

POLITECNICO DI MILANO
SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE
Corso di Laurea Magistrale in Ingegneria Biomedica



Recovery of Contrast Transit Dynamics in CT Brain Angiography for Arteries and Veins Segmentation

Relatori:

Prof. Giuseppe Baselli

Dott. Francesco Cardinale

Dott.ssa Elena De Momi

Tesi di laurea di:

Beatrice Barra

Matr. 817380

Anno Accademico 2014 - 2015

Contents

List of Figures	vi
List of Tables	vii
Abstract	ix
Sommario	xiii
1 Introduction	1
1.1 Vessel Tracking in Neurosurgery	2
1.2 Digital Angiography	3
1.2.1 Temporal Subtraction with Digital Radiography	5
1.3 Arteries and Veins distinct separation	6
2 Methods	11
2.1 Algebraic Reconstruction Technique	11
2.2 Contrast Dynamics Recovery: the ART 3.5 D Algorithm	15
2.3 Problem Dimension Analysis	20
2.3.1 Basis Functions Dimensioning	21
2.3.2 LTV System Dimension Analysis	25
3 Simulation Protocol	29
3.1 Simulation Design	29
3.2 ART 3.5 D Implementation	34
3.2.1 Memory Issues	42
3.3 ART 3.5 D Optimized Implementation	45
3.3.1 Computational Issues	46
3.4 Further Proposed Improvements for the ART 3.5 Optimized Algorithm	48
3.5 Evaluation Protocol	52

3.5.1	Evaluation of the TIC reconstruction	52
3.5.2	ROC Curves Analysis and Evaluation	53
3.5.3	Evaluation of the Arteries and Veins Classification	55
4	Results	57
4.1	TIC Reconstruction	57
4.2	ROC Curves Analysis and Evaluation	58
4.3	Arteries and Veins Classification	58
5	Conclusions and Future Work	71
Appendix A	Tomography	75
A.1	Computed Tomography	75
A.2	Image Reconstruction	77
A.2.1	Image Reconstruction based on CC model	82
A.2.2	Image Reconstruction based on DC model	84
A.2.3	Image Reconstruction based on DD model	85
A.3	Filtered Backprojection	86
A.4	Feldkamp-Davis-Kress (FDK) Algorithm	90
Appendix B	Implementation Details	93
B.1	Simulation	93
B.2	System Matrix	97
B.3	Dynamic System Matrix	103
B.4	Basis Functions	103
B.5	Segmentation	104
Bibliography		107

List of Figures

1.1	Multimodal scene: the multimodal 3D Slicer scene, an image set including models of the pial surface (white, with the precentral and postcentral gyri in blue and red, respectively), the SEEG electrodes (yellow, with brown entry site markers and labels), and the 3D DSA cerebrovasculature (green, semitransparent).	4
1.2	Angiographic Image: and X-ray image was acquired during contrast medium transit and the bone was subtracted leaving almost only the vascular tree.	5
1.3	A Digital Subtraction Angiography obtained subtracting the bone mask to the contrast enhanced CBCT reconstruction, obtained with the O-arm4, Medtronic, Minneapolis, Minnesota	7
1.4	Volume rendering of the arteries (red) and veins (blue) obtained with the K-means clustering by Laue et. al. Image taken from [30]	10
2.1	Schema of a generic X-Ray projection	12
2.2	Graphical illustration of the ART solution iteration. Top: the two lines represents two problem constraints in the 2-dimensional problem. The intersection of the two lines represents the exact problem solution. The vector μ_0 identifies the initialization point in the image space, the vector μ_1 is the image vector obtained after a first update, the vector μ_2 is the image vector obtained after the second update. The projection steps are qualitatively illustrated; Bottom: the steps and the vectors needed for the update computation are illustrated.	14
2.3	Schema of the acquisition geometry seen from above downwards.	17
2.4	Graphical illustration of the integration kernel (i.e the weight $w_{i,j}$) computation for the projection ray i crossing the voxel j	19
2.5	Basis Functions evaluated for the temporal profiles modeling. Top: Sigmoidal Basis Functions; Middle: Rectangular Basis Functions; Bottom: Triangular Basis Functions.	24

2.6	The trend of the median RMSE among the randomly generated TICs and their modeling through a rectangular set of B basis functions with a number B varying from 1 to 50.	25
2.7	A 3D Slicer DSA vessel reconstruction.	28
3.1	Proportionate schema of the O-arm acquisition geometry.	31
3.2	The dynamic of the contrast flow was simulated with two different dynamic behaviours	33
3.3	Two simulated vascular tree phantoms used for simulation.	35
3.4	Sequence of images showing the simulated contrast flow in the vascular tree phantom A.	36
3.5	Block diagram showing the ART 3.5D algorithm structure. All the quantities in input at each block are orange colored, whereas all the quantities in output from each block are green colored.	37
3.6	Computational process for the Dynamic extension of the System Matrix: block (A) represents the basis functions matrix, where the angular samples vary on the rows, while the values of different basis functions vary on columns; block (B) represents the static system matrix W ; the block (C) represents the static matrix reduced eliminating all the voxels that do not belong to the vascular tree, leaving a number $N_{V_{voxel}}$ of voxels; block (D) represents the Dynamic System Matrix, in which the basis functions have been inserted.	39
3.7	Example of an interpolation process from a voxel's stairstep TIC to a smoother sigmoidal trend.	40
3.8	Graphical representation of the arterial AUC (AUC_A) and venous AUC (AUC_V)	41
3.9	Comparison among arterial, capillary and venous phase in angiographic images	43
3.10	Block diagram showing the ART 3.5D algorithm optimized structure for real data processing. All the quantities in input at each block are orange colored, whereas all the quantities in output from each block are green colored.	47
3.11	Fitting curve of computational time data. On the y-axis the computational time values in seconds for a single ART iteration. On the x-axis the image dimension expressed in voxels.	49

3.12	The maximum sum of the maximum distances between the arterial and venous ROC curves points and the random guess diagonal line points defines the best threshold parameters for arteries and veins classification. The red ROC curve is the arterial ROC; the blue ROC curve is the Venous ROC.	54
4.1	Median ϵ index trend over iterations, for each simulation setup.	59
4.2	Median TICs RMSE trend over iterations, for each simulation setup.	60
4.3	Receiver Operating Characteristics (ROCs) for arteries and veins classification. Left: ROC curves for the Vessel Tree A; Left: ROC curves for the Vessel Tree B	64
4.4	Median Arteries TP index trend over iterations, for each simulation setup.	65
4.5	Median AV index trend over iterations, for each simulation setup.	66
4.6	Median VA index trend over iterations, for each simulation setup.	67
4.7	Median Veins TP index trend over iterations, for each simulation setup.	68
4.8	Illustration of the reconstructed and simulated arterial tree for the vessel tree A.	69
4.9	Illustration of the reconstructed and simulated venous tree for the vessel tree A.	70
A.1	The four generations of CT scanners	78
A.2	Generic illustration of a X-Ray path in an object from the point A to the point B.	79
A.3	Graphical representation of the computation of weights. In the specific 2D example the weight $w_{4,1,h}$ is computed as the area of the pixel $[4, 1]$ and represents the contribution of the pixel $[4, 1]$ to the projection p_h	81
A.4	Graphical illustration of the Central Section Theorem. (x,y) are the space variables while (u,v) are the frequencies variables. The angle ω here represents the direction of the projection ray.	83
A.5	Star artifact on a point reconstruction. On the left the artifact for a finite number of back projection angles. On the right, the star artifact when the number of considered directions of projection tends to infinite	87
A.6	Graphical illustration of the back-projection result: all the points on a line obtain a value equal to the value $p_\omega(t)$ of projection along that line.	88
A.7	Comparison among Shepp-Logan Phantom and its backprojected version.	89
A.8	The Cone Beam Geometry is illustrated. The reconstructed volume is composed of a cylinder plus two cones. The set of projection rays can be represented by several inclined fan.	90

A.9	Geometric schema of elevation (κ) and in-plane (γ) angles in a 3D Cone Beam projection	92
B.1	Reference system for the acquisition set up	98
B.2	Three planes delimiting a voxel	99
B.3	Example, on a 3×3 matrix, of the voxel numeration adopted for the system matrix ordering	102

List of Tables

2.1	Medtronic O-ARM4 acquisition parameters	21
2.2	Median RMSE values for TIC modeling with triangular, rectangular and sigmoidal basis functions	22
2.3	Mean RMSE values for TIC modeling with triangular, rectangular and sigmoidal basis functions. TICs are computed with the iterative approach	23
3.1	Simulated Geometric Parameters	32
3.2	Simulated Acquisition Parameters	32
3.3	Simulation setups summarized. For each of the two vessel tree shapes (A and B), two different contrast transit dynamics were simulated, namely a step dynamics and a sigmoidal dynamics.	52
4.1	Statistical values of RMSE among the simulated and reconstructed TICs for each simulation setup. A1, A2, B1 and B2 are the simulation setups names listed in Table 3.3.	58
4.2	Value of the threshold parameter k (multiplied to the total AUC for arteries classification) for each simulation setup.	58
4.3	True positive, false positive, true negative and false negatives for arteries classification are shown for each simulation setup.	61
4.4	True positive, false positive, true negative and false negatives for veins classification are shown for each simulation setup.	62
4.5	Arteries correctly classified, veins incorrectly classified as arteries, arteries incorrectly classified as veins and correctly classified veins are shown for each simulation setup.	62
4.6	Sensitivity and Specificity values for each simulation setup.	63

Abstract

Introduction: Visualization of brain vasculature is truly helpful in many neurosurgical fields and it is gaining more and more importance. In fact, the vessels are of primary importance in both the pre-operative and intra-operative phase for planning, multimodal image registration, image-guided neurosurgery and brain shift compensation, among many other purposes. The pre-operative phase requires the tracking of the whole vascular tree mainly for the intervention planning so as to avoid vessel hurting and subsequent haemorrhage. The pre-operative planning must be subsequently registered to the intra-operative scenario, and in this context Cone-Beam-Computed-Tomography (CBCT) represents a powerful facility, since it is capable to acquire images in the operation room, shortly before surgery. In this context, the distinction between arteries and veins within the reconstructed vascular tree is still lacking, though it is important both for neurosurgical planning and the detection neurological diseases such as ArteroVenous Malformations (MAV). Even if segmentation of arteries and veins has already been produced starting from 4D CT perfusion scans, much longer acquisition protocols are needed with this technique, since several volumes are acquired instead of a single volume. Moreover the radiation dose given to the patient is increased. With 4D CT scans a time profile for each voxel is available, though with heavy limitations in time resolution that can be tentatively encompassed by post-processing techniques. Limits to these attempts are mainly in the need of dedicated scanners, not currently available in the clinical and Operating Room environment. Hence, space is open to approaches, as the one proposed in this thesis, attempting to recover the dynamic information from standard CBCT contrast enhanced scans.

Methods: A theoretical approach for the recovery of temporal information (i.e. the 4th dimension) from standard CBCT contrast enhanced projections is introduced as novel method, to our knowledge. The dataset employed by the method is composed of two standard sets of CBCT projections over 360°: a first data set taken without the employment of contrast medium and a second dataset acquired while the contrast medium is flowing through the vascular tree from arteries to veins. Starting from those two datasets, a vessel tree reconstruction can be achieved through Digital Subtraction

Angiography (DSA), i.e. the volume reconstructed from the non-contrasted dataset is subtracted from the volume reconstructed with the contrast-enhanced dataset. Once the DSA reconstruction is available, the dynamic information must be added to each voxel belonging to the vessel tree. Though, the CBCT dataset is by far insufficient at this purpose. In fact, in the full 4D problem the number of 4D unknowns (i.e., volume changing over time, order N^4) is consistently higher than the number of measures (i.e. CB projections on a 2D flat panel, times angular/temporal samples, order N^3). However, the working hypothesis is that the negligible contrast of the surrounding background thanks to digital subtraction and the constraints furnished by a preliminary vascular segmentation reduce the unknowns to the vascular voxels. Moreover, the predictable time course of the contrast in a vascular voxel is highly regular and characterized by a wash-in and a wash-out phase (in case of bolus injection), or by a wash-in phase only (in case of continuous contrast injection). Hence, further reduction of the number of unknowns can be obtained by describing dynamics through a small number of temporal basis functions. The presented approach extends a standard iterative Algebraic Reconstruction Technique from the Linear Time Invariant (LTI) problem, to a Linear Time Variant (LTV) problem, exploiting information about system dynamics during contrast enhanced CBCT acquisitions. A LTI system has the standard form $\mathbf{b} = \mathbf{A} \cdot \mathbf{x}$, where \mathbf{b} is the experimental measures vector, \mathbf{A} is the system matrix and \mathbf{x} is the vector of the unknowns. \mathbf{A} contains, for each unknown, its contribution to each measure. A standard image reconstruction problem can be expressed as a LTI system, being \mathbf{b} the vector of projections, \mathbf{A} the system matrix containing the attenuation path of each voxel to each projection ray and \mathbf{x} the vector of voxel's attenuation coefficients. This LTI system can be transformed to a LTV system. In fact, during a contrast enhanced CT acquisition, image voxels do not have static values over time, due to the contrast medium flow. Therefore, the content of each voxel can be modeled as a Time Intensity Curve (TIC). In order to reconstruct a TIC for each voxel, dynamic information must be recovered from the projection dataset. In a standard CBCT acquisition protocol the projections are taken on a Flat Panel Device (FPD), which rotates together with the source, around the gantry axis. Therefore, each projection taken at a specific angle θ_k strictly corresponds to a time instant t_k and contains a dynamic piece of information that can be exploited for the Time Intensity Curves (TICs) reconstruction. Since two standard sets of CBCT projections (acquired with and without contrast injection) are in use, a final dataset for TICs reconstruction was created, containing only the contrast enhanced information. Naming the projection set before contrast injection p^0 and the projection set after the contrast injection p^{ci} , the final projection set was extracted as the difference among them, $p = p^{ci} - p^0$, so that

only the contribute of vessels was considered in projections. Moreover, a reduction of the number of the unknowns was introduced taking into account only the vascular voxels for the reconstruction. The TIC of each voxel was modeled as a linear combination of basis functions, whose weighing coefficients had to be estimated. The basis functions set was chosen a priori, basing on the modeling needs and since they were included in the a priori knowledge they were included in the system matrix. Then a new system matrix was introduced, which combined the static weights and the samples of the basis functions at each projection time t_k for each projection at the angle θ_k . Once the system matrix was extended and the projection data set was adapted, the iterative process was started leading, as a solution, to a set of basis functions coefficients for each voxel. The TIC is then extracted as the linear combination of the basis functions weighed by the coefficients. Once the TIC was reconstructed for each voxel, the voxel was classified as belonging to “artery” or “vein” basing on his TIC. In particular, the arterial Area Under The Curve (AUC_A) was calculated as the integral of the TIC from zero to half of the acquisition time and a threshold on this value was imposed for arteries classification: if the value of AUC_A resulted over the threshold, the voxel was classified as “artery”, differently it was classified as “vein”. This algorithmic choice was due to the fact that the continue injection of contrast makes unobservable the contrast wash-out phase. In fact, the contrast wash-out takes place only at the end of the acquisition process, for all the vessels. Differently, the wash-in phase is observable for each voxel, since it presents very different timing in arteries (wash-in phase in the first few instants of the acquisition) and veins (wash-in phase in the last instants of the acquisition). Several simulations were performed. A set of rectangular basis functions was chosen after some experimental trials, since it demonstrated to be the most suitable. Two vessel-tree phantoms were built, with different morphological characteristics, both of them including several branches. The contrast transit dynamics was modeled both as step dynamic and sigmoidal dynamic. The sigmoidal dynamic modeling was introduced in order to make the simulation far more realistic than the step dynamic simulation. Different simulations were performed varying the dimensions of the projection dataset and the dimensions of the reconstructed 4D image volume. Within the performed simulations, the Receiver Operating Characteristic (ROC) curves were built in order to inspect the optimal threshold value on AUC_A for arteries classification. Once the arteries and veins were classified using the optimal threshold value, error measures were evaluated on the reconstructed temporal profiles, namely the Root Mean Square Error (RMSE) among the simulated and reconstructed TICs and the ART objective function ϵ over the iterations. Several indexes were chosen for the classification evaluation, namely true positive, false positives, true negatives

and false negatives for both arteries and veins classification. In addition, cross indexes for the evaluation of exchange of arteries as veins classification (VA index) and vice versa (AV index) were employed. The whole algorithm was implemented in Matlab 2015b on a Mac Book Air , 1.7 GHz Intel Core i7, 8GB DDR3 at 1600 MHz.

Results: For each experimental setup the median results was computed among all the simulation trials. The arteries classification results can be summarized as follows: median true positive value ranges from 0.9737 to 1 varying the experimental setup; median true negative values range from 0.9937 to 0.9997; median false positive values range from 0.0003 to 0.0063; median false negative values range from 0 to 0.0263. The veins classification results can be summarized as follows: median true positive values range from 0.8795 to 0.9885; median true negative values range from 0.9995 to 1; mean false positive values range from 0 to 0.0005; mean false negative values range from 0.0115 to 0.1205. Therefore up to 90% of arteries and veins were correctly classified. Those results show the suitability of the presented method for the recovery of contrast dynamics, leading to good arteries and veins separation, even if only simple simulation results are available.

Conclusions: In conclusion, a method for dynamic information recovery from standard contrast CBCT acquisition is introduced as a novel method, to our knowledge. The recovered dynamic information allows the distinct classification of arteries and veins. A standard iterative Algebraic Reconstruction Technique is extended from a Linear Time Invariant System to a Linear Time Variant System, and a Time Intensity Curve is reconstructed for each voxel. Several simulations were performed and good results were obtained, even if at this stage real data could not be processed due to very high computational times. In fact a brief study of the computational time trend with increasing image dimensions was performed and the needed computational time forecasted for a complete real data processing is up to a month. Several theoretical solutions were proposed for this issue, mainly the exploitation of parallel computation techniques. All the proposed solutions require implementation times, accurate theoretical studies and infrastructures that make the complete solution beyond the aim of this thesis.

Sommario

Introduzione: La ricostruzione e la rappresentazione tridimensionale dell'albero vascolare risultano molto utili in molte applicazioni neuro-chirurgiche e stanno recentemente acquistando sempre più rilevanza. La visualizzazione dei vasi sanguigni è infatti di primaria importanza sia nella fase pre-operatoria che nella fase intra-operatoria in molte applicazioni quali la pianificazione chirurgica, la registrazione di immagini multi-modali, la neurochirurgia assistita e la compensazione dello spostamento corticale in seguito a resezione. La fase pre-operatoria richiede la localizzazione dell'albero vascolare per la pianificazione dell'intervento, in modo da evitare un danneggiamento vascolare con conseguente emorragia. Il piano pre-operatorio deve essere successivamente essere integrato e registrato con la procedura intra-operatoria. A questo scopo la Tomografia Computerizzata a Cone Beam (CBCT) rappresenta una facilitazione consistente, poiché permette l'acquisizione di immagini direttamente nella sala operatoria, poco prima dell'inizio dell'intervento. In questo contesto applicativo, manca una distinzione tra vene e arterie nell'albero vascolare ricostruito, nonostante la grande importanza che tale distinzione avrebbe per il planning neuro-chirurgico e la diagnosi di alcuni disturbi di origine neurologica, quali le Malformazioni Artero-Venose (MAV). Nonostante la distinzione tra vene e arterie sia già stata raggiunta analizzando scansioni 4-dimensionali in perfusione (4D CTP), questa tipologia di acquisizione introduce tempi di scansione molto lunghi, poiché vengono acquisiti molteplici volumi immagine nel tempo. D'altronde, vista la molteplicità dei volumi acquisiti, questa tecnica consente la costruzione di un profilo temporale per ogni voxel, anche se con consistenti limitazioni di risoluzione temporale. Queste tecniche presentano limitazioni soprattutto legate alla necessità di scanner dedicati, che non sono attualmente disponibili nelle sale operatorie.

Questo lavoro di tesi risponde all'esigenza di introdurre una tecnica per il recupero dell'informazione dinamica da acquisizioni CBCT standard con iniezione di mezzo di contrasto, attualmente disponibili in sala operatoria.

Metodi: Per la nostra attuale conoscenza, questo lavoro di tesi introduce un approccio teorico innovativo per il recupero dell'informazione temporale (i.e la quarta dimensione)

da acquisizioni CBCT con mezzo di contrasto. Un protocollo CBCT con mezzo di contrasto prevede l'acquisizione di due dataset di proiezioni a raggi X: il primo senza iniezione di mezzo di contrasto e il secondo con iniezione di mezzo di contrasto nella carotide cerebrale. L'iniezione in carotide garantisce che il mezzo di contrasto scorra prima nelle arterie e poi nelle vene. Da questi due dataset può essere effettuata una ricostruzione dell'albero vascolare tramite una Angiografia Sottrattiva Digitale (DSA). Una volta ottenuta la DSA l'informazione dinamica deve essere ricostruita per ogni voxel appartenente all'albero vascolare. In realtà, il dataset derivante dalla CBCT è assolutamente insufficiente per la risoluzione di un problema 4D completo che presenta un numero di incognite dell'ordine N^4 , quindi consistentemente più alto del numero di misurazioni (ovvero di proiezioni sul Flat Panel Device (FPD) per ogni angolo di campionamento) che risulta dell'ordine N^3 . Tuttavia, l'ipotesi alla base dell'approccio è che le limitazioni fornite da una precedente segmentazione dell'albero vascolare e la concentrazione trascurabile del contrasto nei tessuti circostanti riducono il numero di incognite ai soli voxel appartenenti all'albero vascolare. Inoltre, è ipotizzabile che l'andamento temporale del contrasto in un voxel dell'albero vascolare sia molto regolare e caratterizzato da due fasi distinte di aumento e di diminuzione del contrasto (nel caso di iniezione di bolo) oppure da una sola fase di aumento del contrasto, con conseguente mantenimento della concentrazione (in caso di iniezione continua). Conseguentemente, la dinamica di ogni singolo voxel può essere descritta da un numero finito di funzioni base distribuite nel tempo. L'approccio qui presentato estende una soluzione iterativa di tipo ART (Algebraic Reconstruction Technique) da un sistema Lineare Tempo Invariante (LTI) ad un sistema Lineare Tempo Variante (LTV) sfruttando l'informazione dinamica raccolta durante l'acquisizione CBCT con iniezione di contrasto. Un problema LTI è formulato come $\mathbf{b} = \mathbf{A} \cdot \mathbf{x}$, dove \mathbf{b} è il vettore delle misure sperimentali, \mathbf{A} è la matrice di sistema e \mathbf{x} è il vettore delle incognite. La matrice di sistema contiene il peso di ogni incognita su ogni specifica misura sperimentale in \mathbf{b} . Un classico problema di ricostruzione di immagini da proiezioni può essere ricondotto ad un sistema LTI, dove \mathbf{b} è il vettore dei valori di proiezione, \mathbf{A} è la matrice di sistema (che contiene, per ogni voxel, l'estensione del percorso effettuato da ogni raggio di proiezione in quel voxel) e \mathbf{x} è il vettore dei voxel dell'immagine da ricostruire. Il sistema LTI non è sufficiente per la descrizione della dinamica temporale dei voxel, ma può essere esteso in un sistema LTV, mantenendo la linearità del problema. Infatti, durante un'acquisizione CT con mezzo di contrasto, i voxel non hanno un valore costante durante il tempo di acquisizione, proprio a causa del flusso del mezzo di contrasto. Di conseguenza, il contenuto di ogni voxel può essere rappresentato da un profilo temporale di intensità (TIC, Time Intensity Curve). Per poter ricostruire un profilo temporale per ogni voxel, è neces-

sario recuperare informazione di tipo dinamico dal dataset acquisito. Nel tradizionale protocollo di acquisizione CBCT le proiezioni vengono acquisite su un FPD, che ruota attorno all'asse di rotazione del gantry, in sincronia con la sorgente di raggi X. Ogni proiezione viene acquisita ad uno specifico angolo θ_k , che corrisponde ad un istante di tempo t_k . Il susseguirsi delle proiezioni su FPD ad angoli diversi contiene un'informazione di tipo dinamico che può essere sfruttata per la ricostruzione delle Curve di Intensità rispetto al Tempo (TIC) per ogni voxel. Ricordando che, secondo il protocollo di acquisizione di riferimento, vengono acquisiti due dataset, il primo, p^0 , senza iniezione di mezzo contrasto, e il secondo, p^{ci} durante l'iniezione di mezzo di contrasto, il set di proiezioni usato è stato estratto come la differenza tra i due dataset acquisiti, in modo che nel dataset finale fosse contenuta solo l'informazione riguardante i voxel che contengono mezzo di contrasto, ovvero i voxel appartenenti all'albero vascolare. Avendo a disposizione tale dataset è possibile operare una riduzione del numero delle incognite, riducendole ai soli voxel appartenenti all'albero vascolare. I profili temporali dei voxel sono stati modellati attraverso una combinazione lineare di funzioni base temporali, i cui coefficienti rappresentano il set di incognite per ogni voxel. Le funzioni base, e le loro caratteristiche, vengono scelte a priori, e di conseguenza possono essere inserite all'interno della matrice di sistema, come ulteriore pesatura delle incognite. È necessario precisare che questa ulteriore pesatura introdotta dalle funzioni base è di tipo temporale, non più spaziale, e questa è condizione imprescindibile per l'elaborazione dell'informazione dinamica. Ottenuta la nuova matrice di sistema e il nuovo dataset di proiezioni, il sistema è stato risolto attraverso l'algoritmo ART (Algebraic Reconstruction Technique), una soluzione iterativa algebrica al problema di ricostruzione di immagini da proiezioni. Al termine del processo, ad ogni voxel è stato associato un insieme di coefficienti delle funzioni base e i profili temporali sono stati ottenuti tramite la combinazione lineare delle funzioni base pesate attraverso i loro coefficienti. Una volta ottenuti i profili temporali per tutti i voxel dell'albero vascolare, sulla base di essi i voxel sono stati classificati come appartenenti alle arterie o alle vene. A partire da ogni profilo temporale è stata calcolata l'area sotto la curva "arteriosa" (Arterial Area Under the Curve, AUC_A) come l'integrale da 0 s alla metà del tempo di acquisizione. Su tale valore è stata imposta una soglia: se il valore di AUC_A è superiore alla soglia il voxel è classificato come "arteria", altrimenti come vena. Questa scelta è dettata dal fatto che l'iniezione continua di contrasto non consente di registrare alcuna informazione sulla fase di wash-out del contrasto, che avviene per tutti i vasi solo al termine del processo di acquisizione. Diversamente, resta distinguibile per ogni vaso il fronte di salita, che avviene nei primi istanti di acquisizione per le arterie, e negli ultimi istanti di acquisizione per le vene. Per comprovare il corretto funzionamento dell'algoritmo sono state

effettuate diverse simulazioni. Studi sperimentali hanno condotto alla scelta di un set di funzioni base rettangolari, che si è dimostrato essere più adatto per una ottimale convergenza alla soluzione. Sono stati computazionalmente costruiti due fantocci di albero vascolare, che comprendessero alcune diramazioni. La dinamica del contrasto è stata simulata sia a gradino che con andamento sigmoidale. La dinamica a gradino è stata introdotta inizialmente per semplicità, ma è successivamente stata affiancata dalla dinamica sigmoidale, più verosimile e più adatta ad una simulazione realistica del flusso di contrasto nei vasi cerebrali. Sui risultati delle simulazioni effettuate sono state costruite delle curve ROC (Receiver Operating Characteristic) per poter individuare il valore ottimale del parametro di soglia su AUC_A per la classificazione delle arterie. La valutazione della classificazione di arterie e vene è stata condotta sulla base di diversi indici: i veri positivi, falsi positivi, falsi negativi e veri negativi per arterie e vene; un indice di misclassificazione delle vene in arterie (indice AV); un indice di misclassificazione delle arterie in vene (VA). Inoltre, altre misure di errore sono state valutate sui singoli profili temporali dei voxel: in particolare sono stati valutati il Root Mean Square Error (RMSE) e la cifra di merito minimizzata dall'algoritmo ART ad ogni iterazione ϵ . L'intero algoritmo è stato implementato in Matlab 2015b su un Mac Book Air, 1.7 GHz Intel Core i7, 8GB DDR3 at 1600 MHz.

Risultati: Per ogni set-up sperimentale considerato sono stati calcolati i valori mediani tra le varie simulazioni effettuate. La classificazione delle arterie ha prodotto i seguenti risultati: il valore mediano (sulle varie simulazioni) dei veri positivi va da 0.9737 a 1; il valore mediano dei veri negativi va da 0.9937 a 0.9997; il valore mediano dei falsi positivi va da 0.0003 a 0.0063; il valore mediano dei falsi negativi va da 0 a 0.0263. La classificazione delle vene ha prodotto i seguenti risultati: il valore mediano (sulle varie simulazioni) dei veri positivi va da 0.8795 a 0.9885; il valore mediano dei veri negativi va da 0.9995 a 1; il valore mediano dei falsi positivi va da 0 a 0.0005; il valore mediano dei falsi negativi va da 0.0115 a 0.1205. Si evince dai risultati che più del 90% dei vasi viene correttamente classificato. Questi risultati, benché ottenuti su semplici simulazioni, mostrano l'idoneità del metodo presentato per il recupero dell'informazione dinamica da acquisizioni CBCT con mezzo di contrasto e per la successiva classificazione dei vasi in arterie e vene.

Conclusioni: In conclusione, per la nostra conoscenza attuale, in questo lavoro viene introdotto un metodo innovativo per il recupero di informazione dinamica da acquisizioni CBCT standard con mezzo di contrasto. Il recupero di tale informazione consente la classificazione di arterie e vene all'interno dell'albero vascolare. Una soluzione iterativa algebrica di ricostruzione da proiezioni è stata applicata ad un sistema Lineare Tempo Variante e per ogni voxel è stato ricostruito un profilo temporale. Sono

state effettuate numerose simulazioni che hanno prodotto buoni risultati che hanno confermato il corretto funzionamento del metodo. Purtroppo il tempo previsto di calcolo per l'applicazione dell'algoritmo ad un dataset reale è risultato eccessivo rendendo impraticabile (viste le infrastrutture a disposizione) un test su un dataset reale. È stato effettuato un breve studio sull'evoluzione dei tempi di calcolo all'aumentare della dimensione del problema, che ha portato alla previsione di circa un mese di tempo necessario alla computazione su un dataset reale. Sono state proposte numerose soluzioni a questo problema: la totalità di queste soluzioni richiede infrastrutture, tempi di implementazione e studi teorici che esulano dallo scopo di questo lavoro di tesi.

Chapter 1

Introduction

Multimodal imaging is truly helpful in many neurosurgical fields such as tumor surgery [20], radiosurgery [46] or functional surgery [3]. Among them, the implantation of StereoElectroEncephaloGraphy (SEEG) intracerebral electrodes and Deep Brain Stimulation (DBS) are fairly new procedures particularly demanding as to imaging support to planning and navigation [9, 3, 43]. Within multimodal imaging for neurosurgery the three-dimensional (3D) visualization of brain vasculature is gaining more and more importance. The vessels play a significant role in both the pre-operative and intra-operative phase, for planning, multimodal image registration, image guided neurosurgery and brain shift compensation, among many other purposes. In particular in the planning scenario, neurosurgery requires the tracking of the whole vascular tree in order to avoid vessel hurting. Namely, Cone-Beam-Computed-Tomography (CBCT) represents a powerful facility capable to acquire images also in the operation room, shortly before surgery. In this context, the distinction between arteries and veins within the reconstructed vascular tree is still lacking, although it would be very important for advanced stereotactic planning, intra-operative image guidance and the detection of neurological diseases such as ArteroVenousMalformations (MAV). Although the raw projection data from contrast enhanced CBCT bring dynamic information of the contrast medium flow in vessels (arteries, next capillaries, next veins), common angiographic reconstructions completely overlook dynamics, since they share the same reconstruction methods of static images, as if the contrast was steadily and homogeneously distributed during the scan. This approximation holds on the purely morphological side, since 360° scans are performed (compared to the theoretical minimum of 180°), during which all vascular segments are filled with contrast for a significant percentage of time. The objective of this thesis is to design and implement an algorithm capable to retain the dynamic information in the contrast enhanced CBCT acquisition,

leading to an arteries and veins distinction and representation. In the next Sections of this chapter the importance of vessel tracking in neurosurgical applications is delineated, and the vessel tree reconstruction technique exploited in this work is presented, in order to introduce fundamental concepts. In addition, some 4D CT based arteries and veins segmentation approaches present in literature are introduced.

1.1 Vessel Tracking in Neurosurgery

Vessel tracking is of primary importance in many neurosurgical procedures, such as tumor resection, Deep Brain Stimulation (DBS), cortical resections, StereoElectroEncephaloGraphy (SEEG) and many other procedures. Vessel tracking can benefit both the intra-operative or pre-operative surgical phases, depending on the specific intervention.

In [14] vessel tracking in intra-operative microscopic video sequences was exploited for the cortical displacement estimation. The cortical displacement often occurs during resection procedures, and its estimation represents a fundamental problem, especially when dealing with spatially accurate procedures co-registered to preoperative image studies [45]. The proposed solutions often relies on vessel localization and tracking in many imaging techniques. Moreover the brain shift represents a source of error for the overlap of pre-operative and intra-operative images and for this reason many vessel-based registration techniques were developed for an accurate image-guided procedures [47, 44].

The vessel tracking can affect also the preoperative phase. In fact, many surgical operations require an accurate preoperative planning and a possible example is the placement of StereoElectroEncephaloGraphy (SEEG) intra-cerebral electrodes, application of choice in this work. The SEEG method consists of the implantation of depth electrodes (usually 5 to 15, 11 on average) in a stereotactic orthogonal framework [19] (Figure 1.1). SEEG allows depth electrode recordings and it is of primary importance in the treatment of drug resistant epileptic patients. In fact, with drug resistant epilepsy, the disconnection or removal of the Epileptogenic Zone (EZ) is the only valid alternative. Those interventions need an accurate planning, due to the complexity of the intervention decisions. In many cases the planning is based on non-invasive clinical investigations [27] but in about the 5% a SEEG is needed: in particular the SEEG plays an important role when the surgical resection for treatment requires a very accurate localization of the EZ[50, 51].

In addition, when SEEG is needed, also the placing of such electrode arrays needs a very accurate pre-surgical investigation and planning [13]. Due to the insertion of the

electrodes in the brain, an accurate targeting of brain structures must be performed. Moreover it requires accurate planning in order to avoid vessel hurting and subsequent intracranial bleeding.

The clinical interest in vessel tracking have led to a strong research interest in reconstruction and analysis of the whole vascular tree. Many segmentation and reconstruction techniques have been proposed until today [28], but all of them are beyond the scope of this thesis. In this work, the Digital Subtraction Angiography is the method of choice for the vascular tree reconstruction. In fact, the CBCT DSA acquisition protocol allows the recovery of dynamic information, which is commonly overlooked in the other approaches. Cardinale et al. first described the clinical use of 3D Cone-Beam-Computed-Tomography (3D CBCT) Digital-Subtraction-Angiography (DSA) obtained processing the images scanned with the O-arm4 ([9, 10, 11]), (Medtronic, Minneapolis, Minnesota). The DSA approach will be presented in detail in Section 1.2.

1.2 Digital Angiography

Angiography has in origin been carried on using standard contrast radiography. Since the attenuation of the blood and soft tissue are too similar to be distinguished the use of contrast medium is needed for the visualization of blood vessels. Conventional angiography requires the positioning of a catheter directly into a vessel, in order to get enough contrast. The introduction of a catheter decreases the safety of the procedure and requires longer treatment lasting, including hospitalization, however in intra-operative applications this does not constitute an issue. When the contrast medium is non-selectively injected in the vascular tree the image contrast is not enough, since the contrast medium is too diluted, making necessary a subtraction method. Those methods subtract high contrast structures (i.e. bone or different tissue interfaces) to the data set acquired during the contrast medium flow. Three main techniques can be considered among the subtraction methods [6]:

- **Energy subtraction:** exploits the different attenuation of tissues as a function of X ray. K-edge subtraction is based on directing two beams with energy respectively immediately above and below the absorption value of iodine. The subtraction of images obtained with the two beams will lead to high contrast structures cancellation;
- **Depth subtraction:** the depth information is exploited in order to eliminate the contribution of tissue or bone structures which are superimposed to the vascular tree;

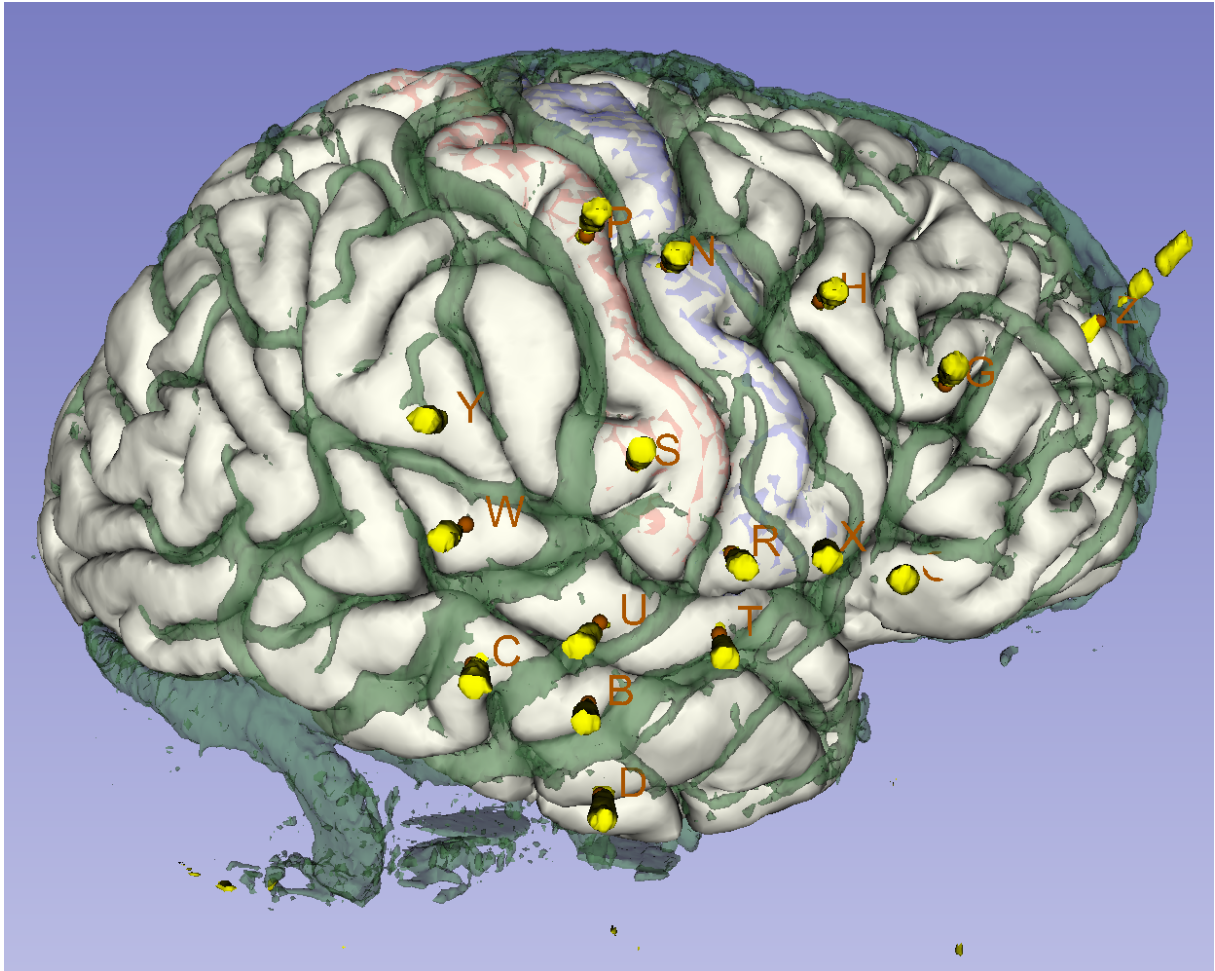


Figure 1.1: Multimodal scene: the multimodal 3D Slicer scene, an image set including models of the pial surface (white, with the precentral and postcentral gyri in blue and red, respectively), the SEEG electrodes (yellow, with brown entry site markers and labels), and the 3D DSA cerebrovasculature (green, semitransparent).

- **Temporal Subtraction:** two datasets must be acquired. The first one, called bone mask, is acquired without the injection of contrast medium. The second one is obtained acquiring after the injection of contrast medium. When the bone mask is subtracted to post contrast image the high contrast structures are cancelled, leaving only the vasculature to be visualized.

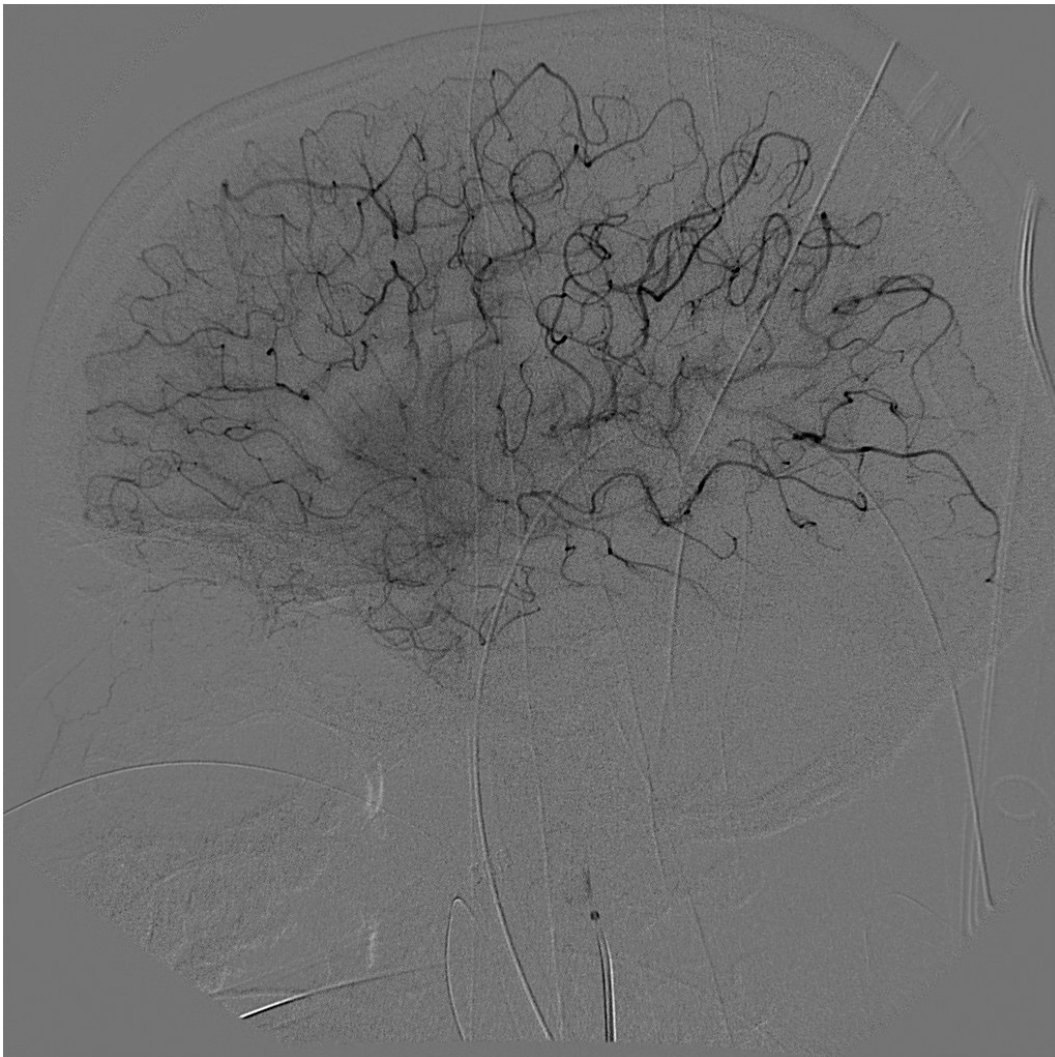


Figure 1.2: Angiographic Image: and X-ray image was acquired during contrast medium transit and the bone was subtracted leaving almost only the vascular tree.

1.2.1 Temporal Subtraction with Digital Radiography

Temporal subtraction with digital radiography accounts many differences with respect to conventional film subtractions. Firstly, the logarithm of the intensity is used for subtraction instead of the intensity itself. Moreover the process becomes significantly

faster, since the images (including the mask) can be stored in the computer memory and a mask can be chosen for subtraction optimization, among many others. Nonetheless, the temporal reconstruction has motion effects limitations. In fact, several seconds elapse amongst the bone mask acquisition and the injection of contrast with the subsequent data acquisition. If in this lapse of time, patient motion occurs, the position mismatch leads to image distortion. This type of artifact can however be compensated and corrected through registration algorithms. A DSA based on temporal subtraction is the vessel reconstruction of choice in this work.

The DSA considerably improves the quality of examination of the vessels and obtains good images of the arteries. Moreover DSA can be obtained with intra-arterial selective contrast injection and this particular case allows a strong enhancement and a good contrast definition despite the use of a small quantity of contrast material. The DSA technique addresses two CBCT acquisitions: the first one is obtained without contrast medium injection and it detects the bone mask; the second one is obtained scanning the images immediately afterwards the injection of contrast medium, detecting the additional signal resulting from the vascular tree. Once both the datasets have been acquired, reconstruction algorithms can be used in order to obtain the 3D reconstructions. Since the image reconstructed starting from the contrast medium dataset contains additional information with respect to the one constituting the bone mask, the bone mask is subtracted from the reconstruction addressing contrast medium flow. Figure 1.3 shows a DSA obtained subtracting two volumes acquired with the O-arm Medtronic [36].

1.3 Arteries and Veins distinct separation

Distinct visualization of arteries and veins can be advisable both for advanced stereotactic planning and intra-operative image guidance. In addition it can be very useful for the detection of neurological diseases such as *Arteriovenous malformations* (MAV) [33]. An arteriovenous malformation consists in an anomalous connection between arteries and veins, that bypasses the capillary system. Since the presence of such malformations can lead to a serious hemorrhage, their detection is important and a distinction among arteries and veins can facilitate the diagnosis of such neurological conditions. Several methods for arteries and veins separation have been proposed, based on the use of both Magnetic Resonance Angiography (MRA) or 4D CT datasets. The MRA techniques mainly exploit correlation algorithms [5], fuzzy connected object delineation principles [33] and many other approaches [42, 32, 31]. However MRA processing techniques are heavily different from those based on CT, mainly due to heavily different acquisition

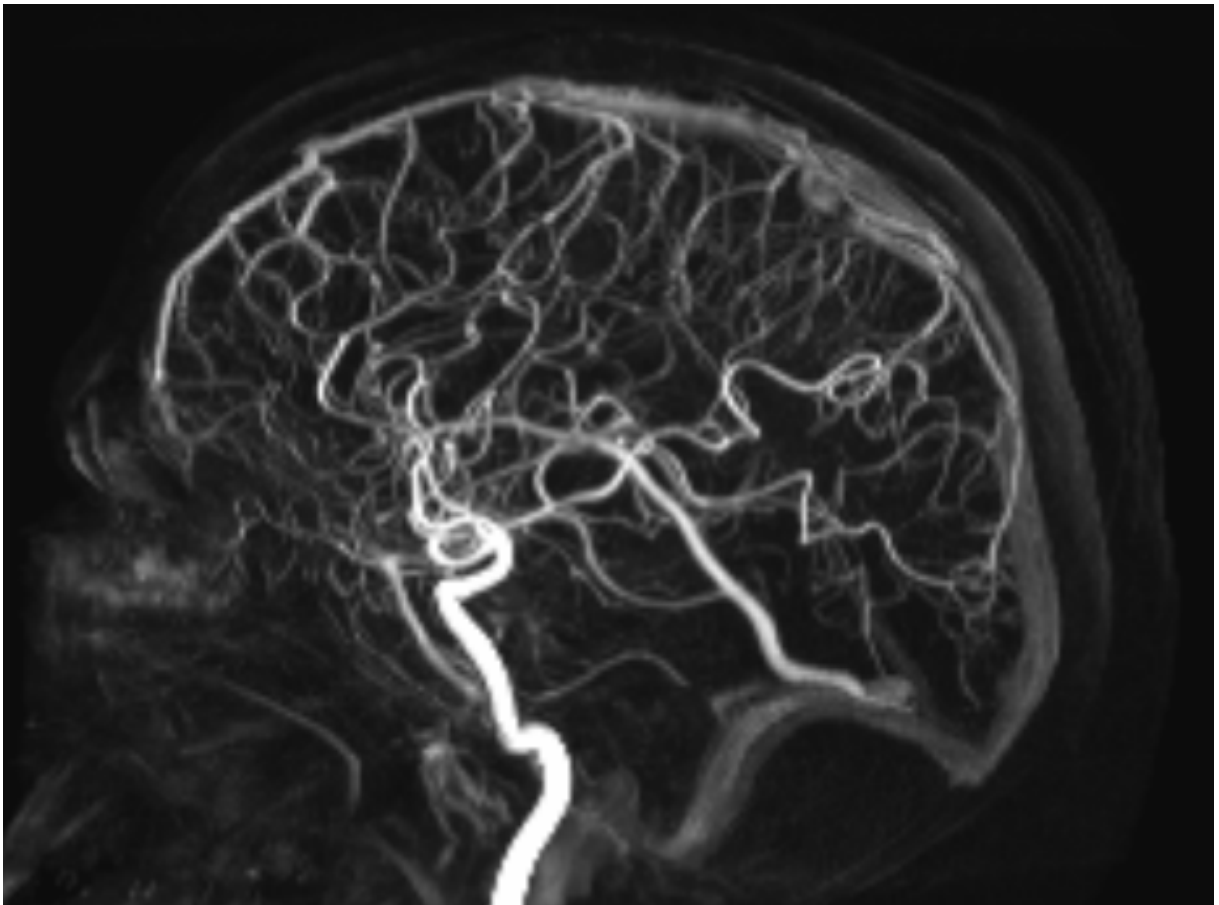


Figure 1.3: A Digital Subtraction Angiography obtained subtracting the bone mask to the contrast enhanced CBCT reconstruction, obtained with the O-arm4, Medtronic, Minneapolis, Minnesota

protocols, therefore a deep study of those technique is beyond the scope of this work, and no further details are here presented.

Another imaging method exploited for arteries and veins differentiation is the 4D computed tomography (4D CT). 4D CT acquires many image volumes while the contrast medium is flowing in the vessels. In this way several volumes changing with time are acquired. In addition, 4D Computed Tomography Perfusion scans can be obtained performing a 4D CT during contrast medium injection, and this very last imaging method is particularly suitable for arteries and veins segmentation techniques. Here some 4D CT based arteries and veins segmentation methods found in literature are presented.

In [30] a method based on the unsupervised classification of the Time Intensity Curve (TIC) is introduced, producing the Artery Input Function (AIF) and Venous Output Function (VOF). The TIC is the image intensity variation over time for each voxel. The AIF describes the contrast medium input to a tissue as a function of time [8] while the VOF describes the output of the contrast medium from the tissue over time. A 4D CT data, comprising 24 volumes, was acquired in a time window of 200 s while 40 ml of non ionic contrast were injected. The first volume was thresholded between 10 and 100 HU, in order to mask the bone and the air outside the whole head. Only the remaining pixels were kept for the study. The first 6 seconds were used for the baseline computation, that was subsequently subtracted in order to exclude the contribution of the non-vessel tissue. The obtained curves in which the baseline was subtracted were named Time Concentration Curves (TCC). Moreover the Areas Under the Curve (AUCs) was computed for all the voxels and the voxels showing at least 95% of the highest AUC were kept. Then K-means clustering was applied and three clusters were identified, namely the arteries the veins and the vessels outside the brain. Some morphological operation were applied in order to remove tracts smaller than 25 voxels, due to noise. The result obtained is shown in Figure 1.4. In conclusion this work showed preliminary results in the feasibility of the K-means clustering for the vessel detection and segmentation.

In [37] the dataset in use consists in a 4D CT perfusion (CTP) scan. Moreover a CT angiography (CTA) is acquired for the vascular segmentation. In fact the CTP is more noisy (due to the reduced radiation dose) and cannot enhance vessel from background. All the 4D CTP volumes are registered using the first volume as a reference. The bone is eliminated using the time-profiles of the CTP scans. Since the 4D CT acquires several volumes over time, for each voxel a time-intensity temporal profile is available. The Gaussian first derivative, very well known for edges detection, is applied to those time profiles, showing the amount of intensity change at a specific time step. In fact,

contrast enhanced vessels show a large amount of intensity change with respect to the background, that remains static. When the vasculature is segmented the Time To Peak (TTP) information is exploited for arteries and veins separation. The zero crossing of the first temporal derivative of the time-intensity profiles, (correspondent to the TTP), is exploited for the creation of a zero crossing image. The zero crossing image is eroded to select only the largest vessels and the arterial peak and the venous peak are identified as the highest peaks. Subsequently all the voxels were labeled as artery or vein depending on their zeros crossing being similar to an arterial or a venous zero crossing. This methods was applied on the CTP scans of 20 patients. For quantitative evaluation two expert observers were asked to label arteries and veins. The ground truth was based on both the observers consecutively, and they were compared to each other. The automatic segmentation was compared to both the observers segmentations, and accuracy, specificity, and sensitivity were computed. Sensitivity values between 0.928 and 0.958 were obtained, together with specificity values between 0.964 and 0.985 and accuracy values between 0.961 and 0.963. A separate arteriogram and venogram was constructed.

In conclusion, methods presented in the literature to separate the brain arterial and venous trees rely on enhanced 4D scans, though with heavy limitations in time resolution, tentatively encompassed by the post-processing techniques. Limits to these attempts are mainly in the need of dedicate scanners, not currently available in the clinical and Operating Room (OR) environment. Hence, space is open to approaches, as the one proposed in the present thesis work, attempting to recover the dynamic information from standard contrast enhanced scans.

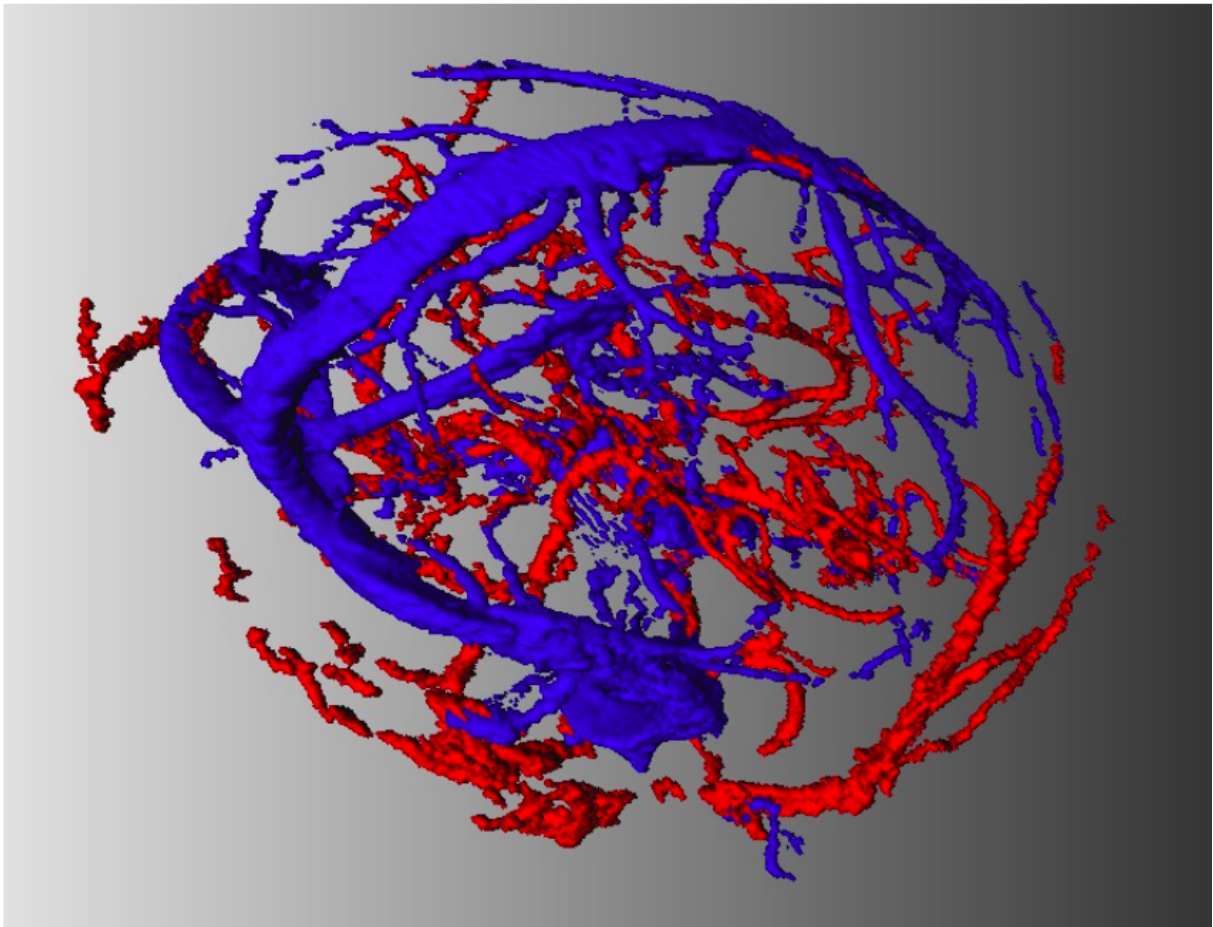


Figure 1.4: Volume rendering of the arteries (red) and veins (blue) obtained with the K-means clustering by Laue et. al. Image taken from [30]

Chapter 2

Methods

In this Chapter, after a brief Algebraic Reconstruction Technique explanation, the theoretical bases of the proposed method are presented. Moreover the problem dimension will be analyzed, in order to verify the feasibility of the proposed approach.

2.1 Algebraic Reconstruction Technique

The Algebraic Reconstruction Technique (ART), is a Discrete-Discrete algebraic iterative approach (see Appendix A). The advantages of these approaches are the possibility to handle with uneven sampling or incomplete data and the good immunity to non idealities, such as metal artifacts. This class of algorithms is especially employed in CT reconstructions where, thanks to the high data statistics, the statistical modeling of the acquisition process can be neglected. ART relies on the method of projections by Kaczmarz [24]. In the general reconstruction problem, J unknown voxels must be reconstructed. Recalling a general formulation of the projection problem (See Appendix A), the i^{th} projection ray produces a projection value p_i . The value p_i is defined as:

$$p_i = \ln \left(\frac{I_{0i}}{I_i} \right) \quad (2.1)$$

where I_{0i} is the initial energy of the i^{th} X-ray and I_i is the energy of the i^{th} X-ray coming out from the sample tissue (Figure 2.1). The reconstructed image is constituted by an attenuation coefficient value μ_j for each j^{th} voxel. The projection value p_i can also been expressed as the sum of the μ_j values weighted by the $w_{i,j}$ coefficients, as follows:

$$p_i = \sum_j w_{i,j} \mu_j \quad (2.2)$$

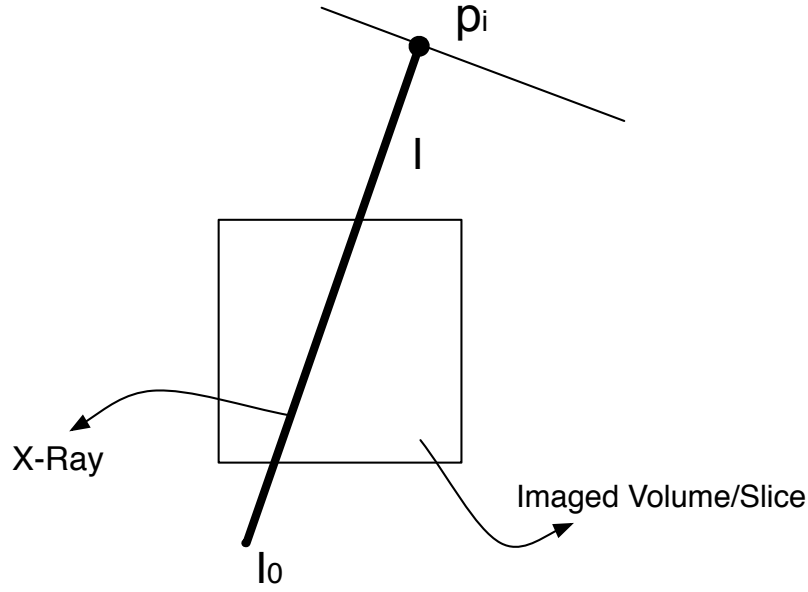


Figure 2.1: Schema of a generic X-Ray projection

The $w_{i,j}$ coefficient weights the contribution of the j^{th} voxel to the i^{th} projection. Being J the total number of image voxels, the entire image is defined by a J -dimensional vector containing the μ_j values and it can be represented as a point ($\boldsymbol{\mu}$) in the J -dimensional space (i.e. the image space). Each equation of the type of Equation 2.2 corresponds to an hyperplane in the J -dimensional space and the solution (i.e the point representing the image) is the hyperplanes intersection [25]. The iterative process is started from an initialization point in the J -dimensional space. Each point in the J -dimensional space is identified by a J -dimensional vector. The initialization vector $\boldsymbol{\mu}_0$ is orthogonally projected on the first hyperplane, i.e the first constraint. The obtained image $\boldsymbol{\mu}_1$ is then orthogonally projected on the second constraint and so on (Figure 2.2). In a single iteration, all the constraints are exploited for the image update. After a predefined number of iterations, the solution point is identified by the final J -dimensional vector. In order to understand the mathematical formulation of the ART update law we can consider the geometrical representation in the two-dimensional case, with two-pixels and two projections, i.e. two equations. In the 2-dimensional space, the hyperplanes are identified by straight lines. With reference to Figure 2.2 we graphically derive the update equations. The iterative process is started from the initialization point identified by the vector $\boldsymbol{\mu}_0$. In order to satisfy the first constraint, the vector $\boldsymbol{\mu}_0$ must be orthogonally projected on it, i.e it must be orthogonally projected on the line representing the first equation $\boldsymbol{\mu} \cdot \mathbf{w}_1 = p_1$. The projection on the first constraint identifies the vector $\boldsymbol{\mu}_1$. The weight vector for the first constraint is identified by the

vector \mathbf{w}_1 , which has a direction orthogonal to the equation. To get the value of μ_1 we must subtract the vector \overrightarrow{AB} , (i.e. the orthogonal distance between the initial guess and the first constraint) to μ_0 . The direction of the vector \overrightarrow{AB} is identified by the unit vector:

$$|\mathbf{w}_1| = \frac{\overrightarrow{w_1}}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \quad (2.3)$$

The norm of the vector \overrightarrow{AB} is computed as the difference between the vector \overrightarrow{OD} and the \overrightarrow{OC} :

$$\overrightarrow{AB} = \overrightarrow{OD} - \overrightarrow{OC} \quad (2.4)$$

The vector \overrightarrow{OD} is the projection of μ_0 on the weight vector \mathbf{w}_1 , and since the projection operation coincides with the scalar product:

$$\overrightarrow{OD} = \frac{\overrightarrow{\mu_0} \cdot \overrightarrow{w_1}}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \cdot |\mathbf{w}_1| \quad (2.5)$$

The vector \overrightarrow{OC} is computed in the same way as:

$$\overrightarrow{OC} = \frac{\overrightarrow{\mu_1} \cdot \overrightarrow{w_1}}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \cdot |\mathbf{w}_1| \quad (2.6)$$

where $\overrightarrow{\mu_1} \cdot \overrightarrow{w_1} = p_1$, then

$$\overrightarrow{OC} = \frac{p_1}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \cdot |\mathbf{w}_1| \quad (2.7)$$

Finally, subtracting \overrightarrow{OC} to \overrightarrow{OD} :

$$\overrightarrow{AB} = \frac{\overrightarrow{\mu_0} \cdot \overrightarrow{w_1}}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \cdot |\mathbf{w}_1| - \frac{p_1}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \cdot |\mathbf{w}_1| \quad (2.8)$$

$$= \left(\frac{\overrightarrow{\mu_0} \cdot \overrightarrow{w_1}}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} - \frac{p_1}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \right) \cdot |\mathbf{w}_1| \quad (2.9)$$

$$= \frac{(\overrightarrow{\mu_0} \cdot \overrightarrow{w_1} - p_1)}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \cdot \frac{\overrightarrow{w_1}}{\sqrt{\overrightarrow{w_1} \cdot \overrightarrow{w_1}}} \quad (2.10)$$

and the μ_1 results:

$$\mu_1 = \mu_0 + \frac{(\overrightarrow{\mu_0} \cdot \overrightarrow{w_1} - p_1)}{\overrightarrow{w_1} \cdot \overrightarrow{w_1}} \cdot \overrightarrow{w_1} \quad (2.11)$$

Generalizing for a generic $\boldsymbol{\mu}_n$ value at the n^{th} iteration and a generic projection p_i , the ART iteration updating is:

$$\boldsymbol{\mu}_{n+1} = \boldsymbol{\mu}_n + \frac{(\vec{\boldsymbol{\mu}}_n \cdot \vec{\boldsymbol{w}}_i - p_i)}{\vec{\boldsymbol{w}}_i \cdot \vec{\boldsymbol{w}}_i} \cdot \vec{\boldsymbol{w}}_i \quad (2.12)$$

The convergence of ART depends on the relative position of the hyperplanes. The more the hyperplanes are orthogonal to each other, the faster will be the convergence. Otherwise, the more the hyperplanes are parallel to each other, the slower will be the convergence. To increase the convergence velocity two different variations on ART has been proposed:

1. Simultaneous Iterative Reconstruction Technique (SIRT): the value of $\boldsymbol{\mu}$ is changed only after having explored all the equations. The updating term for a specific voxel is the average of the updating terms for that voxel [48];
2. Simultaneous Algebraic Reconstruction Technique(SART): the updating terms are simultaneously applied for all the rays in a specific projection [2].

2.2 Contrast Dynamics Recovery: the ART 3.5 D Algorithm

In this Section the theoretical approach for the recovery of temporal (i.e. the 4th dimension) information from dynamic contrast projection is introduced as novel method, to our knowledge. Clearly, a standard set of CBCT projection over 360°, taken while the contrast medium is flowing through the vascular tree from arteries to veins, is by far insufficient to solve a full 4D problem: the number of 4D unknowns (i.e. volume changing over time, order N^4) is consistently higher than the number of measures (i.e. CB projections on a 2D flat panel times angular/temporal samples, order N^3). However, the working hypothesis is that the constraints given by a preliminary vascular segmentation and the negligible contrast of the surrounding background, thanks to digital subtraction, reduce unknowns to the vascular voxels. Moreover, the predictable time course of the contrast in a vascular voxel is highly regular and characterized by a wash-in and a wash-out phase or in alternative, for continue contrast injection, by a wash-in phase only. Hence, further reduction of the number of unknowns can be obtained by describing dynamics through a small number of basis functions.

The presented approach extends an iterative algebraic solution from a Linear Time Invariant (LTI) to a Linear Time Variant (LTV) System exploiting information about

system dynamics during contrast enhanced CBCT acquisitions. The Algebraic Reconstruction Technique was extended to the case of dynamic image reconstruction. In this way a Time Intensity Curve (TIC) was reconstructed for each voxel, instead of a grey value. The vessels were therefore classified as “artery” or “vein” basing on their voxels TICs. A general LTI System is formalized as:

$$\mathbf{p} = \mathbf{W}\boldsymbol{\mu} \quad (2.13)$$

where \mathbf{W} is the system matrix, $\boldsymbol{\mu}$ is the unknowns vector and \mathbf{p} is the measures vector. Avoiding the matrix formulation, the system can be rewritten as :

$$p_i = \sum_{j=1}^J w_{i,j} \cdot \mu_j = |w_{i,1} \dots w_{i,J}| \cdot \begin{vmatrix} \mu_1 \\ \dots \\ \mu_J \end{vmatrix} \quad (2.14)$$

where for the image reconstruction problem p_i is a single projection element, the terms $w_{i,1} \dots w_{i,J}$ are the elements of System Matrix \mathbf{W} and the terms $\mu_1 \dots \mu_J$ are the image voxel values to be reconstructed. Equation 2.14 allows to reconstruct static 3D images (i.e. the vector $\boldsymbol{\mu}$) from standard datasets obtained without contrast medium. In a standard CBCT acquisition protocol, the projections are taken on a Flat Panel Device (FPD), which rotates together with the source, around the gantry axis. Therefore, each projection taken at a specific angle θ_k strictly corresponds to a time instant t_k . Moreover each projection at the angle θ_k leads to $(N \times M)_k$ projection values, where N and M are the dimensions of the FPD in pixels. Therefore, the projection index i is defined by $i = i(\theta_k, a_m, b_n)$, where a_m and b_n are the horizontal and vertical displacements on the FPD, while θ_k is the focal spot rotation angle, which is the relevant parameter in the dynamic problem.

The projection angle θ_k is discrete and the angular displacement between two FPD projections is equal to $\Delta\theta$:

$$\theta_k = k\Delta\theta \quad k = 1, 2 \dots K \quad (2.15)$$

with K number of projection angles. The index k depends on the time instant and so the trend of θ_k orderly follows the projection timing t_k (Figure 2.3).

$$\theta_k = k(t_k)\Delta\theta \quad (2.16)$$

Exploiting this piece of information, normally not accounted for in static problems, a Time Intensity Curve (TIC) was extracted for each voxel. The TIC of each voxel was modeled as a linear combination of basis functions, whose weighing coefficients had

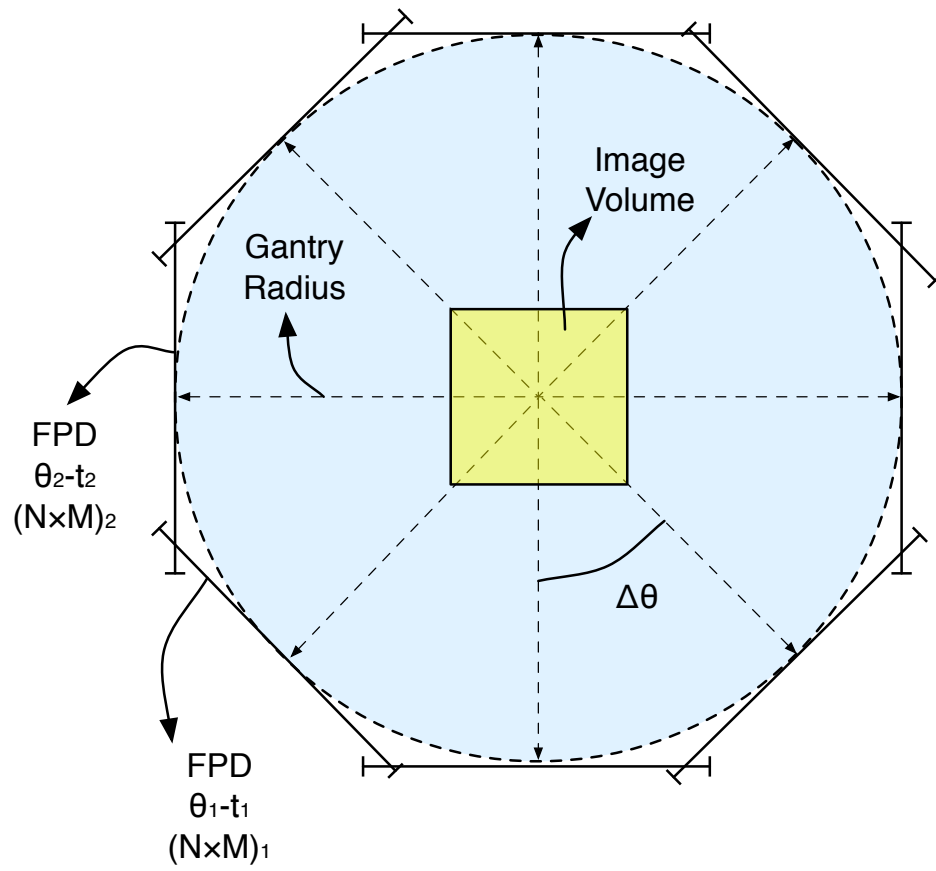


Figure 2.3: Schema of the acquisition geometry seen from above downwards.

to be estimated. The basis functions set was chosen a priori, basing on the modeling need, therefore they were included in the system matrix \mathbf{W} . A new system matrix \mathbf{W}_d was introduced which combined the static weights $w_{i,j}$ and the samples of the basis functions at each projection time t_k . To understand how the system matrix was expanded, and the LTI problem can be rephrased, we can rely on some simple mathematical manipulations. The term μ_j , which was a grey value in the static problem, was transformed in a time function for the dynamic problem, the TIC. The TIC of each voxel j was then expressed by $\mu_j(t)$. This time curve of each voxel was then modeled through a linear combination of basis functions, as previously introduced:

$$\mu_j(t) = \sum_{b=0}^B d_{j,b} \cdot q_b(t) = \begin{vmatrix} d_{j,0} & \dots & d_{j,B} \end{vmatrix} \cdot \begin{vmatrix} q_0(t) \\ \dots \\ q_B(t) \end{vmatrix} \quad (2.17)$$

where the coefficients $d_{j,b}$ weight the basis functions. The set of weights $d_{j,b}$ corresponds to the new set of unknowns, for a number J of angiographic voxels. Substituting the term $\mu_j(t)$ (expressed in Equation 2.17) in the LTI problem formulation (Equation 2.14), the following expression is obtained:

$$p_i = \sum_{j=1}^J w_{i,j} \cdot \mu_j(t) = \sum_{j=1}^J w_{i,j} \cdot \left(\sum_{b=1}^B d_{j,b} \cdot q_b(t_k) \right) \quad (2.18)$$

Rearranging Equation 2.18, the following expression is obtained:

$$p_i = \sum_{j=1}^J \sum_{b=1}^B q_b(t_k) \cdot w_{i,j} \cdot d_{j,b} \quad (2.19)$$

Grouping all the terms known a priori a new system matrix was defined, with elements:

$$w_{i,j}^d = \sum_{b=1}^B w_{i,j} \cdot q_b(t_k) \quad (2.20)$$

In vector notation:

$$\mathbf{w}_{i,j}^d = w_{i,j} \cdot \begin{vmatrix} q_1(t_k) & \dots & q_B(t_k) \end{vmatrix} \quad (2.21)$$

Rephrasing Equation 2.19

$$p_i = \sum_{j=1}^J \sum_{b=1}^B \mathbf{w}_{i,j}^d \cdot \mathbf{d}_{j,b} \quad (2.22)$$

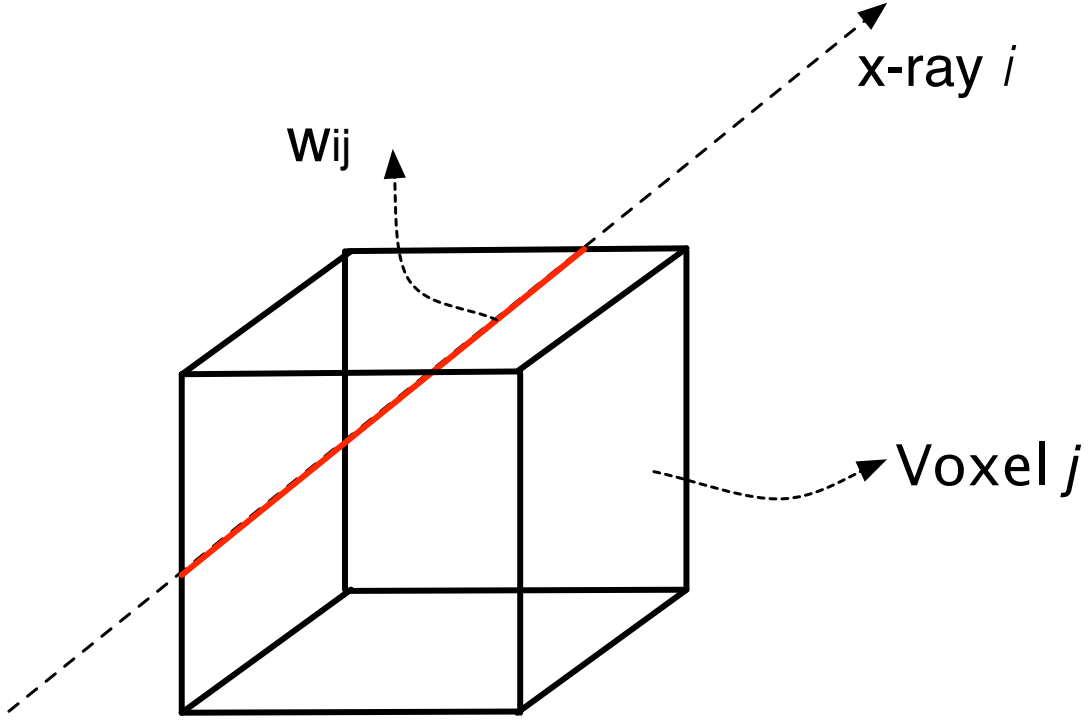


Figure 2.4: Graphical illustration of the integration kernel (i.e the weight $w_{i,j}$) computation for the projection ray i crossing the voxel j .

and in matrix notation

$$\mathbf{p} = \mathbf{W}_d \boldsymbol{\mu} \quad (2.23)$$

where \mathbf{W}_d is the new Dynamic System Matrix. Therefore the proposed method retrieves dynamic information from a standard linear algebraic problem, which can be solved by a reconstruction algorithm.

Since the entire problem was reduced to a linear system, based on the adhibition of basis functions, the approach of this work is attributable to a Discrete-Discrete approach, introduced in Appendix A. In particular, within this approach, five components must be chosen, and in this work the following choices were made:

1. **A model of the physics of the measurement process:** the integration kernels $w_{i,j}$ implies an attenuation characteristics based on the length of the path that a projection ray i runs in a specific voxel j . The integration kernel computation is illustrated in Figure 2.4.
2. **A model of the measurement uncertainty:** no hypotheses on the statistics of measure uncertainty were made because the high CT statistics does not require

additional hypotheses.

3. **A set of basis functions:** since the dynamics was introduced, temporal basis functions were chosen (whereas normally spatial defined basis functions are used in static 3D reconstructions, aiming at define a spatial linear combination for the image to be reconstructed). A set of rectangular basis functions was chosen for the simulations, as illustrated in Figure 2.5. Those basis functions proved to be the most suitable among rectangular, triangular sigmoidal and sinusoidal basis functions. The choice of rectangular basis functions was led from an experimental trial, explained in Section 2.3.1.
4. **An objective function:** the ART objective function is the difference between the projection data acquired scanning the patient and the simulated projection obtained applying a mathematical projection operator \mathcal{P} to the reconstructed image at the considered iteration. Therefore, naming μ_n the image containing the temporal profiles obtained at iteration n and the projection data p , the ART objective function to be minimized is:

$$\epsilon = |p - \mathcal{P}(\mu_n)| \quad (2.24)$$

5. **A numerical algorithm:** an iterative algorithm was chosen, and in particular the Algebraic Reconstruction Technique. As previously introduced in Section 2.1 the advantage of this solution is the possibility to handle with uneven sampling or incomplete data and to be immune to non idealities, such as the metal artifacts. Moreover this class of algorithms is specially employed in CT reconstructions, where thanks to the high statistics, the statistical approach can be avoided. The ART iteration law takes the form:

$$\mu_{n+1} = \mu_n + \frac{(\vec{\mu}_n \cdot \vec{w}_i - p_i)}{\vec{w}_i \cdot \vec{w}_i} \cdot \vec{w}_i \quad (2.25)$$

where n is the current iteration number.

2.3 Problem Dimension Analysis

The problem dimension is defined by two factors mainly: the acquisition protocol and the dimensionality of the dynamic modeling. Concerning the acquisition protocol, the CBCT standard acquisition protocol of the O-ARM4 (Medtronic, Minneapolis,

Minnesota) [36] is taken as reference, and the acquisition parameters are listed in Table 2.1.

Table 2.1: Medtronic O-ARM4 acquisition parameters

Image dimension in voxels ($N_{voxel}^x, N_{voxel}^y, N_{voxel}^z$)	(512, 512, 192)
Voxel Spacing (w_{voxel})	0.415 mm
Slice thickness (t_s)	0.833 mm
FPD dimension in pixel ($N \times M$)	384×1024
FPD Pixel Size (w_{pixel})	0.388
Rotation Radius (R)	647.7 mm
Distance of FPD surface center from iso-center	520.7 mm
Number of Projections (N_θ)	391
Minimum Angle (θ_{min})	0°
Maximum Angle (θ_{max})	360°
Acquisition time (T_a)	12 s

2.3.1 Basis Functions Dimensioning

A brief study on the basis functions suitability for the contrast flow modeling was performed. Three basis functions sets composed of ten basis functions with different shapes were tested for contrast transit modeling, namely a triangular, a rectangular and a sigmoidal basis functions set (Figure 2.5). Several contrast flows were simulated with a sigmoidal trend as:

$$I(t) = \frac{I_{max}}{1 + e^{-k(t-t_{on})}} + I_{min} \quad (2.26)$$

where I_{min} and I_{max} are the minimum and maximum voxel intensity, which here were set to $I_{min} = 0$ and $I_{max} = 1$. Moreover the parameter k is a sigmoidal slope parameter randomly chosen between 0 and 1 for each contrast flow. The parameter t_{on} was randomly chosen as a time instant between 0s and the acquisition time expressed in seconds. Each randomly simulated contrast flow was reproduced with a linear combination of basis functions $q = [q_1, q_2, \dots, q_B]$, in order to get an estimated contrast flow $\hat{I}(t)$ equal to :

$$\hat{I}(t) = \sum_{i=1}^B d_i q_i(t) \quad (2.27)$$

where the terms d_i are the basis functions coefficients and B is the number of basis functions employed for the curve modeling. Enclosing all the time samples in a vector, Equation 2.27 can be rewritten in the vectorial form as:

$$\hat{\mathbf{I}} = \mathbf{d} \cdot \mathbf{q} \quad (2.28)$$

where $\hat{\mathbf{I}}$ is a $[1 \times N_s]$ vector (with N_s number of time samples), \mathbf{d} is a $[B \times 1]$ vector and \mathbf{q} is a $[N_s \times B]$ matrix.

The basis functions coefficients were computed starting from the randomly generated contrast flow \mathbf{I} as:

$$\mathbf{d} = \frac{(\mathbf{q})^{-1} \cdot \mathbf{I}}{(\mathbf{q})^{-1} \cdot \mathbf{q}} \quad (2.29)$$

so that the estimation of the randomly chosen contrast flow became:

$$\hat{\mathbf{I}} = \frac{(\mathbf{q})^{-1} \cdot \mathbf{I} \cdot \mathbf{q}}{(\mathbf{q})^{-1} \cdot \mathbf{q}} \quad (2.30)$$

Afterwards, for each time profile, the Root Mean Square error was computed as:

$$RMSE = \frac{\sqrt{\sum_{t=0}^{t=N_s} (I(t) - \hat{I}(t))^2}}{N_s} \quad (2.31)$$

This process was repeated for a set \mathbf{q}_t of triangular basis functions, a set \mathbf{q}_s of sigmoidal basis functions and a set \mathbf{q}_r of rectangular basis functions (Figure 2.5). The obtained median RMSE results for each basis functions set are reported in Table 2.2.

Table 2.2: Median RMSE values for TIC modeling with triangular, rectangular and sigmoidal basis functions

		Triangular	Rectangular	Sigmoidal
RMSE	Median	0.0263	0.0263	0.0051
	25-quantile	0.0261	0.0261	0.0047
	25-quantile	0.0264	0.0264	0.0058
	Standard Deviation	0.0018	0.0016	0.0010

It is easy to observe that the sigmoidal basis functions are the most suitable for a sigmoidal TIC modeling. Depending on the results of this study, the sigmoidal basis functions seemed to be the most appropriate. However, the coefficients computation through the matrix inversion (described in Equation 2.29) is remarkably different from their iterative computation based on the objective function minimization. Therefore, an additional simple trial was performed in order to test the three basis functions sets (\mathbf{q}_t , \mathbf{q}_s , and \mathbf{q}_r) suitability during the iterative TIC reconstruction. The iterative reconstruction of hundred temporal profiles was performed through the three basis

functions sets, and the RMSE errors were again evaluated on those profiles. The RMSE distribution was tested for normality through the Kolmogorov-Smirnov normality test, and the null hypothesis was rejected at the 5% significance level, meaning that the distribution proved not to be normal. Therefore, the median of the RMSE values was chosen as the reference index and it was computed starting from the whole set of RMSE values. The RMSE results from the iterative TIC reconstruction are reported in Table 2.3.

Table 2.3: Mean RMSE values for TIC modeling with triangular, rectangular and sigmoidal basis functions. TICs are computed with the iterative approach

		Triangular	Rectangular	Sigmoidal
RMSE	Median	0.1139	0.0220	0.0328
	25-quantile	0.0939	0.0206	0.0287
	75-quantile	0.1252	0.0229	0.0362
	Standard Deviation	0.0409	0.0059	0.0059

It easy to observe that the rectangular basis functions set is far more appropriate when dealing with the iterative TIC reconstruction. Therefore, basing on the combination of the two results the set of rectangular basis functions was chosen for the time profile modeling.

Once the shape of the basis functions had been chosen, the number of basis functions to be employed was investigated. Hundred randomly generated TICs were modeled using rectangular basis functions sets, containing a number of basis functions varying from 1 to 50. The RMSE trend among the TICs and the basis functions modeling was computed for each set and the RMSE trend with respect to the number of basis functions employed was inspected. The RMSE trend over the basis functions number (B) is shown in Figure 2.6. With reference to the obtained results a value of $B = 10$ was chosen. In fact, from $B=10$ to $B=17$ the RMSE value oscillates, therefore there is no need of choosing a B higher than 10 and smaller than 17. On the other hand, being $B=17$ the first value of B that guarantees a stable lower RMSE, choosing it would almost double the computational complexity with respect to $B=10$. Moreover a RMSE value of 0.02 was evaluated low enough for our purpose, and the first RMSE value under 0.02 is obtained with $B=10$. Therefore a basis functions set with 10 basis functions represents a good compromise between TIC modeling accuracy and low computational complexity. Moreover, as will be explained in Section 3.2, the criterium for arteries and veins separation is based on the computation of the TIC's Arterial Area Under the Curve (AUC_A), which is defined as the integral in time of the TIC curve from 0 to the

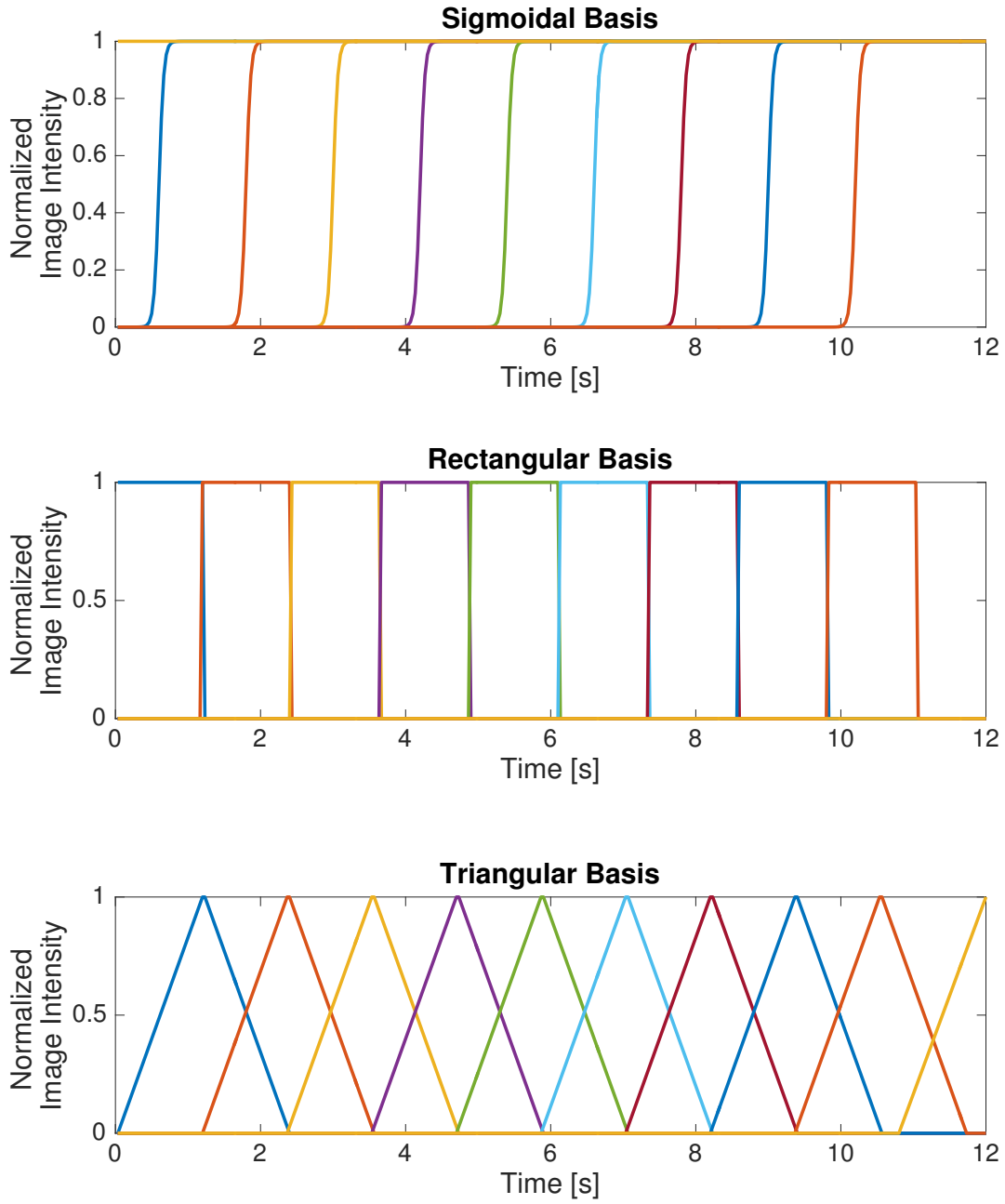


Figure 2.5: Basis Functions evaluated for the temporal profiles modeling. Top: Sigmoidal Basis Functions; Middle: Rectangular Basis Functions; Bottom: Triangular Basis Functions.

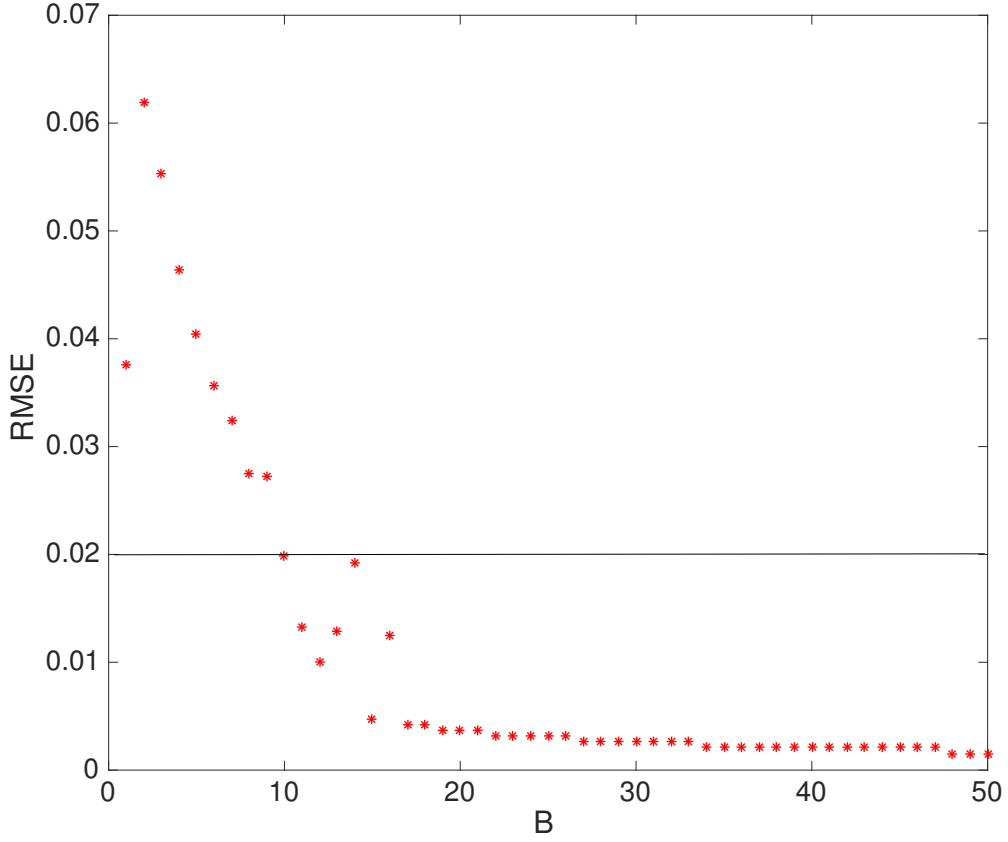


Figure 2.6: The trend of the median RMSE among the randomly generated TICs and their modeling through a rectangular set of B basis functions with a number B varying from 1 to 50.

first half of the Acquisition Time $T_a/2$. A threshold on the AUC_A with respect to total AUC of the TIC is imposed for arteries classification. It is highly probable that errors on TIC modeling would be made uniformly over the time samples, and therefore, being the AUC_A compared with the total AUC of the TIC, a small error in TIC modeling would not affect much the AUC_A thresholding for arteries classification.

2.3.2 LTV System Dimension Analysis

Here the LTV system dimension are analyzed for a hypothetic Medtronic O-arm real dataset, in order to investigate the feasibility of this methods on real patient datasets. Therefore the parameters listed in Table 2.1 are used to dimension the system. Some hypotheses must be firstly introduced. Since the complete dataset to be processed includes both a projection set before (p^0) and after (p^{ci}) the contrast injection, the final projection set (p) containing only the vessels contribution can be extracted as the

difference among them:

$$p = p^{ci} - p^0 \quad (2.32)$$

This leads to three hypothesis of this work.

Hypothesis 2.3.1.

A projection data p_i , at the angle θ_k , can be considered as the sum, along a projection line, of the contribution of all and only the voxels including contrast medium at time t_k .

Hypothesis 2.3.2.

All and only the voxels belonging to the vascular tree reconstruction are needed for the TIC reconstruction.

Hypothesis 2.3.3.

The voxels belonging to the vascular tree reconstruction represents approximately the 1% of the total number of voxels.

The confirmation of all the above hypothesis would lead to an important computational aid. In fact, an analysis limited to the segmented voxels shows a dramatic dimension reduction. In order to include only the needed voxels in the computation, the Digital Subtraction Angiography (See Section 1.2) reconstruction must be exploited. In fact, within the method of this thesis, only the voxels belonging to the DSA vascular tree reconstruction were considered. This dimensionality reduction allowed to expand the LTI System to the time dimension. The above hypotheses, if numerically confirmed, would assure that the dimensionality of the problem could be reduced at least of a factor $p/100$, where p is the percentage of voxels occupied by vessels. In order to have enough constraints for a good problem conditioning, the number of constraints I must be at least equal to the number of coefficients to be reconstructed $J \times B$, where J is the number of voxels and B is the number of coefficients to be estimated for each voxel. The number of available constraints is equal to

$$N_p = N \times M \times N_\theta \quad (2.33)$$

where N and M are the dimensions of the FPD in pixels, and N_θ is the number of acquired angular projections. N_p is then of an order N^3 . Choosing a number of basis functions B , the number of unknowns N_u becomes:

$$N_u = J \times B \quad (2.34)$$

where J is the number of voxels contained in the whole image cube. Therefore N_u is of an order N^4 and $N_u > N_p$, which makes the problem ill-conditioned. Nonetheless, since Hypothesis 2.3.2 implies that it is possible to take into consideration only vascular voxels, N_u becomes:

$$N_u = J \times B \times \frac{p}{100} \quad (2.35)$$

Moreover the third hypothesis (Hypothesis 2.3.3) on the percentage of vessel voxel with respect to the total number of voxels J , implies a dimension reduction of at least 100 times with respect to J , and this would bring down the order of N_c to N^3 , making the problem well conditioned.

Considering the clinical acquisition parameters listed in Table 2.1, here the problem dimension is numerically verified. Therefore, for a Medtronic O-arm system, the number of projection results:

$$N_p = N \times M \times N_\theta = 384 \times 1024 \times 391 \sim 153 \times 10^6 \quad (2.36)$$

and the number of unknowns is equal to:

$$N_u = J \times B = (512 \times 512 \times 192) \times 10 \sim 503 \times 10^6 \quad (2.37)$$

Exploiting the hypotheses 2.3.2 and 2.3.3 a forecasted reduced unknowns dimension is obtained:

$$N_u = J \times B \times \frac{p}{100} = (512 \times 512 \times 192) \times 10 \times \frac{1}{100} \sim 5 \times 10^6 \quad (2.38)$$

which is hundred times less than the initial number of constraints. Therefore, if the hypothesis would be verified, the problem would become well-conditioned. The hypothesis was verified on a real dataset. A vessel segmentation and reconstruction was obtained through a DSA in 3D Slicer 4.3.1 (Figure 2.7). The vessel reconstruction was analyzed and the voxels belonging to the vessel mask were counted resulting in an overall number of vessel voxels equals to 962573, which is approximately the 0,1 % of the total number of image voxels. Then, Hypothesis 2.3.3 is widely respected, proving the problem to be well-conditioned.

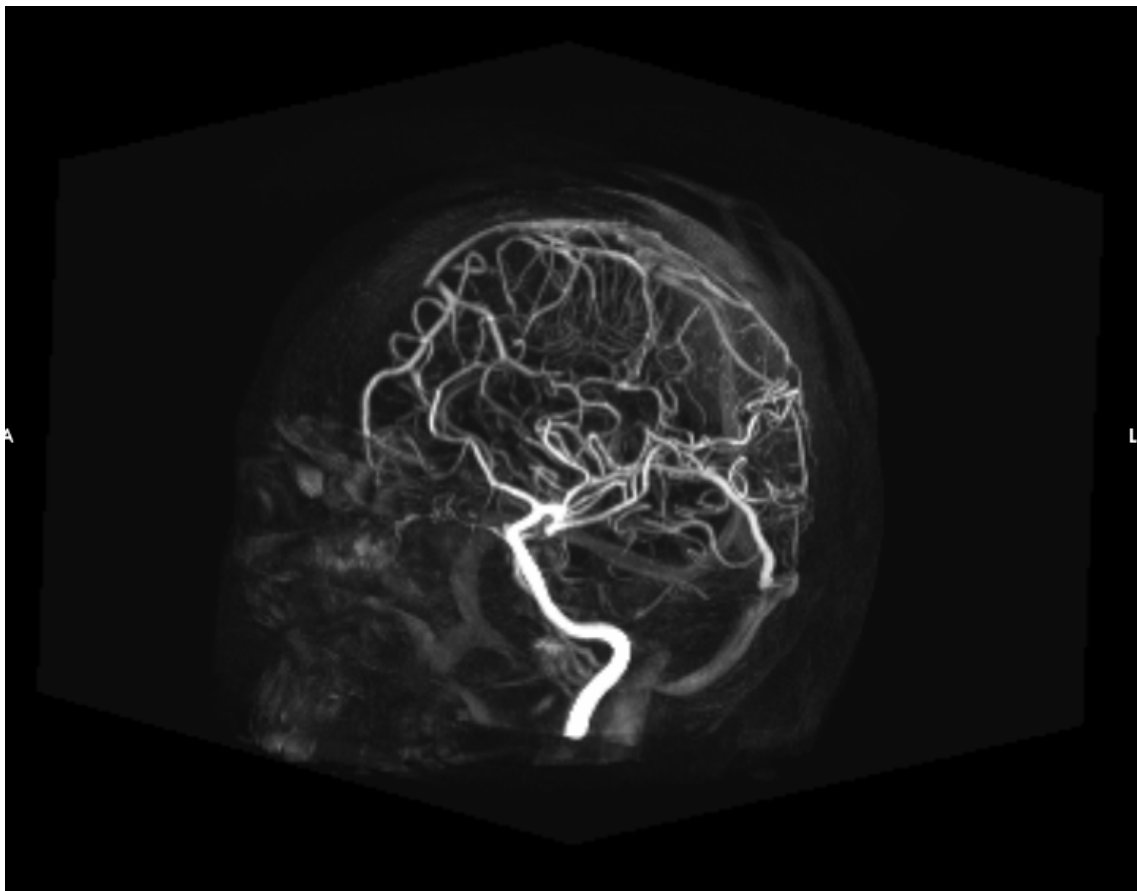


Figure 2.7: A 3D Slicer DSA vessel reconstruction.

Chapter 3

Simulation Protocol

The developed simulation protocol includes two main steps, namely the simulation design and the implementation of the ART 3.5 D Algorithm. The simulation design consist in the creation of a vessel tree phantom and in the simulation of the contrast flow in the phantom. Moreover, it includes the CBCT dataset creation through a simulated projection on the vessel tree phantom. The implementation of the ART 3.5D algorithm includes all the algorithmic steps presented in Section 2.2. However, due to computational and data storage issues some additional implementation optimizations were performed. Moreover the storage and computational issues were investigated and analyzed, leading in the end at several proposes for implementation improvements.

3.1 Simulation Design

A good simulation design is of primary importance in order to guarantee a good reproducibility passing from simulation to reality. Since the problem was verified to be well-conditioned all the parameters and computational quantities have to be simulated and reproduced proportionally to the real problem dimensions. Parameters can be divided in two classes:

- **Geometric parameters:** are all the geometric quantities linked with the O-arm geometric structure: N_{voxel}^x , N_{voxel}^y , N_{voxel}^z , N , M , w_{voxel} , w_{pixel} , t_s and R ;
- **Acquisition parameters:** are the parameters strictly linked with the simulation of the contrast injection in the 3D vessel structures: T_a , N_θ , θ_{min} , θ_{max} , $\Delta\theta$ and δ (the delay in contrast injection with to the instant of acquisition beginning).

All the letter notations were already listed in Table 2.1. In particular the geometric parameters must be chosen in order to guarantee an adequate sampling. The dimen-

sions were chosen respecting the relationships among the real O-arm parameters. The FPD have to be large enough to cover all the image volume and moreover it must have enough detectors to sample the image volume adequately. It is easy to verify that, in the real acquisition protocol, the number of FPD pixels in one generic direction is exactly twice the number of image voxel in the same direction.

$$\frac{N_{pixel}^x}{N_{voxel}^x} = \frac{N_{pixel}^y}{N_{voxel}^y} = \frac{1024}{512} = 2 \quad (3.1)$$

$$\frac{N_{pixel}^z}{N_{voxel}^z} = \frac{384}{192} = 2 \quad (3.2)$$

Therefore, the number of FPD pixels in each dimension must be chosen as twice the number of voxels in that direction.

Similarly, the ratio among the widths of FPD pixel (w_{pixel}) and image voxel (w_{voxel}) size is:

$$\frac{w_{pixel}}{w_{voxel}} = \frac{0.388}{0.415} = 0.93 \quad (3.3)$$

therefore the FPD pixel size must be chosen as the $w_{pixel} = 0.93 \cdot w_{voxel}$. The slice thickness (t_s) can be compared to the width of the voxel:

$$\frac{t_s}{w_{voxel}} = \frac{0.833}{0.415} = 2 \quad (3.4)$$

therefore t_s must be chosen in order to have $t_s = 2 \cdot w_{voxel}$

Also the rotation radius must be chosen appropriately and it can be compared to the length, expressed in millimeters, of the image cube along the radius direction. Therefore, if we consider that in the O-arm acquisition geometry the rotation radius R is equal to 647,7 mm, the ratio with the (x,y)-dimension of the image is

$$\frac{R}{L_{image}} = \frac{R}{N_{voxel}^{x,y} \cdot w_{voxel}} = \frac{647,7\text{mm}}{512 \cdot 0.415} = \frac{647,7\text{mm}}{212\text{mm}} = 3.05 \quad (3.5)$$

The geometric proportion among image volume and FPD obtained respecting this guidelines is qualitatively represented in Figure 3.1.

The chosen geometric and acquisition parameters are listed in Table 3.1 and 3.2 respectively. Cubic image volumes with several image dimensions were considered, with $N_{voxel}^x \times N_{voxel}^y \times N_{voxel}^z$ spanning from $10 \times 10 \times 10$ volumes to $30 \times 30 \times 30$ and several FPD dimensions were considered according to the image volume dimension, spanning from 20×20 up to 60×60 . Several Rotation Radius values were set depending on the image volume in order to guarantee a radius value equal to $R = 3 \times N_{voxel}^{x,y} \times w_{voxel}$ mm

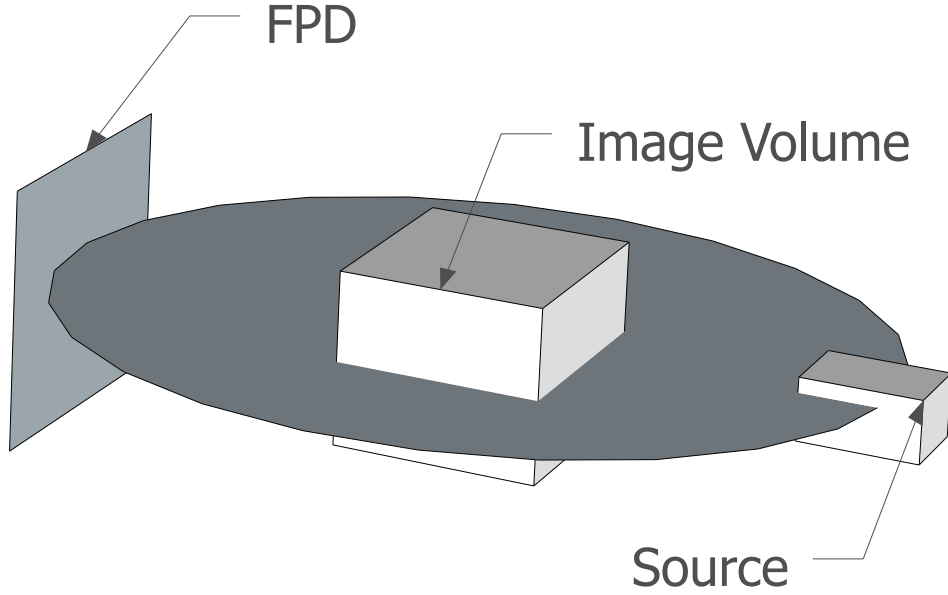


Figure 3.1: Proportionate schema of the O-arm acquisition geometry.

Another aspect that has been considered within the simulation protocol is the transit of the contrast medium bolus. Since the real acquisition protocol requires the continue injection of contrast medium in order to obtain enough contrast and visibility for the vessels, a continue injection was simulated, using an high Transit Time (T_t) value (up to 10 s), with 2 seconds delay δ in time injection. The contrast flow was simulated with two different dynamics. The dynamics represents the local contrast concentration change vs. time and was uniformly propagated along the length of the vessels considering a constant flow velocity, with no attempt of simulating low fluid-dynamics, which was out of our scope.

Firstly a simulation was performed using a simulated contrast flow with step dynamics. This choice was due to an initial simplification, for a first verification of the proper functioning of the method. Afterwards, the contrast flow was simulated as a sigmoidal trend (Figure 3.2), in order to simulate a real contrast transit dynamics. The sigmoidal slope was chosen through some experimental evaluations, in order to simulated a contrast flow as much realistic as possible. The sigmoidal and step minimum and maximum values were set to 0 and 1, in order to express contrast concentration and image intensity as a normalized value.

At last, the vessel 3D reconstruction was simulated. The vascular tree can be

Table 3.1: Simulated Geometric Parameters

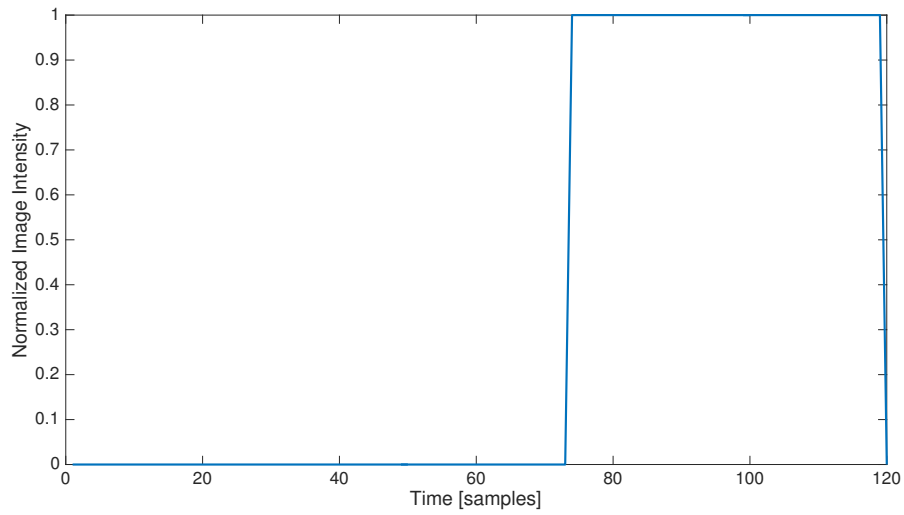
Image dimension in voxels ($N_{voxel}^x, N_{voxel}^y, N_{voxel}^z$)	spanning from (10, 10, 10) to (30, 30, 30)
Voxel Size (w_{voxel})	1mm \times 1mm
Slice thickness (t_s)	1 mm
FPD Pixel Size (w_{pixel})	0.93 mm
FPD dimension in pixel ($N \times M$)	$(2N_{voxel}^z) \times (2N_{voxel}^x)$
Rotation Radius (R)	$R = 3 \times N_{voxel}^{x,y} \times w_{voxel}$ mm

Table 3.2: Simulated Acquisition Parameters

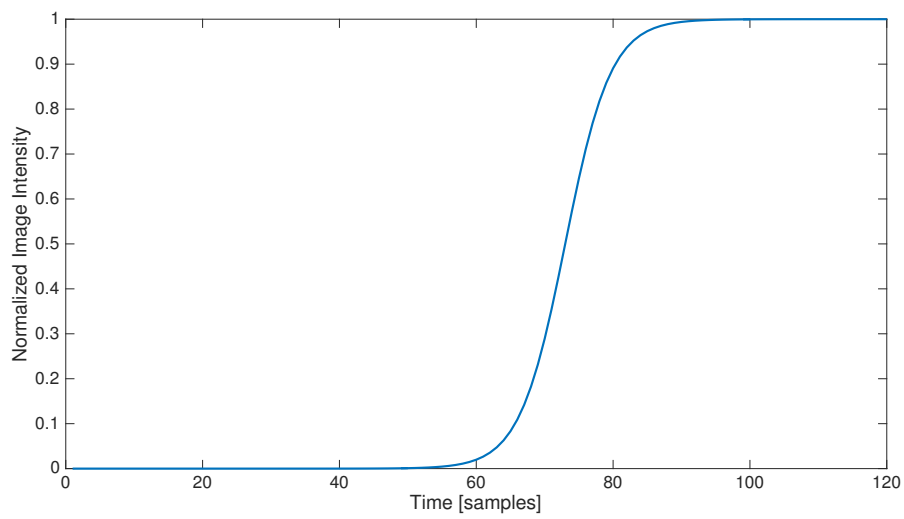
Number of Projections (N_θ)	120
Minimum Angle (θ_{min})	0°
Maximum Angle (θ_{max})	360°
Angular Resolution ($\Delta\theta$)	3°
Acquisition time (T_a)	12 s
Transit Contrast Time (T_t)	10 s

composed as the sum of different vessel tracts, all with the same diameter. The initial and final points of all the tracts must be specified, and all the tracts are then linked one another, giving the birth to the simulated vascular tree (further details in Appendix B). Two simple vascular tree shapes were simulated: one with five branches, and another one with seven, in order to be interpretable and simply verifiable to the human eye. The vessel tree shapes are shown in Figure 3.3. The vessel tree shown in Figure 3.3(a) is composed of a common artery branching in five vessels and subsequently gathering in a common vein again: this vessel tree will be named Vessel Tree A, for convenience. The vessel tree shown in Figure 3.3(b) differs from the previous one since two tracts crossing from one side to the other (*shunts*) are added: it will be called Vessel Tree B, for convenience. Due to obvious practical reasons, the simulation of a realistic tree with arterioles, capillaries, and venules was not attempted. Indeed, the shown branches may resemble artero-venous shunts, which exist only in artero-venous malformations. Nonetheless, this pattern, and the relevant dynamics of contrast transport within, was suited to our purposes and even more challenging than the physiological condition where a consistent delay separates the venous from the arterial compartment due to capillaries. Figure 3.4 shows some instants of the contrast flow inside the Vessel Tree A, during the whole acquisition time.

The simulated dynamic phantom was afterwards exploited for projection simulation,



(a) The contrast medium flow simulated with a step dynamic



(b) The contrast medium flow simulated with a sigmoidal dynamic

Figure 3.2: The dynamic of the contrast flow was simulated with two different dynamic behaviours

in order to obtain the desired projection dataset. The projection values were normalized as well, obtaining projection values spanning from 0 to 1. It has to be noticed that, since the simulated projections are already performed on the simulated dynamic DSA reconstruction (the dynamic phantom), the projection data already contain only vessel's information. Therefore, the subtraction among two datasets (contrast-enhanced and non contrast-enhanced) is not necessary in simulations.

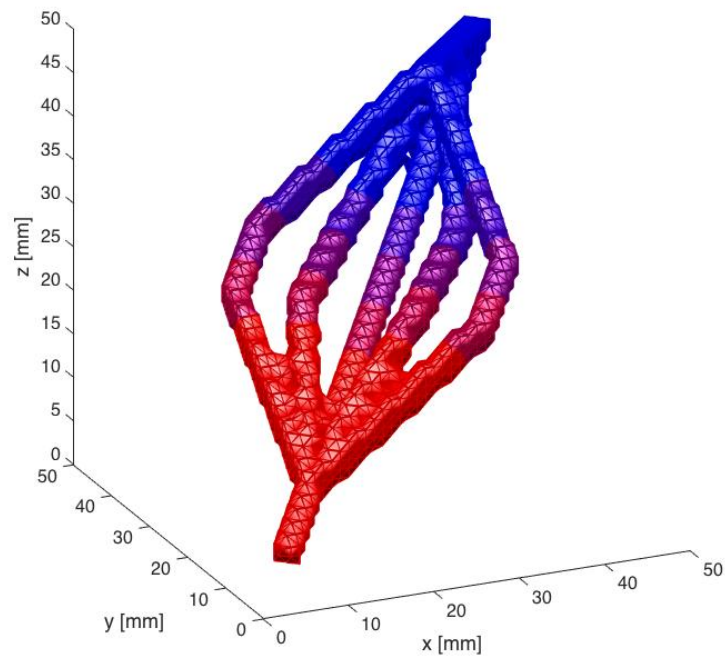
Four simulation setups were taken into account:

- Vessel tree A with Step contrast medium dynamic
- Vessel tree A with Sigmoidal contrast medium dynamic
- Vessel tree B with Step contrast medium dynamic
- Vessel tree B with Sigmoidal contrast medium dynamic

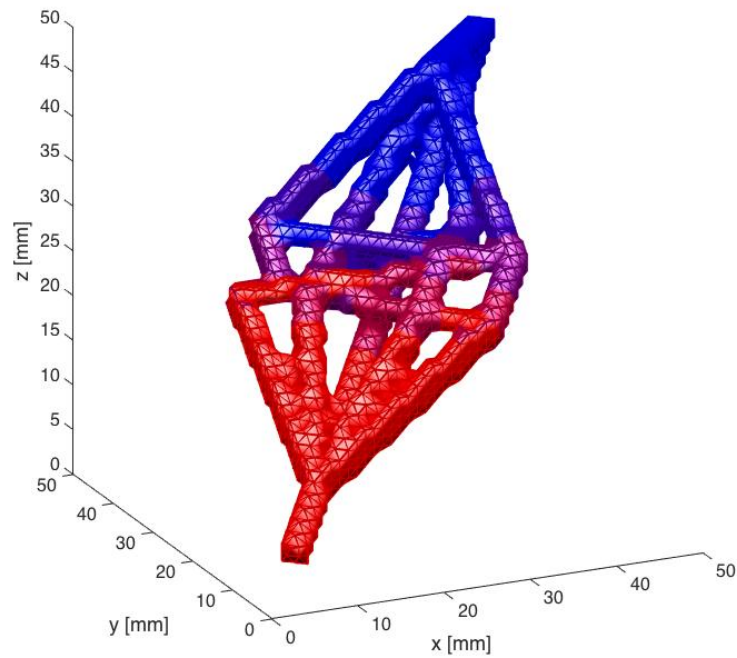
Each simulation condition was repeated for images with different dimensions. In particular image volumes with dimensions (in voxels) spanning from $10 \times 10 \times 10$ voxels to $30 \times 30 \times 30$ voxels were considered, and the FPD dimensions were varied accordingly. The dimensions of voxels and FPD pixels (expressed in mm) were kept constant. Moreover, the diameter of the vessels tracts was kept constant too. Therefore, over the different image volumes, the ratio among the vessels diameter and the vessel tree length varies. For each volume the iterative algorithm performed 10 iterations.

3.2 ART 3.5 D Implementation

In Figure 3.5 the implementation structure is illustrated. Once all the parameters have been chosen in order to respect the limitations listed in Section 3.1 the simulation can start, leading to the creation of the **simulated DSA reconstruction** and to a sequence of volumes representing the fraction of the vascular tree containing the contrast medium for each time sample. Subsequently a **simulation of the projection raw data** is obtained projecting, at each time sample, on the volume correspondent to that time sample. Once all the needed quantities have been simulated the algorithm can start processing. Since the ART needs the whole set of coefficients w_{ij} that weight each unknown j for a specific projection i , the System Matrix must be computed and stored. The **System Matrix computation** needs the geometric parameters as input and computes the system matrix in the static LTI case. However, in this case the dynamic system matrix is needed. The dynamic system matrix coincide to a system matrix in which the basis functions (known a priori) are inserted, leading to a matrix



(a) Vessel Tree A: The artery branches in five tracts and then gathers again in a vein



(b) Vessel Tree B: The artery branches in seven tracts, two of them crossing the vascular tree to one side to the other. All the vessels gathers again in a vein.

Figure 3.3: Two simulated vascular tree phantoms used for simulation.

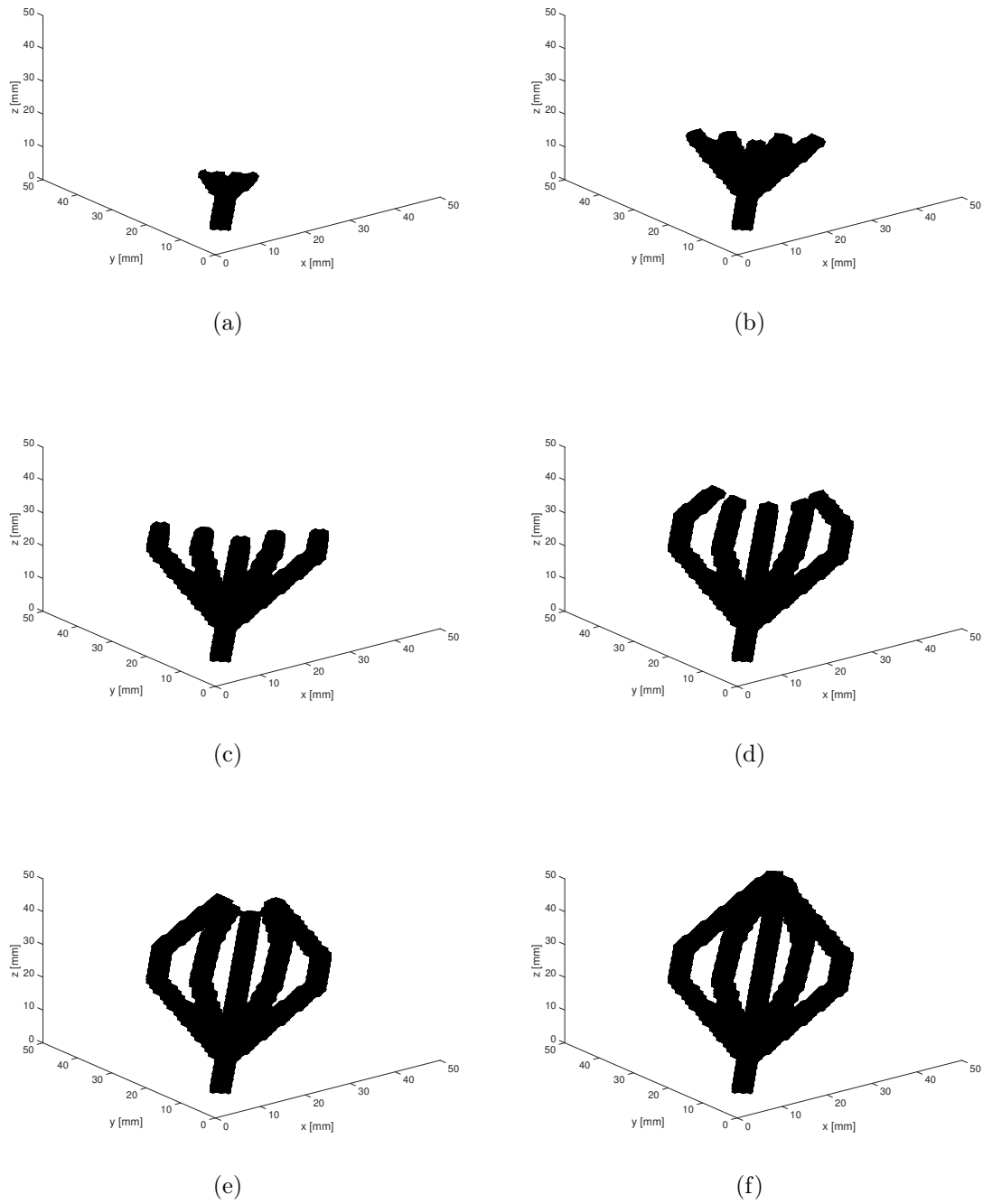


Figure 3.4: Sequence of images showing the simulated contrast flow in the vascular tree phantom A.

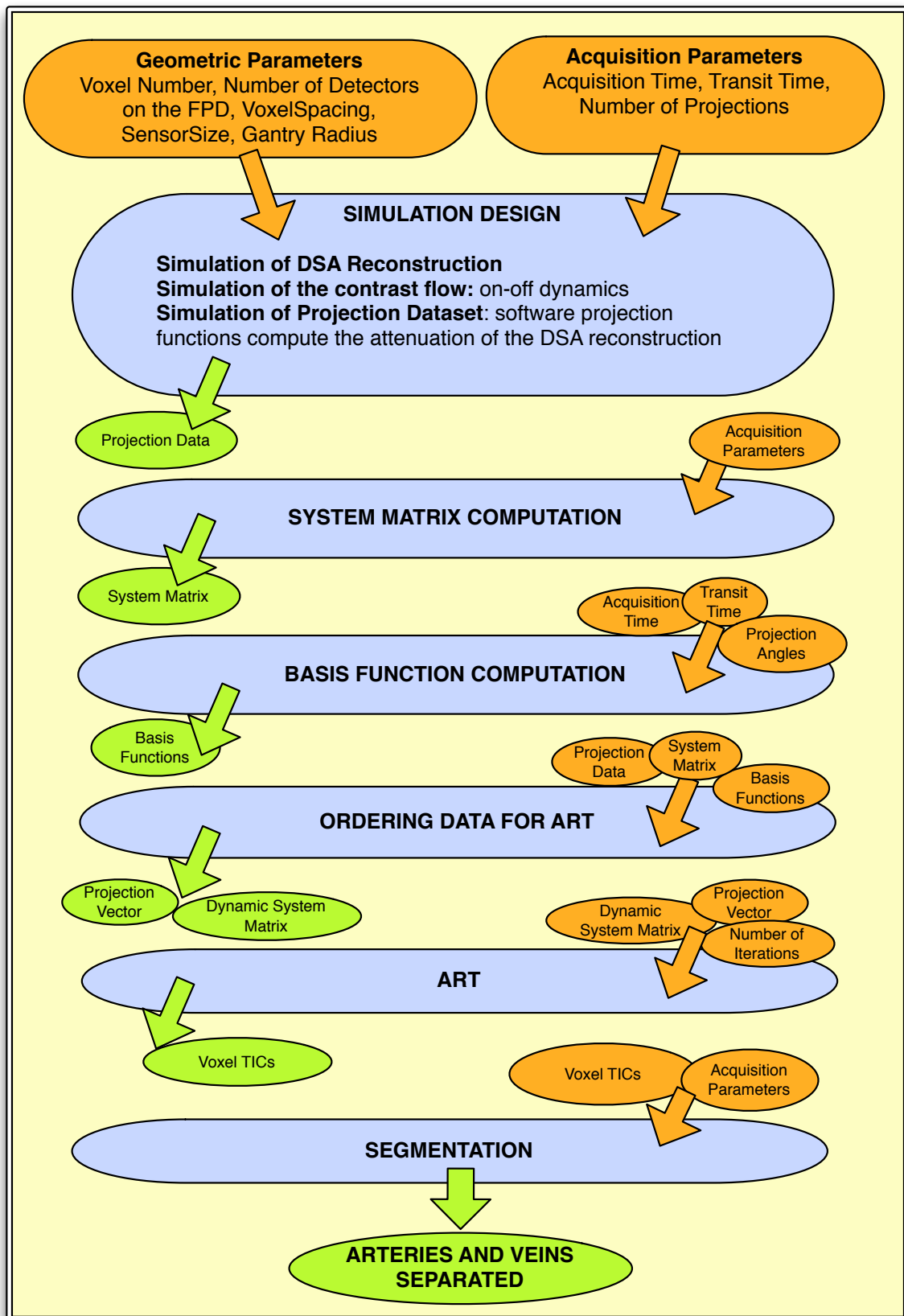


Figure 3.5: Block diagram showing the ART 3.5D algorithm structure. All the quantities in input at each block are orange colored, whereas all the quantities in output from each block are green colored.

expansion. Figure 3.6 outlines the System Matrix expansion procedure. Firstly, the **basis functions must be computed**. Afterwards, the vector of the basis functions samples at the time instant t_k is multiplied to each voxel's static weight for the projection θ_k . The basis functions bring dynamic information inside the matrix, making it suitable for the dynamic reconstruction. Subsequently, **data must be structured and organized** in order to make them suitable for the ART. In fact, only the pixels belonging to the vascular tree must be considered for reconstruction and all the projection data must be orderly stored in a Projection Vector. Once the Dynamic System Matrix and the Projection Vector are ready the number of iteration must be specified and the **ART algorithm can be launched**. The result of the ART is a vector containing, for each voxel, a number B of basis functions coefficients, being B the number of basis functions available. Those coefficients are multiplied to the basis functions, obtaining a TIC for each voxel. Once the voxel TICs are reconstructed the **segmentation of arteries and veins** block is entered. The voxels are classified depending on their temporal profile. Since the TICs reconstructed through the rectangular basis functions present a step like trend, a interpolation was introduced in order to make the TICs smoother. In particular, the medium point of the upper side of the rectangles were used as the reference data for interpolation (Figure 3.7(a)). Then, a number B of data were interpolated over the entire acquisition time T_a , leading to a complete smoother TIC. An example of the comparison among the reconstructed TIC and the interpolated TIC is shown in Figure 3.7.

Then, starting from TICs the Area Under the Curve (AUC) was computed as the integral of the TIC on the whole acquisition time T_a .

$$AUC = \int_0^{T_a} TIC(t)dt \quad (3.6)$$

Since the reconstructed curve was a discretization of a continuous profile, the AUC was actually computed a discrete integral, according to the rectangle method:

$$AUC = \sum_{i=1}^{N_t} TIC_i \Delta t \quad (3.7)$$

where N_t is the number of temporal instants and Δt is the sampling period. Moreover partial AUCs were computed as well: the AUC in the first half of Acquisition Time T_a was named arterial AUC (AUC_A) and the AUC in the second half of Acquisition

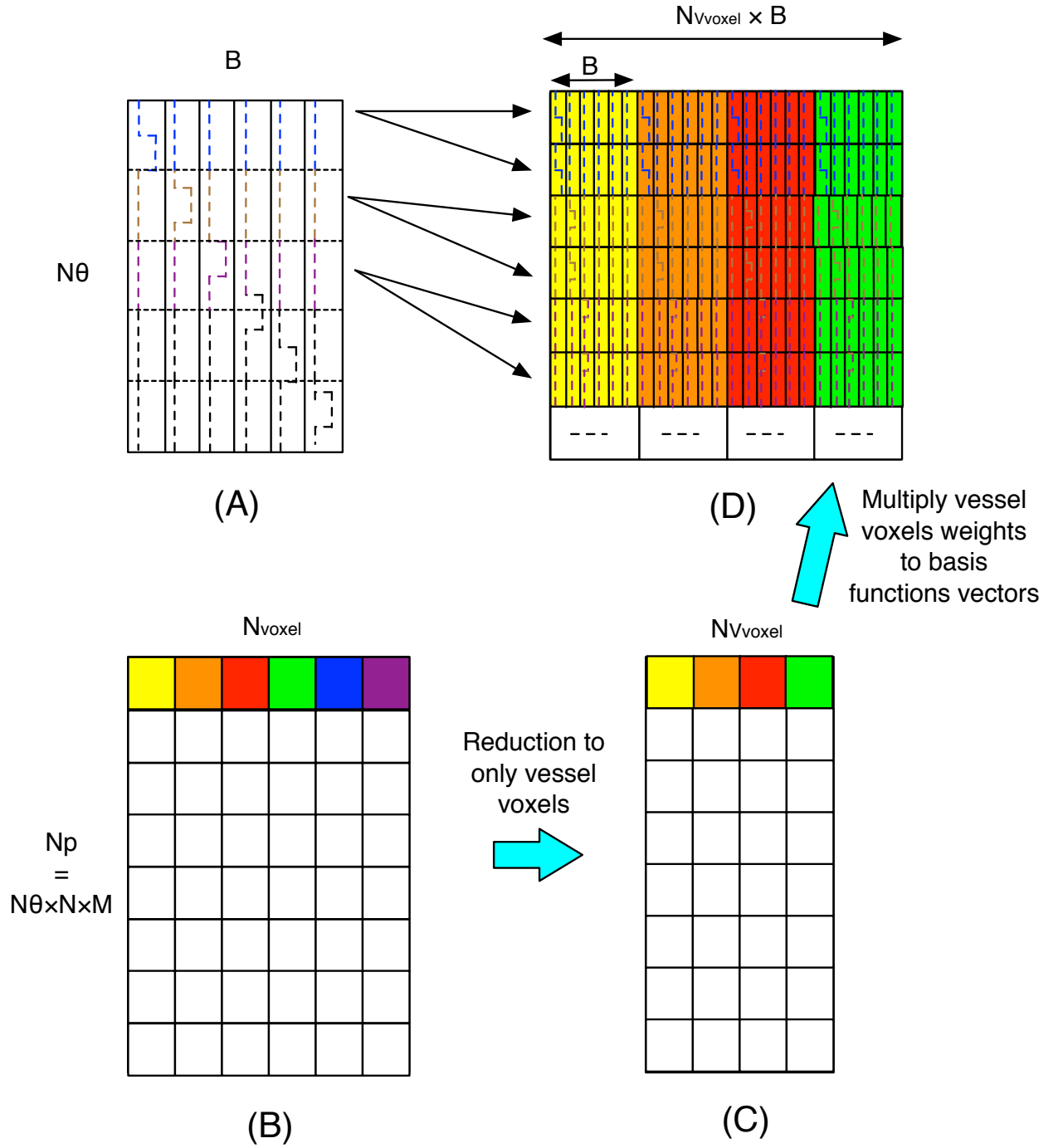
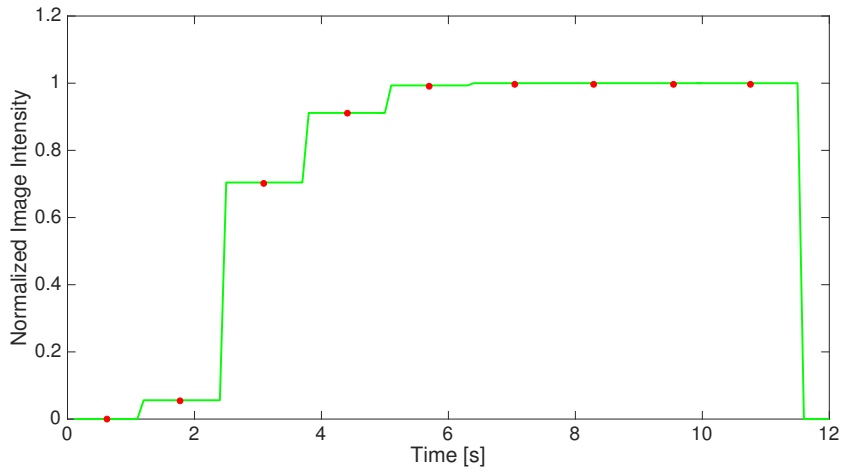
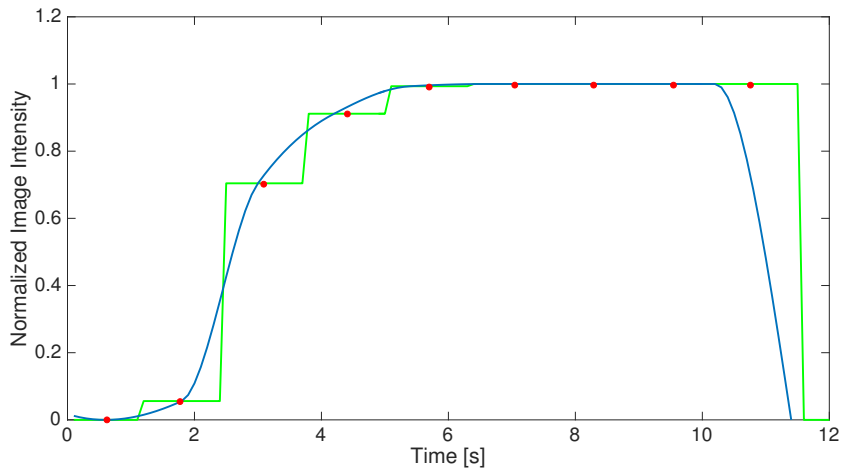


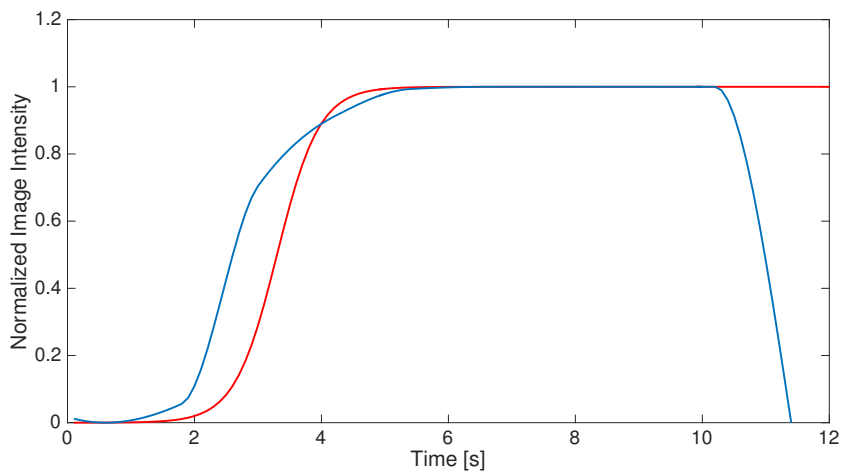
Figure 3.6: Computational process for the Dynamic extension of the System Matrix: block (A) represents the basis functions matrix, where the angular samples vary on the rows, while the values of different basis functions vary on columns; block (B) represents the static system matrix W ; the block (C) represents the static matrix reduced eliminating all the voxels that do not belong to the vascular tree, leaving a number $N_{V_{\text{voxel}}}$ of voxels; block (D) represents the Dynamic System Matrix, in which the basis functions have been inserted.



(a) Data used for interpolation are highlighted on the staircase TIC.



(b) The interpolated trend is shown over the staircase TIC.



(c) The interpolated trend is shown over the real voxel TIC.

Figure 3.7: Example of an interpolation process from a voxel's staircase TIC to a smoother sigmoidal trend.

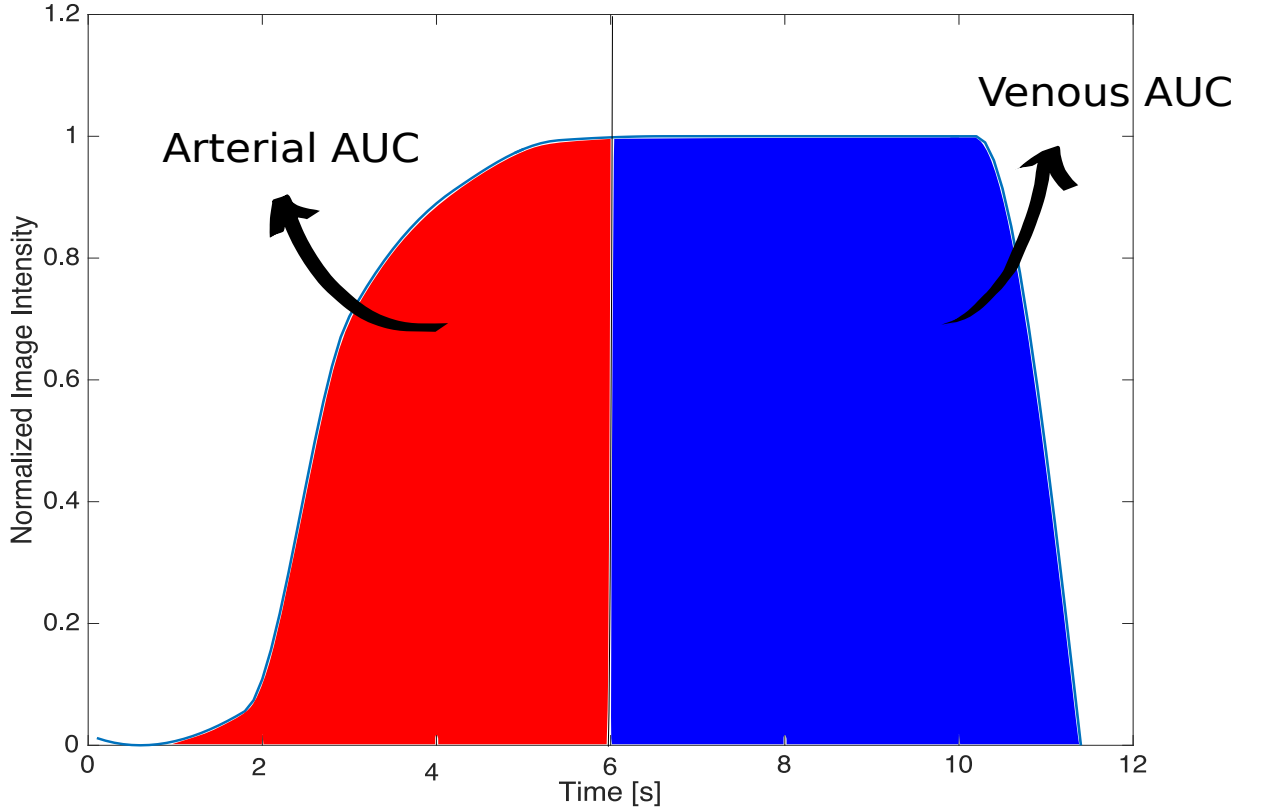


Figure 3.8: Graphical representation of the arterial AUC (AUC_A) and venous AUC (AUC_V)

Time T_a was referred to as the venous AUC (AUC_V) as shown in Figure 3.8.

$$AUC_A = \sum_{i=1}^{\frac{N_t}{2}} TIC_i \Delta t \quad (3.8)$$

$$AUC_V = \sum_{i=\frac{N_t}{2}}^{N_t} TIC_i \Delta t \quad (3.9)$$

Depending on the values of AUC , AUC_A and AUC_V the voxels were classified as artery or vein, as explained below:

$$\begin{aligned} \text{if } AUC_A > kAUC &\rightarrow \text{Artery} \\ \text{else} &\rightarrow \text{Vein} \end{aligned} \quad (3.10)$$

The value of k defines a threshold for arteries and veins classification and its optimal value was established through the use of Receiver Operating Characteristic (ROC)

curves. This classification strategy, which proved to work well in simulation, would probably need some improvements when dealing with real data. In fact, dealing with real cerebral vessel trees, including hundreds of vascular branches, a higher classification error has to be expected. Nonetheless, in future developments several improvements can be exploited, which were not introduced so far. In fact, the arterial and venous phase have deeply different dynamic properties and this can make suitable the insertion of regularizations within the modeling of the TICs. Moreover the capillary phase, which is intermediate among arteries and veins, can be detected and exploited for a most reliable separation of the dynamic behavior of arteries and veins. Capillaries can be detected as they present much lower intensity than arteries and veins during the contrast flow, due to blood and contrast dispersion in tissues which lowers the contrast medium concentration. Figure 3.9 shows three X-ray angiography images evincing the arterial, capillary and venous phase. These images visually prove by eye the possibility of arterial, capillary or venous phases distinction. Considering those experimental observation, if the capillary phase is detected, the voxels can be classified as arteries if their wash-in phase is positioned before the capillary phase, and as veins if their wash-in phase is positioned after the capillary phase. Moreover venous phase is as well much less intense than arterial phase, since much contrast medium has been lost during capillary phase, with a decrease in concentration, and this detail can be exploited for veins detection. In addition some spatial constraints can be introduced, in order to guarantee spatial continuity and avoid errors on isolated groups of voxels. For the interested reader, the implementation details are presented and examined in depth in Appendix B. The entire algorithm was implemented in Matlab 2015b on a Mac Book Air , 1.7 GHz Intel Core i7, 8GB DDR3 at 1600 MHz.

3.2.1 Memory Issues

Referring to the acquisition parameters listed in Table 2.1, and recalling that the system matrix stores for each voxel the amount of contribution to a specific projection value, we can compute the dimension of the static matrix for real data processing. Being N_θ the number of projection angles, $N \times M$ the number of sensors on a FPD, the number of rows of the system matrix r_W is:

$$r_W = N_\theta \times N \times M = 391 \times (1024 \times 384) \quad (3.11)$$

In the same way, being J the number of voxels in a reconstructed image, the number of columns of the system matrix coincides with the number of voxels:

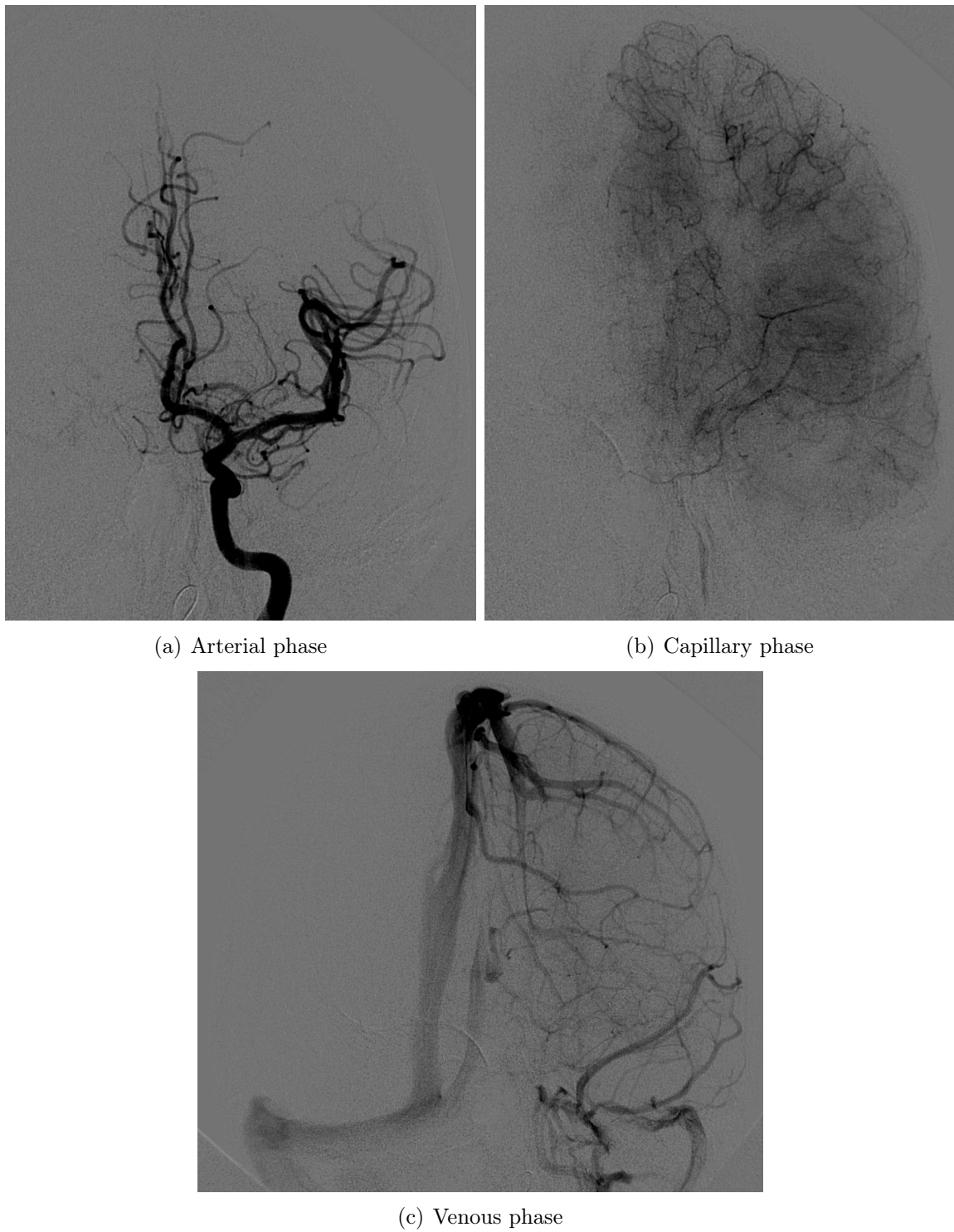


Figure 3.9: Comparison among arterial, capillary and venous phase in angiographic images

$$c_W = J = N_{voxel}^x \times N_{voxel}^y \times N_{voxel}^z = 512 \times 512 \times 192 \quad (3.12)$$

Moreover, recalling that due to the dynamic extension of the Algebraic Reconstruction Technique the basis functions are included in the system matrix, the dimensionality of the new dynamic system matrix is augmented of a factor B , with B being the number of chosen basis functions. Since a number $B = 10$ of basis functions was chosen, the number of columns c_W of the Dynamic System Matrix W_d is equal to:

$$c_W = J \times B = 512 \times 512 \times 192 \times 10 \quad (3.13)$$

However we can hypothesize that only a small percentage of the image voxel is occupied by vessels and, since the algorithm will be applied to only the vessel voxels, the dimension of the dynamic matrix is reduced. Here the conservative hypothesis of a 1% vessel volume is made, therefore the number of columns of the Dynamic System Matrix becomes:

$$c_W = J \times B \times 0,01 = 512 \times 512 \times 192 \times 0.01 = 5\,033\,164.8 \quad (3.14)$$

Therefore the final size of the Dynamic System Matrix is:

$$[W] = [N_\theta \times N \times M, J \times B] = [391 \times (1024 \times 384) ; 512 \times 512 \times 192] \quad (3.15)$$

$$= [153\,747\,456 ; 5\,033\,164.8] \quad (3.16)$$

Since each element of the matrix is represented in the computer memory as a double data type, each element fills 8 bytes. The total number of bytes ($\#Bytes$) allocated for the dynamic system matrix is:

$$\#Bytes = 153\,747\,456 \times 5\,033\,164.8 \times 8 = 6.19 \times 10^{15} \quad (3.17)$$

that equals

$$\#GigaBytes = 153\,747\,456 \times 50\,331\,648 \times 8 = 6\,190\,690GB \quad (3.18)$$

In order to solve this huge memory problem, the algorithm was adapted: only one line of the system matrix is computed at a time. In this way the total memory needed for a line is

$$\#Bytes = 50.331.648 \times 8 = 402\,653\,184 \quad (3.19)$$

that expressed in MegaBytes is

$$\#MegaBytes = 402.65 \quad (3.20)$$

Using this strategy the memory issue is solved but the computational time increases dramatically, since each line of the system matrix has to be computed for each update. This issue will be explained in Section 3.3.

3.3 ART 3.5 D Optimized Implementation

As anticipated in Section 3.2.1 the algorithm adopted in simulation is unsuitable for real data processing, due to heavy memory problems. Since the Dynamic System Matrix is too large, a new strategy must be adopted to make the algorithm suitable for a general computer. Recalling the general update law of ART:

$$\boldsymbol{\mu}_{n+1} = \boldsymbol{\mu}_n + \frac{(\vec{\boldsymbol{\mu}}_n \cdot \vec{\boldsymbol{w}}_i - p_i)}{\vec{\boldsymbol{w}}_i \cdot \vec{\boldsymbol{w}}_i} \cdot \vec{\boldsymbol{w}}_i \quad (3.21)$$

it is clear that just one constraint at once is employed for the update. Therefore just one line of the system matrix is needed at a time. The simplest strategy consists then in the computation of the needed line of the system matrix at every update. The obtained algorithm structure is shown in Figure 3.10. As can be observed in the Figure 3.10, here the geometric and acquisition parameters are known a priori, depending on the O-arm characteristics. Moreover also the DSA vascular tree reconstruction and the projected data are available. The projection data can be extracted as the difference among the contrast enhanced projection dataset and the non-contrast enhanced dataset acquired with the O-arm. The vascular tree reconstruction is obtained through the subtraction of the bone mask to the image volume reconstructed from the contrast enhanced scan. Then, the **computation of a number B of basis function** is needed. The iterative process starts without computing the static and dynamic matrix in advance. The process is executed for N_{it} times, being N_{it} the number of ART iterations. Inside each iteration, for $N_\theta \times N \times M$ times an update is computed (being $N_\theta \times N \times M$ the number of projection data available). Each projection data constitutes a constraint. For each constraint, the correspondent **line of the static system matrix is computed** and the voxel that do not belong to the vascular tree are eliminated. Subsequently, the static system matrix line is expanded to the 4_{th} dimension, becoming a **dynamic**

system vector. Thereafter, the **update term is computed** and the vector of the unknowns is adjourned, with the well known ART law. At the end of the process, a temporal image function is obtained and the TIC of each voxel is analyzed for the **arteries and veins segmentation** as explained in Section 3.2.

3.3.1 Computational Issues

Exploiting this new approach the computational time is heavily increased, due to the fact that the system vector is computed in real time with the ART update term. If the static matrix could be stored, the time employed for its computation would result outsize as well, but it could be computed just once and also, parallel computation would be a feasible solution, since each line would be computed independently from the others. Differently, with this new approach it must be computed line by line, for N_{it} times, where N_{it} is the iteration number. A brief study on the computational times was performed. The trend of computational time versus image dimension was studied. For uniformity, the overall computational time within one ART iteration was considered as the studied time quantity. The image dimensions were varied from 10 voxels to 30 voxels, with a step of one voxel among the trials. The obtained data were fit, through the curve fitting tool, (CFT) Matlab, in order to obtain the computational time for one iteration as a function of the image and FPD dimensions. Subsequently this function was exploited in order to forecast the computational time for one iteration on real data, with their specified dimensions. The obtained fitting curve is a power relation:

$$T_c = ax^b \quad (3.22)$$

were $a = 0.003626$ and $b = 3.275$, with 95% confidence bounds for a [0.001498, 0.005753] and b [3.096, 3.453]. The goodness of fit was evaluated through several indexes:

- SSE (The sum of squares due to error): is measures the error through the sum of the square errors. Having n data samples, and being t_{ci} the i^{th} data sample and $f(D_i)$ (D_i is the image dimension in voxels) the SSE is expressed as:

$$SSE = \sum_{i=1}^n (t_{ci} - f(D_i))^2 \quad (3.23)$$

For the evaluated fit ti has a value of $SSE = 665.9$

- R-square : it is the ration among the data variability and the correctness of the

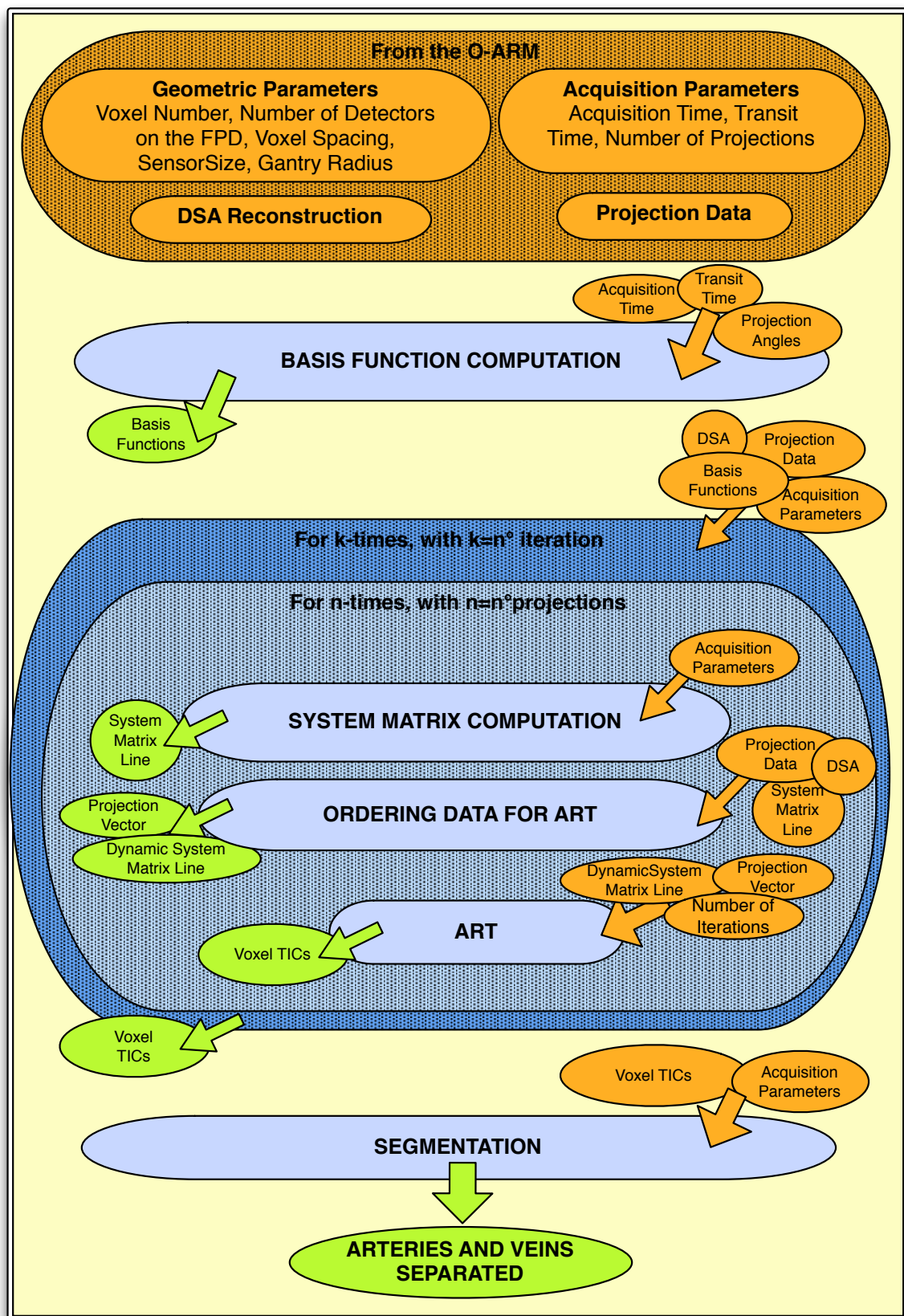


Figure 3.10: Block diagram showing the ART 3.5D algorithm optimized structure for real data processing. All the quantities in input at each block are orange colored, whereas all the quantities in output from each block are green colored.

proposed model. It is computed as

$$R_2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS} \quad (3.24)$$

where $ESS = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$ is the Explained Sum of Squares, $TSS = \sum_{i=1}^n (y_i - \bar{y})^2$ is the Total Sum of Squares and the $RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ is the Residual Sum of Squares. In all the expression y_i are the observed data, \bar{y} is the mean value of data and \hat{y}_i are the estimated data. A R^2 value equal to 0.9942 was obtained.

- RMSE (Root Mean Square Error), computed as :

$$RMSE = \frac{\sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2}}{n} \quad (3.25)$$

represents the mean sample standard deviation. A $RMSE$ value of 5.92 was obtained.

The obtained fit is illustrated in Figure 3.11.

The obtained model was used to produce an estimate of the computational time needed for processing a real patient dataset. It was obtained a real data iteration time T_{rdi} equal to:

$$T_{rdi} = 0.003626 \times 512^{3.275} = 2.705745357471414 \cdot 10^6 s = 31.31 \text{days} \quad (3.26)$$

3.4 Further Proposed Improvements for the ART 3.5 Optimized Algorithm

As reported in Section 3.3.1 the time needed for running the algorithm is very high. Nonetheless, in case the static matrix was computed just once before the running, the issue could be solved with parallel computing techniques. In fact, the computation of each system matrix line is independent from the computation of other lines and therefore it lends itself to a parallel computing approach. Then, the computational time T would be decreased proportionally with the number of cores used for parallel computing. Having a number of computational cores N_{core} the overall time becomes:

$$T_{parallel} = \frac{T}{N_{cores}} \quad (3.27)$$

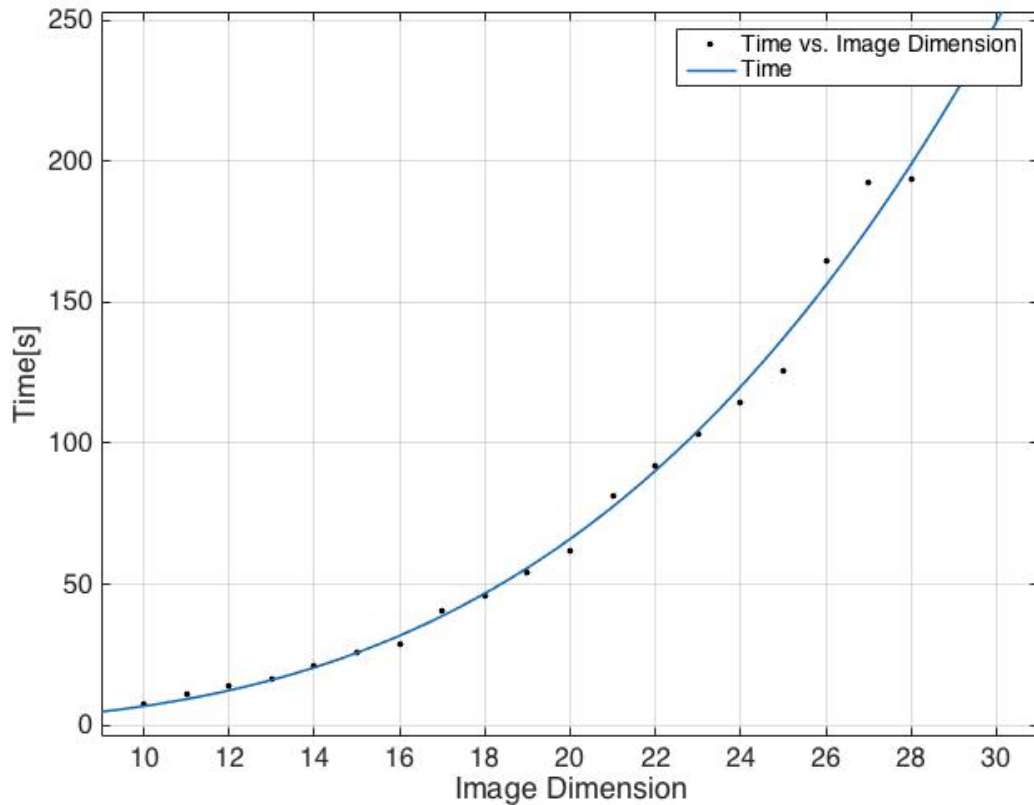


Figure 3.11: Fitting curve of computational time data. On the y-axis the computational time values in seconds for a single ART iteration. On the x-axis the image dimension expressed in voxels.

Differently, if the computation of the matrix line is embedded in the ART iteration (as it is in the real data adapted algorithm) standard parallel solutions are unsuitable, since each iteration depends on the previous iteration. Nonetheless, a certain number of approaches are available for the speeding up of iterative algorithms, many of them aiming at making the iterative process suitable for parallel computing techniques. Here some approaches will be briefly introduced. For interested readers, some reference are available in [1]. The use of subset is firstly presented, followed by a data dependency analysis approach.

Creating and using subsets

One possibility for parallel computation within iterative algorithms is the use of subsets. For *subset* it is intended a set of constraints, grouped together. Considering the ART algorithm, normally a correction term corresponds to one satisfied constraint, i.e. the algorithm satisfies one constraint at a time. It is common practice to organize the

constraints in subsets. The first introduction of the subset concept within the ART algorithm has lead to an approach called SIRT (Simultaneous Iterative Reconstruction Technique, [48]). Within this approach the voxel value is changed only after having explored all the equations and the final correction term is the average of the correction terms for that voxel. Therefore, the correction terms could be computed in parallel and then mediated. Of course many variations can be introduced: subsets can be smaller than the whole set of equations, and the dimension and composition of subsets can be richly varied depending on the needs. It is common practice to create the subsets in order to make them as much orthogonal as possible. In fact, the more the constraints are orthogonal to each other, the faster is the algorithm convergence. The introduction of subsets, although common practice, must be carefully evaluated, since the only introduction of subsets, tends to make the convergence slower than the standard iteration process. Nonetheless, within this work it might improve the performances, since it allows to compute in parallel a number I of correction, where I is the dimensionality of the subset. Only after the computation of all the I correction terms, they are mediated and the voxels are updated. Therefore the computational time can be reduced of a factor I . One disadvantage lies in the approximation introduced averaging the correction terms instead of applying them one by one. In order to repair to this disadvantage, an exact technique is proposed in the following section.

Data Dependencies analysis

In order to answer to the needs described above, an exact mathematical procedure can be exploited, in order to obtain a description of the data dependencies. The dependency analysis is a powerful technique that originates from advanced hardware acceleration techniques. In this work, the dependency analysis relies on a mathematical formal analysis of the iterative algorithm law, aiming at extracting a dependency law among data at different iterations [39]. Therefore, within a standard iterative algorithm starting from a general signal f , after the first iteration we obtain $f_1 = t(f)$ where t is the specific algorithm operation on the f signal. After the second iteration an f_2 signal is obtained as $f_2 = t(f_1)$ and so on. Generalizing on a casual iteration number n :

$$f_n = t(f_{n-1}) = T(f) \quad (3.28)$$

where the operator T encloses all the repetitions of the operator t . Within this framework a mathematical analysis can be applied in order to extract the operator T . Once the operator T is extracted, the iteration formula can be applied only once every n iterations, reducing the computational time of a factor n . In the specific case of ART,

the standard iteration μ_{n+1} is expressed as a function of the result of previous iteration μ_n :

$$\mu_{n+1} = \mu_n + \frac{(\vec{\mu}_n \cdot \vec{w}_i - p_i)}{\vec{w}_i \cdot \vec{w}_i} \cdot \vec{w}_i \quad (3.29)$$

which can be rewritten as:

$$\mu_{n+1} = \mu_n + \frac{\vec{\mu}_n \cdot \vec{w}_i \cdot \vec{w}_i}{\vec{w}_i \cdot \vec{w}_i} - \frac{p_i \cdot \vec{w}_i}{\vec{w}_i \cdot \vec{w}_i} \quad (3.30)$$

$$\mu_{n+1} = 2\mu_n - \frac{p_i}{\vec{w}_i} \quad (3.31)$$

In the same way the term μ_{n+2} can be written as:

$$\mu_{n+2} = 2\mu_{n+1} - \frac{p_{i+1}}{\vec{w}_{i+1}} \quad (3.32)$$

Recalling the Equation 3.31, immediately above we can substitute in Equation 3.32 the term μ_{n+1} expressed as a function of μ_n , getting:

$$\mu_{n+2} = 2 \left(2\mu_n - \frac{p_i}{\vec{w}_i} \right) - \frac{p_{i+1}}{\vec{w}_{i+1}} \quad (3.33)$$

Stepping again once:

$$\mu_{n+3} = 2\mu_{n+2} - \frac{p_{i+2}}{\vec{w}_{i+2}} \quad (3.34)$$

$$= 2 \left(2\mu_{n+1} - \frac{p_{i+1}}{\vec{w}_{i+1}} \right) - \frac{p_{i+2}}{\vec{w}_{i+2}} \quad (3.35)$$

$$= 2 \left[2 \left(2\mu_n - \frac{p_i}{\vec{w}_i} \right) - \frac{p_{i+1}}{\vec{w}_{i+1}} \right] - \frac{p_{i+2}}{\vec{w}_{i+2}} \quad (3.36)$$

Generalizing the above expression for $\mu_{n+k} = f(\mu_n)$ and optimizing it, an exact appropriate transformation T can be extracted.

$$\mu_{n+k} = 2^k \mu_n - \sum_{j=1}^k 2^{k-j} \frac{p_{i+j-1}}{\vec{w}_{i+j-1}} \quad (3.37)$$

This dependency among data can be exploited for making the algorithm suitable for parallel computation.

3.5 Evaluation Protocol

The experimental evaluation of the complete approach includes three evaluation steps:

- Evaluation of the TIC reconstruction
- ROC Curves Analysis and Evaluation
- Evaluation of the arteries and veins classification

Here the experimental conditions already introduced in Section 3.1 are recalled. In particular, two contrast flow dynamics were considered, namely a step dynamic and a sigmoidal dynamic. Moreover, two vessel tree phantoms were built and exploited for simulations. The contrast flow simulation and the vessel tree simulation were then combined in order to obtain six simulation setups, summarized in Table 3.3.

Table 3.3: Simulation setups summarized. For each of the two vessel tree shapes (A and B), two different contrast transit dynamics were simulated, namely a step dynamics and a sigmoidal dynamics.

	Vessel Tree	A	B
Contrast Dynamics		A1	B1
Sigmoidal dynamics		A2	B2

For each simulation setup, several image volumes with different dimensions in terms of voxels were simulated, in order to introduce some variability in the data. Varying the image dimensions in terms of voxels means to vary the ratio among the simulated vessels diameter and the total length of the vessel tree. The number of ART iterations performed (N_{it}) was set to 10. The classification was performed at each iteration step, in order to obtain informations on the convergence of the reconstruction algorithm.

3.5.1 Evaluation of the TIC reconstruction

The quality of the TIC reconstruction was evaluated through two error measures:

- ART Error Measure (ϵ) : it is the objective function ϵ minimized during the algorithm iterations. It is the difference (in terms of projection intensity) among the actual projection value p , and the projection value obtained with the simulated projection operator (\mathcal{P}), i.e. $\mathcal{P}(\mu_n)$, where μ is the reconstructed image at the n^{th} iteration.

$$\epsilon = \|p - \mathcal{P}(\mu_n)\| \quad (3.38)$$

- The Root Mean Square Error (RMSE) was computed for each reconstructed TIC (TIC_i) with respect to the simulated TIC (\overline{TIC}_i), for each image volume, at each iteration, as follows:

$$RMSE = \frac{\sqrt{\sum_{i=1}^n (TIC_i - \overline{TIC}_i)^2}}{N} \quad (3.39)$$

where N is the number of reconstructed TICs. In order to verify the distribution of the RMSE data, a normality test was performed on image volumes for each simulation setup. The Kolmogorov-Smirnov test was employed and the null hypothesis was rejected at the 5% significance level, meaning that the distribution proved not to be normal. Therefore, the median of the RMSE values was chosen as the reference index and it was computed starting from the whole set of RMSE values. Once a median RMSE value was obtained for each image volume the median value on the RMSE values on image volumes with different volume dimensions was computed, so that a median value for each iteration was obtained, and the convergence could be observed.

3.5.2 ROC Curves Analysis and Evaluation

The ROC curves were built in each simulation setup in order to investigate the value of k that, multiplied to the total AUC , defines the threshold of AUC_A for arteries classification. All the multiple values to 0.01 from 0 to 1 were considered among the possible thresholds. The classification on the image volume with dimensions $30 \times 30 \times 30$ was performed employing all the threshold values. For each classification the Sensitivity and the Specificity values were computed for both arteries and veins classification. The sensitivity is defined as:

$$SE = \frac{TP}{TP + FN} \quad (3.40)$$

The specificity is defined as:

$$SP = \frac{TN}{TN + FP} \quad (3.41)$$

Then, the ROC curve is defined as the SE values over the $(1 - SP)$ values. The ROC curve is built for both the arteries and veins classifications. The best parameter k for each simulation setup was then extracted as the k maximizing the SE and SP for both arteries and veins ROC curves. The k value that optimizes both SE and SP values for a

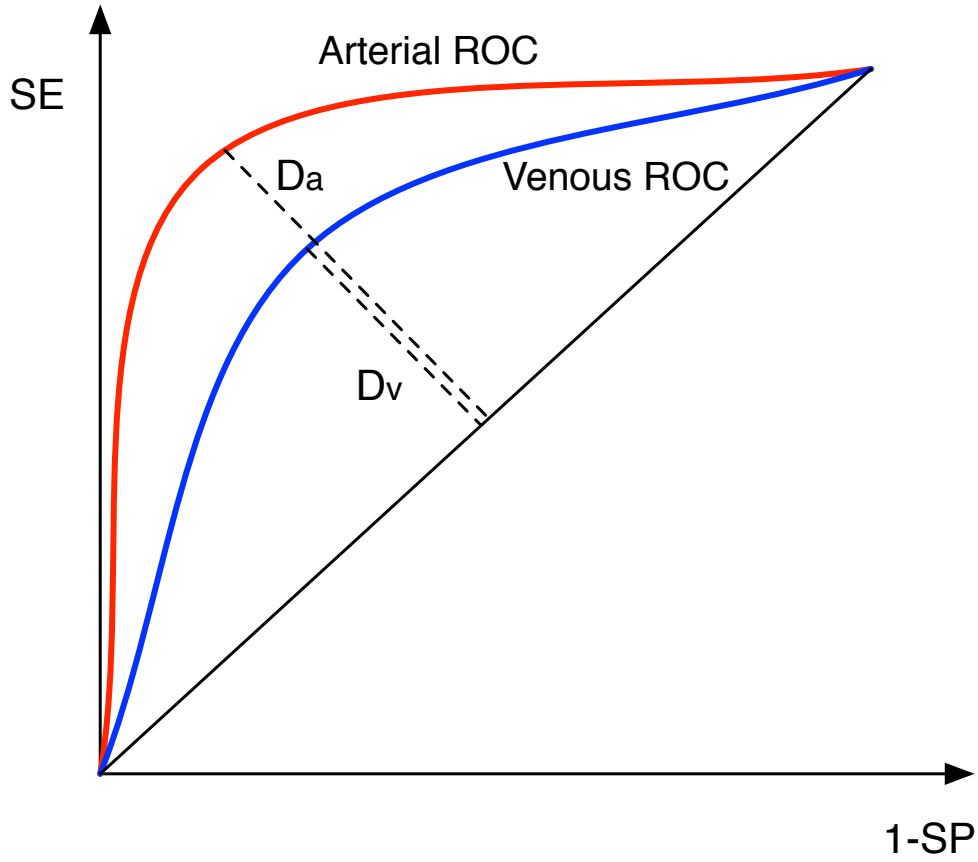


Figure 3.12: The maximum sum of the maximum distances between the arterial and venous ROC curves points and the random guess diagonal line points defines the best threshold parameters for arteries and veins classification. The red ROC curve is the arterial ROC; the blue ROC curve is the Venous ROC.

generical classification is identified by the ROC point that is orthogonally the furthest from the random guess diagonal line (i.e the diagonal passing from the points (0,0) and (1,1)). Here, the parameter that maximizes this distance for both the arterial and venous ROC curve is investigated. Therefore the parameter that maximizes the sum of the distance in the arterial ROC curve (D_a in Figure 3.12) and in the venous ROC curve (D_v in Figure 3.12) is chosen for the classification.

$$k = \underset{k}{\operatorname{argmax}}(D_a + D_v) \quad (3.42)$$

3.5.3 Evaluation of the Arteries and Veins Classification

The easiest way to evaluate the segmentation was to compute the percentage of voxels correctly classified. For both arteries and veins classifications the several indexes were computed. The value of those indexed spans from 0 to 1, where 1 means that the whole set of voxel falls in the evaluated condition and 0 means that not any voxel falls in the evaluated condition.

- True Positive (TP): the number of vessels correctly classified, expressed as

$$TP = \frac{C \cap S}{S} \quad (3.43)$$

where C are the classified vessels and S are the simulated vessels.

- False Negative (FN): the number of vessel incorrectly not classified as arteries or vein.

$$FN = 1 - TP \quad (3.44)$$

- True Negative (TN): the number of vessel correctly not classified as arteries or vein.

$$TN = \frac{(1 - C) \cap (1 - S)}{(1 - S)} \times 100 \quad (3.45)$$

- False Positive (FP): the number of vessel classified as artery or vein incorrectly

$$FP = 1 - TN \quad (3.46)$$

In addition, the sensitivity and specificity indexes were computed for each simulation setup, and the results were compared with results found in literature. Moreover, two additional indexes were evaluated, namely the fraction of arteries voxel incorrectly classified as veins (AV), the fraction of veins voxels incorrectly classified as arteries (VA). These very last indexes can correspond to the veins and arteries false negatives respectively. They have been investigated for information completeness.

- Arteries voxels incorrectly classified as veins, AV

$$AV = \frac{(C_A - S_A) \cap S_V}{S_A} \times 100 \quad (3.47)$$

where C_A are the classified arteries, S_A are the simulated arteries, and S_V are the simulated veins. The voxels incorrectly classified as arteries are compared

with the simulated veins mask. The intersection among those two quantities is the percentage of classified arteries that should have been classified as veins, with respect to the whole number of voxels classified as arteries.

- Veins voxels incorrectly classified as arteries, VA

$$VA = \frac{(C_V - S_V) \cap S_A}{S_V} \times 100 \quad (3.48)$$

where C_V are the classified veins, S_V are the simulated veins and S_A are the simulated arteries. The voxels incorrectly classified as veins are compared with the simulated arteries mask. The intersection among those two quantities is the percentage of classified veins that should have been classified as arteries, with respect to the whole number of voxels classified as veins.

Those indexes were computed for each reconstructed image volume in each experimental setup, and in each experimental set up the median value among all the images was investigated.

Chapter 4

Results

Here the obtained results are presented, namely the results on TIC reconstruction, the ROC curves results and the arteries and veins classification results.

4.1 TIC Reconstruction

The median values of the RMSE among the simulated and reconstructed TICs are listed in Table 4.1, together with quantiles values and the standard deviation. The values are quite homogeneous in the different experimental cases. In addition both the ϵ and the RMSE error trend with respect to the iteration number were investigated. Since ϵ is the ART minimized objective function, its monotonic decrease with an increasing number of iterations was expected. This results is verified (Figure 4.1). Moreover, the RMSE trends in Figure 4.2 shows that the RMSE decreases as well with increasing iteration numbers, notwithstanding it is not minimized by the iterative algorithm. This is a very positive results, since it confirms that the error on TIC reconstruction is minimized from the ART algorithm. Therefore, we can conclude that the ART 3.5D is a very suitable algorithm for TIC reconstruction. As can be seen in Figures, in the most of the TIC reconstruction improvements are achieved in the first iteration. However, inspecting the numerical values, after the first iteration some small oscillations on the RMSE values were noticed. This result was quite unexpected, since increasing the number of iterations, the solution should converge to a more stable result. However, the corrections during the iterative process are made on the ART error ϵ . Therefore the hypothesis is that the adjustments on the ART error ϵ induce continuos adjustments on coefficients and this cause some coefficients oscillations, that have repercussions on the TICs trend. This hypothesis is verified looking at the trend of the error index ϵ (Figure 4.1) with increasing iterations in comparison with trend of the RMSE error. In

fact, the ART error computed for the update during the iterative process monotonically decreases at each step, being almost constant after the second or third iteration.

Table 4.1: Statistical values of RMSE among the simulated and reconstructed TICs for each simulation setup. A1, A2, B1 and B2 are the simulation setups names listed in Table 3.3.

RMSE Statistical Measures	Simulation Setup			
	A1	A2	B1	B2
Median	0.0376	0.0367	0.0366	0.0368
25-quantile	0.0373	0.0367	0.0364	0.0366
75-quantile	0.0378	0.0368	0.0369	0.0369
Standard Deviation	$5.1530 \cdot 10^{-4}$	$1.1698 \cdot 10^{-4}$	$2.7876 \cdot 10^{-4}$	$2.6550 \cdot 10^{-4}$

4.2 ROC Curves Analysis and Evaluation

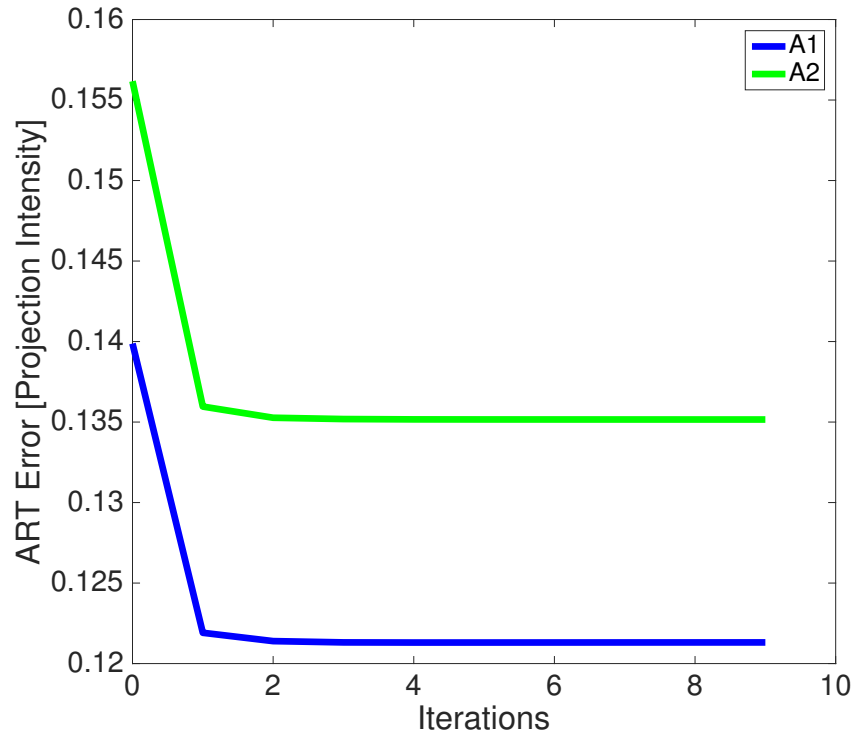
The Receiver Operating Characteristic (ROC) curve was built for each simulation setup in order to inspect the right threshold parameter k for the Arterial Area Under the Curve (AUC_A) computation for the classification. The ROC curves are shown in Figure 4.3. The coefficients that, multiplied to the total Area Under the Curve, give the threshold values for the AUC_A in each experimental condition are listed in Table 4.2. Those values were used for the segmentation.

Table 4.2: Value of the threshold parameter k (multiplied to the total AUC for arteries classification) for each simulation setup.

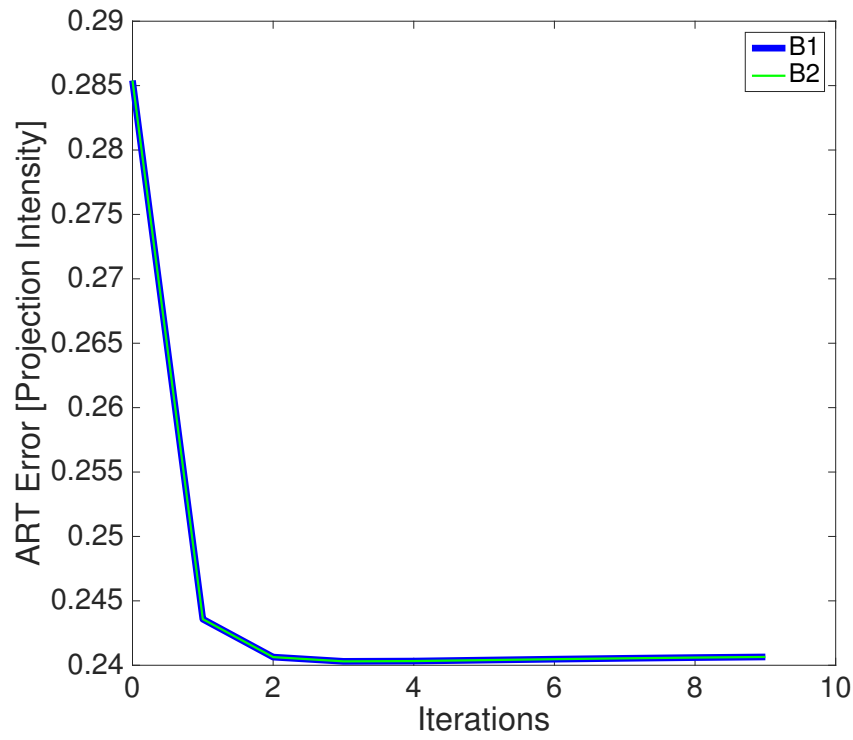
Simulation Setup	k
A1	0.1900
B1	0.2600
A2	0.1600
B2	0.1500

4.3 Arteries and Veins Classification

The results of the segmentation were inspected. The trend of the arteries and veins TP, the AV index and the VA index with an increasing number of iterations for all the simulation setup is shown in Figures 4.4, 4.5, 4.6 and 4.7. As it is possible to see,

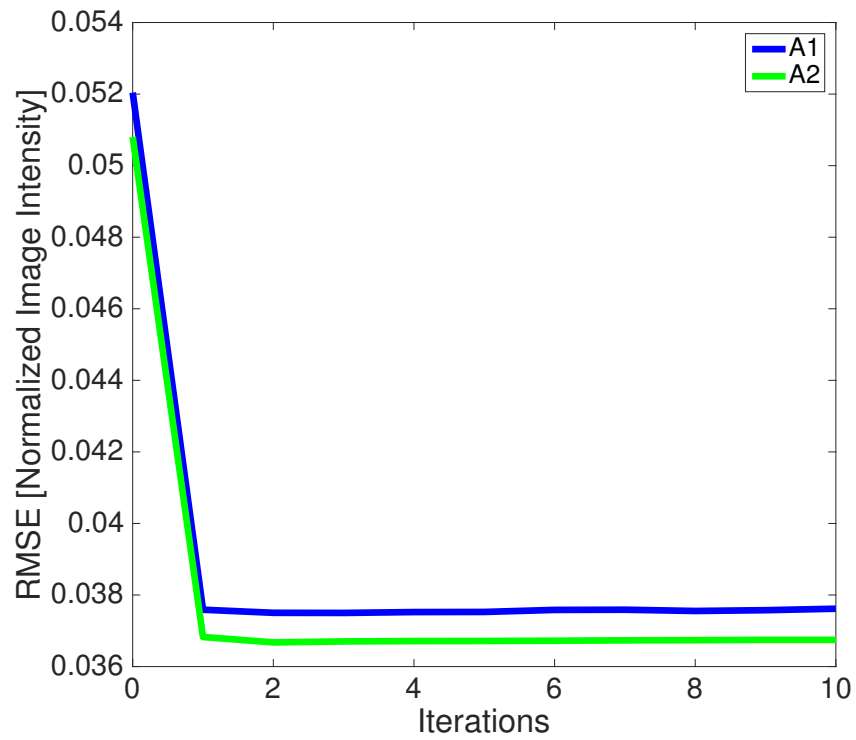


(a) Vessel Tree A

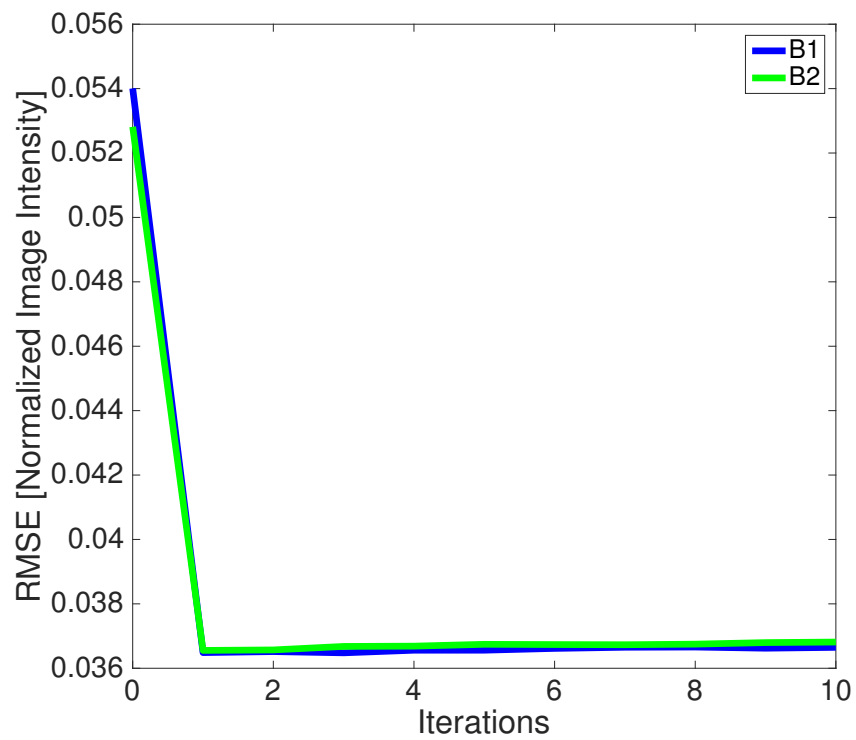


(b) Vessel Tree B

Figure 4.1: Median ϵ index trend over iterations, for each simulation setup.



(a) Vessel Tree A



(b) Vessel Tree B

Figure 4.2: Median TICs RMSE trend over iterations, for each simulation setup.

Table 4.3: True positive, false positive, true negative and false negatives for arteries classification are shown for each simulation setup.

Arteries Classification		Simulated	
		P	N
A1	P	1.0000	0.0035
	N	0	0.9965
A2	P	1.0000	0.0003
	N	0	0.9997
B1	P	0.9737	0.0063
	N	0.0263	0.9937
B2	P	0.9950	0.0036
	N	0.0050	0.9964

the most of the classification improvements are achieved in the first iteration, while in the subsequent iterations some oscillations occurs. This is in agreement with the hypotheses presented above, since the best classification results are obtained at the second or third iteration as well. Moreover it has to be noticed that the oscillations are small enough to be neglected, being up to 0.01 maximum.

Arteries and veins median classification results are shown in Tables 4.3, 4.4, 4.5. Table 4.3 shows true positive, false positive, true negative and false negatives for arteries classification. In the same way Table 4.4 shows true positive, false positive, true negative and false negatives for veins classification. The simulation setups including sigmoidal contrast dynamics show better results. This can be considered a good result, meaning that the sigmoidal contrast dynamics profiles, that are the most realistic ones, are reconstructed with better accuracy. Therefore, this reconstruction method seems suitable for voxels TIC reconstruction.

In addition, Sensitivity and Specificity values for each simulation setup are listed in Table 4.6. Those results can be compared to results found in literature: in fact, Mendrik et al. [37] presented sensitivity and specificity values for their automatic arteries and veins segmentation procedures on real patient data. Their ground truth was based on the labeling of two expert observers. They presented sensitivity values equals to 0.928 and 0.958 (depending on the observer used as ground truth) and specificity values equal to 0.985 and 0.964. Their results confirmed that the method was more accurate than the observers themselves. The sensitivity and specificity results obtained in this thesis seem to be comparable to those obtained in literature, though with better

Table 4.4: True positive, false positive, true negative and false negatives for veins classification are shown for each simulation setup.

Veins Classification		Simulated	
		P	N
A1	P	0.8795	0
	N	0.1205	1
A2	P	0.9885	0
	N	0.0115	1.0000
B1	P	0.8893	0.0005
	N	0.1107	0.9995
B2	P	0.9216	0.0001
	N	0.0784	0.9999

Table 4.5: Arteries correctly classified, veins incorrectly classified as arteries, arteries incorrectly classified as veins and correctly classified veins are shown for each simulation setup.

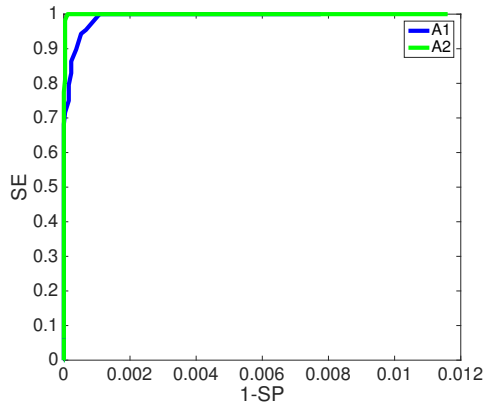
Arteries and Veins Classification		Simulated	
		A	V
A1	A	1	0
	V	0.1205	0.8795
A2	A	1.0000	0
	V	0.0115	0.9885
B1	A	0.9737	0.0263
	V	0.1107	0.8893
B2	A	0.9950	0.0050
	V	0.0784	0.9216

Table 4.6: Sensitivity and Specificity values for each simulation setup.

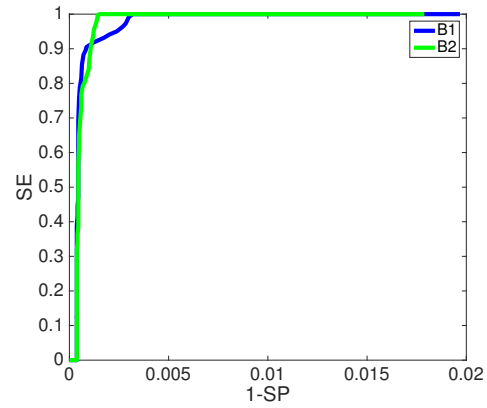
Arteries and Veins SE and SP		SE	SP
A1	Arteries	1	0.9965
	Veins	0.8795	1
A2	Arteries	1	0.9997
	Veins	0.9885	1
B1	Arteries	0.9737	0.9937
	Veins	0.8893	0.9995
B2	Arteries	0.9950	0.9964
	Veins	0.9216	0.9999

performances in sensitivity than in specificity, differently from the results presented in [37]. Nonetheless, a real comparison is not possible, since the presented results are extracted from extremely simple simulations, not pretending to be realistic enough to make possible any precise comparison and result evaluation. However, there are many elements in support of possible reliable classifications on real data-sets through the ART 3.5D algorithm. One of them consists in the wide temporal gap represented by the capillary phase of contrast transit. Since the contrast medium is rapidly washed in through the arteries, virtually disappearing and expanding in the huge network of unresolved capillaries, and next reappearing in veins for the final rapid wash out, the arterial and venous phase can be easily distinguished investigating the capillary phase. These considerations, not yet exploited in our algorithm, would be precious when dealing with real data, when an higher classification error is expected. Future work, will be performed in order to apply the algorithm on a real dataset, as explained in Section 3.4 and Chapter 5.

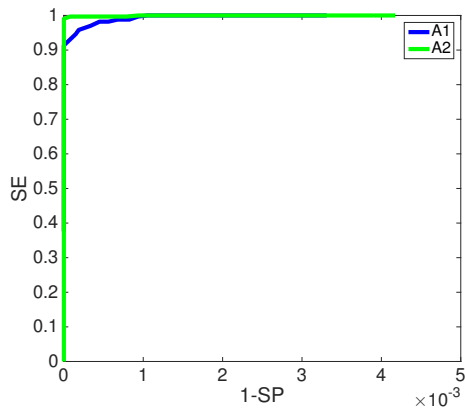
Figure 4.8 qualitatively shows the reconstructed arterial tree in comparison with the phantom arterial tree. Figure 4.9 qualitatively shows the reconstructed venous tree in comparison with the phantom venous tree.



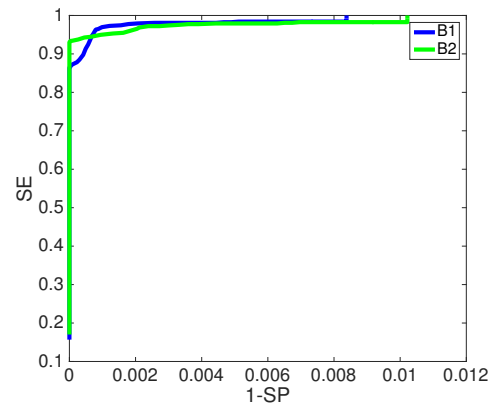
(a) Arteries, Vessel Tree A



(b) Arteries, Vessel Tree B

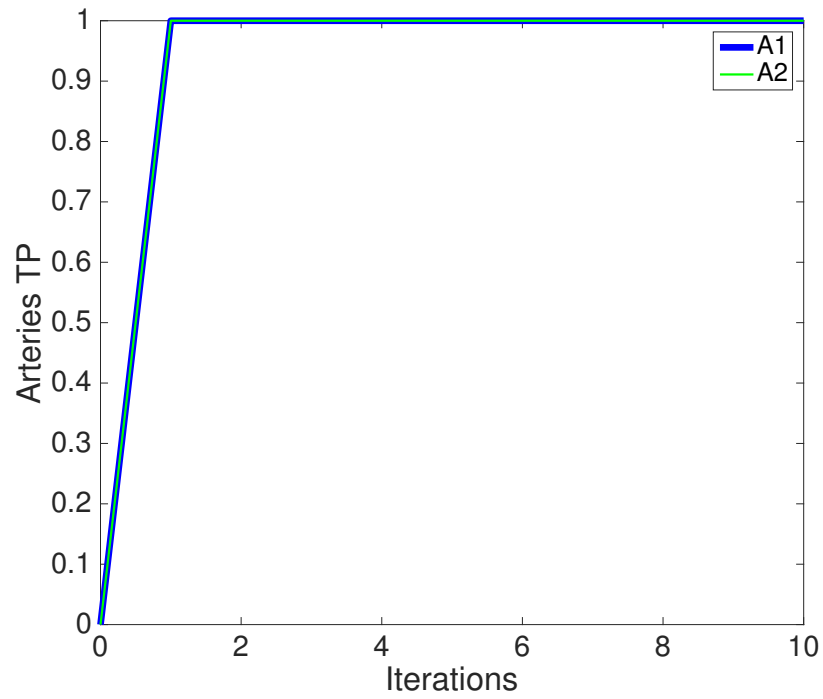


(c) Veins, Vessel Tree A

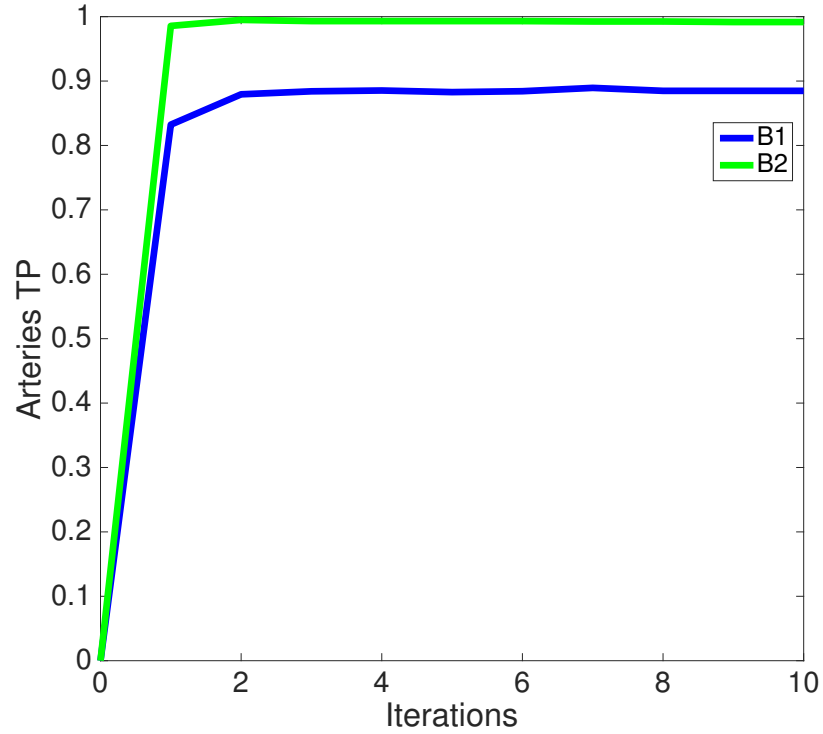


(d) Veins, Vessel Tree B

Figure 4.3: Receiver Operating Characteristics (ROCs) for arteries and veins classification. Left: ROC curves for the Vessel Tree A; Left: ROC curves for the Vessel Tree B

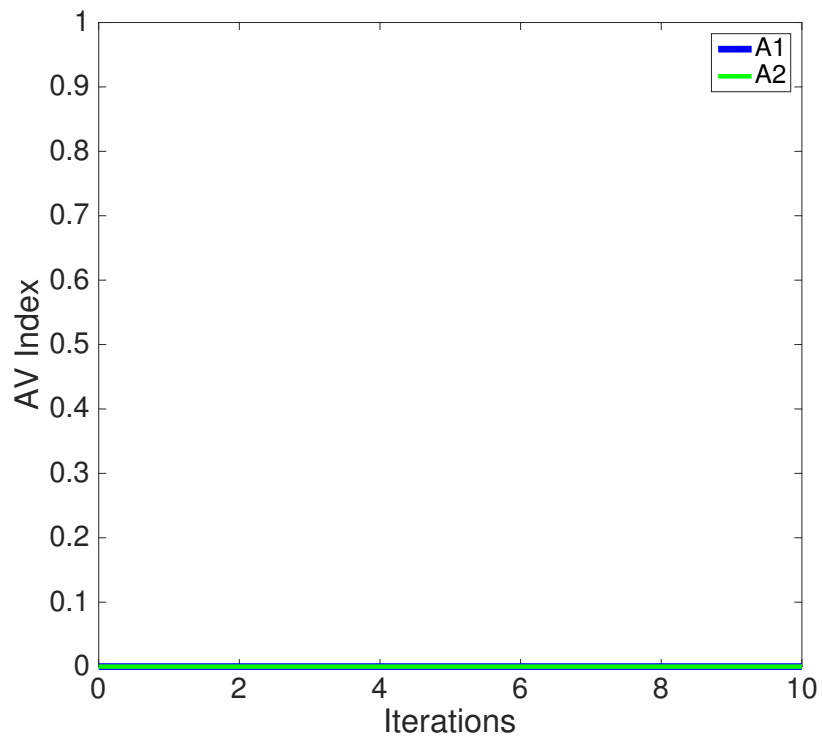


(a) Vessel Tree A

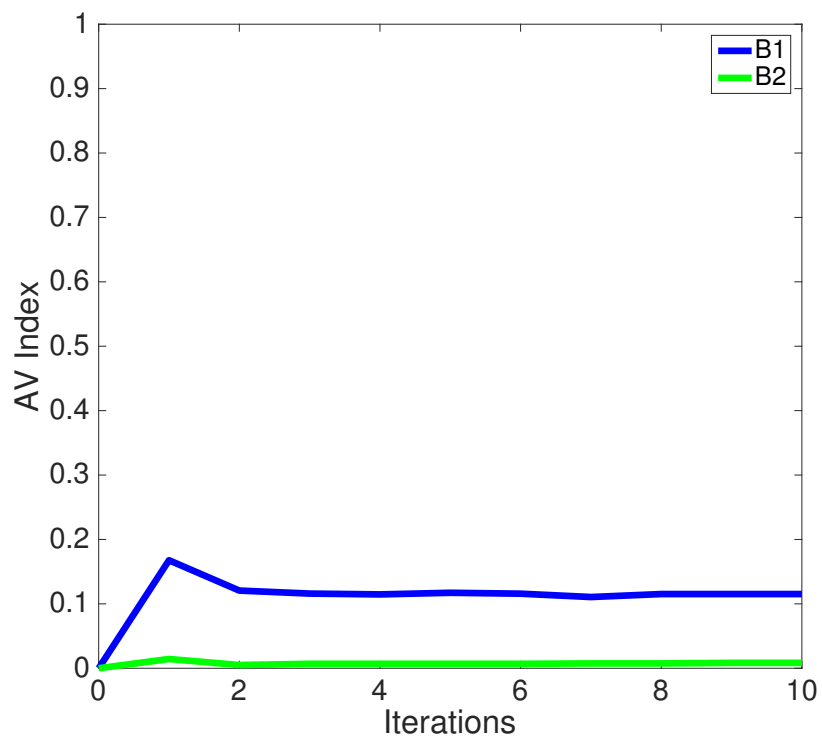


(b) Vessel Tree B

Figure 4.4: Median Arteries TP index trend over iterations, for each simulation setup.

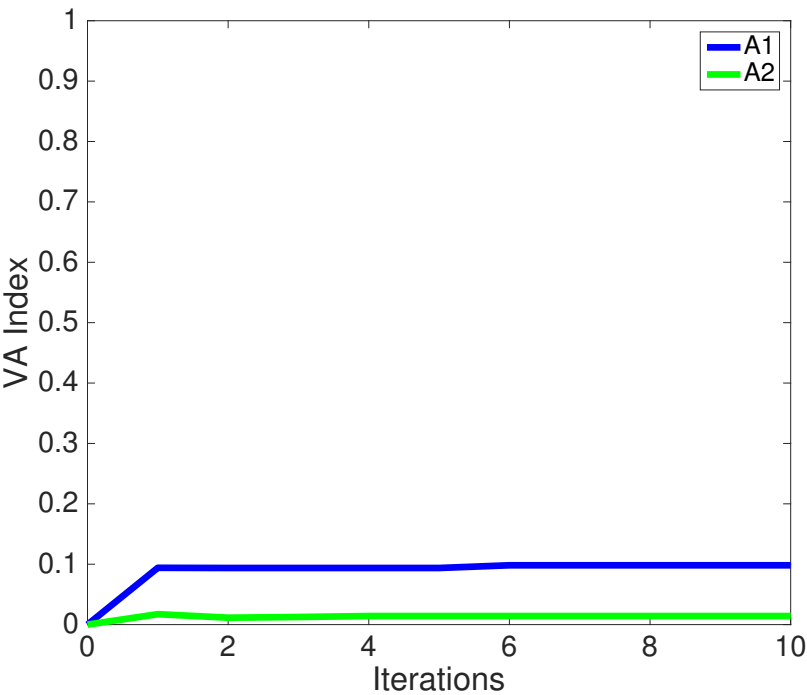


(a) Vessel Tree A

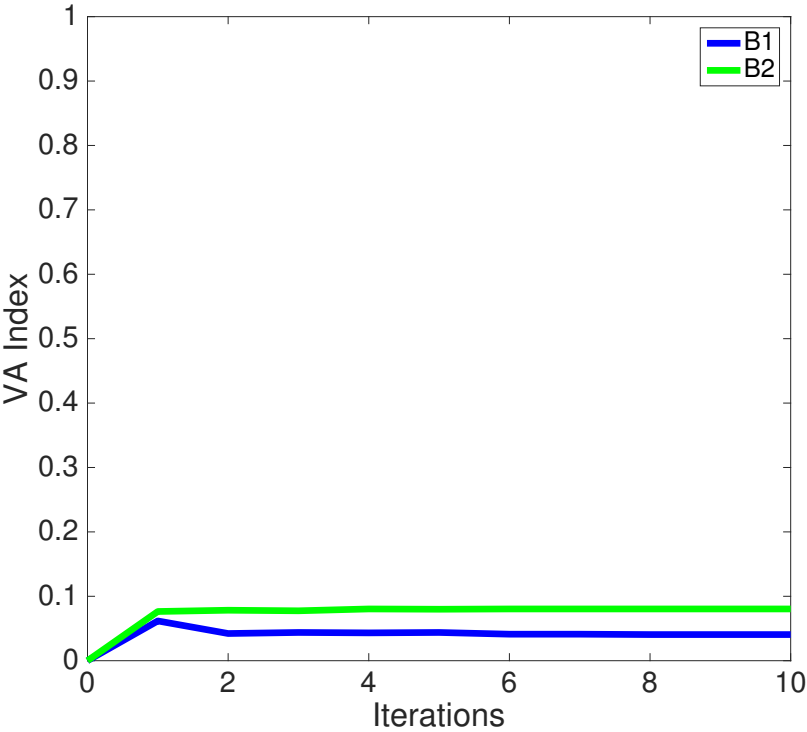


(b) Vessel Tree B

Figure 4.5: Median AV index trend over iterations, for each simulation setup.

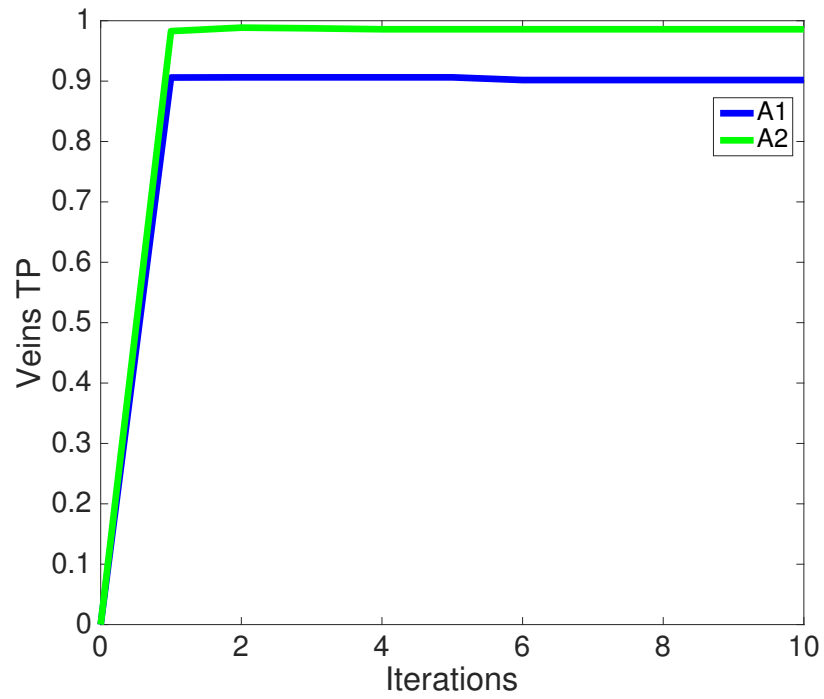


(a) Vessel Tree A

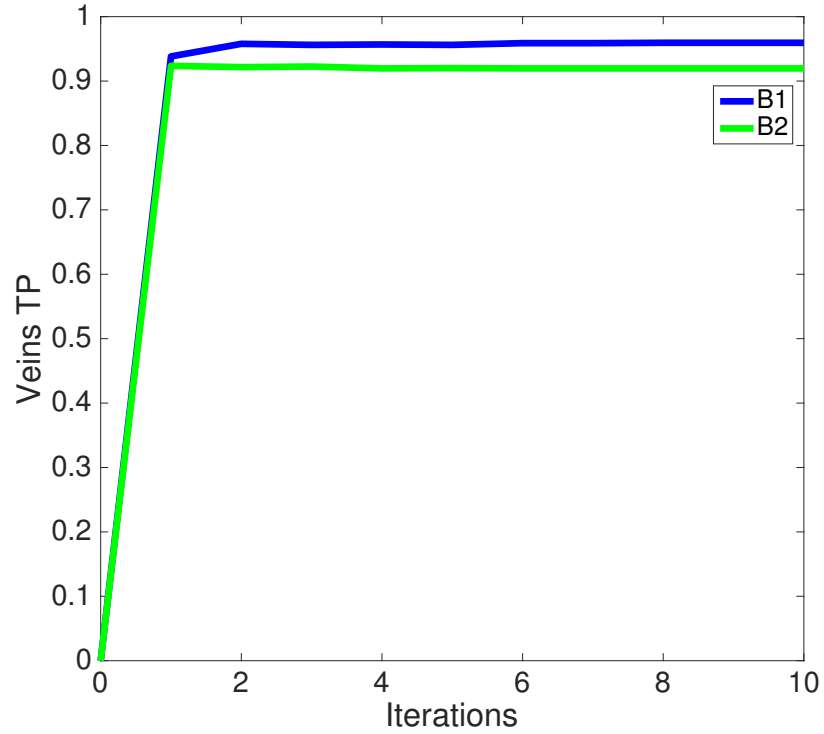


(b) Vessel Tree B

Figure 4.6: Median VA index trend over iterations, for each simulation setup.

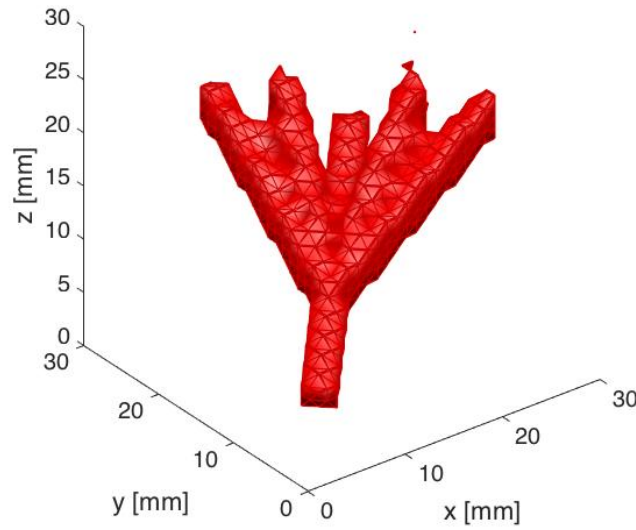


(a) Vessel Tree A

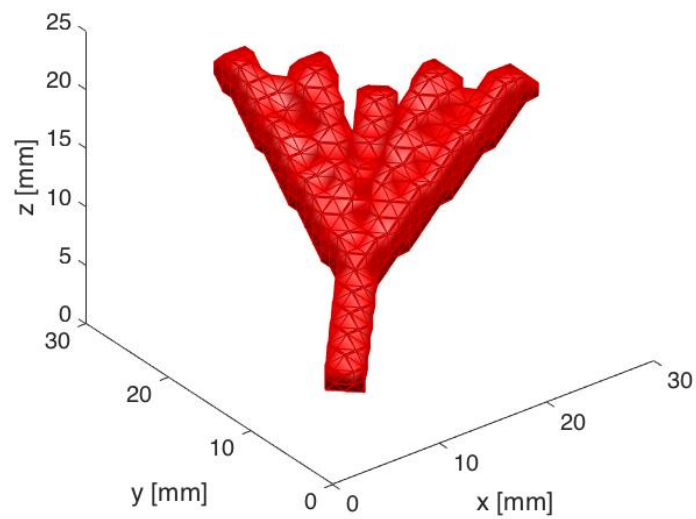


(b) Vessel Tree B

Figure 4.7: Median Veins TP index trend over iterations, for each simulation setup.

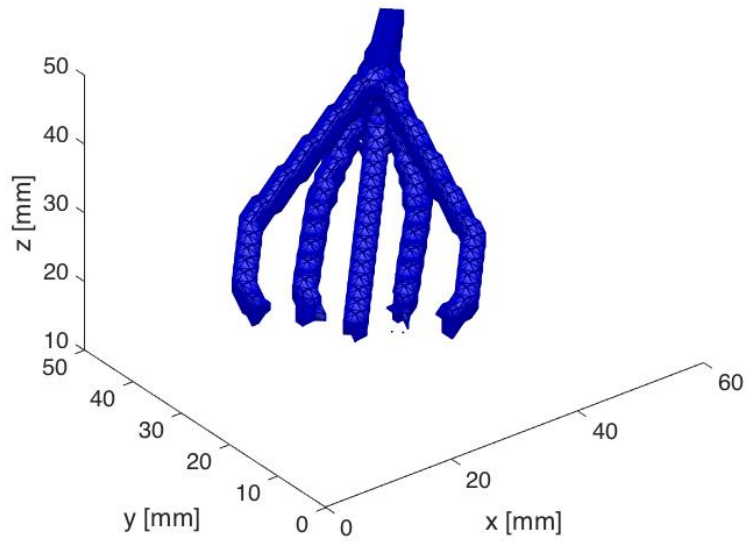


(a) Reconstructed Arterial Tree

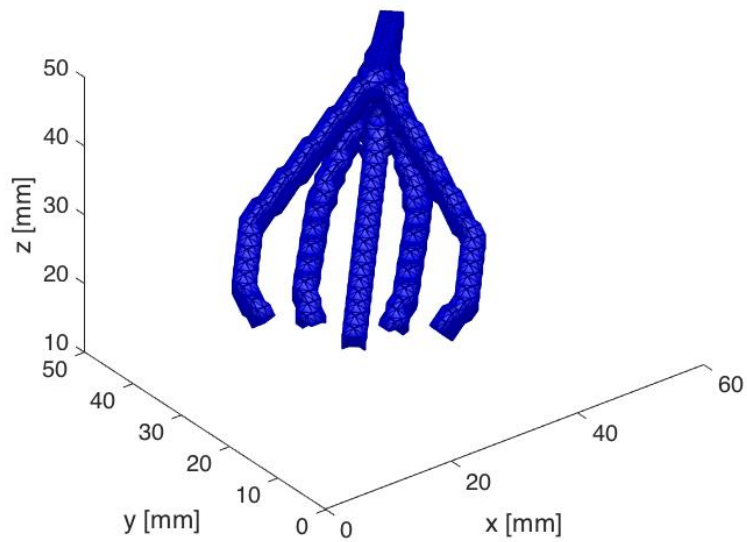


(b) Simulated Phantom Arterial Tree

Figure 4.8: Illustration of the reconstructed and simulated arterial tree for the vessel tree A.



(a) Reconstructed Venous Tree



(b) Simulated Phantom Venous Tree

Figure 4.9: Illustration of the reconstructed and simulated venous tree for the vessel tree A.

Chapter 5

Conclusions and Future Work

The presented work aimed at the classification of arteries and veins, starting from Cone Beam Computed Tomography (CBCT) data. Voxel's Time Intensity Curves (TICs) were reconstructed exploiting the implicit and usually overlooked dynamic information contained in the CBCT acquired data due to the transit of the contrast bolus. Normally, contrast information is exploited only for the segmentation of the vascular tree, yielding an unclassified angiography. Though limited to theoretical analyses and gross digital simulations, the present work was able to demonstrate the feasibility of the proposed principles. Once the TICs were reconstructed the voxels were classified as artery or vein depending on the distribution of the Area Under the Curve (AUC) in time. An iterative algebraic technique, namely the Kaczmarz ART algorithm was exploited for the TICs reconstruction. Obviously, the basic condition for a solution existence and next for algorithm convergence is the availability of a number of projection values I superior higher than the number of unknowns J . Since the faced problem extends the unknown system to 4D by adding the time dimension, the number of available projections values I , furnished by a standard contrast CBCT, would not be enough. Consequently, the problem dimension had to be reduced, in some way. This objective was accomplished considering for TIC reconstruction only the voxels belonging to vessel structures. In fact, by doing that, the problem dimension was reduced up to a hundred times, allowing a problem expansion in the time dimension. Further reduction was obtained by describing TICs by a set of basis functions, suitably shifted over time, describing the wash-in, wash-out contrast transit at an unknown temporal shift. The linear combination of the basis functions produced a TIC, so that each voxel was characterized by a set of basis functions coefficient, still maintaining the equations within the class of linear algebraic problems. Therefore, if a standard image reconstruction process is based on the reconstruction of the attenuation value of the voxel, here the

unknown for each voxels coincides with a set of basis functions coefficients. The algorithm was developed in Matlab 2015 and evaluated in the simulation framework only, for a first feasibility assessment and due to computational times, which grew excessively, given the available HW and a SW development at its first stage. The precision in TIC reconstruction was evaluated by the Root Mean Square Error computation on all the TICs. The reconstructed TICs presented low RMSE values (median value between 0.0366 and 0.0376). The classification was evaluated computing the percentage of correctly classified voxels and incorrectly classified voxels, obtained from the comparison among the computed classification and the simulation a priori information. The classification showed good results, up to more than 90% of correctly classified arteries and veins. This outcome, yet limited to highly simplified angiographic simulations, is promising as to feasibility and deserves some discussion in the perspective of real data analyses. In the latter, including hundreds of vascular branches, a higher classification error has to be expected. Nonetheless, in future developments several improvements can be exploited, which were not introduced so far:

- Time regularization: It was limited here to a coarse separation of TICs into time shifted basis functions, but no constraint relevant to wash-in sequence was imposed and also no insertion or temporal smoothing within the iteration cycles was attempted.
- Spatial continuity: Classification after the reconstruction of TICs was performed on a voxelwise basis. However, region growing techniques could be easily get rid of isolated misclassified voxel on the hypothesis of the continuity of vessels.

A further element in favor of reliable classifications on real data-sets is represented by the wide temporal gap represented by the capillary phase of contrast transit. The contrast medium is rapidly washed in through the arteries, next it virtually disappears expanding in the huge network of unresolved capillaries, and next reappears in veins for the final rapid wash out. This is easily seen by observing the sequence of projections through time and not yet exploited in our algorithm. Further work would be needed in order to make the algorithm suitable for real images processing. In particular, computational time limits might be brightly solved adopting an appropriate computation framework, relying on a more powerful hardware and exploiting parallel computation techniques. Moreover, once the algorithm steps are reliably tested in a general purpose language like Matlab, a translation in a computationally efficient language (e.g. C++) would greatly enhance the processing speed. A core element concerning speed is the tradeoff between memory occupation and computational load. The huge dimension of the algebraic problem matrix of coefficients W (system matrix, in generalized sense)

imposed the repeated computation of the active matrix blocks during the run. Probably, by suitably exploiting the problem symmetries and/or by some simplification this could be avoided, at least in part, thus greatly reducing the computational time.

Appendix A

Tomography

A.1 Computed Tomography

According to Webster's Dictionary [16], *tomography* is derived from the Greek word *tomos*. Tomography can be defined as a “technique of X-ray photography by which a single plane is photographed, with the outline of structures eliminated”. The computed tomography (CT) has been developed to overcome the limitations of conventional radiography. Among them the most heavy limitation consists in the reduction of visibility of the object of interest due to the superimposition of all the anatomic structure in the image. In fact, in conventional radiography all the contribution of the structures along the X-ray direction is summed up and consequently superimposed. The resulting image value (I) can be accordingly defined as the integral value of the attenuation values $\mu(x)$ of all the anatomic structures along the considered ray direction (x) (Eq. A.1).

$$I = I_0 e^{\int_0^L -\mu(x) dx} \quad (\text{A.1})$$

where I_0 is the initial X-ray intensity and L is the length of the distance traveled. This expression is commonly called the Lambert-Beer Law. Obviously, the term μ is a function of the X-ray energy. Usually X-ray with energy of about 70 keV are used. Differently computed tomography aims at representing axial slices of the human body, starting from a set of lateral digital radiographies. The two-dimensional image obtained is a compression along a plane of a three-dimensional slice volume of human body. The thickness of the 3D slice depends on the slice sensitivity profile which measures the axial resolution. The smaller the thickness the greater the resolution. The value obtained for each pixel is measured in a scale relative to water absorption, through units called

Hounsfield units (HU), defined as:

$$\mu(HU) = \frac{\mu - \mu_{H_2O}}{\mu_{H_2O}} \times 1000 \quad (A.2)$$

Therefore low values of HU, close to the 0 of water, are obtained for soft tissues while values up to 1000 and more imply the presence of hard tissue, as the compact bone, and -1000 the air. The tomography approach emerged as early as 1940, when modern computer technology lacked. The basic idea was firstly introduced by Gabriel Frank [49]: he designed the sinogram representation and the optical back-projection techniques. The sinogram is intended as the collection of lines representing the projection values along the X-ray direction. The names comes from the fact that if a point in space is imaged over 180 degrees projections it designs a sinusoidal wave over the lines. Back-projection algorithms consists is the uniform distribution of the sample, along the path that contributed to the sample formation. After twenty years an American Neurologist from Las Vegas tried to perform a series of experiments in order to put in evidence anatomical structures contained in more dense structures. He was William H. Oldendorf, and he built a plastic and iron phantom in order to mimic the desired anatomical geometry [21]. Moreover the radioisotopy-based transverse tomography was introduced in 1963 from David E. Kuhl and Roy O. Edwards [29]. This lead to the modern Emission Computed Tomography (ECT). Two opposite detectors were used and the emission radiation was sampled uniformly in terms of angular sampling. Nonetheless, a precise reconstruction technique was still missing at that time, although in 1917 Radon had already anticipated the mathematical formulation for image reconstruction. Radon introduced the Radon transform, proving that an object could be exactly reconstructed from an infinite set of projections [23]. Around 1956 the inverse Radon transform was introduced as the solution of the image reconstruction problem. In the same years Allan M. Cormack understood how important it was to know the attenuation coefficient distribution of the anatomical part being imaged [12]. He used one of the first CT scanners at that time to prove his mathematical theory for image reconstruction. In 1967 at the Central Research Laboratories of EMI, in England, Godfrey N. Hounsfield lead to the advent of the first clinical scanner and preliminarily estimated that the attenuation coefficients could be estimated with 0.5% accuracy [22]. Since this first scanner many advancements have been achieved and four scanner generations have been designed (Figure A.1). The scanner built in 1971 with the Hounsfield model is called the first-generation CT: the system was composed of the X-ray source and a single detector. Both had to translate in order to acquire a single projection, while they have to rotate in order to change the projection angle. Since two

movements were necessary for the complete read out, the acquisition time was as long as about 5 minutes for a single slice. The need for shorter acquisition times led to the development of the second-generation scanner at EMI in 1975 [17]. In this system n detectors were aligned on a straight line and the X-ray source was collimated in order to produce a fan beam projection. Within this system two movements (translation and rotation) were still required but the acquisition time was reduced of a factor n , leading for an overall time of one minute approximately.

Successively the third-generation scanner was introduced, becoming the most popular scanner geometry. An array of detectors was introduced: the sensor elements are aligned over an arch and the entire array can rotate together with the source. Doing so the translation movement is eliminated and the slice acquisition time is now reduced down to $1/3$, $1/4$ of a second. Immediately thereafter the fourth-generation scanner was introduced: the detectors were fixed, organized in a ring shape and the source is rotating along the ring; however, this geometry was soon abandoned, almost completely. To guarantee the correct functioning of continuously rotating scanners (3^{rd} and 4^{th} generation) *slip ring* technology was introduced, leading to the removal of the connection cables. Further evolution was provided by the development of the spiral CT, that in the 90's became predominant. The spiral CT combines the slip ring technology to a translation of the patient table: this allows to image large body regions through a single scan. Furthermore, several detector arches (up to 256) were set in parallel thus contemporaneously scanning many slices in Multi Slice Spiral CT.

A.2 Image Reconstruction

Tomographic methods require solving the reconstruction from projection problem. Starting from the projections data, a slice of the object being imaged is modeled as a 2D function in the (x, y) space. The mathematical theory for reconstruction was introduced by Radon in 1917. As it was already introduced in Section A.1 the projection image consists in a value representing the attenuation coefficients summation on a line integral along a particular x-ray direction. According to the Lambert Beer Law, if the attenuation coefficient μ is constant in space, the intensity value of the X-ray coming out from the imaged object is equal to :

$$I(x, y) = I_0 e^{-\mu L} \quad (\text{A.3})$$

where I_0 is the initial X-ray intensity and L is the X-ray path length inside the object. In general, the attenuation coefficient is not constant, and in this case the

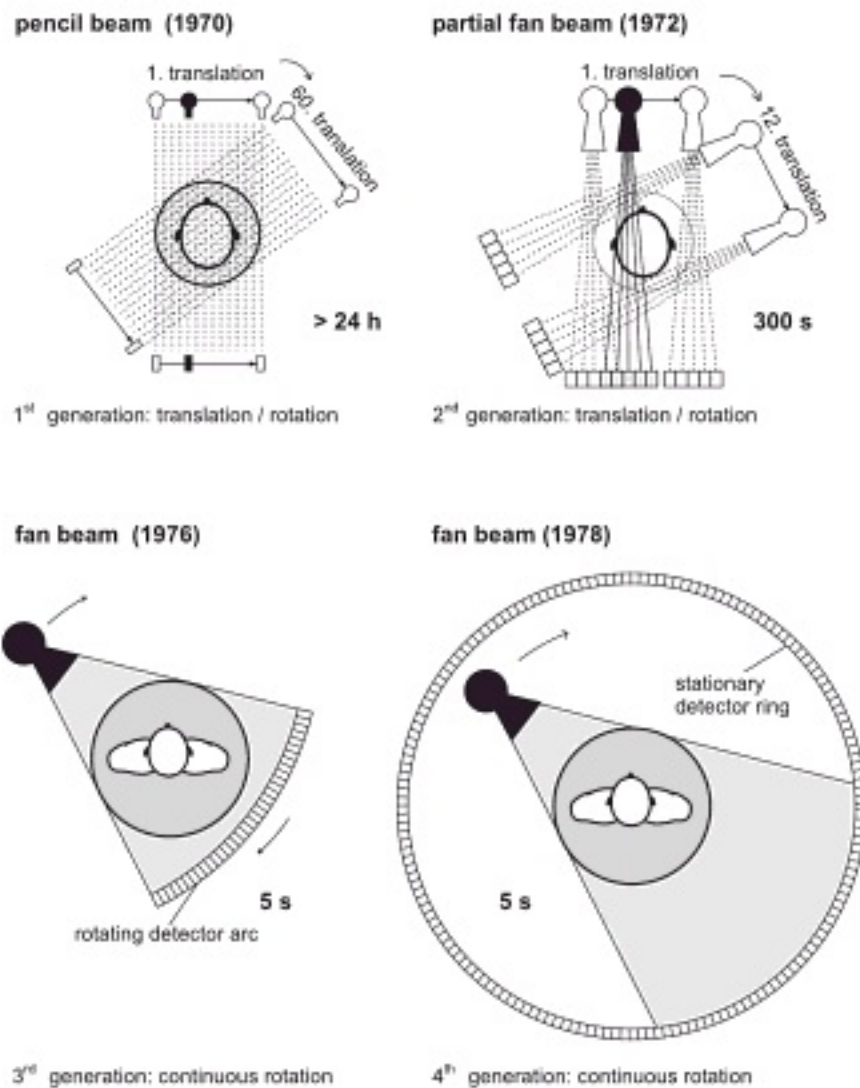


Figure A.1: The four generations of CT scanners

projection ray can be subdivided in smaller paths l_i with uniform attenuation coefficient μ_i , and the intensity of the X-ray becomes:

$$I(x, y) = I_0 e^{-\sum_{i=1}^N \mu_i l_i} \quad (\text{A.4})$$

When l_i approaches zero the summation becomes an integral and the above description becomes :

$$I(x, y) = I_0 e^{-\int_A^B \mu(x, y, z) dl} \quad (\text{A.5})$$

where A and B are the source and the detector position in the plane (x,y) i.e. the X-ray path is the segment \overline{AB} (Figure A.2). Using the logarithmic transformation, the following expression is obtained:

$$\ln \left(\frac{I_0}{I} \right) = \int_A^B \mu(x, y, z) dl \quad (\text{A.6})$$

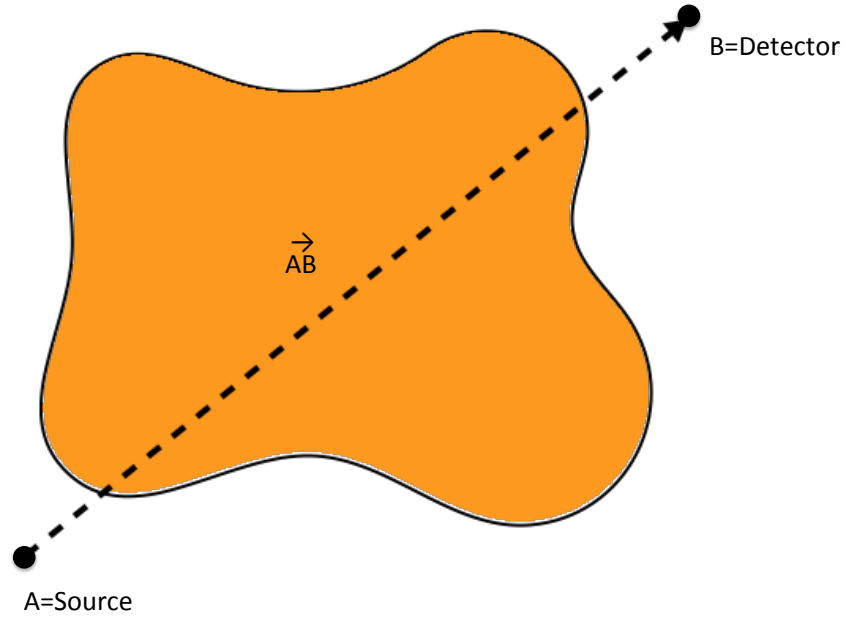


Figure A.2: Generic illustration of a X-Ray path in an object from the point A to the point B.

In CT the projection measurement is defined as the quantity p computed as:

$$p = \ln \left(\frac{I_0}{I} \right) \quad (\text{A.7})$$

Equation A.7 means that the ratio among the input and output X-ray intensity (in the logarithmic scale) represents the attenuation coefficients summation on along the x-ray direction.

When dealing with the computational problem we have to refer to Equation A.4. Switching to the more general 3D problem, if the imaged object is subdivided in volume elements (*voxels*) small enough to have a constant attenuation coefficient μ , the value $\mu_{i,j,k}$ should be estimated for each voxel, i, j, k being the indexes on the space directions (x,y,z). If a general number H of projection rays has been considered, the measure set is composed of H projection values p_h and Equation A.4 can be rewritten as :

$$p_h = \sum_{i,j,k} \mu_{i,j,k} w_{i,j,k,h} \quad (\text{A.8})$$

The $w_{i,j,k,h}$ values represent the weights for each voxel i.e. they represent how much the voxel attenuation coefficient $\mu_{i,j,k}$ has influenced the projection value p_h . Normally, the weights values are computed as the fractional volume of the voxel covered by the x-ray passing through. Taking as an example a two-dimensional image, the weights are representable as the pixel area covered by the x-ray, as illustrated in Fig. A.3

Many approaches can be taken into account for the resolution of the image reconstruction problem. All of them start from the modeling of the process of data collection. Taking as a reference the Lewitt review [34] we can define the process of data acquisition as a discrete-continuous (D-C) model that couples the projection (discrete) data to the function of continuous spatial variables, which corresponds to the image to be reconstructed, denoted here by $\mu(x, y, z)$. Moreover we assume a Linearly Spatially Variant model between the projection data and the function $\mu(x, y, z)$. The DC LSV model can be written as:

$$p_h = \int \int \int_{\Omega} \mu(x, y, z) h_i(x, y, z) dx dy dz \quad h = 1, 2, \dots, H \quad (\text{A.9})$$

where Ω is the spatial domain and $h_i(x, y, z)$ is the integration kernel function over space, which is the weight function. Once the DC model is developed the reconstruction algorithms can proceed in three different ways:

1. Image Reconstruction based on Continuous Continuous model

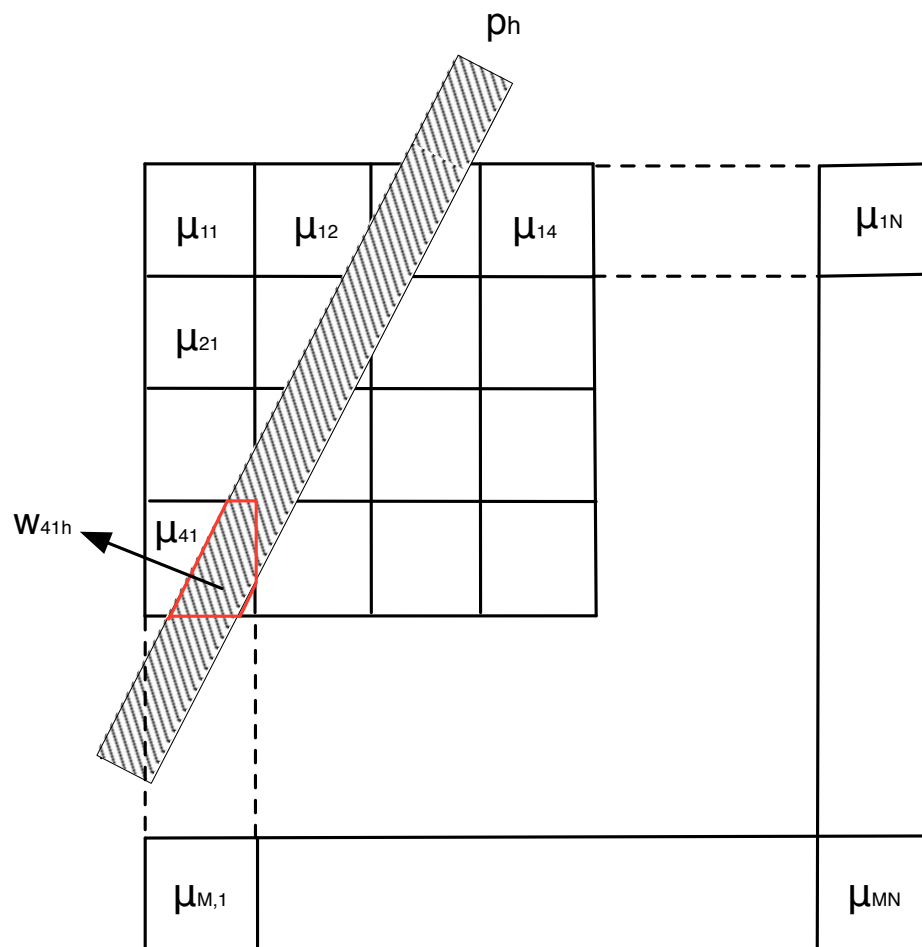


Figure A.3: Graphical representation of the computation of weights. In the specific 2D example the weight $w_{4,1,h}$ is computed as the area of the pixel $[4, 1]$ and represents the contribution of the pixel $[4, 1]$ to the projection p_h

2. Image Reconstruction based on Discrete Continuous model
3. Image Reconstruction based on Discrete Discrete model

A.2.1 Image Reconstruction based on CC model

This approach considers the measure set as a sample of a continuous function in the space of measurements. In this way not further approximation due to discretization is introduced. However, an acquisition model simplification is needed in order to derive an analytic solution. This approach leads to a non-optimal solution, but to a short computational time. Let's take, for example, a parallel ray geometry. We consider a particular projection direction, identified by the a line rotated by an angle Φ about the z -axis and rotated by an angle θ about the x -axis. This rotation transform the general coordinates (x, y, z) to the coordinates (x', y', z') . Let's take for simplicity the projection line as parallel to the y' axis. In addition, the measurement model must be simplified, and in particular the 3D integration region is shrunk to a non-volumetric thin line, so that the 3D-integral must be reduced to a 1D-integral along the line. Now the C-C model can be written as:

$$p(x', z', \theta, \Phi) = \int_L \mu(x, y, z) dy' \quad (\text{A.10})$$

where L is the whole projection line. In this equation, $p(x', z', \theta, \Phi)$ is the function in the measurement space, and the measure set is the sampling of this function. For this type of formulation, the Fourier Central Section Theorem (Figure A.4) establishes that the 2-D Fourier transform of each parallel ray projection corresponds to a plane of the 3-D Fourier transform of the function $\mu(x, y, z)$. Defining the 3D Fourier transform of $\mu(x, y, z)$ as:

$$\hat{\mu}(\omega_x, \omega_y, \omega_z) = \int \int \int_{\Omega} \mu(x, y, z) e^{i(\omega_x x + \omega_y y + \omega_z z)} dx dy dz \quad (\text{A.11})$$

and we select, in the transform, the plane $\omega_y = 0$ we obtain

$$\hat{\mu}(\omega_x, 0, \omega_z) = \int \int \int_{\Omega} \mu(x, y, z) e^{i(\omega_x x + \omega_z z)} dx dy dz \quad (\text{A.12})$$

Moreover the 2D Fourier transform of the projection $p_{\theta\Phi}(x', z') = p(x', z', \theta, \Phi)$ is

$$p_{\theta\Phi}(\omega_{x'}, \omega_{z'}) = \int \int_{\Omega} p_{\theta\Phi}(x', z') e^{-i(\omega_{x'} x' + \omega_{z'} z')} dx' dz' \quad (\text{A.13})$$

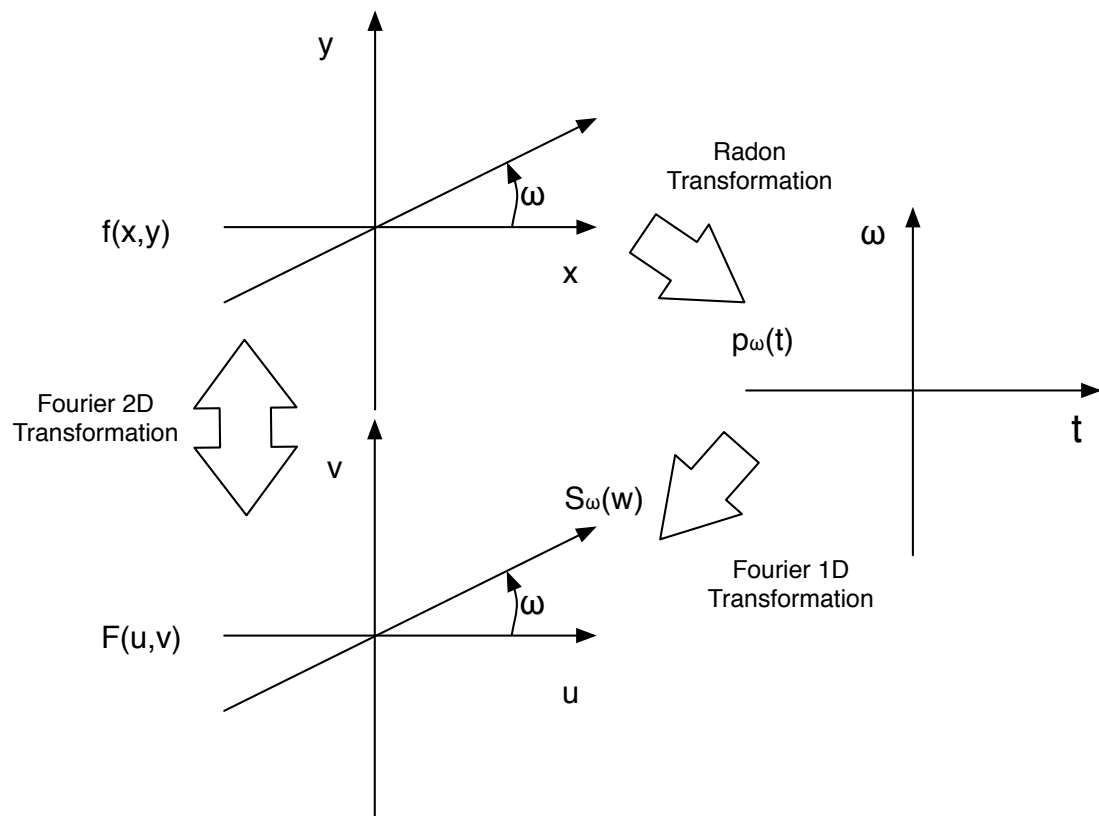


Figure A.4: Graphical illustration of the Central Section Theorem. (x,y) are the space variables while (u,v) are the frequencies variables. The angle ω here represents the direction of the projection ray.

We can identify in Equation A.12 the term $\int \mu(x, y, z) dy = p(x', z', \theta = 0, \Phi = 0)$ obtaining in conclusion:

$$\hat{\mu}(\omega_x, 0, \omega_z) = \int \int p_{\theta=0, \Phi=0}(x', z') e^{i(\omega_x x + \omega_z z)} dx dy dz = p_{\theta=0, \Phi=0}(\omega_{x'}, \omega_{z'}) \quad (\text{A.14})$$

which is the 2-D Fourier transform of the projection. Inverting this relationship the function $\hat{\mu}(\omega_x, \omega_y, \omega_z)$ can be expressed in terms of $p_{\theta\Phi}(\omega_{x'}, \omega_{z'})$ and hereafter $\mu(x, y, z)$ can be derived from $\hat{\mu}(\omega_x, \omega_y, \omega_z)$ by the inverse Fourier transform, obtaining the needed image reconstruction.

A.2.2 Image Reconstruction based on DC model

This approach does not introduce any further discretization and approximation. Since a function of continuous variables can not be estimated starting from a finite set of measures, further conditions must be introduced. In particular the Moore-Penrose generalized solution can minimize the difference $\|\mathcal{P}(\mu) - p\|$, denoting as ρ the projection operator. If another vector q , with I elements i.e. $q = (q_1, q_2, \dots, q_I)$ is introduced, it can be proved [40, 41] that the Moore Penrose solution can be written as:

$$\mu(x, y, z)_{MP} = \sum_{i=1}^I q_i h_i(x, y, z) \quad (\text{A.15})$$

If we substitute this expression into Equation A.9 we obtain

$$p_h = \int \int \int_{\Omega} \sum_{j=1}^J q_j h_j(x, y, z) h_i(x, y, z) dx dy dz \quad h = 1, 2, \dots, H \quad (\text{A.16})$$

We can define the Gram Matrix elements [4] as

$$H_{i,j} = \int \int \int_{\Omega} h_i(x, y, z) h_j(x, y, z) dx dy dz \quad (\text{A.17})$$

in order to reduce the system as a classical linear system:

$$\mathbf{p} = \mathbf{H}\mathbf{q} \quad (\text{A.18})$$

This solution actually requires the construction of a large, non sparse system and the computation of the matrix \mathbf{H} can be not trivial and computationally heavy. In fact the matrix \mathbf{H} tends to be very dense due to the fact that the H_i functions are

not confined. In several methods the basis functions are chosen to be coincident to the integration kernels and in this case we call them natural pixels [7]. Another type of basis functions is the *ridge functions* and they are used when, at each angle, the projection value is a continuous variables function in the perpendicular direction to the projection lines [35, 26].

A.2.3 Image Reconstruction based on DD model

The DD model approach, alias numerical approach, introduces a further approximation. In fact, the image function $\mu(x, y, z)$ is approximated with the linear combination of a finite number J of discretized basis functions.

$$\bar{\mu}(x, y, z) = \sum_{j=1}^J \tilde{\mu}_j b_j(x, y, z) \quad (\text{A.19})$$

If, once again, we substitute the term $\bar{\mu}(x, y, z)$ to the term $\mu(x, y, z)$ in Equation A.9 we obtain:

$$p_h = \int \int \int_{\Omega} \bar{\mu}(x, y, z) h_i(x, y, z) dx dy dz \quad h = 1, 2, \dots, H \quad (\text{A.20})$$

and if we model the contribution of the j^{th} basis functions to the i^{th} data measurements as a term $a_{i,j}$ where :

$$a_{i,j} = \int \int \int_{\Omega} h_i(x, y, z) b_j(x, y, z) dx dy dz \quad (\text{A.21})$$

we finally obtain:

$$p_i = \sum_{j=1}^J a_{i,j} \tilde{\mu}_j \quad (\text{A.22})$$

Having an equation for each projection value, the system can also be expressed in matrix form:

$$\mathbf{p} = \mathbf{A}\boldsymbol{\mu} \quad (\text{A.23})$$

The matrix \mathbf{A} , formed by the elements $a_{i,j}$ is called *System Matrix*. Within this model, the reconstruction algorithm has to compute the vector $\tilde{\mu}$ in order to get a vector \tilde{p} similar to the measurement data p . The DD model includes several approaches that consists in five general components:

1. **A model of the physics of the measurement process:** the characteristics of

attenuation must be taken into consideration in order to design the appropriate integration kernels h_i . In CT, kernels are commonly simplified to uniform thick rays, thus approximating the sensitivity function of sensors to an on-off, the width of which is given by the FWHM of the sensor bell-shaped sensitivity.

2. **A model of the measurement uncertainty:** a probability distribution of the measurements can help in avoiding errors due to acquisition inaccuracies. This is particularly valuable in PET and SPECT techniques, due to their Poissonian statistics. Conversely, at common exposure levels CT noise can be considered Gaussian and statistics is not explicitly considered..
3. **A set of basis functions:** the basis functions can be global or localized. A global basis function influences the whole image spatial region while a localized one only influences one single voxel or a finite set of voxels. The latter is the common approach in CT and will be kept here; nonetheless, our approach is passible of improvements if basis functions targeting vessel segments were used.
4. **An objective function:** it consists in a mathematical quantity to be minimized by the algorithm. It must be chosen in order to optimize some specified parameters, depending on the objective of the algorithm. When statistics is taken into account maximum likelihood is the most appropriate objective function. Conversely, if not, as in CT, the least squares of errors is considered, either implicitly or explicitly.
5. **A numerical algorithm:** given the measure set and the all the conditions chosen through the categories above the appropriate algorithm must be chosen. For example a non iterative algorithm might be a good choice when the analytic solution is not too heavy or complicated, i.e when the dataset is not too big and the objective function is quite simple. Most frequently iterations are needed which implement a simple update operation based on optimizing the objective function towards the entire set, or a sub-set of projections or even a single one. The algorithm repeatedly iterates through the whole data set until a stopping rule is satisfied.

A.3 Filtered Backprojection

One of the most diffuse operator in image reconstruction is the back-projection (BP) operator. Once the projection data are obtained, back-projecting means to give the projection value to all the points belonging to the line $t = x\cos\omega + y\sin\omega$. Summing

all the contribution of the back projection at different angles, the following expression is obtained:

$$\delta \int_0^\pi p_\omega(x\cos\omega + y\sin\omega) \quad (\text{A.24})$$

In Figure A.6 the concept of back-projection is illustrated. The Filtered Back-projection is a back projection in which the projection data are previously filtered, since the reconstructed image is blurred by a simple BP, as shown in Figure A.7. In fact the BP from many directions results in a star and, if the number of considered directions tends to infinite the obtained image equals to the image function convoluted to the Point Spread Function (Figure A.5):

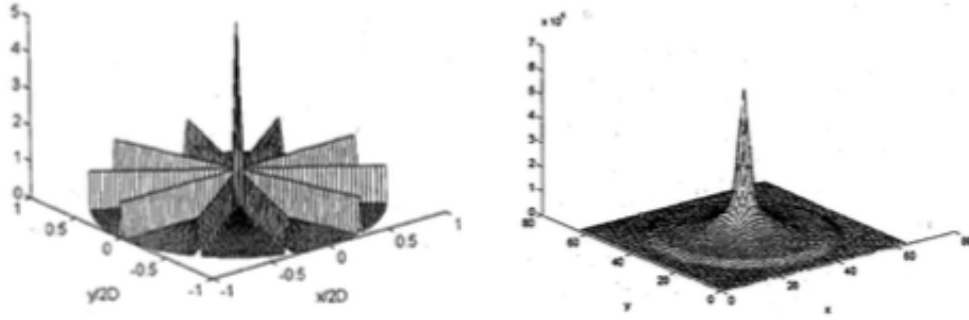


Figure A.5: Star artifact on a point reconstruction. On the left the artifact for a finite number of back projection angles. On the right, the star artifact when the number of considered directions of projection tends to infinite

$$\int_0^\pi p_\omega(x\cos\omega + y\sin\omega) = \mu(x, y, z) \otimes \frac{1}{\sqrt{x^2 + y^2}} \quad (\text{A.25})$$

where $1/\sqrt{x^2 + y^2} = 1/\rho$ is the PSF. Due to the linear property, the Fourier Transform of a star is a star in the frequencies domain with density decreasing as $1/\sqrt{\omega_x^2 + \omega_y^2} = 1/\Omega$ and tends to $1/\Omega$ as the number of direction tends to infinite. This is called Modulation Transfer Function (MTF). Multiplying by $MTF^{-1} = \Omega$ in the Fourier space the correction of the back-projection blurring is obtained.

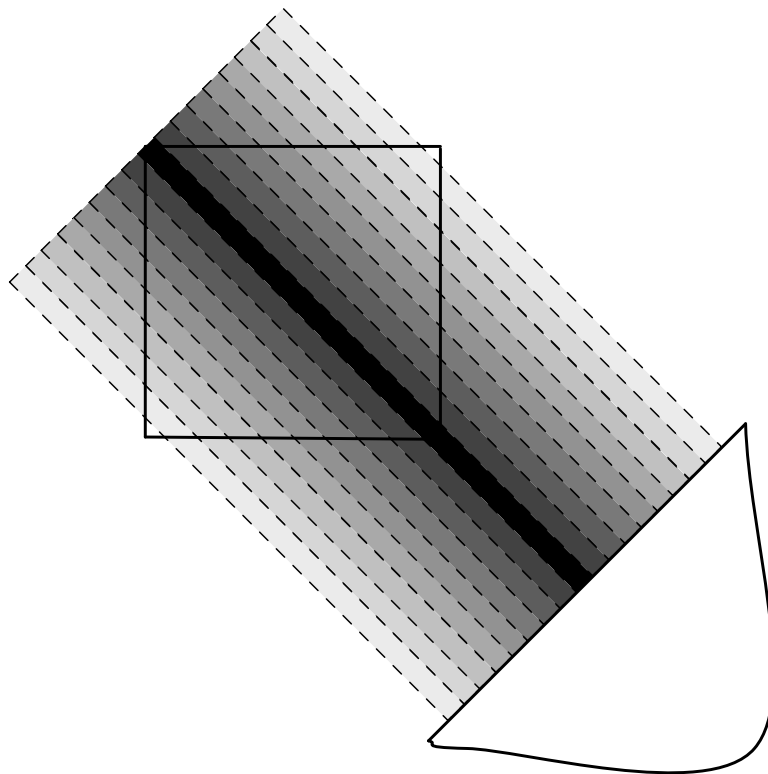
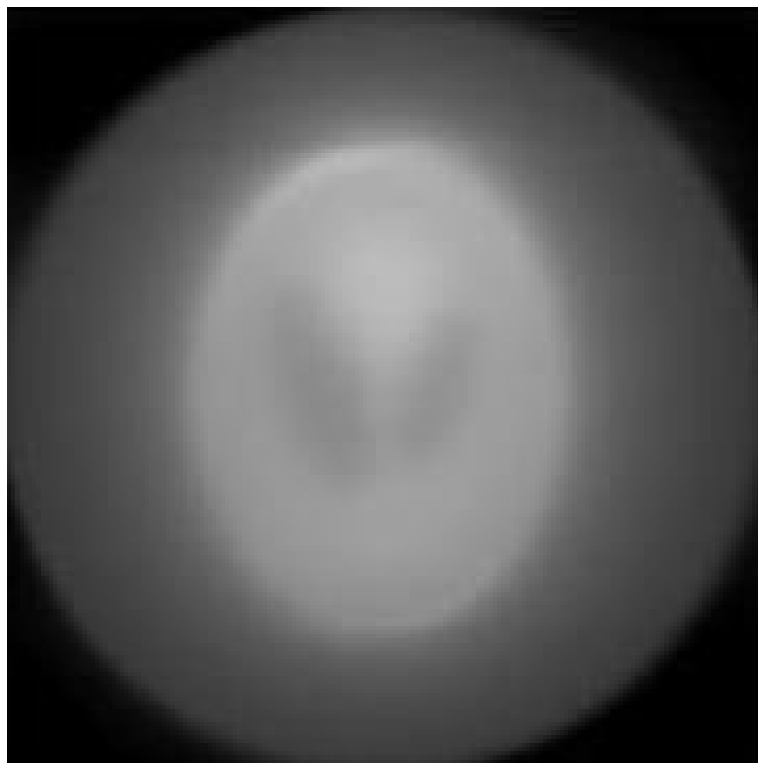


Figure A.6: Graphical illustration of the back-projection result: all the points on a line obtain a value equal to the value $p_{\omega}(t)$ of projection along that line.



(a) Shepp-Logan Phantom



(b) Shepp-Logan Phantom reconstructed with backprojection.
Blurring artifact is noticeable

Figure A.7: Comparison among Shepp-Logan Phantom and its backprojected version.

A.4 Feldkamp-Davis-Kress (FDK) Algorithm

The Feldkamp-Davis-Kress Algorithm is an approximated filtered backprojection for the Cone Beam CT (CBCT) with circular trajectory. It introduces a correction for the in-plane angle aperture and another correction for the elevation angle aperture. In fact, as shown in Figure A.8, the rays are representable as many inclined fans and the whole reconstructed volume coincide to a cylinder plus two cones.

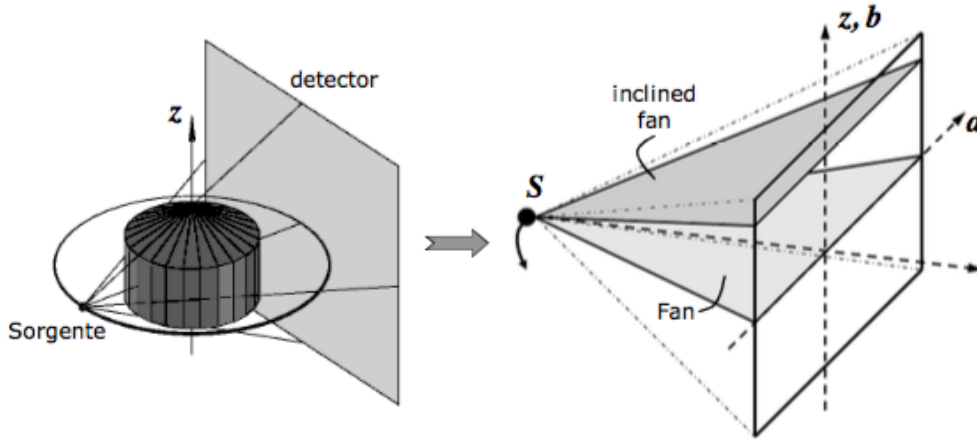


Figure A.8: The Cone Beam Geometry is illustrated. The reconstructed volume is composed of a cylinder plus two cones. The set of projection rays can be represented by several inclined fan.

If we consider each projection ray, it will produce on the panel a projection value that is a function of the panel coordinate a , the other panel coordinate b , and the panel rotation angle β around the gantry axis, as illustrated in Figure A.9. The coordinates a and b are computed as :

$$a = R \tan(\gamma) \quad (\text{A.26})$$

$$b = \sqrt{R^2 + a^2} \tan(\kappa) \quad (\text{A.27})$$

where γ is the ray's in-plane angle, κ is the elevation angle, and R the rotation radius. The FDK algorithm is composed of three main steps:

1. A correction for the cone beam geometry is produced on the projection values depending on which ray produced the projection value. A first correction term is represented by $\cos(\gamma)$, being γ the ray's in-plane angle, as represented in Figure A.9. This correction solves the fan-beam inaccuracy due to the different lengths of the rays path inside the object. A second correction term $\cos(\kappa)$, being κ the elevation angle, solves the same issue of the fan-beam problem but in the

third dimension. The corrected projection $p^c(a, b, \beta)$ is obtained from the initial projection value $p(a, b, \beta)$ as:

$$p^c(a, b, \beta) = p(a, b, \beta) \cos(\gamma) \cos(\kappa) \quad (\text{A.28})$$

2. The ramp filter is applied to every projection row. The ramp filter is defined as:

$$g^p(a) = F_{1D}^{-1} \{|\omega|\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\omega| e^{j\omega t} d\omega \quad (\text{A.29})$$

The filtering operation can be written as:

$$\bar{p}^c(a, b, \beta) = p^c(a, b, \beta) \otimes g^p(a) \quad (\text{A.30})$$

3. The last step is composed of a simple back-projection after a weighing for a factor $\frac{R^2}{U(x, y, \beta)^2}$.

$$\mu(x, y) = \int_0^{2\pi} \frac{R^2}{U(x, y, \beta)^2} \bar{p}^c(a, b, \beta) d\beta \quad (\text{A.31})$$

with:

$$a(x, y, \beta) = R \frac{-x \sin(\beta) + y \sin(\beta)}{R + x \cos(\beta) + y \sin(\beta)} \quad (\text{A.32})$$

$$b(x, y, z, \beta) = z \frac{R}{R + x \cos(\beta) + y \sin(\beta)} \quad (\text{A.33})$$

$$U(x, y, \beta) = R + x \cos(\beta) + y \sin(\beta) \quad (\text{A.34})$$

In conclusion the FDK algorithm is an approximate extension of the fan-beam correction, for the 3D CBCT. It produces some artifacts, know as FDK artifact, mainly consisting in a blurring along the z-axis at the highest elevation angles. However good reconstructions has been demonstrated to be achievable at much higher elevation angles ($40^\circ - 50^\circ$) than the ones commonly used in clinic ($\pm 10^\circ$).

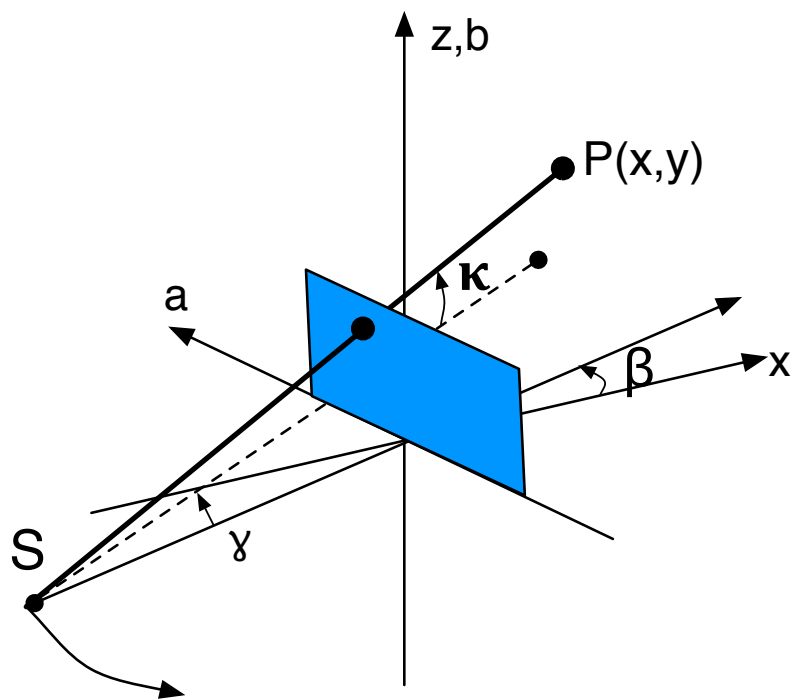


Figure A.9: Geometric schema of elevation (κ) and in-plane (γ) angles in a 3D Cone Beam projection

Appendix B

Implementation Details

In this section a more detailed explanation of the implementation of the algorithm explained in the previous Sections will be presented. Here only the mathematical manipulations are reported. The appendix is divided in categories: Simulation, System Matrix, Dynamic System Matrix, Basis Functions and Segmentation. Each category contains the mathematical explanation of the main functions used for the algorithm implementation. Where needed, some code resources are reported.

B.1 Simulation

The simulation functions compute the simulated Digital Subtraction Angiography (DSA) reconstruction and simulate the contrast transit in the simulated vessels. The structure of the vascular tree is charged to the user. The vascular tree structure can be set through an array, the Vessel Array (VA) containing the extreme of the vessel tracts. Linking those extremes the whole vessel is obtained. The vessel array is composed in the following way

$$VA = [p_1, p_2, p_3 \dots p_{n+1}] \quad (B.1)$$

where n is the number of tracts composing the vessel, and the points $p_1, p_2 \dots p_{n+1}$ are the extremes of the tracts composing the vessel. More than one vessel can be constructed and many vessel can be overlapped, in order to get a vascular tree. The function in charge for the simulation of a single vessel takes as input the dynamic parameters (acquisition time T_a [s], Delay δ [s], contrast transit time T_t [s] , and the coordinates of the vessel tracts to be), the vessel diameter, d_v and the Vessel Array (VA). It gives as output the simulated DSA reconstruction and two time-map images

, containing for each voxel the values of the wash-in and wash-out of the contrast respectively. In general, if n vessels are built, the image volumes V_v^i are summed one another, in order to get the vascular tree volume V_{vt} .

$$V_{vt} = \sum_{i=1}^n V_v^i \quad (\text{B.2})$$

With this method, two vessels can overlap, meaning that a voxel belongs to two different vessels, each having his own dynamics. In this case, two possible t_{on} are associated to the voxel, one for each simulated vessel. In order to manage those shunts, the t_{on} for each voxel is chosen as the earlier instant of time among the several t_{on} values associated to this voxel. This means that we are classifying the vessel basing on the first instant of contrast arrival. Differently in order to get the wash-out map image I_{off} , the I_{off} values are chosen among the maximum value on t_{off} different from zero, favoring the vein with respect to the artery. With this method, if an artery overlaps to a vein the voxel will be modeled with a longer transit time than the other vessels, representing the combined effect of the artery-vein spatial superposition.

Within the function, firstly the radius of the vessel is computed as the half of the vessel diameter:

$$r_v = \frac{d_v}{2} \quad (\text{B.3})$$

A parameter Δ , being the maximum displacement for any segment is set *a priori*. The number of segments in the radius is computed as

$$n_{rad} = \left\lceil \frac{r_v}{\Delta} \right\rceil \quad (\text{B.4})$$

and the angular increment exploited to build disks of vessel cross-section is defined as

$$\delta\phi = \text{atan2}(n_{rad}, r_v) \quad (\text{B.5})$$

Once all this variables are computed the length of each tract of the vessel is computed using the euclidean distance among one extreme of the vessel p_i and the previous extreme p_{i-1} .

$$L_{t(i)} = \sqrt{(p_x^i - p_x^{i-1})^2 + (p_y^i - p_y^{i-1})^2 + (p_z^i - p_z^{i-1})^2} \quad (\text{B.6})$$

and the total vessel length is computed as the sum of all tracts lengths.

$$L_v = \sum_{i=1}^n L_{t(i)} \quad (\text{B.7})$$

Once the total length is computed the contrast flow speed can be calculated as the ratio between the vessel total length and the acquisition time:

$$v_{cm} = \frac{L_v}{T_a} \quad (\text{B.8})$$

Then for each vessel tract \vec{t}^i , the unit vector directed as the vessel tract is computed as:

$$\vec{t}_u^i = \frac{\vec{t}^i}{L_{t(i)}} \quad (\text{B.9})$$

where $\vec{t}_i = |t_x \ t_y \ t_z|$ is the tract vector.

Since many segments are contained in a tract, the number of segments is computed and the steps for vessel growing are computed in the three cartesian directions x,y, and z.

$$n_t^i = \frac{L_t^i}{\Delta} \quad (\text{B.10})$$

$$\begin{cases} dx_t^i = \Delta \cdot \vec{t}_{ux}^i \\ dy_t^i = \Delta \cdot \vec{t}_{uy}^i \\ dz_t^i = \Delta \cdot \vec{t}_{uz}^i \end{cases} \quad (\text{B.11})$$

The increment vector having as components $|dx_t^i \ dy_t^i \ dz_t^i|$ is computed easily as $d\vec{t}^i = \Delta \cdot \vec{t}_u^i$. At this point the vector orthogonal to the vessel direction must be defined. It is approximated with the cross product among \vec{t}_u^i and either the x, y or z unit vector, depending on the \vec{t}_u^i orientation.

$$\begin{cases} \vec{r}_u^i = \vec{t}_u^i \times \vec{x} \text{ if } \left| (\vec{t}_u^i \times \vec{x}) \right| > 0.1 \\ \vec{r}_u^i = \vec{t}_u^i \times \vec{y} \text{ if } \left| (\vec{t}_u^i \times \vec{y}) \right| > 0.1 \\ \vec{r}_u^i = \vec{t}_u^i \times \vec{z} \text{ if } \left| (\vec{t}_u^i \times \vec{z}) \right| > 0.1 \end{cases} \quad (\text{B.12})$$

The first diameter center is set in C , where $C = \vec{p}_i$ and the distance traveled by contrast D_c (which the first time has been initialized to zero $D_c = 0$). The contrast arrival time is computed as:

$$t_{on} = \delta + \frac{D_c}{v_{cm}} \quad (\text{B.13})$$

and for the contrast leaving time the transit time T_t is added to the contrast arrival time.

$$t_{off} = t_{on} + T_t \quad (\text{B.14})$$

Then, the cylinder is filled by the contrast proceeding with angular steps $\Delta\phi$. For each angle ϕ the radius is rotated by ϕ from f exploiting the Rodriguez formula:

$$r_v = (\vec{r}_v \cdot \vec{r}) \cdot \cos(\phi) + \vec{t}_u \times (\vec{r}_v \cdot \vec{r}) \cdot \sin(\phi) + \vec{t}_u \cdot (\vec{t}_u \cdot (\vec{r}_v \cdot \vec{r})) \cdot (1 - \cos(\phi)) \quad (\text{B.15})$$

where \vec{r} is the vector orthogonal to the vessel direction, as defined before, and \vec{t}_u is the unit vector in the vessel direction. The increment in the radial direction is computed as:

$$\Delta r = \frac{\vec{r}_v}{n_{rad}} \quad (\text{B.16})$$

and for all the radial steps the coordinates of the point to fill are computed as:

$$\begin{cases} P_x = P_x + \Delta r_x \\ P_y = P_y + \Delta r_y \\ P_z = P_z + \Delta r_z \end{cases} \quad (\text{B.17})$$

The voxel to be filled by contrast is approximated starting from the point coordinates:

$$\begin{cases} V_x = \text{round}(P_x) \\ V_y = \text{round}(P_y) \\ V_z = \text{round}(P_z) \end{cases} \quad (\text{B.18})$$

and subsequently, if the voxel is not yet filled by contrast it is filled

$$\text{Image}(Vx, Vy, Vz) = 1; \quad (\text{B.19})$$

and the images of arrival and leaving times are filled in the correspondent voxel position with the values t_{on} and t_{off} .

$$\begin{cases} I_{on}(Vx, Vy, Vz) = t_{on} \\ I_{off}(Vx, Vy, Vz) = t_{off} \end{cases} \quad (\text{B.20})$$

In order to obtained a stored 4D image, containing a temporal profile for each voxel,

or ideally being a 3D volume changing in time (the fourth dimension), the information stored in the t_{on} and t_{off} images is exploited, and data are reordered. Within the simulation functions the projection functions are of great importance. In fact, after the simulation of the image volume, the projection data must be obtained. The projection computation is based on the Lambert Beer law. Since the system matrix is computed for the algorithm running, and it contains the values of the x-ray paths along all voxels, it can be exploited to compute the attenuation of the voxels to a specific projection ray. So, for each projection ray i , initializing the x-ray intensity to $I_0 = 1$, the x-ray intensity I obtained after overpassing all the voxels is:

$$I_i = I_0 e^{\sum_{j=1}^J a_{i,j} \cdot l_j} \quad (\text{B.21})$$

where J is the number of voxels and $a_{i,j}$ is a weight that expresses how much the voxel j attenuates the x-ray i in terms of path length.

B.2 System Matrix

The system matrix is composed of coefficients weighing the contribute of each voxel j to each projection i . Within this work the weights are computed as the length of the path of a thin non volumetric ray inside a cubic voxel, as shown in Figure 2.4 in Section 2.2. In order to compute those distances all the projection rays must be computed, along with all the intersection points with the voxel edges. This has been implemented through a cycle on all the projection rays. Each projection ray corresponds to a specific projection angle θ and to a specific position on the FPD (z', x'), so that a projection value is a function $p(\theta, z', x')$, where z' and x' are the axis along the FPD dimensions. The reference system has been chosen with the z -axis coincident to the gantry axis. The reference system is illustrated in Figure B.1.

The FPD points position has been initialized in $\theta = 0$, every time the projection angle is changed from θ to $\theta + \Delta\theta$ the FPD points are rotated of $\Delta\theta$. The rotation is obtained multiplying a rotation matrix R to the FPD points coordinate at the angle θ .

$$R = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & 0 \\ \sin(\Delta\theta) & \cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.22})$$

Within each projection angle, each projection ray impressing a FPD sensor is considered separately. The projection ray is identified with the straight line connecting the source (which rotates together with the panel) and the FPD sensor under exam.

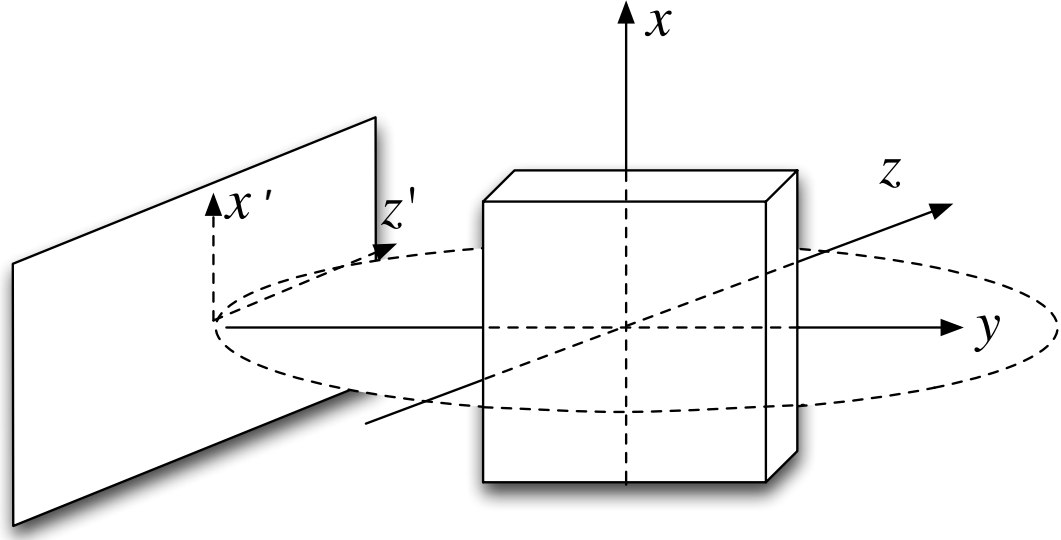


Figure B.1: Reference system for the acquisition set up

Its position in space is identified by an angular coefficient on each plane m_{xy}, m_{yz}, m_{zx} , plus an intercept value for each plane q_{xy}, q_{yz}, q_{zx} . Those coefficients are computed as follows:

$$\begin{cases} m_{xy} = \frac{(s_x - p_x)}{(s_y - p_y)} \\ m_{yz} = \frac{(s_y - p_y)}{(s_z - p_z)} \\ m_{zx} = \frac{(s_z - p_z)}{(s_x - p_x)} \end{cases} \quad (\text{B.23})$$

$$\begin{cases} q_{xy} = s_x - m_{xy}s_y \\ q_{yz} = s_y - m_{yz}s_z \\ q_{zx} = s_z - m_{zx}s_x \end{cases} \quad (\text{B.24})$$

Then, in order to compute all the paths across the voxels, the projection lines have been intercepted with all the planes delimiting the voxels. Figure B.2 shows a representation of only three planes delimiting a voxel.

Therefore, in order to compute the ray intersection with all the planes of the whole set of voxels, the whole set of planes orthogonal to the x , y , and z axis is considered, i.e.:

- Each xy plane between $z = -\frac{N_{voxel}^z}{2} \cdot w_{voxel}$ and $z = \frac{N_{voxel}^z}{2} \cdot w_{voxel}$ stepping from one to another of w_{voxel} . The N_{voxel}^z is the number of voxels along the z dimension,

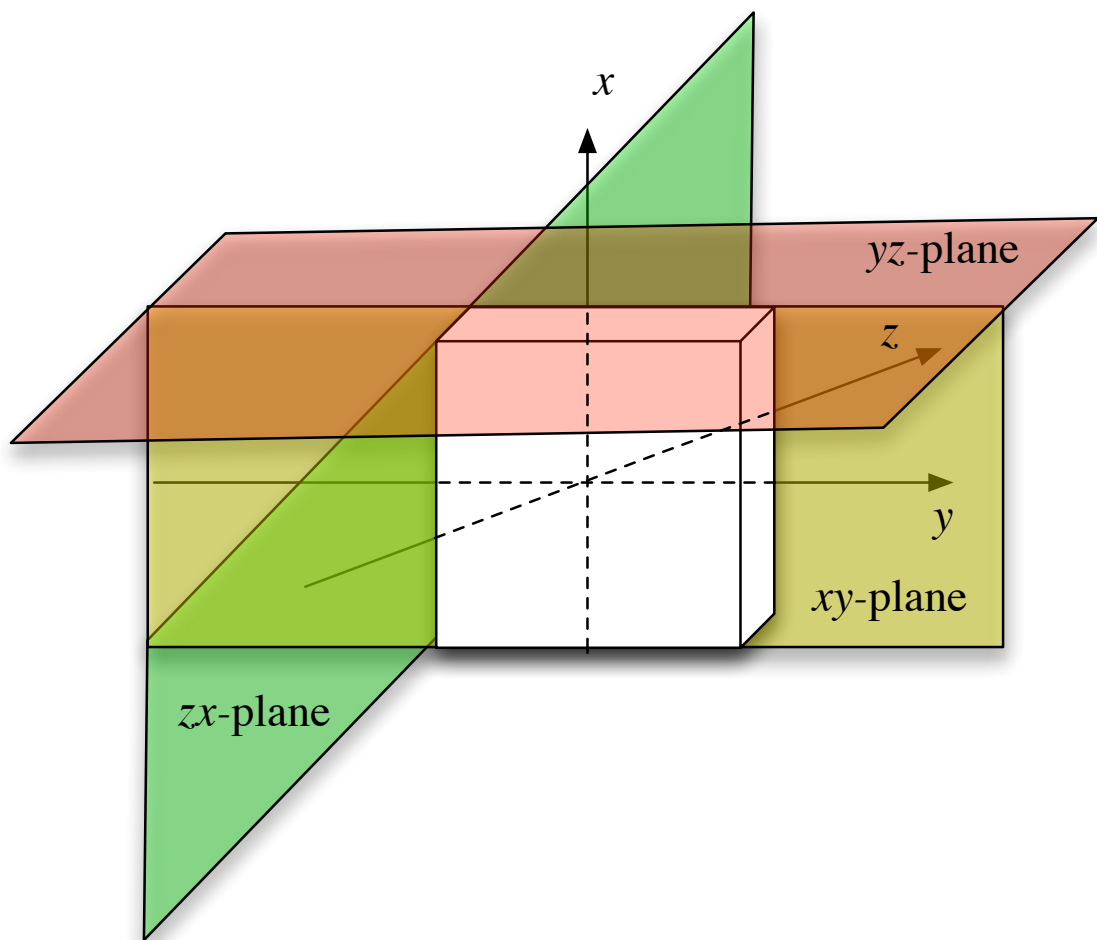


Figure B.2: Three planes delimiting a voxel

while w_{voxel} is the width of the voxel along the z dimension.

- Each yz plane between $x = -\frac{N_{voxel}^x}{2} \cdot w_{voxel}$ and $x = \frac{N_{voxel}^x}{2} \cdot w_{voxel}$ stepping from one to another of w_{voxel} . The N_{voxel}^x is the number of voxels along the x dimension, while w_{voxel} is the width of the voxel along the x dimension.
- Each zx plane between $y = -\frac{N_{voxel}^y}{2} \cdot w_{voxel}$ and $y = \frac{N_{voxel}^y}{2} \cdot w_{voxel}$ stepping from one to another of w_{voxel} . The N_{voxel}^y is the number of voxels along the y dimension, while w_{voxel} is the width of the voxel along the y dimension.

So for each plane value explored, that will be generically called k , the planes $x = k$, $y = k$, $z = k$ are considered at the same time. The intercept values are computed as follows. For the planes $x = k$ the intercept point (x_i, y_i, z_i) is:

$$\begin{cases} x_i = k \\ y_i = m_{yx}x_i + q_{yx} \\ z_i = m_{zy}y_i + q_{zy} \end{cases} \quad (B.25)$$

For the planes $y = k$ the intercept point (x_i, y_i, z_i) is:

$$\begin{cases} x_i = m_{xy}y_i + q_{xy} \\ y_i = k \\ z_i = m_{zx}x_i + q_{zx} \end{cases} \quad (B.26)$$

For the planes $z = k$ the intercept point (x_i, y_i, z_i) is:

$$\begin{cases} x_i = m_{xz}z_i + q_{xz} \\ y_i = m_{yx}x_i + q_{yx} \\ z_i = k \end{cases} \quad (B.27)$$

For each possible value of k , $-\frac{N_{voxel}}{2} \cdot w_{voxel} < k < \frac{N_{voxel}}{2} \cdot w_{voxel}$, the intercept points are exploited to compute the euclidean distance from the x-ray source. Therefore, for each intercept point (x_i, y_i, z_i) the distance from the x-ray source is.

$$D_{si} = \sqrt{(x_i - s_x)^2 + (y_i - s_y)^2 + (z_i - s_z)^2} \quad (B.28)$$

Once the three distances to the source from the three planes $x = k$, $y = k$ and $z = k$ have been calculated, the plane value k is updated to $k = k + w_{voxel}$ and the process is repeated for $x = k = k + w_{voxel}$, $y = k = k + w_{voxel}$ and $z = k = k + w_{voxel}$. All the I D_{si} distances (with I number of interceptions) are saved in

three separate arrays: one for interception with $x = k$ (**DistanceVector_{yz}**), one for interception with $y = k$ (**DistanceVector_{zx}**) and the last one for interception with $z = k$ (**DistanceVector_{xy}**). All these arrays are composed as follows: the distance term has been saved in the first row. The second, third and fourth rows of the vector contain the values x_i , y_i and z_i , so that each intercept point remains associated with the computed distance from the source. Subsequently, the three arrays are merged together in an unique **DistanceVector** array and this last one is sorted in order to have the smallest distance in the first array element. Subsequently, starting from the second-last array element, the ray paths in the voxels are computed as the difference among the source-intercept distances. Those value are saved in the first row of another array, **WeightsVector**:

$$\mathbf{WeightVector}(1, i) = \mathbf{DistanceVector}(1, i + 1) - \mathbf{DistanceVector}(1, i) \quad (\text{B.29})$$

For each column the coordinates x_{ci} , y_{ci} and z_{ci} of the center of the voxel associated to the weight in that column are assigned to the second, third and fourth row of the **WeightVector**. The coordinated x_{ci} , y_{ci} and z_{ci} are approximated as the medium point between $[x_i \ y_i \ z_i]$ and $[x_{i+1} \ y_{i+1} \ z_{i+1}]$

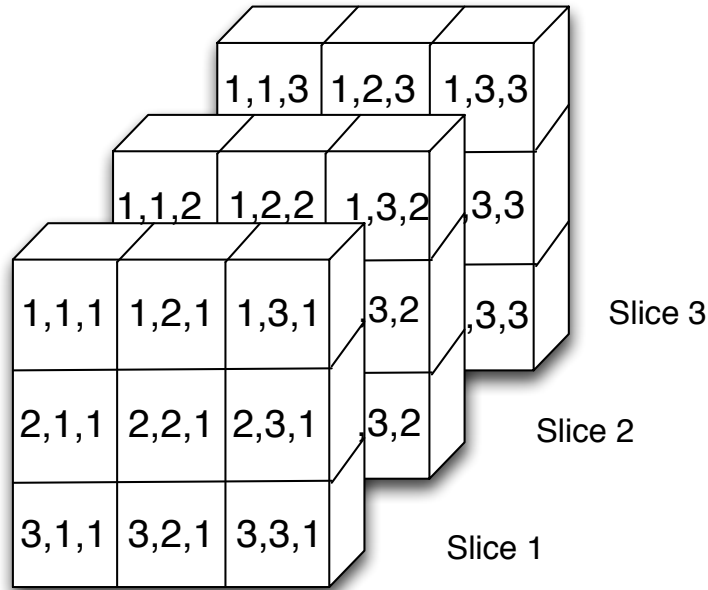
$$\begin{cases} x_{ci} = \frac{x_i + x_{i+1}}{2} \\ y_{ci} = \frac{y_i + y_{i+1}}{2} \\ z_{ci} = \frac{z_i + z_{i+1}}{2} \end{cases} \quad (\text{B.30})$$

The elements of **WeightVector** are the weights of the Static System Matrix. Subsequently they have to be assigned to the matrix in the correct position. The numeration given to the voxels in order to put them in a line within the Static System Matrix is the following: the voxels are taken row by row, and slice by slice, starting from the slice parallel to the plane zx , at $y = -N_{voxel}w_{voxel}$. A voxels numeration example is shown in Figure B.3(b).

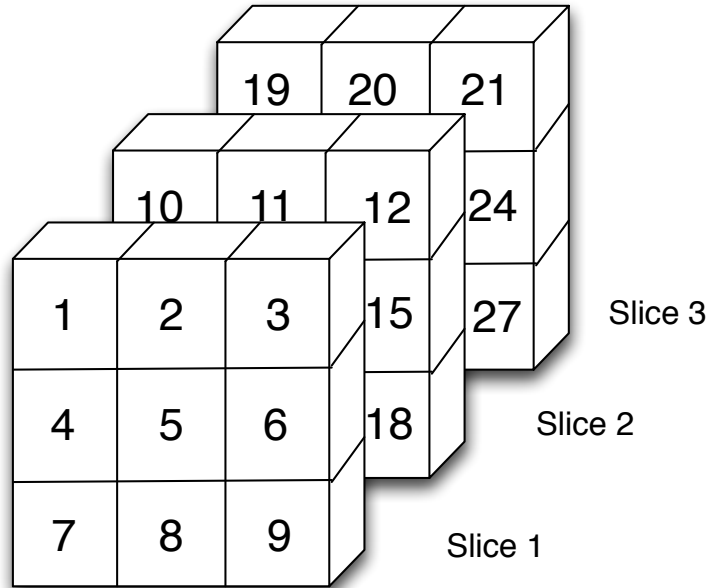
Taking as reference the numeration adopted, the indexing for the weights is computed starting from the voxel center data, in the following way:

$$\begin{cases} x_{index} = \text{ceil}(y_{ci} + N_{vy} \cdot \frac{w_{voxel}}{2}) \\ y_{index} = \text{ceil}(x_{ci} + N_{vx} \cdot \frac{w_{voxel}}{2}) \\ z_{index} = \text{ceil}(|z_{ci} - N_{vz} \cdot \frac{w_{voxel}}{2}|) \end{cases} \quad (\text{B.31})$$

where x_{index} , y_{index} and z_{index} correspond to the indexing illustrated in Figure B.3(a). Afterwards, the unique index I_v is computed as :



(a) Numeration using an index for each space dimension x , y and z ($x_{index}, y_{index}, z_{index}$)



(b) Numeration using a single unique index I_v

Figure B.3: Example, on a 3×3 matrix, of the voxel numeration adopted for the system matrix ordering

$$I_v = (y_{index} - 1)N_{vy} + (z_{index} - 1)N_{vz} + x_{index} \quad (\text{B.32})$$

Once the I_v is computed, the weight element associated to that index is placed in the System Matrix W in the row associated with the projection ray considered, and on the column associated with the index I_v . This overall process is repeated for each projection ray, until all the lines of the System Matrix are filled.

B.3 Dynamic System Matrix

In order to make the System Matrix suitable for the dynamic extension of the ART algorithm, the basis function samples must be inserted in the System Matrix, and only the needed voxels must be kept. In order to do this, the voxels to be kept are identified as the voxels with a non-zero value within the vascular three reconstruction obtained with the DSA. This comes straightforward from the Matlab function *find*. Giving as input the volumetric DSA reconstruction **DSA**, the indexes of the voxels of interest are obtained as

$$[i, j] = \text{find}(\mathbf{DSA}) \quad (\text{B.33})$$

for each image slice.

A global index (instead of the $[i, j]$ indexing) is obtained as in Equation B.32 already illustrated in the above section. Once the voxels of interest are defined, the static System Matrix is extended as dynamic. For each row i , and for each column j , the element $W(i, j)$ is replaced by a vector of length B , with B equal to the number of basis functions used: the vector is composed of the coefficient $W(i, j)$ multiplied by the vector of the basis function values at the time instant correspondent to the projection i , $\mathbf{b}(i)$.

$$W_d(i, j) = W(i, j) \cdot \mathbf{b}(i) = W(i, j) \cdot |b_{i1}b_{i2} \dots b_{iB}| \quad (\text{B.34})$$

For further explanation see Section 3.2.

B.4 Basis Functions

The basis functions chosen for this algorithm are the rectangular ones. This is due to the fact that they are suitable for modeling long transit times of the contrast medium, as normally is in clinic acquisition protocols. The number of basis functions must be decided a priori, depending on the contrast transit time and on the accuracy level

claimed. Once the number of basis functions has been chosen, the temporal acquisition period must be uniformly sampled with the basis functions. In order to guarantee an adequate sampling the width of the B rectangular basis functions is computed as:

$$w_{bf} = \frac{T_a}{B} \quad (\text{B.35})$$

where T_a is the acquisition time. The term w_{bf} is expressed in seconds, and it must be converted in samples. Therefore, defining the quantity Δt as the sampling period expressed in seconds, the width of the basis functions expressed in samples is:

$$w_{bf}^s = \frac{w_{bf}}{\Delta t} \quad (\text{B.36})$$

Then starting from the generic b^{th} basis function q_b , its rise time is set as:

$$t_r = q_b \cdot w_{bf}^s \quad (\text{B.37})$$

Each basis function signal is set to 1 between the rise time t_r and $t_r + w_{bf}$.

B.5 Segmentation

The segmentation process starts from the voxels reconstructed TICs. The area under the curve for arteries and veins is computed as:

$$AUC_a = \sum_{i=1}^{\frac{N_t}{2}} TIC_i \Delta t \quad (\text{B.38})$$

$$AUC_v = \sum_{i=\frac{N_t}{2}}^{N_t} TIC_i \Delta t \quad (\text{B.39})$$

and the total AUC is the sum of the expressions above:

$$AUC = AUC_a + AUC_v \quad (\text{B.40})$$

For each voxel the AUC_a and AUC_v are evaluated leading to the classification of arteries and veins. An artery and a vein mask are created, and each voxel j of the mask is set to one with the following criterion: if the AUC_a is greater then the 10% of the total AUC the voxel is classified as artery while if the voxel is not yet classified as an artery and has an AUC_a smaller then the 10% of the totale AUC the voxel is

classified as a vein.

$$\text{if } AUC_a > 0.1AUC \rightarrow \mathbf{ArteriesMask}(\mathbf{j}) = 1 \quad (\text{B.41})$$

$$\text{else if } \mathbf{ArteriesMask}(\mathbf{j}) = 0 \rightarrow \mathbf{VeinsMask}(\mathbf{j}) = 1 \quad (\text{B.42})$$

Bibliography

- [1] Alfred V Aho, Ravi Sethi, and Jeffrey D Ullman. *Compilers, Principles, Techniques*. Addison wesley, 1986.
- [2] AH Andersen and Avinash C Kak. Simultaneous algebraic reconstruction technique (sart): a superior implementation of the art algorithm. *Ultrasonic imaging*, 6(1):81–94, 1984.
- [3] Silvain Bériault, Abbas F Sadikot, Fahd Alsubaie, Simon Drouin, D Louis Collins, and G Bruce Pike. Neuronavigation using susceptibility-weighted venography: application to deep brain stimulation and comparison with gadolinium contrast: Technical note. *Journal of neurosurgery*, 121(1):131–141, 2014.
- [4] Mario Bertero and Patrizia Boccacci. *Introduction to inverse problems in imaging*. CRC press, 1998.
- [5] Michael Bock, Stefan O Schoenberg, Frank Floemer, and Lothar R Schad. Separation of arteries and veins in 3d mr angiography using correlation analysis. *Magnetic resonance in medicine*, 43(3):481–487, 2000.
- [6] William R Brody. Digital subtraction angiography. *Nuclear Science, IEEE Transactions on*, 29(3):1176–1180, 1982.
- [7] Michael H Buonocore, William R Brody, and Albert Macovski. A natural pixel decomposition for two-dimensional image reconstruction. *Journal of Computer Assisted Tomography*, 5(6):950, 1981.
- [8] Fernando Calamante. Arterial input function in perfusion mri: a comprehensive review. *Progress in nuclear magnetic resonance spectroscopy*, 74:1–32, 2013.
- [9] F Cardinale, A Miserocchi, A Moscato, M Cossu, L Castana, MP Schiariti, F Gozzo, G Pero, L Quilici, A Citterio, et al. Talairach methodology in the multimodal imaging and robotics era. *Stereotaxy and epilepsy neurosurgery. John Libbey Eurotext, London*, pages 245–272, 2012.

- [10] Francesco Cardinale, Massimo Cossu, Laura Castana, Giuseppe Casaceli, Marco Paolo Schiariti, Anna Miserocchi, Dalila Fuschillo, Alessio Moscato, Chiara Caborni, Gabriele Arnulfo, et al. Stereoelectroencephalography: surgical methodology, safety, and stereotactic application accuracy in 500 procedures. *Neurosurgery*, 72(3):353–366, 2013.
- [11] Francesco Cardinale, Guglielmo Pero, Luca Quilici, Mariangela Piano, Paola Colombo, Alessio Moscato, Laura Castana, Giuseppe Casaceli, Dalila Fuschillo, Luciana Gennari, et al. Cerebral angiography for multimodal surgical planning in epilepsy surgery: description of a new three-dimensional technique and literature review. *World neurosurgery*, 2015.
- [12] Allan Macleod Cormack. Representation of a function by its line integrals, with some radiological applications. *Journal of applied physics*, 34(9):2722–2727, 1963.
- [13] E De Momi, C Caborni, F Cardinale, G Casaceli, L Castana, M Cossu, R Mai, F Gozzo, S Francione, L Tassi, et al. Multi-trajectories automatic planner for stereoelectroencephalography (seeg). *International journal of computer assisted radiology and surgery*, 9(6):1087–1097, 2014.
- [14] Siyi Ding, Michael Miga, Thomas S Pheiffer, Amber L Simpson, Reid C Thompson, Benoit M Dawant, et al. Tracking of vessels in intra-operative microscope video sequences for cortical displacement estimation. *Biomedical Engineering, IEEE Transactions on*, 58(7):1985–1993, 2011.
- [15] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention, MICCAI, 98*, pages 130–137. Springer, 1998.
- [16] E Ward Gilman et al. *Webster’s dictionary of English usage*. Merriam-Webster, 1989.
- [17] L Goldman. Principles of ct and the evolution of ct technology. *Categorical Course in Diagnostic Radiology Physics: CT and US Cross-Sectional Imaging*. Oak Brook, IL: Radiological Society of North America, pages 33–52, 2000.
- [18] Jorge Gonzalez-Martinez, Jeffrey Mullin, Sumeet Vadera, Juan Bulacio, Gwyneth Hughes, Stephen Jones, Rei Enatsu, and Imad Najm. Stereotactic placement of depth electrodes in medically intractable epilepsy: Technical note. *Journal of neurosurgery*, 120(3):639–644, 2014.

- [19] Marc Guenot, Jean Isnard, Philippe Ryvlin, Catherine Fischer, Karine Ostrowsky, Franç ois Mauguiere, and Marc Sindou. Neurophysiological monitoring for epilepsy surgery: the talairach seeg method. *Stereotactic and functional neurosurgery*, 77(1-4):29–32, 2001.
- [20] Mehmet V Harput, Pablo Gonzalez-Lopez, and Ugur Türe. Three-dimensional reconstruction of the topographical cerebral surface anatomy for presurgical planning with free osirix software. *Operative Neurosurgery*, 10(3):426–435, 2014.
- [21] William H. Oldendorf. Isolated flying spot detection of radiodensity discontinuities: displaying the internal structural patten of a complex object. In *IEEE Transactions of Biomedical Electronic*, volume 8, pages 68–72, 1961.
- [22] GN Hounsfield. Historical notes on computerized axial tomography. *Journal of the Canadian Association of Radiologists*, 27(3):135–142, 1976.
- [23] Hsieh Jiang. *Computed tomography: principles, design, artifacts, and recent advances*. 2009.
- [24] Stefan Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l’Academie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.
- [25] Avinash C.. Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics, 2001.
- [26] IG Kazantsev, Rik Van de Walle, and Ignace Lemahieu. Ridge functions, natural pixels and minimal norm reconstruction. In *Nuclear Science Symposium, 1999. Conference Record. 1999 IEEE*, volume 3, pages 1309–1313. IEEE, 1999.
- [27] Christine Kilpatrick, Mark Cook, Andrew Kaye, Michael Murphy, and Zelko Matkovic. Non-invasive investigations successfully select patients for temporal lobe surgery. *Journal of Neurology, Neurosurgery & Psychiatry*, 63(3):327–333, 1997.
- [28] Cemil Kirbas and Francis Quek. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys (CSUR)*, 36(2):81–121, 2004.
- [29] David E. Kuhl and Roy O. Edwards. Reorganization data from transverse section scans of the brain using digital processing. In *Radiology*, volume 91, pages 975–983, 1968.

- [30] HOA Laue, MTH Oei, L Chen, IN Kompan, HK Hahn, M Prokop, and R Manniesing. Automated artery and vein detection in 4d-ct data with an unsupervised classification algorithm of the time intensity curves. In *SPIE Medical Imaging*, pages 86691W–86691W. International Society for Optics and Photonics, 2013.
- [31] T Lei, JK Udupa, PK Saha, D Odhner, and R Baum. Artery–vein separation using mr angiographic data: In 25 patients. In *Proc. ISMRM*, volume 3, page 1235, 1999.
- [32] Tianhu Lei, Jayaram K Udupa, Punam K Saha, and Dewey Odhner. 3d mr angiographic visualization and artery-vein separation. In *Medical Imaging’99*, pages 58–66. International Society for Optics and Photonics, 1999.
- [33] Tianhu Lei, Jayaram K Udupa, Punam K Saha, and Dewey Odhner. Artery-vein separation via mra-an image processing approach. *Medical Imaging, IEEE Transactions on*, 20(8):689–703, 2001.
- [34] Robert M Lewitt and Samuel Matej. Overview of methods for image reconstruction from projections in emission computed tomography. *Proceedings of the IEEE*, 91(10):1588–1611, 2003.
- [35] BF Logan, LA Shepp, et al. Optimal reconstruction of a function from its projections. *Duke Math. J*, 42(4):645–659, 1975.
- [36] Medtronic. www.medtronic.com.
- [37] Adriënne Mendrik, Evert-jan Vonken, Bram van Ginneken, Ewoud Smit, Annet Waaijer, Giovanna Bertolini, Max A Viergever, and Mathias Prokop. Automatic segmentation of intracranial arteries and veins in four-dimensional cerebral ct perfusion scans. *Medical physics*, 37(6):2956–2966, 2010.
- [38] JJ Mordang, MTH Oei, R van den Boom, EJ Smit, M Prokop, B van Ginneken, and R Manniesing. A pattern recognition framework for vessel segmentation in 4d ct of the brain. In *SPIE Medical Imaging*, pages 866919–866919. International Society for Optics and Photonics, 2013.
- [39] Alessandro Antonio Nacci, Vincenzo Rana, Francesco Bruschi, Donatella Sciuto, Ivan Beretta, and David Atienza. A high-level synthesis flow for the implementation of iterative stencil loop algorithms on fpga devices. In *Proceedings of the 50th Annual Design Automation Conference*, page 52. ACM, 2013.
- [40] Frank Natterer. *The mathematics of computerized tomography*, volume 32. Siam, 1986.

- [41] Frank Natterer et al. *Mathematical methods in image reconstruction*. Siam, 2001.
- [42] Wiro Niessen, Alexander Montauban van Swijndregt, Bernard Elsman, Onno Wink, Max Viergever, and Willem Mali. Enhanced artery visualization in blood pool mra: results in the peripheral vasculature. In *Information Processing in Medical Imaging*, pages 340–345. Springer, 1999.
- [43] Mark Nowell, Roman Rodionov, Gergely Zombori, Rachel Sparks, Gavin Winston, Jane Kinghorn, Beate Diehl, Tim Wehner, Anna Miserocchi, Andrew W McEvoy, et al. Utility of 3d multimodality imaging in the implantation of intracranial electrodes in epilepsy. *Epilepsia*, 56(3):403–413, 2015.
- [44] Ingerid Reinertsen, Maxime Descoteaux, Kaleem Siddiqi, and D Louis Collins. Validation of vessel-based registration for correction of brain shift. *Medical image analysis*, 11(4):374–388, 2007.
- [45] David W Roberts, Alexander Hartov, Francis E Kennedy, Michael I Miga, and Keith D Paulsen. Intraoperative brain shift and deformation: a quantitative analysis of cortical displacement in 28 cases. *Neurosurgery*, 43(4):749–758, 1998.
- [46] Mina G Safain, Jason P Rahal, Ami Raval, Mark J Rivard, John E Mignano, Julian K Wu, and Adel M Malek. Use of cone-beam computed tomography angiography in planning for gamma knife radiosurgery for arteriovenous malformations: A case series and early report. *Neurosurgery*, 74(6):682–696, 2014.
- [47] Hai Sun, David W Roberts, Alex Hartov, Kyle R Rick, and Keith D Paulsen. Using cortical vessels for patient registration during image-guided neurosurgery: a phantom study. In *Medical Imaging 2003*, pages 183–191. International Society for Optics and Photonics, 2003.
- [48] Jeannot Trampert and Jean-Jacques Leveque. Simultaneous iterative reconstruction technique: Physical interpretation based on the generalized least squares solution. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 95(B8):12553–12559, 1990.
- [49] Simon Webb. Historical experiments predating commercially available computed tomography. *Brain Journal of Radiography*, 65:835–837, 1992.
- [50] Jinxian Yuan, Yangmei Chen, and Edouard Hirsch. Intracranial electrodes in the presurgical evaluation of epilepsy. *Neurological sciences*, 33(4):723–729, 2012.
- [51] Dominik Zumsteg and Heinz Gregor Wieser. Presurgical evaluation: current role of invasive eeg. *Epilepsia*, 41(s3):S55–S60, 2000.

