs resulting in the same $\boldsymbol{q}$ (any of them obtainable from any other by steps consisting of changing θ by 4π or simultaneously taking the negative of $\mathbf{n}$ and θ), restricting θ to $[0, \pi]$ makes $(\mathbf{n}, \theta)$ unique for all $\theta < \pi$. (The scalar component of all such $\boldsymbol{q}$ is positive, whereas the equivalent $-\boldsymbol{q}$ has a negative first component.) For binary rotations (i. e. when $\theta = \pi$), $\boldsymbol{q}$ is a pure unit quaternion, so having to choose between $\boldsymbol{q}$ and $-\boldsymbol{q}$ remains an issue. To resolve this, the author defines the canonization of unit quaternions as follows: Considering the pair $\boldsymbol{q}$ and $-\boldsymbol{q}$, the one whose leftmost nonzero component is positive is said to be in the **canonical form**, where "leftmost" refers to the position of a component with respect to the ordering in Equation (2.17). (The canonical form is indeed well-defined because every unit quaternion has at least one nonzero component.) It is easy to see that, for any $\boldsymbol{q}$, exactly either of $\boldsymbol{q}$ and $-\boldsymbol{q}$ will be in the canonical form, hence the set $\mathbb{H}_{1c}$ of canonical unit quaternions provides a unique representation of rotations. (It should be remarked that $\mathbb{H}_{1c}$ is *not* closed under multiplication: the result of the product $\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0\right) \cdot \left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0\right) = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0\right)$ is not in $\mathbb{H}_{1c}$; its canonical form would look like $\left(\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0, 0\right)$. Therefore, an explicit canonization has to be performed after executing any quaternion operation if the uniqueness of rotations is important.)

2.5 Summary

We have established the mathematical foundations for continuous and discrete tomography, including an overview of problematic topics involved. After a brief introduction of some widespread reconstruction techniques, a few examples were shown regarding how tomographic approaches can be employed for practical applications.

We have also considered some commonly utilized representation schemes for rotations in 3D, discussing the basic properties quaternions in detail that will be later exploited in Section 7.4.

Chapter 3

Physical Background

This chapter first explains the minimally necessary physical foundations of radiography, focusing on the terminology and methods relevant for the practical applications discussed later in the thesis.

The rest of the chapter gives an overview of the fundamentals of X-ray diffraction techniques (especially 3DXRD, described in Chapter 7 in detail), in particular focusing on the imaging process from a computational point of view.

3.1 Radiography

3.1.1 Radiographic Basics

Radiography is an imaging technique that uses some kind of particle **radiation** to illuminate the object under investigation. The amount of radiation passing through and leaving the object is then measured and recorded by a **detector**, forming a so-called **radiograph** or, in the terminology of image processing, a **projection**. In the most typical setup, the object is placed somewhere between the **radiation source** and the detector, the beam is focused in the direction of the object and the radiograph records the radiation transmitted through the object. This kind of configuration, with the radiation source being located *outside* the object, is called **transmission** radiography. In another kind of setup, the detector is still located outside the object, but the radiation source is placed *inside* the object. (It is, of course, also possible that the radiation gets emitted by the materials of the object itself, due to the presence of radioactive isotopes.) This configuration is known as **emission** radiography. Both settings are demonstrated in Figure 3.1.

Whatever the case, the radiation is subject to some alterations as it passes through the object and arrives at the detector. Since the extent / magnitude of these effects usually depends on both material properties and radiation energy levels, experiments are typically carried out with a **monochromatic** beam, that is one comprising particles of a single common energy (or of a narrow energy band). A few examples of such alterations are mentioned below without completeness:

- A fraction of the radiation can be **absorbed** by the object, resulting in an **attenuation** (reduction) of the measured intensity as compared to that of the radiation source. This kind of imaging is dubbed **attenuation contrast** (or **absorption contrast**).

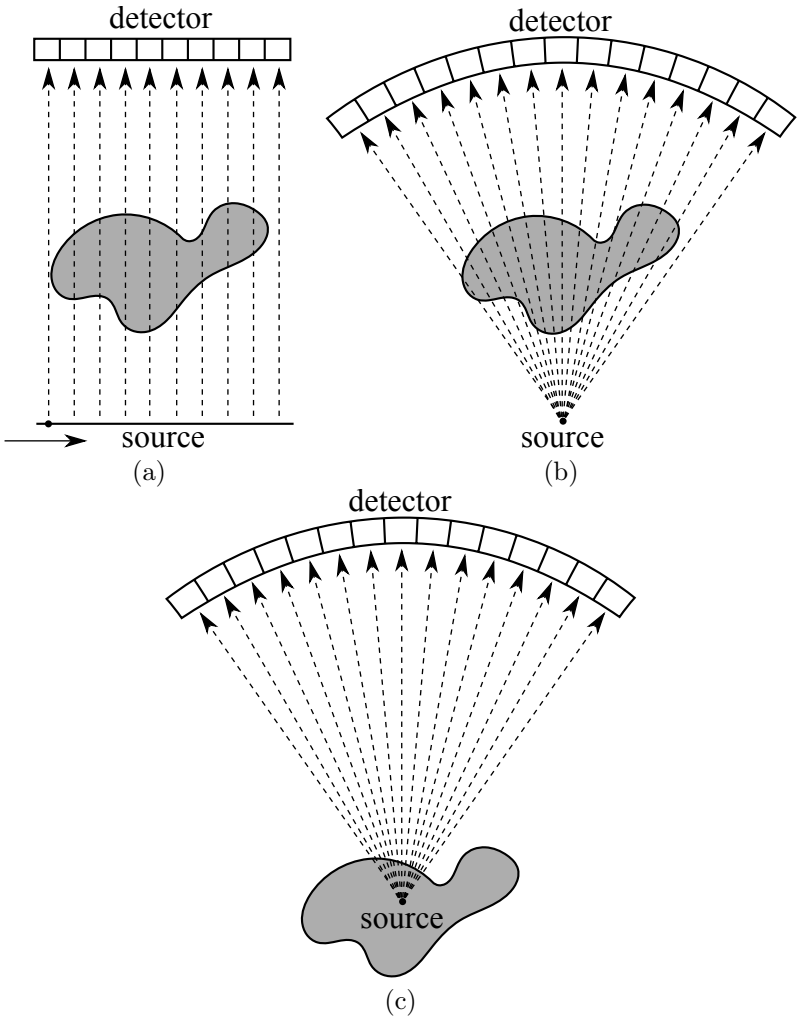


Figure 3.1. Projection geometries for radiograpic experiments. **(a)** Transmission radiography using parallel geometry. **(b)** Transmission radiography using divergent (fan beam) geometry. **(c)** Emission radiography.

- Due to the refractive properties of the materials constituting the object, the **phase** (in the sense of wave propagation) of the measured beam can become shifted with respect to that of the radiation source.[1] This approach is called **phase contrast**.
- When the materials forming the object possess some long-range order (i. e. having a crystalline lattice), a tiny portion of the radiation may undergo **diffraction**, resulting in a collection of "deflected" rays in addition to the incident beam passing through without being diffracted. This case is considered in Section 3.2.1 and Chapters 7–8 of the thesis.

A detailed comparison of the first two approaches can be found in [34, 79]. For the rest of the section, we are going to restrict our attention to the usage of attenuation contrast imaging in the transmission radiography setup.

A unique property of every material is its ability to attenuate radiation, which is expressed by the following term: Let a particle of energy E move perpendicular towards a homogeneous material t of unit thickness (here given in centimeters). If the probability of the particle passing through the material without absorption is $p \in [0, 1]$, then the **linear attenuation coefficient** [138] (LAC) or **absorption coefficient** μ_E^t associated with material t and energy level E is given by

$$\mu_E^t \coloneqq -\ln p \, \frac{1}{\mathrm{cm}} \,. \tag{3.1}$$

Obviously $\mu_E^t \geq 0$ real, with the two extrema being zero attenuation ($\mu_E^t = 0$) and complete attenuation ($\mu_E^t \to +\infty$). As a consequence, if the intensity of radiation entering and leaving the material is denoted by I_{in} and I_{out} (with $0 \leq I_{\mathrm{out}} \leq I_{\mathrm{in}}$), respectively, the following holds:

$$\mu_E^t = -\ln \frac{I_{\mathrm{out}}}{I_{\mathrm{in}}} = \ln \frac{I_{\mathrm{in}}}{I_{\mathrm{out}}} = \ln I_{\mathrm{in}} - \ln I_{\mathrm{out}} \,. \tag{3.2}$$

Furthermore, if the intensity of radiation measured at the source and the detector is designated by I_{source} and I_{det}, respectively, the linearity of μ_E^t implies

$$\ln I_{\mathrm{source}} - \ln I_{\mathrm{det}} = \int_{\mathrm{source}}^{\mathrm{det}} \mu(x) \, \mathrm{d}x \,, \tag{3.3}$$

where $\mu(x)$ denotes the LAC of the material at location x (for some implicitly assumed E), and $\mu \colon \mathcal{D} \to \{0\} \cup \mathbb{R}^+$ with $\mathcal{D}$ defined as in Section 2.1.0.1. Since we are, most of the time, interested in the value of the integral above, but can actually measure only I_{source} and I_{det}, a so-called **logarithmic transformation** needs to be performed as per Equation (3.3). As a matter of fact, the aforementioned integral expresses the Radon transform of the function $\mu(x)$ along the projection line source $\to$ det (cf. Equation (2.2)):

$$[\mathfrak{R}\mu](s, \vartheta) = \ln I_{\mathrm{source}} - \ln I_{\mathrm{det}} = \int_{\mathrm{source}}^{\mathrm{det}} \mu(u) \, \mathrm{d}u \,.$$

Equation (3.2) is, actually, obtained as a special case of the well-known **Beer–Lambert** (or **Beer–Lambert–Bouguer**) law [35, 39, 128] of attenuation in optics:

$$T \coloneqq \frac{I_{\mathrm{out}}}{I_{\mathrm{in}}} = \mathrm{e}^{-\mu l} \,.$$

Here, T is the transmittance (or transmissivity), μ is the LAC of the material (with implicit assumption of t and E), and l is the distance (path length) the radiation travels within the

[1] Obviously, this requires that the beam exiting the radiation source be coherent as well.

material. The result is due to August Beer (German physicist, chemist, and mathematician), Johann Heinrich Lambert (Swiss mathematician, physicist, philosopher and astronomer), and Pierre Bouguer (French mathematician, geophysicist, geodesist, and astronomer).

When multiple projections are available (taken in different directions, e. g. by rotating the object during exposure), the tools of tomography—especially those of discrete tomography—can be employed to reconstruct the internal structure of the object. This combination of techniques leads to the notion of **transmission** (emission, diffraction etc.) **tomography**. In such a setting, the objective is to reconstruct the **attenuation map** $\mu(x)$ from radiographs.

As mentioned earlier, the number of projections practically attainable depends on several factors, such as financial costs or acquisition time. Moreover, in many cases, the time of total exposure—hence the number of projections taken—shall be limited in order to minimize radiation dose (e. g. when irradiating a human patient) and to avoid damaging or "activating" the object (i. e. inducing radioactivity). The low number of projections usually rules out the usage of classic (continuous) tomographic techniques, thus letting them be replaced with discrete tomographic approaches.

3.1.2 Projection Acquisition

The history of radiography can be traced back to German physicist Wilhelm Conrad Röntgen who took the first X-ray of his wife's hand in 1895, just after discovering this new kind of radiation. There has been a lot of development in the field of radiography since then, so experiments can now be carried out using a rich selection of radiation types; a few examples are given here without completeness:

- Electromagnetic radiation:
  - Radio waves: MRI
  - X-ray: CT, industrial X-ray
  - Gamma rays: SPECT, PET, industrial gamma camera
- Elementary particle rays:
  - Electron: electron tomography
  - Neutron
- Heavy ions: atom probe tomography (APT)
- Mechanical waves: acoustic tomography, geotomography

The results we are going to present in Chapter 6 are based on experiments using three **imaging modalities**: **X-rays**, **neutron rays**, and **gamma rays**. X-ray beams can easily pass through "soft", low-density materials (e. g. air, muscle and fatty tissues), while they are largely blocked / absorbed by "hard", high-density ones (e. g. metals, bone tissue). On the other hand, neutron radiation behaves just the opposite way, passing through metals with less attenuation but being absorbed more by light nuclei. Gamma rays can cross "soft" materials as well as most metals, but they are blocked by very high density materials like lead or concrete.

All aforementioned experiments employed transmission radiography using parallel geometry; an outline of the instrument set-up is depicted in Figure 3.2. The object of interest is attached to a rotating stage. The beam leaving the radiation source is mechanically constrained into a thick "block" of rays (mostly with the help of a **collimator**), which is then let pass through the

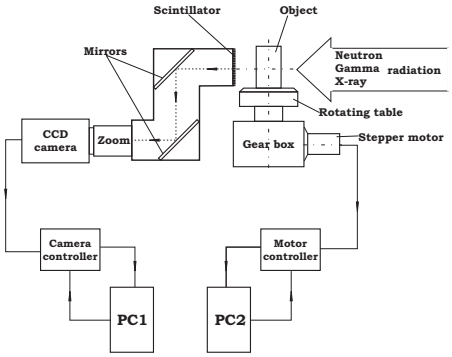


Figure 3.2. Sketch of the radiographic apparatus. (Image is courtesy of Márton Balaskó.)

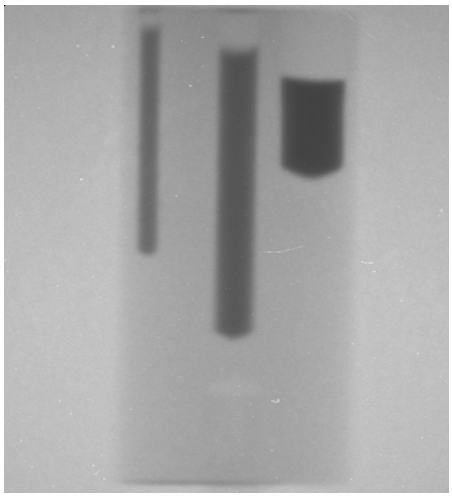


Figure 3.3. Projection of a cylindrical object taken with neutron radiation. No pre-processing has been applied. (Image source: Márton Balaskó [KFKI].)

specimen. During its journey through the object, a fraction of the radiation gets absorbed (or scattered in some direction); the rest will be captured and measured by a 2D **detector** plate placed perpendicular to the beam (again typically having a collimator in front of it in order to record only parallel rays). (The detector usually operates by emitting visible light in response to absorbed neutrons or X-ray / gamma ray photons. Such scintillations are then guided by mirrors and lenses onto a **charge-coupled device** [CCD] where they are recorded as digital electric signals.) After a preset exposure time, the stage—and hence the specimen as well—will be rotated by some amount, then the measurement procedure described above will start over with the new setting.

3.1.3 Practical Issues

Like most physical measurements, real projections are subject to some distortions and artifacts that may make the reconstruction task especially demanding. A few of these issues are listed below, and some of them can be actually observed in Figure 3.3.

- Noise: This is caused by several factors like scattering within the object, statistical errors, detector imperfections, cross-talk between detector pixels, random fluctuations of the CCD, inhomogeneities in the radiation source etc. While the statistical measurement error can be described by the Poisson probability distribution, it is much harder to account for other kinds of noise factors. Noise appears as inhomogeneous, randomized background patches or "salt-and-pepper" intensity distortions. This artifact manifests as a positive projection error term during reconstruction, and may influence the quality of reconstruction results with respect to the theoretical optima.

- Detector / CCD non-uniformity: Despite a constant radiation intensity, one part of the detector (or CCD) may record a different intensity level than other parts. This kind of distortion can be corrected to some extent when an "empty" projection is also available that records only background information (i. e. omitting the object under investigation).

- CCD pixel errors: These can appear as constant black or white spots found at a given location on every projection. They can be efficiently got rid of using median filtering.

- Intensity fluctuation: If the intensity of the beam exiting the radiation source varies in time, or the sensitivity of the CCD changes during the acquisition period, subsequent projections will accordingly exhibit a variation in the maximal intensity available.

- Under- or overexposure: Too short or too long an exposure will result in overly dark or light projections. In either case, the number of effectively available intensity levels will be diminished.

- Detector tilts: Even minute inaccuracies of detector orientation (i. e. when the detector plate is not exactly perpendicular to the beam leaving the radiation source) with respect to the laboratory frame of reference may cause various perspective distortions in projections.

- Precession of sample: Due to the inaccuracies in the placement of the turntable, the sample may precess with respect to the laboratory coordinate system. This can be corrected only if the properties of the motion are known at least approximately.

In order to overcome / eliminate the aforementioned issues, real projections need to be pre-processed in some way. Specific examples for this will be mentioned in Section 6.1.1.

3.2 Crystallography

3.2.1 Crystallographic Basics

In order to be able to compute simulated diffraction patterns, one shall be familiar with the fundamentals of X-ray diffraction as well as some crystallographic terms. Since their description is somewhat complex and they are not essential for the understanding of the rest of the thesis, the associated material has been moved into Appendix C.

3.2.2 Projection Acquisition

The method presented in Chapter 7 is based on reconstruction using high-energy **X-rays** with a set-up similar to that of conventional transmission tomography. The major difference is that in CT the absorption of the incident beam through the sample is probed, while in 3DXRD the diffracted beam is probed as it diverges from the sample on the exit side. The diffraction pattern

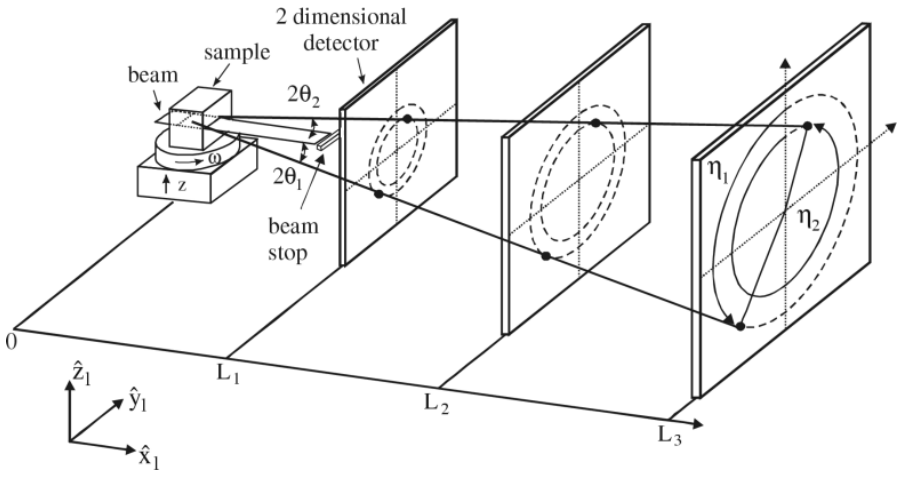


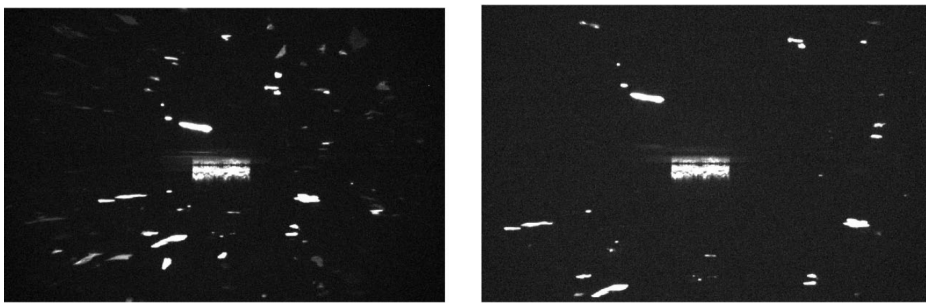
Figure 3.4. Sketch of the 3DXRD geometry. Detectors are positioned perpendicular to the beam at distance L_1, L_2 and L_3. The double 2θ of the Bragg angle, the projection angle ω, and the azimuthal angle η are indicated for a part of the microstructure that happens to give rise to diffraction. (Image source: [156].)

on the detector typically is composed of a set of distinct **diffraction spots**. Acquiring images at a set of rotation angles, each grain gives rise to ≈5–100 spots, with positions and intensity distributions determined by the local orientation of the crystalline lattice.

A sketch of the data collection methodology is shown in Figure 3.4. The apparatus presented there—dubbed a **diffractometer**—comprises the following parts: The specimen of interest is attached to a rotating table. The beam exiting the radiation source is focused into a plane perpendicular to the axis of rotation, then it is used to illuminate the sample at the appropriate vertical position (along the z-axis as shown in the figure). The majority of the radiation simply passes through the material (mostly) unchanged and gets captured by a beam stop. (This is necessary so as to avoid damaging the detector.) A very tiny fraction will, however, be subject to diffraction within the sample and deflected in some other direction (as described in Appendix C). The direction of the diffracted beam is described by two angles: 2θ, where θ is the so-called Bragg angle, and the azimuthal angle η. Diffracted photons will be then subject to interference; those interfering constructively will result in a so-called **reflection** (or diffraction spot) as measured by a 2D **detector** plate placed perpendicular to the beam at distance L from the origin O. (The detector usually operates as described in Section 3.1.2, i. e. via recording the visible light emitted by the detector with a CCD.) After a certain duration of illumination, the turntable—together with the specimen—will be rotated by some amount, then a new set of measurements will be taken using this setting.

Remark: Following the conventions introduced in [153] and elsewhere, the angle of rotation applied to the specimen will be denoted by ω ($\omega \in [-\pi, \pi[$, though the actual range is usually smaller in practice). This clearly conflicts with the analogous notation of ϑ used throughout Sections 2.1, 2.2 and Chapters 4–6, but the author wants to avoid any confusions regarding θ (i. e. the Bragg angle) and ϑ.

Note that the data acquisition described here is similar to what is usual in DT, with the important difference that the diffraction spots do not correspond to measured sums along straight lines, but rather correspond to measured sums taken over more general sets. Our approach differs in this sense from approaches formerly known in the DT literature.



(a) (b)

Figure 3.5. Real diffraction patterns of an aluminum polycrystal taken at various sample–detector distances and a fixed ω. **(a)** Projection taken at $L = 7.6\,\text{mm}$. **(b)** Projection taken at $L = 12.9\,\text{mm}$. (Image source: [156].)

In order to make use of the phenomenon of diffraction and to achieve a substantial penetration depth, monochromatic hard (i. e. high-energy) X-rays are employed, which are supplied by a synchrotron.

3.2.3 Practical Issues

Like most physical measurements, real projections are subject to some distortions and artifacts. For reference, two diffraction patterns are depicted in Figure 3.5.

The following imaging artifacts can be observed:

- Noise: This is caused by several factors like Compton (inelastic) scattering within the sample, attenuation of diffracted photons (the so-called primary extinction), secondary diffraction of already diffracted photons (dubbed the secondary extinction), statistical errors, detector imperfections, cross-talk between detector pixels, random fluctuations of the CCD, inhomogeneities in the X-ray radiation source etc. While the statistical measurement error can be described by the Poisson probability distribution, it is much harder to account for other kinds of noise factors. Noise appears as inhomogeneous, randomized background patches or "salt-and-pepper" intensity distortions (especially apparent in the right image of Figure 3.5). This artifact manifests as a positive projection error term during reconstruction, and may influence the quality of reconstruction results with respect to the theoretical optima.

- Tails of the incident beam: Though the majority of the non-diffracted incident beam is blocked by a beam stop, a small amount may still "slip through" and appear as a high-intensity smeared patch. This can be observed as a large rectangular spot in the center of both images of Figure 3.5. This artifact can be counteracted by manual pre-processing of diffraction patterns, so that all pixels lying in this area are set to zero intensity.

In addition to artifacts, the following matters also make the acquisition of diffraction patterns and subsequent reconstruction troublesome:

- Multiple sample–detector distance settings: The choice of L determines several important properties (e. g. field-of-view for reflections, angular resolution for grain orientations), so in practice measurements may be taken at multiple different settings of L (either in parallel

using multiple detectors, or sequentially by adjusting L after a full scan of ω). Such a scenario is presented in Figure 3.4 with 3 sample–detector distances.

- Detector tilts: Even minute inaccuracies of detector orientation (i. e. when the detector plate is not exactly perpendicular to the beam leaving the radiation source) with respect to the laboratory frame of reference may cause various perspective distortions in projections.
- Precession of sample: Due to the inaccuracies in the placement of the turntable, the sample may precess with respect to the laboratory coordinate system. This will manifest as certain diffraction spots appearing at a wrong location, or even at a wrong ω value.
- Overlapping diffraction spots: It may happen that the projections of two—possibly non-adjacent—grains overlap on the detector. This can be an issue in case of large orientation maps, especially in the highly deformed case, causing a further level of uncertainty.
- Multiple material phases: The presence of multiple crystal structures in the specimen means an extra degree of freedom for the pixel parameters to be determined.
- Absence of material: The reconstruction area may contain **void pixels**—inside as well as outside the specimen—that is, regions of space where sample material is missing.

3.3 Summary

A quick outline has been given for the core radiographic notions and the associated projection acquisition process. Moreover, we have also introduced the physical and crystallographic foundations for 3DXRD, including a crude overview of the imaging apparatus. As has been explained in both cases, real measurements usually involve some imaging artifacts and other complications that, nevertheless, need to be taken into account during reconstruction.

Chapter 4

Algorithm: Reconstruction of Objects Parametrized with Geometrical Primitives

The following mathematical problem was brought up while performing nondestructive testing: We are given a 3D object made of homogeneous materials whose cross-sections comprise an annulus (ring) and some disjoint discs, the number of which constituents are known beforehand. (Some everyday objects indeed have such a structure, such as tubes, ball bearings, cables etc.) The aim is to reconstruct the internal layout of the object from its projections obtained by illuminating it with some radiation (e. g. X-ray or neutron beams). It should be taken into account that the imaging process—hence also the projections—is subject to distortions by noise. Moreover, the reconstruction is to be carried out using as few projections as possible.

This chapter presents a new discrete tomographic technique for solving the aforementioned challenge. The object under consideration is expressed in terms of geometrical primitives, and the reconstruction task is formulated as an optimization problem that can be solved with Simulated Annealing.

The author published his results and achievements in [33, 104, 105, 121–123].

4.1 Motivation

Many industrial processes—like construction, manufacturing, maintenance, or security inspection—require the examination of the internal structure of some object, usually as a means of fault analysis / detection or quality assurance. It is crucial that this evaluation be accomplished without actually damaging the object under investigation. Sometimes it is even impractical or expensive—both in terms of time and money—to disassemble the object, so a different approach needs to be found.

Nondestructive testing (NDT) is a general label for a range of related examination techniques that all aim at evaluating the properties of a material, object or system without causing permanent damage. This can be achieved by a selection of methods: ultrasound, magnetic particles, liquid dyes, radiography, electromagnetic fields, heating, chemicals, surface probes, just to name a few. NDT is in widespread use in engineering, forensic science, medicine, and even art.

4.1.1 Problem Description

The following task was raised by the colleagues at the research reactor Hahn–Meitner Institut, Berlin, Germany (HMI): We are given a hollow cylindrical object containing some solid balls, which are presumably parts subject to wearing / aging. The cylinder is closed, hence only some nondestructive imaging procedure may be applicable to get information about the location and size of the balls inside. In this specific case, neutron imaging (called **neutron radiography**) was chosen. Using **neutron beams** to irradiate an object produces an image (dubbed a **projection** or **radiograph**) quite similar to everyday medical X-rays, the main difference being that neutron rays are less absorbed by metals.

The acquisition of such a radiograph is very *costly* and *time-consuming*, therefore the main goal is to devise a reconstruction method that is able to produce 3D object models of satisfactory quality using only a few (2–4) projections. It shall be noted that, due to the limitations of the imaging apparatus, only profile (i. e. side-view) images can be usually acquired. Furthermore, the reconstruction is also impeded by various measurement distortions like noise effects. For these reasons, the colleagues at HMI are struggling with the application of classical tomographic reconstruction techniques.

A similar practical challenge was met in KFKI Atomic Energy Research Institute, Budapest, Hungary (KFKI). Here, the object under investigation was a solid cylinder containing some round bored holes.

4.1.2 Contemporary Approaches and Pitfalls

As mentioned earlier, classical reconstruction techniques like FBP or ART are not suitable for the scenarios encountered at HMI and KFKI for several reasons. First, the inspected objects are composed of nearly homogeneous materials of a few kinds (the exact composition might even be known in several circumstances). Unfortunately, classical methods invariably produce continuous (i. e. when the range of the image is an infinite subset of $\mathbb{R}$) reconstructions, hence they cannot take advantage of the finite set of material types constituting the object. (There are exceptions, such as BART [84], whose results do have a finite range.) Second, these methods typically require tens, hundreds—or even thousands—of projections in order to yield satisfactory results.

The knowledge of the object's composition suggests that using some discrete tomographic (DT) reconstruction technique may be more appropriate. This is especially so because there are, most of the time, additional *a priori* data available about the object under investigation. Some examples include: statistical distribution about the structure of the object, homogeneity or smoothness of materials, geometrical constraints for object components, or similarity to some ideal / reference object. Together, these two facts—viz. the presence of a finite set of materials and other prior information about the object—enable DT methods to be successful even under unfavorable conditions like a low projection count or a high noise level.

Both practical tasks explained in Section 4.1.1 have certain properties in common that indicate that they should rather be approached as DT problems. In particular, in each case, the object to be reconstructed has a well-defined geometrical structure, so this fact can be taken as *a priori* data in addition to the knowledge of constituting materials. In fact, the reliance on geometrical priors or other parametric image descriptions is not a new idea; there are already algorithms utilizing polygons [142–144], polyhedra [32,144], solid discs [26], ellipses [183], ellipsoids [94], material microstructure [148], prototype objects [150], or reference computer-aided design (CAD) models [173]. There is even a branch of tomography—called geometric tomography—devoted to the reconstruction of arbitrary[1] planar or 3D shapes [65].

[1] To be precise, geometric tomography considers the reconstruction of finite measurable sets.

The technique presented below is a special DT reconstruction algorithm employing the additional *a priori* information that the 3D object to be reconstructed is composed of certain geometrical primitives (namely, tubes, spheres, and cylinders), so that its 2D cross-sections can be described with an annulus (circular ring) and some disjoint discs (i. e. solid circles) inside. As will be demonstrated, unlike some other approaches that resort to solving 2D cross-sections individually, our method can also perform a *truly 3D* reconstruction.

4.2 Reconstructing 2D Objects

Before coping with the original 3D problem in its entirety, we first focus on the analogous sub-problem restricted to 2D. Due to the presence of noise and other distortions in physical measurements, the reconstruction problem is formulated as an optimization task along the lines of Section 2.2.4. Therefore, given a set of 1D input projections, the goal is to reconstruct a 2D cross-section having a prescribed geometrical structure, so that its simulated projections approximate the input projections as closely as possible.

4.2.1 Prerequisites

Let

$$f\colon \mathcal{D} \to \mathcal{A}, \quad (x, y) \mapsto f(x, y) \tag{4.1}$$

denote the digital image to be reconstructed with domain $\mathcal{D} \subset \mathbb{Z}^2$ and range $\mathcal{A} \subset \{0\} \cup \mathbb{R}^+$ (both being finite; exact definitions follow below). In our applications, projections are acquired by illuminating the object of interest with some radiation and then measuring the amount of radiation passing through. The elements of $\mathcal{A}$, thus, express the ability of matter to **attenuate** (reduce) the intensity of the incident beam; these elements will be called **linear attenuation coefficient** (LAC) from now on. (A precise explanation of the measurement process, including the definition of LAC, are given in Section 3.1.1.) Accordingly, f is occasionally dubbed an **attenuation map**.

The domain $\mathcal{D}$ can be chosen according to the needs of the specific application. In the forthcoming discussion, we are going to assume that $\mathcal{D}$ is the usual rectangular lattice of square pixels in the xy plane, forming an image with dimensions SIZE $\times$ SIZE (where SIZE $\in \mathbb{Z}^+$ arbitrary):

$$\mathcal{D} := \{0, \ldots, \text{SIZE} - 1\} \times \{0, \ldots, \text{SIZE} - 1\}\,. \tag{4.2}$$

The reason for considering only squarish images is justified by the shape of the reconstruction area, which is a disc in case of transmission tomography (see Section 2.1.0.1). This definition of $\mathcal{D}$ will be later extended for the 3D case too; see Section 4.3.1, Equation (4.11).

Image Model

The motivating practical applications introduced in Section 4.1.1 suggest that one shall exploit the *a priori* information available about the object, and accept only conforming images f as solutions. In particular, f is said to be **valid** if and only if it satisfies the following restrictions:

- Geometrical structure:
  - f consists of exactly one concentric **annulus** (circular **ring**) and some **discs** (possibly none).

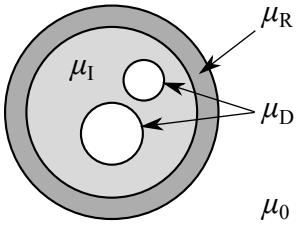


Figure 4.1. Geometrical structure and material composition of the 2D object to be reconstructed.

  - The annulus should fit within the reconstruction area of the image, and its thickness is to be nonnegative. (See below for an explanation why a thickness of zero is also permitted.)
  - Discs must be pairwise disjoint and located entirely inside the annulus. Moreover, their radius should be positive (actually, not smaller than some fixed minimal radius[2] $r_{\min}$).

- Material composition:
  - f contains at most four kinds of materials, as shown in Figure 4.1:
    * The vacuum / air surrounding the object having LAC μ_0. (Actually, $\mu_0 = 0$ can be assumed in most practical applications.)
    * The material of the ring having LAC μ_R.
    * The material of the area inside the ring having LAC μ_I. From now on, this area will be dubbed the **interior of the ring** (or simply the **interior**).
    * The material of the discs having LAC μ_D.
  - Each LAC must be a nonnegative real given in unit $\frac{1}{\text{pixel}}$.
  - Equality among LAC values is permitted. (See below for some remarks.)
  - All aforementioned materials are expected to be completely **homogeneous**.

The range $\mathcal{A}$ of f then becomes

$$\mathcal{A} := \{\mu_0, \mu_R, \mu_I, \mu_D\}\,, \tag{4.3}$$

hence f is a **quaternary** (i. e. four-valued) image. As a special case, the classical **binary** images considered in many other DT applications can be obtained using settings $\mu_0 = \mu_I = 0$ and $\mu_R = \mu_D = 1$. The aforementioned definition of $\mathcal{A}$ will be later extended to cover the 3D model as well; see Section 4.3.1, Equation (4.14).

It is clear that any circle is uniquely determined by its center $O = (x, y)$ and radius r, which will be called the **parameters of the circle** (O, r) henceforth. Therefore, any circle can be defined with a triplet of reals (subject to some geometrical restrictions; see below). The complete structure of the object can, thus, be described with the **configuration** (or **model**)

$$\mathbf{c} := (O_{RE}, r_{RE}, O_{RI}, r_{RI}, O_1, r_1, \ldots, O_N, r_N)\ . \tag{4.4}$$

[2] Having such a lower bound on permitted radii is a sort of heuristics, thus forbidding the creation of tiny (possibly "invisible") shapes.

Here, $(O_{\mathrm{RE}}, r_{\mathrm{RE}})$ and $(O_{\mathrm{RI}}, r_{\mathrm{RI}})$ describe the **external** and **internal circles** forming the ring, respectively. (At the same time, $(O_{\mathrm{RI}}, r_{\mathrm{RI}})$ also determines the boundaries of the interior.) The rest of the parameters (O_i, r_i) $(1 \leq i \leq N)$ define the discs, where $N \in \mathbb{N}$ denotes their number.[3] (The corresponding coordinates of the center are also subscripted in a similar way, e. g. $O_{\mathrm{RE}} = (x_{\mathrm{RE}}, y_{\mathrm{RE}})$.) Since the annulus should be concentric, the setting $O_{\mathrm{RE}} = O_{\mathrm{RI}}$ is enforced in the current implementation.

The solution space $\mathcal{C}$ (also called **configuration space**) contains all candidate configurations:[4]

$$\mathcal{C} \coloneqq \mathbb{R}^{6+3\cdot N}, \quad \mathbf{c} \in \mathcal{C}\,. \tag{4.5}$$

When LAC values are not known exactly, it may be beneficial to include them as well in $\mathbf{c}$ and, hence, in $\mathcal{C}$:

$$\begin{aligned} \mathbf{c} &\coloneqq (\mu_0, \mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}, O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}}, O_1, r_1, \ldots, O_N, r_N) \\ \mathcal{C} &\coloneqq \mathbb{R}^{4+6+3\cdot N} \end{aligned}\,. \tag{4.6}$$

These definitions of $\mathbf{c}$ and $\mathcal{C}$ will be later extended for the 3D case too; see Section 4.3.1, Equations (4.15), (4.16), (4.17) and (4.18).

It should again be stressed that all aforementioned circle parameters are *reals*. That is, contrary to the discreteness of the domain $\mathcal{D}$, we are looking for a *continuous* configuration $\mathbf{c}$. While this choice makes $\mathcal{C}$ much bigger, it also more closely reflects real-life situations.

Given a configuration $\mathbf{c}$, one can define the digital image $f_{\mathbf{c}}$ determined by $\mathbf{c}$:

$$f_{\mathbf{c}} \colon \mathcal{D} \to \mathcal{A}, \quad (x, y) \mapsto f_{\mathbf{c}}(x, y)\,, \tag{4.7}$$

where $\mathcal{D}$ and $\mathcal{A}$ are as discussed earlier. (The exact procedure for how such a $f_{\mathbf{c}}$ is constructed will be described in Section 5.2.1; see Equation (5.5).) Since we are not interested in arbitrary images f but only in valid ones, we shall solely focus on reconstructing $f_{\mathbf{c}}$ hereafter.

Assumptions

The following assumptions are made:

- The object to be reconstructed has the geometrical structure and material composition described above.
- The number N of discs is known exactly.
- LAC values μ_0, μ_{R}, μ_{I} and μ_{D} are known at least approximately.[5]

Geometrical Constraints

The configuration $\mathbf{c}$ is said to be **valid** if and only if it obeys the geometrical restrictions discussed earlier. By definition, the associated $f_{\mathbf{c}}$ is valid only if $\mathbf{c}$ is also valid, therefore it is feasible—and beneficial—to formulate these **constraints** in a geometrical framework. As will be demonstrated, the restrictions are verified *analytically* based on the components of $\mathbf{c}$, that is avoiding any digitization involved in the construction of $f_{\mathbf{c}}$.

[3] Note that the same notations (O_i, r_i) will be later employed to define spheres and cylinders in the 3D configuration. Similarly, N will then denote the number of spheres or cylinders.

[4] This definition is slightly imprecise as it permits both O_{RE} and O_{RI}, even though they are strongly tied together. Moreover, any further restrictions—e. g. the positivity of radii—are ignored here for simplicity.

[5] This can be problematic in case of real physical measurements; the algorithm presented in Section 4.5 will, therefore, offer some means to determine these parameters from the input projections.

1. The annulus is concentric.

   This is established by the conditions

   $$\begin{aligned} O_{\mathrm{RE}} &= O_{\mathrm{RI}} \\ r_{\mathrm{RI}} &< r_{\mathrm{RE}} \end{aligned} .$$

   When concentricity is not strictly enforced, the following needs to hold instead:

   $$\overline{O_{\mathrm{RE}}O_{\mathrm{RI}}} + r_{\mathrm{RI}} < r_{\mathrm{RE}} ,$$

   where $\overline{AB}$ denotes the length of the line segment with end points A and B.

2. The annulus fits within the reconstruction area of the image.

   It is enough to ensure that

   $$\overline{O_{\mathrm{RE}}O_{\mathrm{C}}} + r_{\mathrm{RE}} \leq r_{\max} ,$$

   where

   $$\begin{aligned} O_{\mathrm{C}} &= (x_{\mathrm{C}}, y_{\mathrm{C}}) \\ x_{\mathrm{C}} \coloneqq y_{\mathrm{C}} \coloneqq r_{\max} &\coloneqq \frac{\mathrm{SIZE} - 1}{2} \cdot \end{aligned} \tag{4.8}$$

   Here, the reconstruction area is represented by the circle $(O_{\mathrm{C}}, r_{\max})$. (The aforementioned definition of O_{C} will be later extended to 3D configurations as well; see Section 4.3.1, Equation (4.12).)

3. The thickness of the annulus is nonnegative.

   This requires

   $$r_{\mathrm{RI}} \leq r_{\mathrm{RE}} ,$$

   thus amending the respective condition in item 1. above.

   Permitting a ring of zero thickness can be useful for situations like the one raised by KFKI colleagues. Specifically, it can be used to model the cross-section of a 3D object consisting of a solid cylinder containing some spherical or cylindrical cavities. This also implies that μ_{R} is irrelevant in such cases. (Internally, this is realized by setting μ_{R} to zero.)

4. Discs are located entirely inside the annulus.

   This means that, for $1 \leq i \leq N$,

   $$\overline{O_{\mathrm{RI}}O_i} + r_i \leq r_{\mathrm{RI}} .$$

5. Discs are pairwise disjoint.

   The following needs to hold:

   $$\overline{O_iO_j} \geq r_i + r_j \quad (\forall 1 \leq i, j \leq N,\, i \neq j) .$$

6. The radius of any disc is not smaller than $r_{\min}$. (Actually, this implies that $r_{\mathrm{RE}} \geq r_{\min}$ and $r_{\mathrm{RI}} \geq r_{\min}$.)

   This requires

   $$r_i \geq r_{\min} \quad (\forall 1 \leq i \leq N) .$$

The following conditions handle a few edge cases, ensuring that the reconstruction problem remains well-defined:

7. LAC values μ_{R}, μ_{I} and μ_{D} are not all zero.

8. $\mu_{\mathrm{I}} = \mu_{\mathrm{D}}$ is allowed only if $N = 0$.

   This, essentially, makes μ_{D} irrelevant.

9. $\mu_{\mathrm{R}} = \mu_{\mathrm{I}}$ implies $r_{\mathrm{RI}} = r_{\mathrm{RE}}$.

   Note: As mentioned in item 3. above, $\mu_{\mathrm{R}} = 0$ will be assumed in this case.

10. $\mu_{\mathrm{R}} = \mu_{\mathrm{I}} = 0$ is allowed only if $N > 0$.

    This point allows the modeling of more general objects consisting of a collection of disjoint discs without any enclosing interior or annulus. As a consequence, parameters $(O_{\mathrm{RE}}, r_{\mathrm{RE}})$ and $(O_{\mathrm{RI}}, r_{\mathrm{RI}})$ are rendered irrelevant. (In the current implementation, the settings $O_{\mathrm{RE}} \coloneqq O_{\mathrm{RI}} \coloneqq O_{\mathrm{C}}$ and $r_{\mathrm{RE}} \coloneqq r_{\mathrm{RI}} \coloneqq r_{\max}$ will be applied.)

In order to compensate for numerical round-off errors, all comparisons are carried out using some tolerance. For instance, instead of checking for $a > b$, $c < d$ or $e = f$, the conditions $a - b > \varepsilon$, $c - d < -\varepsilon$ and $|e - f| \leq \varepsilon$ will be verified, where ε is a small positive real. (A similar approach is used when comparing for $\leq$ or $\geq$ involving real values.)

4.2.2 Objective Function

In order to solve the reconstruction of $f_{\mathbf{c}}$ introduced in Section 4.2.1, one needs to find an appropriate configuration $\mathbf{c}$ in the solution space $\mathcal{C}$. It is easy to see that the size of the configuration space is enormous (actually, infinite), namely $|\mathcal{C}| = \mathfrak{c}$ (i. e. the cardinality of the continuum). As mentioned in Section 2.2.2, most direct graph search techniques are simply too slow for practical usage on such large spaces.[6] The reconstruction problem is, therefore, formulated as an optimization task with a suitably chosen objective function $\gamma(f_{\mathbf{c}})$. One major advantage of such an approach is that it stays robust against most of the noise and distortions found in input measurements.

Motivated by the literature of stochastic optimization (see Section 2.2.3), we decided to model $\gamma(f_{\mathbf{c}})$ as a Boltzmann distribution. This, first, needs the notion of the energy (Hamiltonian) of $f_{\mathbf{c}}$:

$$H(f_{\mathbf{c}}) \coloneqq \sum_{\vartheta} \left\| \mathbf{P}_{f_{\mathbf{c}},\vartheta} - \mathbf{P}_{\vartheta} \right\|_2^2 . \tag{4.9}$$

Here, $\mathbf{P}_{f_{\mathbf{c}},\vartheta} \colon \mathcal{D}_{\mathbf{P}} \to \{0\} \cup \mathbb{R}^+$ denotes the simulated discrete projection of $f_{\mathbf{c}}$ at direction ϑ, $\mathbf{P}_{\vartheta} \colon \mathcal{D}_{\mathbf{P}} \to \{0\} \cup \mathbb{R}^+$ the input—i. e. measured—discrete projection at direction ϑ, and $\mathcal{D}_{\mathbf{P}} \subset \mathbb{Z}$ the finite projection domain. (The exact definition of the latter as well as the computation of $\mathbf{P}_{f_{\mathbf{c}},\vartheta}$ will be given later in Section 5.2.3; see Equations (5.10) and (5.15). Moreover, the meaning of these notations will be later extended for the 3D case as well; see Section 4.3.2.)

As can be seen, $H(f_{\mathbf{c}})$ simply calculates the square of the projection error (measured with the ℓ_2 norm) summed over all directions ϑ. While it is superficially similar to the $\gamma(f)$ defined in Equation (2.13), it is in fact of the form of $\gamma_{\Phi}(f)$ shown in Equation (2.14). In particular, $\alpha = 1$, while the fitness $\Phi(f)$ is implicitly encoded by the geometrical model and constraints associated with $\mathbf{c}$.

[6] Even though $\mathcal{C}$ could be made finite by permitting only parameter values up to some fixed precision (i. e. via quantization), the resulting space would *still* be too large for direct searches.

Having defined $H(f_{\mathbf{c}})$, now the aim is to *maximize* the **objective function**

$$\gamma(f_{\mathbf{c}}) \coloneqq \frac{1}{Z_{T;f_{\mathbf{c}}}} \mathrm{e}^{-\frac{H(f_{\mathbf{c}})}{k_{\mathrm{B}}T}}\,, \tag{4.10}$$

where $Z_{T;f_{\mathbf{c}}}$ is the partition function, k_{B} the Boltzmann constant, and T the temperature, respectively (cf. Equation (2.15)). As a matter of fact, the optimal $\hat{f}_{\mathbf{c}}$ that maximizes $\gamma(f_{\mathbf{c}})$ simultaneously *minimizes* $H(f_{\mathbf{c}})$:

$$\hat{f}_{\mathbf{c}} \coloneqq \arg\max_{f_{\mathbf{c}}} \gamma(f_{\mathbf{c}}) = \arg\min_{f_{\mathbf{c}}} H(f_{\mathbf{c}})\,.$$

That is, by maximizing $\gamma(f_{\mathbf{c}})$ we are looking for the $\hat{f}_{\mathbf{c}}$ that results in the lowest projection error—and hence in the lowest $H(f_{\mathbf{c}})$.

It should be stressed that any solution $\hat{f}_{\mathbf{c}}$ must be a *valid* one (with respect to the geometrical constraints). Since $H(f_{\mathbf{c}})$ is basically undefined for invalid configurations, $\gamma(f_{\mathbf{c}})$ has lots of discontinuities that poses extraordinary challenges for the optimization process.

4.2.3 Optimization

The optimization of $\gamma(f_{\mathbf{c}})$ is carried out by utilizing SA, an iterative stochastic global optimization method (see Section 2.2.3). As explained before, instead of trying to reconstruct $f_{\mathbf{c}}$ directly by modifying individual pixels, the optimization takes place in the configuration space $\mathcal{C}$.

In each iterative step, a new candidate configuration $\mathbf{c}'$ is constructed by perturbing the current $\mathbf{c}$ in either of the following ways:

1. The center $O_{\mathrm{RE}} = O_{\mathrm{RI}}$ of the annulus or the O_i of one of the discs is translated along a line in some direction by some distance d.[7]

2. The external (r_{RE}) or internal (r_{RI}) radius of the annulus or the r_i of one of the discs is increased or decreased by some amount Δr.

3. One of the LAC values $\{\mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}\}$ gets increased or decreased by some amount $\Delta\mu$ while keeping them within a prescribed range.[8]

The choice among the aforementioned 3 rules, as well as that among the respective parameters, is made with equal probability. Furthermore, the algorithm first attempts to determine the range R of valid alterations, then picks d, Δr or $\Delta\mu$ from R randomly.[9]

If the $\mathbf{c}'$ obtained so happens to violate some constraints, the magnitude of alteration is corrected as $d \coloneqq \lambda_1 d$ (similarly for Δr and $\Delta\mu$), then $\mathbf{c}'$ gets updated accordingly. (Here the $0 < \lambda_1 < 1$ real determines the size of the correction.) This step is repeated until either $\mathbf{c}'$ becomes valid, or the magnitude of alteration falls below some fixed threshold ε_1. In the latter case, the invalid $\mathbf{c}'$ is discarded and a new iterative step is started.

Depending on the user's choice, the construction of $\mathbf{c}'$ can happen in 3 different manners: (a) allow only geometrical changes (i. e. translation of a center or modification of a radius) and hold $\{\mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}\}$ fixed; (b) allow only changes to LAC values and hold the geometrical model fixed; (c) allow both kinds of changes.[10]

[7] The distance and direction are expressed as polar coordinates.

[8] Actually the algorithm assumes that $\mu_0 \equiv 0$ throughout the optimization.

[9] The present implementation does not aim for a perfect R; its value is only approximately correct when the perturbation involves the translation of a center.

[10] In fact (b) would only make sense if the task was to determine the LAC values with respect to a fixed, known geometrical model. Thus only (a) and (c) are employed in practice.

The optimization procedure can, in theory, be started from an arbitrary initial configuration $\mathbf{c}_0$, but—as explained in Section 2.2.3—having a $\mathbf{c}_0$ close enough to the expected global optimum will result in faster and more successful reconstructions. The exact way of its construction is discussed in Section 4.5.

The execution of SA also needs some further details to be specified: initial and minimal temperatures (T_0, $T_{\min}$), annealing schedule, and termination criteria. These will be given later in Section 5.2.6.

4.3 Reconstructing 3D Objects

Building upon the foundations presented in Section 4.2, we can now continue with the original aim of reconstructing 3D objects whose structure can be described with a couple of geometrical primitives. The algorithm discussed below is a direct extension and generalization of the former 2D approach to 3D. As will be shown, this is done so that *truly 3D* reconstructions can be obtained instead of simply reconstructing individual 2D cross-sections. Like for its 2D counterpart, the reconstruction process is expressed as an optimization task.

4.3.1 Prerequisites

Most concepts introduced in Section 4.2.1 can be redefined for 3D in a pretty straightforward way. For brevity, only the most important ones of these notions will be included here; the rest can be devised by analogy.

We begin with establishing the domain $\mathcal{D} \subset \mathbb{Z}^3$ of the 3D digital image f to be reconstructed (cf. Equation (4.2)):

$$\mathcal{D} \coloneqq \{0, \ldots, \mathrm{SIZE} - 1\} \times \{0, \ldots, \mathrm{SIZE} - 1\} \times \{0, \ldots, \mathrm{HEIGHT} - 1\}\,. \tag{4.11}$$

That is, $\mathcal{D}$ is a square box (square cuboid) so that SIZE gives the length of the sides of 2D cross-sections in the xy plane, while $\mathrm{HEIGHT} \in \mathbb{Z}^+$ denotes the height of the image volume along the z-axis. Accordingly, the reconstruction area becomes the cylinder $(O_\mathrm{C}, r_{\max}, h_{\max})$, where (cf. Equation (4.8))

$$\begin{aligned} O_\mathrm{C} &= (x_\mathrm{C}, y_\mathrm{C}, z_\mathrm{C}) \\ x_\mathrm{C} \coloneqq y_\mathrm{C} \coloneqq r_{\max} &\coloneqq \frac{\mathrm{SIZE} - 1}{2} \\ z_\mathrm{C} &\coloneqq \frac{\mathrm{HEIGHT} - 1}{2} \\ h_{\max} &\coloneqq \mathrm{HEIGHT} \end{aligned}\,. \tag{4.12}$$

Image Model

The image f is said to be valid if and only if it complies with the following restrictions:

- Geometrical structure:
  - f consists of exactly one concentric **tube** and some **spheres** or **cylinders** (possibly none, but only either kind of primitive).
  - The tube should fit within the reconstruction area of the image, and its thickness is to be nonnegative. Furthermore, the **external** and **internal cylinders** forming the tube shall have common centers and heights.

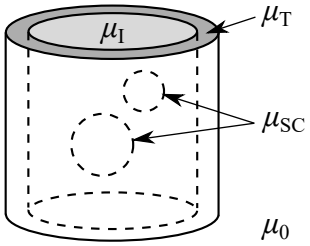


Figure 4.2. Geometrical structure and material composition of the 3D object to be reconstructed, assuming spheres within the interior.

  - Spheres (respectively, cylinders) must be pairwise disjoint and located entirely inside the tube. Moreover, their radius should be not smaller than $r_{\min}$, and the height of cylinders should reach at least $h_{\min}$.
  - The radii and heights of the tube and cylinders are bounded by $r_{\max}$ and $h_{\max}$. Moreover, the radius of spheres has the upper bound

$$r_{\max\mathrm{S}} \coloneqq \min\left\{r_{\max}, \frac{h_{\max}-1}{2}\right\} . \tag{4.13}$$

  - The axis of symmetry of the tube and all the cylinders must be parallel to the z-axis.

- Material composition:
  - f contains at most four kinds of materials, as shown in Figure 4.2:
    * μ_0 and μ_{I} are kept as is.
    * The interior (now also dubbed the **interior of the tube**) means the area inside the tube. These primitives will, therefore, have the same centers and heights.
    * The LAC μ_{T} of the tube replaces μ_{R}, while the LAC μ_{SC} of spheres and cylinders takes the place of μ_{D}.
  - All aforementioned materials are still expected to be completely homogeneous.

Thus the range $\mathcal{A}$ of f can be written as (cf. Equation (4.3))

$$\mathcal{A} \coloneqq \{\mu_0, \mu_{\mathrm{T}}, \mu_{\mathrm{I}}, \mu_{\mathrm{SC}}\} . \tag{4.14}$$

Given the center $O = (x, y, z)$ and radius r, which will be called the **parameters of the sphere**, spheres will be denoted with the shortcut (O, r). Similarly, given the height h (the third **parameter of the cylinder**), cylinders are represented as (O, r, h).

The 3D configuration $\mathbf{c}$ then becomes (cf. Equation (4.4))

$$\mathbf{c} \coloneqq (O_{\mathrm{TE}}, r_{\mathrm{TE}}, h_{\mathrm{TE}}, O_{\mathrm{TI}}, r_{\mathrm{TI}}, h_{\mathrm{TI}}, O_1, r_1, \ldots, O_N, r_N) . \tag{4.15}$$

when having spheres inside the interior, and

$$\mathbf{c} \coloneqq (O_{\mathrm{TE}}, r_{\mathrm{TE}}, h_{\mathrm{TE}}, O_{\mathrm{TI}}, r_{\mathrm{TI}}, h_{\mathrm{TI}}, O_1, r_1, h_1, \ldots, O_N, r_N, h_N) . \tag{4.16}$$

when having cylinders instead. Here, $(O_{\mathrm{TE}}, r_{\mathrm{TE}}, h_{\mathrm{TE}})$ and $(O_{\mathrm{TI}}, r_{\mathrm{TI}}, h_{\mathrm{TI}})$ describe the external and internal cylinders of the tube, respectively. The remaining parameters (O_i, r_i) and (O_i, r_i, h_i)

($1 \leq i \leq N$) correspond to the individual spheres and cylinders inside the interior, respectively. According to the notes above, the settings $O_{\mathrm{TE}} = O_{\mathrm{TI}}$ and $h_{\mathrm{TE}} = h_{\mathrm{TI}}$ are enforced in the current implementation.

The configuration space $\mathcal{C}$ thus turns into (cf. Equation (4.5))

$$\mathcal{C} := \mathbb{R}^{10+4 \cdot N} \tag{4.17}$$

or

$$\mathcal{C} := \mathbb{R}^{10+5 \cdot N} \tag{4.18}$$

depending on whether spheres or cylinders are being used, respectively. (Like in the 2D scenario, $\mathcal{C}$ can optionally be further extended to include LAC values as well.)

As a special case, any 2D configuration can be represented in 3D with the settings HEIGHT $= h_{\mathrm{TE}} = h_{\mathrm{TI}} = h_i = h_{\min} = 1$, so that every disc becomes a cylinder of height 1.

The definitions of f and $f_{\mathbf{c}}$, as well as the assumptions and geometrical constraints for them are taken over and extended from their 2D counterparts.

4.3.2 Objective Function

Everything discussed in Section 4.2.2 remains valid and applicable to 3D configurations as well. In particular, the Hamiltonian $H(f_{\mathbf{c}})$ expresses the "energy" of $f_{\mathbf{c}}$ in terms of the projection error, while $\gamma(f_{\mathbf{c}})$ is formulated as a Boltzmann distribution. The major difference is that the simulated projections $\mathbf{P}_{f_{\mathbf{c}},\vartheta}$ and input projections $\mathbf{P}_{\vartheta}$ become two-dimensional, defined over the domain $\mathcal{D}_{\mathbf{P}} \subset \mathbb{Z}^2$ (see Section 5.2.3, Equation (5.11) for the actual definition).

4.3.3 Optimization

The optimization of $\gamma(f_{\mathbf{c}})$ is performed with SA along the lines of Section 4.2.3; most of the concepts and remarks mentioned there also apply to the 3D scenario. The only change needed is the way how the new candidate configuration $\mathbf{c}'$ is obtained from the current one $\mathbf{c}$:

1. One of the centers $O_{\mathrm{TE}} = O_{\mathrm{TI}}$ or O_i gets translated along a line in some direction by some distance.[11]
2. Either radius r_{TE}, r_{TI}, r_i is increased or decreased by some amount.
3. A height parameter $h_{\mathrm{TE}} = h_{\mathrm{TI}}$ or h_i is increased or decreased by some amount.
4. One of the LAC values $\{\mu_{\mathrm{T}}, \mu_{\mathrm{I}}, \mu_{\mathrm{SC}}\}$ gets modified by some amount.

Like before, the choice among these rules and the associated configuration parameters is made with equal probability.

Similarly to the 2D case, optimization is started from an initial configuration $\mathbf{c}_0$ based on the input projections $\mathbf{P}_{\vartheta}$; see Section 4.5 for more details about its construction.

4.4 Generating Random Configurations

During the implementation and subsequent evaluation of the reconstruction algorithm described earlier in this chapter, it became clear that it would have been desirable to be able to randomly generate the parameters of geometrical primitives, or even to generate entire configurations $\mathbf{c}$.

[11] The distance and direction are expressed as spherical coordinates.

Specifically, such an ability would be useful for generating test configurations for reconstruction, building random initial configurations $\mathbf{c}_0$, or repairing an incomplete $\mathbf{c}_0$ by adding missing geometrical primitives; these are explained in Sections 4.5 and 5.2.2.

The approach devised by the author is guided by the following principles:

1. The generated configuration $\mathbf{c}$ is valid, i. e. it satisfies the structural requirements and geometrical constraints discussed earlier.
2. It is possible to generate all elements of the configuration space $\mathcal{C}$.
3. Any potential $\mathbf{c}$ is chosen uniformly (i. e. with equal probability).
4. The generation process is reasonably robust. Especially, it tries to avoid getting into infinite loops.
5. The generation of $\mathbf{c}$ is fast and has modest resource requirements.
6. The process is totally automatic with some user-tunable parameters.

The generation algorithm is presented in Appendix D in its entirety; here we resort to a simplified description only. The overview of the process for 2D configurations is shown in Algorithm 4.1. As can be seen, the parameters of the ring are generated first, followed by the centers O_i of discs (with the temporary setting $r_i \coloneqq r_{\min}$), and finally the radii r_i.

Algorithm 4.1. Generation of a 2D random configuration (summary).

```
1  Let c be the empty configuration
2  Choose O_RE = O_RI randomly
3  Choose r_RE and r_RI randomly
4  Add annulus to c
5  foreach 1 ≤ i ≤ N do
6      Determine if it is possible to add a new disc (O_i, r_i) to c with r_i ≥ r_min
7      if No more discs can be added then
8          Error
9      Choose O_i randomly by assuming r_i := r_min
10     Add disc (O_i, r_i) to c with r_i = r_min
11 foreach 1 ≤ i ≤ N do
12     Determine the valid range for r_i with respect to other geometrical primitives in c
13     Choose r_i randomly and update it in c
14 c contains the generated configuration
```

When generating O_i, the algorithm needs to decide if there is a chance at all to find a suitable disc center. This step, shown in line 6, tries to estimate $r' \coloneqq \max r_i$ based on the current partial configuration $\mathbf{c}$; this is controlled by the positive real δ that determines the precision of r'.

The generation of 3D configurations is performed by an analogous approach, but of course establishing parameters for the tube, spheres and cylinders. For sake of easier testing of special cases, it is also possible to optionally restrict the z coordinate of the centers of all spheres and cylinders to a given range.

4.5 Constructing Initial Configurations

In order to speed up the convergence of the reconstruction process, as well as to hopefully avoid unwanted local optima, the optimization of $\gamma(f_{\mathbf{c}})$ is started from a suitably chosen initial

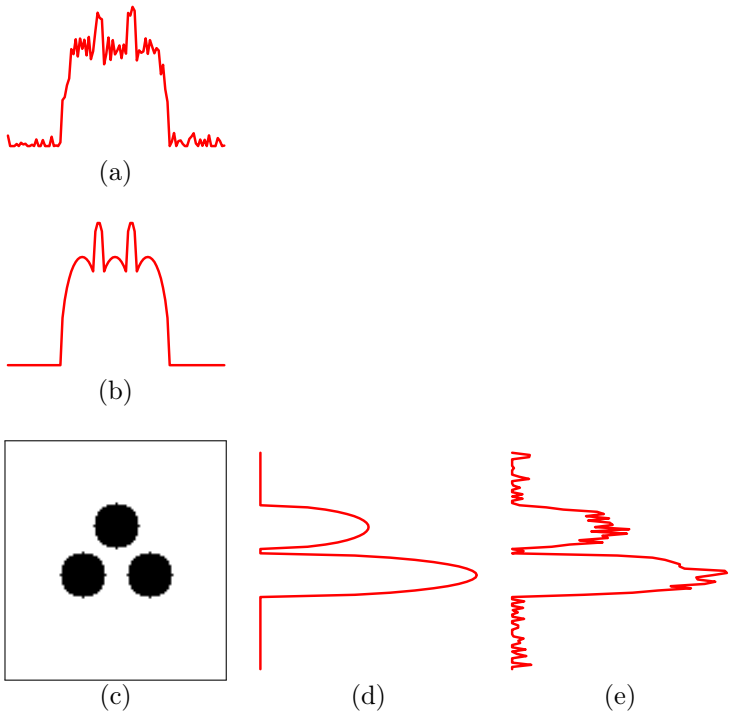


Figure 4.3. Superposition of the projection of multiple discs. Pixel intensities correspond to 0 / background (white) and 1 / foreground (black). **(a)** Vertical projection of (c) at 10 % noise level. **(b)** Noiseless vertical projection of (c). **(c)** Binary image containing three discs. (Frame added for better visibility.) **(d)** Noiseless horizontal projection of (c). **(e)** Horizontal projection of (c) at 10 % noise level.

configuration $\mathbf{c}_0$. In particular, $\mathbf{c}_0$ is constructed by scanning the input projections $\mathbf{P}_\vartheta$ so as to extract the approximate parameters of constituting geometrical shapes. (The exact procedure is discussed below in detail.)

This is, however, a very demanding task in general and cannot be solved by geometrical means alone. On the one hand, measurements can become reasonably distorted as compared to ideal, noiseless projections. On the other hand, every time a projection line crosses multiple shapes, the measured value will equal the **superposition** (i. e. sum) of respective elementary projections. In practice, this may manifest both as fewer and as more maxima than expected. This situation is demonstrated in Figure 4.3. The image shown there contains three discs; still, the number of maxima appearing is 2 (horizontal projection) and 5 (vertical projection), respectively. The problem only gets worse with the addition of noise or other distortions.

The algorithm developed by the author is somewhat complex, the application of which will nevertheless highly improve the overall reconstruction success rate. (See Sections 5.2.4 and 5.3 for the relevant results.) Therefore a high-level summary will only be presented here; the complete description, theoretical background and related pseudocode can be found in Appendix E.

The approach chosen is based on geometrical principles combined with some heuristics. As explained above, no perfect result is expected from this procedure alone. In particular, the algorithm may not always find all the discs, spheres or cylinders, or their parameters may not match their ideal settings. Such discrepancies are, however, totally acceptable since $\mathbf{c}_0$ will be

subject to further optimization.

As will be demonstrated, the algorithm attempts to perform a direct *geometrical reconstruction* by analyzing input projections $\mathbf{P}_\vartheta$; this is done in an entirely automated and deterministic[12] fashion. The procedure takes just a handful of input parameters: $\mathbf{P}_\vartheta$, set $\{\vartheta_1, \ldots, \vartheta_n\}$ of projection angles, N, and $\mathcal{A}$.[13] It is also assumed that the projection geometry and the computation of simulated projections are defined as shown later in Section 5.2.3.

The 3D initial configuration $\mathbf{c}_0$ will be constructed as follows:

1. First, the initial locations of the tube and its interior are estimated from the projections (see Figure 4.4(a)). This is done after using filtered versions of the input projections, obtained by applying a Gaussian and several averaging filtering kernels to every projection.[14] The boundary of the projection of the tube, namely the coordinates of its axis of symmetry, its external radius, and its upper and lower ends can be found by defining a threshold for the background noise, and scanning the filtered projections for the leftmost, rightmost, uppermost, and lowermost values above this threshold. The boundary of the interior of the tube can be determined in a similar way. The threshold is calculated as a user-specified percentage of the global maximum value of the projections, based on an assumed level of noise.

2. Using the location of the tube and its interior, their projections can be subtracted from the input projections.[15] This is shown in Figure 4.4(b).

3. Next, the initial positions of the spheres or cylinders within the interior are to be guessed based on the reduced projections (see Figure 4.5). In the present implementation this task has, for simplicity, been reduced to 2D sub-problems. Currently this procedure is performed in two steps:

   (a) First, the locations of discs are found in each cross-section separately (see Figure 4.5(a)). That is, the intersections of the spheres or cylinders with the 2D cross-sections orthogonal to the z-axis have to be found. Exact details are to follow below. (Note: The number of discs in a given cross-section is unknown; this may be smaller than N, so partial 2D configurations are permitted.)

   (b) Lastly, the initial positions of the spheres or cylinders are determined from the discs found in neighboring cross-sections. Specifically, the discs detected in the previous step are treated as solid cylinders of unit height. An auxiliary 3D configuration can be built by stacking the cross-sections onto each other, that is placing these discs into a 3D coordinate system. This configuration is then examined to find candidate spheres and cylinders that will be included in $\mathbf{c}_0$. (These are located by successively merging the stacked discs between neighboring[16] cross-sections: a disc on section z_i and another on section $z_i + 1$ will be merged into a single object when their centers are close enough to each other and the difference between their radii is smaller than some threshold.) This is demonstrated in Figure 4.6.

The processing of 2D cross-sections is accomplished by the following greedy algorithm:

[12] This, of course, does not apply to the discs, spheres and cylinders generated randomly in case of a partial configuration. See later for an explanation.

[13] In fact, $\mu_0 \equiv 0$ is assumed during the course of the algorithm.

[14] In general, filtering and other "destructive" operations will be always applied to a temporary copy of the input projections; the actual $\mathbf{P}_\vartheta$ will be left intact and used later during the optimization of $\gamma(f_\mathbf{c})$.

[15] Again, this is only done to a copy of $\mathbf{P}_\vartheta$.

[16] Optionally, merging of discs can be also enabled between nearby—but not necessarily adjacent—cross-sections.

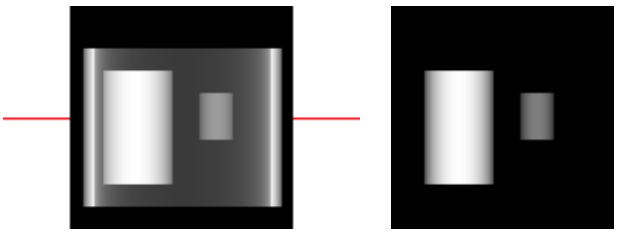


Figure 4.4. Elimination of the tube from 2D projections. Gray levels represent the amount of attenuation from zero (black) to maximum (white). **(a)** One of the noiseless input projections of the 3D object. The red lines indicate the cross-section considered in Figure 4.5(a). **(b)** Reduced projection obtained from (a) by subtracting the projection of the tube and its interior.

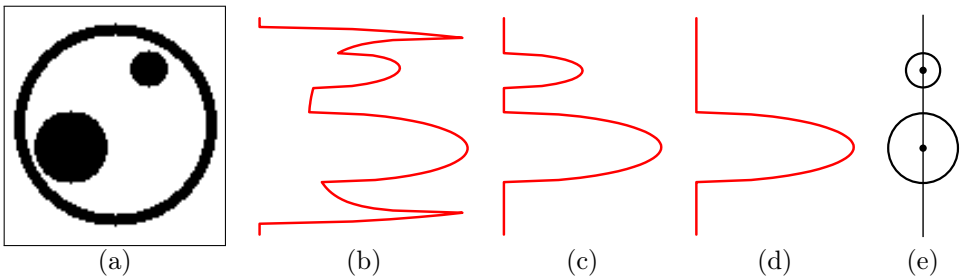


Figure 4.5. Detection of discs in 2D cross-sections after the elimination of the tube from 2D projections (or the ring from 1D projections). Pixel intensities correspond to 0 / background (white) and 1 / foreground (black). **(a)** Binary cross-section of the object depicted in Figure 4.4(a). (Frame added for better visibility.) **(b)** Noiseless 1D horizontal projection of (a). **(c)** Reduced projection obtained from (b) by eliminating the ring. **(d)** Elimination of the projection of a disc from (c). **(e)** Candidate discs detected in (c).

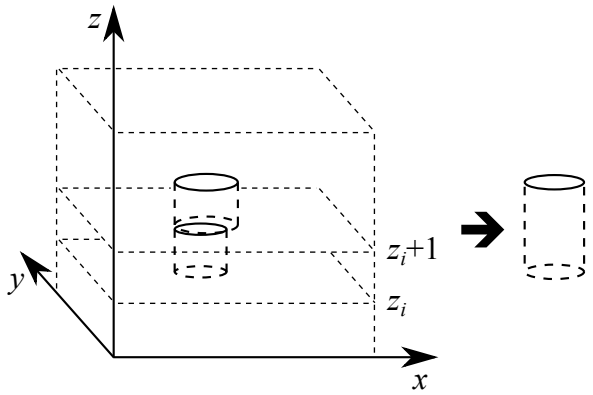


Figure 4.6. Building a 3D initial configuration by stacking the 2D configurations determined for individual cross-sections.

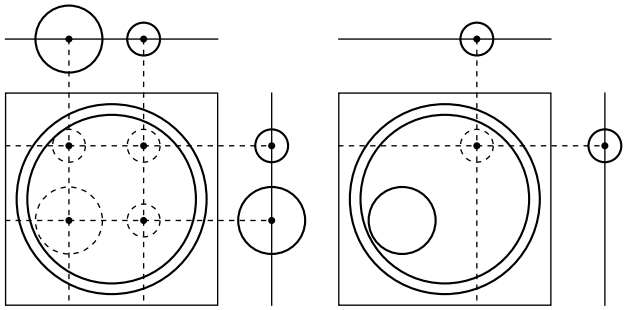


Figure 4.7. Back-projection of candidate discs detected in 1D projections. **(a)** Initial intersections of the cross-section shown in Figure 4.5(a) using a horizontal and a vertical projection. Intersections are formed by the crossings of the middle projection lines of respective candidate discs found in the aforementioned projections. Dashed circles represent the candidate discs centered about these intersections. **(b)** Deletion of intersections after choosing the candidate disc (O_i, r_i) with the largest radius (drawn with a solid line); (O_i, r_i) will be immediately added to the initial configuration $\mathbf{c}_0$. The dashed circle centered about the sole remaining intersection will be chosen in the next iteration, thus completing $\mathbf{c}_0$.

1. Possible projections of discs are located within the 1D projections of the cross-sections (see Figure 4.5(c)) by the application of **model fitting** [159, Chapter 15], i. e. disc parameters are estimated. In particular, a deterministic iterative algorithm called gradient descent [159, Section 10.6] [48] is used to find the parameters of the projection of a disc which best fits (a part of) the 1D reduced projections. After storing these parameters, the projection of the disc is eliminated from the already reduced projections (see Figure 4.5(d)). This process is repeated until no more discs can be detected (see Figure 4.5(e)).

2. The centers of the candidate discs found in the previous step are "projected back" into the plane of the 2D cross-section, thus forming several pairs of intersection points and radii of the discs associated with them (see Figure 4.7(a)). Those intersections whose distance from one another is below a given limit will be merged into a single intersection. Whatever the case, the radius associated with a particular intersection is calculated by taking the minimum of the radii of the corresponding discs found in the 1D projections.

3. The discs actually located in the 2D cross-section are selected by a greedy strategy: The center of a candidate disc is chosen to be the intersection which was defined by the most projections. Should there be more than one such intersections, the one with the largest radius associated with it will be taken. The intersection chosen is instantly removed from the list of intersections. We should remark here that some additional intersections may be deleted as well in order to retain consistency between the intersections and the 1D projections (see Figure 4.7(b)). If this disc results in a valid configuration, it will be permanently added to the 2D configuration of the cross-section. Otherwise, another intersection will be chosen. This procedure is repeated until no more intersections are left.

If, after the completion of the aforementioned algorithm, $\mathbf{c}_0$ is incomplete, the missing discs, spheres and cylinders will be generated using the approach presented in Section 4.4.[17]

[17] This is the only step that may make the construction process non-deterministic.

As a "side-effect" of the elimination of the tube and its interior, the algorithm computes estimated LAC values from $\mathbf{P}_\vartheta$: $\{\hat{\mu}_\mathrm{R}, \hat{\mu}_\mathrm{I}\}$ for 2D configurations, and $\{\hat{\mu}_\mathrm{T}, \hat{\mu}_\mathrm{I}\}$ for 3D ones. (The remaining estimates $\hat{\mu}_\mathrm{D}$ and $\hat{\mu}_\mathrm{SC}$ need to be determined manually, though.) Whereas these estimates are not actually used during reconstruction, they are nonetheless a great aid for physical measurements where LAC values may not be always known exactly.

The algorithm is governed by the following user-defined parameters: $\tau > 0$ real (expected noise level in $\mathbf{P}_\vartheta$), $0 < \lambda < 1$ real (factor of adjustment for radii and heights when encountering an invalid configuration), positive reals MERGE_O and MERGE_r (upper bounds for the center distance and radius difference when merging discs found in nearby 2D cross-sections into a single 3D geometrical primitive).

4.6 Summary

A stochastic DT reconstruction method has been presented that is able to reconstruct 2D images of objects which can be described as a composition of simple geometrical primitives (namely, discs and annuli). The approach was then extended to the analogous 3D setting, but allowing tubes, cylinders and spheres. It shall be stressed that the latter algorithm provides a native 3D reconstruction, i. e. the 3D result is not obtained by simply stacking the reconstructions of individual 2D cross-sections. Furthermore, the reconstruction process operates in the configuration space $\mathcal{C}$, that is finding solutions in terms of the parameters of constituting geometrical primitives.

In order to be able to test the efficiency of the reconstruction approach, an algorithm has been shown that can automatically generate random configurations. Moreover, it has also been demonstrated how one can devise initial configurations based on the input projections, which can then become the starting point of the reconstruction methods.

Chapter 5

Simulations: Reconstruction of Objects Parametrized with Geometrical Primitives

The efficacy of the techniques is demonstrated by numerous simulation experiments, in order to quantify how the reconstruction quality is influenced by the geometrical complexity of the configuration, by the number of projections, or by the amount of noise. Moreover, the benefits of using an automatically determined initial configuration (as opposed to a random one) are also justified.

The results presented herein have been published in [33, 104, 105, 121–123].

5.1 Implementation Details

5.1.1 Applications, Source Codes and Development Environment

The algorithms described in Chapter 4 have been implemented as a software package written in ANSI C, accompanied by a bunch of supporting Unix shell scripts. The following listing describes each application and its purpose:

pgm2dct Converts raw detector data (`.pgm` bitmap images) to the internal projection format (DIRECT, see below), supporting both 1D and 2D projections.[1]

projconv Supports the conversion of 1D projections among various formats: DIRECT, PRN (the format used by HMI), and SNARK `file11` [2]. (Note: 2D projections are currently supported in the DIRECT format only.)

projgen Allows the computation of simulated projections (1D and 2D), application of artificial noise to projections, and generation of random configurations (2D and 3D).

specrec Entry point to the reconstruction algorithms for obtaining a 2D or 3D configuration from a set of 1D or 2D projections, including the setting up of an initial configuration (obtained automatically from the projections or generated randomly). Allows the tuning

[1] The 1D version of this tool is due to Zoltán Kiss; it was later extended by the author of the thesis for the 2D scenario.

of all reconstruction parameters, visualization of configurations, as well as the calculation of quality metrics for results.

Both reconstruction methods support the projection format[2] of the DIscrete REConstruction Techniques framework (DIRECT) [1] for easier co-operation. (DIRECT is a framework for DT reconstruction approaches developed at the Department of Image Processing and Computer Graphics, University of Szeged.) Moreover, the 2D algorithm was also made available online as part of DIRECT.[3]

Each application mentioned above has only a command-line interface, expecting inputs and producing outputs via files.

Some source code statistics: 99 files, ≈ 1.09 MB, 32 224 total lines of code.

All the timings mentioned later were measured under GNU Linux (Fedora 4 distribution) on a PC having 1.25 GB RAM and a 2.8 GHz single-core 32-bit processor (Intel Pentium 4).

5.1.2 Speeding Up the Reconstruction Process

Stochastic methods are renowned not only for their ability to easily solve complicated problems, but also because they tend to be slow as compared to deterministic approaches. Therefore, considerable effort has been invested in optimizing the program code. The most important of these improvements is presented below.

Quick Evaluation of the Objective Function

Let $\mathbf{c}$ and $\mathbf{c}'$ denote the current and candidate configurations, respectively (see Section 4.2.3). As discussed before, the execution of SA needs one to compute the Boltzmann factor of the objective function $\gamma(f_{\mathbf{c}})$ (see Equations (2.11) and (4.10)):

$$\frac{\gamma(f_{\mathbf{c}'})}{\gamma(f_{\mathbf{c}})} = \mathrm{e}^{-\frac{H(f_{\mathbf{c}'})-H(f_{\mathbf{c}})}{k_{\mathrm{B}}T}} .$$

Considering the numerator of the exponent only, the task is to calculate (cf. Equation (4.9))

$$H(f_{\mathbf{c}'}) - H(f_{\mathbf{c}}) = \sum_{\vartheta} \left(\left\| \mathbf{P}_{f_{\mathbf{c}'},\vartheta} - \mathbf{P}_{\vartheta} \right\|_2^{\,2} - \left\| \mathbf{P}_{f_{\mathbf{c}},\vartheta} - \mathbf{P}_{\vartheta} \right\|_2^{\,2} \right) , \tag{5.1}$$

where $\mathbf{P}_{f_{\mathbf{c}},\vartheta}$ and $\mathbf{P}_{f_{\mathbf{c}'},\vartheta}$ can be expanded as shown later in Section 5.2.3 (see Equation (5.15)). When having 2D configurations, these expressions can be calculated directly since the amount of projection values involved is rather small.

The scenario looks more complicated for the 3D case, however. There the amount of projection pixels is increased manifold, namely with a factor of HEIGHT (see Equation (5.11)), therefore the direct computation of Equation (5.1) would be very costly. After expansion using $(a-b)^2 = a^2 - 2ab + b^2$, we get

$$H(f_{\mathbf{c}'}) - H(f_{\mathbf{c}}) = \sum_{\vartheta} \left(\mathbf{P}_{f_{\mathbf{c}'},\vartheta}{}^2 - 2\mathbf{P}_{f_{\mathbf{c}'},\vartheta}\mathbf{P}_{\vartheta} - \mathbf{P}_{f_{\mathbf{c}},\vartheta}{}^2 + 2\mathbf{P}_{f_{\mathbf{c}},\vartheta}\mathbf{P}_{\vartheta} \right) ,$$

where multiplication and exponentiation is meant element-wise and summed over all (s, z) coordinates. A further expansion and simplification[4] then leaves only the terms depending on the kind of alteration in $\mathbf{c}'$ vs $\mathbf{c}$, whose computation is significantly faster than the direct evaluation of Equation (5.1). In the current implementation, this new formulation alone had resulted in a 10–12-fold speed-up.

[2] Extensions specific to this thesis were introduced into the DIRECT format in cooperation with László Ruskó.
[3] This integration was entirely accomplished by Zoltán Kiss.
[4] This is not shown here for sake of brevity.

5.2 Simulation Setup

5.2.1 Visualization

Visualizing 2D Configurations

The digital image $f_{(O_i,r_i)}$ determined by disc (O_i, r_i) $(1 \leq i \leq N)$ is obtained as

$$f_{(O_i,r_i)}(x,y) \coloneqq \begin{cases} \mu_{\mathrm{D}} & \text{if } (x-x_i)^2 + (y-y_i)^2 \leq {r_i}^2\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.2}$$

Analogously, the image $f_{(O_{\mathrm{RI}},r_{\mathrm{RE}},r_{\mathrm{RI}})}$ of the ring $(O_{\mathrm{RI}}, r_{\mathrm{RE}}, r_{\mathrm{RI}})$ (with the assumption $O_{\mathrm{RE}} = O_{\mathrm{RI}}$), as well as the image $f_{(O_{\mathrm{RI}},r_{\mathrm{RI}})}$ of the interior $(O_{\mathrm{RI}}, r_{\mathrm{RI}})$, can be formulated as

$$\begin{aligned} f_{(O_{\mathrm{RI}},r_{\mathrm{RE}},r_{\mathrm{RI}})}(x,y) &\coloneqq \begin{cases} \mu_{\mathrm{R}} & \text{if } {r_{\mathrm{RI}}}^2 < (x-x_{\mathrm{RI}})^2 + (y-y_{\mathrm{RI}})^2 \leq {r_{\mathrm{RE}}}^2\,, \\ 0 & \text{otherwise}\,, \end{cases} \\ f_{(O_{\mathrm{RI}},r_{\mathrm{RI}})}(x,y) &\coloneqq \begin{cases} \mu_{\mathrm{I}} & \text{if } (x-x_{\mathrm{RI}})^2 + (y-y_{\mathrm{RI}})^2 \leq {r_{\mathrm{RI}}}^2 \\ & \text{and } f_{(O_i,r_i)}(x,y) = 0,\ \forall 1 \leq i \leq N\,, \\ 0 & \text{otherwise}\,. \end{cases} \end{aligned} \tag{5.3}$$

Furthermore, every pixel belonging to the background (air / vacuum surrounding the object) gets assigned μ_0 in the corresponding image f_0:

$$f_0(x,y) \coloneqq \begin{cases} \mu_0 & \text{if } {r_{\mathrm{RE}}}^2 < (x-x_{\mathrm{RI}})^2 + (y-y_{\mathrm{RI}})^2 \leq {r_{\max}}^2\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.4}$$

Finally, the digital image $f_{\mathbf{c}}$ determined by the 2D configuration $\mathbf{c}$ can be composed as[5]

$$\begin{aligned} f_{\mathbf{c}}(x,y) \coloneqq & f_0(x,y) + f_{(O_{\mathrm{RI}},r_{\mathrm{RE}},r_{\mathrm{RI}})}(x,y) + \\ & f_{(O_{\mathrm{RI}},r_{\mathrm{RI}})}(x,y) + \sum_{i=1}^{N} f_{(O_i,r_i)}(x,y)\,. \end{aligned} \tag{5.5}$$

Moreover, the rasterization of $f_{\mathbf{c}}$ is carried out using nearest neighbor sampling, without employing anti-aliasing or considering partial volume effects.

For sake of a consistent output irrespective of actual LAC values, the 2D configuration $\mathbf{c}$ gets usually rendered as a gray-scale image $f_{\mathbf{c}}^{\mathrm{gray}}$:

$$f_{\mathbf{c}}^{\mathrm{gray}}(x,y) \coloneqq \begin{cases} 0 \text{ (black)} & \text{if } f_{\mathbf{c}}(x,y) = \mu_{\mathrm{R}}\,, \\ \frac{1}{2} I_{\max} \text{ (dark-gray)} & \text{if } f_{\mathbf{c}}(x,y) = \mu_{\mathrm{I}}\,, \\ \frac{3}{4} I_{\max} \text{ (light-gray)} & \text{if } f_{\mathbf{c}}(x,y) = \mu_{\mathrm{D}}\,, \\ I_{\max} \text{ (white)} & \text{if } f_{\mathbf{c}}(x,y) \in \{0, \mu_0\}\,, \end{cases} \tag{5.6}$$

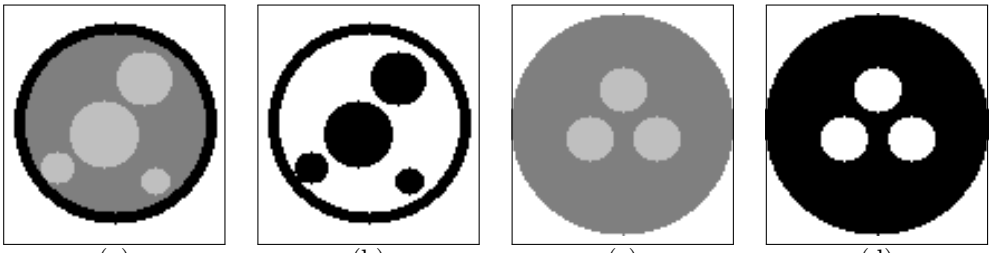
where $I_{\max}$ denotes the maximal gray-scale intensity.

For convenience, $f_{\mathbf{c}}$ can be also displayed as a binary[6] digital image $f_{\mathbf{c}}^{\mathrm{bin}}$:

$$f_{\mathbf{c}}^{\mathrm{bin}}(x,y) \coloneqq \begin{cases} 0 \text{ (white / background)} & \text{if } f_{\mathbf{c}}(x,y) \in \{0, \mu_0, \mu_{\mathrm{I}}\}\,, \\ 1 \text{ (black / foreground)} & \text{if } f_{\mathbf{c}}(x,y) \in \{\mu_{\mathrm{R}}, \mu_{\mathrm{D}}\}\,, \end{cases} \tag{5.7}$$

[5] Strictly speaking, $f_{\mathbf{c}}$ will actually be quinary (5-valued) when $\mu_0 \neq 0$, namely $\mathcal{A} = \{0, \mu_0, \mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}\}$, with all the pixels outside the reconstruction area being set to 0.

[6] Even though the usual convention is to map 0 to black and 1 to white, the setting shown here better suits the rendering of images against a white background (viz. the one used for this thesis).



(a) (b) (c) (d)

Figure 5.1. Visualization of 2D configurations. Gray levels are as per Equations (5.6), (5.7) and (5.8). (Frame added for better visibility.) **(a)** A general configuration with $r_{\mathrm{RE}} > r_{\mathrm{RI}}$ rendered as a gray-scale image. **(b)** (a) rendered as a binary image. **(c)** A configuration with $r_{\mathrm{RE}} = r_{\mathrm{RI}}$ rendered as a gray-scale image. **(d)** (c) rendered as a binary image.

when $r_{\mathrm{RE}} > r_{\mathrm{RI}}$ and

$$f_{\mathbf{c}}^{\mathrm{bin}}(x,y) := \begin{cases} 0 \text{ (white / background)} & \text{if } f_{\mathbf{c}}(x,y) \in \{0, \mu_0, \mu_{\mathrm{D}}\}\,, \\ 1 \text{ (black / foreground)} & \text{if } f_{\mathbf{c}}(x,y) = \mu_{\mathrm{I}}\,, \end{cases} \tag{5.8}$$

when $r_{\mathrm{RE}} = r_{\mathrm{RI}}$. (This distinction is necessary because essentially $\mu_{\mathrm{R}} = 0$ when the thickness of the annulus is zero.)

Let $\mathbf{c}_{\mathrm{o}}$ and $\mathbf{c}_{\mathrm{r}}$ be the original (i. e. ideal target) and reconstructed configurations, and $f_{\mathbf{c}_{\mathrm{o}}}^{\mathrm{bin}}$ and $f_{\mathbf{c}_{\mathrm{r}}}^{\mathrm{bin}}$ denote the binary digital images determined by these configurations. Then one can define the gray-scale difference image $f_{\mathbf{c}_{\mathrm{o}},\mathbf{c}_{\mathrm{r}}}^{\mathrm{gray}}$ as

$$f_{\mathbf{c}_{\mathrm{o}},\mathbf{c}_{\mathrm{r}}}^{\mathrm{gray}}(x,y) := \begin{cases} 0 \text{ (black)} & \text{if } f_{\mathbf{c}_{\mathrm{o}}}^{\mathrm{bin}}(x,y) - f_{\mathbf{c}_{\mathrm{r}}}^{\mathrm{bin}}(x,y) = 1\,, \\ \frac{1}{2} I_{\max} \text{ (dark-gray)} & \text{if } f_{\mathbf{c}_{\mathrm{o}}}^{\mathrm{bin}}(x,y) - f_{\mathbf{c}_{\mathrm{r}}}^{\mathrm{bin}}(x,y) = -1\,, \\ I_{\max} \text{ (white)} & \text{if } f_{\mathbf{c}_{\mathrm{o}}}^{\mathrm{bin}}(x,y) - f_{\mathbf{c}_{\mathrm{r}}}^{\mathrm{bin}}(x,y) = 0\,, \end{cases} \tag{5.9}$$

so that common pixels appear white, foreground pixels found only in $f_{\mathbf{c}_{\mathrm{o}}}^{\mathrm{bin}}$ are painted black, and those found only in $f_{\mathbf{c}_{\mathrm{r}}}^{\mathrm{bin}}$ are drawn in dark-gray.

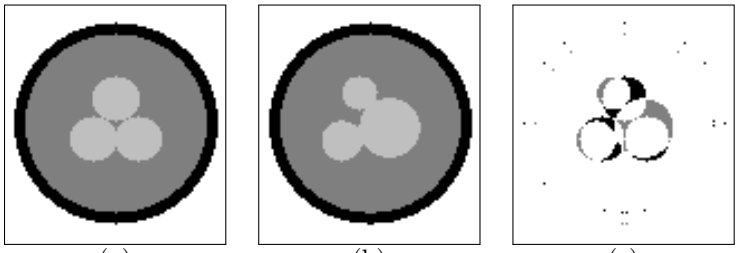
In all the aforementioned cases, the origin of the coordinate system is placed in the lower-left corner of the image, so that the positive x-axis and y-axis run along the lower and left side of the image, respectively.

Figures 5.1 and 5.2 give a few examples for the visualization options described above.

Visualizing 3D Configurations

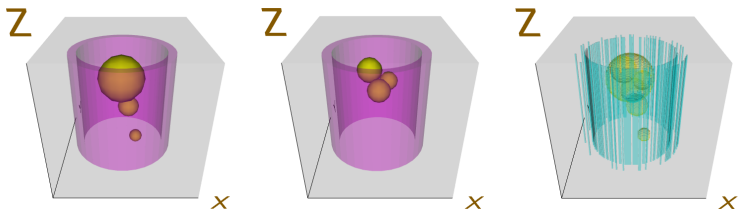
3D configurations can be presented in two ways: First, they can be rendered as a series of gray-scale or binary images corresponding to individual 2D cross-sections; these images are produced exactly like described above for 2D configurations. In addition, it is possible to turn any 3D configuration into a Virtual Reality Modeling Language (VRML) [4] model (to be viewed with internet browser plugins or external utilities), thus providing a means for native 3D visualization and interactive exploration. (There are, of course, other alternatives for rendering 3D discrete volumes, e. g. iso-surface extraction and volume rendering, but the simplicity and interactivity of VRML is sufficient for our needs.)

In analogy with the 2D case, one can construct the difference volume of the original (i. e. ideal target) and reconstructed configurations. The voxels of this 3D volume can then be also



(a) (b) (c)

Figure 5.2. Visualizing the difference of 2D configurations. Gray levels are as per Equations (5.6) and (5.9). (Frame added for better visibility.) **(a)** Original (i. e. ideal target) configuration rendered as a gray-scale image. **(b)** Unsuccessful reconstructed configuration rendered as a gray-scale image. **(c)** The difference of (a) and (b).



(a) (b) (c)

Figure 5.3. Visualization of 3D configurations using VRML models (oblique view). **(a)** Original (i. e. ideal target) configuration. **(b)** Unsuccessful reconstructed configuration. **(c)** The difference of (a) and (b).

visualized as a VRML model, so that each voxel gets drawn as a transparent cube. Specifically, common voxels are hidden, voxels found only in the original configuration are drawn in yellow, while those found only in the reconstructed configuration are drawn in cyan. This is demonstrated in Figure 5.3.

5.2.2 Software Phantoms

The behavior of the reconstruction algorithms have been inspected using several test images (commonly called **phantoms**); a few examples are shown in Figures 5.4 and 5.5. While some of them were constructed manually, most of the experiments employed randomly generated configurations produced with the approach described in Section 4.4.

5.2.3 Computation of Simulated Projections

Acquisition Set-up

The acquisition set-up imitated the conditions found in physical experiments (see Section 3.1.2). Moreover, computations were carried out in a somewhat idealized fashion. In particular, the following assumptions had been taken:

- Instrument geometry: Measurements are based on attenuation contrast transmission tomography using parallel geometry. The radiation beam is monochromatic with negligible

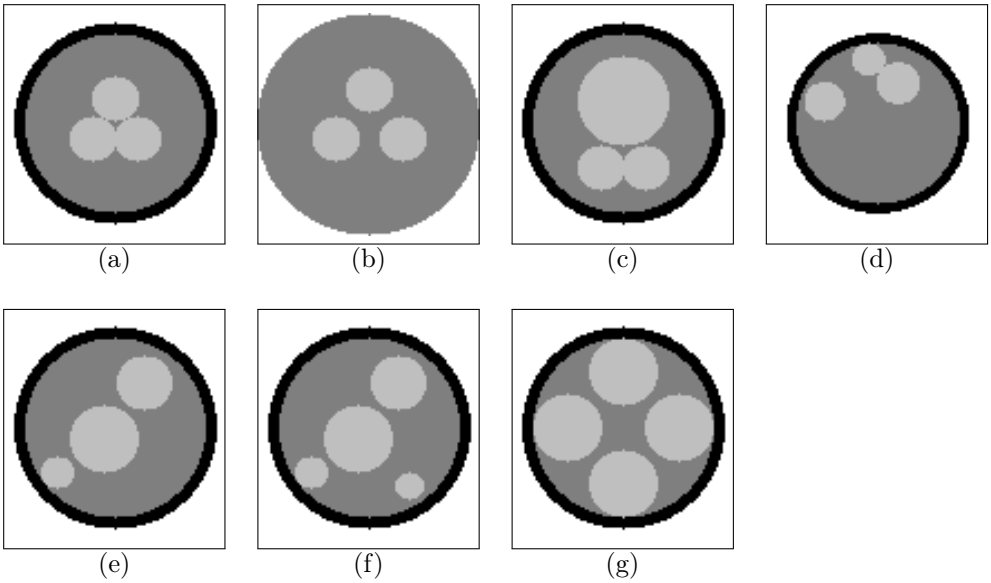


Figure 5.4. A few 2D software phantoms used for later reconstructions. Gray levels are as per Equation (5.6). Phantoms (d) and (f) were generated automatically, the rest were constructed manually. (Frame added for better visibility.)

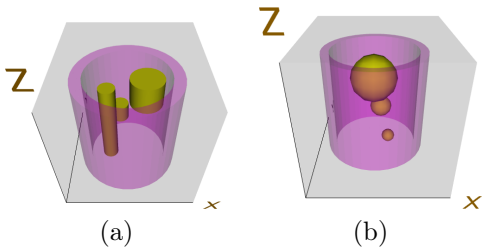


Figure 5.5. A few 3D software phantoms used for later reconstructions (oblique VRML view). Both configurations were generated automatically.

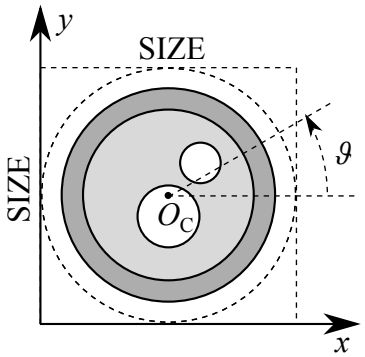


Figure 5.6. Projection geometry for the 2D object to be reconstructed. Features depicted: coordinate system, image domain $\mathcal{D}$ (dashed square), reconstruction area (dashed circle), center of rotation O_C, projection angle ϑ.

divergence and energy spread.[7] There is no detector tilting or sample precession during rotation. Projections are taken along equidistant projection lines at sample rotations $\{\vartheta_1, \ldots, \vartheta_n\}$ $(n \geq 2)$.[8] Projection directions are defined as shown in Figures 5.6 and 5.7, so that the object is rotated counterclockwise about O_C (in 2D) or about the axis of rotation passing through the reconstruction area and incident with O_C (in 3D).

- Sample geometry: The sample—as well as the image f containing it—can be described with square pixels / cubic voxels. The detector spacing matches the sample pixel spacing, so that each detector pixel is determined by a single projection line hitting the pixel in its center. Thus the projection domain becomes

$$\mathcal{D}_{\mathbf{P}} \coloneqq \{0, \ldots, \text{SIZE} - 1\} \tag{5.10}$$

for 2D configurations and

$$\mathcal{D}_{\mathbf{P}} \coloneqq \{0, \ldots, \text{SIZE} - 1\} \times \{0, \ldots, \text{HEIGHT} - 1\} \tag{5.11}$$

for 3D objects.

- Imaging system: A sample pixel always contributes the same amount of absorption, no matter how large fraction of the projection line passes through it. (That is, partial volume effects are ignored.) The projection error term $\|\mathbf{P}_{f_\mathbf{c},\vartheta} - \mathbf{P}_\vartheta\|_2^2$ is computed, for any ϑ, by comparing corresponding pixels in $\mathbf{P}_{f_\mathbf{c},\vartheta}$ and $\mathbf{P}_\vartheta$.

Projection of a 2D Configuration

Let ϑ, (x, y) and $[\mathfrak{R}f]\,(s, \vartheta)$ be as defined earlier in Section 2.1.0.1. Since $f_\mathbf{c}$ is entirely determined by configuration $\mathbf{c}$, the simulated discrete projection $\mathbf{P}_{f_\mathbf{c},\vartheta}$ of $f_\mathbf{c}$ at direction ϑ can be also defined

[7] In other words, we assume that projections do not need intensity or beam hardening corrections. The actual spectral distribution is otherwise irrelevant to us.

[8] For simplicity, all simulations were carried out using equiangular (i. e. equally spaced) projection directions, even though the algorithm allows arbitrary ϑ_i values. On the other hand, it has been shown that such a setting may not be always optimal: projections taken at different directions have varying amount of information that could help avoid ambiguities during reconstruction [187].

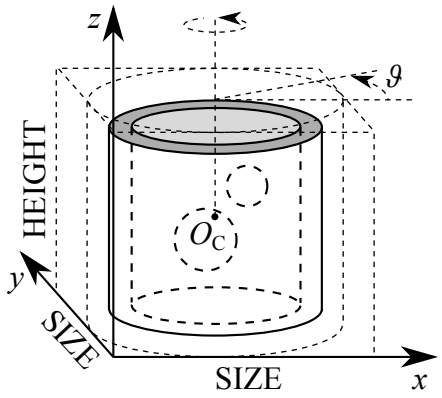


Figure 5.7. Projection geometry for the 3D object to be reconstructed. Features depicted: coordinate system, image domain $\mathcal{D}$ (dashed block), reconstruction area (enclosing dashed cylinder), axis of rotation passing through O_C, projection angle ϑ.

analytically—i. e. as if a continuous Radon transform—in terms of $\mathbf{c}$. This is achieved in multiple steps, as shown below. In order to avoid the need of computing the intersections of $\mathbf{c}$ with projection lines having arbitrary orientations, $\mathbf{c}$ will be rotated about O_C (see Equation (4.8)) by an angle $-\vartheta$, essentially implying $(s,u) \equiv (x,y)$. Therefore, it is enough to calculate the vertical projection of the rotated configuration as follows.

The projection $\mathbf{P}_{(O_i,r_i),\vartheta}$ of disc (O_i, r_i) $(1 \le i \le N)$ at direction ϑ is obtained as

$$\mathbf{P}_{(O_i,r_i),\vartheta}(s) \coloneqq \begin{cases} 2\,(\mu_D - \mu_I)\,\sqrt{{r_i}^2 - (s - x_{i,\vartheta})^2} & \text{if } |s - x_{i,\vartheta}| \le r_i\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{5.12}$$

where

$$x_{i,\vartheta} \coloneqq (x_i - x_C)\cdot\cos\vartheta + (y_i - y_C)\cdot\sin\vartheta + x_C\,.$$

Similarly, the projections $\mathbf{P}_{(O_{RI},r_{RE}),\vartheta}$ and $\mathbf{P}_{(O_{RI},r_{RI}),\vartheta}$ of (O_{RI}, r_{RE}) (external disc of the ring, with the assumption $O_{RE} = O_{RI}$) and of the interior (O_{RI}, r_{RI}) at direction ϑ, respectively, can be expressed as

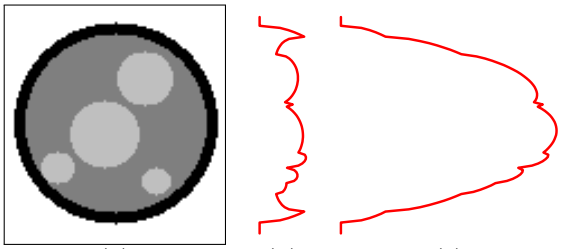
$$\begin{aligned} \mathbf{P}_{(O_{RI},r_{RE}),\vartheta}(s) &\coloneqq \begin{cases} 2\,(\mu_R - \mu_0)\,\sqrt{{r_{RE}}^2 - (s - x_{RI,\vartheta})^2} & \text{if } |s - x_{RI,\vartheta}| \le r_{RE}\,, \\ 0 & \text{otherwise}\,, \end{cases} \\ \mathbf{P}_{(O_{RI},r_{RI}),\vartheta}(s) &\coloneqq \begin{cases} 2\,(\mu_I - \mu_R)\,\sqrt{{r_{RI}}^2 - (s - x_{RI,\vartheta})^2} & \text{if } |s - x_{RI,\vartheta}| \le r_{RI}\,, \\ 0 & \text{otherwise}\,, \end{cases} \end{aligned} \tag{5.13}$$

where

$$x_{RI,\vartheta} \coloneqq (x_{RI} - x_C)\cdot\cos\vartheta + (y_{RI} - y_C)\cdot\sin\vartheta + x_C\,.$$

Moreover, the projection $\mathbf{P}_0$ of the background (air / vacuum surrounding the object)—irrespective of direction ϑ—is given by

$$\mathbf{P}_0(s) \coloneqq \begin{cases} 2\mu_0\sqrt{{r_{max}}^2 - (s - x_C)^2} & \text{if } |s - x_C| \le r_{max}\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.14}$$



(a) (b) (c)

Figure 5.8. Effects of LAC values on the projection. **(a)** 2D configuration to be reconstructed. **(b)** Noiseless horizontal projection of (a) with $(\mu_{\mathrm{R}} = 1, \mu_{\mathrm{I}} = 0, \mu_{\mathrm{D}} = 1)$. **(c)** Noiseless horizontal projection of (a) with $(\mu_{\mathrm{R}} = 1, \mu_{\mathrm{I}} = 2, \mu_{\mathrm{D}} = 3)$.

Thus $\mathbf{P}_{f_{\mathbf{c}},\vartheta}$ can be expanded as

$$\begin{aligned}\mathbf{P}_{f_{\mathbf{c}},\vartheta}(s) \coloneqq [\mathfrak{R} f_{\mathbf{c}}](s,\vartheta) \coloneqq &\mathbf{P}_0(s) + \mathbf{P}_{(O_{\mathrm{RI}},r_{\mathrm{RE}}),\vartheta}(s) + \\ &\mathbf{P}_{(O_{\mathrm{RI}},r_{\mathrm{RI}}),\vartheta}(s) + \sum_{i=1}^{N} \mathbf{P}_{(O_i,r_i),\vartheta}(s)\,.\end{aligned} \tag{5.15}$$

As can be expected, projections of a given configuration computed with different sets of LAC values may appear very different from each other; this is demonstrated in Figure 5.8.

Projection of a 3D Configuration

The computation of projections for 3D configurations is carried out in analogy with their 2D companions. The only differences are that the configuration gets rotated about the line passing through O_{C} (see Equation (4.12)) and parallel to the z-axis, and $\mathbf{P}_{f_{\mathbf{c}},\vartheta}$ is composed of the projections of individual geometrical primitives (namely, the tube, the cylinder of the interior, and the spheres or cylinders enclosed by the interior).

Modeling Noise

In order to come close to the quality of real transmission radiographs, varying levels of artificial **noise** were applied to the simulated projections. The distortions of measurements are caused by different kinds of phenomena: scattering, fluctuation of the intensity of the particle beam, statistical error (the so-called Poisson or quantum noise), cross-talk between neighboring detector pixels, etc. As an approximation to the cumulative effects of these distortions, the following simple additive noise model was employed: Let us suppose that the value $v \coloneqq \mathbf{P}_{f_{\mathbf{c}},\vartheta}(s)$ of a 1D projection is to be distorted by $L\,\%$ of noise ($L \geq 0$ real), and let M denote

$$M \coloneqq \max_{s,\vartheta} \mathbf{P}_{f_{\mathbf{c}},\vartheta}(s)\,,$$

that is the global maximal projection value among all directions ϑ. The noisy projection value v_{noisy} was then defined as a uniformly distributed random number taken from $\left[v - M\frac{L}{100}, v + M\frac{L}{100}\right]$, subject to the constraint of non-negativity. It is evident that v_{noisy} can be positive even if $v = 0$ (i. e. the noise also affects background pixels).

For 2D projections (of 3D configurations) noise is applied by the direct extension of the above method. The only difference is that M will then be computed over 2D projections.

5.2.4 Initial Configurations

As presented in Section 4.5, the reconstruction process is started from an automatically determined initial configuration $\mathbf{c}_0$. In order to assess the effect of $\mathbf{c}_0$ on reconstruction quality, however, it is also possible to entirely skip the procedure mentioned before and start from a random $\mathbf{c}_0$. Such random initial configurations are also generated using the approach presented in Section 4.4. As will be shown in Section 5.3, it is indeed justified to use the automatically determined $\mathbf{c}_0$ for it brings a lot of improvement in terms of reconstruction success rate.

5.2.5 Measuring the Quality of Reconstructions

The quality of the results were measured by four **figure-of-merit** (FOM) functions, $\mathrm{FOM}_{\mathrm{RME}}$, $\mathrm{FOM}_{\mathrm{SE}}$, $\mathrm{FOM}_{\mathrm{VE}}$ and $\mathrm{FOM}_{\mathrm{CD}}$, all given in percents. (The first 3 were adapted from [117,158].)

Let $\mathbf{c}_{\mathrm{o}}$ and $\mathbf{c}_{\mathrm{r}}$ be the original (i. e. ideal target) and reconstructed configurations, and $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}$ and $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}$ denote the binary digital images determined by these configurations (as per Section 5.2.1). Then, the "relative mean error" is computed as

$$\mathrm{FOM}_{\mathrm{RME}} \coloneqq \frac{\sum_{x,y} \left| f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}(x,y) - f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}(x,y) \right|}{\sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}(x,y)} \cdot 100\,\%\,, \tag{5.16}$$

expressing the ratio of the total pixel errors vs the number of foreground pixels in $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}$. Although $\mathrm{FOM}_{\mathrm{RME}}$ is unbounded by default due to the accidental division by zero (when $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}$ only contains background pixels), this can be alleviated by using $\max\{1, d\}$ as the new denominator, where d denotes the original denominator in Equation (5.16). With this correction, $\mathrm{FOM}_{\mathrm{RME}} \leq |\mathcal{D}| \cdot 100\,\%$ holds. (The maximum is reached when $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}$ contains a sole foreground pixel and $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}$ is the inverted image of $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}$.)

The "shape error" is given by

$$\mathrm{FOM}_{\mathrm{SE}} \coloneqq \frac{2 \cdot \sum_{x,y} \left| f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}(x,y) - f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}(x,y) \right|}{\sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}(x,y) + \sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}(x,y)} \cdot 100\,\%\,, \tag{5.17}$$

that is the ratio of the total pixel errors vs the total number of foreground pixels in both images. (The factor 2 in the numerator is present to scale its range to match that of the denominator.) It can be checked that $\mathrm{FOM}_{\mathrm{SE}} \leq 200\,\%$, reaching the maximum when $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}$ is the inverted image of $f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}$.

The "volume / area error" is obtained as

$$\mathrm{FOM}_{\mathrm{VE}} \coloneqq \frac{2 \left| \sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}(x,y) - \sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}(x,y) \right|}{\sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{o}}}(x,y) + \sum_{x,y} f^{\mathrm{bin}}_{\mathbf{c}_{\mathrm{r}}}(x,y)} \cdot 100\,\%\,, \tag{5.18}$$

providing the ratio of the difference of the number of foreground pixels vs the total number of foreground pixels in both images. (Again, the factor 2 is necessary in order to get identical ranges for the numerator and denominator.) It can be verified that $\mathrm{FOM}_{\mathrm{VE}} \leq 200\,\%$, reaching the maximum when either image only contains background pixels and the other one contains at least 1 foreground pixels.

As can be seen, all the aforementioned FOM functions are based on the digital images rendered from the configurations. On the other hand, the fourth FOM, called the "configuration distance[9]", is quite different: it expresses the normalized vector distance of configurations $\mathbf{c}_{\mathrm{o}}$ and

[9] This FOM has not been published; it was devised as an extension and improvement upon the results published earlier.

$\mathbf{c}_\mathrm{r}$. Specifically, it is defined as

$$\mathrm{FOM}_\mathrm{CD} \coloneqq \frac{1}{d_\mathrm{max}} \min_{\mathbf{c}'_\mathrm{r}} \|\mathbf{c}_\mathrm{o} - \mathbf{c}'_\mathrm{r}\|_2 \cdot 100\,\%\,, \tag{5.19}$$

where d_max is an approximate upper bound[10] for the vector distance, while $\mathbf{c}'_\mathrm{r}$ is obtained from $\mathbf{c}_\mathrm{r}$ by permuting its parameters (O_i, r_i) $(1 \leq i \leq N)$ of the discs (and analogously for spheres and cylinders in the 3D case). The minimum operator runs over all such $\mathbf{c}'_\mathrm{r}$, i. e. finding the permutation that results in the smallest vector distance. By definition, $\mathrm{FOM}_\mathrm{CD} \leq 100\,\%$, but the maximum can never be reached because the d_max used here is not a tight bound.

As can be expected, each of these measures have different advantages and shortcomings:

$\mathrm{FOM}_\mathrm{RME}$ Its value is too biased towards $f^\mathrm{bin}_{\mathbf{c}_\mathrm{o}}$; it gets exaggerated especially when the number of foreground pixels is much lower in $f^\mathrm{bin}_{\mathbf{c}_\mathrm{o}}$ than in $f^\mathrm{bin}_{\mathbf{c}_\mathrm{r}}$.

FOM_SE This measure performs quite similarly to $\mathrm{FOM}_\mathrm{RME}$; its main advantage is the smaller upper bound and the absence of bias.

FOM_VE Unfortunately it says nothing about how close $\mathbf{c}_\mathrm{o}$ and $\mathbf{c}_\mathrm{r}$ actually are (e. g. it can have a pretty low value in case of a switching component, like the scenario shown later in Figure 5.9). Moreover, it is sometimes "inconsistent" with the previous two measures in the sense that if $\mathrm{FOM}_\mathrm{RME}$ (or FOM_SE) is better for one particular reconstruction than for some other, FOM_VE can still be worse for the first reconstruction than for the second one.

FOM_CD Its main advantage is that it is not prone to the pixel errors that are quite common due to the simple rendering approach (namely the nearest neighbor sampling and the lack of anti-aliasing; see Section 5.2.1). On the other hand, the number of $\mathbf{c}'_\mathrm{r}$ to try equals $N!$, so this measure is not practical above $N = 10$ (where $10! = 3\,628\,800$).

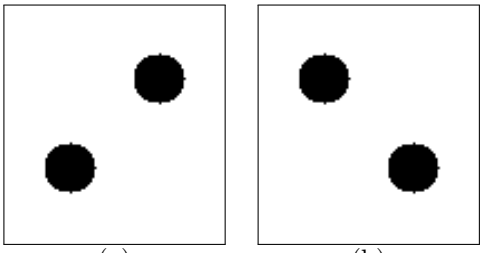
In all the cases, smaller values correspond to better reconstruction qualities, so that a perfect reconstruction has an associated 0 % error using either FOM. According to our experience, a reconstruction is successful (i. e. "of good quality") when $\mathrm{FOM}_\mathrm{RME} < 10\,\%$, $\mathrm{FOM}_\mathrm{SE} < 10\,\%$, $\mathrm{FOM}_\mathrm{VE} < 3\,\%$, and $\mathrm{FOM}_\mathrm{CD} < 1.5\,\%$. Furthermore, $\mathrm{FOM}_\mathrm{RME} \approx \mathrm{FOM}_\mathrm{SE}$ (except for pathological cases), $0.05 \leq \frac{\mathrm{FOM}_\mathrm{VE}}{\mathrm{FOM}_\mathrm{RME}} \approx \frac{\mathrm{FOM}_\mathrm{VE}}{\mathrm{FOM}_\mathrm{SE}} \leq 0.15$, and $0.15 \leq \frac{\mathrm{FOM}_\mathrm{CD}}{\mathrm{FOM}_\mathrm{RME}} \approx \frac{\mathrm{FOM}_\mathrm{CD}}{\mathrm{FOM}_\mathrm{SE}} \leq 0.4$ most of the time.

5.2.6 Reconstruction Parameter Settings

The configuration geometry was set up so as to mimic the circumstances met in physical experiments. All phantoms were, therefore, constructed with varying number of discs / spheres / cylinders whose radii and heights exceeded at least 5 pixels. As discussed earlier, $O_\mathrm{RE} = O_\mathrm{RI}$ was assumed in order to force the annulus / tube to be concentric. The image domain had the dimensions of 100×100 or $100 \times 100 \times 100$ pixels.

μ_0 was fixed at zero, while the remaining 3 LAC parameters $(\mu_\mathrm{R}, \mu_\mathrm{I}, \mu_\mathrm{D})$—and similarly $(\mu_\mathrm{T}, \mu_\mathrm{I}, \mu_\mathrm{SC})$—were assigned either of 2 sets of values: $(1, 0, 1)$ or $(1, 2, 3)$. Again, to come close to the limitations of real applications, only 2 (at $\vartheta = 0°$ and 90°) or 4 projections (at $\vartheta = 0°$, 45°, 90° and 135°) were taken most of the time, usually distorted by 10 % of artificial noise.

[10] In 2D, $\left(\overline{O_1 O_2}\right)^2 \leq (2r_\mathrm{max})^2$ and $(r_1 - r_2)^2 \leq {r_\mathrm{max}}^2$ for any circle centers O_1 and O_2 and radii r_1 and r_2. Since the annulus requires one center and two radii and any disc needs one center and one radius, $d_\mathrm{max} \coloneqq r_\mathrm{max}\sqrt{(4+2)+(4+1)\cdot N}$ can be taken. The 3D case is slightly more complicated: $d_\mathrm{max} \coloneqq \sqrt{((4+2)+4N)\cdot {r_\mathrm{max}}^2 + (2+N)\cdot {h_\mathrm{max}}^2 + N{r_\mathrm{maxS}}^2}$ when having spheres, and $d_\mathrm{max} \coloneqq \sqrt{((4+2)+(4+1)\cdot N)\cdot {r_\mathrm{max}}^2 + (2+2N)\cdot {h_\mathrm{max}}^2}$ when having cylinders.



(a) (b)

Figure 5.9. A pair of binary images having identical horizontal and vertical projections. Pixel intensities correspond to 0 / background (white) and 1 / foreground (black). (Frame added for better visibility.)

The optimization was typically started from an automatically determined $\mathbf{c}_0$. For the 3D case, the merging of discs on neighboring or nearby cross-sections was also allowed during the construction of $\mathbf{c}_0$. All LAC values were held fixed throughout the optimization. τ was set 5% higher than the noise level.

The temperature was decreased at every 20 attempted iterations using a simple linear annealing schedule (see case 3 of Equation (2.8)). Even though there are several more sophisticated schedules available (e. g. [71,164] or [103,149,150]), we concluded that the aforementioned simple strategy suffices when having a good enough $\mathbf{c}_0$.

The optimization procedure was stopped when the Hamiltonian $H(f_{\mathbf{c}})$ has fallen below a preset threshold (currently: 10), or the number of configuration candidates $\mathbf{c}'$ rejected subsequently (i. e. in a row, without any accepted configurations in between) has exceeded some upper bound (5000 in the present implementation). It would also be possible to terminate the process after a certain amount of iterations, but this was never realized.

See Table 5.1 for a summary of settings in place.

5.3 Results

Like in other applications of DT, there are situations when multiple global optima exist. One such case is depicted in Figure 5.9, showing a pair of binary images of common dimensions with two discs in each. Both discs have the same radius, and their centers lie on opposing corners of a square (a rectangle in general). The location, size and orientation of this "imaginary" square is the same in both images. It is easy to see that these images have identical horizontal projections, and the same is true for the vertical projections. (Essentially, these images behave as generalized switching components, with discs playing the role of standalone pixels.) Therefore, when using only 2 projections taken at the aforementioned directions[11], exactly two solutions exist, and both are "correct" from a mathematical point of view. Besides the effects of artificial noise added to projections, the presence of such ambiguities also pose some complications to the reconstruction process (albeit the ratio of such pathological cases over the rest is rather small).

The rest of the section is devoted to presenting the simulation results obtained for 2D as well as 3D configurations.

[11] It should be noted that the reconstruction becomes a unique task—having a single solution—when using projections in some other direction (e. g. along image diagonals) that contain more information about the relation of discs with respect to each other [187].

Table 5.1. Parameter settings for simulation studies using $\gamma(f_{\mathbf{c}})$.

Parameter	Setting
Configuration geometry	
SIZE	100
HEIGHT	100
O_{RE}	O_{RI}
$r_{\min}$	7 when generating phantoms, 5 otherwise
$h_{\min}$	5
N	varies for each phantom
δ	10^{-3}
Projections	
n	4 by default; sometimes 2 or 8
μ_0	0
$(\mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}})$, $(\mu_{\mathrm{T}}, \mu_{\mathrm{I}}, \mu_{\mathrm{SC}})$	$(1, 0, 1)$ by default; sometimes $(1, 2, 3)$
noise level	10 % by default; sometimes 0 %, 20 % or 40 %
Optimization	
$\mathbf{c}_0$	automatic by default; sometimes random
λ	0.95
λ_1	0.5
τ	5 % higher than the expected noise level
MERGE_O	15
MERGE_r	10
ε_1	10^{-2}
k_{B}	1
annealing schedule	linear ($T_0 = 100$, $\Delta T = 0.1$ @ 20 iters, $T_{\min} = 1$)
Miscellaneous	
ε	10^{-6}

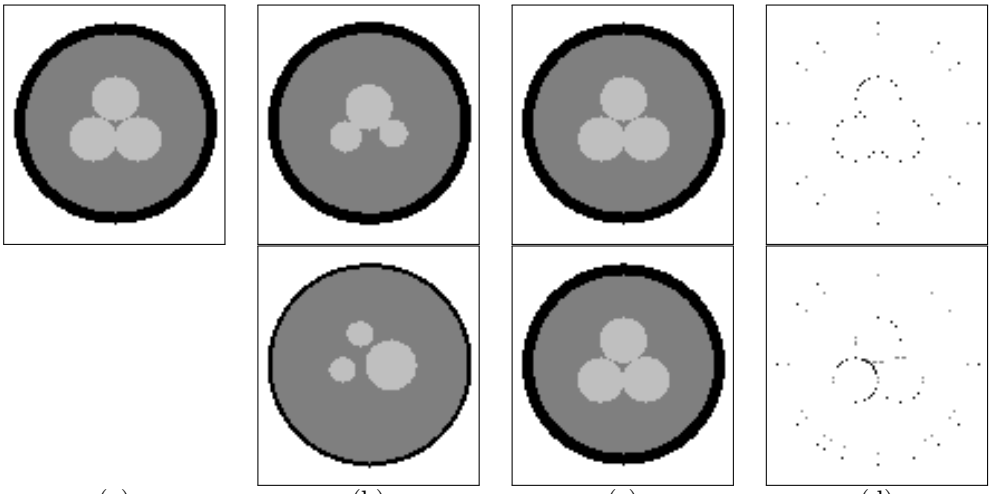


Figure 5.10. Reconstruction of a 2D software phantom from 4 projections at 0 % and 10 % noise levels. (Frame added for better visibility.) Top row: noiseless projections; $\mathrm{FOM}_{\mathrm{RME}} = 2.318\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.059\,\%$. Bottom row: 10 % noise level; $\mathrm{FOM}_{\mathrm{RME}} = 3.281\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.237\,\%$. **(a)** Original configuration. **(b)** Initial configuration. **(c)** Reconstructed configuration. **(d)** Difference of (a) and (c).

5.3.1 Reconstructions of 2D Objects

The reconstruction algorithm was first tested with a couple of 2D software phantoms, some constructed manually but most of them were generated randomly. All the reconstructions of these phantoms were performed using 4 projections and the LAC settings $(\mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}) = (1, 0, 1)$. (See the end of this subsection regarding the case when other LAC values are taken.) As can be seen in Figure 5.10, though the quality of the reconstructed configuration degrades (very) slightly in the presence of noise, the result can still be considered an excellent quality reconstruction with $\mathrm{FOM}_{\mathrm{CD}} = 0.237\,\%$. Further examples are shown in Appendix F, namely in Figures F.1–F.6. It should be noted that, in most cases, the "precision" of the initial configuration constructed from the input projections is remarkably good, partly thanks to the 4 projection directions.

The purpose of the next series of test runs was to give a quantitative measure of the algorithm's sensitivity regarding two factors: the geometrical complexity of the phantom, and the amount of noise added to the simulated input projections. For each scenario, a collection of 10 random software phantoms were generated and reconstructed independently, then the resulting FOM values got collected and aggregated into plots for easier visualization. Unless noted otherwise, all tests assumed the following defaults: $(\mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}) = (1, 0, 1)$, 2 projections, 3 discs, 10 % noise level, and automatic initial configurations.

As can be seen in Figure 5.11, the approach is able to reconstruct up to 3 discs with an acceptable quality when using 2 projections, and up to 5 discs when having 4 projections. It is also evident that, irrespective of the number of discs, worse quality measures are obtained when the reconstruction process is started from a random initial configuration.

As indicated in Figure 5.12, good quality reconstructions can be expected even in the presence of substantial amount of noise ($\approx 25\,\%$ or less), but that requires 4 or more projections. When having only 2 projections, the upper bound for a "safe" noise level is estimated somewhere around 5–10 %. To get a feeling of what a noise level of 20 % or 40 % does to simulated projections, see

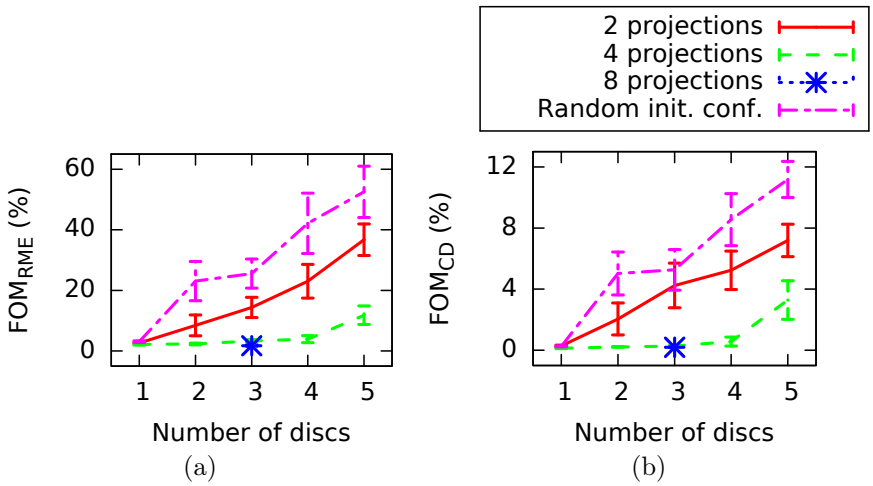


Figure 5.11. Quality of the reconstructions for 2D configurations at 10 % noise level as the function of geometrical complexity, based on 10 repetitions. Error bars indicate the standard error. Smaller values correspond to better results. **(a)** $\mathrm{FOM}_{\mathrm{RME}}$ versus the number of discs. **(b)** $\mathrm{FOM}_{\mathrm{CD}}$ versus the number of discs.

Figure 5.13.

The speed of convergence of the objective function $\gamma(f_{\mathbf{c}})$ is depicted in Figure 5.14 for a typical reconstruction of the test object given in Figure F.2 using noiseless projections. As can be seen, only 170 out of 37 428 SA iterations ($\approx 0.45\,\%$) resulted in the acceptance of a new candidate configuration $\mathbf{c}'$; the rest of the proposals were rejected. During this time, the exponent of $\gamma(f_{\mathbf{c}})$ rose from the initial $-2\,960$ up to ≈ -8.4, spending altogether about 4 seconds. It is evident that the objective function kept improving steadily for the first ≈ 40 accepted $\mathbf{c}'$, but slowed down dramatically thereafter. As can be observed, the objective function can get quite close to 0 in case of perfect LAC values and noiseless projections.

Some of the aforementioned reconstructions were repeated with a different set of LAC values, namely $(\mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}) = (1, 2, 3)$. Besides influencing the value of $\gamma(f_{\mathbf{c}})$, the reconstruction quality is otherwise unaffected by such changes, implying that the algorithm is robust in this sense.

5.3.2 Reconstructions of 3D Objects

In analogy with the 2D studies presented in Section 5.3.1, the quality of reconstructions was verified via numerous executions using 3D configurations. Two examples of such objects are given in Figures F.7 and F.8, both showing nearly perfect reconstructions. As for the 2D scenario, the automatically determined initial configurations are already pretty close to the ideal ones.

Similarly to the 2D case, we would like to determine how much the reconstruction quality is affected by the geometrical complexity of configurations as well as the noise level in simulated projections. Contrary to 2D executions where the geometrical complexity is solely related to the number of discs within the interior, 3D configurations can comprise spheres as well as cylinders. It may then come as a surprise that the algorithm behaves quite differently depending on which kind of geometric primitive is being used. Specifically, good quality reconstructions can be obtained even when having 6 spheres but only 2 projections (see Figure F.9)! On the other hand, switching to cylinders makes the graphs resemble more to their 2D counterparts, as is apparent in Figure F.10 (cf. Figure 5.11). It is also clear that using automatically constructed

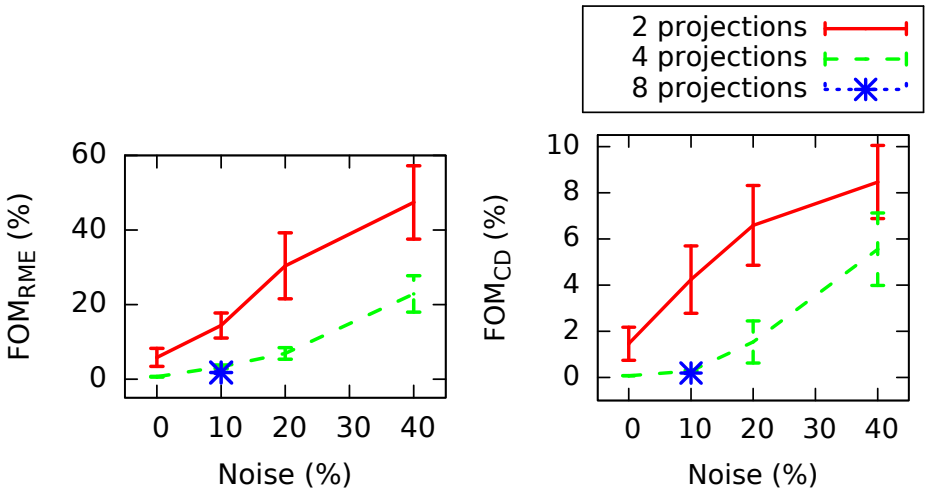


Figure 5.12. Quality of the reconstructions for 2D configurations containing 3 discs as the function of noise level, based on 10 repetitions. Error bars indicate the standard error. Smaller values correspond to better results. **(a)** $\mathrm{FOM}_{\mathrm{RME}}$ versus the level of noise. **(b)** $\mathrm{FOM}_{\mathrm{CD}}$ versus the level of noise.

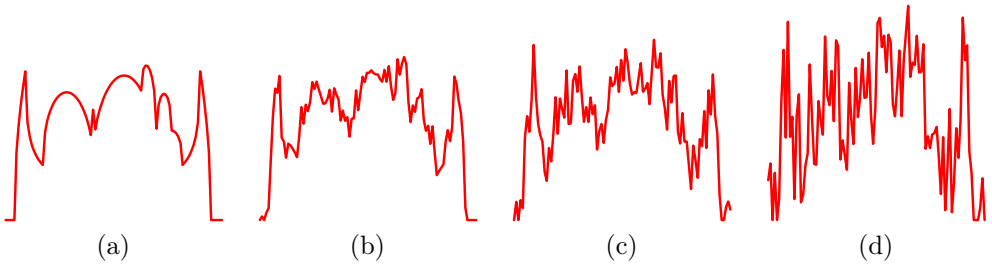
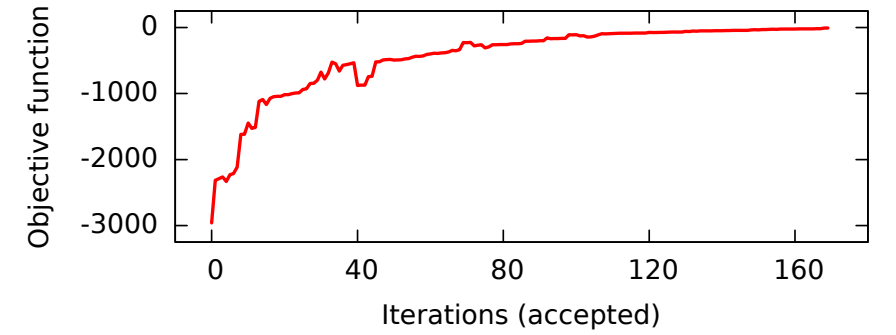
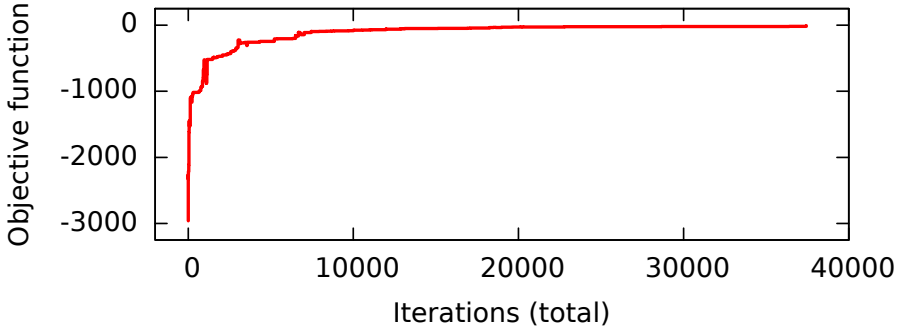


Figure 5.13. Effects of the noise level on simulated 1D projections. **(a)** Noiseless original projection. **(b)** Original projection at 10 % noise level. **(c)** Original projection at 20 % noise level. **(d)** Original projection at 40 % noise level.



(a)



(b)

Figure 5.14. Speed of convergence of $\gamma(f_{\mathbf{c}})$ as the function of iterative steps. Reconstruction of the 2D software phantom shown in Figure F.2 using noiseless projections. **(a)** Exponent of $\gamma(f_{\mathbf{c}})$ versus the number of accepted candidate configurations $\mathbf{c}'$ (out of 37 428 iterations in total). **(b)** Exponent of $\gamma(f_{\mathbf{c}})$ versus the total number of iterations.

initial configurations always yields better results in 3D too. (As a matter of fact, the usage of random initial configurations seems to have a stronger detrimental effect in 3D than in 2D.)

Regarding sensitivity with respect to the noise level, the associated plots are in sync with their 2D companions; this is demonstrated in Figure F.11 (cf. Figure 5.12). The extent of distortions caused by individual noise levels can be seen in Figure F.12.

Finally, Figure F.13 shows the typical convergence rate expected for 3D objects in the absence of noise. Though exact iteration numbers and $\gamma(f_{\mathbf{c}})$ values are different than those shown for the respective 2D case (cf. Figure 5.14), they both exhibit quite similar trends.

5.4 Summary

The performance of the algorithms has been investigated using numerous randomly generated configurations as well as with a few manually constructed phantoms. It was demonstrated that 3 or fewer discs can be successfully reconstructed from 2 noisy projections (assuming a noise level not exceeding 10 %), and that reconstruction quality stays high for configurations with up to 5 discs when having 4 noisy projections. Good quality results can be expected up to ≈ 25 % of noise when having 4 projections or more. Similar statistics have been gathered for 3D configurations containing cylinders.

It is remarkable that 3D configurations can be retrieved from 2 projections with pretty good quality when having 6 or fewer spheres, so this scenario is apparently easier for our technique than the one involving cylinders.

As expected, having a suitable initial configuration $\mathbf{c}_0$—as determined automatically from input projections—always guarantees better quality results. On the other hand, starting from a random $\mathbf{c}_0$ invariably yields much worse outcomes: the chance of a successful reconstruction diminishes when having 2 or more discs or cylinders. (The detrimental effect of a random $\mathbf{c}_0$ is especially striking in the 3D case.)

The aforementioned numbers give a justification for the viability of DT approaches based on geometrical priors, even when facing reasonably high amount of noise in the measurements.

Chapter 6

Physical Phantoms: Reconstruction of Objects Parametrized with Geometrical Primitives

In order to verify the real-world applicability and robustness of the algorithms, several successful reconstructions are presented based on physical measurements. To get a better insight into how sensitive the algorithms are to projection distortions, measurements are taken with 3 different radiation modalities.

The results shown below have been published in [22, 33, 104, 105, 121–123].

6.1 Experiment Setup

Except where noted otherwise, the assumptions, approaches and settings described earlier in Section 5.2 are still valid for the physical experiments. In particular, the following aspects remain unchanged: visualization of configurations, and computation of simulated projections.

Contrary to simulations, physical experiments are always started from an automatically determined initial configuration. Moreover, the quality of results can only be verified visually, either by comparing them with the outcome of an alternative reconstruction method (FBP in this case), or by assessing how close they fit the schematic of the test object (if known). Specifically, since the original (reference) configuration is unavailable, no FOM values can be determined.

6.1.1 Phantoms and Physical Measurements

Experimental studies were carried out using five series of measurements. The first of them (Test Case I) was received from HMI[1], while the rest (Test Cases II–V) were contributed by KFKI[2]. As a matter of fact, all the KFKI experiments concerned the same phantom (a so-called "reference cylinder"; see Figure 6.1), a solid cylindrical object with three round bored holes of varying depth.

A short description of the test cases follows below; a few additional technical details are given in Table 6.1.

[1] These measurements were provided by Markus Strobl and Wolfgang Treimer (HMI).
[2] These measurements were supplied by Márton Balaskó (KFKI).

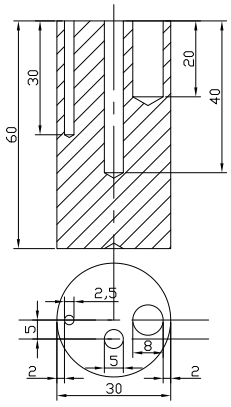


Figure 6.1. Diagram of the phantom considered in Test Cases II–V. (Dimensions are shown in mm.)

Table 6.1. Technical attributes of physical phantoms. $\Delta\vartheta$ denotes the angular difference between successive projection directions. Imaging modalities are abbreviated as: X-ray (X), neutron (N), gamma (G).

	2D Phantom		3D Phantom				
					Case IV		
Parameter	Case I	Case II	Case II	Case III	Full	Clipped	Case V
SIZE	90	155	155	137	113	113	365
HEIGHT	–	–	113	70	185	70	400
N	4	3	3	3	2	3	3
n	60	73	73	73	73	73	360
$\Delta\vartheta$ (°)	3	5	5	5	5	5	1
Modality	N	X	X	N	G	G	N
# of materials	3	2	2	2	3	3	4

Case I (series "jun1_a") A 2D cross-section of an object composed of an annulus and four discs inside; reconstructions of the cross-section obtained via FBP are depicted in Figure 6.2(a)–(b). Whereas this object was supposed to contain only two materials (air for the interior, brass for the annulus and discs), the effective μ_{R} and μ_{D} have been found to differ substantially (with an approximate ratio of $2:1$). Furthermore, input projections $\mathbf{P}_\vartheta$ were only received partially: the left and right sides were absent, hence most of the annulus's projection had to be guessed and completed manually. This is apparent in Figure 6.2(c)–(d).

Case II (series "GF") A Plexiglas (poly(methyl methacrylate) [PMMA]) cylinder with air in all the holes. The deepest (middle) hole contained some aluminum screws at the bottom as a radiocontrast agent. Sadly, these turned out to be completely radiopaque, thus causing 0 intensity pixels in $\mathbf{P}_\vartheta$ (effectively corresponding to an infinite LAC). We had to, therefore, discard the lower part of the input projections. As this series of measurements was the first one received from KFKI, we also had the opportunity to reconstruct just a 2D cross-section of it besides doing a full 3D reconstruction. (Test Cases III–V have only been reconstructed in 3D.) Since some of the projections were damaged in the lower 180° direction range, only those taken at $\vartheta \in [180°, 355°]$ were used. See Figure 6.3 for an overview of the input projections.

Case III (series "AF") An aluminum cylinder[3] containing some amount of fluid in all the holes: ethanol in the thinnest hole and water in the other ones. The upper part of every hole is filled with air. Since both kinds of fluids were completely radiopaque to the neutron radiation, the associated pixels of $\mathbf{P}_\vartheta$ had an intensity of 0, so the majority of the projections again had to be thrown away. This is demonstrated in Figure 6.4.

Case IV (series "FF1") An iron cylinder containing some lead in the two larger holes; the rest of each hole is filled with air. As will be later shown in Section 6.2, it is possible to reconstruct the lead-containing[4] holes on their own using the whole projection, as well as to reconstruct the upper part of all holes containing air separately. For a summary of the respective projections, see Figure 6.5.

Case V (series "Polish1") Similarly to Test Case III, this one is also an aluminum cylinder with varying amounts of fluid in each of the holes, the rest of them being filled with air. (Here, the fluid was acetone for the hole with the largest diameter, and water for the other two holes.) On the other hand, the fluid-containing parts were *not* radiopaque this time, so it was possible to reconstruct them using the whole projection.[5] Even though projections were of reasonably high resolution, they were also suffering from serious distortions and noise.

In all the cases, input projections $\mathbf{P}_\vartheta$ were acquired using the instruments and setup discussed in Section 3.1.2. (More details regarding the radiation source parameters, detector types etc. used for Test Cases II–V can be found in [22].) For Test Cases II–V, $\mathbf{P}_\vartheta$ were provided as raw detector measurements that first had to be converted by applying a suitable logarithmic transform (as per Equation (3.3)) and intensity thresholding / windowing. (The I_{source} required

[3] The usage of Test Cases III and IV has not been published; they were investigated as an extension and improvement upon the results published earlier.

[4] The air-containing upper parts of the holes will then appear as excessive "noise" in the projection error term of $\gamma(f_{\mathbf{c}})$.

[5] Surprisingly, the LAC values of the two kinds of fluids were found to be approximately equal. Moreover, the air-containing upper parts of the holes will contribute as "noise" to the projection error term of $\gamma(f_{\mathbf{c}})$.

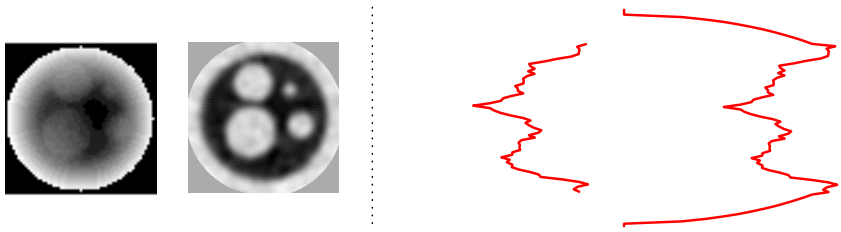


Figure 6.2. Input projection and FBP reconstruction of Test Case I. **(a)** Reconstruction from 60 projections using SNARK [2]. **(b)** Reconstruction from an unknown number (perhaps even hundreds) of projections, including top-view measurements as well. **(c)** Original projection at $\vartheta = 0°$. (Zero level shown as a dotted line.) **(d)** Corrected projection obtained from (c) by completing the missing parts of the ring. (Image (a) is courtesy of Zoltán Kiss and László Ruskó. Image source for (b): HMI.)

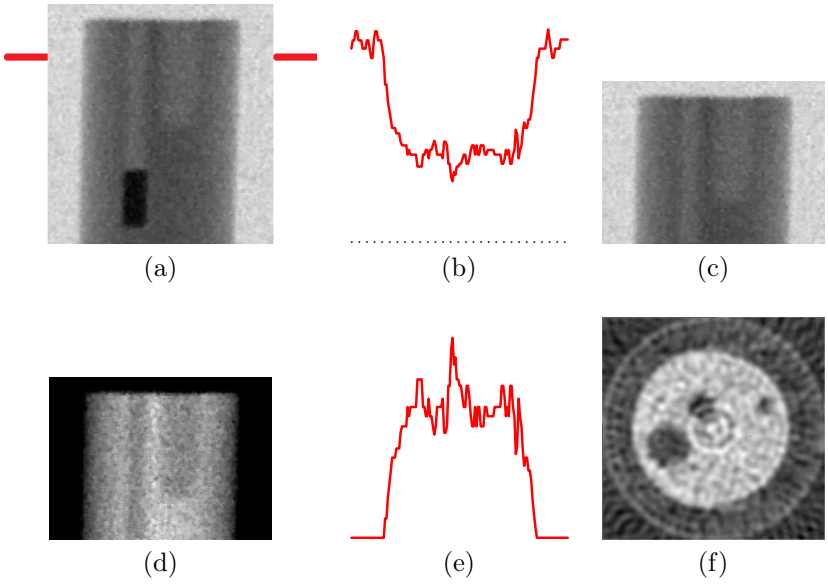
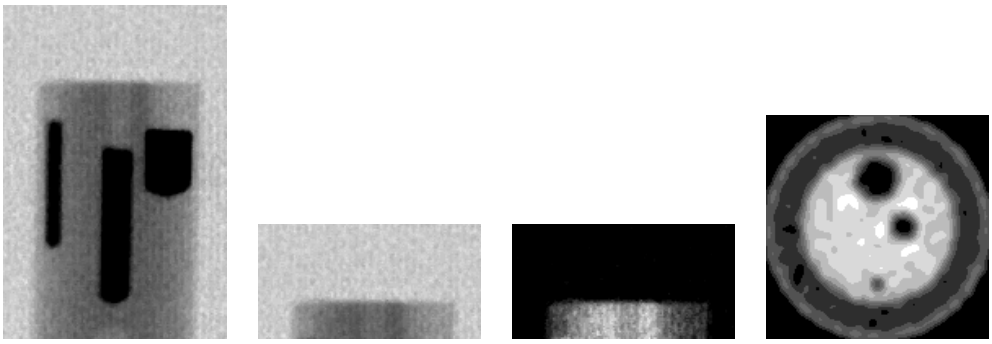
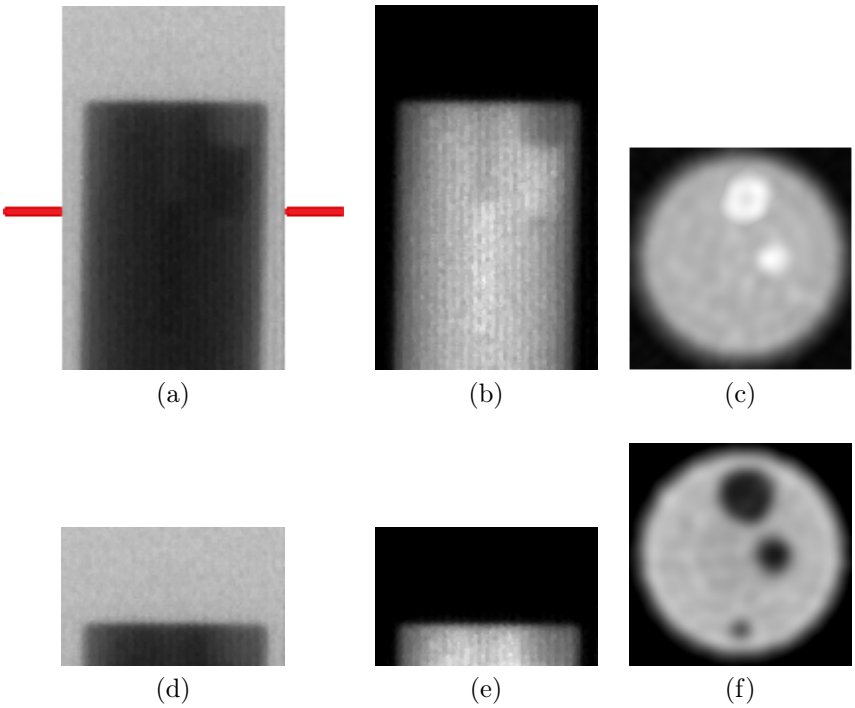


Figure 6.3. Input projection and FBP reconstruction of Test Case II. **(a)** Original projection at $\vartheta = 180°$. **(b)** Projection profile of the cross-section marked in (a) with red lines. (Zero level shown as a dotted line.) **(c)** The remainder of (a) after cropping the unusable part. **(d)** Projection obtained from (c) by applying the logarithmic transform. **(e)** Projection profile of the same cross-section in (d). **(f)** FBP reconstruction of the cross-section from 73 projections produced by SNARK [2]. (Image (f) is courtesy of Zoltán Kiss and László Ruskó.)



(a) (b) (c) (d)

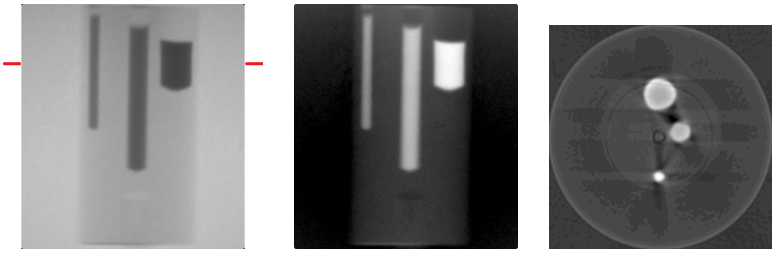
Figure 6.4. Input projection and FBP reconstruction of Test Case III. **(a)** Original projection at $\vartheta = 0°$. **(b)** The remainder of (a) after cropping the unusable part. **(c)** Projection obtained from (b) by applying the logarithmic transform. **(d)** FBP reconstruction of one of the cross-sections from 37 projections produced by SNARK [2]. (Image (d) is courtesy of Zoltán Kiss and László Ruskó.)



(a) (b) (c)

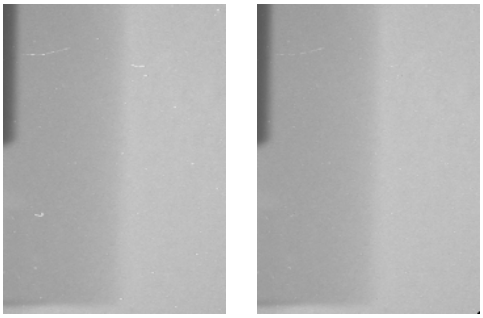
(d) (e) (f)

Figure 6.5. Input projection and FBP reconstruction of Test Case IV. **(a)** Original projection at $\vartheta = 0°$. **(b)** Projection obtained from (a) by applying the logarithmic transform. **(c)** FBP reconstruction of the cross-section marked in (a) with red lines from 37 projections produced by SNARK [2]. **(d)** The remainder of (a) after cropping the upper part. **(e)** Projection obtained from (d) by applying the logarithmic transform. **(f)** FBP reconstruction of one of the cross-sections of (e) from 37 projections produced by SNARK. (Images (c) and (f) are courtesy of Zoltán Kiss and László Ruskó.)



(a) (b) (c)

Figure 6.6. Input projection and FBP reconstruction of Test Case V. **(a)** Original projection at $\vartheta = 0°$. **(b)** Projection obtained from (a) by applying the logarithmic transform. **(c)** FBP reconstruction of the cross-section marked in (a) with red lines from 180 projections produced by SNARK [2]. (Image (c) is courtesy of Zoltán Kiss and László Ruskó.)



(a) (b)

Figure 6.7. Benefits of the pre-processing phase for input projections. **(a)** Cropped-out region of the raw projection of Test Case V at $\vartheta = 0°$. **(b)** Projection obtained from (a) by applying pre-processing steps. (Image (a) is adapted from Figure 6.6(a).)

for the logarithmic transform was determined experimentally as the average background intensity in the raw measurements.) On the other hand, these steps were not necessary for Test Case I because we had already received $\mathbf{P}_\vartheta$ as Radon transform values.

As explained in Section 3.1.3, physical measurements are sometimes prone to various distortions and imperfections. The experiments considered here are not exceptions either, unfortunately. Therefore, before performing the aforementioned logarithmic transform and intensity thresholding, $\mathbf{P}_\vartheta$ had to be **pre-processed** using several techniques: uniformity and intensity correction, median filtering, motion correction using image registration techniques [42, 182], etc.[6] The benefits of these steps are demonstrated in Figure 6.7.

6.1.2 Reconstruction Parameter Settings

SIZE and HEIGHT were set to the corresponding dimension of the input projections $\mathbf{P}_\vartheta$. Moreover, the number N of discs / cylinders were given as part of the *a priori* information.

Reconstructions were always carried out from 4 projections (at $\vartheta = 0°$, $45°$, $90°$ and $135°$ for all experiments except for Test Case II where the directions $180°$, $225°$, $270°$ and $315°$ were taken)

[6] All the pre-processing steps mentioned here were entirely implemented and accomplished by Zoltán Kiss; more details can be found in his thesis [103].

Table 6.2. Parameter settings for physical experiments using $\gamma(f_{\mathbf{c}})$.

	2D Phantom		3D Phantom				
					Case IV		
Parameter	Case I	Case II	Case II	Case III	Full	Clipped	Case V
$r_{\min}$	2.5	3	3	3	3	3	5
$h_{\min}$	–	–	5	5	5	5	5
$\hat{\mu}_{\mathrm{R}}$, $\hat{\mu}_{\mathrm{T}}$	60.126	0	0	0	0	0	0
$\hat{\mu}_{\mathrm{I}}$	0.0013	1.945	1.82	1.858	2.016	2.016	0.181
μ_{R}, μ_{T}	60.5	0	0	0	0	0	0
μ_{I}	0	1.93	1.75	1.9	2	2	0.21
μ_{D}, μ_{SC}	31.85	0	0	0	3.2	0.75	3.825
τ (%)	15	25	15	15	15	15	20
MERGE_O	–	–	10	15	15	15	25
MERGE_r	–	–	5	10	10	10	15

for sake of robustness and due to the bad quality of projections. (Even after the pre-processing corrections mentioned in Section 6.1.1, some distortions could not be entirely eliminated from the input projections, e. g. non-uniformity, precession of the object, or the vertical striping probably caused by detector defects.)

Estimated LAC values $\hat{\mu}_{\mathrm{R}}$, $\hat{\mu}_{\mathrm{T}}$ and $\hat{\mu}_{\mathrm{I}}$ were determined automatically as part of the procedure described in Section 4.5, while $\hat{\mu}_{\mathrm{D}}$ and $\hat{\mu}_{\mathrm{SC}}$ were devised manually after the elimination of the tube and its interior from $\mathbf{P}_{\vartheta}$. All the LAC parameters were then held fixed during reconstruction.

Finally, the noise threshold level τ had to be settled experimentally since the actual noise level was unknown.

Table 6.2 gives a summary of all parameter settings in effect. With the exception of these changes, the values listed in Section 5.2.6 were employed for all remaining parameters.

6.2 Results

This section presents the results obtained from physically measured projections for 2D as well as 3D configurations.

6.2.1 Reconstructions of 2D Objects

The reconstruction of Test Case I, the first 2D physical phantom, is depicted in Figure 6.8. As is apparent, the relative sizes and positions of all the discs approximately match their original settings when compared to the reconstructions obtained with FBP. For reference, one of the simulated projections is shown there as well; this is also in agreement with the respective input projection.

Figure 6.9 shows one of the cross-sections reconstructed from Test Case II, along with the reference FBP reconstruction. Though the two larger holes seem to have the correct locations

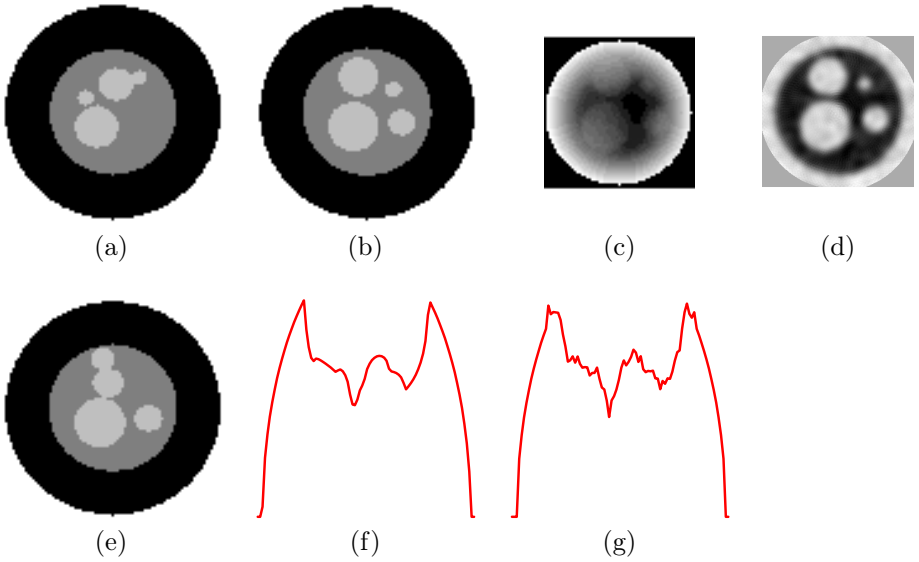


Figure 6.8. Reconstruction of Test Case I from 4 projections. **(a)** Initial configuration. The two smaller discs were added randomly. **(b)** Reconstructed configuration. **(c)–(d)** FBP reconstructions. **(e)** Failed reconstructed configuration from 2 projections. **(f)** Simulated projection of (b) at $\vartheta = 0°$. **(g)** Input projection at $\vartheta = 0°$. (Images (c), (d) and (g) are taken from Figure 6.2.)

and radii, the third one has too small a radius and is somewhat shifted from its ideal position.

Despite the rather noisy input projections (especially for Test Case II), the initial configuration constructed by the method is still quite close to the ideal one (which is unknown, of course). The only issue is that discs with very small radii cannot be detected, and hence have to be substituted with randomly generated ones; this can be seen in Figures 6.8(a) and 6.9(a).

As demonstrated by Figures 6.8(e) and 6.9(d), the algorithm sometimes gets stuck in a local optimum when using only 2 projections. We hypothesize this might be caused by the distortions remaining in input projections.

6.2.2 Reconstructions of 3D Objects

The reconstruction of the full 3D model of Test Case II is shown in Figure 6.10. While the hole with the smallest radius cannot be detected (and thus gets supplied randomly), the algorithm can nevertheless produce a pretty good result. In particular, the radius and position of the thinnest cylinder is also correct, which is definitely an improvement over the 2D case (cf. Figure 6.9).

Test Case V could be also reconstructed with good precision (at least to the extent this can be determined by visual comparisons); see Figure 6.11. This phantom was especially challenging due to the large amount of distortions (e. g. non-uniformity, precession of the object).

Further results[7] for Test Cases III and IV are given in Appendix F; see Figures F.14, F.15 and F.16. As mentioned in Section 6.1.1, Test Case IV could, actually, be reconstructed in two ways: first with a focus on the two lead-containing parts of the holes (completely ignoring the air-filled parts), and second by keeping only the upper region of input projections and targeting

[7]The results of Test Cases III and IV have not been published; they were obtained as an extension and improvement upon the results published earlier.

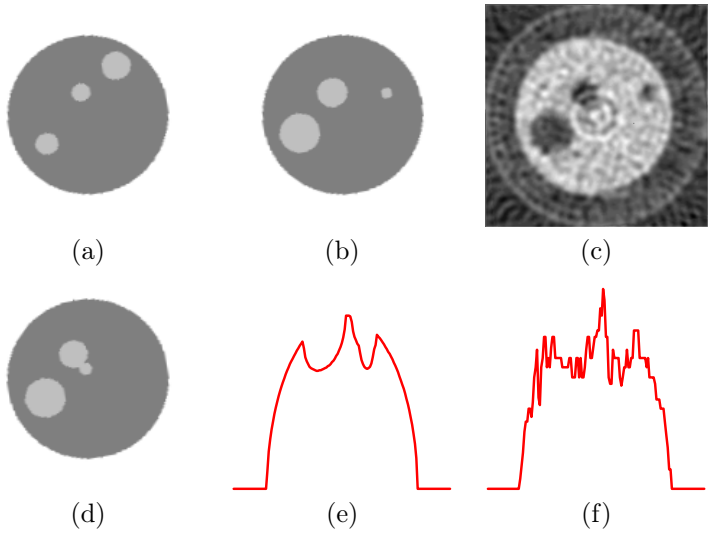


Figure 6.9. Reconstruction of a cross-section of Test Case II from 4 projections. The location of the cross-section is marked in Figure 6.3(a) with red lines. **(a)** Initial configuration. The largest discs at top right was added randomly. **(b)** Reconstructed configuration. **(c)** FBP reconstruction. **(d)** Failed reconstructed configuration from 2 projections. **(e)** Simulated projection of (b) at $\vartheta = 180°$. **(f)** Input projection at $\vartheta = 180°$. (Image (c) is taken and image (f) is adapted from Figure 6.3.)

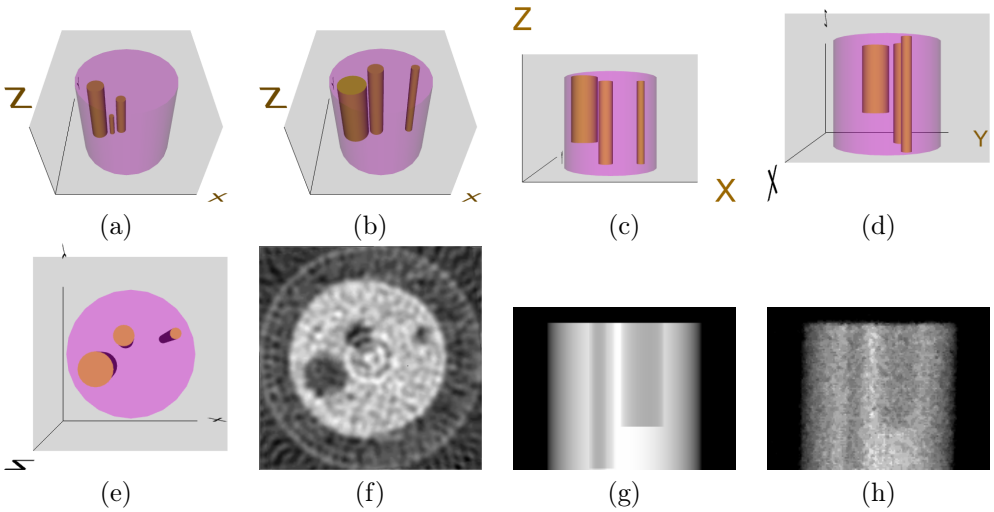


Figure 6.10. Reconstruction of Test Case II from 4 projections. **(a)** Initial configuration. The smallest cylinder was added randomly. **(b)–(e)** Different views of the reconstructed configuration. **(f)** FBP reconstruction of the cross-section marked in Figure 6.3(a) with red lines. **(g)** Simulated projection of (b)–(e) at $\vartheta = 180°$. **(h)** Input projection at $\vartheta = 180°$. (Images (f) and (h) are taken from Figure 6.3.)

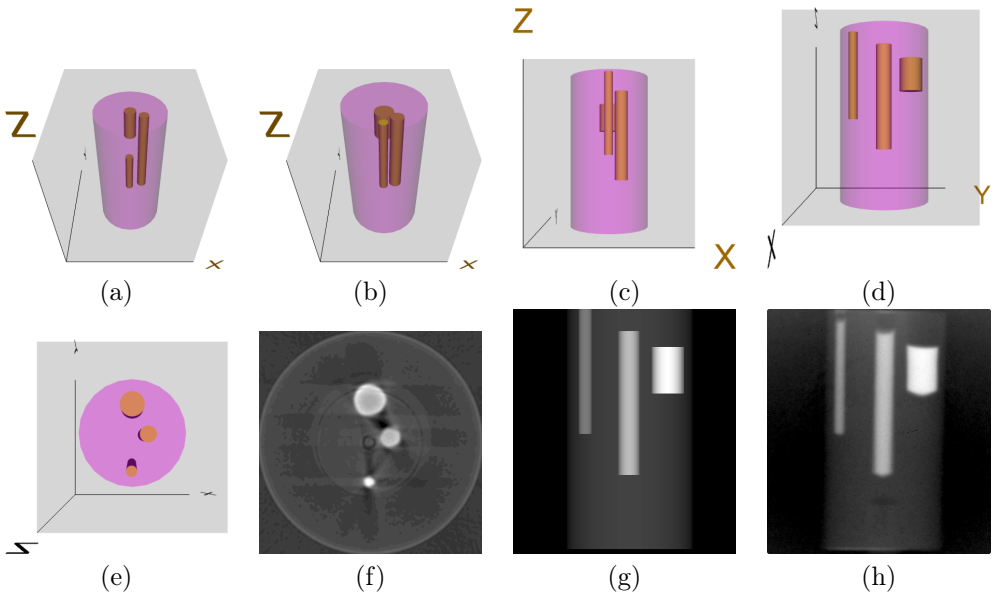


Figure 6.11. Reconstruction of Test Case V from 4 projections. **(a)** Initial configuration. **(b)**–**(e)** Different views of the reconstructed configuration. **(f)** FBP reconstruction of the cross-section marked in Figure 6.6(a) with red lines. **(g)** Simulated projection of (b)–(e) at $\vartheta = 0°$. **(h)** Input projection at $\vartheta = 0°$. (Images (f) and (h) are taken from Figure 6.6.)

the three air-filled holes. As is apparent in Figure F.16, the second kind of reconstruction of Test Case IV is only satisfactory at best: the radii of the two smaller holes are larger than expected.

Like earlier for 2D objects, the automatically determined initial configurations are still very good in most cases.

6.3 Summary

We have demonstrated that the new DT approach introduced earlier is able to produce successful reconstructions from physical measurements as well. In particular, the algorithm was tested with two 2D and four 3D objects whose projections had been taken with 3 different radiation modalities. In all the cases, the accuracy of reconstructions was verified by comparing 2D cross-sections—or top-view VRML renderings—with the result of a classical reconstruction technique (FBP).

It was also shown that it is not always possible to obtain good results when taking only 2 projections (probably due to their bad quality even after pre-processing corrections), but this issue goes away when having at least 4 projections.

Chapter 7

Algorithm: Reconstruction of Deformed Polycrystalline Samples

The determination of the mesoscopic structure of poly-crystalline structures is a demanding and fundamental task in crystallography. This chapter offers a new approach for rendering a 2D grain map of a polycrystal based on an orientation map reconstructed from X-ray diffraction patterns. The orientation and grain maps are produced by a Bayesian discrete tomographic algorithm, applying image-modeling Gibbs priors and a homogeneity condition. The optimization of the objective function is accomplished via the Metropolis algorithm. In order to express the structure of the orientation map, the similarity of orientations is defined by means of unit quaternions.

The author published his results and achievements in [16, 165, 166].

7.1 Motivation

Many materials, natural or artificial, are found in crystalline forms (e. g., metals, ceramics, minerals, precious gems, ice, bones and drugs). As such, crystalline structures are of basic interest to various branches of science, including materials science, physics, geophysics, chemistry, biology, pharmacy, and forensic science. Some typical fields of application incorporate structure determination, deformation analysis (e. g., metals, plastic deformation), and examining the effects of recrystallization.

Solid state physics deals with the structures and properties of rigid matter including crystals [21, 106], while **crystallography** deals with the symmetry and experimental determination of crystal structures. The **crystal structure** is determined by the **crystalline lattice** (the so-called **Bravais lattice**) and the (crystalline) **basis**. The latter is a physical unit (like a collection of atoms or molecules) that is to be located at every lattice point. It is the crystal structure that governs many of the physical, chemical, and mechanical properties of a crystalline material. As a matter of fact, **materials science**, an interdisciplinary field, is occupied with investigating the relationship between the microscopic structure of materials and their macroscopic properties. For more on this topic, see [45, 52, 93, 107] and the aforementioned references.

7.1.1 Problem Description

Most crystalline materials are **polycrystals**, i. e., they are composed of an assemblage of crystals, called **grains** or **crystallites**. In case of an *undeformed* specimen, the lattice within each grain

is typically near-perfect. As such, each grain can be associated with one orientation. In case of *deformed* specimens, the grains will further sub-divide into near-perfect sub-grains. For moderate degrees of deformation the orientations within a grain will vary within a certain orientation range around the so-called **basic orientation**. This spread will typically be smaller than the difference between the basic orientations of neighboring grains.

A piece of specimen may, in fact, be a conglomeration of several different materials. On the other hand, it is also possible to encounter vastly distinct crystalline solid states for a given material, which also may occasionally coexist in a specimen. In either case, a region of material that is chemically uniform and physically distinct (e. g. the crystal structures differ) is dubbed a **material phase** (or simply **phase** for short). Accordingly, polycrystals consisting of a single crystal structure (that is, when all the grains have a common crystal structure) are called **monophase**, otherwise they are regarded as **multiphase**.

In the problem under investigation, the aim is to generate a so-called **orientation map** [7, 45, 80, 82, 107, 147, 163]: an image associating the underlying crystalline orientation with each pixel. In addition, sometimes it is beneficial to establish a **grain map**: a labeling (coloring) of a polycrystalline specimen, where each grain is represented by a color—or direction vector—according to its basic orientation. (That is, sample pixels belonging to a particular grain are to receive the same "grain label.") As a matter of fact, the physical, chemical and mechanical properties of a crystalline material is to a large extent governed by the geometrical features of this 3D complex, including neighboring effects (such as the correlation between the orientation of two neighboring grains and the morphology of the boundary separating them). Hence, the ability to study the grain map as well is very important for certain materials science applications.

There is an inherent difficulty with the problem as stated above: even with the knowledge of the location of the crystal lattice points in space (relative to some fixed coordinate system), the "basic orientation" of the grain (in the sense of how it had to be rotated relative to some reference position in order to get into its current position) is not defined in an obvious and unambiguous manner. Consider, for example, the **simple cubic lattice** whose lattice points are at locations (x, y, z) with $x, y, z \in \mathbb{Z}$. Suppose further that we rotate the lattice by $\frac{3\pi}{4}$ radians counterclockwise around the positive z-axis. The resulting arrangement of lattice points is indistinguishable from what would be obtained by rotating the lattice by $\frac{\pi}{4}$ radians counterclockwise (or, for that matter, clockwise) around the positive z-axis. Thus, due to the lattice structure (more specifically the so-called **crystal symmetry**), the same observable arrangement of lattice points can be obtained by very different rotations; which of these should then be selected for the definition of "orientation"? This important point will be revisited in Section 7.4, but its consequences will be ignored in the discussion until then.

7.1.2 Contemporary Approaches and Pitfalls

Until a few years ago the only techniques for determining grain maps were surface probes such as optical and electron microscopy (EM). X-ray scanning methods have recently been demonstrated with sub-micrometer resolution [129]. The major disadvantage of these techniques is that they are *destructive*, since the sample has to be prepared and sliced before examination. Besides generating only 2D maps, these tools are inherently slow, thereby ruling out any study of the dynamics of the individual grains during typical processes such as annealing or deformation. As a consequence of this lack of experimental studies, existing models of basic industrial processes, such as deformation and annealing, are grossly simplified and typically deal only with average properties of the grains. On the other hand, X-ray and neutron tomography allow for the nondestructive characterization, but with serious limitations: no information is gathered about orientations (as measurements convey only information on material density and radiation

absorption), and the methods apply only to multiphase systems (where these techniques may be nevertheless employed for discovering the distribution of material phases).

A breakthrough was the introduction of three-dimensional X-ray diffraction (3DXRD) microscopy, as has been established at beamline ID11 of the European Synchrotron Radiation Facility (ESRF). 3DXRD is a *nondestructive* imaging technique that uses high-energy monochromatic **X-rays** and utilizes the physical phenomenon known as **diffraction** to produce an image of a 2D layer of the sample in the form of **diffraction patterns**. These patterns (also called **projections**) are recorded by a detector plate while the sample is rotated about an axis perpendicular to the X-ray beam. As this technique is nondestructive, it makes dynamic studies feasible, e. g. allowing for the creation of movies by repeated *in situ* acquisitions of grain maps during the relevant treatment of the sample. In addition, it has an exceptional penetration depth (several millimeters for steel and some centimeters for aluminum), and, as its name suggests, even 3D reconstructions can be performed by imaging multiple cross-sections. Combined with dynamic studies, in favorable cases 3D movies can be produced of grain growth processes [174, 175]. The acquisition of such data is seen as a prerequisite to establish first-principles models for industrially very important processes such as recrystallization and phase transformation. (Detailed descriptions of 3DXRD and information on its software implementation can be found in [14, 153, 154].)

Several methods based on 3DXRD have been proposed to extract grain maps and orientation data. At the time of research this meant that, in all cases, the diffraction spots were first sorted with respect to grains by some external tool, for instance performed by GRAINDEX [130]. Grains were then reconstructed one-by-one either by projecting the outline of the relevant diffraction spots back onto the sample plane [156], or by using an adaptation of ART [31, 136, 155]. These techniques impose severe restrictions on the specimen:

- The number of grains should be small in order to have as few overlapping diffraction spots as possible (since these have to be discarded); and
- The in-grain local variation of orientation ought to be negligible. (In other words, the sample must be *undeformed*, requiring that the so-called mosaic spread of each grain be below one degree. [153])

Even if these restrictions are met, the reconstructed grain map sometimes contains void regions or overlapping grains—both phenomena are undesirable in practice.

The author was introduced to this field and the aforementioned topics during his scholarship at City University of New York[1] (CUNY) and internship at Risø National Laboratory[2] (Risø for short), supervised by professors Gabor T. Herman and Henning Friis Poulsen, respectively. This culminated in a new method, as presented in the rest of the chapter, which can also be regarded as an application of 3DXRD. However, it differs from all aforementioned techniques due to its reliance on DT, which may be a suitable tool to get good results even if the aforesaid restrictions are not satisfied. In particular, as will be demonstrated, we can study *deformed* specimens with the new approach.

7.2 Reconstructing Orientation Maps

To overcome the problems mentioned in Section 7.1.2, we have chosen the following approach. Two-dimensional sections (layers) of the sample are illuminated sequentially and reconstructed

[1] Discrete Imaging and Graphics Group, Department of Computer Science, The Graduate Center, City University of New York, New York, NY, USA

[2] Center for Fundamental Research: "Metal Structures in Four Dimensions", Risø National Laboratory, Roskilde, Denmark

separately (and this can be achieved in practical settings). Instead of tracking and selecting diffraction spots to build a grain map, the unaltered projections are used to reconstruct a 2D orientation map from which we derive the grain map. That is, the aim is to reconstruct a vector-valued image of a cross-section of the specimen from a set of X-ray diffraction patterns acquired via 3DXRD. To the knowledge of the author, no transform or algebraic reconstruction algorithm for reconstructing such a vector field was readily available at the time of research (see, e. g., [153]).

This procedure is more natural than the techniques mentioned above, in the sense that it takes advantage of every measurement, and that it returns a discrete solution as opposed to the continuous one produced by ART. (In fact, these GRAINDEX-based methods are only applicable to undeformed samples; they cannot work for deformed specimens, in which—by the nature of plastic deformation—a range of orientations is associated with each grain.)

7.2.1 Prerequisites

The following assumptions are made:

- The sample is a *monophase polycrystal* with a known crystal structure (i. e., every grain has the same crystal structure);
- The circumference of the sample in the layer to be reconstructed and a discretization of this layer are both established;
- Orientations are defined with respect to the image acquisition system whose geometry and technical parameters are known; and
- Orientations are allowed to vary arbitrarily in general—hence permitting both undeformed and *deformed* samples—, but the probability of encountering a given orientation map is to be driven by certain *a priori* known properties it exhibits:
  - Neighboring sample pixels are encouraged to have similar orientations (within some range);
  - The presence of a major dissimilarity between the orientation of adjacent pixels constitutes a grain boundary; and
  - The layout of such boundaries can be described with a small set of pixel neighborhood configurations (i. e. local image features).

Furthermore, since there are finitely many grains in the cross-section (though the total number may be unknown), the set of possible basic orientations is also finite (although the value of the particular elements is unknown). This way the problem becomes a DT reconstruction task. This still holds even when allowing variations in the orientation of individual pixels of a grain: though the set of permissible orientations seems infinite, it can be considered finite—though of enormous cardinality—for practical purposes due to the aforementioned assumptions and the properties of diffraction and the imaging system (see Section 3.2). (In fact, as will be later discussed, the set $\mathcal{O}$ of crystalline orientations has been indeed chosen so that it is known and finite; see Section 8.1.2, Equation (8.1).)

7.2.2 Objective Function

Our reconstruction procedure is based on the Bayesian technique described in [51]. Whereas the method from [51] deals with gray-valued images, some details have to be adjusted to be applicable to orientations instead of gray levels.

Let a finite 2D section of the specimen of interest, $\mathcal{D} \subset \mathbb{Z}^2$, be discretized into a set of pixels i. Furthermore, let

$$o\colon \mathcal{D} \to \mathcal{O}, \quad i \mapsto o(i) \tag{7.1}$$

denote a 2D digital **orientation map**, that is a vector-valued image whose pixels i are assigned an orientation $o(i) \in \mathcal{O}$. While it is fairly straightforward to define pixel values in terms of density (no matter whether it describes an absorption or an intensity), the representation of rotations—let alone crystalline orientations—is more complicated. It is also important to see that the choice of representation is a critical decision, for it has many effects on the feasibility and the efficiency of an algorithm (such as how to define and compute the "distance" of orientations), and thus it needs some deliberation. These issues are discussed in the forthcoming Sections 7.4 and 8.1.2; the exact definition of $\mathcal{O}$ will be given thereafter (see Equations (7.20) and (8.1)).

Most DT methods employ some kind of *a priori* information about the object to be reconstructed in order to get a reconstruction appropriate for the field of application. Since the orientation of a pixel inside a particular grain is likely to be similar to that of its neighbors, we decided to model orientation maps as a **Markov random field** (MRF), that can be characterized by its joint Gibbs distribution [40, 71, 192]. Using an MRF image model has the advantage that only local features need to be specified, allowing for a compact image description and an efficient algorithm. Such a formulation had earlier been successfully used for binary [47, 132, 137] and multilevel DT [50, 133].

The probability of occurrence of an orientation map o is given by the Gibbs distribution (cf. Equation (2.12))

$$\pi(o) := \frac{1}{Z_{\beta;o}} \mathrm{e}^{-\beta H(o)}, \tag{7.2}$$

where β and $Z_{\beta;o}$ correspond to the inverse of the temperature and to the partition function, respectively. It is the construction of the Hamiltonian $H(o)$ that carries the desired properties of an orientation map o. As presented more precisely below, this function is calculated as a weighted sum of **clique potentials**. It is defined as

$$H(o) := H_1(o) + H_2(o), \tag{7.3}$$

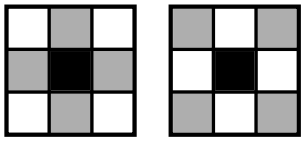
The first term of $H(o)$ establishes a **homogeneity** condition:

$$H_1(o) := -\left(\sum_{C \in \mathcal{C}_+} \lambda_1 \Phi_C(o) + \sum_{C \in \mathcal{C}_\times} \lambda_2 \Phi_C(o) \right), \tag{7.4}$$

with

$$\Phi_{\{i,j\}}(o) := \mathrm{e}^{-\frac{\left(d\left(o(i),o(j)\right)\right)^2}{2\delta^2}}. \tag{7.5}$$

$\mathcal{C}_+$ denotes the set of all horizontal and vertical **pair cliques** (i. e., pairs of adjacent pixel indexes), while $\mathcal{C}_\times$ is that of all diagonal pair cliques. Actually, both classes comprise so-called doubleton (i. e. 2nd order) pixel cliques well known in digital topology: $\mathcal{C}_+$ represents all the **4-connected neighbors**, while $\mathcal{C}_\times$ covers the remaining cliques of **8-connected neighbors**. This is demonstrated in Figure 7.1. The real-valued coefficients λ_1 and λ_2 determine the contribution of each type of interaction (i. e. the associated clique potential), and the free parameter $\delta >$



(a) (b)

Figure 7.1. 2nd order cliques formed by the black pixel in the center and one gray pixel. **(a)** $\mathcal{C}_+$: 4-connected neighborhood. **(b)** $\mathcal{C}_\times$: 8-connected neighborhood (excluding neighbors that are also 4-connected).

0, together with the λ_k ($k \in \{1,2\}$), controls the degree of homogeneity. (As a matter of fact, δ is related to the maximal expected **orientation spread** over all the grains, or, more explicitly, to the maximal distance likely to be encountered between the orientations of adjacent pixels within any grain.) The nonnegative function $d\big(o(i),o(j)\big)$ measures the **distance** of two orientations (i. e. it is a measure of the **disorientation**), a smaller value indicates more similar orientations. Since the definition of $d\big(o(i),o(j)\big)$ depends inherently on the representation of crystalline orientations of $\mathcal{O}$, it will be specified later in Section 7.4 (see Equations (7.24) and (7.25)). Finally, $\Phi_{\{i,j\}}(o)$ measures the **similarity** of orientations associated with the pair clique $\{i,j\}$ of o (inspired by Gaussian radial basis functions used in e. g. machine learning).

The second term in $H(o)$, namely $H_2(o)$, models the **borders** between neighboring grains:

$$H_2(o) \coloneqq -\sum_{C\in\mathcal{C}_{3\times3}} \big(\kappa_1 I_C^1(o) + \kappa_2 I_C^2(o) + \kappa_3 I_C^3(o)\big)\ . \tag{7.6}$$

$\mathcal{C}_{3\times3}$ denotes the set of all cliques with 3×3 blocks of pixels, κ_k are scalars ($k\in\{1,2,3\}$), and $I_C^k(o)$ are $\{0,1\}$-valued functions corresponding to the clique configurations shown in Figure 7.2. For instance, $I_C^1(o)$ is defined as

$$I_C^1(o) \coloneqq \begin{cases} 1, & \text{if } d\big(o(k),o(\mathrm{c})\big)\le\delta\ \ \forall k\in\{\mathrm{w},\mathrm{nw},\mathrm{n},\mathrm{ne},\mathrm{e}\} \\ & \text{and } d\big(o(k),o(\mathrm{s})\big)\le\delta\ \ \forall k\in\{\mathrm{sw},\mathrm{se}\} \text{ and } d\big(o(\mathrm{n}),o(\mathrm{s})\big)>\Delta\,, \\ 1, & \text{if } d\big(o(k),o(\mathrm{c})\big)\le\delta\ \ \forall k\in\{\mathrm{w},\mathrm{sw},\mathrm{s},\mathrm{se},\mathrm{e}\} \\ & \text{and } d\big(o(k),o(\mathrm{n})\big)\le\delta\ \ \forall k\in\{\mathrm{nw},\mathrm{ne}\} \text{ and } d\big(o(\mathrm{n}),o(\mathrm{s})\big)>\Delta\,, \\ 1, & \text{if } d\big(o(k),o(\mathrm{c})\big)\le\delta\ \ \forall k\in\{\mathrm{s},\mathrm{sw},\mathrm{w},\mathrm{nw},\mathrm{n}\} \\ & \text{and } d\big(o(k),o(\mathrm{e})\big)\le\delta\ \ \forall k\in\{\mathrm{se},\mathrm{ne}\} \text{ and } d\big(o(\mathrm{w}),o(\mathrm{e})\big)>\Delta\,, \\ 1, & \text{if } d\big(o(k),o(\mathrm{c})\big)\le\delta\ \ \forall k\in\{\mathrm{s},\mathrm{se},\mathrm{e},\mathrm{ne},\mathrm{n}\} \\ & \text{and } d\big(o(k),o(\mathrm{w})\big)\le\delta\ \ \forall k\in\{\mathrm{sw},\mathrm{nw}\} \text{ and } d\big(o(\mathrm{w}),o(\mathrm{e})\big)>\Delta\,, \\ 0, & \text{otherwise}\,. \end{cases} \tag{7.7}$$

Indexes c, n, ne, e, se, etc. (denoting center, north, northeast, east, and southeast, respectively) are used to select individual pixels within a 3×3 clique. The first case in Equation (7.7), for example, corresponds to the upper-left clique configuration in Figure 7.2 (labeled "horizontal") in which the orientation of every pixel labeled by "`x`" is similar to that of pixels labeled by "`a`". Each $I_C^k(o)$ models borders in a particular set of directions, while the factors κ_k govern the strength of each kind of interaction (i. e. the associated clique potential). It is obvious that δ, which also occurs in Equation (7.5), directly determines the maximal distance of similar orientations. Likewise, $\Delta\ge\delta$ represents the minimal **degree of separation** of the basic orientations of neighboring grains (called the **misorientation** in materials science [145, 146]).

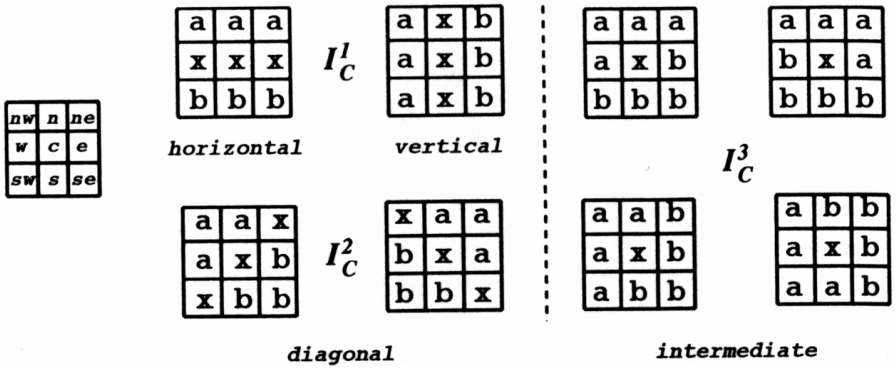


Figure 7.2. Clique configurations for borders between different orientations. Pixels labeled by "a" and "b" have dissimilar orientations. Pixels labeled as "x" all have orientations similar to those labeled by either "a" or "b". The legend on the left shows the meaning of pixel indexes used in Equation (7.7). (Image source: [51].)

The actual value of the coefficients κ_k ($k \in \{1,2,3\}$) and λ_l ($l \in \{1,2\}$) has to be determined experimentally. As recommended in [51], they may be chosen so that $\frac{1}{2}\kappa_3 = \kappa_2 = \kappa_1$ and $\lambda_2 = \frac{1}{\sqrt{2}}\lambda_1$. The first condition compensates for the fact that horizontal, vertical, and diagonal border elements can always be found in pairs at the border between two regions. The second encourages spatially isotropic interaction in the modeling of homogeneity. See Section 8.2.6 for an overview of exact parameter values taken for reconstructions.

Having defined the desired distribution of orientation maps, we can now formulate the **posterior distribution** in terms of the physical measurements. According to **Bayes' theorem**, the probability of an orientation map o given the set of input diffraction patterns $\mathbf{P}$ and the prior $\pi(o)$ is

$$\Pr(o|\mathbf{P}) = \frac{L(\mathbf{P}|o)\pi(o)}{\Pr(\mathbf{P})}\,, \tag{7.8}$$

where $L(\mathbf{P}|o)$ is a probabilistic **likelihood** function for modeling noise [14]. We are looking for the orientation map o that maximizes $\Pr(o|\mathbf{P})$. However, as $\Pr(\mathbf{P})$ is independent of o, it is sufficient to maximize the numerator of Equation (7.8) over the set of all possible orientation maps, because the o that maximizes $\Pr(o|\mathbf{P})$ is the same as the one that maximizes $L(\mathbf{P}|o)\pi(o)$:

$$\arg\max_o \Pr(o|\mathbf{P}) = \arg\max_o L(\mathbf{P}|o)\pi(o)\,.$$

For convenience, we chose

$$L(\mathbf{P}|o) := \mathrm{e}^{-\beta\left(\alpha\|\mathbf{P}_o - \mathbf{P}\|_1\right)}\,, \tag{7.9}$$

where $\mathbf{P}_o$ denotes the set of simulated diffraction patterns given the orientation map o, $\|\cdot\|_1$ is the usual ℓ_1 norm, and $\alpha > 0$ determines the relative importance of the projection error term $\|\mathbf{P}_o - \mathbf{P}\|_1$.

Accordingly, the aim is to *maximize* the following **objective function** (cf. Equation (2.16)):

$$\begin{aligned}\gamma(o) := L(\mathbf{P}|o)\pi(o) &= \frac{1}{Z_{\beta;o}}\mathrm{e}^{-\beta\left(H(o)+\alpha\|\mathbf{P}_o - \mathbf{P}\|_1\right)} \\ &= \frac{1}{Z_{\beta;o}}\mathrm{e}^{-\beta\left(H_1(o)+H_2(o)+\alpha\|\mathbf{P}_o - \mathbf{P}\|_1\right)}\,.\end{aligned} \tag{7.10}$$

This, effectively, implies that the sum $H(o) + \alpha \|\mathbf{P}_o - \mathbf{P}\|_1$ needs to be *minimized*. (This is achieved by having $H(o)$ as negative and the projection error term $\|\mathbf{P}_o - \mathbf{P}\|_1$ as close to zero as possible.) Such an optimization can be performed, for example, using the stochastic techniques discussed in Section 2.2.3. Using either of these schemes means that the optimization progresses iteratively, so that in each iteration step the orientation of a random pixel is altered conforming to the probability distribution $\gamma(o)$. It is an important aspect that the computation of the posterior distribution $\Pr(o|\mathbf{P})$—hence the computation of $\gamma(o)$ as well—is local, due to the MRF properties of $\pi(o)$, so it can be accomplished in a quite efficient way.

7.2.3 Optimization

In the present implementation the optimization of $\gamma(o)$ is carried out using the Metropolis algorithm (see Section 2.2.3). A major advantage of this iterative Monte Carlo approach is that it does not need the value of $\gamma(o)$ itself during optimization.

At every iteration of the Metropolis algorithm, a new approximation o' is constructed according to the following principle:

1. The orientation $o(i)$ of a pixel i may be replaced with one of the nearest neighbors in orientation space of $o(i)$ itself or of any of its 4-connected neighboring pixels. (See below for an explanation.)

More specifically, o' is generated from o as shown in Algorithm 7.1. Line 4 relies on the set $\mathcal{N}\left(o(i)\right)$ of **neighbors** of orientation $o(i)$. Since the interpretation of "orientation adjacency" heavily depends on the choice of $\mathcal{O}$, its definition will be given later (see Equation (8.4)).

Algorithm 7.1. One iterative step for obtaining a new orientation map o'.

1 Let $o'(i) \coloneqq o(i)$ for all $i \in \mathcal{D}$
2 Randomly choose a pixel $i \in \mathcal{D}$
3 Let $J \coloneqq \{\, j \in \mathcal{D} \mid j \text{ is a 4-connected neighbor of } i \,\}$
4 Randomly pick $o'(i)$ from $\{o(i)\} \cup \mathcal{N}\left(o(i)\right) \cup \bigcup_{j \in J} \Big(\{o(j)\} \cup \mathcal{N}\left(o(j)\right) \Big)$
5 Decide on whether to accept o' based on the Metropolis probability $p_{o'}$

The optimization procedure can, in theory, be started from an arbitrary orientation map o_0, but—as in the case of parametric object reconstructions discussed in Chapter 4—the closer the initial o_0 is to the expected global optimum, the faster the convergence of the algorithm will be (and the less the chance will be to get stuck in local optima). The exact way of its construction is discussed in Section 8.2.4.

Iterations continue until some termination criteria are satisfied; these will be later detailed in Section 8.2.6.

7.3 Reconstructing Grain Maps Directly

Once the optimization of $\gamma(o)$ has finished, it is a rather straightforward procedure to derive a grain map based on the optimal orientation map o. For instance, one can apply a standard **connected component** (or **labeling**) algorithm [29, 177, 181] to o, using the disorientation $d\left(o(i), o(j)\right)$ as a similarity / distance measure (in particular, looking for grain boundaries based on a fixed misorientation threshold). (These techniques classify the pixels of an image into so-called **regions** by assigning a unique region label to every pixel.) However, as explained in Section 8.3.1, the optimization of $\gamma(o)$ progresses rather slowly for practical purposes. To

overcome this concern, we make explicit use of the discrete and nonrandom nature of the microstructure. We assume that it is meaningful to associate the specimen with a grain map g as well as an orientation map o, and attempt reconstructing both simultaneously. This is motivated by the fact that the grain structure exhibits additional properties, which may be exploited in the reconstruction:

- The grains can be approximated by discrete objects in the sense that a given voxel in the sample either fully belongs to a given grain or not at all.
- The grains are simply-connected (i. e. 4-connected) 3D space filling objects within the borders given by the sample geometry.
- The grain boundaries are smooth.
- The physics underlying the generation of the structure implies that grain maps tend to resemble each other, once they have been scaled to the average grain size. Hence, it is possible to describe local properties statistically.

The simultaneous reconstruction of o and g has several advantages. First, the configuration space can be vastly reduced, bringing about a serious improvement in the speed of computation; see Section 8.3.2. Moreover, this also allows for an incorporation of additional information about the microstructure to be incorporated into the solution. Finally, as mentioned earlier, many applications of materials science demand an accurate determination of both orientation maps and grain maps.

We start by letting

$$g\colon \mathcal{D} \to \mathcal{G}, \quad i \mapsto g(i) \tag{7.11}$$

denote a 2D digital **grain map**, that is a scalar-valued image whose pixels i are assigned a grain label $g(i) \in \mathcal{G}$. The latter, in turn, is defined as

$$\mathcal{G} \coloneqq \{1, \ldots, G\} \tag{7.12}$$

where G denotes the number of grains in section $\mathcal{D}$ of the specimen. Clearly, if i and j are pixels in the same grain, then $g(i) = g(j)$. The notation g_ℓ will be used to designate the set $\{\, i \in \mathcal{D} \mid g(i) = \ell \,\}$ of indexes of all the pixels i associated with the grain labeled $\ell \in \mathcal{G}$. In addition, the special grain label 0 will also be employed during the course of reconstruction in order to indicate so-called **ambiguous pixels** whose grain membership is not yet decided.

Two more concepts need to be established before proceeding. The **basic orientation** of a set of orientations $o(i_1), \ldots, o(i_R)$ is represented by the orientation $o_{\text{basic}} \in \mathcal{O}$ that minimizes $\sum_{r=1}^{R} d\left(o(i_r), o_{\text{basic}}\right)$. (The exact definition of $\mathcal{O}$ and $d\left(o(i), o(j)\right)$ will be given later; see Equations (7.20) and (8.1), as well as Equations (7.24) and (7.25), respectively.) The **orientation spread** of grain g_ℓ is defined as the maximal distance between the basic orientation $o_{\text{basic};\ell}$ of g_ℓ and the orientation of any pixel constituting g_ℓ:

$$\max_{i \in g_\ell} d\left(o(i), o_{\text{basic};\ell}\right). \tag{7.13}$$

7.3.1 Prerequisites

Besides the assumptions taken in Section 7.2.1, the following preconditions shall also hold:

- The number G of grains located in the 2D sample layer of interest is fixed and known beforehand.

- If an initially undeformed specimen is deformed, the boundaries between grains get distorted. At the same time the lattice is no longer homogeneous within each grain, but there is a spatial variation of orientation. With increasing amount of deformation, these distortions become larger and larger. In the following we confine ourselves to **moderate** degrees of **deformation**, which refers to the case in which it is possible to derive the corresponding (deformed) grain map from a given orientation map. This restriction is expected to allow reconstructions of acceptable quality even under unfavorable circumstances (e. g., high background radiation level or low photon count).
- An initial guess of basic orientations, orientation spreads, centers-of-mass, and approximate morphologies of grains should be available. A program called GRAINSWEEPER [153] is under development, its specific purpose is to provide such information on grain maps and orientation maps via the analysis of 3DXRD diffraction patterns.
- The statistical distribution of typical grain morphologies in moderately deformed orientation maps is available.

7.3.2 Objective Function

Our reconstruction procedure is an extension of the Bayesian technique described in Section 7.2. Furthermore, the algorithm presented below is a generalization of the stochastic algorithm for undeformed specimens devised in [14, 15] to moderately deformed samples. Since the orientation of a pixel within a particular grain is likely to be similar to that of its neighbors, and all moderately deformed grain maps show similar morphological features (depending on the material and the magnitude of deformation, of course), we decided to model both maps simultaneously by a Gibbs distribution. This has the advantage that only *local* features need to be specified, making the description compact and the algorithm efficient.

The probability of occurrence of a grain map / orientation map pair (g, o) is given by, similarly to Equation (7.2),

$$\pi(g,o) := \frac{1}{Z_{\beta;g,o}} \mathrm{e}^{-\beta H(g,o)} \,, \tag{7.14}$$

where the Hamiltonian $H(g,o)$ carries the desired properties of a grain map g and of an orientation map o. It is defined as

$$H(g,o) := H_1(g,o) + H_2(g) \,. \tag{7.15}$$

The first term of $H(g,o)$, again, establishes a homogeneity condition:

$$H_1(g,o) := -\sum_{\ell=1}^{G} \left(\sum_{C \in \mathcal{C}_{+,\ell}} \lambda_{1;g} \Phi_C(o) + \sum_{C \in \mathcal{C}_{\times,\ell}} \lambda_{2;g} \Phi_C(o) \right) \,, \tag{7.16}$$

with $\Phi_{\{i,j\}}(o)$ exactly as defined in Equation (7.5). $\mathcal{C}_{+,\ell}$ denotes the set of all horizontal and vertical pair cliques (i. e., pairs of adjacent pixel indexes) within grain g_ℓ, while $\mathcal{C}_{\times,\ell}$ is that of all diagonal pair cliques inside grain g_ℓ. Finally, the real-valued coefficients $\lambda_{1;g}$ and $\lambda_{2;g}$ determine the contribution of each type of interaction (i. e. the associated clique potential).

The second term in $H(g,o)$, namely $H_2(g)$, models the borders between neighboring grains. That is, the purpose of this term is to capture typical grain-like features of moderately deformed maps. This is achieved by expressing $H_2(g)$ in terms of appropriately chosen set of clique configurations and associated potentials. Before proceeding with its exact definition, it should be noted that this Gibbs distribution is, in fact, defined on multicolored images (viz. g). However,

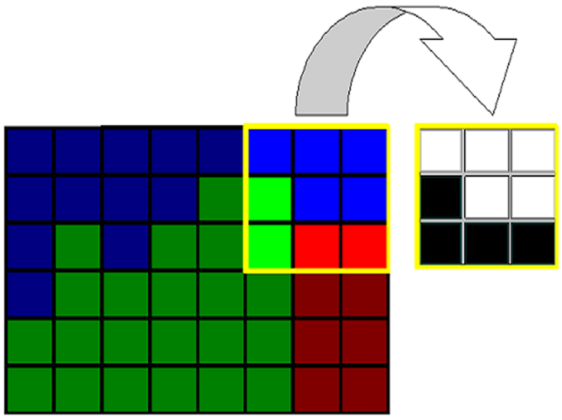


Figure 7.3. Obtaining 3×3 binary local configurations from multicolored grain map images. Pixels having the grain label of the central pixel are considered white; the rest are mapped to black. (Adapted from: [15].)

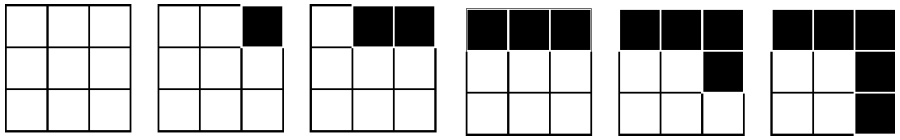


Figure 7.4. The configurations $\mathcal{C}_1, \ldots, \mathcal{C}_6$ of a 3×3 clique that we use in our model of grain morphologies. Configurations not in any of $\mathcal{C}_1, \ldots, \mathcal{C}_6$ are put into $\mathcal{C}_0$ (not shown). (Adapted from: [14–16].)

in order to keep the number of parameters low, we compute $H_2(g)$ by suitably resorting to binary configurations, as is explained below.

For each pixel in the multicolored grain map g, we define its **local configuration** as a 3×3 array of black and white pixels as follows: the central pixel is always white and any other pixel is white if, and only if, it belongs to the same grain as the central pixel. This is demonstrated in Figure 7.3. These configurations are partitioned into seven equivalence classes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_6$, each containing configurations of similar morphology, such as "grain interior," "grain edge," etc. For $1 \leq k \leq 6$, the class $\mathcal{C}_k$ consists of the configurations illustrated in Figure 7.4 and all the configurations that can be obtained from it by a sequence of 90° rotations around the center and mirror images about the central vertical line. Configurations not in any of $\mathcal{C}_1, \ldots, \mathcal{C}_6$ are put into $\mathcal{C}_0$. (The choice of these clique configurations was motivated by [14, 15, 132].)

Now we can define $H_2(g)$ as

$$H_2(g) \coloneqq -\sum_{k=0}^{6} \kappa_{k;g} N(\mathcal{C}_k, g)\,, \tag{7.17}$$

where the factors $\kappa_{k;g}$ are scalars ($0 \leq k \leq 6$), governing the strength of each kind of interaction (i. e. the clique potential associated with class $\mathcal{C}_k$) [132]. On the other hand, $N(\mathcal{C}_k, g)$ counts the number of times a configuration from $\mathcal{C}_k$ occurs in g.

Notably, the definition of $H_2(g)$ does not require a parameter analogous to Δ that was needed for $H_2(o)$. This is so because $H_2(g)$ defines grain boundaries based on $\mathcal{C}_k$ and g, whereas $H_2(o)$

uses the notion of misorientation of adjacent $\big(o(i), o(j)\big)$ pixels belonging to different grains using the distance threshold Δ.

The actual value of the coefficients $\kappa_{k;g}$ $(0 \leq k \leq 6)$ and $\lambda_{l;g}$ $(l \in \{1, 2\})$ has to be settled experimentally. $\lambda_{l;g}$ can be, for instance, determined as λ_l since they serve an analogous purpose. As to $\kappa_{k;g}$, these potentials shall be chosen so that random samples from the Gibbs distribution resemble the arrangement of grains in polycrystals. As demonstrated in [14, 15], the effects of such a term are very robust with respect to the choice of clique potentials. Moreover, moderately deformed grains, by definition, used to be undeformed before deformation. As the morphology changes during deformation are relatively small, we claim that the $\kappa_{k;g}$ employed in aforementioned papers and the set of configurations shown in Figure 7.4 are appropriate for moderately deformed grain maps too. Exact parameter values taken for reconstructions are listed in Section 8.2.6.

Having defined the desired distribution of grain map / orientation map pairs, we can now formulate the posterior distribution

$$\Pr(g, o|\mathbf{P}) = \frac{L(\mathbf{P}|g, o)\pi(g, o)}{\Pr(\mathbf{P})}\,, \tag{7.18}$$

where $L(\mathbf{P}|g, o)$ is a probabilistic likelihood function for modeling noise. We are looking for the grain map / orientation map pair (g, o) that maximizes $\Pr(g, o|\mathbf{P})$. For convenience, we chose $L(\mathbf{P}|g, o) \coloneqq L(\mathbf{P}|o)$ because diffraction patterns are solely determined by o and independent of g.

Accordingly, the aim is to *maximize* the following **objective function** (cf. Equation (7.10)):

$$\begin{aligned}\gamma(g, o) \coloneqq L(\mathbf{P}|g, o)\pi(g, o) &= \frac{1}{Z_{\beta;g,o}} \mathrm{e}^{-\beta\left(H(g,o)+\alpha\|\mathbf{P}_o-\mathbf{P}\|_1\right)} \\ &= \frac{1}{Z_{\beta;g,o}} \mathrm{e}^{-\beta\left(H_1(g,o)+H_2(g)+\alpha\|\mathbf{P}_o-\mathbf{P}\|_1\right)}\,.\end{aligned} \tag{7.19}$$

Such an optimization can be performed using the same techniques employed for $\gamma(o)$ before.

7.3.3 Optimization

Like in the case of $\gamma(o)$, the optimization of $\gamma(g, o)$ is carried out using the Metropolis algorithm. At every iteration, a new pair (g', o') of approximations is constructed according to three principles:

1. An ambiguous pixel may only be given the label and orientation of an adjacent non-ambiguous pixel.

2. A non-ambiguous pixel may retain its association with a grain while its orientation may be replaced with one of its nearest neighbors in orientation space.

3. A non-ambiguous pixel at a grain boundary may be given the same label and orientation as a neighboring pixel associated with an adjacent grain. Note that the change of the grain label should only be permitted if it did not result in the complete "vanishing" of any grains. (That is, the condition $|g_\ell| \geq 1$ must hold all the time for every $\ell \in \mathcal{G}$.)

More specifically, (g', o') is generated from (g, o) as shown in Algorithm 7.2. As is apparent, line 10 utilizes the notion $\mathcal{N}\big(o(i)\big)$ of orientation neighbors introduced earlier in Section 7.2.3.

The optimization procedure can, in theory, be started from an arbitrary pair (g_0, o_0) of maps, but g_0 is usually initialized so that most of the pixels be assigned the ambiguous label (at which

Algorithm 7.2. One iterative step for obtaining a new grain map / orientation map pair (g', o').

```
Let g'(i) := g(i) and o'(i) := o(i) for all i ∈ 𝒟
Randomly choose a pixel i ∈ 𝒟
if at least one 4-connected neighbor of i is non-ambiguous then
    Randomly choose a non-ambiguous 4-connected neighbor, pixel j
    if pixel i is ambiguous then
        Set g'(i) := g(j) and o'(i) := o(j)
        Accept (g', o') unconditionally
    else
        if g(i) = g(j) then
            Randomly pick o'(i) from {o(i), o(j)} ∪ 𝒩(o(i)) ∪ 𝒩(o(j))
            Decide on whether to accept (g', o') based on the Metropolis probability p_{g',o'}
        else
            if the grain with label g(i) has at least two pixels then
                g'(i) := g(j)
            o'(i) := o(j)
            Decide on whether to accept (g', o') based on the Metropolis probability p_{g',o'}
```

pixels o_0 is left undefined). The exact way of their construction is discussed in Section 8.2.4. Whatever the case, no ambiguous pixels are expected to remain in g after the completion of the Metropolis algorithm.

Like before, iterations are carried on until the fulfillment of some termination criteria; see Section 8.2.6 for more details.

7.4 Similarity of Orientations

Based on the material presented in Sections 2.3 and 2.4, we begin by defining the orientation of (a pixel of) a grain as the (proper) rotation required to obtain the current placement of the crystalline lattice of the grain with respect to a hypothetical crystal aligned with the reference frame of the acquisition system. Specifically, rotations are to be given as unit quaternions interpreted according to Equation (2.29). The set $\mathcal{O}$ of crystalline orientations then becomes

$$\mathcal{O} := \mathbb{H}_1 \,. \tag{7.20}$$

Hence each pixel $o(i)$ of the orientation map will be assigned a 4-element real vector comprising the components of a unit quaternion. As mentioned earlier, mapping such objects to colors for sake of visualization is not trivial; this issue will be dealt with later in Section 8.2.1. (It should be also noted that this definition of $\mathcal{O}$ is not complete; the final one will be presented in Section 8.1.2, Equation (8.1).)

We now return to the important issue of how to define the distance $d\big(o(i), o(j)\big)$ between the orientations at pixels $o(i)$ and $o(j)$ of the orientation map o. Initially we ignore the underlying crystalline lattice, and discuss the **distance**, denoted by $r(\boldsymbol{q}_1, \boldsymbol{q}_2)$, between rotations $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

According to the heuristic stated in Section 2.4, such a distance between two rotations should be a monotonic function of the smallest nonnegative angle $\theta_{1\to2}$ such that the first rotation followed by a third rotation $(\mathbf{n}_{1\to2}, \theta_{1\to2})$, for some unit vector $\mathbf{n}_{1\to2}$, will result in the second rotation. If the first rotation is represented by the unit quaternion $\boldsymbol{q}_1$, and the second by the unit quaternion $\boldsymbol{q}_2$, then a unit quaternion $\boldsymbol{q}_{1\to2}$ that corresponds to the desired third rotation is

$$\boldsymbol{q}_{1\to2} := \boldsymbol{q}_2 \bar{\boldsymbol{q}}_1 \,, \tag{7.21}$$

since

$$\boldsymbol{q}_{1\to 2}\boldsymbol{q}_1 = (\boldsymbol{q}_2\bar{\boldsymbol{q}}_1)\boldsymbol{q}_1 = \boldsymbol{q}_2(\bar{\boldsymbol{q}}_1\boldsymbol{q}_1) = \boldsymbol{q}_2(1,\boldsymbol{0}) = \boldsymbol{q}_2\,.$$

The $\boldsymbol{q}_{1\to 2} = (a_{1\to 2}, b_{1\to 2}, c_{1\to 2}, d_{1\to 2})$ defined in Equation (7.21) is called the **transition quaternion** expressing the rotation from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$ (see e. g. [82] or [125, Section 8.3]). (Analogously, the inverse transition quaternion $\bar{\boldsymbol{q}}_{1\to 2} = \boldsymbol{q}_{2\to 1} = \boldsymbol{q}_1\bar{\boldsymbol{q}}_2$ corresponds to the rotation from $\boldsymbol{q}_2$ to $\boldsymbol{q}_1$.) Since $\boldsymbol{q}_{1\to 2}$ and $-\boldsymbol{q}_{1\to 2}$ define the same rotation, the first component $a_{1\to 2}$ is nonnegative for at least one of them, and the corresponding smallest nonnegative $\theta_{1\to 2}$ must lie between 0 and π. This is the value that indicates the dissimilarity, and so

$$r(\boldsymbol{q}_1, \boldsymbol{q}_2) \coloneqq 1 - |a_{1\to 2}| \tag{7.22}$$

can be used as the definition of distance.

Looking at the first row of Equation (2.24) for the product $\boldsymbol{q}_2\bar{\boldsymbol{q}}_1$, we see that

$$a_{1\to 2} = a_1a_2 + b_1b_2 + c_1c_2 + d_1d_2\,.$$

It is clear that $a_{1\to 2}$ is nothing but the dot product of $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ when they are considered as 4D unit vectors. In particular, the $|a_{1\to 2}|$, and hence $r(\boldsymbol{q}_1, \boldsymbol{q}_2)$, does not depend on the choice of the unit quaternions used to represent the first two rotations: if one used the alternative representations $-\boldsymbol{q}_1$ and $-\boldsymbol{q}_2$ for one or both of these rotations, then the value of $|a_{1\to 2}|$ would remain the same. Moreover, taking the conjugates (i. e. inverses) of both rotations at the same time also leaves $|a_{1\to 2}|$ unchanged. It can be verified that $r(\boldsymbol{q}_1, \boldsymbol{q}_2) = 0$ if and only if $\boldsymbol{q}_1 = \pm\boldsymbol{q}_2$, and that $r(\boldsymbol{q}_1, \boldsymbol{q}_2) = r(\boldsymbol{q}_2, \boldsymbol{q}_1)$ (in other words, we have the desirable property that the distance from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$ is the same as the distance from $\boldsymbol{q}_2$ to $\boldsymbol{q}_1$). Also, it is the case that

$$r((1,\boldsymbol{0}), \boldsymbol{q}_2\bar{\boldsymbol{q}}_1) = r(\boldsymbol{q}_1, \boldsymbol{q}_2) = r(\bar{\boldsymbol{q}}_1, \bar{\boldsymbol{q}}_2) = r((1,\boldsymbol{0}), \bar{\boldsymbol{q}}_2\boldsymbol{q}_1) \tag{7.23}$$

(here we made use of the fact that the conjugate of the conjugate of a quaternion is the quaternion itself).

Now we return to the issue raised in Section 7.1, namely that orientations are associated with a crystal structure, and hence the rotational crystal symmetries have to be taken into account. To illustrate this problem, let $\boldsymbol{q}$ denote a fixed unit quaternion describing an arbitrary rotation, and $\boldsymbol{s}$ be a unit quaternion describing a symmetry rotation of the crystal lattice (i. e., a rotation that maps every lattice point onto a lattice point). It is easy to see that the rotation $\boldsymbol{qs}$ results in the same orientation of the lattice as the rotation $\boldsymbol{q}$. Thus $\boldsymbol{q}$ and $\boldsymbol{qs}$ are said to be **crystallographically equivalent**. Indeed, the study of relative orientations in the presence of crystal symmetries has been an active field of research in the analysis of the texture of materials for over 30 years [76, 77, 80, 82, 107, 147].

We are now ready to define the distance $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$ between two orientations of the lattice brought about by rotations represented by unit quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. If the quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ correspond to the orientations of two neighboring grains, or of two neighboring pixels $o(i_1)$ and $o(i_2)$ in the orientation map, then $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$ should measure the size of the disorientation between the grains / pixels, taking into consideration the underlying crystal symmetries. Following the methodology presented in [76, 77, 82], this can be done as follows. Let $\mathcal{S}$ denote the set of all (proper) symmetry rotations of the lattice (i. e. the crystallographic point group). (The exact setting of $\mathcal{S}$ used in experiments will be presented in Section 8.2.3.) Then, by definition,

$$d(\boldsymbol{q}_1, \boldsymbol{q}_2) \coloneqq \min_{\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathcal{S}} r(\boldsymbol{q}_1\boldsymbol{s}_1, \boldsymbol{q}_2\boldsymbol{s}_2)\,, \tag{7.24}$$

that is, the smallest distance between a quaternion that is crystallographically equivalent to $\boldsymbol{q}_1$ and another quaternion that is crystallographically equivalent to $\boldsymbol{q}_2$.

We are now going to show that in fact

$$d(\boldsymbol{q}_1, \boldsymbol{q}_2) = \min_{\boldsymbol{s}\in\mathcal{S}} r\big((1,\boldsymbol{0}), \boldsymbol{s}\boldsymbol{q}_2\bar{\boldsymbol{q}}_1\big) = \min_{\boldsymbol{s}\in\mathcal{S}} r\big((1,\boldsymbol{0}), \boldsymbol{s}\bar{\boldsymbol{q}}_2\boldsymbol{q}_1\big)\,. \tag{7.25}$$

We first recall that $r(\boldsymbol{q}_1\boldsymbol{s}_1, \boldsymbol{q}_2\boldsymbol{s}_2) = r((1,\boldsymbol{0}), \bar{\boldsymbol{s}}_2\bar{\boldsymbol{q}}_2\boldsymbol{q}_1\boldsymbol{s}_1)$, see Equation (7.23). The definition of the latter needs the first component of

$$\bar{\boldsymbol{s}}_2\bar{\boldsymbol{q}}_2\boldsymbol{q}_1\boldsymbol{s}_1 = (\bar{\boldsymbol{s}}_1\boldsymbol{s}_1)\bar{\boldsymbol{s}}_2\bar{\boldsymbol{q}}_2\boldsymbol{q}_1\boldsymbol{s}_1 = \bar{\boldsymbol{s}}_1(\boldsymbol{s}_1\bar{\boldsymbol{s}}_2\bar{\boldsymbol{q}}_2\boldsymbol{q}_1)\boldsymbol{s}_1\,.$$

Recalling Equation (2.28), we see that the first component of this product is the same as that of $\boldsymbol{s}_1\bar{\boldsymbol{s}}_2\bar{\boldsymbol{q}}_2\boldsymbol{q}_1$. Since $(1,\boldsymbol{0})$ is a crystal symmetry, the conjugate of a crystal symmetry is a crystal symmetry, and obviously the product of two crystal symmetries is also a crystal symmetry, the claim in Equation (7.25) follows. Therefore, computing $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$ requires finding the minimum of $r\big((1,\boldsymbol{0}), \boldsymbol{s}\boldsymbol{q}_2\bar{\boldsymbol{q}}_1\big)$ among $|\mathcal{S}|$ crystallographically equivalent quaternions. Possibilities for further speedups will be discussed in Section 8.1.2.

Since the components of a unit quaternion have a magnitude less than or equal to 1,

$$\begin{aligned} 0 \leq r(\boldsymbol{q}_1, \boldsymbol{q}_2) \leq 1\,,\\ 0 \leq d(\boldsymbol{q}_1, \boldsymbol{q}_2) \leq 1\,, \end{aligned}$$

where the smallest and largest distances correspond to values 0 and 1, respectively. The smallest distance is measured when $\boldsymbol{q}_1 = \pm\boldsymbol{q}_2$ (for $r(\boldsymbol{q}_1, \boldsymbol{q}_2)$) or $\boldsymbol{q}_1 = \pm\boldsymbol{s}\boldsymbol{q}_2$ (for $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$). On the other hand, while the largest distance of $r(\boldsymbol{q}_1, \boldsymbol{q}_2) = 1$ is encountered for $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ being orthogonal as 4D vectors (i. e. associated with a transition rotation by π), the maximal value of $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$ encountered in practice depends on $\mathcal{S}$ and can actually be smaller than 1.

The valid range of δ and Δ defined earlier (see Equations (7.5) and (7.7)) also matches the range of $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$. For convenience, these parameters can be also expressed in terms of the **disorientation angle** $\theta_{1\to2}$ corresponding to $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$ as per Equation (7.22):

$$\left|\cos\left(\tfrac{1}{2}\theta_{1\to2}\right)\right| = 1 - d(\boldsymbol{q}_1, \boldsymbol{q}_2)\,. \tag{7.26}$$

Using the previously stated assumption $\theta_{1\to2} \in [0, \pi]$, the absolute value can be dropped around the cosine. For instance, when $\mathcal{S}$ contains the (proper) symmetry rotations of the simple cubic lattice introduced earlier, the maximal disorientation angle $\theta_{\max}$ can be derived as [76,77,82,145]

$$\tan\left(\tfrac{1}{2}\theta_{\max}\right) := \sqrt{23 - 16\sqrt{2}}\,, \tag{7.27}$$

yielding $\theta_{\max} \approx 62.7994°$ and corresponding to the maximal orientation distance $d_{\max} \approx 0.1465$.

7.5 Summary

Two related stochastic DT reconstruction techniques have been presented. The first one is very modest in terms of the amount of prior information required, and it can deliver an orientation map that most accurately matches the input set of projections. The second approach builds upon the aforementioned foundations, extending it with the ability to produce a grain map and an orientation map simultaneously, at the price of needing further *a priori* data in the form of grain statistics, approximate locations and basic orientations.

In both cases, orientations are represented using unit quaternions, for this representation scheme of rotations has been found to most suit our needs. In particular, by building on some basic properties of quaternions, we were able to devise a means for expressing the similarity of orientations in the presence of crystal symmetries.

Chapter 8

Simulations: Reconstruction of Deformed Polycrystalline Samples

Numerous simulations have been performed to optimize the free parameters of the algorithms and to quantitatively characterize the quality of the reconstructions as functions of magnitude of orientation spread within grains, those of degree of morphological complexity of grain maps, and of the artificial noise.

The results presented herein have been published in [16, 165, 166].

8.1 Implementation Details

8.1.1 Applications, Source Codes and Development Environment

The algorithms described in Chapter 7 have been implemented as a software package written in ANSI C, accompanied by a bunch of supporting Unix shell scripts. The following listing describes each application and its purpose:

diffrec Entry point to the reconstruction algorithms for obtaining an orientation map o or grain map / orientation map pair (g, o) from a set of diffraction patterns. Allows the tuning of all reconstruction parameters, as well as the measurement of various statistics regarding the quality of results.

gmaptool Allows the manipulation of grain maps (introduction of ambiguous pixels), construction of grain maps from orientation maps, construction of new orientation maps based on a subset of grains (using unchanged or homogeneous orientations within grains), calculation of grain statistics etc.

omapconv Provides convenience functions for manipulating and converting orientation maps. Supports some basic transformations (cropping, quantization and canonization of orientations), generation functions (constant or random orientation map, crystallographic equivalents, add noise to orientation map pixels) and conversions between representation formats (Euler angles, rotation matrices, unit quaternions).

projcalc Allows the generation of simulated projections (i. e. diffraction patterns), application of artificial noise, and calculation of diffraction pattern statistics.

Each application has only a command-line interface, expecting inputs and producing outputs via files. The sole exception is **diffrec** that is able to show a graphical progress of the orientation map; this feature uses the C++ based cross-platform Qt library (specifically, Qt/X11 v3.3.2).

Some source code statistics: 50 files, ≈ 736 KB, ≈ 19 300 total lines of code.

All the timings mentioned later were measured in the same hardware and software environment described in Section 5.1.1.

8.1.2 Speeding Up the Reconstruction Process

As mentioned in Section 5.1.2, performance optimizations are very beneficial—and sometimes essential—for reconstruction algorithms based on a stochastic method. For reference purposes, some of these enhancements are summarized below.

Look-up Table for Binary Local Grain Map Configurations

Several look-up tables are utilized throughout the program to store pre-computed values of certain functions whose evaluation should be as fast as possible; this approach had been already used elsewhere with much success [186]. For instance, the potentials associated with any 3×3 binarized grain map configurations (i. e. the $\mathcal{C}_1, \ldots, \mathcal{C}_6$ required for the computation of $H_2(g)$) are stored in a 9D look-up table, as motivated by [15]. Another example is given below.

Look-up Table for Orientation Distances

Though it would be possible to determine the value of $r(\cdot, \cdot)$ in Equation (7.25) for each $\boldsymbol{s} \in \mathcal{S}$ one-by-one, and then choosing the smallest, this would really slow down the reconstruction procedure. (For example, $\mathcal{S}$ has 24 elements for cubic lattices.) A more efficient method is to use a **look-up table** of the values of $\min_{\boldsymbol{s}\in\mathcal{S}} r\big((1,\boldsymbol{0}), \boldsymbol{sq}\big)$, based on a **quantization** (discretization) of the unit quaternion $\boldsymbol{q}$. We now show that there is a fairly easy way to implement such a look-up table.

First, it should be recalled that the components of a unit quaternion have a magnitude less than or equal to 1. For any rotation, the first component a of $\boldsymbol{q}$ can be chosen to be nonnegative. Since a is uniquely determined by the other components, it can be ignored for the construction of the look-up table. For the sake of simplicity, let us suppose that the whole range $[-1, 1]$ of the other three components (b, c, d) is sampled using Q values ($Q \in \mathbb{Z}^+$ odd). This way we get a table T of Q^3 values, such that, at those locations for which $b^2 + c^2 + d^2 \leq 1$, the element t_{bcd} is the needed value for the unit quaternion $\boldsymbol{q} = (a, b, c, d)$ for the corresponding nonnegative a. If $b^2 + c^2 + d^2 > 1$, the value of t_{bcd} is undefined. This convention makes the table easily addressable, at the cost of some wasted memory. The sampling resolution Q may simply be chosen based on memory considerations. For example, having $Q := 101$ and using 8 bytes to represent real numbers, T consumes $8 \cdot 101^3$ bytes (about 7.86 MB). (See Table 8.1 for a statistics on T versus various Q settings.)

To find the desired $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$, calculate $\boldsymbol{q}_{1\to2} = (a_{1\to2}, b_{1\to2}, c_{1\to2}, d_{1\to2})$ according to Equation (7.21). If $a_{1\to2}$ is negative, replace $\boldsymbol{q}_{1\to2}$ with $-\boldsymbol{q}_{1\to2}$. Then $d(\boldsymbol{q}_1, \boldsymbol{q}_2)$ is approximated by $t_{b'_{1\to2}c'_{1\to2}d'_{1\to2}}$, where $b'_{1\to2}$, $c'_{1\to2}$, and $d'_{1\to2}$ are the sampling points nearest to $b_{1\to2}$, $c_{1\to2}$, and $d_{1\to2}$, respectively.

Quantization of Orientations

As a matter of fact, discretizing unit quaternions as discussed above helps not only in the fast computation of disorientations but is also beneficial for reducing the search space of orientations

Table 8.1. Effects of the sampling resolution Q on the cardinality of $\mathbb{H}_{1c}^{Q}$ and on the look-up table T for $d(\cdot,\cdot)$. For the latter, 8-byte `double` elements are assumed.

		T		T & $\mathbb{H}_{1c}^{Q}$
Q	Δ_Q	Storage size (MB)	Total elements	Valid elements
101	0.02	≈ 7.86	1 030 301	523 249 ($\approx 50.79\,\%$)
201	0.01	≈ 61.96	8 120 601	4 187 801 ($\approx 51.57\,\%$)
401	0.005	≈ 491.95	64 481 201	33 507 829 ($\approx 51.97\,\%$)

during the optimization of $\gamma(o)$ or $\gamma(g,o)$. (This is not only an issue of memory requirements or reconstruction speed. Specifically, it is a fundamental property of 3DXRD that the obtainable spatial resolution—with respect to grain position inside the specimen—and the angular resolution—related to the precision with which orientations can be measured—of detectors are not independent. See [153] for a deeper discussion.) In addition, the usage of unit quaternions in the canonical form allows for a unique representation for all possible orientations (disregarding crystallographic equivalence). Letting $\mathbb{H}_{1c}^{Q}$ denote the set of quantized unit quaternions in the canonical form for some sampling resolution Q, the final set $\mathcal{O}$ of crystalline orientations then becomes

$$\mathcal{O} \coloneqq \mathbb{H}_{1c}^{Q}\,. \tag{8.1}$$

Clearly, $\mathcal{O}$ is known and finite, hence placing the reconstruction problems raised in Chapter 7 in the realms of DT. Specifically, $|\mathcal{O}| \propto Q^3$, but the actual cardinality is always less than Q^3 because about half of the combinations (b,c,d) are invalid (viz. when $b^2+c^2+d^2>1$). (For instance, having $Q \coloneqq 101$ yields $Q^3 = 1\,030\,301$, of which only 523 249 [$\approx 50.79\,\%$] combinations result in valid unit quaternions. Table 8.1 gathers similar statistics for some additional choices of Q.)

When orientations are represented by such quantized unit quaternions $\boldsymbol{q} = (a,b,c,d) \in \mathbb{H}_{1c}^{Q}$ for some Q, one can speak of the set $\mathcal{N}_Q(\boldsymbol{q})$ of **neighbors** of $\boldsymbol{q}$:

$$\mathcal{N}_Q(\boldsymbol{q}) \coloneqq \left\{ \begin{array}{l} (a_1, b-\Delta_Q, c, d), \\ (a_2, b+\Delta_Q, c, d), \\ (a_3, b, c-\Delta_Q, d), \\ (a_4, b, c+\Delta_Q, d), \\ (a_5, b, c, d-\Delta_Q), \\ (a_6, b, c, d+\Delta_Q) \end{array} \right\}, \tag{8.2}$$

where Δ_Q denotes the **sampling unit** of quaternion components (b,c,d),

$$\Delta_Q \coloneqq \frac{2}{Q-1}\,. \tag{8.3}$$

(For example, having $Q \coloneqq 101$ results in a sampling unit of $\Delta_Q = 0.02$.) Moreover, the a_i $(1 \le i \le 6)$ in Equation (8.2) shall be determined so that the resulting quaternion stays in $\mathbb{H}_{1c}^{Q}$. If no such a_i exists, the associated (a_i, b_i, c_i, d_i) will be discarded. (Therefore, $1 \le |\mathcal{N}_Q(\boldsymbol{q})| \le 6$ for any Q and $\boldsymbol{q}$.) With these definitions in place, we can now formulate the set $\mathcal{N}\big(o(i)\big)$ of neighbors of orientation $o(i)$:

$$\mathcal{N}\big(o(i)\big) \coloneqq \mathcal{N}_Q(\boldsymbol{q}) \tag{8.4}$$

for some Q and $\boldsymbol{q} \in \mathbb{H}_{1c}^{Q}$ representing $o(i)$.

Reduction of Crystallographically Equivalent Orientations

Given that crystallographically equivalent orientations lead to the same arrangement of lattice points, it seems natural to partition $\mathcal{O}$ into equivalence classes by the equivalence of orientations, then pick one representative element from each class and use the reduced collection $\mathcal{O}'$ of such elements instead of $\mathcal{O}$. When the representative elements $(\mathbf{n}, \theta)$ are chosen so that θ is minimal and $\mathbf{n}$ falls within a certain subset of the 3D unit sphere, the resulting $\mathcal{O}'$ is known as the **fundamental zone** in materials science (see [60] and [76, 77] for derivations using Rodrigues vectors and quaternions, respectively). Though it would be tempting—and indeed valuable in further reducing the search space of orientations—to restrict the range of an orientation map o to $\mathcal{O}'$, this has *not* been actually employed by the author due to the inherent complications involved regarding the computation of orientation distances $d\big(o(i), o(j)\big)$ and the notion of "neighboring orientations" $\mathcal{N}\big(o(i)\big)$ within $\mathcal{O}'$.

Quick Evaluation of the Objective Function

The objective function $\gamma(g, o)$ needs to be completely evaluated only once, right before optimization is started. Whenever a pair (g', o') of maps is to be tested for acceptance—as per $p_{g',o'}$—as the new approximation of the optimal maps (with (g, o) being the current pair of maps), $\gamma(g', o')$ is computed by updating $\gamma(g, o)$ according to the changes caused by switching from (g, o) to (g', o'). In particular, terms $H_1(g, o)$, $H_1(g', o')$, $H_2(g)$, and $H_2(g')$ are affected by only those 3×3 cliques that contain the pixel i being modified. (There are only 9 such cliques.) Furthermore, the simulated projections $\mathbf{P}_{o'}$, and thus the error $\|\mathbf{P}_{o'} - \mathbf{P}\|_1$, can be directly derived from $\mathbf{P}_o$ by subtracting the projections on the detector generated by the old orientation $o(i)$ and then by adding those obtained with the new orientation $o'(i)$. A similar trick is employed to speed up the computation of $\gamma(o)$.

8.2 Simulation Setup

8.2.1 Visualization of Orientations

In order to be able to visually inspect 2D orientation maps, a mapping of orientations to colors is to be defined, so that o can be displayed as a regular color image. As explained in [7,162,193,194], however, there is no such single color mapping that would fit all applications and purposes. In particular, it is exceedingly hard to establish a color mapping that associates "similar" (i. e. close, with respect to $d(\cdot, \cdot)$) orientations with similar colors. (This is especially so in the presence of crystal symmetries.)

For these reasons, we have decided to use the following simple mapping: Components (b, c, d) of a unit quaternion $\boldsymbol{q} = (a, b, c, d)$ are assigned the color

$$(\text{red}, \text{green}, \text{blue}) \coloneqq I_{\max} \left(1 - |b|, 1 - |c|, 1 - |d|\right) ,$$

where $I_{\max}$ designates the maximal intensity of any color primary. (As discussed earlier, a is uniquely determined by the other components when using a canonical representation, hence it can be ignored for sake of visualization.)

It is easy to see that $\boldsymbol{q}$ and $-\boldsymbol{q}$ are assigned the same color, which is desirable. A less welcome consequence is that $\bar{\boldsymbol{q}}$ and $-\bar{\boldsymbol{q}}$ are also represented with this color. As a matter of fact, the 16 quaternions $(\pm a, \pm b, \pm c, \pm d)$ are all mapped to a common color—admittedly, this is an unfortunate side-effect. A major shortcoming of this approach is its complete disregard of crystal symmetries; equivalent orientations will thus be rendered with different colors. Moreover,

such a linear mapping cannot possibly express the "similarity" of orientations via the choice of similar colors. (After all, the RGB color space is unsuitable for measuring distances. The latter aim could be better achieved by employing a more sophisticated color space like CIE 1976 (L^*, a^*, b^*).)

An interesting property of the aforementioned mapping is that it never produces the following colors: black, pure red, pure green, pure blue. On the other hand, the special unit quaternions $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, and $(0, 0, 0, 1)$ are mapped to white, cyan, magenta, and yellow, respectively. The darkest shade of gray is associated with $b = c = d = \pm\frac{1}{\sqrt{3}}$ (implying $a = 0$).

8.2.2 Software Phantoms

Four 64×64 pixel test maps of deformed monophase aluminum samples of varying complexity were used. In all cases the **reference orientation map** was generated by **electron backscattering diffraction** (EBSD) [6, 110]—a technique utilizing a **scanning electron microscope** (SEM) to determine the distribution of grain orientations on a surface section of the specimen.[1] These maps are depicted in Figure 8.1 using the color-mapping method described in Section 8.2.1. (In order to make the orientation spreads within grains more visible, a contrast enhanced / "equalized" variant of the maps is also shown, with as well as without edges overlaid.) They represent different complexities in terms of the number of grains and of the orientation spread inside grains:

Case I (series "G015-a2") This map comprises 11 grains; on average the orientation spread—as per Equation (7.13)—within grains is 0.0003427 (i. e., equivalent to the disorientation angle of $\approx 3°$; see Equation (7.26)), while one grain has a spread of 0.0018652 ($\approx 7°$).

Case II (series "g3_15") This map comprises 26 grains; the average orientation spread is around 0.0018652 ($\approx 7°$), but the spread within grains varies from 0.0003427 up to 0.0183728 ($\approx 3°$–$22°$). The subdivision of some grains into sub-grains is also noticeable.

Case III (series "G01") This map comprises 3 grains; the average orientation spread is 0.0074538 ($\approx 14°$).

Case IV (series "Na4-5cX") The material is, in this case, too deformed to comply with being "moderately deformed" as defined in Section 7.3.1. Grains cannot be identified in an unambiguous way and the orientation spread sometimes surpasses 0.0109841 ($\approx 17°$). This case is included to test the limitations of the algorithms. (It should be remarked that this map apparently contains some grains that seem to be 8-connected but not 4-connected; this is in clear contradiction with our earlier assumptions.)

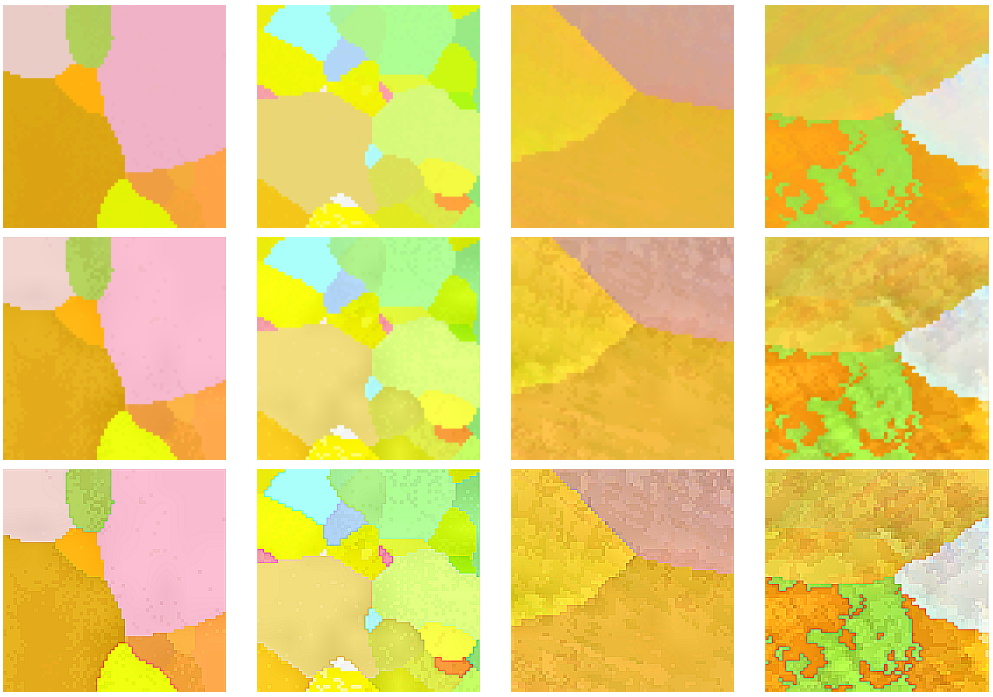
Hence, all data sets but Case IV represent orientation maps that were subject to *moderate* levels of deformation. The exact details of the deformation process are summarized in Table 8.2.

In order to assess the applicability of the algorithms to *undeformed* samples too, the 128×128 pixel map of a monophase aluminum specimen examined in [14–16] has been taken.[2] Its reference orientation map has, again, been generated by EBSD and is shown in Figure 8.2. This map exhibits the following properties:

Case V (series "abstrM1") This map comprises 44 grains; the orientation spread is invariably 0 (0°). This map is special because 5 986 out of 16 384 ($\approx 36.54\,\%$) pixels are void. (The exact way of handling of these pixels is described in Section 8.2.3.)

[1] All EBSD maps were provided by Jacob R. Bowen (Risø).

[2] The usage of Test Case V has not been published; this map was investigated as an extension and improvement upon the results published earlier.

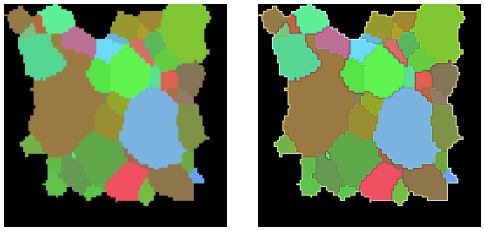


(a) (b) (c) (d)

Figure 8.1. Deformed reference orientation maps. Top row: original color mapping. Middle row: contrast-enhanced map. Bottom row: like middle row with edges overlaid. **(a)** Test Case I. **(b)** Test Case II. **(c)** Test Case III. **(d)** Test Case IV.

Table 8.2. Technical attributes of deformed reference orientation maps. $\varepsilon_{\mathrm{vm}}$ denotes the strain (a normalized measure of the deformation), RT stands for room temperature, ECAE is a shorthand for equal channel angular extrusion (a kind of deformation procedure).

Test map	Deformation	Comment
Case I	15 pass ECAE at RT, $\varepsilon_{\mathrm{vm}} = 10$, annealed for 1h at 400 °C	a little mosaic spread, clear grain boundaries
Case II	15 pass ECAE at 300 °C, $\varepsilon_{\mathrm{vm}} = 10$, as deformed	more grains, a little more spread, yet still clear grain boundaries
Case III	1 pass ECAE at RT, $\varepsilon_{\mathrm{vm}} = 0.67$ ($\approx 50\,\%$), as deformed	few grains, much spread within grains
Case IV	40 % cold rolled	few grains, much spread within grains



(a) (b)

Figure 8.2. Undeformed reference orientation map of Test Case V. **(a)** Original color mapping. Black pixels represent void regions. **(b)** Like (a) with edges overlaid.

For each case, first a **reference grain map** was determined from the reference orientation map via a connected component technique [29, 177, 181]. This procedure used $d\big(o(i), o(j)\big)$ as a similarity / distance measure in tandem with a fixed misorientation threshold, establishing a grain boundary whenever the orientation distance of the pixel under consideration from its already visited neighbors exceeded the misorientation threshold. After some experimentation, the following thresholds were used for Test Cases I–IV: 0.00095 ($\approx 5°$), 0.0053 ($\approx 11.8°$), 0.0038 ($\approx 10°$), 0.0024 ($\approx 8°$). For Test Case V, the threshold 0.000038 ($\approx 1°$) was chosen.

The next step consisted of the determination of the center-of-mass pixel and the basic orientation of each of the grains. These were later used to initialize (g_0, o_0) before the optimization process as described in Section 8.2.4. These grain features, together with the number G of grains, would constitute the minimal outcome of an external method (e. g., GRAINSWEEPER [153]) to be used to retrieve such information by the pre-processing of 3DXRD diffraction patterns.

8.2.3 Computation of Simulated Diffraction Patterns

The usage of simulated diffraction patterns instead of relying on real—i. e. physically measured—ones is due to several reasons. First, physical measurements are usually accompanied by certain issues whose handling during reconstruction can be pretty complex (e. g. presence of void sample regions or distortions caused by imperfect instrument geometry; see Section 3.2.3).

Second and most important, it is very complicated to establish a "ground truth", i. e. validate the reconstruction result via some independent method. At the time of research, the only alternative to 3DXRD was the usage of surface scans with SEM. 3DXRD cannot be employed near specimen boundaries (e. g. to illuminate the topmost layer of the sample), whereas SEM can only inspect grain orientations on the specimen surface. If 3DXRD were to be used on some lower layer of the specimen, the sample would have to be polished to get rid of upper layers in order to bring the layer of interest to the surface, so that it becomes accessible to SEM. Unfortunately, polishing itself causes serious deformations to the specimen, essentially "destroying" the original orientation map that was examined earlier with 3DXRD.[3]

Acquisition Set-up

For every test case, 3DXRD diffraction patterns $\mathbf{P}$ associated with the reference orientation maps were simulated[4] (see Equation (C.5)), optionally applying some synthetic noise as well. We

[3] It should be noted that some more sophisticated polishing techniques have appeared since the time of research; these do not introduce any significant deformations to the sample.

[4] The original diffraction core subroutine is courtesy of Erik Bergbäck Knudsen (Risø). Additional improvements have been made by the author and Arun K. Kulshreshth (CUNY).

generated 91 projections corresponding to equally spaced rotation angles ω_r from the interval $[-45°, 45°]$. The acquisition set-up imitated that of the 3DXRD microscope at the ESRF at the time (see Section 3.2.2). Moreover, computations were carried out in a somewhat idealized fashion. In particular, the following assumptions had been taken:

- Instrument geometry: Measurements are performed at a single sample–detector distance setting with $L = 4.186\,\text{mm}$ to the center of the sample. The X-ray beam is monochromatic at 50 keV, uses parallel geometry, and has negligible divergence and energy spread. The detector has a resolution of 1024×1536 pixels, pixel size $2.3 \times 2.3\,\mu\text{m}^2$, and field-of-view $2.3552 \times 3.5328\,\text{mm}^2$. There is no detector tilting or sample precession during rotation.

- Sample geometry: The sample area consists of 64×64 (Cases I–IV) or 128×128 (Case V) pixels, with pixel size $2.3 \times 2.3\,\mu\text{m}^2$, and occupies dimensions $147.2 \times 147.2\,\mu\text{m}^2$ or $294.4 \times 294.4\,\mu\text{m}^2$ (Cases I–IV versus Case V). For Test Case V, void regions are known beforehand exactly[5]. The vertical direction in the maps shown were all set to be perpendicular to the incoming beam for $\omega = 0°$.

- Sample composition: The sample is in monophase (made of aluminum), moderately deformed (Cases I–IV) or undeformed (Case V), and has a minimal associated diffraction spot overlapping.

- Imaging system: All orientation maps—including reference data sets[6]—contain orientations represented using elements of $\mathbb{H}_{1c}^{Q}$ with $Q := 101$. (This setting of Q was chosen for speed and memory reasons.) A sample pixel always contributes the same orientation, no matter how large fraction of the projection line passes through it. A reflection event is realized as the deposition of a single photon impact into a single detector pixel.[7] The projection error term $\|\mathbf{P}_o - \mathbf{P}\|_1$ is computed, for any ω_r, by comparing corresponding pixels in $\mathbf{P}_o$ and $\mathbf{P}$.[8]

The crystal structure of aluminum can be described with a **face-centered cubic** (FCC) lattice—crystallographic space group 225, to be exact—as shown in Figure 8.3(a). This lattice is part of the more general family of **cubic crystal lattices** (the other family members being the simple cubic and the body-centered cubic lattices). These lattices have the same crystal symmetries $\mathcal{S}$: there are 48 symmetry operations, out of which 24 $(= 1 + 3 \cdot (4 - 1) + 6 \cdot (2 - 1) + 4 \cdot (3 - 1)$; see Figure 8.3(b)) are pure rotations and the rest are rotoinversions. (As discussed earlier, only proper rotations are relevant for describing crystalline orientations.) So, for every rotation $\boldsymbol{q}$, there are 23 other rotations that are crystallographically equivalent with it. Many important metals have a cubic lattice. For instance, iron has a body-centered cubic lattice, while gold, silver, copper, and lead all have a face-centered one (besides aluminum).

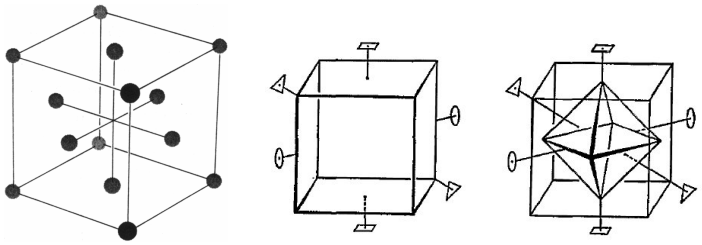
Diffraction patterns were computed by taking into account the following family of reflections only (a total of 58 Miller indexes): {111}, {200}, {220}, {222}, and {311}. These correspond

[5] All orientation maps o—reference, initial, intermediate, reconstructed—are permitted to contain void pixels (represented with the zero quaternion $(0, \mathbf{0})$ and a special grain label); they will be simply ignored during the computation of $\mathbf{P}_o$, $\gamma(o)$ and $\gamma(g, o)$. However, their exact position needs to be fixed; no void pixels will be removed or introduced during reconstruction.

[6] The only reason for quantizing orientations in the reference map is to be able to reduce the projection error term $\|\mathbf{P}_o - \mathbf{P}\|_1$ to zero. This condition will not, of course, hold for physical measurements, thus manifesting as a nonzero projection error.

[7] Real diffraction measurements are expected to be better describable by a more complex point spread function (PSF).

[8] Admittedly, this naive approach is not perfect. A small change in orientation $o(i)$ of pixel i may result in a "leap" in ω, i. e. a diffraction spot previously appearing at ω_r may "jump over" to ω_{r-1} or ω_{r+1}. This will certainly yield a higher projection error, even though the orientation difference in $o(i)$ can be tiny.



(a) (b)

Figure 8.3. Crystal structure of aluminum. **(a)** Face-centered cubic lattice. (Image source: [21].) **(b)** Rotational symmetries of a cube and a regular octahedron. Proper rotation operations: identity, 3 fourfold axes, 6 twofold axes, 4 threefold axes. (Image source: [58].)

to diffracted directions having the smallest 2θ value (see Section 3.2.2), thus being the most prominent ones in practice. (The contribution of other Miller indexes with a greater 2θ angle is negligible.)

Modeling Noise

In order to come close to the quality of real diffraction patterns, varying levels of artificial **noise** were applied to the simulated projections. The distortions of measurements are caused by different kinds of phenomena: scattering, fluctuation of the intensity of the X-ray beam, statistical error (the so-called Poisson or quantum noise), cross-talk between neighboring detector pixels, etc. Out of these, perhaps the Poisson noise is the most prominent, therefore only this sort of deviation was taken into account. Let us suppose that the intensity (photon count) I_0 of every detector pixel is to be distorted by $N\,\%$ of noise ($N \geq 0$ real). The noisy intensity I_{noisy} was then, as an approximation to the Poisson distribution, defined as a uniformly distributed random number taken from $\left[I_0\left(1-\frac{N}{100}\right), I_0\left(1+\frac{N}{100}\right)\right]$, subject to the constraint of non-negativity. A trivial property of such a multiplicative noise is that $I_{\text{noisy}} = 0$ whenever $I_0 = 0$ (i. e. no noise gets introduced at background pixels); this can be contrasted with the additive noise model described in Section 5.2.3 for parametric objects and attenuation contrast transmission radiographs.

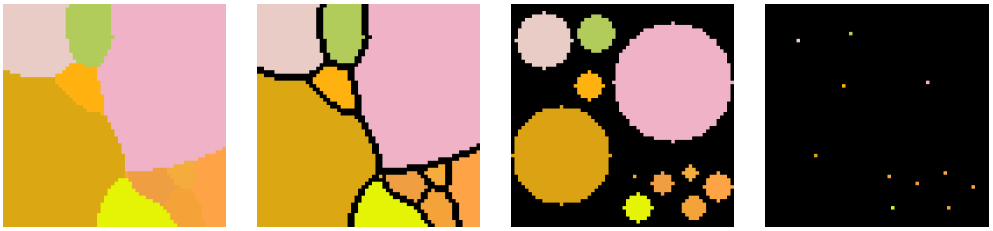
8.2.4 Initial Grain and Orientation Maps

Before the optimization of $\gamma(g, o)$ could be started, one first needs a suitable initial (g_0, o_0). The following options were considered:

1. Let g_0 be identical with the reference grain map. (Hence, there are no ambiguous pixels in g_0.) Moreover, construct o_0 so that all the pixels of a given grain be set to the basic orientation of the grain, thus arriving at a completely homogeneous map. (Basically, the o_0 acquired so will represent an undeformed orientation map.)

2. Start off from the outcome of option 1 and introduce some ambiguous pixels to g_0 in the vicinity of grain borders. This can be achieved, for instance, by applying binary morphological erosion to each grain of g_0.

3. Start with the outcome of option 1, then, for each $\ell \in \mathcal{G}$, keep the grain label at pixels i that fall within the "incircle" of g_ℓ and set the rest of g_ℓ in g_0 to ambiguous. (By "incircle" we mean the discrete circle of maximal radius about the grain centroid so that the circle completely lies within the grain.)



(a) (b) (c) (d)

Figure 8.4. Options for setting up initial orientation maps for $\gamma(g, o)$ (original color mapping). The initial grain map can be obtained by assigning unique labels to non-black connected homogeneous regions and setting black pixels to ambiguous. **(a)** Complete homogeneous orientation map. Each pixel of any grain is assigned the basic orientation. **(b)** Morphological erosion applied to (a). **(c)** Circles of maximal radii about the grain centroids of (a). **(d)** Grain centroids of (a).

4. Modify the output of option 1 so that grain labels are kept in g_0 only at pixels corresponding to grain centroids, and set the rest of g_0 to ambiguous. (These non-ambiguous pixels are dubbed **seed points**.)

These scenarios are demonstrated in Figure 8.4. Whatever the case, pixels of o_0 having the ambiguous label in g_0 will be left undefined. Of the aforementioned possibilities, option 4 seems to be the most realistic and feasible from a practical point of view, so this approach was used to obtain (g_0, o_0). (For Test Case IV, somewhat arbitrarily a set of 31 seeds were determined, corresponding to 31 "grains". However, these are not grains in the same physically meaningful sense as in the other three cases.)

In the case of $\gamma(o)$, the initial orientation map o_0 can be also obtained in multiple ways:

1. Homogeneous (i. e. undeformed) map, like option 1 above.
2. Random map.
3. Constant map, that is $o_0(i) \equiv \boldsymbol{q}_0$ $(\forall i \in \mathcal{D})$, where $\boldsymbol{q}_0 \in \mathbb{H}^Q_{1c}$ fixed.
4. Constant map with seed points, like option 4 above.

Reconstructions have been carried out employing all but the first of these possibilities. (Note: Option 4 is somewhat unrealistic since the availability of grain statistics is not expected; see Section 7.2.1 for a list of priors assumed for $\gamma(o)$.)

8.2.5 Measuring the Quality of Reconstructions

The quality of the results were measured by two **figure-of-merit** (FOM) functions, FOM_g and FOM_o.

$$\mathrm{FOM}_g := 1 - \frac{M}{|\mathcal{D}|}\,, \tag{8.5}$$

where M denotes the number of mismatching grain labels between corresponding pixels of the reference and reconstructed grain maps, and $|\mathcal{D}|$ is the number of pixels. (FOM_g is undefined for $\gamma(o)$, of course.) The second figure-of-merit is related to the distance between the original

orientation $o_{\mathrm{orig}}(i)$ and the resulting orientation $o_{\mathrm{rec}}(i)$ for each pixel i:

$$\mathrm{FOM}_o := 1 - \frac{1}{d_{\max}\,|\mathcal{D}|} \sum_{i \in \mathcal{D}} d\left(o_{\mathrm{orig}}(i), o_{\mathrm{rec}}(i)\right), \tag{8.6}$$

where, for $i \in \mathcal{D}$, $o_{\mathrm{orig}}(i)$ and $o_{\mathrm{rec}}(i)$ denote the orientations at pixel i in the original and in the reconstructed maps, and $d_{\max}$ represents the maximal possible value of $d(\cdot,\cdot)$, which equals about 0.1465 ($\approx 62.7994°$; see Equation (7.27)) for the FCC lattice of aluminum. It can be easily checked that both measures have the range $[0,1]$. Evidently $\mathrm{FOM}_g = \mathrm{FOM}_o = 1$ for perfectly reconstructed maps, while $\mathrm{FOM}_g \approx 0$ and $\mathrm{FOM}_o \approx 0.5$ for a random reconstruction. (These can be contrasted with the properties of the measures defined in Section 5.2.5 for parametric objects.)

8.2.6 Reconstruction Parameter Settings

The choice of the Gibbs potentials $\kappa_{k;g}$ ($0 \le k \le 6$) was motivated by [14, 15] (there denoted as U^1). Moreover, as recommended in [51], the ratios $\frac{1}{2}\kappa_3 = \kappa_2 = \kappa_1$ and $\lambda_2 = \lambda_{2;g} = \frac{1}{\sqrt{2}}\lambda_1 = \frac{1}{\sqrt{2}}\lambda_{1;g}$ were also respected. The value of $\kappa_1 = \kappa_2$ was chosen somewhat arbitrarily as 0.94. (As a matter of fact, this is the optimal estimated value of κ_1 established in [51] for a certain software phantom; it might not be optimal for our applications.)

The value of δ was always guessed from the reference orientation map and kept constant all the time. In particular, δ was set to 0.005 ($\approx 11.5°$), 0.01 ($\approx 16.2°$), 0.008 ($\approx 14.5°$), and 0.021 ($\approx 23.5°$) for Test Cases I–IV, respectively. For Test Case V, the setting $\delta = 0.000625$ ($\approx 4°$) was applied.

In the case of $\gamma(o)$, Δ was set equal to δ. (As a reminder, $\Delta \ge \delta$ shall hold in general.)

Based on the five test cases, associated input projections, and initial maps as defined above, a series of reconstructions were performed with varying system parameters α, β, $\lambda_1 = \lambda_{1;g}$, and number of Metropolis iterations, as well as different noise levels. The variations of the FOMs with respect to the aforementioned free parameters turned out to be small, implying that the algorithm is robust. For the following the values $\alpha = \beta = \lambda_1 = \lambda_{1;g} = 1$ have been employed. The final set of parameters is listed in Table 8.3.

The optimization process was terminated when the projection error term $\|\mathbf{P}_o - \mathbf{P}\|_1$ reached 0 or the number of Metropolis iterations exceeded a preset threshold, whichever occurred earlier.

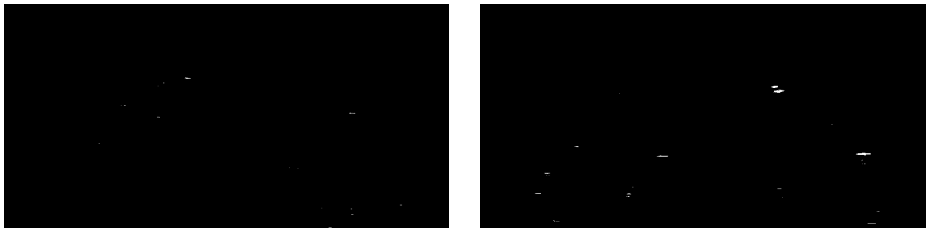
8.3 Results

Due to the properties of diffraction, projections are very sparse (i. e., they contain very few measurements); the fraction of detector pixels changed is around 0.01 % for Test Cases I–III, climbs to ≈ 0.017 % for Case V and reaches ≈ 0.024 % for Case IV. (This is clearly demonstrated by the simulated diffraction patterns depicted in Figure 8.5.) For each grain, the nature of the diffraction process implies that there is only a finite set of possible projections. The exact number depends on the experiment, but it may be as low as 5—pretty small a value as compared to the number 91 of projections taken. (For our choice of Miller indexes described in Section 8.2.3, any pixel i of o gave rise to ≈ 14 reflections on average.) Intensities (i. e. photon counts) of non-background detector pixels varied between 1–49 (Case V) down to 1–16 (Case IV), with average intensities of ≈ 6.2 (Case V) down to ≈ 1.7 (Case IV). This sparsity of data interferes with the aim of obtaining high quality grain maps.

The rest of the section is devoted to presenting the simulation results obtained using either objective function.

Table 8.3. Parameter settings for $\gamma(o)$ and $\gamma(g, o)$.

Parameter	Setting
$\kappa_1 = \kappa_2$	0.94
κ_3	1.88
$(\kappa_{0;g}, \kappa_{1;g}, \kappa_{2;g}, \kappa_{3;g}, \kappa_{4;g}, \kappa_{5;g}, \kappa_{6;g})$	$(0, 1.4, 0.71, 0.61, 0.79, 0.5, 0.61)$
$\lambda_1 = \lambda_{1;g}$	1
$\lambda_2 = \lambda_{2;g}$	$\frac{1}{\sqrt{2}}$
α	1
β	1
δ, Δ	varies for each Test Case



(a) (b)

Figure 8.5. Simulated diffraction patterns of Test Case II in the absence of noise. Intensities have been binarized for easier visibility; black pixels denote a photon count of zero, white ones have a positive photon count. Only the lower third of the total detector area is shown; the rest is completely black. The non-diffracted incident X-ray beam passes just below the lower side of the detector (not indicated). **(a)** Projection at $\omega = -43°$. **(b)** Projection at $\omega = 27°$.

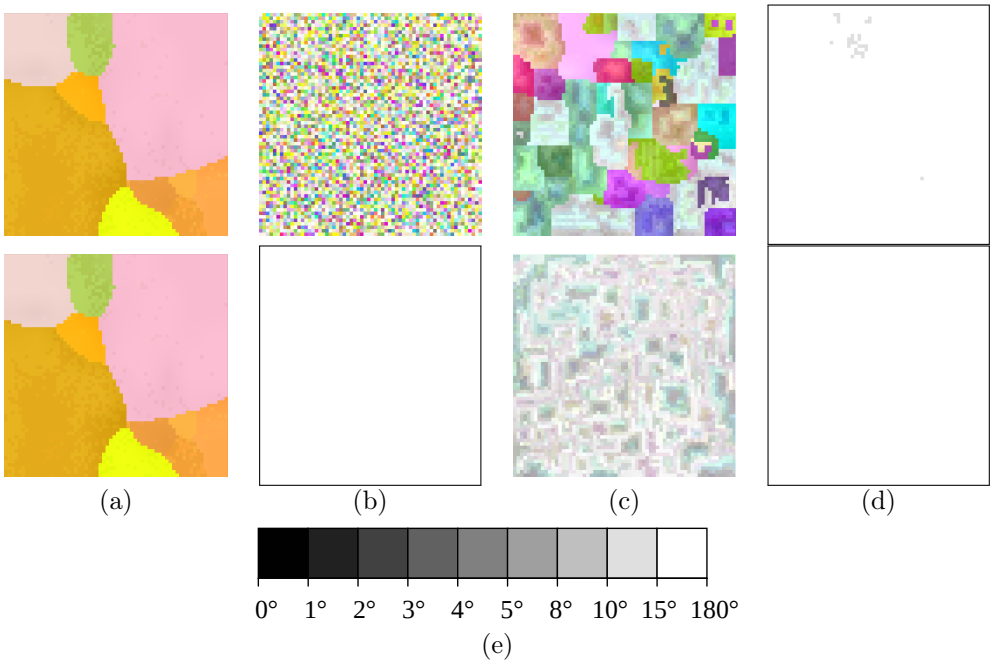


Figure 8.6. Failed reconstructions of Test Case I using $\gamma(o)$ and noiseless projections. Top row: random o_0; $\mathrm{FOM}_o = 0.528$. Bottom row: constant o_0 with $\boldsymbol{q}_0 \coloneqq (1, \mathbf{0})$; $\mathrm{FOM}_o = 0.501$. **(a)** Reference orientation map (contrast-enhanced). **(b)** Initial orientation map. (Frame added in bottom row for better visibility.) **(c)** Reconstructed orientation map (contrast-enhanced). **(d)** Difference of the reference and the reconstructed orientation maps. The intensity of the pixels is determined by the distance (disorientation angle) of corresponding orientation pairs, as shown in (e). (Frame added for better visibility.)

8.3.1 Reconstructions using $\gamma(o)$

Reconstructions[9] for the four deformed test cases were attempted using different initial orientation maps. In particular, every pixel of o_0 was generated either by picking a random element from $\mathcal{O}$ or by setting it to some fixed unit quaternion, thus resulting in a random or a constant map, respectively. Sadly, both approaches invariably produced unusable reconstructions; two such examples are shown in Figure 8.6 for Test Case I using ideal diffraction patterns. This is very disappointing, and it indicates that the algorithm got stuck in a local optimum.

As a workaround, executions were repeated by using an o_0 that was almost completely constant but also contained seed points placed at grain centroids, with each seed having been set to the basic orientation of the associated grain. With this change in place, the algorithm was finally able to yield very good results in at most 2.5 million iterations (corresponding to about 17 minutes of computer time), as demonstrated in Figure 8.7 and some further results found in Appendix G. The qualities of the reconstructions of Test Cases I and III are all very good with FOM_o being at or above 0.9996, implying that reconstructed maps are able to reproduce the variations in orientation to high accuracy.

As can be seen in Figure G.1, the result obtained for Test Case II is not nearly optimal,

[9]The results using $\gamma(o)$ have not been published; they were obtained as an extension and improvement upon the results published earlier.

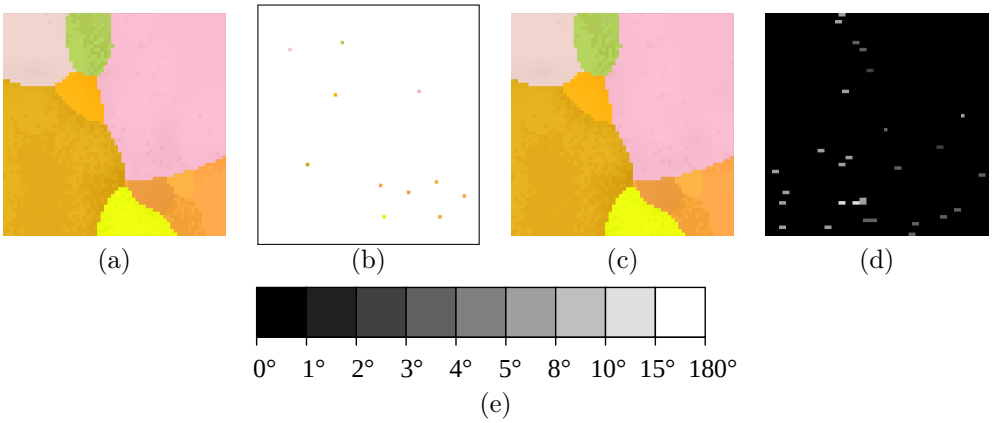


Figure 8.7. Successful reconstruction of Test Case I using $\gamma(o)$ and noiseless projections; $\mathrm{FOM}_o = 0.99987$. **(a)** Reference orientation map (contrast-enhanced). **(b)** Initial orientation map: constant o_0 with $\boldsymbol{q}_0 \coloneqq (1, \mathbf{0})$ and seed points. (Frame added for better visibility.) **(c)** Reconstructed orientation map (contrast-enhanced). **(d)** Difference of the reference and the reconstructed orientation maps. The intensity of the pixels is determined by the distance (disorientation angle) of corresponding orientation pairs, as shown in (e).

yielding a FOM_o around 0.97. Interestingly, this could not be helped by raising δ, Δ or the number of iterations. (We hypothesize that including further 1–2 seeds placed in the erroneous region could be a solution, but this has not been actually verified.)

The reconstruction of Test Case IV was also problematic; the result presented in Figure G.3—with $\mathrm{FOM}_o \approx 0.99$—was achieved by setting $\Delta = \delta = 0.035$ ($\approx 30.4°$). When using the value 0.021 ($\approx 23.5°$) mentioned in Section 8.2.6, the algorithm was unable to correctly reproduce the upper part of the orientation map, yielding a $\mathrm{FOM}_o \approx 0.95$.

For comparison, Figure G.4 demonstrates a reconstruction of Case V using noiseless projections. Actually, the map shown there is a perfect reconstruction with $\mathrm{FOM}_o = 1$ and zero projection error $\|\mathbf{P}_o - \mathbf{P}\|_1$. (Of course, slightly worse results can obtained in less lucky situations, but the quality still remains high with FOM_o values around 0.9999.) It should be noted that this reconstruction was acquired by allowing 5 million iterations, taking about half an hour to complete.

The speed of convergence of $\gamma(o)$ is illustrated in Figure 8.8 for the reconstruction of Case I in the absence of noise. As can be observed, only 856 567 out of 2.5 million Metropolis iterations ($\approx 34.3\,\%$) resulted in the acceptance of a new o' map; the rest of the proposals were rejected. During this time, the exponent of $\gamma(o)$ rose from the initial $\approx -96\,811.1$ up to $\approx 12\,878.7$, spending a bit more than 16 minutes. It is clear that the objective function kept improving reasonably fast for the first $\approx 250\,000$ accepted o' maps, but slowed down considerably thereafter.

To sum up, the application of $\gamma(o)$ in the present circumstances is very questionable. One either needs additional *a priori* data (e. g. in the form of seed points), or optimization shall be carried out via some more sophisticated procedure like SA with a proper annealing schedule. Unfortunately, SA and similar approaches are expected to involve even longer running times. (In fact, reconstructions obtained with $\gamma(g, o)$ are already twice as fast at least, as will be discussed below.) Moreover, $\gamma(o)$ cannot possibly deliver a grain map directly, and the Gibbs model being employed is far not optimal for modeling grain boundaries in orientation maps. For these reasons, no further investigations had been performed regarding the robustness of $\gamma(o)$ (e. g. with respect

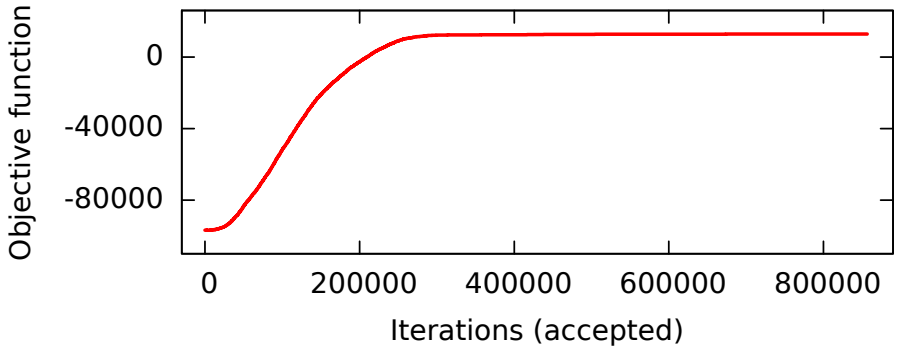


Figure 8.8. Speed of convergence of $\gamma(o)$ as the function of iterative steps. Reconstruction of Test Case I using noiseless projections as per Figure 8.7. Value plotted: exponent of $\gamma(o)$ versus the number of accepted o' maps (out of 2.5 million iterations in total).

to the effect of Gibbs terms or noise), and all efforts were focused on $\gamma(g, o)$ instead.

8.3.2 Reconstructions using $\gamma(g, o)$

Reconstructions tended to converge quite rapidly, needing not more than 2.5 million iterations (corresponding to about 8 minutes of computer time). Figures 8.9 through 8.12 present the results for the four deformed test cases using ideal diffraction patterns. The qualities of the reconstructions in the first three cases are all very good with both FOM values being at or above 0.99. Not only the grain contours but also the gradients (bands and domains) within the grains are generally reproduced to high accuracy. Remarkably, a high quality orientation map—with $\mathrm{FOM}_o = 0.986$—is derived also for Case IV. (In this case, we do not have a physically meaningful grain map and thus FOM_g is irrelevant.)

For comparison, Figure 8.13 demonstrates a reconstruction[10] of Case V in the absence of noise. As a matter of fact, this is a perfect reconstruction with $\mathrm{FOM}_o = 1$ and zero projection error $\|\mathbf{P}_o - \mathbf{P}\|_1$. (It should be remarked that not all results are so flawless in general; the worst one encountered out of 10 random reconstructions had $\mathrm{FOM}_o = 0.999894$ with a total of 4 mismatching pixels in o.) The presence of a substantial grain map error ($\mathrm{FOM}_g = 0.987$) is somewhat surprising. After checking the reference orientation map, this turns out to be caused by two adjacent brownish grains that happen to have almost identical orientations ($(\approx 0.19, -0.38, -0.52, -0.74)$ versus $(\approx 0.15, -0.4, -0.52, -0.74)$, a disorientation of only $\approx 5^\circ$).

Similar reconstructions for 100 % of noise are shown in Figures 8.14–8.15, and figures-of-merit are summarized in Figure 8.16 for noise levels 0 %, 100 % and 200 %, respectively. (The latter plot was acquired by repeating every reconstruction 10 times using different seeds for the pseudo-random number generator, where the error bars indicate the standard error due to this variability.) Additional reconstruction results can be found in Appendix G. The noise in experimental data is estimated to be of order 10 %, so the effect is clearly exaggerated in these simulations. Nevertheless, the FOM values of the reconstructions remain high.

In general, as expected, the errors in the grain map are primarily found to build up at the grain boundaries. The results also reflect expectations that the orientation errors are usually higher in those grains that have the larger orientation spreads. There is though a surprising fact:

[10] The results of Test Case V have not been published; they were obtained as an extension and improvement upon the results published earlier.

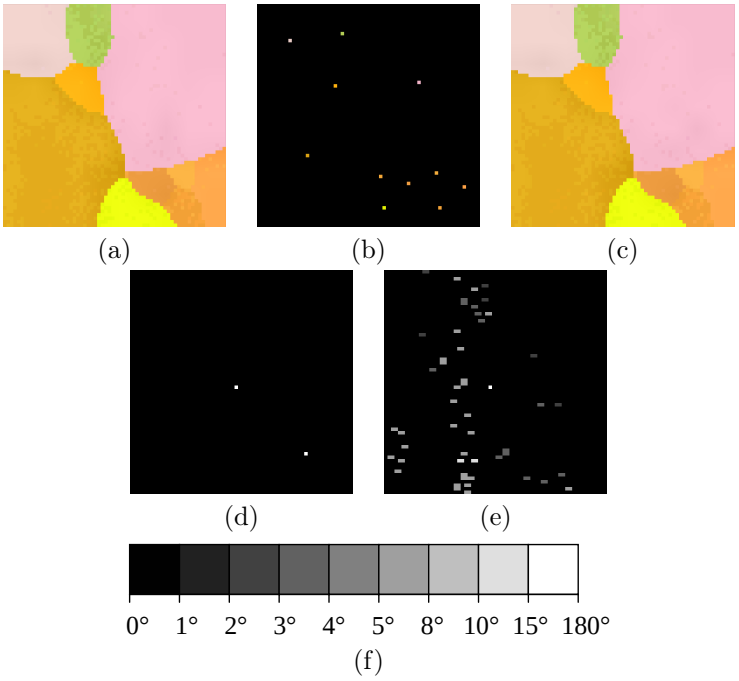


Figure 8.9. The reconstruction of Test Case I using $\gamma(g, o)$ and noiseless projections; $\mathrm{FOM}_g = 0.9995$, $\mathrm{FOM}_o = 0.9996$. **(a)** Reference orientation map (contrast-enhanced). **(b)** Initial orientation map. The initial grain map can be obtained by assigning unique labels to non-black pixels and setting the rest to ambiguous. **(c)** Reconstructed orientation map (contrast-enhanced). **(d)** Difference of the reference and the reconstructed grain maps. Black pixels denote identical grain labels, white pixels represent mismatching ones. **(e)** Difference of the reference and the reconstructed orientation maps. The intensity of the pixels is determined by the distance (disorientation angle) of corresponding orientation pairs, as shown in (f).

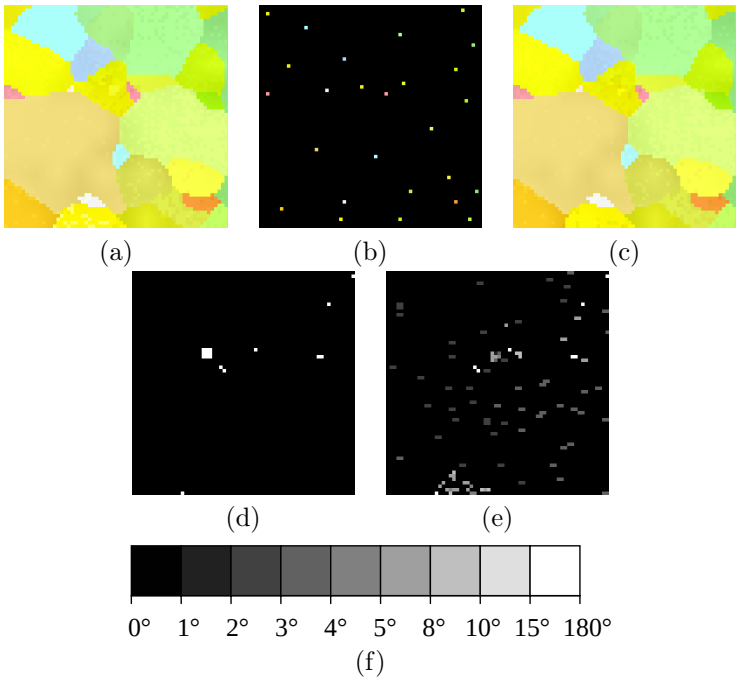


Figure 8.10. The reconstruction of Test Case II using $\gamma(g, o)$ and noiseless projections; $\text{FOM}_g = 0.996$, $\text{FOM}_o = 0.9994$. Map arrangement and gray scales as for Figure 8.9.

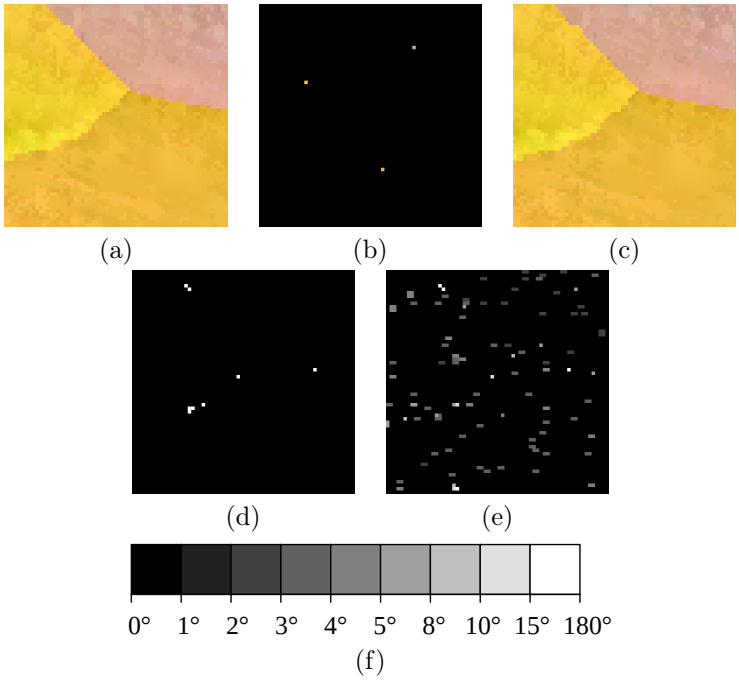


Figure 8.11. The reconstruction of Test Case III using $\gamma(g, o)$ and noiseless projections; $\text{FOM}_g = 0.998$, $\text{FOM}_o = 0.9994$. Map arrangement and gray scales as for Figure 8.9.

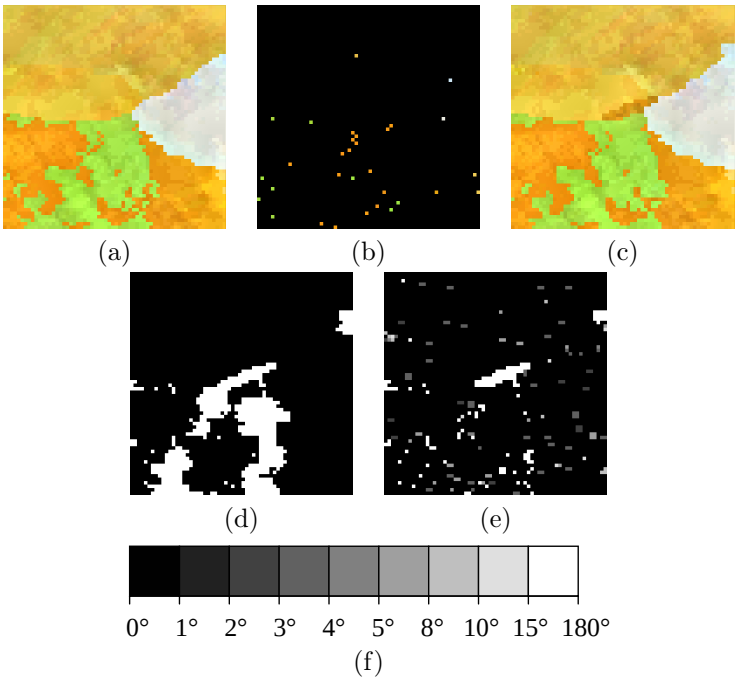


Figure 8.12. The reconstruction of Test Case IV using $\gamma(g, o)$ and noiseless projections; $\mathrm{FOM}_g = 0.857$, $\mathrm{FOM}_o = 0.984$. Map arrangement and gray scales as for Figure 8.9. (Note: (d) is irrelevant due to the lack of a meaningful definition of grains in this case; see text.)

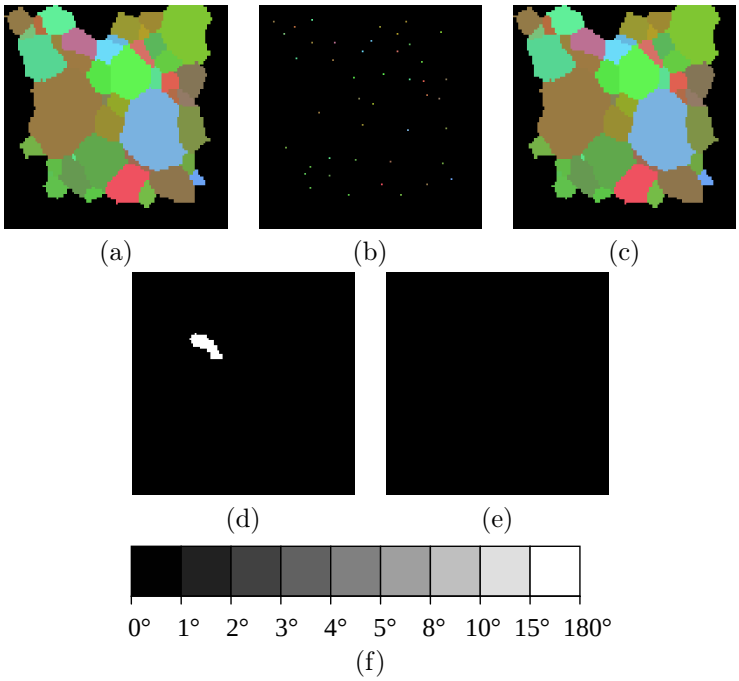


Figure 8.13. The reconstruction of Test Case V using $\gamma(g, o)$ and noiseless projections (original color mapping); $\text{FOM}_g = 0.987$, $\text{FOM}_o = 1$. Map arrangement and gray scales as for Figure 8.9. Black pixels in orientation maps represent void regions. (Note: see text for an explanation of the error in (d).)

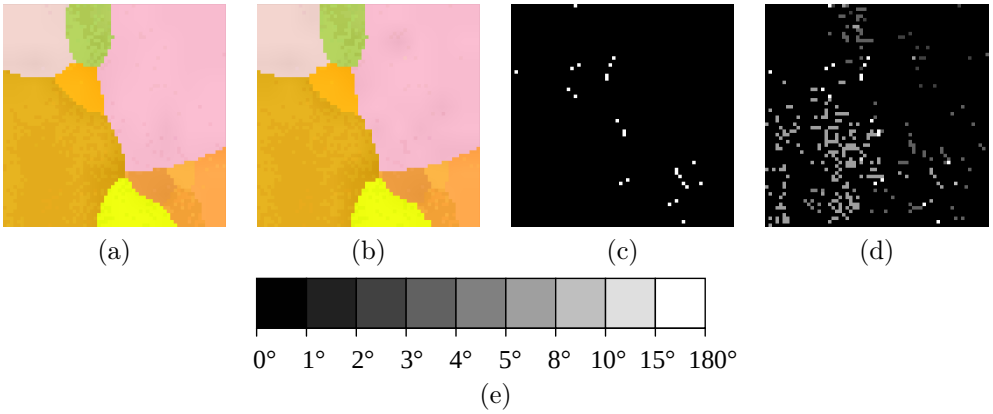


Figure 8.14. The reconstruction of Test Case I using $\gamma(g, o)$ at 100 % noise level; $\mathrm{FOM}_g = 0.994$, $\mathrm{FOM}_o = 0.997$. **(a)** Reference orientation map (contrast-enhanced). **(b)** Reconstructed orientation map (contrast-enhanced). **(c)** Difference of the reference and the reconstructed grain maps. Black pixels denote identical grain labels, white pixels represent mismatching ones. **(d)** Difference of the reference and the reconstructed orientation maps. The intensity of the pixels is determined by the distance (disorientation angle) of corresponding orientation pairs, as shown in (e).

Case III surpassed the others—including Case V—in quality by yielding figures-of-merit quite close to 1. (Except for noise level 0 % where Case V yielded the highest FOM_o.)

The impact of terms $H_1(g, o)$ and $H_2(g)$ was investigated by performing some experiments for Case II when one or both of these terms had been disabled. To get reliable statistics, all simulations were repeated 10 times using different seeds for the pseudo-random number generator. The results are presented in Figure 8.17 for noise levels 0 %, 50 %, 100 %, 200 %, 300 % and 400 %, respectively. It is evident that including at least one of the additional terms is beneficial at higher noise levels, and that $H_2(g)$ has a slightly higher influence than $H_1(g, o)$.

The speed of convergence of $\gamma(g, o)$ is depicted in Figure 8.18 for a typical reconstruction of Case II using noiseless projections. As can be seen, only 804 658 out of 2.5 million Metropolis iterations ($\approx 32.2\,\%$) resulted in the acceptance of a new (g', o') pair of maps; the rest of the proposals were rejected. During this time, the exponent of $\gamma(g, o)$ rose from the initial -55 352 up to $\approx 11\,543.73$, spending altogether slightly over 7 minutes. It is evident that the objective function kept improving steadily for the first $\approx 100\,000$ accepted (g', o') pairs, but slowed down dramatically thereafter.

8.4 Summary

We have demonstrated that the discrete tomography approach yields substantial improvements over the continuous approach (such as ART) when both grain map and orientation map reconstructions are desired. The discrete nature of the reconstruction task was exploited in two ways:

1. By introducing labels and discrete grain maps. This has reduced the size of solution space by orders of magnitude, thereby enabling the use of stochastic routines. We showed that we provide reconstructions of a quality that is clearly superior to previous approaches. The stochastic routines do need *a priori* information in terms of initial maps. However,

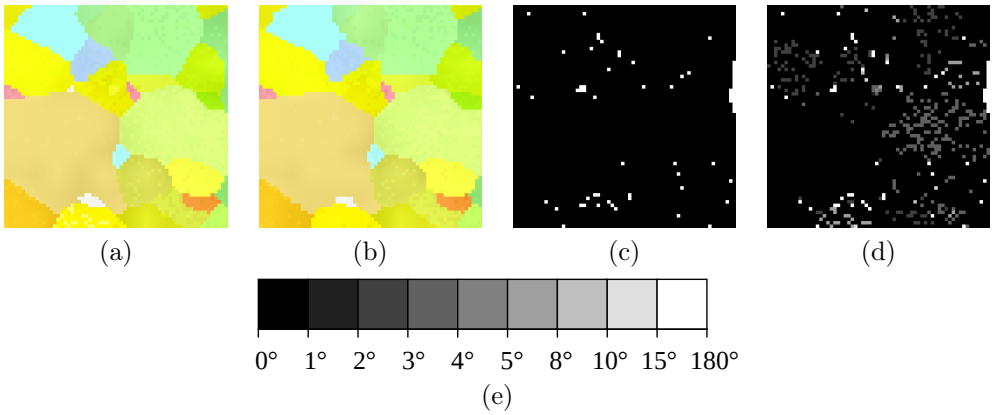


Figure 8.15. The reconstruction of Test Case II using $\gamma(g, o)$ at 100 % noise level; $\mathrm{FOM}_g = 0.984$, $\mathrm{FOM}_o = 0.996$. Map arrangement and gray scales as for Figure 8.14.

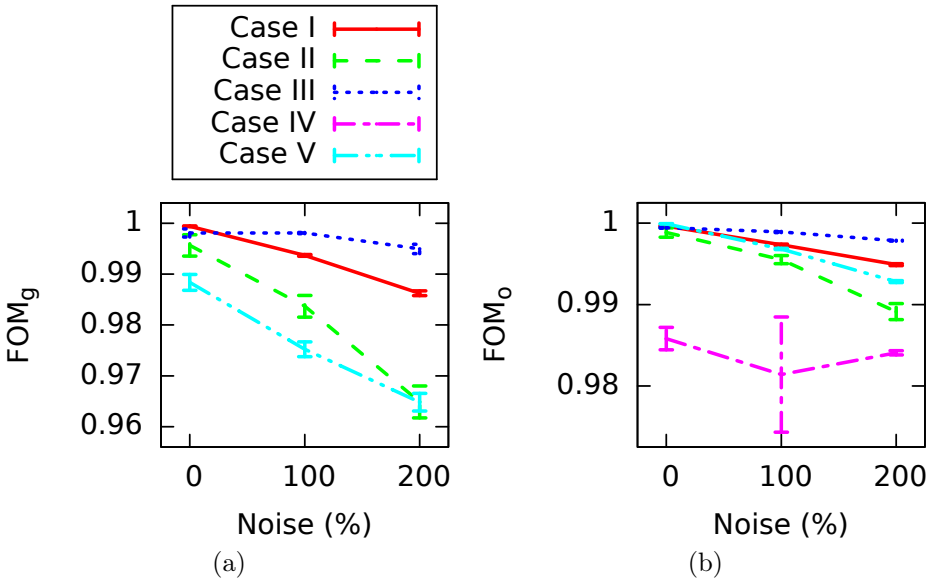


Figure 8.16. Quality of the reconstructions using $\gamma(g, o)$ as the function of noise level, based on 10 repetitions. Error bars indicate the standard error. Larger values correspond to better results. **(a)** FOM_g versus the level of noise. Test Case IV is not shown due to the lack of a physically meaningful grain map. **(b)** FOM_o versus the level of noise.

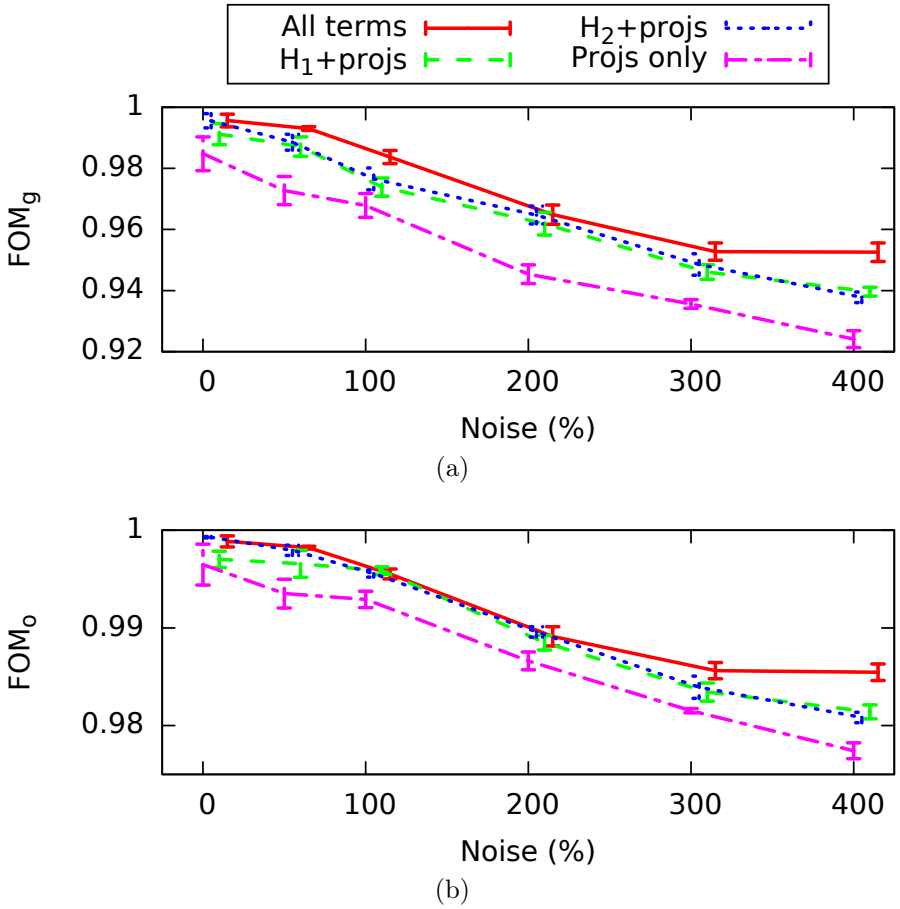


Figure 8.17. Importance of various terms of $\gamma(g, o)$ as the function of noise level, based on 10 repetitions of the reconstruction of Test Case II. Error bars indicate the standard error. Larger values correspond to better results. **(a)** FOM_g versus the level of noise. **(b)** FOM_o versus the level of noise.

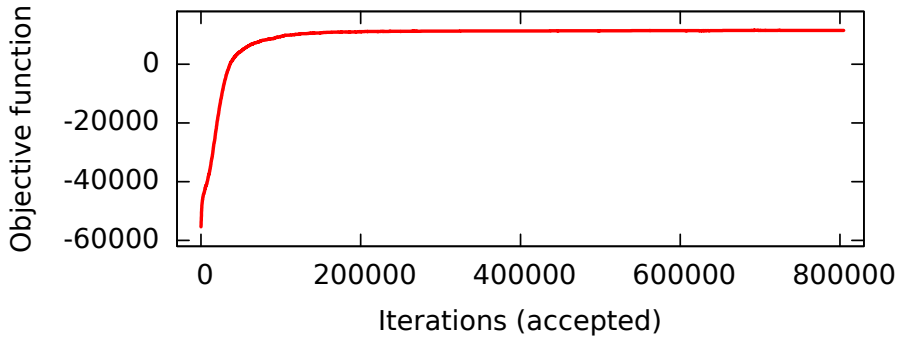


Figure 8.18. Speed of convergence of $\gamma(g, o)$ as the function of iterative steps. Reconstruction of Test Case II using noiseless projections as per Figure 8.10. Value plotted: exponent of $\gamma(g, o)$ versus the number of accepted (g', o') pairs (out of 2.5 million iterations in total).

this requirement is not that demanding; prior data can be supplied by external tools like GRAINSWEEPER [153] as a pre-processing step.

2. By use of priors. To enable this, the methodology has been generalized from a two-level (binary) to multi-level systems. The priors were shown to provide better maps in the cases where the reconstructions based only on the projections deteriorated (few projections, high noise, initial maps with little information). The results indicate that they are not strictly required for noiseless data, but their inclusion is likely to improve substantially the time resolution of dynamic measurements by allowing the use of data with very low count rates.

The case of $\gamma(o)$ is, clearly, a negative result, as it could not deliver satisfactory reconstructions without a proper initial map o_0.

The presented work serves as a basis for further improvements. In fact, the algorithms shown here have been incorporated in the crystallographic software package developed under the auspices of the research program TotalCryst [3, 157]. Moreover, the present approach was used as an initialization step for another method providing grain map / orientation map pairs of moderately deformed specimens using importance sampling and backward projection of 3DXRD diffraction patterns [126].

Appendices

Appendix A

Summary in English

Tomography is a tool of image processing for determining (reconstructing) an image (or a function, in general) from a set of measurements over it (called projections). Discrete tomography (DT), a relatively new field of image processing, deals with the special case when the range of the images / functions is a known, finite set. The latter constraint is, actually, rather easy to satisfy in many real-life problems, thus enabling the usage of DT in these cases.

The present thesis discusses two very different applications of DT: The first one is concerned with the reconstruction of images of objects composed of some geometrical primitives like tubes, cylinders and spheres. The specific scenario considered here arose in industrial nondestructive testing of objects using radiographic measurements. The second application, on the other hand, involves the reconstruction of orientation maps and grain maps of deformed polycrystalline material samples from X-ray diffraction patterns. These tasks can be very challenging but they are also crucial for several materials scientific concepts.

Reconstruction of Objects Parametrized with Geometrical Primitives

In order to tackle the industrial problem brought up in the motivation, the author devised and developed a new stochastic DT reconstruction method that can reconstruct 2D cross-sectional images of objects. These objects are assumed to possess a specific geometrical structure; namely, they can be described as a composition of discs and annuli. Moreover, the object is allowed to be composed of 4 kinds of homogeneous materials that will appear as different pixel intensity levels. The algorithm expects a small number of projections as input taken with parallel beam geometry. To improve robustness against measurement errors, the reconstruction problem is formulated as an optimization task so that solutions are represented in terms of the parameters of constituting geometrical primitives.

Next, the author extended the above approach to enable the reconstruction of 3D objects containing tubes, cylinders and spheres. Instead of being reduced to 2D sub-problems, the algorithm provides a native 3D reconstruction.

Driven by the need to have a large selection of test images, the author also developed an algorithm that can automatically generate random configurations of object parameters in a way that any configuration is chosen with equal probability.

In order to ensure a faster convergence of the optimization process, the author devised and implemented a method for automatically and deterministically constructing initial configurations

based on the input projections. Since the precise pixel intensities are sometimes unavailable for physical measurements, the algorithm also provides estimates for these values.

The effectiveness of the reconstruction techniques, as well as their sensitivity to various factors, were first investigated using simulation studies. The following parameters were considered: the geometrical complexity of the configuration, the number of input projections, the amount of noise present in the projections, and the quality of the initial configuration. In order to get closer to the circumstances found in physical measurements, the author implemented an additive noise model that can be used to distort simulated projections.

Besides using an appropriate initial configuration to start the optimization procedure, further speed-ups were achieved by carrying out several optimizations in the algorithm logic.

Finally, the author also had the opportunity to test the algorithms with several physical measurements. These included 2D as well as 3D objects examined with X-ray, neutron and gamma radiation.

The results shown above have been published in [22, 33, 104, 105, 121–123].

Reconstruction of Deformed Polycrystalline Samples

The scope of the second industrial application is to get an insight into the microstructure of polycrystalline specimens. As a first attempt targeting the general case, the author devised and implemented a new stochastic DT reconstruction technique that can reconstruct the orientation map in a 2D cross-section of a deformed polycrystal from X-ray diffraction measurements. The sample is presumed to be in monophase, i. e. being composed of one material and a single crystal structure. For better robustness against measurement inaccuracies, the reconstruction problem is formulated as an optimization task that seeks a solution in the space of all possible orientation map images. After considering several alternatives, the author eventually decided to represent orientations with unit quaternions and model orientation maps as a Markov random field.

Since the aforementioned general case seemed to be hard to handle efficiently, the author extended the reconstruction method to be able to simultaneously produce both an orientation map and a grain map, both of which are modeled as Markov random fields. This approach is applicable to moderately deformed specimens where one can derive a grain map from the orientation map in a sensible way.

For both techniques rely on the notion of orientation similarity, the author defined and realized a process for expressing and efficiently computing this quantity in the presence of crystal symmetries.

In order to make the reconstruction procedure fast enough for practical usage, the author applied numerous optimizations in the algorithm logic.

Both reconstruction methods were subject to a series of simulation studies using undeformed as well as moderately deformed orientation maps that were all acquired during physical experiments. These investigations helped the author determine the influence of some factors on the reconstruction quality: magnitude of orientation spread within grains, degree of morphological complexity of grain maps, and the amount of noise present in the projections. So as to get a more realistic test setup imitating physical measurements, the author implemented a multiplicative noise model that can be used to distort simulated projections.

The results presented herein have been published in [16, 165, 166].

Conclusions

Two industrial problems have been considered. While quite dissimilar in nature, they nonetheless had a couple of common points: both assumed that the range of the image to be retrieved is a finite, known set, and also that only a limited number of measurements are typically available. These issues together pose serious obstacles to classic tomographic approaches.

With the help of some *a priori* information about the image sought, DT techniques can still be successful in such situations. In the cases presented in the thesis, these priors included an expected geometrical structure, as well as a statistical distribution of grain morphologies and some other local image features. As has been demonstrated, the new methods developed by the author could deliver good quality reconstructions by incorporating prior knowledge into the reconstruction process.

Key Points of the Dissertation

The findings of the research presented in the dissertation can be divided into two thesis groups. Table A.1 shows which thesis point is described in which publication by the author.

Reconstruction of Objects Parametrized with Geometrical Primitives

The results were published in the conference proceedings [122], papers [22, 104, 105, 121, 123], and the book chapter [33].

I/1. The author devised and developed a new stochastic DT reconstruction method that is able to reconstruct 2D discrete images of objects which can be described as a composition of simple geometrical primitives (namely, discs and annuli), and are composed of 4 kinds of homogeneous materials (represented as different pixel intensities). The algorithm expects a small number of projections as input taken with parallel beam geometry. The reconstruction problem is formulated as an optimization task operating in the configuration space, that is looking for solutions in terms of the parameters of constituting geometrical primitives. [122, 123] (Section 4.2)

I/2. The author extended the approach to enable the reconstruction of 3D objects containing tubes, cylinders and spheres, still composed of 4 kinds of homogeneous materials. Unlike some other techniques, this method provides a native 3D reconstruction, i. e. the 3D result is not obtained by simply stacking the reconstructions of individual 2D cross-sections. [33, 104, 105, 121] (Section 4.3)

I/3. In order to be able to test the efficiency of the reconstruction approach, the author developed an algorithm that can automatically generate random configurations of object parameters. This automated process strives to be able to generate all valid potential configurations with equal probability. [104, 122, 123] (Section 4.4)

I/4. The author devised and implemented a method for automatically and deterministically constructing initial configurations based on the input projections, which can then become the starting point of the reconstruction methods. The approach chosen is based on geometrical principles combined with some heuristics. As part of this procedure, the author enhanced the algorithm so that it can provide estimates for the pixel intensity levels needed for the reconstruction of the object, which may not be known accurately enough in case of physical measurements. [104] (Section 4.5)

I/5. The author investigated the efficacy of the reconstruction techniques using numerous simulation experiments, which included mostly randomly generated configurations as well as a few manually constructed ones in both 2D and 3D. The goal of this study was to quantify how sensitive the algorithms were regarding the following factors: the geometrical complexity of the configuration, the number of input projections, and the amount of noise present in the projections. The influence of using an automatically determined initial configuration vs a random one was also determined. In order to simulate the imprecise nature of physical measurements, the author implemented an additive noise model that can be used to distort simulated projections. In all the cases, the precision of reconstruction results was measured with several figure-of-merit functions; one of them was devised and implemented by the author. For the sake of improving reconstruction performance, the author carried out several optimizations in the algorithm logic. As a crucial step, the evaluation of the objective function—required for the optimization process—was sped up by executing computations in an incremental way, so that the current value of the objective function gets updated according to the alteration made in the proposed configuration. [33, 104, 105, 121–123] (Sections 5.1.2 and 5.3)

I/6. The author tested the algorithms with several physical measurements. For the 2D scenario, two test objects were reconstructed whose projections had been acquired with X-rays and neutron radiation, respectively. The technique was also evaluated in case of reconstructing 3D objects; in particular, the same test object was reconstructed from X-ray, neutron and gamma radiation based projections. The accuracy of reconstructions was verified by comparing 2D cross-sections with the result of a classical reconstruction technique (FBP). [22, 33, 104, 105, 121–123] (Section 6.2)

Reconstruction of Deformed Polycrystalline Samples

The results were published in papers [165, 166] and the book chapter [16].

II/1. The author devised and implemented a new stochastic DT reconstruction technique that can reconstruct the orientation map in a 2D cross-section of a deformed polycrystal from X-ray diffraction measurements. Unlike some other approaches, this method utilizes the unaltered projections (diffraction patterns) and produces a discrete solution. The algorithm is applicable to specimens consisting of one known material and a single crystal structure. The reconstruction problem is formulated as an optimization task operating in the space of all possible orientation map images, where each pixel represents the local orientation of the crystalline lattice at that location expressed as a unit quaternion. Orientation maps are modeled as a Markov random field based on a combination of a homogeneity term and a collection of clique configurations conveying local image features (namely, grain boundaries). [165] (Section 7.2)

II/2. The author extended the reconstruction method to be able to simultaneously produce both an orientation map and a grain map. Besides the input projections (X-ray diffraction patterns), the algorithm also requires some *a priori* data in the form of statistics about typical grain morphologies, approximate locations and basic orientations. The extended approach is applicable to moderately deformed specimens where a grain map can be meaningfully derived from the orientation map. In order to take advantage of local image features, both maps are modeled as Markov random fields. [16, 166] (Section 7.3)

II/3. The author defined and realized a process for expressing the similarity of orientations in the presence of crystal symmetries. Besides the basic definition, the author also gave a

Table A.1. The connection between the thesis points and the author's publications.

	Publication									
Thesis point	[16]	[22]	[33]	[104]	[105]	[121]	[122]	[123]	[165]	[166]
I/1.							•	•		
I/2.			•	•	•	•				
I/3.				•			•	•		
I/4.				•						
I/5.			•	•	•	•	•	•		
I/6.		•	•	•	•	•	•	•		
II/1.									•	
II/2.	•									•
II/3.	•								•	•
II/4.	•								•	•
II/5.	•									•

formulation for the efficient computing of this measure. [16, 165, 166] (Section 7.4)

II/4. For the sake of improving reconstruction performance, the author carried out several optimizations in the algorithm logic. One such enhancement was the use of look-up tables for computation-intensive expressions. Additional speed-ups were gained by relying on quantized unit quaternions to represent orientations, and by updating the objective function value incrementally according to the alteration made in the proposed map or pair of maps. [16, 165, 166] (Section 8.1.2)

II/5. The author run numerous simulations to quantitatively characterize the quality of the reconstructions, based on an undeformed and 4 moderately deformed orientation maps that were all acquired during physical experiments. The aim of these investigations was to determine the sensitivity of the algorithms regarding the following factors: magnitude of orientation spread within grains, degree of morphological complexity of grain maps, and the amount of noise present in the projections. In order to mimic the inaccuracies found in physical measurements, the author implemented a multiplicative noise model that can be used to distort simulated projections. In all the cases, the precision of reconstruction results was measured with a pair of figure-of-merit functions, one defined over the grain map and the other over the orientation map. [16, 166] (Section 8.3)

B. függelék

Összefoglaló Magyar Nyelven

A tomográfia egy képfeldolgozási eszköz képek (vagy, általánosságban, függvények) megállapítására (rekonstrukciójára) az azokon vett mérések (vetületek) alapján. A diszkrét tomográfia (DT), a képfeldolgozás egy viszonylag új területe, azzal a speciális esettel foglalkozik, amikor a képek / függvények értékkészlete egy ismert véges halmaz. Ez utóbbi feltételt sok valós életbeli probléma esetén igen könnyű teljesíteni, ily módon lehetővé téve a DT használatát ezekben az esetekben.

Jelen dolgozat a DT két eltérő alkalmazását tárgyalja: Az első olyan tárgyak képeinek rekonstrukciójával foglalkozik, amik néhány geometriai alakzatból állnak, mint pl. csövek, hengerek és gömbök. Az itt tekintett konkrét eset ipari nemroncsoló anyagvizsgálat közben merült fel, ahol tárgyakat vizsgáltak radiográfiai mérésekkel. Másrészt, a második alkalmazás deformált polikristályos anyagminták orientáció- és szemcsetérképének rekonstrukciójával foglalkozik, méghozzá az azok röntgendiffrakciós mintáiból. Bár ezek a feladatok igen komplikáltak, mégis kritikusak sok anyagtudományi téma szempontjából.

Geometriai Alakzatokkal Paraméterezhető Tárgyak Rekonstrukciója

Azért, hogy a motivációban felvetett problémát kezelni tudja, a szerző egy új sztochasztikus DT rekonstrukciós módszert dolgozott és fejlesztett ki, amely tárgyak 2D keresztmetszeti képeit képes rekonstruálni. Ezekről a tárgyakról feltételezzük, hogy egy konkrét geometriai szerkezettel bírnak: egész pontosan, körlapok és körgyűrűk kompozíciójaként írhatók le. Továbbá, a tárgy 4-féle homogén anyagból állhat, amelyek eltérő pixel intenzitásszintekként fognak megjelenni. Az algoritmus bemenetként kevés számú, párhuzamos geometriával vett vetületet vár. A mérési hibákkal szembeni robusztusság céljából a rekonstrukciós problémát optimalizációs feladatként fogalmazzuk meg oly módon, hogy a megoldásokat a tárgyat alkotó geometriai alakzatok paramétereivel fejezzük ki.

A szerző ezután kiterjesztette a fenti megközelítést, hogy az képes legyen csöveket, hengereket és gömböket tartalmazó 3D tárgyak rekonstrukciójára. Ez az algoritmus natív 3D rekonstrukciót nyújt ahelyett, hogy 2D alproblémákra vezetné vissza a problémát.

Mivel a későbbi szimulációs futásokhoz nagyméretű teszt adathalmazra volt szükség, a szerző kifejlesztett egy olyan algoritmust, ami a tárgyparaméterek véletlen konfigurációit képes automatikusan generálni oly módon, hogy bármely konfigurációt azonos valószínűséggel válasszuk.

Azon célból, hogy biztosítsa az optimalizációs folyamat gyorsabb konvergálását, a szerző ki-

fejlesztett és megvalósított egy módszert a kezdőkonfigurációk automatikus és determinisztikus konstruálására a bemeneti vetületek alapján. Mivel a fizikai mérések során a pontos pixel intenzitások néha nem állnak rendelkezésre, az algoritmus ezek értékéről is szolgáltat egy becslést.

A rekonstrukciós módszerek hatásosságát, valamint azok érzékenységét különféle tényezőkkel szemben először szimulációs tesztekkel vizsgáltuk. A következő paramétereket tekintettük: a konfiguráció geometriai bonyolultsága, a bemeneti vetületek száma, a vetületekben jelenlevő zaj mennyisége, valamint a kezdőkonfiguráció minősége. Azért, hogy minél közelebb kerüljünk a fizikai méréseknél tapasztalható körülményekhez, a szerző egy additív zajmodellt fejlesztett ki, amit a szimulált vetületek eltorzítására használhatunk.

Azon kívül, hogy megfelelő kezdőkonfigurációból indítjuk az optimalizációs folyamatot, további gyorsulást értünk el az algoritmus logikájának optimalizálásával.

Végezetül, a szerzőnek lehetősége volt számos fizikai méréssel is tesztelni az algoritmust. Ezek között 2D és 3D tárgyak is voltak, amiket röntgen-, neutron- és gamma-sugárzással vizsgáltak.

A fentebb bemutatott eredmények itt kerültek publikálásra: [22, 33, 104, 105, 121–123].

Deformált Polikristályos Minták Rekonstrukciója

A második ipari alkalmazás során a feladat az, hogy polikristályos minták mikroszerkezetét határozzuk meg. Az általános esetet megcélzó első próbaként, a szerző kitalált és megvalósított egy új sztochasztikus DT rekonstrukciós technikát, ami egy deformált polikristály 2D keresztmetszetében képes rekonstruálni az orientáció-térképet röntgendiffrakciós mérésekből. A mintáról feltételezzük, hogy egyfázisú, azaz egyféle anyagból és egyetlen kristályszerkezetből áll. Azért, hogy robusztusabb legyen a mérési pontatlanságokkal szemben, a rekonstrukciós problémát optimalizációs feladatként fogalmazzuk meg, ami az összes lehetséges orientáció-térképet tartalmazó térben keres megoldást. Számos alternatíva áttekintése után a szerző végül úgy döntött, hogy az orientációkat egységkvaterniókkal reprezentálja, valamint az orientáció-térképet Markov valószínűségi mezőként modellezi.

Mivel a fent említett általános esetet látszólag nehéz volt hatékonyan kezelni, a szerző kiterjesztette a rekonstrukciós módszert, hogy az képes legyen szimultán egy orientáció- és egy szemcsetérképet gyártani úgy, hogy mindkét térképet Markov valószínűségi mezőként modellezzük. Ez a megközelítés mérsékelten deformált mintákhoz használható, amik esetében értelmes módon lehetséges az orientáció-térképből a szemcsetérképet kinyerni.

Mivel mindkét technika az orientációk hasonlóságának fogalmán alapszik, a szerző definiált és megvalósított egy módszert, amivel ezt a mennyiséget kristályszimmetriák jelenléte esetén kifejezhetjük és hatékonyan kiszámolhatjuk.

A rekonstrukciós módszer gyorsabbá tétele érdekében a szerző számos optimalizálást alkalmazott az algoritmus logikájában, hogy a módszer megfelelő teljesítményt nyújtson gyakorlati használatra.

Mindkét rekonstrukciós módszert teszteltük egy sor szimulációval, mind deformálatlan, mind mérsékelten deformált orientáció-térképekkel, amik mindegyikét fizikai kísérletek során nyerték ki. Ezek a vizsgálatok segítették a szerzőt, hogy meghatározza néhány tényezőnek a behatását a rekonstrukciós minőségre: az orientáció szórásának nagysága a szemcséken belül, a szemcsetérképek morfológiai bonyolultságának foka, és a vetületekben jelenlevő zaj mennyisége. Ahhoz, hogy minél valósághűbb tesztkörnyezetet kapjunk, ami a fizikai méréseket imitálja, a szerző egy multiplikatív zajmodellt valósított meg, amivel eltorzíthatók a szimulált vetületek,

Az itt bemutatott eredmények a következő helyeken lettek publikálva: [16, 165, 166].

Konklúziók

Két ipari problémával foglalkoztunk. Miközben eléggé eltérő természetűek, mégis volt pár közös pontjuk: mindkettő feltételezte, hogy a kinyerendő kép értékkészlete egy véges, ismert halmaz, továbbá hogy tipikusan csak limitált számú mérések állnak rendelkezésre. Ezek a problémák együtt komoly akadályokat állítanak a klasszikus tomográfiai megközelítések elé.

A keresett képről rendelkezésre álló néhány előzetes információ segítségével a DT technikák még ilyen szituációkban is sikeresek lehetnek. A dolgozatban bemutatott esetekben ezek az előzetes információk egyrészt a várt geometriai szerkezetből, másrészt a szemcse-morfológiák statisztikai eloszlásából és pár egyéb lokális képjellemzőből álltak. Amint demonstráltuk, a szerző által kifejlesztett új módszerek jó rekonstrukciós minőséget tudnak nyújtani azáltal, hogy az előzetes információkat beépítik a rekonstrukciós folyamatba.

A Disszertáció Tézispontjai

A disszertációban bemutatott eredmények két téziscsoportba oszthatók. A B.1 táblázat mutatja, melyik tézispont a szerző melyik publikációjában van leírva.

Geometriai Alakzatokkal Paraméterezhető Tárgyak Rekonstrukciója

Az eredmények a [122] konferenciakiadványban, a [22,104,105,121,123] cikkekben, valamint a [33] könyvfejezetben kerültek publikálásra.

I/1. A szerző kifejlesztett és megvalósított egy új sztochasztikus DT rekonstrukciós módszert, ami olyan tárgyak 2D diszkrét képeit képes rekonstruálni, amik egyszerű geometriai alakzatok (konkrétan, körlapok és körgyűrűk) kompozíciójaként írhatók le, és 4-féle homogén anyagból állnak (amelyek különféle pixel intenzitásként jelennek meg). Az algoritmus bemenetként kevés számú, párhuzamos geometriájú vetületet vár. A rekonstrukciós problémát optimalizációs feladatként fogalmazzuk meg a konfigurációk terében, tehát a megoldásokat a tárgyat alkotó geometriai alakzatok paramétereivel fejezzük ki. [122, 123] (4.2. alfejezet)

I/2. A szerző kiterjesztette a megközelítést, hogy az képes legyen csöveket, hengereket és gömböket tartalmazó 3D tárgyak rekonstrukciójára, amik továbbra is 4-féle homogén anyagból állnak. Néhány más technikától eltérően ez a módszer natív 3D rekonstrukciót ad, azaz a 3D eredmény nem úgy születik, hogy az egyes 2D keresztmetszeteket egyszerűen egymásra halmozzuk. [33, 104, 105, 121] (4.3. alfejezet)

I/3. Azért, hogy lehetőség legyen a rekonstrukciós megközelítés hatékonyságának tesztelésére, a szerző kifejlesztett egy algoritmust, ami a tárgyparaméterek véletlen konfigurációit képes automatikusan generálni. Ez az automatikus folyamat arra törekszik, hogy bármely potenciális konfigurációt azonos valószínűséggel válasszuk. [104, 122, 123] (4.4. alfejezet)

I/4. A szerző kifejlesztett és megvalósított egy módszert a kezdőkonfigurációk automatikus és determinisztikus konstruálására a bemeneti vetületek alapján, amely konfigurációk utána a rekonstrukciós módszerek kezdőpontjává válhatnak A megközelítés geometriai elvekkel kombinált heurisztikára épül. Ezen eljárás részeként a szerző kiterjesztette az algoritmust, hogy az becslést tudjon adni a tárgy rekonstrukciójához szükséges pixel intenzitásokról, melyek a fizikai mérések során esetleg nem elég pontossággal ismertek. [104] (4.5. alfejezet)

I/5. A szerző számos szimulációs kísérlet segítségével megvizsgálta a rekonstrukciós technikák hatékonyságát, amelyek nagyrészt véletlenül generált konfigurációt, másrészt néhány manuálisan konstruált konfigurációt használtak mind 2D-ben, mind 3D-ben. Ennek a tanulmánynak az volt a célja, hogy meghatározza, mennyire érzékenyek az algoritmusok a következő tényezőkkel szemben: a konfiguráció geometriai bonyolultsága, a bemeneti vetületek száma, és a vetületekben jelenlevő zaj mennyisége. Az is meg lett határozva, mi a befolyása, ha automatikusan megállapított kezdőkonfigurációt használunk véletlen helyett. Azon célból, hogy szimulálhassuk a fizikai mérések pontatlan természetét, a szerző egy additív zajmodellt fejlesztett ki, amit a szimulált vetületek eltorzítására használhatunk. A rekonstrukciós eredményének pontosságát minden esetben több mérőszámmal mértük; ezek közül az egyiket a szerző találta ki és valósította meg. A rekonstrukció teljesítményének javítása érdekében a szerző számos optimalizálást hajtott végre az algoritmus logikáján. Az egyik kritikus lépésként a célfüggvény – amire az optimalizáció miatt van szükség – kiértékelése fel lett gyorsítva azáltal, hogy a számolásokat inkrementális módon hajtjuk végre, tehát a célfüggvény aktuális értéke frissítésre kerül a javasolt konfigurációban elkövetett változtatás alapján. [33, 104, 105, 121–123] (5.1.2. és 5.3. alfejezet)

I/6. A szerző számos fizikai méréssel tesztelte az algoritmusokat. A 2D esetben két tárgy lett rekonstruálva, amelyek vetülete röntgen- ill. neutron-sugárzással lett begyűjtve. A technikát 3D tárgyak rekonstrukciójával is kiértékeltük; konkrétan, ugyanaz a tárgy lett rekonstruálva röntgen-, neutron- és gamma-sugárzással készült vetületekből. A rekonstrukció pontosságát a 2D keresztmetszetek és egy klasszikus technika (FBP) eredményének összevetésével ellenőriztük. [22, 33, 104, 105, 121–123] (6.2. alfejezet)

Deformált Polikristályos Minták Rekonstrukciója

Az eredmények a [165, 166] cikkekben és a [16] könyvfejezetben kerültek publikálásra.

II/1. A szerző kifejlesztett és megvalósított egy új sztochasztikus DT rekonstrukciós módszert, ami egy deformált polikristály 2D keresztmetszetében képes rekonstruálni az orientáció-térképet röntgendiffrakciós mérésekből. Néhány más megközelítéstől eltérően ez a módszer a változatlan vetületeket (diffrakciós mintákat) használja, és diszkrét megoldást nyújt. Az algoritmus egyféle anyagból és egyetlen kristályszerkezetből álló mintákkal használható. A rekonstrukciós problémát optimalizációs feladatként fogalmazzuk meg, ami az összes lehetséges orientáció-térképet tartalmazó térben keres megoldást, ahol minden pixel a kristályrács azon pontján mért lokális orientációját reprezentálja egységkvaterniókkal. Az orientáció-térképeket Markov valószínűségi mezőként modellezzük egy homogenitási feltétel és klikk konfigurációk kombinációja segítségével, ahol az utóbbiak lokális képjellemzőket fejeznek ki (konkrétan, szemcsehatárokat). [165] (7.2. alfejezet)

II/2. A szerző kiterjesztette a rekonstrukciós módszert, hogy az képes legyen szimultán egy orientáció- és egy szemcsetérképet gyártani. A bemeneti vetületek (röntgendiffrakciós minták) mellett az algoritmusnak egy kis előzetes információra is szüksége van, mégpedig a tipikus szemcse-morfológiák statisztikája, valamint a szemcsék megközelítő helyzete és alaporientációja formájában. A kiterjesztett megközelítés mérsékelten deformált mintákhoz használható, amik esetében értelmes módon lehetséges az orientáció-térképből a szemcsetérképet kinyerni. A lokális képjellemzők előnyeinek kihasználása érdekében mindkét térképet Markov valószínűségi mezőként modellezzük. [16, 166] (7.3. alfejezet)

II/3. A szerző definiált és megvalósított egy módszert, amivel az orientációk hasonlóságát kristályszimmetriák jelenléte esetén kifejezhetjük. Az alapdefiníció mellett a szerző arra is

B.1. táblázat. A tézispontok és a szerző publikációi közötti kapcsolat.

	Publikáció									
Tézispont	[16]	[22]	[33]	[104]	[105]	[121]	[122]	[123]	[165]	[166]
I/1.							•	•		
I/2.			•	•	•	•				
I/3.				•			•	•		
I/4.				•						
I/5.			•	•	•	•	•	•		
I/6.		•	•	•	•	•	•	•		
II/1.									•	
II/2.	•									•
II/3.	•								•	•
II/4.	•								•	•
II/5.	•									•

kitért, hogyan lehet ezt a mennyiséget hatékonyan kiszámolni. [16, 165, 166] (7.4. alfejezet)

II/4. A rekonstrukció teljesítményének javítása érdekében a szerző számos optimalizálást alkalmazott az algoritmus logikájában. Az egyik ilyen ötlet a look-up táblák használata volt számolásigényes kifejezésekben. További gyorsulást értünk el kvantált egységkvaterniók használatával az orientációk reprezentációja esetén, továbbá a célfüggvény értékének inkrementális frissítésével a javasolt térképben vagy térkép-párban elkövetett változtatás alapján. [16, 165, 166] (8.1.2. alfejezet)

II/5. A szerző számos szimulációt futtatott, hogy számszerűsítse a rekonstrukciók minőségét, egyrészt egy deformálatlan, valamint 4 mérsékelten deformált orientáció-térképpel, amik mindegyikét fizikai kísérletek során nyerték ki. Ezeknek a vizsgálatoknak az volt a céljuk, hogy megállapítsák, mennyire érzékenyek az algoritmusok a következő tényezőkkel szemben: az orientáció szórásának nagysága a szemcséken belül, a szemcsetérképek morfológiai bonyolultságának foka, és a vetületekben jelenlevő zaj mennyisége. Azért, hogy jobban utánozzuk a fizikai mérésekben megtalálható pontatlanságokat, a szerző egy multiplikatív zajmodellt valósított meg, amivel eltorzíthatók a szimulált vetületek. A rekonstrukciók eredményének pontosságát minden esetben két mérőszámmal mértük, ahol az egyik a szemcsetérképen, a másik az orientáció-térképen volt definiálva. [16, 166] (8.3. alfejezet)

Appendix C

Crystallographic Basics

This appendix gives a brief introduction to the crystallographic notions relevant for the thesis; thus it complements Section 3.2.1. Furthermore, the mathematical relationship between the object and its projections is explained (following [130, 153]). For a deeper introduction we refer to [21, 58, 106, 189].

The contents of this appendix was adapted from [14] in its entirety; it is reproduced here with the permission of the authors.

C.1 Lattices

In a crystalline material the atoms are arranged in a three-dimensional discrete lattice. The **crystalline lattice** can be fully characterized by three properties:

1. The crystal's regularity, i. e., the periodic array in which the repeated units of the crystal are arranged, is captured by the concept of the **Bravais lattice**,

$$\mathcal{L} = \{\, \mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x} = \alpha_1 \mathbf{a} + \alpha_2 \mathbf{b} + \alpha_3 \mathbf{c}, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{Z} \,\} \ ,$$

   where $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$ are given vectors not all lying in the same plane. Notably, for a given lattice, a number of symmetry operations exist (such as translations, inversions, rotations and mirror operations) that map the Bravais lattice onto itself. Lattices that are invariant under such operations are said to exhibit the same **crystal symmetry**. Further definitions and properties of the Bravais lattice are summarized below.

2. The (crystalline) **basis**: This is the configuration of atoms that is repeated at each point in the Bravais lattice. It can be as small as a single atom (e. g. an unalloyed metal, leading to a **monatomic** basis), a collection of atoms / ions (e. g. rock salt, formed by sodium and chloride ions), or as large as a complex molecule (e. g. frozen and crystallized proteins).

3. The **orientation** of the lattice. Notably, due to crystal symmetry the orientation is not uniquely defined (see Section 7.4 for more details). A **polycrystal**, as mentioned in Section 7.1, consists of smaller crystals (**grains** or **crystallites**) at different orientations.

The vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ above are called the **primitive vectors** of the lattice. In the crystallographic literature, one often specifies a lattice by six **lattice parameters** $(a, b, c, \alpha, \beta, \gamma) \in \mathbb{R}^6$. These parameters define the lengths of the primitive vectors and their interaxial angles, i. e.,

$$a \coloneqq \|\mathbf{a}\|_2 \,, \quad b \coloneqq \|\mathbf{b}\|_2 \,, \quad c \coloneqq \|\mathbf{c}\|_2$$

and

$$\alpha \coloneqq \arccos \frac{\mathbf{b} \cdot \mathbf{c}}{bc}, \quad \beta \coloneqq \arccos \frac{\mathbf{c} \cdot \mathbf{a}}{ca}, \quad \gamma \coloneqq \arccos \frac{\mathbf{a} \cdot \mathbf{b}}{ab},$$

where $\cdot$ denotes the scalar product.

Polycrystals containing two or more grains with different lattice structure are said to be in **multiphase**, otherwise they are in **monophase**. It is this latter case that is considered in this thesis.

Given a Bravais lattice with primitive vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, one defines the **reciprocal lattice** to be the lattice with primitive vectors $\mathbf{a}^*$, $\mathbf{b}^*$ and $\mathbf{c}^*$ fulfilling

$$\mathbf{a} \cdot \mathbf{a}^* = \mathbf{b} \cdot \mathbf{b}^* = \mathbf{c} \cdot \mathbf{c}^* = 2\pi$$

and

$$\mathbf{a} \cdot \mathbf{b}^* = \mathbf{a} \cdot \mathbf{c}^* = \mathbf{b} \cdot \mathbf{a}^* = \mathbf{b} \cdot \mathbf{c}^* = \mathbf{c} \cdot \mathbf{a}^* = \mathbf{c} \cdot \mathbf{b}^* = 0\,.$$

Lattice planes—defined as planes containing at least three noncollinear Bravais lattice points—are an essential notion for describing the diffraction process. These lattice planes are usually described by **Miller indexes**, which are integers $h, k, l \in \mathbb{Z}$, with greatest common divisor equal to 1, defining the lattice plane normal to the vector $h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$ in the reciprocal lattice. Miller indexes corresponding to a particular set of lattice planes are usually designated by (hkl). For instance, (111) denotes the set of lattice planes whose normal is parallel to the vector $\mathbf{a}^* + \mathbf{b}^* + \mathbf{c}^*$ in the reciprocal lattice. The distance d between two adjacent planes among the set of lattice planes associated with some Miller indexes (hkl) is called the **lattice spacing**. Furthermore, the notation $\{hkl\}$ is a shorthand for the (Miller indexes of the) set of all lattice planes that are equivalent to (hkl) due to the symmetries of the lattice.

The orientation of a grain can be described by a (proper) rotation in 3D space. Given a fixed coordinate system it is well known that such a rotation can be described by three angles $(\psi, \varphi_1, \varphi_2)$, the so-called **Euler angles** [45], expressing a sequence of 3 rotations. In this notation, the first rotation is by an angle $\psi \in [0, 2\pi)$ about the z-axis, the second is by an angle $\varphi_1 \in [0, \pi]$ about the x-axis, and the third is by an angle $\varphi_2 \in [0, 2\pi)$ again about the z-axis (all angles measured counterclockwise). Straightforward calculations of the three rotation matrices yields the orientation matrix

$$\mathbf{U} \coloneqq \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}, \tag{C.1}$$

where

$$\begin{aligned}
u_{11} &\coloneqq \cos\varphi_1 \cos\varphi_2 - \sin\varphi_1 \sin\varphi_2 \cos\psi\,, \\
u_{12} &\coloneqq -\cos\varphi_1 \sin\varphi_2 - \sin\varphi_1 \cos\varphi_2 \cos\psi\,, \\
u_{13} &\coloneqq \sin\varphi_1 \sin\psi\,, \\
u_{21} &\coloneqq \sin\varphi_1 \cos\varphi_2 + \cos\varphi_1 \sin\varphi_2 \cos\psi\,, \\
u_{22} &\coloneqq -\sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 \cos\psi\,, \\
u_{23} &\coloneqq -\cos\varphi_1 \sin\psi\,, \\
u_{31} &\coloneqq \sin\varphi_2 \sin\psi\,, \\
u_{32} &\coloneqq \cos\varphi_2 \sin\psi\,, \\
u_{33} &\coloneqq \cos\psi\,.
\end{aligned}$$

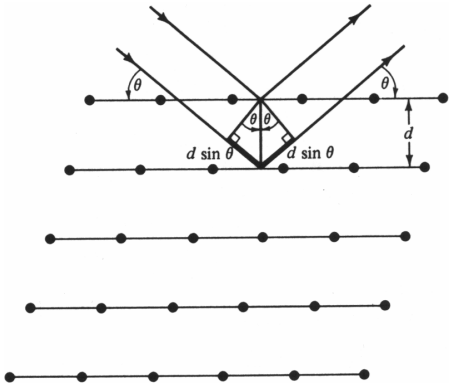


Figure C.1: A Bragg reflection from a particular family of lattice planes, with lattice spacing d and angle of incidence θ. (Image source: [21].)

C.2 Diffraction

The physical principle that governs the data acquisition process of the 3DXRD microscope is the principle of **X-ray diffraction** [58, 153, 189]. Diffraction spots on the detector appear if X-ray beams reflected on parallel lattice planes interfere constructively. **Bragg's law** states that the condition for constructive interference for a wave of wavelength λ against a series of lattice planes with lattice spacing d, rotated by the angle θ off the plane of reflection[1] (called the **Bragg angle**), is

$$n\lambda = 2d\sin\theta, \quad n \in \mathbb{N}\,. \tag{C.2}$$

This is illustrated in Figure C.1. Reformulated in terms of the scattering vector $\mathbf{g}$ (defined as the difference between the vector describing the incident beam and the reflected beam, normalized so that $\|\mathbf{g}\|_2 = 2\pi/d$), Bragg's law states:

$$\|\mathbf{g}\|_2 = \frac{4\pi}{n\lambda}\sin\theta\,. \tag{C.3}$$

For given Miller indexes (hkl) only the case $n = 1$ is relevant, as any higher order harmonics is equivalent to an $n = 1$ case for another distinct Miller indexes $(h'k'l')$. For example, the $n = 3$ case for (111) is equivalent to $n = 1$ for (333).

C.3 Calculating Diffraction Spots

Three Cartesian coordinate systems are utilized: the laboratory system, the rotation table system and the Cartesian grain system. In particular, the origins of the laboratory system and the rotation stage coincide.

Let $(\mathbf{x}_C, \mathbf{y}_C, \mathbf{z}_C)$ denote the Cartesian grain system which is fixed with respect to the reciprocal lattice $(\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*)$ in the grain.

Let $\mathbf{g}_{hkl}$ denote a scattering vector represented by the Miller indexes (hkl) in the reciprocal lattice system. The correspondence between the Cartesian grain system and reciprocal space is

[1] Contrary to the convention in optics, the Bragg angle is indeed measured from the plane of reflection rather than from the plane normal. See e. g. [21, Chapter 6].

given by the matrix $\mathbf{B}$: $\mathbf{g}_C = \mathbf{B}\mathbf{g}_{hkl}$, with

$$\mathbf{B} := \begin{pmatrix} a^* & b^* \cos\gamma^* & c^* \cos\beta^* \\ 0 & b^* \sin\gamma^* & -c^* \sin\beta^* \cos\alpha \\ 0 & 0 & c^* \sin\beta^* \sin\alpha \end{pmatrix}$$

and

$$\cos\alpha = \frac{\cos\beta^* \cos\gamma^* - \cos\alpha^*}{\sin\gamma^* \sin\beta^*} .$$

Here $(a, b, c, \alpha, \beta, \gamma)$ and $(a^*, b^*, c^*, \alpha^*, \beta^*, \gamma^*)$ symbolize the lattice parameters in direct and reciprocal space, respectively.

Given the orientation of the grain by the Euler angles $(\psi, \varphi_1, \varphi_2)$, this scattering vector further transforms by applying the orientation matrix $\mathbf{U}$ of Equation (C.1). Finally, taking the rotation by an angle ω on the rotation table into account, we derive the representation in the laboratory system by

$$\mathbf{g}_l = \mathbf{\Omega}\mathbf{U}\mathbf{B}\mathbf{g}_{hkl} ,$$

where

$$\mathbf{\Omega} := \begin{pmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

Given the angles θ and η as in Figure 3.4 we deduce that

$$\mathbf{g}_l = \frac{2\pi}{\sqrt{2 - 2\cos(2\theta)}} \begin{pmatrix} \cos(2\theta) - 1 \\ -\sin(2\theta)\sin\eta \\ \sin(2\theta)\cos\eta \end{pmatrix} .$$

Notice that $\mathbf{g}_l$ is the diffraction vector, i. e., it is derived by subtracting the incoming beam vector from the diffracted beam vector (in polar coordinates). Together with Bragg's law in Equation (C.3) we derive

$$\mathbf{g}_l = \frac{2\pi}{\lambda} \begin{pmatrix} \cos(2\theta) - 1 \\ -\sin(2\theta)\sin\eta \\ \sin(2\theta)\cos\eta \end{pmatrix} . \tag{C.4}$$

In summary, given a lattice plane we can calculate by Equation (C.4) the angles η and θ fulfilling Bragg's law, and thus infer that the beam is diffracted along a straight line with direction vector

$$\mathbf{v}_l = \begin{pmatrix} \cos(2\theta) \\ -\sin(2\theta)\sin\eta \\ \sin(2\theta)\cos\eta \end{pmatrix} .$$

Now, let $(y_{det}(0), z_{det}(0))$ denote the intersection of the detector plane with the incoming (non-diffracted) ray passing through the origin O of the laboratory system, and L the distance between O and the detector plane. Given a point (x_l, y_l, z_l) of the grain (with respect to the laboratory system), it easily follows that the associated diffraction point on the detector (L, y_{det}, z_{det}) can be calculated by

$$\begin{aligned} y_{det} &= -(L - x_l)\tan(2\theta)\sin\eta + y_l - y_{det}(0) , \\ z_{det} &= (L - x_l)\tan(2\theta)\cos\eta + z_l - z_{det}(0) . \end{aligned} \tag{C.5}$$

Appendix D

Generating Random Configurations for 2D Parametric Objects

This appendix explains the exact process of generating random configurations for 2D parametric objects; it complements and expands upon the notions introduced in Section 4.4. The execution of the algorithm starts at Function GenerateConfiguration. The input of this function is the target number N of discs to be included, and its result is a valid generated configuration $\mathbf{c}$ containing exactly N discs.

Generating the Annulus

The construction of the new configuration starts with laying down the annulus (see Procedure AddRing). This is performed in two steps: First, the center $O_{\mathrm{RE}} = O_{\mathrm{RI}}$ is generated randomly within a certain circular neighborhood of O_{C} (determined by a user-defined radius about O_{C}). The two radii r_{RE} and r_{RI} are then chosen within certain limits so that the configuration remains valid. (The range of r_{RE} as well as the permissible ring thickness are controlled by another set of user-defined parameters.)

Objectives

Having fixed the location and size of the ring, the process continues with the generation of disc parameters. This step deserves some deliberation, though: On the one hand, it is only possible to obtain a meaningful statistics on the quality of reconstructions when every element of the configuration space $\mathcal{C}$ can be generated *equally likely*, i. e. if $\mathbf{c}$ is chosen with uniform probability. On the other hand, the algorithm should also be *robust* enough in order to avoid non-terminating conditions (namely, entering an infinite loop in the absence of any valid configurations). The approach described below tries hard to satisfy these requirements as much as possible.

Generating Discs

The algorithm proceeds with randomly fixing the centers of all the discs at once, after which disc radii will be generated individually (see Procedure AddDisc). The choice of disc centers is driven by two guidelines: all radii should be greater than or equal to r_{min} (again specified by the user), and it should be possible to decide if a new disc with some given radius r_i can be added to $\mathbf{c}$ so that it remains valid. Actually, the algorithm is able to answer a harder question:

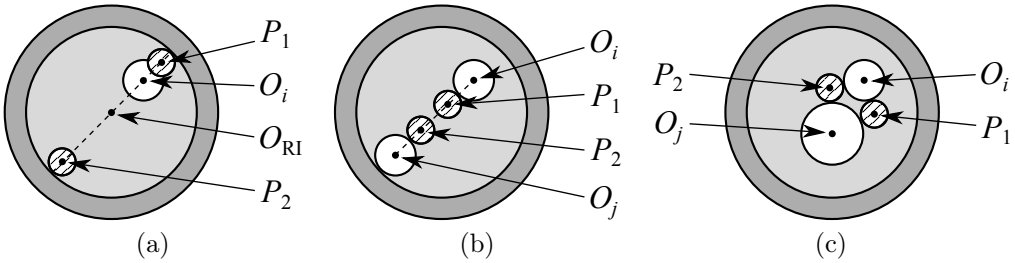


Figure D.1: Verifying if a new disc (drawn striped) with radius r can be added to $\mathbf{c}$ without violating geometrical constraints. All mentioned algorithm lines are from Function RadiusGood. **(a)** The case mentioned in line 4. **(b)** The case mentioned in line 11. **(c)** The case mentioned in line 13.

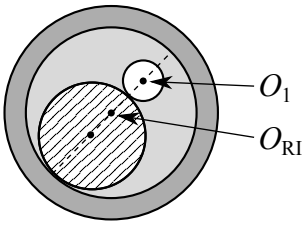


Figure D.2: Finding the maximal radius r_2 of a new disc (drawn striped) in the presence of a single disc (O_1, r_1) in $\mathbf{c}$; see line 4 of Function MaxRadiusNewDisc.

Function MaxRadiusNewDisc—with the help of Function RadiusGood—gives an approximate upper bound for r_i. Having generated disc centers, Function MaxRadiusExistingDisc determines the maximal feasible radius for each newly added disc one by one (based on the position of the disc with respect to the annulus and other neighboring discs); this information is then used to randomly pick the final radius from the feasible range.

The aim of Function RadiusGood is to decide if a new disc with radius r can be added to $\mathbf{c}$ while keeping the latter valid. This task is, in fact, related to the so-called **Apollonius' problem** (named after Greek geometer Apollonius of Perga) that seeks to find circles tangent to three given circles, lines or points in a plane. The algorithm presented here is a sort of heuristics devised by the author; it is based on checking certain candidate centers as shown in Figure D.1. The task is trivial to solve when $\mathbf{c}$ contains at most one disc; see Figure D.2 for the case of a single existing disc. Using this information, Function MaxRadiusNewDisc finds an estimate for the maximal feasible radius using a binary search over the interval $[0, r_{\mathrm{RI}}]$. The iterative search stops when the width of the current interval falls below a given positive real δ.

Final Remarks

Regarding the aforementioned objectives, the probability of any given $\mathbf{c}$ mostly depends on the properties of the pseudo-random number generator, as well as on the reliability[1] of the estimated maximal feasible radius. On the other hand, the algorithm might be unable to generate all necessary disc centers when choosing too high a $r_{\min}$. In the latter case, potential infinite loops are spotted by setting a strict upper bound on the number of center candidates being tried.

[1] This is unknown, unfortunately; no analysis has been carried out to assess the actual precision / quality of the radius estimate given here.

The generation of 3D configurations is performed by a direct extension and adaptation of the aforementioned procedure. It is, of course, reasonably more complex due to the extra space dimension, as well as due to the need for supporting two kinds of objects (spheres and cylinders) within the interior.

Function GenerateConfiguration(N)

Input : number of discs the generated configuration shall contain (N)
Output : generated configuration

```
1  c := empty configuration
2  AddRing(c)
3  AddDisc(c, N)
4  return c
```

Procedure AddRing(**c**)

Input : configuration to amend (**c**)

```
1  Generate O_RE
2  O_RI := O_RE
3  Generate r_RE and r_RI
4  Add ring parameters to c
```

Procedure AddDisc(**c**, N)

Input : configuration to amend (**c**), number of discs to add (N)

```
1  N' := number of discs in c
   /* Generating disc centers */
2  for i := N' + 1 to N' + N do
3  |   r := MaxRadiusNewDisc(c)
4  |   if r < r_min then
5  |   |   Error
6  |   r_i := r_min                                   /* Temporary setting */
7  |   Generate O_i using temporary r_i
8  |   if generation of O_i failed then
9  |   |   Error
10 |   Add disc (O_i, r_i) to c
   /* Generating disc radii */
11 for i := N' + 1 to N' + N do
12 |   r := MaxRadiusExistingDisc(c, i)
13 |   Generate r_i so that r_min ≤ r_i ≤ r
14 |   Update r_i in c
```

Function MaxRadiusNewDisc(**c**)

Input : configuration to inspect (**c**)
Output : approximate maximal radius of a new disc that may be added to **c** without violating geometric constraints

if there are no discs in **c** **then**
  $r \coloneqq r_{\mathrm{RI}}$
else if there is only a single disc in **c** **then**
  $r \coloneqq \frac{r_{\mathrm{RI}} - r_1 + \overline{O_{\mathrm{RI}} O_1}}{2}$
else
  /* Find approximate maximal radius using binary search */
  $r \coloneqq 0$ /* May be smaller than $r_{\mathtt{min}}$ in worst case */
  $r^{'} \coloneqq r_{\mathrm{RI}}$
  while $r^{'} - r \geq \delta$ **do**
    $x \coloneqq \frac{r + r^{'}}{2}$
    if RadiusGood(**c**, x) **then**
      $r \coloneqq x$
    else
      $r^{'} \coloneqq x$
return r

Function RadiusGood(**c**, r)

Input : configuration to inspect (**c**), radius to test for (r)
Output : **true** if a new disc with radius r may be added to **c** without violating geometric constraints; **false** otherwise

$N^{'} \coloneqq$ number of discs in **c**
/* According to the logic in Function MaxRadiusNewDisc, $N^{'} \geq 2$ */
for $i \coloneqq 1$ **to** $N^{'}$ **do**
  if $O_i \neq O_{\mathrm{RI}}$ **then**
    Let P_1 and P_2 denote the centers of the circles with radius r along line $O_{\mathrm{RI}} O_i$ that are tangent to the interior
    if either of the discs (P_1, r) and (P_2, r) may be added to **c** without violating geometric constraints **then**
      return true
for $i \coloneqq 1$ **to** $N^{'} - 1$ **do**
  for $j \coloneqq i + 1$ **to** $N^{'}$ **do**
    $r^{'} \coloneqq \frac{\overline{O_i O_j} - r_i - r_j}{2}$
    if $r \leq r^{'}$ **then**
      Let P_1 and P_2 denote the centers of the circles with radius r along line segment $O_i O_j$ that are tangent to either (O_i, r_i) or (O_j, r_j)
    else
      Let P_1 and P_2 denote the intersections of circles $(O_i, r_i + r)$ and $(O_j, r_j + r)$; these are the centers of the circles with radius r tangent to both (O_i, r_i) and (O_j, r_j)
    if either of the discs (P_1, r) and (P_2, r) may be added to **c** without violating geometric constraints **then**
      return true
return false

Function MaxRadiusExistingDisc($\mathbf{c}$, i)

Input : configuration to inspect ($\mathbf{c}$), index of disc to test for (i)
Output : maximal possible radius of disc (O_i, r_i) in $\mathbf{c}$ without violating geometric constraints while keeping O_i fixed

```
1 N' := number of discs in c
2 r := r_RI − O_RI O_i
3 for j := 1 to N' do
4     if i ≠ j then
5         r := min(r, O_i O_j − r_j)
6 return r
```

Appendix E

Constructing Initial Configurations for 2D Parametric Objects

This appendix describes the exact way of constructing initial configurations for 2D parametric objects; it complements and expands upon the notions introduced in Section 4.5. Before discussing the complete procedure, however, we begin with recalling the definition of a well-known operation of digital signal processing: Let $m \geq n$ be positive integers. The **discrete convolution** of vectors $\mathbf{f} = (f_0, \ldots, f_{m-1})$ and $\mathbf{g} = (g_0, \ldots, g_{n-1})$ is obtained as the following vector $\mathbf{h} = (h_0, \ldots, h_{m-1})$:

$$h_i := \begin{cases} \sum_{j=0}^{n-1} f_{i-j} \cdot g_j & \text{if } i \geq n-1\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

$\mathbf{f}$ is dubbed the **digital signal**, its elements are **samples**, while $\mathbf{g}$ is called the **convolution kernel**. The kernel is usually given in a **normalized** form, i. e. $\sum_{j=0}^{n-1} |g_j| = 1$. If n is odd, $\mathbf{h}$ can be also computed as follows:

$$h_i := \begin{cases} \sum_{j=0}^{n-1} f_{i+j-\frac{n-1}{2}} \cdot g_j & \text{if } \frac{n-1}{2} \leq i < m - \frac{n-1}{2}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

In the formulas above, elements near the left and right ends of $\mathbf{h}$ do not necessarily have to be set to 0. In order to avoid treating these elements specially, the signal $\mathbf{f}$ needs to be "extended" in some way for all respective indexes outside the range $[0, m-1]$. There are many options for this: (a) "threshold" invalid indexes to stay within the valid range; (b) consider $\mathbf{f}$ as a periodic signal; (c) extrapolate missing signal values. Since projection vectors are definitely non-periodic (namely, there is no physical justification for them being treated as periodic measurements), and extrapolation would probably yield no additional gain, option (a) was finally chosen. Thus, missing elements f_i were assumed to equal either f_0 or f_{m-1}, depending on which end of the interval $[0, m-1]$ was closer to i.

Algorithm Entry Point

The execution of the algorithm starts at Function InitialConfiguration. The input of this function is the set $\{\vartheta_1, \ldots, \vartheta_n\}$ of projection angles, the input discrete sinogram $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$ (with all the $\mathbf{P}_{\vartheta_i}$ being 1D projections), the target number N of discs to be included in $\mathbf{c}_0$, and the LAC values constituting $\mathcal{A}$. (As mentioned earlier, $\mu_0 \equiv 0$ is assumed throughout the algorithm.) The function produces a valid configuration $\mathbf{c}_0$ containing exactly N discs.

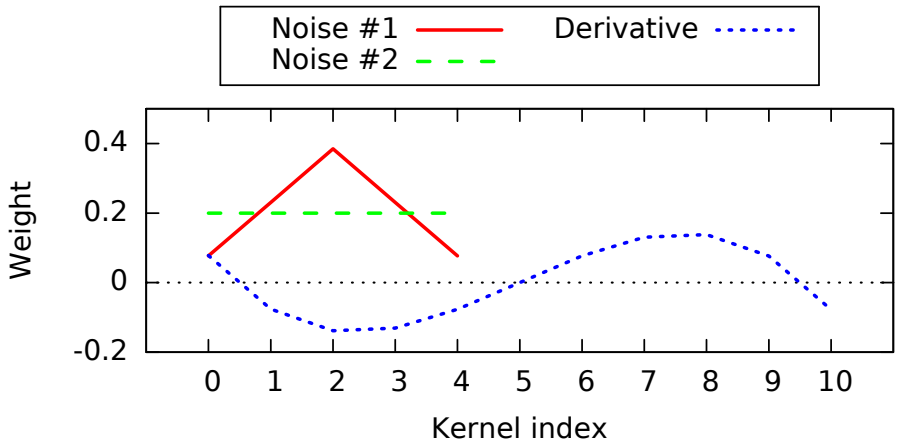


Figure E.1: Normalized convolution kernels used for noise filtering and for approximating the 1st numerical derivative.

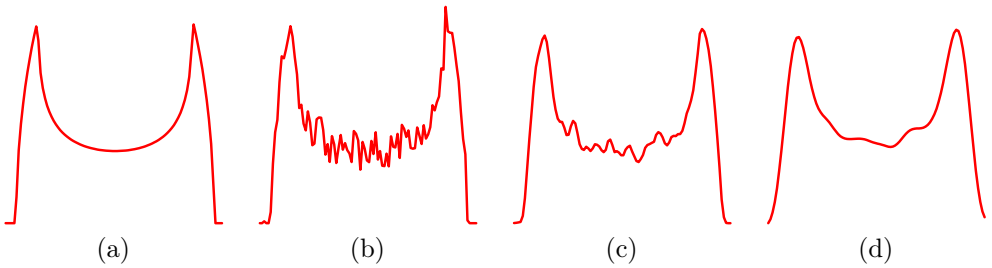


Figure E.2: Phases of noise filtering. **(a)** Noiseless original projection. **(b)** Original projection at 10 % noise level. **(c)** Result of the Gaussian filtering of (b). **(d)** Result of the averaging filtering of (c).

Locating the Annulus

The procedure sets off with trying to locate the annulus in the projections; see Function LocalizeRing. Even though the rest of the algorithm operates on the original—possibly noisy—projections, the localization of the annulus can be made more robust by first applying some noise filtering. $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$ will be, thus, subject to two phases of convolutions: first with kernel $\left(\frac{1}{13}, \frac{3}{13}, \frac{5}{13}, \frac{3}{13}, \frac{1}{13}\right)$—a rough approximation of a Gaussian—, then using the simple averaging kernel $\left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right)$ 5 times in succession. (Figure E.1 shows a plot of both aforementioned kernels.) As can be seen in Figure E.2, such a combination of filters is indeed effective in smoothing out projections even in the presence of noise.[1]

The outer boundaries of the ring are are determined by scanning the *original* (unfiltered) projections for the "leftmost" (s_1) and "rightmost" (s_4) value exceeding a given threshold t. As can be seen in the definition of the latter (see line 2 of Function LocalizeRing), the $\tau > 0$ real denotes the estimated noise level in agreement with the noise model presented earlier in Section 5.2.3.

The location of the inner boundaries of the ring (which happen to define the boundaries of

[1]It should be noted that, as a side-effect of smoothing, these filters may slightly alter the height—and shift the location—of local extrema.

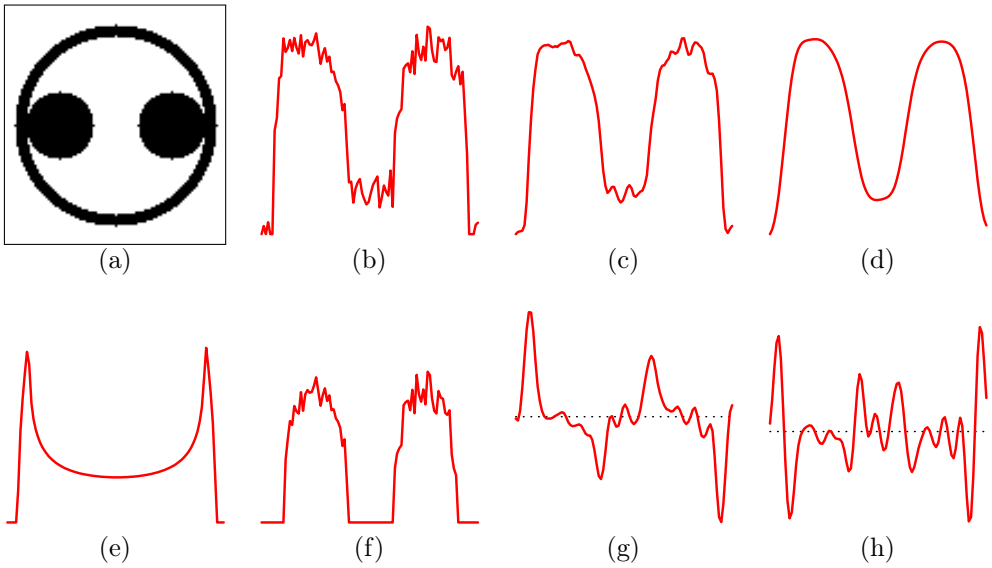


Figure E.3: Phases of locating the annulus. **(a)** Binary configuration to be reconstructed; $\mu_{\mathrm{R}} = \mu_{\mathrm{D}} = 1$, $\mu_{\mathrm{I}} = 0$. (Frame added for better visibility.) **(b)** Original vertical projection of (a) at 10 % noise level. **(c)** Result of the Gaussian filtering of (b). **(d)** Result of the averaging filtering of (c). **(e)** Noiseless simulated projection of the annulus and the interior determined from (b). **(f)** Projection remaining after the subtraction of (e) from (b). **(g)** 1st numerical derivative of (c). (Zero level shown as a dotted line.) **(h)** 1st numerical derivative of (g) (i. e. 2nd numerical derivative of (c)).

the interior as well) is slightly harder to find. Usually it is enough to look for the "leftmost" (s_2) and "rightmost" (s_3) local maxima of the noise-filtered projection (based on its first derivative), but this may not be sufficient in scenarios where a disc is tangent to (i. e. touches) the inner circle of the ring (see Figure E.3). This can be overcome by inspecting the second derivative to find the first inflection point (if such exists) of the filtered projection to the "left" and to the "right" of the local maxima. The numerical derivatives are approximated using the Savitzky–Golay convolution filter [159, Sections 5.7 and 14.8] [170, 171, 179]. (A plot of the associated kernel is depicted in Figure E.1.)

The aforementioned s indexes will then be used to compute some intermediate values that, for each ϑ_i, are saved in vectors EXTRADIUS, DIFFRADIUS and LINE (storing the approximate r_{RE}, $r_{\mathrm{RE}} - r_{\mathrm{RI}}$ and O_{RE} values determined for a given projection angle, respectively). Having processed all the projections, annulus parameters $(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$ will be obtained by averaging the intermediate values mentioned above (with the usual assumption that $O_{\mathrm{RE}} = O_{\mathrm{RI}}$).

Function LocalizeRing concludes with calculating the estimates $\{\hat{\mu}_{\mathrm{R}}, \hat{\mu}_{\mathrm{I}}\}$, as well as refining r_{RI} using model fitting. The exact purpose and explanation of these steps will be given later.

Locating Discs in Projections

The algorithm proceeds with the detection of projections of individual discs appearing in the unfiltered $\mathbf{P}_{\vartheta_i}$; see Function LocalizeDiscs. This first requires the "elimination" of the ring and

the interior from the input projections.[2] This is followed by the **normalization** of the remaining projection data via dividing each element by $(\mu_{\mathrm{D}} - \mu_{\mathrm{I}})$ (as per Section 5.2.3). The latter step ensures two properties: 1. the projection of any disc remains positive; 2. the projection of any disc appears as if it had been taken at $\mu_{\mathrm{D}} = 1$. Finally, the remaining projection is once more thresholded to get rid of any random spikes that presumably belong to the background. Any elements located outside the annulus are also zeroed out. The effect of these procedures is shown in Figure E.3(f).

After all this pre-processing, the remaining projection data is further inspected in an iterative fashion. In each iteration, the widest interval $[s_1, s_2] \subseteq \mathcal{D}_{\mathbf{P}}$ is chosen so that the restriction of $\mathbf{P}'_{\vartheta_i}$ to $[s_1, s_2]$ is all positive. The iteration is immediately aborted if no such interval exists. Otherwise, an attempt is made to find the parameters (O, r) of a candidate disc whose projection lies within $[s_1, s_2]$ and matches $\mathbf{P}'_{\vartheta_i}$ as closely as possible. Like in case of the annulus above, this is carried out using the technique model fitting (details are to follow later). If the fitting has been successful, (O, r) is added to vector DISC and the projection of the disc gets subtracted from $\mathbf{P}'_{\vartheta_i}$. If no candidate disc has been found, $[s_1, s_2]$ is deemed to contain only noise and discarded by setting the respective elements of $\mathbf{P}'_{\vartheta_i}$ to zero. This process continues until $\mathbf{P}'_{\vartheta_i}$ becomes constant 0.

The aforementioned procedure may occasionally detect more candidate discs than desired (e. g. due to the high amount of noise or other distortions). Such situations are resolved using a greedy strategy: only the N discs having the largest radius will be kept in DISC_i for any particular ϑ_i. It may as well happen that fewer than N discs are found for some projection angle; these cases are signaled to the user but not given any special attention at this point.

Back-projecting Candidate Discs

The algorithm continues with Function GetIntersections that, basically, provides a purely geometrical implementation for the back-projection algorithm. Specifically, each candidate disc (O, r) in DISC_i for some ϑ_i determines a line that passes through O and has a slope parallel to the projection lines in direction ϑ_i. For all $(\vartheta_i, \vartheta_j)$, $i \neq j$, the intersections of the respective lines defined so are formed and stored in the vector INTSECT. Each such intersection has, besides its coordinates, an associated radius that is computed as the minimum of the radii of the candidate discs involved, as well as a record of the corresponding set $\{\vartheta_i, \vartheta_j\}$ of directions.

In order to make the procedure more robust against noise effects, any two intersections (I_1, I_2) in INTSECT being sufficiently close to each other will be merged into a single one. This new intersection will be obtained as follows: its coordinates are computed as the average (centroid) of (I_1, I_2); its radius becomes the minimum of the respective radii of (I_1, I_2); and the set of projection angles will be formed by taking the union of the corresponding sets of (I_1, I_2). For consistency, I_1 and I_2 may be merged only if 1. their corresponding sets of projection angles are disjoint; or 2. for all common directions ϑ_i, I_1 and I_2 pass through the center of the same candidate disc in DISC_i.

Constructing the Initial Configuration

The algorithm ends with Function BuildInitialCfg whose task is to construct $\mathbf{c}_0$ based on the outcome of the earlier steps. First, the annulus is added to an empty $\mathbf{c}_0$, then the process proceeds in an iterative fashion. In each iteration, a new disc (O, r) is to be added to the current approximation of $\mathbf{c}_0$, the parameters of which are determined from INTSECT. Like before, the

[2] Please note that this step is actually carried out on a copy of $\mathbf{P}_{\vartheta_i}$; the input projections are never modified in any way!

choice of $I \in \text{INTSECT}$ is guided by a greedy strategy: the I associated with the most projection angles will be picked; if there are multiple such elements, the one having the largest associated radius will be preferred. (If the latter criterion still results in more than one elements, the one encountered the earliest in INTSECT will be picked.[3])

No matter how I gets chosen, it gets immediately removed from INTSECT and its coordinates and radius will become O and r of the trial disc, respectively. Disc (O, r) will then be tentatively added to $\mathbf{c}_0$. If the new configuration obtained so remains valid, (O, r) becomes finalized and every DISC_i involved gets updated by dropping all associated candidate discs. This, in turn, will be followed with the removal of all conflicting elements of INTSECT to keep $\mathbf{c}_0$ consistent with the input projections.

If the inclusion of (O, r) violates some geometrical constraint(s), r is gradually reduced by being multiplied with the $0 < \lambda < 1$ real. These adjustments are repeated until either $\mathbf{c}_0$ becomes valid, or r drops below $r_{\min}$. In the latter case, (O, r) is removed from $\mathbf{c}_0$ and the procedure starts over with selecting a new intersection.

The aforementioned process continues until $\mathbf{c}_0$ contains the desired number of discs (N). In less fortunate cases it may also happen, however, that INTSECT gets depleted before all discs could have been found. In order to resolve this, $\mathbf{c}_0$ will be completed with randomly generated discs by calling Procedure AddDisc discussed earlier in Appendix D.

Model Fitting

The aim of **data modeling** is to have a compact representation of some data—typically a measurement sample—by a set of free parameters $p_1, \ldots, p_j$ and some function $m_{p_1,\ldots,p_j}(x)$ depending on them. Here, the univariate (with x being the variable) function m is dubbed the **model** and $p_1, \ldots, p_j$ its parameters. (This definition of m can be extended to the multivariate case as well.) **Model fitting** [159, Chapter 15] is the process of finding the parameter values that yield the *best-fitting* m—the one that approximates the target data as closely as possible with respect to some measure of **fitness**.[4]

Model fitting is, thus, a special kind of optimization problem (with respect to the fitness of fitting m to the target data). While it can be also dealt with using general (global) optimization techniques, there are more sophisticated approaches that perform better (in terms of efficiency) for such problems. Of these, the author had first tried the Levenberg–Marquardt Method [159, Section 15.5], an iterative technique suitable for non-linear models and based on the second partial derivatives of the fitness function, and then an adaptation of the iterated conditional modes (ICM) [36], but eventually switched to the much simpler gradient descent [159, Section 10.6] [48] (also known as steepest descent). In both cases, the so-called chi-square measure [159, Section 15.1] (expressing the weighted least-squares error) was taken as fitness.[5]

As mentioned above, model fitting is utilized in several occasions during the execution of the algorithm producing $\mathbf{c}_0$. In all cases, the target data to fit against is the set $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$ of input projections, while the models represent the simulated projections of individual geometrical shapes constituting $\mathbf{c}_0$ (as per Section 5.2.3) and, hence, are all non-linear.

- Fitting the annulus (see line 19 of Function LocalizeRing): Aims to refine r_{RI} while $O_{\text{RE}} = O_{\text{RI}}$ and r_{RE} are held fixed. Starting at $r_{\text{RI}} := r_{\text{RE}} - 1$, r_{RI} is gradually decremented

[3] This is better then making an arbitrary choice among them, because this way the algorithm stays deterministic.

[4] Besides merely giving optimal parameter values, any model fitting procedure does also convey further details: whether or not—and to what extent—the chosen model is appropriate to describe the target data, and error in optimal parameter values. While very important in other applications, these aspects will be ignored for our purposes hereafter.

[5] For simplicity, the standard deviation was assumed constant 1 for each data point.

by 1 down to zero while collecting fitness values alongside in a vector $\mathbf{f}$. The optimal r_{RI} is then found by looking for the longest convergent leading subseries in $\mathbf{f}$ (whose limit is deemed r_{RI}). Depending on the user's choice, it is also possible to fit the LAC values $\{\mu_{\mathrm{R}}, \mu_{\mathrm{I}}\}$ within some range in tandem with r_{RI}.

- Estimating LAC values (see lines 18 and 20 of Function LocalizeRing): The goal is to give estimations $\{\hat{\mu}_{\mathrm{R}}, \hat{\mu}_{\mathrm{I}}\}$ for the materials forming the ring and its interior. The process is based on the "comparison" of input projections with the simulated projections of the ring and the interior using LAC values of unity (i. e. settings $\mu_{\mathrm{R}} = \mu_{\mathrm{D}} = 1$ and $\mu_{\mathrm{I}} = 0$ for computing $\hat{\mu}_{\mathrm{R}}$, and settings $\mu_{\mathrm{R}} = \mu_{\mathrm{D}} = 0$ and $\mu_{\mathrm{I}} = 1$ for $\hat{\mu}_{\mathrm{I}}$). By "comparison" we mean calculating the quotient of respective projections (namely, the input divided by the simulated one); the piecewise quotients are stored in a vector $\mathbf{q}$. The estimations $\hat{\mu}_{\mathrm{R}}$ and $\hat{\mu}_{\mathrm{I}}$ are then found by sorting $\mathbf{q}$ in ascending order and looking for the leftmost longest convergent subseries. While not actually used anywhere during the optimization of $\gamma(f_{\mathbf{c}})$, these estimations are quite useful for physical measurements where $\{\mu_{\mathrm{R}}, \mu_{\mathrm{I}}\}$ may not be known to sufficient accuracy.

  - Note: The estimation $\hat{\mu}_{\mathrm{D}}$ still needs to be determined manually; this is easy if a projection is known (or suspected) to contain an isolated candidate disc.

- Fitting candidate discs in $\mathbf{P}_{\vartheta_i}$ (see line 9 of Function LocalizeDiscs): Strives to determine the parameters (O, r) of a candidate disc whose simulated projection best-fits $\mathbf{P}_{\vartheta_i}$ over some interval $[s_1, s_2]$ while keeping μ_{D} fixed. As a matter of fact, the fitting always fails when $s_2 - s_1 < 2r_{\min}$. Otherwise, fitting is attempted using either of three approaches (in this order):

  - Covering ("spanning") the whole $[s_1, s_2]$ with a single disc, so that $O := \frac{s_1+s_2}{2}$ and $\frac{s_2-s_1}{2} - 2 \leq r \leq \frac{s_2-s_1}{2}$ (see Figure E.4(b)).
  - Individually fitting over all 6-element-wide sub-intervals of $[s_1, s_2]$, and looking for the leftmost (i. e. closest to s_1) longest convergent subseries consisting of at least 3 elements (see Figure E.4(d)–(k)).
  - Covering a "left" sub-interval of $[s_1, s_2]$ with a single disc having as large a radius r as possible (see Figure E.4(c)). That is, starting with $r := r_{\min}$ and going up to $\frac{s_2-s_1}{2}$ in increments of 1, O is always set to $s_1 + r$. The optimal r is found as the largest trial value so that the simulated projection of (O, r) best-fits $\mathbf{P}_{\vartheta_i}$ over $[s_1, s_1 + 2r]$.

Extension to 3D

Though it would be theoretically possible to generalize and extend the aforementioned algorithm to 3D configurations and 2D input projections, this had not been attempted. Instead, the 3D case was reduced to the individual 2D sub-problems corresponding to cross-sections taken perpendicular to the axis of rotation of the object. The exact details of this process are described in Section 4.5.

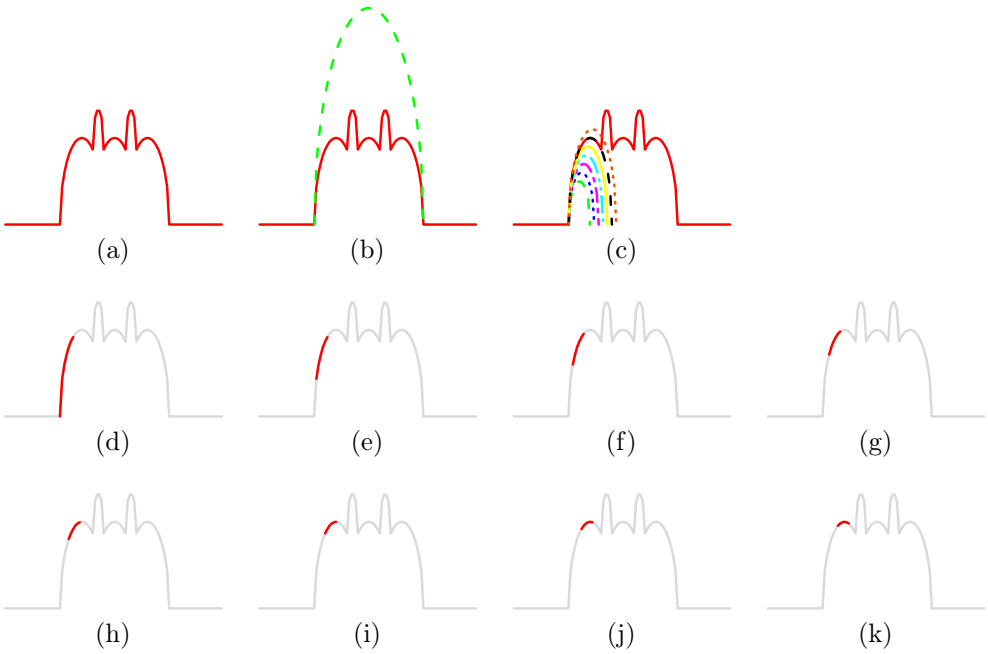


Figure E.4: Fitting candidate discs. **(a)** Noiseless pre-processed input projection after the elimination of the annulus and its interior, normalization and noise thresholding. The interval $[s_1, s_2]$ under inspection contains most of the projection except for the constant zero parts on the left and right side. **(b)** A failed attempt to cover ("span") the whole $[s_1, s_2]$ in (a) with a single disc (drawn with a green dashed line). **(c)** Covering the left sub-intervals of $[s_1, s_2]$ in (a) with successively larger discs (drawn with dashed and dotted lines). The second largest disc drawn is the optimal fit. **(d)**–**(k)** Fitting over 6-element-wide sub-intervals of $[s_1, s_2]$ in (a) (drawn in red in front of the rest of the input projection shown in light-gray). The leftmost 8 such sub-intervals are shown here; together they will define an optimum quite close to the one found in (c).

Function InitialConfiguration($\{\vartheta_1, \ldots, \vartheta_n\}$, $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$, N, $\mathcal{A}$)

Input : set of projection angles ($\{\vartheta_1, \ldots, \vartheta_n\}$), input discrete sinogram ($\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$), number of discs the constructed configuration shall contain (N), LAC values ($\mathcal{A} = \{\mu_0, \mu_\mathrm{R}, \mu_\mathrm{I}, \mu_\mathrm{D}\}$)
Output : initial configuration

`/* Assumption:` $\mu_0 \equiv 0$ `*/`

1 $\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$:= a perfect copy of $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$
2 $(O_\mathrm{RE}, r_\mathrm{RE}, O_\mathrm{RI}, r_\mathrm{RI})$:= LocalizeRing($\{\vartheta_1, \ldots, \vartheta_n\}$, $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$, $\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$, μ_R, μ_I)
3 $\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$:= a perfect copy of $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$ (to counteract any changes made by LocalizeRing)
4 DISC := LocalizeDiscs($\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$, N, $\mathcal{A}$, $(O_\mathrm{RE}, r_\mathrm{RE}, O_\mathrm{RI}, r_\mathrm{RI})$)
5 INTSECT := GetIntersections($\{\vartheta_1, \ldots, \vartheta_n\}$, DISC)
6 **return** BuildInitialCfg(N, $(O_\mathrm{RE}, r_\mathrm{RE}, O_\mathrm{RI}, r_\mathrm{RI})$, DISC, INTSECT)

Function LocalizeRing($\{\vartheta_1, \ldots, \vartheta_n\}$, $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$, $\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$, μ_{R}, μ_{I})

Input : set of projection angles ($\{\vartheta_1, \ldots, \vartheta_n\}$), input discrete sinogram ($\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$), copy of input discrete sinogram ($\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$), LAC of the annulus ($\mu_{\mathrm{R}}$), LAC of the interior ($\mu_{\mathrm{I}}$)

Output : parameters of the annulus

Let EXTRADIUS, DIFFRADIUS and LINE be n-element vectors

`/* Determine external and internal boundaries of the ring */`

$t \coloneqq \tau \cdot \max_{s,i} \mathbf{P}_{\vartheta_i}(s)$, where $\tau \in \mathbb{R}^+$ is the estimated noise level

for $i \coloneqq 1$ **to** n **do**

- $s_1 \coloneqq$ the smallest s so that $\mathbf{P}_{\vartheta_i}(s) > t$
- $s_4 \coloneqq$ the largest s so that $\mathbf{P}_{\vartheta_i}(s) > t$
- Apply noise filtering to $\mathbf{P}'_{\vartheta_i}$ (Gaussian and averaging kernels)
- Compute first and second numerical derivatives of $\mathbf{P}'_{\vartheta_i}$
- $s_2 \coloneqq$ the smallest $s > s_1$ associated with a local maximum or inflection point in $\mathbf{P}'_{\vartheta_i}$, whichever happens earlier
- $s_3 \coloneqq$ the largest $s < s_4$ associated with a local maximum or inflection point in $\mathbf{P}'_{\vartheta_i}$, whichever happens later
- $\mathrm{EXTRADIUS}_i \coloneqq \frac{s_4 - s_1}{2}$
- $\mathrm{DIFFRADIUS}_i \coloneqq \frac{(s_2 - s_1) + (s_4 - s_3)}{2}$
- $\mathrm{LINE}_i \coloneqq$ projection line in $\mathbf{P}'_{\vartheta_i}$ associated with $\frac{s_1 + s_4}{2}$

`/* Compute ring parameters */`

$r_{\mathrm{RE}} \coloneqq$ average of EXTRADIUS

$d \coloneqq$ average of DIFFRADIUS

$r_{\mathrm{RI}} \coloneqq r_{\mathrm{RE}} - d$ `/* Temporary setting */`

$O_{\mathrm{RE}} \coloneqq$ average (centroid) of all the pairwise line intersections of LINE

$O_{\mathrm{RI}} \coloneqq O_{\mathrm{RE}}$

Calculate estimated LAC values $\{\hat{\mu}_{\mathrm{R}}, \hat{\mu}_{\mathrm{I}}\}$ based on $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$ and $(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$

Determine r_{RI}—and possibly $\{\mu_{\mathrm{R}}, \mu_{\mathrm{I}}\}$—more precisely using model fitting based on $\left(\mathbf{P}_{\vartheta_1}, \ldots, \mathbf{P}_{\vartheta_n}\right)$, $(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$, μ_{R} and μ_{I}

Re-calculate estimated LAC values $\{\hat{\mu}_{\mathrm{R}}, \hat{\mu}_{\mathrm{I}}\}$

return $(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$

Function LocalizeDiscs($\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$, N, $\mathcal{A}$, $(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$)

Input : copy of input discrete sinogram ($\left(\mathbf{P}'_{\vartheta_1}, \ldots, \mathbf{P}'_{\vartheta_n}\right)$), number of discs the constructed configuration shall contain (N), LAC values ($\mathcal{A} = \{\mu_0, \mu_{\mathrm{R}}, \mu_{\mathrm{I}}, \mu_{\mathrm{D}}\}$), parameters of the annulus ($(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$)
Output : disc parameters for each projection

DISC $\coloneqq$ n-element vector of empty lists
$t \coloneqq \tau \cdot \max_{s,i} \mathbf{P}'_{\vartheta_i}(s)$, where $\tau \in \mathbb{R}^+$ is the estimated noise level
for $i \coloneqq 1$ **to** n **do**
    `/* Pre-process` $\mathbf{P}'_{\vartheta_i}$ `*/`
    Subtract the respective projection of the ring and the interior from $\mathbf{P}'_{\vartheta_i}$
    Normalize $\mathbf{P}'_{\vartheta_i}$ (divide each element by $(\mu_{\mathrm{D}} - \mu_{\mathrm{I}})$)
    Set each $\mathbf{P}'_{\vartheta_i}(s) \leq \frac{t}{|\mu_{\mathrm{D}} - \mu_{\mathrm{I}}|}$ to zero, as well as those located outside the ring
    `/* Detect discs */`
    while $\mathbf{P}'_{\vartheta_i}$ is not constant 0 **do**
        Let $[s_1, s_2] \subseteq \mathcal{D}_{\mathbf{P}}$ be the widest interval so that $\forall s \in [s_1, s_2] : \mathbf{P}'_{\vartheta_i}(s) > 0$
        Try to find the projection of a candidate disc (O, r) in $\mathbf{P}'_{\vartheta_i}$ restricted to $[s_1, s_2]$ using model fitting; here O essentially becomes the s in $\mathbf{P}'_{\vartheta_i}$ associated with the middle projection line passing through the disc
        if Candidate (O, r) found **then**
            Subtract the projection of disc (O, r) from $\mathbf{P}'_{\vartheta_i}$, then set all negative elements of $\mathbf{P}'_{\vartheta_i}$ to zero
            if $r \geq r_{\min}$ **then**
                Add (O, r) to DISC_i
        else
            `/*` $[s_1, s_2]$ `contains no useful data, so let us discard it */`
            Set $\mathbf{P}'_{\vartheta_i}(s)$ to zero for all $s \in [s_1, s_2]$
    Sort DISC_i by r descending
    Keep the first N elements in DISC_i (i. e. those having the largest radii)
return DISC

Function GetIntersections($\{\vartheta_1, \ldots, \vartheta_n\}$, DISC)

Input : set of projection angles ($\{\vartheta_1, \ldots, \vartheta_n\}$), disc parameters for each projection (DISC)
Output : intersections (candidate disc centers, along with associated radii) formed from DISC

INTSECT $\coloneqq$ empty list
for $i \coloneqq 1$ **to** $n - 1$ **do**
    for $j \coloneqq i + 1$ **to** n **do**
        Add all pairwise intersections of the middle projection lines of discs in DISC_i and DISC_j to INTSECT, along with the minimum of the radii of the discs involved
Merge intersections in INTSECT whose distance is less than $2 \cdot r_{\min}$; the new intersection will possess the average (centroid) of the coordinates and the minimum of the radii of the original intersections involved
return INTSECT

Function BUILDINITIALCFG(N, $(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$, DISC, INTSECT)

Input : number of discs the constructed configuration shall contain (N), parameters of the annulus ($(O_{\mathrm{RE}}, r_{\mathrm{RE}}, O_{\mathrm{RI}}, r_{\mathrm{RI}})$), disc parameters for each projection (DISC), intersections (INTSECT)
Output : initial configuration

```
1  c := empty configuration
2  Add ring (O_RE, r_RE, O_RI, r_RI) to c
3  i := 1
4  while i ≤ N and INTSECT ≠ ∅ do
5      I := the element of INTSECT associated with the most projections and having the largest
       corresponding disc
6      Remove I from INTSECT
7      (O_i, r_i) := the disc determined by I
8      Add candidate disc (O_i, r_i) to c                          /* Might be temporary */
9      while c violates geometric constraints and r_i ≥ r_min do
10         r_i := λ r_i , where 0 < λ < 1 is the factor of adjustment
11         Update r_i in c
12     if r_i < r_min then
13         Remove invalid candidate disc (O_i, r_i) from c
14     else
           /* Found a new disc (O_i, r_i) in c */
15         for j := 1 to n do
16             if I is associated with P_{ϑ_j} then
17                 (O, r) := the disc in DISC_j with the largest radius and whose middle projection line is
                   associated with I
18                 Remove disc (O, r) from DISC_j
19                 if DISC_j contains no more discs with center O then
20                     Update all elements of INTSECT associated with the middle projection line of
                       (O, r), so that they will no longer be associated with P_{ϑ_j}
21                     Remove all elements of INTSECT that are only associated with a single projection
22         i := i + 1
23 if i ≤ N then
       /* Partial configuration; add missing discs randomly */
24     ADDDISC(c, N − (i − 1))
25 return c
```

Appendix F

Additional Reconstruction Results for Objects Parametrized with Geometrical Primitives

This appendix shows some additional reconstructions[1] of software and physical phantoms for parametric objects. It complements the results presented in Sections 5.3 and 6.2.

[1]The results of Test Cases III and IV have not been published; they were obtained as an extension and improvement upon the results published earlier.

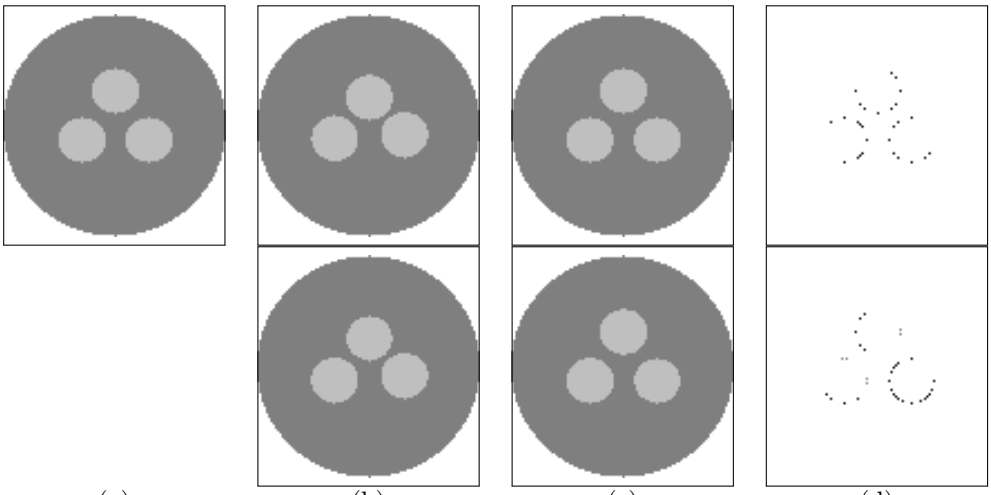


Figure F.1: Reconstruction of a 2D software phantom from 4 projections at 0 % and 10 % noise levels. (Frame added for better visibility.) Top row: $\mathrm{FOM}_{\mathrm{RME}} = 2.217\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.065\,\%$. Bottom row: $\mathrm{FOM}_{\mathrm{RME}} = 2.613\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.169\,\%$. Image arrangement as for Figure 5.10.

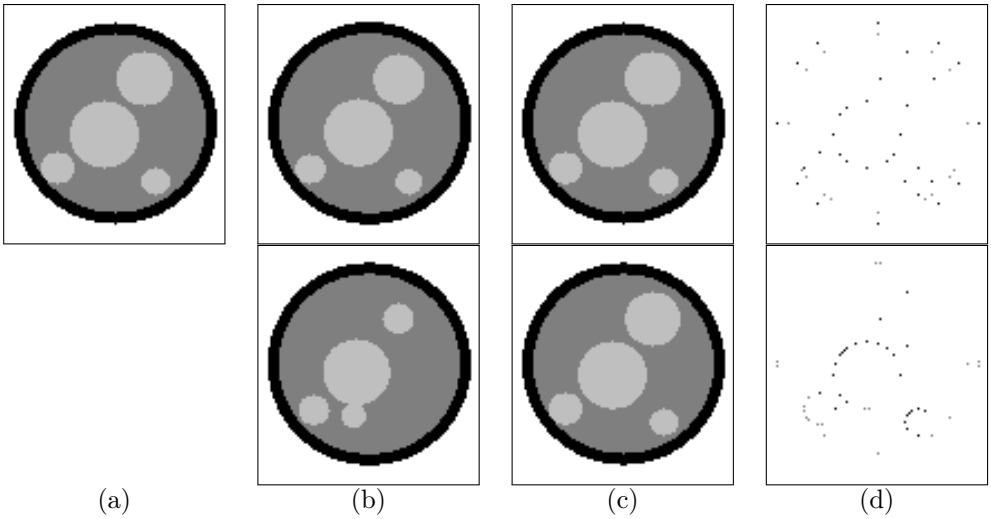
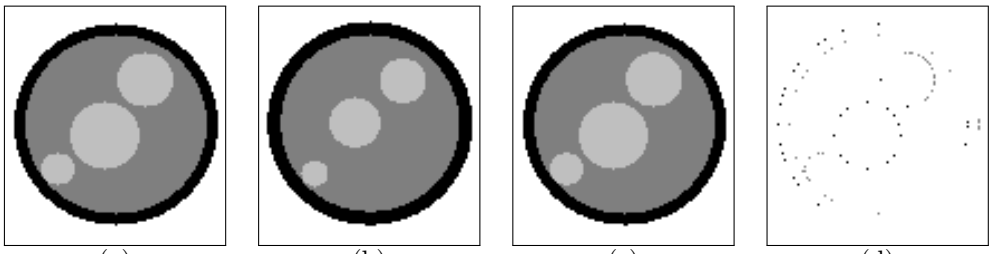
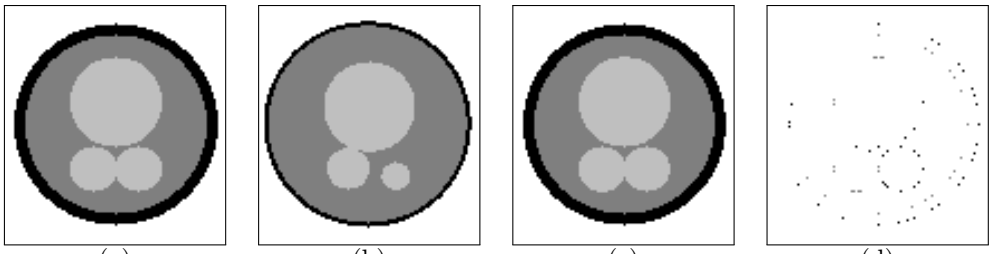


Figure F.2: Reconstruction of a 2D software phantom from 4 projections at 0 % and 10 % noise levels. (Frame added for better visibility.) Top row: $\mathrm{FOM}_{\mathrm{RME}} = 1.747\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.051\,\%$. Bottom row: $\mathrm{FOM}_{\mathrm{RME}} = 1.71\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.158\,\%$. Image arrangement as for Figure 5.10.



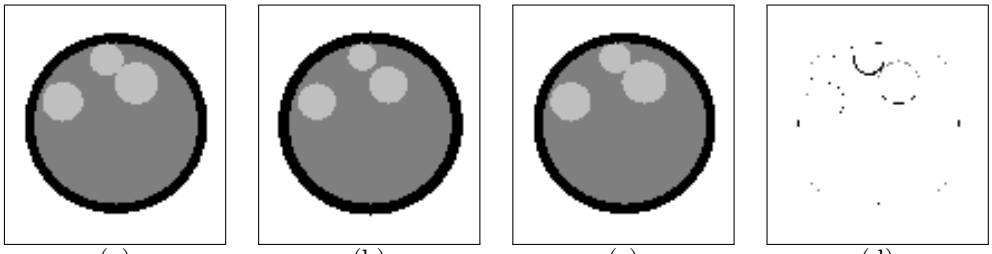
(a) (b) (c) (d)

Figure F.3: Reconstruction of a 2D software phantom from 4 projections at 10 % noise level; $\mathrm{FOM}_{\mathrm{RME}} = 2.884\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.171\,\%$. (Frame added for better visibility.) **(a)** Original configuration. **(b)** Initial configuration. **(c)** Reconstructed configuration. **(d)** Difference of (a) and (c).



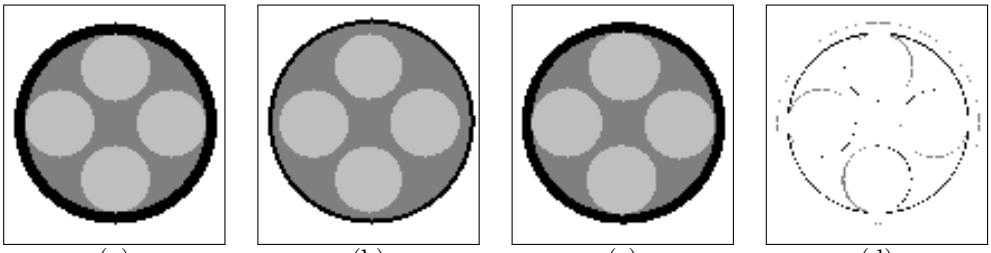
(a) (b) (c) (d)

Figure F.4: Reconstruction of a 2D software phantom from 4 projections at 10 % noise level; $\mathrm{FOM}_{\mathrm{RME}} = 2.294\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.166\,\%$. (Frame added for better visibility.) Image arrangement as for Figure F.3.



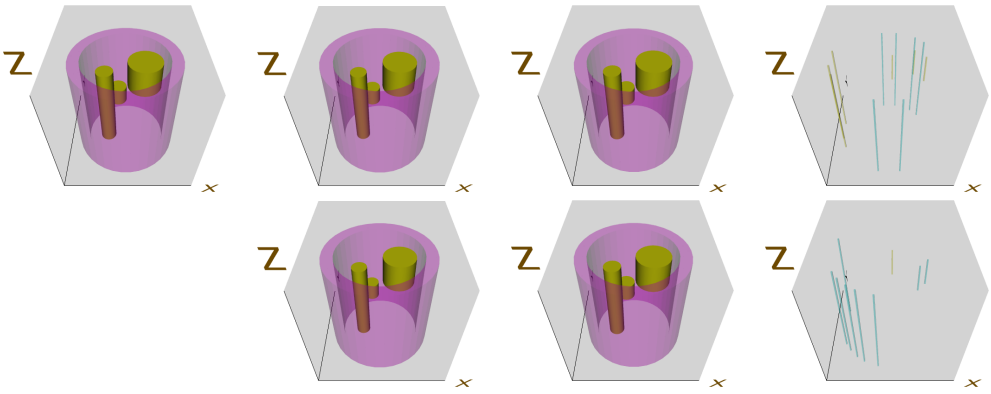
(a) (b) (c) (d)

Figure F.5: Reconstruction of a 2D software phantom from 4 projections at 10 % noise level; $\mathrm{FOM}_{\mathrm{RME}} = 3.952\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.372\,\%$. (Frame added for better visibility.) Image arrangement as for Figure F.3.



(a) (b) (c) (d)

Figure F.6: Reconstruction of a 2D software phantom from 4 projections at 10 % noise level; $\mathrm{FOM}_{\mathrm{RME}} = 7.79\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.585\,\%$. (Frame added for better visibility.) Image arrangement as for Figure F.3.



(a) (b) (c) (d)

Figure F.7: Reconstruction of a 3D software phantom containing cylinders from 4 projections at 0 % and 10 % noise levels. Top row: noiseless projections; $\mathrm{FOM}_{\mathrm{RME}} = 0.459\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.32\,\%$. Bottom row: 10 % noise level; $\mathrm{FOM}_{\mathrm{RME}} = 0.32\,\%$, $\mathrm{FOM}_{\mathrm{CD}} = 0.349\,\%$. **(a)** Original configuration. **(b)** Initial configuration. **(c)** Reconstructed configuration. **(d)** Difference of (a) and (c).

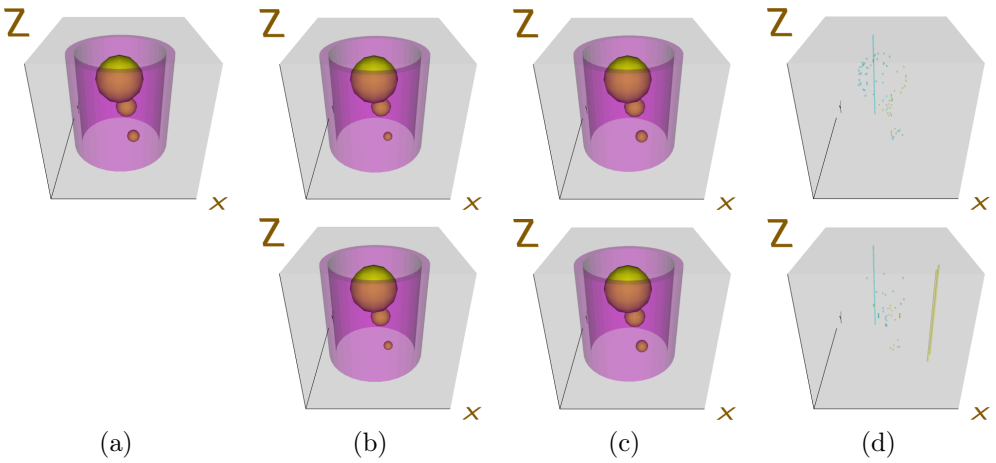


Figure F.8: Reconstruction of a 3D software phantom containing spheres from 4 projections at 0 % and 10 % noise levels. Top row: $\mathrm{FOM_{RME}} = 0.129\,\%$, $\mathrm{FOM_{CD}} = 0.027\,\%$. Bottom row: $\mathrm{FOM_{RME}} = 0.226\,\%$, $\mathrm{FOM_{CD}} = 0.307\,\%$. Image arrangement as for Figure F.7.

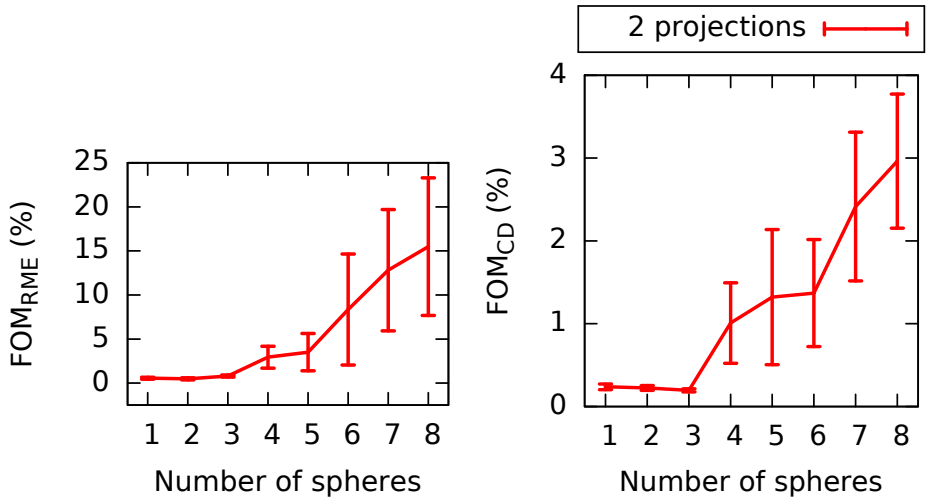
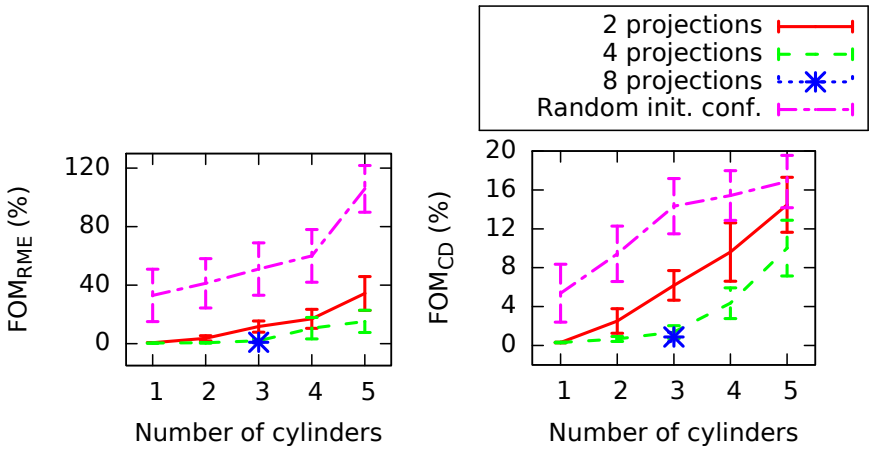
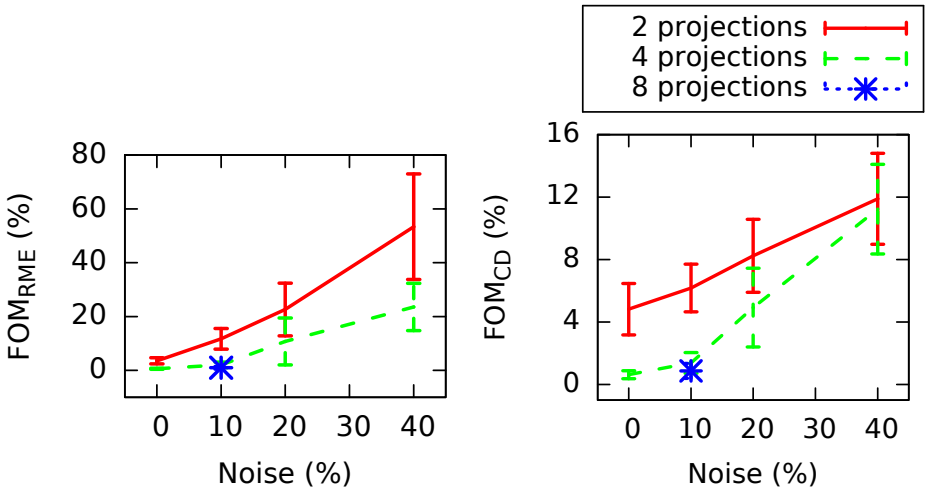


Figure F.9: Quality of the reconstructions for 3D configurations containing spheres at 10 % noise level as the function of geometrical complexity, based on 10 repetitions. Error bars indicate the standard error. Smaller values correspond to better results. **(a)** $\mathrm{FOM_{RME}}$ versus the number of spheres. **(b)** $\mathrm{FOM_{CD}}$ versus the number of spheres.



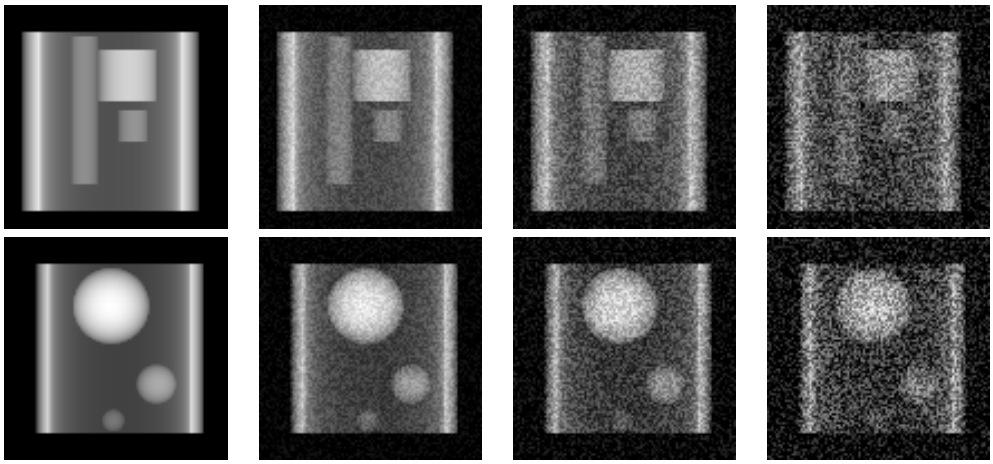
(a) (b)

Figure F.10: Quality of the reconstructions for 3D configurations containing cylinders at 10 % noise level as the function of geometrical complexity, based on 10 repetitions. Error bars indicate the standard error. Smaller values correspond to better results. **(a)** $\mathrm{FOM}_{\mathrm{RME}}$ versus the number of cylinders. **(b)** $\mathrm{FOM}_{\mathrm{CD}}$ versus the number of cylinders.



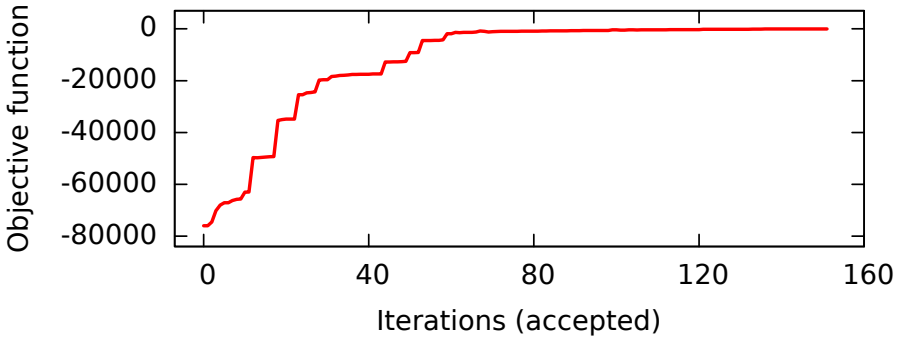
(a) (b)

Figure F.11: Quality of the reconstructions for 3D configurations containing 3 cylinders as the function of noise level, based on 10 repetitions. Error bars indicate the standard error. Smaller values correspond to better results. **(a)** $\mathrm{FOM}_{\mathrm{RME}}$ versus the level of noise. **(b)** $\mathrm{FOM}_{\mathrm{CD}}$ versus the level of noise.

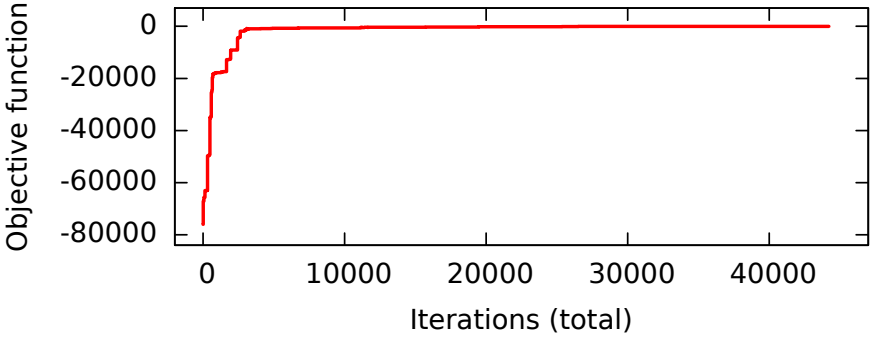


(a) (b) (c) (d)

Figure F.12: Effects of the noise level on simulated 2D projections. Top row: 3D configuration containing cylinders. Bottom row: 3D configuration containing spheres. **(a)** Noiseless original projection. **(b)** Original projection at 10 % noise level. **(c)** Original projection at 20 % noise level. **(d)** Original projection at 40 % noise level.



(a)



(b)

Figure F.13: Speed of convergence of $\gamma(f_{\mathbf{c}})$ as the function of iterative steps. Reconstruction of the 3D software phantom shown in Figure F.8 using noiseless projections. **(a)** Exponent of $\gamma(f_{\mathbf{c}})$ versus the number of accepted candidate configurations $\mathbf{c}'$ (out of 44 220 iterations in total). **(b)** Exponent of $\gamma(f_{\mathbf{c}})$ versus the total number of iterations.

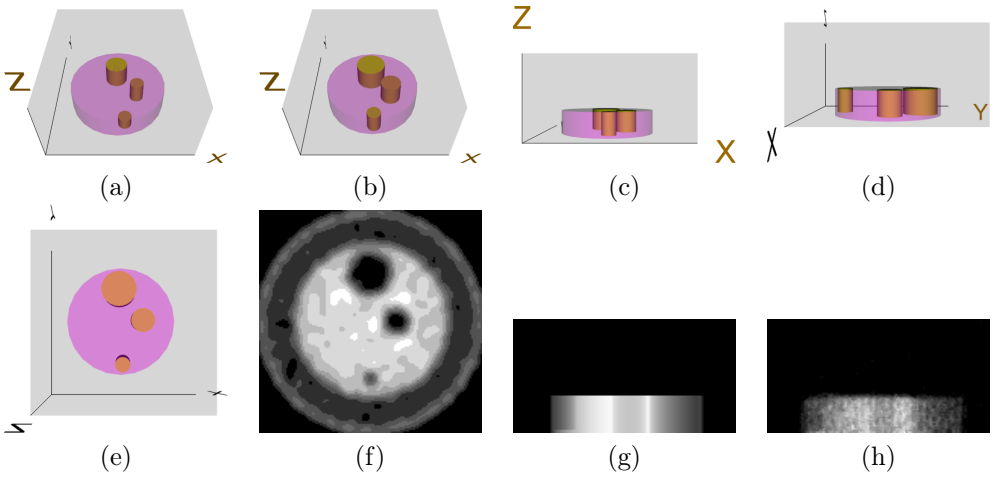


Figure F.14: Reconstruction of Test Case III from 4 projections. **(a)** Initial configuration. **(b)**–**(e)** Different views of the reconstructed configuration. **(f)** FBP reconstruction of a cross-section. **(g)** Simulated projection of (b)–(e) at $\vartheta = 0°$. **(h)** Input projection at $\vartheta = 0°$. (Images (f) and (h) are taken from Figure 6.4.)

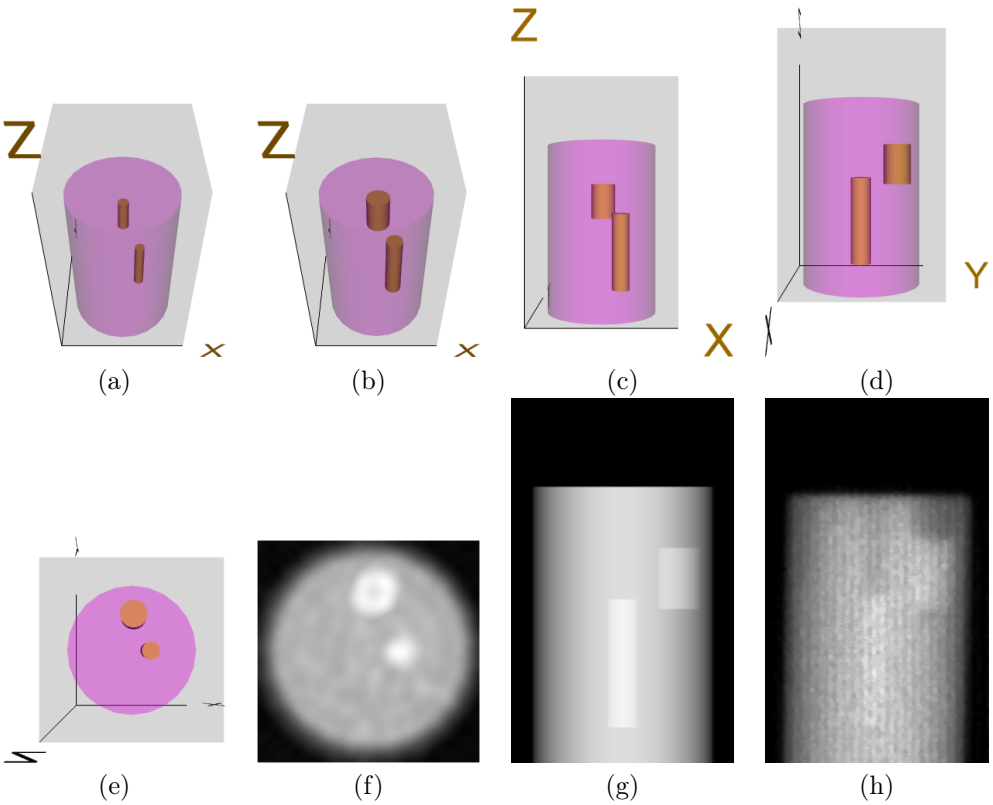


Figure F.15: Reconstruction of the lead-containing holes of Test Case IV from 4 projections. **(a)** Initial configuration. **(b)**–**(e)** Different views of the reconstructed configuration. **(f)** FBP reconstruction of the cross-section marked in Figure 6.5(a) with red lines. **(g)** Simulated projection of (b)–(e) at $\vartheta = 0°$. **(h)** Input projection at $\vartheta = 0°$. (Images (f) and (h) are taken from Figure 6.5.)

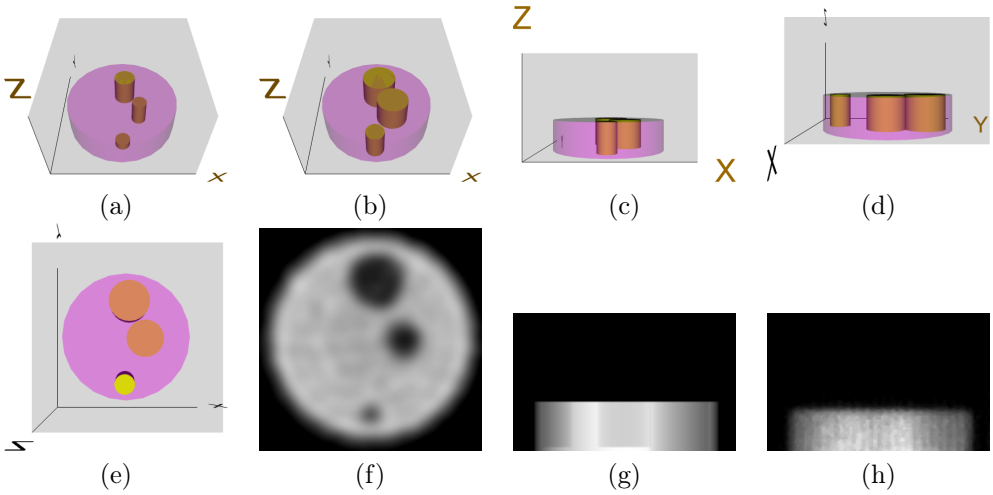


Figure F.16: Reconstruction of the air-filled holes of Test Case IV from 4 projections. **(a)** Initial configuration. **(b)**–**(e)** Different views of the reconstructed configuration. **(f)** FBP reconstruction of a cross-section. **(g)** Simulated projection of (b)–(e) at $\vartheta = 0°$. **(h)** Input projection at $\vartheta = 0°$. (Images (f) and (h) are taken from Figure 6.5.)

Appendix G

Additional Reconstruction Results for Deformed Polycrystalline Samples

This appendix shows some additional reconstructions of polycrystalline software phantoms using $\gamma(o)$ and $\gamma(g, o)$ from noiseless as well as noisy projections.[1] It complements the results presented in Section 8.3.

[1] The results using $\gamma(o)$ and those of Test Case V have not been published; they were obtained as an extension and improvement upon the results published earlier.

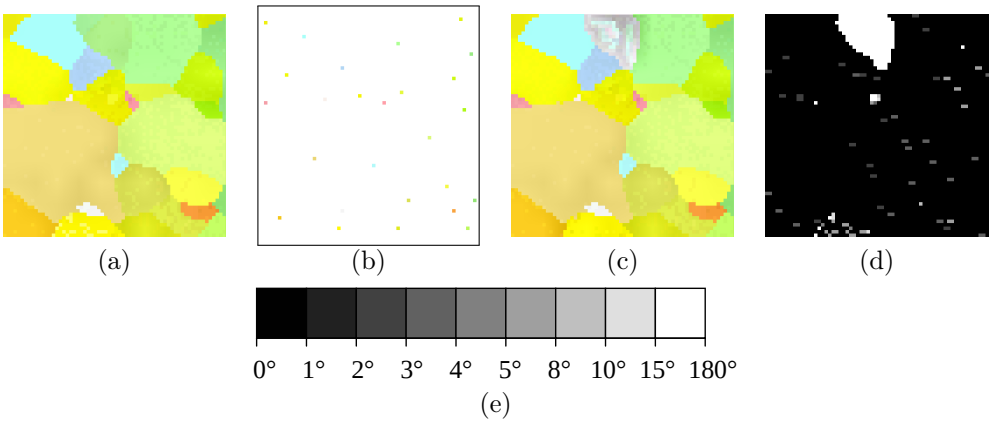


Figure G.1: The reconstruction of Test Case II using $\gamma(o)$ and noiseless projections; $\mathrm{FOM}_o = 0.972$. Map arrangement and gray scales as for Figure 8.7.

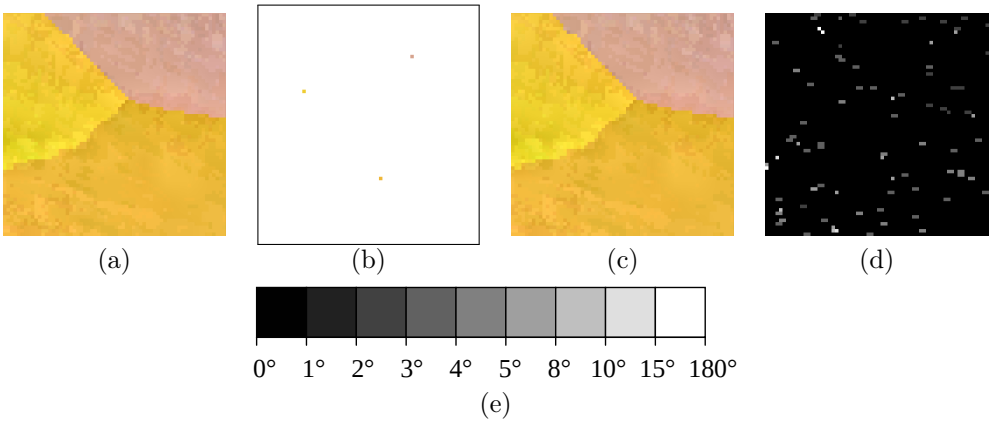


Figure G.2: The reconstruction of Test Case III using $\gamma(o)$ and noiseless projections; $\mathrm{FOM}_o = 0.9996$. Map arrangement and gray scales as for Figure 8.7.

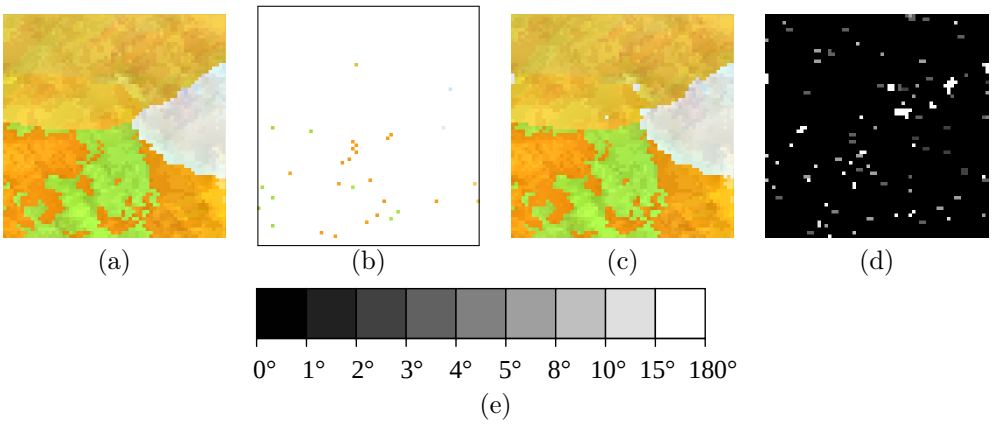


Figure G.3: The reconstruction of Test Case IV using $\gamma(o)$ and noiseless projections; $\mathrm{FOM}_o = 0.996$. Map arrangement and gray scales as for Figure 8.7.

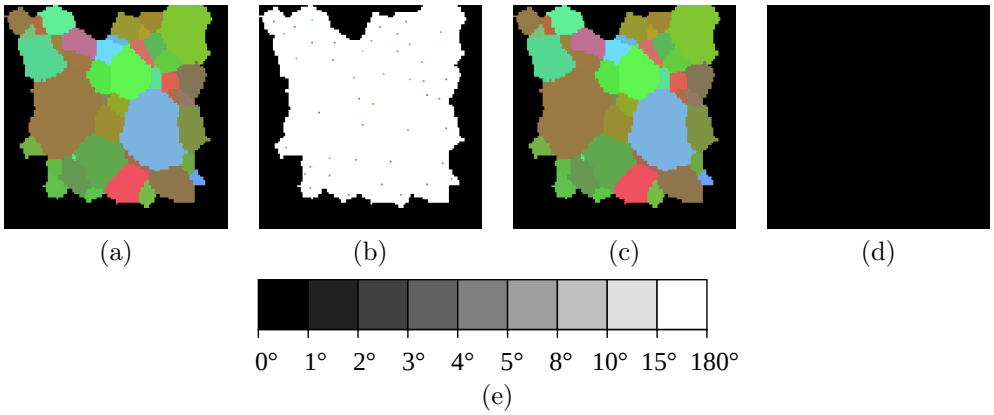


Figure G.4: The reconstruction of Test Case V using $\gamma(o)$ and noiseless projections (original color mapping); $\mathrm{FOM}_o = 1$. Map arrangement and gray scales as for Figure 8.7. Black pixels in orientation maps represent void regions.

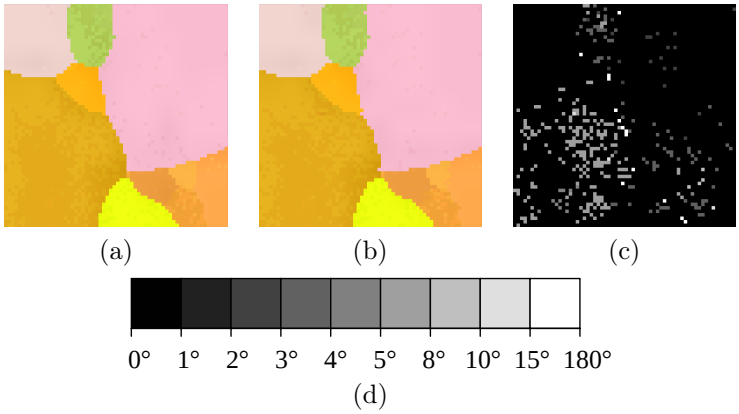


Figure G.5: The reconstruction of Test Case I using $\gamma(o)$ at 100 % noise level; $\mathrm{FOM}_o = 0.997$. **(a)** Reference orientation map (contrast-enhanced). **(b)** Reconstructed orientation map (contrast-enhanced). **(c)** Difference of the reference and the reconstructed orientation maps. The intensity of the pixels is determined by the distance (disorientation angle) of corresponding orientation pairs, as shown in (d).

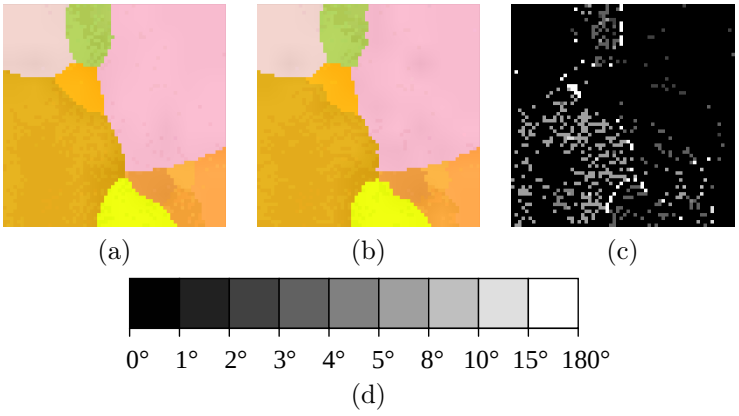


Figure G.6: The reconstruction of Test Case I using $\gamma(o)$ at 200 % noise level; FOM_o = 0.995. Map arrangement and gray scales as for Figure G.5.

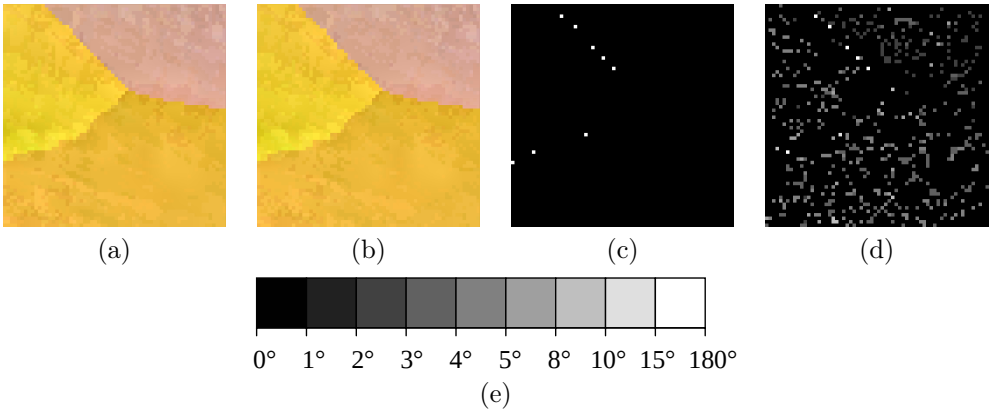


Figure G.7: The reconstruction of Test Case III using $\gamma(g, o)$ at 100 % noise level; FOM_g = 0.998, FOM_o = 0.9988. Map arrangement and gray scales as for Figure 8.14.

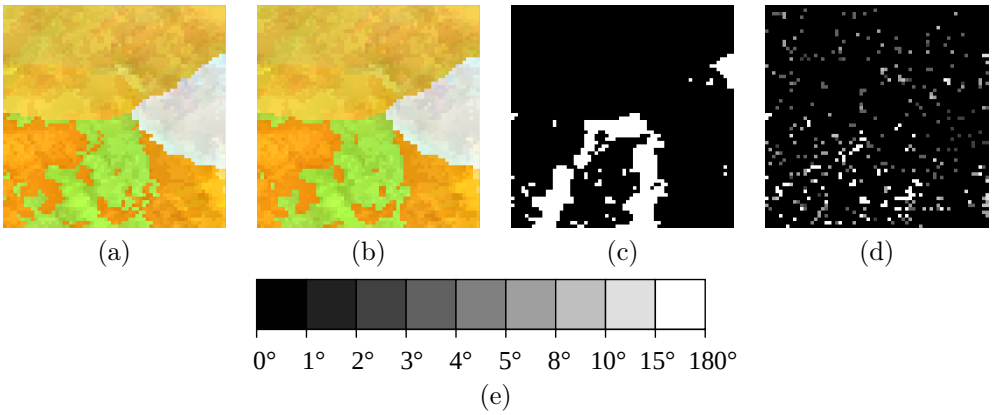


Figure G.8: The reconstruction of Test Case IV using $\gamma(g, o)$ at 100 % noise level; FOM_g = 0.873, FOM_o = 0.989. Map arrangement and gray scales as for Figure 8.14. (Note: (c) is irrelevant due to the lack of a meaningful definition of grains in this case; see text.)

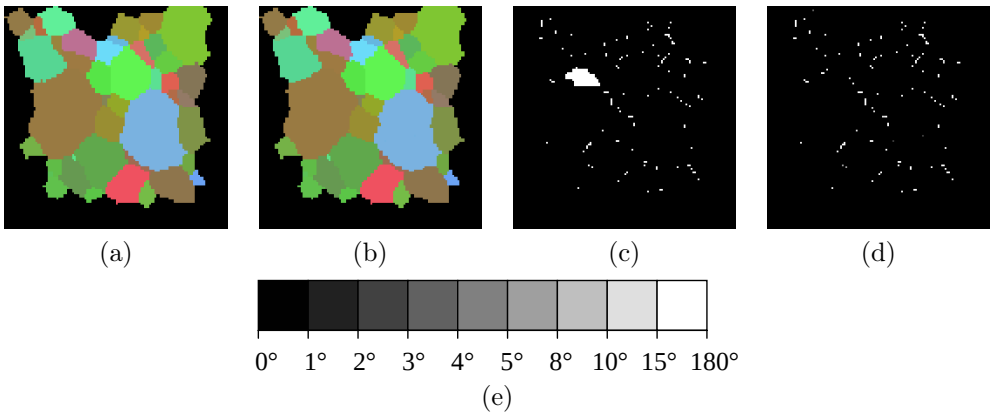


Figure G.9: The reconstruction of Test Case V using $\gamma(g, o)$ at 100 % noise level (original color mapping); $\mathrm{FOM}_g = 0.975$, $\mathrm{FOM}_o = 0.997$. Map arrangement and gray scales as for Figure 8.14. Black pixels in orientation maps represent void regions.

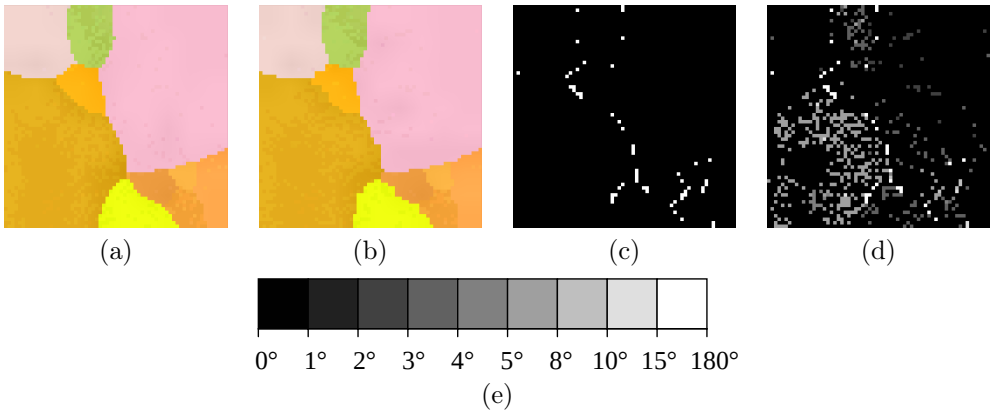


Figure G.10: The reconstruction of Test Case I using $\gamma(g, o)$ at 200 % noise level; $\mathrm{FOM}_g = 0.987$, $\mathrm{FOM}_o = 0.995$. Map arrangement and gray scales as for Figure 8.14.

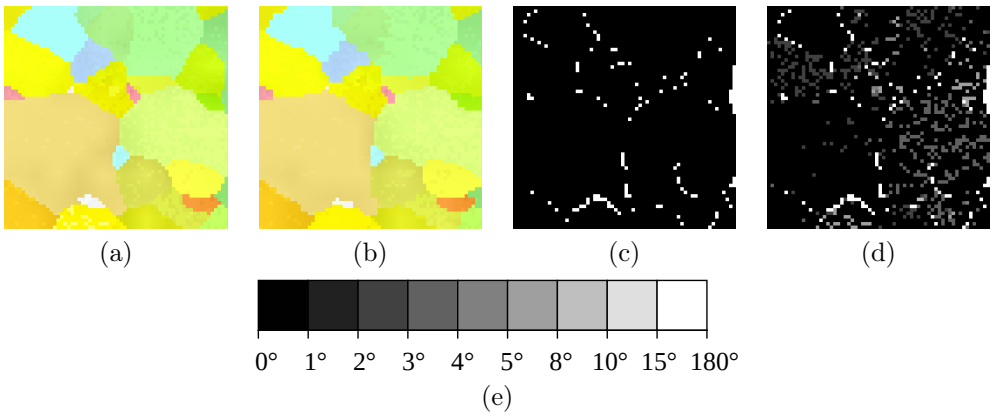


Figure G.11: The reconstruction of Test Case II using $\gamma(g, o)$ at 200 % noise level; $\mathrm{FOM}_g = 0.971$, $\mathrm{FOM}_o = 0.991$. Map arrangement and gray scales as for Figure 8.14.

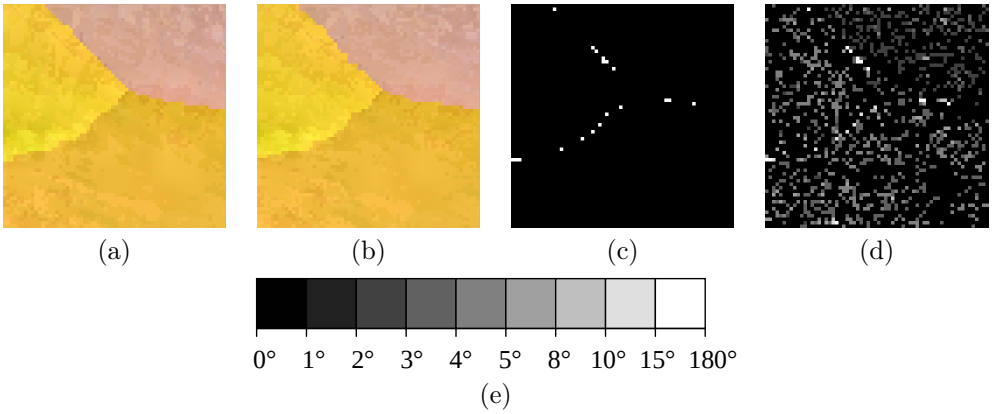


Figure G.12: The reconstruction of Test Case III using $\gamma(g, o)$ at 200 % noise level; $\text{FOM}_g = 0.995$, $\text{FOM}_o = 0.998$. Map arrangement and gray scales as for Figure 8.14.

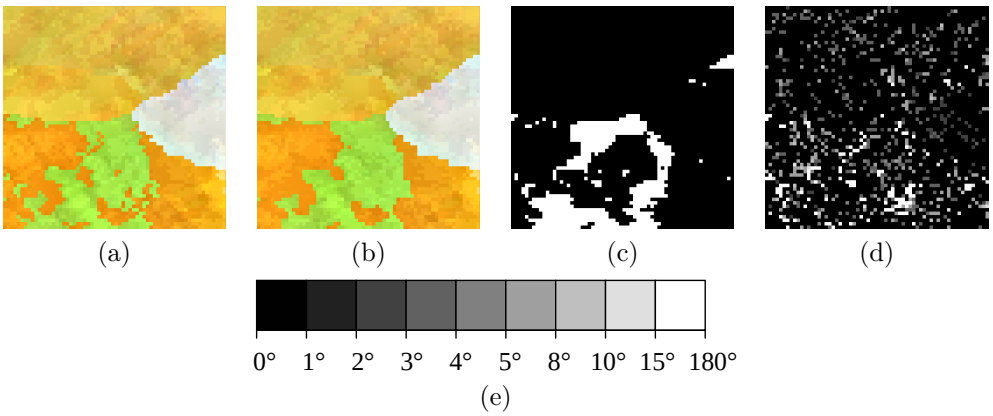


Figure G.13: The reconstruction of Test Case IV using $\gamma(g, o)$ at 200 % noise level; $\text{FOM}_g = 0.862$, $\text{FOM}_o = 0.985$. Map arrangement and gray scales as for Figure 8.14. (Note: (c) is irrelevant due to the lack of a meaningful definition of grains in this case; see text.)

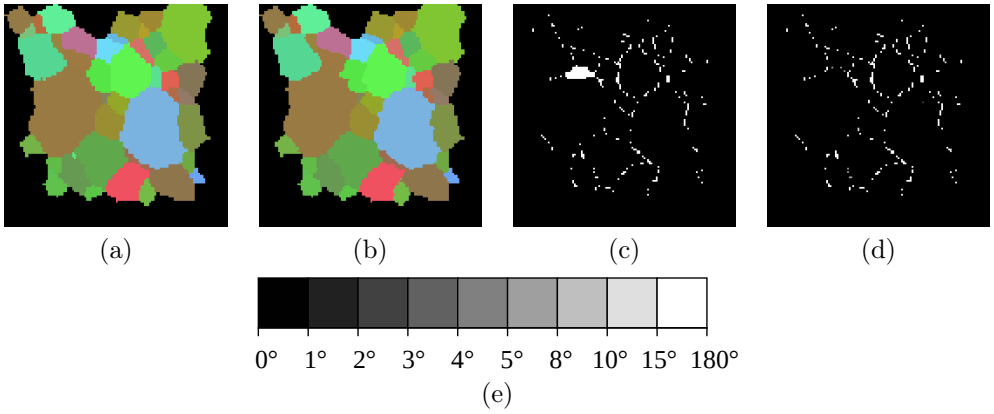


Figure G.14: The reconstruction of Test Case V using $\gamma(g, o)$ at 200 % noise level (original color mapping); $\text{FOM}_g = 0.967$, $\text{FOM}_o = 0.993$. Map arrangement and gray scales as for Figure 8.14. Black pixels in orientation maps represent void regions.

Bibliography

[1] DIscrete REConstruction Techniques (DIRECT) framework. http://www.inf.u-szeged.hu/~direct/, http://www.inf.u-szeged.hu/ipcg/projects/DIRECT.html.
cited on: 54

[2] SNARK09 framework. http://www.dig.cs.gc.cuny.edu/software/snark09/.
cited on: xxv, xxv, xxv, xxvi, xxvi, 53, 74, 75, 76

[3] Total Crystallography: Structure and dynamics of polycrystals. http://www.totalcryst.dk/.
cited on: 119

[4] Virtual Reality Modeling Language (VRML97). http://www.web3d.org/x3d/vrml/.
cited on: 56

[5] E. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. Princeton University Press, Princeton, 2003.
cited on: 9

[6] M. N. Alam, M. Blackman, and D. W. Pashley. High-angle Kikuchi patterns. *P. Roy. Soc. A–Math. Phy.*, 221(1145):224–242, jan 1954.
cited on: 101

[7] Bowen Alpern, Larry Carter, Matt Grayson, and Chris Pelkie. Orientation maps: Techniques for visualizing rotations. In *Proceedings of IEEE Conference on Visualization '93*, pages 183–188, oct 1993.
cited on: 16, 82, 100

[8] Andreas Alpers. *Instability and stability in discrete tomography*. PhD thesis, Technische Universität München, 2003.
cited on: 8

[9] Andreas Alpers. A short introduction to tomographic grain map reconstruction. *Pure Mathematics and Applications*, 20(1–2):157–163, 2009. PU.M.A.
cited on: 16

[10] Andreas Alpers and Sara Brunetti. Stability results for the reconstruction of binary pictures from two projections. *Image Vision Comput.*, 25(10):1599–1608, oct 2007.
cited on: 8

[11] Andreas Alpers, Richard J. Gardner, Stefan König, Robert S. Pennington, Chris B. Boothroyd, Lothar Houben, Rafal E. Dunin-Borkowski, and Kees Joost Batenburg. Geometric reconstruction methods for electron tomography. *Ultramicroscopy*, 128:42–54, may

2013.
cited on: 14

[12] Andreas Alpers and Peter Gritzmann. On stability, error correction, and noise compensation in discrete tomography. *SIAM J. Discrete Math.*, 20(1):227–239, aug 2006.
cited on: 7, 8

[13] Andreas Alpers, Peter Gritzmann, and Lionel Thorens. Stability and instability in discrete tomography. In Gilles Bertrand, Atsushi Imiya, and Reinhard Klette, editors, *Lect. Notes Comput. Sc.*, volume 2243, pages 175–186. Springer Berlin Heidelberg, 2001. Digital and Image Geometry: Advanced Lectures.
cited on: 8

[14] Andreas Alpers, Erik Knudsen, Henning Friis Poulsen, and Gabor T. Herman. Resolving ambiguities in reconstructed grain maps using discrete tomography. *Electron. Notes Discrete Math.*, 20:419–437, jul 2005. Proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: xxvii, 14, 17, 83, 87, 90, 91, 92, 101, 107, 135

[15] Andreas Alpers, Henning Friis Poulsen, Erik Knudsen, and Gabor T. Herman. A discrete tomography algorithm for improving the quality of three-dimensional X-ray diffraction grain maps. *J. Appl. Crystallogr.*, 39(4):582–588, aug 2006.
cited on: xxvi, xxvii, 9, 14, 90, 91, 92, 98, 101, 107

[16] Andreas Alpers, Lajos Rodek, Henning Friis Poulsen, Erik Knudsen, and Gabor T. Herman. Discrete tomography for generating grain maps of polycrystals. In Herman and Kuba [90], chapter 13, pages 271–301. Based on the proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: xxvii, 81, 91, 97, 101, 124, 126, 127, 130, 132, 133

[17] Andreas Alpers and Rob Tijdeman. The two-dimensional Prouhet–Tarry–Escott problem. *J. Number Theory*, 123(2):403–412, apr 2007.
cited on: 7

[18] Simon L. Altmann. *Rotations, Quaternions, and Double Groups*. Clarendon Press, Oxford, 1986.
cited on: 16, 17, 19, 20, 22

[19] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART). *Ultrasonic Imaging*, 6:81–94, 1984.
cited on: 6

[20] S. R. Arridge. Optical tomography in medical imaging. *Inverse Probl.*, 15(2):R41–R93, apr 1999.
cited on: 6

[21] Neil W. Ashcroft and N. David Mermin. *Solid State Physics*. Brooks/Cole, Australia, 1976.
cited on: xxvii, xxviii, 81, 105, 135, 137

[22] Márton Balaskó, Attila Kuba, Antal Nagy, Zoltán Kiss, Lajos Rodek, and László Ruskó. Neutron-, gamma- and X-ray three-dimensional computed tomography at the Budapest research reactor site. *Nucl. Instrum. Meth. A*, 542A:22–27, apr 2005. Proceedings of the

Fifth International Topical Meeting on Neutron Radiography (ITMNR-5, 2004).
cited on: 71, 73, 124, 125, 126, 127, 130, 131, 132, 133

[23] Márton Balaskó, Erzsébet Sváb, Attila Kuba, Zoltán Kiss, Lajos Rodek, and Antal Nagy. Pipe corrosion and deposit study using neutron- and gamma- radiation sources. *Nucl. Instrum. Meth. A*, 542A:302–308, apr 2005. Proceedings of the Fifth International Topical Meeting on Neutron Radiography (ITMNR-5, 2004).
cited on: 14

[24] Péter Balázs. Complexity results for reconstructing binary images with disjoint components from horizontal and vertical projections. *Discrete Appl. Math.*, 161:2224–2235, 2013.
cited on: 8

[25] Péter Balázs, Emese Balogh, and Attila Kuba. Reconstruction of 8-connected but not 4-connected *hv*-convex discrete sets. *Discrete Appl. Math.*, 147:149–168, 2005.
cited on: 8

[26] Péter Balázs and Mihály Gara. An evolutionary approach for object-based image reconstruction using learnt priors. In Arnt-Børre Salberg, Jon Yngve Hardeberg, and Robert Jenssen, editors, *Lect. Notes Comput. Sc.*, volume 5575, pages 520–529. Springer Berlin Heidelberg, 2009. Image Analysis: 16th Scandinavian Conference, SCIA 2009, Oslo, Norway, June 15-18, 2009. Proceedings.
cited on: 9, 36

[27] Emese Balogh, Attila Kuba, Csaba Dévényi, and Alberto Del Lungo. Comparison of algorithms for reconstructing hv-convex discrete sets. *Linear Algebra Appl.*, 339(1–3):23–35, dec 2001.
cited on: 8, 9

[28] Elena Barcucci, Andrea Frosini, Attila Kuba, Antal Nagy, Simone Rinaldi, Martin Šámal, and Steffen Zopf. Emission discrete tomography. In Herman and Kuba [90], chapter 15, pages 333–366.
cited on: 8, 14

[29] H. Bässmann and P. W. Besslich. *Bildverarbeitung Ad Oculos*. Springer Berlin, 1991.
cited on: 88, 103

[30] K. Joost Batenburg and Jan Sijbers. DART: A practical reconstruction algorithm for discrete tomography. *IEEE T. Image Process.*, 20(9):2542–2553, sep 2011.
cited on: 9

[31] K. Joost Batenburg, Jan Sijbers, Henning Friis Poulsen, and Erik Knudsen. DART: A robust algorithm for fast reconstruction of three-dimensional grain maps. *J. Appl. Crystallogr.*, 43(6):1464–1473, dec 2010.
cited on: 9, 14, 83

[32] Xavier L. Battle, G. S. Cunningham, and K. M. Hanson. Tomographic reconstruction using 3D deformable models. *Phys. Med. Biol.*, 43(4):983–990, 1998.
cited on: 9, 14, 36

[33] Joachim Baumann, Zoltán Kiss, Sven Krimmel, Attila Kuba, Antal Nagy, Lajos Rodek, Burkhard Schillinger, and Jürgen Stephan. Discrete tomography methods for nondestructive testing. In Herman and Kuba [90], chapter 14, pages 303–331. Based on the proceedings

of the Workshop on Discrete Tomography and its Applications 2005.
cited on: 35, 53, 71, 124, 125, 126, 127, 130, 131, 132, 133

[34] Felix Beckmann, Ulrich Bonse, and Theodor Biermann. New developments in attenuation and phase-contrast microtomography using synchrotron radiation with low and high photon energies. *Proc. SPIE*, 3772:179, sep 1999. Developments in X-Ray Tomography II, July 18, 1999, Denver, CO, USA.
cited on: 27

[35] August Beer. Bestimmung der Absorption des rothen Lichts in farbigen Flüssigkeiten. *Annalen der Physik und Chemie*, 86:78–88, 1852.
cited on: 27

[36] Julian Besag. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc. B*, 48(3):259–302, 1986.
cited on: 9, 149

[37] Griff L. Bilbro, Wesley E. Snyder, Stephen J. Garnier, and James W. Gault. Mean field annealing: A formalism for constructing GNC-like algorithms. *IEEE T. Neural Networ.*, 3(1):131–138, jan 1992.
cited on: 9

[38] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. Artificial Intelligence. MIT Press, Cambridge, MA, 1987.
cited on: 9

[39] Pierre Bouguer. *Essai d'optique sur la gradation de la lumière*. Claude Jombert, Paris, France, 1729.
cited on: 27

[40] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulations, and Queues*. Springer New York, 1999.
cited on: 9, 10, 13, 85

[41] S. Brocchi, Andrea Frosini, and C. Picouleau. Reconstruction of binary matrices under fixed size neighborhood constraints. *Theor. Comput. Sci.*, 406(1–2):43–54, oct 2008.
cited on: 9

[42] L. G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24:325–376, 1992.
cited on: 76

[43] Richard A. Brualdi. *Combinatorial matrix classes*. Cambridge: Cambridge University Press, 2006.
cited on: 7

[44] Sara Brunetti and Alain Daurat. Stability in discrete tomography: Linear programming, additivity and convexity. In Ingela Nyström, Gabriella Sanniti di Baja, and Stina Svensson, editors, *Lect. Notes Comput. Sc.*, volume 2886, pages 398–407. Springer Berlin Heidelberg, 2003. Discrete Geometry for Computer Imagery: 11th International Conference, DGCI 2003, Naples, Italy, November 19-21, 2003. Proceedings.
cited on: 8, 9

[45] H. J. Bunge. *Texture Analysis in Materials Science*. Butterworths, London, 1982.
cited on: 17, 81, 82, 136

[46] Weiwei Cai and Lin Ma. Comparison of approaches based on optimization and algebraic iteration for binary tomography. *Comput. Phys. Commun.*, 181(12):1974–1981, dec 2010.
cited on: 9

[47] Bruno M. Carvalho, Gabor T. Herman, Samuel Matej, C. Salzberg, and Eilat Vardi. Binary tomography for triplane cardiography. In Attila Kuba, Martin Šámal, and Andrew Todd-Pokropek, editors, *Information Processing in Medical Imaging*, pages 29–41. Springer Berlin, 1999.
cited on: 14, 85

[48] Augustin Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. In Bachelier (Paris), editor, *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, volume 25, pages 536–538. Académie des sciences, France, jul 1847.
cited on: 9, 50, 149

[49] B. Chalmond, F. Coldefy, and B. Lavayssière. Tomographic reconstruction from non-calibrated noisy projections in non-destructive evaluation. *Inverse Probl.*, 15(2):399–411, apr 1999.
cited on: 14

[50] Michael T. Chan, Gabor T. Herman, and Emanuel Levitan. Bayesian image reconstruction using image-modeling Gibbs priors. *Int. J. Imag. Syst. Tech.*, 9(2–3):85–98, 1998.
cited on: 85

[51] Michael T. Chan, Gabor T. Herman, and Emanuel Levitan. Probabilistic modeling of discrete images. In Herman and Kuba [88], chapter 9, pages 213–235.
cited on: xxvi, 85, 87, 107

[52] J. W. Christian. *The Theory of Transformations in Metals and Alloys*. Pergamon, Oxford, 1981.
cited on: 81

[53] John H. Conway and Derek A. Smith. *On Quaternions and Octonions: Their Geometry, Arithmetic, and Symmetry*. A. K. Peters, Natick, MA, 2003.
cited on: 15, 16, 19, 22

[54] Dominique de Werra, Marie-Christine Costa, Christophe Picouleau, and Bernard Ries. On the use of graphs in discrete tomography. *Ann. Oper. Res.*, 175(1):287–307, mar 2010.
cited on: 8

[55] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B*, 39:1–38, 1977.
cited on: 6

[56] Ferenc Divós and Péter Divós. Resolution of stress wave based acoustic tomography. In Friedrich-Wilhelm Bröker, editor, *Proceedings of the 14th International Symposium on Nondestructive Testing of Wood (Wood-NDT 2005)*, Holz- und Forstwirtschaft, pages 307–314. University of Applied Sciences, Germany, Eberswalde, Shaker Verlag, apr 2005.
cited on: 6

[57] Christoph Dürr, Flavio Guiñez, and Martin Matamala. Reconstructing 3-colored grids from horizontal and vertical projections is NP-hard: A solution to the 2-atom problem in discrete tomography. *SIAM J. Discrete Math.*, 26(1):330–352, 2012.
cited on: 8

[58] P. P. Ewald, editor. *Fifty Years of X-Ray Diffraction.* N.V.A. Oosthoek, 1962.
cited on: xxvii, 105, 135, 137

[59] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *J. Global Optim.*, 6:109–133, 1995.
cited on: 9

[60] F. C. Frank. Orientation mapping. *Metall. Mater. Trans. A*, A19(3):403–408, mar 1988.
cited on: 18, 19, 100

[61] Thomas Frese, Charles A. Bouman, and Ken Sauer. Multiscale Bayesian methods for discrete tomography. In Herman and Kuba [88], chapter 10, pages 237–264.
cited on: 9

[62] Anja Frost, Eike Renners, Michael Hötter, and Jörn Ostermann. Probabilistic evaluation of three-dimensional reconstructions from X-ray images spanning a limited angle. *Sensors-Basel*, 13(1):137–151, 2013.
cited on: 14

[63] David Gale. A theorem on flows in networks. *Pac. J. Math*, 7(2):1073–1082, jun 1957.
cited on: 7

[64] Mihály Gara, Tamás Sámuel Tasi, and Péter Balázs. Machine learning as a preprocessing phase in discrete tomography. In Ullrich Köthe, Annick Montanvert, and Pierre Soille, editors, *Lect. Notes Comput. Sc.*, volume 7346, pages 109–124. Springer Berlin Heidelberg, 2012. Applications of Discrete Geometry and Mathematical Morphology: First International Workshop, WADGMM 2010, Istanbul, Turkey, August 22, 2010, Revised Selected Papers.
cited on: 9

[65] Richard J. Gardner. *Geometric Tomography*, volume 58 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, New York, NY, 1995.
cited on: 8, 36

[66] Richard J. Gardner and Peter Gritzmann. Discrete tomography: Determination of finite sets by X-rays. *T. Am. Math. Soc.*, 349(6):2271–3395, 1997.
cited on: 14

[67] Richard J. Gardner and Peter Gritzmann. Uniqueness and complexity in discrete tomography. In Herman and Kuba [88], chapter 4, pages 85–113.
cited on: 8, 14

[68] Richard J. Gardner, Peter Gritzmann, and D. Prangenberg. On the computational complexity of reconstructing lattice sets from their X-rays. *Discrete Math.*, 202:45–71, 1999.
cited on: 7, 8

[69] Richard J. Gardner and P. McMullen. On Hammer's X-ray problem. *J. Lond. Math. Soc.*, 21:171–175, 1980.
cited on: 14

[70] R. J. Gaudette, D. H. Brooks, C. A. DiMarzio, M. E. Kilmer, E. L. Miller, T. Gaudette, and D. A. Boas. A comparison study of linear reconstruction techniques for diffuse optical tomographic imaging of absorption coefficient. *Phys. Med. Biol.*, 45:1051–1070, 2000.
cited on: 6

[71] Stuart Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE T. Pattern Anal.*, 6(6):721–741, nov 1984.
cited on: 8, 9, 13, 64, 85

[72] P. Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *J. Theor. Biol.*, 6:105–117, 1972.
cited on: 6

[73] F. W. Glover and M. Laguna. *Tabu Search*. Springer-Verlag Berlin, 1997.
cited on: 9

[74] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman, Boston, 1989.
cited on: 9

[75] R. Gordon, R. Bender, and G. T. Herman. Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theor. Biol.*, 29:471–482, 1970.
cited on: 6

[76] Hans Grimmer. Disorientations and coincidence rotations for cubic lattices. *Acta Crystallogr. A*, A30:685–688, 1974.
cited on: 15, 94, 95, 100

[77] Hans Grimmer. A unique description of the relative orientation of neighbouring grains. *Acta Crystallogr. A*, A36:382–389, 1980.
cited on: 15, 94, 95, 100

[78] P. Gritzmann, S. de Vries, and M. Wiegelmann. Approximating binary images from discrete X-rays. *SIAM J. Optimiz.*, 11:522–546, 2000.
cited on: 9

[79] Dieter Hahn, Pierre Thibault, Martin Bech, Marco Stockmar, Simone Schleede, Irene Zanette, Alexander Rack, Timm Weitkamp, Aniko Sztrókay, Thomas Schlossbauer, Fabian Bamberg, Maximilian Reiser, and Franz Pfeiffer. Numerical comparison of X-ray differential phase contrast and attenuation contrast. *Biomed. Opt. Express*, 3(6):1141–1148, jun 2012.
cited on: 27

[80] L. Hansen, J. Pospiech, and K. Lücke. *Tables of Texture Analysis of Cubic Crystals*. Springer Berlin, 1978.
cited on: 82, 94

[81] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, apr 1970.
cited on: 9, 10

[82] A. Heinz and P. Neumann. Representation of orientation and disorientation data for cubic, hexagonal, tetragonal and orthorombic crystals. *Acta Crystallogr. A*, A47:780–789, 1991.
cited on: 15, 82, 94, 95

[83] Sigurdur Helgason. *The Radon Transform*, volume 5 of *Progress in Mathematics*. Birkhäuser Boston, second edition, aug 1999.
cited on: 4

[84] Gabor T. Herman. Reconstruction of binary patterns from a few projections. In A. Günther, B. Levrat, and H. Lipps, editors, *International Computing Symposium 1973*, pages 371–378. North-Holland Publ. Co., Amsterdam, 1974.
cited on: 9, 36

[85] Gabor T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Advances in Computer Vision and Pattern Recognition. Springer-Verlag London, second edition, 2009.
cited on: 6

[86] Gabor T. Herman, Michael T. Chan, Yair Censor, Emanuel Levitan, Robert M. Lewitt, and T. K. Narayan. Maximum a posteriori image reconstruction from projections. In S. E. Levinson and L. Shepp, editors, *Image Models (and their Speech Model Cousins)*, volume 80 of *The IMA Volumes in Mathematics and its Applications*, pages 53–89. Springer New York, 1996.
cited on: 6

[87] Gabor T. Herman and Ran Davidi. Image reconstruction from a small number of projections. *Inverse Probl.*, 24(4):045011, aug 2008.
cited on: 6

[88] Gabor T. Herman and Attila Kuba, editors. *Discrete Tomography: Foundations, Algorithms, and Applications*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 1999.
cited on: 7, 9, 179, 180, 183, 184, 187

[89] Gabor T. Herman and Attila Kuba. Discrete tomography in medical imaging. *Proc. IEEE*, 91(10):1612–1626, oct 2003.
cited on: 14

[90] Gabor T. Herman and Attila Kuba, editors. *Advances in Discrete Tomography and Its Applications*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2007.
cited on: 7, 9, 176, 177, 184

[91] Gabor T. Herman, Roberto Marabini, José-María Carazo, Edgar Garduño, Robert M. Lewitt, and Samuel Matej. Image processing approaches to biological three-dimensional electron microscopy. *Int. J. Imag. Syst. Tech.*, 11(1):12–29, 2000.
cited on: 14

[92] Gabor T. Herman and L. B. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE T. Med. Imaging*, 12(3):600–609, sep 1993.
cited on: 6

[93] F. J. Humphreys and M. Hatherly. *Recrystallization and Related Annealing Phenomena*. Pergamon, Oxford, 1995.
cited on: 81

[94] Seema Jaggi, William C. Karl, and Alan S. Willsky. Estimation of dynamically evolving ellipsoids with applications to medical imaging. *IEEE T. Med. Imaging*, 14(2):249–258, jun 1995.
cited on: 36

[95] Avinash C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, New York, NY, 1988.
cited on: 6

[96] Akira Kaneko and Lei Huang. Reconstruction of plane figures from two projections. In Herman and Kuba [88], chapter 5, pages 115–135.
cited on: 8

[97] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, USA, 1972. Proceedings of a symposium on the Complexity of Computer Computations, March 20–22, 1972.
cited on: 8

[98] Zoltán Kató, Josiane Zerubia, and Marc Berthod. Satellite image classification using a modified Metropolis dynamics. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992 (ICASSP-92)*, volume 3, pages 573–576. IEEE, San Francisco, CA, mar 1992.
cited on: 9

[99] Zoltán Kató, Josiane Zerubia, and Marc Berthod. Bayesian image classification using Markov random fields. In Ali Mohammad-Djafari and G. Demoment, editors, *Maximum Entropy and Bayesian Methods: Paris, France, 1992*, volume 53 of *Fundamental Theories of Physics*, pages 375–382. Kluwer Academic Publishers, 1993.
cited on: 9

[100] Alia S. Khaled and Thomas J. Beck. Successive binary algebraic reconstruction technique: An algorithm for reconstruction from limited angle and limited number of projections decomposed into individual components. *J. X-Ray Sci. Tech.*, 21(1):9–24, 2013.
cited on: 14

[101] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.
cited on: 8, 9, 11

[102] C. Kisielowski, P. Schwander, F. H. Baumann, M. Seibt, Y. Kim, and A. Ourmazd. An approach to quantitative high-resolution transmission electron microscopy of crystalline materials. *Ultramicroscopy*, 58(2):131–155, may 1995.
cited on: 14

[103] Zoltán Kiss. *A Pixel-based Discrete Tomographic Technique and Its Applications*. PhD thesis, University of Szeged, Hungary, 2011.
cited on: 9, 64, 76

[104] Zoltán Kiss, Lajos Rodek, and Attila Kuba. Image reconstruction and correction methods in neutron and X-ray tomography. *Acta Cybernetica*, 17(3):557–587, 2006.
cited on: 35, 53, 71, 124, 125, 126, 127, 130, 131, 132, 133

[105] Zoltán Kiss, Lajos Rodek, Antal Nagy, Attila Kuba, and Márton Balaskó. Reconstruction of pixel-based and geometric objects by discrete tomography. Simulation and physical experiments. *Electron. Notes Discrete Math.*, 20:475–491, jul 2005. Proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: 35, 53, 71, 124, 125, 126, 127, 130, 131, 132, 133

[106] Charles Kittel. *Introduction to Solid State Physics*. Wiley, New York, NY, 7th. edition, 1995.
cited on: 81, 135

[107] U. F. Kocks, C. N. Tomé, and H.-R. Wenk. *Texture and Anisotropy*. Cambridge University Press, Cambridge, 1998.
cited on: 81, 82, 94

[108] D. Kölzow, Attila Kuba, and A. Volčič. An algorithm for reconstructing convex bodies from their projections. *Discrete Comput. Geom.*, 4(1):205–237, dec 1989.
cited on: 8, 14

[109] T. Yung Kong and Gabor T. Herman. Tomographic equivalence and switching operations. In Herman and Kuba [88], chapter 3, pages 59–84.
cited on: 8

[110] N. C. Krieger Lassen, Dorte Juul Jensen, and K. Conradsen. Image processing procedures for analysis of electron back scattering patterns. *Scann. Microsc.*, 6:115–121, 1992.
cited on: 101

[111] Sven Krimmel, Joachim Baumann, Zoltán Kiss, Attila Kuba, Antal Nagy, and Jürgen Stephan. Discrete tomography for reconstruction from limited view angles in non-destructive testing. *Electron. Notes Discrete Math.*, 20:455–474, jul 2005. Proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: 14

[112] Attila Kuba. The reconstruction of two-directionally connected binary patterns from their two orthogonal projections. *Comput. Vis. Graph. Image Process.*, 27(3):249–265, sep 1984.
cited on: 8

[113] Attila Kuba. Reconstruction of unique binary matrices with prescribed elements. *Acta Cybernetica*, 12(1):57–70, 1995.
cited on: 8

[114] Attila Kuba. Reconstruction of two-valued functions and matrices. In Herman and Kuba [88], chapter 6, pages 137–162.
cited on: 8

[115] Attila Kuba and Gabor T. Herman. Discrete tomography: A historical overview. In Herman and Kuba [88], chapter 1, pages 3–34.
cited on: 8

[116] Attila Kuba and Gabor T. Herman. Introduction. In Herman and Kuba [90], chapter 1, pages 1–16.
cited on: 8

[117] Attila Kuba, Gabor T. Herman, Samuel Matej, and Andrew Todd-Pokropek. Medical applications of discrete tomography. In Ding-Zhu Du, Panos M. Pardalos, and Jie Wang, editors, *Discrete Mathematical Problems with Medical Applications*, volume 55 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 195–208. DIMACS Center, Rutgers University, Piscataway, NJ, AMS, Rhode Island, 2000. Proceedings of the DIMACS Workshop on Discrete Mathematical Problems and Medical Applications 1999.
cited on: 14, 62

[118] Attila Kuba and Antal Nagy. Reconstruction of hv-convex binary matrices from their absorbed projections. *Electron. Notes Theor. Comput. Sci.*, 46:371–380, aug 2001. Proceedings of the 8th International Workshop on Combinatorial Image Analysis (IWCIA 2001).
cited on: 8, 14

[119] Attila Kuba, Antal Nagy, and Emese Balogh. Reconstructing hv-convex binary matrices from their absorbed projections. *Discrete Appl. Math.*, 139(1–3):137–148, apr 2004. Proceedings of The 2001 International Workshop on Combinatorial Image Analysis (IWCIA 2001).
cited on: 8, 14

[120] Attila Kuba and Maurice Nivat. Reconstruction of discrete sets with absorption. *Linear Algebra Appl.*, 339(1–3):171–194, dec 2001.
cited on: 8, 14

[121] Attila Kuba, Lajos Rodek, Zoltán Kiss, László Ruskó, Antal Nagy, and Márton Balaskó. Discrete tomography in neutron radiography. *Nucl. Instrum. Meth. A*, 542A:376–382, apr 2005. Proceedings of the Fifth International Topical Meeting on Neutron Radiography (ITMNR-5, 2004).
cited on: 35, 53, 71, 124, 125, 126, 127, 130, 131, 132, 133

[122] Attila Kuba, László Ruskó, Lajos Rodek, and Zoltán Kiss. Application of discrete tomography in neutron imaging. In Piero Chirco and Roberto Rosa, editors, *Proceedings of The Seventh World Conference on Neutron Radiography (WCNR-7, 2002)*, pages 361–371, Rome, Italy, sep 2002. ENEA (Italian National Agency for New Technologies, Energy and the Environment).
cited on: 35, 53, 71, 124, 125, 126, 127, 130, 131, 132, 133

[123] Attila Kuba, László Ruskó, Lajos Rodek, and Zoltán Kiss. Preliminary studies of discrete tomography in neutron imaging. *IEEE T. Nucl. Sci.*, 52(1):380–385, feb 2005. Based on the proceedings of The Seventh World Conference on Neutron Radiography (WCNR-7, 2002).
cited on: 35, 53, 71, 124, 125, 126, 127, 130, 131, 132, 133

[124] Attila Kuba and A. Volčič. Characterisation of measurable plane sets which are reconstructable from their two projections. *Inverse Probl.*, 4(2):513–527, may 1988.
cited on: 8

[125] Jack B. Kuipers. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, Princeton, NJ, 1999.
cited on: 16, 17, 19, 22, 94

[126] Arun K. Kulshreshth, Andreas Alpers, Gabor T. Herman, Erik Knudsen, Lajos Rodek, and Henning Friis Poulsen. A greedy method for reconstructing polycrystals from three-dimensional X-ray diffraction data. *Inverse Probl. Imag.*, 3(1):69–85, feb 2009.
cited on: 9, 14, 119

[127] Árpád Kurusa. The Radon transform on half sphere. *Acta Sci. Math.*, 58(1–4):143–158, 1993.
cited on: 4

[128] Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Eberhardt Klett, Augsburg, Germany, 1760.
cited on: 27

[129] B. C. Larson, W. Yang, G. E. Ice, J. D. Budai, and J. Z. Tischler. Three-dimensional X-ray structural microscopy with submicrometre resolution. *Nature*, 415:887–890, 2002.
cited on: 82

[130] Erik M. Lauridsen, Søren Schmidt, R. M. Suter, and Henning Friis Poulsen. Tracking: A method for structural characterization of grains in powders or polycrystals. *J. Appl. Crystallogr.*, 34:744–750, 2001.
cited on: 83, 135

[131] Emanuel Levitan and Gabor T. Herman. A maximum a posteriori probability expectation maximization for image reconstruction in emission tomography. *IEEE T. Med. Imaging*, 6:185–192, 1987.
cited on: 6

[132] Hstau Y. Liao and Gabor T. Herman. Automated estimation of the parameters of Gibbs priors to be used in binary tomography. *Discrete. Appl. Math*, 139(1–3):149–170, apr 2004. Proceedings of The 2001 International Workshop on Combinatorial Image Analysis (IWCIA 2001).
cited on: 85, 91

[133] Hstau Y. Liao and Gabor T. Herman. Discrete tomography with a very few views, using Gibbs priors and a Marginal Posterior Mode approach. *Electron. Notes Discrete Math.*, 20:399–418, jul 2005. Proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: 85

[134] George G. Lorentz. A problem of plane measure. *Am. J. Math.*, 71(2):417–426, apr 1949.
cited on: 7

[135] Pertti Lounesto. *Clifford Algebras and Spinors*, volume 286 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, second edition, 2001.
cited on: 15

[136] T. Markussen, X. Fu, L. Margulies, Erik M. Lauridsen, Søren Fæster Nielsen, Søren Schmidt, and Henning Friis Poulsen. An algebraic algorithm for generation of three-dimensional grain maps based on diffraction with a wide beam of hard X-rays. *J. Appl. Crystallogr.*, 37:96–102, 2004.
cited on: 14, 83

[137] Samuel Matej, Avi Vardi, Gabor T. Herman, and Eilat Vardi. Binary tomography using Gibbs priors. In Herman and Kuba [88], chapter 8, pages 191–212.
cited on: 9, 85

[138] A. D. McNaught and A. Wilkinson, editors. *IUPAC Compendium of Chemical Terminology*. Blackwell Scientific Publications, Oxford, second edition, 1997.
cited on: 27

[139] Donald Allan McQuarrie. *Statistical Mechanics*. University Science Books, Sausalito, California, USA, 2000.
cited on: 12

[140] L. Meister and H. Schaeben. A concise quaternion geometry of rotations. *Math. Method. Appl. Sci.*, 28(1):101–126, jan 2005.
cited on: 4, 22

[141] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.
cited on: 9, 10

[142] Ali Mohammad-Djafari. Binary polygonal shape image reconstruction from a small number of projections. *Elektrik*, 5(1):127–138, 1997.
cited on: 36

[143] Ali Mohammad-Djafari. Shape reconstruction in X-ray tomography. *Proc. SPIE*, 3170:240–251, oct 1997. Image Reconstruction and Restoration II.
cited on: 9, 14, 36

[144] Ali Mohammad-Djafari and Charles Soussen. Compact object reconstruction. In Herman and Kuba [88], chapter 14, pages 317–342.
cited on: 36

[145] A. Morawiec. Misorientation-angle distribution of randomly oriented symmetric objects. *J. Appl. Crystallogr.*, 28:289–293, 1995.
cited on: 18, 86, 95

[146] A. Morawiec. Distributions of misorientation angles and misorientation axes for crystallites with different symmetries. *Acta Crystallogr. A*, A53:273–285, 1997.
cited on: 18, 86

[147] A. Morawiec. *Orientations and Rotations. Computations in Crystallographic Textures*. Springer Berlin, 2004.
cited on: 19, 82, 94

[148] Roberto Moreno-Atanasio, Richard A. Williams, and Xiaodong Jia. Combining X-ray microtomography with computer simulation for analysis of granular and porous materials. *Particuology*, 8(2):81–99, apr 2010.
cited on: 14, 36

[149] Antal Nagy and Attila Kuba. Reconstruction of binary matrices from fan-beam projections. *Acta Cybernetica*, 17(2):359–383, 2005.
cited on: 4, 64

[150] Antal Nagy and Attila Kuba. Parameter settings for reconstructing binary matrices from fan-beam projections. *J. Comput. Inform. Technol.*, 14(2):101–110, 2006.
cited on: 4, 36, 64

[151] F. Natterer. *The Mathematics of Computerized Tomography.* Teubner Stuttgart, 1986.
cited on: 6

[152] Henning Friis Poulsen. A six-dimensional approach to microstructure analysis. *Philos. Mag.*, 83:2761–2778, 2003.
cited on: 18

[153] Henning Friis Poulsen. *Three-Dimensional X-ray Diffraction Microscopy: Mapping Polycrystals and Their Dynamics.* Springer Berlin, 2004.
cited on: 17, 18, 31, 83, 84, 90, 99, 103, 119, 135, 137

[154] Henning Friis Poulsen. An introduction to three-dimensional X-ray diffraction microscopy. *J. Appl. Crystallogr.*, 45(6):1084–1097, dec 2012.
cited on: 83

[155] Henning Friis Poulsen and X. Fu. Generation of grain maps by an algebraic reconstruction technique. *J. Appl. Crystallogr.*, 36:1062–1068, 2003.
cited on: 14, 83

[156] Henning Friis Poulsen, Søren Fæster Nielsen, Erik M. Lauridsen, Søren Schmidt, R. M. Suter, U. Lienert, L. Margulies, T. Lorentzen, and Dorte Juul Jensen. Three-dimensional maps of grain boundaries and the stress state of individual grains in polycrystals and powders. *J. Appl. Crystallogr.*, 34(6):751–756, dec 2001.
cited on: xxiii, xxiii, 31, 32, 83

[157] Henning Friis Poulsen, Søren Schmidt, J. Wright, H. O. Sørensen, J. Oddershede, and Andreas Alpers. 3DXRD and TotalCryst geometry. Technical report, SourceForge, 2009.
cited on: 119

[158] G. P. M. Prause and D. G. W. Onnasch. Binary reconstruction of the heart chambers from biplane angiographic image sequences. *IEEE T. Med. Imaging*, 15(4):532–546, aug 1996.
cited on: 62

[159] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.
cited on: 9, 11, 50, 147, 149

[160] Johann Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Ber. Sächs. Akad. Wiss. Leipzig, Math. Phys. Kl.*, 69:262–277, 1917.
cited on: 4

[161] Johann Radon. On the determination of functions from their integral values along certain manifolds. *IEEE T. Med. Imaging*, 5(4):170–176, dec 1986. Translator: Parks, P. C.
cited on: 4

[162] Valerie Randle. *Microtexture Determination and its Applications.* The Institute of Materials, London, England, 1992.
cited on: 100

[163] Valerie Randle and Olaf Engler. *Introduction to Texture Analysis, Macrotexture, Microtexture, and Orientation Mapping*. Gordon & Breach, 2000.
cited on: 82

[164] Normand Robert, Françoise Peyrin, and Martin J. Yaffe. Binary vascular reconstruction from a limited number of cone beam projections. *Med. Phys.*, 21(12):1839–1851, dec 1994.
cited on: 64

[165] Lajos Rodek, Erik Knudsen, Henning Friis Poulsen, and Gabor T. Herman. Discrete tomographic reconstruction of 2D polycrystal orientation maps from X-ray diffraction projections using Gibbs priors. *Electron. Notes Discrete Math.*, 20:439–453, jul 2005. Proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: 81, 97, 124, 126, 127, 130, 132, 133

[166] Lajos Rodek, Henning Friis Poulsen, Erik Knudsen, and Gabor T. Herman. A stochastic algorithm for reconstruction of grain maps of moderately deformed specimens based on X-ray diffraction. *J. Appl. Crystallogr.*, 40(2):313–321, apr 2007.
cited on: 81, 97, 124, 126, 127, 130, 132, 133

[167] Olinde Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *J. de Mathématiques Pures et Appliquées*, 5:380–440, 1840.
cited on: 18, 22

[168] Herbert John Ryser. Combinatorial properties of matrices of zeros and ones. *Can. J. Math.*, 9:371–377, 1957.
cited on: 7, 8

[169] Herbert John Ryser. *Combinatorial Mathematics*, volume 14 of *Carus Mathematical Monographs*. The Mathematical Association of America, Washington, DC, sep 1963.
cited on: 7, 8

[170] Abraham Savitzky. A historic collaboration. *Anal. Chem.*, 61(15):921A–923A, aug 1989.
cited on: 147

[171] Abraham Savitzky and Marcel J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, 36(8):1627–1639, jul 1964.
cited on: 147

[172] Burkhard Schillinger. *Neue Entwicklungen zu Radiographie und Tomographie mit thermischen Neutronen und zu deren routinemäßigem Einsatz*. PhD thesis, Technische Universität München, 1999.
cited on: 6, 14

[173] Burkhard Schillinger. Proposed combination of CAD data and discrete tomography for the detection of coking and lubricants in turbine blades or engines. *Electron. Notes Discrete Math.*, 20:493–499, jul 2005. Proceedings of the Workshop on Discrete Tomography and its Applications 2005.
cited on: 36

[174] Søren Schmidt, Søren Fæster Nielsen, C. Gundlach, L. Margulies, X. Huang, and Dorte Juul Jensen. Watching the growth of bulk grains during recrystallization of deformed

metals. *Science*, 305:229–232, 2004.
cited on: 83

[175] Søren Schmidt, U. L. Olsen, Henning Friis Poulsen, H. O. Sørensen, Erik M. Lauridsen, L. Margulies, C. Maurice, and Dorte Juul Jensen. Direct observation of 3D grain growth in Al-0.1%Mn. *Scripta Mater.*, 59:491–494, 2008.
cited on: 83

[176] Thomas Schüle, Christoph Schnörr, Stefan Weber, and Joachim Hornegger. Discrete tomography by convex–concave regularization and D.C. programming. *Discrete Appl. Math.*, 151(1–3):229–243, oct 2005.
cited on: 8, 9

[177] L. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall, 2002.
cited on: 88, 103

[178] L. Shepp and Y. Vardi. Maximum likelihood reconstruction in emission tomography. *IEEE T. Med. Imaging*, 1:113–121, 1982.
cited on: 6

[179] Jean Steinier, Yves Termonia, and Jules Deltour. Comments on smoothing and differentiation of data by simplified least square procedure. *Anal. Chem.*, 44(11):1906–1909, sep 1972.
cited on: 147

[180] M. Strobl, I. Manke, N. Kardjilov, A. Hilger, M. Dawson, and J. Banhart. Advances in neutron radiography and tomography. *J. Phys. D Appl. Phys.*, 42(24):243001, 2009.
cited on: 14

[181] Kenji Suzuki, Isao Horiba, and Noboru Sugie. Linear-time connected-component labeling based on sequential local operations. *Comput. Vis. Image Und.*, 89(1):1–23, jan 2003.
cited on: 88, 103

[182] Attila Tanács and Attila Kuba. Evaluation of a fully automatic medical image registration algorithm based on mutual information. *Acta Cybernetica*, 16:327–336, 2003.
cited on: 76

[183] Ashley B. Tarokh, Eric L. Miller, and David Boas. A new flexible parameterization for the estimation of 3D shape structure from scattered field data. *Proc. SPIE*, 5299:304–314, may 2004. Computational Imaging II, San Jose, CA, January 18, 2004.
cited on: 36

[184] Nobuhisa Ueda and Tadaaki Nagao. NP-completeness results for NONOGRAM via parsimonious reductions. Technical Report TR96-0008, Department of Computer Science, Tokyo Institute of Technology, may 1996.
cited on: 14

[185] Peter J. M. van Laarhoven and Emile H. L. Aarts. *Simulated Annealing: Theory and Applications*, volume 37 of *Mathematics and Its Applications*. Springer Netherlands, reprint edition, 1987.
cited on: 8, 9, 11

[186] Eilat Vardi, Gabor T. Herman, and T. Yung Kong. Speeding up stochastic reconstructions of binary images from limited projection directions. *Linear Algebra Appl.*, 339(1–3):79–89, dec 2001.
cited on: 9, 98

[187] László Varga, Péter Balázs, and Antal Nagy. Direction-dependency of binary tomographic reconstruction algorithms. *Graph. Models*, 73(6):365–375, nov 2011.
cited on: 9, 14, 59, 64

[188] Vlado Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Optimiz. Theory App.*, 45(1):41–51, jan 1985.
cited on: 9, 11

[189] Bertram Eugene Warren. *X-Ray Diffraction*. Dover Books on Physics. Dover Publications, New York, reprint edition, jun 1990.
cited on: 135, 137

[190] Stefan Weber, Antal Nagy, Thomas Schüle, Christoph Schnörr, and Attila Kuba. A benchmark evaluation of large-scale optimization approaches to binary tomography. In Attila Kuba, László G. Nyúl, and Kálmán Palágyi, editors, *Lect. Notes Comput. Sc.*, volume 4245, pages 146–156. Springer Berlin Heidelberg, 2006. Discrete Geometry for Computer Imagery: 13th International Conference, DGCI 2006, Szeged, Hungary, October 25-27, 2006. Proceedings.
cited on: 9

[191] Stefan Weber, Thomas Schüle, Attila Kuba, and Christoph Schnörr. Binary tomography with deblurring. In Ralf Reulke, Ulrich Eckardt, Boris Flach, Uwe Knauer, and Konrad Polthier, editors, *Lect. Notes Comput. Sc.*, volume 4040, pages 375–388. Springer Berlin Heidelberg, 2006. Combinatorial Image Analysis: 11th International Workshop on Combinatorial Image Analysis (IWCIA 2006), Berlin, Germany, June 19-21, 2006. Proceedings.
cited on: 9, 14

[192] Gerhard Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Number 27 in Applications of Mathematics: Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, second edition, 2003.
cited on: 9, 10, 13, 85

[193] Stuart I. Wright, Brent L. Adams, and Karsten Kunze. Application of a new automatic lattice orientation measurement technique to polycrystalline aluminum. *Mat. Sci. Eng. A-Struct.*, 160A(2):229–240, feb 1993.
cited on: 100

[194] Boris Yamrom, John A. Sutliff, and Andrew P. Woodfield. Visualizing polycrystalline orientation microstructures with spherical color maps. In *Proceedings of IEEE Conference on Visualization '94*, pages 46–51, oct 1994.
cited on: 100

[195] Rafal Zdunek. On image reconstruction algorithms for binary electromagnetic geotomography. *Theor. Comput. Sci.*, 406(1–2):160–170, oct 2008. Discrete Tomography and Digital Geometry: In memory of Attila Kuba.
cited on: 9, 14

Index