Bayesian iterative reconstruction methods for 3D X-ray Computed Tomography

Camille Chapdelaine

▶ **To cite this version:**

Camille Chapdelaine. Bayesian iterative reconstruction methods for 3D X-ray Computed Tomography. Signal and Image processing. Université Paris-Sud, 2019. English. NNT : . tel-02110033v1

HAL Id: tel-02110033

https://hal.science/tel-02110033v1

Submitted on 17 Apr 2019 (v1), last revised 25 Apr 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2019SACLS092

Thèse de doctorat

Bayesian iterative reconstruction methods for 3D X-ray Computed Tomography

Thèse de doctorat de l'Université Paris-Saclay
préparée à Université Paris-Sud

Ecole doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Traitement du signal et des images

Thèse présentée et soutenue à Gif-sur-Yvette, le 12 avril 2019, par

CAMILLE CHAPDELAINE

Composition du Jury :

Jeffrey A. Fessler Professeur, University of Michigan (EECS)	Rapporteur
Nicolas Gac Maître de conférences, Université Paris-Sud (L2S)	Co-encadrant
Jean-François Giovannelli Professeur, Université de Bordeaux (IMS)	Rapporteur
Ali Mohammad-Djafari Ancien Directeur de Recherche (CNRS) et fondateur de International Science Consulting and Training (ICST)	Directeur de thèse (Invité)
Estelle Parra Responsable Sanction Automatique, SAFRAN Tech (TSI)	Co-encadrante
Simon Rit Chargé de recherche, CNRS (CREATIS)	Examinateur
Thomas Rodet Professeur, ENS Paris-Saclay (SATIE)	Président du jury
Sabine Rolland du Roscoat Maître de conférences, Université Grenoble Alpes (3SR)	Examinatrice
Charles Soussen Professeur, CentraleSupélec (L2S)	Directeur de thèse

Contents

1 X-ray Computed Tomography : an inverse problem **14**
1.1 Definition of the reconstruction problem . . . 14
1.2 Uncertainties and ill-posedness of the reconstruction problem . . . 16

2 Reconstruction methods in X-ray Computed Tomography **20**
2.1 Geometrical model . . . 20
2.2 Mathematical foundations . . . 22
2.3 Analytical reconstruction methods . . . 24
2.4 Algebraic reconstruction methods . . . 26
2.5 Model-based iterative reconstruction (MBIR) methods . . . 27
2.6 MBIR methods with Total Variation (TV) regularization . . . 29
2.6.1 Alternating Direction Method of Multipliers (ADMM) . . . 29
2.6.2 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) . . . 31
2.6.3 Chambolle-Pock (CP) and Primal-Dual Frank-Wolfe (PDFW) algorithms . . . 32
2.7 Perspectives to develop MBIR methods for SAFRAN . . . 35

3 Projection and backprojection operators **37**
3.1 Unmatched ray-driven/voxel-driven (RD/VD) pair . . . 38
3.1.1 Ray-driven projector . . . 38
3.1.2 Voxel-driven backprojector . . . 38
3.2 Matched Separable Footprint (SF) pair . . . 40
3.2.1 SF projector on the GPU . . . 42
3.2.2 SF backprojector on the GPU . . . 45
3.2.3 Merging the two kernels of the GPU SF projector . . . 46
3.3 Validation . . . 50
3.3.1 Projector and backprojector as single modules . . . 50
3.3.2 Projector and backprojector in a full MBIR method . . . 53
3.4 Conclusion and perspectives . . . 57

4 Gauss-Markov-Potts prior model for joint reconstruction and segmentation **59**
4.1 Forward model . . . 60
4.2 Gauss-Markov-Potts prior model on the volume . . . 61
4.3 Joint Maximization A Posteriori (JMAP) algorithm . . . 62
4.3.1 Estimation of the volume . . . 64
4.3.2 Estimation of the variances of the uncertainties . . . 65
4.3.3 Estimation of the labels . . . 65
4.3.4 Estimation of the means of the classes . . . 67
4.3.5 Estimation of the variances of the classes . . . 67
4.3.6 Summary of the algorithm and fixation of the parameters . . . 68
4.4 Simulation results . . . 69
4.5 Results on real data . . . 76

4.5.1 Evaluation of the accuracy : Image Quality Indicator (IQI) volume . . . 76
4.5.2 Evaluation on Composite Material with Titanium on the Edges (CMTE) . . . 77
4.5.3 Evaluation on a metallic part . . . 84
4.6 Conclusion and perspectives . . . 84

5 Error-splitting forward model and its application with Gauss-Markov-Potts prior 87
5.1 Error-splitting forward model . . . 89
5.1.1 Derivation of the error-splitting forward model . . . 89
5.1.2 Measurement uncertainties . . . 90
5.1.3 Linear model uncertainties . . . 90
5.1.4 Fixing the parameters of the error-splitting forward model . . . 90
5.2 Joint Maximization A Posteriori (JMAP) using the error-splitting forward model . . . 93
5.2.1 Conditional distribution of the "true" projections . . . 94
5.2.2 Conditional distribution of the variances of the measurement uncertainties . . . 95
5.2.3 Conditional distribution of the volume . . . 95
5.2.4 Conditional distribution of the variances of the linear model uncertainties . . . 95
5.2.5 Conditional distribution of the hyperparameters of the prior model on the volume . . . 95
5.2.6 Discussion . . . 96
5.3 Comparison with the usual forward model . . . 96
5.4 Combination of the error-splitting forward model with Gauss-Markov-Potts prior on the volume . . . 98
5.5 Results on real data . . . 99
5.5.1 Framework for the experiments . . . 99
5.5.2 Evaluation of the accuracy : Image Quality Indicator (IQI) volume . . . 99
5.5.3 Evaluation of the robustness : Composite Material with Titanium on the Edges (CMTE) and metallic part . . . 103
5.5.4 Discussion . . . 107
5.6 Conclusion and perspectives . . . 107

6 Towards the estimation of the uncertainties on the reconstruction by Variational Bayesian Approach 114
6.1 Variational Bayesian Approach . . . 114
6.1.1 Mean Field Approximation . . . 116
6.1.2 Partially factorized approximation . . . 117
6.2 Application of VBA to 3D X-ray CT with Gauss-Markov-Potts prior . . . 119
6.2.1 Approximate distribution for the volume . . . 122
6.2.2 Approximate distribution for the labels . . . 123
6.2.3 Approximate distribution of the inverses of the variances of the uncertainties . . . 123
6.2.4 Approximate distribution for the means of the classes . . . 123
6.2.5 Approximate distribution for the inverses of the variances of the classes . . . 123
6.2.6 Summary and initialization of the VBA algorithm . . . 124
6.3 Difficulty of implementation : computation of the diagonal coefficients . . . 125
6.4 Simulation results . . . 130
6.5 Conclusion and perspectives . . . 131

7 Conclusion and perspectives 132

A Derivation of the algebraic forward model used in X-ray CT 138

B Projection and backprojection algorithms 140
B.1 Fast Siddon's ray-tracing method . . . 141
B.2 Separable Footprint (SF) projector and backprojector . . . 145
B.2.1 Description of the SF pair . . . 145
B.2.2 CPU implementation of the SF projector . . . 149

B.2.3 CPU implementation of the SF backprojector . . . 152
B.3 Projection and backprojection simulations . . . 154

C Derivation of the error-splitting forward model for X-ray CT **157**

D Calculations for fixing the parameters of the error-splitting forward model **159**

E Estimation theory **160**
E.1 Maximum a posteriori . . . 160
E.2 Minimum Mean Square Error . . . 160
E.3 Minimum Mean Absolute Error . . . 161

F Calculations for Variational Bayesian Approach **162**
F.1 Calculations of the approximate distributions . . . 162
F.1.1 Calculation of the approximate distribution for the volume . . . 162
F.1.2 Calculation of the approximate distribution for the labels . . . 164
F.1.3 Calculation of the approximate distribution for the inverses of the variances of the uncertainties on the projections . . . 166
F.1.4 Calculation of the approximate distribution for the means of the classes . . . 167
F.1.5 Calculation of the approximate distribution for the inverses of the variances of the classes . . . 167
F.2 Calculation of the negative free energy . . . 168
F.2.1 Calculation of the entropy of the approximate joint posterior distribution . . . 168
F.2.2 Calculation of the expectation of the log-joint posterior distribution . . . 169
F.2.3 Expression of the negative free energy . . . 170

List of notations

In this thesis, we use several notations which are summarized here in order to facilitate reading and examination.

- $\boldsymbol{g}$: the (vectorized) measurements, called the projections. The i^{th} element g_i of $\boldsymbol{g}$ corresponds to the projection at angle ϕ onto the cell located at position (u, v) of the detector.
- $\boldsymbol{f}$: the (vectorized) volume to reconstruct. The j^{th} element f_j of $\boldsymbol{f}$ corresponds to the attenuation coefficient of voxel j.
- $\boldsymbol{H}$: the projection matrix. The ideal forward model reads : $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f}$.
- $\boldsymbol{H}^T$: the backprojection matrix, which is the transpose of $\boldsymbol{H}$.

The variables related to the acquisition are :

- I_0 : mean number of photons sent by the source
- I_i : photon-count measured by the detector for ray i
- r_i : mean number of background events in measurement i of photon-count
- D : the source-to-detector distance.
- R : the source-to-object distance.
- N_ϕ : the number of projection angles. The projection angles are uniformly distributed over $[0, 2\pi]$.
- N_u and N_v : respectively the number of columns and the number of rows of cells on the detector. The number of cells of the detector is $N_u \times N_v$.
- δ_u and δ_v : the sizes of a cell respectively in u and v-directions.
- M : the size of vector $\boldsymbol{g}$. By definition, we have : $M = N_u \times N_v \times N_\phi$.
- N_x, N_y and N_z : the number of voxels respectively in x, y and z-directions.
- δ : the sizes of a voxel in x, y and z-directions. In this thesis, the voxels are considered as cubic of side δ.
- N : the size of vector $\boldsymbol{f}$. By definition, we have : $N = N_x \times N_y \times N_z$.

Some mathematical notations are :

- $p(\boldsymbol{\psi})$: the probability distribution of variable $\boldsymbol{\psi}$.
- $\mathcal{S}t_g(.|\alpha, \beta)$: generalized Student-t distribution, with shape parameter α and scale parameter β.
- $q(\boldsymbol{\psi})$: the approximate probability distribution of variable $\boldsymbol{\psi}$.
- $\mathcal{H}(q)$: the entropy of probability distribution q.
- $KL(q||p)$: the Kullback-Leibler divergence from q to p.

- $\mathcal{F}f$: the Fourier transform of f.
- $\mathfrak{H}f$: the Hilbert transform of f.
- diag $[\boldsymbol{x}]$: the square diagonal matrix whose the diagonal elements are the elements of vector $\boldsymbol{x}$.
- $\|\boldsymbol{y}\|^2_{\boldsymbol{V}} = \boldsymbol{y}^T\boldsymbol{V}^{-1}\boldsymbol{y}$ for all $\boldsymbol{y}$ and fixed diagonal matrix $\boldsymbol{V}$ with positive and non-zeros diagonal coefficients.
- $\lfloor x \rceil$: the nearest integer to real number x

Other notations used in this thesis are :

- $\mathcal{M}$: the prior model on the volume.
- $\boldsymbol{\theta}$: the hyperparameters of the prior model on the volume.
- $\mathcal{U}$: the usual forward model, with one term of uncertainties on the projections.
- ζ_i : the total uncertainty on projection i. The vector $\boldsymbol{\zeta} = (\zeta_i)_i$ is size M.
- v_{ζ_i} : the variance of total uncertainty ζ_i. The vector $\boldsymbol{v}_\zeta = (v_{\zeta_i})_i$ is size M.
- α_{ζ_0} : the shape parameter of the Inverse Gamma prior on the variances $\boldsymbol{v}_\zeta$ of the total uncertainties.
- β_{ζ_0} : the scale parameter of the Inverse Gamma prior on the variances $\boldsymbol{v}_\zeta$ of the total uncertainties.
- K : the number of materials in the volume.
- k : the label assigned to voxels in material k. Each material is associated to a class, which is also denoted by k. By definition, $k \in \{1, \dots, K\}$.
- $\mathcal{R}_k$: the set of voxels in class k.
- m_k : the mean of class k. The vector $\boldsymbol{m} = (m_k)_k$ is size K.
- v_k : the variance of class k. The vector $\boldsymbol{v} = (v_k)_k$ is size K.
- m_0 : the mean of the Gaussian prior on the means of the classes.
- v_0 : the variance of the Gaussian prior on the means of the classes.
- α_0 : the shape parameter of the Inverse Gamma prior on the variances of the classes.
- β_0 : the scale parameter of the Inverse Gamma prior on the variances of the classes.
- γ_0 : Potts coefficient, which tunes the compacity of the classes.
- α_k : parameter giving a prior probability e^{α_k} to be in class k. The vector $\boldsymbol{\alpha} = (\alpha_k)_k$ is size K.
- $\mathcal{S}$: the error-splitting forward model, which splits the uncertainties on the projections into two terms : the measurement uncertainties and the linear model uncertainties.
- ϵ_i : the measurement uncertainty on projection i. The vector $\boldsymbol{\epsilon} = (\epsilon_i)_i$ is size M.
- v_{ϵ_i} : the variance of measurement uncertainty ϵ_i. The vector $\boldsymbol{v}_\epsilon = (v_{\epsilon_i})_i$ is size M.
- ξ_i : the linear model uncertainty on projection i. The vector $\boldsymbol{\xi} = (\xi_i)_i$ is size M.
- v_{ξ_i} : the variance of linear model uncertainty ξ_i. The vector $\boldsymbol{v}_\xi = (v_{\xi_i})_i$ is size M.
- α_{ϵ_0} : the shape parameter of the Inverse Gamma prior on the variances $\boldsymbol{v}_\epsilon$ of the measurement uncertainties.
- β_{ϵ_0} : the scale parameter of the Inverse Gamma prior on the variances $\boldsymbol{v}_\epsilon$ of the measurement uncertainties.
- α_{ξ_0} : the shape parameter of the Inverse Gamma prior on the variances $\boldsymbol{v}_\xi$ of the linear model uncertainties.
- β_{ξ_0} : the scale parameter of the Inverse Gamma prior on the variances $\boldsymbol{v}_\xi$ of the linear model uncertainties.

List of Figures

1.1 Cone-beam acquisition process . . . 15
1.2 Summary of the deviations in the projections with respect to ideal linear projections $\boldsymbol{H}\boldsymbol{f}$ . . . 16
1.3 Toy example to illustrate the reconstruction problem . . . 17
1.4 Illustration of the non-existence of solution . . . 17
1.5 Illustration of the non-uniqueness of the solution . . . 17
1.6 Illustration of ill-conditioning . . . 18

2.1 Geometrical model . . . 21
2.2 Divergent-beam transform . . . 23
2.3 Fan-beam CT . . . 25
2.4 Radon transform inversion by the use of projection-slice theorem . . . 26
2.5 Interpolation step from polar to Cartesian grid in Fourier domain [Rod02] . . . 26
2.6 General hierarchical model to classify reconstruction methods . . . 36

3.1 Ray-driven projector . . . 39
3.2 Voxel-driven backprojector . . . 39
3.3 Trapezoidal shape of the transaxial footprint of a voxel at different projection angles . . . 41
3.4 Rectangular shape of the axial footprint of a voxel . . . 41
3.5 Voxels considered by the GPU SF projector depending on whether the main direction is x (a) or y (b) 43
3.6 Modified Shepp-Logan phantom (middle slice) . . . 50
3.7 CPU SF projection (a) followed by matched CPU SF backprojection (b) . . . 52
3.8 GPU SF projection with two kernels (a) followed by matched GPU SF backprojection (b) . . . 52
3.9 GPU SF projection with one kernel (a) followed by matched GPU SF backprojection (b) . . . 52
3.10 GPU RD projection (a) followed by unmatched GPU VD backprojection (b) . . . 53
3.11 Noisy projection of Shepp-Logan phantom, with SNR$=20$ db . . . 54
3.12 Reconstruction of Shepp-Logan phantom by the FDK method [FDK84] . . . 54
3.13 Reconstruction of Shepp-Logan phantom by PDFW [OMBF18] with the unmatched RD/VD pair . . . 54
3.14 Reconstruction of Shepp-Logan phantom by PDFW [OMBF18] with the matched SF pair . . . 54
3.15 Difference between the original volume and the reconstruction by PDFW, with the unmatched RD/VD pair . . . 54
3.16 Difference between the original volume and the reconstruction by PDFW, with the matched SF pair . 54
3.17 Estimated dual variable $\boldsymbol{p}$ (see algorithm 7) in PDFW [OMBF18] with the matched SF pair. The variable is size of the projections. . . . 55
3.18 Estimated dual variable $\tilde{\boldsymbol{z}}$ (see algorithm 7) in PDFW [OMBF18] with the matched SF pair. The variable is size of the volume. . . . 55
3.19 Evolution of the criterion during PDFW (a) with the RD/VD and SF pairs. A zoom on the last iterations is provided in (b). . . . 55
3.20 Evolution of the Root-Mean Square Difference (RMSD) between the estimate $\boldsymbol{f}^{(t)}$ and the real volume during PDFW (a) with the RD/VD and SF pairs. A zoom on the last iterations is provided in (b). . . . 56

3.21 Zoom on a region-of-interest with little details for : the real volume (a), the reconstruction by PDFW [OMBF18] with the unmatched RD/VD pair (b), and the reconstruction by PDFW [OMBF18] with the matched SF pair (c). The differences with respect to the original volume are shown in (d) for the unmatched RD/VD pair, and in (e) for the matched SF pair. 57

4.1 Potts field $\boldsymbol{z}$ for different values of γ_0 61
4.2 Hierarchical model with Gauss-Markov-Potts prior 63
4.3 Division like a chessboard in "white" and "black" voxels, in order to perform the segmentation in JMAP algorithm by ICM. The division is illustrated in 2D. The figure is inspired from [Aya10, chap. 4.5.2]. 66
4.4 Probability density function of $\mathcal{IG}(.|\alpha_0, \beta_0)$ for several values of (α_0, β_0) 70
4.5 Shepp-Logan phantom (a) and its segmentation (b). We see that it contains $K = 5$ classes. 72
4.6 FDK reconstruction of Shepp-Logan phantom (a) and its segmentation (b) used as initialization of JMAP 72
4.7 Joint reconstruction (a) and segmentation (b) of Shepp-Logan phantom obtained by JMAP with the unmatched RD/VD pair 73
4.8 Joint reconstruction (a) and segmentation (b) of Shepp-Logan phantom obtained by JMAP with the matched SF pair 73
4.9 Convergence metrics for JMAP (stopping criterion (a) and RMSD with respect to the real phantom (b)), with the unmatched RD/VD pair and the matched SF pair 74
4.10 Reconstructions of Shepp-Logan phantom by PDFW [OMBF18] with the unmatched RD/VD pair (a) and the matched SF pair (b) 75
4.11 Comparison of the evolution of the RMSD between JMAP and PDFW, with the unmatched RD/VD pair (a) and the matched SF pair (b) 75
4.12 Profile of Shepp-Logan phantom to compare the reconstructions by PDFW and JMAP, with the unmatched RD/VD pair (a) and with the matched SF pair 76
4.13 Reconstruction from 300 projections of IQI volume by filtered backprojection [FDK84] (bottom (a) and top (b)) 78
4.14 Reconstruction from 300 projections of IQI volume by PDFW [OMBF18] (bottom (a) and top (b)) . . 78
4.15 Joint reconstruction (bottom (a) and top (b)) and segmentation (bottom (c) and top (d)) of IQI volume from 300 projections by JMAP 79
4.16 Convergence of JMAP for the joint reconstruction and segmentation of IQI volume from 300 projections 80
4.17 Profiles of the holes of IQI volume for the reconstructions by filtered backprojection [FDK84], PDFW [OMBF18] and JMAP 80
4.18 Reconstruction of CMTE from 300 projections by filtered backprojection [FDK84] 81
4.19 Reconstruction of CMTE from 300 projections by PDFW [OMBF18] 81
4.20 Joint reconstruction (a) and segmentation (b) of CMTE from 300 projections by JMAP 81
4.21 Convergence of JMAP for the joint reconstruction and segmentation of CMTE from 300 projections . 82
4.22 First profile of CMTE for each reconstruction by filtered backprojection [FDK84], PDFW [OMBF18] and JMAP 82
4.23 Second profile of CMTE for each reconstruction by filtered backprojection [FDK84], PDFW [OMBF18] and JMAP 83
4.24 Reconstruction of the metallic part from 500 projections by filtered backprojection [FDK84] 84
4.25 Joint reconstruction (a) and segmentation (b) of the metallic part from 500 projections by JMAP . . . 85
4.26 Convergence of JMAP for the joint reconstruction and segmentation of the metallic part from 500 projections 85

5.1 Probability density function of $\mathcal{S}t_g(.|\alpha_{\xi_0}, \beta_{\xi_0})$ for several values of $(\alpha_{\xi_0}, \beta_{\xi_0})$ 91
5.2 Illustration of the dependence of the Signal-to-Noise Ratio (SNR) on the thickness of the material crossed by the rays 92
5.3 General hierarchical model including the error-splitting forward model 93

5.4 Hierarchical model combining the error-splitting forward model and Gauss-Markov-Potts prior on the volume . . . 97
5.5 Reference for IQI, reconstructed from 2400 projections (algorithm 12 in chapter 4) (bottom (a) and top (b)) . . . 100
5.6 Reconstructions from 300 projections of IQI volume by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (bottom (a) and top (b)), the usual forward model and modelling ζ as Gaussian (bottom (c) and top (d)), and the error-splitting forward model (bottom (e) and top (f)) . . . 101
5.7 Segmentations obtained jointly with the reconstruction from 300 projections of IQI volume by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (bottom (a) and top (b)), the usual forward model and modelling ζ as Gaussian (bottom (c) and top (d)), and the error-splitting forward model (bottom (e) and top (f)) . . . 102
5.8 Accuracy : profiles of the holes of IQI volume for each reconstruction, with Gauss-Markov-Potts prior for JMAP . . . 104
5.9 Convergence for JMAP with the error-splitting forward model and Gauss-Markov-Potts prior for IQI reconstruction from 300 projections . . . 105
5.10 Convergence to the reference volume for IQI reconstruction by PDFW [OMBF18] . . . 105
5.11 Convergence to the reference volume for IQI reconstruction by JMAP from 300 projections with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model . . . 105
5.12 Reference for CMTE, reconstructed from 1500 projections (algorithm 12 in chapter 4) (middle slice) . 107
5.13 Joint reconstructions and segmentations (middle slices) from 300 projections of the CMTE volume by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (reconstruction (a) and segmentation (b)), the usual forward model modelling ζ as Gaussian (reconstruction (c) and segmentation (d)), and the error-splitting forward model (reconstruction (e) and segmentation (f)) . . . 108
5.14 Convergence for JMAP with the error-splitting forward model and Gauss-Markov-Potts prior for CMTE reconstruction from 300 projections . . . 109
5.15 Convergence to the reference volume for CMTE reconstruction by PDFW [OMBF18] . . . 109
5.16 Convergence to the reference volume for CMTE reconstruction by JMAP from 300 projections with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model . . . 109
5.17 Profile of the CMTE volume for each reconstruction, with Gauss-Markov-Potts prior for JMAP . . . 110
5.18 Reference for the metallic part, reconstructed from 1500 projections (algorithm 12 in chapter 4) . . . 110
5.19 Joint reconstructions and segmentations from 500 projections of the metallic part by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (reconstruction (a) and segmentation (b)), the usual forward model and modelling ζ as Gaussian (reconstruction (c) and segmentation (d)), and the error-splitting forward model (reconstruction (e) and segmentation (f)) 111
5.20 Convergence to the reference volume for the metallic part reconstruction by JMAP from 500 projections with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model . . . 112
5.21 Convergence for JMAP with the error-splitting forward model and Gauss-Markov-Potts prior for the reconstruction of the metallic part from 500 projections . . . 112

6.1 Iterative algorithm to compute approximating distribution $q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})$ . . . 121
6.2 Diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$. These coefficients are the size of projections, of which one image is shown. . . . 126
6.3 Diagonal coefficients of $\boldsymbol{H}^T\boldsymbol{H}$. These coefficients are the size of a volume, of which the middle slice is shown. . . . 126
6.4 SF projection of a volume fully filled with ones, to compare to the diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$. . . . 126
6.5 SF backprojection of projections fully filled with ones, to compare to the diagonal coefficients of $\boldsymbol{H}^T\boldsymbol{H}$. 126
6.6 Joint reconstructions and segmentations by JMAP (a-b) and VBA (c-d) algorithms . . . 128

6.7 Probability maps for each voxel to be in each class, estimated by the VBA algorithm 129
6.8 Convergence of the VBA algorithm 130
6.9 Variances obtained by the VBA algorithm jointly with the reconstruction. The variances are shown in logarithmic scale. 130

B.1 Geometrical model 140
B.2 Voxel (x_e, y_e) (in 2D) pierced by ray i, with the representation of lengths $\lambda_x(x_e - \frac{1}{2}), \lambda_x(x_e + \frac{1}{2}), \lambda_y(y_e - \frac{1}{2}), \lambda_y(y_e + \frac{1}{2})$ 142
B.3 Trapezoidal transaxial footprint of a voxel at different projection angles 147
B.4 Rectangular axial footprint of a voxel 147
B.5 Azimuthal angles of the ray connecting the source to the center of a cell (u_e, v_e) and of the ray connecting the source to the center of a voxel (x_e, y_e, z_e) 150
B.6 Polar angles of the ray connecting the source to the center of a cell (u_e, v_e) and of the ray connecting the source to the center of a voxel (x_e, y_e, z_e) 150
B.7 Different overlaps of the transaxial footprint of a voxel with a cell 151
B.8 Different overlaps of the axial footprint of a voxel with a cell 151
B.9 Analytical projections of the Shepp-Logan phantom 156
B.10 Siddon-Jacobs-Han's projection on the CPU 156
B.11 SF projection on the CPU 156
B.12 SF backprojection on the CPU 156

List of Tables

3.1 Comparison of SF models in terms of computation speed and accuracy . . . 40
3.2 Matched GPU SF projector and backprojector as single modules . . . 51
3.3 Unmatched GPU RD projector and VD backprojector as single modules . . . 51
3.4 Coupling degree for the GPU RD/VD pair, the CPU SF pair and the GPU SF pairs . . . 51
3.5 Comparison of the results of PDFW with the RD/VD pair and the SF pair . . . 53
3.6 Relative Mean Absolute Error (RMAE) of PDFW in small region-of-interest Ω, with the RD/VD pair and the SF pair . . . 57

4.1 Parameters of JMAP to reconstruct Shepp-Logan phantom . . . 71
4.2 Comparison of the computational cost of JMAP in terms of projection and backprojection operations, depending on whether the used P/BP pair is matched or unmatched . . . 71
4.3 Comparison of the results of PDFW and JMAP, with the RD/VD pair and the SF pair . . . 74
4.4 Parameters of JMAP to reconstruct IQI volume . . . 77
4.5 Comparison of the computation time between PDFW and JMAP, for the reconstruction from 300 projections of IQI volume . . . 77
4.6 Parameters of JMAP to reconstruct CMTE . . . 77
4.7 Comparison of the computation times of PDFW and JMAP, for the reconstruction of CMTE from 300 projections . . . 84
4.8 Parameters of JMAP to reconstruct the metallic part . . . 84

5.1 Parameters of the error-splitting forward model for IQI reconstruction . . . 100
5.2 Parameters for Gauss-Markov-Potts prior model on IQI volume . . . 100
5.3 Performance comparison on IQI reconstruction from 300 projections, of PDFW and JMAP with Gauss-Markov-Potts prior model and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model . . . 103
5.4 Parameters of the error-splitting forward model for CMTE reconstruction . . . 103
5.5 Parameters for Gauss-Markov-Potts prior model on the CMTE volume . . . 106
5.6 Performance comparison on CMTE reconstruction from 300 projections, of TV-PDFW and JMAP with Gauss-Markov-Potts prior model and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model . . . 106

6.1 Parameters for the VBA algorithm . . . 130
6.2 Comparaison of JMAP and VBA algorithms . . . 131

B.1 Comparison of the Siddon-Jacobs-Han's projector and the SF projector on the CPU . . . 154
B.2 Computation time of the SF backprojector on the CPU . . . 154

List of Algorithms

1 ADMM algorithm [BPC+11] . . . 30
2 Scaled ADMM algorithm [BPC+11] . . . 31
3 Iterative Shrinkage-Thresholding Algorithm (ISTA) [BT09] . . . 32
4 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [BT09] . . . 32
5 Generic Chambolle-Pock (CP) algorithm [CP11] . . . 34
6 Chambolle-Pock (CP) algorithm to solve TV-penalized weighted least-squares [SJP+12] . . . 35
7 Primal-Dual Frank-Wolfe (PDFW) algorithm [OMBF18] . . . 35
8 Kernel $proj_x_ker$ for SF projection on GPU . . . 45
9 Kernel $proj_y_ker$ for SF projection on GPU . . . 46
10 Kernel $back_ker$ for SF backprojection on GPU . . . 47
11 Unified kernel $proj_ker$ for SF projection on GPU . . . 49
12 JMAP algorithm with Gauss-Markov-Potts prior . . . 68
13 JMAP algorithm using the error-splitting forward model . . . 96
14 JMAP algorithm combining the error-splitting forward model and Gauss-Markov-Potts prior . . . 99
15 VBA algorithm with Gauss-Markov-Potts prior . . . 124
16 Siddon-Jacobs-Han's projector [Sid85, JSDS+98, HLY99] . . . 144
17 SF projection algorithm on the CPU [LFB10] . . . 153
18 SF backprojection algorithm on the CPU [LFB10] . . . 155

Introduction

SAFRAN is a leading multinational group in the aircraft propulsion and equipments markets. The group inspects its manufactured aeronautical parts by imaging their interior thanks to Non-Destructive Testing (NDT) techniques. Among these techniques, 3D X-ray Computed Tomography (CT), using cone-beam acquisition process, is used to obtain an accurate reconstruction of the whole interior of the parts. Nevertheless, the inspected parts are characterized by pronounced asymmetries and very absorbing materials, causing many image artifacts in the reconstructions produced by the tomographs bought by SAFRAN. In order to enhance image quality for NDT of its aeronautical parts by 3D X-ray cone-beam Computed Tomography (CBCT), SAFRAN has wanted to investigate the potential of iterative reconstruction methods. To this end, the research and technogy center of SAFRAN, Safran Tech, through its Signal and Information Processing team (TSI : "Traitement du Signal et de l'Information"), has been working with Inverse Problems Group (GPI) of Signal and Systems Laboratory (L2S) since 2014.

GPI researchers have a long experience in many inverse problems such as CT. The specificity of their works lies in the use of a Bayesian framework to introduce complex prior information on the quantity to estimate. For a few years, GPI researchers have conducted several works applying their Bayesian methods to CBCT. Since the inspected volumes are very large, GPI researchers have also developed skills in high parallel computing on Graphical Processor Units (GPUs).

Aiming at better imaging aeronautical parts of SAFRAN, this thesis, resulting from the collaboration between Safran Tech and GPI, proposes iterative reconstruction methods thanks to a Bayesian approach. In particular, the proposed methods take benefit from the fact that the inspected parts are piecewise-constant and enhance reconstruction quality by enforcing a Gauss-Markov-Potts prior model on the volume. In addition, the Bayesian approach is also used to better model the acquisition process, so several uncertainties on the measurements, such as those resulting from the monochromatic approximation in X-rays' modelling, can be taken into account.

Since this work deals with huge volumes discretized in $512^3, 1024^3, \ldots$, voxels, the proposed model-based iterative reconstruction (MBIR) methods are designed to be highly parallelizable on GPU. Furthermore, MBIR methods very often have parameters which can be difficult to tune. In this thesis, a particular effort is made to propose strategies to fix these parameters thanks to physical interpretations provided by Bayesian approach.

Chapter 1 of this thesis introduces the reconstruction problem. The acquisition of the measurements, which are called the projections, is described, as well as the origins of the uncertainties on it. The ill-conditioning of the reconstruction problem is outlined and illustrated. Due to this ill-conditioning, volume estimation has to be made more robust, so a regularization is needed. Chapter 2 presents a state-of-the-art on analytical and iterative reconstruction methods in CT. At the end of this chapter, motivations for choosing Bayesian approach when developing MBIR methods for the aimed industrial application, are presented.

In all iterative reconstruction methods, the main computational burden lies in repeated projection and backprojection operations which model the attenuation of X-rays through the materials. Chapter 3 presents the implementation of several projection and backprojection operators (also called projector and backprojector) used in this thesis. Mathematically, the backprojector is defined as the adjoint of the projector. Nevertheless, in order to alleviate the computational cost of projection and backprojection operations, not complying with this definition to use so-called unmatched pairs of projector and backprojector, has become very common in the CT community. Such an unmatched pair of projector and backprojector (P/BP) is presented. Being aware that the use of an unmatched P/BP pair is a mathematical approximation, we also investigate the use of a matched pair, which is the Separable Footprint (SF) pair. For this pair, a new GPU implementation is proposed.

Chapter 4 focuses on the prior model used for SAFRAN's industrial parts to reconstruct. Since these parts are piecewise-constant volumes, we propose to use a Gauss-Markov-Potts prior model, which labels each voxel depending

on the material to which it belongs. Based on this model, a joint reconstruction and segmentation algorithm is derived. Experiments on simulated and real data show that the proposed algorithm takes advantage from its segmentation step to enhance reconstruction quality. Comparisons are also made between the uses of unmatched and matched P/BP pairs.

In X-ray CT, the projections have many uncertainties with different origins. In most iterative reconstruction methods, these uncertainties are all gathered in only one term in the model for the projections, called the forward model. Chapter 5 introduces an error-splitting forward model which distinguishes uncertainties called linear model uncertainties and measurement uncertainties. The use of this new forward model is motivated by adding an error term in the mean of Poisson statistics of photon-counts, in order to take into account the approximation error induced by the monochromatic model of X-rays with respect to their actual polychromacy. A general MBIR algorithm using the error-splitting forward model and valid for any prior model on the volume is then presented. Next, the error-splitting forward model is combined with Gauss-Markov-Potts prior to reconstruct aeronautical parts of SAFRAN. Compared to the usual forward model used in chapter 4, experiments show that the error-splitting forward model, combined with Gauss-Markov-Potts prior, brings more robustness and accuracy to the reconstruction.

At last, chapter 6 addresses the estimation of the uncertainties on the reconstruction. For this purpose, we propose an algorithm which jointly performs the estimation of the reconstruction and of the uncertainties on it. This algorithm is an instance of variational Bayesian approach (VBA) which looks for a simple approximation of the posterior distribution of the unknowns in order to easily estimate the posterior variances. The approximating distribution is chosen as almost fully factorized and minimizes Kullback-Leibler divergence with respect to the true posterior distribution of the unknowns. With Gauss-Markov-Potts prior model, the chosen approximating distribution is chosen to preserve the dependence between the value of a voxel and its label. New updating formulae for VBA using conditional approximating posterior distribution are proven in chapter 6. In addition, we highlight that the computation of approximate posterior variances requires one to use a matched P/BP pair in order to apply the algorithm in a reasonable time. For this purpose, SF pair presented in chapter 3 is efficiently used.

The thesis shows how to take benefit from each part of the reconstruction problem, i.e. : the modelling of the volume to reconstruct, the modelling of the decrease of X-rays, and the modelling of the uncertainties on the projections. Chapter 7 presents conclusions, draws perspectives for the industrial use of the proposed methods, and summarize open problems which have emerged during the three years of PhD studies.

Chapter 1

X-ray Computed Tomography : an inverse problem

In this chapter, we present the reconstruction problem in X-ray Computed Tomography (CT). In section 1.1, we first describe the acquisition process and the ideal physical model used to obtain a virtual reconstruction of the volume based on the measurements, called the projections of the volume in X-ray CT. In section 1.2, we show that the ideal physical model suffers from many uncertainties on the projections. In addition, we highlight the ill-posedness of the reconstruction problem, which hinders the exact estimation of the volume.

1.1 Definition of the reconstruction problem

The goal of X-ray CT is to virtually reconstruct the interior of a volume in order to assess its structure or microstructure. X-ray CT is used both in medicine on human or other organisms, and in industry on designed industrial parts. In industry, it is said to be a non-destructive testing (NDT) technique since the inspected part is not altered by the testing process.

The acquisition process of measurements used in SAFRAN industries is cone-beam and is illustrated in figure 1.1. The volume is placed between a source and a flat detector. The source sends a cone X-ray beam through the volume, so the intensities of the rays decrease before reaching the detector. The measurements of the decreased intensities by the cells of the detector form an image which is called a projection of the volume onto the detector. In order to acquire several views, the object is rotated around z-axis by a projection angle $(-\phi)$ as shown in figure 1.1. Projection angles are commonly uniformly distributed over $[0; 2\pi[$. Hence, the measurements are a collection of N_ϕ projections of the volume.

The decrease of X-rays' intensities is dependent on the materials encountered in the volume. For measurement i, photon-count I_i for ray i is given by Beer-Lambert law [SB93] : in the case of monochromatic rays measured in a perfectly unnoisy configuration, it reads

$$I_i = I_0 \exp\left[-\int_{\mathcal{L}_i} f(\boldsymbol{r})dl\right] \tag{1.1}$$

where I_0 is the mean number of photons sent by the source, and $\mathcal{L}_i$ is the line followed by ray i when passing through the volume. Quantity $f(\boldsymbol{r})$, expressed in m^{-1}, denotes the attenuation coefficient in the volume at point $\boldsymbol{r} \in \mathbb{R}^3$. In order to image the interior of the volume, the reconstruction process consists in estimating $f(\boldsymbol{r})$ for all $\boldsymbol{r}$ in the field-of-view. For this purpose, the recontruction problem can be linearized by estimating $f(\boldsymbol{r})$ from

$$g_i = \ln\left(\frac{I_0}{I_i}\right) = \int_{\mathcal{L}_i} f(\boldsymbol{r})dl \tag{1.2}$$

instead of I_i.

The number of cells of the detector is denoted by $N_u \times N_v$, which corresponds to the size of each projection. Typical values for N_u and N_v are 512, 1024 or 2048. Since we have N_ϕ projections, the number of measurements is

$$M = N_u \times N_v \times N_\phi. \tag{1.3}$$

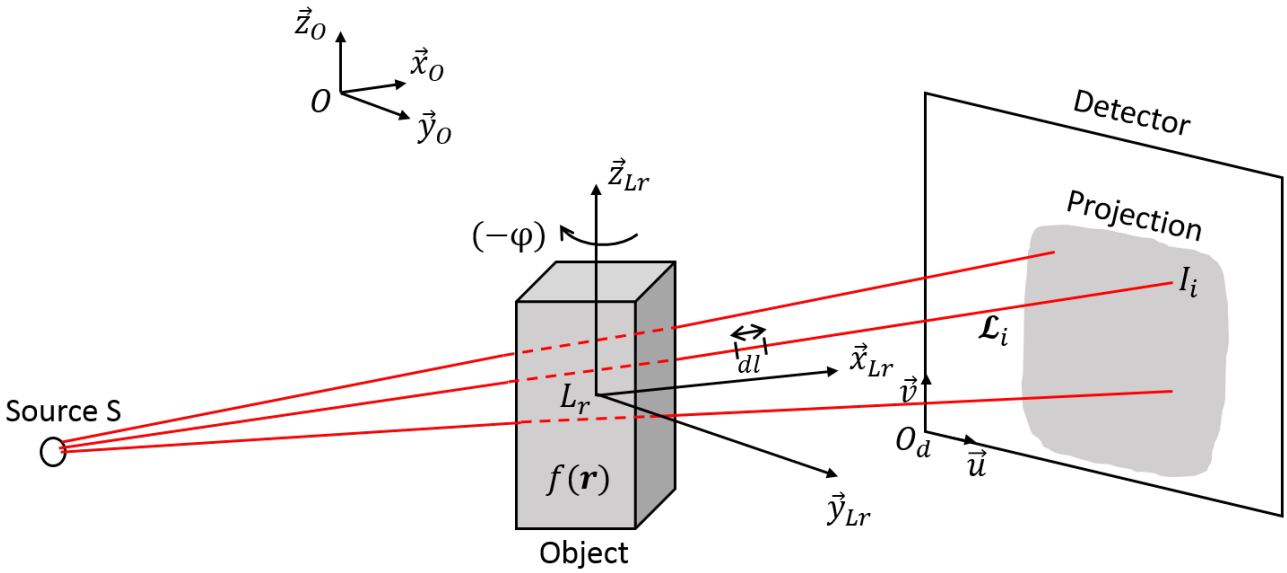


Figure 1.1: Cone-beam acquisition process

In order to have a good reconstruction, the number of projections N_ϕ can be very large : in this thesis, typical values for N_ϕ we will consider go from $N_\phi = 300$ to $N_\phi = 2400$. As a result, the number M of collected data can be very huge and require several Gigabytes of memory storing. Since the inspected volumes are industrial parts and not human organisms, the problem of reducing the dose is not really considered in NDT. Nevertheless, proposing robust reconstruction methods, which give good results with a reduced number of projections as with a lot of projections, can be interesting in order to gain time during the acquisition process and to save memory usage when analyzing the projections in some post-processing applications.

The field-of-view where the controlled volume lies is discretized in

$$N = N_x \times N_y \times N_z \tag{1.4}$$

voxels. Like the number M of measurements, N can be very huge : typical values for N are 512^3, 1024^3 or 2048^3. At SAFRAN, some industrial parts are too big to lie in the full field-of-view. In this case, imaging only a small region-of-interest (ROI) in these parts is difficult, since the projections are impacted by voxels which are out of the ROI. The problem is that the external voxels contributing to the projections are not the same at all the projection angles. For this reason, the projections are said to be truncated data. Reconstructing the ROI from these truncated projections is the challenge undertaken in local tomography [PM17, PDM17]. In this work, we will consider small industrial parts, so the object entirely lies in the field-of-view.

The attenuation coefficient for voxel j, $1 \leqslant j \leqslant N$, is denoted by f_j : analytical equation (1.2) can be expressed by an algebraic equation with matrix-vector product :

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} \tag{1.5}$$

where $\boldsymbol{g} = \{g_i, 1 \leqslant i \leqslant M\}$ and $\boldsymbol{f} = \{f_j, 1 \leqslant j \leqslant N\}$. Matrix $\boldsymbol{H} \in \mathbb{R}^{M \times N}$ is the projection matrix. Equivalently, it is also called the projection operator or the projector. Based on equation (1.2), an interpretation for the projection matrix coefficients is that H_{ij} corresponds to the length crossed by ray i in voxel j. Since M and N are very large, the size of matrix $\boldsymbol{H}$ is huge : for instance, if $M = 1024 \times 1024 \times 1500$ and $N = 1024^3$, the size of $\boldsymbol{H}$ is $M \times N = 1.69 \times 10^{18}$, i.e. several millions of Terabytes. Since a voxel j is crossed by few rays and not all rays i, it is known that matrix $\boldsymbol{H}$ is sparse. Nevertheless, the number of non-zero coefficients is still very huge, a

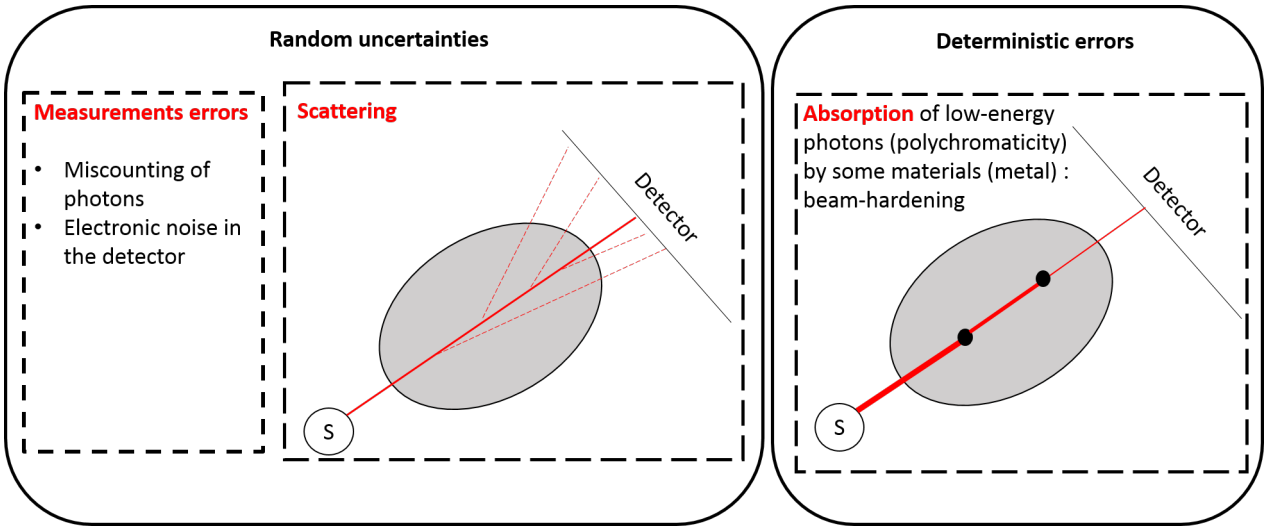


Figure 1.2: Summary of the deviations in the projections with respect to ideal linear projections $\boldsymbol{Hf}$

fortiori in 3D applications. As a result, the coefficients of $\boldsymbol{H}$ cannot be stored in memory, and accessing to each of them is an unfeasible task in a reasonable time. Consequently, in practice, only matrix-vector products $\boldsymbol{Hf}$ can be calculated, by computing relevant coefficients of $\boldsymbol{H}$ on-the-fly, as done for instance in ASTRA toolbox [Ble15]. In order to accelerate projection operations and to deal with huge volumes, the use of high-parallel computing (HPC) on *Graphical Processor Unit* (GPU) [SK10] has become very popular in the last ten years. GPUs are also used to perform the backprojection operation

$$\boldsymbol{b} = \boldsymbol{H}^T \boldsymbol{g} \tag{1.6}$$

which is represented by the adjoint $\boldsymbol{H}^T$ of projector $\boldsymbol{H}$. Matrix $\boldsymbol{H}^T$ is called the backprojection matrix, or, equivalently, the backprojection operator or the backprojector. In iterative reconstruction methods, the applications of projection and backprojection operators $\boldsymbol{H}$ and $\boldsymbol{H}^T$ correspond to the main computational burden. As a consequence, implementing efficient and accurate projector and backprojector on GPU is a key aspect to make iterative reconstruction methods fast. This topic will be covered in chapter 3.

1.2 Uncertainties and ill-posedness of the reconstruction problem

The forward model given by equation (1.1) (or equivalently by equation (1.5)) is the ideal forward model for X-ray CT. In practice, it suffers from several uncertainties. First, photons are miscounted by the cells. As a consequence, it is better to model photon-count I_i by a Poisson process of mean given by Beer-Lambert law [LC+84, KPTF13] :

$$p(I_i|\boldsymbol{f}, r_i) = \mathcal{P}\left(I_0 e^{-[\boldsymbol{Hf}]_i} + r_i\right). \tag{1.7}$$

In this statistics is added a term denoted by r_i, which corresponds to the mean number of background events. This term enables to take into account electronic noise introduced by the detector itself [NDMF+13]. The mean of background events is precomputed before the acquisition, so it can be included in the linearization of the reconstruction problem by

$$g_i = \ln\left(\frac{I_0}{I_i - r_i}\right), \forall i. \tag{1.8}$$

Given that the number of photons reaching the detector is large in X-ray CT, a second order Taylor expansion of Poisson statistics (1.7) of photon-counts can be performed [SB93, TBSH06, TSBH07], leading to the forward model accounting for uncertainties

$$\boldsymbol{g} = \boldsymbol{Hf} + \boldsymbol{\zeta} \tag{1.9}$$

where uncertainties $\zeta_i, \forall i$, are modeled as Gaussian [SB93, KPTF13]. In appendix A, we show how to derive this forward model from Poisson statistics (1.7). The proof is similar to the one presented in [SB93]. We give it in appendix for the sake of comprehensiveness, since we will re-consider the used forward model in chapter 5.

Analogy of the reconstruction problem

?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
12	12	8	8	

Figure 1.3: Toy example to illustrate the reconstruction problem

Non-existence of solution

?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
12	12	9	8	

Figure 1.4: Illustration of the non-existence of solution

One problem :

?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
12	12	8	8	

Many different solutions :

3	3	2	2	10
3	3	2	2	10
3	3	2	2	10
3	3	2	2	10
12	12	8	8	

10	0	0	0	10
2	8	0	0	10
0	4	6	0	10
0	0	2	8	10
12	12	8	8	

1	3	2	4	10
5	2	2	1	10
4	3	1	2	10
2	4	3	1	10
12	12	8	8	

Figure 1.5: Illustration of the non-uniqueness of the solution

Poisson noise on photon-counts is only one source of uncertainties in X-ray CT. Several other phenomena happen in practice. In particular, formula (1.7) models X-rays as monochromatic, while they are actually polychromatic. The polychromacy of X-rays means that photons of different energies will not interact in the same way with a same material. For instance, some photons scatter in the volume, so they are recorded by the detector in a location different from the one given by Beer-Lambert law. This results in artifacts in the reconstruction if not taken into account [NDMF+13]. Another object-dependent phenomenon is beam-hardening : photons of low energies are absorbed when they encounter some materials such as metal [GDMJ+16]. Beam-hardening causes strong artifacts in the reconstruction, which require complex methods in order to reduce it, such as corrections of the projections based on a first reconstruction [KKF08].

The deviations in real projections $\boldsymbol{g}$ from ideal theoretical projections $\boldsymbol{Hf}$ are summarized in figure 1.2. Like measurement errors such as Poisson noise or electronic noise in the detector, scattering is a random uncertainty on the projections, while beam-hardening is a deterministic error which is systematically the same if the scan is repeated. Nevertheless, scattering and beam-hardening are both object-dependent phenomena which are difficult to model in a tractable way in 3D X-ray CT.

The presence of uncertainties in projections $\boldsymbol{g}$ makes the estimation of $\boldsymbol{f}$ only based on $\boldsymbol{g}$ unstable, due to the

Two very close problems :

?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
?	?	?	?	10
12	12	8	8	

?	?	?	?	10
?	?	?	?	11
?	?	?	?	10
?	?	?	?	10
12	12	9	8	

Two very different solutions :

3	3	2	2	10
3	3	2	2	10
3	3	2	2	10
3	3	2	2	10
12	12	8	8	

2	4	2	2	10
4	3	2	2	11
3	3	3	1	10
3	2	2	3	10
12	12	9	8	

Figure 1.6: Illustration of ill-conditioning

conditioning of the reconstruction problem : for a variation $\Delta \boldsymbol{g}$ of data $\boldsymbol{g}$, the variation $\Delta \hat{\boldsymbol{f}}$ of estimation $\hat{\boldsymbol{f}}$ is bounded

$$\frac{\|\Delta \hat{\boldsymbol{f}}\|}{\|\hat{\boldsymbol{f}}\|} \leqslant \chi_{\boldsymbol{H}} \frac{\|\Delta \boldsymbol{g}\|}{\|\boldsymbol{g}\|} \tag{1.10}$$

where $\chi_{\boldsymbol{H}} \geqslant 0$ is the condition number of matrix $\boldsymbol{H}$. From equation (1.10), we see that small $\chi_{\boldsymbol{H}}$ ensures that a small variation $\Delta \boldsymbol{g}$ results in a small variation $\Delta \hat{\boldsymbol{f}}$ in volume estimation. In this case, the reconstruction problem is said to be well-conditioned. Unfortunately, the reconstruction problem is not well-conditioned but ill-conditioned [Idi08] : because of large $\chi_{\boldsymbol{H}} \gg 1$, we are not sure whether a small variation $\Delta \boldsymbol{g}$ will not lead to a large variation $\Delta \hat{\boldsymbol{f}}$.

This ill-conditioning is one of the reasons for which the reconstruction problem is *ill-posed*. An inverse problem is said to be ill-posed if it is not *well-posed* in Hadamard's sense. An inverse problem is said to be well-posed if and only if the three following conditions are met :

- a solution to the problem exists,
- this solution is unique,
- the problem is well-conditioned, i.e. the condition number is small : $\chi_{\boldsymbol{H}} \ll 1$. If the problem is well-conditioned, the solution changes smoothly with the data according to equation (1.10).

In order to make the reader more familiar with these notions, we show in figure 1.3 a toy example which is an analogy of the reconstruction problem in X-ray CT. The problem consists in finding the coefficients of one matrix, of which we only know the sums over each row and each column. The coefficients of the matrix have to be positive integers. Although it is rather easy to find a solution to the problem in figure 1.3, we may notice in figure 1.4 that, for some values of the data, it is possible that no solution exists. In addition, as illustrated in figure 1.5, if one solution exists, then this solution may be not unique. At last, figure 1.6 illustrates the impact of ill-conditioning. In this figure, we see that the two presented problems are very close, since only two numbers are different in the second problem, compared to the first. Nevertheless, in the solutions we find, much more than two coefficients are different. This is typically

an ill-conditioned problem, which shows how an estimation only relying on the data can be unstable. In order to make the reconstruction problem well-posed, it is necessary to regularize the reconstruction problem by incorporating prior information on the volume to reconstruct, so the estimation is a tradeoff between the data and the regularization [Tik63]. We present this notion more precisely in chapter 2.

Chapter 2

Reconstruction methods in X-ray Computed Tomography

The first mathematical foundations for X-ray CT were provided by the works of Radon [Rad17]. Analytical reconstruction methods rely on these mathematical foundations to propose approximate inversion formulae in order to solve the reconstruction problem [FDK84, SK85, RRT$^+$18]. In industry, the Feldkamp-Davis-Kress (FDK) method [FDK84] is widely used for NDT but requires a large amount of projections to obtain reconstructions of good quality. Due to scattering, beam-hardening and other uncertainties in the projections, these reconstructions suffer from many artifacts which are hard to reduce even with a lot of projections. In order to enhance image quality, model-based iterative reconstruction (MBIR) methods enforce a prior model $\mathcal{M}$ on the volume to reconstruct. This prior model brings a regularization of the reconstruction problem which leads to a better conditioning and to a more robust estimation with respect to the uncertainties in the projections. In our context, MBIR methods are interesting in order to inspect industrial parts, since, with same acquisition conditions, they increase the reconstruction quality compared to analytical reconstruction methods. In addition, they can also obtain good results even when the angle distribution is limited [Fes00]. In the medical context, MBIR methods are appealing because they enable to reduce the dose while preserving a good reconstruction quality.

This chapter briefly reviews reconstruction methods in X-ray CT. After having introduced the used geometrical model in section 2.1, the main mathematical foundations of X-ray CT are presented in section 2.2. Then, section 2.3 focuses on analytical reconstruction methods and section 2.4 on first algebraic reconstruction methods proposed in the literature [GBH70]. These methods only rely on the projections to perform the reconstruction. In section 2.5, MBIR methods combines the information brought by the data with a prior model, which leads to enhanced reconstructions compared to methods presented in sections 2.3 and 2.4. Next, section 2.6 focuses on the case of total variation (TV) regularization, in order to present several optimization algorithms which have been proposed in the literature. At the end of the chapter, a summary about the reviewed methods is provided and the chosen approach to develop reconstruction algorithms for SAFRAN is justified.

2.1 Geometrical model

The used geometrical model is shown in figure 2.1. In this model, the source and the detector are simultaneously rotated by a projection angle ϕ around z-axis $(L_r\vec{z}_{Lr})$ while the volume is fixed : this is equivalent to rotate the volume by an angle $(-\phi)$ as presented in chapter 1, and is simpler to handle for geometrical calculations. We consider cubic voxels of side length δ. The volume is discretized into $N = N_x \times N_y \times N_z$ voxels. Coordinates (x, y, z) are given in the optical frame $(O, \vec{x}_O, \vec{y}_O, \vec{z}_O)$. The position of each voxel is given by the normalized coordinates of its center (x_e, y_e, z_e), $0 \leqslant x_e < N_x$, $0 \leqslant y_e < N_y$, $0 \leqslant z_e < N_z$. The corresponding coordinates in the object frame are

$$\begin{cases} x_v = x_e \times \delta \\ y_v = y_e \times \delta \\ z_v = z_e \times \delta \end{cases} . \tag{2.1}$$

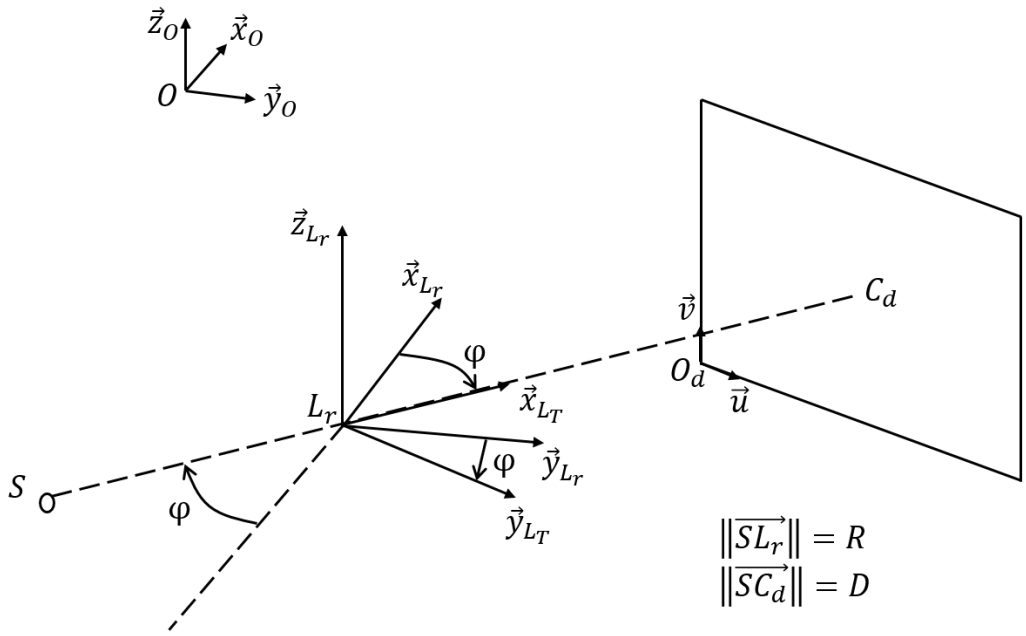


Figure 2.1: Geometrical model

The position of the source S is denoted by $(x_S(\phi), y_S(\phi), z_S(\phi))$. The position of the detector is given by the position of its center C_d $(x_{C_d}(\phi), y_{C_d}(\phi), z_{C_d}(\phi))$. In the ideal axial cone-beam geometry, the source and the center of the detector are aligned with the rotation center L_r and lie in the same horizontal plane. In practice, this configuration is very hard to get and some calibration is required in order to avoid artifacts in the reconstruction [SHH07, KLH$^+$08]. In this work, we assume the geometry is correctly calibrated, so we have :

$$z_{C_d}(\phi) = z_S(\phi) = z_{L_r}, \forall \phi. \tag{2.2}$$

The cells of the detector are all the same size $\delta_u \times \delta_v$. The local geometrical frame of the detector is $(O_d, \vec{u}, \vec{v})$. In this frame, the location of one cell is given by the normalized coordinates of its center (u_e, v_e), $0 \leqslant u_e < N_u$, $0 \leqslant v_e < N_v$. Their unnormalized coordinates are

$$\begin{cases} u = u_e \times \delta u \\ v = v_e \times \delta v \end{cases} . \tag{2.3}$$

In $(O_d, \vec{u}, \vec{v})$, the normalized coordinates of the center of the detector are

$$\begin{cases} u_{C_{d_e}} = \frac{u_{C_d}}{\delta u} = \frac{N_u - 1}{2} \\ v_{C_{d_e}} = \frac{v_{C_d}}{\delta v} = \frac{N_v - 1}{2} \end{cases} . \tag{2.4}$$

The coordinates of the rotation center L_r in the optical frame $(O, \vec{x}_O, \vec{y}_O, \vec{z}_O)$ are

$$\begin{cases} x_{Lr} = x_{Lr_e}\delta = \frac{N_x - 1}{2}\delta \\ y_{Lr} = y_{Lr_e}\delta = \frac{N_y - 1}{2}\delta \\ z_{Lr} = z_{Lr_e}\delta = \frac{N_z - 1}{2}\delta \end{cases} . \tag{2.5}$$

The normalized coordinates of one voxel in the source-detector frame $(Lr, \vec{x}_{Lr}, \vec{y}_{Lr}, \vec{z}_{Lr})$ are denoted by

$$\begin{cases} x'_e = x_e - x_{Lr_e} \\ y'_e = y_e - y_{Lr_e} \\ z'_e = z_e - z_{Lr_e} \end{cases} . \tag{2.6}$$

The source-to-detector distance

$$\|\overrightarrow{SC_d}\| = D \tag{2.7}$$

and the source-to-object distance

$$\|\overrightarrow{SLr}\| = R \tag{2.8}$$

give the magnification

$$Magn = \frac{D}{R}. \tag{2.9}$$

With our geometrical model, the coordinates of the source in the optical frame are

$$\begin{cases} x_S(\phi) &= x_{Lr} - R\cos\phi \\ y_S(\phi) &= y_{Lr} - R\sin\phi \\ z_S(\phi) &= z_{Lr} \end{cases} \tag{2.10}$$

and the coordinates of the detector are

$$\begin{cases} x_{C_d}(\phi) &= x_{Lr} + (D-R)\cos\phi \\ y_{C_d}(\phi) &= y_{Lr} + (D-R)\sin\phi \\ z_{C_d}(\phi) &= z_{Lr} \end{cases} \tag{2.11}$$

for each projection angle ϕ. The coordinates of a cell $P(u_e, v_e)$ at one projection angle ϕ in the frame $(C_d, \vec{u}, \vec{v})$ are denoted by

$$\begin{cases} s &= s_{u_e} \times \delta_u = (u_e - u_{C_{d_e}}) \times \delta_u \\ t &= t_{v_e} \times \delta_v = (v_e - v_{C_{d_e}}) \times \delta_v \end{cases}, \tag{2.12}$$

where

$$\overrightarrow{C_dP} = s\vec{u} + t\vec{v}. \tag{2.13}$$

Given that the detector is rotated by ϕ with respect to the optical frame

$$\begin{cases} \vec{u} = -\vec{x}_{Lr}\sin\phi + \vec{y}_{Lr}\cos\phi = -\vec{x}_O\sin\phi + \vec{y}_O\cos\phi \\ \vec{v} = \vec{z}_{Lr} = \vec{z}_O \end{cases}, \tag{2.14}$$

the coordinates of cell P in the optical frame are

$$\begin{cases} x(u_e, \phi) = x_{Lr} + (D-R)\cos\phi - s\sin\phi \\ y(u_e, \phi) = y_{Lr} + (D-R)\sin\phi + s\cos\phi \\ z(v_e) = z_{Lr} + t \end{cases}. \tag{2.15}$$

2.2 Mathematical foundations

The position of the source is denoted by $\boldsymbol{S} = \vec{OS}$. In figure 2.2, the vector $\boldsymbol{r}_i$ corresponds to the unit direction vector of the line $\mathcal{L}_i$. From equation (1.5), ideal unnoisy projection can be expressed as the divergent-beam transform of the volume :

$$g_i = \int_{\mathcal{L}_i} f(\boldsymbol{r})dl = \int_0^{+\infty} f(\boldsymbol{S} + \sigma\boldsymbol{r}_i)d\sigma \triangleq \mathcal{D}f(\boldsymbol{r}_i, \boldsymbol{S}) \tag{2.16}$$

where $\mathcal{D}f(\boldsymbol{r}_i, \boldsymbol{S})$ denotes the divergent-beam transform of f [SK85]. The X-ray transform (XT) performs the integration over $\mathbb{R}$ [SK85, Rod02] :

$$\mathcal{X}f(\boldsymbol{r}_i, \boldsymbol{S}) = \int_{-\infty}^{+\infty} f(\boldsymbol{S} + \sigma\boldsymbol{r}_i)d\sigma = \mathcal{D}f(\boldsymbol{r}_i, \boldsymbol{S}) + \mathcal{D}f(-\boldsymbol{r}_i, \boldsymbol{S}). \tag{2.17}$$

Since the source is out of the volume, we have $f(\boldsymbol{S} + \sigma\boldsymbol{r}_i) = 0$ for $\sigma < 0$. Hence, $\mathcal{X}f(\boldsymbol{r}_i, \boldsymbol{S})$ is equal to $\mathcal{D}f(\boldsymbol{r}_i, \boldsymbol{S})$.

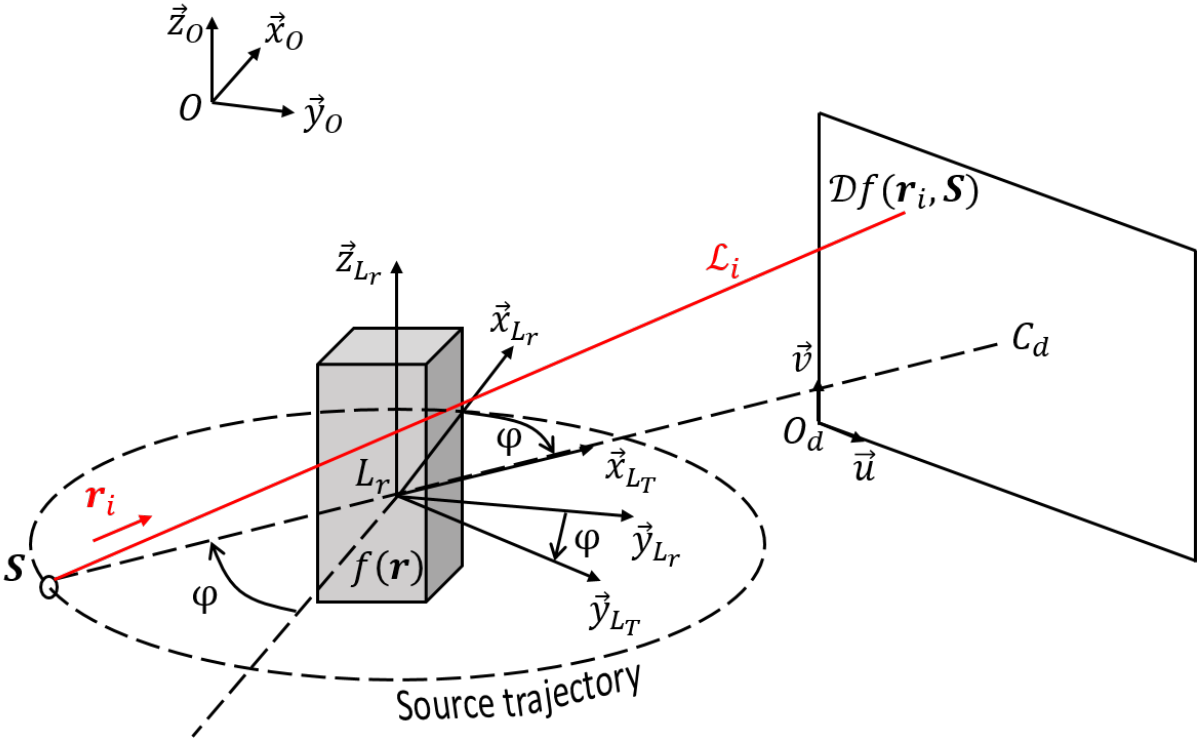


Figure 2.2: Divergent-beam transform

The historical fundamental transform in X-ray CT is the Radon transform (RT) [Rad17]. It performs integration of the volume over a hyperplane : for an unit vector $\boldsymbol{r} \in \mathbb{R}^n$, the Radon transform is defined as [SK85]

$$\mathcal{R}f(\boldsymbol{r}, \tau) \triangleq \int_{\boldsymbol{r}' : \langle \boldsymbol{r}', \boldsymbol{r} \rangle = \tau} f(\boldsymbol{r}') d\boldsymbol{r}', \forall \tau \in \mathbb{R}. \tag{2.18}$$

In 2D, hyperplanes correspond to lines, so the Radon transform is the same as the X-ray transform. For all unit vector $\boldsymbol{r} \in \mathbb{R}^2$, we define an orthogonal unit vector $\boldsymbol{r}_\perp$ such that $\langle \boldsymbol{r}_\perp, \boldsymbol{r} \rangle = 0$. For all $\tau \in \mathbb{R}$ and $\boldsymbol{r} \in \mathbb{R}^2$, the Radon transform performs an integration over a hyperplane which is the set of vectors $\boldsymbol{r}'$ such that $\langle \boldsymbol{r}', \boldsymbol{r} \rangle = \tau$. Hence, the Radon transform reads [Rod02]

$$\mathcal{R}f(\boldsymbol{r}, \tau) = \int_{\boldsymbol{r}' : \langle \boldsymbol{r}', \boldsymbol{r} \rangle = \tau} f(\boldsymbol{r}') d\boldsymbol{r}' = \int_{\boldsymbol{r}' \in \boldsymbol{r}^\perp} f(\tau \boldsymbol{r} + \boldsymbol{r}') d\boldsymbol{r}' = \int_{-\infty}^{+\infty} f(\tau \boldsymbol{r} + \sigma \boldsymbol{r}_\perp) d\sigma = \mathcal{X}f(\boldsymbol{r}_\perp, \tau \boldsymbol{r}). \tag{2.19}$$

On the opposite, in nD, for $n > 2$, the Radon transform and the X-ray transform are not equal. Indeed, in 3D, the X-ray transform is an integration over a line, while the Radon transform performs an integration over a plane.

Defining, for all $\tau \in \mathbb{R}$, $\boldsymbol{S} \in \mathbb{R}^n$ and $\boldsymbol{r} \in \mathbb{R}^n$ such that $\|\boldsymbol{r}\| = 1$,

$$\mathcal{R}_{\boldsymbol{r}} f(\tau) = \mathcal{R}f(\boldsymbol{r}, \tau), \tag{2.20}$$

projection-slice theorem 2.2.1, also called the Fourier slice theorem, provides a direct relation between the Fourier transform of the volume and the Fourier transform of $\mathcal{R}_{\boldsymbol{r}} f$ [Rod02].

Theorem 2.2.1. ***Projection-slice theorem :***

For $f : \mathbb{R}^n \to \mathbb{R}$, we have, for all $\bar{\tau} \in \mathbb{R}$, $\bar{\boldsymbol{S}} \in \mathbb{R}^n$ and $\boldsymbol{r} \in \mathbb{R}^n$ such that $\|\boldsymbol{r}\| = 1$:

$$\mathcal{F}\mathcal{R}_{\boldsymbol{r}} f(\bar{\tau}) = (2\pi)^{\frac{n-1}{2}} \mathcal{F}f(\bar{\tau}\boldsymbol{r}), \tag{2.21}$$

where

$$\mathcal{F}f(\bar{\boldsymbol{S}}) = \frac{1}{(2\pi)^{\frac{n}{2}}} \int_{\mathbb{R}^n} f(\boldsymbol{r}') \, \mathrm{e}^{-i\langle \bar{\boldsymbol{S}}, \boldsymbol{r}' \rangle} \, d\boldsymbol{r}', \forall \bar{\boldsymbol{S}} \tag{2.22}$$

is the Fourier transform of f.

For the 2D fan-beam case, the Radon transform can be inverted using Radon inversion formula [Rad17, HLN76]. Concerning the divergent-beam transform in 3D, Tuy derived an inversion formula [Tuy83, Rod02], which is valid if the necessary condition in theorem 2.2.2 is fulfilled.

Theorem 2.2.2. ***Tuy's condition [Tuy83]***

Divergent-beam transform $\mathcal{D}f(., \boldsymbol{S})$ is invertible from projections acquired along a bounded trajectory of the source $\boldsymbol{S}$ if every plane crossing the to-be-imaged region-of-interest also crosses the trajectory of $\boldsymbol{S}$ in a non-tangential way.

As shown in figure 2.2, the trajectory of the source is circular for the axial cone-beam case. Consequently, each plane of equation $z \neq z_{L_r}$ crossing the field-of-view (for instance $z = z_{L_r} + 1$), does not intercept the trajectory of the source. Therefore, Tuy's condition is not satisfied. As a result, only approximate reconstruction can be achieved in 3D axial cone-beam CT.

2.3 Analytical reconstruction methods

Analytical reconstruction methods are based on the mathematical fundations presented in section 2.3 [Rad17, HLN76, SK85]. Historically, many analytical methods have been developed for fan-beam CT, illustrated in figure 2.3, in order to reconstruct a 2D slice of an object. For the 2D fan-beam case, applying Radon inversion formula is a straightforward way to perform the reconstruction [HLN76]. Nevertheless, for this purpose, using projection-slice theorem 2.2.1 is also a very simple way, as shown in figure 2.4. Since it requires much less computations than applying Radon inversion formula [PK83], the procedure illustrated in figure 2.4 has been very popular [Mer76, SWPH81a, SWPH81b]. The main problem with this method is that the projection-slice theorem gives the Fourier transform of $f(\boldsymbol{r})$ on a polar grid, while, in order to obtain $f(\boldsymbol{r})$ sampled on the grid of voxels after having applied the Fourier transform inversion, it is necessary to have it on a Cartesian grid. Going from polar to Cartesian grid in Fourier domain requires an interpolation step, as illustrated in figure 2.5. This step is very difficult to perform since imprecisions in the interpolation cause artifacts in the reconstruction [SWPH81a, SWPH81b, JMNM91].

As an alternative to the projection-slice theorem, the reconstruction for the 2D fan-beam case can be performed by applying Radon inversion formula [Rad17, HLN76, FDK84] :

$$f(x, y) = \frac{1}{4\pi^2} \int_{\phi=0}^{2\pi} \mathrm{d}\phi \int_{s=-\infty}^{+\infty} \frac{\mathrm{d}s}{x\cos\phi + y\sin\phi - s} \frac{\partial}{\partial s} g(s, \phi). \tag{2.23}$$

Defining the function $g_\phi : s \mapsto g(s, \phi)$, we see that the integral over s in Radon inversion formula (2.23) is a convolution, evaluated at $(x\cos\phi + y\sin\phi)$, between the first derivative of g_ϕ and $s \mapsto \frac{1}{s}$: this convolution is the Hilbert transform $\mathfrak{H}$ of $\frac{\partial}{\partial s} g_\phi$ [Rod02, RRT+18] :

$$\mathfrak{H}\left[\frac{\partial}{\partial s} g_\phi\right](s') = \frac{1}{\pi}\left[\frac{1}{s} * \frac{\partial}{\partial s} g_\phi\right](s') = \frac{1}{\pi} \int_{s=-\infty}^{+\infty} \frac{\mathrm{d}s}{s' - s} \frac{\partial}{\partial s} g_\phi(s) \tag{2.24}$$

where $s' = x\cos\phi + y\sin\phi$ in Radon inversion formula (2.23). Going to Fourier domain, Radon inversion formula

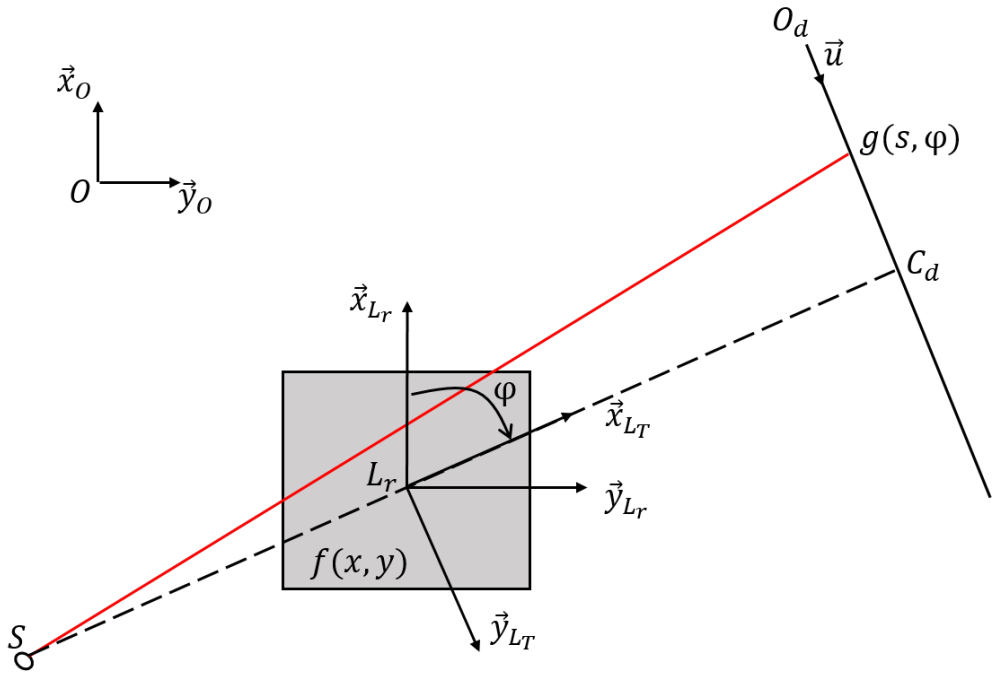


Figure 2.3: Fan-beam CT

(2.23) reads [FDK84]

$$\begin{aligned} f(x,y) &= \frac{1}{4\pi^2}\int_{\phi=0}^{2\pi} \mathrm{d}\phi \left[\frac{1}{s} * \frac{\partial}{\partial s} g(s,\phi)\right](x\cos\phi + y\sin\phi) \\ &= \frac{1}{8\pi^3}\int_{\phi=0}^{2\pi} \mathrm{d}\phi \int_{\omega=-\infty}^{+\infty} i\omega \times \mathcal{F}g_\phi(\omega) \times \mathcal{F}\left[s \mapsto \frac{1}{s}\right](\omega) \times \exp\left[i\omega(x\cos\phi + y\sin\phi)\right] \mathrm{d}\omega \\ &= \frac{1}{8\pi^3}\int_{\phi=0}^{2\pi} \mathrm{d}\phi \int_{\omega=-\infty}^{+\infty} i\omega \times \mathcal{F}g_\phi(\omega) \times (-i\pi\,\mathrm{sign}(\omega)) \times \exp\left[i\omega(x\cos\phi + y\sin\phi)\right] \mathrm{d}\omega \\ &= \frac{1}{8\pi^2}\int_{\phi=0}^{2\pi} \mathrm{d}\phi \int_{\omega=-\infty}^{+\infty} |\omega| \times \mathcal{F}g_\phi(\omega) \times \exp\left[i\omega(x\cos\phi + y\sin\phi)\right] \mathrm{d}\omega. \end{aligned} \tag{2.25}$$

Formula (2.25) gives an intuitive interpretation of what Radon inversion formula does. The values of the projections along each ray are backpropagated and accumulated in each pixel, which corresponds to the backprojection operation. Before this backpropagation, a ramp filter, of frequency response $\omega \mapsto |\omega|$, is applied on each projection [FDK84]. The ideal ramp filter cannot be implemented in practice, so a cutoff frequency ω_c needs to be defined [WWH05] :

$$f(x,y) \approx \frac{1}{8\pi^2}\int_{\phi=0}^{2\pi} \mathrm{d}\phi \int_{\omega=-\omega_c}^{\omega_c} |\omega| \times \mathcal{F}g_\phi(\omega) \times \exp\left[i\omega(x\cos\phi + y\sin\phi)\right] \mathrm{d}\omega. \tag{2.26}$$

Therefore, the application of Radon inversion formula corresponds to a Filtered Back-Projection (FBP).

As mentioned in the previous section, in 3D, the Tuy's conditions are not satisfied for the cone-beam acquisition case. Hence, no inversion formula is available. The FDK method [FDK84] performs an approximate inversion by applying FBP to each row of the projections, as if it were the result of a 2D fan-beam acquisition. By this method, the middle slice is exactly reconstructed, while the error for the other slices grows as the distance from the middle slice increases [Lon11]. Short formulae for the FDK method can be found in [Lon11, Chapter 2][RRT+18]. The FDK method is a very practical and fast cone-beam reconstruction algorithm. For this reason, it is widely used in medicine and in industry. In particular, the FDK method is implemented in the tomographs bought by SAFRAN. Nevertheless, since the frequency response of the ramp filter $\omega \mapsto |\omega|$ goes to infinity as ω increases, the ramp filter amplifies the noise. Consequently, the FDK method is not optimal to perform the reconstruction from noisy projections.

Recently, a new analytical method has been proposed, called the backprojection-filtration (BPF) method [RRT+18]. It uses the fact that the ramp filter can be decomposed as a spatial derivative along each row of the projections, followed

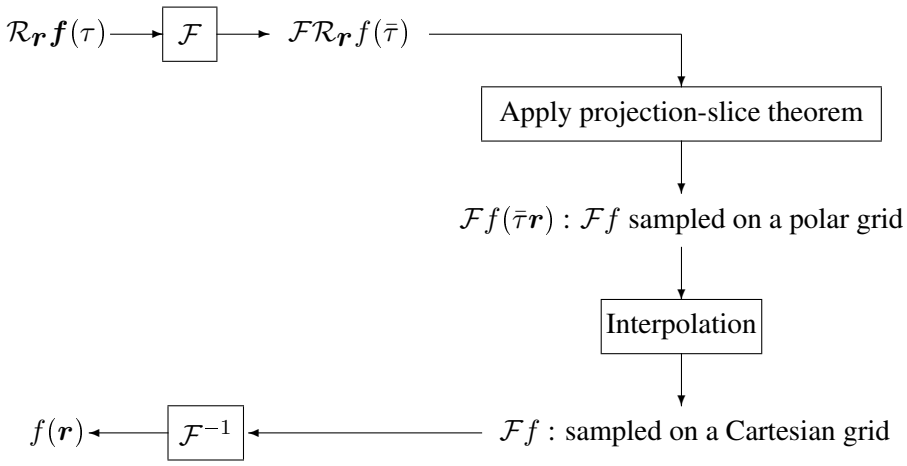


Figure 2.4: Radon transform inversion by the use of projection-slice theorem

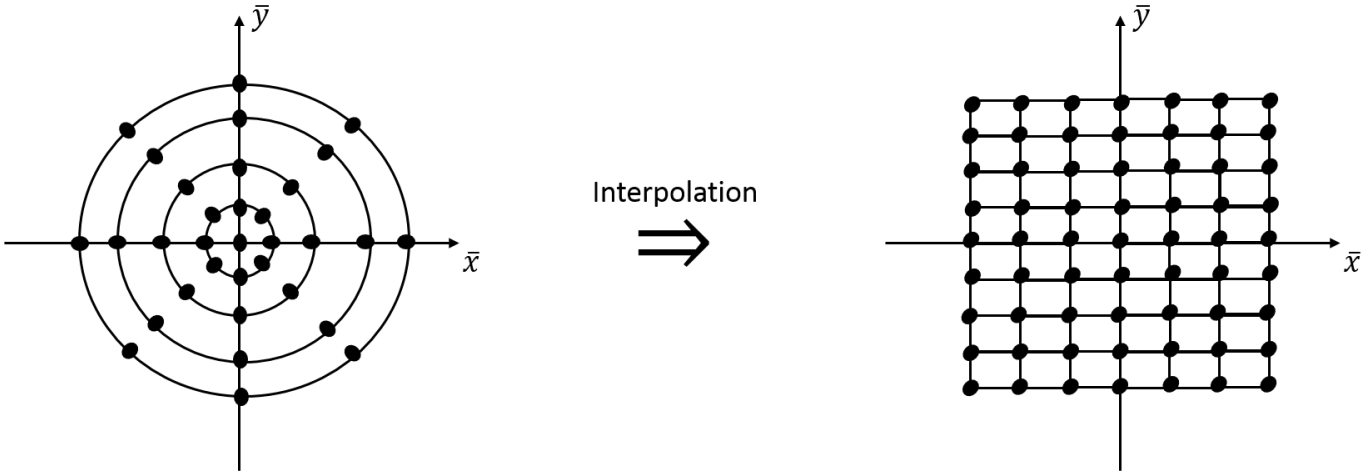


Figure 2.5: Interpolation step from polar to Cartesian grid in Fourier domain [Rod02]

by a Hilbert transform. The BPF method reverses the applications of the Hilbert transform and the backprojection operation in the FDK reconstruction formula. Like FBP, the BPF method is exact in the 2D fan-beam case [RRT$^+$18]. Differently from the FDK method, BPF enables to apply directional filtering according to which backprojected direction the Hilbert transform is applied [RRT$^+$18]. The main interest of BPF is that it performs better reconstructions of regions-of-interest than FDK [RRT$^+$18].

2.4 Algebraic reconstruction methods

Algebraic reconstruction methods are based on the discretized formulation of the reconstruction problem presented in chapter 1. They are called iterative reconstruction methods since they refine the estimation of the volume after having obtained a first reconstruction. The first algebraic reconstruction methods which have been proposed estimate the volume which is the most likely to have given the data $\boldsymbol{g}$. In this sense, they are maximum-likelihood techniques :

$$\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} p(\boldsymbol{g}|\boldsymbol{f}). \tag{2.27}$$

Among these maximum-likelihood methods, ART (Algebraic Reconstruction Technique) [GBH70] tries to match the real and the theoretical projections $\boldsymbol{g}$ and $\boldsymbol{H}\boldsymbol{f}$ by solving the linear system $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f}$. The estimation procedure

used in ART considers each ray separately, so one iteration of ART contains M updates of the volume. Due to the ill-conditioning, it gives poor reconstruction results [Gil72]. After ART, SIRT (Simultaneous Iterative Reconstruction Technique) has been proposed : it updates the volume considering all the projections simultaneously, leading to enhanced reconstruction quality [Gil72]. SART (Simultaneous Algebraic Reconstruction Technique) aims at improving the convergence speed of SIRT by proposing a different update formula for the volume [AK84].

SART takes into account discretization uncertainties in the discretized formulation of the reconstruction problem

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\zeta}. \tag{2.28}$$

Although forward model (2.28) looks the same as forward model (1.9), its mathematical sense is rather different, since SART introduces uncertainties $\boldsymbol{\zeta}$ to take into account errors inherent to discretization [AK84], while uncertainties $\boldsymbol{\zeta}$ in forward model (1.9) are physically justified by statistical noise [SB93], as explained in chapter 1. As a result, SART considers that the projections are equally important and performs a least-squares estimation :

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|_2^2, \tag{2.29}$$

which is equivalent to consider uncertainties $\boldsymbol{\zeta}$ as Gaussian with same variance v_ζ. Thanks to the approach provided in [SB93], we know that uncertainties have different variances

$$p(\zeta_i|v_{\zeta_i}) = \mathcal{N}(\zeta_i|0, v_{\zeta_i}), \forall i, \tag{2.30}$$

where

$$v_{\zeta_i} = \frac{I_i}{(I_i - r_i)^2}, \forall i, \tag{2.31}$$

as shown in appendix A. Consequently, it is more appropriate, from a maximum-likelihood perspective, to perform weighted least-squares estimation :

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|_{\boldsymbol{V}_\zeta}^2 \tag{2.32}$$

where $\boldsymbol{V}_\zeta = \text{diag}\,[\boldsymbol{v}_\zeta]$ and $\|\boldsymbol{y}\|_{\boldsymbol{V}}^2 = \boldsymbol{y}^T\boldsymbol{V}^{-1}\boldsymbol{y}$ for all $\boldsymbol{y}$ and fixed diagonal matrix $\boldsymbol{V}$ with positive and non-zeros diagonal coefficients. The derivation of expression (2.31) in [SB93] does not take into account scattering and beam-hardening. For this reason, expression (2.31) appears as rather suboptimal in order to weight the projections.

2.5 Model-based iterative reconstruction (MBIR) methods

In the algebraic reconstruction methods which have been presented in section 2.4, the estimation of the volume is only based on the projections. Due to the ill-conditioning, these methods are very sensitive to the noise introduced by physical uncertainties such as scattering, beam-hardening or cracks in the cells. The estimation needs to be made more robust by including a prior model on the volume in the reconstruction algorithm [MDD88]. While this is difficult to do in analytic reconstruction methods, algebraic reconstruction methods provide a natural framework to introduce such prior, which leads to Model-Based Iterative Reconstruction (MBIR) methods. MBIR methods perform Maximum A Posteriori (MAP) estimation : they aim at maximizing posterior distribution of the volume $p(\boldsymbol{f}|\boldsymbol{g})$, which is given by Bayes' rule :

$$p(\boldsymbol{f}|\boldsymbol{g}) = \frac{p(\boldsymbol{g}|\boldsymbol{f})p(\boldsymbol{f})}{p(\boldsymbol{g})} \tag{2.33}$$

where

- $p(\boldsymbol{g}|\boldsymbol{f})$ is the likelihood,
- $p(\boldsymbol{f})$ is the prior distribution of volume $\boldsymbol{f}$,
- $p(\boldsymbol{g})$ is the evidence.

Since evidence $p(\boldsymbol{g})$ does not depend on the volume, MAP estimation is equivalent to

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} -\ln(p(\boldsymbol{g}|\boldsymbol{f})) - \ln(p(\boldsymbol{f})). \tag{2.34}$$

Given the likelihood and removing constant terms, this leads to penalized weighted least-squares (PWLS) [SB93, TSBH07] :

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta} - \ln(p(\boldsymbol{f})). \tag{2.35}$$

Depending on the physical properties of the volume to reconstruct, many prior models have been proposed, leading to different MBIR algorithms. A general form for the prior distribution on $\boldsymbol{f}$ is [KPTF13, KRF15, MF15]

$$p(\boldsymbol{f}) \propto \exp(-R(\boldsymbol{D}\boldsymbol{f})), \tag{2.36}$$

where $R : \mathbb{R} \to \mathbb{R}$ is a convex potential function and $\boldsymbol{D}$ is a linear transform (or a concatenation of linear transforms) on the volume. Prior $p(\boldsymbol{f})$ in (2.36) can be possibly improper (i.e. its integral over $\mathbb{R}^N$ is $+\infty$), as long as the posterior distribution remains proper. Given general form (2.36), PWLS reads

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta} + R(\boldsymbol{D}\boldsymbol{f}). \tag{2.37}$$

Depending on chosen R and $\boldsymbol{D}$, optimization (2.37) is more or less difficult to achieve. All possible priors and all possible ways to solve (2.37) result in a wide variety of MBIR algorithms.

In order to enhance image quality, the prior model very often encourages volumes divided in compact and smooth regions, which leads to choose an edge-preserving potential R [KRF15]. Many regularizations aim at enforcing sparsity of the volume in $\boldsymbol{D}$-transform domain :

$$R(\boldsymbol{D}\boldsymbol{f}) = \lambda\|\boldsymbol{D}\boldsymbol{f}\|_p^q \tag{2.38}$$

where $0 \leqslant p \leqslant 2$, $1 \leqslant q \leqslant 2$ and $\lambda \geqslant 0$. Parameter λ tunes the tradeoff between the data-matching term and the regularization term. In many cases, optimal λ is difficult to find. As a result, fixing λ requires repeated experiments, which can be tedious due to the high dimension in 3D CT.

One standard prior model is Total Variation (TV) regularization [SJP+12]. Denoting by ∇ the operator concatenating the first-order finite differences ∇_ι in each considered direction $\iota = 1, \ldots, I$ of the 3D space

$$\nabla\boldsymbol{f} = \begin{pmatrix} \nabla_1\boldsymbol{f} \\ \vdots \\ \nabla_\iota\boldsymbol{f} \\ \vdots \\ \nabla_I\boldsymbol{f} \end{pmatrix}, \tag{2.39}$$

the anisotropic total variation regularization is given by

$$R(\boldsymbol{D}\boldsymbol{f}) = \lambda\sum_j \left(\sum_{\iota=1}^{I} \left|(\nabla_\iota\boldsymbol{f})_j\right|\right) = \lambda\|\nabla\boldsymbol{f}\|_1, \tag{2.40}$$

while the isotropic total variation regularization reads

$$R(\boldsymbol{D}\boldsymbol{f}) = \lambda\sum_j \sqrt{\sum_{\iota=1}^{I} (\nabla_\iota\boldsymbol{f})_j^2}. \tag{2.41}$$

In this thesis, we deal with $I = 3$ directions which correspond to the first-order neighbours. In order to reconstruct piecewise-constant volumes, Potts model [BVZ01]

$$R(\boldsymbol{D}\boldsymbol{f}) = \lambda\sum_{\iota=1}^{I} \|\nabla_\iota\boldsymbol{f}\|_0 \tag{2.42}$$

has also been considered. In this case, reconstruction problem (2.37) has been solved by joint reconstruction and segmentation algorithms [RR07, SWFU15, SRUW17]. Transform $\boldsymbol{D}$ can be chosen different from the gradient. For $\boldsymbol{D}$ being the second-order derivative, sparse contours are favoured in the reconstruction. Other prior models focus on sparsity in a wavelet domain [VGVH$^+$13, NHBF16, WMDG17]. An optimal sparse dictionary representation can be also pre-learnt [XYM$^+$12, CXLF17, ZRLF18]. The main difficulty of methods based on dictionary-learning is that the learning dataset has to contain sufficiently varied reconstructions of good quality. In order to reduce the number of required volumes, patch-based learning has been investigated [ZRLF18].

For a fixed regularization, the reconstruction can be accelerated by dividing the projections into several Ordered Subsets (OS) [HL94, EF99, KPTF13]. In this case, at each iteration of the reconstruction algorithm, each subset is considered one after the other. In practice, OS methods show improved convergence speed but their convergence properties remain uncertain [MF15]. In particular, dealing with too small subsets make the algorithm diverge [KPTF13]. OS strategy can be combined with other optimization techniques such as separable quadratic surrogate [KPTF13] or Nesterov's momentum [KRF15].

Parameters p and q tune the sparsity of $\boldsymbol{D}\boldsymbol{f}$. For $p = q$ and $1 < p \leqslant 2$, minimization problem (2.37) is convex and differentiable, which enables to use simple or conjugate gradient descent as reconstruction algorithms [She94]. The sparsity in the transform domain increases as parameter p tends to zero. In this case, performing optimization (2.37) requires more complex optimization methods since the criterion to minimize in (2.37) is not differentiable for $p \leqslant 1$. In the next section, we review some of these methods for the case of anisotropic TV regularization.

2.6 MBIR methods with Total Variation (TV) regularization

For TV-penalized weighted least-squares, the criterion to minimize is

$$J(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta} + \lambda\|\nabla\boldsymbol{f}\|_1 = \Phi(\boldsymbol{f}) + \Psi(\boldsymbol{f}), \tag{2.43}$$

where $\boldsymbol{V}_\zeta = \operatorname{diag}[\boldsymbol{v}_\zeta]$. Variances $v_{\zeta_i}, \forall i$, are given by (2.31). $\Phi(\boldsymbol{f})$ is the data-matching term :

$$\Phi(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta}, \tag{2.44}$$

and $\Psi(\boldsymbol{f})$ is the regularization :

$$\Psi(\boldsymbol{f}) = \lambda\|\nabla\boldsymbol{f}\|_1. \tag{2.45}$$

The difficulty to perform optimization

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} J(\boldsymbol{f}) \tag{2.46}$$

is that $\Psi(\boldsymbol{f})$ is not differentiable. Hence, gradient descent methods are not applicable. Nevertheless, $\Psi(\boldsymbol{f})$ is still convex, which enables to apply several convex optimization algorithms we now describe here. Many of these algorithms use proximity operators, defined, for a convex function h, by

$$\operatorname{prox}_h(\boldsymbol{f}) = \arg\min_{\boldsymbol{f}'} \left\{\frac{1}{2}\|\boldsymbol{f}' - \boldsymbol{f}\|_2^2 + h(\boldsymbol{f}')\right\}. \tag{2.47}$$

2.6.1 Alternating Direction Method of Multipliers (ADMM)

The idea of Alternating Direction Method of Multipliers (ADMM) [BPC$^+$11, Ess09] is to divide the optimization problem (2.46) into two simpler sub-problems. This is done by performing a variable-splitting into the expression of the criterion :

$$J(\boldsymbol{f}) = \Phi(\boldsymbol{f}) + \Psi(\boldsymbol{f}) = \Phi(\boldsymbol{f}) + \tilde{\Psi}(\nabla\boldsymbol{f}) = \Phi(\boldsymbol{f}) + \tilde{\Psi}(\boldsymbol{z}) \tag{2.48}$$

where $\boldsymbol{z} = \nabla\boldsymbol{f}$ and

$$\tilde{\Psi}(\boldsymbol{z}) = \lambda\|\boldsymbol{z}\|_1. \tag{2.49}$$

ADMM solves the reconstruction problem with respect to $\boldsymbol{f}$ and $\boldsymbol{z}$:

$$(\hat{\boldsymbol{f}}, \hat{\boldsymbol{z}}) = \arg\min_{\boldsymbol{f},\boldsymbol{z}} \left\{ \Phi(\boldsymbol{f}) + \tilde{\Psi}(\boldsymbol{z}) \right\} \text{ such that } \boldsymbol{z} = \nabla \boldsymbol{f} \tag{2.50}$$

which is equivalent to problem (2.46) thanks to constraint $\boldsymbol{z} = \nabla \boldsymbol{f}$. In order to deal with this constraint, ADMM introduces the augmented Lagragian :

$$L_\mu(\boldsymbol{f}, \boldsymbol{z}; \boldsymbol{b}) = \Phi(\boldsymbol{f}) + \tilde{\Psi}(\boldsymbol{z}) + \boldsymbol{b}^T(\nabla \boldsymbol{f} - \boldsymbol{z}) + \frac{\mu}{2}\|\nabla \boldsymbol{f} - \boldsymbol{z}\|_2^2 \tag{2.51}$$

where $\boldsymbol{b}$ is the vector of Lagrange multipliers (or dual variables), and $\mu > 0$ is the penalty parameter [BPC$^+$11]. The augmented Lagrangian is minimized with respect to primal variables $(\boldsymbol{f}, \boldsymbol{z})$ and maximized with respect to dual variables $\boldsymbol{b}$. This is summarized by saddle-point primal-dual formulation [BPC$^+$11]

$$\min_{\boldsymbol{f},\boldsymbol{z}} \max_{\boldsymbol{b}} L_\mu(\boldsymbol{f}, \boldsymbol{z}; \boldsymbol{b}). \tag{2.52}$$

In order to solve (2.52), ADMM performs one primal descent with respect to $\boldsymbol{f}$ and $\boldsymbol{z}$ and one dual ascent with respect to $\boldsymbol{b}$, as summarized in algorithm 1. Steps for $\boldsymbol{f}$ and $\boldsymbol{z}$ updates read as one step of alternate minimization of the augmented Lagrangian in the primal space. If several steps of this alternate minimization are performed, then algorithm 1 looks like Split-Bregman method (SBM) [GO09, Ess09]. Considering the scaled dual variables

$$\tilde{\boldsymbol{b}} = \frac{\boldsymbol{b}}{\mu} \tag{2.53}$$

instead of the unscaled dual variables $\boldsymbol{b}$, the augmented Lagragian can be rewritten as [BPC$^+$11]

$$L_\mu(\boldsymbol{f}, \boldsymbol{z}; \tilde{\boldsymbol{b}}) = \Phi(\boldsymbol{f}) + \tilde{\Psi}(\boldsymbol{z}) + \frac{\mu}{2}\|\nabla \boldsymbol{f} - \boldsymbol{z} + \tilde{\boldsymbol{b}}\|_2^2 - \frac{\mu}{2}\|\tilde{\boldsymbol{b}}\|_2^2. \tag{2.54}$$

Solving primal-dual problem (2.52) with respect to $(\boldsymbol{f}, \boldsymbol{z}; \tilde{\boldsymbol{b}})$ instead of $(\boldsymbol{f}, \boldsymbol{z}; \boldsymbol{b})$ leads to the scaled formulation of ADMM given in algorithm 2. In this algorithm, the step for $\boldsymbol{z}$ update uses the proximity operator of $\frac{1}{\mu}\tilde{\Psi} = \frac{\lambda}{\mu}\|.\|_1$, which is the soft thresholding operator [BT09] :

$$\left(\text{prox}_{\frac{1}{\mu}\tilde{\Psi}}(\tilde{\boldsymbol{z}}^{(t)})\right)_j = \text{sign}(\tilde{z}_j) \max\left(|\tilde{z}_j| - \frac{\lambda}{\mu}, 0\right), \forall j, \tag{2.55}$$

where

$$\tilde{\boldsymbol{z}}^{(t)} = \nabla \boldsymbol{f}^{(t+1)} + \tilde{\boldsymbol{b}}^{(t)}. \tag{2.56}$$

Algorithm 1 ADMM algorithm [BPC$^+$11]

Initialize $\boldsymbol{f}^{(0)}$, $\boldsymbol{z}^{(0)}$ (primal variables)
Initialize $\boldsymbol{b}^{(0)} = \boldsymbol{0}$ (dual variable)
for $t = 0, \ldots, t_{max} - 1$ **do**
  Primal descent : $\boldsymbol{f}^{(t+1)} = \arg\min_{\boldsymbol{f}} L_\mu\left(\boldsymbol{f}, \boldsymbol{z}^{(t)}; \boldsymbol{b}^{(t)}\right)$
  Primal descent : $\boldsymbol{z}^{(t+1)} = \arg\min_{\boldsymbol{z}} L_\mu\left(\boldsymbol{f}^{(t+1)}, \boldsymbol{z}; \boldsymbol{b}^{(t)}\right)$
  Dual ascent : $\boldsymbol{b}^{(t+1)} = \boldsymbol{b}^{(t)} + \mu\left(\nabla \boldsymbol{f}^{(t+1)} - \boldsymbol{z}^{(t+1)}\right)$
end for

As emphasized in the unscaled version of ADMM in algorithm 1, penalty parameter $\mu > 0$ acts as a fixed stepsize in the dual space. One advantage of ADMM is that the algorithm converges whatever the value of this parameter is [BPC$^+$11]. In particular, the primal residuals, which, in our case, correspond to

$$\boldsymbol{r}^{(t)} = \nabla \boldsymbol{f}^{(t)} - \boldsymbol{z}^{(t)}, \tag{2.57}$$

Algorithm 2 Scaled ADMM algorithm [BPC+11]

Initialize $\boldsymbol{f}^{(0)}$, $\boldsymbol{z}^{(0)}$ (primal variables)
Initialize $\tilde{\boldsymbol{b}}^{(0)} = \mathbf{0}$ (dual variable)
for $t = 0, \ldots, t_{max} - 1$ **do**
  Primal descent : $\boldsymbol{f}^{(t+1)} = \arg\min_{\boldsymbol{f}} \left\{ \Phi(\boldsymbol{f}) + \frac{\mu}{2} \|\nabla \boldsymbol{f} - \boldsymbol{z}^{(t)} + \tilde{\boldsymbol{b}}^{(t)}\|_2^2 \right\}$
  Primal descent : $\boldsymbol{z}^{(t+1)} = \arg\min_{\boldsymbol{z}} \left\{ \tilde{\Psi}(\boldsymbol{z}) + \frac{\mu}{2} \|\nabla \boldsymbol{f}^{(t+1)} - \boldsymbol{z} + \tilde{\boldsymbol{b}}^{(t)}\|_2^2 \right\} = \text{prox}_{\frac{1}{\mu}\tilde{\Psi}} \left(\nabla \boldsymbol{f}^{(t+1)} + \tilde{\boldsymbol{b}}^{(t)} \right)$
  Dual ascent : $\tilde{\boldsymbol{b}}^{(t+1)} = \tilde{\boldsymbol{b}}^{(t)} + \nabla \boldsymbol{f}^{(t+1)} - \boldsymbol{z}^{(t+1)}$
end for

converge to zero as $t \to +\infty$. Nevertheless, in practice, since a finite number t_{max} of iterations is performed, one may want to achieve the convergence as fast as possible in order to be sure that the primal residuals are sufficiently small at the end of the algorithm, so the constraint $\nabla \boldsymbol{f} = \boldsymbol{z}$ is satisfied within a desired tolerance [BPC+11, sec. 3.3.1]. From this perspective, one has to apply a careful selection strategy of μ in order to avoid slow convergence speed [GTSJ15]. For this purpose, the penalty parameter selection problem is covered in [GTSJ15] but only for the case of $\mathcal{L}_2$-regularization and not for TV. Consequently, finding optimal penalty parameter μ for ADMM in the case of TV-regularization is a difficult and non-trivial problem.

2.6.2 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

As shown in algorithm 2, ADMM does not need to have Φ or $\tilde{\Psi}$ differentiable. If one of these functions is differentiable, as it is the case for Φ in the TV-penalized weighted least-squares problem (2.43), then it is interesting to consider the Iterative Shrinkage-Thresholding Algorithm (ISTA) and its fast version, the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [BT09]. ISTA assumes that the gradient of Φ, in criterion (2.43), is L_Φ-Lipschitz continuous

$$\|\nabla \Phi(\boldsymbol{f}_1) - \nabla \Phi(\boldsymbol{f}_2)\|_2 \leqslant L_\Phi \|\boldsymbol{f}_1 - \boldsymbol{f}_2\|_2. \tag{2.58}$$

In our case, this is true, with $L_\Phi = |||\boldsymbol{H}^T \boldsymbol{V}_\zeta^{-1} \boldsymbol{H}|||_2$. $|||.|||_2$ denotes Frobenius norm :

$$|||\boldsymbol{A}|||_2 = \text{Tr}\left[\boldsymbol{A}^T \boldsymbol{A}\right]^{\frac{1}{2}}. \tag{2.59}$$

Given (2.58), at iteration t, ISTA considers

$$Q_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) = \Phi(\boldsymbol{f}^{(t)}) + \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)^T \left(\boldsymbol{f} - \boldsymbol{f}^{(t)}\right) + \frac{L}{2} \|\boldsymbol{f} - \boldsymbol{f}^{(t)}\|_2^2 \tag{2.60}$$

which is greater than $\Phi(\boldsymbol{f})$ for $L \geqslant L_\Phi$ [BT09]. Iterate $\boldsymbol{f}^{(t+1)}$ of ISTA is computed by

$$\boldsymbol{f}^{(t+1)} = \arg\min_{\boldsymbol{f}} \left\{ \tilde{Q}_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) \right\} \tag{2.61}$$

where

$$\tilde{Q}_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) = Q_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) + \Psi(\boldsymbol{f}). \tag{2.62}$$

Since $Q_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) \geqslant \Phi(\boldsymbol{f})$ for $L \geqslant L_\Phi$, we have $\tilde{Q}_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) \geqslant J(\boldsymbol{f})$. Hence, ISTA can be seen as a majorize-minimize algorithm. Given that

$$\begin{aligned}
&Q_L(\boldsymbol{f}, \boldsymbol{f}^{(t)}) = \Phi(\boldsymbol{f}^{(t)}) + \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)^T \left(\boldsymbol{f} - \boldsymbol{f}^{(t)}\right) + \frac{L}{2} \left\|\boldsymbol{f} - \boldsymbol{f}^{(t)}\right\|_2^2 \\
&= \Phi(\boldsymbol{f}^{(t)}) + 2 \times \frac{L}{2} \times \frac{1}{L} \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)^T \left(\boldsymbol{f} - \boldsymbol{f}^{(t)}\right) + \frac{L}{2} \left\|\boldsymbol{f} - \boldsymbol{f}^{(t)}\right\|_2^2 + \frac{L}{2} \left\|\frac{1}{L} \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)\right\|_2^2 - \frac{L}{2} \left\|\frac{1}{L} \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)\right\|_2^2 \\
&= \Phi(\boldsymbol{f}^{(t)}) + \frac{L}{2} \left\|\boldsymbol{f} - \boldsymbol{f}^{(t)} + \frac{1}{L} \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)\right\|_2^2 - \frac{L}{2} \left\|\frac{1}{L} \nabla \Phi\left(\boldsymbol{f}^{(t)}\right)\right\|_2^2
\end{aligned}$$

$$= \Phi(\boldsymbol{f}^{(t)}) + \frac{L}{2}\left\|\boldsymbol{f} - \left(\boldsymbol{f}^{(t)} - \frac{1}{L}\nabla\Phi\left(\boldsymbol{f}^{(t)}\right)\right)\right\|_2^2 - \frac{L}{2}\left\|\frac{1}{L}\nabla\Phi\left(\boldsymbol{f}^{(t)}\right)\right\|_2^2, \tag{2.63}$$

ISTA iterate can be rewritten, removing constant terms, as

$$\boldsymbol{f}^{(t+1)} = \arg\min_{\boldsymbol{f}}\left\{\frac{L}{2}\left\|\boldsymbol{f} - \left(\boldsymbol{f}^{(t)} - \frac{1}{L}\nabla\Phi\left(\boldsymbol{f}^{(t)}\right)\right)\right\|_2^2 + \Psi(\boldsymbol{f})\right\}. \tag{2.64}$$

Hence, $\boldsymbol{f}^{(t+1)}$ in ISTA can be expressed using the proximity operator of Ψ, as done in algorithm 3.

Algorithm 3 Iterative Shrinkage-Thresholding Algorithm (ISTA) [BT09]

Initialize $\boldsymbol{f}^{(0)}$
for $t = 0, \ldots, t_{max} - 1$ **do**
  $\boldsymbol{f}^{(t+1)} = \mathrm{prox}_{\frac{1}{L}\Psi}\left(\boldsymbol{f}^{(t)} - \frac{1}{L}\nabla\Phi\left(\boldsymbol{f}^{(t)}\right)\right)$
end for

The fast version FISTA adds two steps of Nesterov's acceleration technique [Nes83] to ISTA [BT09, CD15]. FISTA is summarized in algorithm 4. As we see, FISTA is more memory-costly than ISTA since it requires to store one further volume. The application of the proximity operator of $\frac{1}{L}\Psi$ corresponds to a soft thresholding operation

Algorithm 4 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [BT09]

Initialize $\boldsymbol{f}^{(0)}$
Initialize $\tilde{\boldsymbol{f}}^{(0)} = \boldsymbol{f}^{(0)}$, $\sigma_1 = 1$
for $t = 0, \ldots, t_{max} - 1$ **do**
  ISTA step : $\boldsymbol{f}^{(t+1)} = \mathrm{prox}_{\frac{1}{L}\Psi}\left(\tilde{\boldsymbol{f}}^{(t)} - \frac{1}{L}\nabla\Phi\left(\tilde{\boldsymbol{f}}^{(t)}\right)\right)$
  Nesterov's acceleration step : $\sigma_{t+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4\sigma_t^2}\right)$
  Nesterov's acceleration step : $\tilde{\boldsymbol{f}}^{(t+1)} = \boldsymbol{f}^{(t+1)} + \frac{\sigma_t - 1}{\sigma_{t+1}}\left(\boldsymbol{f}^{(t+1)} - \boldsymbol{f}^{(t)}\right)$
end for

in the case of $\mathcal{L}_1$-regularization $\Psi(\boldsymbol{f}) = \lambda\|\boldsymbol{f}\|_1$. With such a regularization, the proximal operator has closed-form expression given by (2.55). Unfortunately, this is not the case for TV-regularization $\Psi(\boldsymbol{f}) = \lambda\|\nabla\boldsymbol{f}\|_1$. As a result, computing $\boldsymbol{f}^{(t+1)}$ requires one to run sub-iterations in FISTA. For this computation, a dual approach has been proposed for image denoising [KF17]. Nevertheless, this approach would require to store too many variables for our 3D application.

2.6.3 Chambolle-Pock (CP) and Primal-Dual Frank-Wolfe (PDFW) algorithms

If the involved proximal operators have no closed-form expressions in ADMM, ISTA and FISTA, their computation requires to run subiterations. This can be seen as a drawback since, in doing so, the iterates $\boldsymbol{f}^{(t)}$ depend on the number of subiterations performed at each iteration of the algorithm. In order to avoid subiterations, Chambolle-Pock (CP) algorithm can be used as an alternative. Like ADMM, CP algorithm is a primal-dual method which alternates one step of primal descent and one step of dual ascent to solve a saddle-point problem. The difference with ADMM is that CP algorithm does not consider any Lagrangian or augmented Lagrangian : the saddle-point problem is directly derived from the minimization problem (2.46). A pillar for this is Fenchel-Legendre transform, defined, for a convex function h, by [BV04, chap. 3.3]

$$h^*(\boldsymbol{f}) = \max_{\boldsymbol{f}'}\left\{\langle\boldsymbol{f}, \boldsymbol{f}'\rangle - h(\boldsymbol{f}')\right\}. \tag{2.65}$$

Fenchel-Legendre transform h^* is also called the conjugate of h. We also define the biconjugate function of h by

$$h^{**}(\boldsymbol{f}) = \max_{\boldsymbol{f}'}\left\{\langle\boldsymbol{f}, \boldsymbol{f}'\rangle - h^*(\boldsymbol{f}')\right\}. \tag{2.66}$$

According to Fenchel-Moreau theorem, $h^{**} = h$ if and only if h is convex and lower-semicontinuous [BL06, chap. 4.2].

In order to apply Fenchel-Legendre transform and Fenchel-Moreau theorem to the reconstruction problem, we define

$$\tilde{\Phi}(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{g} - \boldsymbol{p}\|_2^2 \tag{2.67}$$

such that

$$\tilde{\Phi}\left(\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{f}\right) = \Phi(\boldsymbol{f}). \tag{2.68}$$

$\tilde{\Phi}$ given by (2.67) and $\tilde{\Psi}$ given by (2.49) are convex and continuous so Fenchel-Moreau theorem can be applied

$$\Phi(\boldsymbol{f}) = \tilde{\Phi}(\boldsymbol{H}\boldsymbol{f}) = \max_{\boldsymbol{p}}\left\{\langle \boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{f}, \boldsymbol{p}\rangle - \tilde{\Phi}^*(\boldsymbol{p})\right\} = \max_{\boldsymbol{p}}\left\{\langle \boldsymbol{f}, \boldsymbol{H}^T\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{p}\rangle - \tilde{\Phi}^*(\boldsymbol{p})\right\} \tag{2.69}$$

and

$$\Psi(\boldsymbol{f}) = \tilde{\Psi}(\nabla\boldsymbol{f}) = \max_{\boldsymbol{z}}\left\{\langle \nabla\boldsymbol{f}, \boldsymbol{z}\rangle - \tilde{\Psi}^*(\boldsymbol{z})\right\} = \max_{\boldsymbol{z}}\left\{\langle \boldsymbol{f}, \nabla^T\boldsymbol{z}\rangle - \tilde{\Psi}^*(\boldsymbol{z})\right\}, \tag{2.70}$$

where $\nabla^T = -\operatorname{div}$ [SJP$^+$12]. Replacing Φ and Ψ by (2.69) and (2.70) leads to the saddle-point problem [SJP$^+$12, OMBF18] :

$$\min_{\boldsymbol{f}} J(\boldsymbol{f}) = \min_{\boldsymbol{f}} \Phi(\boldsymbol{f}) + \Psi(\boldsymbol{f}) = \min_{\boldsymbol{f}}\max_{\boldsymbol{p},\boldsymbol{z}} \langle \boldsymbol{f}, \boldsymbol{H}^T\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{p} + \nabla^T\boldsymbol{z}\rangle - \tilde{\Phi}^*(\boldsymbol{p}) - \tilde{\Psi}^*(\boldsymbol{z}). \tag{2.71}$$

Convex conjugates $\tilde{\Phi}^*$ and $\tilde{\Psi}^*$ are derived from (2.67) and (2.49) respectively and are given by

$$\tilde{\Phi}^*(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p} + \boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{g}\|_2^2 - \frac{1}{2}\|\boldsymbol{g}\|_{\boldsymbol{V}_\zeta}^2 \tag{2.72}$$

and [SJP$^+$12, OMBF18]

$$\tilde{\Psi}^*(\boldsymbol{z}) = i_{B_\infty(\lambda)}(\boldsymbol{z}) \tag{2.73}$$

where $B_\infty(\lambda)$ is the ball of radius λ according to $\mathcal{L}_\infty$-norm

$$B_\infty(\lambda) = \{\boldsymbol{z} : \|\boldsymbol{z}\|_\infty \leqslant \lambda\} = \{\boldsymbol{z} : |z_j| \leqslant \lambda, \forall j\} \tag{2.74}$$

and $i_{B_\infty(\lambda)}$ is the indicator function of $B_\infty(\lambda)$ [KF17, Pal17]

$$i_{B_\infty(\lambda)}(\boldsymbol{z}) = \begin{cases} 0 \text{ if } \boldsymbol{z} \in B_\infty(\lambda) \\ +\infty \text{ otherwise} \end{cases}. \tag{2.75}$$

Given (2.72) and (2.73), the saddle-point problem reads

$$\min_{\boldsymbol{f}}\max_{\boldsymbol{p},\boldsymbol{z}} \langle \boldsymbol{f}, \boldsymbol{H}^T\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{p} + \nabla^T\boldsymbol{z}\rangle - \frac{1}{2}\|\boldsymbol{p} + \boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{g}\|_2^2 + \frac{1}{2}\|\boldsymbol{g}\|_{\boldsymbol{V}_\zeta}^2 - i_{B_\infty(\lambda)}(\boldsymbol{z}). \tag{2.76}$$

This problem has been solved by applying an instance of Chambolle-Pock (CP) algorithm [CP11, SJP$^+$12, Pal17]. CP algorithm considers the generic saddle-point problem

$$\min_{\boldsymbol{x}}\max_{\boldsymbol{y}} \langle \boldsymbol{x}, \boldsymbol{K}^T\boldsymbol{y}\rangle + G(\boldsymbol{x}) - F^*(\boldsymbol{y}) \tag{2.77}$$

where $\boldsymbol{K}$ is a linear continuous operator and G and F^* are lower-semicontinuous convex functions [CP11]. Generic CP algorithm to solve (2.77) is given in algorithm 5 [CP11]. In order to ensure convergence, primal and dual stepsizes μ_p and μ_d in CP algorithm have to be chosen such that [CP11]

$$\sqrt{\mu_p\mu_d}|||\boldsymbol{K}|||_2 < 1 \tag{2.78}$$

Algorithm 5 Generic Chambolle-Pock (CP) algorithm [CP11]

Initialize $\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}$
Initialize $\tilde{\boldsymbol{x}}^{(0)} = \boldsymbol{x}^{(0)}$
Choose primal and dual stepsizes $\mu_p > 0$ and $\mu_d > 0$ such that $\sqrt{\mu_p \mu_d}|||\boldsymbol{K}|||_2 < 1$
Choose over-relaxation parameter $\sigma \in [0, 1]$
for $t = 0, \dots, t_{max} - 1$ **do**
  Dual ascent : $\boldsymbol{y}^{(t+1)} = \text{prox}_{\mu_d F*}\left(\boldsymbol{y}^{(t)} + \mu_d \boldsymbol{K}\tilde{\boldsymbol{x}}^{(t)}\right)$
  Primal descent : $\boldsymbol{x}^{(t+1)} = \text{prox}_{\mu_p G}\left(\boldsymbol{x}^{(t)} - \mu_p \boldsymbol{K}^T \boldsymbol{y}^{(t+1)}\right)$
  Over-relaxation : $\tilde{\boldsymbol{x}}^{(t+1)} = \boldsymbol{x}^{(t+1)} + \sigma\left(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\right)$
end for

where $|||\boldsymbol{K}|||_2$ can be computed using the power method [SJP+12]. Instead of fixed μ_p, μ_d and σ, CP algorithm can use sequences of over-relaxation parameters $\sigma^{(t)}$ and primal and dual stepsizes $\mu_p^{(t)}$ and $\mu_d^{(t)}$ [CP11].

The over-relaxation step in algorithm 5 is optional and can be omitted fixing $\sigma = 0$ in order to save memory usage [ZC08]. With $\sigma = 0$, dual ascent and primal descent in algorithm 5 are equivalent to proximal-point steps [ZC08]

$$\begin{cases} \boldsymbol{y}^{(t+1)} = \arg\max_{\boldsymbol{y}} \langle \boldsymbol{x}^{(t)}, \boldsymbol{K}^T\boldsymbol{y}\rangle - F^*(\boldsymbol{y}) - \frac{1}{2\mu_d}\|\boldsymbol{y} - \boldsymbol{y}^{(t)}\|_2^2 \\ \boldsymbol{x}^{(t+1)} = \arg\min_{\boldsymbol{x}} \langle \boldsymbol{x}, \boldsymbol{K}^T\boldsymbol{y}^{(t)}\rangle + G(\boldsymbol{x}) + \frac{1}{2\mu_p}\|\boldsymbol{x} - \boldsymbol{x}^{(t)}\|_2^2 \end{cases}, \tag{2.79}$$

so new iterates are close to the previous ones. This formulation can be used in order to derive the instance of CP algorithm for reconstruction problem (2.76) :

$$\begin{cases} \boldsymbol{p}^{(t+1)} = \arg\max_{\boldsymbol{p}} \langle \boldsymbol{f}^{(t)}, \boldsymbol{H}^T\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{p}\rangle - \frac{1}{2}\|\boldsymbol{p} + \boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{g}\|_2^2 + \frac{1}{2}\|\boldsymbol{g}\|_{\boldsymbol{V}_\zeta}^2 - \frac{1}{2\mu_d}\|\boldsymbol{p} - \boldsymbol{p}^{(t)}\|_2^2 \\ \boldsymbol{z}^{(t+1)} = \arg\max_{\boldsymbol{z}} \langle \boldsymbol{f}^{(t)}, \nabla^T\boldsymbol{z}\rangle - i_{B_\infty(\lambda)}(\boldsymbol{z}) - \frac{1}{2\mu_d}\|\boldsymbol{z} - \boldsymbol{z}^{(t)}\|_2^2 \\ \boldsymbol{f}^{(t+1)} = \arg\min_{\boldsymbol{f}} \langle \boldsymbol{f}, \boldsymbol{H}^T\boldsymbol{V}_\zeta^{-\frac{1}{2}}\boldsymbol{p}^{(t+1)} + \nabla^T\boldsymbol{z}^{(t+1)}\rangle + \frac{1}{2\mu_p}\|\boldsymbol{f} - \boldsymbol{f}^{(t)}\|_2^2 \end{cases}. \tag{2.80}$$

In order to compute $\boldsymbol{z}^{(t+1)}$, the proximal operator of $i_{B_\infty(\lambda)}$ is the projection $P_{B_\infty(\lambda)}$ onto $B_\infty(\lambda)$, which reads [KF17]

$$P_{B_\infty(\lambda)}(\boldsymbol{z}) = \text{sign}\,(\boldsymbol{z}) \min\{\lambda, |\boldsymbol{z}|\}, \tag{2.81}$$

where $\text{sign}(.)$ and $\min\{\lambda, .\}$ are applied entrywise. Adding over-relaxation step for primal variable $\boldsymbol{f}$, updates (2.80) lead to algorithm 6 [SJP+12].

Reminding equation (2.39), the size of the gradient $\nabla\boldsymbol{f}$ is I times the size of the volume, where I is the number of directions in which first-order differences are computed. Hence, in algorithm 6, dual variable $\boldsymbol{z}$ is very memory-consuming. In order to reduce this memory cost, Primal Dual Frank-Wolfe (PDFW) [OMBF18] algorithm introduces the auxiliary variable

$$\tilde{\boldsymbol{z}} = \nabla^T\boldsymbol{z} = \sum_{\iota=1}^{I} \nabla_\iota^T \boldsymbol{z}_\iota, \tag{2.82}$$

and deals with $\tilde{\boldsymbol{z}}$ instead of $\boldsymbol{z}$. Since $\tilde{\boldsymbol{z}}$ is the size of the volume, PDFW results in memory-efficiency. The proximal-point step for $\boldsymbol{z}$ in algorithm 6 is replaced by a Frank-Wolfe (FW) step [FW56, OMBF18] :

$$\boldsymbol{z}_\iota^{(t+1)} = (1 - \mu_{fw}^{(t)})\boldsymbol{z}_\iota + \mu_{fw}^{(t)}\lambda\,\text{sign}\left(\nabla_\iota\tilde{\boldsymbol{f}}^{(t)}\right), \forall\iota \in \{1, \dots, I\} \tag{2.83}$$

which, thanks to the linearity in $\boldsymbol{z}_\iota$ of (2.83), enables to deal with $\tilde{\boldsymbol{z}}$ instead of $\boldsymbol{z}$, by accumulating over all $\iota \in \{1, \dots, I\}$ [OMBF18] :

$$\tilde{\boldsymbol{z}}^{(t+1)} = (1 - \mu_{fw}^{(t)})\tilde{\boldsymbol{z}}^{(t)} + \mu_{fw}^{(t)}\lambda\sum_{\iota=1}^{I}\nabla_\iota^T\,\text{sign}\left(\nabla_\iota\tilde{\boldsymbol{f}}^{(t)}\right). \tag{2.84}$$

PDFW is summarized in algorithm 7. Choices for sequences of stepsizes $\left(\mu_p^{(t)}, \mu_d^{(t)}, \mu_{fw}^{(t)}\right)$ are given in [OMBF18]. At the time we present this work, the convergence is only ensured fixing $\sigma = 0$. Due to its appealing memory-efficiency, PDFW will be used in this thesis to compare the proposed methods with TV-regularized weighted least-squares.

Algorithm 6 Chambolle-Pock (CP) algorithm to solve TV-penalized weighted least-squares [SJP+12]

Initialize $\boldsymbol{f}^{(0)}$
Initialize $\boldsymbol{p}^{(0)} = \boldsymbol{0}$, $\boldsymbol{z}^{(0)} = \boldsymbol{0}$
Initialize $\tilde{\boldsymbol{f}}^{(0)} = \boldsymbol{f}^{(0)}$
Choose primal and dual stepsizes $\mu_p > 0$ and $\mu_d > 0$ such that $\sqrt{\mu_p \mu_d} \left\| \begin{pmatrix} \boldsymbol{V}_\zeta^{-\frac{1}{2}} \boldsymbol{H} \\ \nabla \end{pmatrix} \right\|_2 < 1$
Choose over-relaxation parameter $\sigma \in [0, 1]$
for $t = 0, \ldots, t_{max} - 1$ **do**
  Dual ascent : $\boldsymbol{p}^{(t+1)} = \frac{1}{1+\mu_d} \boldsymbol{p}^{(t)} + \frac{\mu_d}{1+\mu_d} \boldsymbol{V}_\zeta^{-\frac{1}{2}} (\boldsymbol{H} \tilde{\boldsymbol{f}}^{(t)} - \boldsymbol{g})$
  Dual ascent : $\boldsymbol{z}^{(t+1)} = \operatorname{sign}\left(\boldsymbol{z}^{(t)} + \mu_d \nabla \tilde{\boldsymbol{f}}^{(t)}\right) \min\left\{\lambda, \left|\boldsymbol{z}^{(t)} + \mu_d \nabla \tilde{\boldsymbol{f}}^{(t)}\right|\right\}$
  Primal descent : $\boldsymbol{f}^{(t+1)} = \boldsymbol{f}^{(t)} - \mu_p \left(\boldsymbol{H}^T \boldsymbol{V}_\zeta^{-\frac{1}{2}} \boldsymbol{p}^{(t+1)} + \nabla^T \boldsymbol{z}^{(t+1)}\right)$
  Over-relaxation : $\tilde{\boldsymbol{f}}^{(t+1)} = \boldsymbol{f}^{(t+1)} + \sigma \left(\boldsymbol{f}^{(t+1)} - \boldsymbol{f}^{(t)}\right)$
end for

Algorithm 7 Primal-Dual Frank-Wolfe (PDFW) algorithm [OMBF18]

Initialize $\boldsymbol{f}^{(0)}$
Initialize $\boldsymbol{p}^{(0)} = \boldsymbol{0}$, $\tilde{\boldsymbol{z}}^{(0)} = \boldsymbol{0}$
Initialize $\tilde{\boldsymbol{f}}^{(0)} = \boldsymbol{f}^{(0)}$
Choose primal and dual stepsizes $\mu_p^{(t)}, \mu_d^{(t)}, \mu_{fw}^{(t)}$
Choose over-relaxation parameter $\sigma \in [0, 1]$
for $t = 0, \ldots, t_{max} - 1$ **do**
  Dual ascent : $\boldsymbol{p}^{(t+1)} = \frac{1}{1+\mu_d^{(t)}} \boldsymbol{p}^{(t)} + \frac{\mu_d^{(t)}}{1+\mu_d^{(t)}} \boldsymbol{V}_\zeta^{-\frac{1}{2}} \left(\boldsymbol{H} \tilde{\boldsymbol{f}}^{(t)} - \boldsymbol{g}\right)$
  Dual ascent by Frank-Wolfe : $\tilde{\boldsymbol{z}}^{(t+1)} = \left(1 - \mu_{fw}^{(t)}\right) \tilde{\boldsymbol{z}}^{(t)} + \mu_{fw}^{(t)} \lambda \sum_{\iota=1}^{I} \nabla_\iota^T \operatorname{sign}\left(\nabla_\iota \tilde{\boldsymbol{f}}^{(t)}\right)$
  Primal descent : $\boldsymbol{f}^{(t+1)} = \boldsymbol{f}^{(t)} - \mu_p^{(t)} \left(\boldsymbol{H}^T \boldsymbol{V}_\zeta^{-\frac{1}{2}} \boldsymbol{p}^{(t+1)} + \tilde{\boldsymbol{z}}^{(t+1)}\right)$
  Over-relaxation : $\tilde{\boldsymbol{f}}^{(t+1)} = \boldsymbol{f}^{(t+1)} + \sigma \left(\boldsymbol{f}^{(t+1)} - \boldsymbol{f}^{(t)}\right)$
end for

2.7 Perspectives to develop MBIR methods for SAFRAN

Due to the presence in its aeronautical parts of very diffusing and absorbing materials such as metal, the projections acquired at SAFRAN are very noisy, leading to unsatisfactory reconstructions by maximum-likelihood techniques, which try to match the data too much. In addition, the designed industrial parts often present strong asymmetries, making the noise not uniformly distributed over all the projection angles. These strong asymmetries advocate for weighting the data differently according to the projection direction, which is a task difficultly handled by filtered backprojection methods.

Since SAFRAN designs and manufactures its parts, SAFRAN has the possibility to enhance reconstruction quality for NDT by taking benefit from the knowledge of its parts, in terms of structure as in terms of shape. For this purpose, developing MBIR methods for reconstructing parts made by SAFRAN appears as a natural way, since these methods, as seen in this chapter, have the ability to insert prior information on the inspected volume, as weights for the projections.

MBIR methods have hyperparameters $\boldsymbol{\theta}$ which can be difficult to fix in practice. This is for instance the case of tradeoff parameter λ in sections 2.5 and 2.6. This parameter often needs to be tuned by repeated experiments. This can be very long for 3D NDT. In addition, the tuning procedure must be re-done each time the acquisition protocol is changed. As a consequence, the problem of finding optimal hyperparameters may arise repeatedly and hinder

the industrialization of proposed MBIR methods. For this reason, in this thesis, we aim at proposing MBIR methods which, starting from a prior model $\mathcal{M}$, estimate both optimal volume $\boldsymbol{f}$ and hyperparameters $\boldsymbol{\theta}$, by joint maximization a posteriori (JMAP) :

$$(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\theta}}) = \arg\max_{\boldsymbol{f},\boldsymbol{\theta}} p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{M}), \tag{2.85}$$

where joint posterior of the unknowns $\boldsymbol{\psi} = (\boldsymbol{f}, \boldsymbol{\theta})$ is given by Bayes' rule :

$$p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{M}) = \frac{p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{\theta}; \mathcal{M}) p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{M})}{p(\boldsymbol{g}|\mathcal{M})}. \tag{2.86}$$

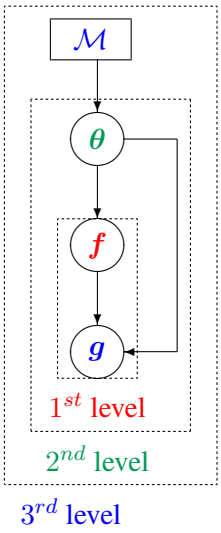


Figure 2.6: General hierarchical model to classify reconstruction methods

Figure 2.6 summarizes the general hierarchical model derived from equation (2.86) : data $\boldsymbol{g}$ result from volume $\boldsymbol{f}$ through projection operation $\boldsymbol{H}$, while volume $\boldsymbol{f}$ is given by prior model $\mathcal{M}$ and its hyperparameters $\boldsymbol{\theta}$. Depending on which stages of the hierarchical model are considered, reconstruction methods can be divided into three classes :

- first-level methods are maximum-likelihood techniques presented in sections 2.3 and 2.4, which aim to match theoretical and actual projections : $\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{\theta}; \mathcal{M})$,
- second-level methods add prior information but fix hyperparameters : $\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} p(\boldsymbol{f}|\boldsymbol{g}; \boldsymbol{\theta}, \mathcal{M})$,
- third-level methods optimize both the volume and the hyperparameters, leading to joint maximization a posteriori : $(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\theta}}) = \arg\max_{\boldsymbol{f},\boldsymbol{\theta}} p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{M})$.

Reconstruction methods at each level have their pros and cons. Compared to first-level methods, second-level and third-level reconstruction methods are robust to uncertainties in the projections and enhance reconstruction quality. Third-level reconstruction methods optimize the hyperparameters jointly with the volume. Nevertheless, optimizing $\boldsymbol{\theta}$ prevents from using improper priors, i.e., priors for which the integral over $\mathbb{R}^N$ is not finite. As a result, several improper strong sparsity-inducing priors, such as anisotropic and isotropic TV (2.41) and (2.40), or Potts model (2.42), cannot be considered in third-level reconstruction methods : for these priors, the hyperparameters have to be tuned empirically, which may be difficult. At last, first-level reconstruction methods such as filtered backprojection do not introduce bias in the estimation and are fast, while MBIR methods are known to be computationally burdensome due to repeated projection and backprojection operations. In order to alleviate this computational cost, MBIR methods have to be massively parallelizable on GPUs.

Chapter 3

Projection and backprojection operators

As we have seen in chapter 2, projection and backprojection operators $\boldsymbol{H}$ and $\boldsymbol{H}^T$ are called repeatedly in all MBIR methods. Since they are very computationally-intensitive, this makes MBIR methods very slow compared to FBP methods. In order to alleviate their computational cost, it has become common during the last decade to parallelize existing projection and backprojection algorithms [Sid85, Jos82, DMB04, LFB10, NL15] on GPUs (Graphical Processor Units) [SK10]. One GPU can run thousands of threads simultaneously, so GPUs are very interesting devices for high parallel computing (HPC). The threads of the GPU execute the same code, called the *kernel* [SK10]. Hence, parallelizing $\boldsymbol{H}$ (respectively $\boldsymbol{H}^T$) on the GPU consists in transposing the projection (respectively the backprojection) algorithm into kernels. As a result, the projections of thousands of rays (respectively the backprojections in thousands of voxels) can be computed at the same time, leading to a significant acceleration of MBIR methods, compared to a CPU implementation [LFDMY17].

When parallelizing a pair of projector and backprojector (P/BP) on the GPU, it is necessary to make sure that the final result of the projection of one ray is written by only one thread. Similarly, the backprojection in one voxel has to be written by only one thread. If this is not the case, this generates writing conflicts between the threads, so the accumulation of threads' contributions gives a completely random result [PSL14]. In order to avoid this, the accumulation can be done using atomic operations [SK10, Chapter 9], so we are sure that only one thread at a time brings its contribution to the accumulation. Nevertheless, this slows down the projection or backprojection computation.

Efficient models for projection and backprojection operations are a key aspect to ensure speed and accuracy of MBIR methods. Unfortunately, an efficient projector $\boldsymbol{H}$ does not necessarily implies that the adjoint backprojector $\boldsymbol{H}^T$ is also efficient. For instance, Siddon's projector, presented in appendix B.1, is fast [Sid85, JSDS$^+$98, HLY99], but its adjoint backprojector not, even parallelized on the GPU [PSL14, NL15]. Similarly, while the ray-driven (RD) projector, presented in section 3.1.1, is very suited to high parallelization, this is not the case for its adjoint backprojector. The same problem occurs for the voxel-driven (VD) backprojector detailed in section 3.1.2, of which the adjoint projector is difficult to accelerate on the GPU [DYXW17].

Due to the difficulty to use adjoint projector and backprojector which are both fast, accurate and easily parallelizable on GPUs, it has become very common in the CT community to work with P/BP pairs which are unmatched, i.e. pairs in which the used backprojector is not the adjoint of the used projector [ZG00]. For instance, in section 3.1, we present the unmatched ray-driven/voxel-driven (RD/VD) pair, in which the projector is ray-driven (RD) and the backprojector is voxel-driven (VD). Due to the computational efficiency of both the projector and the backprojector, this pair implemented on the GPU enables to perform very fast projection and backprojection operations.

Using an unmatched pair can be valid for very simple reconstruction algorithms, such as gradient descent to solve unregularized least-squares [ZG00]. Nevertheless, it remains a mathematical approximation, since the convergence proofs of reconstruction algorithms are derived considering a matched pair of projector and backprojector. As a consequence, an unmatched P/BP pair may lead to suboptimal reconstruction and even hinder the convergence of complex algorithms such as ADMM or PWLS [ASM16]. In order to ensure the convergence of MBIR methods, computationally-efficient matched P/BP pairs have been proposed [DMB04, LFB10]. In section 3.2, we focus on the matched Separable Footprint (SF) pair [LFB10] and present a new GPU implementation. The RD/VD and the SF pairs are validated and compared in section 3.3, as single modules and in a full iterative reconstruction method

which is PDFW presented in section 2.6. Perspectives for the new GPU implementation of the SF pair are presented in section 3.4.

3.1 Unmatched ray-driven/voxel-driven (RD/VD) pair

3.1.1 Ray-driven projector

For each projection angle ϕ and each cell (u_e, v_e) of the detector, the ray-driven (RD) projector traces a ray connecting the source and the center of the cell, as illustrated in 2D in figure 3.1. In the field-of-view, this ray is regularly sampled with step δ, which is the side length of the voxels. At each sample point (x_k, y_k, z_k), the value $f(x_k, y_k, z_k)$ is calculated by trilinear interpolation. According to the discretization of the integral in Beer-Lambert law

$$g(u_e, v_e, \phi) = \int_{\mathcal{L}(u_e, v_e, \phi)} f(\boldsymbol{r}) dl, \tag{3.1}$$

the result of this interpolation is added to the projection, multiplied by the sampling stepsize δ :

$$\begin{aligned} g(u_e, v_e, \phi) + = \quad & \delta \times \big[(1-\epsilon_x)(1-\epsilon_y)(1-\epsilon_z) f(x_e, y_e, z_e) \\ & + \epsilon_x (1-\epsilon_y)(1-\epsilon_z) f(x_e+1, y_e, z_e) \\ & + (1-\epsilon_x)\epsilon_y (1-\epsilon_z) f(x_e, y_e+1, z_e) \\ & + \epsilon_x \epsilon_y (1-\epsilon_z) f(x_e+1, y_e+1, z_e) \\ & + (1-\epsilon_x)(1-\epsilon_y)\epsilon_z f(x_e, y_e, z_e+1) \\ & + \epsilon_x (1-\epsilon_y)\epsilon_z f(x_e+1, y_e, z_e+1) \\ & + (1-\epsilon_x)\epsilon_y \epsilon_z f(x_e, y_e+1, z_e+1) \\ & + \epsilon_x \epsilon_y \epsilon_z f(x_e+1, y_e+1, z_e+1) \big]. \end{aligned} \tag{3.2}$$

In the trilinear interpolation, $(\epsilon_x, \epsilon_y, \epsilon_z) \in [0,1]^3$ are the normalized distances in x, y and z-directions with the nearest neighbour (x_e, y_e, z_e) of the sample point, as shown in figure 3.1. The RD projector is naturally highly-parallelizable on the GPU, since one ray can simply be handled by one thread. In order to accelerate the computations, the volume is copied on the texture memory of the GPU [SK10, Chapter 7]. If the volume was simply stored in the global memory of the GPU, the values of spatially neighbouring voxels would be far from each other : while $f(x_e, y_e, z_e)$ and $f(x_e+1, y_e, z_e)$ would be placed at neighbouring locations, $f(x_e, y_e, z_e)$ and $f(x_e, y_e+1, z_e)$ would be distant by N_x, and $f(x_e, y_e, z_e)$ and $f(x_e, y_e, z_e+1)$ by $N_x \times N_y$. Since N_x and N_y are very large, performing interpolation (3.2) would result in a high memory traffic in order to read the values of each voxel. On the contrary, the texture memory stores the volume in a cache such that the accesses to spatially neighbouring voxels are made faster [NVI18, Section 3.2.11.1]. Since the memory accesses to perform interpolation (3.2) are only spatially local, using the texture memory results in a reduced memory traffic and the interpolation is done faster. The texture memory has another advantage, which is that trilinear interpolation (3.2) can be performed by the hardware of the GPU [NVI18, Section 3.2.11.1] : this results in a further acceleration of the calculations, and makes the RD projector very fast.

On the contrary, the adjoint RD backprojector is difficult to parallelize since one thread has to handle one voxel in order to avoid writing conflicts. In this case, one thread computing the backprojection in one voxel has to find the rays for which the voxel has contributed to the projections. Then, each of these rays has to be projected to compute the exact contribution of the voxel. Hence, in the RD backprojector, many threads have to compute the projections of the same rays, which is very redundant and slow. Therefore, the RD backprojector is very unefficient on the GPU. In addition, due to the interpolation step, artifacts are visible in the RD backprojection [DMB02, DMB04]. For these reasons, an unmatched backprojector can be preferred in order to accelerate the calculations [ZG00]. In this thesis, we choose the voxel-driven backprojector described in section 3.1.2.

3.1.2 Voxel-driven backprojector

For each voxel (x_e, y_e, z_e), the voxel-driven (VD) backprojector traces a ray connecting the source and the center of the voxel. The projection of this ray is located at a position $(u(\phi; x_e, y_e, z_e), v(\phi; x_e, y_e, z_e))$ on the detector, as shown in figure 3.2 in 2D. The value of the projection at this point is calculated by a bilinear interpolation. The projections

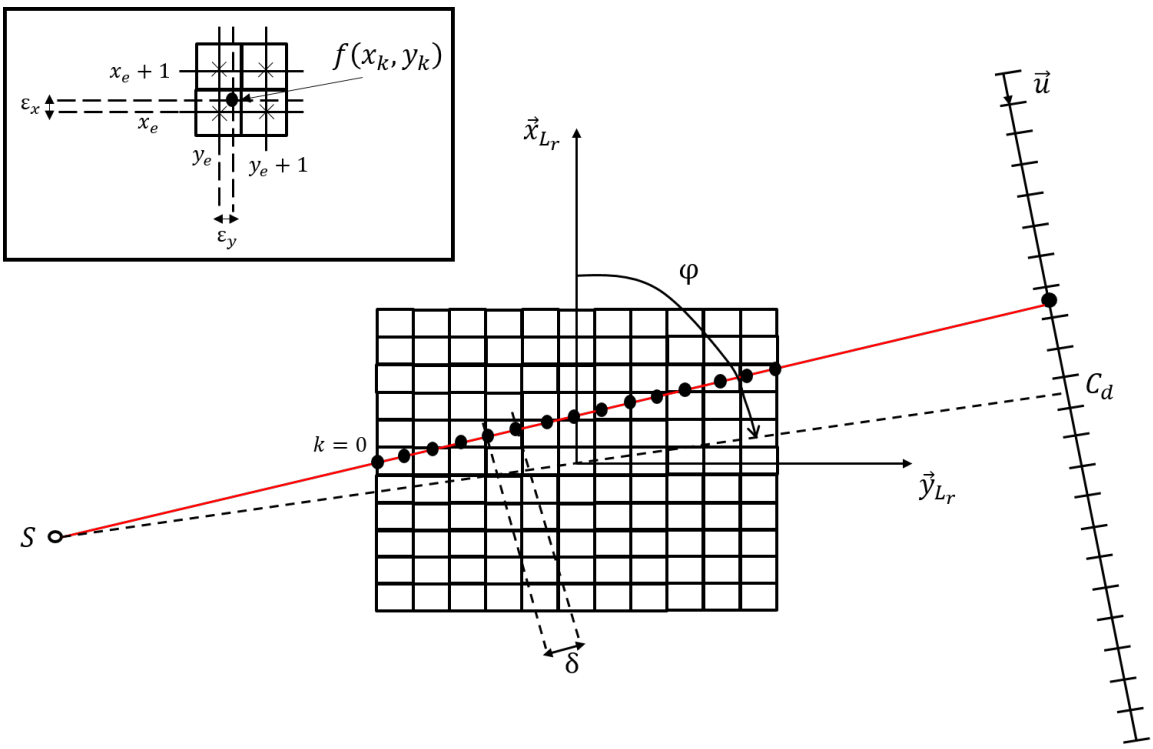


Figure 3.1: Ray-driven projector

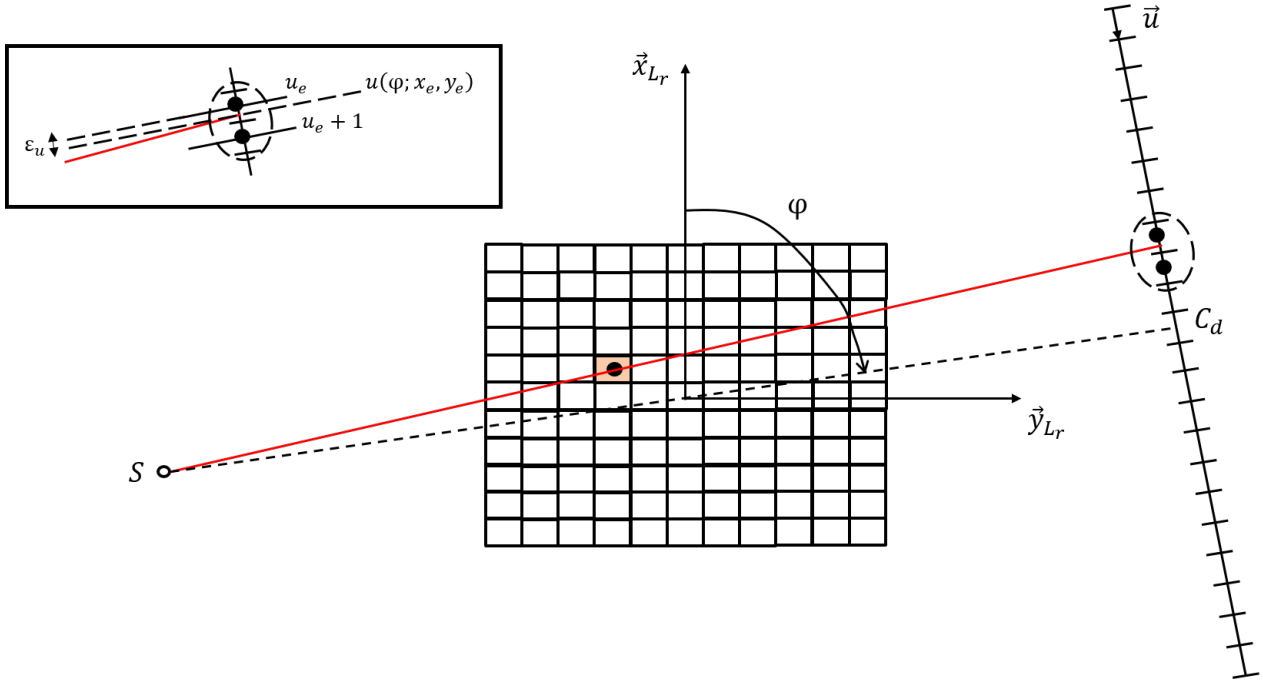


Figure 3.2: Voxel-driven backprojector

are copied on the texture memory of the GPU for local memory accesses. Similarly to the RD projector, the VD backprojector takes advantage of the texture memory since the bilinear interpolation is done by the hardware. The result of the interpolation is accumulated in the backprojection. This operation is repeated for each projection angle ϕ. The voxel-driven backprojection reads

$$b(x_e, y_e, z_e) = \sum_{\phi} g_{interp}(u(\phi; x_e, y_e, z_e), v(\phi; x_e, y_e, z_e), \phi) \tag{3.3}$$

where

$$\begin{aligned} g_{interp}(u(\phi; x_e, y_e, z_e), v(\phi; x_e, y_e, z_e), \phi) \quad &= (1-\epsilon_u)(1-\epsilon_v) g(u_e, v_e, \phi) \\ &+\epsilon_u (1-\epsilon_v) g(u_e+1, v_e, \phi) \\ &+(1-\epsilon_u)\epsilon_v g(u_e, v_e+1, \phi) \\ &+\epsilon_u \epsilon_v g(u_e+1, v_e+1, \phi). \end{aligned} \tag{3.4}$$

In the bilinear interpolation, $(\epsilon_u, \epsilon_v) \in [0,1]^2$ are the normalized distances in u and v-directions with the nearest neighbour (u_e, v_e) of the projection point $(u(\phi; x_e, y_e, z_e), v(\phi; x_e, y_e, z_e))$, as shown in figure 3.2. Like the RD projector, the VD backprojector is easily highly parallelizable and very fast, while this is not the case for its adjoint VD projector [DYXW17]. Furthermore, the adjoint VD projector suffers from the same interpolation artifacts as the RD backprojector [DMB04]. Due to these considerations in terms of image quality and computational speed, in this work, we use the VD backprojector with the unmatched RD projector described in section 3.1.1.

3.2 Matched Separable Footprint (SF) pair

The Separable Footprint (SF) pair [LFB10] approximates the footprint of a voxel onto the detector as a separable function with respect to the axial and the transaxial directions. This approximation leads to factorizations of the computations in the projection and the backprojection formulae, and then to reasonable computation times for both the projector and the backprojector, even on the CPU. Several separable footprint approximations have been proposed [LFB10, Lon11].

In the Separable Footprint Trapezoidal-Rectangular (SFTR) pair, the footprint of a voxel is modeled as trapezoidal in the transaxial direction, and rectangular in the axial direction [LFB10]. The relevance of this choice is well illustrated in figures 3.3 and 3.4. It has been shown that the matched distance-driven (DD) pair [DMB02, DMB04] is a kind of Separable Footprint Rectangular-Rectangular (SFRR) pair, in which the approximating function in the transaxial direction is rectangular as in the axial direction [Lon11, chap. 3.2.5]. Due to the rectangular approximation in the transaxial direction, the DD pair is less accurate than the SFTR pair, in particular for projection angles near $45°$. Another SF pair, the Separable Footprint Trapezoidal-Trapezoidal (SFTT) pair, has also been studied in [LFB10]. This pair is theoretically more precise than the SFTR pair since it models the footprint as trapezoidal both in the transaxial and axial directions. Nevertheless, it has been shown that this further approximation implies much more computations than the SFTR pair, without providing significant gain in the reconstruction precision [LFB10]. Table 3.1 shows a comparison of the different SF pairs.

SF P/BP pair	Approximation in the transaxial direction	Approximation in the axial direction	Accuracy	Computation speed
SFTR	Trapezoidal	Rectangular	++	+
SFTT	Trapezoidal	Trapezoidal	+++	–
SFRR	Rectangular	Rectangular	+	++
DD	Rectangular	Rectangular	+	++

Table 3.1: Comparison of SF models in terms of computation speed and accuracy

Despite their advantages, the matched DD and SF pairs require more computations than unmatched P/BP pairs such as the RD/VD pair. In addition, they are more complicated to parallelize on the GPU [WF11, XML+17,

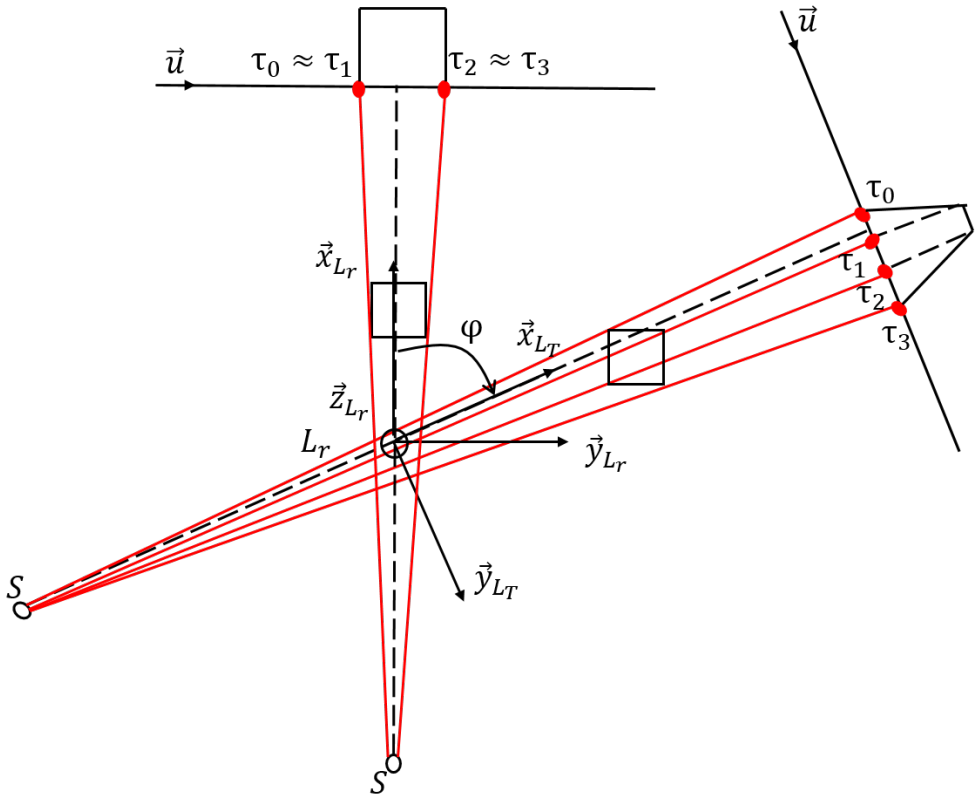


Figure 3.3: Trapezoidal shape of the transaxial footprint of a voxel at different projection angles

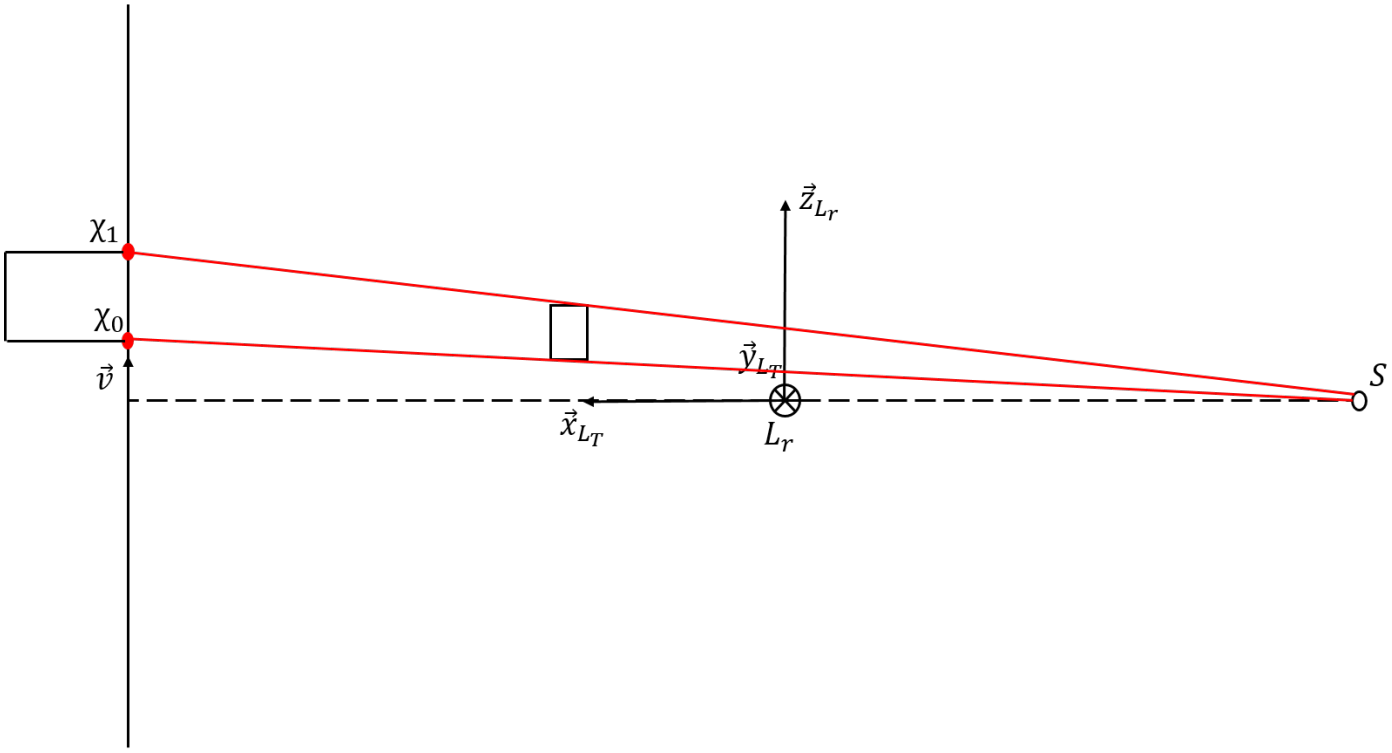


Figure 3.4: Rectangular shape of the axial footprint of a voxel

LFDMY17]. As a consequence, their use is still not widespread in the CT community. Nevertheless, they offer perspectives to increase the accuracy of MBIR methods, as to ensure their convergence. In table 3.1, the SFTR pair appears as the best tradeoff between accuracy and computation speed. For this reason, in this work, motivated by using a matched P/BP pair at SAFRAN (for MBIR methods, but perhaps for other applications such as registration), we focus on the SFTR pair. For the sake of completeness and in order to detail the formulae adapted to our geometrical model, the SFTR pair originally proposed in [LFB10] and its CPU implementation are fully presented in appendix B.2. In the following, since we do not discuss the other SF models anymore, we abbreviate the name of the SFTR pair and simply call it the SF pair.

For this pair, two GPU implementations have been proposed [WF11, XML$^+$17], the one of [XML$^+$17] having been shown faster than the one of [WF11]. Both these implementations have a CPU-loop on the projection angles, which computes many intermediate volumes and implies many memory transfers between CPU and GPU. These memory transfers are known to be the main bottleneck for GPU computing. Moreover, when working on a multi-GPU implementation of the reconstruction algorithm, the latency due to the high amount of transfers between GPUs can be hidden by overlapping copies with computations [MWF18]. With the existing GPU implementations of the SF pair [WF11, XML$^+$17], the CPU-loop hinders to perform these overlaps efficiently, since the copies are only achievable at the last iteration of the loop [MWF18]. For all these reasons, we proposed in [CGMDP18] a new GPU implementation of the SF pair where no outer loop is present. This implementation is reviewed in sections 3.2.1 and 3.2.2. The proposed GPU SF backprojector runs only one kernel which is voxel-driven, i.e., one thread updates one voxel, while the proposed GPU SF projector is ray-driven. In [CGMDP18], the GPU SF projector runs two independent kernels, each kernel handling rays depending on whether the source is closer to x-axis or y-axis. Dealing with several kernels is not an easy task for GPU computation optimizations. In section 3.2.3, inspired by the work of [DH17] for Joseph's projector [Jos82], we merge the two kernels of the GPU SF projector into only one kernel.

3.2.1 SF projector on the GPU

The SF projection of volume $\boldsymbol{f}$ on cell (u_e, v_e) at projection angle ϕ reads

$$g(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) \sum_{x_e, y_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e) \tag{3.5}$$

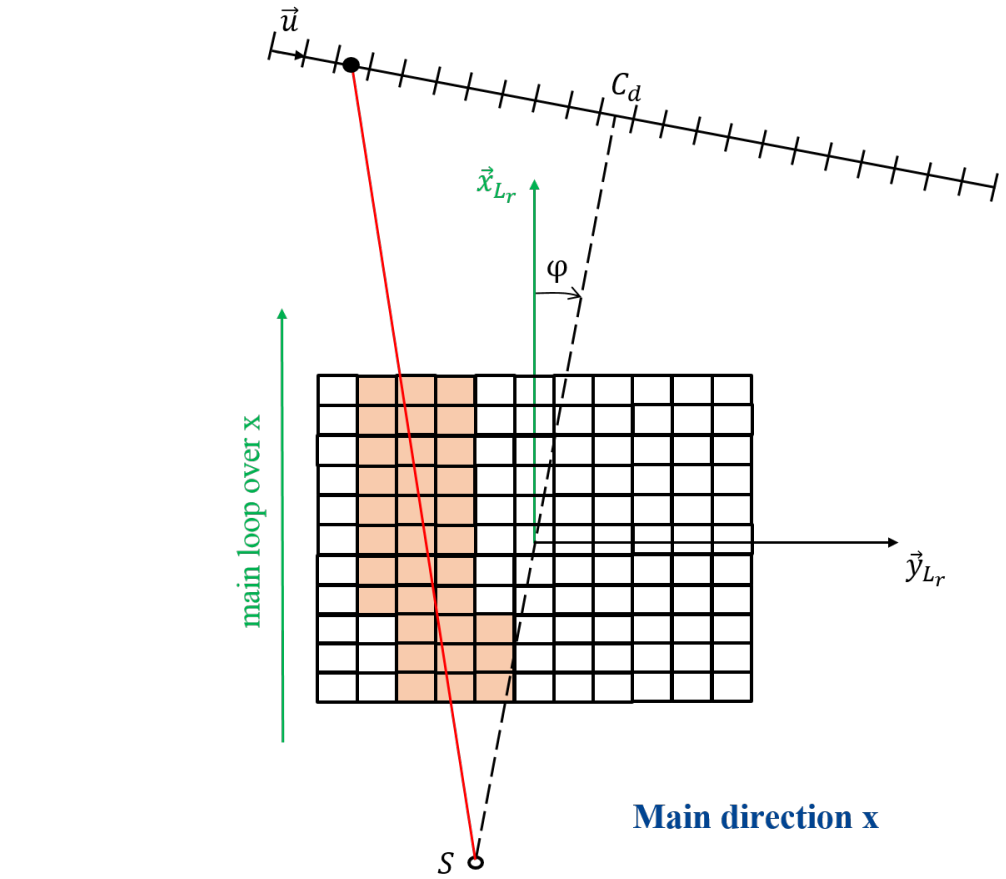
where $F_{trans}(u_e, \phi; x_e, y_e)$ is the transaxial footprint and $F_{ax}(v_e, \phi; x_e, y_e, z_e)$ the axial footprint of voxel (x_e, y_e, z_e) on cell (u_e, v_e) with projection angle ϕ [LFB10]. Amplitude functions $l_{\theta_c}(u_e, v_e)$ and $l_{\psi_v}(\phi; x_e, y_e)$ are given by the A2 method described in appendix B.2 [LFB10]. The formulae for the transaxial and the axial footprints, adapted to our geometrical model, are also given in appendix B.2.

In order to avoid writing conflicts between threads, our GPU implementation of the SF projector is ray-driven, i.e. one thread computes the SF projection of one ray defined by (u_e, v_e, ϕ). We follow each ray according to its primary direction, which is x-axis if the source is closer to x-axis, and y-axis otherwise [XML$^+$17]. Each thread advances along this primary direction, as illustrated in figure 3.5, and computes the voxels for which the transaxial and the axial footprints are both non-zero. As for the RD projector, the volume is copied on texture memory to ensure local memory accesses. Furthermore, variables related to the geometry are copied in constant memory.

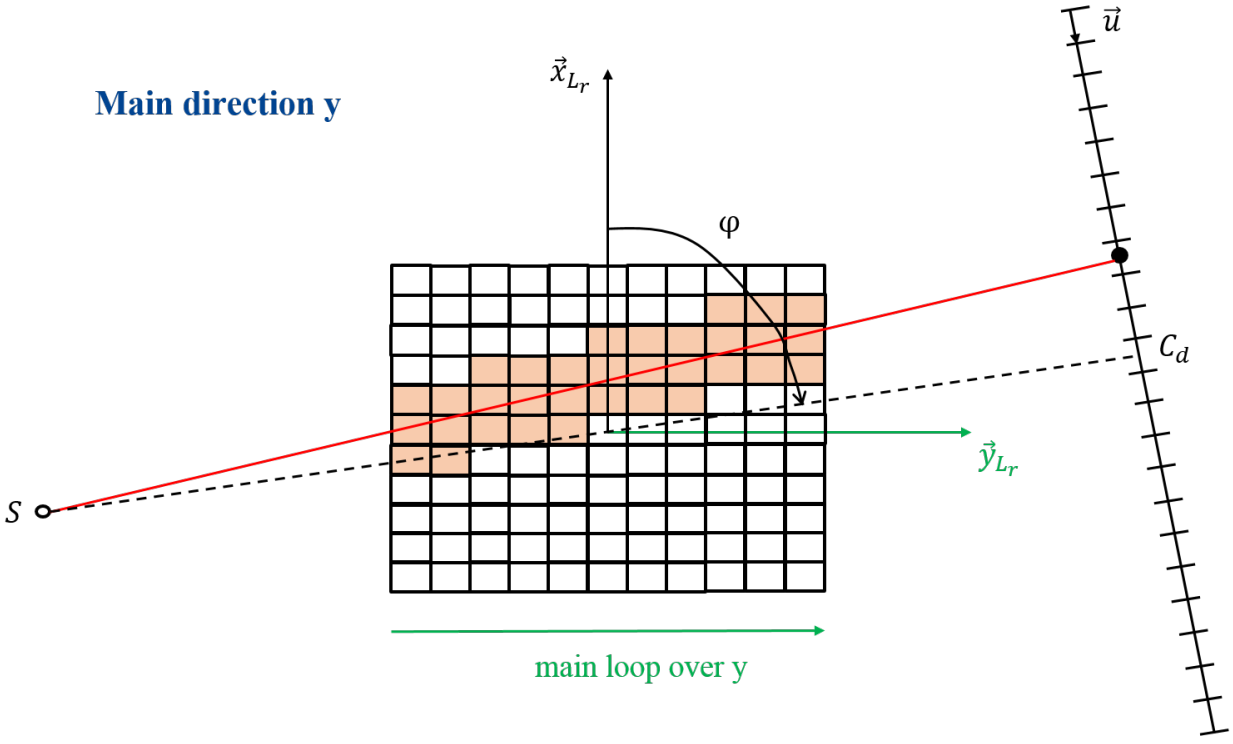
Considering rays with primary direction x or y leads to different calculations we need to do separately. For this reason, in [CGMDP18], our GPU implementation of the SF projector runs two kernels. First kernel $proj_x_ker$ handles rays with primary direction x, while second kernel $proj_y_ker$ handles rays with primary direction y. Since they compute disjoint sets of projections, these two kernels are independent. They can be run successively or on two different GPUs. Nevertheless, having several kernels does not facilitate the developments in order to accelerate the computations by using multi-streaming or multi-GPU optimizations [SK10, Chapters 10 and 11]. In section 3.2.3, we go back to this problem and show how to merge these kernels.

For a ray with primary direction x, kernel $proj_x_ker$ performs its main loop over x_e, $0 \leqslant x_e < N_x$. For each x_e, the intersecting location $(x_e, y_e(x_e))$ with the ray is computed similarly to Joseph's method :

$$y_e(x_e) = \frac{1}{\delta} \left(y_S(\phi) + \frac{y(u_e, \phi) - y_S(\phi)}{x(u_e, \phi) - x_S(\phi)} (x_e \delta - x_S(\phi)) \right). \tag{3.6}$$



(a)



(b)

Figure 3.5: Voxels considered by the GPU SF projector depending on whether the main direction is x (a) or y (b)

For current x_e, the thread looks the neighbouring pixels of $(x_e, y_e(x_e))$ in the middle plane, i.e. y_e such that

$$y_{e_{min}} \leqslant y_e \leqslant y_{e_{max}}, \quad \begin{cases} y_{e_{min}} = \lfloor y_e(x_e) \rceil - 1 \\ y_{e_{max}} = \lfloor y_e(x_e) \rceil + 1 \end{cases}, \tag{3.7}$$

where $\lfloor y_e(x_e) \rceil$ is the nearest integer to $y_e(x_e)$. The bounds for y_e, $y_{e_{min}}$ and $y_{e_{max}}$, are different from those proposed in [CGMDP18], which were over-estimated [Jos82, XML+17]. Hence, considering each y_e between $y_{e_{min}}$ and $y_{e_{max}}$ leads to a small loop of size 3 in kernel $proj_x_ker$. For rays with primary direction y, the calculations are the same, except that the roles of x and y are reversed : the main loop is over y_e, $0 \leqslant x_e < N_y$, and we have a sub-loop over x_e, $x_{e_{min}} \leqslant x_e \leqslant x_{e_{max}}$. Here, we see the interest of dealing with rays with different primary directions in different kernels, in order to avoid divergence between threads.

For each considered (x_e, y_e) (with y_e or x_e varying in a very little set depending on the executed kernel), the scaled transaxial footprint

$$F'_{trans}(u_e, \phi; x_e, y_e) = l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) \tag{3.8}$$

is computed. Next, each thread finds the indices z_e for which $F_{ax}(v_e, \phi; x_e, y_e, z_e) \neq 0$. Thanks to the rectangular shape of the axial footprint, these indices are very simple to compute :

$$z_{e_{min}} \leqslant z_e \leqslant z_{e_{max}} \tag{3.9}$$

where

$$\begin{cases} z_{e_{min}} = \lfloor z_{Lr_e} - 0.5 + \frac{x_{\phi_e}\delta_v}{D}(v_e - v_{C_{d_e}} - 0.5) \rfloor \\ z_{e_{max}} = \lceil z_{Lr_e} + 0.5 + \frac{x_{\phi_e}\delta_v}{D}(v_e - v_{C_{d_e}} + 0.5) \rceil \end{cases}, \tag{3.10}$$

and

$$x_{\phi_e} = \frac{R}{\delta} + (x_e - x_{Lr_e}) \cos\phi + (y_e - y_{Lr_e}) \sin\phi. \tag{3.11}$$

Knowing the bounds for z_e, each thread runs a loop over z_e to compute

$$F'_{ax}(v_e, \phi; x_e, y_e) = \sum_{z_e = z_{e_{min}}}^{z_{e_{max}}} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e). \tag{3.12}$$

This loop is very small and is typically size 3. Iteratively, through the double loop over x_e and y_e, the threads calculate the sum

$$g'(u_e, v_e, \phi) = \sum_{x_e} \sum_{y_e = y_{e_{min}}}^{y_{e_{max}}} F'_{trans}(u_e, \phi; x_e, y_e) F'_{ax}(v_e, \phi; x_e, y_e) \tag{3.13}$$

in kernel $proj_x_ker$, and

$$g'(u_e, v_e, \phi) = \sum_{y_e} \sum_{x_e = x_{e_{min}}}^{x_{e_{max}}} F'_{trans}(u_e, \phi; x_e, y_e) F'_{ax}(v_e, \phi; x_e, y_e) \tag{3.14}$$

in kernel $proj_y_ker$. Finally, the thread handling ray (u_e, v_e, ϕ) computes the final value for the projection

$$g(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) g'(u_e, v_e, \phi), \tag{3.15}$$

which is stored from GPU to CPU. The pseudo code for kernels $proj_x_ker$ and $proj_y_ker$ is given in algorithms 8 and 9.

Algorithm 8 Kernel $proj_x_ker$ for SF projection on GPU

Handle ray (u_e, v_e, ϕ)
Initialize the projection : $p_0 := 0$
Compute : $x_{SP} = x(u_e, \phi) - x_S(\phi)$ and $y_{SP} = y(u_e, \phi) - y_S(\phi)$
if $|x_{SP}| \geqslant |y_{SP}|$ **then**
  for $x_e = 0, \ldots, (N_x - 1)$ **do**
    Compute $y_e(x_e)$ by (3.6)
    Compute $y_{e_{min}}$ and $y_{e_{max}}$ by (3.7)
    for $y_e = y_{e_{min}}, \ldots, y_{e_{max}}$ **do**
      Compute and sort the projections of $(x_e \pm 0.5, y_e \pm 0.5) : \tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$
      Compute transaxial footprint F_{trans}
      Compute $z_{e_{min}}$ and $z_{e_{max}}$ by (3.10)
      Initialize the sum of the axial footprints : $F'_{ax} := 0$
      for $z_e = z_{e_{min}}, \ldots, z_{e_{max}}$ **do**
        Compute axial footprint F_{ax}
        if $(x_e - x_{Lr_e})^2 + (y_e - y_{Lr_e})^2 \leqslant \left(\frac{N_x}{2}\right)^2$ **then**
          Update : $F'_{ax} += F_{ax} \times f(x_e, y_e, z_e)$
        end if
      end for
      Compute amplitude l_{ψ_v} (A2 method)
      Update the projection : $p_0 = l_{\psi_v} \times F_{trans} \times F'_{ax}$
    end for
  end for
end if
Compute amplitude l_{θ_c} (A2 method)
Store the projection : $g(u_e, v_e, \phi) := l_{\theta_c} \times p_0$

3.2.2 SF backprojector on the GPU

Since the SF projector and backprojector are matched, the SF backprojection for a voxel (x_e, y_e, z_e) is

$$b(x_e, y_e, z_e) = \sum_{\phi} \sum_{u_e} F_{trans}(u_e, \phi; x_e, y_e) l_{\psi_v}(\phi; x_e, y_e) \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi). \quad (3.16)$$

To prevent writing conflicts between threads, we compute $\boldsymbol{b}$ by running a kernel $back_ker$ which is voxel-driven : one thread calculates the backprojection in one voxel (x_e, y_e, z_e). Kernel $back_ker$ has its main loop over projection angles ϕ. For each projection angle, each thread finds cells (u_e, v_e) overlapped by the transaxial and axial footprints of voxel (x_e, y_e, z_e). The u_e-coordinates of these cells are given by ordering the projections of the four corners of pixel (x_e, y_e) in the middle plane [LFB10]. These projections $\tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$ are computed as explained in appendix B.2. Hence, the u_e-coordinates for which $F_{trans}(u_e, \phi; x_e, y_e) \neq 0$ are

$$u_{e_{min}} \leqslant u_e \leqslant u_{e_{max}}, \begin{cases} u_{e_{min}} = \lfloor u_{C_{d_e}} - 0.5 + \frac{\tau_0}{\delta_u} \rfloor \\ u_{e_{max}} = \lceil u_{C_{d_e}} + 0.5 + \frac{\tau_3}{\delta_u} \rceil \end{cases} . \quad (3.17)$$

The v_e-coordinates for which $F_{ax}(v_e, \phi; x_e, y_e, z_e) \neq 0$ are

$$v_{e_{min}} \leqslant v_e \leqslant v_{e_{max}}, \begin{cases} v_{e_{min}} = \lfloor v_{C_{d_e}} - 0.5 + \frac{\chi_0}{\delta_v} \rfloor \\ v_{e_{max}} = \lceil v_{C_{d_e}} + 0.5 + \frac{\chi_1}{\delta_v} \rceil \end{cases} \quad (3.18)$$

where χ_0 and χ_1 are the projections of $(x_e, y_e, z_e - 0.5)$ and $(x_e, y_e, z_e + 0.5)$ respectively [LFB10], and are calculated as in appendix B.2. Knowing the bounds for u_e and v_e, the threads run a small double-loop over u_e and v_e. The size of this double loop is approximately the same for each voxel [WF11]. Projections $\boldsymbol{g}$ are copied on texture memory. Because of the separation of the footprint of the voxel with respect to the transaxial and axial directions, the double

Algorithm 9 Kernel $proj_y_ker$ for SF projection on GPU

Handle ray (u_e, v_e, ϕ)
Initialize the projection : $p_0 := 0$
Compute : $x_{SP} = x(u_e, \phi) - x_S(\phi)$ and $y_{SP} = y(u_e, \phi) - y_S(\phi)$
if $|x_{SP}| < |y_{SP}|$ **then**
  for $y_e = 0, \ldots, (N_y - 1)$ **do**
    Compute $x_e(y_e)$ similarly to (3.6)
    Compute $x_{e_{min}}$ and $x_{e_{max}}$ similarly to (3.7)
    for $x_e = x_{e_{min}}, \ldots, x_{e_{max}}$ **do**
      Compute and sort the projections of $(x_e \pm 0.5, y_e \pm 0.5) : \tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$
      Compute transaxial footprint F_{trans}
      Compute $z_{e_{min}}$ and $z_{e_{max}}$ by (3.10)
      Initialize the sum of the axial footprints : $F'_{ax} := 0$
      for $z_e = z_{e_{min}}, \ldots, z_{e_{max}}$ **do**
        Compute axial footprint F_{ax}
        if $(x_e - x_{Lr_e})^2 + (y_e - y_{Lr_e})^2 \leqslant \left(\frac{N_x}{2}\right)^2$ **then**
          Update : $F'_{ax} += F_{ax} \times f(x_e, y_e, z_e)$
        end if
      end for
      Compute amplitude l_{ψ_v} (A2 method)
      Update the projection : $p_0 = l_{\psi_v} \times F_{trans} \times F'_{ax}$
    end for
  end for
end if
Compute amplitude l_{θ_c} (A2 method)
Store the projection : $g(u_e, v_e, \phi) := l_{\theta_c} \times p_0$

loop can be done over u_e then v_e or over v_e then u_e indifferently. Here, the main loop is over u_e : for each u_e between $u_{e_{min}}$ and $u_{e_{max}}$, a loop is run over v_e, $v_{e_{min}} \leqslant v_e \leqslant v_{e_{max}}$. Like for the loop over z_e in SF projector, this loop is typically size 3 and calculates

$$b_{ax}(u_e, \phi; x_e, y_e, z_e) = \sum_{v_e=v_{e_{min}}}^{v_{e_{max}}} F_{ax}(v_e, \phi; x_e, y_e, z_e) l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi). \tag{3.19}$$

Hence, the double loop performs the summation

$$b_\phi(\phi; x_e, y_e, z_e) = \sum_{u_e=u_{e_{min}}}^{u_{e_{max}}} F_{trans}(u_e, \phi; x_e, y_e) l_{\psi_v}(\phi; x_e, y_e) b_{ax}(u_e, \phi; x_e, y_e, z_e). \tag{3.20}$$

Then, the backprojection can be updated

$$b(x_e, y_e, z_e) += b_\phi(\phi; x_e, y_e, z_e) \tag{3.21}$$

until all the projection angles have been considered. The pseudo code for kernel $back_ker$ is given in algorithm 10.

3.2.3 Merging the two kernels of the GPU SF projector

In section 3.2.1, the GPU SF projector has two independent kernels which are here run successively. In this section, inspired by the generalized Joseph's projector [DH17], we merge kernels $proj_x_ker$ and $proj_y_ker$ into only one kernel, so GPU computing optimizations are easier to implement. Like $proj_x_ker$ and $proj_y_ker$, the generalized kernel for the SF projector, called $proj_ker$, is ray-driven. Since it does not distinguish rays with primary directions x or y, kernel $proj_ker$ is branchless.

Algorithm 10 Kernel $back_ker$ for SF backprojection on GPU

Handle voxel (x_e, y_e, z_e)
Initialize the backprojection : $b_0 := 0$
if $(x_e - x_{Lr_e})^2 + (y_e - y_{Lr_e})^2 \leqslant \left(\frac{N_x}{2}\right)^2$ **then**
  for ϕ, projection angle, **do**
    Compute and sort the projections of $(x_e \pm 0.5, y_e \pm 0.5) : \tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$
    Compute amplitude l_{ψ_v} (A2 method)
    Compute $u_{e_{min}}$ and $u_{e_{max}}$ by (3.17)
    Compute $v_{e_{min}}$ and $v_{e_{max}}$ by (3.18)
    for $u_e = u_{e_{min}}, \ldots, u_{e_{max}}$ **do**
      Initialize the sum of the axial footprints : $b_{ax} := 0$
      for $v_e = v_{e_{min}}, \ldots, v_{e_{max}}$ **do**
        Compute amplitude l_{θ_c} (A2 method)
        Compute axial footprint F_{ax} and update : $b_{ax} += F_{ax} \times l_{\theta_c} \times g(u_e, v_e, \phi)$
      end for
      Multiply by the amplitude : $b_{ax} := b_{ax} \times l_{\psi_v}$ (A2 method)
      Compute transaxial footprint F_{trans}
      Update : $b_0 += F_{trans} \times b_{ax}$
    end for
  end for
end if
Store the backprojection : $b(x_e, y_e, z_e) := b_0$

Like [DH17], the generalized SF projector computes the primary direction

$$d = \arg \max_{\iota \in \{`x`, `y`\}} \{|\iota(u_e, \phi) - \iota_S(\phi)|\}. \tag{3.22}$$

We define the arithmetic sequences

$$\begin{cases} x_e(\iota_e) = \tilde{\theta}_x + \tilde{r}_x \iota_e \\ y_e(\iota_e) = \tilde{\theta}_y + \tilde{r}_y \iota_e \end{cases} \tag{3.23}$$

defined for all ι_e, $0 \leqslant \iota_e < N_\iota$, where

$$N_\iota = \begin{cases} N_x \text{ if } d = `x` \\ N_y \text{ if } d = `y` \end{cases}. \tag{3.24}$$

For all ι_e, $0 \leqslant \iota_e < N_\iota$, we have

$$\begin{cases} x_e(\iota_e + 1) = x_e(\iota_e) + \tilde{r}_x \\ y_e(\iota_e + 1) = y_e(\iota_e) + \tilde{r}_y \end{cases}. \tag{3.25}$$

In (3.23), the common differences and the offsets for sequences $x_e(\iota_e)$ and $y_e(\iota_e)$ are defined by

$$\begin{cases} \tilde{r}_x = \frac{x(u_e,\phi) - x_S(\phi)}{d(u_e,\phi) - d_S(\phi)} \\ \\ \tilde{r}_y = \frac{y(u_e,\phi) - y_S(\phi)}{d(u_e,\phi) - d_S(\phi)} \end{cases} \tag{3.26}$$

and

$$\begin{cases} \tilde{\theta}_x = \frac{1}{\delta}\left(x_S(\phi) - \tilde{r}_x d_S(\phi)\right) \\ \\ \tilde{\theta}_y = \frac{1}{\delta}\left(y_S(\phi) - \tilde{r}_y d_S(\phi)\right) \end{cases}. \tag{3.27}$$

The key to understand how these definitions enable to merge the two kernels is to see what happens when the primary

direction is x or y. If x is the primary direction, $d = `x`$, then

$$\begin{cases} \tilde{r}_x = \frac{x(u_e,\phi)-x_S(\phi)}{x(u_e,\phi)-x_S(\phi)} = 1 \\ \\ \tilde{r}_y = \frac{y(u_e,\phi)-y_S(\phi)}{x(u_e,\phi)-x_S(\phi)} \\ \tilde{\theta}_x = \frac{1}{\delta}\left(x_S(\phi) - \tilde{r}_x x_S(\phi)\right) = \frac{1}{\delta}\left(x_S(\phi) - x_S(\phi)\right) = 0 \\ \\ \tilde{\theta}_y = \frac{1}{\delta}\left(y_S(\phi) - \tilde{r}_y x_S(\phi)\right) = \frac{1}{\delta}\left(y_S(\phi) - \frac{y(u_e,\phi)-y_S(\phi)}{x(u_e,\phi)-x_S(\phi)} x_S(\phi)\right) \end{cases} . \tag{3.28}$$

Consequently,

$$x_e(\iota_e) = \tilde{\theta}_x + \tilde{r}_x \iota_e = \iota_e \tag{3.29}$$

and

$$\begin{aligned} y_e(\iota_e) &= \tilde{\theta}_y + \tilde{r}_y \iota_e = \frac{1}{\delta}\left(y_S(\phi) - \frac{y(u_e,\phi)-y_S(\phi)}{x(u_e,\phi)-x_S(\phi)} x_S(\phi)\right) + \frac{y(u_e,\phi)-y_S(\phi)}{x(u_e,\phi)-x_S(\phi)} \iota_e \\ &= \frac{1}{\delta}\left(y_S(\phi) + \frac{y(u_e,\phi)-y_S(\phi)}{x(u_e,\phi)-x_S(\phi)} (\xi_e \delta - x_S(\phi))\right) \\ &= y_e(x_e) \end{aligned} \tag{3.30}$$

according to (3.6) and because $x_e(\iota_e) = \iota_e$ if x is the primary direction. Similarly, $y_e(\iota_e) = \iota_e$ and $x_e(\iota_e) = x_e(y_e)$ if y is the primary direction. Hence, for all ι_e, $0 \leqslant \iota_e < N_\iota$, pixel $(x_e(\iota_e), y_e(\iota_e))$ corresponds to the intersecting location with the ray, whatever the primary direction is.

In order to fully merge the two kernels of the SF projector, we now have to find a unified expression for the visited pixels (x_e, y_e) in the middle plane, independent from primary direction d. If the primary direction is x, according to the definitions of $y_{e_{min}}$ and $y_{e_{max}}$ in (3.7), the set of the considered pixels is

$$\begin{aligned} \mathcal{X}(x_e) = \{&(x_e, \lfloor y_e(x_e) \rceil - 1), \\ &(x_e, \lfloor y_e(x_e) \rceil), \\ &(x_e, \lfloor y_e(x_e) \rceil + 1)\} \end{aligned} \tag{3.31}$$

and, if y is the primary direction, this set is

$$\begin{aligned} \mathcal{Y}(y_e) = \{&(\lfloor x_e(y_e) \rceil - 1, y_e), \\ &(\lfloor x_e(y_e) \rceil, y_e), \\ &(\lfloor x_e(y_e) \rceil + 1, y_e)\} . \end{aligned} \tag{3.32}$$

According to (3.26) and to definition (3.22) of primary direction d, one shall notice that

$$\begin{cases} \tilde{r}_x = 1 \\ |\tilde{r}_y| < 1 \end{cases} \quad \text{if } x \text{ is the primary direction,} \tag{3.33}$$

and reversely if y is the primary direction. Now, we consider the set

$$\begin{aligned} \Xi(\iota_e) = \{&(\lfloor x_e(\iota_e) \rceil - \lceil 1 - |\tilde{r}_x| \rceil, \lfloor y_e(\iota_e) \rceil - \lceil 1 - |\tilde{r}_y| \rceil), \\ &(\lfloor x_e(\iota_e) \rceil, \lfloor y_e(\iota_e) \rceil), \\ &(\lfloor x_e(\iota_e) \rceil + \lceil 1 - |\tilde{r}_x| \rceil, \lfloor y_e(\iota_e) \rceil + \lceil 1 - |\tilde{r}_y| \rceil)\} . \end{aligned} \tag{3.34}$$

According to (3.29), (3.30) and (3.33), we see that

$$\begin{cases} \Xi(\iota_e) = \mathcal{X}(x_e) & \text{if } x \text{ is the primary direction} \\ \Xi(\iota_e) = \mathcal{Y}(y_e) & \text{if } y \text{ is the primary direction} \end{cases} . \tag{3.35}$$

Hence, $\Xi(\iota_e)$ is an unified expression of $\mathcal{X}(x_e)$ and $\mathcal{Y}(y_e)$. In the generalized SF projector, the projection is computed by

$$g(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) \sum_{\iota_e} \sum_{(x'_e, y'_e) \in \Xi(\iota_e)} F'_{trans}(u_e, \phi; x'_e, y'_e) F'_{ax}(v_e, \phi; x'_e, y'_e), \tag{3.36}$$

independently from the primary direction. The main loop of unified kernel $proj_ker$ is over ι_e, $0 \leqslant \iota_e < N_\iota$. Indices $x_e(\iota_e)$ and $y_e(\iota_e)$ are initialized to $\tilde{\theta}_x$ and $\tilde{\theta}_y$ respectively, and incremented by $\tilde{r}_x$ and $\tilde{r}_y$ at the end of each iteration of the loop. For each ι_e, according to (3.36), the transaxial and axial footprints are computed for each (x'_e, y'_e) in $\Xi(\iota_e)$ in an unrolled loop. The pseudo code for unified kernel $proj_ker$ is given in algorithm 11. Since the scaled transaxial footprint (3.8) is the same for cells in a same column of the detector, it is only computed by threads for which $v_e = 0$. After the calculation, the scaled transaxial footprint is stored in shared memory, so the other threads can use it. This optimization can also be made for kernels $proj_x_ker$ and $proj_y_ker$ in algorithms 8 and 9.

Algorithm 11 Unified kernel $proj_ker$ for SF projection on GPU

Handle ray (u_e, v_e, ϕ)
Create table : $F'_{trans}(u_e)$ in shared memory
Compute : $x_{SP} = x(u_e, \phi) - x_S(\phi)$ and $y_{SP} = y(u_e, \phi) - y_S(\phi)$
if $|x_{SP}| \geqslant |y_{SP}|$ **then**
  Set : $\tilde{r}_x = 1$, $\tilde{r}_y = \frac{y_{SP}}{x_{SP}}$, $\tilde{\theta}_x = 0$, $\tilde{\theta}_y = \frac{1}{\delta}\left(y_S(\phi) - \tilde{r}_y x_S(\phi)\right)$ and $N_\iota = N_x$
else
  Set : $\tilde{r}_x = \frac{x_{SP}}{y_{SP}}$, $\tilde{r}_y = 1$, $\tilde{\theta}_x = \frac{1}{\delta}\left(x_S(\phi) - \tilde{r}_x y_S(\phi)\right)$, $\tilde{\theta}_y = 0$ and $N_\iota = N_y$
end if
Initialize : $p_0 := 0$, $x_e = \tilde{\theta}_x$ and $y_e = \tilde{\theta}_y$
for $\iota = 0, \ldots, (N_\iota - 1)$ **do**
  for each pixel of the middle plane $(x'_e, y'_e) \in \Xi(\iota)$ (defined by (3.34)), **do**
    Compute $z_{e_{min}}$ and $z_{e_{max}}$ by (3.10)
    Initialize the sum of the axial footprints : $F'_{ax} := 0$
    for $z'_e = z_{e_{min}}, \ldots, z_{e_{max}}$ **do**
      Compute axial footprint F_{ax}
      if $(x'_e - x_{Lr_e})^2 + (y'_e - y_{Lr_e})^2 \leqslant \left(\frac{N_x}{2}\right)^2$ **then**
        Update : $F'_{ax} += F_{ax} \times f(x'_e, y'_e, z'_e)$
      end if
    end for
    if $v_e = 0$ **then**
      Compute and sort the projections of $(x'_e \pm 0.5, y'_e \pm 0.5) : \tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$
      Compute transaxial footprint F_{trans}
      Compute amplitude l_{ψ_v} (A2 method)
      Store in shared memory : $F'_{trans}(u_e) = l_{\psi_v} \times F_{trans}$
    end if
    Update the projection : $p_0 = F'_{trans}(u_e) \times F'_{ax}$
  end for
  Update : $x_e += \tilde{r}_x$ and $y_e += \tilde{r}_y$
end for
Compute amplitude l_{θ_c} (A2 method)
Store the projection : $g(u_e, v_e, \phi) := l_{\theta_c} \times p_0$

Before dealing with the validation, we shall warn the reader that when $\tilde{r}_x$ and $\tilde{r}_y$ are both equal to 1 (i.e. when the azimuthal angle of the ray is $45^o[90^o]$), all $(x'_e, y'_e) \in \Xi(\iota_e)$ correspond to the same pixel, which is $(\lfloor x_e(\iota_e) \rceil, \lfloor y_e(\iota_e) \rceil)$. In order to avoid to compute the same footprint three times, we add the condition that the sum of the axial footprints $F'_{ax}(v_e, \phi; x_e, y_e)$ is calculated for pixels

$$(\lfloor x_e(\iota_e) \rceil - \lceil 1 - |\tilde{r}_x| \rceil, \lfloor y_e(\iota_e) \rceil - \lceil 1 - |\tilde{r}_y| \rceil) \text{ and } (\lfloor x_e(\iota_e) \rceil + \lceil 1 - |\tilde{r}_x| \rceil, \lfloor y_e(\iota_e) \rceil + \lceil 1 - |\tilde{r}_y| \rceil)$$

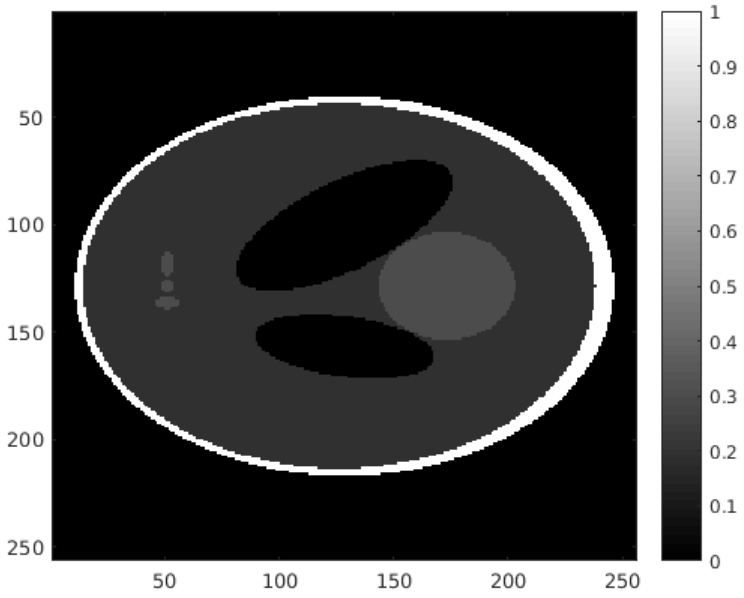


Figure 3.6: Modified Shepp-Logan phantom (middle slice)

only if $|\tilde{r}_x| < 1$ or $|\tilde{r}_y| < 1$. Otherwise, the sum of the axial footprints is set to zero for these two pixels. This is a precaution in order to be sure to cover all the possible cases. The if-statement taking this condition into acount is added just before accumulating in F'_{ax} in line 16 of algorithm 11, so the possible divergence caused by this condition is small. Furthermore, this divergence does not often happen in practice, since, due to numerical approximations, having $\tilde{r}_x$ and $\tilde{r}_y$ both equal to 1 is very unlikely. For this reason, and for the sake of simplicity, the condition does not appear explicitly in the pseudo code of the kernel in algorithm 11.

3.3 Validation

We now validate the RD/VD and the SF pairs on the GPU. For this purpose, we use modified Shepp-Logan phantom [SL74], shown in figure 3.6. The field-of-view is sampled into 256^3 voxels and is size $6.05 \times 6.05 \times 5.86$ mm^3. The source-to-detector distance is $D = 230$ mm, and the source-to-object distance is $R = 98$ mm. We use 64 projections of the volume uniformly distributed over $[0, 2\pi]$. The projections are sampled into 256^2 pixels. For our tests, we use one GPU which is Tesla P100-PCIE-12GB.

3.3.1 Projector and backprojector as single modules

First, we show the results of our proposed GPU implementation of the SF pair. We compare the results when the SF projector has two kernels or only one unified kernel. Like [XML+17], we show in table 3.2 the normalized root mean square error (NRMSE) with respect to our CPU version, presented in appendix B.2 [LFB10] :

$$NRMSE = \sqrt{\frac{1}{N_y}\sum_{i=1}^{N_y}\left(\frac{y_i^{(GPU)} - y_i^{(CPU)}}{y_i^{(CPU)}}\right)^2}, \tag{3.37}$$

where y_i denotes the projection or the backprojection for ray or voxel i respectively. As we see in table 3.2, the NRMSE is very low for both our GPU SF projector and backprojector, so our implementation implies no deviation with respect to our CPU version. This is visible in figures 3.7, 3.8 and 3.9, where the obtained images look the same.

In particular, the NRMSE is the lowest when using the SF projector with one kernel. This validates our merging of the two kernels of the GPU SF projector. In addition, we see in table 3.2 that the use of shared memory in kernel $proj_ker$ in algorithm 11 accelerates the GPU SF projector with one kernel.

Operator	Computation time	NRMSE with respect to CPU
SF projector with two kernels	561 ms	0.023 %
SF projector with one kernel (without using shared memory)	563 ms	**0.018 %**
SF projector with one kernel (using shared memory)	**373 ms**	**0.018 %**
SF backprojector (after SF projection with two kernels)	**406 ms**	0.006 %
SF backprojector (after SF projection with one kernel)	**406 ms**	**0.005 %**

Table 3.2: Matched GPU SF projector and backprojector as single modules

Operator	Computation time	NRMSE with respect to CPU SF pair
RD projector	**41 ms**	0.043 %
VD backprojector (after RD projection)	**21 ms**	0.083 %

Table 3.3: Unmatched GPU RD projector and VD backprojector as single modules

The images obtained by the unmatched RD/VD pair in figure 3.10 are similar to the ones obtained by the SF pair. In table 3.3, the NRMSE with respect to the CPU version of SF pair is very low. As expected, the computation times for the RD projector and the VD backprojector are much smaller than the ones for the SF pair.

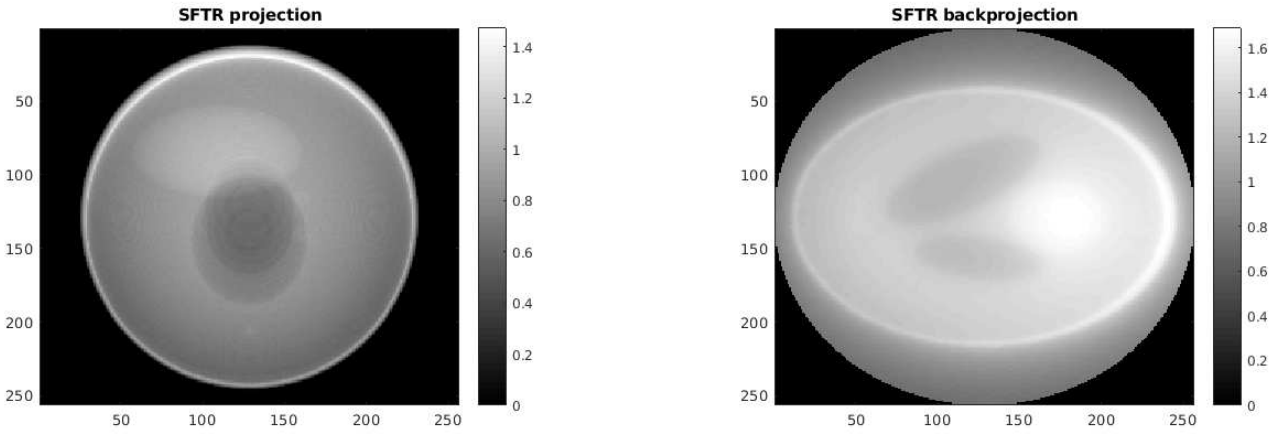
P/BP pair	Coupling degree
Unmatched RD/VD pair	1.0040
Matched SF pair on CPU	**1.0000**
Matched SF pair on GPU, with the projector having two kernels	1.0001
Matched SF pair on GPU, with the projector having one kernel	1.0001

Table 3.4: Coupling degree for the GPU RD/VD pair, the CPU SF pair and the GPU SF pairs

In order to confirm that our GPU implementation of the SF pair is matched, we compute the coupling degree, defined by [ASM16]

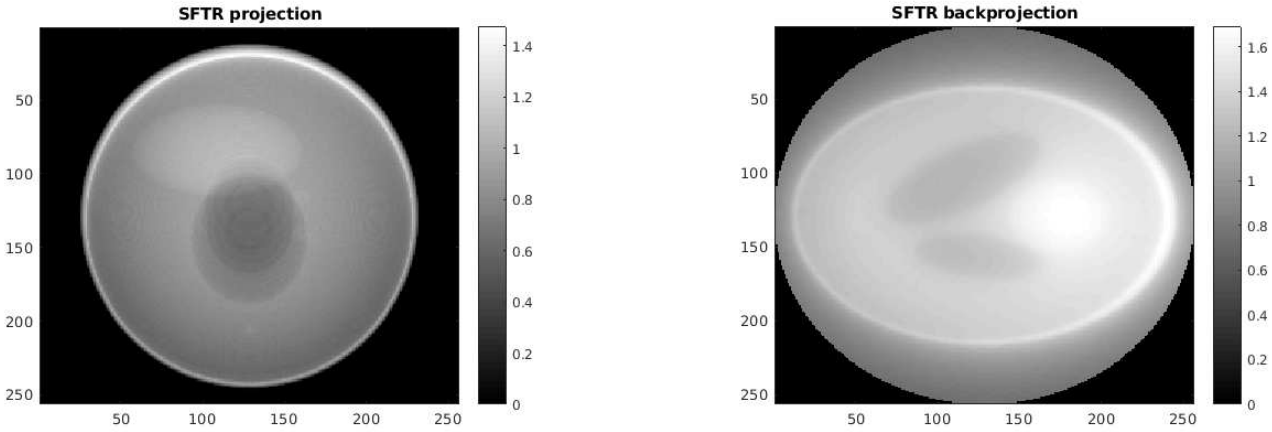
$$c = \frac{\langle \boldsymbol{g}, \boldsymbol{H}\boldsymbol{f} \rangle}{\langle \boldsymbol{B}\boldsymbol{g}, \boldsymbol{f} \rangle} \tag{3.38}$$

for random projections $\boldsymbol{g}$ and random volume $\boldsymbol{f}$. In (3.38), $\boldsymbol{H}$ denotes the used projector and $\boldsymbol{B}$ the used backprojector. For an unmatched pair, $\boldsymbol{B} \neq \boldsymbol{H}^T$, so $c \neq 1$. On the contrary, for a matched pair, $\boldsymbol{B} = \boldsymbol{H}^T$, so $c = 1$. In practice, c is never perfectly equal to 1 because of the approximation errors due to the finite precision of the calculations done by the computer. Nevertheless, measuring the difference between the coupling degree and 1 gives a good idea of how the used P/BP pair is matched : the smaller this difference is, the better the projector and the backprojector are coupled [ASM16]. In table 3.4, we show the coupling degree for the different pairs presented in this chapter. For our CPU implementation of the SF pair described in appendix B.2, the operators are very well coupled since, using floating-point precision, the coupling degree is 1.0000. Using MATLAB's long format for display, we saw that this coupling degree is equal to 1 until the 7^{th} digit. For both of our GPU implementations of the SF pair, the coupling degree is 1.0001. It is less close to 1 than the CPU version of the pair because, in our GPU implementation, some factors, which theoretically correspond to the same value, are independently computed by several threads, leading to slight differences from one thread to another due to numerical errors. For instance, on CPU, the transaxial footprint $F_{trans}(u_e, \phi; x_e, y_e)$ is stored, so the same value is used to compute the backprojection in voxels (x_e, y_e, z_e) and $(x_e, y_e, z_e + 1)$, while, on GPU, the threads associated to voxels (x_e, y_e, z_e) and $(x_e, y_e, z_e + 1)$ both compute $F_{trans}(u_e, \phi; x_e, y_e)$ independently, and, consequently, are cause of a slight loss in the coupling of the pair compared to its CPU version. Nevertheless, our GPU SF projector and backprojector are more coupled than the RD projector and VD backprojector, since, for the unmatched RD/VD pair in table 3.4, the coupling degree has one less non-zero digit than the one of our GPU SF pair.



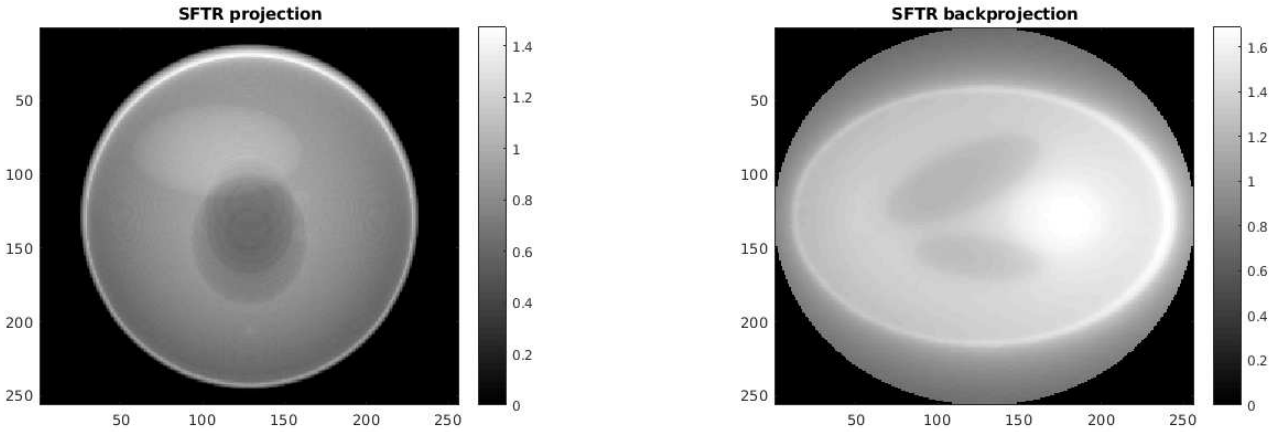
(a) (b)

Figure 3.7: CPU SF projection (a) followed by matched CPU SF backprojection (b)



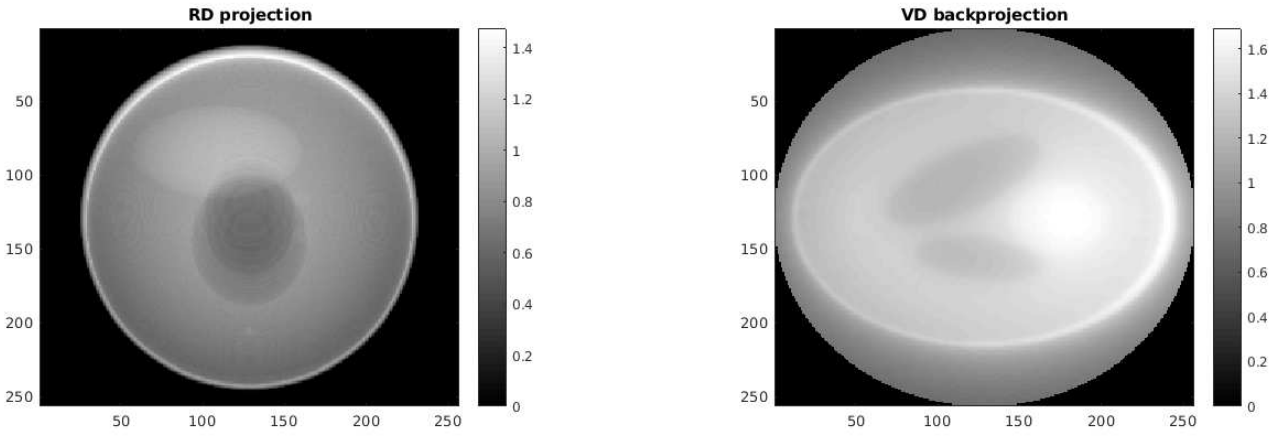
(a) (b)

Figure 3.8: GPU SF projection with two kernels (a) followed by matched GPU SF backprojection (b)



(a) (b)

Figure 3.9: GPU SF projection with one kernel (a) followed by matched GPU SF backprojection (b)



(a) (b)

Figure 3.10: GPU RD projection (a) followed by unmatched GPU VD backprojection (b)

3.3.2 Projector and backprojector in a full MBIR method

We now validate the presented P/BP pairs in a full MBIR method. Concerning the SF pair, in the previous section, our version with one kernel for the projector has given the best NRMSE with respect to the CPU version. In addition, dealing with only one kernel is easier than dealing with two in order to develop applications on the GPU. Hence, in the following, we only consider our GPU implementation of the SF pair with one kernel for the projector. We compare its use with the one of the unmatched RD/VD pair.

In this section, the MBIR method in which we incorporate our GPU implementations of the RD/VD pair and the SF pair is PDFW [OMBF18], presented in section 2.6. We reconstruct Shepp-Logan phantom, of size 256^3 voxels, from 64 projections with 256^2 pixels. The projections are noisy by adding a white Gaussian noise of Signal-to-Noise Ratio (SNR) equal to 20 db, as shown in figure 3.11. We run 500 iterations of PDFW. Concerning the statistical weights, we set : $\boldsymbol{V}_\zeta = \boldsymbol{I}_M$. The selected value for the regularization parameter is $\lambda = 0.5$. In order to comply with the conditions of the convergence theorem presented in [OMBF18], we do not introduce the over-relaxation step in PDFW. For the same reason, the set of primal and dual stepsizes we choose are the same as the ones denoted by "S1" in [OMBF18].

P/BP pair	Computation time	$\Delta_2 \boldsymbol{f}$	$\Delta_2 \boldsymbol{g}$
Unmatched RD/VD	**168.5 s**	3.93 %	0.76 %
Matched SF	510.5 s	**3.90 %**	**0.73 %**

Table 3.5: Comparison of the results of PDFW with the RD/VD pair and the SF pair

The results of PDFW with the RD/VD pair and the SF pair are shown in figures 3.13 and 3.14 respectively. Compared to the FDK method [FDK84] in figure 3.12, for both pairs, the reconstruction quality is greatly enhanced thanks to the regularization. As shown in figures 3.15 and 3.16 by looking at the errors with respect to the original volume, no difference depending on the used P/BP pair is visible in the reconstructions. Additionally, for the SF pair, in figures 3.17 and 3.18, we also show the obtained dual variables $\boldsymbol{p}$ and $\tilde{\boldsymbol{z}}$ which appear in PDFW in algorithm 7. Dual variable $\boldsymbol{p}$ is the size of the projections, while dual variable $\tilde{\boldsymbol{z}}$ is the size of the volume. Interestingly, if we compare with figures 3.11 and 3.12, and if we do not care about the gray levels, $\boldsymbol{p}$ looks like as it has captured the noise in the projections, while $\tilde{\boldsymbol{z}}$ seems to have captured the noise in the backprojection. With the unmatched RD/VD pair, the dual variables we have found are similar to those obtained with the SF pair.

In table 3.5, we evaluate the $\mathcal{L}_2$-relative error

$$\Delta_2 \boldsymbol{f} = \frac{\|\hat{\boldsymbol{f}} - \boldsymbol{f}_0\|_2^2}{\|\boldsymbol{f}_0\|_2^2} \tag{3.39}$$

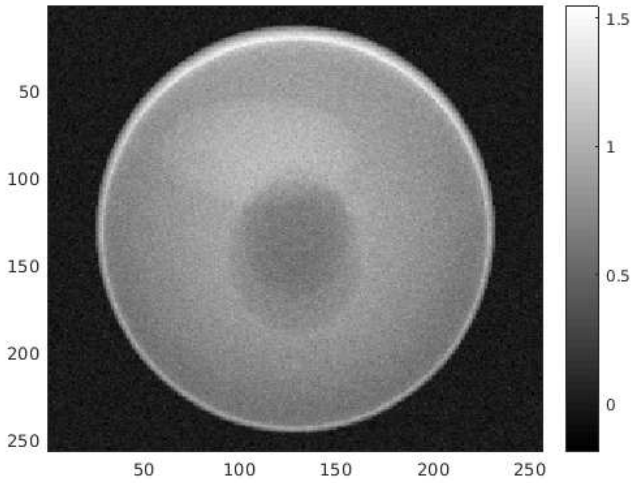


Figure 3.11: Noisy projection of Shepp-Logan phantom, with SNR= 20 db

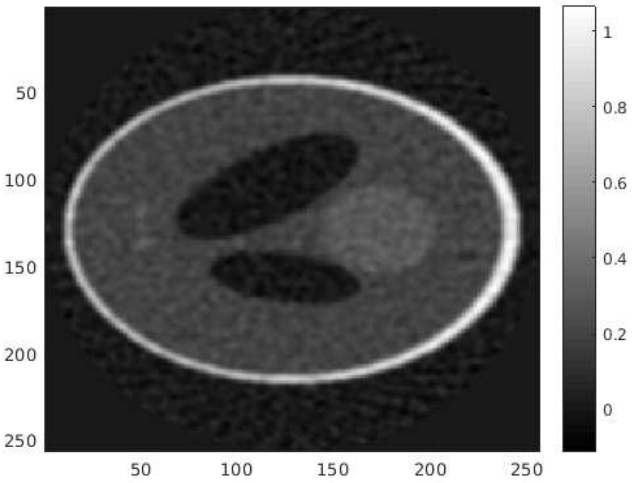


Figure 3.12: Reconstruction of Shepp-Logan phantom by the FDK method [FDK84]

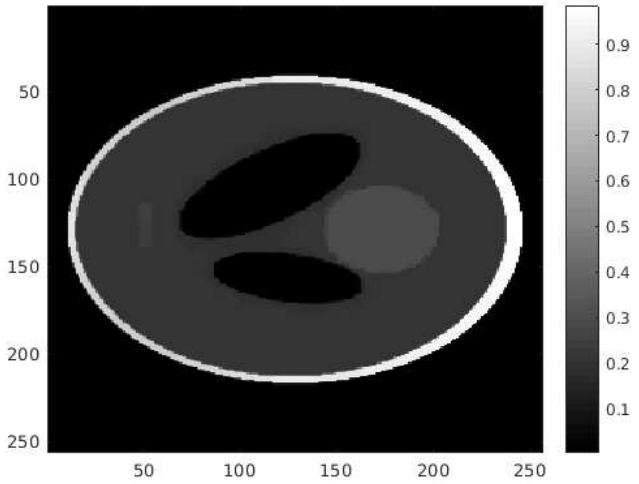


Figure 3.13: Reconstruction of Shepp-Logan phantom by PDFW [OMBF18] with the unmatched RD/VD pair

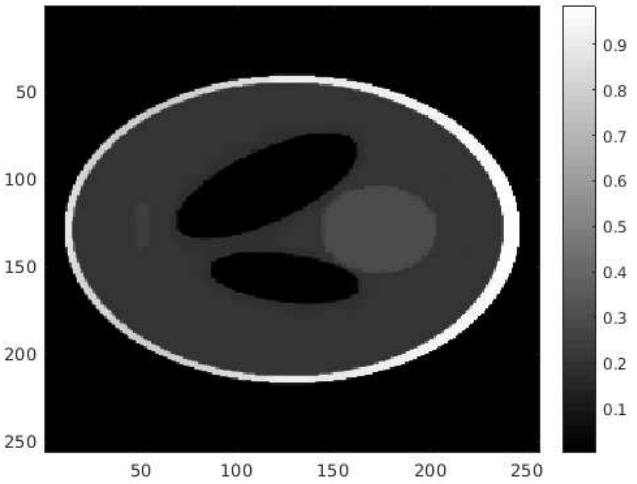


Figure 3.14: Reconstruction of Shepp-Logan phantom by PDFW [OMBF18] with the matched SF pair

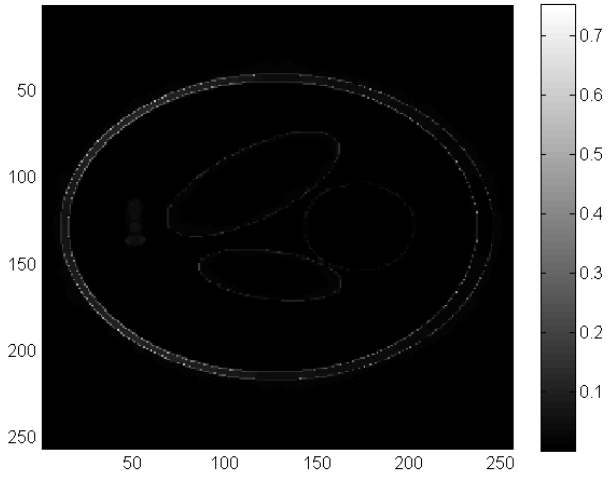


Figure 3.15: Difference between the original volume and the reconstruction by PDFW, with the unmatched RD/VD pair

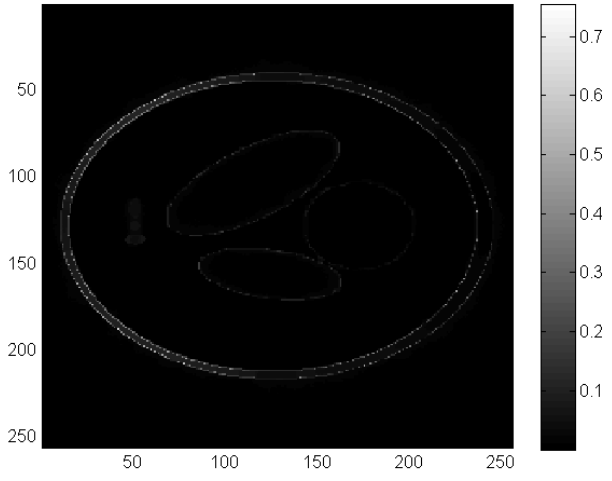


Figure 3.16: Difference between the original volume and the reconstruction by PDFW, with the matched SF pair

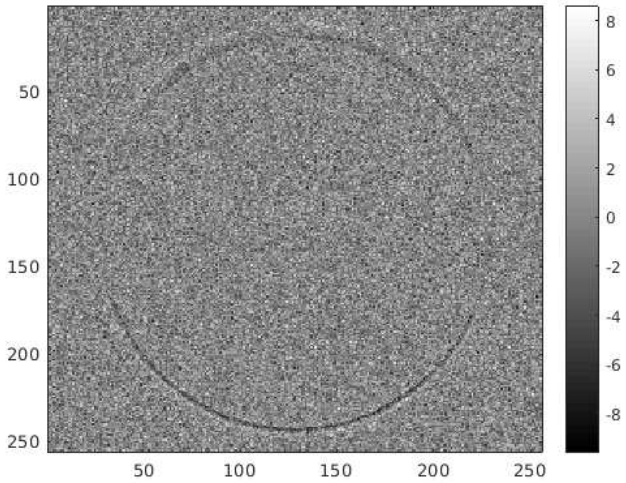


Figure 3.17: Estimated dual variable $\boldsymbol{p}$ (see algorithm 7) in PDFW [OMBF18] with the matched SF pair. The variable is size of the projections.

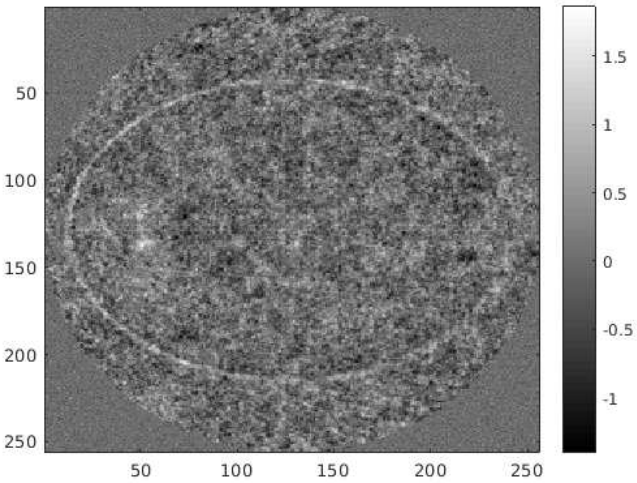
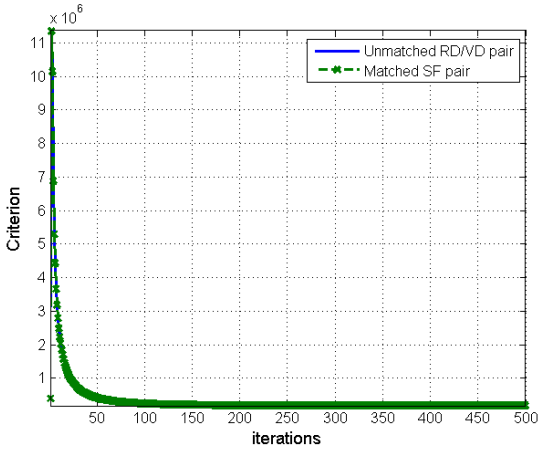
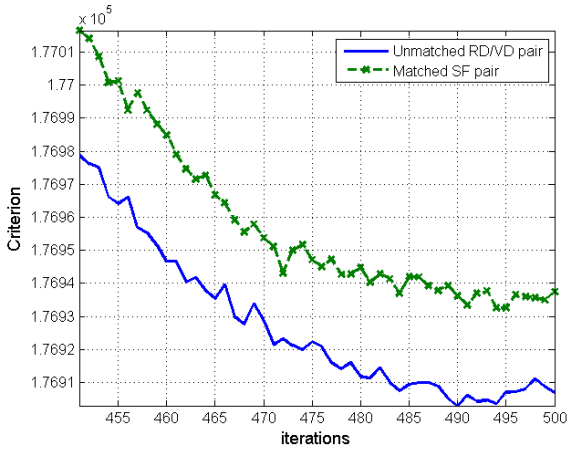


Figure 3.18: Estimated dual variable $\tilde{\boldsymbol{z}}$ (see algorithm 7) in PDFW [OMBF18] with the matched SF pair. The variable is size of the volume.

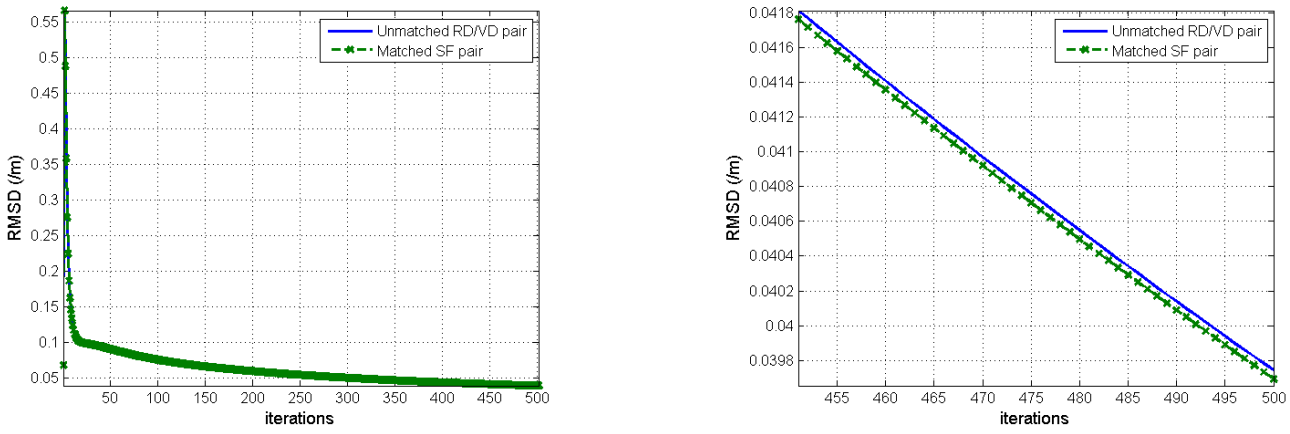


(a)



(b)

Figure 3.19: Evolution of the criterion during PDFW (a) with the RD/VD and SF pairs. A zoom on the last iterations is provided in (b).



(a) (b)

Figure 3.20: Evolution of the Root-Mean Square Difference (RMSD) between the estimate $\boldsymbol{f}^{(t)}$ and the real volume during PDFW (a) with the RD/VD and SF pairs. A zoom on the last iterations is provided in (b).

between the reconstruction $\hat{\boldsymbol{f}}$ and the real volume $\boldsymbol{f}_0$. We also evaluate the $\mathcal{L}_2$-relative error

$$\Delta_2 \boldsymbol{g} = \frac{\|\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}\|_2^2}{\|\boldsymbol{g}\|_2^2} \tag{3.40}$$

between the actual projections $\boldsymbol{g}$ and the theoretical ones $\boldsymbol{H}\hat{\boldsymbol{f}}$. We see that $\Delta_2 \boldsymbol{f}$ and $\Delta_2 \boldsymbol{g}$ are small with both the RD/VD and the SF pairs. They are slightly better for the SF pair but the difference with the RD/VD pair is not very pronounced. This confirms that the reconstructions obtained with the RD/VD and the SF pairs are similar. Nevertheless, the computation time with the unmatched RD/VD pair is shorter than the one with the matched SF pair. This is not surprising since, as we have seen in the previous section, the projection and backprojection operators are faster.

We also analyze the behaviours of PDFW through its iterations depending on the used P/BP pair. Figure 3.19 shows the evolution of the criterion given by equation (2.43). We see it decreases with both of the pairs. Zooming on the last iterations, the criterion is the lowest with the RD/VD pair. Nevertheless, this does not mean that the RD/VD pair has better results than the SF pair. Indeed, the criterion is not exactly the same for both pairs, since the used projectors $\boldsymbol{H}$ in (2.43) are different. In order to check the convergence independently from the used pair, in figure 3.20, we observe the evolution of the Root Mean Square Difference (RMSD)

$$RMSD = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(f_j^{(t)} - f_{0_j}\right)^2} \tag{3.41}$$

between the estimate $\boldsymbol{f}^{(t)}$ at iteration t and the real volume $\boldsymbol{f}_0$. We see that the RMSD decreases for both pairs. Zooming on the last iterations, the RMSD with the matched SF pair appears to be slighly lower than the one with the unmatched RD/VD pair. This confirms that the unmatched pair gives a suboptimal result [ASM16]. Nevertheless, in our case, as the difference between the RMSD of the pairs is very small, the suboptimality looks very negligible.

Since RMSD is not sensitive to minor localized errors, we deepen the analysis of the obtained reconstructions by focusing on a small region-of-interest (ROI) Ω in which little details are present. The zooms on this ROI in the reconstructions obtained with the RD/VD and the SF pairs, as the differences with respect to the original volume, are shown in figure 3.21. We see that, qualitatively, the reconstructions do no greatly differ. The Relative Mean Absolute Error (RMAE) computed within the ROI

$$RMAE = \frac{1}{|\Omega|}\sum_{j\in\Omega}\left|\frac{\hat{f}_j - f_{0_j}}{f_{0_j}}\right| \tag{3.42}$$

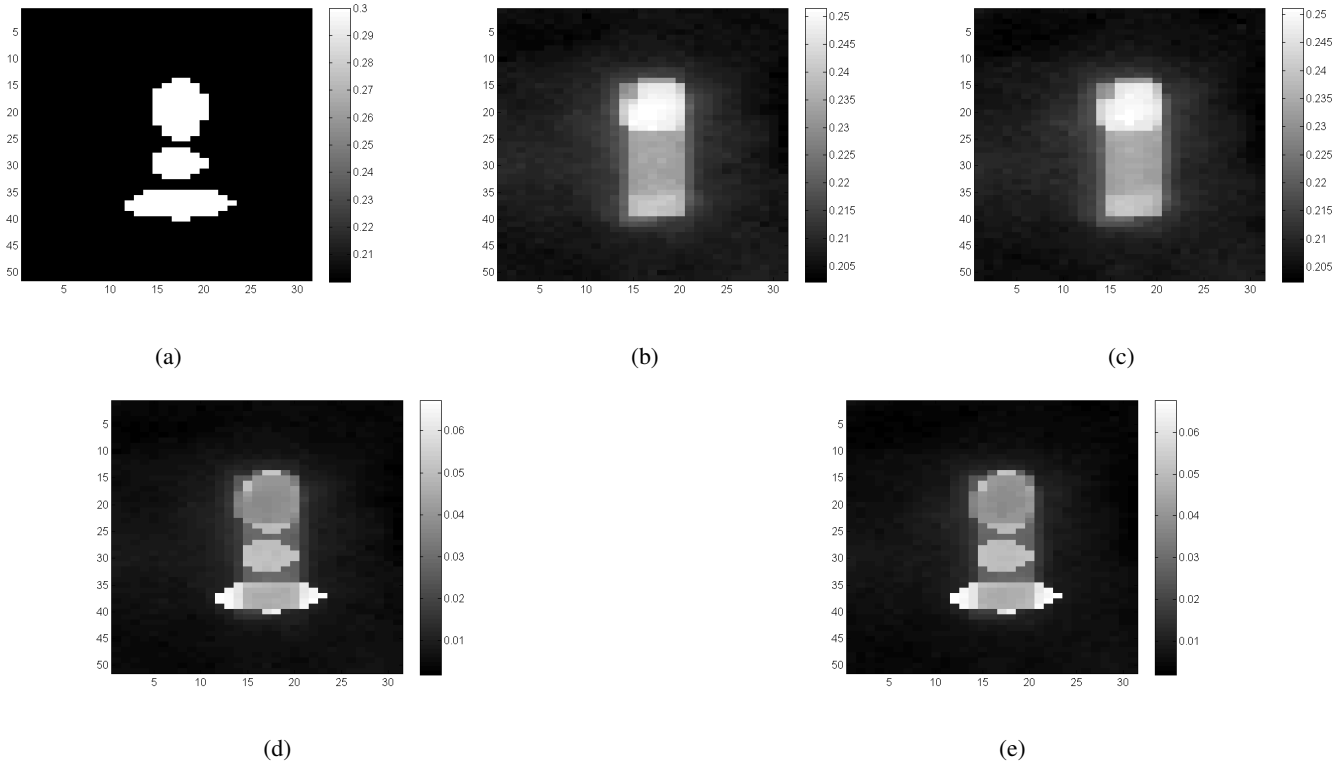


Figure 3.21: Zoom on a region-of-interest with little details for : the real volume (a), the reconstruction by PDFW [OMBF18] with the unmatched RD/VD pair (b), and the reconstruction by PDFW [OMBF18] with the matched SF pair (c). The differences with respect to the original volume are shown in (d) for the unmatched RD/VD pair, and in (e) for the matched SF pair.

with respect to the real volume is given in table 3.6 with the RD/VD pair and the SF pair. We see that it is practically the same with the RD/VD or the SF pair. Hence, the reconstructions obtained with the matched and the unmatched pairs seem to be equivalent.

P/BP pair	RMAE
Unmatched RD/VD	1.12 %
Matched SF	**1.10 %**

Table 3.6: Relative Mean Absolute Error (RMAE) of PDFW in small region-of-interest Ω, with the RD/VD pair and the SF pair

3.4 Conclusion and perspectives

In this chapter, we have compared an unmatched and a matched pair of projector and backprojector. The unmatched RD/VD pair is very suited to parallelization on the GPU and is consequently very fast. For the matched SF pair, we have investigated a new GPU implementation for the SF projector and backprojector which minimizes the memory transfers compared to its previous GPU implementations [WF11, XML$^+$17]. In particular, we have enhanced our work presented in [CGMDP18], by merging the two kernels of our GPU SF projector.

We have validated our unmatched and matched pairs as single modules and in a full MBIR method which was PDFW [OMBF18]. The results we obtained with each pair for the reconstruction of Shepp-Logan phantom were quite similar. The main difference between the pairs is the computational speed which is greater for the unmatched RD/VD pair. Nevertheless, we have also identified a small suboptimality for the unmatched RD/VD pair when checking the convergence to the real phantom. In chapter 4, after having detailed a joint reconstruction and segmentation algorithm, we will deepen the comparison by analyzing the behaviour of the algorithm with each of the pairs. In

addition, in chapter 6, we will show how the matched pair can be useful for the estimation of the uncertainties on the reconstruction, while this task is difficult and weakly reliable with the unmatched pair.

Concerning the implementations of the pairs, the unmatched RD/VD pair has been developed for almost ten years at L2S [BGMD$^+$15], and fully adapted for volumes of huge size such as 1024^3 or 2048^3 voxels, by multi-streaming and multi-GPUs calculations. For the moment, our GPU implementation of the SF pair is still very recent, and the computation time is quite long for volumes of great size. For this reason, in this chapter, we have limited our experiments to relatively small volumes of size 256^3 voxels. Accelerating and scaling our implementation for much larger volumes is the next step of our developments of the SF pair. For this purpose, further optimizations need to be studied. For instance, our proposed kernels for the GPU SF projector and backprojector do a lot of calculations. As a result, the number of registers used by the threads is large, which prevents from running many active warps. Hence, the number of registers needs to be reduced by taking advantage from other types of memory in a GPU, such as the shared memory. Furthermore, since the amount of memory used by each thread limits the number of active warps during the execution of the SF projector and backprojector, increasing the number of data processed by one thread may be a relevant way to reduce the computation time of our implementation. In addition, this would enable to further leverage the factorizations induced by the SF pair. After having proceeded with these optimizations and adapted the code for multi-streaming and multi-GPUs computing, comparisons with the previous GPU implementations of the SF pair [WF11, XML$^+$17] will remain to be done, as further studies of the use of the pair in applications for SAFRAN.

Chapter 4

Gauss-Markov-Potts prior model for joint reconstruction and segmentation

Many industrial parts at SAFRAN have in common that they are piecewise-constant (or quasi-piecewise-constant) volumes. These volumes are composed of several materials, each material filling one or several compact and quasi-homogeneous regions. Based on this prior information, we aim at proposing a MBIR algorithm in order to reconstruct such parts for NDT purposes.

Reconstructing piecewise-constant volumes has been widely considered in X-ray CT, for applications such as industrial inspection [BS11, ZKS+16], or, as a medical example, assessment of bone microstructure [WSRP16]. Intuitively, the piecewise-constant assumption can be introduced in the reconstruction by enforcing a sparse prior on the gradient of the volume. In this context, TV regularization gives good results. Thanks to the use of the l_1-norm, TV regularization is non-smooth [Nik02], so the boundaries in the reconstruction are sharp, as we have seen in section 3.3. On the contrary, Tikhonov regularization [Tik63] is not a good choice to reconstruct piecewise-constant volumes since it does not preserve the edges and tends to blur the reconstruction [NNZC08]. Compared to TV prior, a stronger sparsity for the gradient can be achieved by minimizing its l_0-norm : this is the Potts model [SWFU15]. Contrary to the case of TV, the reconstruction problem with the Potts model is NP-hard [SWFU15]. An algorithm, which is an instance of ADMM [BPC+11], has been proposed in [SWFU15] in order to solve the reconstruction problem regularized by the Potts model. The drawback of this algorithm is that the variable-splitting performed in ADMM makes it very memory-costly. Consequently, its applications are limited to 2D images [SWFU15], or to small 3D volumes [SRUW17]. Another prior suited to piecewise-constant volumes is sparse Haar transform [WMDG17]. Nevertheless, Haar transform is difficult to parallelize on the GPU, since computing a multilevel wavelet transformation requires to store many intermediate results which lead to many memory transactions between the CPU and the GPU [QJ16]. In addition, the reconstruction algorithm proposed in [WMDG17] has many variables of the size of the volume, which make it very memory-costly.

A widespread approach to reconstruct piecewise-constant or quasi-piecewise-constant volumes is to perform a segmentation of the volume jointly with its reconstruction [BS11, SWFU15]. The Discrete Algebraic Reconstruction Technique (DART) [BS11] is a typical example of this approach, in which a reconstruction step is alternated with a segmentation step. The specificity of DART is that, knowing the segmentation estimated at the previous iteration, DART reduces the computational complexity of the reconstruction step by only updating the voxels on the contours and, inside the regions, a given percentage of randomly selected "free" voxels. The reconstruction step is performed by applying a least-squares method which is SART [AK84]. An extension of DART to the polychromatic forward model has been proposed in [SBS18]. The segmentation step in DART is a thresholding algorithm for which the thresholds are given by the discrete gray values corresponding to each class in the volume. DART assumes that these gray values are perfectly known, which is never the case in practice. In addition, the reconstruction step in the original DART does not have regularization : this tends to concentrate the noise on the free voxels. Lastly, the percentage of free voxels to update is a parameter which is difficult to tune. In order to tackle all these problems, TVR-DART (total-variation-regularized DART) has been proposed [ZPB16]. As mentioned in the name of the algorithm, this reconstruction step is regularized by a total variation term [ZPB16]. In TVR-DART, the gray values and the thresholds are automatically estimated by including Newton's method steps in the algorithm. Nevertheless, these steps add projection operations to

the original version of DART. Contrary to DART, all the voxels are updated in the reconstruction step of TVR-DART.

In this chapter, we propose to reconstruct quasi-piecewise-constant volumes by a MBIR method which is a joint reconstruction and segmentation algorithm based on a Gauss-Markov-Potts prior model. This prior model has shown its effectiveness in microwave imaging [FDMD05], diffraction imaging [ADMD10] and image restoration [AMD10]. Gauss-Markov-Potts prior model assigns a prior model on the to-be-estimated volume and on the segmentation itself, which is an advantage compared to DART [BS11] and its derivatives [ZPB16, SBS18]. The prior model on the volume depends on the segmentation, while the prior on the segmentation is a Potts model which promotes compact regions in the reconstruction. In addition, since the inspected parts are not *exactly* piecewise-constant, Gauss-Markov-Potts prior model allows variability in the classes by introducing variances which are estimated, as are the means of the classes [AMD10]. Furthermore, our algorithm also optimizes the weights on the projections which are introduced in order to take into account that the noise is not uniformly distributed over the measurements.

Compared to the previously mentioned algorithms [SWFU15, WMDG17], our proposed method is less memory-costly, since it has few auxiliary variables which are of large sizes : while the estimated weights are the size of the projections, the segmentation is the size of the volume but requires less memory to be stored since it corresponds to labels which are integers. The remainder of this chapter presents all the details of our algorithm and its results on simulated and real data. In section 4.1, the used forward model is presented. In section 4.2, Gauss-Markov-Potts prior model for the volume is described. Based on this prior and on the used forward model, in section 4.3, the joint reconstruction and segmentation algorithm is derived by Joint Maximization A Posteriori (JMAP) estimation. Then, experimental results in simulation and on real data from SAFRAN are presented. The results are compared to TV-penalized weighted least-squares solved by PDFW [OMBF18] presented in chapter 2. Conclusion and perspectives for this work are given in section 4.6.

A first attempt to use Gauss-Markov-Potts prior in 3D X-ray CT for joint reconstruction and segmentation has been made in [FSVMD07]. Nevertheless, the authors in [FSVMD07] only validate the used pair of projector and backprojector and do not present results of the algorithm. In addition, the purpose was to reconstruct very little volumes (typically of size 64^3 voxels) in microtomography. Due to the little dimensions of the data, it was possible to perform the segmentation step by a Monte Carlo Markov Chain (MCMC) technique [FSVMD07]. In our industrial context, this MCMC step is not feasible due to the fact that the inspected volumes are far bigger (typically discretized into 512^3 or 1024^3 voxels). In order to deal with the huge dimension, we replace this MCMC step by Iterated Conditional Modes (ICM) algorithm [Bes86], which performs local optimization. Furthermore, in [FSVMD07], the role of the parameters of the algorithm is not studied : in section 4.3.6, the influence of each parameter is explained, and clear strategies are provided in order to fix it. At last, in our simulation experiments in section 4.4, we compare the influence on the algorithm of using a matched or an unmatched pair of projector and backprojector.

4.1 Forward model

The used forward model is denoted by $\mathcal{U}$. As explained in chapter 1, this forward model is linear and takes the uncertainties into account :

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\zeta} \tag{4.1}$$

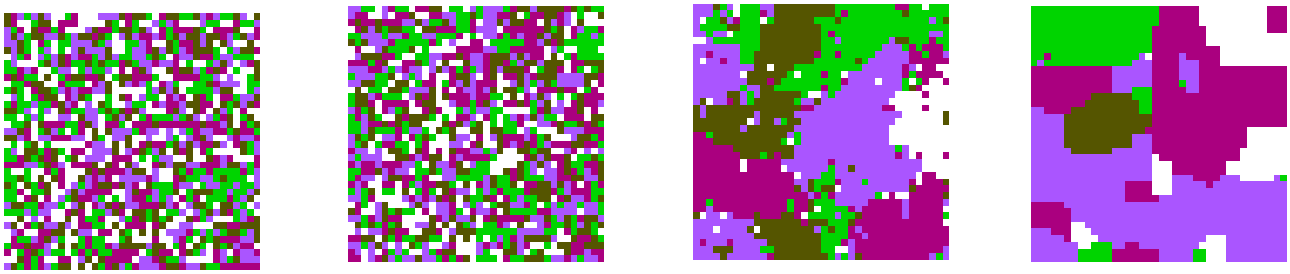
and uncertainties $\boldsymbol{\zeta}$ are modeled as Gaussian

$$p(\zeta_i|v_{\zeta_i}) = \mathcal{N}(\zeta_i|0, v_{\zeta_i}), \forall i \in \{1, \dots, M\}\,. \tag{4.2}$$

Expression (2.31) can be used for the variances of the uncertainties $\boldsymbol{v}_\zeta = (v_{\zeta_i})_i$. Nevertheless, as pointed out in section 2.4, scattering and beam-hardening are not taken into account in the derivation of this expression which is detailed in appendix A. Consequently, expression (2.31) can be used for the variances of the uncertainties can be seen as suboptimal. For this reason, we choose to estimate $\boldsymbol{v}_\zeta$ jointly with the volume. To do so, we assign a conjugate Inverse-Gamma prior on variances $\boldsymbol{v}_\zeta$

$$p(v_{\zeta_i}|\alpha_{\zeta_0}, \beta_{\zeta_0}) = \mathcal{IG}(v_{\zeta_i}|\alpha_{\zeta_0}, \beta_{\zeta_0}) = \frac{\beta_{\zeta_0}^{\alpha_{\zeta_0}}}{\Gamma(\alpha_{\zeta_0})} v_{\zeta_i}^{-\alpha_{\zeta_0}-1} \exp\left[-\frac{\beta_{\zeta_0}}{v_{\zeta_i}}\right], v_{\zeta_i} > 0, \forall i, \tag{4.3}$$

where Γ denotes Euler Gamma function, and α_{ζ_0} and β_{ζ_0} are fixed parameters.



$\gamma_0 = 0.5$ $\gamma_0 = 0.7$ $\gamma_0 = 0.8$ $\gamma_0 = 1.6$

Figure 4.1: Potts field $\boldsymbol{z}$ for different values of γ_0

4.2 Gauss-Markov-Potts prior model on the volume

Gauss-Markov-Potts prior model enables one to link the gray level f_j of a voxel j to the material to which it belongs. This material is identified by a label z_j assigned to voxel j : we set $z_j = k$ if voxel j is in material k, $k \in \{1, \ldots, K\}$, where K is the number of materials in the volume. The compacity of a region composed of material k is enforced thanks to a Potts model for labels $\boldsymbol{z}$. This model is markovian in the sense that the label of one voxel depends on the labels of its neighbours : a voxel of which the neighbours are in a same class k has a high probability to be also in this class. Knowing the labels of the neighbours of a voxel j, the probability for this voxel to be in class k, depending on the labels in its neighbourhood $\mathcal{V}(j)$, reads

$$p(z_j = k | z_i, i \in \mathcal{V}(j); \alpha_k, \gamma_0) \propto \exp\left[\alpha_k + \gamma_0 \sum_{i \in \mathcal{V}(j)} \delta(k - z_i)\right], \tag{4.4}$$

for all $k \in \{1, \ldots, K\}$. Hence, the prior for label z_j is

$$p(z_j | z_i, i \in \mathcal{V}(j); \alpha_k, \gamma_0) \propto \exp\left[\sum_{k=1}^{K} \alpha_k \delta(z_j - k) + \gamma_0 \sum_{i \in \mathcal{V}(j)} \delta(z_j - z_i)\right]. \tag{4.5}$$

In (4.4) and (4.5), singletons' energies $\boldsymbol{\alpha} = (\alpha_k)_k$ insert a prior knowledge about the probability for voxel j to be in class k. This probability is given by e^{α_k}, so we have the constraint [AMD10]

$$\sum_{k=1}^{K} e^{\alpha_k} = 1. \tag{4.6}$$

Another parameter in (4.4) and (4.5) is γ_0, which is called the Potts coefficient or, equivalently, the granularity coefficient [PDBT13] : it weights the dependence of label z_j on the neighbours and, consequently, has a huge influence on the granularity of the field $\boldsymbol{z}$. An interesting fact is that there exists a critical value γ_c for Potts coefficient γ_0 so that the regions are compact if $\gamma_0 \geqslant \gamma_c$ [Ons44, Hua87]. Figure 4.1 illustrates the influence of γ_0 by showing the middle slices of generated $64 \times 64 \times 64$ Potts fields with $K = 5$ classes, for different values of γ_0. Rather surprisingly, the granularity of the field sharply changes when we increase γ_0 from 0.7 to 0.8, which means that $0.7 \leqslant \gamma_c \leqslant 0.8$. The critical value γ_c can only be computed for the Ising field case (for which we have $K = 2$ classes) in 2D [Gio10].

Using Hammersley-Clifford theorem [Bes74], the joint prior distribution for labels $\boldsymbol{z}$ is

$$p(\boldsymbol{z} | \boldsymbol{\alpha}, \gamma_0) \propto \exp\left[\sum_{j=1}^{N} \left(\sum_{k=1}^{K} \alpha_k \delta(z_j - k) + \gamma_0 \sum_{i \in \mathcal{V}(j)} \delta(z_j - z_i)\right)\right] \tag{4.7}$$

The markovian term

$$\Phi_{in}(\boldsymbol{z}) = \sum_{j=1}^{N} \gamma_0 \sum_{i \in \mathcal{V}(j)} \delta(z_j - z_i) \tag{4.8}$$

is called the inner energy of Potts field $\boldsymbol{z}$ and is related to its compacity, while the term

$$\Phi_{out}(\boldsymbol{z}) = \sum_{j=1}^{N} \sum_{k=1}^{K} \alpha_k \delta(z_j - k) \tag{4.9}$$

is the outer energy of the field.

In a region exclusively composed of material k, the distribution of gray levels is modeled as Gaussian, with mean and variance m_{jk} and v_{jk} which depend on the label :

$$p(f_j|z_j = k, m_{jk}, v_{jk}) = \mathcal{N}(f_j|m_{jk}, v_{jk}) . \tag{4.10}$$

Several choices can be made for m_{jk} and v_{jk}. First, they can be chosen independent from j : this leads to model the volume as a Mixture of Independent Gaussians (MIG) [FDMD05, ADMD10, AMD10] :

$$\begin{cases} m_{jk} = m_k \\ v_{jk} = v_k \end{cases} \quad \text{(MIG)}. \tag{4.11}$$

Other Gauss-Markov-Potts prior models have been proposed : in [AMD10], a Markovian Mixture of Gaussians (MMG) is considered but, compared to the MIG model, it is more costly due to the computation of the contours of the regions. Recently, another Gauss-Markov-Potts prior model has been presented in order to account for textured classes [GV17]. Nevertheless, this model is very memory-costly in 3D since it requires to store K textured patches of the size of the volume. In the following, we only deal with the MIG Gauss-Markov-Potts prior model for the volume. For this model, the prior distribution of the volume given the labels, the means and the variances of the classes is

$$p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v}) = (2\pi)^{-\frac{N}{2}} \det(\boldsymbol{V}_{\boldsymbol{z}})^{-1/2} \exp\left[-\frac{1}{2}\|\boldsymbol{f} - \boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}\right], \tag{4.12}$$

where

$$m_{z_j} = m_k \text{ if } z_j = k, \forall j, \tag{4.13}$$

$\boldsymbol{V}_{\boldsymbol{z}} = \text{diag}[\boldsymbol{v}_{\boldsymbol{z}}]$, and

$$v_{z_j} = v_k \text{ if } z_j = k, \forall j. \tag{4.14}$$

We consider means $\boldsymbol{m}$ and variances $\boldsymbol{v}$ of the classes as unknowns and assign conjugate priors to it. These priors are respectively Gaussian and Inverse-Gamma :

$$\begin{cases} p(m_k|m_0, v_0) = \mathcal{N}(m_k|m_0, v_0) \\ p(v_k|\alpha_0, \beta_0) = \mathcal{IG}(v_k|\alpha_0, \beta_0) \end{cases}, \forall k, \tag{4.15}$$

where m_0, v_0, α_0 and β_0 are fixed parameters.

4.3 Joint Maximization A Posteriori (JMAP) algorithm

Figure 4.2 shows the overall hierarchical model when Gauss-Markov-Potts prior is used. Projections $\boldsymbol{g}$ result from volume $\boldsymbol{f}$ (through projection operator $\boldsymbol{H}$) and from uncertainties $\boldsymbol{\zeta}$ with variances $\boldsymbol{v}_\zeta$. Volume $\boldsymbol{f}$ is described by its Gauss-Markov-Potts prior model $\mathcal{M}$ with hyperparameters $\boldsymbol{\theta} = (\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v})$. Labels $\boldsymbol{z}$ are assigned a Potts model with singletons' energies $\boldsymbol{\alpha}$ and with granularity coefficient γ_0. Means $\boldsymbol{m}$ of the classes have a Gaussian prior of mean m_0 and variance v_0, while variances $\boldsymbol{v}$ of the classes have an Inverse-Gamma prior of shape parameter α_0 and intensity parameter β_0. Variances $\boldsymbol{v}_\zeta$ of the uncertainties are also assigned an Inverse-Gamma prior of parameters $(\alpha_{\zeta_0}, \beta_{\zeta_0})$ according to forward model $\mathcal{U}$. The unknowns we want to estimate are

$$\boldsymbol{\psi} = (\boldsymbol{f}, \boldsymbol{v}_\zeta, \boldsymbol{\theta}) = (\boldsymbol{f}, \boldsymbol{v}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v}). \tag{4.16}$$

According to Bayes' rule, their joint posterior distribution reads

$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{v}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v}|\boldsymbol{g}; \mathcal{U}, \mathcal{M}) \propto \quad & p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{v}_\zeta) p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v}) p(\boldsymbol{v}_\zeta|\alpha_{\zeta_0}, \beta_{\zeta_0}) \\ & p(\boldsymbol{z}|\boldsymbol{\alpha}; \gamma_0) p(\boldsymbol{m}|m_0, v_0) p(\boldsymbol{v}|\alpha_0, \beta_0) \end{aligned} \tag{4.17}$$

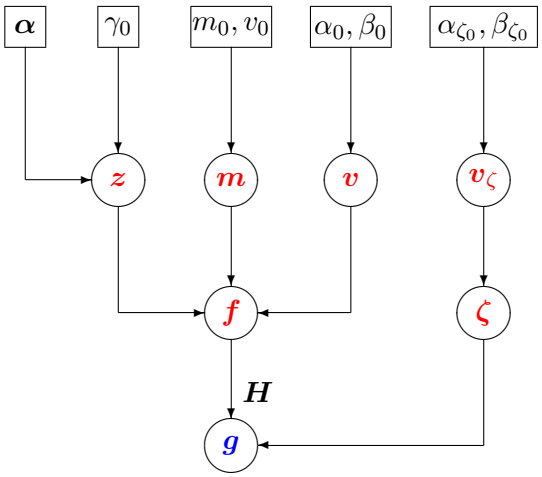


Figure 4.2: Hierarchical model with Gauss-Markov-Potts prior

where

$$p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{v}_\zeta) \propto \det\left(\boldsymbol{V}_\zeta\right)^{-1/2} \exp\left[-\frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta}\right], \tag{4.18}$$

$$p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{v}) \propto \det\left(\boldsymbol{V}_{\boldsymbol{z}}\right)^{-1/2} \exp\left[-\frac{1}{2}\|\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}\right], \tag{4.19}$$

$$p(\boldsymbol{v}_\zeta|\alpha_{\zeta_0},\beta_{\zeta_0}) \propto \exp\left[-\sum_{i=1}^{M}\left((\alpha_{\zeta_0}+1)\ln v_{\zeta_i}+\frac{\beta_{\zeta_0}}{v_{\epsilon_i}}\right)\right], \tag{4.20}$$

$$p(\boldsymbol{z}|\boldsymbol{\alpha};\gamma_0) \propto \exp\left[\sum_{j=1}^{N}\left(\sum_{k=1}^{K}\alpha_k\delta(z_j-k)+\gamma_0\sum_{i\in\mathcal{V}(j)}\delta(z_j-z_i)\right)\right], \tag{4.21}$$

$$p(\boldsymbol{m}|m_0,v_0) \propto v_0^{-\frac{K}{2}} \exp\left[-\frac{1}{2v_0}\sum_{k=1}^{K}(m_k-m_0)^2\right], \tag{4.22}$$

$$p(\boldsymbol{v}|\alpha_0,\beta_0) \propto \exp\left[-\sum_{k=1}^{K}\left((\alpha_0+1)\ln v_k+\frac{\beta_0}{v_k}\right)\right]. \tag{4.23}$$

In order to estimate the unknowns, Monte-Carlo Markov-Chain (MCMC) method by Gibbs sampling can be used [FDMD05, ADMD10]. Nevertheless, the computational complexity of Gibbs sampling with these priors has been studied [ZBKT16] and shows that these methods are unaffordable for our application with large 3D volumes. For this reason, we choose to perform a joint maximization a posteriori (JMAP) :

$$(\hat{\boldsymbol{f}},\hat{\boldsymbol{v}}_\zeta,\hat{\boldsymbol{z}},\hat{\boldsymbol{m}},\hat{\boldsymbol{v}}) = \underset{(\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{v})}{\arg\max} \left\{p(\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{v}|\boldsymbol{g};\mathcal{U},\mathcal{M})\right\}. \tag{4.24}$$

Because the posterior distribution (4.17) is multi-modal, estimation (4.24) is done approximately by the following alternate optimization algorithm :

$$\begin{cases}
\boldsymbol{f}^{(t+1)} = \arg\max_{\boldsymbol{f}}\left\{p(\boldsymbol{f},\boldsymbol{v}_\zeta^{(t)},\boldsymbol{z}^{(t)},\boldsymbol{m}^{(t)},\boldsymbol{v}^{(t)}\mid\boldsymbol{g};\mathcal{U},\mathcal{M})\right\} & \text{(4.25a)}\\
\boldsymbol{v}_\zeta^{(t+1)} = \arg\max_{\boldsymbol{v}_\zeta}\left\{p(\boldsymbol{f}^{(t+1)},\boldsymbol{v}_\zeta,\boldsymbol{z}^{(t)},\boldsymbol{m}^{(t)},\boldsymbol{v}^{(t)}\mid\boldsymbol{g};\mathcal{U},\mathcal{M})\right\} & \text{(4.25b)}\\
\boldsymbol{z}^{(t+1)} = \arg\max_{\boldsymbol{z}}\left\{p(\boldsymbol{f}^{(t+1)},\boldsymbol{v}_\zeta^{(t+1)},\boldsymbol{z},\boldsymbol{m}^{(t)},\boldsymbol{v}^{(t)}\mid\boldsymbol{g};\mathcal{U},\mathcal{M})\right\} & \text{(4.25c)}\\
\boldsymbol{m}^{(t+1)} = \arg\max_{\boldsymbol{m}}\left\{p(\boldsymbol{f}^{(t+1)},\boldsymbol{v}_\zeta^{(t+1)},\boldsymbol{z}^{(t+1)},\boldsymbol{m},\boldsymbol{v}^{(t)}\mid\boldsymbol{g};\mathcal{U},\mathcal{M})\right\} & \text{(4.25d)}\\
\boldsymbol{v}^{(t+1)} = \arg\max_{\boldsymbol{v}}\left\{p(\boldsymbol{f}^{(t+1)},\boldsymbol{v}_\zeta^{(t+1)},\boldsymbol{z}^{(t+1)},\boldsymbol{m}^{(t+1)},\boldsymbol{v}\mid\boldsymbol{g};\mathcal{U},\mathcal{M})\right\} & \text{(4.25e)}
\end{cases}$$

Due to the estimation of both the volume and the labels, our algorithm is a joint reconstruction and segmentation algorithm. The stopping criterion $\mathcal{L}$ is derived from the logarithm of the joint posterior distribution (4.17), from which constants are removed :

$$\begin{aligned}\mathcal{L}(\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{v}) = &-\frac{1}{2}\sum_{i=1}^{M}\frac{(g_i-[\boldsymbol{H}\boldsymbol{f}]_i)^2}{v_{\zeta_i}} - \frac{1}{2}\sum_{i=1}^{M}\ln(v_{\zeta_i}) - \sum_{i=1}^{M}\left((\alpha_{\zeta_0}+1)\ln v_{\zeta_i} + \frac{\beta_{\zeta_0}}{v_{\zeta_i}}\right)\\ &+\sum_{j=1}^{N}\left(\sum_{k=1}^{K}\left[\alpha_k - \frac{1}{2}\left(\ln(v_k) + \frac{(f_j-m_k)^2}{v_k}\right)\right]\delta(z_j-k) + \gamma_0\sum_{i\in\mathcal{V}(j)}\delta(z_j-z_i)\right)\\ &-\frac{1}{2v_0}\sum_{k=1}^{K}(m_k-m_0)^2 - \sum_{k=1}^{K}\left((\alpha_0+1)\ln v_k + \frac{\beta_0}{v_k}\right).\end{aligned} \tag{4.26}$$

According to Bayes' rule, each step of the algorithm (4.25) is equivalent to maximizing the conditional distribution of each unknown given the others. In the following of this section, we detail these steps one after the other. For the sake of readibility, in each subsection describing the steps of the algorithm, subscripts t, $0 \leqslant t < t_{max}$, corresponding to the iterations of the algorithm, are omitted. Then, in a final subsection, we discuss the initialization and the choice of the parameters.

4.3.1 Estimation of the volume

The conditional distribution of the volume given the other unknowns reads

$$\begin{aligned}p(\boldsymbol{f}|\boldsymbol{g},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{v};\boldsymbol{g},\mathcal{U},\mathcal{M}) &\propto p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{v}_\zeta)p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{v})\\ &\propto \exp\left[-\frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta} - \frac{1}{2}\|\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}\right].\end{aligned} \tag{4.27}$$

Hence, step (4.25a) leads to the minimization of the criterion

$$J_f(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta} + \frac{1}{2}\|\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}. \tag{4.28}$$

As we see, this minimization is an instance of penalized weighted least-squares presented in chapter 2. The regularization $\frac{1}{2}\|\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}$, derived from Gauss-Markov-Potts prior model, is very intuitive and can be interpreted : since the volume is quasi-piecewise-constant, the value of a voxel is a priori approximately equal to the mean of its material, with a certain tolerance given by the variance of the class.

The optimum of $J_f(\boldsymbol{f})$

$$\boldsymbol{f}^* = \left(\boldsymbol{H}^T\boldsymbol{V}_\zeta^{-1}\boldsymbol{H} + \boldsymbol{V}_{\boldsymbol{z}}^{-1}\right)^{-1}\left(\boldsymbol{H}^T\boldsymbol{V}_\zeta^{-1}\boldsymbol{g} + \boldsymbol{V}_{\boldsymbol{z}}^{-1}\boldsymbol{m}_{\boldsymbol{z}}\right) \tag{4.29}$$

cannot be computed exactly due to the inversion of the huge matrix $\left(\boldsymbol{H}^T\boldsymbol{V}_\zeta^{-1}\boldsymbol{H} + \boldsymbol{V}_{\boldsymbol{z}}^{-1}\right)$. As a result, we need to run an optimization algorithm in order to get an approximation of $\boldsymbol{f}^*$. Since the data-matching and the regularization terms in (4.28) are both quadratic, we can apply a simple gradient descent

$$\boldsymbol{f}^{(n+1)} = \boldsymbol{f}^{(n)} - \mu_n\nabla J_f\left(\boldsymbol{f}^{(n)}\right) \tag{4.30}$$

where the gradient of the criterion is

$$\nabla J(\boldsymbol{f}) = \boldsymbol{H}^T\boldsymbol{V}_\zeta^{-1}(\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g}) + \boldsymbol{V}_{\boldsymbol{z}}^{-1}(\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}) \tag{4.31}$$

and where the stepsize $\mu_n > 0$ is adapted such that

$$\mu_n = \arg\min_{\mu>0} J_f\left(\boldsymbol{f}^{(n)} - \mu\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\right) = \frac{\|\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|_2^2}{\nabla J_f\left(\boldsymbol{f}^{(n)}\right)^T\boldsymbol{H}^T\boldsymbol{V}_\zeta^{-1}\boldsymbol{H}\nabla J_f\left(\boldsymbol{f}^{(n)}\right) + \|\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}}. \tag{4.32}$$

The gradient descent is stopped when n reaches the maximum number of iterations n_{vol} which is fixed by the user for this descent. Concerning formulae (4.31) and (4.32), they are valid for the case a matched pair is used. If the used pair is unmatched, the adjoint operator of $\boldsymbol{H}$ is approximated by a backprojector $\boldsymbol{B} \approx \boldsymbol{H}^T$ in order to accelerate the computations, as seen in chapter 3. Hence, for the case of an unmatched pair, we replace $\boldsymbol{H}^T$ by $\boldsymbol{B}$ in formulae (4.31) and (4.32), so the gradient is approximately computed by

$$\nabla J(\boldsymbol{f}) \approx \boldsymbol{B}\boldsymbol{V}_{\zeta}^{-1}(\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g}) + \boldsymbol{V}_{\boldsymbol{z}}^{-1}(\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}) \tag{4.33}$$

and the expression of the adapted stepsize is

$$\mu_n \approx \frac{\|\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|_2^2}{\nabla J_f\left(\boldsymbol{f}^{(n)}\right)^T \boldsymbol{B}\boldsymbol{V}_{\zeta}^{-1}\boldsymbol{H}\nabla J_f\left(\boldsymbol{f}^{(n)}\right) + \|\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|_{\boldsymbol{V}_{\boldsymbol{z}}}^2}. \tag{4.34}$$

4.3.2 Estimation of the variances of the uncertainties

The conditional distribution of $\boldsymbol{v}_{\zeta}$ given the other unknowns is fully factorized

$$\begin{aligned} p(\boldsymbol{v}_{\zeta}|\boldsymbol{f},\boldsymbol{z},\boldsymbol{m},\boldsymbol{v};\boldsymbol{g},\mathcal{U},\mathcal{M}) &\propto p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{v}_{\zeta})p(\boldsymbol{v}_{\zeta}|\alpha_{\zeta_0},\beta_{\zeta_0}) \\ &\propto \prod_i \frac{1}{v_{\zeta_i}^{\frac{1}{2}}} \exp\left(-\frac{1}{2v_{\zeta_i}}(g_i-[\boldsymbol{H}\boldsymbol{f}]_i)^2\right) \times \prod_i \frac{1}{v_{\zeta_i}^{\alpha_{\zeta_0}+1}} \exp\left(-\frac{\beta_{\zeta_0}}{v_{\zeta_i}}\right) \\ &\propto \prod_i \frac{1}{v_{\zeta_i}^{\alpha_{\zeta_0}+\frac{3}{2}}} \exp\left(-\frac{1}{v_{\zeta_i}}\left(\beta_{\zeta_0}+\frac{1}{2}(g_i-[\boldsymbol{H}\boldsymbol{f}]_i)^2\right)\right). \end{aligned} \tag{4.35}$$

Given (4.35), the conditional distribution of v_{ζ_i} is an Inverse-Gamma distribution

$$p(v_{\zeta_i}|\boldsymbol{f},\boldsymbol{z},\boldsymbol{m},\boldsymbol{v};\boldsymbol{g},\mathcal{U},\mathcal{M}) = \mathcal{IG}(v_{\zeta_i}|\hat{\alpha}_{\zeta_{0_i}},\hat{\beta}_{\zeta_{0_i}}) \tag{4.36}$$

of which the shape parameter $\hat{\alpha}_{\zeta_{0_i}}$ and the intensity parameter $\hat{\beta}_{\zeta_{0_i}}$ are

$$\begin{cases} \hat{\alpha}_{\zeta_{0_i}} = \alpha_{\zeta_0} + \frac{1}{2} \\ \hat{\beta}_{\zeta_{0_i}} = \beta_{\zeta_0} + \frac{1}{2}(g_i-[\boldsymbol{H}\boldsymbol{f}]_i)^2 \end{cases}, \forall i \in \{1,\ldots,M\}. \tag{4.37}$$

This distribution is maximized at

$$\hat{v}_{\zeta_i} = \frac{\hat{\beta}_{\zeta_{0_i}}}{\hat{\alpha}_{\zeta_{0_i}}+1}, \tag{4.38}$$

which is the update formula for v_{ζ_i}, $\forall i \in \{1,\ldots,M\}$, in step (4.25b) of JMAP algorithm.

4.3.3 Estimation of the labels

The conditional distribution of the labels $\boldsymbol{z}$ reads

$$\begin{aligned} &p(\boldsymbol{z}|,\boldsymbol{f},\boldsymbol{v}_{\zeta},\boldsymbol{m},\boldsymbol{v};\boldsymbol{g},\mathcal{U},\mathcal{M})) \propto p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{v})p(\boldsymbol{z}|\gamma_0,\boldsymbol{\alpha}) \\ &\propto \exp\left[\sum_{j=1}^{N}\left(\sum_{k=1}^{K}\left[\alpha_k - \frac{1}{2}\left(\ln(v_k) + \frac{(f_j-m_k)^2}{v_k}\right)\right]\delta(z_j-k) + \gamma_0 \sum_{i\in\mathcal{V}(j)}\delta(z_j-z_i)\right)\right] \\ &\propto \exp\left[E\left(\boldsymbol{z}|\boldsymbol{f},\boldsymbol{m},\boldsymbol{v};\boldsymbol{\alpha},\gamma_0\right)\right] \end{aligned} \tag{4.39}$$

where

$$E\left(\boldsymbol{z}|\boldsymbol{f},\boldsymbol{m},\boldsymbol{v};\boldsymbol{\alpha},\gamma_0\right) = \sum_{j=1}^{N}\left(\sum_{k=1}^{K}\left[\alpha_k - \frac{1}{2}\left(\ln(v_k) + \frac{(f_j-m_k)^2}{v_k}\right)\right]\delta(z_j-k) + \gamma_0 \sum_{i\in\mathcal{V}(j)}\delta(z_j-z_i)\right) \tag{4.40}$$

is Potts energy. It can be rewritten as

$$E\left(\boldsymbol{z}|\boldsymbol{f},\boldsymbol{m},\boldsymbol{v};\boldsymbol{\alpha},\gamma_0\right)=\sum_{j=1}^{N}\left(\sum_{k=1}^{K}\alpha_{jk}\delta(z_j-k)+\gamma_0\sum_{i\in\mathcal{V}(j)}\delta(z_j-z_i)\right) \tag{4.41}$$

where

$$\alpha_{jk}=\alpha_k-\frac{1}{2}\left(\ln\left(v_k\right)+\frac{\left(f_j-m_k\right)^2}{v_k}\right),\forall j,k. \tag{4.42}$$

As we see in (4.39), step (4.25c) of JMAP algorithm consists in finding $\boldsymbol{z}$ which maximizes Potts energy. Due to the Markovian term in (4.41), the exact computation of the optimum of (4.41) is unfeasible [ZBS01]. As a result, we need to get an approximate solution by applying a segmentation algorithm. Among possible algorithms, simulated annealing (SA) [GG84] is able to converge to the global optimum, but on the condition that a slow annealing is performed [HKK+97]. Consequently, SA is too slow for an industrial application. Other algorithms looking for a global optimum are graph-cut methods [BVZ01], which transform the problem of the maximization of (4.41) in a problem of maximum flow in a graph. Nevertheless, parallelizing a maximum flow algorithm in order to deal with huge volumes, as it is our case in 3D CT, is not feasible [GSS82], and prevents from the use of graph-cut methods to achieve the estimation of $\boldsymbol{z}$.

Another segmentation method is the Iterated Conditional Modes (ICM) [Bes86] algorithm, which converges fast and is easily and highly parallelizable. The idea of ICM is to divide the voxels into two disjoint subsets, called ”black” voxels and ”white” voxels. These subsets are such that the neighbours of a ”white” voxel are only ”black” voxels, and vice versa. Figure 4.3 illustrates this subdivision in 2D. We denote by $\boldsymbol{z}_B$ the labels for ”black” voxels, and by $\boldsymbol{z}_W$ the labels for ”white” voxels. After having done this subdivision, we see that, given the ”white” voxels, ”black” voxels are independent from each other, and this is the same for ”white” voxels given the ”black” voxels. Since the subsets are disjoint, we can write Potts energy as a function of $\boldsymbol{z}_B$ and $\boldsymbol{z}_W$:

$$E\left(\boldsymbol{z}|\boldsymbol{f},\boldsymbol{m},\boldsymbol{v};\boldsymbol{\alpha},\gamma_0\right)=E\left(\boldsymbol{z}_B,\boldsymbol{z}_W|\boldsymbol{f},\boldsymbol{m},\boldsymbol{v};\boldsymbol{\alpha},\gamma_0\right). \tag{4.43}$$

Hence, Potts energy can be maximized alternately with respect to $\boldsymbol{z}_B$ and $\boldsymbol{z}_W$. The labellings for ”white” voxels and ”black” voxels are performed one after the other by alternating maximization. This algorithm is clearly massively parallelizable since all the ”white” voxels (respectively all the ”black” voxels) in figure 4.3 can be labelled at the same time. ICM is known to converge to a solution very fast but its disadvantage is that, contrary to SA and graph-cut methods, this solution is only a local optimum [HD92, HKK+97]. Hence, its initialization is very important. We may also underline that, in order to make ICM applicable, the neighbourhood $\mathcal{V}(j)$ of every voxel j must be its first-order neighbourhood, i.e. its six nearest neighbours and only them. If this is not the case, then labels $\boldsymbol{z}_W$ (respectively $\boldsymbol{z}_B$) are not jointly independent anymore. One may note that this constraint on the choice of the neighbourhood is not necessary in order to apply SA or graph-cut methods. Nevertheless, considering only the first-order neighbours is a good tradeoff between the computation time and modelling a Markovian field [HKK+97].

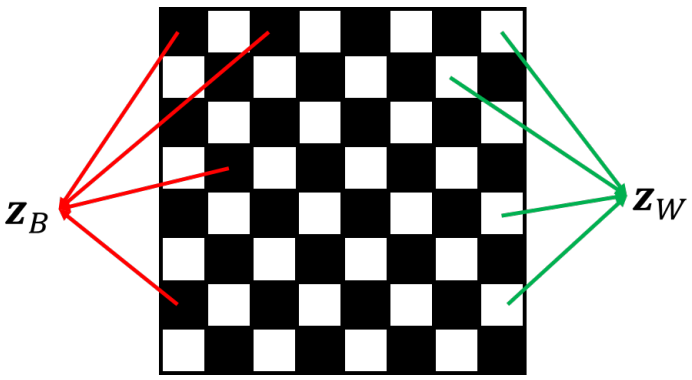


Figure 4.3: Division like a chessboard in "white" and "black" voxels, in order to perform the segmentation in JMAP algorithm by ICM. The division is illustrated in 2D. The figure is inspired from [Aya10, chap. 4.5.2].

Due to its appealing properties of parallelization, we choose to apply ICM algorithm in order to update the labels in step (4.25c) of JMAP algorithm. We denote by n_{seg} the number of iterations of ICM performed at each global iteration of JMAP. Like n_{vol} for the estimation of the volume, this number has to be fixed by the user.

4.3.4 Estimation of the means of the classes

Like for the variances $\boldsymbol{v}_\zeta$ of the uncertainties, the conditional distribution of the means $\boldsymbol{m}$ of the classes is fully factorized

$$\begin{aligned}
&p(\boldsymbol{m}|\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{v};\boldsymbol{g},\mathcal{U},\mathcal{M}) \propto p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{v})p(\boldsymbol{m}|m_0,v_0) \\
&\propto \exp\left[-\frac{1}{2}\left(\sum_{j=1}^{N}\sum_{k=1}^{K}\frac{(f_j-m_k)^2}{v_k}\delta(z_j-k)+\sum_{k=1}^{K}\frac{(m_k-m_0)^2}{v_0}\right)\right] \\
&\propto \exp\left[-\frac{1}{2}\sum_{k=1}^{K}\left(\left[\frac{1}{v_0}+\frac{1}{v_k}\sum_{j=1}^{N}\delta(z_j-k)\right]m_k^2-2m_k\left[\frac{m_0}{v_0}+\frac{1}{v_k}\sum_{j=1}^{N}f_j\delta(z_j-k)\right]\right)\right] \\
&\propto \exp\left[-\frac{1}{2}\sum_{k=1}^{K}\left(\left[\frac{1}{v_0}+\frac{N_k}{v_k}\right]m_k^2-2m_k\left[\frac{m_0}{v_0}+\frac{1}{v_k}\sum_{j\in\mathcal{R}_k}f_j\right]\right)\right]
\end{aligned} \tag{4.44}$$

where $\mathcal{R}_k$ is the set of voxels in class k, and $N_k = |\mathcal{R}_k|$ is the number of voxels in $\mathcal{R}_k$. Hence, the conditional distribution of m_k is a Gaussian

$$p(m_k|\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{v};\boldsymbol{g},\mathcal{U},\mathcal{M}) = \mathcal{N}(\hat{m}_{0_k},\hat{v}_{0_k}) \tag{4.45}$$

of which the mean $\hat{m}_{0_k}$ and the variance $\hat{v}_{0_k}$ are

$$\left\{\begin{array}{l}\hat{v}_{0_k} = \left(\frac{1}{v_0}+\frac{N_k}{v_k}\right)^{-1} \\ \hat{m}_{0_k} = \hat{v}_{0_k}\left(\frac{m_0}{v_0}+\frac{1}{v_k}\sum_{j\in\mathcal{R}_k}f_j\right)\end{array}\right. , \forall k \in \{1,\ldots,K\}\,. \tag{4.46}$$

A Gaussian distribution is maximized at its mean, so step (4.25d) of JMAP algorithm sets

$$\hat{m}_k = \hat{m}_{0_k}, \forall k \in \{1,\ldots,K\}\,. \tag{4.47}$$

4.3.5 Estimation of the variances of the classes

The conditional distribution of the variances $\boldsymbol{v}$ of the classes is also fully factorized

$$\begin{aligned}
&p(\boldsymbol{v}|\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{m};\boldsymbol{g},\mathcal{U},\mathcal{M}) \propto p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{v})p(\boldsymbol{v}|\alpha_0,\beta_0) \\
&\propto \exp\left[-\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\frac{(f_j-m_k)^2}{v_k}\right)\delta(z_j-k)-\sum_{k=1}^{K}\left((\alpha_0+1)\ln(v_k)-\frac{\beta_0}{v_k}\right)\right].
\end{aligned} \tag{4.48}$$

Hence, the conditional distribution of v_k is an Inverse-Gamma distribution

$$p(v_k|\boldsymbol{f},\boldsymbol{v}_\zeta,\boldsymbol{z},\boldsymbol{m};\boldsymbol{g},\mathcal{U},\mathcal{M}) = \mathcal{IG}\left(v_k|\hat{\alpha}_{0_k},\hat{\beta}_{0_k}\right) \tag{4.49}$$

of which the shape parameter $\hat{\alpha}_{0_k}$ and the intensity parameter $\hat{\beta}_{0_k}$ are

$$\left\{\begin{array}{l}\hat{\alpha}_{0_k} = \alpha_0 + \frac{N_k}{2} \\ \hat{\beta}_{0_k} = \beta_0 + \frac{1}{2}\sum_{j\in\mathcal{R}_k}(f_j-m_k)^2\end{array}\right. , \forall k \in \{1,\ldots,K\}\,. \tag{4.50}$$

The maximum is reached at

$$\hat{v}_k = \frac{\hat{\beta}_{0_k}}{\hat{\alpha}_{0_k}+1}, \tag{4.51}$$

which is the update formula for v_k, $\forall k \in \{1,\ldots,K\}$, in step (4.25e) of the algorithm.

4.3.6 Summary of the algorithm and fixation of the parameters

The proposed joint reconstruction and segmentation algorithm is summarized in algorithm 12. Due to the fact that the joint posterior distribution (4.17) is multimodal, the algorithm is a local optimization algorithm. In addition, we may underline that it has no convergence guarantee. As a consequence, the initialization is of great importance. The volume $\boldsymbol{f}^{(0)}$ required for the initialization is provided by a FBP method such as FDK [FDK84]. From this initial reconstruction, the initial variances $\boldsymbol{v}_\zeta^{(0)}$ of the uncertainties are computed using the update formula (4.38) of the algorithm. The initial volume is segmented in order to obtain the initial labels $\boldsymbol{z}^{(0)}$. Based on this initial segmentation, the initial means $\boldsymbol{m}^{(0)}$ and variances $\boldsymbol{v}^{(0)}$ of the classes are computed.

Algorithm 12 JMAP algorithm with Gauss-Markov-Potts prior

Initialize $\boldsymbol{f}^{(0)}, \boldsymbol{v}_\zeta^{(0)}, \boldsymbol{z}^{(0)}, \boldsymbol{m}^{(0)}, \boldsymbol{v}^{(0)}$
for $t = 0, \ldots, t_{max} - 1$ **do**
  1 : Update $\boldsymbol{f}$ by minimizing criterion (4.28) through n_{vol} iterations of gradient descent (4.30)
  2 : Update $\boldsymbol{v}_\zeta$ by (4.38)
  3 : Update $\boldsymbol{z}$ by maximizing Potts energy (4.40) through n_{seg} iterations of ICM algorithm [Bes86]
  4 : Update $\boldsymbol{m}$ by (4.47)
  5 : Update $\boldsymbol{v}$ by (4.51)
end for

Since we apply a local optimization algorithm to update the labels in algorithm 12, the initial segmentation has to be done very carefully. In our experiments, we have used histogram-based thresholding methods which are Otsu's method [Ots79] and a peak-picking method detailed in [KSH01]. These multi-thresholding methods are suited to get a good initial segmentation since they can handle classes with different variances. This is not the case for other algorithms such as K-means [Mac67], which we do not recommend to use. Originally developed for $K = 2$ classes, Otsu's method computes an optimal threshold by maximizing the between-class variance [Ots79]. For $K > 2$, we rely on MATLAB implementation, in which the thresholds are calculated by a locally convergent search-based optimization. Contrary to Otsu's method, the peak-picking algorithm given in [KSH01] is designed for multi-thresolding. This algorithm is an adaptation of a non-parametric clustering method [KNF76]. For each bin in the histogram, we seek the nearest peak in its neighbourhood : this peak is called the parent-bin of the current bin, which is called a child-bin of the peak. Each parent-bin and its children are seen as a class, i.e. the voxels in the children-bins of parent-bin k are assigned the initial class k. When we know the number of classes, each parent-bin is assigned as the child-bin of a higher parent-bin in its neighbourhood until there are only K parents-bins.

The parameters of the proposed algorithm are quite numerous : K, $\boldsymbol{\alpha}$, γ_0, m_0, v_0, α_0, β_0, α_{ζ_0} and β_{ζ_0}. We fix the number of classes K to its theoretical value which is known in industry. The initialization can be used to fix several of the other parameters. In the initial segmentation, we compute the number of voxels $N_k^{(0)}$ in each class k. We fix the singletons' energies to

$$\alpha_k = \ln\left(\frac{N_k^{(0)}}{N}\right), \forall k \in \{1, \ldots, K\}, \tag{4.52}$$

which complies with constraint (4.6). Hence, the a priori probability for a voxel to be in class k is

$$p_k = \frac{N_k^{(0)}}{N}. \tag{4.53}$$

For the prior on the means of the classes, parameter m_0 is fixed to the approximate median of the gray values, computed based on the initialization :

$$m_0 = \frac{1}{2}\left(\max_j f_j^{(0)} + \min_j f_j^{(0)}\right). \tag{4.54}$$

Given this value, parameter v_0 is fixed sufficiently large such that means of the classes m_k can take any value in a large interval containing the set of possible gray values.

The Potts coefficient is fixed such that the regions in the volume are compact. Hence, as shown in section 4.2, it has to be fixed greater than its critical value γ_c. The calculation of γ_c requires to know the partition function of the labels, i.e. the normalizing constant of (4.7) :

$$Z(\boldsymbol{\alpha}, \gamma_0) = \sum_{\boldsymbol{z}' \in \{1,\dots,N\}^K} \sum_{j=1}^{N} \left(\sum_{k=1}^{K} \alpha_k \delta(z'_j - k) + \gamma_0 \sum_{i \in \mathcal{V}(j)} \delta(z'_j - z'_i) \right), \tag{4.55}$$

This partition function has only an analytical expression in 2D for the Ising field case for which $K = 2$, which enables to estimate the value of γ_c [Gio10]. In the other cases, the partition function is untractable [MDZ97]. Hence, the critical value γ_c cannot be computed in general, whatever the dimension and the value of K are. In 2D, works have been done in order to estimate an optimal value of γ_0 jointly with the labels, without dealing with the partition function of $\boldsymbol{z}$ [PDBT13, SSHF16]. Unfortunately, these strategies are computationally costly because they are based on MCMC methods [PDBT13], or not parallelizable [SSHF16]. As a result, they cannot be used for our 3D application. For this reason, we fix γ_0 to an optimal value which is found empirically.

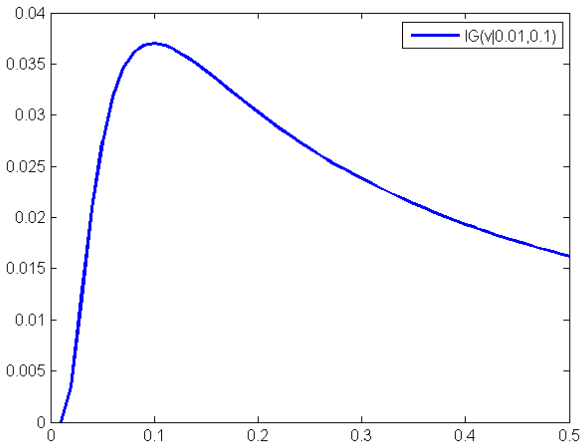
Concerning parameters (α_0, β_0) for the Inverse-Gamma prior on the variances of the classes, since we want to reconstruct (quasi-)piecewise-constant objects, they are fixed in order to encourage small v_k, $1 \leqslant k \leqslant K$. Figure 4.4 shows the probability density function of Inverse-Gamma distribution for different values of its parameters. We see that, for $\alpha_0 > 1$ and $\beta_0 < 1$, small variances are modeled as very likely and high variances as very unlikely, which is exactly what we want. For $\alpha_0 < \beta_0 < 1$, high variances are modeled as very likely, which does not correspond to our piecewise-constant prior. For $\beta_0 < \alpha_0 < 1$, small variances are more likely than high variances, but the probability for high variances is not zero as it is the case for $\alpha_0 > 1$ and $\beta_0 < 1$. Fixing $\alpha_0 > \beta_0 > 1$ leads to promote the variances in an interval which does not contain zero. At last, fixing $\alpha_0 < 1$ and $\beta_0 > 1$ or $\beta_0 > \alpha_0 > 1$ does not correspond at all to what we want, since high variances are modeled as very likely and small variances as very unlikely. As a conclusion, it appears from figure 4.4 that fixing $\alpha_0 > 1$ and $\beta_0 < 1$ is the optimal choice for reconstructing quasi-piecewise-constant volumes.

The route to fix α_0 and β_0 cannot be the same for parameters α_{ζ_0} and β_{ζ_0} related to the variances of the uncertainties on the projections. Indeed, since the derivation of the forward model (4.1) does not take into account beam-hardening and scattering, we cannot have a particular prior on how large variances $\boldsymbol{v}_\zeta$ are. As a result, these parameters are fixed empirically based on the different plots presented in figure 4.4. If we know that high scattering and beam-hardening are present, then it is preferable to model high variances $\boldsymbol{v}_\zeta$ as likely. If, on the contrary, there is only few scattering, then we can constrain $v_{\zeta_i}, \forall i$, to be in an interval of small variances, as it is the case in plot (e) of figure 4.4, with $\alpha_{\zeta_0} > \beta_{\zeta_0} > 1$. Another possibility is, if the noise is very small, to encourage very small $v_{\zeta_i}, \forall i$, by fixing $\alpha_{\zeta_0} > 1$ and $\beta_{\zeta_0} < 1$, as it is the case in plot (c) of figure 4.4.

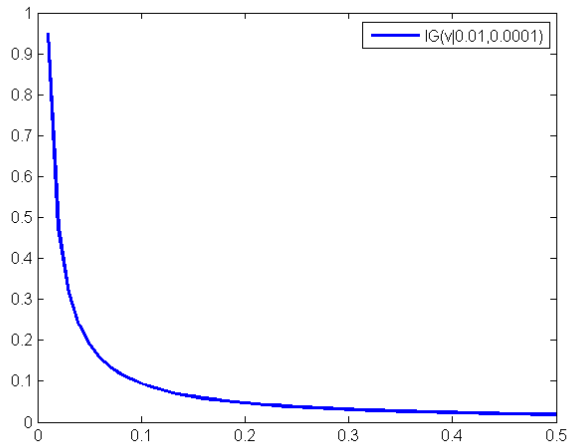
Other parameters which are hidden are the numbers of subiterations for the estimation of the volume and the labels, n_{vol} and n_{seg}. These parameters are tuned experimentally. When many beam-hardening and scattering effects are present, it is preferable to fix n_{vol} and n_{seg} small. With small n_{seg}, we avoid that scattering and beam-hardening effects contaminate the segmentation because of the compactness constraint enforced by γ_0. Similarly, with small n_{vol}, we avoid to over-trust the data $\boldsymbol{g}$ which are very far from the ideal noiseless projections $\boldsymbol{Hf}$ due to high absorption and scattering.

4.4 Simulation results

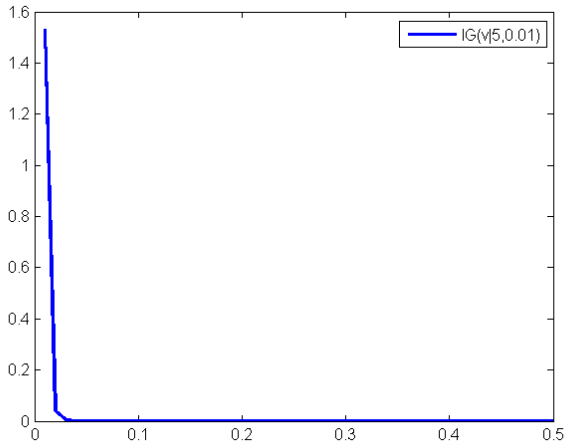
We first test our method in simulation on the Shepp-Logan phantom with the same geometrical configuration used in section 3.3. The 64 used projections are noisy with a SNR of 20 db. Since the phantom is exactly piecewise-constant, we can compute the actual number of classes which is, as shown in figure 4.5, $K = 5$. The values for the parameters of JMAP are given in table 4.1 and fixed according to the strategies explained in section 4.3.6. We run JMAP during 20 iterations. In each of these iterations, the maximum number of subiterations for the estimation of the volume is fixed to $n_{vol} = 20$, while the one for ICM in order to estimate the labels is fixed to $n_{seg} = 10$. The gradient descent can be stopped sooner if criterion (4.28) does not change anymore. Similarly, ICM can be stopped before having performed n_{seg} iterations if Potts energy does not change significantly. In practice, after few global iterations, the actual number of performed iterations for ICM is 2 or 3.



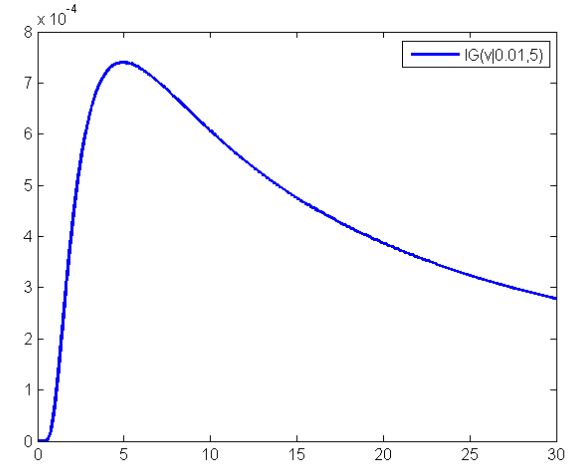
(a) $\alpha_0 < 1, \beta_0 < 1, \alpha_0 < \beta_0$



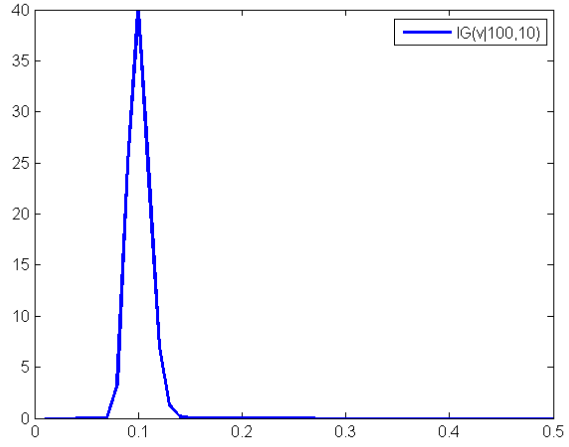
(b) $\alpha_0 < 1, \beta_0 < 1, \alpha_0 > \beta_0$



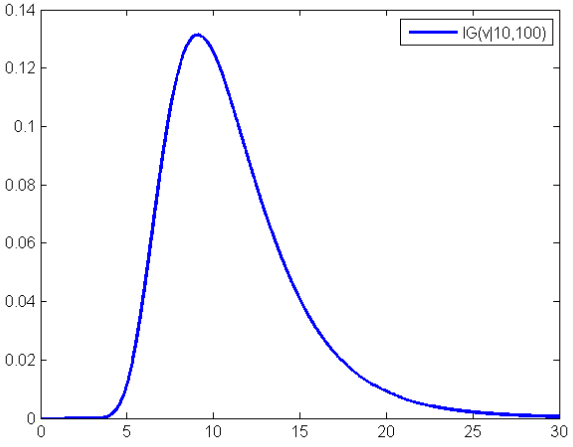
(c) $\alpha_0 > 1, \beta_0 < 1$



(d) $\alpha_0 < 1, \beta_0 > 1$



(e) $\alpha_0 > 1, \beta_0 > 1, \alpha_0 > \beta_0$



(f) $\alpha_0 > 1, \beta_0 > 1, \alpha_0 < \beta_0$

Figure 4.4: Probability density function of $\mathcal{IG}(.|\alpha_0, \beta_0)$ for several values of (α_0, β_0)

Parameters	K	γ_0	v_0	α_0	β_0	α_{ζ_0}	β_{ζ_0}	n_{vol}	n_{seg}
Fixed values	5	6	1	5	0.01	200	1	20	10

Table 4.1: Parameters of JMAP to reconstruct Shepp-Logan phantom

We compare the uses in JMAP of the unmatched RD/VD pair of projector and backprojector (P/BP) and of the matched SF pair presented in chapter 3. If the used P/BP pair is the SF pair, then the expression (4.32) of the stepsize in the gradient descent for the estimation of the volume can be simplified

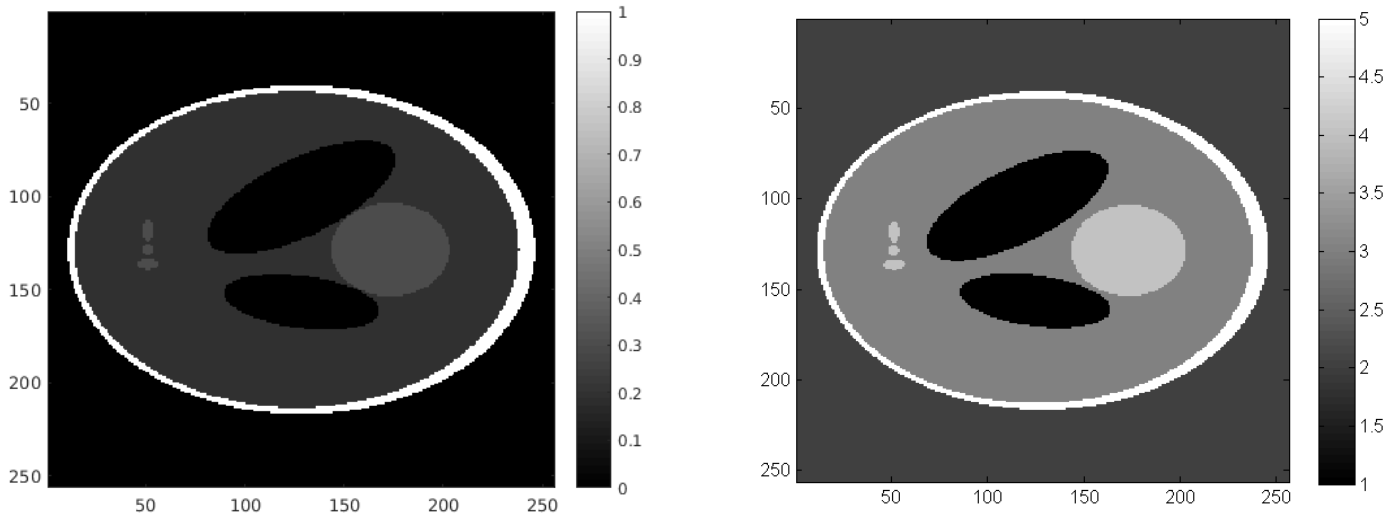
$$\mu_n = \arg\min_{\mu>0} J_f\left(\boldsymbol{f}^{(n+1)}\right) = \frac{\|\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|_2^2}{\|\boldsymbol{H}\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|_{\boldsymbol{V}_\zeta}^2 + \|\nabla J_f\left(\boldsymbol{f}^{(n)}\right)\|_{\boldsymbol{V}_{\boldsymbol{z}}}^2}, \tag{4.56}$$

since the pair is matched. On the contrary, if the used P/BP pair is unmatched such as the RD/VD pair, we have observed in our experiments that applying formula (4.56) makes the algorithm diverge. In this case, we have found that it is preferable to apply formula (4.34) in order to compute the optimal stepsize. Compared to the use of formula (4.56), this adds a backprojection operation at each iteration of the descent. Since one iteration of the descent is performed $n_{vol} \times t_{max}$ times in JMAP, the number of saved backprojection operations when the used P/BP pair is matched is high. This is well shown in table 4.2. With the matched SF pair, one iteration of the gradient descent only requires one backprojection operation, while two backprojections have to be done with the unmatched pair. Hence, if the gradient descent has $n_{vol} = 20$ iterations as it is the case in our tests, then we have 20 backprojections with the matched pair and 40 with the unmatched one. For both pairs, 40 projections are required in each gradient descent, as one projection to update the variances of the uncertainties by formula (4.38). In total, if we run JMAP algorithm during 20 global iterations, we need to perform 820 projections whatever the used pair is, but 400 backprojections instead of 800 if the used P/BP pair is matched. Hence, JMAP algorithm requires to perform 1620 projection and backprojection operations with an unmatched pair, and only 1220 with a matched one, which corresponds to a saving of 25 % of P/BP operations.

Used P/BP pair		Unmatched RD/VD	Matched SF
One iteration of gradient descent to update the volume	Projections	2	2
	Backprojections	2	1
Update the volume in $n_{vol} = 20$ iterations	Projections	40	40
	Backprojections	40	20
Update the variances $\boldsymbol{v}_\zeta$ of the uncertainties by (4.38)	Projections	1	1
	Backprojections	0	0
$t_{max} = 20$ iterations of JMAP algorithm	Projections	820	820
	Backprojections	800	400
Total number of projection and backprojection operations in JMAP algorithm		1620	1220

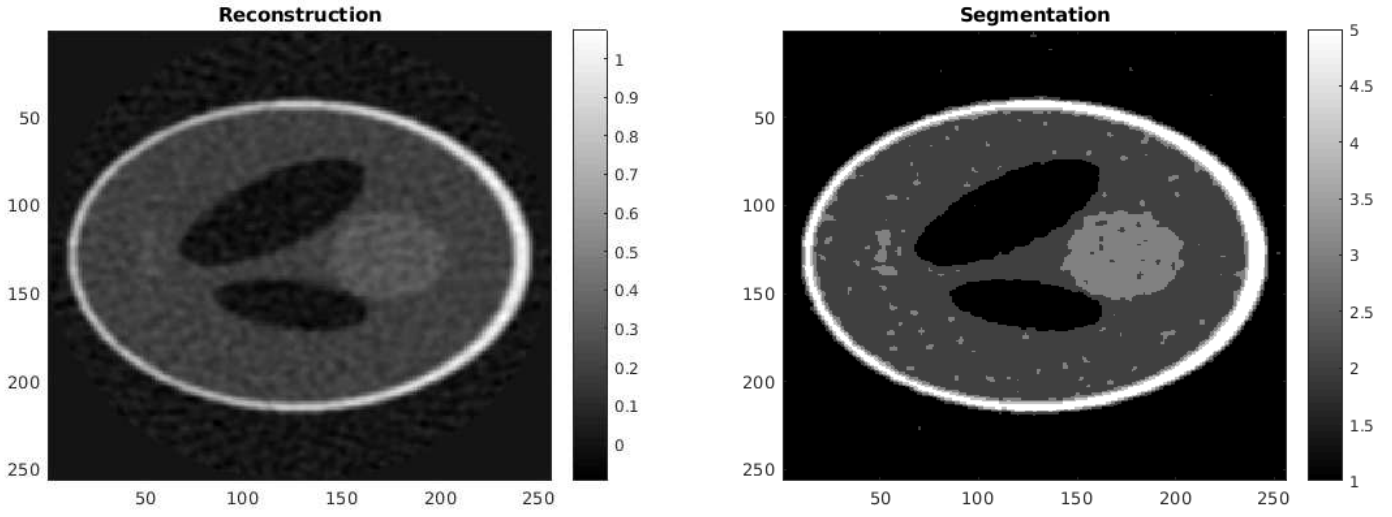
Table 4.2: Comparison of the computational cost of JMAP in terms of projection and backprojection operations, depending on whether the used P/BP pair is matched or unmatched

Figures 4.7 and 4.8 show the joint reconstructions and segmentations of Shepp-Logan phantom obtained by JMAP, with the RD/VD pair and the SF pair respectively. As expected, the obtained reconstructions are piecewise-constant, with compact and homogeneous regions. The reconstructions are near the original phantom, as shown in table 4.3, with a relative error $\Delta_2 \boldsymbol{f}$ equal to 8.82 % with the unmatched RD/VD pair and 8.81 % with the matched SF pair. The reconstructions obtained with the different pairs appear very similar, as the segmentations jointly returned by the algorithm. These segmentations look very near the original one in figure 4.5. In table 4.3, the similarity between the original and the estimated segmentations is measured by the Rand index [Ran71] : it is approximately 95 % whatever the used P/BP is. Moreover, the convergence of the algorithm, shown in figure 4.9, does not seem to be impacted by the mismatching of the projection and backprojection operators, but, once again, this is because we have applied



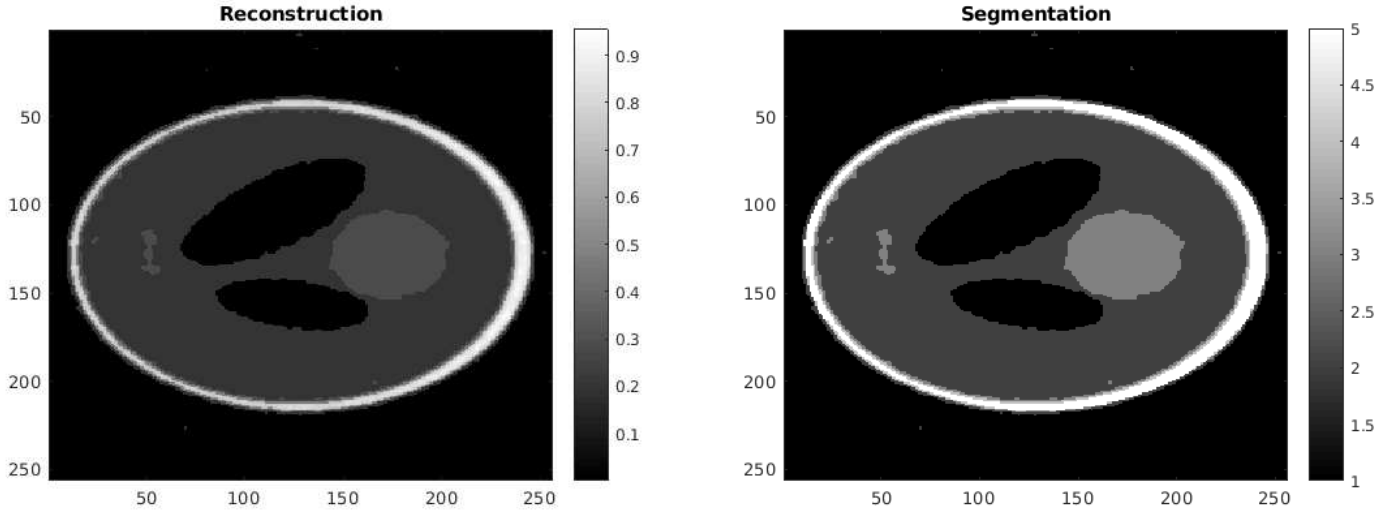
(a) (b)

Figure 4.5: Shepp-Logan phantom (a) and its segmentation (b). We see that it contains $K = 5$ classes.



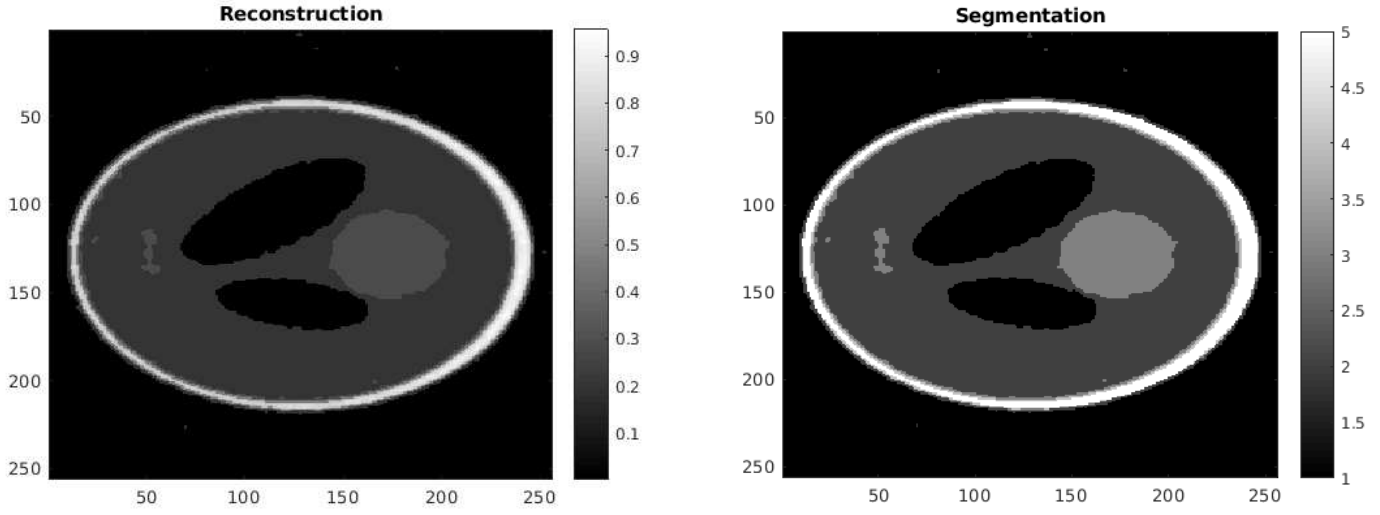
(a) (b)

Figure 4.6: FDK reconstruction of Shepp-Logan phantom (a) and its segmentation (b) used as initialization of JMAP



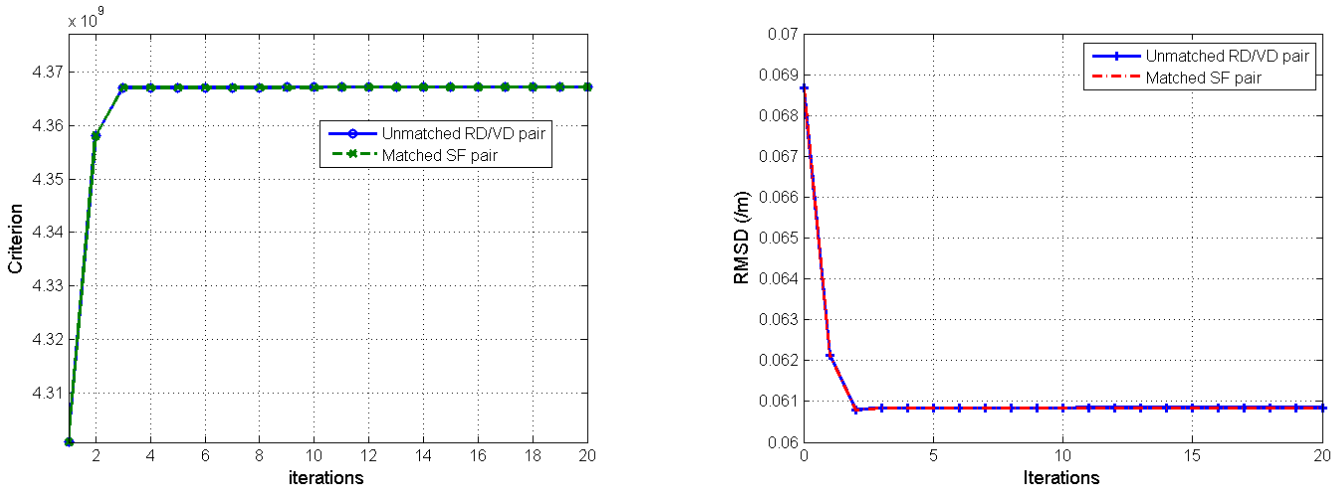
(a) (b)

Figure 4.7: Joint reconstruction (a) and segmentation (b) of Shepp-Logan phantom obtained by JMAP with the unmatched RD/VD pair



(a) (b)

Figure 4.8: Joint reconstruction (a) and segmentation (b) of Shepp-Logan phantom obtained by JMAP with the matched SF pair



(a) (b)

Figure 4.9: Convergence metrics for JMAP (stopping criterion (a) and RMSD with respect to the real phantom (b)), with the unmatched RD/VD pair and the matched SF pair

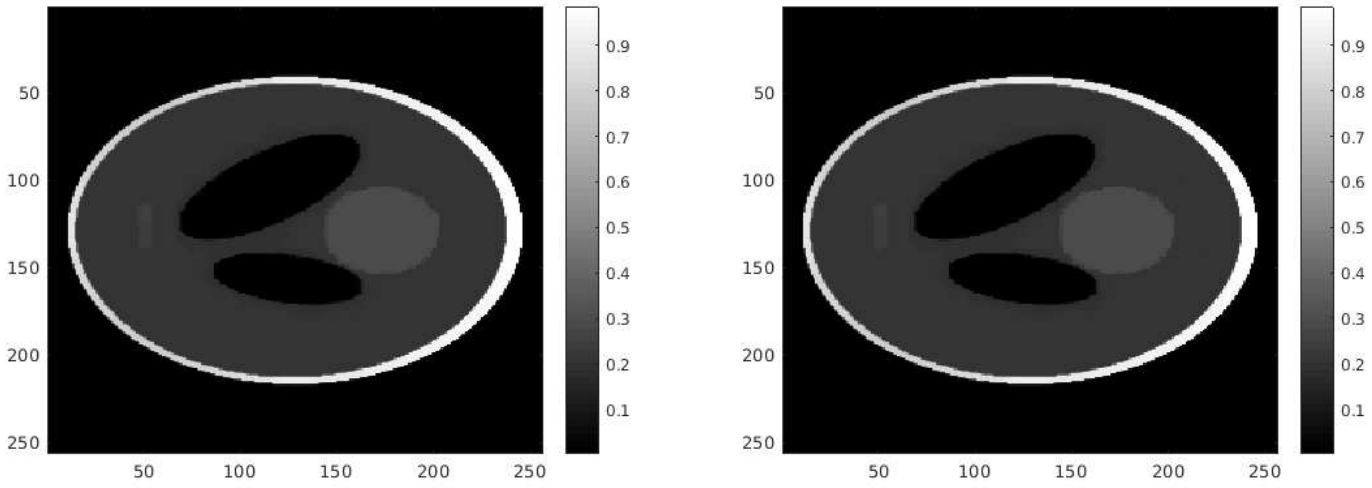
formula (4.32) instead of (4.56) for stepsize μ_n when the unmatched pair was used. In table 4.3, the computation time with the matched SF pair is larger than the one with the RD/VD pair, since the SF model is more accurate.

Reconstruction algorithm	Computation time	$\Delta_2 \boldsymbol{f}$	$\Delta_2 \boldsymbol{g}$	Rand index
PDFW with the unmatched RD/VD pair	**168.5 s**	3.93 %	0.76 %	/
PDFW with the matched SF pair	510.5 s	**3.90 %**	0.73 %	/
JMAP with the unmatched RD/VD pair	584.3 s	8.82 %	0.67 %	**94.7 %**
JMAP with the matched SF pair	873.1 s	8.81 %	**0.66 %**	**94.7 %**

Table 4.3: Comparison of the results of PDFW and JMAP, with the RD/VD pair and the SF pair

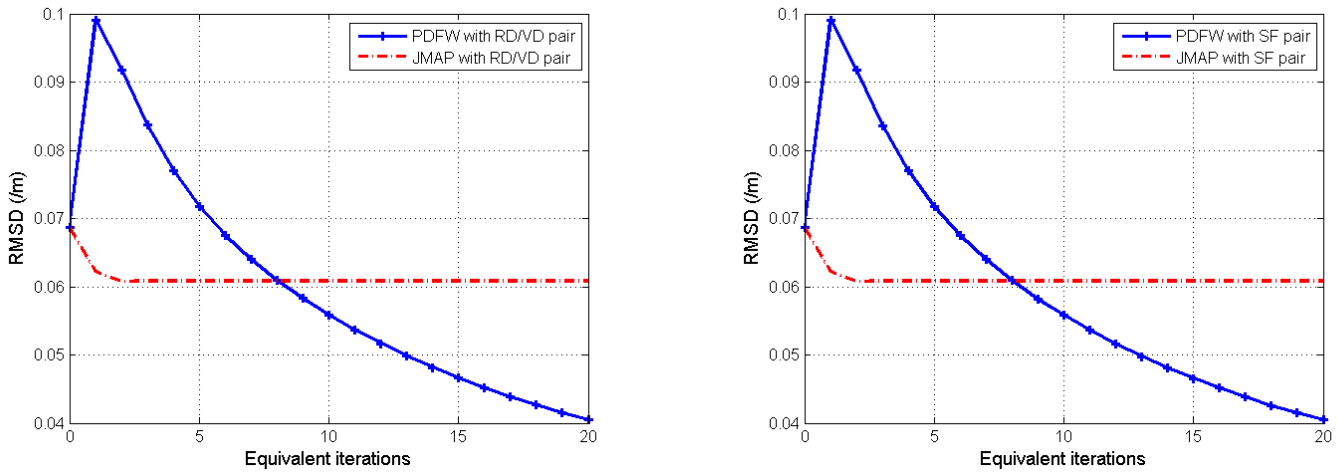
We compare our results with TV-regularized least-squares solved by PDFW [OMBF18] presented in chapter 2. The results obtained by this algorithm are re-shown in figure 4.10. In table 4.3, the computation time is much greater for JMAP than for PDFW. This is due to the fact that PDFW is mainly composed of projection and backprojection operations, while, in JMAP, the segmentation step is also computationally demanding. In this work, the segmentation step is done on the CPU and its parallelization on the GPU is a perspective. In figure 4.11, PDFW appears to converge to a solution which is a bit closer to the original phantom than the one obtained by JMAP. Nevertheless, this proximity is only in terms of the RMSD. If we focus on the properties of the estimation, then the solution obtained by JMAP appears closer to the original phantom. Indeed, if we look at a profile of the phantom as done in figure 4.12 and if we zoom in the central points, we see that JMAP complies with the piecewise-constant assumption much more than PDFW. This is not surprising, since TV-regularization in PDFW looks for a solution of which the gradient has minimum $\mathcal{L}_1$-norm, while the piecewise-constant assumption is enforced if we consider the $\mathcal{L}_0$-norm of the gradient [SWFU15, SRUW17]. Nevertheless, as pointed out in the introduction of this chapter, the method in [SWFU15, SRUW17] requires a lot of memory, contrary to PDFW [OMBF18]. In JMAP, there are two huge variables which need to be stored in addition to the volume. The first is the variances $\boldsymbol{v}_\zeta$ of the uncertainties which are the size of the projections, like the dual variable $\boldsymbol{p}$ in PDFW (see algorithm 7). The second huge variable in JMAP corresponds to the labels $\boldsymbol{z}$ which are the size of the volume but have the advantage to be integers : hence, they require less memory than the second dual variable in PDFW which is also the size of the volume (see algorithm 7).

As a conclusion of this section, our experiments in simulation have validated our reconstruction algorithm since reconstructions of good quality have been obtained. The assets of JMAP in terms of memory usage have been highlighted. The algorithm has also appeared as more computationally costly than other reconstruction algorithms such as PDFW due to its additional segmentation step. Nevertheless, JMAP has been shown to enforce the piecewise-



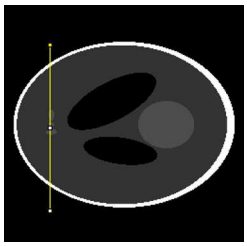
(a) (b)

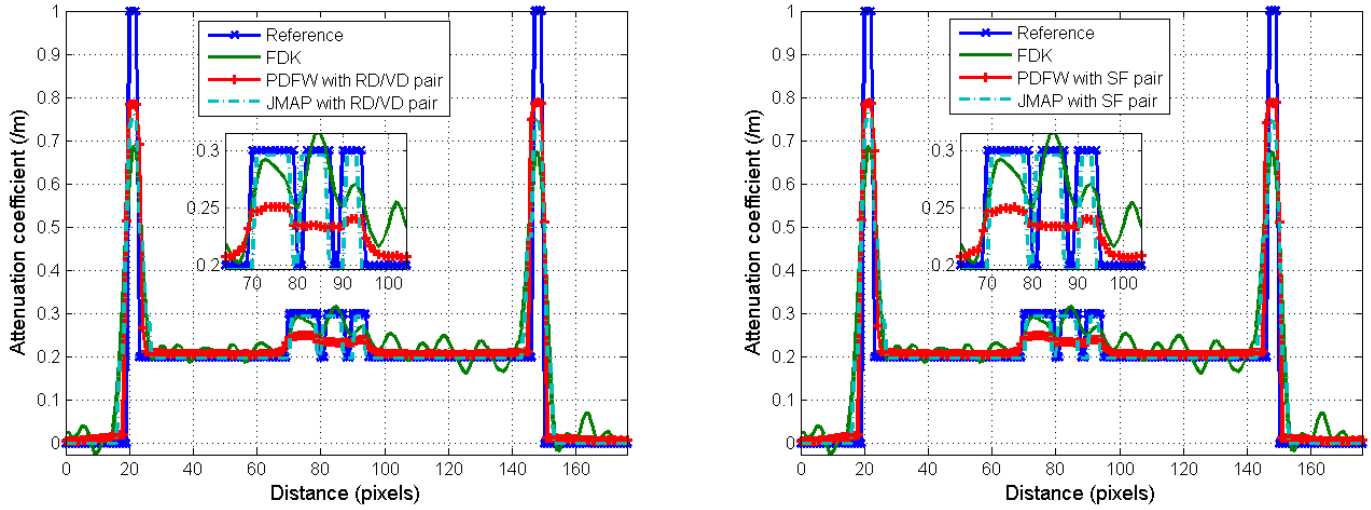
Figure 4.10: Reconstructions of Shepp-Logan phantom by PDFW [OMBF18] with the unmatched RD/VD pair (a) and the matched SF pair (b)



(a) (b)

Figure 4.11: Comparison of the evolution of the RMSD between JMAP and PDFW, with the unmatched RD/VD pair (a) and the matched SF pair (b)





(a) (b)

Figure 4.12: Profile of Shepp-Logan phantom to compare the reconstructions by PDFW and JMAP, with the unmatched RD/VD pair (a) and with the matched SF pair

constant prior better than PDFW. In the next section, we apply JMAP on real data of SAFRAN. Because our GPU implementation of the SF pair is still not completely operational and can for the moment only be used to reconstruct small volumes, and because, we do our experiments only with the RD/VD pair in the following. This use is reasonable since, thanks to the use of formula (4.32) instead of formula (4.56) for stepsize μ_n, we have not noticed troubles with this unmatched pair.

4.5 Results on real data

4.5.1 Evaluation of the accuracy : Image Quality Indicator (IQI) volume

We first evaluate the accuracy of the proposed method on an Image Quality Indicator (IQI) volume [GARC16]. The field-of-view is sampled into $512 \times 512 \times 256$ voxels. In order to reconstruct this volume by JMAP, 300 projections uniformly distributed over $[0, 2\pi]$ are used. The detector is sampled into 512×256 pixels. The source-to-rotation-center distance is 975 mm, and the source-to-detector distance is 1300 mm. The IQI volume is a cylindrical plate in which holes are drilled [GARC16]. It is obliquely positioned in order to minimize the attenuation of X-rays inside it. From 300 projections, the filtered backprojection [FDK84], which initializes JMAP, is shown in figure 4.13. The values for the parameters of JMAP are given in table 4.4. The volume is composed of nylon, polytetrafluoroethylene (PTFE), polyamide and air, hence we fix $K = 4$. We run JMAP during 50 iterations. Since, for this reconstruction, each iteration of JMAP contains $n_{vol} = 10$ subiterations to estimate the volume, we run PDFW with TV-regularization [OMBF18] during 500 iterations. The reconstruction by PDFW is shown in figure 4.14.

The joint reconstruction and segmentation by JMAP are shown in figure 4.15. The convergence of the stop criterion is given in figure 4.16. As pointed out in simulation, the computation time for JMAP is larger than the one for PDFW due to the segmentation step. Like the TV-reconstruction in figure 4.14, the reconstruction by JMAP has compact and homogeneous regions thanks to the use of Gauss-Markov-Potts prior model. To compare the accuracy

Parameters	K	γ_0	v_0	α_0	β_0	α_{ζ_0}	β_{ζ_0}	n_{vol}	n_{seg}
Fixed values	4	3	1	5	0.01	200	1	10	4

Table 4.4: Parameters of JMAP to reconstruct IQI volume

of each reconstruction, we focus on the little holes present in the volume, as shown in figure 4.17. For the two biggest holes, JMAP reaches the lowest value and is better than FDK and PDFW. Nevertheless, this value is slightly under the theoretical value of 0 due to the fact that, like in PDFW, no non-negativity constraint is enforced in JMAP. For the two intermediate holes, TV-reconstruction's profile goes lower than JMAP. For the two most little holes, the accuracy is better for JMAP. As noticed in simulation, the reconstruction by JMAP has sharper transitions than PDFW since it enforces a quasi-piecewise-constant assumption.

Reconstruction algorithm	Computation time
PDFW	**1180 s**
JMAP	3542 s

Table 4.5: Comparison of the computation time between PDFW and JMAP, for the reconstruction from 300 projections of IQI volume

4.5.2 Evaluation on Composite Material with Titanium on the Edges (CMTE)

We evaluate the robustness of our algorithm on challenging data with metal : the inspected volume is made of composite material with titanium on the edges. We denote it by CMTE (Composite Material with Titanium on the Edges). The field-of-view is sampled into 512^3 voxels. The reconstruction is performed with 300 projections uniformly distributed over $[0, 2\pi]$. The detector is sampled into 512^2 pixels. The source-to-rotation-center distance is 432 mm, and the source-to-detector distance is 807 mm. The reconstruction by filtered backprojection is shown in figure 4.18. As we see, the titanium is cause of many scattering and beam-hardening. The parameters for JMAP are given in table 4.6. The field-of-view is composed of air and of the materials of CMTE which are composite material and titanium : hence, we fix $K = 3$. Since there are strong beam-hardening and scattering, we fix $\alpha_{\zeta_0} < \beta_{\zeta_0} < 1$ as explained in section 4.3.6.

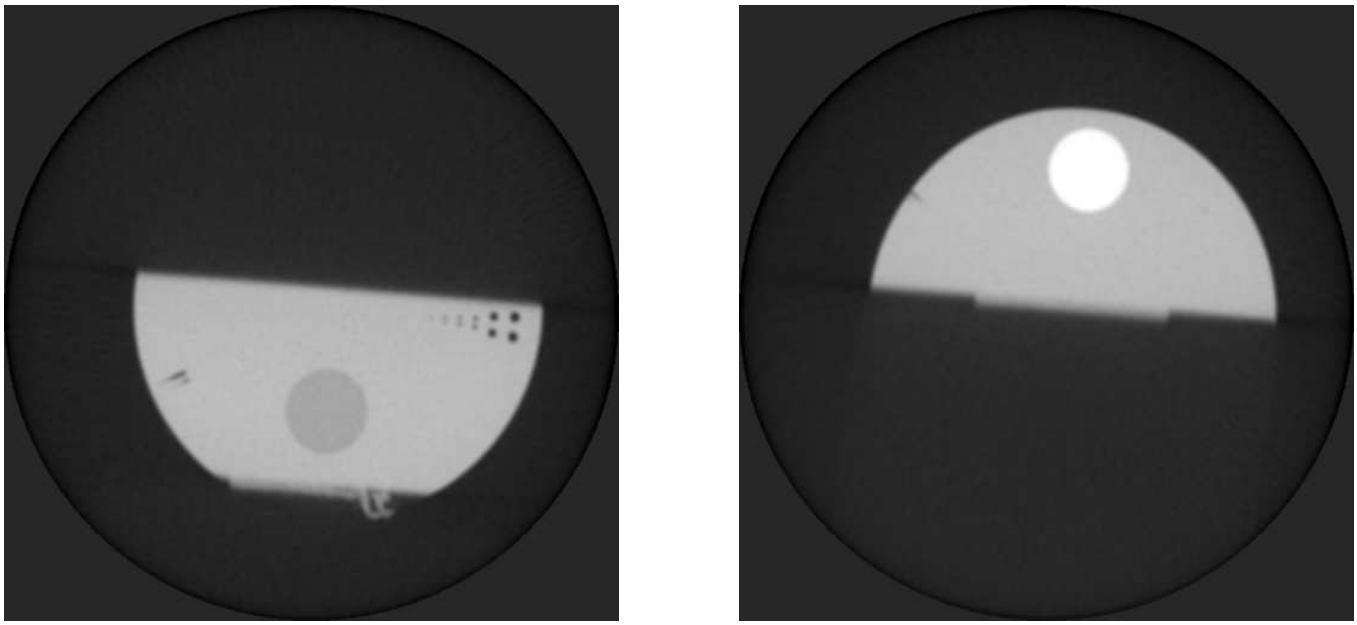
Parameters	K	γ_0	v_0	α_0	β_0	α_{ζ_0}	β_{ζ_0}	n_{vol}	n_{seg}
Fixed values	3	3	1	5	0.01	0.01	0.1	2	3

Table 4.6: Parameters of JMAP to reconstruct CMTE

JMAP is run during 50 iterations and the convergence of its criterion is shown in figure 4.21. The results of JMAP in figure 4.20 are compared with those of PDFW in figure 4.19. For PDFW, we fix $\lambda = 4$ and the weights are added to the data-matching term in order to take into account the pronounced asymmetry of the CMTE volume. The computation times for PDFW and JMAP are given in table 4.7.

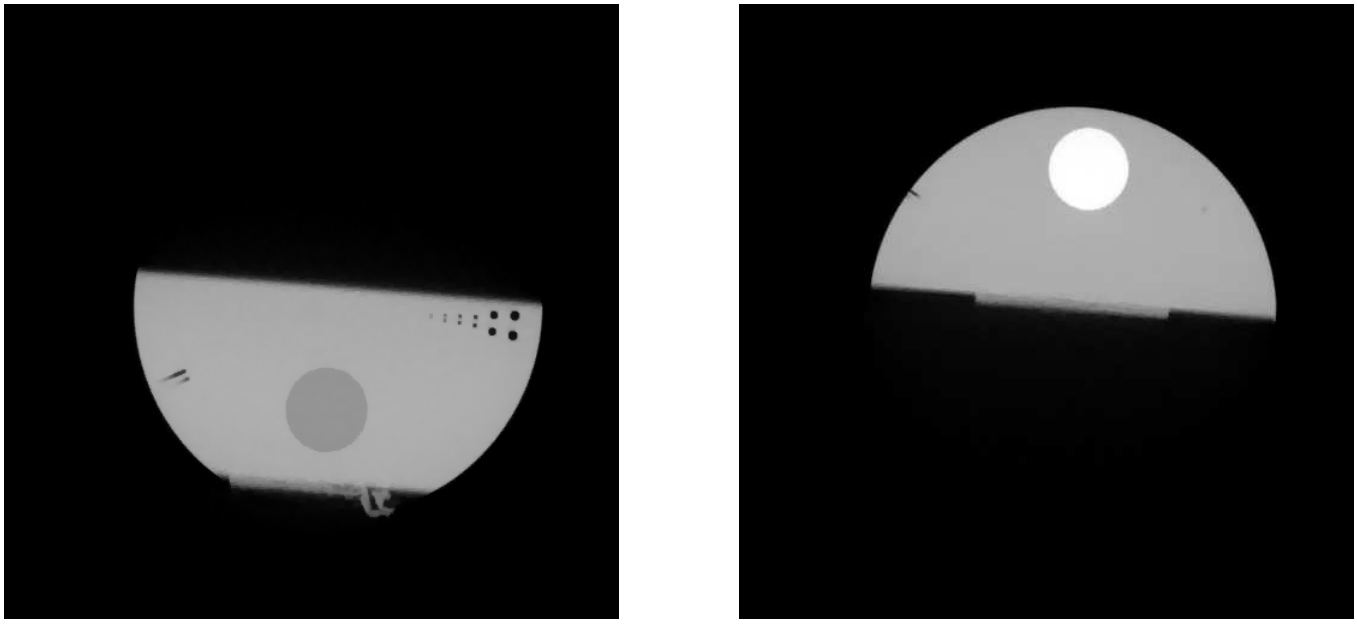
The scattering in CMTE has its values which are near those of the composite material. As a result, in figure 4.20, JMAP fails in removing this scattering since the segmentation step gives the same label to the scattering and to the composite material. On the contrary, TV-regularization succeeds in smoothing the scattering. Hence, the first profile of CMTE we observe in figure 4.22 shows that PDFW obtains sharper transitions than JMAP due to the removing of the scattering.

The beam-hardening in CMTE is characterized by strong negative values at the extremities of the titanium edges. These negative values, visible in the second profile of CMTE observed in figure 4.23 for FDK and PDFW reconstructions, can be very distinguished from the other materials during the segmentation step of JMAP. As a result, the beam-hardening is assigned the same label as the air. Due to the fact that, according to our Gauss-Markov-Potts prior, the values of the voxels of this class have to be close to their mean which is approximately zero, JMAP succeeds in completely erasing the beam-hardening, while artifacts are still present in the reconstruction by PDFW.



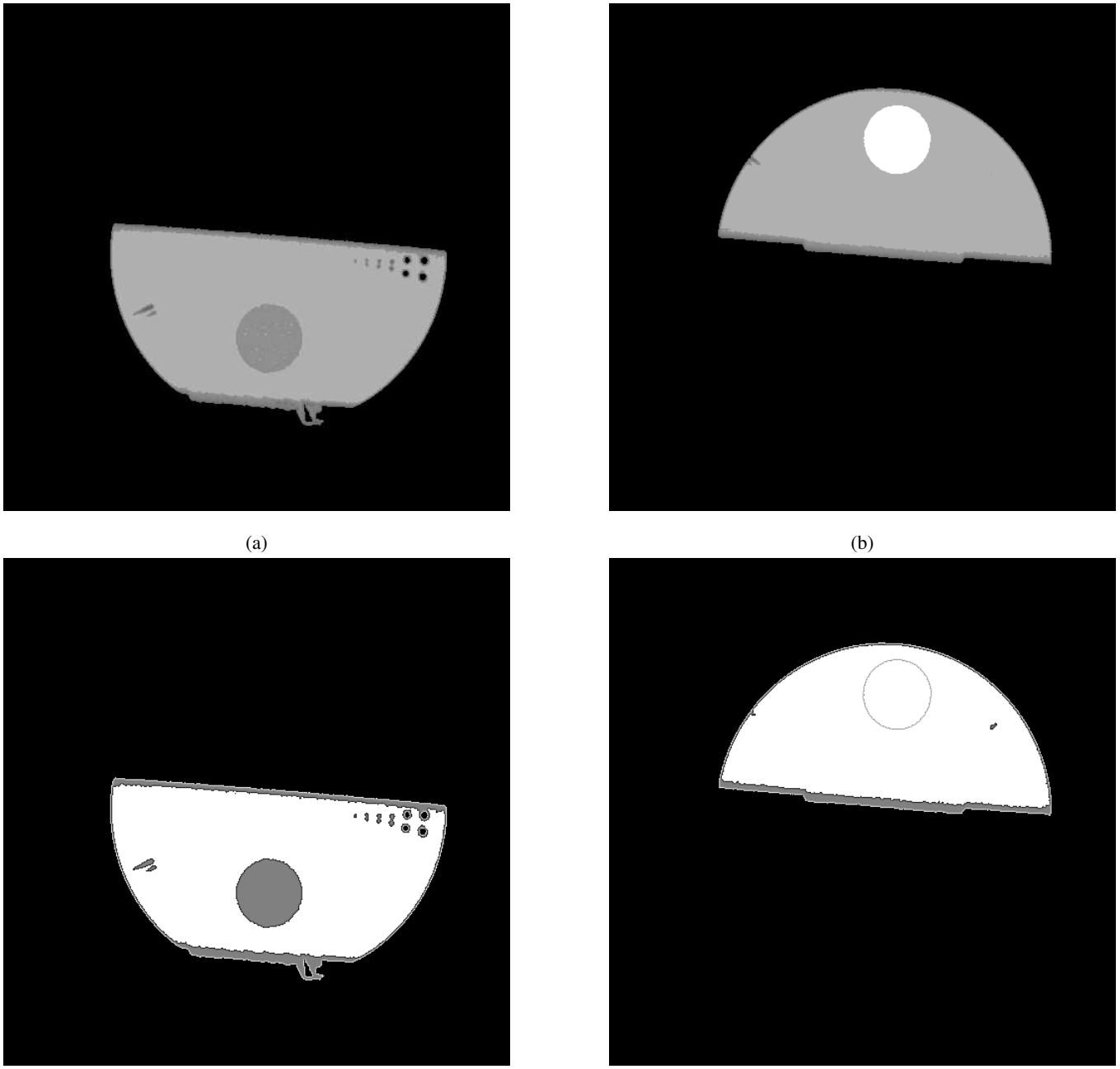
(a) (b)

Figure 4.13: Reconstruction from 300 projections of IQI volume by filtered backprojection [FDK84] (bottom (a) and top (b))



(a) (b)

Figure 4.14: Reconstruction from 300 projections of IQI volume by PDFW [OMBF18] (bottom (a) and top (b))



(a) (b)

(c) (d)

Figure 4.15: Joint reconstruction (bottom (a) and top (b)) and segmentation (bottom (c) and top (d)) of IQI volume from 300 projections by JMAP

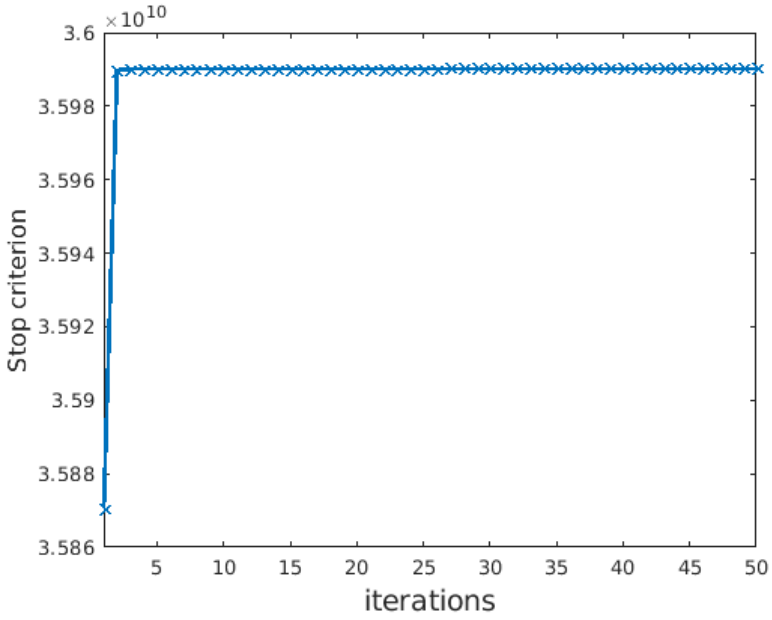


Figure 4.16: Convergence of JMAP for the joint reconstruction and segmentation of IQI volume from 300 projections

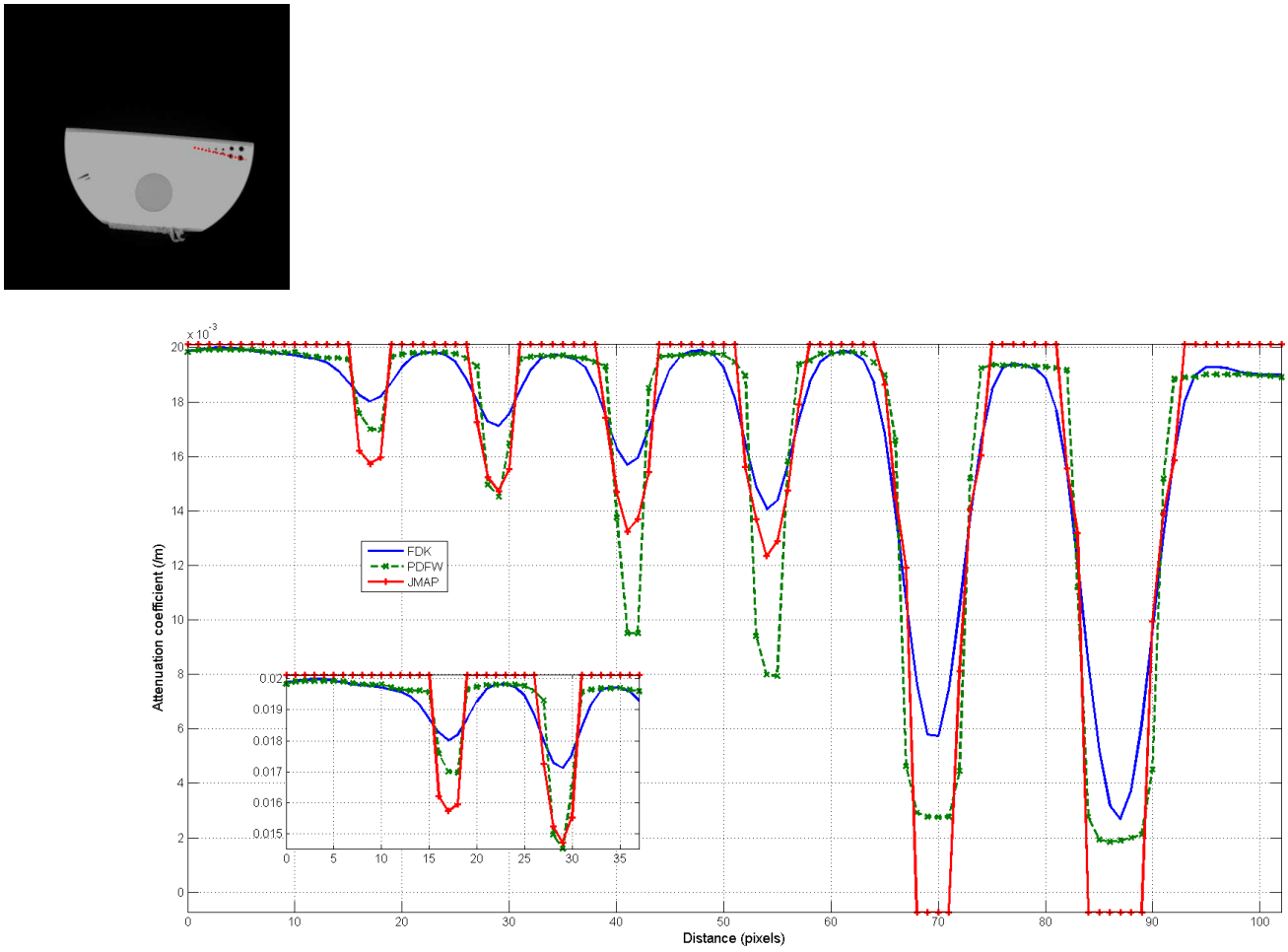


Figure 4.17: Profiles of the holes of IQI volume for the reconstructions by filtered backprojection [FDK84], PDFW [OMBF18] and JMAP

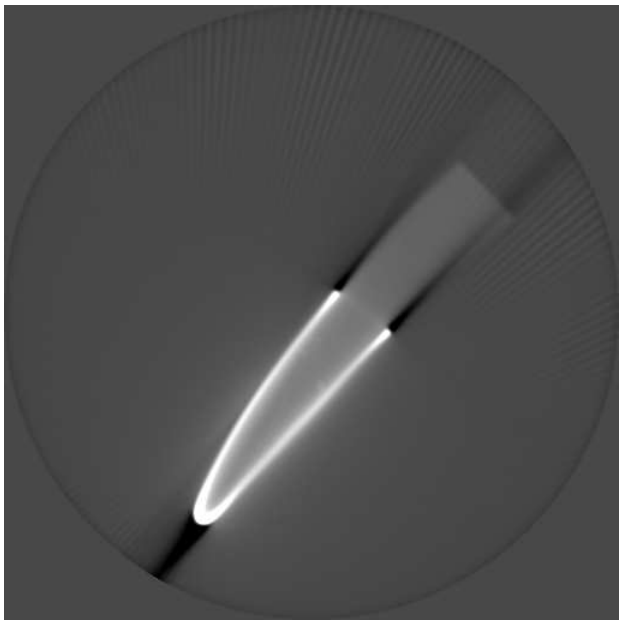
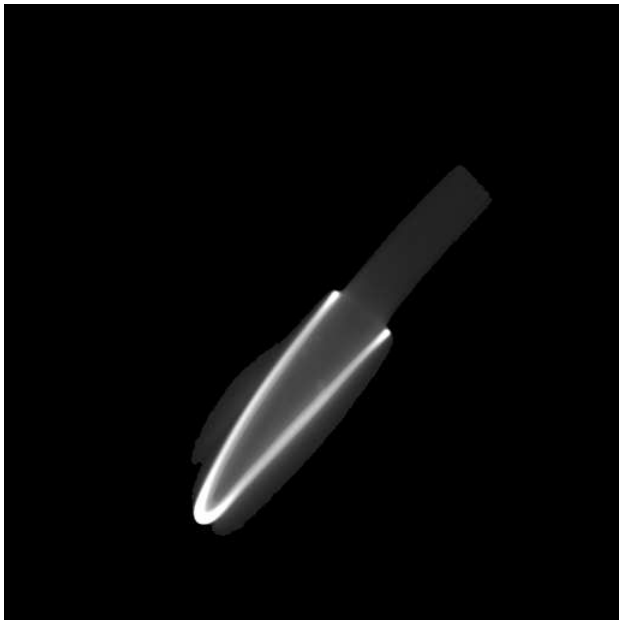
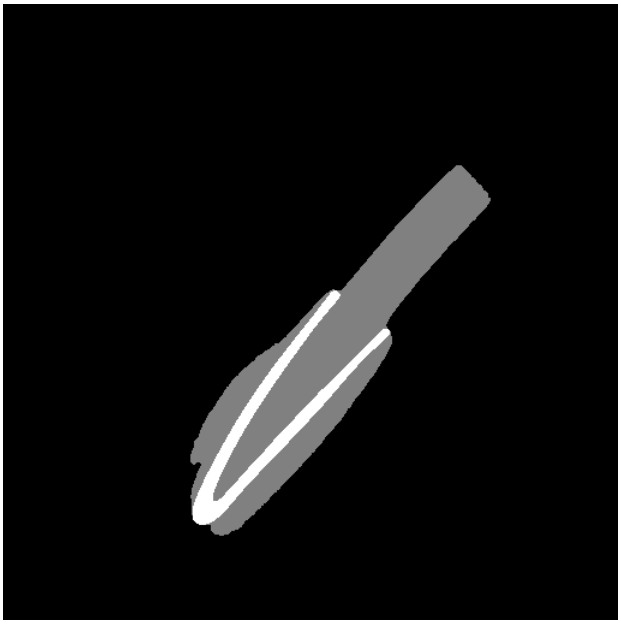


Figure 4.18: Reconstruction of CMTE from 300 projections by filtered backprojection [FDK84]

Figure 4.19: Reconstruction of CMTE from 300 projections by PDFW [OMBF18]



(a)



(b)

Figure 4.20: Joint reconstruction (a) and segmentation (b) of CMTE from 300 projections by JMAP

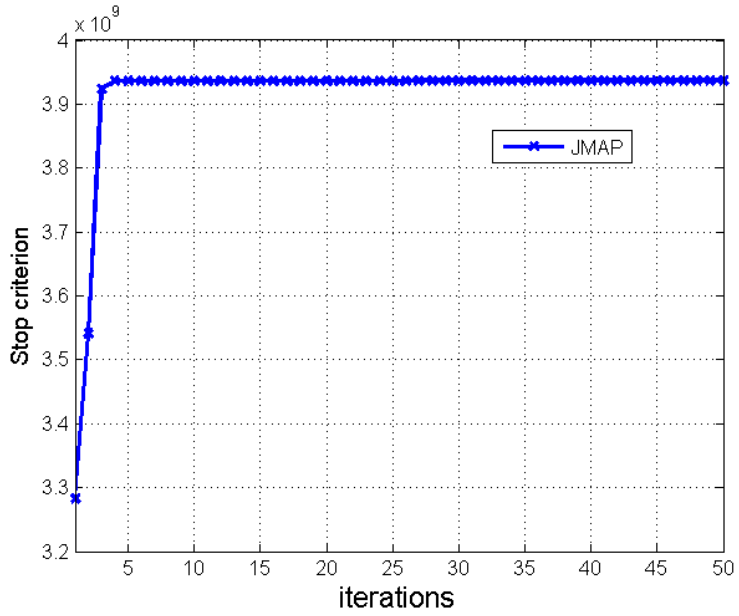
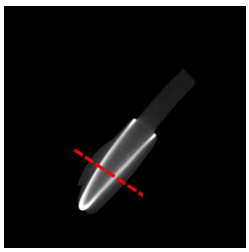


Figure 4.21: Convergence of JMAP for the joint reconstruction and segmentation of CMTE from 300 projections



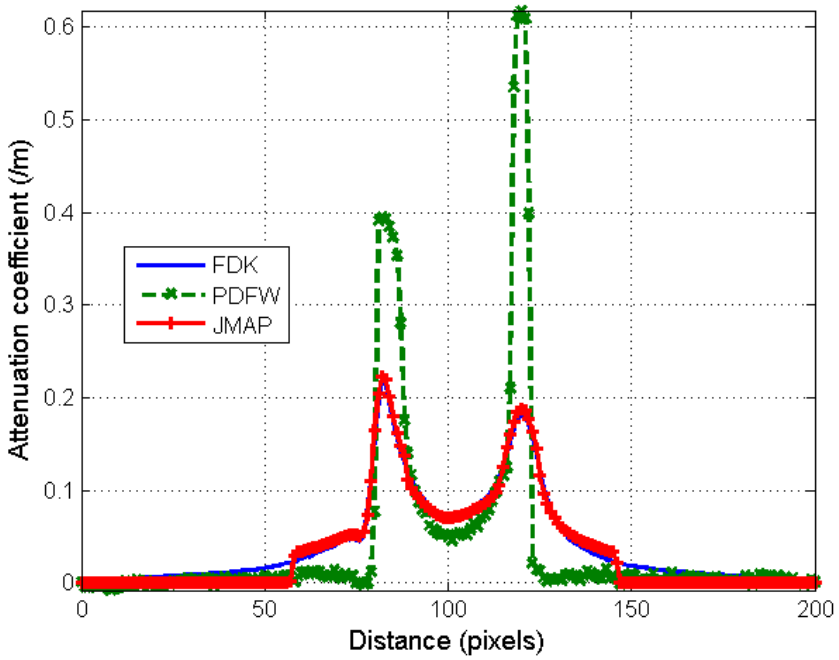
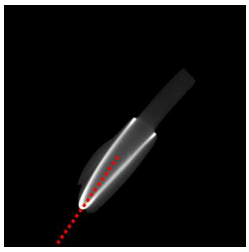


Figure 4.22: First profile of CMTE for each reconstruction by filtered backprojection [FDK84], PDFW [OMBF18] and JMAP



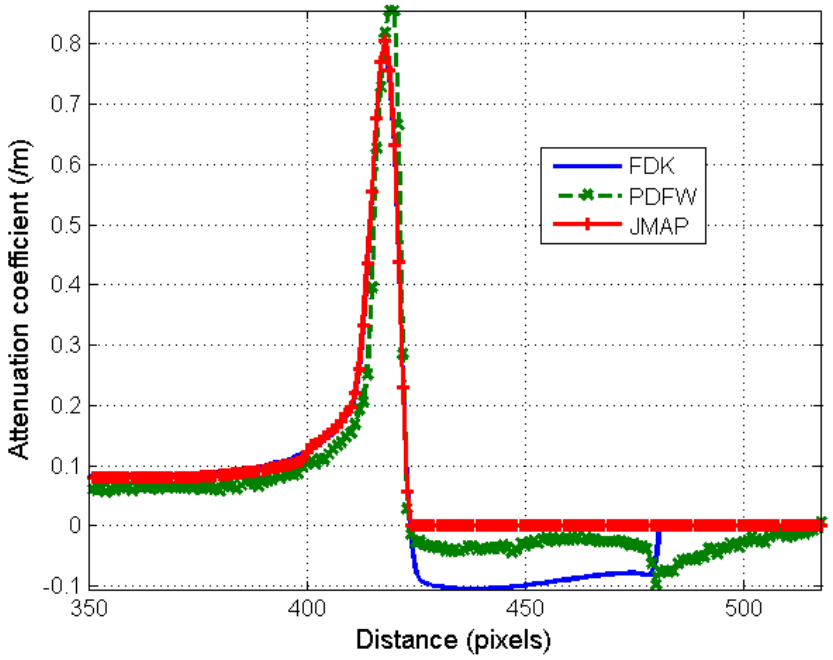


Figure 4.23: Second profile of CMTE for each reconstruction by filtered backprojection [FDK84], PDFW [OMBF18] and JMAP

Reconstruction algorithm	Computation time
PDFW	**2352 s**
JMAP	5074 s

Table 4.7: Comparison of the computation times of PDFW and JMAP, for the reconstruction of CMTE from 300 projections

Figure 4.24: Reconstruction of the metallic part from 500 projections by filtered backprojection [FDK84]

4.5.3 Evaluation on a metallic part

We do further experiments on a metallic part. The field-of-view is sampled into $1024 \times 1024 \times 1024$ voxels[1]. We use 500 projections uniformly distributed over $[0, 2\pi]$. The detector has 1024×1024 cells. The source-to-detector distance is 1077 mm, and the source-to-object distance is 700 mm. The part is only composed of metal : including the air, we fix $K = 2$. Since the volume is very huge and due to the fact that all the steps of our algorithm are still not parallelized on the GPU, we run JMAP during 20 iterations. The values for the parameters are given in table 4.8. Once again, we fix $\alpha_{\zeta_0} < \beta_{\zeta_0} < 1$ to take strong scattering and beam-hardening into account.

Parameters	K	γ_0	v_0	α_0	β_0	α_{ζ_0}	β_{ζ_0}	n_{vol}	n_{seg}
Fixed values	2	3	1	5	0.01	0.01	0.1	2	2

Table 4.8: Parameters of JMAP to reconstruct the metallic part

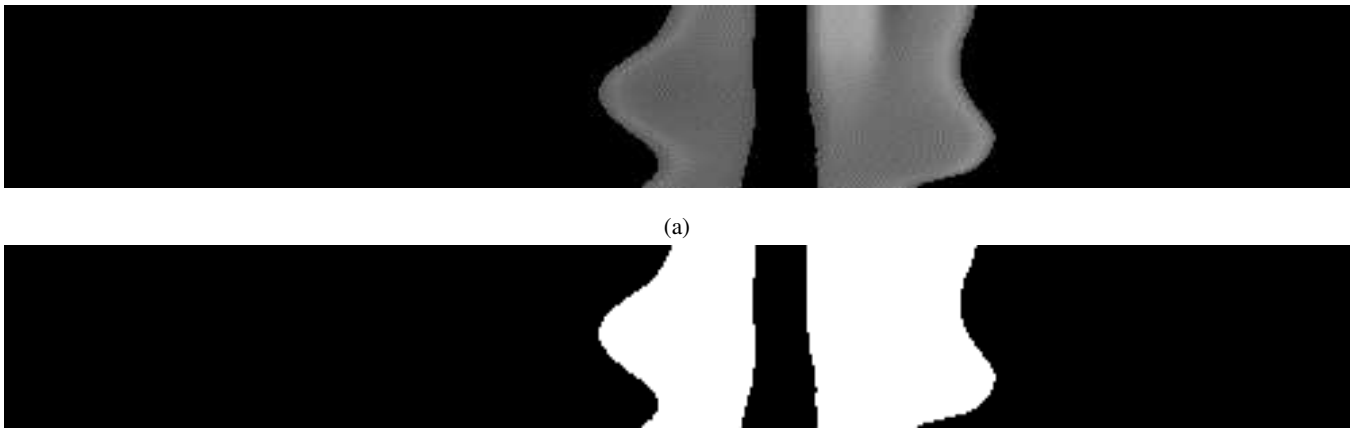
As for the CMTE volume, the metal is cause of many scattering and beam-hardening in the filtered backprojection shown in figure 4.24. The segmentation step in JMAP succeeds in removing these artifacts in the reconstruction shown in figure 4.25, since the gray values of scattering and beam-hardening are far from the one of the metal. Consequently, JMAP returns a reconstruction of good quality, for which the piecewise-constant prior has been enforced. Contrary to the filtered backprojection, the reconstruction by JMAP has sharp transitions between the air and the metal.

The convergence of JMAP is shown in figure 4.26. This reconstruction of a 1024^3 voxels has taken a long time which is $1\text{h}45$, since the segmentation step is performed on the CPU. Parallelizing this step on the GPU (which is feasible due our choice of ICM as the segmentation algorithm in JMAP) should allow to perform this reconstruction in less than one hour.

4.6 Conclusion and perspectives

In this chapter, we have presented a joint reconstruction and segmentation algorithm for 3D X-ray CT based on a Gauss-Markov-Potts prior on the volume, in order to reconstruct quasi-piecewise-constant industrial parts of SAFRAN. We have made efforts to explain how to correctly tune each parameter. In order to deal with huge volumes, we have justified our choice for highly-parallelizable ICM algorithm in order to perform the segmentation step. In simulation, we have shown that the algorithm can be made robust to the use of an unmatched pair of projector and backprojector. Nevertheless, we have also explained that this mismatching requires to perform much more backprojections than with a matched pair. We have experimentally shown the convergence of the algorithm. Nevertheless, we have underlined that the proposed algorithm performs a local optimization. As a result, we have insisted on the importance of its initialization. In particular, the results obtained by the algorithm are very dependent on the initial segmentation.

[1]For confidentiality reasons, the metallic part is only partly shown in figures 4.24 and 4.25.



(a)

(b)

Figure 4.25: Joint reconstruction (a) and segmentation (b) of the metallic part from 500 projections by JMAP

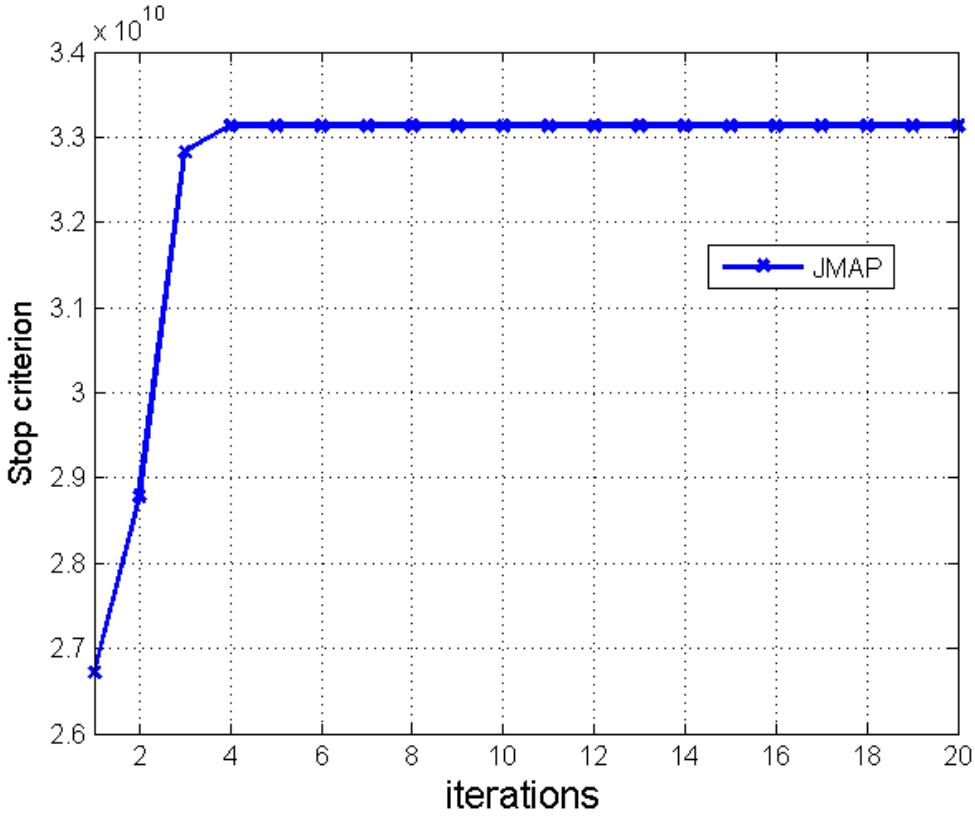


Figure 4.26: Convergence of JMAP for the joint reconstruction and segmentation of the metallic part from 500 projections

Our experiments on real data of SAFRAN have shown that the algorithm is actually able to enhance the reconstruction and to remove scattering and beam-hardening thanks to the segmentation step. Nevertheless,we have seen with the CMTE volume that the segmentation can be misestimated if the scattering is too close to a material in terms of gray values. Otherwise, the algorithm is efficient. Since the titanium is well separated from the scattering and the composite material in JMAP, a possible way to enhance the reconstruction is to perform the reconstruction of the composite material by another algorithm (such as PDFW or another), but inserting as prior information that we know where the titanium is. Since we have shown in our experiments that PDFW can remove the scattering in the CMTE volume, the knowledge about the position of the titanium should enable to avoid the artifacts due to the beam-hardening. This idea will be tested in future works.

A perspective to accelerate JMAP is to implement the segmentation step on the GPU as we have seen in our experiments with the 1024^3 metallic part. Another way to study is the reconstruction of textured materials such as woven composite materials. In further experiments, which are not presented in this thesis, we have seen that Gauss-Markov-Potts prior oversmoothes the weaving. Finding an appropriate prior for these materials remains an open problem.

Chapter 5

Error-splitting forward model and its application with Gauss-Markov-Potts prior

In the previous chapter, as in other MBIR methods (such as the ones presented in chapter 2), we have used the forward model

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\zeta}, \tag{5.1}$$

where $\boldsymbol{g}$ contains the projections, $\boldsymbol{f}$ is the volume and $\boldsymbol{\zeta}$ is a term accounting for the uncertainties on the projections, as explained in chapter 1. By enforcing a prior model $\mathcal{M}$ on the volume to reconstruct, we have seen that MBIR methods significantly ehance the reconstruction quality, particularly when the number of projections is low [Fes00]. In chapter 2, we have listed many prior models which have been applied in X-ray CT [SJP$^+$12, OMBF18, CMDGP17, SWFU15, RR07, VGVH$^+$13, WMDG17, XYM$^+$12, ZRLF18]. We have also seen that prior model $\mathcal{M}$ very often has hyperparameters $\boldsymbol{\theta}$ which can be difficult to tune. In chapter 4, we have proposed a Bayesian MBIR method which estimates optimal hyperparameters jointly with the volume

$$p(\boldsymbol{f}, \boldsymbol{\theta}|\mathcal{M}) = p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M}), \tag{5.2}$$

as it is done in [CMDGP17, WMDG17].

The reliability of the forward model (5.1), and therefore the accuracy of MBIR methods, depends on the prior model for uncertainties $\boldsymbol{\zeta}$. To derive the prior for uncertainties ζ_i for each ray i modeled as monochromatic, we have shown in chapter 1 that the expression of the Poisson statistics of photon count has been considered as [KPTF13] :

$$p(I_i|\boldsymbol{f}, r_i) = \mathcal{P}\left(I_0 e^{-[\boldsymbol{H}\boldsymbol{f}]_i} + r_i\right), \tag{5.3}$$

where I_0 denotes the mean number of photons sent by the source and r_i denotes the mean number of background events [KPTF13]. These background events can be for instance Compton scatter interactions undergone by X-ray photons inside the detector [NDMF$^+$13]. As said in chapter 1, the mean number r_i of background events is precomputed before the acquisition and taken into account in the linearization of the reconstruction problem

$$g_i = \ln\left(\frac{I_0}{I_i - r_i}\right) \forall i. \tag{5.4}$$

Given (5.4), we have shown in appendix A that uncertainties $\boldsymbol{\zeta}$ are well modeled by a Gaussian distribution

$$p(\zeta_i|v_{\zeta_i}) = \mathcal{N}(\zeta_i|0, v_{\zeta_i}), \forall i, \tag{5.5}$$

thanks to a second order Taylor series expansion of the Poisson statistics (5.3) of photons counts [SB93, TBSH06, TSBH07]. Inverses of variances $\boldsymbol{v}_\zeta$ weight the data and translate how the actual physical phenomena occuring in each measurement are far from or near the ideal physics described by Beer-Lambert's law. These variances can be pre-computed [TSBH07] or, as we did in chapter 4, optimized during the reconstruction [CMDGP17, WMDG17].

Although the commonly-used forward model (5.1) has given good results in 3D X-ray CT when combined with a prior model on the volume, the expression of the mean of photon counts in equation (5.3) does not take into account

all the phenomena involved in 3D X-ray CT. This led us to estimate optimal $\boldsymbol{v}_\zeta$ in chapter 4. As explained in chapter 1, the further phenomena involved in X-ray CT correspond to object-dependent uncertainties, such as scattering inside the volume, but also absorption of low energy photons occuring in presence of some materials such as metal [GDMJ$^+$16]. These further phenomena can be taken into account by considering the actual polychromacy of X-rays. The polychromatic model is obtained by integrating the attenuation term in (5.3) over all the energy spectrum :

$$p(I_i|\boldsymbol{f}(E), r_i) = \mathcal{P}\left(\int_E I_0(E) e^{-[\boldsymbol{H}\boldsymbol{f}(E)]_i} dE + r_i\right). \tag{5.6}$$

Based on this model, MBIR methods aiming at compensating beam-hardening address the reconstruction problem by estimating the density of material in each voxel, which is part of attenuation coefficients' expression [JBS15, MFDA18]. This estimation is preferred to the one of attenuation coefficients because the density of material only depends on voxel location [JBS15, MFDA18] : hence, the estimation of one reconstruction per energy is avoided. Nevertheless, the integration in polychromatic model (5.6) hinders the linearization of the reconstruction problem done for the monochromatic model by formula (1.8). Consequently, the use of polychromatic model (5.6) adds complexity to MBIR methods which are already computationally-demanding due to projection and backprojection operations. For this reason, even recently, polychromatic model (5.6) has only been used for 2D reconstruction problems [JBS15, MFDA18].

In order to tackle this problem and to reconstruct 3D images with a more accurate forward model than (5.1), in this chapter, we propose to make a tradeoff between the accuracy of polychromatic model (5.6) and the simplicity of monochromatic model (5.3). Since monochromatic model (5.3) is an approximation of polychromatic model (5.6), we suggest to add an unknown error term n_i in monochromatic model (5.3), which corresponds to the approximation error with respect to the polychromatic model :

$$p(I_i|\boldsymbol{f}, n_i, r_i) = \mathcal{P}\left(I_0 e^{-[\boldsymbol{H}\boldsymbol{f}]_i} + n_i + r_i\right), \forall i. \tag{5.7}$$

Furthermore, unknown error n_i enables to handle other physical phenomena described in [NDMF$^+$13], which would be too complex to model in a MBIR method.

Based on expression (5.7), we propose to derive a new linear forward model accounting for uncertainties, different from (5.1), in order to use it in a full MBIR method, and to compare results with those obtained with the usual forward model (5.1). In section 5.1, we perform a second order Taylor series expansion from Poisson statistics (5.7), in order to show that an error-splitting forward model of the form

$$\begin{cases} \boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\epsilon} \\ \boldsymbol{g}_0 = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\xi} \end{cases} \tag{5.8}$$

can be used for iterative reconstruction in 3D X-ray CT. Next, we assign different priors to uncertainties $\boldsymbol{\epsilon}$ and $\boldsymbol{\xi}$. Uncertainties $\boldsymbol{\epsilon}$ are called the measurement uncertainties and are modeled as Gaussian, while uncertainties $\boldsymbol{\xi}$ are called linear model uncertainties and are more difficult to model. In order to introduce robustness with respect to outliers, we assign a heavy-tailed prior to $\boldsymbol{\xi}$, which is in this chapter a generalized Student-t distribution [Dum16]. Both priors for $\boldsymbol{\epsilon}$ and $\boldsymbol{\xi}$ introduce parameters about which we give strategies to fix them. Then, in section 5.2, we propose a general MBIR algorithm for maximizing the joint posterior distribution of the unknowns and compare its cost to Joint Maximization A Posteriori (JMAP) with usual forward model (5.1). To implement this algorithm, conditional distributions of the unknowns in the error-splitting forward model are derived. The proposed algorithm is valid for any proper prior model on the volume. In order to reconstruct piecewise-constant parts for SAFRAN, we propose to combine the error-splitting forward model with Gauss-Markov-Potts prior model on the volume presented in chapter 4. We show results on real data in section 5.5 and give conclusions in section 5.6. The results compare the uses of usual forward model (5.1) and of the proposed forward model, with Gauss-Markov-Potts prior model on the volume. Compared to usual forward model (5.1), we show that, with Gauss-Markov-Potts prior, the use of the error-splitting forward model leads to more robust and accurate reconstructions.

5.1 Error-splitting forward model

5.1.1 Derivation of the error-splitting forward model

The photon count I_i measured by the detector for projection i is modeled as a Poisson process of which the statistics is given by equation (5.7), where r_i is the mean of background events (i.e. the dark scan), and n_i is the approximation error induced by monochromatic model (5.3) with respect to polychromatic model (5.6). Contrary to background events mean r_i which is precomputed, error n_i is unknown because of its dependance on both the acquisition and the inspected volume $\boldsymbol{f}$. Thus, error n_i cannot be pre-computed and taken into account in the linearization of the reconstruction problem in (5.4). From equation (5.7), we have

$$-\ln(p(I_i|\boldsymbol{f},n_i,r_i)) = \ln(I_i!) + I_0 e^{-[\boldsymbol{H}\boldsymbol{f}]_i} + n_i + r_i - I_i \ln\left(I_0 e^{-[\boldsymbol{H}\boldsymbol{f}]_i} + n_i + r_i\right). \tag{5.9}$$

After reparametrization (5.4), a quadratic approximation from second order Taylor series expansion of (5.9) is derived in appendix C. It leads to

$$-\ln(p(g_i|\boldsymbol{f},n_i,r_i) \approx \ln(I_i!) + I_i + n_i - I_i \ln(I_i+n_i) + \frac{a_i}{2}\left(g_i - [\boldsymbol{H}\boldsymbol{f}]_i + \frac{b_i}{a_i}\right)^2 - \frac{b_i^2}{2a_i} \tag{5.10}$$

where

$$\begin{cases} a_i = (I_i - r_i)\left(1 - \frac{I_i(r_i+n_i)}{(I_i+n_i)^2}\right) \\ b_i = \frac{n_i(I_i-r_i)}{I_i+n_i} \end{cases}. \tag{5.11}$$

By denoting

$$\xi_i = -\frac{b_i}{a_i} = -\frac{n_i(I_i+n_i)}{(I_i+n_i)^2 - I_i(r_i+n_i)} \tag{5.12}$$

and

$$v_{\epsilon_i} = \frac{1}{a_i}, \forall i, \tag{5.13}$$

we have the following expression for the log-likelihood

$$\ln(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{n},\boldsymbol{r})) = C - \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f} - \boldsymbol{\xi}\|^2_{\boldsymbol{V}_\epsilon} \tag{5.14}$$

where $\boldsymbol{V}_\epsilon = \operatorname{diag}[\boldsymbol{v}_\epsilon]$, $\boldsymbol{v}_\epsilon = (v_{\epsilon_i})_i$ and

$$C = -\sum_{i=1}^{M}\left[\ln(I_i!) + I_i + n_i - I_i\ln(I_i+n_i) - \frac{b_i^2}{2a_i}\right]. \tag{5.15}$$

As already mentioned, from a physical standpoint, error n_i depends on the inspected volume $\boldsymbol{f}$. Nevertheless, taking into account the relationship between n_i and $\boldsymbol{f}$ would lead to a non-linear forward model which would be computationally untractable in 3D. For this reason, from a mathematical standpoint, n_i is seen as a variable independent from $\boldsymbol{f}$, which only compensates the difference between the polychromatic and the monochromatic models of the rays. Consequently, since it does not include $\boldsymbol{f}$, term C given by equation (5.15) can be seen as a constant. Hence, the new forward model we now derive from the approximate log-likelihood (5.14) is linear. In (5.14), term $\boldsymbol{\xi}$ appears in the data-fidelity term as a bias which is unknown due to error n_i. By introducing intermediate projections

$$g_{0_i} = [\boldsymbol{H}\boldsymbol{f}]_i + \xi_i, \forall i, \tag{5.16}$$

we are able to write the new forward model

$$\begin{cases} g_i = g_{0_i} + \epsilon_i \text{ where } p(\epsilon_i|v_{\epsilon_i}) = \mathcal{N}(\epsilon_i|0,v_{\epsilon_i}), \\ g_{0_i} = [\boldsymbol{H}\boldsymbol{f}]_i + \xi_i \end{cases} \forall i, \tag{5.17}$$

which leads to the same log-likelihood than (5.14), up to an additive constant. The vector form of this forward model is given by equation (5.8) : we see that it splits the uncertainties into two terms : $\boldsymbol{\epsilon}$ and $\boldsymbol{\xi}$. For this reason, we call it the *error-splitting forward model*. Since uncertainties $(\epsilon_i)_i$ are related to measurements g_i, we call $\boldsymbol{\epsilon}$ the *measurement uncertainties*. Concerning uncertainties $(\xi_i)_i$, since it corresponds in (5.14) to a bias between actual and theoretical projections $\boldsymbol{g}$ and $\boldsymbol{H}\boldsymbol{f}$, we call $\boldsymbol{\xi}$ the *linear model uncertainties*.

5.1.2 Measurement uncertainties

As we have seen in section 5.1.1, the measurement uncertainties $(\epsilon_i)_i$ in

$$g_i = g_{0_i} + \epsilon_i \tag{5.18}$$

are zero-mean Gaussian :

$$p(\epsilon_i|v_{\epsilon_i}) = \mathcal{N}(\epsilon_i|0, v_{\epsilon_i}), \forall i, \tag{5.19}$$

where v_{ϵ_i} is given by (5.13). We see that v_{ϵ_i} is the inverse of a_i which depends on n_i in (5.11). Since n_i is unknown, we jointly estimate v_{ϵ_i} with the volume. In order to have analytically tractable update formulae, v_{ϵ_i} is assigned a conjugate prior which is an Inverse-Gamma distribution, of which the expression has been already seen in chapter 4 :

$$p(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) = \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) = \frac{\beta_{\epsilon_0}^{\alpha_{\epsilon_0}}}{\Gamma(\alpha_{\epsilon_0})} v_{\epsilon_i}^{-\alpha_{\epsilon_0}-1} \exp\left[-\frac{\beta_{\epsilon_0}}{v_{\epsilon_i}}\right], v_{\epsilon_i} > 0, \forall i. \tag{5.20}$$

Parameters α_{ϵ_0} and β_{ϵ_0} are fixed. We will discuss about the choice of these parameters in section 5.1.4.

5.1.3 Linear model uncertainties

As we have seen in section 5.1.1, linear model uncertainties $\boldsymbol{\xi}$ are a bias which can be separated from measurement uncertainties $\boldsymbol{\epsilon}$ by introducing intermediate projections

$$\boldsymbol{g}_0 = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\xi}. \tag{5.21}$$

From

$$\boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\epsilon}, \tag{5.22}$$

projections $\boldsymbol{g}_0$ appear as the projections unnoisy by the measurement uncertainties. For this reason, we call $\boldsymbol{g}_0$ the "true" projections. In order to estimate volume $\boldsymbol{f}$ from $\boldsymbol{g}_0$ taking $\boldsymbol{\xi}$ into account, we need to assign a prior to linear model uncertainties $\boldsymbol{\xi}$. This is not obvious because, contrary to $\boldsymbol{\epsilon}$, this prior cannot be deduced from the second order Taylor series expansion derived in section 5.1.1. Since $\boldsymbol{\xi}$ is difficult to physically model due to its high dependency on the inspected volume, we prefer to assign a heavy-tailed prior to it. The use of a heavy-tailed prior enables to model that possibly high ξ_i can be present in projections $\boldsymbol{g}_0$. Consequently, compared, for instance, to a Gaussian prior, it brings more robustness to the estimation of $\boldsymbol{f}$ from $\boldsymbol{g}_0$.

Among heavy-tailed distributions, zero-mean normal variance mixtures have appealing properties for simple computations in linear inverse problems [DGWMD17]. Among them, generalized Student-t distribution $\mathcal{S}t_g$ [Dum16]

$$p(\xi_i|\alpha_{\xi_0}, \beta_{\xi_0}) = \mathcal{S}t_g(\xi_i|\alpha_{\xi_0}, \beta_{\xi_0}), \forall i, \tag{5.23}$$

corresponds to a mixture on the variances between a zero-mean Gaussian and an Inverse-Gamma distributions

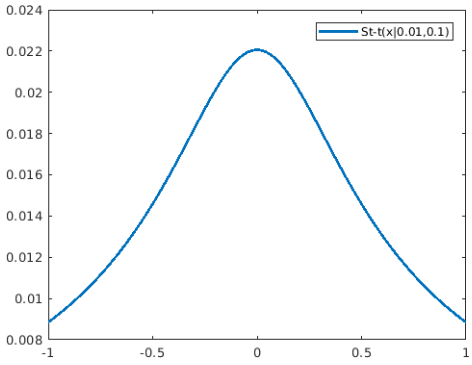
$$\begin{cases} p(\xi_i|v_{\xi_i}) = \mathcal{N}(\xi_i|0, v_{\xi_i}) \\ p(v_{\xi_i}|\alpha_{\xi_0}, \beta_{\xi_0}) = \mathcal{IG}(v_{\xi_i}|\alpha_{\xi_0}, \beta_{\xi_0}) \end{cases}, \forall i. \tag{5.24}$$

By introducing hidden variances $\boldsymbol{v}_\xi$ in the reconstruction problem, generalized Student-t distribution can be easily dealt with thanks to simple calculations involving normal distribution for $\boldsymbol{\xi}$ and conjugate Inverse-Gamma prior for $\boldsymbol{v}_\xi$. Parameters α_{ξ_0} and β_{ξ_0} are fixed according to the strategy presented in section 5.1.4.

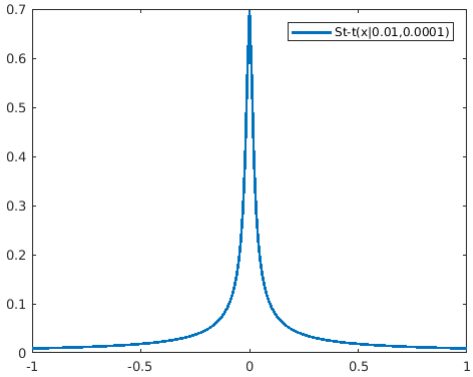
5.1.4 Fixing the parameters of the error-splitting forward model

From (5.19) and (5.20), since we have assigned a conjugate prior on variances $\boldsymbol{v}_\epsilon$, the prior for measurement uncertainties is in fact a normal variance mixture

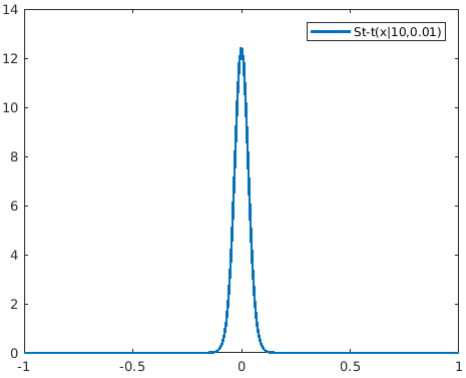
$$\begin{cases} p(\epsilon_i|v_{\epsilon_i}) = \mathcal{N}(\epsilon_i|0, v_{\epsilon_i}) \\ p(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) = \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \end{cases}, \forall i, \tag{5.25}$$



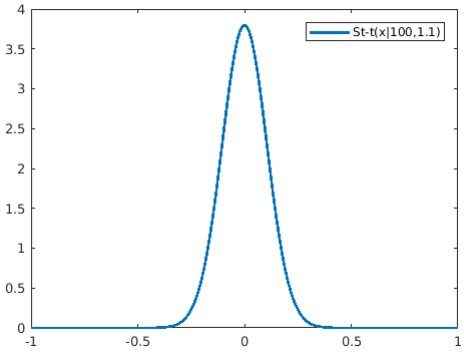
(a) $\alpha_{\xi_0} < 1, \beta_{\xi_0} < 1, \alpha_{\xi_0} < \beta_{\xi_0}$



(b) $\alpha_{\xi_0} < 1, \beta_{\xi_0} < 1, \alpha_{\xi_0} > \beta_{\xi_0}$



(c) $\alpha_{\xi_0} > 1, \beta_{\xi_0} < 1$



(d) $\alpha_{\xi_0} > 1, \beta_{\xi_0} > 1, \alpha_{\xi_0} > \beta_{\xi_0}$

Figure 5.1: Probability density function of $\mathcal{S}t_g(.|\alpha_{\xi_0}, \beta_{\xi_0})$ for several values of $(\alpha_{\xi_0}, \beta_{\xi_0})$

which is a generalized Student-t distribution as for the linear model uncertainties. This is a problem because $\boldsymbol{\epsilon}$ has to be Gaussian according to (5.17). In order to see how to actually assign a Gaussian prior to measurement uncertainties keeping $\boldsymbol{v}_\epsilon$ unknown, we write the marginal prior of ϵ_i with respect to v_{ϵ_i}

$$p(\epsilon_i|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) = \int_0^{+\infty} \mathcal{N}(\epsilon_i|0, v_{\epsilon_i}) \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \, \mathrm{d}v_{\epsilon_i} \tag{5.26}$$

which reads [DGWMD17]

$$p(\epsilon_i|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \propto \left(1 + \frac{\epsilon_i^2}{2\beta_{\epsilon_0}}\right)^{-\left(\alpha_{\epsilon_0} + \frac{1}{2}\right)} \tag{5.27}$$

where normalizing constants are omitted. From formula (5.27), we see that, if we fix β_{ϵ_0} such that

$$\left|\frac{\epsilon_i^2}{\beta_{\epsilon_0}}\right| \ll 1 \quad \forall i, \tag{5.28}$$

the following approximation holds :

$$p(\epsilon_i|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \underset{\left|\frac{\epsilon_i^2}{\beta_{\epsilon_0}}\right| \ll 1}{\sim} \exp\left[-\left(\alpha_{\epsilon_0} + \frac{1}{2}\right) \frac{\epsilon_i^2}{2\beta_{\epsilon_0}}\right] \tag{5.29}$$

and means that ϵ_i can actually be modeled as zero-mean Gaussian

$$p(\epsilon_i|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \underset{\left|\frac{\epsilon_i^2}{\beta_{\epsilon_0}}\right| \ll 1}{\sim} \mathcal{N}\left(\epsilon_i \middle| 0, \frac{\beta_{\epsilon_0}}{\alpha_{\epsilon_0} + \frac{1}{2}}\right), \forall i. \tag{5.30}$$

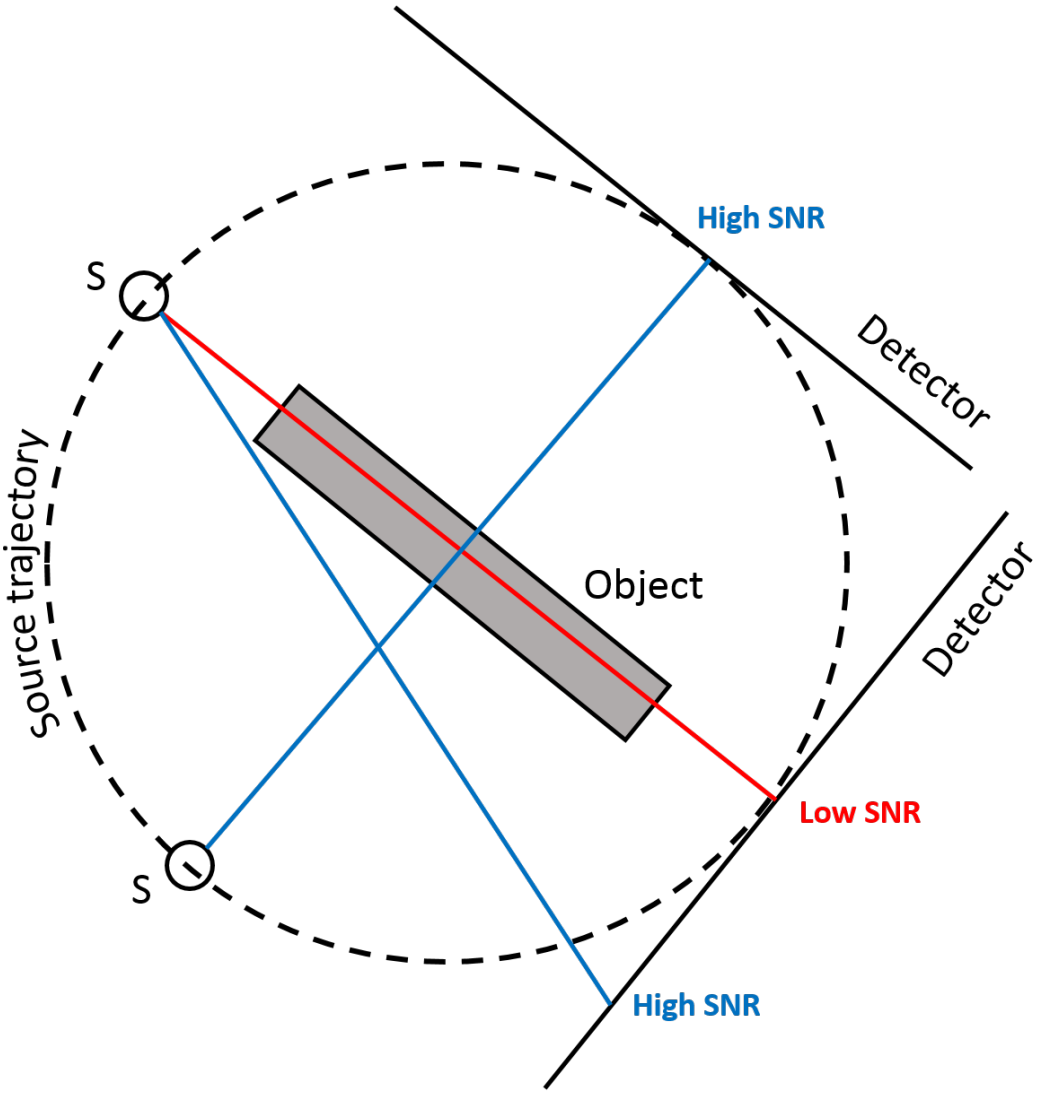


Figure 5.2: Illustration of the dependence of the Signal-to-Noise Ratio (SNR) on the thickness of the material crossed by the rays

Thus, generalized Student-t distribution is sufficiently flexible to assign Gaussian prior for measurement uncertainties $\boldsymbol{\epsilon}$ and heavy-tailed prior for linear model uncertainties $\boldsymbol{\xi}$.

For fixed β_{ϵ_0} complying with constraint (5.28), we derive in appendix D a formula to automatically set α_{ϵ_0} :

$$\alpha_{\epsilon_0} = M \times \frac{\beta_{\epsilon_0}}{\|\boldsymbol{g}\|_2^2} \times \left(1 + 10^{\frac{SNR}{10}}\right) - \frac{1}{2}, \tag{5.31}$$

where SNR corresponds to a prior on the *Signal-to-Noise Ratio* (SNR). In this work, SNR is a parameter which is fixed manually by the user based on physical intuition. Actually, the SNR for each projection is different depending on the materials crossed by the rays and on the thickness of these materials. For instance, the SNR in rays going through air is much higher than the one in rays going through thick materials. As shown in figure 5.2, it is also clear that the SNR depends on the projection direction : if the projection direction is such that the path of a ray inside the inspected object is short, then the SNR in this ray is high. On the opposite, if the projection direction is such that this path is much longer, then the SNR is much lower. Given these observations, one improvement would be to assign a different SNR_i to each ray i (and consequently, based on formula (5.31), a different $\alpha_{\epsilon_{0_i}}$ to each measurement uncertainty ϵ_i). Parameter SNR_i would be fixed to a more or less high value depending on the length of the path of ray i inside the object. This length would be approximately computed based on a segmentation of the volume initializing the reconstruction, obtained as discussed in section 4.3.6. Although this idea seems interesting to test, including it in this work would complicate the comparison between the usual forward model (5.1) and the proposed error-splitting one, which is the scope of this chapter. For this reason, this enhancement will be considered in future works.

Concerning parameters α_{ξ_0} and β_{ξ_0} for linear model uncertainties, their fixation tunes the tail of the generalized Student-t distribution in order to introduce robustness as motivated in section 5.1.3. Figure 5.1 shows the probability density function of $\mathcal{S}t_g(.|\alpha_{\xi_0}, \beta_{\xi_0})$ for different values of α_{ξ_0} and β_{ξ_0}. We see that generalized Student-t distribution is heavy-tailed for $\alpha_{\xi_0} < 1$. On the contrary, the tail is short for $\alpha_{\xi_0} > 1$, which is not good to enforce robustness. Hence, we fix $\alpha_{\xi_0} < 1$. For $\alpha_{\xi_0} < 1$, we see that $\mathcal{S}t_g(.|\alpha_{\xi_0}, \beta_{\xi_0})$ is concentrated around 0 for $\beta_{\xi_0} < 1$ and $\alpha_{\xi_0} > \beta_{\xi_0}$: this makes this setting appropriate when linear model uncertainties are high in a small number of measurements. In the other case, fixing $\alpha_{\xi_0} < \beta_{\xi_0}$ and $\beta_{\xi_0} < 1$ makes the tail very heavy : it is more appropriate when linear model uncertainties are high in all the data. Consequently, we fix $\beta_{\xi_0} < \alpha_{\xi_0} < 1$ when scattering and absorption are low, and $\alpha_{\xi_0} < \beta_{\xi_0} < 1$ when they are high.

Now we have explained strategies to fix the parameters, we detail in section 5.2 a reconstruction algorithm using the error-splitting forward model.

5.2 Joint Maximization A Posteriori (JMAP) using the error-splitting forward model

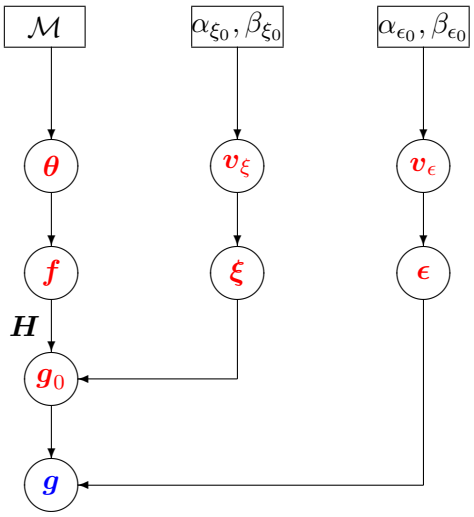


Figure 5.3: General hierarchical model including the error-splitting forward model

The error-splitting forward model is summarized by equations

$$\begin{cases} \boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\epsilon} \\ \boldsymbol{g}_0 = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\xi} \end{cases} . \tag{5.32}$$

The model developed in section 5.1.1 for splitted uncertainties :

$$\begin{cases} p(\epsilon_i|v_{\epsilon_i}) = \mathcal{N}(\epsilon_i|0, v_{\epsilon_i}) \\ p(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) = \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\xi_i|v_{\xi_i}) = \mathcal{N}(\xi_i|0, v_{\xi_i}) \\ p(v_{\xi_i}|\alpha_{\xi_0}, \beta_{\xi_0}) = \mathcal{IG}(v_{\xi_i}|\alpha_{\xi_0}, \beta_{\xi_0}) \end{cases} \forall i, \tag{5.33}$$

is denoted by $\mathcal{S}$. We consider a general prior model $\mathcal{M}$ on the volume with hyperparameters $\boldsymbol{\theta}$ which are estimated jointly with the other unknowns. Figure 5.3 summarizes the general hierarchical model when the error-splitting forward model is used : data $\boldsymbol{g}$ are the combination of "true" projections $\boldsymbol{g}_0$ and of measurement uncertainties $\boldsymbol{\epsilon}$ with variances $\boldsymbol{v}_\epsilon$. Projections $\boldsymbol{g}_0$ result from volume $\boldsymbol{f}$ (through projection operation $\boldsymbol{H}$), and from linear model uncertainties $\boldsymbol{\xi}$ with variances $\boldsymbol{v}_\xi$. At last, volume $\boldsymbol{f}$ is described by prior model $\mathcal{M}$ with hyperparameters $\boldsymbol{\theta}$. Hyperparameters $\boldsymbol{\theta}$ can be assigned a prior distribution depending on $\mathcal{M}$, while variances $\boldsymbol{v}_\xi$ and $\boldsymbol{v}_\epsilon$ have Inverse-Gamma priors with parameters $(\alpha_{\xi_0}, \beta_{\xi_0})$ and $(\alpha_{\epsilon_0}, \beta_{\epsilon_0})$ respectively. The joint posterior distribution of the unknowns is given by Bayes's

rule :

$$p(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{S}, \mathcal{M}) \propto p(\boldsymbol{g}|\boldsymbol{g}_0, \boldsymbol{v}_\epsilon) p(\boldsymbol{g}_0|\boldsymbol{f}, \boldsymbol{v}_\xi) p(\boldsymbol{v}_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0})\, p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M})\, p(\boldsymbol{v}_\xi|\alpha_{\xi_0}, \beta_{\xi_0})\, p(\boldsymbol{\theta}|\mathcal{M}). \tag{5.34}$$

Similarly to the proposed reconstruction method in chapter 4, we choose to perform a joint maximization a posteriori (JMAP)

$$(\hat{\boldsymbol{g}}_0, \hat{\boldsymbol{v}}_\epsilon, \hat{\boldsymbol{f}}, \hat{\boldsymbol{v}}_\xi, \hat{\boldsymbol{\theta}}) = \underset{(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{\theta})}{\arg\max} \{p(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{S}, \mathcal{M})\} \tag{5.35}$$

by an alternate optimization algorithm

$$\begin{cases} \boldsymbol{g}_0^{(t+1)} = \arg\max_{\boldsymbol{g}_0} \left\{ p(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon^{(t)}, \boldsymbol{f}^{(t)}, \boldsymbol{v}_\xi^{(t)}, \boldsymbol{\theta}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}) \right\} & \text{(5.36a)} \\ \boldsymbol{v}_\epsilon^{(t+1)} = \arg\max_{\boldsymbol{v}_\epsilon} \left\{ p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon, \boldsymbol{f}^{(t)}, \boldsymbol{v}_\xi^{(t)}, \boldsymbol{\theta}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}) \right\} & \text{(5.36b)} \\ \boldsymbol{f}^{(t+1)} = \arg\max_{\boldsymbol{f}} \left\{ p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}, \boldsymbol{v}_\xi^{(t)}, \boldsymbol{\theta}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}) \right\} & \text{(5.36c)} \\ \boldsymbol{v}_\xi^{(t+1)} = \arg\max_{\boldsymbol{v}_\xi} \left\{ p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\xi, \boldsymbol{\theta}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}) \right\} & \text{(5.36d)} \\ \boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\xi^{(t+1)}, \boldsymbol{\theta} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}) \right\} & \text{(5.36e)} \end{cases}$$

The criterion to maximize is derived from joint posterior distribution (5.34), from which constants are removed :

$$\begin{aligned} \mathcal{L}(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{\theta}) = & -\frac{1}{2}\|\boldsymbol{g} - \boldsymbol{g}_0\|^2_{\boldsymbol{V}_\epsilon} - \frac{1}{2}\|\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\xi} - \sum_{i=1}^{M} \left[\left(\alpha_{\epsilon_0} + \frac{3}{2} \right) \ln(v_{\epsilon_i}) + \frac{\beta_{\epsilon_0}}{v_{\epsilon_i}} \right] \\ & + \ln(p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M})) - \sum_{i=1}^{M} \left[\left(\alpha_{\xi_0} + \frac{3}{2} \right) \ln(v_{\xi_i}) + \frac{\beta_{\xi_0}}{v_{\xi_i}} \right] + \ln(p(\boldsymbol{\theta}|\mathcal{M})). \end{aligned} \tag{5.37}$$

Like in chapter 4, according to Bayes's rule, each step of the algorithm is equivalent to maximizing the conditional distribution of each unknown given the others. In the following of this section, we detail these steps for the general case with any prior model $\mathcal{M}$ on the volume.

5.2.1 Conditional distribution of the "true" projections

According to Bayes's rule, the conditional distribution of the "true" projections given the other unknowns reads

$$\begin{aligned} p(\boldsymbol{g}_0|\boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{\theta}; \boldsymbol{g}, \mathcal{S}, \mathcal{M}) &\propto p(\boldsymbol{g}|\boldsymbol{g}_0, \boldsymbol{v}_\epsilon) p(\boldsymbol{g}_0|\boldsymbol{f}, \boldsymbol{v}_\xi) \\ &\propto \exp\left[-\frac{1}{2}\|\boldsymbol{g} - \boldsymbol{g}_0\|^2_{\boldsymbol{V}_\epsilon} - \frac{1}{2}\|\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\xi} \right] \end{aligned} \tag{5.38}$$

where $\boldsymbol{V}_\epsilon = \text{diag}[\boldsymbol{v}_\epsilon]$ and $\boldsymbol{V}_\xi = \text{diag}[\boldsymbol{v}_\xi]$. Hence, conditional distribution (5.38) is a Gaussian

$$p(\boldsymbol{g}_0|\boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{\theta}; \boldsymbol{g}, \mathcal{S}, \mathcal{M}) = \mathcal{N}(\boldsymbol{g}_0|\boldsymbol{m}_{g_0}, \boldsymbol{V}_{g_0}) \tag{5.39}$$

where

$$\begin{cases} \boldsymbol{V}_{g_0} = \left(\boldsymbol{V}_\epsilon^{-1} + \boldsymbol{V}_\xi^{-1} \right)^{-1} \\ \boldsymbol{m}_{g_0} = \boldsymbol{V}_{g_0} \left(\boldsymbol{V}_\epsilon^{-1}\boldsymbol{g} + \boldsymbol{V}_\xi^{-1}\boldsymbol{H}\boldsymbol{f} \right) \end{cases}. \tag{5.40}$$

Hence, step (5.36a) sets

$$\hat{\boldsymbol{g}}_0 = \boldsymbol{m}_{g_0} = \left(\boldsymbol{V}_\epsilon^{-1} + \boldsymbol{V}_\xi^{-1} \right)^{-1} \left(\boldsymbol{V}_\epsilon^{-1}\boldsymbol{g} + \boldsymbol{V}_\xi^{-1}\boldsymbol{H}\boldsymbol{f} \right) \tag{5.41}$$

which is simple to compute since $\boldsymbol{V}_{g_0}$ is diagonal.

5.2.2 Conditional distribution of the variances of the measurement uncertainties

By applying Bayes's rule, we have

$$\begin{aligned} p(\boldsymbol{v}_\epsilon|\boldsymbol{g}_0,\boldsymbol{f},\boldsymbol{v}_\xi,\boldsymbol{\theta};\boldsymbol{g},\mathcal{S},\mathcal{M}) &\propto p(\boldsymbol{g}|\boldsymbol{g}_0,\boldsymbol{v}_\epsilon)p(\boldsymbol{v}_\epsilon|\alpha_{\epsilon_0},\beta_{\epsilon_0}) \\ &\propto \prod_{i=1}^{M} \exp\left[-\left(\alpha_{\epsilon_0}+\frac{3}{2}\right)-\frac{1}{v_{\epsilon_i}}\left(\beta_{\epsilon_0}+\frac{1}{2}(g_i-g_{0_i})^2\right)\right]. \end{aligned} \tag{5.42}$$

Hence, the conditional distribution of the variances of the measurement uncertainties is Inverse-Gamma

$$p(v_{\epsilon_i}|\boldsymbol{g}_0,\boldsymbol{f},\boldsymbol{v}_\xi,\boldsymbol{\theta};\boldsymbol{g},\mathcal{S},\mathcal{M}) = \mathcal{IG}(v_{\epsilon_i}|\hat{\alpha}_{\epsilon_{0_i}},\hat{\beta}_{\epsilon_{0_i}}) \tag{5.43}$$

where

$$\left\{\begin{array}{l} \hat{\alpha}_{\epsilon_{0_i}} = \alpha_{\epsilon_0}+\frac{1}{2} \\ \hat{\beta}_{\epsilon_{0_i}} = \beta_{\epsilon_0}+\frac{1}{2}\left(g_i-g_{0_i}\right)^2 \end{array}\right. ,\forall i\in\{1,\ldots,M\}. \tag{5.44}$$

So, step (5.36b) sets

$$\hat{v}_{\epsilon_i} = \frac{\hat{\beta}_{\epsilon_{0_i}}}{\hat{\alpha}_{\epsilon_{0_i}}+1}, \forall i\in\{1,\ldots,M\}. \tag{5.45}$$

5.2.3 Conditional distribution of the volume

The conditional distribution of the volume given the other unknowns depends on its prior model

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{g}_0,\boldsymbol{v}_\epsilon,\boldsymbol{f},\boldsymbol{v}_\xi,\boldsymbol{\theta};\boldsymbol{g},\mathcal{S},\mathcal{M}) &\propto p(\boldsymbol{g}_0|\boldsymbol{f},\boldsymbol{v}_\xi)p(\boldsymbol{f}|\boldsymbol{\theta};\mathcal{M}) \\ &\propto \exp\left[-\frac{1}{2}\|\boldsymbol{g}_0-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\xi}\right] p(\boldsymbol{f}|\boldsymbol{\theta};\mathcal{M}) \end{aligned} \tag{5.46}$$

So, step (5.36c) consists in minimizing

$$J_f(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{g}_0-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\xi} - \ln\left(p(\boldsymbol{f}|\boldsymbol{\theta};\mathcal{M})\right) \tag{5.47}$$

by an optimization algorithm which is dependent on the prior model on the volume.

5.2.4 Conditional distribution of the variances of the linear model uncertainties

Thanks to the use of conjugate prior, we have :

$$p(v_{\xi_i}|\boldsymbol{g}_0,\boldsymbol{v}_\epsilon,\boldsymbol{f},\boldsymbol{\theta};\boldsymbol{g},\mathcal{S},\mathcal{M}) = \mathcal{IG}(v_{\xi_i}|\hat{\alpha}_{\xi_{0_i}},\hat{\beta}_{\xi_{0_i}}) \tag{5.48}$$

where

$$\left\{\begin{array}{l} \hat{\alpha}_{\xi_{0_i}} = \alpha_{\xi_0}+\frac{1}{2} \\ \hat{\beta}_{\xi_{0_i}} = \beta_{\xi_0}+\frac{1}{2}\left(g_{0_i}-[\boldsymbol{H}\boldsymbol{f}]_i\right)^2 \end{array}\right. ,\forall i\in\{1,\ldots,M\}. \tag{5.49}$$

So, step (5.36d) sets

$$\hat{v}_{\xi_i} = \frac{\hat{\beta}_{\xi_{0_i}}}{\hat{\alpha}_{\xi_{0_i}}+1}, \forall i\in\{1,\ldots,M\}. \tag{5.50}$$

5.2.5 Conditional distribution of the hyperparameters of the prior model on the volume

The conditional distribution of hyperparameters $\boldsymbol{\theta}$

$$p(\boldsymbol{\theta}|\boldsymbol{g}_0,\boldsymbol{v}_\epsilon,\boldsymbol{f},\boldsymbol{v}_\xi,;\boldsymbol{g},\mathcal{S},\mathcal{M}) \propto p(\boldsymbol{f}|\boldsymbol{\theta};\mathcal{M})p(\boldsymbol{\theta}|\mathcal{M}) \tag{5.51}$$

only depends on the prior model on the volume and not on the used forward model.

Algorithm 13 JMAP algorithm using the error-splitting forward model

Initialize $\boldsymbol{v}_\epsilon^{(0)}$, $\boldsymbol{f}^{(0)}$, $\boldsymbol{v}_\xi^{(0)}$ and $\boldsymbol{\theta}^{(0)}$
for $t = 0, \dots, t_{max} - 1$ **do**
1 : Update $\boldsymbol{g}_0$ by (5.41)
2 : Update $\boldsymbol{v}_\epsilon$ by (5.45)
3 : Update $\boldsymbol{f}$ by minimizing criterion (5.47) through n_{vol} iterations of an optimization algorithm
4 : Update $\boldsymbol{v}_\xi$ by (5.50)
5 : Update $\boldsymbol{\theta}$ by maximizing (5.51)
end for

5.2.6 Discussion

The overall algorithm is summarized in algorithm 13. Initial volume $\boldsymbol{f}^{(0)}$ is obtained by a filtered backprojection method such as FDK [FDK84]. On this first reconstruction, initial values $\boldsymbol{\theta}^{(0)}$ of hyperparameters $\boldsymbol{\theta}$ can be computed, as we did in chapter 4 for $\boldsymbol{\theta} = (\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v})$. "True" projections $\boldsymbol{g}_0$ do not need to be initialized since they are immediately estimated at the beginning of the algorithm. We use formulae (5.45) and (5.50) in order to initialize $\boldsymbol{v}_\epsilon$ and $\boldsymbol{v}_\xi$. Since the "true" projections cannot be known at initialization, we replace $\boldsymbol{g}_0$ by $\boldsymbol{g}$ and $\boldsymbol{H}\boldsymbol{f}$ respectively :

$$\hat{v}_{\epsilon_i}^{(0)} = \frac{\beta_{\epsilon_0} + \frac{1}{2}\left(g_i - \left[\boldsymbol{H}\boldsymbol{f}^{(0)}\right]_i\right)^2}{\alpha_{\epsilon_0} + 1}, \forall i \in \{1, \dots, M\}. \tag{5.52}$$

and

$$\hat{v}_{\xi_i}^{(0)} = \frac{\beta_{\xi_0} + \frac{1}{2}\left(g_i - \left[\boldsymbol{H}\boldsymbol{f}^{(0)}\right]_i\right)^2}{\alpha_{\xi_0} + \frac{3}{2}}, \forall i \in \{1, \dots, M\}. \tag{5.53}$$

Like our joint reconstruction and segmentation method in chapter 4, algorithm 13 has a hidden parameter which is the number n_{vol} of subiterations to perform in order to update the volume. This parameter is fixed in the same way as the one explained in section 4.3.6. As already pointed out, the algorithm is applicable with any proper prior model on the volume. In section 5.4, we combine the error-splitting forward model with Gauss-Markov-Potts prior model on the volume used in chapter 4, in order to complete the algorithm with explicit updates of volume $\boldsymbol{f}$ and hyperparameters $\boldsymbol{\theta}$. Before that, we compare the cost of the use of the error-splitting forward model with respect to the usual one.

5.3 Comparison with the usual forward model

With the usual forward model (5.1), total uncertainties $\boldsymbol{\zeta}$ are Gaussian, with variances $\boldsymbol{v}_\zeta$, as explained in chapter 1. Like in chapter 4, we denote this forward model by $\mathcal{U}$. A conjugate prior can be assigned to $\boldsymbol{v}_\zeta$, as done in chapter 4 [CMDGP17, WMDG17].

With the usual forward model (5.1), true projections $\boldsymbol{g}_0$ do not appear since the uncertainties are not splitted. To estimate volume $\boldsymbol{f}$, variances $\boldsymbol{v}_\zeta$ and hyperparameters $\boldsymbol{\theta}$, their joint posterior distribution

$$p(\boldsymbol{f}, \boldsymbol{v}_\zeta, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{U}, \mathcal{M}) \propto p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{v}_\zeta)\, p(\boldsymbol{v}_\zeta|\mathcal{U})\, p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M})\, p(\boldsymbol{\theta}|\mathcal{M}) \tag{5.54}$$

is maximized

$$(\hat{\boldsymbol{f}}, \hat{\boldsymbol{v}}_\zeta, \hat{\boldsymbol{\theta}}) = \underset{(\boldsymbol{f}, \boldsymbol{v}_\zeta, \boldsymbol{\theta})}{\arg\max} \{p(\boldsymbol{f}, \boldsymbol{v}_\zeta, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\} \tag{5.55}$$

by the alternate optimization [CMDGP17, WMDG17]

$$\begin{cases} \boldsymbol{f}^{(t+1)} = \arg\max_{\boldsymbol{f}} \left\{p(\boldsymbol{f}, \boldsymbol{v}_\zeta^{(t)}, \boldsymbol{\theta}^{(t)}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right\} & \text{(5.56a)} \\ \boldsymbol{v}_\zeta^{(t+1)} = \arg\max_{\boldsymbol{v}_\zeta} \left\{p(\boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\zeta, \boldsymbol{\theta}^{(t)}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right\} & \text{(5.56b)} \\ \boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \left\{p(\boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\zeta^{(t+1)}, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right\} & \text{(5.56c)} \end{cases}$$

In section 5.2, using the error-splitting forward model, step (5.36c) reads

$$\boldsymbol{f} \leftarrow \underset{\boldsymbol{f}}{\arg\min} \left\{ \frac{1}{2} \|\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\xi} - \ln\left(p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M})\right) \right\} \tag{5.57}$$

while step (5.56a) with usual forward model (5.1) has the form

$$\boldsymbol{f} \leftarrow \underset{\boldsymbol{f}}{\arg\min} \left\{ \frac{1}{2} \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta} - \ln\left(p(\boldsymbol{f}|\boldsymbol{\theta}; \mathcal{M})\right) \right\}, \tag{5.58}$$

so minimization (5.57) can be performed in the same way as (5.58) by replacing $\boldsymbol{g}$ by $\boldsymbol{g}_0$, and $\boldsymbol{v}_\zeta$ by $\boldsymbol{v}_\xi$. The update of hyperparameters $\boldsymbol{\theta}$ is done as in section 5.2. Hence, alternate optimizations (5.36) or (5.56) only differ by the estimation of $\boldsymbol{g}_0$, $\boldsymbol{v}_\epsilon$ and $\boldsymbol{v}_\xi$ with the error-splitting forward model, and by the estimation of $\boldsymbol{v}_\zeta$ with the usual forward model. As we have seen in section 5.2, the estimations of $\boldsymbol{g}_0$, $\boldsymbol{v}_\epsilon$ and $\boldsymbol{v}_\xi$ have a very little computational cost since they are performed by analytical formulae thanks to the use of conjugate priors. Therefore, especially when a conjugate prior is assigned to $\boldsymbol{v}_\zeta$ (such as Inverse-Gamma prior as in chapter 4), the cost of one global iteration of JMAP when using the error-splitting forward model is approximately the same as the one with usual forward model (5.1), for any prior model on the volume.

One drawback of our error-splitting approach is the memory cost. The error-splitting forward model introduces variables $\boldsymbol{g}_0$, $\boldsymbol{v}_\epsilon$ and $\boldsymbol{v}_\xi$ in place of variances $\boldsymbol{v}_\zeta$ in the usual forward model. These variables are all the size of projections $\boldsymbol{g}$, which can be very large. In order to save memory usage, one way is to assign only one variance v_{cell_c} to all measurement uncertainties related to cell c of the detector :

$$v_{\epsilon_i} = v_{cell_c}, \quad \text{for all ray } \ i \ \text{ hitting cell } \ c, \tag{5.59}$$

so variances of measurement uncertainties are only size of an image with $N_u \times N_v$ pixels. This is what we do in our experiments in section 5.5. By this way, the variances of the measurement uncertainties are made independent from the projection angle, which is an approximation according to our derivation of the error-splitting forward model in section 5.1.1. Nevertheless, compared to the case where we have one v_{ϵ_i} per ϵ_i, we did not notice troubles due to this memory saving.

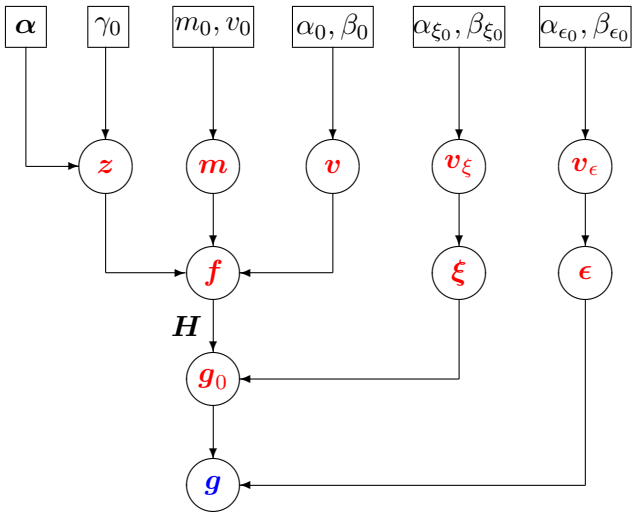


Figure 5.4: Hierarchical model combining the error-splitting forward model and Gauss-Markov-Potts prior on the volume

5.4 Combination of the error-splitting forward model with Gauss-Markov-Potts prior on the volume

We now combine our error-splitting forward model with Gauss-Markov-Potts prior presented in chapter 4. Figure 5.4 shows the hierarchical model merging figures 4.2 and 5.3. Given the joint posterior distribution of the unknowns

$$\begin{aligned} p\left(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}\right) \propto & p(\boldsymbol{g}|\boldsymbol{g}_0, \boldsymbol{v}_\epsilon) p(\boldsymbol{g}_0|\boldsymbol{f}, \boldsymbol{v}_\xi)\, p(\boldsymbol{v}_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0})\, p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v}) \\ & p(\boldsymbol{v}_\xi|\alpha_{\xi_0}, \beta_{\xi_0})\, p(\boldsymbol{z}|\boldsymbol{\alpha}; \gamma_0)\, p(\boldsymbol{m}|m_0, v_0)\, p(\boldsymbol{v}|\alpha_0, \beta_0), \end{aligned} \tag{5.60}$$

the joint maximization a posteriori

$$(\hat{\boldsymbol{g}}_0, \hat{\boldsymbol{v}}_\epsilon, \hat{\boldsymbol{f}}, \hat{\boldsymbol{v}}_\xi, \hat{\boldsymbol{z}}, \hat{\boldsymbol{m}}, \hat{\boldsymbol{v}}) = \underset{(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v})}{\arg\max} \left\{p\left(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon, \boldsymbol{f}, \boldsymbol{v}_\xi, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M}\right)\right\}, \tag{5.61}$$

is performed as described in section 5.2 :

$$\begin{cases} \boldsymbol{g}_0^{(t+1)} = \arg\max_{\boldsymbol{g}_0} \left\{p(\boldsymbol{g}_0, \boldsymbol{v}_\epsilon^{(t)}, \boldsymbol{f}^{(t)}, \boldsymbol{v}_\xi^{(t)}, \boldsymbol{z}^{(t)}, \boldsymbol{m}^{(t)}, \boldsymbol{v}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62a)} \\ \boldsymbol{v}_\epsilon^{(t+1)} = \arg\max_{\boldsymbol{v}_\epsilon} \left\{p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon, \boldsymbol{f}^{(t)}, \boldsymbol{v}_\xi^{(t)}, \boldsymbol{z}^{(t)}, \boldsymbol{m}^{(t)}, \boldsymbol{v}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62b)} \\ \boldsymbol{f}^{(t+1)} = \arg\max_{\boldsymbol{f}} \left\{p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}, \boldsymbol{v}_\xi^{(t)}, \boldsymbol{z}^{(t)}, \boldsymbol{m}^{(t)}, \boldsymbol{v}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62c)} \\ \boldsymbol{v}_\xi^{(t+1)} = \arg\max_{\boldsymbol{v}_\xi} \left\{p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\xi, \boldsymbol{z}^{(t)}, \boldsymbol{m}^{(t)}, \boldsymbol{v}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62d)} \\ \boldsymbol{z}^{(t+1)} = \arg\max_{\boldsymbol{z}} \left\{p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\xi^{(t+1)}, \boldsymbol{z}, \boldsymbol{m}^{(t)}, \boldsymbol{v}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62e)} \\ \boldsymbol{m}^{(t+1)} = \arg\max_{\boldsymbol{m}} \left\{p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\xi^{(t+1)}, \boldsymbol{z}^{(t+1)}, \boldsymbol{m}, \boldsymbol{v}^{(t)} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62f)} \\ \boldsymbol{v}^{(t+1)} = \arg\max_{\boldsymbol{v}} \left\{p(\boldsymbol{g}_0^{(t+1)}, \boldsymbol{v}_\epsilon^{(t+1)}, \boldsymbol{f}^{(t+1)}, \boldsymbol{v}_\xi^{(t+1)}, \boldsymbol{z}^{(t+1)}, \boldsymbol{m}^{(t+1)}, \boldsymbol{v} \mid \boldsymbol{g}; \mathcal{S}, \mathcal{M})\right\} & \text{(5.62g)} \end{cases}$$

The criterion minimized to update the volume, corresponding to (5.47), is

$$J_f(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\xi} + \frac{1}{2}\|\boldsymbol{f} - \boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}} \tag{5.63}$$

where, like in chapter 4, $m_{z_j} = m_k$ and $v_{z_j} = v_k$ if $z_j = k$, and $\boldsymbol{V}_{\boldsymbol{z}} = \text{diag}\left[\boldsymbol{v}_{\boldsymbol{z}}\right]$. The minimization is done by a gradient descent similar to the one presented in section 4.3.1.

For Gauss-Markov-Potts prior model, the hyperparameters are $\boldsymbol{\theta} = (\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v})$. As we have seen, their update do not depend on the used forward model. Consequently, $\boldsymbol{z}$, $\boldsymbol{m}$ and $\boldsymbol{v}$ are updated as done in chapter 4. The labels are updated by running few iterations of Iterated Conditional Mode (ICM) algorithm [Bes86] which maximizes Potts energy (4.40). Next, the updating formulae for $\boldsymbol{m}$ and $\boldsymbol{v}$ are the same as (4.47) and (4.51) derived in chapter 4 :

$$\hat{m}_k = \frac{\frac{m_0}{v_0} + \frac{1}{v_k}\sum_{j=1}^{N} f_j \delta(z_j - k)}{\frac{1}{v_0} + \frac{N_k}{v_k}}, \forall k \tag{5.64}$$

and

$$\hat{v}_k = \frac{\beta_0 + \frac{1}{2}\sum_{j=1}^{N}(f_j - m_k)^2 \delta(z_j - k)}{\alpha_0 + \frac{N_k}{2} + 1}, \forall k \tag{5.65}$$

where N_k is the number of voxels in class k. The full MBIR algorithm combining Gauss-Markov-Potts prior and the error-splitting forward model is given in algorithm 14 and is simply an instance of algorithm 13. Initial labels $\boldsymbol{z}^{(0)}$, means $\boldsymbol{m}^{(0)}$ and variances $\boldsymbol{v}^{(0)}$ are computed as explained in section 4.3.6 with the usual forward model. The strategies to fix the parameters related to Gauss-Markov-Potts prior : K, γ_0, $(\alpha_k)_k$, m_0, v_0, α_0 and β_0, are the same as presented in section 4.3.6.

Algorithm 14 JMAP algorithm combining the error-splitting forward model and Gauss-Markov-Potts prior

Initialize $\boldsymbol{v}_\epsilon^{(0)}$, $\boldsymbol{f}^{(0)}$, $\boldsymbol{v}_\xi^{(0)}$, $\boldsymbol{z}^{(0)}$, $\boldsymbol{m}^{(0)}$ and $\boldsymbol{v}^{(0)}$
for $t = 0, \ldots, t_{max} - 1$ **do**
1 : Update $\boldsymbol{g}_0$ by (5.41)
2 : Update $\boldsymbol{v}_\epsilon$ by (5.45)
3 : Update $\boldsymbol{f}$ by minimizing criterion (5.63) through n_{vol} iterations of gradient descent (4.30)
4 : Update $\boldsymbol{v}_\xi$ by (5.50)
5 : Update $\boldsymbol{z}$ by maximizing Potts energy (4.40) through n_{seg} iterations of ICM algorithm [Bes86]
6 : Update $\boldsymbol{m}$ by (5.64)
7 : Update $\boldsymbol{v}$ by (5.65)
end for

5.5 Results on real data

We evaluate the error-splitting forward model combined with Gauss-Markov-Potts prior by applying algorithm 14 to real data. We compare the results with those obtained with usual forward model (5.1). Through the iterations of each reconstruction algorithm, we observe the evolution of *Root Mean Square Difference* (RMSD) :

$$RMSD = \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left(f_j - f_j^* \right)^2} \tag{5.66}$$

between estimation $\boldsymbol{f}$ and a reference volume $\boldsymbol{f}^*$. We run each algorithm during 50 iterations.

The used pair of projector and backprojector is the unmatched RD/VD pair presented in chapter 3. We did not notice troubles due to the mismatching of the pair, and the algorithm have converged in all the experiments. As in chapter 4, the segmentation step, corresponding to step 5 of algorithm 14, is performed on the CPU.

5.5.1 Framework for the experiments

In order to use usual forward model (5.1), we need to assign a prior to variances $\boldsymbol{v}_\zeta$ of total uncertainties $\boldsymbol{\zeta}$: to do simple calculations, we consider a conjugate prior, which is Inverse-Gamma :

$$p(v_{\zeta_i} | \alpha_{\zeta_0}, \beta_{\zeta_0}) = \mathcal{IG}(v_{\zeta_i} | \alpha_{\zeta_0}, \beta_{\zeta_0}) \tag{5.67}$$

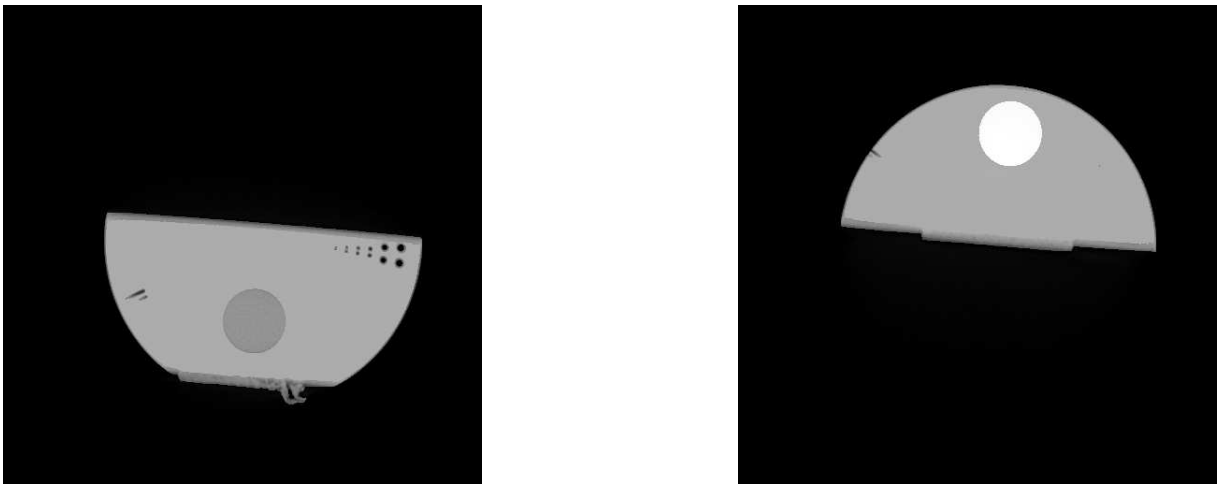
with parameters $(\alpha_{\zeta_0}, \beta_{\zeta_0})$, as done in chapter 4. With this prior, step (5.56b) presented in section 5.3 is done similarly to updates (5.36b) and (5.36d). In addition to chapter 4, we see that

$$\begin{cases} p(\zeta_i | v_{\zeta_i}) = \mathcal{N}(\zeta_i | 0, v_{\zeta_i}) \\ p(v_{\zeta_i} | \alpha_{\zeta_0}, \beta_{\zeta_0}) = \mathcal{IG}(v_{\zeta_i} | \alpha_{\zeta_0}, \beta_{\zeta_0}) \end{cases}, \tag{5.68}$$

so the distribution of ζ_i marginalized with respect to v_{ζ_i} is a generalized Student-t. As a result, two ways are possible to fix α_{ζ_0} and β_{ζ_0}. One way is to model total uncertainties as Gaussian, and, in this case, parameters α_{ζ_0} and β_{ζ_0} are fixed similarly to α_{ϵ_0} and β_{ϵ_0} in section 5.1.4 for measurement uncertainties $\boldsymbol{\epsilon}$. Another way is to enforce robustness with respect to outliers in projections $\boldsymbol{g}$, so a heavy-tailed prior is more appropriate for $\boldsymbol{\zeta}$. In this case, α_{ζ_0} and β_{ζ_0} are fixed similarly to α_{ξ_0} and β_{ξ_0} in section 5.1.4 for linear model uncertainties $\boldsymbol{\xi}$. This interpretation of α_{ζ_0} and β_{ζ_0} is a bit more precise compared to the analysis we made in section 4.3.6.

5.5.2 Evaluation of the accuracy : Image Quality Indicator (IQI) volume

We first evaluate the accuracy of the error-splitting forward model on the Image Quality Indicator (IQI) volume already used in section 4.5.1 [GARC16]. The field-of-view is sampled into $512 \times 512 \times 256$ voxels. We still use 300 projections uniformly distributed over $[0, 2\pi]$. The detector is sampled into 512×256 pixels. The source-to-rotation-center distance is 975 mm, and the source-to-detector distance is 1300 mm. Reference volume is reconstructed from



(a) (b)

Figure 5.5: Reference for IQI, reconstructed from 2400 projections (algorithm 12 in chapter 4) (bottom (a) and top (b))

2400 projections by algorithm 12 presented in chapter 4, and is shown in figure 5.5. To tune JMAP, we consider that the SNR is 20 db, which is a realistic value. We set $\beta_{\epsilon_0} = 1$ to ensure that constraint (5.28) is fullfilled. The parameters of the error-splitting forward model for IQI reconstruction are summarized in table 5.1. They are fixed according to the strategy explained in section 5.1.4.

Parameters	SNR	α_{ϵ_0}	β_{ϵ_0}	α_{ξ_0}	β_{ξ_0}
Values	20	Use (5.31)	1	0.01	0.0001

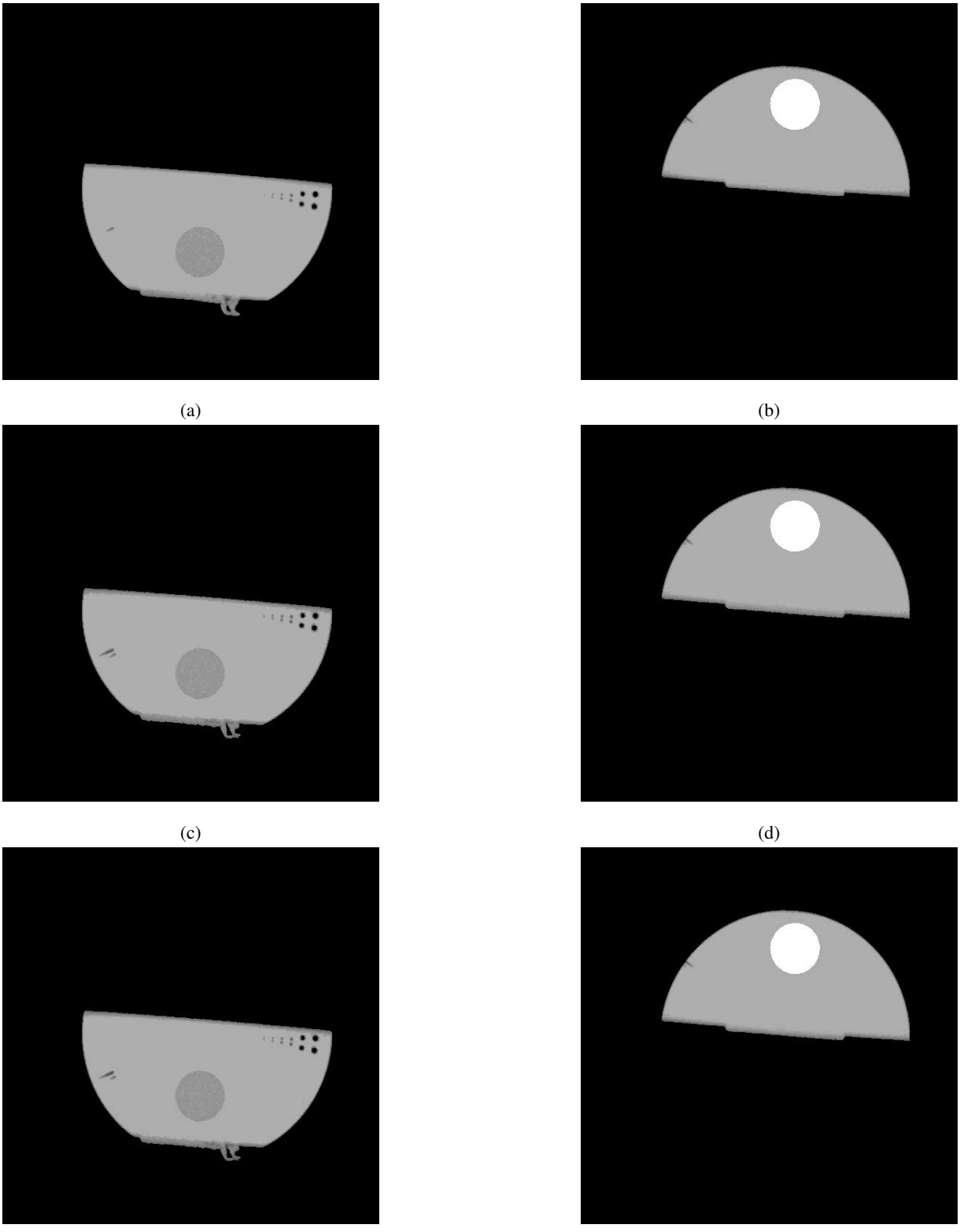
Table 5.1: Parameters of the error-splitting forward model for IQI reconstruction

In order to compare the tested forward models, the values of the parameters of the Gauss-Markov-Potts prior on the volume remain the same whatever the forward model is. These values are summarized in table 5.2 and are the same as the ones used in chapter 4.

Parameters	K	γ_0	v_0	α_0	β_0
Fixed values	4	3	1	5	0.01

Table 5.2: Parameters for Gauss-Markov-Potts prior model on IQI volume

The reconstructions of IQI volume by JMAP with Gauss-Markov-Potts prior are shown in figure 5.6. As we see, the reconstructions obtained with the different forward models look the same and are similar to the one obtained in chapter 4 : SSIM indices [WBSS04] computed between these reconstructions are approximately 1. The reconstructions by JMAP have compact and homogeneous regions thanks to the use of Gauss-Markov-Potts prior model. For each forward model, the segmentations obtained jointly with the reconstruction are shown in figure 5.7 : as we see, they are the same. To compare the accuracy of each reconstruction, as in chapter 4, we focus on the little holes of the volume. We show their profiles in figure 5.8, with the ones of the reference and FDK and TV reconstructions. FDK and TV reconstructions are respectively given in figures 4.13 and 4.14 in chapter 4. We observe that JMAP with Gauss-Markov-Potts prior enhances the contrast, making transitions between holes much sharper than FDK and TV. For the two biggest holes, whatever the used forward model is, JMAP reaches the lowest value and is better than FDK and TV. Nevertheless, this value is slightly under the theoretical value of 0 due to the fact that no non-negativity constraint is enforced in our algorithm. For the two intermediate holes, TV-reconstruction's profile goes lower than JMAP. For these holes, the profile is the same for JMAP reconstructions whatever the forward model is. For the two most little holes, the accuracy is quite similar for JMAP with the error-splitting forward model, JMAP with the usual



(a) (b)

(c) (d)

(e) (f)

Figure 5.6: Reconstructions from 300 projections of IQI volume by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (bottom (a) and top (b)), the usual forward model and modelling ζ as Gaussian (bottom (c) and top (d)), and the error-splitting forward model (bottom (e) and top (f))

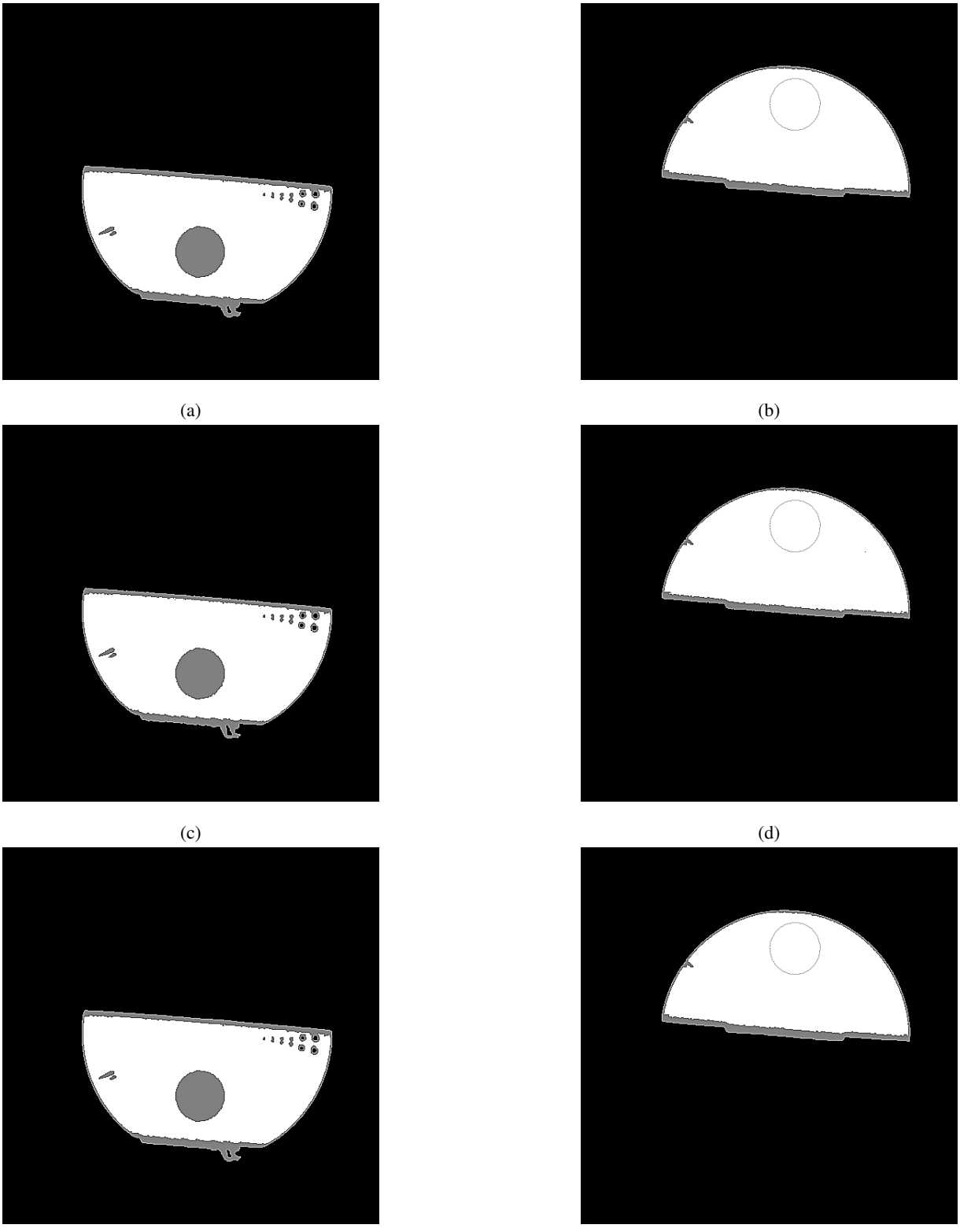


Figure 5.7: Segmentations obtained jointly with the reconstruction from 300 projections of IQI volume by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (bottom (a) and top (b)), the usual forward model and modelling ζ as Gaussian (bottom (c) and top (d)), and the error-splitting forward model (bottom (e) and top (f))

forward model modelling $\boldsymbol{\zeta}$ as Gaussian, and TV-reconstruction. The accuracy is lower for JMAP with the usual forward model modelling $\boldsymbol{\zeta}$ as heavy-tailed. Hence, we see that the error-splitting forward model does not imply a loss of accuracy, and is even better than the usual forward model for a certain set of parameters.

We analyze the convergence of JMAP with each forward model. For the error-splitting forward model, the criterion (5.37) of algorithm 14 converges to a maximum value as shown in figure 5.9. Computation times are given in table 5.3 : because the error-splitting forward model adds very few operations, the reconstruction times for JMAP with each forward model are similar and are approximately 30 minutes. For volume update step at each iteration of JMAP, $n_{vol} = 10$ sub-iterations are performed. Hence, because PDFW does not have sub-iterations and because we run JMAP during 50 iterations, TV-reconstruction is obtained in 500 iterations. The reconstruction time for TV is approximately 20 minutes. The computation time for JMAP is larger than the one for TV due to the segmentation step done on the CPU. We show the evolution of the RMSD with respect to reference volume in figure 5.10 for TV and in figure 5.11 for JMAP. Compared to TV, in table 5.3, we see that a lower value for the RMSD is reached for JMAP with Gauss-Markov-Potts prior, whatever the forward model is. Nevertheless, in figure 5.11, we see that the RMSD reaches its lowest value with the error-splitting forward model. Thus, for this reconstruction with Gauss-Markov-Potts prior, JMAP with the error-splitting forward model is more accurate than JMAP with usual forward model (5.1).

Forward model	Usual	Usual	Usual	Error-splitting
Prior model	TV	Gauss-Markov-Potts	Gauss-Markov-Potts	Gauss-Markov-Potts
Algorithm	PDFW	JMAP-Heavy-tailed	JMAP-Gaussian	JMAP-Error-splitting
Computation time	**1180 s**	1876 s	1777 s	1854 s
Final value of RMSD (/m)	22.10×10^{-4}	9.09×10^{-4}	8.86×10^{-4}	$\mathbf{8.78 \times 10^{-4}}$

Table 5.3: Performance comparison on IQI reconstruction from 300 projections, of PDFW and JMAP with Gauss-Markov-Potts prior model and : the usual forward model modelling $\boldsymbol{\zeta}$ as heavy-tailed, the usual forward model modelling $\boldsymbol{\zeta}$ as Gaussian, and the error-splitting forward model

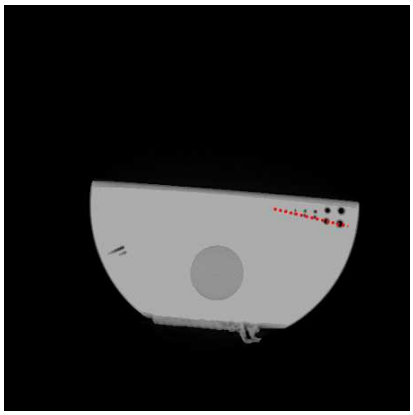
5.5.3 Evaluation of the robustness : Composite Material with Titanium on the Edges (CMTE) and metallic part

We now evaluate the robustness of the error-splitting forward model on challenging data with metal. First, we test algorithm 14 on Composite Material with Titanium on the Edges (CMTE) used in chapter 4. The field-of-view is sampled into 512^3 voxels. The reconstruction is performed with 300 projections uniformly distributed over $[0, 2\pi]$. The detector is sampled into 512^2 pixels. The source-to-rotation-center distance is 432 mm, and the source-to-detector distance is 807 mm. Reference volume is reconstructed from 1500 projections by algorithm 12 presented in chapter 4 : it is shown in figure 5.12. As we see, even with a lot of projections, scattering remains around the titanium edges. This scattering is hard to remove because the values for scattering are near the values for the composite material. The parameters of the error-splitting forward model are summarized in table 5.4. Because we have a lot of absorption and scattering due to the presence of titanium, we fix $\alpha_{\xi_0} < \beta_{\xi_0} < 1$, as explained in section 5.1.4. The values of the parameters for Gauss-Markov-Potts prior model on CMTE volume are presented in table 5.5 and are the same as the ones used in chapter 4.

Parameters	SNR	α_{ϵ_0}	β_{ϵ_0}	α_{ξ_0}	β_{ξ_0}
Values	20	Use (5.31)	1	0.01	0.1

Table 5.4: Parameters of the error-splitting forward model for CMTE reconstruction

Because of high beam-hardening, the number of sub-iterations for volume update in JMAP has to be fixed carefully in order to avoid local optima : at each iteration of JMAP, $n_{vol} = 2$ sub-iterations are performed for volume



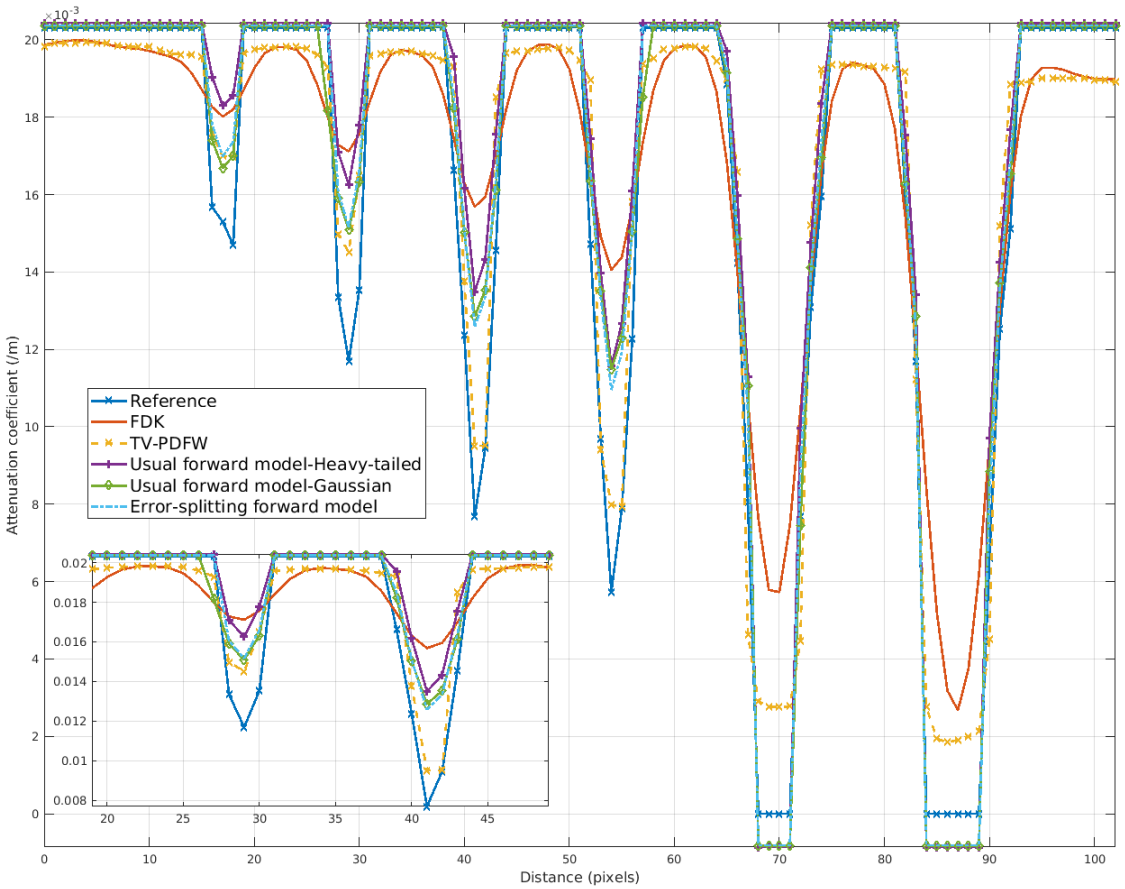


Figure 5.8: Accuracy : profiles of the holes of IQI volume for each reconstruction, with Gauss-Markov-Potts prior for JMAP

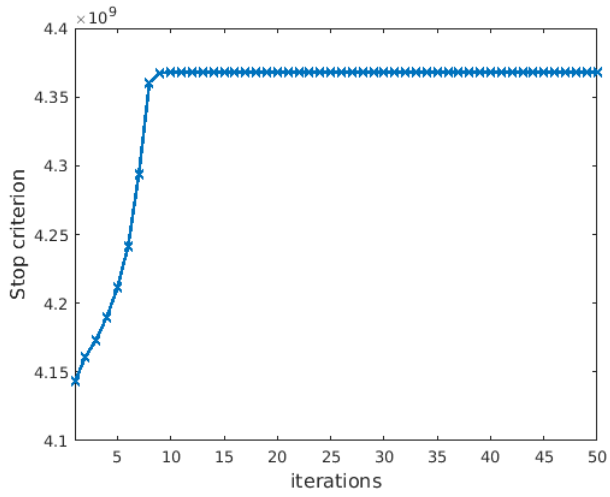


Figure 5.9: Convergence for JMAP with the error-splitting forward model and Gauss-Markov-Potts prior for IQI reconstruction from 300 projections

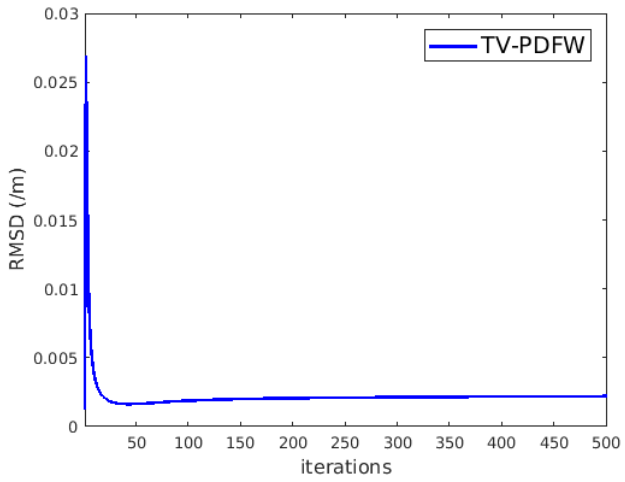


Figure 5.10: Convergence to the reference volume for IQI reconstruction by PDFW [OMBF18]

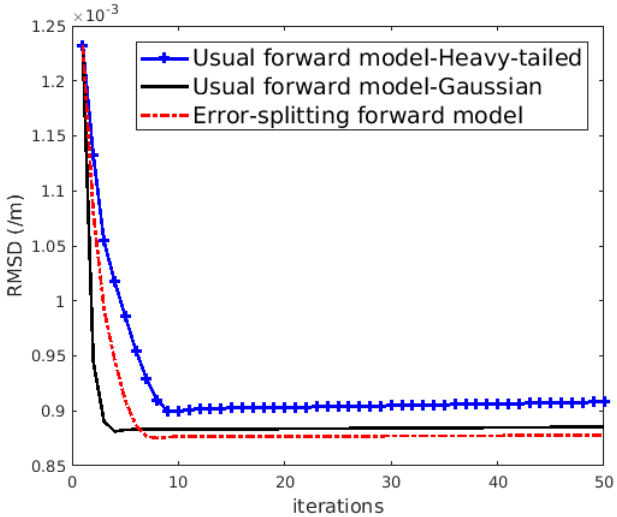


Figure 5.11: Convergence to the reference volume for IQI reconstruction by JMAP from 300 projections with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model

Parameters	K	γ_0	v_0	α_0	β_0
Fixed values	3	3	1	5	0.01

Table 5.5: Parameters for Gauss-Markov-Potts prior model on the CMTE volume

Forward model	Usual	Usual	Usual	Error-splitting
Prior model	TV	Gauss-Markov-Potts	Gauss-Markov-Potts	Gauss-Markov-Potts
Algorithm	PDFW	JMAP-Heavy-tailed	JMAP-Gaussian	JMAP-Error-splitting
Computation time	**2352 s**	5074 s	5359 s	5496 s
Final value of RMSD (/m)	43.383×10^{-3}	$\mathbf{1.765 \times 10^{-3}}$	8.167×10^{-3}	1.785×10^{-3}

Table 5.6: Performance comparison on CMTE reconstruction from 300 projections, of TV-PDFW and JMAP with Gauss-Markov-Potts prior model and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model

estimation. We run JMAP during 50 iterations. Hence, TV-reconstruction is obtained in 100 iterations. Joint reconstructions and segmentations of the CMTE volume by JMAP with Gauss-Markov-Potts prior are shown in figure 5.13. Figure 5.14 shows the convergence of the criterion of JMAP with the error-splitting forward model. In figure 5.13, we see that the reconstructions by JMAP still have scattering around the titanium. Because the values for scattering are very near the values for composite material, scattering is misclassified in the segmentation and is in the same class as the composite material. With the error-splitting forward model and with the usual forward model modelling ζ as heavy-tailed, JMAP is very close to the reference. This is not the case with the usual forward model modelling ζ as Gaussian : the reconstruction has very strong beam-hardening artifacts at the extremities of the titanium. We see in figure 5.16 that JMAP with the usual forward model modelling ζ as Gaussian diverges far from the reference volume. Indeed, because the prior on ζ is Gaussian, i.e. short-tailed, the algorithm tries to be too close to the data, which is not a good point, since high absorption is present in the data due to the titanium edges. The same phenomenon occurs for TV-reconstruction in which beam-hardening is also present, as shown in the profile in figure 5.17. In table 5.6, we see that TV-reconstruction reaches a higher value for the RMSD than JMAP.

On the opposite, JMAP with the error-splitting forward model and JMAP with the usual forward model modelling ζ as heavy-tailed manage to remove beam-hardening artifacts, and, as a result, converge to the reference volume, as shown by the evolution of the RMSD in figure 5.16. In table 5.6, we see that the final RMSD is practically the same for these reconstructions.

For the metallic part used in chapter 4, we obtain similar results. The field-of-view is sampled into 1024^3 voxels[1]. We use 500 projections uniformly distributed over $[0, 2\pi]$. The detector has 1024^2 cells. The source-to-detector distance is 1077 mm, and the source-to-object distance is 700 mm. The reference volume is reconstructed from 1500 projections by algorithm 12 in chapter 4 and shown in figure 5.18. The values for the parameters of the error-splitting forward model are the same than the ones for the CMTE volume in table 5.4. Those for the parameters of Gauss-Markov-Potts prior are the same used in chapter 4 and presented in table 4.8. The obtained joint reconstructions and segmentations are shown in figure 5.19. The reconstruction by JMAP with the error-splitting forward model and with the usual forward modelling ζ as heavy-tailed are of good quality, while the reconstruction with the usual forward model modelling ζ as Gaussian has strong scattering. As a result, in figure 5.20, the reconstructions with the error-splitting forward model and with the usual forward modelling ζ as heavy-tailed converge to the reference volume, while the reconstruction with the usual forward model modelling ζ as Gaussian diverge. In figure 5.21, we see that the criterion of JMAP with the error-splitting forward model reaches a maximum value.

From all our experiments reconstructing the CMTE volume and the metallic part with Gauss-Markov-Potts prior, we can draw the conclusion that JMAP with the error-splitting forward model is as robust as JMAP with the usual

[1]For confidentiality reasons, the metallic part is only partly shown in figures 5.18 and 5.19.

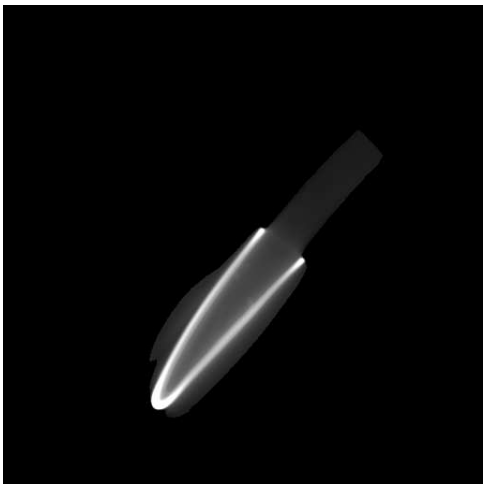


Figure 5.12: Reference for CMTE, reconstructed from 1500 projections (algorithm 12 in chapter 4) (middle slice)

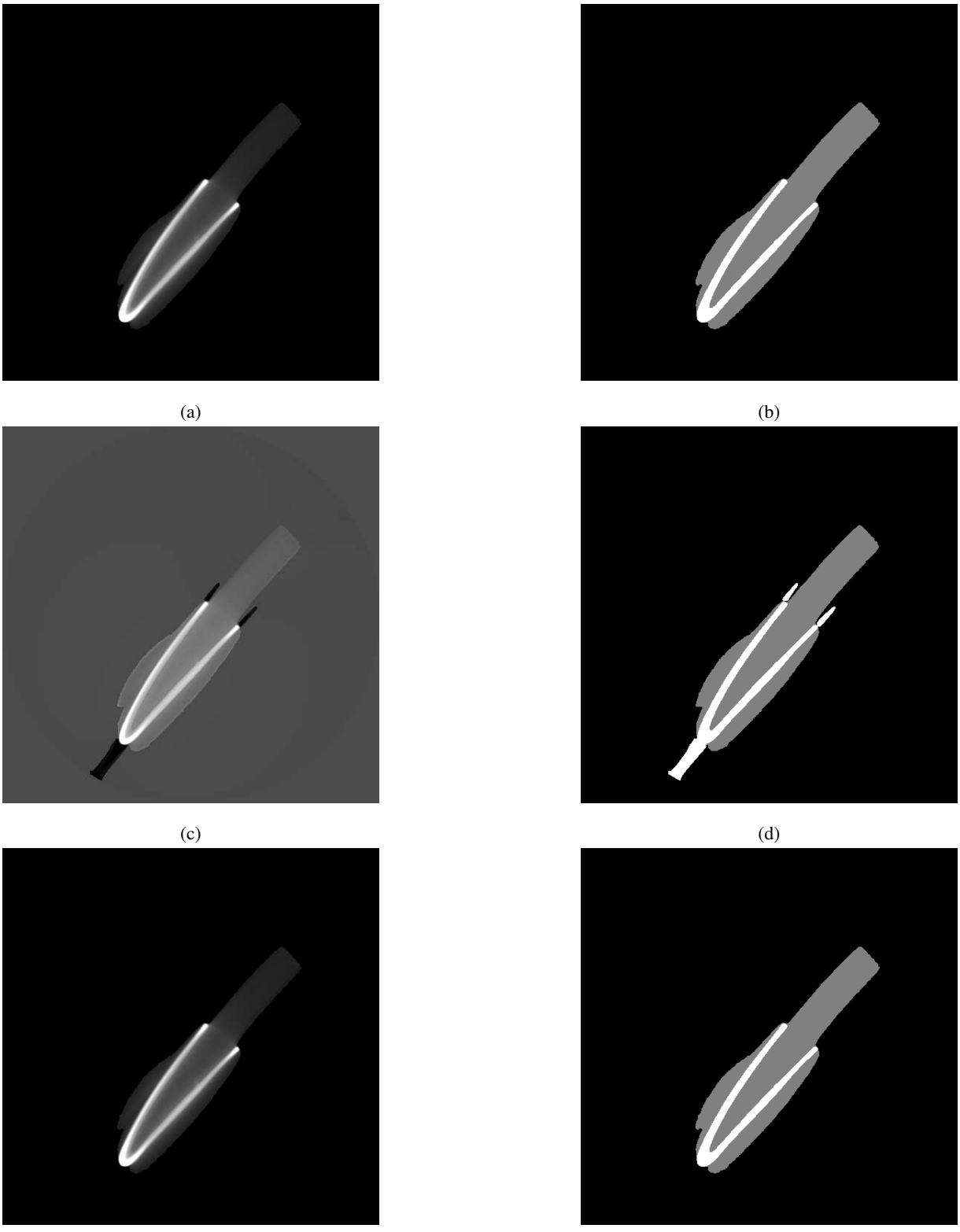
forward model modelling ζ as heavy-tailed. The interesting point is that, when evaluating the accuracy on IQI volume in the previous section, JMAP with the error-splitting forward model obtains the same accuracy than JMAP with the usual forward model modelling ζ as Gaussian, while the reconstruction is less accurate with heavy-tailed prior on ζ in the usual forward model. Indeed, because there are not high uncertainties on the projections for IQI volume, modelling ζ as short-tailed for JMAP with the usual forward model enables to be close to the data, and then to be more accurate. But, with the CMTE volume and the metallic part, being too close to the data leads to a failure in removing strong artifacts. For the error-splitting forward model, our better physical modelling leads to the best accuracy for IQI reconstruction, and, thanks to the heavy-tailed prior on linear model uncertainties, to robustness for the CMTE and the metallic part reconstructions. Hence, in our experiments with Gauss-Markov-Potts prior, JMAP with the error-splitting forward model combines accuracy and robustness, while, with the usual forward model, accuracy is only achieved by modelling ζ as Gaussian, and robustness only by modelling ζ as heavy-tailed.

5.5.4 Discussion

Our experiments have shown that the error-splitting forward model enables one to get robustness with heavy-tailed prior on linear model uncertainties $\boldsymbol{\xi}$, and accuracy with short-tailed prior on measurement uncertainties $\boldsymbol{\epsilon}$, while the usual forward model only enables to get one of these features. These results have been obtained with Gauss-Markov-Potts prior for industrial quasi-piecewise-constant volumes. Nevertheless, Gauss-Markov-Potts prior fails in reconstructing texture-rich details such as in woven composite materials. Since our derivation of the error-splitting forward model in section 5.1.1 is valid for any X-ray CT imaging task, future works will mainly focus on combining the error-splitting forward model with appropriate priors for textured, or more generally no longer piecewise-constant, volumes. In particular, a combination with TV-regularization will be interesting to study.

5.6 Conclusion and perspectives

Taking into account the error in the monochromatic model with respect to the polychromatic one, we have included a noise term in the mean of Poisson statistics of photon-count in 3D X-ray CT. From this addition, we have derived a new forward model for Model-Based Iterative Reconstruction (MBIR) methods, where two terms of uncertainties are present instead of one. This error-splitting forward model distinguishes what we called measurement uncertainties and linear model uncertainties. Measurement uncertainties appeared as Gaussian in our second order Taylor series expansion of the Poisson log-likelihood, while linear model uncertainties appeared as an unknown bias between measured and theroretical projections $\boldsymbol{g}$ and $\boldsymbol{Hf}$. Thanks to the use of generalized Student-t distribution, a heavy-tailed prior has been assigned to this bias in order to ensure robustness in the reconstruction process. We have given details about how to fix each parameter of the error-splitting forward model, so each uncertainty was modeled as we wished.



(a) (b)

(c) (d)

(e) (f)

Figure 5.13: Joint reconstructions and segmentations (middle slices) from 300 projections of the CMTE volume by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (reconstruction (a) and segmentation (b)), the usual forward model modelling ζ as Gaussian (reconstruction (c) and segmentation (d)), and the error-splitting forward model (reconstruction (e) and segmentation (f))

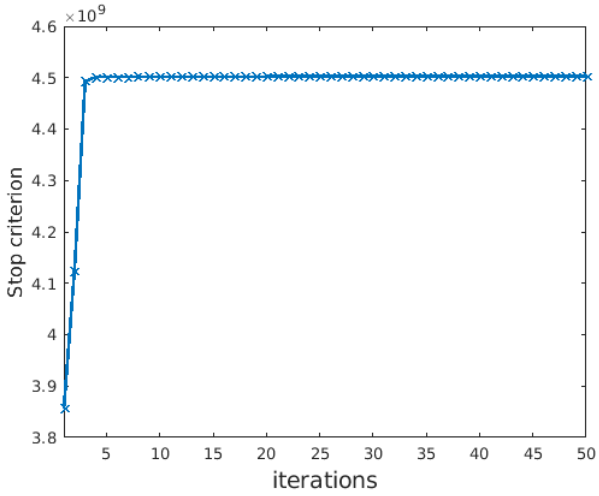


Figure 5.14: Convergence for JMAP with the error-splitting forward model and Gauss-Markov-Potts prior for CMTE reconstruction from 300 projections

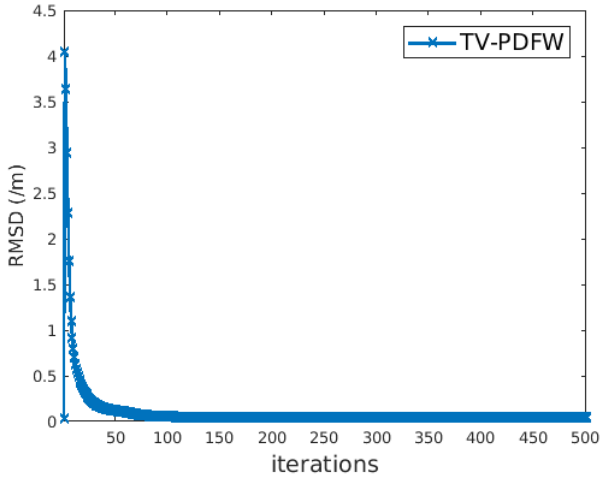


Figure 5.15: Convergence to the reference volume for CMTE reconstruction by PDFW [OMBF18]

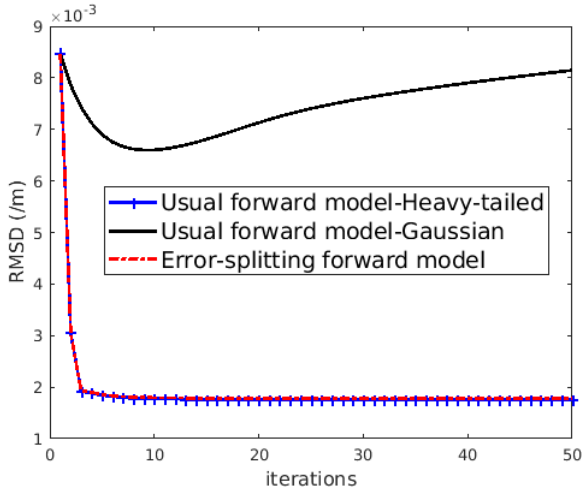
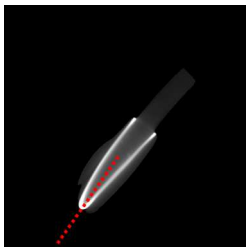


Figure 5.16: Convergence to the reference volume for CMTE reconstruction by JMAP from 300 projections with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model



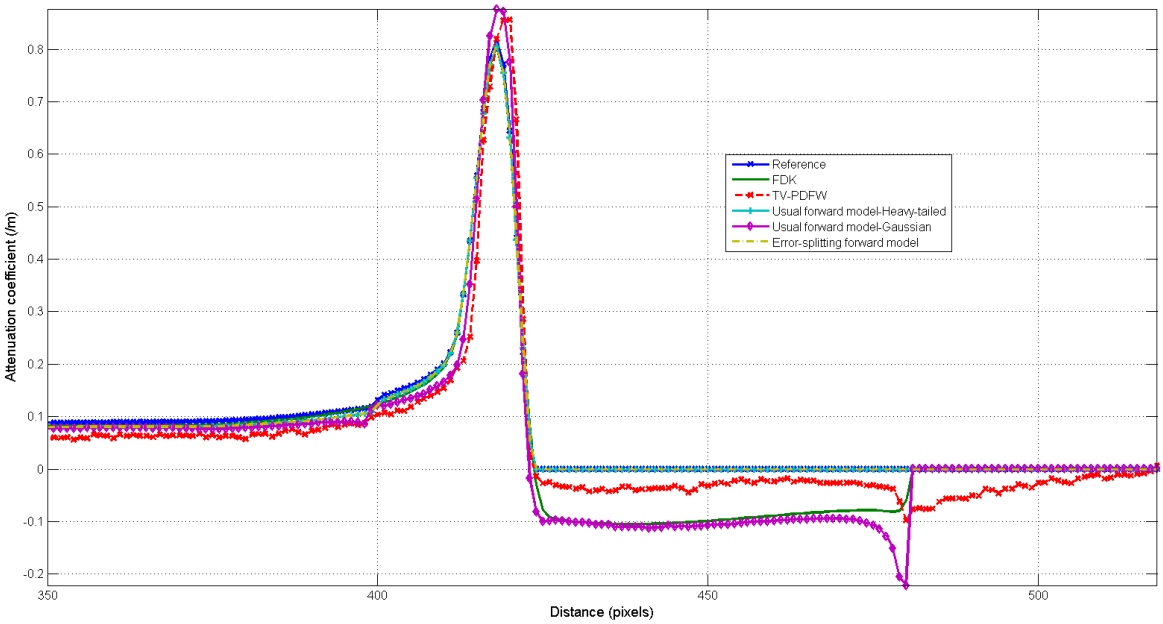


Figure 5.17: Profile of the CMTE volume for each reconstruction, with Gauss-Markov-Potts prior for JMAP

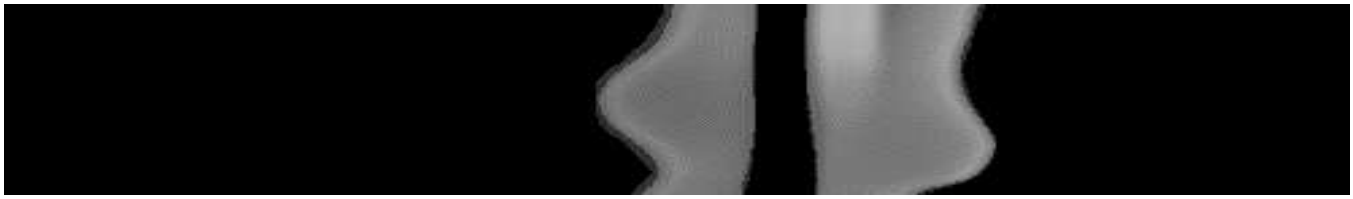
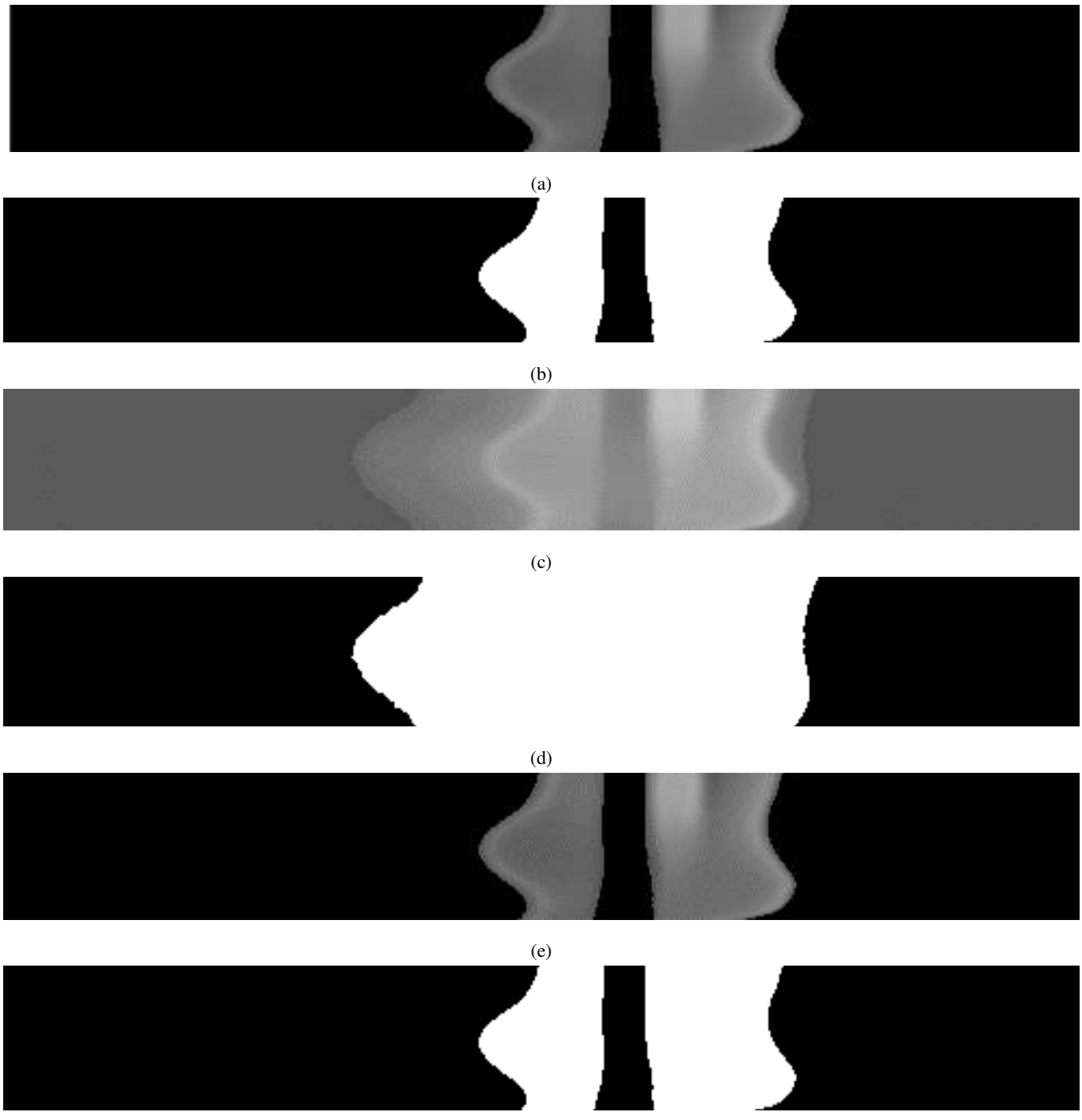


Figure 5.18: Reference for the metallic part, reconstructed from 1500 projections (algorithm 12 in chapter 4)



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.19: Joint reconstructions and segmentations from 500 projections of the metallic part by JMAP with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed (reconstruction (a) and segmentation (b)), the usual forward model and modelling ζ as Gaussian (reconstruction (c) and segmentation (d)), and the error-splitting forward model (reconstruction (e) and segmentation (f))

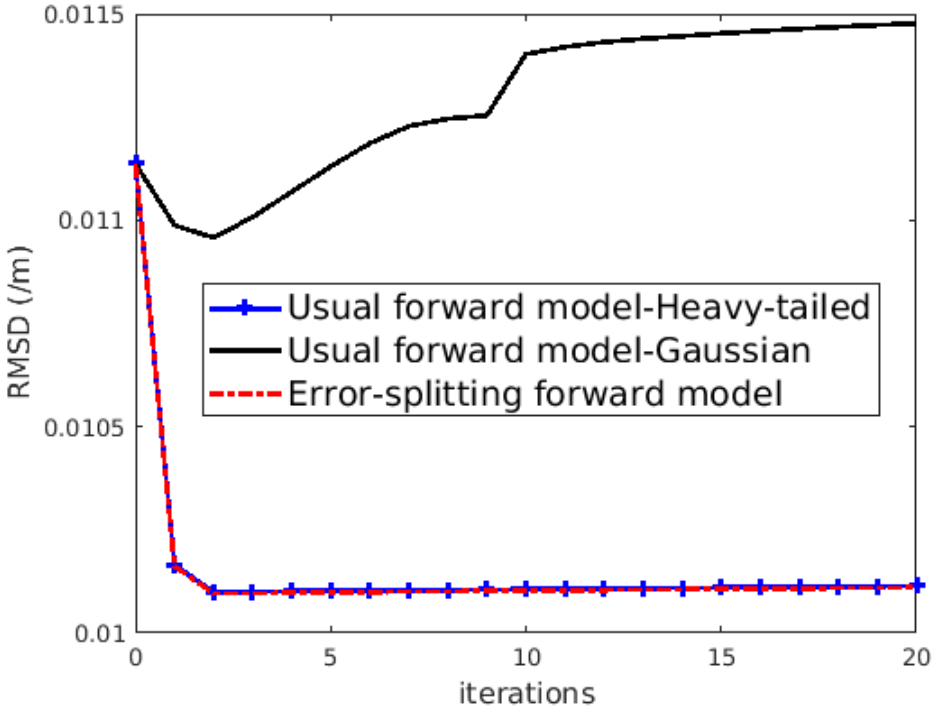


Figure 5.20: Convergence to the reference volume for the metallic part reconstruction by JMAP from 500 projections with Gauss-Markov-Potts prior and : the usual forward model modelling ζ as heavy-tailed, the usual forward model modelling ζ as Gaussian, and the error-splitting forward model

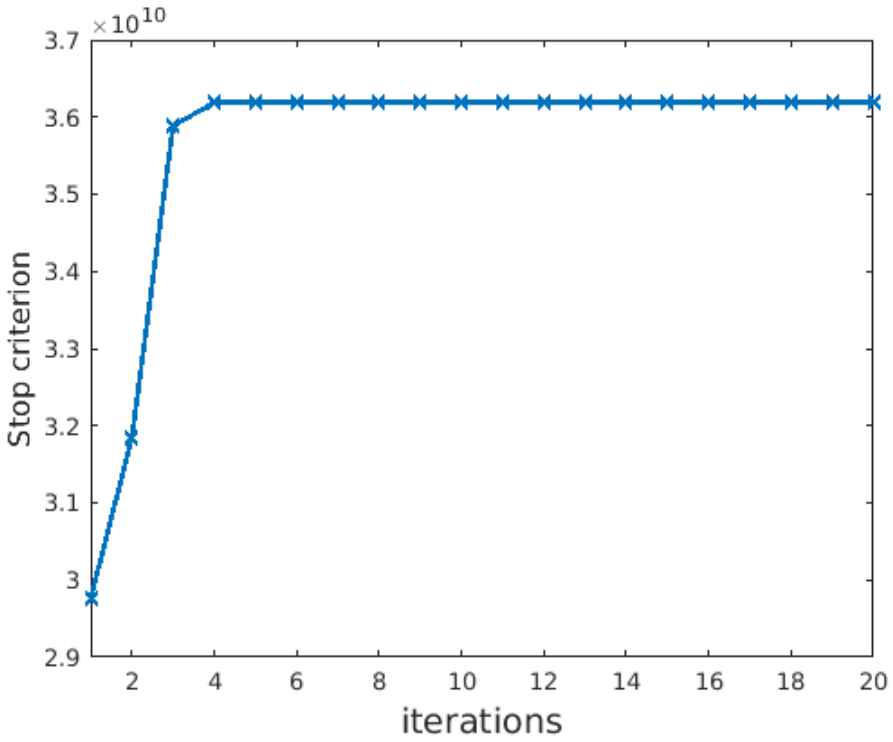


Figure 5.21: Convergence for JMAP with the error-splitting forward model and Gauss-Markov-Potts prior for the reconstruction of the metallic part from 500 projections

Compared to the usual forward model, the error-splitting forward model adds very few calculations thanks to the simplicity of generalized Student-t distribution. We have combined this new forward model and Gauss-Markov-Potts prior in order to reconstruct piecewise-constant volumes. In our experiments with real data, we have compared our new forward model with the usual one. Thanks to our better consideration of uncertainty, we have shown that, with Gauss-Markov-Potts prior, using the error-splitting forward model enables to obtain more accurate and more robust reconstructions, while only one of these two features is achievable with the usual forward model. Since our experiments have been done with Gauss-Markov-Potts prior model on the volume, in future works, the emphasis will be on the impact of the error-splitting forward model combined with other priors.

Particularly in order to tackle the strong scattering around the titanium edges of the CMTE volume, efforts will also be made to see how to improve the error-splitting forward model. For this purpose, we have emphasized in section 5.1.4 that a way could be to assign a different prior on the Signal-to-Noise Ratio (SNR) to each measurement uncertainty. Another possible way is to enhance the modelling of the linear model uncertainties. Indeed, in section 5.1.1, we have noticed that the prior distribution for these uncertainties cannot be deduced from our second-order Taylor expansion, and generalized Student-t distribution has been chosen due its properties to enforce robustness and to perform simple calculations. Nevertheless, as shown in figure 5.1, this distribution is over all real numbers, while, according to equation (5.12), linear model uncertainties are nonpositive, since we have $I_i \geqslant 0$ and $n_i \geqslant 0$, for all i. Consequently, an intuitively better prior on linear model uncertainties would be a heavy-tailed distribution only over nonpositive real numbers. In this case, a mixture on the variances between a Gaussian and another distribution could not be used anymore. As a result, a possible difficulty in this improvement for future works could be the increase of the computational cost.

Chapter 6

Towards the estimation of the uncertainties on the reconstruction by Variational Bayesian Approach

In chapters 4 and 5, we have presented MBIR methods providing reconstructions of enhanced quality compared to conventional filtered backprojection. These methods have the potential to be advantageously exploited at SAFRAN in order to better inspect and analyze its manufactured industrial parts. Due to the high dimension and to the fact that the reconstruction problem is ill-posed, we have seen that the exact estimation of the volume to inspect is not possible : as a result, our proposed algorithms, like other MBIR methods, only compute an approximate solution of the reconstruction problem. Consequently, knowing the uncertainties on the reconstructed estimation can be a desirable tool for NDT in post-reconstruction processing.

In [Fes96], an iterative method is proposed to estimate the uncertainties after the reconstruction has been performed. Nevertheless, its high computational cost makes it only applicable to a few voxels of interest [Fes96]. In [Per17], a method is detailed in order to compute the confidence regions on the Maximum A Posteriori (MAP) estimator, but this procedure is difficult to apply for discrete-continuous channels estimation, as it is our case for joint reconstruction and segmentation. In this chapter, we propose to compute Posterior Mean (PM) rather than MAP. In the framework of estimation theory, PM and MAP estimators are defined as the minimizers of distinct cost functions, as detailed in appendix E. For PM estimator, the uncertainties on the reconstruction correspond to the variances. Our algorithm estimates these variances jointly with the reconstruction based on variational Bayesian approach (VBA) [PSC+16].

We first present the principles of VBA. Then, we detail our reconstruction algorithm based on VBA, applied with a Gauss-Markov-Potts prior on the volume to reconstruct. In order to implement this algorithm, the main difficulty is the computation of the diagonal coefficients of the posterior covariance matrix, which are linked to projection and backprojection operators (P/BP) : we solve this issue thanks to the use of a matched pair which is here the Separable Footprint (SF) pair [LFB10] presented in chapter 3. We present simulation results and compare the obtained reconstruction with the one given by joint maximization a posteriori (JMAP) detailed in chapter 4. To the best of our knowledge, this work is the first attempt to apply VBA to a very general 3D inverse problem such as 3D X-ray CT.

6.1 Variational Bayesian Approach

In this chapter, the prior model $\mathcal{M}$ on the volume is a Gauss-Markov-Potts prior described in chapter 4. The usual forward model $\mathcal{U}$ with only one term of uncertainties :

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\zeta} \tag{6.1}$$

is considered instead of the error-splitting forward model for the sake of simplicity. We have seen in chapter 4 that the priors for the variances of the classes and the variances of the uncertainties are Inverse-Gamma distributions :

$$\begin{cases} p(v_k|\alpha_0, \beta_0) = \mathcal{IG}(v_k|\alpha_0, \beta_0), \forall k \in \{1, \dots, K\}, \\ p(v_{\zeta_i}|\alpha_{\zeta_0}, \beta_{\zeta_0}) = \mathcal{IG}(v_{\zeta_i}|\alpha_{\zeta_0}, \beta_{\zeta_0}), \forall i \in \{1, \dots, M\}. \end{cases} \tag{6.2}$$

In this chapter, we aim at computing the posterior mean. For an Inverse-Gamma distribution $\mathcal{IG}(.|\alpha,\beta)$, the mean is not defined for $\alpha < 1$. In order to avoid this constraint on α_0 and α_{ζ_0}, we rather consider the inverses of v_k and v_{ζ_i}, $\forall k, i$,

$$\begin{cases} \rho_k = \frac{1}{v_k} \\ \rho_{\zeta_i} = \frac{1}{v_{\zeta_i}} \end{cases}. \tag{6.3}$$

Given that v_k and v_{ζ_i}, $\forall k, i$, are Inverse-Gamma, ρ_k and ρ_{ζ_i} follow a Gamma distribution

$$\begin{cases} p(\rho_k|\alpha_0,\beta_0) = \mathcal{G}(\rho_k|\alpha_0,\beta_0), \forall k \in \{1,\dots,K\}, \\ p(\rho_{\zeta_i}|\alpha_{\zeta_0},\beta_{\zeta_0}) = \mathcal{G}(\rho_{\zeta_i}|\alpha_{\zeta_0},\beta_{\zeta_0}), \forall i \in \{1,\dots,M\}, \end{cases} \tag{6.4}$$

where

$$\mathcal{G}(\rho|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\rho_{\zeta_i}^{\alpha-1}\exp\left[-\beta\rho\right], \forall \rho > 0. \tag{6.5}$$

The rationale for this change of variables is that the mean of a Gamma distribution $\mathcal{G}(.|\alpha,\beta)$ is defined for all $\alpha > 0$ and $\beta > 0$. Hence, we have no constraint on α_0 and α_{ζ_0}.

After reparametrization (6.3), the unknowns to estimate are

$$\boldsymbol{\psi} = (\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}), \tag{6.6}$$

of which the posterior distribution is

$$\begin{aligned} p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M}) &= p(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\boldsymbol{g};\mathcal{U},\mathcal{M}) \\ &= \frac{p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho})p(\boldsymbol{\rho}_\zeta|\alpha_{\zeta_0},\beta_{\zeta_0})p(\boldsymbol{z}|\boldsymbol{\alpha},\gamma_0)p(\boldsymbol{m}|m_0,v_0)p(\boldsymbol{\rho}|\alpha_0,\beta_0)}{p(\boldsymbol{g}|\mathcal{U},\mathcal{M})}. \end{aligned} \tag{6.7}$$

According to chapter 4, we have

$$p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta) = (2\pi)^{-\frac{M}{2}}\det\left(\boldsymbol{V}_\zeta\right)^{-1/2}\exp\left[-\frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta}\right], \tag{6.8}$$

$$p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}) = (2\pi)^{-\frac{N}{2}}\det\left(\boldsymbol{V}_{\boldsymbol{z}}\right)^{-1/2}\exp\left[-\frac{1}{2}\|\boldsymbol{f}-\boldsymbol{m}_{\boldsymbol{z}}\|^2_{\boldsymbol{V}_{\boldsymbol{z}}}\right], \tag{6.9}$$

$$p(\boldsymbol{\rho}_\zeta|\alpha_{\zeta_0},\beta_{\zeta_0}) = \frac{\beta_{\zeta_0}^{\alpha_{\zeta_0}}}{\Gamma(\alpha_{\zeta_0})}\exp\left[\sum_{i=1}^{M}\left(((\alpha_{\zeta_0}-1)\ln\rho_{\zeta_i}-\beta_{\zeta_0}\rho_{\zeta_i})\right)\right], \tag{6.10}$$

$$p(\boldsymbol{z}|\boldsymbol{\alpha},\gamma_0) = \frac{1}{Z(\boldsymbol{\alpha},\gamma_0)}\exp\left[\sum_j\left(\sum_{k=1}^{K}\alpha_k\delta(z_j-k)+\gamma_0\sum_{i\in\mathcal{V}(j)}\delta(z_j-z_i)\right)\right] \tag{6.11}$$

$$p(\boldsymbol{m}|m_0,v_0) = (2\pi)^{-\frac{K}{2}}{v_0}^{-\frac{K}{2}}\exp\left[-\frac{1}{2v_0}\sum_{k=1}^{K}(m_k-m_0)^2\right], \tag{6.12}$$

$$p(\boldsymbol{\rho}|\alpha_0,\beta_0) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\exp\left[\sum_{k=1}^{K}\left((\alpha_0-1)\ln\rho_k-\beta_0\rho_k\right)\right], \tag{6.13}$$

where $v_{\zeta_i} = \rho_{\zeta_i}^{-1}$, $m_{z_j} = m_k$ and $v_{z_j} = \rho_k^{-1}$ if $z_j = k$, $\boldsymbol{V}_{\boldsymbol{z}} = \text{diag}\left[\boldsymbol{v}_{\boldsymbol{z}}\right]$ and $\boldsymbol{V}_\zeta = \text{diag}\left[\boldsymbol{v}_\zeta\right]$. The partition function $Z(\boldsymbol{\alpha},\gamma_0)$ of the labels is defined by formula (4.55) in chapter 4.

The posterior distribution (6.7) is too complex to have an analytical expression of its mean. A first way to get an approximation of the posterior mean is to generate samples of the posterior distribution by MCMC approaches such as Gibbs sampling [ADMD10, PDBT13, ZBKT16]. But, as said in chapter 4, these approaches are unaffordable in 3D for our industrial application. A way which seems more tractable in 3D for the calculation of the posterior mean is Variational Bayesian Approach (VBA), which consists in approximating the true posterior distribution p of

the unknowns by a distribution q which is simpler in the sense that the computation of its mean is analytically feasible [MDA09, PSC+16]. By this way, the mean of q obtained by VBA is known to be a good estimation of the true posterior mean [GBJ15, PSC+16]. The approximate posterior distribution q is calculated by solving the optimization problem

$$\min_{q \in \mathcal{Q}} KL(q(\boldsymbol{\psi})||p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})) \tag{6.14}$$

where $\mathcal{Q}$ is a subset of tractable distributions in which we know that we are able to compute the mean. In optimization problem (6.14), the best approximation of the posterior distribution $p(.|\boldsymbol{g};\mathcal{U},\mathcal{M})$ is defined as the distribution $q \in \mathcal{Q}$ which minimizes the Kullback-Leibler divergence [ŠQ06, PSC+16]

$$KL(q(\boldsymbol{\psi})||p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})) = \int_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \ln\left(\frac{q(\boldsymbol{\psi})}{p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})}\right) d\boldsymbol{\psi}. \tag{6.15}$$

The Kullback-Leibler divergence has all the properties of a distance, except that it is not symmetric, that is

$$KL(q(\boldsymbol{\psi})||p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})) \neq KL(p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})||q(\boldsymbol{\psi})) \tag{6.16}$$

in general. For this reason, $KL(q(\boldsymbol{\psi})||p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M}))$ is not said to be the Kullback-Leibler divergence between $q(\boldsymbol{\psi})$ and $p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M}))$: in order to take the non-symmetry into account, it is more appropriate to say that $KL(q(\boldsymbol{\psi})||p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M}))$ is the Kullback-Leibler divergence from $q(\boldsymbol{\psi})$ to $p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M}))$ [ŠQ06, Section 3.2.2].

More details about the Kullback-Leibler divergence can be found in [ŠQ06, Section 3.2.2]. In this chapter, we will use that

$$KL(q(\boldsymbol{\psi})||p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})) = \ln\left(p(\boldsymbol{g}|\mathcal{U},\mathcal{M})\right) - \mathcal{F}(q(\boldsymbol{\psi})) \tag{6.17}$$

where

$$\begin{aligned}\mathcal{F}(q(\boldsymbol{\psi})) &= \int_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \ln\left(\frac{p(\boldsymbol{g},\boldsymbol{\psi}|\mathcal{U},\mathcal{M})}{q(\boldsymbol{\psi})}\right) d\boldsymbol{\psi} \\ &= -\int_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \ln\left(q(\boldsymbol{\psi})\right) d\boldsymbol{\psi} + \int_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \ln\left(p(\boldsymbol{g},\boldsymbol{\psi}|\mathcal{U},\mathcal{M})\right) d\boldsymbol{\psi}\end{aligned} \tag{6.18}$$

is the negative free energy [ŠQ06, AMD10]. The quantity

$$\mathcal{H}(q(\boldsymbol{\psi})) = -\int_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \ln\left(q(\boldsymbol{\psi})\right) d\boldsymbol{\psi}. \tag{6.19}$$

is called the entropy of the approximate posterior distribution q. According to formula (6.17), since the log-evidence $\ln\left(p(\boldsymbol{g}|\mathcal{U},\mathcal{M})\right)$ is a constant, minimizing the Kullback-Leibler divergence is equivalent to maximizing the negative free energy.

The subset $\mathcal{Q}$ in which the approximate distribution must lie is chosen by the user. This subset has to make a tradeoff between several requirements. As already pointed out, first of all, $\mathcal{Q}$ must only contain distributions which are tractable in order to compute the mean. But, additionally, $\mathcal{Q}$ must exclude distributions which would be too gross approximations of the posterior distribution and which would give a very bad estimation of the posterior mean. Given these constraints, we describe two possible choices for $\mathcal{Q}$.

6.1.1 Mean Field Approximation

The simplest and the most common choice for $\mathcal{Q}$ is the subset of distributions q which are fully factorized in univariate distributions q_j :

$$q(\boldsymbol{\psi}) = \prod_j q_j(\psi_j). \tag{6.20}$$

Due to the full factorization, the correlations between variables $(\psi_j)_j$ are completely neglected in order to get a simple approximate distribution. As a result, for a variable ψ_j, the influence of the other variables on ψ_j is replaced by the mean of their influence : for this reason, this approximation is called the Mean Field Approximation (MFA) [Zha92].

Given all the other distributions $q_{j'}, j' \neq j$, the expression of q_j which minimizes the Kullback-Leibler divergence can be derived without the need of theorems of derivation under an integral [ŠQ06, Section 3.3.1]. By introducing $\boldsymbol{\psi}_{/j}$ which is $\boldsymbol{\psi}$ without its component ψ_j, and

$$q_{/j}(\boldsymbol{\psi}_{/j}) = \prod_{j' \neq j} q_{j'}(\psi_{j'}), \tag{6.21}$$

minimizing Kullback-Leibler divergence with respect to q_j leads to the expression [ŠQ06, Section 3.3.1]

$$q_j(\psi_j) \propto \exp\left(\mathbb{E}_{q_{/j}}\left(\ln\left(p(\boldsymbol{\psi}|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\right), \tag{6.22}$$

which is the formula used for q_j. We see that, through $q_{/j}$, this expression depends on the other distributions $q_{j'}, j' \neq j$. Consequently, in practice, VBA turns into an iterative algorithm in which each q_j is updated one after the other using formula (6.22), until the convergence of the negative free energy is achieved [QJ07, ADMD12, GBJ15, PSC$^+$16].

6.1.2 Partially factorized approximation

By neglecting all the correlations between the variables, the mean field approximation can be too gross in some cases [PSC$^+$16]. In particular, the experiments we conducted showed that the mean field approximation gives very bad results for our reconstruction problem with Gauss-Markov-Potts prior. In order to complexify the form of the approximate distribution, a partial factorization with respect to subvectors $\boldsymbol{\psi}_s$ of $\boldsymbol{\psi}$ can be chosen [YFCL17, MZCP17]

$$q(\boldsymbol{\psi}) = \prod_s q_s(\boldsymbol{\psi}_s). \tag{6.23}$$

Using the same proof as in [ŠQ06, Section 3.3.1], the expression of each q_s given the others is similar to formula (6.22). Nevertheless, the partially factorized approach is limited by the size of the subvectors $\boldsymbol{\psi}_s$: if these subvectors are too big, then VBA leads to the computation of covariance matrices which are not attainable due to the high dimension. In this case, the calculation of these covariance matrices can only be done approximately by specific strategies, such as Monte Carlo techniques [MZCP17]. In this chapter, we propose to deal with an approximation of the form

$$q(\boldsymbol{\psi}) = q(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) = \prod_j q_{1_j}(\psi_{1_j}|\psi_{2_j}) q_{2_j}(\psi_{2_j}) \tag{6.24}$$

where $\boldsymbol{\psi}$ is divided into subvectors $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$. These subvectors are the same size and a dependence is preserved between one scalar component of $\boldsymbol{\psi}_1$ and one scalar component of $\boldsymbol{\psi}_2$. Given the form (6.24) of the approximate distribution, we need to derive the updating formulae for q_{1_j} and q_{2_j}. Indeed, due to the dependence between ψ_{1_j} and ψ_{2_j} in q_{1_j}, the updating formula (6.22), used in the case of the mean field approximation, is not applicable. Inspired by the proof in [ŠQ06, Section 3.3.1] for the case of mean field approximation, we first write the Kullback-Leibler divergence

$$\begin{aligned}
KL\left(q(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right) &= \int_{\psi_{1_j},\psi_{2_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j}) q_{2_j}(\psi_{2_j}) \ln\left(q_{1_j}(\psi_{1_j}|\psi_{2_j}) q_{2_j}(\psi_{2_j})\right) \mathrm{d}\psi_{1_j}\, \mathrm{d}\psi_{2_j} \\
&+ \int_{\boldsymbol{\psi}_{1/j},\boldsymbol{\psi}_{2/j}} q_{1/j}(\boldsymbol{\psi}_{1/j}|\boldsymbol{\psi}_{2/j}) q_{2/j}(\boldsymbol{\psi}_{2/j}) \ln\left(q_{1/j}(\psi_{1_j}|\boldsymbol{\psi}_{2_j}) q_{2/j}(\boldsymbol{\psi}_{2_j})\right) \mathrm{d}\boldsymbol{\psi}_{1/j}\, \mathrm{d}\boldsymbol{\psi}_{2/j} \\
&- \int_{\psi_{1_j},\psi_{2_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j}) q_{2_j}(\psi_{2_j}) \\
&\quad \times \left(\int_{\boldsymbol{\psi}_{1/j},\boldsymbol{\psi}_{2/j}} q_{1/j}(\boldsymbol{\psi}_{1/j}|\boldsymbol{\psi}_{2/j}) q_{2/j}(\boldsymbol{\psi}_{2/j}) \ln\left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right) \mathrm{d}\boldsymbol{\psi}_{1/j}\, \mathrm{d}\boldsymbol{\psi}_{2/j}\right) \mathrm{d}\psi_{1_j}\, \mathrm{d}\psi_{2_j}.
\end{aligned} \tag{6.25}$$

We denote by

$$a_j = \int_{\boldsymbol{\psi}_{1/j},\boldsymbol{\psi}_{2/j}} q_{1/j}(\boldsymbol{\psi}_{1/j}|\boldsymbol{\psi}_{2/j}) q_{2/j}(\boldsymbol{\psi}_{2/j}) \ln\left(q_{1/j}(\psi_{1_j}|\boldsymbol{\psi}_{2_j}) q_{2/j}(\boldsymbol{\psi}_{2_j})\right) \mathrm{d}\boldsymbol{\psi}_{1/j}\, \mathrm{d}\boldsymbol{\psi}_{2/j} \tag{6.26}$$

the term in $KL\left(q(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)$ which is independent from q_{1_j} and q_{2_j}, and we introduce

$$\begin{aligned}&\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\\&\qquad=\int_{\boldsymbol{\psi}_{1/j},\boldsymbol{\psi}_{2/j}} q_{1/j}(\boldsymbol{\psi}_{1/j}|\boldsymbol{\psi}_{2/j})q_{2/j}(\boldsymbol{\psi}_{2/j})\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\,\mathrm{d}\boldsymbol{\psi}_{1/j}\,\mathrm{d}\boldsymbol{\psi}_{2/j}.\end{aligned}\tag{6.27}$$

Given (6.26) and (6.27), Kullback-Leibler divergence (6.25) reads

$$\begin{aligned}KL\left(q(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)&=\int_{\psi_{1_j},\psi_{2_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j})q_{2_j}(\psi_{2_j})\ln\left(q_{1_j}(\psi_{1_j}|\psi_{2_j})q_{2_j}(\psi_{2_j})\right)\,\mathrm{d}\psi_{1_j}\,\mathrm{d}\psi_{2_j}\\&+a_j-\int_{\psi_{1_j},\psi_{2_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j})q_{2_j}(\psi_{2_j})\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\,\mathrm{d}\psi_{1_j}\,\mathrm{d}\psi_{2_j}\end{aligned}\tag{6.28}$$

which leads to

$$\begin{aligned}KL\left(q(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)&=\int_{\psi_{1_j},\psi_{2_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j})q_{2_j}(\psi_{2_j})\ln\left(q_{1_j}(\psi_{1_j}|\psi_{2_j})\right)\,\mathrm{d}\psi_{1_j}\,\mathrm{d}\psi_{2_j}\\&+\int_{\psi_{2_j}} q_{2_j}(\psi_{2_j})\ln\left(q_{2_j}(\psi_{2_j})\right)\,\mathrm{d}\psi_{2_j}\\&+a_j-\int_{\psi_{1_j},\psi_{2_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j})q_{2_j}(\psi_{2_j})\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\,\mathrm{d}\psi_{1_j}\,\mathrm{d}\psi_{2_j}.\end{aligned}\tag{6.29}$$

First, we focus on the updating formula of q_{1_j}. Denoting the entropy of q_{2_j} by

$$\mathcal{H}(q_{2_j})=-\int_{\psi_{2_j}} q_{2_j}(\psi_{2_j})\ln\left(q_{2_j}(\psi_{2_j})\right)\,\mathrm{d}\psi_{2_j}\tag{6.30}$$

and introducing

$$Z_{1_j}(\psi_{2_j})=\int_{\psi_{1_j}}\exp\left[\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\right]\,\mathrm{d}\psi_{1_j},\tag{6.31}$$

equation (6.29) reads

$$\begin{aligned}&KL\left(q(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)=a_j-\mathcal{H}(q_{2_j})\\&+\int_{\psi_{2_j}} q_{2_j}(\psi_{2_j})\left(\int_{\psi_{1_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j})\ln\left(\frac{q_{1_j}(\psi_{1_j}|\psi_{2_j})\times\frac{1}{Z_{1_j}(\psi_{2_j})}}{\frac{1}{Z_{1_j}(\psi_{2_j})}\exp\left[\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\right]}\right)\mathrm{d}\psi_{1_j}\right)\mathrm{d}\psi_{2_j}\end{aligned}\tag{6.32}$$

which leads to

$$\begin{aligned}&KL\left(q(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)=a_j-\mathcal{H}(q_{2_j})-\int_{\psi_{2_j}} q_{2_j}(\psi_{2_j})\ln\left(Z_{1_j}(\psi_{2_j})\right)\,\mathrm{d}\psi_{2_j}\\&+\int_{\psi_{2_j}} q_{2_j}(\psi_{2_j})KL\left(q_{1_j}(\psi_{1_j}|\psi_{2_j})\,\middle|\middle|\,\frac{1}{Z_{1_j}(\psi_{2_j})}\exp\left[\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\right]\right)\,\mathrm{d}\psi_{2_j}.\end{aligned}\tag{6.33}$$

Hence, by the same arguments as the ones used in section 6.1.1 for the mean field approximation, the updating formula for q_{1_j} is, dropping the normalizing constant,

$$q_{1_j}(\psi_{1_j}|\psi_{2_j})\propto\exp\left[\mathbb{E}_{q_{/1_j,2_j}}\left(\ln\left(p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)\right)\right].\tag{6.34}$$

Next, we have to find the updating formula for q_{2_j} given the other distributions. From equation (6.29), we have

$$KL\left(q(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2)||p(\boldsymbol{\psi}_1,\boldsymbol{\psi}_2|\boldsymbol{g};\mathcal{U},\mathcal{M})\right)=a_j+\int_{\psi_{2_j}} q_{2_j}(\psi_{2_j})\ln\left(q_{2_j}(\psi_{2_j})\right)\,\mathrm{d}\psi_{2_j}$$

$$- \int_{\psi_{2_j}} q_{2_j}(\psi_{2_j}) \left[\int_{\psi_{1_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j}) \mathbb{E}_{q_{/1_j,2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) \mathrm{d}\psi_{1_j} \right.$$
$$\left. - \int_{\psi_{1_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j}) \ln \left(q_{1_j}(\psi_{1_j}|\psi_{2_j}) \right) \mathrm{d}\psi_{1_j} \right] \mathrm{d}\psi_{2_j}. \tag{6.35}$$

The entropy of q_{1_j} given ψ_{2_j} is denoted by

$$\mathcal{H}(q_{1_j}|\psi_{2_j}) = - \int_{\psi_{1_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j}) \ln \left(q_{1_j}(\psi_{1_j}|\psi_{2_j}) \right) \mathrm{d}\psi_{1_j}. \tag{6.36}$$

Introducing

$$\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) = \int_{\psi_{1_j}} q_{1_j}(\psi_{1_j}|\psi_{2_j}) \mathbb{E}_{q_{/1_j,2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) \mathrm{d}\psi_{1_j}, \tag{6.37}$$

we have from (6.35)

$$\begin{aligned}
KL\left(q(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) || p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M})\right) &= a_j + \int_{\psi_{2_j}} q_{2_j}(\psi_{2_j}) \ln \left(q_{2_j}(\psi_{2_j}) \right) \mathrm{d}\psi_{2_j} \\
&\quad - \int_{\psi_{2_j}} q_{2_j}(\psi_{2_j}) \left[\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) + \mathcal{H}(q_{1_j}|\psi_{2_j}) \right] \mathrm{d}\psi_{2_j} \\
&= a_j + \int_{\psi_{2_j}} q_{2_j}(\psi_{2_j}) \ln \left(\frac{q_{2_j}(\psi_{2_j})}{\exp \left[\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) + \mathcal{H}(q_{1_j}|\psi_{2_j}) \right]} \right) \mathrm{d}\psi_{2_j} \\
&= a_j + \int_{\psi_{2_j}} q_{2_j}(\psi_{2_j}) \ln \left(\frac{q_{2_j}(\psi_{2_j}) \times \frac{1}{Z_{2_j}}}{\frac{1}{Z_{2_j}} \exp \left[\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) + \mathcal{H}(q_{1_j}|\psi_{2_j}) \right]} \right) \mathrm{d}\psi_{2_j},
\end{aligned} \tag{6.38}$$

where

$$Z_{2_j} = \int_{\psi_{2_j}} \exp \left[\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) + \mathcal{H}(q_{1_j}|\psi_{2_j}) \right] \mathrm{d}\psi_{2_j} \tag{6.39}$$

is a normalizing constant independent from q_{2_j}. Hence, equation (6.38) reads

$$\begin{aligned}
KL\left(q(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) || p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M})\right) &= a_j - \ln \left(Z_{2_j} \right) \\
&\quad + KL \left(q_{2_j}(\psi_{2_j}) \left| \left| \frac{1}{Z_{2_j}} \exp \left[\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) + \mathcal{H}(q_{1_j}|\psi_{2_j}) \right] \right. \right. \right).
\end{aligned} \tag{6.40}$$

As a result, the updating formula for q_{2_j} given the other distributions is, dropping the normalizing constant,

$$q_{2_j}(\psi_{2_j}) \propto \exp \left[\mathbb{E}_{q_{/2_j}} \left(\ln \left(p(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2 | \boldsymbol{g}; \mathcal{U}, \mathcal{M}) \right) \right) + \mathcal{H}(q_{1_j}|\psi_{2_j}) \right]. \tag{6.41}$$

As we see, this formula depends on the logarithm of the posterior distribution but also on the entropy of q_{1_j}. This additional term makes formula (6.41) rather different from other updating formulae (6.22) and (6.34) encountered in VBA.

6.2 Application of VBA to 3D X-ray CT with Gauss-Markov-Potts prior

For our reconstruction problem and with our priors recapped in section 6.1, we choose an approximate posterior distribution of the form

$$q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}) = \prod_{j=1}^{N} q_{f_j}(f_j|z_j) \times \prod_{j=1}^{N} q_{z_j}(z_j)$$

$$\times \prod_{i=1}^{M} q_{\rho_{\zeta_i}}(\rho_{\zeta_i}) \times \prod_{k=1}^{K} q_{m_k}(m_k) \times \prod_{k=1}^{K} q_{\rho_k}(\rho_k). \tag{6.42}$$

As in section 6.1.2, this approximation performs a partial separation, since the dependence between attenuation coefficient f_j and label z_j is preserved. This approximation has been proposed for the first time in [AMD10] for image restoration. Nevertheless, in [AMD10], the general updating formulae (6.34) and (6.41) are not derived and the authors only minimize the Kullback-Leibler divergence expressed with explicit priors.

The details of the calculations to compute the approximate distributions q_{f_j}, q_{z_j}, q_{ρ_i}, q_{m_k} and q_{ρ_k} are given in appendix F. These calculations are based on the applications of the updating formula (6.34) for q_{f_j}, of the updating formula (6.41) for q_{z_j}, since the separation between the volume $\boldsymbol{f}$ and the labels $\boldsymbol{z}$ is partial. For q_{ζ_i}, q_{m_k} and q_{v_k}, the separation is full, so we apply the updating formula (6.22) of mean field approximation. After the calculations in appendix F, we find

$$\begin{cases} q_{f_j}(f_j|z_j=k) = \mathcal{N}(f_j|\tilde{m}_{jk}, \tilde{v}_{jk}), \forall k \\ q_{z_j}(z_j) = \frac{\exp\left[\sum_{k=1}^{K}\left(\tilde{\alpha}_{jk}+\gamma_0 \sum_{i\in\mathcal{V}(j)} q_{z_i}(k)\right)\delta(z_j-k)\right]}{\sum_{k=1}^{K}\exp\left[\tilde{\alpha}_{jk}+\gamma_0 \sum_{i\in\mathcal{V}(j)} q_{z_i}(k)\right]} \\ q_{\rho_{\zeta_i}}(\rho_{\zeta_i}) = \mathcal{G}(\rho_{\zeta_i}|\tilde{\alpha}_{\zeta_{0_i}}, \tilde{\beta}_{\zeta_{0_i}}) \\ q_{m_k}(m_k) = \mathcal{N}(m_k|\tilde{m}_{0_k}, \tilde{v}_{0_k}) \\ q_{\rho_k}(\rho_k) = \mathcal{G}(\rho_k|\tilde{\alpha}_{0_k}, \tilde{\beta}_{0_k}) \end{cases} \tag{6.43}$$

where $q_{z_i}(k)$ in the expression of $q_{z_j}(k)$ is the value of $q_{z_i}(k)$ at previous iteration. Denoting

$$\begin{cases} q_{\boldsymbol{f}}(\boldsymbol{f}|\boldsymbol{z}) = \prod_{j=1}^{N} q_{f_j}(f_j|z_j) \\ q_{\boldsymbol{z}}(\boldsymbol{z}) = \prod_{j=1}^{N} q_{z_j}(z_j) \\ q_{\boldsymbol{\rho}_\zeta}(\boldsymbol{\rho}_\zeta) = \prod_{i=1}^{M} q_{\rho_{\zeta_i}}(\rho_{\zeta_i}) \\ q_{\boldsymbol{m}}(\boldsymbol{m}) = \prod_{k=1}^{K} q_{m_k}(m_k) \\ q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) = \prod_{k=1}^{K} q_{\rho_k}(\rho_k) \end{cases}, \tag{6.44}$$

the VBA algorithm in figure 6.1 gives the order of the updates of the approximate distributions. The distributions of the variables which are approximated as independent are immediatly replaced by their updates. The first two steps of the algorithm in figure 6.1 have to be seen as the update of the joint approximate distribution of the volume and the labels

$$q^{(t)}_{\boldsymbol{f},\boldsymbol{z}}(\boldsymbol{f},\boldsymbol{z}) = q^{(t)}_{\boldsymbol{f}}(\boldsymbol{f}|\boldsymbol{z}) q^{(t)}_{\boldsymbol{z}}(\boldsymbol{z}). \tag{6.45}$$

For this reason, the update of $q^{(t)}_{\boldsymbol{z}}(\boldsymbol{z})$ is done using $q^{(t-1)}_{\boldsymbol{f}}(\boldsymbol{f}|\boldsymbol{z})$.

As shown in equations (6.43), the forms of the approximate distributions do not change through the algorithm. Consequently, the VBA algorithm turns into iterative updates of the parameters of the distributions in equation (6.43). The updating formulae involve the *digamma* function

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}, \forall x \in \mathbb{R}^*_+, \tag{6.46}$$

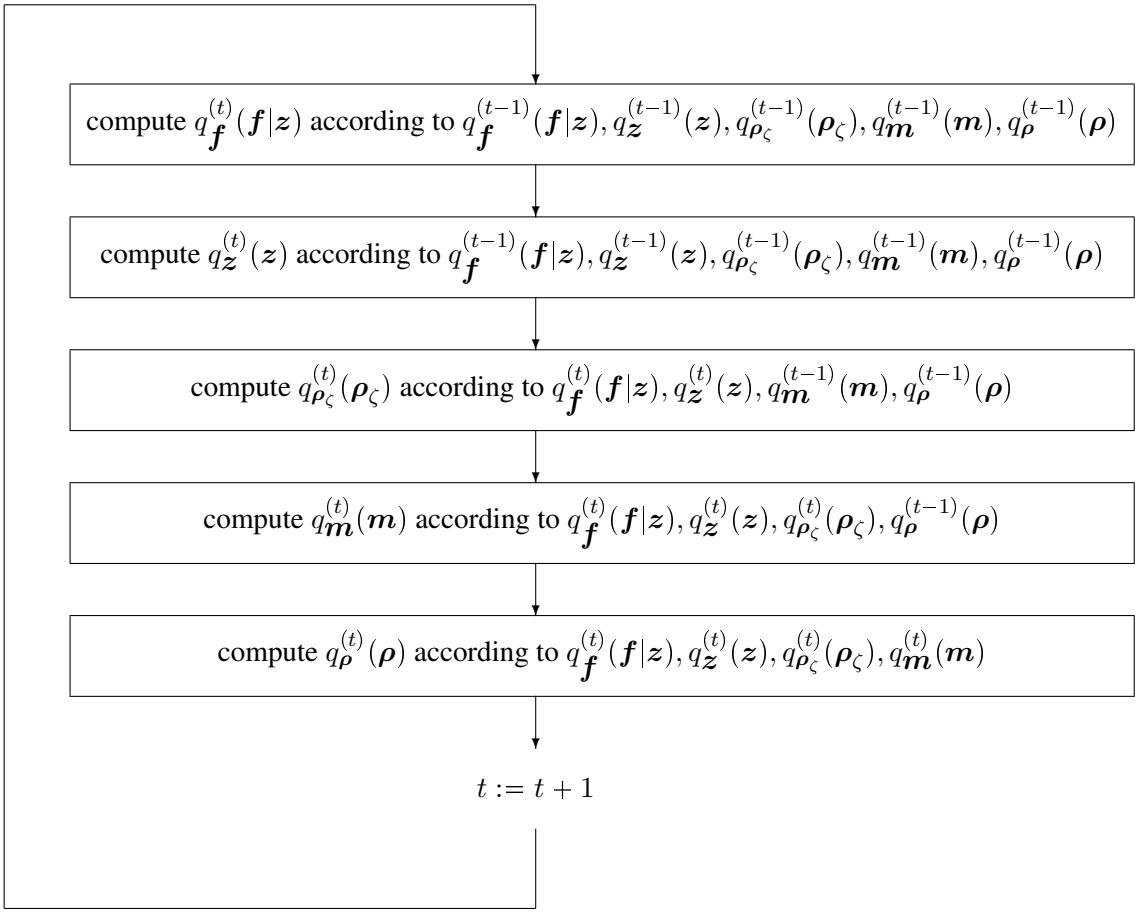


Figure 6.1: Iterative algorithm to compute approximating distribution $q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})$

as several auxiliary variables :

$$\begin{cases} \tilde{m}_j = \sum_{k=1}^{K} \tilde{m}_{jk} q_{z_j}(k) \\ \tilde{v}_j = \sum_{k=1}^{K} \tilde{v}_{jk} q_{z_j}(k) \\ \tilde{m}_j^{(2)} = \sum_{k=1}^{K} \left(\tilde{m}_{jk} - \tilde{m}_j\right)^2 q_{z_j}(k) = \sum_{k=1}^{K} \tilde{m}_{jk}^2 q_{z_j}(k) - \tilde{m}_j^2 \\ \tilde{v}_j^{(2)} = \tilde{v}_j + \tilde{m}_j^{(2)} \\ \tilde{v}_{\zeta_i} = \frac{\tilde{\beta}_{\zeta_{0_i}}}{\tilde{\alpha}_{\zeta_{0_i}}} \end{cases} \tag{6.47}$$

and the diagonal matrix $\tilde{\boldsymbol{V}}_\zeta = \text{diag}\left[\tilde{\boldsymbol{v}}_\zeta\right]$. At the end of the algorithm, the unknowns are estimated by their expectation with respect to the approximate distribution q, excepted for the labels which are estimated by maximum a posteriori,

due to the fact that they are discrete variables :

$$\begin{cases} \hat{f}_j = \mathbb{E}_q(f_j) = \sum_{k=1}^K \mathbb{E}_q(f_j|z_j=k) q_{z_j}(k) = \sum_{k=1}^K \tilde{m}_{jk} q_{z_j}(k) = \tilde{m}_j \\ \hat{\rho}_{\zeta_i} = \mathbb{E}_q(\rho_{\zeta_i}) = \frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}} \\ \hat{z}_j = \arg\max_k \left\{ q_{z_j}(k) \right\} \\ \hat{m}_k = \mathbb{E}_q(m_k) = \tilde{m}_{0_k} \\ \hat{\rho}_k = \mathbb{E}_q(\rho_k) = \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} \end{cases} . \tag{6.48}$$

Concerning the posterior variances, denoted by $\boldsymbol{v_f} = (v_{f_j})_j$, they are estimated at the end of the VBA algorithm by

$$\hat{v}_{f_j} = \mathbb{V}\text{ar}_q(f_j) = \sum_{k=1}^K \mathbb{E}_q(f_j^2|z_j=k) q_{z_j}(k) - \tilde{m}_j^2 = \sum_{k=1}^K (\tilde{v}_{jk} + \tilde{m}_{jk}^2) q_{z_j}(k) - \tilde{m}_j^2 = \tilde{v}_j^{(2)}, \forall j. \tag{6.49}$$

The stopping criterion of the VBA algorithm in figure 6.1 is the negative free energy from which constants are removed :

$$\begin{aligned} \mathcal{L}(q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})) = & \frac{1}{2} \sum_{j=1}^N \sum_{k=1}^K \ln(\tilde{v}_{jk}) q_{z_j}(k) - \sum_{j=1}^N \sum_{k=1}^K q_{z_j}(k) \ln\left(q_{z_j}(k)\right) \\ & + \sum_{i=1}^M \left[\ln(\Gamma(\tilde{\alpha}_{\zeta_{0_i}})) - \ln(\tilde{\beta}_{\zeta_{0_i}}) + \tilde{\alpha}_{\zeta_{0_i}} - \left(\tilde{\alpha}_{\zeta_{0_i}} - 1\right) \psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right) \right] \\ & + \frac{1}{2} \sum_{k=1}^K \ln(\tilde{v}_{0_k}) + \sum_{k=1}^K \left[\ln(\Gamma(\tilde{\alpha}_{0_k})) - \ln(\tilde{\beta}_{0_k}) + \tilde{\alpha}_{0_k} - (\tilde{\alpha}_{0_k} - 1) \psi(\tilde{\alpha}_{0_k}) \right] \\ & - \frac{1}{2} \sum_{i=1}^M \left[\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right) \right] - \frac{1}{2} \|\boldsymbol{g} - \boldsymbol{H}\tilde{\boldsymbol{m}}\|^2_{\tilde{\boldsymbol{V}}_\zeta} - \frac{1}{2} \sum_{j=1}^N \tilde{v}_j^{(2)} \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H} \right]_{jj} \\ & - \frac{1}{2} \sum_{j=1}^N \sum_{k=1}^K \left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} \left[\tilde{v}_{jk} + \tilde{v}_{0_k} + (\tilde{m}_{jk} - \tilde{m}_{0_k})^2 \right] + \ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k}) \right) q_{z_j}(k) \\ & + \sum_{j=1}^N \sum_{k=1}^K \left(\alpha_k + \gamma_0 \sum_{i \in \mathcal{V}(j)} q_{z_i}(k) \right) q_{z_j}(k) \\ & - (\alpha_{\zeta_0} - 1) \sum_{i=1}^M (\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi(\tilde{\alpha}_{\zeta_{0_i}})) - \beta_{\zeta_0} \sum_{i=1}^M \frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}} \\ & - \frac{1}{2v_0} \sum_{k=1}^K \left(\tilde{v}_{0_k} + (\tilde{m}_{0_k} - m_0)^2 \right) - (\alpha_0 - 1) \sum_{k=1}^K (\ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})) - \beta_0 \sum_{k=1}^K \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}. \end{aligned} \tag{6.50}$$

The details of the calculations of the stopping criterion (F.65) are given in appendix F. In the following of this section, we give the updating formulae of the parameters of the approximate distributions. Like in the previous chapters, the subscripts t, $0 \leqslant t < t_{max}$, corresponding to the iterations of the algorithm, are omitted for the sake of readibility.

6.2.1 Approximate distribution for the volume

By applying formula (6.34), we have :

$$q_{f_j}(f_j|z_j=k) = \mathcal{N}(f_j|\tilde{m}_{jk}, \tilde{v}_{jk}), \forall j, \forall k \tag{6.51}$$

where

$$\left\{ \begin{array}{l} \tilde{v}_{jk} = \left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} + \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H} \right]_{jj} \right)^{-1} \\ \tilde{m}_{jk} = \tilde{m}_j + \tilde{v}_{jk} \left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} (\tilde{m}_{0_k} - \tilde{m}_j) + \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} (\boldsymbol{g} - \boldsymbol{H}\tilde{\boldsymbol{m}}) \right]_j \right) \end{array} \right. \tag{6.52}$$

6.2.2 Approximate distribution for the labels

By applying formula (6.41), we have :

$$q_{z_j}(z_j) = \frac{\exp\left[\sum_{k=1}^K \left(\tilde{\alpha}_{jk} + \gamma_0 \sum_{i \in \mathcal{V}(j)} q_{z_i}(k) \right) \delta(z_j - k) \right]}{\sum_{k=1}^K \exp\left[\tilde{\alpha}_{jk} + \gamma_0 \sum_{i \in \mathcal{V}(j)} q_{z_i}(k) \right]} \tag{6.53}$$

where $q_{z_i}(k), \forall i \in \mathcal{V}(j)$, is the value of $q_{z_i}(k)$ at the previous iteration of the algorithm presented in figure 6.1. We have

$$\begin{aligned} \tilde{\alpha}_{jk} = \alpha_k - \frac{1}{2} \left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} \left[\tilde{v}_{jk} + \tilde{v}_{0_k} + (\tilde{m}_{jk} - \tilde{m}_{0_k})^2 \right] + \ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k}) \right) + \frac{1}{2} \ln(\tilde{v}_{jk}) \\ - \frac{1}{2} \left(\left(\tilde{v}_{jk} + \tilde{m}_{jk}^2 \right) \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H} \right]_{jj} - 2\tilde{m}_{jk} \left(\tilde{m}_j \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H} \right]_{jj} + \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} (\boldsymbol{g} - \boldsymbol{H}\tilde{\boldsymbol{m}}) \right]_j \right) \right) \end{aligned} \tag{6.54}$$

It is worth to notice that this step does not imply the calculation of the backprojection of the errors and of diagonal coefficients $\left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H} \right]_{jj}$, for all j, since they have been computed before in order to update the approximate distribution of the volume, according to the order of the updates shown in figure 6.1.

6.2.3 Approximate distribution of the inverses of the variances of the uncertainties

By applying formula (6.22), we have :

$$q_{\rho_{\zeta_i}}(\rho_{\zeta_i}) = \mathcal{G}(\rho_{\zeta_i} | \tilde{\alpha}_{\zeta_{0_i}}, \tilde{\beta}_{\zeta_{0_i}}) \tag{6.55}$$

where

$$\left\{ \begin{array}{l} \tilde{\alpha}_{\zeta_{0_i}} = \alpha_{\zeta_0} + \frac{1}{2} \\ \tilde{\beta}_{\zeta_{0_i}} = \beta_{\zeta_0} + \frac{1}{2} \left((g_i - [\boldsymbol{H}\tilde{\boldsymbol{m}}]_i)^2 + \left(\boldsymbol{H} \tilde{\boldsymbol{V}}^{(2)} \boldsymbol{H}^T \right)_{ii} \right) \end{array} \right. , \forall i \in \{1, \ldots, M\}. \tag{6.56}$$

and $\tilde{\boldsymbol{V}}^{(2)} = \text{diag}\left[\tilde{\boldsymbol{v}}^{(2)} \right]$.

6.2.4 Approximate distribution for the means of the classes

By applying formula (6.22), we have :

$$q_{m_k}(m_k) = \mathcal{N}(m_k | \tilde{m}_{0_k}, \tilde{v}_{0_k}) \tag{6.57}$$

where

$$\left\{ \begin{array}{l} \tilde{v}_{0_k} = \left(\frac{1}{v_0} + \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} \sum_{j=1}^N q_{z_j}(k) \right)^{-1} \\ \tilde{m}_{0_k} = \tilde{v}_{0_k} \left(\frac{m_0}{v_0} + \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} \sum_{j=1}^N \tilde{m}_{jk} q_{z_j}(k) \right) \end{array} \right. \tag{6.58}$$

6.2.5 Approximate distribution for the inverses of the variances of the classes

By applying formula (6.22), we have :

$$q_{\rho_k}(\rho_k) = \mathcal{G}(\rho_k | \tilde{\alpha}_{0_k}, \tilde{\beta}_{0_k}) \tag{6.59}$$

where

$$\left\{ \begin{array}{l} \tilde{\alpha}_{0_k} = \alpha_0 + \frac{1}{2} \sum_{j=1}^N q_{z_j}(k) \\ \tilde{\beta}_{0_k} = \beta_0 + \frac{1}{2} \sum_{j=1}^N \left(\tilde{v}_{0_k} + \tilde{v}_{jk} + (\tilde{m}_{jk} - \tilde{m}_{0_k})^2 \right) q_{z_j}(k) \end{array} \right. \tag{6.60}$$

6.2.6 Summary and initialization of the VBA algorithm

The proposed VBA algorithm is summarized in algorithm 15. The updating formulae are applied according to the order given in figure 6.1. In particular, for the computation of $q_{z_j}(k)$ at iteration t, $\tilde{m}_{jk}^{(t-1)}$ and $\tilde{v}_{jk}^{(t-1)}$ must be used, and not $\tilde{m}_{jk}^{(t)}$ and $\tilde{v}_{jk}^{(t)}$, in order to ensure the convergence of the algorithm.

Algorithm 15 VBA algorithm with Gauss-Markov-Potts prior

Initialize $\tilde{m}_{jk}^{(0)}, \tilde{v}_{jk}^{(0)}, q_{z_j}(k)^{(0)}, \tilde{\alpha}_{\zeta_{0_i}}^{(0)}, \tilde{\beta}_{\zeta_{0_i}}^{(0)}, \tilde{m}_{0_k}^{(0)}, \tilde{v}_{0_k}^{(0)}, \tilde{\alpha}_{0_k}^{(0)}, \tilde{\beta}_{0_k}^{(0)}, \forall j, i, k$
for $t = 0, \ldots, t_{max} - 1$ **do**
1 : Update $\tilde{m}_{jk}$ and $\tilde{v}_{jk}$ by (6.52), $\forall j, k$
2 : Update $q_{z_j}(k)$ by (6.53), $\forall j, k$
3 : Update $\tilde{\alpha}_{\zeta_{0_i}}$ and $\tilde{\beta}_{\zeta_{0_i}}$ by (6.56), $\forall i$
4 : Update $\tilde{m}_{0_k}$ and $\tilde{v}_{0_k}$ by (6.58), $\forall k$
5 : Update $\tilde{\alpha}_{0_k}$ and $\tilde{\beta}_{0_k}$ by (6.60), $\forall k$
end for
Return $\hat{\boldsymbol{f}}, \hat{\boldsymbol{v}}_{\boldsymbol{f}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{\rho}}_{\zeta}, \hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{\rho}}$ by (6.48) and (6.49)

Since the number of variables is quite high, the initialization of algorithm 15 is not trivial. First, based on an initial reconstruction $\boldsymbol{f}^{(0)}$ obtained by filtered backprojection [FDK84], an initial segmentation $\boldsymbol{z}^{(0)}$ is performed by applying a fast method [Ots79, KSH01], as it is done for Joint Maximization A Posteriori (JMAP) in chapter 4. From this initial segmentation, initial means $\boldsymbol{m}^{(0)}$ and variances $\boldsymbol{v}^{(0)}$ of the classes are computed. Then, the means $\tilde{m}_{jk}, \forall j, k$ are initialized to $f_j^{(0)}$ if voxel j is in class k at initialization, and to $m_k^{(0)}$ otherwise :

$$\tilde{m}_{jk}^{(0)} = \begin{cases} f_j^{(0)} \text{ if } z_j^{(0)} = k \\ m_k^{(0)} \text{ otherwise} \end{cases} \tag{6.61}$$

Concerning the variances of the approximate distribution of the volume, inspired by their updating formula (6.52), we initialize $\tilde{v}_{jk}$ by :

$$\tilde{v}_{jk}^{(0)} = \left(\frac{1}{v_k^{(0)}} + \frac{\alpha_{\zeta_0}}{\beta_{\zeta_0}} \left[\boldsymbol{H}^T \boldsymbol{H} \right]_{jj} \right)^{-1}. \tag{6.62}$$

The probabilities $q_{z_j}(k), \forall j, \forall k$, for each voxel to be in each class are initialized by 0 or 1 :

$$q_{z_j}(k)^{(0)} = \begin{cases} 1 \text{ if } z_j^{(0)} = k \\ 0 \text{ otherwise} \end{cases} \tag{6.63}$$

The approximate distributions of $\boldsymbol{\rho}_{\zeta}$, $\boldsymbol{m}$ and $\boldsymbol{\rho}$ are initialized to the conditional distributions given the other unknowns. These distributions are derived in chapter 4. We set

$$\tilde{\alpha}_{\zeta_{0_i}}^{(0)} = \alpha_{\zeta_0} + \frac{1}{2}, \forall i, \tag{6.64}$$

$$\tilde{\beta}_{\zeta_{0_i}}^{(0)} = \beta_{\zeta_0} + \frac{1}{2} \left(g_i - \left[\boldsymbol{H} \boldsymbol{f}^{(0)} \right]_i \right)^2, \forall i, \tag{6.65}$$

$$\tilde{v}_{0_k}^{(0)} = \left(\frac{1}{v_0} + \frac{N_k^{(0)}}{v_k^{(0)}} \right)^{-1}, \forall k, \tag{6.66}$$

$$\tilde{m}_{0_k}^{(0)} = \tilde{v}_{0_k}^{(0)} \left(\frac{m_0}{v_0} + N_k^{(0)} \frac{m_k^{(0)}}{v_k^{(0)}} \right), \forall k, \tag{6.67}$$

$$\tilde{\alpha}_{0_k}^{(0)} = \alpha_0 + \frac{N_k^{(0)}}{2}, \forall k, \tag{6.68}$$

$$\tilde{\beta}_{0_k}^{(0)} = \beta_0 + \frac{N_k^{(0)}}{2} v_k^{(0)}, \forall k, \tag{6.69}$$

where $N_k^{(0)}$ is the number of voxels in class k in the initial segmentation $\boldsymbol{z}^{(0)}$.

After these initializations, algorithm 15 can be applied. Nevertheless, for the case of 3D X-ray CT, the algorithm has an intrinsic difficulty we now describe in the next section.

6.3 Difficulty of implementation : computation of the diagonal coefficients

At one iteration of the algorithm, for any voxel j, the diagonal coefficient used to compute v_{jk} by (6.52) is

$$d_{v_j} = \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H}\right]_{jj} = \|\boldsymbol{H}\boldsymbol{e}^{(j)}\|^2_{\tilde{\boldsymbol{V}}_\zeta} \tag{6.70}$$

where $e_i^{(j)} = \delta(j-i), \forall i$. As $\boldsymbol{d}_v = (d_{v_j})_j$ has the size of a volume, formula (6.70) implies to compute N projections, which is very long, even if the projector implemented on GPU is very fast. We calculated that, if we have to reconstruct a volume of size $N = 256^3$ voxels, and if one projection operation takes only 10 milliseconds, computing all diagonal coefficients $d_{v_j}, \forall j$, for only one iteration of algorithm 15, would require more than 40 hours. Due to this huge computational cost, we prefer to consider the algebraic formula :

$$d_{v_j} = \left[\boldsymbol{H}^T \tilde{\boldsymbol{V}}_\zeta^{-1} \boldsymbol{H}\right]_{jj} = \sum_{i=1}^{M} H_{ij}^2 \tilde{v}_{\zeta_i}^{-1}, \forall j. \tag{6.71}$$

From this formula, diagonal coefficients $\boldsymbol{d}_v = (d_{v_j})_j$ appear to be similar to a backprojection of $\tilde{\boldsymbol{v}}_\zeta^{-1} = (\tilde{v}_{\zeta_i}^{-1})_i$, except that coefficients H_{ij} are replaced by their squares $H_{ij}^2, \forall i, j$. Similarly, diagonal coefficients

$$d_{\zeta_i} = \left[\boldsymbol{H}\tilde{\boldsymbol{V}}\boldsymbol{H}^T\right]_{ii} = \sum_{j=1}^{N} H_{ij}^2 \tilde{v}_j, \forall i, \tag{6.72}$$

appear like a projection of volume $\tilde{\boldsymbol{v}}$, with H_{ij}^2 instead of H_{ij}. Given formulae (6.71) and (6.72), we implement a *squared-projector* $\boldsymbol{H}^{(2)}$ such that $H_{ij}^{(2)} = H_{ij}^2, \forall i, j$, and a *squared-backprojector* $(\boldsymbol{H}^{(2)})^T$, which are projection-like and backprojection-like operators, in which coefficients H_{ij} are replaced by their squares H_{ij}^2. In order to ensure the validity of formulae (6.71) and (6.72), and therefore the convergence of our algorithm, we use a matched P/BP pair, which is here the Separable Footprint (SF) pair [LFB10] described in chapter 3. We recall that the SF projection and backprojection operations respectively read

$$\begin{aligned} g(u_e, v_e, \phi) = g_i &= [\boldsymbol{H}\boldsymbol{f}]_i \\ &= l_{\theta_c}(u_e, v_e) \sum_{x_e} \sum_{y_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e) \end{aligned} \tag{6.73}$$

and

$$\begin{aligned} b(x_e, y_e, z_e) = b_j &= \left[\boldsymbol{H}^T \boldsymbol{g}\right]_j \\ &= \sum_{\phi} \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) \sum_{u_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi), \end{aligned} \tag{6.74}$$

where $F_{trans}(u_e, \phi; x_e, y_e)$ is the trapezoidal transaxial footprint of voxel (x_e, y_e, z_e) onto cell (u_e, v_e) at projection angle ϕ, $F_{ax}(v_e, \phi; x_e, y_e, z_e)$ is the rectangular axial footprint, and $l_{\theta_c}(u_e, v_e)$ and $l_{\psi_v}(\phi; x_e, y_e)$ are amplitude functions [LFB10]. Consequently, for the SF pair, a squared-projection operation is defined as

$$d_{\zeta_i} = \left[\boldsymbol{H}^{(2)} \tilde{\boldsymbol{v}}^{(2)}\right]_i = d_\zeta(u_e, v_e, \phi)$$

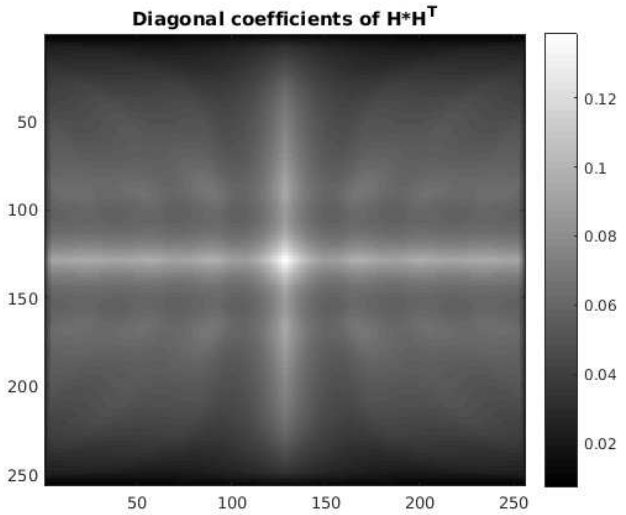


Figure 6.2: Diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$. These coefficients are the size of projections, of which one image is shown.

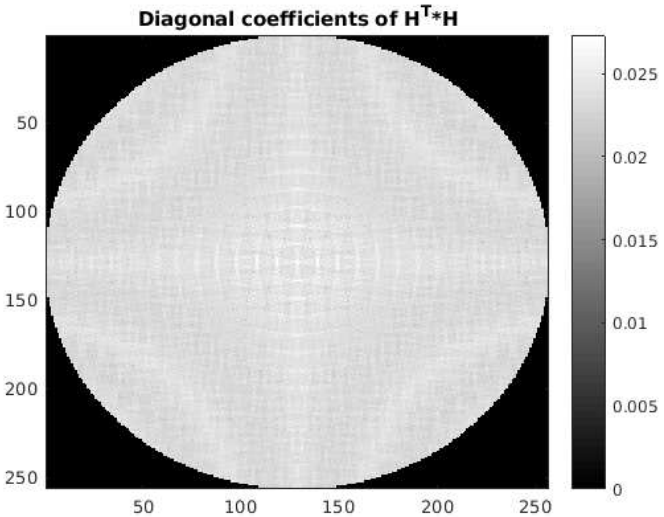


Figure 6.3: Diagonal coefficients of $\boldsymbol{H}^T\boldsymbol{H}$. These coefficients are the size of a volume, of which the middle slice is shown.

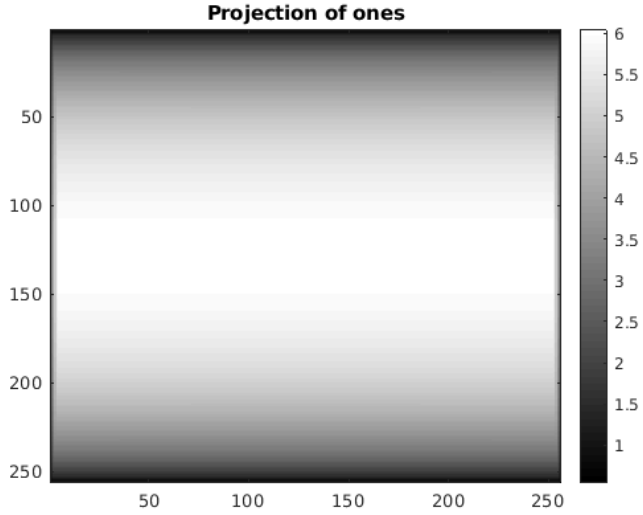


Figure 6.4: SF projection of a volume fully filled with ones, to compare to the diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$.

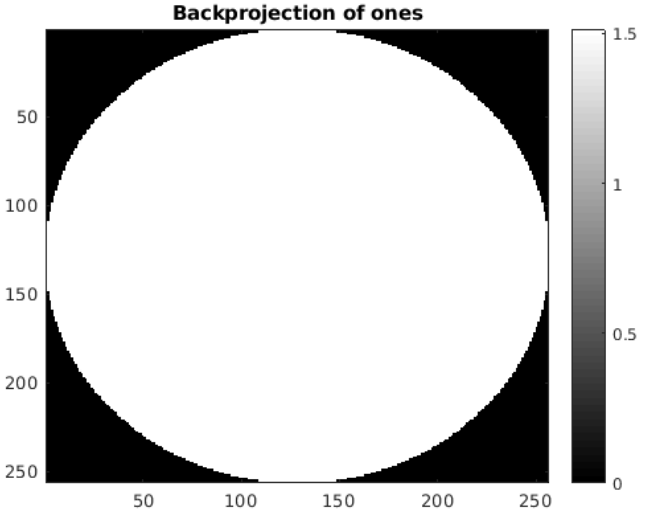


Figure 6.5: SF backprojection of projections fully filled with ones, to compare to the diagonal coefficients of $\boldsymbol{H}^T\boldsymbol{H}$.

$$= l_{\theta_c}(u_e, v_e)^2 \sum_{x_e} \sum_{y_e} l_{\psi_v}(\phi; x_e, y_e)^2 F_{trans}(u_e, \phi; x_e, y_e)^2 \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e)^2 \tilde{v}^{(2)}(x_e, y_e, z_e) \tag{6.75}$$

and a squared-backprojection is

$$\begin{aligned} d_{v_j} &= \left[\boldsymbol{H}^{(2)^T} \tilde{\boldsymbol{v}}_{\zeta}^{-1} \right]_j = d_v(x_e, y_e, z_e) \\ &= \sum_{\phi} \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e)^2 \sum_{u_e} l_{\psi_v}(\phi; x_e, y_e)^2 F_{trans}(u_e, \phi; x_e, y_e)^2 l_{\theta_c}(u_e, v_e)^2 \frac{1}{\tilde{v}_{\zeta}(u_e, v_e, \phi)}. \end{aligned} \tag{6.76}$$

Both the squared SF projector and the squared SF backprojector are implemented on the GPU exactly like the SF projector and the SF backprojector in chapter 3. Thanks to the use of these operators, in one iteration of our algorithm, diagonal coefficients $d_{v_j}, \forall j$, are simultaneously computed by applying $(\boldsymbol{H}^{(2)})^T$, which is very fast because it takes the same time as a backprojection, instead of N projections. Similarly, diagonal coefficients $d_{\zeta_i}, \forall i$, are simultaneously computed by applying $\boldsymbol{H}^{(2)}$, as fast as one projection, instead of M backprojections. Here, we see the advantage of using a matched P/BP pair, since this fast computation of the diagonal coefficients is not feasible with an unmatched pair. Figures 6.2 and 6.3 show the diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$ and $\boldsymbol{H}^T\boldsymbol{H}$, computed by $\boldsymbol{H}^{(2)}$ and $(\boldsymbol{H}^{(2)})^T$ respectively. The diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$ have the size of projections and are shown as it in figure 6.2, while those of $\boldsymbol{H}^T\boldsymbol{H}$ are shown as a volume. Given formula (6.75), the diagonal coefficients of $\boldsymbol{H}\boldsymbol{H}^T$ correspond to the squared projection of a volume fully filled with ones, denoted by $\mathbf{1}_N$:

$$\begin{aligned} (\boldsymbol{H}\boldsymbol{H}^T)_{ii} &= \left[\boldsymbol{H}^{(2)} \mathbf{1}_N \right]_i \\ &= l_{\theta_c}(u_e, v_e)^2 \sum_{x_e} \sum_{y_e} l_{\psi_v}(\phi; x_e, y_e)^2 F_{trans}(u_e, \phi; x_e, y_e)^2 \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e)^2, \end{aligned} \tag{6.77}$$

while, given formula (6.76), the diagonal coefficients of $\boldsymbol{H}^T\boldsymbol{H}$ correspond to the squared backprojection of projections fully filled with ones, denoted by $\mathbf{1}_M$:

$$\begin{aligned} (\boldsymbol{H}^T\boldsymbol{H})_{jj} &= \left[\boldsymbol{H}^{(2)^T} \mathbf{1}_M \right]_j \\ &= \sum_{\phi} \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e)^2 \sum_{u_e} l_{\psi_v}(\phi; x_e, y_e)^2 F_{trans}(u_e, \phi; x_e, y_e)^2 l_{\theta_c}(u_e, v_e)^2. \end{aligned} \tag{6.78}$$

In figures 6.4 and 6.5, we show the results of formulae (6.77) and (6.78) without the squares, i.e. the projection of $\mathbf{1}_N$

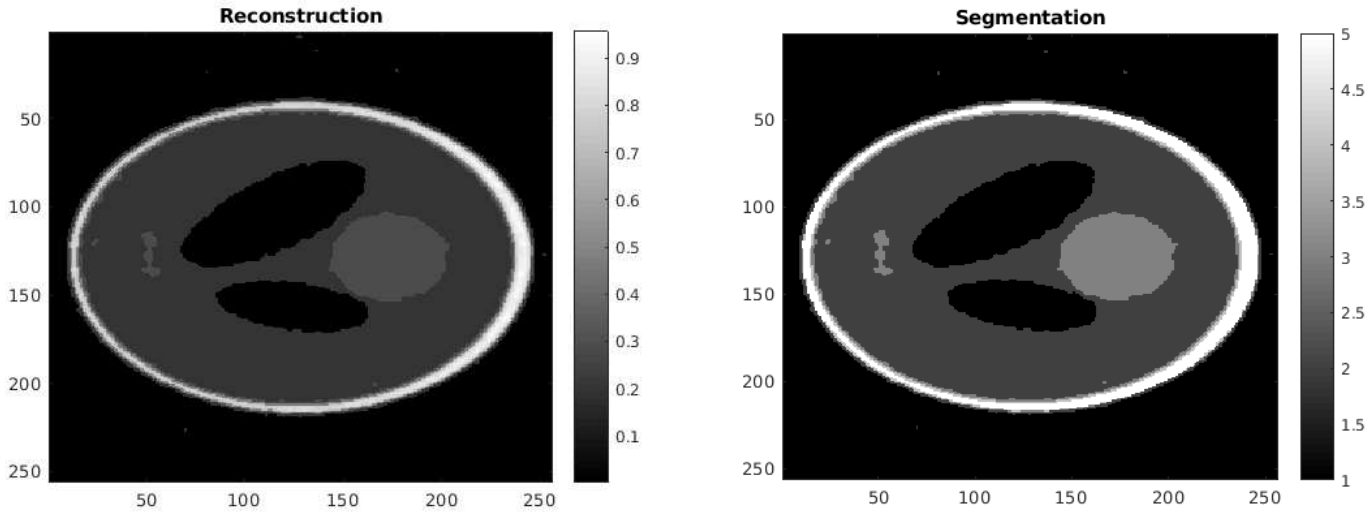
$$[\boldsymbol{H}\mathbf{1}_N]_i = l_{\theta_c}(u_e, v_e) \sum_{x_e} \sum_{y_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e), \tag{6.79}$$

and the backprojection of $\mathbf{1}_M$

$$\left[\boldsymbol{H}^T\mathbf{1}_M\right]_j = \sum_{\phi} \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) \sum_{u_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) l_{\theta_c}(u_e, v_e), \tag{6.80}$$

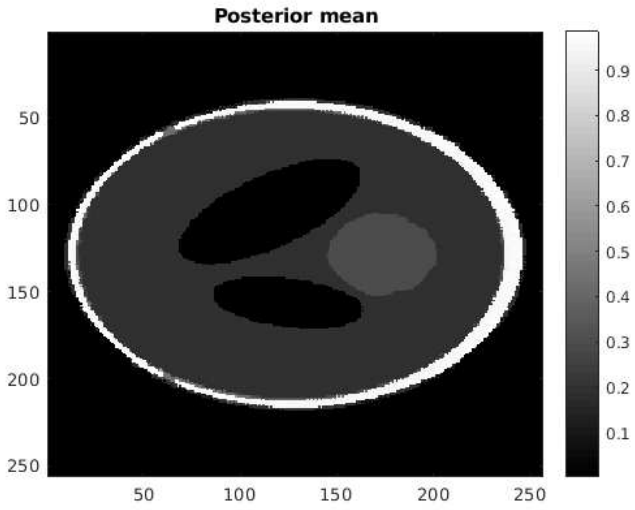
respectively. Comparing these figures to figures 6.2 and 6.3, we see that the orders of magnitude of the coefficients of $\boldsymbol{H}^{(2)}\mathbf{1}_N$ and $\boldsymbol{H}^{(2)^T}\mathbf{1}_M$ are very different from the ones of $\boldsymbol{H}\mathbf{1}_N$ and $\boldsymbol{H}^T\mathbf{1}_M$, since H_{ij}, which is squared in $\boldsymbol{H}^{(2)}$, approximately corresponds to the length crossed by ray i in voxel j, which is very little. The use of the squared projector and backprojector is not limited to the application of VBA to 3D CT. These operators can also be useful to implement regularizers such that the resolution of the obtained penalized-likelihood estimators is nearly space-invariant, for instance in 3D Positron Emission Tomography (PET) [FR96].

We now apply algorithm 15 to simulated data, and compare the estimated PM with JMAP described in chapter 4. For our comparisons, JMAP is performed by algorithm 12 and applied with the SF pair.

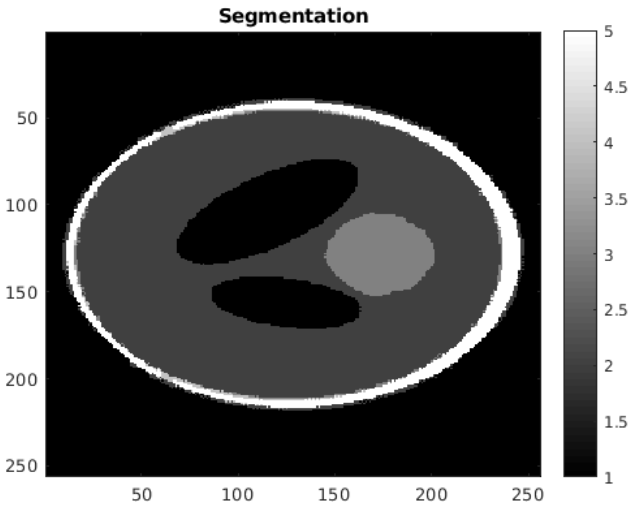


(a)

(b)



(c)



(d)

Figure 6.6: Joint reconstructions and segmentations by JMAP (a-b) and VBA (c-d) algorithms

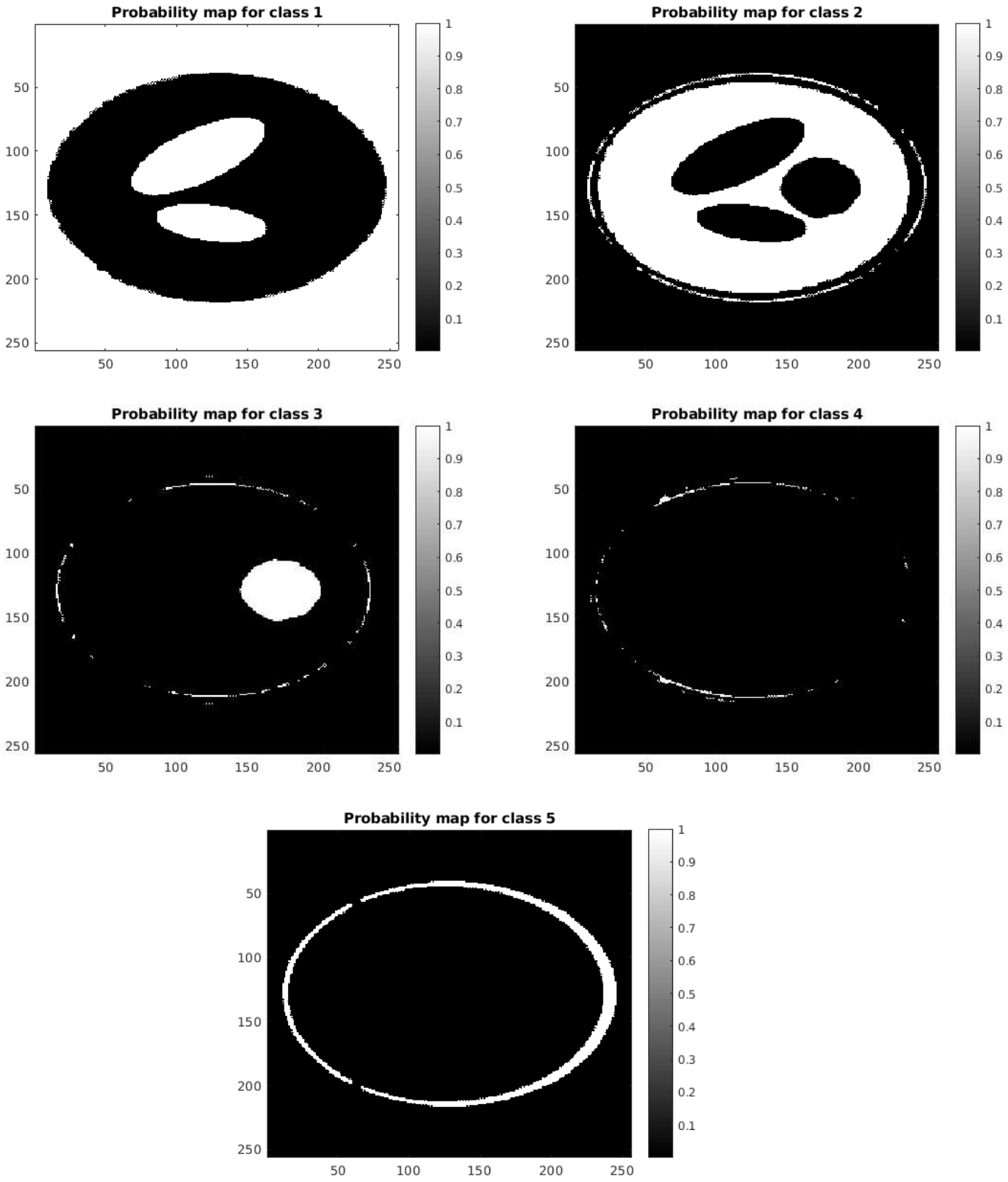


Figure 6.7: Probability maps for each voxel to be in each class, estimated by the VBA algorithm

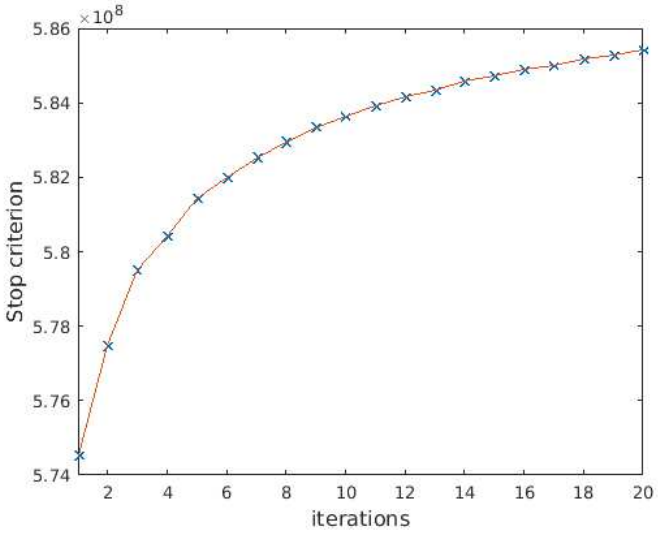


Figure 6.8: Convergence of the VBA algorithm

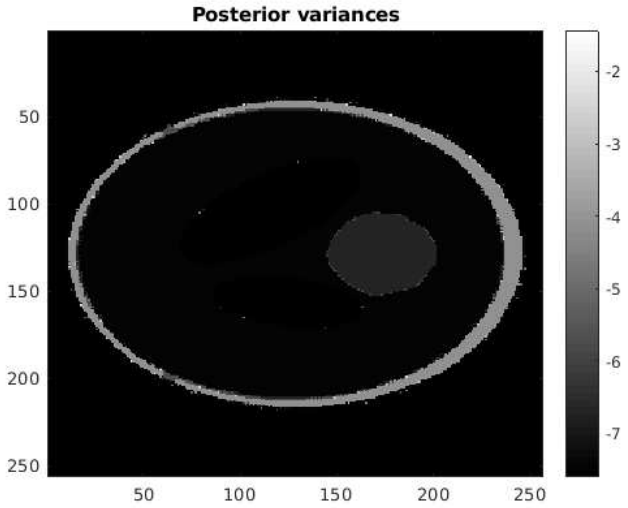


Figure 6.9: Variances obtained by the VBA algorithm jointly with the reconstruction. The variances are shown in logarithmic scale.

6.4 Simulation results

We test algorithm 15 on the Shepp-Logan phantom, already used in chapters 3 and 4. The source-to-detector distance is 230 mm, and the source-to-object distance is 98 mm. The phantom has $K = 5$ classes. The field-of-view is size $6.05 \times 6.05 \times 5.86$ mm^3 and is discretized into 256^3 voxels. We reconstruct it from 64 projections of size 256^2 pixels, uniformly distributed over $[0, 2\pi]$. The projections are noisy with a signal-to-noise ratio equal to 20 db.

Concerning the fixation of the parameters of the VBA algorithm, we use the same strategies as the ones explained in chapter 4 for $\boldsymbol{\alpha}$, γ_0, m_0 and v_0. In the experiments we conducted, we saw that the ways to fix $(\alpha_{\zeta_0}, \beta_{\zeta_0}, \alpha_0, \beta_0)$ for JMAP lead to very bad results for VBA. This is due to the fact that we do not compute the same estimator, which is the PM in VBA and the MAP in JMAP. Consequently, the parameters $(\alpha_{\zeta_0}, \beta_{\zeta_0}, \alpha_0, \beta_0)$ of the VBA algorithm cannot be tuned in the same way as in JMAP. Experimentally, we found the best results when fixing α_{ζ_0}, β_{ζ_0}, α_0 and β_0 near the Jeffreys prior [Jef46]. Nevertheless, during our experiments, we have noticed that the VBA algorithm is more sensitive to the choice of the parameters than JMAP. Table 6.1 gives the used values for the parameters, except for m_0 and $\boldsymbol{\alpha}$ which are fixed automatically as in chapter 4. For JMAP, the parameters are fixed to the same values used in section 4.4.

Parameters	K	γ_0	v_0	α_{ζ_0}	β_{ζ_0}	α_0	β_0
Values	5	6	1	10^{-4}	10^{-2}	10^{-6}	10^{-2}

Table 6.1: Parameters for the VBA algorithm

Figure 6.6 shows the reconstructions obtained by JMAP and VBA respectively. Thanks to the use of Gauss-Markov-Potts prior model, JMAP and VBA reconstructions have compact and well-distinguishable regions. We see that the reconstruction by VBA has smoother contours than the one by JMAP. Nevertheless, in table 6.2, we see that the VBA algorithm obtains a higher error $\Delta_2 \boldsymbol{f}$ than JMAP, as a lower Rand index [Ran71]. Indeed, in figure 6.6, we see that, compared to JMAP, some details are lost by the VBA algorithm. This is due to the factorized form of the approximate posterior distribution, which overlooks correlations between neighbouring voxels.

The evolution of the stopping criterion (F.65) is shown in figure 6.8. As we see, it increases through the iterations. As presented in chapter 4, one iteration of JMAP contains 20 sub-iterations for the volume estimation and few sub-iterations for the segmentation step. The VBA algorithm does not have sub-iterations. Consequently, in table 6.2, the computation time of the VBA algorithm is much less than the one of JMAP. Nevertheless, our experiments showed that the VBA algorithm has a high sensitivity to the number of iterations. Indeed, for a too large number of iterations of VBA, the reconstruction is over-regularized. This is a drawback of the VBA algorithm compared to JMAP. In

Algorithm	Computation time	$\Delta_2 \boldsymbol{f}$	$\Delta_2 \boldsymbol{g}$	Rand index
JMAP	873.1 s	**8.81 %**	**0.66 %**	**94.7 %**
VBA	**150.0 s**	13.5 %	2.67 %	92.8 %

Table 6.2: Comparaison of JMAP and VBA algorithms

addition, the memory cost of the VBA algorithm is much higher than the one of JMAP. This makes it only applicable to small regions-of-interest (ROI), typically of size 256^3. Based on a reconstruction of high quality (for instance, obtained by JMAP), the reconstruction of a ROI can be performed following the method of [ZNG08], as done for instance in [LFB10]. Combining this method with the VBA algorithm should enable to estimate the uncertainties in a ROI. This point will be covered in future works.

By the estimation of $q_{z_j}(k)$, for all $j \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$, the VBA algorithms returns K maps giving the probability for each voxel to be in each class. These probability maps are shown in figure 6.7. The variances of the posterior distribution of the volume estimated by VBA are shown in figure 6.9. Unsurprisingly, the highest variances are on the thinest part of the phantom which is the bone. Nevertheless, the loss of details in the reconstruction is not highlighted by the posterior variances. This is due to the fact that uncertainties are under-estimated in VBA when considering Kullback-Leibler divergence $KL(q||p)$ [GBJ15]. Considering the reversed Kullback-Leibler divergence $KL(p||q)$, like in Expectation-Propagation (EP) [Min01, Min05, PSC+16], should lead to a better estimation of the uncertainties. Nevertheless, minimizing $KL(p||q)$ is more difficult since this divergence is an integral with respect to p which is unknown. In addition, EP uses an approximate disitribution q which has the same factorization than the true posterior distribution p. This is not the case for our approximate distribution (6.42). Consequently, minimizing $KL(p||q)$ instead of $KL(q||p)$ considering our chosen form for q remains an open problem.

6.5 Conclusion and perspectives

In this chapter, we have presented an application for 3D X-ray CT of variational Bayesian approach (VBA) with Gauss-Markov-Potts prior model. We have presented new updating formulae for partially factorized approximate distributions preserving a dependence between variables. By computing the posterior mean (PM) thanks to VBA, we have been able to jointly perform the reconstruction and the estimation of the posterior variances, which give the uncertainties on the reconstruction. In order to compute these variances, we have seen that the huge dimension in 3D X-ray CT hinders to easily get diagonal coefficients, due to the fact that projection and backprojection operators cannot be stored in memory. In order to tackle this problem, we have taken benefit from the use of a matched pair of projector and backprojector, which was the Separable Footprint (SF) one : based on this pair, we have implemented "squared" projector and backprojector which have enabled us to compute diagonal coefficients on-the-fly. The GPU implementation for these squared operators was the same proposed for the SF P/BP pair in chapter 3.

Our tests on simulated data and comparisons with joint maximization a posteriori (JMAP) have shown that VBA obtains smoother contours than JMAP and converges faster. Although the memory cost of VBA is higher than the one of JMAP, we have underlined that the algorithm can be applied in order to estimate the uncertainties in a small region-of-interest (ROI). Nevertheless, our experiments have also highlighted that the estimation of the uncertainties shall be enhanced. The main part of the future works will focus on developing other variational Bayesian algorithms in order to get a better estimation of the uncertainties.

Applications of the algorithm to real data also remain to do. Indeed, we have shown that using a matched pair such as the SF pair detailed in chapter 3 is necessary to apply the VBA algorithm. Consequently, before dealing with real data of huge size, the GPU implementation of the SF pair have to be fully optimized and adapted to large volumes as explained in the conclusion of chapter 3. Once this work will be done, experiments on real data will be performed. These further tests will be an opportunity to find optimal strategies in order to fix the parameters of the VBA algorithm.

Chapter 7

Conclusion and perspectives

In this thesis, we have proposed several iterative reconstruction methods in X-ray Computed Tomography (CT), based on a Bayesian approach, in order to inspect piecewise-constant industrial parts produced by SAFRAN. In order to model the attenuation of X-rays through the to-be-imaged volume, in chapter 3, we have studied a matched pair of projector and backprojector (P/BP), which was the Separable Footprint (SF) [LFB10], and an unmatched P/BP pair, with a ray-driven (RD) projector and a voxel-driven (VD) backprojector. Both these pairs have been validated on Graphical Processor Unit (GPU) in order to massively parallelize the computations and, therefore, to reconstruct volume of huge dimensions in reasonable times. In particular, we have proposed a new GPU implementation of the SF pair in order to minimize the memory transfers between the CPU and the GPU. As our first implementation of the SF projector run two kernels depending on the main direction of the rays, we have proposed a method in order to merge these kernels into only one. As pointed out in the conclusion of chapter 3, the perspectives for the GPU implementation of the SF pair are to adapt it to very large volumes and to compare it with the other existing GPU implementations [XML$^+$17]. After this work, it will be interesting to compare the effects of using a matched or an unmatched P/BP pair on real data, and to see if the matched pair brings a gain in precision and robustness.

After the validation of the used P/BP pairs, in chapter 4, a new algorithm for joint reconstruction and segmentation, based on a Gauss-Markov-Potts prior model on the volume, has been detailed. The algorithm mainly consists in alternating a reconstruction and a segmentation steps. As advantages compared to other joint reconstruction and segmentation approaches [BS11, SWFU15, ZPB16], the algorithm estimates the means and the variances of the classes in a very simple way. In addition, we have underlined that the algorithm has a low memory cost. The algorithm has been shown to give results of good quality, both in simulation and on real data provided by SAFRAN. Nevertheless, on challenging data like the Composite Material with Titanium on the Edges (CMTE), we have seen that the segmentation step fails in removing the scattering due to the titanium, since the attenuation coefficients due to the scattering are very close to the ones of the composite material. In order to reduce this scattering, in section 4.6, we have suggested a method which would be interesting to test in future works. Based on a segmentation which enables to locate the titanium in the field-of-view, an iterative method would be applied in order to reconstruct the composite material and to avoid artifacts thanks to the knowledge about the position of the titanium. The segmentation would be provided by the algorithm presented in chapter 4, but it may alternatively be the Computer-Assisted-Design (CAD) of the part. Nevertheless, using the CAD requires to register it in the field-of-view, which is not easy to do in an unsupervised way. In any case, another benefit from the segmentation would be to weight the projections depending on how much there are scattering or absorbing materials in each direction.

Although the proposed algorithm in chapter 4 has many parameters, we have been able to explain the role of each parameter thanks to our Bayesian approach and to give strategies in order to fix it. Nevertheless, optimally tuning the number of sub-iterations executed during the reconstruction step and the segmentation step remains difficult and often requires repeated experiments on challenging data. In addition, the impact of K needs to be deepened in the perspective of non-destructive testing. Indeed, in this work, K has been fixed to its known theoretical value. Making this choice, experiments still have to be conducted in order to make sure that, in the case when extra materials are present, the reconstruction method does not hide it, or, at least, diverges such that we are able to identify that something is going wrong in the inspected volume. At first sight, a safe way would be to fix K to a far over-estimated value, but, doing so, we have seen in further experiments (not presented in this thesis) that the quality of the final

reconstruction is degraded and not better than a filtered backprojection. This is explained by the fact that, when over-estimating K, the scattering and the beam-hardening can be identified as classes, hence they are not removed during the segmentation process. Another possible way would be to adapt K to an optimal value estimated during the execution of the algorithm itself. Nevertheless, given the reconstruction method, this appears difficult to do, and remains a completely open problem. In any case, the estimation of K will require much more experiments. In order to save computation time, the segmentation step of the algorithm will have to be parallelized on GPU. Concerning this point, a valuable point for industrialization and effective use will be to implement the full method on multi-GPU. This will require to manage the data transfers between GPUs in an effective way, as it is done for instance in [MWF18].

In chapter 4, we have also highlighted that the parameters related to the uncertainties on the projections are not easy to tune, due to the fact that the forward model used in chapter 4 is derived from a monochromatic model of X-rays, while they are actually polychromatic. For this reason, in chapter 5, we have inserted an error term in the mean of the Poisson statistics of photon-counts, standing for the deviation introduced by the monochromatic model of the rays with respect to the polychromatic one. Following the same route as [SB93], we have derived a new forward model for X-ray CT, called the error-splitting forward model, since it splits the uncertainties on the projections into two terms. The first term has appeared as measurement uncertainties which can be well-modeled by a Gaussian distribution. The second term has been shown as playing the role of a bias in the forward model, hence it has been said to correspond to linear model uncertainties. In our derivation of the error-splitting forward model, no clear statistics has appeared for linear model uncertainties, contrary to measurement uncertainties. As a consequence, we have proposed to assign a heavy-tailed prior on these uncertainties, thanks to the use of the generalized Student-t distribution. The choice for a heavy-tailed prior has been motivated by the fact that we wanted to enforce robustness with respect to outliers corresponding to strong scattering and beam-hardening. Based on the error-splitting forward model, we have proposed a general reconstruction algorithm, applicable with any proper prior model on the volume. Then, we have detailed its combination with Gauss-Markov-Potts prior. In our experiments on real data of SAFRAN, the error-splitting forward model with Gauss-Markov-Potts prior has been shown to combine robustness and accuracy, while only one of these features was achievable with the usual forward model previously used in chapter 4. A perspective for this work is to use the error-splitting forward model with priors on the volume different from Gauss-Markov-Potts, in order to see if it brings the same advantages. Since the reconstruction method proposed in chapter 5 is only valid with proper priors on the volume, it will be also interesting to find a way to combine the error-splitting forward model with improper priors such as minimum Total Variation (TV) [CP11, OMBF18]. As mentioned in chapter 5, several improvements of the error-splitting forward model shall be studied, such as enforcing a nonpositivity constraint on linear model uncertainties, or assigning different priors on measurement uncertainties depending on the signal-to-noise ratio in each ray. Another way to enhance the reconstruction quality would be to combine the proposed methods with first-order beam-hardening correction approaches often used in clinical CT as a pre-processing step [GDMJ$^+$16]. Since the object materials are known, a first-order correction of beam-hardening would consist in correcting the non-linearities in the data due to the dominant material [BDC76]. Therefore, the methods proposed in this thesis would only have to remove the second-order effects due to the other materials. Equivalently to water correction in clinical CT, the first-order correction would require the use of a calibration phantom only made of the dominant material of the inspected part [KSK06].

Since the reconstruction obtained by an iterative algorithm is only an estimation of the true volume, in chapter 6, we have proposed an algorithm which performs an estimation of the uncertainties on the reconstruction. This algorithm is an instance of Variational Bayesian Approach (VBA), about which we have given details in chapter 6 and for which we have derived new updating formulae when the chosen form of the approximate posterior distribution preserves a dependence between some variables, as it has been the case between the attenuation coefficients and the labels with Gauss-Markov-Potts prior model on the volume. In order to compute the approximate diagonal coefficients of the posterior covariance matrix in a reasonable time, we have also highlighted that it is necessary to use a matched P/BP pair. To this end, we have taken benefit from our GPU implementation of the SF pair presented in chapter 3. In our experiments on simulated data, it has appeared that, although the algorithm gives quite good results, the uncertainties are under-estimated. In order to improve this estimation, other VBA algorithms, for instance minimizing the reversed Kullback-Leibler divergence, remain to study.

The presented algorithms in this thesis have focused on reconstructing a particular class of industrial parts which are quasi-piecewise-constant. These parts are not the only ones made by SAFRAN. Another interesting class is the

class of woven composite materials, which have not been considered in this thesis. As pointed out at the end of chapter 4, Gauss-Markov-Potts prior is not suited to these parts since it over-smoothes the weaving. Consequently, finding a tractable prior model for these parts in order to reconstruct it by iterative reconstruction methods is a possible future work. For this purpose, it will be interesting to re-use the error-splitting forward model in order to gain in precision and robustness.

At last, we may underline that there are industrial parts at SAFRAN which are too big to entirely lie in the field-of-view. This is a problem because, in such cases, there is information in the projections which is related to voxels out of the field-of-view. This problem in CT corresponds to the field of local tomography [PM17, PDM17]. Adapting the algorithms proposed in this thesis to the context of local tomography in order to apply it on big parts made by SAFRAN can be studied. Nevertheless, the markovian nature of the prior model on the volume may lead to difficulties when dealing with the voxels on the border of the field-of-view.

Publications

Journal papers

- *Error-Splitting Forward Model for Iterative Reconstruction in X-ray Computed Tomography and application with Gauss-Markov-Potts prior*, Camille Chapdelaine, Ali Mohammad-Djafari, Nicolas Gac, Estelle Parra; *IEEE Transactions on Computational Imaging*, accepted for publication.
- *A 3D Bayesian Computed Tomography Reconstruction Algorithm with Gauss-Markov-Potts Prior Model and its Application to Real Data*, Camille Chapdelaine, Ali Mohammad-Djafari, Nicolas Gac, Estelle Parra; 2017, *Fundamenta Informaticae*, 155(4), 373-405.

International conferences

- *Multi-streaming and multi-GPU optimization for a matched pair of Projector and Backprojector*, Nicolas Georgin, Camille Chapdelaine, Nicolas Gac, Ali Mohammad-Djafari, Estelle Parra; *The 15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, accepted, 2019.
- *Bayesian inference with error variable splitting and sparsity enforcing priors for linear inverse problems*, Ali Mohammad-Djafari, Mircea Dumitru, Camille Chapdelaine, Nicolas Gac; *The 26th European Signal Processing Conference*, pages 440-444, 2018.
- *New GPU implementation of Separable Footprint (SF) Projector and Backprojector : first results*, Camille Chapdelaine, Nicolas Gac, Ali Mohammad-Djafari, Estelle Parra; *The 5th International Conference on Image Formation in X-Ray Computed Tomography*, pages 314-317, 2018.
- *Joint Reconstruction and Segmentation of Real 3D Data in Computed Tomography thanks to a Gauss-Markov-Potts Prior Model*, Camille Chapdelaine, Ali Mohammad-Djafari, Nicolas Gac, Estelle Parra; *The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pages 810-813, 2017.
- *A joint Segmentation and Reconstruction Algorithm for 3D Bayesian Computed Tomography using Gauss-Markov-Potts prior model*, Camille Chapdelaine, Ali Mohammad-Djafari, Nicolas Gac, Estelle Parra; *The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017 (PhD forum).

Acknowledgements

This work would not have been possible without the help of my supervisors. First, I would like to express my gratitude to Ali Mohammad-Djafari. Thanks to him, I discovered the field of Bayesian approaches. Ali really gave me the ability to understand each of the mathematical tools I was dealing with and I think it is the most important thing if one wants to make progress in his or her field of research. I am also grateful to Ali for having been an ambitious supervisor, always interested in exploring new areas. In particular, I remember the day when we had the idea of splitting the uncertainties in the forward model. This work has surely been the main contribution of this thesis. I also would like to thank Ali for having very soon encouraged me to publish my work in selective journals and conferences, as to make contact with other researchers.

I am also grateful to Nicolas Gac, without whom I would not have been able to deal with GPU programming. Nicolas has been a very patient professor, always available to answer to my questions or to solve problems with my code. I also want to thank Nicolas for his deepened insight into how a computer program works and for his ability to explain it in a very simple and rigorous way. I am particularly grateful to Nicolas for our fruitful work on the Separable Footprint pair : it has been a very interesting part of my PhD studies and I think we still have work to do about the fascinating problem of matched pairs for 3D Computed Tomography.

I also want to express my gratitude to Estelle Parra. She really helped me to make progress in the presentations of my work and to focus on the essential points. In addition, she has been a great manager, always eager to explore novel approaches in the interest of SAFRAN. She also helped me to discover all the interesting activities of SAFRAN. As a new member of the Signal and Information Processing team at Safran Tech, I hope I will have many occasions to work with her in the future. I also want to thank Estelle for her trust and for having given me the opportunity to supervise an internship at the end of my PhD studies while I was writting my dissertation.

Concerning this internship, I would like to thank Nicolas Georgin. I really appreciated to mentor him during six months. Nicolas did a very good work to accelerate my GPU implementation of the Separable Footprint pair and to adapt it to multi-streaming and multi-GPU computations. He has been a very impressive student, who very fast understood all the different issues in X-ray Computed Tomography. I am happy that he could present his work in a conference paper. I hope we will have further opportunities to work together.

I also thank Charles Soussen for having accepted to supervise my PhD studies after Ali's retirement in June 2018. He gave me very valuable advices to prepare my PhD defence. Charles, with Nicolas, also did his best to compose my doctoral committee. At this step, I would like to thank all the committee members for their careful review of my dissertation and their very interesting questions during the defence. First, I am very grateful to Professor Jeffrey Fessler and Professor Jean-François Giovannelli for their very detailed reports on my dissertation. In addition, I would also like to thank Professor Jeffrey Fessler for our discussions on the Separable Footprint pair and on image reconstruction algorithms. I equally thank the other examiners of the doctoral committee : Professor Thomas Rodet, Simon Rit and Sabine Rolland du Roscoat. I really appreciated to discuss with them and to answer to their questions during the defence.

I thank L2S members, PhD students, post-docs and interns with whom I discussed a lot each time I went to the laboratory : François Orieux, Patrice Brault, Li Wang, Mircea Dumitru, Amine Hadj-Youcef, Mickaël Seznec, Guillaume Révillon, Thi-Thanh Nguyen, Maxime Martelli, Raphael Chinchilla, Youssouf Sameut Bouhaik, Fangchen Feng, Alina Meresescu and Chao Zhang. I also thank Edward Romero, Alexis Huck and Yann Le Guilloux at Safran Tech for the times we discussed on my work and on its applications. I would like to thank Lionel Gay, Nicolas Cochennec and Clément Remacha who provided me the real data for my experiments. I also want to thank all the members of the Image Processing for Manufacturing and NDT team led by Estelle : Véronique Brion, Nacim Belkhir, Marie Kuate Lienou, Arturo Mendoza Quispe, Jorge Hernandez Londono, Vincent Morard, David Tourais, Haithem Boussaid,

Julian Betancur, Roger Trullo and Gaël Michelin. I also express my gratitude to my former professors at Supélec : Laurent Le Brusquet, Emmanuel Vazquez, Stéphane Font, José Picheral, Arthur Tenenhaus, Elisabeth Lahalle, Gilles Chardon, Julien Bect, Hani Hamdan, Hana Baili, Michel Kieffer, Frédéric Pascal, Vincent Lescarret and Pascal Bondon. I am also grateful to Frédéric Desprez, Céline Labrude and Thomas Cuidu for their help to use the GPUs of the laboratory. I would like to thank other L2S members : Maryvonne Giron, Stéphanie Douesnard, Aurélia Fraysse, Matthieu Kowalski, Pascale Debever, Laurence Antunes, Sami Tliba, Gilles Duc, Anne Batalie and Silviu Niculescu; as my other colleagues at SAFRAN : Daniel Duclos, Sylvaine Picard, Olivier Ghibaudo, Stéphane Amiel, Benoît Gérardin, Alexiane Arnaud, Philippe Rodrigues, Mathieu Ducousso, Frédéric Jenson, Thierry Le Polles, Michel Moukari, Mehdi Regina, Jennifer Vandoni, Yann Lifchitz, Héléna Vorobieva, Andrei Bursuc, Ahmed Benaichouche, Franck Michaud, Axel Barrau, Elisa Hubert, Yosra Marnissi, Sébastien Razakarivony, Paul Chauchat, Edouard Pineau, Alexandre Reiffers-Masson, Gianni Franchi, Marc Lelarge, Dany Abboud, Lennart Gulikers, Alonso Silva Allende, Sin-Seok Seo, Morgane Barbet-Massin, Nicolas Fouquet, Yrvann Emzivat, Olivier Brune, Amadou Assoumane, Dohy Hong, Mohamed El Badaoui, Sébastien Da Veiga, Murielle Garcia, Mathilde Coutelier and Jeff Innocent. I also thank Stéphane Roux at LMT and Aymeric Reshef at GE Healthcare.

I would like to thank all my friends and my family. In particular, I thank Rémi Bisognin with whom I closely worked on the development of a deconvolution algorithm for his PhD studies. I thank his supervisor, Gwendal Fève, and Ali for their enthusiasm during this unexpected and very pleasant collaboration. Last but not least, I would like to deeply thank my parents Patrick and Jocelyne Chapdelaine, my grandparents Roger and Raymonde Bellot and Raymonde and Philippe Chapdelaine, and my twin brother Kévin Chapdelaine.

Appendix A

Derivation of the algebraic forward model used in X-ray CT

As explained in chapter 1, the Poisson statistics of photon-counts reads

$$p(I_i|\boldsymbol{f}, r_i) = \mathcal{P}\left(I_0 e^{-[\boldsymbol{Hf}]_i} + r_i\right) \tag{A.1}$$

where I_0 is the mean number of photons sent by the source and r_i is the mean number of background events. We have

$$-\ln(p(I_i|\boldsymbol{f}, r_i)) = \ln(I_i!) + I_0 e^{-[\boldsymbol{Hf}]_i} + r_i - I_i \ln\left(I_0 e^{-[\boldsymbol{Hf}]_i} + r_i\right). \tag{A.2}$$

The reconstruction problem is linearized considering

$$g_i = \ln\left(\frac{I_0}{I_i - r_i}\right), \forall i. \tag{A.3}$$

Since many photons reach the detector, we can write a second-order Taylor expansion [SB93] :

$$\begin{aligned} I_0 e^{-[\boldsymbol{Hf}]_i} + r_i &= (I_i - r_i)\exp\left[g_i - [\boldsymbol{Hf}]_i\right] + r_i \\ &= I_i + (I_i - r_i)(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}(I_i - r_i)(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right). \end{aligned} \tag{A.4}$$

Hence, we have

$$\begin{aligned} I_i \ln\left(I_0 e^{-[\boldsymbol{Hf}]_i} + r_i\right) &= I_i \ln\left((I_i - r_i)\exp\left[g_i - [\boldsymbol{Hf}]_i\right] + r_i\right) \\ &= I_i \ln\left(I_i + (I_i - r_i)(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}(I_i - r_i)(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right) \\ &= I_i \left[\ln(I_i) + \ln\left(1 + \frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right)\right] \\ &= I_i \left[\ln(I_i) + \frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i)^2 \right. \\ &\qquad \left. -\frac{1}{2}\left(\frac{I_i - r_i}{I_i}\right)^2 (g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \\ &= I_i \left[\ln(I_i) + \frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\left[\frac{I_i - r_i}{I_i} - \left(\frac{I_i - r_i}{I_i}\right)^2\right](g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \\ &= I_i \left[\ln(I_i) + \frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{I_i - r_i}{I_i}\left[1 - \frac{I_i - r_i}{I_i}\right](g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \\ &= I_i \left[\ln(I_i) + \frac{I_i - r_i}{I_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{(I_i - r_i)r_i}{I_i^2}(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \end{aligned}$$

$$= I_i \ln(I_i) + (I_i - r_i)(g_i - [\boldsymbol{H}\boldsymbol{f}]_i) + \frac{1}{2}\frac{(I_i - r_i)r_i}{I_i}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right). \tag{A.5}$$

Gathering (A.2), (A.4) and (A.5), we have a cancellation of first-order terms :

$$\begin{aligned} -\ln(p(g_i|\boldsymbol{f}, r_i)) &= \ln(I_i!) + I_i - I_i \ln(I_i) + \left[(I_i - r_i) - (I_i - r_i)\right](g_i - [\boldsymbol{H}\boldsymbol{f}]_i) \\ &\quad + \frac{1}{2}\left[(I_i - r_i) - \frac{(I_i - r_i)r_i}{I_i}\right](g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right) \qquad \text{(A.6)} \\ &= \ln(I_i!) + I_i - I_i \ln(I_i) + \frac{1}{2v_{\zeta_i}}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right) \qquad \text{(A.7)} \end{aligned}$$

where

$$v_{\zeta_i} = \left[(I_i - r_i) - \frac{(I_i - r_i)r_i}{I_i}\right]^{-1} = \left[(I_i - r_i)\left(1 - \frac{r_i}{I_i}\right)\right]^{-1} = \frac{I_i}{(I_i - r_i)^2}. \tag{A.8}$$

As a result, the log-likelihood for the linearized reconstruction problem reads

$$\ln(p(g_i|\boldsymbol{f}, v_{\zeta_i})) \approx C_i - \frac{1}{2v_{\zeta_i}}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 \tag{A.9}$$

where

$$C_i = -\left[\ln(I_i!) + I_i - I_i \ln(I_i)\right] \tag{A.10}$$

does not include $\boldsymbol{f}$. Because of the quadratic term in (A.9), forward model

$$g_i = [\boldsymbol{H}\boldsymbol{f}]_i + \zeta_i \;\text{ with }\; p(\zeta_i|v_{\zeta_i}) = \mathcal{N}(\zeta_i|0, v_{\zeta_i}) \tag{A.11}$$

leads to the same log-likelihood, up to an additive constant. Hence, the following vectorized linear forward model accounting for uncertainties can be used for X-ray CT [SB93] :

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\zeta}, \tag{A.12}$$

where $\boldsymbol{\zeta}$ is modeled as Gaussian.

Appendix B

Projection and backprojection algorithms

Several algorithms have been proposed to compute the projection of a volume [Jos82, Sid85, DMB04, LFB10]. A standard and fast ray-tracing method is Siddon's method, which follows the ray into the volume and calculates the intersection length with the encountered voxels. According to the discretization of Beer-Lambert law, for each pierced voxel, this intersection length is the contribution of the voxel to the projection, i.e. coefficient H_{ij} of the projection matrix is equal to the intersection length between ray i and voxel j. Therefore, Siddon's method models the rays as pencil beams. In Siddon's method, the intersection length with each of these voxels is calculated very fast by differentiating pre-calculated coefficients. Fast versions of the method compute these coefficients on-the-fly [JSDS$^+$98, HLY99], which leads to Siddon-Jacobs-Han's projector described in section B.1. Nevertheless, this projector has strong limitations in the context of iterative reconstruction methods. Indeed, to compute the adjoint Siddon-Jacobs-Han's backprojection in one voxel, all the rays which have pierced this voxel during the projection simulation have to be identified. Due to the iterative nature of Siddon-Jacobs-Han's projector, this is not easy to do and leads to very intensive calculations which make the adjoint Siddon-Jacobs-Han's backprojection very slow, even parallelized on the GPU [NL15]. In order to alleviate this computational cost, the Separable Footprint (SF) pair [LFB10], described in section B.2, approximates the footprint of a voxel onto the detector in order to make the calculations efficient both for the projection and the backprojection. For the sake of readability, the used geometrical model presented in section 2.1 is re-illustrated in figure B.1.

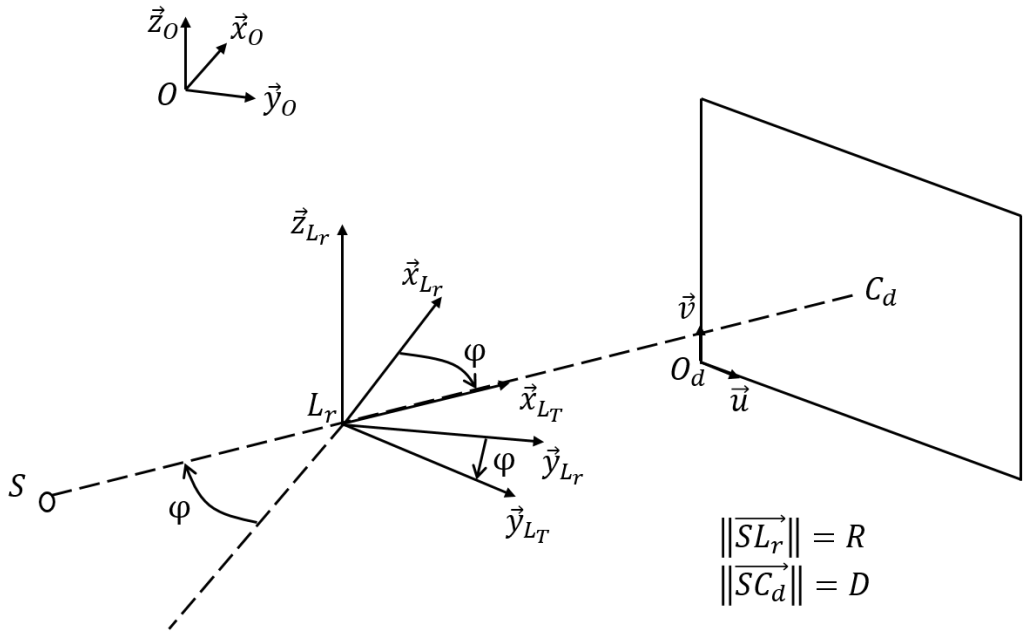


Figure B.1: Geometrical model

B.1 Fast Siddon's ray-tracing method

A popular method to compute the radiological path of a ray in a fast way has been introduced by Siddon [Sid85]. We consider a ray $i \in \{1, \dots, M\}$ which is projected on a cell $P(u_e, v_e)$ of the detector at projection angle ϕ. The coordinates of P in the frame $(C_d, \vec{u}, \vec{v})$ are denoted by

$$\left\{ \begin{array}{ll} s & = s_{u_e} \times \delta_u = (u_e - u_{C_{d_e}}) \times \delta_u \\ t & = t_{v_e} \times \delta_v = (v_e - v_{C_{d_e}}) \times \delta_v \end{array} \right. , \tag{B.1}$$

where

$$\overrightarrow{C_d P} = s\vec{u} + t\vec{v}. \tag{B.2}$$

Given that the detector is rotated by ϕ with respect to the optical frame

$$\left\{ \begin{array}{l} \vec{u} = -\vec{x}_{Lr} \sin\phi + \vec{y}_{Lr} \cos\phi = -\vec{x}_O \sin\phi + \vec{y}_O \cos\phi \\ \vec{v} = \vec{z}_{Lr} = \vec{z}_O \end{array} \right. , \tag{B.3}$$

the coordinates of cell P in the optical frame are

$$\left\{ \begin{array}{l} x(u_e, \phi) = x_{Lr} + (D - R)\cos\phi - s\sin\phi \\ y(u_e, \phi) = y_{Lr} + (D - R)\sin\phi + s\cos\phi \\ z(v_e) = z_{Lr} + t \end{array} \right. . \tag{B.4}$$

Given the distance between the source and the cell

$$L = \left((x(u_e, \phi) - x_S(\phi))^2 + (y(u_e, \phi) - y_S(\phi))^2 + (z(v_e) - z_S(\phi))^2 \right)^{\frac{1}{2}}, \tag{B.5}$$

the parametric equation of ray i between the source and cell P at projection angle ϕ is [HLY99]

$$\left\{ \begin{array}{l} x(\lambda) = x_S(\phi) + \frac{\lambda}{L}(x(u_e, \phi) - x_S(\phi)) \\ y(\lambda) = y_S(\phi) + \frac{\lambda}{L}(y(u_e, \phi) - y_S(\phi)) \\ z(\lambda) = z_S(\phi) + \frac{\lambda}{L}(z(u_e, \phi) - z_S(\phi)) \end{array} \right. , \ \lambda \in [0, L] \tag{B.6}$$

which gives

$$\left\{ \begin{array}{l} x(\lambda) = x_{Lr} - R\cos\phi + \frac{\lambda}{L}(D\cos\phi - s\sin\phi) \\ y(\lambda) = y_{Lr} - R\sin\phi + \frac{\lambda}{L}(D\sin\phi + s\cos\phi) \\ z(\lambda) = z_{Lr} + \frac{\lambda}{L} \times t \end{array} \right. , \ \lambda \in [0, L] . \tag{B.7}$$

For a voxel j with discrete coordinates (x_e, y_e, z_e), Siddon's method defines three lengths :

$$\left\{ \begin{array}{ll} \lambda_x(x_e) & = \frac{L}{x(u_e,\phi) - x_S(\phi)} (x_e \times \delta - x_S(\phi)) \\ \lambda_y(y_e) & = \frac{L}{y(u_e,\phi) - y_S(\phi)} (y_e \times \delta - y_S(\phi)) \\ \lambda_z(z_e) & = \frac{L}{z(v_e) - z_S(\phi)} (z_e \times \delta - z_S(\phi)) \end{array} \right. . \tag{B.8}$$

Figure B.2 shows a representation of these lengths, in 2D for the sake of readability. With this figure, the intersection length between ray i and voxel j appears to be [Sid85, JSDS+98, HLY99]

$$l = \mathbf{1}_{\{\lambda_{max}(x_e,y_e,z_e) > \lambda_{min}(x_e,y_e,z_e)\}} (\lambda_{max}(x_e, y_e, z_e) - \lambda_{min}(x_e, y_e, z_e)) = H_{ij} \tag{B.9}$$

where

$$\left\{ \begin{array}{ll} \lambda_{min}(x_e, y_e, z_e) & = \max\left\{0, \lambda_x^{(min)}(x_e), \lambda_y^{(min)}(y_e), \lambda_z^{(min)}(z_e)\right\} \\ \lambda_{max}(x_e, y_e, z_e) & = \min\left\{L, \lambda_x^{(max)}(x_e), \lambda_y^{(max)}(y_e), \lambda_z^{(max)}(z_e)\right\} \end{array} \right. \tag{B.10}$$

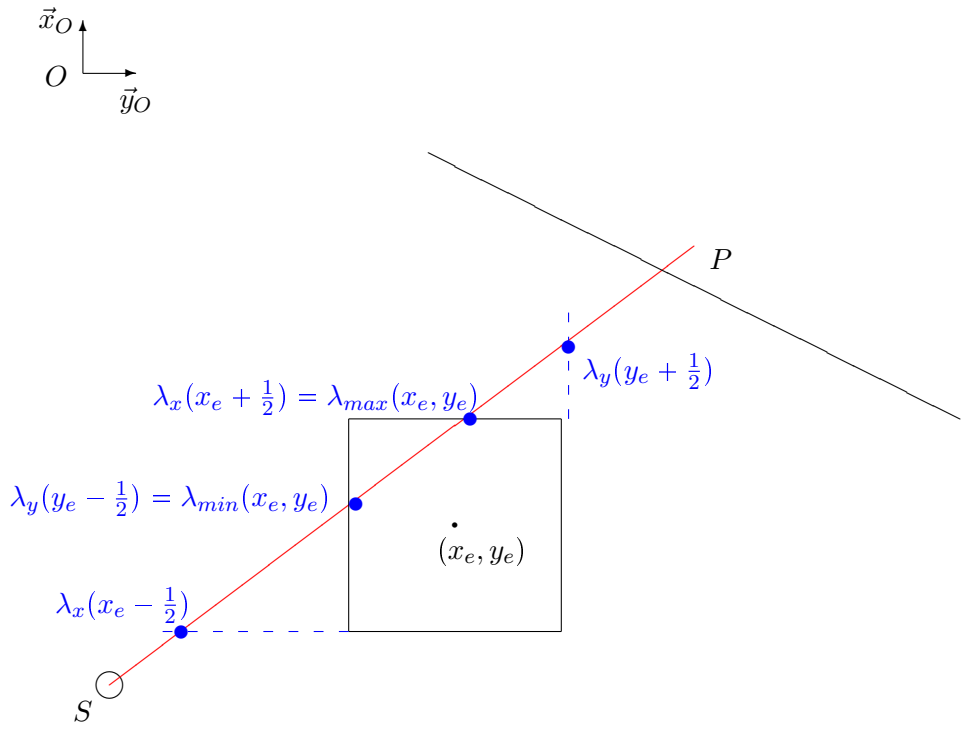


Figure B.2: Voxel (x_e, y_e) (in 2D) pierced by ray i, with the representation of lengths $\lambda_x(x_e - \frac{1}{2}), \lambda_x(x_e + \frac{1}{2}), \lambda_y(y_e - \frac{1}{2}), \lambda_y(y_e + \frac{1}{2})$

and

$$\begin{cases} \lambda_x^{(min)}(x_e) & = \min\left\{\lambda_x\left(x_e - \frac{1}{2}\right), \lambda_x\left(x_e + \frac{1}{2}\right)\right\} \\ \lambda_y^{(min)}(y_e) & = \min\left\{\lambda_y\left(y_e - \frac{1}{2}\right), \lambda_y\left(y_e + \frac{1}{2}\right)\right\} \\ \lambda_z^{(min)}(z_e) & = \min\left\{\lambda_z\left(z_e - \frac{1}{2}\right), \lambda_z\left(z_e + \frac{1}{2}\right)\right\} \\ \lambda_x^{(max)}(x_e) & = \max\left\{\lambda_x\left(x_e - \frac{1}{2}\right), \lambda_x\left(x_e + \frac{1}{2}\right)\right\} \\ \lambda_y^{(max)}(y_e) & = \max\left\{\lambda_y\left(y_e - \frac{1}{2}\right), \lambda_y\left(y_e + \frac{1}{2}\right)\right\} \\ \lambda_z^{(max)}(z_e) & = \max\left\{\lambda_z\left(z_e - \frac{1}{2}\right), \lambda_z\left(z_e + \frac{1}{2}\right)\right\} \end{cases} \tag{B.11}$$

In definition (B.8), one of the denominators can be zero. In this case, we define :

- if $x(u_e, \phi) = x_S(\phi)$ then $\lambda_x^{(min)}(x_e) = 0$ and $\lambda_x^{(max)}(x_e) = L$,
- if $y(u_e, \phi) = y_S(\phi)$ then $\lambda_y^{(min)}(y_e) = 0$ and $\lambda_y^{(max)}(y_e) = L$,
- if $z(v_e) = z_S(\phi)$ then $\lambda_z^{(min)}(z_e) = 0$ and $\lambda_z^{(max)}(z_e) = L$.

Siddon's method pre-computes length sequences λ_x, λ_y and λ_z and arrange it in the ascending order [Sid85]. Denoting 'x', 'y' and 'z' by 'ι', one can see the following relation

$$\lambda_\iota\left(\iota_e + \frac{1}{2}\right) - \lambda_\iota\left(\iota_e - \frac{1}{2}\right) = \frac{L\delta}{\iota(u_e, v_e, \phi) - \iota_S(\phi)}, \forall \iota, \tag{B.12}$$

where

$$\iota(u_e, v_e, \phi) = \begin{cases} x(u_e, \phi) & \text{if '}\iota\text{'='}x\text{'} \\ y(u_e, \phi) & \text{if '}\iota\text{'='}y\text{'} \\ z(v_e) & \text{if '}\iota\text{'='}z\text{'} \end{cases}. \tag{B.13}$$

From relation (B.12), it is clear that if $\iota_S(\phi) \leqslant \iota(u_e, v_e, \phi)$ then

$$\left\{ \begin{array}{ll} \lambda_\iota^{(min)}(\iota_e) & = \lambda_\iota\left(\iota_e - \frac{1}{2}\right) \\ \lambda_\iota^{(max)}(\iota_e) & = \lambda_\iota\left(\iota_e + \frac{1}{2}\right) \end{array} \right. , \tag{B.14}$$

otherwise :

$$\left\{ \begin{array}{ll} \lambda_\iota^{(min)}(\iota_e) & = \lambda_\iota\left(\iota_e + \frac{1}{2}\right) \\ \lambda_\iota^{(max)}(\iota_e) & = \lambda_\iota\left(\iota_e - \frac{1}{2}\right) \end{array} \right. . \tag{B.15}$$

For $\text{`}\iota\text{`} \in \{\text{`}x\text{`}, \text{`}y\text{`}, \text{`}z\text{`}\}$, we can use this fact to compute needed lengths $\lambda_\iota^{(min)}(\iota_e)$ and $\lambda_\iota^{(max)}(\iota_e)$ on-the-fly, as done in [JSDS$^+$98, HLY99], instead of pre-computing length sequences λ_x, λ_y and λ_z.

We denote by $N_{u_e,v_e,\phi}$ the (a priori unknown) number of voxels $(x_{e_n}, y_{e_n}, z_{e_n})$, $n = 1, \ldots, N_{u_e,v_e,\phi}$, pierced by the ray. For these voxels, we have the relation

$$\lambda_{min}(x_{e_n}, y_{e_n}, z_{e_n}) = \lambda_{max}(x_{e_{n-1}}, y_{e_{n-1}}, z_{e_{n-1}}), \forall n \in \{2, \ldots, N_{u_e,v_e,\phi}\} . \tag{B.16}$$

Hence, a dynamic ray-tracing method only needs to get $\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n})$ in order to compute the intersection length (B.9), since, as the algorithm considers each pierced voxel one after the other, $\lambda_{min}(x_{e_n}, y_{e_n}, z_{e_n})$ has been determined before according to relation (B.16) [JSDS$^+$98, HLY99]. In order to get $\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n})$, the algorithm needs to know $\lambda^{(max)}(x_{e_n})$, $\lambda^{(max)}(y_{e_n})$ and $\lambda^{(max)}(z_{e_n})$. Based on figure B.2, the computation of $\lambda^{(max)}(\iota_{e_n})$, for $\text{`}\iota\text{`} \in \{\text{`}x\text{`}, \text{`}y\text{`}, \text{`}z\text{`}\}$, can be simplified compared to definition (B.11) [JSDS$^+$98, HLY99] :

$$\lambda_\iota^{(max)}(\iota_{e_n}) = \max\left\{\lambda_\iota\left(\iota_{e_n} - \frac{1}{2}\right), \lambda_\iota\left(\iota_{e_n} + \frac{1}{2}\right)\right\} = \left\{ \begin{array}{l} \lambda_\iota\left(\iota_{e_n} + \frac{1}{2}\right) \text{ if } \iota_S(\phi) \leqslant \iota(u_e, v_e, \phi) \\ \\ \lambda_\iota\left(\iota_{e_n} - \frac{1}{2}\right) \text{ otherwise} \end{array} \right. . \tag{B.17}$$

The selection of the minimum $\lambda_\iota^{(max)}(\iota_{e_n})$ gives $\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n})$ according to definition (B.10). In addition, the selection of the minimum $\lambda_\iota^{(max)}(\iota_{e_n})$ gives the coordinates of the following voxel pierced by the ray :

$$\iota_{e_{n+1}} = \left\{ \begin{array}{l} \iota_{e_n} + 1 \text{ if } \lambda_\iota^{(max)}(\iota_{e_n}) = \lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) \text{ and } \iota_S(\phi) \leqslant \iota(u_e, v_e, \phi) \\ \iota_{e_n} - 1 \text{ if } \lambda_\iota^{(max)}(\iota_{e_n}) = \lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) \text{ and } \iota_S(\phi) > \iota(u_e, v_e, \phi) \\ \iota_{e_n} \text{ otherwise} \end{array} \right. , \tag{B.18}$$

and iterate $\lambda_\iota^{(max)}(\iota_{e_n})$ for the corresponding moving direction ι can be incremented according to relation (B.12) :

$$\lambda_\iota^{(max)}(\iota_{e_{n+1}}) = \left\{ \begin{array}{l} \lambda_\iota^{(max)}(\iota_{e_n}) + \left|\frac{L\delta}{\iota(u_e,v_e,\phi) - \iota_S(\phi)}\right| \text{ if } \lambda_\iota^{(max)}(\iota_{e_n}) = \lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) \\ \\ \lambda_\iota^{(max)}(\iota_{e_n}) \text{ otherwise} \end{array} \right. . \tag{B.19}$$

Siddon-Jacobs-Han's projector based on this dynamic ray-tracing is summarized in algorithm 16. At its initialization, this algorithm needs

- to know if the ray has crossed the field-of-view,
- to compute the coordinates of the first pierced voxel $(x_{e_1}, y_{e_1}, z_{e_1})$,
- to compute $\lambda_{min}(x_{e_1}, y_{e_1}, z_{e_1})$ in order to get the intersection length for $(x_{e_1}, y_{e_1}, z_{e_1})$.

The field-of-view is the cylinder of axis $[L_r z)$ and of radius $\frac{\delta N_x}{2}$. The ray crosses the field-of-view if there exists λ such that

$$(x(\lambda) - x_{L_r})^2 + (y(\lambda) - y_{L_r})^2 = \left(\frac{\delta N_x}{2}\right)^2 \tag{B.20}$$

where $x(\lambda)$ and $y(\lambda)$ are given by (B.7). Equation (B.20) reads

$$A_\lambda \lambda^2 + 2B_\lambda \lambda + C_\lambda = 0 \tag{B.21}$$

Algorithm 16 Siddon-Jacobs-Han's projector [Sid85, JSDS+98, HLY99]

for ϕ, projection angle, **do**
  for $u_e, v_e \in \{0, \dots, N_u - 1\} \times \{0, \dots, N_v - 1\}$ **do**
    Compute Δ_λ by (B.23) to know whether the ray has crosses the field-of-view or not
    Initialize the projection of the ray : $g(u_e, v_e, \phi) = 0$
    if $\Delta_\lambda \geqslant 0$ **then**
      Compute $\lambda_f^{(min)}$ and $\lambda_f^{(max)}$ which are the ordered solutions of (B.21)
      Compute the first voxel $(x_{e_1}, y_{e_1}, z_{e_1})$ pierced by the ray, using (B.26)
      Set : $\lambda_{min}(x_{e_1}, y_{e_1}, z_{e_1}) = \lambda_f^{(min)}$
      Initialize : $\iota_{e_n} := \iota_{e_1}, \forall \iota \in \{`x`, `y`, `z`\}$
      Compute : $\lambda_\iota^{(max)}(\iota_{e_1})$ by (B.17), $\forall \iota \in \{`x`, `y`, `z`\}$
      Initialize : $\lambda_\iota^{(max)}(\iota_{e_n}) := \lambda_\iota^{(max)}(\iota_{e_1})$, $\forall \iota \in \{`x`, `y`, `z`\}$
      Initialize : $\lambda^{(max)}(x_{e_n}, y_{e_n}, z_{e_n}) := 0$ and $\lambda_{max}(x_{e_{n-1}}, y_{e_{n-1}}, z_{e_{n-1}}) := \lambda_{min}(x_{e_1}, y_{e_1}, z_{e_1})$
      while $\lambda^{(max)}(x_{e_n}, y_{e_n}, z_{e_n}) \leqslant \lambda_f^{(max)}$ **do**
        Compute : $\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) = \min\left\{L, \lambda_x^{(max)}(x_{e_n}), \lambda_y^{(max)}(y_{e_n}), \lambda_z^{(max)}(z_{e_n})\right\}$
        if $\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) > \lambda_{max}(x_{e_{n-1}}, y_{e_{n-1}}, z_{e_{n-1}})$ **then**
          Compute the intersection length : $l = \lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) - \lambda_{max}(x_{e_{n-1}}, y_{e_{n-1}}, z_{e_{n-1}})$
        else
          $l = 0$
        end if
        Update the projection : $g(u_e, v_e, \phi) += l \times f(x_{e_n}, y_{e_n}, z_{e_n})$
        Compute the coordinates $(x_{e_{n+1}}, y_{e_{n+1}}, z_{e_{n+1}})$ of the next voxel by (B.18)
        Set : $\iota_{e_n} := \iota_{e_{n+1}}, \forall \iota \in \{`x`, `y`, `z`\}$
        Set : $\lambda_{max}(x_{e_{n-1}}, y_{e_{n-1}}, z_{e_{n-1}}) := \lambda^{(max)}(x_{e_n}, y_{e_n}, z_{e_n})$
        Compute $\lambda_\iota^{(max)}(\iota_{e_{n+1}})$ by (B.19), $\forall \iota \in \{`x`, `y`, `z`\}$
        Set : $\lambda_\iota^{(max)}(\iota_{e_n}) := \lambda_\iota^{(max)}(\iota_{e_{n+1}})$, $\forall \iota \in \{`x`, `y`, `z`\}$
      end while
    end if
  end for
end for

where

$$\begin{cases} A_\lambda &= \frac{(x_P - x_S)^2 + (y_P - y_S)^2}{L^2} = \frac{D^2 + s^2}{L^2} \\ B_\lambda &= \frac{(x_S - x_{L_r})(x_P - x_S) + (y_S - y_{L_r})(y_P - y_S)}{L} = -\frac{RD}{L} \\ C_\lambda &= (x_S - x_{L_r})^2 + (y_S - y_{L_r})^2 - \left(\frac{\delta N_x}{2}\right)^2 = R^2 - \left(\frac{\delta N_x}{2}\right)^2 \end{cases} . \tag{B.22}$$

Hence, the ray crossed the field-of-view if

$$\Delta_\lambda = B_\lambda^2 - A_\lambda C_\lambda \geqslant 0. \tag{B.23}$$

The ray enters in the field-of-view at

$$\iota_{min} = \iota_S(\phi) + \frac{\lambda_f^{(min)}}{L}(\iota(u_e, v_e, \phi) - \iota_S(\phi)) \tag{B.24}$$

and goes out the field-of-view at

$$\iota_{max} = \iota_S(\phi) + \frac{\lambda_f^{(max)}}{L}(\iota(u_e, v_e, \phi) - \iota_S(\phi)), \forall \iota \in \{`x`, `y`, `z`\}. \tag{B.25}$$

These coordinates depend on $\lambda_f^{(min)}$ and $\lambda_f^{(max)}$ which are the ordered solutions of (B.21) defined if $\Delta_\lambda \geqslant 0$. The coordinates of the first voxel pierced by the ray are

$$\iota_{e_1} = \begin{cases} \left\lfloor \frac{\iota_{min}}{\delta} - 0.5 \right\rfloor \text{ if } \iota_S(\phi) \leqslant \iota(u_e, v_e\phi) \\ \left\lceil \frac{\iota_{min}}{\delta} + 0.5 \right\rceil \text{ otherwise} \end{cases}, \tag{B.26}$$

and

$$\lambda_{min}(x_{e_1}, y_{e_1}, z_{e_1}) = \lambda_f^{(min)}. \tag{B.27}$$

Moreover, algorithm 16 is stopped when $\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n})$ has reached its maximum value, i.e. when

$$\lambda_{max}(x_{e_n}, y_{e_n}, z_{e_n}) > \lambda_f^{(max)}. \tag{B.28}$$

B.2 Separable Footprint (SF) projector and backprojector

B.2.1 Description of the SF pair

The Separable Footprint (SF) pair of projector and backprojector (P/BP) approximates the footprint of a voxel onto the detector as a separable function in the transaxial and axial directions. Among possible SF approximations [LFB10, Lon11], the SFTR pair (Separable Footprint Trapezoidal-Rectangular) models the footprint as trapezoidal in the transaxial direction, and rectangular in the axial direction [LFB10]. In this section, we detail the SFTR P/BP pair introduced in [LFB10]. We adapt the calculations to our geometrical model presented in section 2.1. Based on [LFB10], we also give the pseudo-codes of our CPU implementation of the SFTR projector and backprojector. In the following, "SFTR pair" is abbreviated by "SF pair", since we do not mention the other SF models.

For a voxel (x_e, y_e, z_e), its trapezoidal transaxial footprint is defined as [LFB10]

$$e_{trans}(s_e, \phi; x_e, y_e) = \begin{cases} \frac{s_e - \tau_0}{\tau_1 - \tau_0} \text{ if } \tau_0 \leqslant s_e < \tau_1 \\ 1 \text{ if } \tau_1 \leqslant s_e < \tau_2 \\ \frac{\tau_3 - s_e}{\tau_3 - \tau_2} \text{ if } \tau_2 \leqslant s_e < \tau_3 \\ 0 \text{ otherwise} \end{cases} \tag{B.29}$$

where $\tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$ are the ordered projections of the fours corners $(x_e \pm 0.5, y_e \pm 0.5, 0)$ of pixel (x_e, y_e) in the median plane :

$$\begin{cases} p_{-,-}(x_e, y_e) &= \frac{D}{\delta_u} \frac{y_{\phi_e} + 0.5 \times (\sin\phi - \cos\phi)}{x_{\phi_e} + 0.5 \times (-\cos\phi - \sin\phi)} \\ p_{+,-}(x_e, y_e) &= \frac{D}{\delta_u} \frac{y_{\phi_e} + 0.5 \times (-\sin\phi - \cos\phi)}{x_{\phi_e} + 0.5 \times (\cos\phi - \sin\phi)} \\ p_{-,+}(x_e, y_e) &= \frac{D}{\delta_u} \frac{y_{\phi_e} + 0.5 \times (\sin\phi + \cos\phi)}{x_{\phi_e} + 0.5 \times (-\cos\phi + \sin\phi)} \\ p_{+,+}(x_e, y_e) &= \frac{D}{\delta_u} \frac{y_{\phi_e} + 0.5 \times (-\sin\phi + \cos\phi)}{x_{\phi_e} + 0.5 \times (\cos\phi + \sin\phi)} \end{cases} \tag{B.30}$$

where

$$\begin{cases} x_{\phi_e} &= x'_e \cos\phi + y'_e \sin\phi + \frac{R}{\delta} \\ y_{\phi_e} &= -x'_e \sin\phi + y'_e \cos\phi \\ z_{\phi_e} &= z'_e \end{cases} \tag{B.31}$$

are the coordinates of the voxel in the rotating frame $(S, \vec{x}_{L_T}, \vec{y}_{L_T}, \vec{z}_{L_r})$. These projections are illustrated in figure B.3 at different projection angles. The rectangular axial footprint of voxel (x_e, y_e, z_e) is [LFB10]

$$e_{ax}(t_e, \phi; x_e, y_e, z_e) = \begin{cases} 1 \text{ if } \chi_0 \leqslant t_e \leqslant \chi_1 \\ 0 \text{ otherwise} \end{cases} \tag{B.32}$$

where $\chi_0 \leqslant \chi_1$ are the projections of $(x_e, y_e, z_e \pm 0.5)$, as illustrated in figure B.4 :

$$\begin{cases} \chi_0 = \frac{D}{\delta_v} \frac{z_{\phi_e} - 0.5}{x_{\phi_e}} \\ \chi_1 = \frac{D}{\delta_v} \frac{z_{\phi_e} + 0.5}{x_{\phi_e}} \end{cases} . \tag{B.33}$$

In order to be more accurate, a trapezoidal shape has also been studied for the axial footprint : nevertheless, it implies too many calculations for a not significant gain in precision [LFB10].

On the contrary to Siddon-Jacobs-Han's projector, the SF pair takes into account the finite size of the cells. The projection of voxel (x_e, y_e, z_e) measured by cell (u_e, v_e) reads

$$\begin{aligned} &F(u_e, v_e, \phi; x_e, y_e, z_e) = \\ &\qquad l(u_e, v_e, \phi; x_e, y_e, z_e) \int_{s_e} \int_{t_e} h(s_{u_e} - s_e, t_{v_e} - t_e) e_{trans}(s_e, \phi; x_e, y_e) e_{ax}(t_e, \phi; x_e, y_e, z_e) \, \mathrm{d}s_e \, \mathrm{d}t_e \end{aligned} \tag{B.34}$$

where $h(s_e, t_e)$ is the (shift-invariant) detector blur :

$$h(s_e, t_e) = \mathbf{1}_{\left\{-\frac{1}{2} \leqslant s_e \leqslant \frac{1}{2}\right\}} \times \mathbf{1}_{\left\{-\frac{1}{2} \leqslant t_e \leqslant \frac{1}{2}\right\}} \tag{B.35}$$

and $l(u_e, v_e, \phi; x_e, y_e, z_e)$ is an amplitude function [LFB10]. Thanks to the separability in s_e and t_e of the detector blur and of the approximating footprint

$$e(s_e, t_e, \phi; x_e, y_e, z_e) \approx e_{trans}(s_e, \phi; x_e, y_e) e_{ax}(t_e, \phi; x_e, y_e, z_e), \tag{B.36}$$

the measured projection of voxel (x_e, y_e, z_e) is :

$$F(u_e, v_e, \phi; x_e, y_e, z_e) = l(u_e, v_e, \phi; x_e, y_e, z_e) F_{trans}(u_e, \phi; x_e, y_e) F_{ax}(v_e, \phi; x_e, y_e, z_e). \tag{B.37}$$

The blurred transaxial footprint $F_{trans}(u_e, \phi; x_e, y_e)$ reads [LFB10]

$$\begin{aligned} F_{trans}(u_e, \phi; x_e, y_e) &= \int_{s_{u_e} - 0.5}^{s_{u_e} + 0.5} e_{trans}(s_e, \phi; x_e, y_e) \, \mathrm{d}s_e \\ &= F_{trans}^{(01)}(\max\{s_{u_e} - 0.5, \tau_0\}, \min\{s_{u_e} + 0.5, \tau_1\}) \\ &+ F_{trans}^{(12)}(\max\{s_{u_e} - 0.5, \tau_1\}, \min\{s_{u_e} + 0.5, \tau_2\}) \\ &+ F_{trans}^{(23)}(\max\{s_{u_e} - 0.5, \tau_2\}, \min\{s_{u_e} + 0.5, \tau_3\}) \end{aligned} \tag{B.38}$$

where

$$F_{trans}^{(01)}(s_{e_1}, s_{e_2}) = \frac{1}{2} \frac{(s_{e_2} - \tau_0)^2 - (s_{e_1} - \tau_0)^2}{\tau_1 - \tau_0} \times \mathbf{1}_{\{s_{e_1} < s_{e_2}\}}, \tag{B.39}$$

$$F_{trans}^{(12)}(s_{e_1}, s_{e_2}) = (s_{e_2} - s_{e_1}) \mathbf{1}_{\{s_{e_1} < s_{e_2}\}}, \tag{B.40}$$

$$F_{trans}^{(23)}(s_{e_1}, s_{e_2}) = \frac{1}{2} \frac{(\tau_3 - s_{e_1})^2 - (\tau_3 - s_{e_2})^2}{\tau_3 - \tau_2} \times \mathbf{1}_{\{s_{e_1} < s_{e_2}\}}. \tag{B.41}$$

Similarly, the blurred axial footprint $F_{ax}(v_e, \phi; x_e, y_e, z_e)$ reads [LFB10]

$$\begin{aligned} F_{ax}(v_e, \phi; x_e, y_e, z_e) &= \int_{t_{v_e} - 0.5}^{t_{v_e} + 0.5} e_{ax}(t_e, \phi; x_e, y_e, z_e) \, \mathrm{d}t_e \\ &= F_{ax}^{(01)}(\max\{t_{v_e} - 0.5, \chi_0\}, \min\{t_{v_e} + 0.5, \chi_1\}) \end{aligned} \tag{B.42}$$

where

$$F_{ax}^{(01)}(t_{e_1}, t_{e_2}) = (t_{e_2} - t_{e_1}) \mathbf{1}_{\{t_{e_1} < t_{e_2}\}}. \tag{B.43}$$

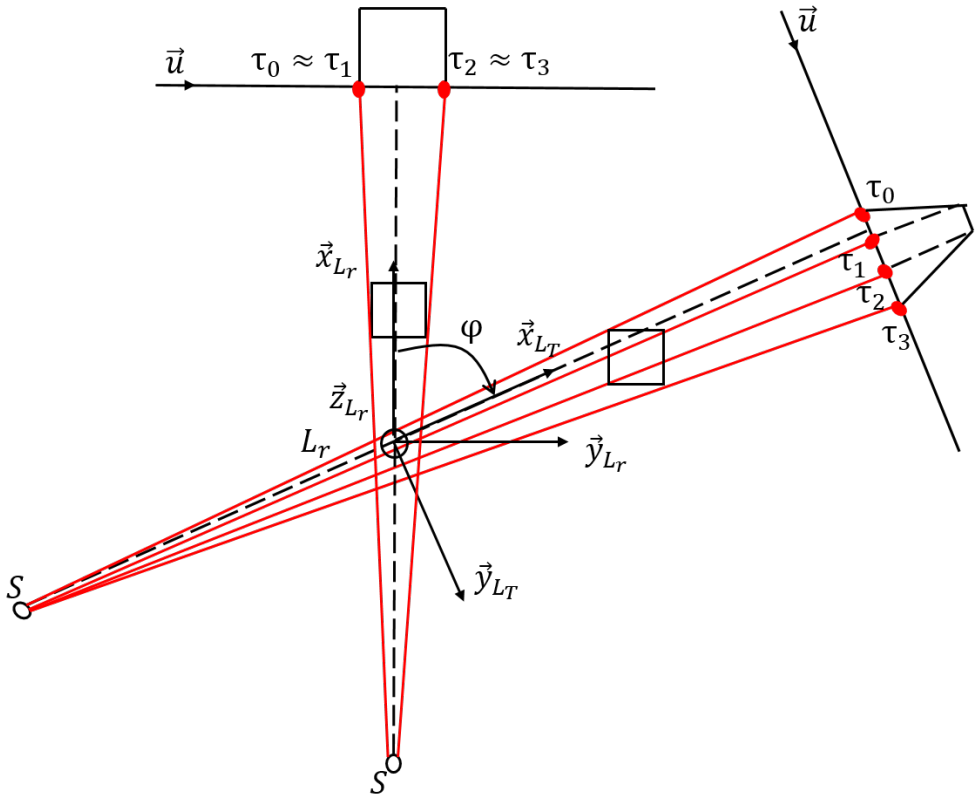


Figure B.3: Trapezoidal transaxial footprint of a voxel at different projection angles

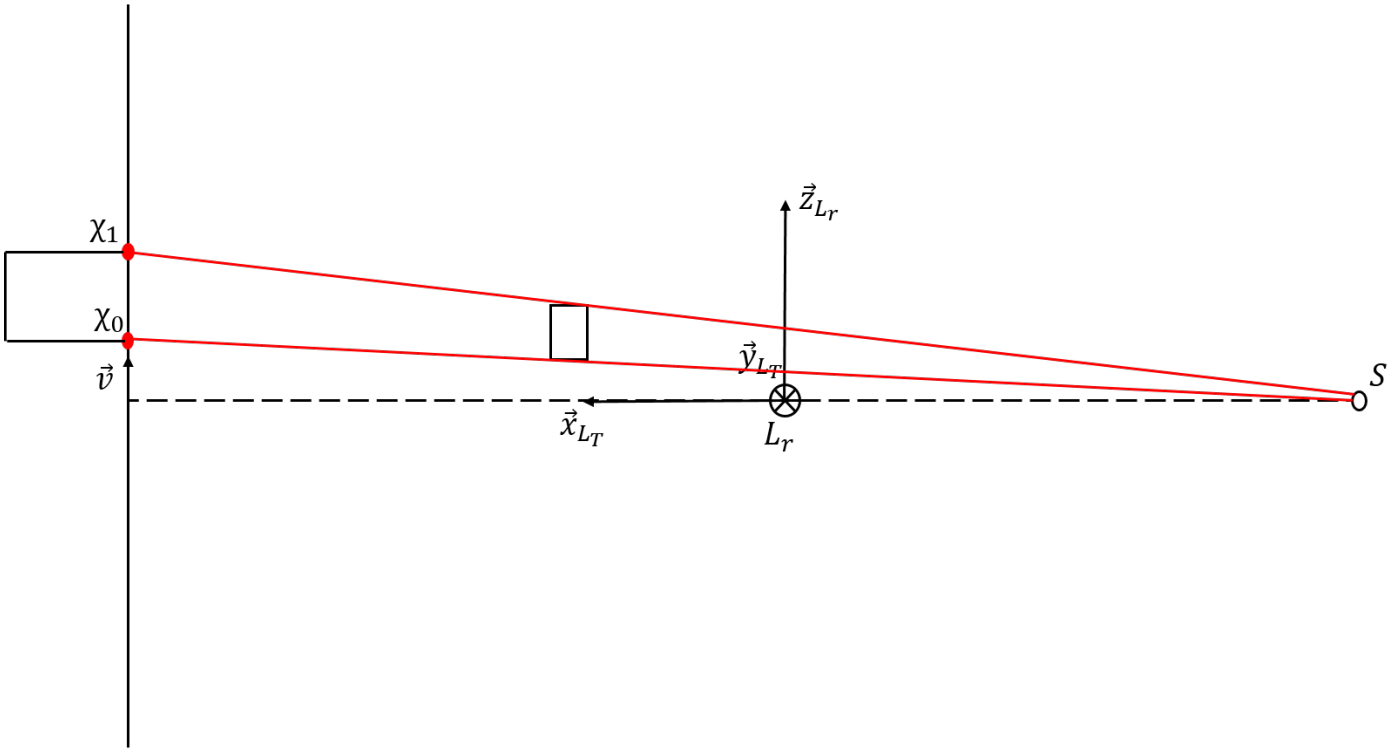


Figure B.4: Rectangular axial footprint of a voxel

The blurred transaxial and axial footprints (B.38) and (B.42) are of amplitude 1. Hence, they need to be multiplied by amplitude function $l(u_e, v_e, \phi; x_e, y_e, z_e)$ in order to get a correct value for ideal measured projection $F(u_e, v_e, \phi; x_e, y_e, z_e)$ in (B.37). This amplitude function is given by the geometry of the acquisition [LFB10]. Several choices are studied in [LFB10], mostly inspired by Joseph's method [Jos82]. The A1 method considers the amplitude function [LFB10]

$$l(u_e, v_e, \phi; x_e, y_e, z_e) = l_1(u_e, v_e, \phi) = l_{\psi_c}(u_e, \phi) l_{\theta_c}(u_e, v_e) \tag{B.44}$$

where

$$l_{\psi_c}(u_e, \phi) = \frac{\delta}{\max\{|\cos\psi_c(u_e, \phi)|, |\sin\psi_c(u_e, \phi)|\}} \tag{B.45}$$

and

$$l_{\theta_c}(u_e, v_e) = \frac{1}{|\cos\theta_c(u_e, v_e)|}. \tag{B.46}$$

In (B.45) and (B.46), $\psi_c(u_e, \phi)$ and $\theta_c(u_e, v_e)$ are the azimuthal and the polar angles of the ray connecting the source to the center of the cell [LFB10]. These angles are illustrated in figures B.5 and B.6 respectively. Based on these figures, we have

$$\psi_c(u_e, \phi) = \phi + \arctan\left(\frac{s_{u_e}\delta_u}{D}\right) \tag{B.47}$$

and

$$\theta_c(u_e, v_e) = \arctan\left(\frac{t_{v_e}\delta_v}{\sqrt{D^2 + (s_{u_e}\delta_u)^2}}\right). \tag{B.48}$$

Hence, amplitudes (B.45) and (B.46) read

$$l_{\psi_c}(u_e, \phi) = \frac{\delta\sqrt{D^2 + (s_{u_e}\delta_u)^2}}{\max\{|D\cos\phi - s_{u_e}\delta_u\sin\phi|, |s_{u_e}\delta_u\cos\phi + D\sin\phi|\}} \tag{B.49}$$

and

$$l_{\theta_c}(u_e, v_e) = \sqrt{1 + \frac{(t_{v_e}\delta_v)^2}{D^2 + (s_{u_e}\delta_u)^2}}. \tag{B.50}$$

While the A1 method is ray-driven, the A3 method is voxel-driven since it considers the amplitude [LFB10]

$$l(u_e, v_e, \phi; x_e, y_e, z_e) = l_3(\phi; x_e, y_e, z_e) = l_{\psi_v}(\phi; x_e, y_e) l_{\theta_v}(\phi; x_e, y_e, z_e) \tag{B.51}$$

where

$$l_{\psi_v}(\phi; x_e, y_e) = \frac{\delta}{\max\{|\cos\psi_v(\phi; x_e, y_e)|, |\sin\psi_v(\phi; x_e, y_e)|\}} \tag{B.52}$$

and

$$l_{\theta_v}(\phi; x_e, y_e, z_e) = \frac{1}{|\cos\theta_v(\phi; x_e, y_e, z_e)|}. \tag{B.53}$$

$\psi_v(\phi; x_e, y_e)$ and $\theta_v(\phi; x_e, y_e, z_e)$ are the azimuthal and polar angles of the ray connecting the source to the center of the voxel (x_e, y_e, z_e). These angles are also illustrated in figures B.5 and B.6 :

$$\psi_v(\phi; x_e, y_e) = \phi + \arctan\left(\frac{y_{\phi_e}}{x_{\phi_e}}\right) \tag{B.54}$$

and

$$\theta_v(\phi; x_e, y_e, z_e) = \arctan\left(\frac{z_{\phi_e}}{\sqrt{x_{\phi_e}^2 + y_{\phi_e}^2}}\right). \tag{B.55}$$

Hence,

$$l_{\psi_v}(\phi; x_e, y_e) = \frac{\delta\sqrt{x_{\phi_e}^2 + y_{\phi_e}^2}}{\max\left\{|x_{\phi_e}\cos\phi - y_{\phi_e}\sin\phi|, |x_{\phi_e}\sin\phi + y_{\phi_e}\cos\phi|\right\}} \tag{B.56}$$

and

$$l_{\theta_v}(\phi; x_e, y_e, z_e) = \sqrt{1 + \frac{z_{\phi_e}^2}{x_{\phi_e}^2 + y_{\phi_e}^2}}. \tag{B.57}$$

At last, the A2 method is both ray-driven and voxel-driven, combining parts of the A1 and A3 methods [LFB10]

$$l(u_e, v_e, \phi; x_e, y_e, z_e) = l_2(u_e, v_e, \phi; x_e, y_e) = l_{\psi_v}(\phi; x_e, y_e) l_{\theta_c}(u_e, v_e), \tag{B.58}$$

where $l_{\psi_v}(\phi; x_e, y_e)$ and $l_{\theta_c}(u_e, v_e)$ are respectively given by (B.56) and (B.50).

It has been shown that the A1, A2 and A3 methods give the same results in terms of precision [LFB10]. Nevertheless, while the A1 and the A2 methods have the same computation times, the A3 method implies much more calculations [LFB10]. Since all the three methods give the same results, in this work, we choose to apply the A2 method for the amplitude function, since it is a tradeoff between the A1 and the A3 methods. This choice has also been made in the GPU implementations of the SF pair [WF11, XML+17]. With this method, the ideally measured projection (B.37) reads

$$F(u_e, v_e, \phi; x_e, y_e, z_e) = l_{\psi_v}(\phi; x_e, y_e) l_{\theta_c}(u_e, v_e) F_{trans}(u_e, \phi; x_e, y_e) F_{ax}(v_e, \phi; x_e, y_e, z_e). \tag{B.59}$$

As a result, the expressions for the SF projector and backprojector are [LFB10]

$$\begin{aligned} g_i = g(u_e, v_e, \phi) = (\boldsymbol{H}\boldsymbol{f})_i &= \sum_{x_e}\sum_{y_e}\sum_{z_e} F(u_e, v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e) \\ &= l_{\theta_c}(u_e, v_e) \sum_{x_e}\sum_{y_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e) \end{aligned} \tag{B.60}$$

and

$$\begin{aligned} b_j = b(x_e, y_e, z_e) = (\boldsymbol{H}^T\boldsymbol{g})_j &= \sum_{\phi}\sum_{u_e}\sum_{v_e} F(u_e, v_e, \phi; x_e, y_e, z_e) g(u_e, v_e, \phi) \\ &= \sum_{\phi}\sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) \sum_{u_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi). \end{aligned} \tag{B.61}$$

B.2.2 CPU implementation of the SF projector

Thanks to the factorizations induced by the separable footprint approximation, the SF projection of a volume $\boldsymbol{f}$

$$g(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) \sum_{x_e}\sum_{y_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e) \tag{B.62}$$

can be computed very fast on the CPU. The set of relevant voxels contributing to the projection $g(u_e, v_e, \phi)$ are those for which the transaxial and the axial footprints overlap with cell (u_e, v_e) at projection angle ϕ. Thanks to the separation in u_e and v_e, this set is easy to compute.

The CPU implementation of the SF projector has its main loop over projection angles ϕ [LFB10]. Next, for each pixel (x_e, y_e) in the middle plane, its four corners $(x_e \pm 0.5, y_e \pm 0.5)$ are projected and their projections ordered : $\tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$. Knowing τ_0 and τ_3 gives the cells overlapping with the transaxial footprint, i.e. the cells for which $F_{trans}(u_e, \phi; x_e, y_e) \neq 0$. As illustrated in figure B.7, these cells are such that their left side is before τ_3

$$s_{u_e} - 0.5 = u_e - u_{C_{d_e}} - 0.5 \leqslant \tau_3 \Leftrightarrow u_e \leqslant u_{C_{d_e}} + 0.5 + \tau_3 \tag{B.63}$$

and their right side after τ_0

$$s_{u_e} + 0.5 = u_e - u_{C_{d_e}} + 0.5 \geqslant \tau_0 \Leftrightarrow u_e \geqslant u_{C_{d_e}} - 0.5 + \tau_0. \tag{B.64}$$

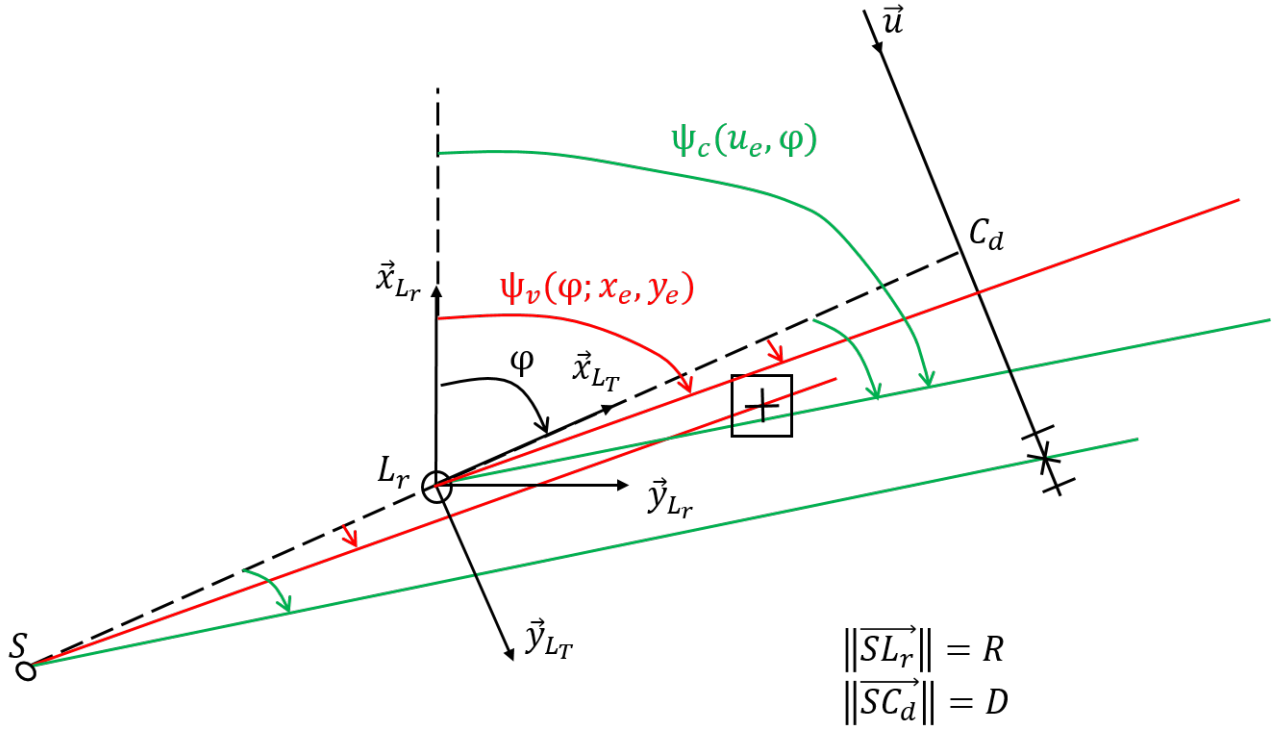


Figure B.5: Azimuthal angles of the ray connecting the source to the center of a cell (u_e, v_e) and of the ray connecting the source to the center of a voxel (x_e, y_e, z_e)

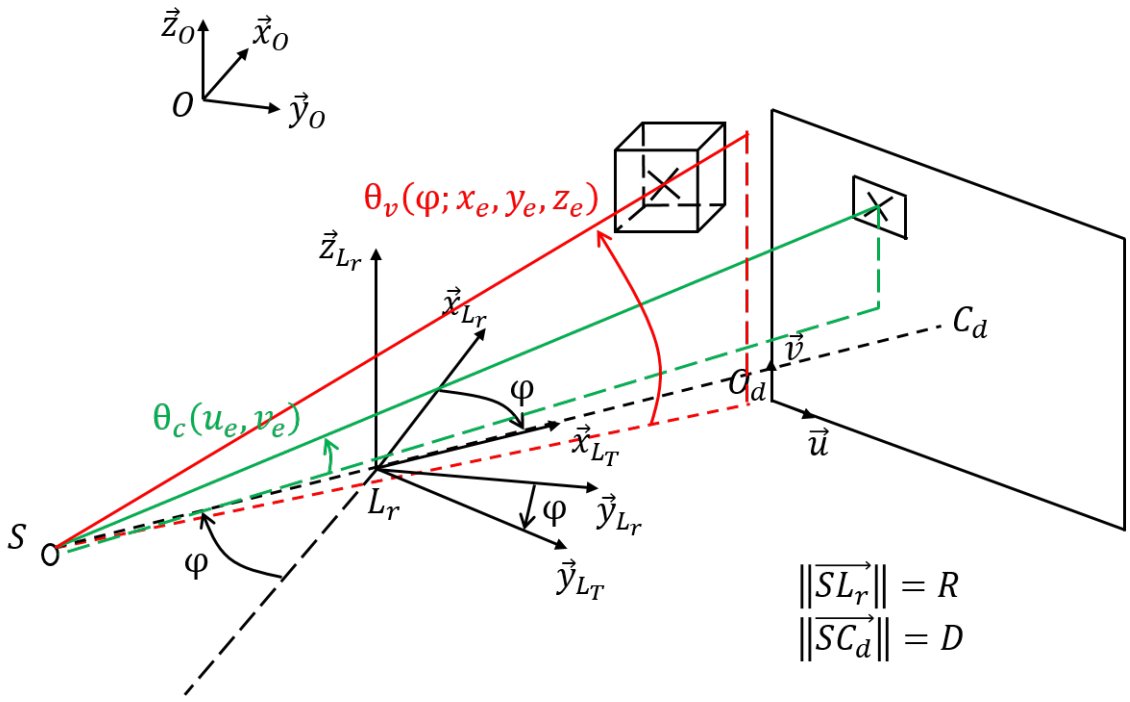


Figure B.6: Polar angles of the ray connecting the source to the center of a cell (u_e, v_e) and of the ray connecting the source to the center of a voxel (x_e, y_e, z_e)

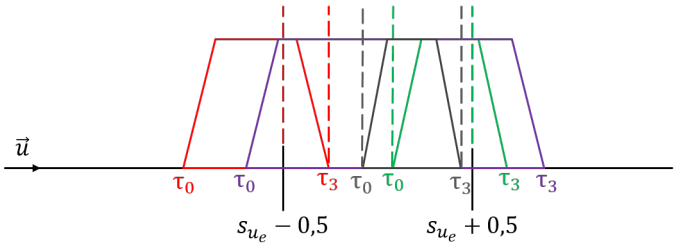


Figure B.7: Different overlaps of the transaxial footprint of a voxel with a cell

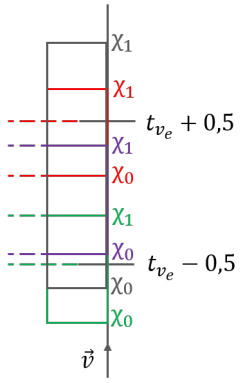


Figure B.8: Different overlaps of the axial footprint of a voxel with a cell

Therefore,

$$F_{trans}(u_e, \phi; x_e, y_e) \neq 0 \Leftrightarrow u_{e_{min}} \leqslant u_e \leqslant u_{e_{max}} \tag{B.65}$$

where

$$\begin{cases} u_{e_{min}} &= \lceil \max\{0, u_{C_{d_e}} - 0.5 + \tau_0\} \rceil \\ u_{e_{max}} &= \lceil \min\{N_u - 1, u_{C_{d_e}} + 0.5 + \tau_3\} \rceil \end{cases}. \tag{B.66}$$

The blurred and scaled transaxial footprint

$$F'_{trans}(u_e, \phi; x_e, y_e) = l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e), \forall u_e \in \{u_{e_{min}}, \dots, u_{e_{max}}\} \tag{B.67}$$

is computed and stored for each u_e between $u_{e_{min}}$ and $u_{e_{max}}$.

Next, for each v_e, indices z_e for which $F_{ax}(v_e, \phi; x_e, y_e, z_e) \neq 0$ are computed. As illustrated in figure B.8, these z_e are such that the bottom of the axial footprint of voxel (x_e, y_e, z_e) is under the top of the cell

$$\begin{aligned} & \chi_0 \leqslant t_{v_e} + 0.5 \\ \Leftrightarrow & \frac{D}{\delta_v} \frac{z_{\phi_e} - 0.5}{x_{\phi_e}} \leqslant t_{v_e} + 0.5 \\ \Leftrightarrow & z_{\phi_e} - 0.5 \leqslant \frac{x_{\phi_e}}{D}(t_{v_e} + 0.5)\delta_v \\ \Leftrightarrow & z_e \leqslant z_{L_{r_e}} + 0.5 + \frac{x_{\phi_e}}{D}(t_{v_e} + 0.5)\delta_v \end{aligned} \tag{B.68}$$

and the top of the footprint is above the bottom of the cell

$$\begin{aligned} & \chi_1 \geqslant t_{v_e} - 0.5 \\ \Leftrightarrow & \frac{D}{\delta_v} \frac{z_{\phi_e} + 0.5}{x_{\phi_e}} \geqslant t_{v_e} - 0.5 \\ \Leftrightarrow & z_{\phi_e} + 0.5 \geqslant \frac{x_{\phi_e}}{D}(t_{v_e} - 0.5)\delta_v \\ \Leftrightarrow & z_e \geqslant z_{L_{r_e}} - 0.5 + \frac{x_{\phi_e}}{D}(t_{v_e} - 0.5)\delta_v. \end{aligned} \tag{B.69}$$

Hence, for each v_e,

$$F_{ax}(v_e, \phi; x_e, y_e, z_e) \neq 0 \Leftrightarrow z_{e_{min}} \leqslant z_e \leqslant z_{e_{max}} \tag{B.70}$$

where

$$\begin{cases} z_{e_{min}} &= \lfloor \max\left\{0, z_{L_{r_e}} - 0.5 + \frac{x_{\phi_e}}{D}(t_{v_e} - 0.5)\delta_v\right\} \rfloor \\ z_{e_{max}} &= \lceil \min\left\{N_z - 1, z_{L_{r_e}} + 0.5 + \frac{x_{\phi_e}}{D}(t_{v_e} + 0.5)\delta_v\right\} \rceil \end{cases} . \tag{B.71}$$

Knowing $z_{e_{min}}$ and $z_{e_{max}}$, a small loop over z_e is run to compute the sum of the blurred axial footprints

$$F'_{ax}(v_e, \phi; x_e, y_e) = \sum_{z_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e) = \sum_{z_e = z_{e_{min}}}^{z_{e_{max}}} F_{ax}(v_e, \phi; x_e, y_e, z_e) f(x_e, y_e, z_e). \tag{B.72}$$

After this loop, the CPU accumulates the products of $F'_{ax}(v_e, \phi; x_e, y_e)$ with $F'_{trans}(u_e, \phi; x_e, y_e)$, for $u_{e_{min}} \leqslant u_e \leqslant u_{e_{max}}$:

$$g'(u_e, v_e, \phi) += F'_{trans}(u_e, \phi; x_e, y_e) F'_{ax}(v_e, \phi; x_e, y_e). \tag{B.73}$$

where

$$g'(u_e, v_e, \phi) = \sum_{x_e} \sum_{y_e} F'_{trans}(u_e, \phi; x_e, y_e) F'_{ax}(v_e, \phi; x_e, y_e). \tag{B.74}$$

After having considered all pixels (x_e, y_e), a final double loop is run over u_e and v_e in order to scale the projections according to the A2 method :

$$g(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) g'(u_e, v_e, \phi). \tag{B.75}$$

This double loop ends the computation of the SF projection on the CPU. The overall algorithm is summarized in algorithm 17.

B.2.3 CPU implementation of the SF backprojector

The SF backprojection of projections $\boldsymbol{g}$ is the adjoint operation of (B.62)

$$b(x_e, y_e, z_e) = \sum_{\phi} \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) \sum_{u_e} l_{\psi_v}(\phi; x_e, y_e) F_{trans}(u_e, \phi; x_e, y_e) l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi). \tag{B.76}$$

As for the projector, the SF backprojection has its main loop over projection angles ϕ [LFB10]. First, the projections are multiplied by the polar amplitude function of the A2 method :

$$\bar{g}(u_e, v_e, \phi) = l_{\theta_c}(u_e, v_e) g(u_e, v_e, \phi). \tag{B.77}$$

Similarly to the SF projector, the CPU implementation of the SF backprojector runs a double loop over pixels (x_e, y_e) in the middle plane, and, for each u_e between $u_{e_{min}}$ and $u_{e_{max}}$ defined by (B.66), the blurred and scaled transaxial footprint (B.67) is computed and stored. Next, for each v_e, the CPU computes and stores :

$$\bar{b}_{trans}(v_e, \phi; x_e, y_e) = \sum_{u_e} F'_{trans}(u_e, \phi; x_e, y_e) \bar{g}(u_e, v_e, \phi) = \sum_{u_e = u_{e_{min}}}^{u_{e_{max}}} F'_{trans}(u_e, \phi; x_e, y_e) \bar{g}(u_e, v_e, \phi). \tag{B.78}$$

Then, a loop over z_e is run. For each z_e, we determine indices v_e for which the blurred axial footprint is non-zero. Based on figure B.8, these indices are such that :

$$t_{v_e} + 0.5 \geqslant \chi_0 \Leftrightarrow v_e \geqslant v_{C_{d_e}} - 0.5 + \chi_0 \tag{B.79}$$

and

$$t_{v_e} - 0.5 \geqslant \chi_1 \Leftrightarrow v_e \leqslant v_{C_{d_e}} + 0.5 + \chi_1. \tag{B.80}$$

Hence, for each z_e,

$$F_{ax}(v_e, \phi; x_e, y_e, z_e) \neq 0 \Leftrightarrow v_{e_{min}} \leqslant v_e \leqslant v_{e_{max}}, \tag{B.81}$$

Algorithm 17 SF projection algorithm on the CPU [LFB10]

Initialize $g(u_e, v_e, \phi) := 0, \forall (u_e, v_e, \phi)$
for ϕ, projection angle, **do**
  for $x_e, y_e \in \{0, \ldots, (N_x - 1)\} \times \{0, \ldots, (N_y - 1)\}$ **do**
    $x'_e := x_e - x_{Lr_e}$ and $y'_e := y_e - y_{Lr_e}$
    if $(x'_e)^2 + (y'_e)^2 \leqslant \left(\frac{N_x}{2}\right)^2$ **then**
      Rotate : $x_{\phi_e} := x'_e \cos\phi + y'_e \sin\phi + \frac{R}{\delta}$ and $y_{\phi_e} := -x'_e \sin\phi + y'_e \cos\phi$
      Compute l_{ψ_v} by (B.56) (A2 method)
      Compute and sort the projections of $(x_e \pm 0.5, y_e \pm 0.5)$ by (B.30) : $\tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$
      Compute $u_{e_{min}} = \lfloor \max\{0, u_{min}\} \rfloor$ and $u_{e_{max}} = \lceil \min\{N_u - 1, u_{max}\} \rceil$ by (B.66)
      if $u_{min} \leqslant N_u - 1$ and $u_{max} \geqslant 0$ **then**
        for $u_e = u_{e_{min}}, \ldots, u_{e_{max}}$ **do**
          $F_{trans} := 0$ (initialize the transaxial footprint)
          $s_{e_1} := \max\{s_{u_e} - 0.5, \tau_0\}$ and $s_{e_2} := \min\{s_{u_e} + 0.5, \tau_1\}$
          if $s_{e_1} < s_{e_2}$ **then** $F_{trans} += 0.5 \times \left((s_{e_2} - \tau_0)^2 - (s_{e_1} - \tau_0)^2\right) / (\tau_1 - \tau_0)$ **end if**
          $s_{e_1} := \max\{s_{u_e} - 0.5, \tau_1\}$ and $s_{e_2} := \min\{s_{u_e} + 0.5, \tau_2\}$
          if $s_{e_1} < s_{e_2}$ **then** $F_{trans} += (s_{e_2} - s_{e_1})$ **end if**
          $s_{e_1} := \max\{s_{u_e} - 0.5, \tau_2\}$ and $s_{e_2} := \min\{s_{u_e} + 0.5, \tau_3\}$
          if $s_{e_1} < s_{e_2}$ **then** $F_{trans} += 0.5 \times \left((\tau_3 - s_{e_1})^2 - (\tau_3 - s_{e_2})^2\right) / (\tau_3 - \tau_2)$ **end if**
          $F'_{trans}(u_e; x_e, y_e) := l_{\psi_v} F_{trans}$
        end for
        for $v_e = 0, \ldots, (N_v - 1)$ **do**
          Compute $z_{e_{min}} = \lfloor \max\{0, z_{min}\} \rfloor$ and $z_{e_{max}} = \lceil \min\{N_z - 1, z_{max}\} \rceil$ by (B.71)
          if $z_{min} \leqslant N_z - 1$ and $z_{max} \geqslant 0$ **then**
            $F'_{ax} := 0$ (initialize the summed axial footprint)
            for $z_e = z_{e_{min}}, \ldots, z_{e_{max}}$ **do**
              Compute χ_0 and χ_1 by (B.33)
              $s_{e_1} := \max\{t_{v_e} - 0.5, \chi_0\}$ and $s_{e_2} := \min\{t_{v_e} + 0.5, \chi_1\}$
              if $s_{e_1} < s_{e_2}$ **then**
                $F'_{ax} += (s_{e_2} - s_{e_1}) \times f(x_e, y_e, z_e)$
              end if
            end for
            for $u_e = u_{e_{min}}, \ldots, u_{e_{max}}$ **do**
              $g(u_e, v_e, \phi) += F'_{ax} \times F'_{trans}(u_e; x_e, y_e)$
            end for
          end if
        end for
      end if
    end if
  end for
  for $v_e = 0, \ldots, (N_v - 1)$ **do**
    for $u_e = 0, \ldots, (N_u - 1)$ **do**
      Scale $g(u_e, v_e, \phi)$ by $l_{\theta_c}(u_e, v_e)$, given by (B.50) (A2 method)
    end for
  end for
end for

where

$$\begin{cases} v_{e_{min}} &= \lfloor \max\left\{0, v_{C_{d_e}} - 0.5 + \chi_0\right\} \rfloor \\ v_{e_{max}} &= \lceil \min\left\{N_v - 1, v_{C_{d_e}} + 0.5 + \chi_1\right\} \rceil \end{cases}. \tag{B.82}$$

Running a small loop over v_e, $v_{e_{min}} \leqslant v_e \leqslant v_{e_{max}}$, the backprojection at angle ϕ

$$\begin{aligned} b_\phi(x_e, y_e, z_e) &= \sum_{v_e} F_{ax}(v_e, \phi; x_e, y_e, z_e) \bar{b}_{trans}(v_e, \phi; x_e, y_e) \\ &= \sum_{v_e = v_{e_{min}}}^{v_{e_{max}}} F_{ax}(v_e, \phi; x_e, y_e, z_e) \bar{b}_{trans}(v_e, \phi; x_e, y_e) \end{aligned} \tag{B.83}$$

is computed and accumulated into the total backprojection

$$b(x_e, y_e, z_e) += b_\phi(x_e, y_e, z_e). \tag{B.84}$$

The overall algorithm is given in algorithm 18.

B.3 Projection and backprojection simulations

The simulations compute 64 projections of the Shepp-Logan phantom [SL74] shown in figure 3.6. The source-to-object distance is 98 mm, while the source-to-detector is 230 mm. The phantom is $5 \times 5 \times 5$ mm^3 and is sampled into 256^3 voxels. The detector has 256^2 cells and is 14.2×14.2 mm^2. By computing the Normalized Root Mean Square Error (NRMSE), given by formula (3.37), with respect to analytical projections shown in figure B.9, we compare the projections obtained by the Siddon-Jacobs-Han's projector and the SF projector on the CPU, respectively shown in figures B.10 and B.11.

Projector (on the CPU)	Computation time	NRMSE with respect to the analytical projections
Siddon-Jacobs-Han's projector	**15.8 s**	2.26 %
SF projector	155.1 s	**0.71 %**

Table B.1: Comparison of the Siddon-Jacobs-Han's projector and the SF projector on the CPU

Backprojector (on the CPU)	Computation time
SF backprojector	118.3 s

Table B.2: Computation time of the SF backprojector on the CPU

Due to the fact that it does not take into account the detector blur, the Siddon-Jacobs-Han's method presented in section B.1 returns projections which are very non-smooth. On the contrary, the SF projector gives images of better quality, which look more similar to the analytical projections. This is confirmed in table B.1, where the NRMSE with respect to the analytical projections is the lowest for the SF projector. Concerning the computation time, the Siddon-Jacobs-Han's projector is faster than the SF projector due to its iterative procedure. Nevertheless, this iterative nature makes it more sensitive to computational errors. Furthermore, this iterative nature is responsible for the unefficiency of its adjoint backprojector, even on the GPU [NL15].

The computation time of the SF backprojector on the CPU is given in table B.2 and is rather the same as the SF projector due to the fact that the operators are coupled. In figure B.12, we see that the obtained SF backprojection is smooth and does not present artifacts due to possible computational errors or mismatching. Consequently, our implementation of the SF pair on the CPU is fully validated and can be used as a reference for the validation of the SF projector and the SF backprojector on the GPU in chapter 3.

Algorithm 18 SF backprojection algorithm on the CPU [LFB10]

Initialize $b(x_e, y_e, z_e) := 0, \forall (x_e, y_e, z_e)$
for ϕ, projection angle, **do**
  for $v_e = 0, \ldots, (N_v - 1)$ **do**
    for $u_e = 0, \ldots, (N_u - 1)$ **do**
      Compute $\bar{g}(u_e, v_e, \phi)$ by (B.77) (A2 method)
    end for
  end for
  for $x_e, y_e \in \{0, \ldots, (N_x - 1)\} \times \{0, \ldots, (N_y - 1)\}$ **do**
    $x'_e := x_e - x_{Lr_e}$ and $y'_e := y_e - y_{Lr_e}$
    if $(x'_e)^2 + (y'_e)^2 \leqslant \left(\frac{N_x}{2}\right)^2$ **then**
      Rotate : $x_{\phi_e} := x'_e \cos\phi + y'_e \sin\phi + \frac{R}{\delta}$ and $y_{\phi_e} := -x'_e \sin\phi + y'_e \cos\phi$
      Compute l_{ψ_v} by (B.56) (A2 method)
      Compute and sort the projections of $(x_e \pm 0.5, y_e \pm 0.5)$ by (B.30) : $\tau_0 \leqslant \tau_1 \leqslant \tau_2 \leqslant \tau_3$
      Compute $u_{e_{min}} = \lfloor \max\{0, u_{min}\} \rfloor$ and $u_{e_{max}} = \lceil \min\{N_u - 1, u_{max}\} \rceil$ by (B.66)
      if $u_{min} \leqslant N_u - 1$ and $u_{max} \geqslant 0$ **then**
        for $u_e = u_{e_{min}}, \ldots, u_{e_{max}}$ **do**
          $F_{trans} := 0$ (initialize the transaxial footprint)
          $s_{e_1} := \max\{s_{u_e} - 0.5, \tau_0\}$ and $s_{e_2} := \min\{s_{u_e} + 0.5, \tau_1\}$
          if $s_{e_1} < s_{e_2}$ **then** $F_{trans} += 0.5 \times \left((s_{e_2} - \tau_0)^2 - (s_{e_1} - \tau_0)^2\right) / (\tau_1 - \tau_0)$ **end if**
          $s_{e_1} := \max\{s_{u_e} - 0.5, \tau_1\}$ and $s_{e_2} := \min\{s_{u_e} + 0.5, \tau_2\}$
          if $s_{e_1} < s_{e_2}$ **then** $F_{trans} += (s_{e_2} - s_{e_1})$ **end if**
          $s_{e_1} := \max\{s_{u_e} - 0.5, \tau_2\}$ and $s_{e_2} := \min\{s_{u_e} + 0.5, \tau_3\}$
          if $s_{e_1} < s_{e_2}$ **then** $F_{trans} += 0.5 \times \left((\tau_3 - s_{e_1})^2 - (\tau_3 - s_{e_2})^2\right) / (\tau_3 - \tau_2)$ **end if**
          $F'_{trans}(u_e; x_e, y_e) := l_{\psi_v} F_{trans}$
        end for
        for $v_e = 0, \ldots, (N_v - 1)$ **do**
          Initialize $\bar{b}_{trans}(v_e) := 0$
          for $u_e = u_{e_{min}}, \ldots, u_{e_{max}}$ **do**
            $\bar{b}_{trans}(v_e) += \bar{g}(u_e, v_e, \phi) \times F'_{trans}(u_e; x_e, y_e)$
          end for
        end for
        for $z_e = 0, \ldots, (N_z - 1)$ **do**
          Compute χ_0 and χ_1 by (B.33)
          Compute $v_{e_{min}} = \lfloor \max\{0, v_{min}\} \rfloor$ and $v_{e_{max}} = \lceil \min\{N_v - 1, v_{max}\} \rceil$ by (B.82)
          if $v_{min} \leqslant N_v - 1$ and $v_{max} \geqslant 0$ **then**
            $b_\phi := 0$
            for $v_e = v_{e_{min}}, \ldots, v_{e_{max}}$ **do**
              $s_{e_1} := \max\{t_{v_e} - 0.5, \chi_0\}$ and $s_{e_2} := \min\{t_{v_e} + 0.5, \chi_1\}$
              if $s_{e_1} < s_{e_2}$ **then**
                $b_\phi += (s_{e_2} - s_{e_1})\, \bar{b}_{trans}(v_e)$
              end if
            end for
            $b(x_e, y_e, z_e) += b_\phi$
          end if
        end for
      end if
    end if
  end for
end for

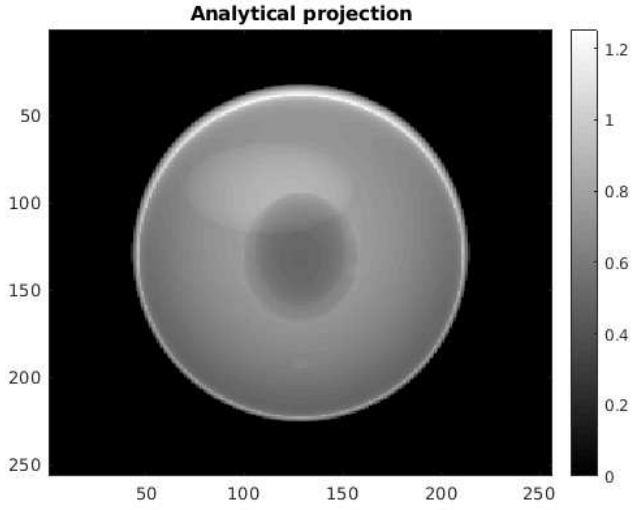


Figure B.9: Analytical projections of the Shepp-Logan phantom

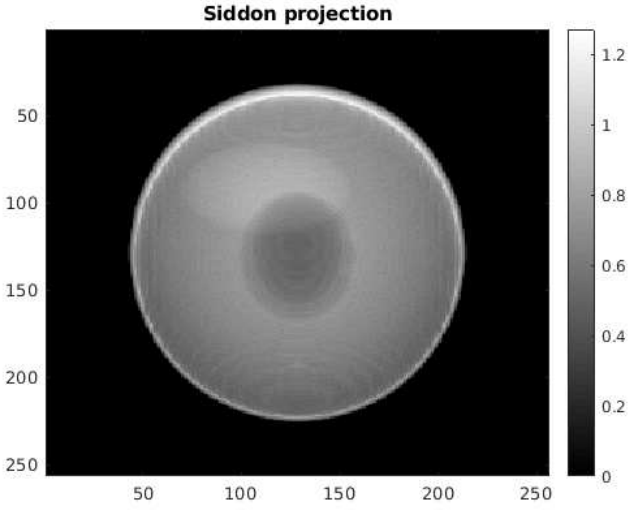


Figure B.10: Siddon-Jacobs-Han's projection on the CPU

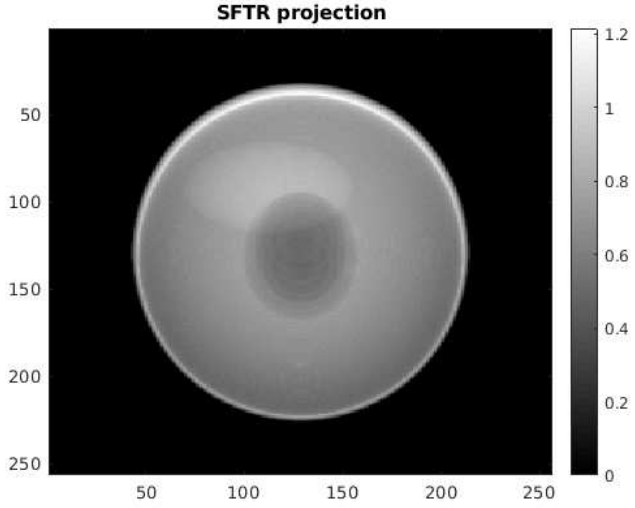


Figure B.11: SF projection on the CPU

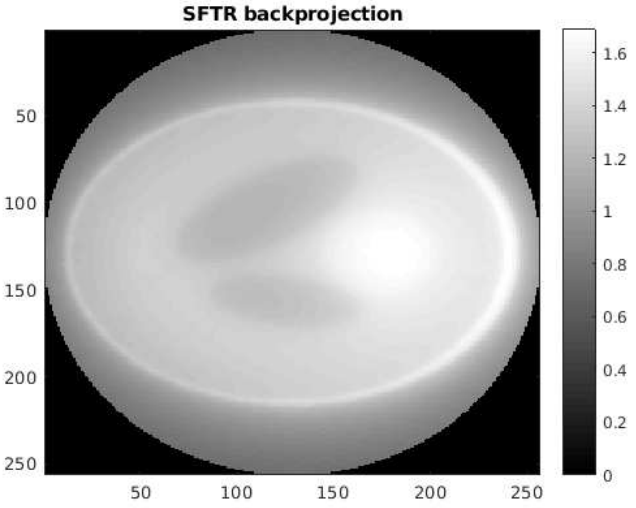


Figure B.12: SF backprojection on the CPU

Appendix C

Derivation of the error-splitting forward model for X-ray CT

We consider the following Poisson statistics for photon-count

$$p(I_i|\boldsymbol{f}, n_i, r_i) = \mathcal{P}\left(I_0 e^{-[\boldsymbol{Hf}]_i} + n_i + r_i\right) \tag{C.1}$$

where I_0 is the mean number of photons sent by source, r_i is the mean number of background events and n_i is a noise term involving object-dependent scattering and absorption occuring inside the volume. I_0 is known, and r_i is pre-computed, while n_i is unknown, due to the fact that it is closely related to the volume. We introduce the reparametrization

$$g_i = \ln\left(\frac{I_0}{I_i - r_i}\right). \tag{C.2}$$

The opposite of the log-likelihood is

$$-\ln(p(I_i|\boldsymbol{f}, n_i, r_i)) = \ln(I_i!) + I_0 e^{-[\boldsymbol{Hf}]_i} + n_i + r_i - I_i \ln\left(I_0 e^{-[\boldsymbol{Hf}]_i} + n_i + r_i\right). \tag{C.3}$$

From equation (C.2) and by a second order Taylor series expansion, we have

$$\begin{aligned} I_0 e^{-[\boldsymbol{Hf}]_i} + n_i + r_i &= (I_i - r_i)\exp\left[g_i - [\boldsymbol{Hf}]_i\right] + n_i + r_i \\ &= I_i + n_i + (I_i - r_i)(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}(I_i - r_i)(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right). \end{aligned} \tag{C.4}$$

Hence

$$\begin{aligned} I_i \ln\left(I_0 e^{-[\boldsymbol{Hf}]_i} + n_i + r_i\right) &= I_i \ln\left((I_i - r_i)\exp\left[g_i - [\boldsymbol{Hf}]_i\right] + n_i + r_i\right) \\ &= I_i \ln\left(I_i + n_i + (I_i - r_i)(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}(I_i - r_i)(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right) \\ &= I_i \left[\ln(I_i + n_i) + \ln\left(1 + \frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right)\right] \\ &= I_i \left[\ln(I_i + n_i) + \frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{Hf}]_i)^2 - \frac{1}{2}\left(\frac{I_i - r_i}{I_i + n_i}\right)^2 (g_i - [\boldsymbol{Hf}]_i)^2 \right. \\ &\qquad \left. + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \\ &= I_i \left[\ln(I_i + n_i) + \frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\left[\frac{I_i - r_i}{I_i + n_i} - \left(\frac{I_i - r_i}{I_i + n_i}\right)^2\right](g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \\ &= I_i \left[\ln(I_i + n_i) + \frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{Hf}]_i) + \frac{1}{2}\frac{I_i - r_i}{I_i + n_i}\left[1 - \frac{I_i - r_i}{I_i + n_i}\right](g_i - [\boldsymbol{Hf}]_i)^2 + o\left((g_i - [\boldsymbol{Hf}]_i)^2\right)\right] \end{aligned}$$

$$= I_i \left[\ln(I_i + n_i) + \frac{I_i - r_i}{I_i + n_i}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i) + \frac{1}{2}\frac{(I_i - r_i)(n_i + r_i)}{(I_i + n_i)^2}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right) \right]$$

$$= I_i \ln(I_i + n_i) + \frac{I_i(I_i - r_i)}{I_i + n_i}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i) + \frac{1}{2}\frac{I_i(I_i - r_i)(n_i + r_i)}{(I_i + n_i)^2}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right). \quad \text{(C.5)}$$

Thus, after reparametrization (C.2), we have for (C.3), by gathering (C.4) and (C.5) :

$$\begin{aligned} -\ln(p(g_i|\boldsymbol{f}, n_i, r_i)) &= \ln(I_i!) + I_i + n_i - I_i \ln(I_i + n_i) + \left[(I_i - r_i) - \frac{I_i(I_i - r_i)}{I_i + n_i}\right](g_i - [\boldsymbol{H}\boldsymbol{f}]_i) \\ &\quad + \frac{1}{2}\left[(I_i - r_i) - \frac{I_i(I_i - r_i)(n_i + r_i)}{(I_i + n_i)^2}\right](g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right) \\ &= \ln(I_i!) + I_i + n_i - I_i \ln(I_i + n_i) + b_i(g_i - [\boldsymbol{H}\boldsymbol{f}]_i) + \frac{a_i}{2}(g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2 + o\left((g_i - [\boldsymbol{H}\boldsymbol{f}]_i)^2\right) \end{aligned} \quad \text{(C.6)}$$

where

$$b_i = (I_i - r_i) - \frac{I_i(I_i - r_i)}{I_i + n_i} = (I_i - r_i)\left[1 - \frac{I_i}{I_i + n_i}\right] = \frac{n_i(I_i - r_i)}{I_i + n_i} \quad \text{(C.7)}$$

and

$$a_i = (I_i - r_i) - \frac{I_i(I_i - r_i)(n_i + r_i)}{(I_i + n_i)^2} = (I_i - r_i)\left(1 - \frac{I_i(n_i + r_i)}{(I_i + n_i)^2}\right). \quad \text{(C.8)}$$

Hence, from (C.6),

$$\begin{aligned} -\ln(p(g_i|\boldsymbol{f}, n_i, r_i)) &\approx \ln(I_i!) + I_i + n_i - I_i \ln(I_i + n_i) + \frac{a_i}{2}\left(g_i - [\boldsymbol{H}\boldsymbol{f}]_i + \frac{b_i}{a_i}\right)^2 - \frac{b_i^2}{2a_i} \\ &= C_i + \frac{1}{2v_{\epsilon_i}}\left(g_i - [\boldsymbol{H}\boldsymbol{f}]_i - \xi_i\right)^2 \end{aligned} \quad \text{(C.9)}$$

where

$$v_{\epsilon_i} = \frac{1}{a_i} \quad \text{(C.10)}$$

and

$$\xi_i = -\frac{b_i}{a_i}. \quad \text{(C.11)}$$

Term

$$C_i = \ln(I_i!) + I_i + n_i - I_i \ln(I_i + n_i) - \frac{b_i^2}{2a_i} \quad \text{(C.12)}$$

does not include $\boldsymbol{f}$. We have the bias

$$\xi_i = -\frac{b_i}{a_i} = -\frac{n_i(I_i - r_i)}{I_i + n_i} \times \frac{(I_i + n_i)^2}{(I_i - r_i)\left[(I_i + n_i)^2 - I_i(n_i + r_i)\right]} = -\frac{n_i(I_i + n_i)}{(I_i + n_i)^2 - I_i(n_i + r_i)} \quad \text{(C.13)}$$

which we call the linear model uncertainty for projection i.

Appendix D

Calculations for fixing the parameters of the error-splitting forward model

We suppose β_{ϵ_0} is correctly fixed to comply with constraint (5.28), so measurement uncertainties are actually Gaussian. In order to do simple calculations, we make the assumption that "true" projections $\boldsymbol{g}_0$ and measurement uncertainties $\boldsymbol{\epsilon}$ are independent, and we remove the expectations :

$$\|\boldsymbol{g}\|_2^2 = \|\boldsymbol{g}_0\|_2^2 + \|\boldsymbol{\epsilon}\|_2^2. \tag{D.1}$$

We define parameter SNR by :

$$SNR = 10 \log \left(\frac{\|\boldsymbol{g}_0\|_2^2}{\|\boldsymbol{\epsilon}\|_2^2} \right) = 10 \log \left(\frac{\|\boldsymbol{g}\|_2^2}{\|\boldsymbol{\epsilon}\|_2^2} - 1 \right). \tag{D.2}$$

Since we approximate $\|\boldsymbol{\epsilon}\|_2^2$ by its expectation, we have

$$\|\boldsymbol{\epsilon}\|_2^2 \approx \mathbb{E}\left(\|\boldsymbol{\epsilon}\|_2^2 | \alpha_{\epsilon_0}, \beta_{\epsilon_0} \right) = \sum_{i=1}^{M} \mathbb{E}\left(\epsilon_i^2 | \alpha_{\epsilon_0}, \beta_{\epsilon_0} \right) \tag{D.3}$$

and

$$\mathbb{E}\left(\epsilon_i^2 | \alpha_{\epsilon_0}, \beta_{\epsilon_0} \right) = \frac{\beta_{\epsilon_0}}{\alpha_{\epsilon_0} + \frac{1}{2}} \tag{D.4}$$

according to (5.30). Combining equations (D.2) and (D.3), we obtain

$$\alpha_{\epsilon_0} = M \times \frac{\beta_{\epsilon_0}}{\|\boldsymbol{g}\|_2^2} \times \left(1 + 10^{\frac{SNR}{10}} \right) - \frac{1}{2}, \tag{D.5}$$

in order to automatically fix α_{ϵ_0} with respect to the data and parameter SNR.

Appendix E

Estimation theory

An estimator $\hat{\boldsymbol{\psi}}$ of unknowns $\boldsymbol{\psi}_0$ is defined as the minimizer of a cost function C which penalizes the distance between $\hat{\boldsymbol{\psi}}$ and $\boldsymbol{\psi}_0$. Since $\boldsymbol{\psi}_0$ is unknown, $\hat{\boldsymbol{\psi}}$ minimizes this cost in average over $\boldsymbol{\psi}_0$:

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} \mathbb{E}_{\boldsymbol{\psi}_0}(C(\boldsymbol{\psi}_0, \boldsymbol{\psi})|\boldsymbol{g}) \tag{E.1}$$

where the expectation

$$\mathbb{E}_{\boldsymbol{\psi}_0}(C(\boldsymbol{\psi}_0, \boldsymbol{\psi})|\boldsymbol{g}) = \int_{\boldsymbol{\psi}_0} C(\boldsymbol{\psi}_0, \boldsymbol{\psi}) p(\boldsymbol{\psi}_0|\boldsymbol{g})\, \mathrm{d}\boldsymbol{\psi}_0 \tag{E.2}$$

is computed with respect to the posterior distribution of $\boldsymbol{\psi}_0$ given data $\boldsymbol{g}$. Depending on the choice of $C(\boldsymbol{\psi}_0, \boldsymbol{\psi})$, several estimators $\hat{\boldsymbol{\psi}}$ are possible.

E.1 Maximum a posteriori

If

$$C(\boldsymbol{\psi}_0, \boldsymbol{\psi}) = 1 - \delta(\boldsymbol{\psi}_0 - \boldsymbol{\psi}), \tag{E.3}$$

where δ denotes the Dirac delta function, then we have :

$$\mathbb{E}_{\boldsymbol{\psi}_0}(C(\boldsymbol{\psi}_0, \boldsymbol{\psi})|\boldsymbol{g}) = 1 - \mathbb{E}_{\boldsymbol{\psi}_0}(\delta(\boldsymbol{\psi}_0 - \boldsymbol{\psi})|\boldsymbol{g}) = 1 - p(\boldsymbol{\psi}|\boldsymbol{g}). \tag{E.4}$$

As a result, $\hat{\boldsymbol{\psi}}$ corresponds to the maximum a posteriori (MAP) :

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} \{1 - p(\boldsymbol{\psi}|\boldsymbol{g})\} = \arg\max_{\boldsymbol{\psi}} p(\boldsymbol{\psi}|\boldsymbol{g}) \tag{E.5}$$

E.2 Minimum Mean Square Error

If

$$C(\boldsymbol{\psi}_0, \boldsymbol{\psi}) = \|\boldsymbol{\psi}_0 - \boldsymbol{\psi}\|_2^2, \tag{E.6}$$

then the estimator

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} \mathbb{E}_{\boldsymbol{\psi}_0}(\|\boldsymbol{\psi}_0 - \boldsymbol{\psi}\|_2^2|\boldsymbol{g}) \tag{E.7}$$

is the Minimum Mean Square Error (MMSE). The derivative of the averaged cost function reads

$$\frac{\partial \mathbb{E}_{\boldsymbol{\psi}_0}(C(\boldsymbol{\psi}_0, \boldsymbol{\psi}))}{\partial \boldsymbol{\psi}} = \frac{\partial}{\partial \boldsymbol{\psi}} \left(\|\boldsymbol{\psi}\|_2^2 + \mathbb{E}_{\boldsymbol{\psi}_0}(\|\boldsymbol{\psi}_0\|_2^2|\boldsymbol{g}) - 2\boldsymbol{\psi}^T \mathbb{E}_{\boldsymbol{\psi}_0}(\boldsymbol{\psi}_0|\boldsymbol{g}) \right) = 2 \times \left(\boldsymbol{\psi} - \mathbb{E}_{\boldsymbol{\psi}_0}(\boldsymbol{\psi}_0|\boldsymbol{g}) \right). \tag{E.8}$$

Hence, the derivative is zero at

$$\hat{\boldsymbol{\psi}} = \mathbb{E}_{\boldsymbol{\psi}_0}(\boldsymbol{\psi}_0|\boldsymbol{g}). \tag{E.9}$$

Consequently, the MMSE corresponds to the Posterior Mean (PM).

E.3 Minimum Mean Absolute Error

If

$$C(\boldsymbol{\psi}_0, \boldsymbol{\psi}) = \|\boldsymbol{\psi}_0 - \boldsymbol{\psi}\|_1 = \sum_j |\psi_{0_j} - \psi_j|, \tag{E.10}$$

then the estimator

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} \mathbb{E}_{\boldsymbol{\psi}_0}(\|\boldsymbol{\psi}_0 - \boldsymbol{\psi}\|_1 | \boldsymbol{g}) \tag{E.11}$$

is the Minimum Mean Absolute Error (MMAE). The averaged cost function reads

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\psi}_0}(\|\boldsymbol{\psi}_0 - \boldsymbol{\psi}\|_1 | \boldsymbol{g}) &= \sum_j \int_{-\infty}^{+\infty} |\psi_{0_j} - \psi_j| p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} \\
&= \sum_j \left[\int_{-\infty}^{\psi_j} (\psi_{0_j} - \psi_j) p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} - \int_{\psi_j}^{+\infty} (\psi_{0_j} - \psi_j) p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} \right] \\
&= \sum_j \left[\psi_j \left(\int_{-\infty}^{\psi_j} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} - \int_{\psi_j}^{+\infty} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} \right) \right. \\
&\quad \left. - \int_{-\infty}^{\psi_j} \psi_{0_j} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} + \int_{\psi_j}^{+\infty} \psi_{0_j} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} \right].
\end{aligned} \tag{E.12}$$

The derivative of the averaged cost function (E.12) is

$$\begin{aligned}
\frac{\partial \mathbb{E}_{\boldsymbol{\psi}_0}(C(\boldsymbol{\psi}_0, \boldsymbol{\psi}))}{\partial \psi_j} &= \left(\int_{-\infty}^{\psi_j} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} - \int_{\psi_j}^{+\infty} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} \right) \\
&\quad + \psi_j \left(p(\psi_j | \boldsymbol{g}) + p(\psi_j | \boldsymbol{g}) \right) - \psi_j p(\psi_j | \boldsymbol{g}) - \psi_j p(\psi_j | \boldsymbol{g}) \\
&= \int_{-\infty}^{\psi_j} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} - \int_{\psi_j}^{+\infty} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j}.
\end{aligned} \tag{E.13}$$

Consequently, the derivative of the averaged cost function is zero at $\hat{\boldsymbol{\psi}}$ such that

$$\int_{-\infty}^{\hat{\psi}_j} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j} = \int_{\hat{\psi}_j}^{+\infty} p(\psi_{0_j} | \boldsymbol{g}) \, \mathrm{d}\psi_{0_j}, \forall j. \tag{E.14}$$

Given equation (E.14), we see that the MMAE corresponds to the median of the posterior distribution of $\boldsymbol{\psi}$. Computing the median requires to get samples at the tail of the posterior distribution and on its modes. This requires numerous numerical simulations. On the contrary, in order to compute the posterior mean, only samples on the modes of the posterior distribution are necessary. These samples are simpler to get thanks to the use of dedicated MCMC algorithms which aim at accelerating the convergence of the generated Markov chain towards the modes of the posterior distribution [MBBCP14]. Given these points, the MMSE is generally preferred to the MMAE.

Appendix F

Calculations for Variational Bayesian Approach

In section F.1, we detail the calculations of the approximate distribution

$$q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}) = \prod_{j=1}^{N} q_{f_j}(f_j|z_j) \times \prod_{j=1}^{N} q_{z_j}(z_j) \times \prod_{i=1}^{M} q_{\rho_{\zeta_i}}(\rho_{\zeta_i}) \times \prod_{k=1}^{K} q_{m_k}(m_k) \times \prod_{k=1}^{K} q_{\rho_k}(\rho_k). \tag{F.1}$$

for the VBA algorithm described in chapter 6. Then, in section F.2, we give the calculations of the negative free energy which is the stopping criterion of the VBA algorithm. We omit subscripts corresponding to the iterations of the algorithm for the sake of readability. In the calculations, we drop unuseful terms in the expressions of many logarithms. In order to show these droppings, we use the notation

$$\ln a \propto b$$

when

$$\ln a = b + C,$$

where C is a term unuseful for the calculations. We remind the auxiliary variables used in the calculations :

$$\begin{cases} \tilde{m}_j = \sum_{k=1}^{K} \tilde{m}_{jk} q_{z_j}(k) \\ \tilde{v}_j = \sum_{k=1}^{K} \tilde{v}_{jk} q_{z_j}(k) \\ \tilde{m}_j^{(2)} = \sum_{k=1}^{K} \left(\tilde{m}_{jk} - \tilde{m}_j\right)^2 q_{z_j}(k) = \sum_{k=1}^{K} \tilde{m}_{jk}^2 q_{z_j}(k) - \tilde{m}_j^2 \\ \tilde{v}_j^{(2)} = \tilde{v}_j + \tilde{m}_j^{(2)} \\ \tilde{v}_{\zeta_i} = \frac{\tilde{\beta}_{\zeta_{0_i}}}{\tilde{\alpha}_{\zeta_{0_i}}} \end{cases} \tag{F.2}$$

and $\tilde{\boldsymbol{V}}_\zeta = \text{diag}\left[\tilde{\boldsymbol{v}}_\zeta\right]$.

F.1 Calculations of the approximate distributions

F.1.1 Calculation of the approximate distribution for the volume

We compute $q_{f_j}(f_j|z_j = k)$ for fixed $j \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$. According to Bayes' rule, we have

$$\ln\left(p(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right) \propto \ln\left(p\left(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{\rho}_\zeta\right)\right) + \ln\left(p\left(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}\right)\right). \tag{F.3}$$

We drop terms independent from f_j and z_j and take the expectation with respect to the approximate distribution as required by formula (6.34). For the first term in (F.3), we have

$$\begin{aligned}
\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right) &= -\frac{1}{2}\sum_{i=1}^{M}\rho_{\zeta_i}\left(g_i-[\boldsymbol{H}\boldsymbol{f}]_i\right)^2-\frac{M}{2}\ln(2\pi)+\frac{1}{2}\sum_{i=1}^{M}\ln\left(\rho_{\zeta_i}\right)\\
&= -\sum_{i=1}^{M}\rho_{\zeta_i}\frac{g_i^2}{2}-\frac{M}{2}\ln(2\pi)+\frac{1}{2}\sum_{i=1}^{M}\ln\left(\rho_{\zeta_i}\right)-\sum_{i=1}^{M}\rho_{\zeta_i}\frac{[\boldsymbol{H}\boldsymbol{f}]_i^2}{2}+\sum_{i=1}^{M}\rho_{\zeta_i}g_i\left[\boldsymbol{H}\boldsymbol{f}\right]_i\\
&\propto -\sum_{i=1}^{M}\rho_{\zeta_i}\frac{[\boldsymbol{H}\boldsymbol{f}]_i^2}{2}+\sum_{i=1}^{M}\rho_{\zeta_i}g_i\left[\boldsymbol{H}\boldsymbol{f}\right]_i\\
&\propto -\frac{f_j^2}{2}\left(\sum_{i=1}^{M}\rho_{\zeta_i}H_{ij}^2\right)+f_j\sum_{i=1}^{M}H_{ij}\rho_{\zeta_i}\left(g_i-\sum_{\substack{l=1\\l\neq j}}^{N}H_{il}f_l\right)\\
&\propto -\frac{f_j^2}{2}\left(\sum_{i=1}^{M}\rho_{\zeta_i}H_{ij}^2\right)+f_j\sum_{i=1}^{M}H_{ij}\rho_{\zeta_i}\left(g_i-\sum_{\substack{l=1\\l\neq j}}^{N}H_{il}\sum_{k'=1}^{K}f_l\delta(z_l-k')\right).
\end{aligned}\tag{F.4}$$

According to section F.1.3,

$$q_{\rho_{\zeta_i}}(\rho_{\zeta_i})=\mathcal{G}(\rho_{\zeta_i}|\tilde{\alpha}_{\zeta_{0_i}},\tilde{\beta}_{\zeta_{0_i}}).\tag{F.5}$$

Hence, given $z_j=k$, the expectation reads

$$\mathbb{E}_{q_{/f_j,z_j}}\left(\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)\right)\propto -\frac{f_j^2}{2}\sum_{i=1}^{M}H_{ij}^2\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}+f_j\sum_{i=1}^{M}H_{ij}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}\left(g_i-\sum_{\substack{l=1\\l\neq j}}^{N}H_{il}\tilde{m}_l\right)\tag{F.6}$$

where

$$\tilde{m}_{lk'}=\mathbb{E}_{q_{f_l}}(f_l|z_j=k').\tag{F.7}$$

and

$$\tilde{m}_l=\sum_{k'=1}^{K}\tilde{m}_{lk'}q_{z_l}(k').\tag{F.8}$$

Introducing

$$\tilde{v}_{\zeta_i}=\frac{\tilde{\beta}_{\zeta_{0_i}}}{\tilde{\alpha}_{\zeta_{0_i}}}\tag{F.9}$$

and $\tilde{\boldsymbol{V}}_\zeta=\text{diag}\left[\tilde{\boldsymbol{v}}_\zeta\right]$, we have

$$\sum_{i=1}^{M}H_{ij}^2\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}=\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}.\tag{F.10}$$

Consequently, the expectation can be expressed as

$$\mathbb{E}_{q_{/f_j,z_j}}\left(\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)\right)\propto -\frac{f_j^2}{2}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}+f_j\left(\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}(\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j+\tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\right).\tag{F.11}$$

We do the same for the second term in (F.3) :

$$\begin{aligned}
\ln\left(p\left(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}\right)\right) &= -\frac{1}{2}\sum_{k=1}^{K}\sum_{j\in\mathcal{R}_k}\rho_k(f_j-m_k)^2-\frac{N}{2}\ln\left(2\pi\right)+\frac{1}{2}\sum_{k=1}^{K}\sum_{j\in\mathcal{R}_k}\ln(\rho_k)\\
&\propto -\frac{\rho_k}{2}\left(f_j-m_k\right)^2\propto -\frac{f_j^2}{2}\rho_k+f_j\rho_k m_k.
\end{aligned}\tag{F.12}$$

According to sections F.1.4 and F.1.5, the approximate distributions for the means and the inverses of the variances of the classes are

$$q_{m_k}(m_k) = \mathcal{N}(m_k|\tilde{m}_{0_k}, \tilde{v}_{0_k}) \tag{F.13}$$

and

$$q_{\rho_k}(\rho_k) = \mathcal{G}(\rho_k|\tilde{\alpha}_{0_k}, \tilde{\beta}_{0_k}). \tag{F.14}$$

Consequently, it leads to

$$\mathbb{E}_{q_{/f_j,z_j}}\left(\ln\left(p\left(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}\right)\right)\right) \propto -\frac{f_j^2}{2}\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} + f_j\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\tilde{m}_{0_k}. \tag{F.15}$$

Gathering equations (F.11) and (F.15), the approximate distribution for the volume is

$$q_{f_j}(f_j|z_j = k) = \mathcal{N}(f_j|\tilde{m}_{jk}, \tilde{v}_{jk}), \forall j,\, k \tag{F.16}$$

where

$$\left\{\begin{array}{l} \tilde{v}_{jk} = \left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} + \left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\right)^{-1} \\ \tilde{m}_{jk} = \tilde{v}_{jk}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\tilde{m}_{0_k} + \left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}(\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j + \tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\right) \end{array}\right. . \tag{F.17}$$

Given that

$$\tilde{v}_{jk}\tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj} = \tilde{v}_{jk}\tilde{m}_j\left(\frac{1}{\tilde{v}_{jk}} - \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\right) = \tilde{m}_j - \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\tilde{v}_{jk}\tilde{m}_j, \tag{F.18}$$

we obtain the updating formulae for the mean and the variance of $q_{f_j}(.|z_j = k)$:

$$\left\{\begin{array}{l} \tilde{v}_{jk} = \left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}} + \left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\right)^{-1} \\ \tilde{m}_{jk} = \tilde{m}_j + \tilde{v}_{jk}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}(\tilde{m}_{0_k} - \tilde{m}_j) + \left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}(\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j\right) \end{array}\right. . \tag{F.19}$$

F.1.2 Calculation of the approximate distribution for the labels

We compute q_{z_j} for fixed $j \in \{1,\ldots,N\}$. This approximate distribution is calculated based on the formula (6.41) proven in section 6.1.2. Since a dependence is preserved between f_j and z_j in the approximate joint posterior distribution (F.1), q_{z_j} appears in the expectation of the log-likelihood $\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)$. For this reason, the log-likelihood is not dropped in Bayes' rule

$$\ln\left(p(\boldsymbol{f},\boldsymbol{\rho}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}|\boldsymbol{g};\mathcal{U},\mathcal{M})\right) \propto \ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right) + \ln\left(p\left(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}\right)\right) + \ln\left(p\left(\boldsymbol{z}|\boldsymbol{\alpha},\gamma_0\right)\right). \tag{F.20}$$

For the log-likelihood, we keep only the terms dependent on f_j :

$$\begin{aligned}\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right) \propto & -\sum_{k=1}^{K}\frac{f_j^2}{2}\delta(z_j-k)\left(\sum_{i=1}^{M}\rho_{\zeta_i}H_{ij}^2\right) \\ & + \sum_{k=1}^{K} f_j\delta(z_j-k)\sum_{i=1}^{M}H_{ij}\rho_{\zeta_i}\left(g_i - \sum_{\substack{l=1\\l\neq j}}^{N}H_{il}\sum_{k'=1}^{K}f_l\delta(z_l-k')\right).\end{aligned} \tag{F.21}$$

Hence, according to sections F.1.1 and F.1.3,

$$\begin{aligned}&\mathbb{E}_{q_{/z_j}}\left(\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)\right) \\ &\propto -\frac{1}{2}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\left(\sum_{k=1}^{K}\left(\tilde{v}_{jk}+\tilde{m}_{jk}^2\right)\delta(z_j-k)\right) + \left(\sum_{k=1}^{K}\tilde{m}_{jk}\delta(z_j-k)\right)\left(\sum_{i=1}^{M}H_{ij}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}\left(g_i - \sum_{\substack{l=1\\l\neq j}}^{N}H_{il}\tilde{m}_l\right)\right)\end{aligned}$$

$$\propto -\frac{1}{2}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\left(\sum_{k=1}^{K}\left(\tilde{v}_{jk}+\tilde{m}_{jk}^2\right)\delta(z_j-k)\right)+\left(\sum_{k=1}^{K}\tilde{m}_{jk}\delta(z_j-k)\right)\left(\sum_{i=1}^{M}H_{ij}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}\left(g_i-\sum_{l=1}^{N}H_{il}\tilde{m}_l\right)\right)$$
$$+\left(\sum_{k=1}^{K}\tilde{m}_{jk}\delta(z_j-k)\right)\tilde{m}_j\left(\sum_{i=1}^{M}H_{ij}^2\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}\right)$$
$$\propto -\frac{1}{2}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\left(\sum_{k=1}^{K}\left(\tilde{v}_{jk}+\tilde{m}_{jk}^2\right)\delta(z_j-k)\right)+\left(\sum_{k=1}^{K}\tilde{m}_{jk}\delta(z_j-k)\right)\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}(\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j$$
$$+\left(\sum_{k=1}^{K}\tilde{m}_{jk}\delta(z_j-k)\right)\tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}$$
$$\propto\sum_{k=1}^{K}\left(-\frac{1}{2}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\left(\tilde{v}_{jk}+\tilde{m}_{jk}^2\right)+\tilde{m}_{jk}\left(\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}(\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j+\tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\right)\right)\delta(z_j-k)$$
$$\propto\left(-\frac{1}{2}\right)\sum_{k=1}^{K}\left(\left(\tilde{v}_{jk}+\tilde{m}_{jk}^2\right)\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}\right.$$
$$\left.-2\tilde{m}_{jk}\left(\tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}+\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}(\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j\right)\right)\delta(z_j-k). \tag{F.22}$$

For the second term in (F.20), we have, according to sections F.1.1, F.1.4 and F.1.5,

$$\mathbb{E}_{q_{/z_j}}\left(\ln\left(p\left(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}\right)\right)\right)\propto-\frac{1}{2}\sum_{k=1}^{K}\left(\mathbb{E}_{q_{f_j}q_{m_k}q_{\rho_k}}\left(\rho_k(f_j-m_k)^2|z_j=k\right)-\mathbb{E}_{q_{\rho_k}}(\ln(\rho_k))\right)\delta(z_j-k)$$
$$\propto-\frac{1}{2}\sum_{k=1}^{K}\left(\mathbb{E}_{q_{\rho_k}}(\rho_k)\left[\mathbb{E}_{q_{f_j}}\left(f_j^2|z_j=k\right)+\mathbb{E}_{q_{m_k}}\left(m_k^2\right)-2\mathbb{E}_{q_{f_j}}(f_j|z_j=k)\mathbb{E}_{q_{m_k}}(m_k)\right]-\mathbb{E}_{q_{\rho_k}}(\ln(\rho_k))\right)\delta(z_j-k)$$
$$\propto-\frac{1}{2}\sum_{k=1}^{K}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\left[\tilde{v}_{jk}+\tilde{v}_{0_k}+\left(\tilde{m}_{jk}-\tilde{m}_{0_k}\right)^2\right]+\ln(\tilde{\beta}_{0_k})-\psi(\tilde{\alpha}_{0_k})\right)\delta(z_j-k), \tag{F.23}$$

where ψ denotes the digamma function :

$$\psi(x)=\frac{\Gamma'(x)}{\Gamma(x)},\forall x\in\mathbb{R}_+^*. \tag{F.24}$$

The expectation of the third term in (F.20) is

$$\mathbb{E}_{q_{/z_j}}\left(\ln\left(p\left(\boldsymbol{z}|\boldsymbol{\alpha},\gamma_0\right)\right)\right)\propto\sum_{k=1}^{K}\left[\alpha_k+\gamma_0\sum_{i\in\mathcal{V}(j)}q_{z_i}(k)\right]\delta(z_j-k) \tag{F.25}$$

where $q_{z_i}(k),\forall i\in\mathcal{V}(j)$, is $q_{z_i}(k)$ computed at the previous iteration of the VBA algorithm. In order to apply the formula (6.41), we have to compute the entropy of the approximate distribution q_{f_j}. Since, according to section F.1.1,

$$q_{f_j}(f_j|z_j=k)=\mathcal{N}(f_j|\tilde{m}_{jk},\tilde{v}_{jk}), \tag{F.26}$$

the entropy reads

$$\mathcal{H}(q_{f_j}|z_j=k)=\frac{1}{2}(1+\ln(2\pi))+\frac{1}{2}\ln\left(\tilde{v}_{jk}\right). \tag{F.27}$$

Consequently, removing constant terms, we have

$$\mathcal{H}(q_{f_j}|z_j)\propto\frac{1}{2}\sum_{k=1}^{K}\ln\left(\tilde{v}_{jk}\right)\delta(z_j-k). \tag{F.28}$$

Then, gathering equations (F.22), (F.23), (F.25) and (F.28), we obtain the expression of q_{z_j} :

$$q_{z_j}(z_j) = \frac{\exp\left[\sum_{k=1}^{K}\left(\tilde{\alpha}_{jk} + \gamma_0 \sum_{i\in\mathcal{V}(j)} q_{z_i}(k)\right)\delta(z_j - k)\right]}{\sum_{k=1}^{K}\exp\left[\tilde{\alpha}_{jk} + \gamma_0 \sum_{i\in\mathcal{V}(j)} q_{z_i}(k)\right]} \tag{F.29}$$

where $q_{z_i}(k), \forall i \in \mathcal{V}(j)$, has been computed at the previous of the algorithm, and

$$\begin{aligned}\tilde{\alpha}_{jk} = \alpha_k - \frac{1}{2}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\left[\tilde{v}_{jk} + \tilde{v}_{0_k} + (\tilde{m}_{jk} - \tilde{m}_{0_k})^2\right] + \ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})\right) + \frac{1}{2}\ln(\tilde{v}_{jk}) \\ - \frac{1}{2}\left(\left(\tilde{v}_{jk} + \tilde{m}_{jk}^2\right)\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_{\zeta}^{-1}\boldsymbol{H}\right]_{jj} - 2\tilde{m}_{jk}\left(\tilde{m}_j\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_{\zeta}^{-1}\boldsymbol{H}\right]_{jj} + \left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_{\zeta}^{-1}(\boldsymbol{g} - \boldsymbol{H}\tilde{\boldsymbol{m}})\right]_j\right)\right).\end{aligned} \tag{F.30}$$

F.1.3 Calculation of the approximate distribution for the inverses of the variances of the uncertainties on the projections

Since the inverses of the variances of the uncertainties are completely separated in the chosen form of the approximate joint posterior distribution (F.1), we apply the formula (6.22) of the mean field approximation. According to Bayes's rule,

$$\ln\left(p(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right) \propto \ln\left(p\left(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{\rho}_\zeta\right)\right) + \ln\left(p\left(\boldsymbol{\rho}_\zeta|\alpha_{\zeta_0}, \beta_{\zeta_0}\right)\right). \tag{F.31}$$

where

$$\ln\left(p\left(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{\rho}_\zeta\right)\right) = -\frac{M}{2}\ln(2\pi) + \frac{1}{2}\sum_{i=1}^{M}\ln(\rho_{\zeta_i}) - \frac{1}{2}\sum_{i=1}^{M}\rho_{\zeta_i}\left(g_i - [\boldsymbol{H}\boldsymbol{f}]_i\right)^2 \tag{F.32}$$

and

$$\ln\left(p\left(\boldsymbol{\rho}_\zeta|\alpha_{\zeta_0}, \beta_{\zeta_0}\right)\right) \propto \sum_{i=1}^{M}\left[(\alpha_{\zeta_0} - 1)\ln(\rho_{\zeta_i}) - \beta_{\zeta_0}\rho_{\zeta_i}\right]. \tag{F.33}$$

For fixed $i \in \{1, \ldots, M\}$, we have

$$\mathbb{E}_{q_{/\rho_{\zeta_i}}}\left(\ln\left(p(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\boldsymbol{g}, \mathcal{M})\right)\right) \propto \left(\alpha_{\zeta_0} - \frac{1}{2}\right)\ln(\rho_{\zeta_i}) - \rho_{\zeta_i}\left[\beta_{\zeta_0} + \frac{1}{2}\mathbb{E}_{q_{/\rho_{\zeta_i}}}\left(\left(g_i - [\boldsymbol{H}\boldsymbol{f}]_i\right)^2\right)\right]. \tag{F.34}$$

Using $\tilde{m}_j$ and $\tilde{v}_j^{(2)}$ defined in (F.2), the expectation of the first term in (F.33) is

$$\begin{aligned}\mathbb{E}_{q_{/\rho_{\zeta_i}}}\left(\left(g_i - [\boldsymbol{H}\boldsymbol{f}]_i\right)^2\right) &= g_i^2 - 2g_i\sum_{j=1}^{N}H_{ij}\mathbb{E}_{q_{f_j}q_{z_j}}(f_j) + \sum_{j=1}^{N}H_{ij}^2\mathbb{E}_{q_{f_j}q_{z_j}}\left(f_j^2\right) \\ &\quad + \sum_{\substack{(j,l)\in\{1,\ldots,N\}^2 \\ j\neq l}} H_{ij}H_{il}\mathbb{E}_{q_{f_j}q_{z_j}}(f_j)\mathbb{E}_{q_{f_l}q_{z_l}}(f_l) \\ &= g_i^2 - 2g_i\left[\boldsymbol{H}\tilde{\boldsymbol{m}}\right]_i + \sum_{j=1}^{N}H_{ij}^2\sum_{k=1}^{K}\left(\tilde{v}_{jk} + \tilde{m}_{jk}^2\right)q_{z_j}(k) + \sum_{\substack{(j,l)\in\{1,\ldots,N\}^2 \\ j\neq l}} H_{ij}H_{il}\tilde{m}_j\tilde{m}_l \text{ according to section F.1.1} \\ &= g_i^2 - 2g_i\left[\boldsymbol{H}\tilde{\boldsymbol{m}}\right]_i + \sum_{j=1}^{N}H_{ij}^2\left(\sum_{k=1}^{K}\left(\tilde{v}_{jk} + \tilde{m}_{jk}^2\right)q_{z_j}(k) - \tilde{m}_j^2\right) + \sum_{(j,l)\in\{1,\ldots,N\}^2} H_{ij}H_{il}\tilde{m}_j\tilde{m}_l \\ &= \left(g_i - [\boldsymbol{H}\tilde{\boldsymbol{m}}]_i\right)^2 + \left(\boldsymbol{H}\tilde{\boldsymbol{V}}^{(2)}\boldsymbol{H}^T\right)_{ii} \text{ where } \tilde{\boldsymbol{V}}^{(2)} = \operatorname{diag}\left[\tilde{\boldsymbol{v}}^{(2)}\right].\end{aligned} \tag{F.35}$$

Consequently, $q_{\rho_{\zeta_i}}$ is a Gamma distribution

$$q_{\rho_{\zeta_i}}(\rho_{\zeta_i}) = \mathcal{G}(\rho_{\zeta_i}|\tilde{\alpha}_{\zeta_{0_i}}, \tilde{\beta}_{\zeta_{0_i}}) \tag{F.36}$$

where

$$\begin{cases} \tilde{\alpha}_{\zeta_{0_i}} = \alpha_{\zeta_0} + \frac{1}{2} \\ \tilde{\beta}_{\zeta_{0_i}} = \beta_{\zeta_0} + \frac{1}{2}\left((g_i - [\boldsymbol{H}\tilde{\boldsymbol{m}}]_i)^2 + \left(\boldsymbol{H}\tilde{\boldsymbol{V}}^{(2)}\boldsymbol{H}^T\right)_{ii}\right) \end{cases}, \forall i \in \{1, \ldots, M\}. \tag{F.37}$$

F.1.4 Calculation of the approximate distribution for the means of the classes

For the means of the classes, we also apply the formula (6.22) of the mean field approximation. We have

$$\ln\left(p(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right) \propto \ln\left(p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})\right) + \ln\left(p(\boldsymbol{m}|m_0, v_0)\right) \tag{F.38}$$

where, for fixed $k \in \{1, \ldots, K\}$,

$$\begin{aligned} \mathbb{E}_{q_{/m_k}}\left(p\left(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}\right)\right) &\propto -\frac{1}{2}\mathbb{E}_{q_{\rho_k}}\left(\rho_k\right)\sum_{j=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q_{f_j}}\left((f_j - m_k)^2|z_j = k\right)\mathbb{E}_{q_{z_j}}(\delta(z_j - k)) \\ &\propto -\frac{1}{2}\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\sum_{j=1}^{N}\left(m_k^2 - 2m_k\tilde{m}_{jk}\right)q_{z_j}(k), \end{aligned} \tag{F.39}$$

according to sections F.1.1 and F.1.5, and

$$\mathbb{E}_{q_{/m_k}}\left(\ln\left(p(m_k|m_0, v_0)\right)\right) \propto -\frac{(m_k - m_0)^2}{2v_0}. \tag{F.40}$$

Consequently, q_{m_k} is a Gaussian distribution

$$q_{m_k}(m_k) = \mathcal{N}(m_k|\tilde{m}_{0_k}, \tilde{v}_{0_k}) \tag{F.41}$$

where

$$\begin{cases} \tilde{v}_{0_k} = \left(\frac{1}{v_0} + \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\sum_{j=1}^{N} q_{z_j}(k)\right)^{-1} \\ \tilde{m}_{0_k} = \tilde{v}_{0_k}\left(\frac{m_0}{v_0} + \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\sum_{j=1}^{N}\tilde{m}_{jk}q_{z_j}(k)\right) \end{cases}, \forall k \in \{1, \ldots, K\}. \tag{F.42}$$

F.1.5 Calculation of the approximate distribution for the inverses of the variances of the classes

We also use the updating formula (6.22) of the mean field approximation. Since

$$\ln\left(p(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\boldsymbol{g}; \mathcal{U}, \mathcal{M})\right) \propto \ln\left(p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})\right) + \ln\left(p(\boldsymbol{\rho}|\alpha_0, \beta_0)\right), \tag{F.43}$$

where, according to sections F.1.1 and F.1.4,

$$\begin{aligned} \mathbb{E}_{q_{/\rho_k}}\left(\ln\left(p\left(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}\right)\right)\right) &\propto \frac{1}{2}\ln(\rho_k)\sum_{j=1}^{N}\mathbb{E}_{q_{z_j}}(\delta(z_j - k)) - \frac{\rho_k}{2}\sum_{j=1}^{N}\mathbb{E}_{q_{f_j}q_{m_k}}\left((f_j - m_k)^2|z_j = k\right)\mathbb{E}_{q_{z_j}}(\delta(z_j - k)) \\ &\propto \frac{1}{2}\ln(\rho_k)\sum_{j=1}^{N}q_{z_j}(k) - \frac{\rho_k}{2}\sum_{j=1}^{N}\left(\tilde{v}_{jk} + \tilde{m}_{jk}^2 + \tilde{v}_{0_k} + \tilde{m}_{0_k}^2 - 2\tilde{m}_{jk}\tilde{m}_{0_k}\right)q_{z_j}(k) \\ &\propto \frac{1}{2}\ln(\rho_k)\sum_{j=1}^{N}q_{z_j}(k) - \frac{\rho_k}{2}\sum_{j=1}^{N}\left(\tilde{v}_{0_k} + \tilde{v}_{jk} + (\tilde{m}_{jk} - \tilde{m}_{0_k})^2\right)q_{z_j}(k) \end{aligned} \tag{F.44}$$

and

$$\mathbb{E}_{q_{/\rho_k}}\left(\ln\left(p(\rho_k|\alpha_0, \beta_0)\right)\right) \propto (\alpha_0 - 1)\ln(\rho_k) - \beta_0\rho_k, \tag{F.45}$$

the approximate distribution q_{ρ_k} is a Gamma distribution

$$q_{\rho_k}(\rho_k) = \mathcal{G}(\rho_k|\tilde{\alpha}_{0_k}, \tilde{\beta}_{0_k}) \tag{F.46}$$

where

$$\begin{cases} \tilde{\alpha}_{0_k} = \alpha_0 + \frac{1}{2}\sum_{j=1}^{N} q_{z_j}(k) \\ \tilde{\beta}_{0_k} = \beta_0 + \frac{1}{2}\sum_{j=1}^{N}\left(\tilde{v}_{0_k} + \tilde{v}_{jk} + (\tilde{m}_{jk} - \tilde{m}_{0_k})^2\right) q_{z_j}(k) \end{cases}, \forall k \in \{1, \ldots, K\}. \tag{F.47}$$

F.2 Calculation of the negative free energy

The negative free energy is the stopping criterion of the VBA algorithm. For the approximate joint posterior distribution (F.1), it reads

$$\begin{aligned} \mathcal{F}(q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})) &= \mathcal{H}(q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})) \\ &\quad + \int q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}) \ln\left(p(\boldsymbol{g}; \boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\mathcal{U}, \mathcal{M})\right) d\boldsymbol{f} d\boldsymbol{\rho}_\zeta d\boldsymbol{z} d\boldsymbol{m} d\boldsymbol{\rho} \\ &= \mathcal{H}(q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})) + \mathbb{E}_q\left(\ln\left(p(\boldsymbol{g}; \boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho}|\mathcal{U}, \mathcal{M})\right)\right). \end{aligned} \tag{F.48}$$

In the following, we give the expressions of the two terms in formula (F.48).

F.2.1 Calculation of the entropy of the approximate joint posterior distribution

Given the separations in the approximate joint posterior distribution, we have

$$\mathcal{H}(q(\boldsymbol{f}, \boldsymbol{\rho}_\zeta, \boldsymbol{z}, \boldsymbol{m}, \boldsymbol{\rho})) = \sum_{j=1}^{N}\sum_{k=1}^{K} \mathcal{H}(q_{f_j}|z_j = k) + \sum_{i=1}^{M} \mathcal{H}(q_{\rho_{\zeta_i}}) + \sum_{j=1}^{N} \mathcal{H}(q_{z_j}) + \sum_{k=1}^{K} \mathcal{H}(q_{m_k}) + \sum_{k=1}^{K} \mathcal{H}(q_{\rho_k}), \tag{F.49}$$

where, according to our previous calculations,

$$\sum_{j=1}^{N}\sum_{k=1}^{K} \mathcal{H}(q_{f_j}|z_j = k) = \frac{N}{2}(1 + \ln(2\pi)) + \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K} \ln(\tilde{v}_{jk})\, q_{z_j}(k), \tag{F.50}$$

$$\sum_{j=1}^{N} \mathcal{H}(q_{z_j}) = -\sum_{j=1}^{N}\sum_{k=1}^{K} q_{z_j}(k) \ln\left(q_{z_j}(k)\right), \tag{F.51}$$

$$\sum_{i=1}^{M} \mathcal{H}(q_{\rho_{\zeta_i}}) = \sum_{i=1}^{M}\left[\ln(\Gamma(\tilde{\alpha}_{\zeta_{0_i}})) - \ln(\tilde{\beta}_{\zeta_{0_i}}) + \tilde{\alpha}_{\zeta_{0_i}} - (\tilde{\alpha}_{\zeta_{0_i}} - 1)\psi(\tilde{\alpha}_{\zeta_{0_i}})\right], \tag{F.52}$$

$$\sum_{k=1}^{K} \mathcal{H}(q_{m_k}) = \frac{K}{2}(1 + \ln(2\pi)) + \frac{1}{2}\sum_{k=1}^{K} \ln(\tilde{v}_{0_k}), \tag{F.53}$$

and

$$\sum_{k=1}^{K} \mathcal{H}(q_{\rho_k}) = \sum_{k=1}^{K}\left[\ln(\Gamma(\tilde{\alpha}_{0_k})) - \ln(\tilde{\beta}_{0_k}) + \tilde{\alpha}_{0_k} - (\tilde{\alpha}_{0_k} - 1)\psi(\tilde{\alpha}_{0_k})\right]. \tag{F.54}$$

F.2.2 Calculation of the expectation of the log-joint posterior distribution

According to Bayes' rule

$$\mathbb{E}_q\left(\ln\left(p(\boldsymbol{g};\boldsymbol{f},\boldsymbol{\rho}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho}|\mathcal{U},\mathcal{M})\right)\right) = \mathbb{E}_q\left(\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)\right) + \mathbb{E}_q\left(\ln\left(p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho})\right)\right) + \mathbb{E}_q\left(\ln\left(p(\boldsymbol{\rho}_\zeta|\alpha_{\zeta_0},\beta_{\zeta_0})\right)\right) + \mathbb{E}_q\left(\ln\left(p(\boldsymbol{z}|\boldsymbol{\alpha},\gamma_0)\right)\right) + \mathbb{E}_q\left(\ln\left(p(\boldsymbol{m}|m_0,v_0)\right)\right) + \mathbb{E}_q\left(\ln\left(p(\boldsymbol{\rho}|\alpha_0,\beta_0)\right)\right). \tag{F.55}$$

For the log-likelihood, we have

$$\mathbb{E}_q\left(\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)\right) = -\frac{M}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{M}\left[\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right)\right] - \frac{1}{2}\mathbb{E}_q\left(\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta}\right) \tag{F.56}$$

where

$$\begin{aligned}\mathbb{E}_q\left(\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{V}_\zeta}\right) &= \mathbb{E}_q\left(\sum_{i=1}^{M}\rho_{\zeta_i}\left(g_i-[\boldsymbol{H}\boldsymbol{f}]_i\right)^2\right) = \sum_{i=1}^{M}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}\mathbb{E}_q\left(\left(g_i-[\boldsymbol{H}\boldsymbol{f}]_i\right)^2\right)\\ &= \sum_{i=1}^{M}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}\left[\left(g_i-[\boldsymbol{H}\tilde{\boldsymbol{m}}]_i\right)^2 + \sum_{j=1}^{N}H_{ij}^2\tilde{v}_j^{(2)}\right] \text{ according to section F.1.3}\\ &= \|\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}}\|^2_{\tilde{\boldsymbol{V}}_\zeta} + \sum_{j=1}^{N}\tilde{v}_j^{(2)}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}.\end{aligned} \tag{F.57}$$

Consequently,

$$\begin{aligned}\mathbb{E}_q\left(\ln\left(p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\rho}_\zeta)\right)\right) = &-\frac{M}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{M}\left[\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right)\right]\\ &- \frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}}\|^2_{\tilde{\boldsymbol{V}}_\zeta} - \frac{1}{2}\sum_{j=1}^{N}\tilde{v}_j^{(2)}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj}.\end{aligned} \tag{F.58}$$

According to our previous calculations, the expressions for the other terms in (F.55) are

$$\begin{aligned}\mathbb{E}_q\left(\ln\left(p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{m},\boldsymbol{v})\right)\right) = &-\frac{N}{2}\ln(2\pi)\\ &- \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\left[\tilde{v}_{jk}+\tilde{v}_{0_k}+\left(\tilde{m}_{jk}-\tilde{m}_{0_k}\right)^2\right] + \ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})\right)q_{z_j}(k),\end{aligned} \tag{F.59}$$

$$\begin{aligned}\mathbb{E}_q\left(\ln\left(p(\boldsymbol{\rho}_\zeta|\alpha_{\zeta_0},\beta_{\zeta_0})\right)\right) = &-M\left(\ln(\Gamma(\alpha_{\zeta_0})) - \alpha_{\zeta_0}\ln(\beta_{\zeta_0})\right)\\ &- (\alpha_{\zeta_0}-1)\sum_{i=1}^{M}(\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi(\tilde{\alpha}_{\zeta_{0_i}})) - \beta_{\zeta_0}\sum_{i=1}^{M}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}},\end{aligned} \tag{F.60}$$

$$\mathbb{E}_q\left(\ln\left(p(\boldsymbol{z}|\boldsymbol{\alpha},\gamma_0)\right)\right) = -\ln\left(Z(\boldsymbol{\alpha},\gamma_0)\right) + \sum_{j=1}^{N}\sum_{k=1}^{K}\left(\alpha_k + \gamma_0\sum_{i\in\mathcal{V}(j)}q_{z_i}(k)\right)q_{z_j}(k), \tag{F.61}$$

$$\mathbb{E}_q\left(\ln\left(p(\boldsymbol{m}|m_0,v_0)\right)\right) = -\frac{K}{2}\ln(2\pi) - \frac{K}{2}\ln(v_0) - \frac{1}{2v_0}\sum_{k=1}^{K}\left(\tilde{v}_{0_k}+\left(\tilde{m}_{0_k}-m_0\right)^2\right), \tag{F.62}$$

and

$$\mathbb{E}_q\left(\ln\left(p(\boldsymbol{\rho}|\alpha_0,\beta_0)\right)\right) = -K\left(\ln(\Gamma(\alpha_0)) - \alpha_0\ln(\beta_0)\right) - (\alpha_0-1)\sum_{k=1}^{K}(\ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})) - \beta_0\sum_{k=1}^{K}\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}. \tag{F.63}$$

F.2.3 Expression of the negative free energy

Gathering the results of our calculations in sections F.2.1 and F.2.2, we obtain the expression of the negative free energy

$$
\begin{aligned}
\mathcal{F}(q(\boldsymbol{f},\boldsymbol{\rho}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho})) &= \frac{N}{2}(1+\ln(2\pi)) + \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K}\ln\left(\tilde{v}_{jk}\right)q_{z_j}(k) - \sum_{j=1}^{N}\sum_{k=1}^{K}q_{z_j}(k)\ln\left(q_{z_j}(k)\right) \\
&+ \sum_{i=1}^{M}\left[\ln(\Gamma(\tilde{\alpha}_{\zeta_{0_i}})) - \ln(\tilde{\beta}_{\zeta_{0_i}}) + \tilde{\alpha}_{\zeta_{0_i}} - \left(\tilde{\alpha}_{\zeta_{0_i}}-1\right)\psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right)\right] \\
&+ \frac{K}{2}(1+\ln(2\pi)) + \frac{1}{2}\sum_{k=1}^{K}\ln\left(\tilde{v}_{0_k}\right) + \sum_{k=1}^{K}\left[\ln(\Gamma(\tilde{\alpha}_{0_k})) - \ln(\tilde{\beta}_{0_k}) + \tilde{\alpha}_{0_k} - \left(\tilde{\alpha}_{0_k}-1\right)\psi\left(\tilde{\alpha}_{0_k}\right)\right] \\
&- \frac{M}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{M}\left[\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right)\right] - \frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}}\|^2_{\tilde{\boldsymbol{V}}_\zeta} - \frac{1}{2}\sum_{j=1}^{N}\tilde{v}_j^{(2)}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj} \\
&- \frac{N}{2}\ln(2\pi) - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\left[\tilde{v}_{jk}+\tilde{v}_{0_k}+(\tilde{m}_{jk}-\tilde{m}_{0_k})^2\right] + \ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})\right)q_{z_j}(k) \\
&- \ln\left(Z(\boldsymbol{\alpha},\gamma_0)\right) + \sum_{j=1}^{N}\sum_{k=1}^{K}\left(\alpha_k + \gamma_0\sum_{i\in\mathcal{V}(j)}q_{z_i}(k)\right)q_{z_j}(k) \\
&- M\left(\ln(\Gamma(\alpha_{\zeta_0})) - \alpha_{\zeta_0}\ln(\beta_{\zeta_0})\right) - (\alpha_{\zeta_0}-1)\sum_{i=1}^{M}(\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi(\tilde{\alpha}_{\zeta_{0_i}})) - \beta_{\zeta_0}\sum_{i=1}^{M}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}} \\
&- \frac{K}{2}\ln(2\pi v_0) - \frac{1}{2v_0}\sum_{k=1}^{K}\left(\tilde{v}_{0_k} + (\tilde{m}_{0_k}-m_0)^2\right) \\
&- K\left(\ln(\Gamma(\alpha_0)) - \alpha_0\ln(\beta_0)\right) - (\alpha_0-1)\sum_{k=1}^{K}(\ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})) - \beta_0\sum_{k=1}^{K}\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}. \qquad \text{(F.64)}
\end{aligned}
$$

Dropping the constant terms, the expression of the stopping criterion of the VBA algorithm is

$$
\begin{aligned}
\mathcal{L}(q(\boldsymbol{f},\boldsymbol{\rho}_\zeta,\boldsymbol{z},\boldsymbol{m},\boldsymbol{\rho})) &= \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K}\ln\left(\tilde{v}_{jk}\right)q_{z_j}(k) - \sum_{j=1}^{N}\sum_{k=1}^{K}q_{z_j}(k)\ln\left(q_{z_j}(k)\right) \\
&+ \sum_{i=1}^{M}\left[\ln(\Gamma(\tilde{\alpha}_{\zeta_{0_i}})) - \ln(\tilde{\beta}_{\zeta_{0_i}}) + \tilde{\alpha}_{\zeta_{0_i}} - \left(\tilde{\alpha}_{\zeta_{0_i}}-1\right)\psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right)\right] \\
&+ \frac{1}{2}\sum_{k=1}^{K}\ln\left(\tilde{v}_{0_k}\right) + \sum_{k=1}^{K}\left[\ln(\Gamma(\tilde{\alpha}_{0_k})) - \ln(\tilde{\beta}_{0_k}) + \tilde{\alpha}_{0_k} - \left(\tilde{\alpha}_{0_k}-1\right)\psi\left(\tilde{\alpha}_{0_k}\right)\right] \\
&- \frac{1}{2}\sum_{i=1}^{M}\left[\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi\left(\tilde{\alpha}_{\zeta_{0_i}}\right)\right] - \frac{1}{2}\|\boldsymbol{g}-\boldsymbol{H}\tilde{\boldsymbol{m}}\|^2_{\tilde{\boldsymbol{V}}_\zeta} - \frac{1}{2}\sum_{j=1}^{N}\tilde{v}_j^{(2)}\left[\boldsymbol{H}^T\tilde{\boldsymbol{V}}_\zeta^{-1}\boldsymbol{H}\right]_{jj} \\
&- \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}\left[\tilde{v}_{jk}+\tilde{v}_{0_k}+(\tilde{m}_{jk}-\tilde{m}_{0_k})^2\right] + \ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})\right)q_{z_j}(k) \\
&+ \sum_{j=1}^{N}\sum_{k=1}^{K}\left(\alpha_k + \gamma_0\sum_{i\in\mathcal{V}(j)}q_{z_i}(k)\right)q_{z_j}(k) \\
&- (\alpha_{\zeta_0}-1)\sum_{i=1}^{M}(\ln(\tilde{\beta}_{\zeta_{0_i}}) - \psi(\tilde{\alpha}_{\zeta_{0_i}})) - \beta_{\zeta_0}\sum_{i=1}^{M}\frac{\tilde{\alpha}_{\zeta_{0_i}}}{\tilde{\beta}_{\zeta_{0_i}}}
\end{aligned}
$$

$$- \frac{1}{2v_0} \sum_{k=1}^{K} \left(\tilde{v}_{0_k} + (\tilde{m}_{0_k} - m_0)^2 \right) - (\alpha_0 - 1) \sum_{k=1}^{K} (\ln(\tilde{\beta}_{0_k}) - \psi(\tilde{\alpha}_{0_k})) - \beta_0 \sum_{k=1}^{K} \frac{\tilde{\alpha}_{0_k}}{\tilde{\beta}_{0_k}}. \quad \text{(F.65)}$$

Bibliography

[ADMD10] Hacheme Ayasso, Bernard Duchêne, and Ali Mohammad-Djafari. Bayesian inversion for optical diffraction tomography. *Journal of Modern Optics*, 57(9):765–776, 2010.

[ADMD12] Hacheme Ayasso, Bernard Duchêne, and Ali Mohammad-Djafari. Optical diffraction tomography within a variational Bayesian framework. *Inverse Problems in Science and Engineering*, 20(1):59–73, 2012.

[AK84] Anders H Andersen and Avinash C Kak. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. *Ultrasonic imaging*, 6(1):81–94, 1984.

[AMD10] Hacheme Ayasso and Ali Mohammad-Djafari. Joint NDT image restoration and segmentation using Gauss–Markov–Potts prior models and variational bayesian computation. *IEEE Transactions on Image Processing*, 19(9):2265–2277, 2010.

[ASM16] Filippo Arcadu, Marco Stampanoni, and Federica Marone. On the crucial impact of the coupling projector-backprojector in iterative tomographic reconstruction. *arXiv preprint arXiv:1612.05515*, 2016.

[Aya10] Hacheme Ayasso. *Une approche bayésienne de l'inversion. Application à l'imagerie de diffraction dans les domaines micro-onde et optique*. PhD thesis, Université Paris Sud-Paris XI, 2010.

[BDC76] Rodney A Brooks and Giovanni Di Chiro. Beam hardening in X-ray reconstructive tomography. *Physics in medicine & biology*, 21(3):390, 1976.

[Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.

[Bes86] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[BGMD+15] Thomas Boulay, Nicolas Gac, Ali Mohammad-Djafari, Estelle Parra-Denis, and Yann Le Guilloux. TomoBayes v1.0 - logiciel de reconstruction en tomographie CT - Ref CNRS du pré-dépôt APP 11562-03 (num IDDN prochainement disponible), December 2015.

[BL06] Jonathan Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization : theory and examples*. Springer Science & Business Media, 2006.

[Ble15] Folkert Bleichrodt. *Improving robustness of tomographic reconstruction methods*. PhD thesis, Mathematical Institute, Faculty of Science, Leiden University, 2015.

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[BS11] Kees Joost Batenburg and Jan Sijbers. DART: a practical reconstruction algorithm for discrete tomography. *Image Processing, IEEE Transactions on*, 20(9):2542–2553, 2011.

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[CD15] Antonin Chambolle and Ch Dossal. On the convergence of the iterates of the "Fast Iterative Shrinkage/Thresholding Algorithm". *Journal of Optimization theory and Applications*, 166(3):968–982, 2015.

[CGMDP18] Camille Chapdelaine, Nicolas Gac, Ali Mohammad-Djafari, and Estelle Parra. New GPU implementation of Separable Footprint Projector and Backprojector : first results. In *The 5th International Conference on Image Formation in X-Ray Computed Tomography*, pages 314–317, 2018.

[CMDGP17] Camille Chapdelaine, Ali Mohammad-Djafari, Nicolas Gac, and Estelle Parra. A 3D Bayesian Computed Tomography Reconstruction Algorithm with Gauss-Markov-Potts Prior Model and its Application to Real Data. *Fundamenta Informaticae*, 155(4):373–405, 2017.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[CXLF17] Il Yong Chun, Zheng Xuehang, Yong Long, and Jeffrey A. Fessler. Sparse View X-Ray CT Reconstruction Using l_1 Regularization with Learned Sparsifying Transforms. In *The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2017.

[DGWMD17] Mircea Dumitru, Nicolas Gac, Li Wang, and Ali Mohammad-Djafari. Unsupervised sparsity enforcing iterative algorithms for 3D image reconstruction in X-ray Computed Tomography. In *The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pages 359–363, 2017.

[DH17] Jonas Dittmann and Randolf Hanke. Simple and efficient raycasting on modern GPU's read-and-write memory for fast forward projections in iterative CBCT reconstruction. In *The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2017.

[DMB02] Bruno De Man and S Basu. Distance-driven projection and backprojection. In *Nuclear Science Symposium Conference Record, 2002 IEEE*, volume 3, pages 1477–1480. IEEE, 2002.

[DMB04] Bruno De Man and Samit Basu. Distance-driven projection and backprojection in three dimensions. *Physics in medicine and biology*, 49(11):2463, 2004.

[Dum16] Mircea Dumitru. *A Bayesian Approach for Periodic Components Estimation for Chronobiological Signals*. PhD thesis, Université Paris-Saclay, 2016.

[DYXW17] Yi Du, Gongyi Yu, Xincheng Xiang, and Xiangang Wang. GPU accelerated voxel-driven forward projection for iterative reconstruction of cone-beam CT. *Biomedical engineering online*, 16(1):2, 2017.

[EF99] Hakan Erdogan and Jeffrey A Fessler. Ordered subsets algorithms for transmission tomography. *Physics in medicine and biology*, 44(11):2835, 1999.

[Ess09] Ernie Esser. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *CAM report*, 9:31, 2009.

[FDK84] LA Feldkamp, LC Davis, and JW Kress. Practical cone-beam algorithm. *JOSA A*, 1(6):612–619, 1984.

[FDMD05] Olivier Féron, Bernard Duchêne, and Ali Mohammad-Djafari. Microwave imaging of inhomogeneous objects made of a finite number of dielectric and conductive materials from experimental data. *Inverse Problems*, 21(6):95–115, 2005.

[Fes96] Jeffrey A Fessler. Mean and Variance of Implicitly Defined Biased Estimators (such as Penalized Maximum Likelihood): Applications to Tomography. *IEEE Transactions on Image Processing*, 5(3):493–506, 1996.

[Fes00] Jeffrey A Fessler. Statistical image reconstruction methods for transmission tomography. *Handbook of medical imaging*, 2:1–70, 2000.

[FR96] Jeffrey A Fessler and W Leslie Rogers. Spatial resolution properties of penalized-likelihood image reconstruction: space-invariant tomographs. *IEEE Transactions on Image processing*, 5(9):1346–1358, 1996.

[FSVMD07] Sofia Fekih Salem, Alexandre Vabre, and Ali Mohammad-Djafari. Bayesian tomographic reconstruction of microsystems. In *AIP Conference Proceedings*, volume 954, pages 372–380. AIP, 2007.

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[GARC16] Lionel Gay, Philippe Arslan, Yves Rambourg, and André Chandelle. Fantôme destiné à être utilisé pour le contrôle de la qualité d'images tomographiques, June 2016.

[GBH70] Richard Gordon, Robert Bender, and Gabor T Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of theoretical Biology*, 29(3):471–481, 1970.

[GBJ15] Ryan J Giordano, Tamara Broderick, and Michael I Jordan. Linear response methods for accurate covariance estimates from Mean Field Variational Bayes. In *Advances in Neural Information Processing Systems*, pages 1441–1449, 2015.

[GDMJ+16] Lars Gjesteby, Bruno De Man, Yannan Jin, Harald Paganetti, Joost Verburg, Drosoula Giantsoudi, and Ge Wang. Metal artifact reduction in CT: Where are we after four decades ? *IEEE Access*, 4:5826–5849, 2016.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[Gil72] Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of theoretical biology*, 36(1):105–117, 1972.

[Gio10] Jean-François Giovannelli. Estimation of the Ising field parameter thanks to the exact partition function. In *ICIP*, pages 1441–1444, 2010.

[GO09] Tom Goldstein and Stanley Osher. The split Bregman method for L1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

[GSS82] Leslie M Goldschlager, Ralph A Shaw, and John Staples. The maximum flow problem is log space complete for p. *Theoretical Computer Science*, 21(1):105–111, 1982.

[GTSJ15] Euhanna Ghadimi, André Teixeira, Iman Shames, and Mikael Johansson. Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2015.

[GV17] Jean-François Giovannelli and Cornelia Vacar. Deconvolution-Segmentation for Textured Images. In *25th European Signal Processing Conference (EUSIPCO)*, pages 201–205, 2017.

[HD92] Y Hu and Tim J Dennis. Simulated annealing and iterated conditional modes with selective and confidence enhanced update schemes. In *Computer-Based Medical Systems, 1992. Proceedings., Fifth Annual IEEE Symposium on*, pages 257–264. IEEE, 1992.

[HKK+97] Karsten Held, E Rota Kops, Bernd J Krause, William M Wells, Ron Kikinis, and H-W Muller-Gartner. Markov random field segmentation of brain mr images. *IEEE transactions on medical imaging*, 16(6):878–886, 1997.

[HL94] H Malcolm Hudson and Richard S Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE transactions on medical imaging*, 13(4):601–609, 1994.

[HLN76] GT Herman, AV Lakshminarayanan, and A Naparstek. Convolution reconstruction techniques for divergent beams. *Computers in biology and medicine*, 6(4):259–271, 1976.

[HLY99] Guoping Han, Zhengrong Liang, and Jiangsheng You. A fast ray-tracing technique for TCT and ECT studies. In *Nuclear Science Symposium, 1999. Conference Record. 1999 IEEE*, volume 3, pages 1515–1518. IEEE, 1999.

[Hua87] Kerson Huang. *Statistical mechanics*. Wiley, 1987.

[Idi08] Jérôme Idier. *Bayesian approach to inverse problems*. John Wiley & Sons, 2008.

[JBS15] Pengchong Jin, Charles A Bouman, and Ken D Sauer. A Model-Based Image Reconstruction Algorithm With Simultaneous Beam Hardening Correction for X-Ray CT. *IEEE Trans. Computational Imaging*, 1(3):200–216, 2015.

[Jef46] Harold Jeffreys. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London a: mathematical, physical and engineering sciences*, volume 186, pages 453–461. The Royal Society, 1946.

[JMNM91] John I Jackson, Craig H Meyer, Dwight G Nishimura, and Albert Macovski. Selection of a convolution function for Fourier inversion using gridding [computerized tomography application]. *Medical Imaging, IEEE Transactions on*, 10(3):473–478, 1991.

[Jos82] Peter M Joseph. An improved algorithm for reprojecting rays through pixel images. *IEEE transactions on medical imaging*, 1(3):192–196, 1982.

[JSDS+98] Filip Jacobs, Erik Sundermann, Bjorn De Sutter, Mark Christiaens, and Ignace Lemahieu. A fast algorithm to calculate the exact radiological path through a pixel or voxel space. *CIT. Journal of computing and information technology*, 6(1):89–94, 1998.

[KF17] Donghwan Kim and Jeffrey A Fessler. Accelerated dual gradient-based methods for total variation image denoising/deblurring problems. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6230–6234. IEEE, 2017.

[KKF08] M Krumm, S Kasperl, and M Franz. Reducing non-linear artifacts of multi-material objects in industrial 3D Computed Tomography. *Ndt & E International*, 41(4):242–251, 2008.

[KLH+08] Y Kyriakou, RM Lapp, L Hillebrand, D Ertel, and WA Kalender. Simultaneous misalignment correction for approximate circular cone-beam computed tomography. *Physics in Medicine & Biology*, 53(22):6267, 2008.

[KNF76] Warren L. G. Koontz, Patrenahalli M. Narendra, and Keinosuke Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Transactions on Computers*, 100(9):936–944, 1976.

[KPTF13] Donghwan Kim, Debashish Pal, Jean-Baptiste Thibault, and Jeffrey A Fessler. Accelerating ordered subsets image reconstruction for X-ray CT using spatially nonuniform optimization transfer. *IEEE transactions on medical imaging*, 32(11):1965–1978, 2013.

[KRF15] Donghwan Kim, Sathish Ramani, and Jeffrey A Fessler. Combining ordered subsets and momentum for accelerated X-ray CT image reconstruction. *IEEE transactions on medical imaging*, 34(1):167–178, 2015.

[KSH01] Fatih Kurugollu, Bülent Sankur, and A Emre Harmanci. Color image segmentation using histogram multithresholding and fusion. *Image and vision computing*, 19(13):915–928, 2001.

[KSK06] Marc Kachelrieß, Katia Sourbelle, and Willi A Kalender. Empirical cupping correction: A first-order raw data precorrection for cone-beam computed tomography. *Medical physics*, 33(5):1269–1274, 2006.

[LC+84] Kenneth Lange, Richard Carson, et al. EM reconstruction algorithms for emission and transmission tomography. *J Comput Assist Tomogr*, 8(2):306–16, 1984.

[LFB10] Yong Long, Jeffrey A Fessler, and James M Balter. 3D forward and back-projection for X-ray CT using separable footprints. *IEEE transactions on medical imaging*, 29(11):1839–1850, 2010.

[LFDMY17] Rui Liu, Lin Fu, Bruno De Man, and Hengyong Yu. GPU-based Branchless Distance-Driven Projection and Backprojection. *IEEE Transactions on Computational Imaging*, 3(4):617–632, 2017.

[Lon11] Yong Long. *Statistical Image Reconstruction and Motion Estimation for Image-Guided Radiotherapy*. PhD thesis, University of Michigan, 2011.

[Mac67] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[MBBCP14] Yosra Marnissi, Amel Benazza-Benyahia, Emilie Chouzenoux, and J-C Pesquet. Majorize-Minimize adapted Metropolis-Hastings algorithm. Application to multichannel image recovery. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 1332–1336. IEEE, 2014.

[MDA09] Ali Mohammad-Djafari and Hacheme Ayasso. Variational Bayes and mean field approximations for Markov field unsupervised estimation. In *2009 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2009.

[MDD88] Ali Mohammad-Djafari and Guy Demoment. Maximum entropy image reconstruction in X-ray and diffraction tomography. *Medical Imaging, IEEE Transactions on*, 7(4):345–354, 1988.

[MDZ97] Robin Morris, Xavier Descombes, and Josiane Zerubia. Fully Bayesian image segmentation - An engineering perspective. In *Image Processing, 1997. Proceedings., International Conference on*, volume 3, pages 54–57. IEEE, 1997.

[Mer76] Russell M Mersereau. Direct Fourier transform techniques in 3-D image reconstruction. *Computers in biology and medicine*, 6(4):247–IN4, 1976.

[MF15] Madison G McGaffin and Jeffrey A Fessler. Alternating dual updates algorithm for X-ray CT reconstruction on the GPU. *IEEE transactions on computational imaging*, 1(3):186–199, 2015.

[MFDA18] C. Martinez, J. A. Fessler, M. Desco, and M. Abella. Statistical Image Reconstruction with Sample-Based Beam-Hardening compensation for X-ray CT. In *The Fifth International Conference on Image Formation in X-Ray Computed Tomography*, pages 11–14, 2018.

[Min01] Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[Min05] Thomas P Minka. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

[MWF18] Harini Muthukrishnan, Thomas F. Wenisch, and Jeffrey A. Fessler. Improving GPU Scaling for X-ray CT. In *The Fifth international conference on image formation in X-ray Computed Tomography*, 2018.

[MZCP17] Yosra Marnissi, Yuling Zheng, Emilie Chouzenoux, and Jean-Christophe Pesquet. A variational bayesian approach for image restoration. application to image deblurring with poisson-gaussian noise. *IEEE Transactions on Computational Imaging*, 2017.

[NDMF+13] Johan Nuyts, Bruno De Man, Jeffrey A Fessler, Wojciech Zbijewski, and Freek J Beekman. Modelling the physics in the iterative reconstruction for transmission computed tomography. *Physics in medicine and biology*, 58(12):R63, 2013.

[Nes83] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In *Doklady AN USSR*, volume 269, pages 543–547, 1983.

[NHBF16] Thibault Notargiacomo, Dominique Houzet, Guillaume Bernard, and Vincent Fristot. Sparse Regularization of CBCT Reconstruction Using 3D Dual-Tree Complex Wavelet Transform and Dictionary Learning Techniques. In *The 4th International Conference on Image Formation in X-Ray Computed Tomography*, 2016.

[Nik02] Mila Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002.

[NL15] Van-Giang Nguyen and Soo-Jin Lee. Parallelizing a matched pair of ray-tracing projector and backprojector for iterative cone-beam CT reconstruction. *IEEE Transactions on Nuclear Science*, 62(1):171–181, 2015.

[NNZC08] Mila Nikolova, Michael K Ng, Shuqin Zhang, and Wai-Ki Ching. Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization. *SIAM journal on Imaging Sciences*, 1(1):2–25, 2008.

[NVI18] NVIDIA. *CUDA C Programming Guide*, October 2018.

[OMBF18] Greg Ongie, Naveen Murthy, Laura Balzano, and Jeffrey A. Fessler. A Memory-Efficient Algorithm for Large-Scale Sparsity Regularized Image Reconstruction. In *The Fifth International Conference on Image Formation in X-Ray Computed Tomography*, pages 20–23, 2018.

[Ons44] Lars Onsager. Crystal statistics. I. a two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117, 1944.

[Ots79] N Otsu. Thresholds selection method form grey-level histograms. *IEEE Trans. On Systems, Man and Cybernetics*, 9(1):1979, 1979.

[Pal17] Pierre Paleo. *Iterative Methods in regularized tomographic reconstruction*. PhD thesis, Université Grenoble Alpes, 2017.

[PDBT13] Marcelo Pereyra, Nicolas Dobigeon, Hadj Batatia, and Jean-Yves Tourneret. Estimating the granularity coefficient of a Potts-Markov random field within a Markov chain Monte Carlo algorithm. *Image Processing, IEEE Transactions on*, 22(6):2385–2397, 2013.

[PDM17] Pierre Paleo, Michel Desvignes, and Alessandro Mirone. A practical local tomography reconstruction algorithm based on a known sub-region. *Journal of synchrotron radiation*, 24(1):257–268, 2017.

[Per17] Marcelo Pereyra. Maximum-A-Posteriori estimation with Bayesian Confidence Regions. *SIAM Journal on Imaging Sciences*, 10(1):285–302, 2017.

[PK83] S Pan and A Kak. A computational study of reconstruction algorithms for diffraction tomography: Interpolation versus filtered-backpropagation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(5):1262–1275, 1983.

[PM17] Pierre Paleo and Alessandro Mirone. Efficient implementation of a local tomography reconstruction algorithm. *Advanced structural and chemical imaging*, 3(1):5, 2017.

[PSC+16] Marcelo Pereyra, Philip Schniter, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Yves Tourneret, Alfred O Hero, and Steve McLaughlin. A survey of stochastic simulation and optimization methods in signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):224–241, 2016.

[PSL14] Hyeong-Gyu Park, Yeong-Gil Shin, and Ho Lee. A fully GPU-based ray-driven backprojector via a ray-culling scheme with voxel-level parallelization for cone-beam CT reconstruction. *Technology in cancer research & treatment*, pages tcrt–2012, 2014.

[QJ07] Yuan Qi and Tommi S Jaakkola. Parameter expanded variational Bayesian methods. *Advances in Neural Information Processing Systems*, 19:1097, 2007.

[QJ16] Tran Minh Quan and Won-Ki Jeong. A fast discrete wavelet transform using hybrid parallelism on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 27(11):3088–3100, 2016.

[Rad17] Johann Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Classic papers in modern diagnostic radiology*, 5, 1917.

[Ran71] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[Rod02] Thomas Rodet. *Algorithmes rapides de reconstruction en tomographie par compression des calculs: application à la tomofluoroscopie 3D*. PhD thesis, Grenoble, INPG, 2002.

[RR07] Ronny Ramlau and Wolfgang Ring. A Mumford–Shah level-set approach for the inversion and segmentation of X-ray tomography data. *Journal of Computational Physics*, 221(2):539–557, 2007.

[RRT+18] Aymeric Reshef, Cyril Riddell, Yves Trousset, Saïd Ladjal, and Isabelle Bloch. Divergent-beam backprojection-filtration formula with applications to region-of-interest imaging. In *The Fifth International Conference on Image Formation in X-Ray Computed Tomography*, 2018.

[SB93] Ken Sauer and Charles Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing*, 41(2):534–548, 1993.

[SBS18] Nathanaël Six, Jan De Beenhouwer, and Jan Sijbers. pDART : Discrete algebraic reconstruction using a polychromatic forward model. In *The Fifth International Conference on Image Formation in X-Ray Computed Tomography*, 2018.

[She94] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[SHH07] Yi Sun, Ying Hou, and Jiasheng Hu. Reduction of artifacts induced by misaligned geometry in cone-beam CT. *IEEE transactions on biomedical engineering*, 54(8):1461–1471, 2007.

[Sid85] Robert L Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical physics*, 12(2):252–255, 1985.

[SJP+12] Emil Y Sidky, HJ Jakob, Xiaochuan Pan, et al. Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle & Pock algorithm. *Physics in medicine and biology*, 57(10):3065, 2012.

[SK85] Kennan T Smith and F Keinert. Mathematical foundations of computed tomography. *Applied Optics*, 24(23):3950–3957, 1985.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming, Portable Documents*. Addison-Wesley Professional, 2010.

[SL74] Lawrence A Shepp and Benjamin F Logan. The Fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, 21(3):21–43, 1974.

[ŠQ06] Václav Šmídl and Anthony Quinn. *The variational Bayes method in signal processing*. Springer Science & Business Media, 2006.

[SRUW17] Martin Storath, Dennis Rickert, Michael Unser, and Andreas Weinmann. Fast segmentation from blurred data in 3D fluorescence microscopy. *IEEE Transactions on Image Processing*, 26(10):4856–4870, 2017.

[SSHF16] Sanming Song, Bailu Si, J Michael Herrmann, and Xisheng Feng. Local Autoencoding for Parameter Estimation in a Hidden Potts-Markov Random Field. *IEEE Transactions on Image Processing*, 25(5):2324–2336, 2016.

[SWFU15] Martin Storath, Andreas Weinmann, Jürgen Frikel, and Michael Unser. Joint image reconstruction and segmentation using the Potts model. *Inverse Problems*, 31(2):025003, 2015.

[SWPH81a] Henry Stark, John Woods, Indraneel Paul, and Rajesh Hingorani. Direct Fourier reconstruction in computer tomography. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):237–245, 1981.

[SWPH81b] Henry Stark, John W Woods, Indraneel Paul, and Rajesh Hingorani. An investigation of computerized tomography by direct Fourier inversion and optimum interpolation. *IEEE Transactions on Biomedical Engineering*, (7):496–505, 1981.

[TBSH06] Jean-Baptiste Thibault, Charles A Bouman, Ken D Sauer, and Jiang Hsieh. A recursive filter for noise reduction in statistical iterative tomographic imaging. In *Electronic Imaging 2006*, pages 60650X–60650X. International Society for Optics and Photonics, 2006.

[Tik63] Andrey Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, volume 5, pages 1035–1038, 1963.

[TSBH07] Jean-Baptiste Thibault, Ken D Sauer, Charles A Bouman, and Jiang Hsieh. A three-dimensional statistical approach to improved image quality for multislice helical CT. *Medical physics*, 34(11):4526–4544, 2007.

[Tuy83] Heang K Tuy. An inversion formula for cone-beam reconstruction. *SIAM Journal on Applied Mathematics*, 43(3):546–552, 1983.

[VGVH+13] Bert Vandeghinste, Bart Goossens, Roel Van Holen, Christian Vanhove, Aleksandra Pižurica, Stefaan Vandenberghe, and Steven Staelens. Iterative CT reconstruction using shearlet-based regularization. *IEEE Transactions on Nuclear Science*, 60(5):3305–3317, 2013.

[WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[WF11] Meng Wu and Jeffrey A Fessler. GPU acceleration of 3D forward and backward projection using separable footprints for X-ray CT image reconstruction. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, volume 6, page 021911. Citeseer, 2011.

[WMDG17] Li Wang, Ali Mohammad-Djafari, and Nicolas Gac. X-ray Computed Tomography using a sparsity enforcing prior model based on Haar transformation in a Bayesian framework. *Fundamenta Informaticae*, 155(4):449–480, 2017.

[WSRP16] L Wang, B Sixou, S Rit, and F Peyrin. Binary tomography reconstruction from few projections with Total Variation regularization for bone microstructure studies. *Journal of X-ray science and technology*, 24(2):177–189, 2016.

[WWH05] Yuchuan Wei, Ge Wang, and Jiang Hsieh. An intuitive discussion on the ideal ramp filter in computed tomography (I). *Computers & Mathematics with Applications*, 49(5-6):731–740, 2005.

[XML+17] Xiaobin Xie, Madison G McGaffin, Yong Long, Jeffrey A Fessler, Minhua Wen, and James Lin. Accelerating separable footprint (SF) forward and back projection on GPU. In *SPIE Medical Imaging*, pages 101322S–101322S. International Society for Optics and Photonics, 2017.

[XYM+12] Qiong Xu, Hengyong Yu, Xuanqin Mou, Lei Zhang, Jiang Hsieh, and Ge Wang. Low-dose X-ray CT reconstruction via dictionary learning. *IEEE Transactions on Medical Imaging*, 31(9):1682–1697, 2012.

[YFCL17] Linxiao Yang, Jun Fang, Hong Cheng, and Hongbin Li. Sparse Bayesian dictionary learning with a Gaussian hierarchical model. *Signal Processing*, 130:93–104, 2017.

[ZBKT16] Ningning Zhao, Adrian Basarab, Denis Kouame, and Jean-Yves Tourneret. Joint segmentation and deconvolution of ultrasound images using a hierarchical Bayesian model based on generalized Gaussian priors. *IEEE transactions on Image Processing*, 25(8):3736–3750, 2016.

[ZBS01] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.

[ZC08] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 34, 2008.

[ZG00] Gengsheng L Zeng and Grant T Gullberg. Unmatched projector/backprojector pairs in an iterative reconstruction algorithm. *IEEE transactions on medical imaging*, 19(5):548–555, 2000.

[Zha92] Jun Zhang. The Mean Field theory in EM procedures for Markov random fields. *IEEE Transactions on signal processing*, 40(10):2570–2583, 1992.

[ZKS+16] Matthias Zisler, Jörg Hendrik Kappes, Claudius Schnörr, Stefania Petra, and Christoph Schnörr. Non-binary discrete tomography by continuous non-convex optimization. *IEEE Transactions on Computational Imaging*, 2(3):335–347, 2016.

[ZNG08] Andy Ziegler, Tim Nielsen, and Michael Grass. Iterative reconstruction of a region of interest for transmission tomography. *Medical physics*, 35(4):1317–1327, 2008.

[ZPB16] Xiaodong Zhuge, Willem Jan Palenstijn, and Kees Joost Batenburg. TVR-DART: a more robust algorithm for discrete tomography from limited projection data with automated gray value estimation. *IEEE Transactions on Image Processing*, 25(1):455–468, 2016.

[ZRLF18] Xuehang Zheng, Saiprasad Ravishankar, Yong Long, and Jeffrey A Fessler. PWLS-ULTRA: An efficient clustering and learning-based approach for low-dose 3D CT image reconstruction. *IEEE transactions on medical imaging*, 37(6):1498–1510, 2018.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Méthodes de reconstruction iteratives en tomographie 3D par rayons X via une approche bayésienne

Mots clés : Tomographie à rayons X, reconstruction 3D, inférence bayésienne, calcul parallèle, GPU

Résumé : Dans un contexte industriel, la tomographie 3D par rayons X vise à imager virtuellement une pièce afin d'en contrôler l'intérieur. Le volume virtuel de la pièce est obtenu par un algorithme de reconstruction, prenant en entrées les projections de rayons X qui ont été envoyés à travers la pièce. Beaucoup d'incertitudes résident dans ces projections à cause de phénomènes non contrôlés tels que la diffusion et le durcissement de faisceau, causes d'artefacts dans les reconstructions conventionnelles par rétroprojection filtrée. Afin de compenser ces incertitudes, les méthodes de reconstruction dites itératives tentent de faire correspondre la reconstruction à un modèle a priori, ce qui, combiné à l'information apportée par les projections, permet d'améliorer la qualité de reconstruction. Dans ce contexte, cette thèse propose de nouvelles méthodes de reconstruction itératives pour le contrôle de pièces produites par le groupe SAFRAN. Compte tenu de nombreuses opérations de projection et de rétroprojection modélisant le processus d'acquisition, les méthodes de reconstruction itératives peuvent être accélérées grâce au calcul parallèle haute performance sur processeur graphique (GPU). Dans cette thèse, les implémentations sur GPU de plusieurs paires de projecteur-rétroprojecteur sont décrites. En particulier, une nouvelle implémentation pour la paire duale dite à empreinte séparable est proposée. Beaucoup de pièces produites par SAFRAN pouvant être vues comme des volumes constants par morceaux, un modèle a priori de Gauss-Markov-Potts est introduit, à partir duquel est déduit un algorithme de reconstruction et de segmentation conjointes. Cet algorithme repose sur une approche bayésienne permettant d'expliquer le rôle de chacun des paramètres. Le caractère polychromatique des rayons X par lequel s'expliquent la diffusion et le durcissement de faisceau est pris en compte par l'introduction d'un modèle direct séparant les incertitudes sur les projections. Allié à un modèle de Gauss-Markov-Potts sur le volume, il est montré expérimentalement que ce nouveau modèle direct apporte un gain en précision et en robustesse. Enfin, l'estimation des incertitudes sur la reconstruction est traitée via l'approche bayésienne variationnelle. Pour obtenir cette estimation en un temps de calcul raisonnable, il est montré qu'il est nécessaire d'utiliser une paire duale de projecteur-rétroprojecteur.

Title : Bayesian iterative reconstruction methods for 3D X-ray Computed Tomography

Keywords : X-ray Computed Tomography, 3D reconstruction, Bayesian inference, parallel computing, GPU

Abstract : In industry, 3D X-ray Computed Tomography aims at virtually imaging a volume in order to inspect its interior. The virtual volume is obtained thanks to a reconstruction algorithm based on projections of X-rays sent through the industrial part to inspect. In order to compensate uncertainties in the projections such as scattering or beam-hardening, which are cause of many artifacts in conventional filtered backprojection methods, iterative reconstruction methods bring further information by enforcing a prior model on the volume to reconstruct, and actually enhance the reconstruction quality. In this context, this thesis proposes new iterative reconstruction methods for the inspection of aeronautical parts made by SAFRAN group. In order to alleviate the computational cost due to repeated projection and backprojection operations which model the acquisition process, iterative reconstruction methods can take benefit from the use of high-parallel computing on Graphical Processor Unit (GPU). In this thesis, the implementation on GPU of several pairs of projector and backprojector is detailed. In particular, a new GPU implementation of the matched Separable Footprint pair is proposed. Since many of SAFRAN's industrial parts are piecewise-constant volumes, a Gauss-Markov-Potts prior model is introduced, from which a joint reconstruction and segmentation algorithm is derived. This algorithm is based on a Bayesian approach which enables to explain the role of each parameter. The actual polychromacy of X-rays, which is responsible for scattering and beam-hardening, is taken into account by proposing an error-splitting forward model. Combined with Gauss-Markov-Potts prior on the volume, this new forward model is experimentally shown to bring more accuracy and robustness. At last, the estimation of the uncertainties on the reconstruction is investigated by variational Bayesian approach. In order to have a reasonable computation time, it is highlighted that the use of a matched pair of projector and backprojector is necessary.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France