UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE FÍSICA

Implementação de Algoritmos de Reconstrução de Imagens de Tomossíntese Utilizando Processamento Paralelo em GPU

Carlos Pereira Duarte

Mestrado Integrado em Engenharia Biomédica e Biofísica
Radiações em Diagnóstico e Terapia

Dissertação orientada por:
Prof. Doutor Nuno Matela

2016

Agradecimentos

A primeira pessoa a quem devo agradecer é claramente o meu orientador, o Professor Doutor Nuno Matela, pela sua disponibilidade e ajuda em todos os momentos deste trabalho. Mesmo quando ainda não tinha tema para a dissertação e precisava de orientação, foi a sua sugestão e colaboração que me levaram a escolher este projecto e a realizar este trabalho. De todos os professores com quem lidei ao longo do meu percurso académico, nenhum apresentou tanta disponibilidade e paciência para com os alunos como o professor Matela. Sem o seu contributo não me teria sido possível escrever esta dissertação.

Uma pessoa que também foi indispensável, e sem a qual, este trabalho não seria possível, foi o Pedro Ferreira. Ele deu início ao projecto no qual trabalhei, e auxiliou-me em todas as etapas do seu desenvolvimento. Também pela sua disponibilidade e amizade lhe agradeço.

Um agradecimento especial também para o professor Pedro Medeiros, da Faculdade de Ciências e Tecnologias da Universidade Nova, pela sua disponibilidade e aconselhamento na vertente informática e de programação em GPU, que me ajudaram bastante na fase inicial do trabalho.

Tenho também que agradecer aos professores e investigadores do IBEB, que me receberam durante este estágio, com especial agradecimento ao professor Alexandre Andrade, que me auxiliou em diversas situações e me incentivou a não desistir do curso de Engenharia Biomédica e a terminar esta dissertação. Aqui também deixo um agradecimento a todos os professores da Faculdade de Ciências que me ajudaram ao longo de todo o curso, e que me permitiram adquirir os conhecimentos e as bases necessárias a realizar este trabalho.

Gostaria também de agradecer à minha família, pelo apoio e dedicação que sempre demonstraram. Aos meus colegas e amigos, que me acompanharam nos últimos anos e com quem partilhei muitos momentos de trabalho e estudo mas também de descontracção e diversão. Finalmente devo agradecer à minha namorada, a pessoa mais importante na minha vida, que tem feito parte do meu percurso académico desde o início e sem a qual não teria conseguido terminar este trabalho.

Resumo

O cancro da mama é o tipo de cancro com maior incidência no género feminino, sendo considerado um dos maiores e mais importantes problemas de saúde pública à escala global. A mamografia é a técnica de imagem médica referência para o rastreio e diagnóstico do cancro da mama, no entanto tem associada algumas limitações bem conhecidas: elevada taxa de falsos-negativos (até 66% em mulheres sintomáticas) e falsos-positivos (até 60%). Estes dados estão sobretudo relacionados com o efeito da sobreposição de tecidos na imagem e têm vindo a gerar grande controvérsia na comunidade médica e científica, no que diz respeito ao uso da mamografia como técnica de rastreio, principalmente em mulheres mais jovens (< 50 anos).

A DBT (*Digital Breast Tomosynthesis*) é uma técnica radiológica tridimensional que produz uma pilha de imagens paralelas que representam as várias profundidades do tecido mamário, reduzindo assim o efeito de sobreposição. Existe actualmente evidência da redução das taxas de falsos negativos e de falsos positivos associados à utilização da DBT como técnica de rastreio, quando comparadas às da mamografia digital (FFDM - *Full-Field Digital Mammography*), sobretudo em mulheres com tecido mamário mais denso. As imagens são reconstruídas a partir de projecções bidimensionais recorrendo a algoritmos computacionais. Estes algoritmos têm um papel fundamental no processo de reconstrução, sobretudo em contexto clínico, uma vez que há a necessidade de implementar um processo que seja simultaneamente preciso e rápido. Actualmente os algoritmos mais utilizados para reconstruir imagens DBT em ambiente clínico são os analíticos, por apresentarem rapidez e simplicidade de execução. Os algoritmos iterativos têm, no entanto, demonstrado produzir imagens de melhor qualidade. O seu uso clínico é actualmente rejeitado devido ao seu elevado tempo de reconstrução e exigência computacional.

As capacidades computacionais das unidades de processamento gráfico (GPU - *Graphical Processing Units*), nomeadamente a capacidade de cálculo paralelo intensivo, foram já utilizadas por vários grupos de investigação para a optimização destes algoritmos iterativos. Um destes estudos foi desenvolvido no Instituto de Biofísica e Engenharia Biomédica (IBEB), na Faculdade de Ciências da Universidade de Lisboa, e consiste na implementação heterogénea (CPU+GPU) do processo de reconstrução de imagens DBT com algoritmos iterativos. Na GPU é executada um dos blocos mais exigentes de todo o algoritmo (o cálculo da matriz de sistema), por intermédio da linguagem de programação CUDA (*Compute Unified Device Architecture*). Esta implementação apresenta, no entanto, algumas falhas, que se reflectem em ligeiras alterações nas imagens, quando comparadas com as imagens reconstruídas pelo processo puramente sequencial

(CPU). O trabalho realizado na presente dissertação visa a correcção destas falhas, de modo a eliminar as diferenças observadas nas imagens reconstruídas, e a aplicação da abordagem heterogénea a outros blocos do algoritmo, de forma a optimizar ainda mais o processo de reconstrução.

Palavras-chave: Tomossíntese, Algoritmos Iterativos de Reconstrução, Computação em GPU, CUDA.

Abstract

Breast cancer is the most prevalent type of cancer in women, considered one of the largest and most relevant public health problems worldwide. Mammography is the state-of-the-art medical imaging technique for screening and diagnosis of breast cancer, but has some limitations: high rates of both false-positive and false-negative (up to 66 % and up to 60 %, respectively). This is mainly a result of the overlapping-tissue effect, a fact that has been generating great controversy in the medical-scientific community regarding the use of mammography as a screening technique, especially in younger women (< 50 years).

DBT (Digital Breast Tomosynthesis) is a three-dimensional x-ray technique that produces a parallel stack of images representing various depths of the breast volume, thereby reducing the overlapping effect. This technique has shown a reduction in false-negative and false-positive rates, when compared with FFMD (Full-Field Digital Mammography), especially in dense breasts. Images are reconstructed from two-dimensional projections using computer algorithms. These algorithms have a central role in the reconstruction process, especially in clinical context, since it is desirable to implement a process both accurate and fast. Currently the most popular algorithms in clinical setting are analytical, because of their fast and simple execution process. However, iterative algorithms have shown to produce better quality images. The problem is they require a lot of computational capability and thus take more time to reconstruct, making them unattractive to clinical implementation.

The computational capabilities of GPUs (Graphical Processing Units), namely intensive parallel computing, have been used by several research groups to optimize these iterative algorithms. One of these studies was developed at the Institute of Biophysics and Biomedical Engineering (IBEB), here at the Faculty of Sciences, and consists of heterogeneous implementation (CPU + GPU) of the DBT image reconstruction process with iterative algorithms. The GPU executes one of the heaviest blocks of the entire reconstruction process (the system matrix calculation), using CUDA (Compute Unified Device Architecture). Still, this implementation has some flaws, which are reflected in slight differences between the reconstructed images and the original ones (images reconstructed by the purely sequential process). My work aims at correcting these flaws in order to eliminate the image errors. Moreover, I intend to generalize this heterogeneous approach to another algorithm blocks, in order to further optimize the reconstruction process.

Keywords: Digital Breast Tomosynthesis, Iterative Reconstruction Algorithms, GPU Computing, CUDA.

Índice

Agradecimentos **i**

Resumo **iii**

Abstract **v**

Índice **vii**

Lista de Figuras **ix**

Lista de Tabelas **xi**

Acrónimos **xiii**

1 Introdução **1**
1.1 Apresentação da Dissertação 1
1.2 Enquadramento/Contextualização 4
1.3 Objectivos 4

2 Fisiologia e Patologia da Mama **7**
2.1 Anatomia e Fisiologia da Mama 7
2.2 Cancro da Mama 9

3 Imagiologia do Cancro da Mama **13**
3.1 Mamografia 15
3.1.1 Princípios Físicos 16
3.1.2 Geometria do Equipamento 17
3.1.3 Clínica e Dosimetria 18
3.1.4 Detectores 20
3.2 Tomossíntese 21
3.2.1 Processo de Aquisição 21
3.2.2 Dosimetria 23
3.2.3 Reconstrução de Imagens em Tomossíntese 24
3.2.3.1 Algoritmos de Reconstrução 28
SART 30
OS-EM e ML-EM 31
3.2.4 Cálculo da Matriz de Sistema 31

4 Computação em GPU **37**
4.1 Contextualização Histórica 37
4.2 Programação em GPU 40
4.3 CUDA 41
4.3.1 Biblioteca Thrust 44

5 Metodologia **47**
5.1 Sistema DBT 47
5.2 Implementação Sequencial 49
5.3 Implementação Heterogénea 53
5.4 Correcção da Implementação Heterogénea 59

6 Resultados e Discussão **65**
6.1 Avaliação das Imagens Reconstruídas 65
6.2 Avaliação do Tempo de Reconstrução 66

7 Considerações Finais **69**

Referências **73**

Lista de Figuras

2.1 Anatomia da mama 8

3.1 Geometria de aquisição na mamografia 18
3.2 Projecções cranio-caudal e médio-lateral oblíqua 19
3.3 Comparação entre imagens de mamografia e DBT 22
3.4 Geometria de aquisição em DBT 23
3.5 Esquema representativo do processo de estimação 25
3.6 Algoritmo iterativo 29
3.7 Interacção dos feixes com a FOV 33
3.8 Cálculo das distâncias (2D) 33
3.9 Atribuição das distâncias aos vóxeis da FOV 35

4.1 Arquitecturas CPU e GPU 39
4.2 Hierarquia de execução em CUDA 42

5.1 Especificações técnicas da aquisição DBT 48
5.2 Esquema do algoritmo de reconstrução iterativo 51
5.3 Integração CUDA-IDL 53
5.4 Divisão do detector em blocos 57
5.5 Artefactos nas imagens 59

6.1 Resultados das correcções na qualidade das imagens 66

Lista de Tabelas

2.1 Estadiamento do cancro da mama 10

3.1 Categorias BI-RADS 19

6.1 Resultados da optimização temporal 67

Acrónimos

API	(do inglês ***A**pplication **P**rogramming **I**nterface*)
ART	(do inglês ***A**lgebraic **R**econstruction **T**echnique*)
BI-RADS	(do inglês ***B**reast **I**maging - **R**eporting **A**nd **D**ata **S**ystem*)
CPU	Unidade Central de Processamento (do inglês ***C**entral **P**rocessing **U**nit*)
CUDA	(do inglês ***C**ompute **U**nified **D**evice **A**rchitecture*)
DBT	Tomossíntese (do inglês ***D**igital **B**reast **T**omosynthesis*)
DCIS	Carcinoma Local dos Ductos (do inglês ***D**uctal **C**arcinoma **I**n **S**itu*)
DIC	Carcinoma Invasivo dos Ductos (do inglês ***D**uctal **I**nvasive **C**arcinoma*)
FBP	Retroprojecção Filtrada (do inglês ***F**iltered **B**ack**P**rojection*)
FDA	(do inglês ***F**ood and **D**rug **A**dministration*)
FFDM	Mamografia Digital (do inglês ***F**ull-**F**ield **D**igital **M**ammography*)
FOV	Campo de Visão (do inglês ***F**ield **O**f **V**iew*)
GPU	Unidade de Processamento Gráfico (do inglês ***G**rafics **P**rocessing **U**nit*)
IDL	(do inglês ***I**teractive **D**ata **L**anguage*)
ILP	(do inglês ***I**nstruction **L**evel **P**arallelism*)
LCIS	Carcinoma Local dos Lóbulos (do inglês ***L**obular **C**arcinoma **I**n **S**itu*)
LIC	Carcinoma Invasivo dos Lóbulos (do inglês ***L**obular **I**nvasive **C**arcinoma*)

LOR Linha de Resposta (do inglês ***L**ine **O**f **R**esponse*)

ML-EM (do inglês ***M**aximum **L**ikelihood – **E**xpectation **M**aximization*)

MS **M**atriz de **S**istema

OS-EM (do inglês ***O**rdered **S**ubsets – **E**xpectation **M**aximization*)

RM **R**essonância **M**agnética

SAA (do inglês ***S**hift-**A**nd-**A**dd*)

SART (do inglês ***S**imultaneous **A**lgebraic **R**econstruction **T**echnique*)

TC **T**omografia **C**omputorizada

Capítulo 1

Introdução

1.1 Apresentação da Dissertação

O cancro da mama é uma doença conhecida da humanidade há muitos séculos, no entanto apenas passou a ter um impacto social à escala global há cerca de 50 anos, com o desenvolvimento da mamografia. É actualmente o tipo de cancro mais prevalente e o segundo mais mortífero entre as mulheres na generalidade dos países ocidentais. Nos últimos 15 anos, com a implementação de rastreios periódicos à população de risco, tem-se verificado um aumento significativo do número de casos diagnosticados mas uma redução da mortalidade associada ao cancro da mama. A mamografia tem a capacidade de diagnosticar a doença antes do início dos sintomas ou da presença de qualquer massa palpável, no entanto a sua utilização como técnica de rastreio tem causado grande controvérsia, sobretudo devido à sua fraca especificidade em mulheres mais jovens. A principal limitação apontada à mamografia relaciona-se com o efeito da sobreposição de tecidos e a fraca capacidade que apresenta em distinguir estruturas tridimensionais.

A tomossíntese (DBT - *Digital Breast Tomosynthesis*) surge como alternativa à mamografia, criando soluções para as estas limitações: imagens em três dimensões, sem sobreposição de estruturas e com doses de radiações equivalentes à mamografia. A DBT é hoje utilizada em contexto clínico, na maioria das vezes como complemento à mamografia. Esta técnica acaba por não ser tão popular devido principalmente ao maior tempo de execução e análise das imagens. Os algoritmos de reconstrução utilizados

em DBT associados a uma maior qualidade de imagem e um melhor diagnóstico (algoritmos iterativos) são computacionalmente muito exigentes e demorados, tendo sido preteridos pelos algoritmos analíticos, mais rápidos e simples, mas com menor qualidade de imagem.

A reconstrução realizada pelos algoritmos iterativos requer a execução de uma tarefa computacionalmente exigente - o cálculo da matriz de sistema (MS). Este cálculo ocupa a maior parte do tempo despendido em todo o processo de reconstrução. No entanto é uma operação que pode ser agilizada pela abordagem paralela: os elementos desta matriz podem ser calculados de forma independente.

As unidades de processamento gráfico (GPU - *Graphical Processing Units*) foram inicialmente desenvolvidas com o objectivo de realizar renderização de gráficos num monitor, e têm evoluído bastante desde então, sobretudo devido ao desenvolvimento da indústria dos jogos de vídeo. No entanto, o interesse na sua utilização para a realização de computação de âmbito geral (*general-purpose*) tem crescido a uma velocidade incrível na última década. As GPUs têm sido procuradas principalmente pela sua grande capacidade de computação paralela intensiva. Este fenómeno tornou-se ainda mais evidente com o aparecimento de linguagens baseadas em C/C++ que permitiram uma programação mais directa e intuitiva das GPUs, como é o caso da linguagem CUDA[1](*Compute Unified Device Architecture*). Desde então têm havido um aumento significativo das aplicações da computação em GPU nas mais diversas áreas, que não exclusivamente renderização de gráficos. De entre essas aplicações destaca-se a imagem médica, onde as unidades GPU têm sido utilizadas como auxiliares na computação de grandes quantidades de dados.

Foi desenvolvida no Instituto de Biofísica e Engenharia Biomédica (IBEB), uma alternativa de implementação do processo de reconstrução em DBT com algoritmos iterativos, utilizando computação heterogénea (CPU+GPU) [1]. A computação do cálculo da MS é realizado na GPU de forma integrada com a implementação sequencial no CPU (*Central Processing Unit*), já desenvolvida. Este projecto surge com o objectivo de acelerar os algoritmos iterativos, de modo a permitir a sua utilização em ambiente clínico e assim melhorar a qualidade das imagens em DBT. Esta optimização pode eventualmente permitir a utilização da DBT como modalidade independente da mamografia.

[1]CUDA é uma marca registada da NVIDIA.

A implementação heterogénea apresenta resultados promissores, com uma redução do tempo total de reconstrução de 1,6 vezes, no entanto contém alguns erros de concepção. Estes erros manifestam-se a partir de ligeiras diferenças existentes entre as imagens reconstruídas e as imagens originais (reconstruídas com a implementação puramente sequencial).

O trabalho desenvolvido nesta dissertação baseia-se na correcção destes erros e na aplicação da computação em GPU a outros blocos do algoritmo de reconstrução, de forma a optimizar ainda mais o processo. A dissertação encontra-se organizada em **7** capítulos, com a seguinte distribuição:

- Capítulo 1 - o presente capítulo introduz a dissertação e enquadra o leitor no tema discutido;
- Capítulo 2 - este capítulo realiza uma revisão da literatura introduzindo alguns conceitos teóricos necessários à compreensão deste trabalho no que diz respeito à anatomia e fisiologia da mama (seção 2.1) e também uma breve introdução ao cancro da mama (secção 2.2);
- Capítulo 3 - este capítulo aborda as várias técnicas de imagem da mama, dando especial ênfase à mamografia (secção 3.1) e à tomossíntese (secção 3.2);
- Capítulo 4 - este capítulo introduz o leitor ao tema da computação em GPU, realizando primeiro uma contextualização histórica (secção 4.1), e depois abordando alguns conceitos básicos sobre GPUs (secção 4.2) e CUDA (secção 4.3);
- Capítulo 5 - neste capítulo é apresentada a metodologia seguida na realização deste trabalho. São aqui apresentadas as especificações do dispositivo DBT, os trabalhos realizados anteriormente por outros autores, as alterações realizadas à implementação heterogénea e de que modo como foi realizada a avaliação dos resultados;
- Capítulo 6 - aqui são apresentados os resultados deste trabalho. É realizada uma avaliação da implementação no que diz respeito à qualidade das imagens reconstruídas (secção 6.1) e optimização dos tempos de reconstrução (secção 6.2);
- Capítulo 7 - por fim são apresentadas as conclusões do trabalho, feito um sumário e apresentadas algumas considerações finais e perspectivas de trabalho futuro.

1.2 Enquadramento/Contextualização

Este trabalho encontra-se inserido num projecto de investigação financiado pela FCT (Fundação para a Ciência e Tecnologia) e desenvolvido numa colaboração entre o IBEB, o Hospital da Luz, o Instituto Superior Técnico, o LIP (Laboratório de Instrumentação e Física Experimental de Partículas) de Lisboa e de Coimbra. Este projecto está em curso desde Junho de 2013 e intitula-se "Melhoria da Qualidade de Imagem e Redução de Dose em Tomossíntese para Mamografia, com Recurso a Algoritmos Estatísticos de Reconstrução de Imagem". A unidade de investigação do IBEB encontra-se dividida em várias linhas temáticas e a presente dissertação insere-se na linha temática intitulada *"Medical Imaging and Diagnosis"*.

No contexto deste projecto de investigação foi desenvolvida uma implementação em IDL[2] (*Interactive Data Language*) que executa o processo de reconstrução das imagens de DBT, recorrendo a algoritmos iterativos (ART, ML-EM ou OS-EM). Foi a partir desta implementação que Ferreira integrou o cálculo da MS em GPU. Esta nova implementação heterogénea foi também desenvolvida no IBEB, em parceria com a FCT-UNL (Faculdade de Ciências e Tecnologias - Universidade Nova de Lisboa), no âmbito da sua dissertação de mestrado [1].

O estudo desenvolvido na presente dissertação teve início em Novembro de 2014 e surge no contexto da continuidade do trabalho desenvolvido por Ferreira na implementação referida anteriormente. Numa fase inicial foi necessária a contextualização da informação e o estudo das técnicas de programação em GPU, sendo que a frequência do curso online coursera, de programação em CUDA, foi fundamental no processo de aprendizagem. Esta fase serviu essencialmente para reunir os conhecimentos e a prática necessária à realização do trabalho proposto, tanto ao nível da programação em GPU, quer ao nível da manipulação do código IDL.

1.3 Objectivos

Este trabalho tem como objectivo principal dar continuidade aos bons resultados obtidos com a implementação heterogénea do processo de reconstrução em DBT. Esta

[2]IDL é uma marca registada de Exelis Visual Information Systems, Inc.

optimização não só demonstra o enorme potencial da computação em GPU em imagem médica como também disponibiliza uma alternativa viável ao rastreio e diagnóstico do cancro da mama: a utilização de algoritmos iterativos na reconstrução de imagens DBT em ambiente clínico.

Neste contexto, os objectivos concretos deste trabalho resumem-se essencialmente a dois pontos:

- Identificar os erros da implementação heterogénea responsáveis pelos artefactos na imagem e se possível corrigi-los. É de extrema importância a identificação do problema na medida em que é necessário garantir que esta implementação não prejudica a qualidade das imagens, distorcendo a informação nelas contida;
- Optimizar a utilização das capacidades de computação paralela da GPU, implementando a mesma metodologia noutros blocos do algoritmo que possam ser paralelizados.

Capítulo 2

Fisiologia e Patologia da Mama

2.1 Anatomia e Fisiologia da Mama

O estudo adequado da anatomia e fisiologia da mama e das suas estruturas adjacentes é de grande importância na compreensão das patologias associadas, nomeadamente o cancro da mama.

A mama, ou seio, é um órgão par, de forma e dimensão vaiável, situado na zona anterior do tórax, sobre a fáscia do músculo peitoral maior [2]. É constituída essencialmente pela glândula mamária (uma glândula sudorípara modificada) e tecido adiposo. É considerada como pertencente ao sistema reprodutivo e encontra-se mais desenvolvida no género feminino. O seu desenvolvimento inicia-se no período pré-natal e é interrompido durante a infância. Durante a puberdade, devido às alterações hormonais, dá-se o seu maior desenvolvimento. A sua principal função é a produção e transmissão de leite durante a amamentação do recém-nascido [3].

A mama é constituída por dois tipos principais de tecido: fibroglandular e adiposo. A porção glandular corresponde às glândulas mamárias, divididas em vários lóbulos, e a porção fibrosa é constituída por fibras de tecido conjuntivo que lhes conferem sustentação. Cada lóbulo da glândula mamária possui várias estruturas saculares de tecido epitelial, os alvéolos, responsáveis pela produção de leite durante a amamentação. O número de alvéolos e o seu estado de desenvolvimento é regulado pela presença de estrogénio e progesterona [4]. Os ductos lactíferos são os canais que transportam o leite dos lóbulos até aos orifícios exteriores, no mamilo. Todas estas estruturas encontram-se

envolvidas por tecido adiposo [5]. À medida que o corpo envelhece, o tecido fibroso é gradualmente substituído por tecido adiposo, diminuindo a densidade da mama [3]. A figura 2.1 representa esquematicamente os vários constituintes da mama e a sua organização.

A mama é revestida exteriormente por pele e apresenta uma zona de coloração mais escura, na região central, de formato cilíndrico, onde se localizam os orifícios dos ductos lactíferos - o mamilo (ou papila). A aréola é a região periférica ao mamilo, com uma coloração semelhante, composta essencialmente por glândulas sebáceas especializadas [2].

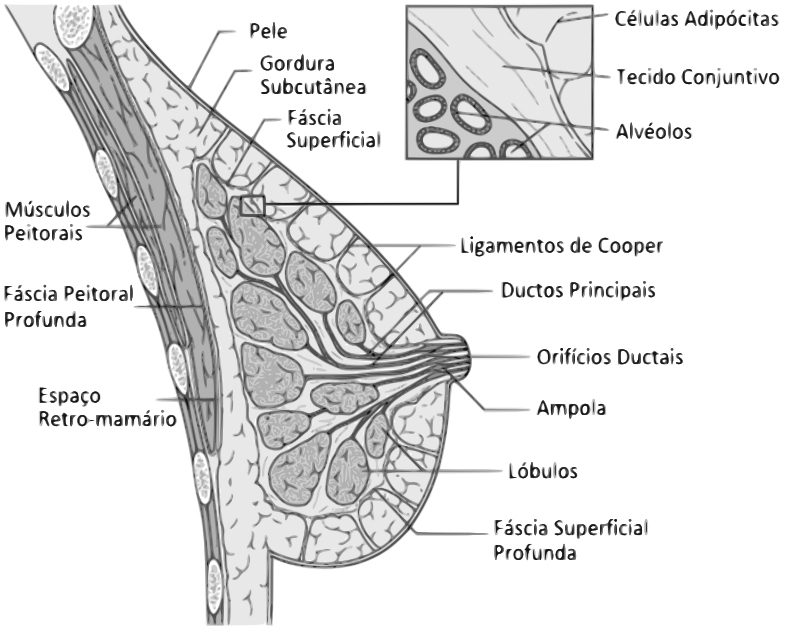

FIGURA 2.1: Esquema anatómico da mama feminina [3].

A rede de nervos, vasos sanguíneos e vasos linfáticos, presentes no tecido conjuntivo fibroso, constituem também uma componente importante do tecido mamário. Os nódulos linfáticos desempenham um papel particularmente importante no diagnóstico e estadiamento do cancro da mama, pois são locais preferenciais de invasão por parte de células cancerígenas e muitas vezes lugar de metástases. A rede linfática axilar representa a principal via de drenagem linfática na mama (97 a 99%) [2]. Por vezes é realizada uma biópsia ao nódulo sentinela, com o objectivo de identificar os nódulos afectados, assim que é identificado um carcinoma na mama [6]. O conhecimento da rede de vasos

linfáticos da mama é, por isso, extremamente importante no estudo e tratamento do cancro da mama.

2.2 Cancro da Mama

O cancro é uma doença que se caracteriza pelo crescimento anormal e descontrolado de células. Estas células dividem-se e espalham-se pelos tecidos saudáveis adjacentes, podendo eventualmente proliferar por todo o organismo, interferindo com as funções fisiológicas normais dos tecidos e órgãos saudáveis, podendo levar progressivamente à morte do indivíduo [7]. Existem vários tipos de cancro, consoante o tipo de tecido e a sua localização no organismo. A sua identificação e caracterização são fundamentais, pois delas dependem o método de abordagem e possível tratamento.

O cancro da mama pode ser classificado em vários tipos, consoante o local de aparecimento das células tumorais. A maioria dos cancros da mama são carcinomas, um tipo de cancro que tem início nas células epiteliais. Em rigor, são chamados de adenocarcinomas - carcinomas que surgem em tecido glandular. No entanto existem outros tipos de cancro que podem ocorrer na mama, como o sarcoma, que tem origem em células de tecido conjuntivo, adiposo ou muscular [8]. Podem ainda ser classificados consoante o nível de invasão do tumor: local (*in situ*) ou invasivo. Os carcinomas locais são os que permanecem dentro dos lóbulos (LCIS-*Lobular Carcinoma In Situ*) ou dos ductos (DCIS-*Ductal Carcinoma In Situ*). Enquanto que os carcinomas invasivos (LIC-*Lobular Invasive Carcinoma*, DIC-*Ductal Invasive Carcinoma*) penetram a membrana basal e proliferam para o tecido envolvente [6]. Os carcinomas invasivos representam cerca de 80% de todos os carcinomas da mama, e por sua vez, a grande maioria dos carcinomas invasivos são do tipo DIC (aproximadamente 70-80%) [9].

O estadiamento do cancro da mama é essencial quando existe um diagnóstico. Este procedimento envolve a determinação da extensão da doença na mama afectada, avaliando os nódulos linfáticos locais e a presença de metástases distantes [10]. Um dos sistemas de classificação mais utilizados pelos profissionais de saúde é o sistema TNM (Tumor-Nódulo-Metástase) [11]. Este sistema classifica um cancro a partir de três parâmetros diferentes: T - que pode tomar uma classificação entre 1 e 4, consoante a dimensão do tumor; N - que pode assumir os valores entre 0 a 3, consoante a presença de células

tumorais nos nódulos linfáticos locais, a sua localização e aderência a tecidos vizinhos; e M - que assume os valores 1 ou 0, dependendo da existência, ou não, de evidência de metástases distantes, respectivamente. Assim, com base nesta classificação é possível dividir o cancro da mama em cinco estádios (0, I, II, III e IV) e respectivos sub-estádios (tabela 2.1).

Tabela 2.1: Estadiamento do cancro da mama segundo o sistema TNM. **Tis** é referente a *in situ*. Adaptado de [11].

Estágio	**T**	**N**	**M**
0	Tis	N0	M0
I	T1	N0	M0
IIA	T0	N1	M0
	T1	N1	
	T2	N0	
IIB	T2	N1	M0
	T3	N0	
IIIA	T0	N2	M0
	T1		
	T2		
	T3	N1	
	T3	N2	
IIIB	T4	N0	M0
	T4	N1	
	T4	N2	
IIIC	Qualquer T	N3	M0
IV	Qualquer T	Qualquer N	M1

O cancro da mama pode ainda ser unifocal, multifocal (duas ou mais lesões localizadas no mesmo quadrante) ou multicêntrico (lesões localizadas em mais do que um quadrante). Adicionalmente, pode ser classificado como unilateral ou bilateral (se afectar as duas mamas) [12]. No caso de existirem múltiplas lesões, a classificação TNM é sempre elaborada tendo por base a lesão de maior dimensão.

O cancro da mama é o segundo tipo de cancro mais comum em todo o mundo e o quinto mais mortífero. Considerando apenas o género feminino é, de longe, o mais comum com cerca de 1,6 milhões de novos casos todos os anos (25% de todos os casos de cancro em mulheres) e também o com maior taxa de mortalidade (cerca de 500 mil por ano) [13]. É também um dos mais investigados e discutidos no panorama científico mundial. A sua etiologia não é totalmente conhecida, no entanto têm sido identificados alguns factores de risco como a idade, a história familiar (factores genéticos), a exposição a hormonas, contraceptivos orais, obesidade e consumo de álcool [6, 14]. Tem-se observado também

um aumento no aparecimento de cancro da mama em mulheres expostas a grandes quantidades de radiação [15].

Cerca de 6 a 10 % dos cancros da mama são hereditários, e estão maioritariamente associados a mutações dos genes BRCA1 e BRCA2 (*BReast CAncer 1 and 2*). A probabilidade de uma mulher vir a desenvolver cancro da mama depende da mutação específica que se observa num destes genes [16].

Para ser possível a detecção do cancro da mama é necessário que a paciente apresente sintomas, o que muitas vezes só acontece num estado já avançado da doença. No entanto, existe a opção de rastreio à população, que pode ser feito através do exame físico (palpação), ou recorrendo a técnicas imagiológicas.

Capítulo 3

Imagiologia do Cancro da Mama

Existem várias modalidades de imagem da mama: mamografia, ecografia, elastografia, ressonância magnética, espectroscopia de impedância eléctrica (EIS - *Electrical Impedance Spectroscopy*), de microondas (MIS *Microwave Imaging Spectroscopy*), e também de infravermelhos (NIS - *Near Infrared Spectroscopic Imaging*) e até mesmo técnicas de medicina nuclear como tomografia de emissão de positrões (PET - *Positron Emission Tomography*) [17]. Cada uma destas técnicas apresenta vantagens e desvantagens, e encontra-se indicada para uma determinada situação específica.

A ecografia é muitas vezes utilizada como exame de seguimento de uma massa previamente detectada pela mamografia ou ainda como técnica auxiliar a biópsias ou outras intervenções [18]. Tem a capacidade de distinguir massas tumorais de quistos e é muitas vezes utilizada como exame de investigação de massas palpáveis em mulheres grávidas, ou em jovens com tecido mamário denso [19, 20]. A ecografia é uma técnica com bastantes vantagens, uma vez que os ultra-sons apresentam baixo custo e não possuem radiação ionizante, no entanto possuiu algumas limitações: pouca resolução, que impossibilita a identificação e caracterização de lesões de menor dimensão; depende da destreza do operador; não permite a diferenciação precisa entre lesões benignas e malignas [17, 21].

A elastografia mamária é uma técnica de ultra-sonografia utilizada para avaliar a elasticidade dos tecidos e que tem vindo a ser introduzida para melhorar a caracterização e distinção entre lesões malignas e benignas. Esta técnica permite a avaliação da rigidez de lesões, a partir da deformação dos tecidos, quando aplicada uma compressão externa. Tal como a ecografia, a elastografia é uma técnica de baixo custo, que não implica a

utilização de radiação ionizante, contudo é por vezes difícil a diferenciação entre lesões malignas e benignas. As limitações associadas são a elasticidade dos tecidos que pode não corresponder às características reais das lesões e a força de compressão executada, que varia consoante o operador, apresentando-se como outro factor que influencia o diagnóstico e a elasticidade dos tecidos [21].

A ressonância magnética mamária tem sido indicada como uma das técnicas mais promissoras, com várias aplicações. Algumas dessas aplicações são consensuais e indicadas como vantajosas em determinadas situações, tais como: diagnóstico de lesões tumorais primárias ocultas a outras técnicas de imagem [22], determinação da extensão de lesões [23], avaliação da resposta à quimioterapia [24], diagnóstico de recorrências [25] e rastreio de pacientes de alto risco [22]. As restantes aplicações são actualmente alvo de controvérsia e discussão e estão presentes sobretudo em ambiente de investigação [26]. A maior limitação da ressonância magnética mamária prende-se com a sua baixa especificidade, que, em conjunto com uma elevada sensibilidade, pode levar a biopsias desnecessárias e aumento da ansiedade das pacientes.

A espectroscopia por RM baseia-se numa sequência específica, que permite a detecção dos elevados níveis de colina produzidos pelos tumores malignos. A colina é uma molécula necessária para a síntese de ácidos gordos nas membranas celulares [27, 28]. Esta técnica apresenta grande potencial como complemento à RM mamária convencional, aumentando a sua especificidade, no entanto possui uma grande limitação relacionada com o facto de nem todos os tumores expressarem colina [29].

As técnicas de medicina nuclear proporcionam uma análise funcional das lesões da mama a um nível celular e metabólico. Estas técnicas estão limitadas no que diz respeito à resolução espacial, apresentando pouca capacidade de detecção de lesões de pequena dimensão, no entanto, com o desenvolvimento de equipamentos exclusivamente dedicados à mama, tem sido possível detectar lesões da ordem do centímetro. Estas modalidades são bastante promissoras para as situações já indicadas para a RM: determinação da extensão da lesão, rastreio de pacientes de alto risco e avaliação da resposta à terapêutica. No entanto, e ao contrário do que acontece na RM, as técnicas de medicina nuclear expõem as pacientes a radiação ionizante [30–32].

A mamografia destaca-se de entre todas as outras, sendo universalmente aceite como a mais indicada para o rastreio e diagnóstico do cancro da mama [17].

3.1 Mamografia

A mamografia é sobretudo eficaz na detecção de lesões numa fase precoce de desenvolvimento, com grande potencial de reduzir a mortalidade. É uma técnica de custo reduzido e com boa resolução de imagem, permitindo a identificação de lesões de reduzidas dimensões. Apresenta, no entanto, algumas desvantagens relacionadas com a sobreposição de estruturas na imagem, consequência da sua natureza bidimensional. Esta limitação produz dois efeitos na capacidade dos radiologistas em distinguir lesões nas imagens de mamografia: por um lado, uma lesão maligna pode ser dissimulada por tecido fibroglandular que se encontre sobreposto, produzindo um falso-negativo; por outro lado, a sobreposição de tecido fibroglandular normal pode produzir a ilusão da presença de uma lesão que não existe, criando assim um falso-positivo [33]. As pacientes estão também expostas a doses de radiação, uma vez que a mamografia é uma técnica que utiliza radiação ionizante [34].

A utilização de radiação para o estudo de lesões da mama remonta ao início do século XX, por Salomon [35], em 1913. Desde então surgiram vários estudos de lesões benignas e malignas da mama, recorrendo à radiação X [36, 37], e o primeiro protótipo de um equipamento especificamente dedicado à mamografia é construído por Egan [38], na década de 60. Egan melhorou significativamente a qualidade da imagem ao utilizar uma fonte de raio-X de alta amperagem e baixa voltagem e um detector composto por filme fotográfico industrial intercalado com camadas de ecrã intensificador. Com a introdução destas alterações, Egan tornou também possível a disseminação global desta tecnologia, que até então se encontrava pouco conhecida. Desde então foram realizados vários ensaios clínicos randomizados que mostraram uma redução significativa da mortalidade (cerca de 30%) associada ao cancro da mama em mulheres rastreadas com mamografia [39]. Esta redução verificou-se sobretudo em mulheres com idades superiores a 50 anos, enquanto que em mulheres mais novas (tipicamente com tecido mamário mais denso) a diferença de taxas de mortalidade entre grupos rasteados e não rastreados não se mostrou estatisticamente significativo [40]. Este assunto tem sido alvo de muita discussão e controvérsia, no entanto, a maioria dos autores concorda com o benefício do rastreio para mulheres entre os 50 e 70 anos de idade [40–42]. A maioria dos países ocidentais têm promovido programas de rastreio do cancro da mama com exames de mamografia,

de dois em dois anos, a mulheres com idade superior a 50 anos, com base nos resultados obtidos nestes ensaios clínicos [41].

3.1.1 Princípios Físicos

A mamografia, tal como acontece na radiografia convencional, tem por base a interacção da radiação X com os diferentes tecidos biológicos. A radiação X consiste em ondas electromagnéticas com comprimentos de onda inferior à luz visível (entre 0,01 e 10 nm), bastante mais energética e com capacidade de ionização. Ao interagir com os electrões atómicos presentes no tecido biológico, os fotões podem sofrer três processos distintos: transmissão, dispersão e absorção. Devido à dispersão e absorção, o feixe incidente é atenuado e perde intensidade. A atenuação A sofrida por um feixe de raio-X, ao atravessar um determinado tecido, é dada pela lei de Beer-Lambert:

$$A = \frac{I}{I_0} = e^{-\mu_m \rho x} = e^{-\mu x}, \tag{3.1}$$

onde I representa a intensidade do feixe medida no detector depois de atravessar uma espessura x de tecido com um coeficiente linear de atenuação μ. I_0 representa a intensidade do feixe incidente não atenuado. O termo μ_m representa o coeficiente de atenuação mássico e ρ a densidade do tecido. Como a atenuação compreende dois processos distintos, a absorção e a dispersão, o cálculo do coeficiente μ é determinado pela ponderação destes dois componentes:

$$\mu = \mu_a + \mu_s, \tag{3.2}$$

onde μ_a representa o coeficiente linear de absorção e μ_s representa o coeficiente linear de dispersão [43].

Como se pode concluir pela equação 3.1, a atenuação dos feixes de radiação está dependente da densidade (ρ) e espessura (x) dos tecidos atravessados mas também da sua composição em termos atómicos (μ_m). Ou seja, não só um tecido mais denso ou mais

espesso vai ser representado com maior intensidade na imagem, mas também um tecido formado por elementos de massa atómica mais elevada. Por exemplo, uma calcificação, que contém um elemento relativamente pesado (Ca), vai ser representada na imagem com uma intensidade superior ao tecido fibroglandular, que apenas contém elementos orgânicos mais leves.

O contraste C criado na imagem devido à atenuação da radiação em dois tecidos diferentes ($\mu_1 \neq \mu_2$) ao longo da mesma espessura x é definido pela seguinte expressão:

$$C = \ln \frac{A_2}{A_1} = \ln \frac{e^{-\mu_2 x}}{e^{-\mu_1 x}} = (\mu_1 - \mu_2)x \ . \tag{3.3}$$

Todas estas relações encontram-se expressas para um feixe de radiação mono-energético, no entanto deve ser considerado o espectro de energias característico da radiação X, e realizada uma ponderação dos vários valores de μ, uma vez que este é dependente da energia do feixe. A sua relação é inversamente proporcional: quanto maior a energia do feixe, menor o valor de μ e consequentemente o contraste C [43].

3.1.2 Geometria do Equipamento

Os raios-X são produzidos na fonte (ou ampola) e emitidos em cone na direcção da mama. Os feixes são atenuados ao atravessar os vários tecidos e a sua intensidade é registada no detector. O sistema é disposto de forma a que a mama da paciente se encontre sempre no trajecto percorrido pelos feixes de raio-X desde a fonte até ao detector (figura 3.1). Devido ao formato de cone formado pelo feixe de raio-X, as estruturas são ampliadas na imagem formada no detector [44].

De modo a obter uma melhor qualidade de imagem e uma menor exposição à radiação, a mamografia é adquirida com compressão dos tecidos. Isto é conseguido colocando a mama da paciente entre duas plataformas planas que a comprimem ligeiramente. Este procedimento permite não só uma diminuição da espessura de tecido, como também a sua homogeneização, permitindo uma penetração uniforme dos feixes de radiação [45]. Outra das vantagens da compressão relaciona-se com a dispersão das estruturas da mama ao longo de uma área superior, diminuindo a sua sobreposição. Este procedimento é

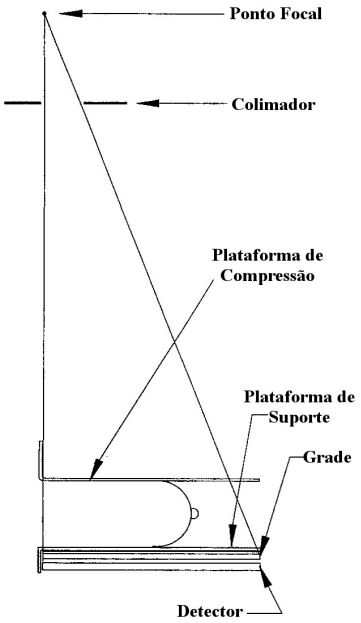

FIGURA 3.1: Esquema da geometria de aquisição da imagem em mamografia [44].

bastante desconfortável para as pacientes, podendo mesmo chegar a ser doloroso [45–47].

A aquisição da imagem em mamografia pode ser realizada em várias posições, tendo em conta a incidência mais adequada ao que se pretende visualizar. São várias as incidências possíveis mas as mais utilizadas no rastreio do cancro da mama são a cranio-caudal (CC) e médio-lateral oblíqua (MLO). No caso concreto destas incidências, a paciente encontra-se em posição vertical (de pé). Na prática clínica, estas duas projecções são normalmente utilizadas em conjunto com o objectivo de produzir alguma noção, mesmo que limitada, da localização tridimensional das estruturas observadas. A geometria da aquisição destas duas projecções encontra-se representada na figura 3.2. Existem também incidências mais específicas, como por exemplo a médio-lateral (ML) e a cranio-caudal exagerada lateralmente (CCEL), que são utilizadas em casos especiais [48].

3.1.3 Clínica e Dosimetria

O Colégio Americano de Radiologia desenvolveu um sistema de classificação universal dos resultados das mamografias: BI-RADS (*Breast Imaging Reporting and Data System*). Este sistema foi criado com o objectivo de categorizar os resultados das mamografias (tabela 3.1), e assim criar uma base universal de auxílio à tomada de decisão dos clínicos

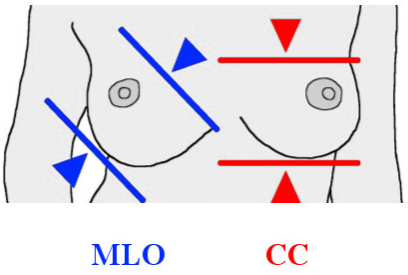

FIGURA 3.2: Representação esquemática das projecções cranio-caudal (CC) e médio-lateral oblíqua (MLO). As setas representadas na figura indicam a direcção de compressão [49].

[50]. A escala BI-RADS foi concebida originalmente para a mamografia, mas a sua utilização tem sido estendida a técnicas como a RM e ecografia mamárias.

TABELA 3.1: Nesta tabela estão representadas as várias categorias BI-RADS, com as recomendações e prognósticos correspondentes. **P de Malignidade** - Probabilidade de Malignidade; **N/A** - Não Aplicável. Adaptado de [50]

	Categoria	**Recomendação**	**P de Malignidade**
0 -	Inconclusivo	Realização de exame imagiológico adicional	N/A
1 -	Negativo	Rastreio de rotina com mamografia	0%
2 -	Benigno	Rastreio de rotina com mamografia	0%
3 -	Provavelmente Benigno	Seguimento e rastreio de curta duração (6 meses) com mamografia	<2%
4 -	Suspeito	Diagnóstico histológico	2-95%
5 -	Altamente Sugestivo de Malignidade	Diagnóstico histológico	>95%
6 -	Malignidade Confirmada	Excisão cirúrgica	N/A

A dose de radiação recebida por um determinado tecido é a quantidade de energia absorvida, por unidade de massa desse tecido. A unidade SI de dose é o Gray (Gy = J/Kg). A maioria dos equipamentos de mamografia expõe as pacientes a uma dose glandular média de 1 a 2 mGy por projecção [51]. O Colégio Americano de Radiologia recomenda que uma mama com 4.2 cm de espessura não deve ser exposta a uma dose glandular média superior a 3 mGy por imagem.

3.1.4 Detectores

A mamografia convencional (ou analógica) utiliza detectores fabricados com filmes fotográficos. Estes filmes funcionam como meios de detecção da radiação X e, ao mesmo tempo, como suporte de armazenamento da informação. No entanto, apesar do seu desenvolvimento nas últimas décadas, os filmes fotográficos possuem um gradiente de intensidades limitado, tornando difícil a identificação de lesões localizadas em regiões correspondentes aos limites deste intervalo de intensidades (zonas muito opacas ou muito brilhantes) [44]. A mamografia digital vem superar esta limitação, uma vez que cada processo (aquisição, visionamento e armazenamento das imagens) é realizado em separado. Assim, a imagem pode ser adquirida com determinadas características e manipulada posteriormente (*à posteriori*), consoante as necessidades do radiologista. Esta técnica, mais conhecida como mamografia digital de campo total (FFDM - *Full-Field Digital Mammography*) utiliza uma matriz de detectores electrónicos fotossensíveis, que possibilita a redução da intensidade da radiação, conservando a qualidade de imagem [52]. Nos últimos anos tem vindo a ganhar popularidade sobre a vertente analógica, uma vez que também apresenta menos custos e maior facilidade de manipulação de imagem [53].

Para além da exposição à radiação ionizante, a mamografia apresenta também outras desvantagens. A mais evidente prende-se com o facto de existir uma sobreposição de tecidos na imagem, principal responsável pela elevada taxa de falsos-negativos (entre 4% a 34%) [34, 54, 55]. Assumindo uma taxa de falsos positivos de 30% e uma taxa de redução da mortalidade por rastreio com mamografia de 15%: seria necessário o rastreio a 2000 mulheres assintomáticas durante 10 anos para impedir uma morte por cancro da mama, ao mesmo tempo que 10 mulheres saudáveis iriam receber tratamento desnecessário, e outras 200 iriam experienciar stress psicológico e ansiedade por vários anos devido a achados falsos-positivos [41]. A sensibilidade desta técnica é sobretudo baixa em mulheres com tecido mamário denso. O desconforto devido à compressão da mama e a variabilidade na interpretação radiológica são também desvantagens importantes relacionadas com a mamografia [8, 17].

O aparecimento e implementação da FFDM permitiu também o desenvolvimento de novas técnicas ainda mais sofisticadas, como é o caso da tomossíntese.

3.2 Tomossíntese

A tomossíntese mamária (DBT - *Digital Breast Tomosynthesis*) é uma técnica de imagem médica tomográfica que se baseia na mamografia digital. Na DBT as imagens são obtidas a partir de um conjunto de projecções, adquiridas à medida que a fonte de raios-X descreve um movimento em arco em torno da mama [55, 56]. Esta técnica, muitas vezes chamada de mamografia 3D, permite a reconstrução do volume tridimensional da mama e a sua visualização em cortes individuais [57]. Esta capacidade de visualização das várias estruturas a diferentes profundidades elimina o efeito da sobreposição de tecidos e permite ao radiologista distinguir entre uma verdadeira massa e uma ilusão criada pela sobreposição de várias estruturas presentes em diferentes planos [58]. A figura 3.3 ilustra a situação descrita: é observada uma alteração na imagem de mamografia (3.3.a), no entanto, nas imagens de tomossíntese (de 3.3.b a 3.3.d) observa-se que a alteração não passa de uma soma de sinal correspondente a estruturas sobrepostas.

Os fundamentos teóricos da tomossíntese foram estabelecidos nos anos 30 do século XX [59], mas a sua concretização e aplicação clínica só se tornaram possíveis várias décadas mais tarde com a criação de detectores digitais (*flat-panel digital display detectors*), com o desenvolvimento do processamento informático e com o avanço dos algoritmos de reconstrução e pós-processamento [60]. Esta modalidade tornou-se disponível para uso clínico em 2009, foi aprovada pela FDA[1] em 2011 e é actualmente utilizada em contexto clínico em várias regiões do mundo, incluindo os EUA, Canada, Ásia e Europa [56]. Particularmente em Portugal é já utilizada na maioria dos hospitais e clínicas como técnica de diagnóstico do cancro da mama.

3.2.1 Processo de Aquisição

A aquisição das projecções em DBT tem por base os mesmos princípios que a mamografia. A diferença fundamental encontra-se na geometria do equipamento e no movimento que descreve durante a aquisição. Durante o processo de aquisição, a fonte de raio-X emite vários disparos (entre 9 e 25) de baixa dose ao longo do movimento em arco que descreve em torno da mama (figura 3.4) [57]. Este movimento, ao contrário do que acontece na TC (*Tomografia Computorizada*), tem uma amplitude limitada (de 15º a

[1] DBT foi aprovada pela FDA (Food and Drug Administration) nos EUA, como técnica complementar à mamografia [61].

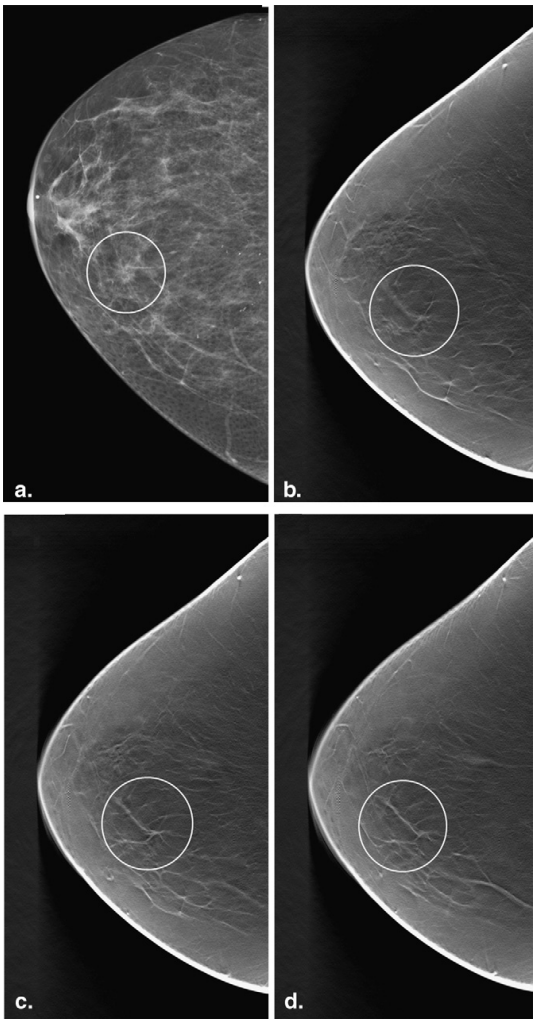

FIGURA 3.3: Comparação de imagem de mamografia, contendo um falso-positivo, com as imagens de tomossíntese correspondentes. **(a)** Imagem de mamografia, vista cranio--caudal, onde se observa uma alteração na arquitectura glandular (*círculo*). **(b-d)** Imagens de tomossíntese que ilustram os cortes cranio-caudais individuais, onde se observa a arquitectura normal do tecido fibroglandular na mesma região (*círculo*) [57].

50º) e pode ser contínuo ou discreto (*step-and-shot*). Durante todo este processo, a mama, o sistema de compressão, e o detector mantêm-se imóveis. No final do processo de aquisição são obtidas várias imagens, projecções que correspondem às várias posições da fonte. As projecções são posteriormente utilizadas no processo de reconstrução para criar um volume tridimensional da mama, recorrendo a algoritmos computacionais [56].

Estes algoritmos computacionais de reconstrução produzem um conjunto de imagens paralelas com 1mm de espessura, e permitem aos radiologistas visualizar e percorrer estas imagens ao longo da direcção vertical (ou cranio-caudal). Estas imagens são normalmente visualizadas em formato de vídeo como uma sequência de imagens consecutivas

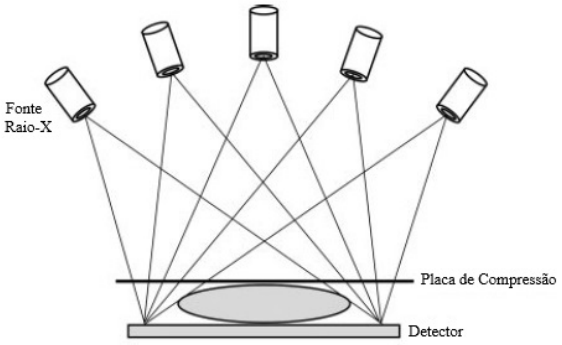

FIGURA 3.4: Representação esquemática do movimento da fonte de raio-X em torno da mama, num exame de tomossíntese [54].

[33].

A tomossíntese é tipicamente utilizada no rastreio e diagnóstico do cancro da mama como técnica complementar à mamografia. Ambas as modalidades utilizam o mesmo equipamento e a mesma compressão do tecido mamário. Em muitos locais que utilizam mamografia começa a surgir a tendência da substituição desta técnica pela DBT, sobretudo desde que os fabricantes começaram a disponibilizar a criação de imagens de mamografia sintetizadas digitalmente a partir do volume DBT reconstruído. Esta imagem sintetizada permite uma análise prévia do estado da doença e poupa as pacientes à realização dos dois exames.

3.2.2 Dosimetria

A dose de radiação correspondente a cada projecção de DBT é relativamente reduzida, fazendo com que a dose total à qual as pacientes são expostas durante uma aquisição completa (15 projecções) seja comparável à de uma aquisição de FFDM [58]. No entanto, em contexto de rastreio, quando são utilizadas as duas técnicas em simultâneo, a dose total do exame aumenta para o dobro. Esta situação é ainda mais evidente em mulheres mais jovens ou com tecido mamário mais denso. As novas técnicas de DBT que permitem a criação da imagem 2D sintética a partir do volume DBT reconstruído limitam assim a dose à de um exame apenas [56].

A utilização de DBT como técnica complementar à mamografia no diagnóstico do cancro da mama apresenta várias vantagens:

- Ajuda na detecção de alterações na arquitectura fibroglandular, que são muitas vezes imperceptíveis na imagem de mamografia [62];
- Melhora a capacidade de delineação das lesões, tornando a sua caracterização mais adequada e a classificação BI-RADS mais precisa.
- Permite a localização espacial da lesão sem necessidade de recorrer às duas projecções da mamografia.

Apesar da evidencia que suporta o uso da DBT como complemento à mamografia no diagnóstico do cancro da mama, a sua utilização como técnica de rastreio apresenta algumas limitações: dose de radiação ligeiramente superior à FFDM; maior custo associado à tecnologia de aquisição e armazenamento; uma maior dificuldade de interpretação por parte dos clínicos; e tempos de aquisição e visualização superiores [55].

A qualidade das imagens em DBT está altamente dependente da geometria do sistema e escolha dos parâmetros de aquisição, e visualização óptimos, mas sobretudo do processo de reconstrução.

3.2.3 Reconstrução de Imagens em Tomossíntese

O processo de reconstrução possui um papel fundamental no desempenho das modalidades de imagem tomográficas, como DBT ou TC. De um modo geral, este processo baseia-se na criação de uma estimativa tridimensional de um objecto desconhecido a partir de projecções bidimensionais, obtidas de várias perspectivas (ou ângulos). É um processo de elevada complexidade, uma vez que implica a inferência de dados desconhecidos. Esta questão é especialmente importante em DBT, uma vez que o número de projecções é menor e o ângulo de rotação da fonte é mais limitado, comparativamente ao que acontece em TC.

O termo *tomografia* deriva da palavra grega *tomos* (corte ou secção) e significa o processo pelo qual se obtém uma imagem das estruturas internas de um objecto sólido através do registo das diferentes atenuações sofridas pelas ondas que atravessam essas estruturas.

O processo de reconstrução encontra-se representado no esquema da figura 3.5 de uma forma muito simplificada. Este esquema representa um objecto composto por quatro

elementos de densidades desconhecidas. Se considerarmos cada uma das setas como sendo um feixe de raios-X que atravessa este objecto em diferentes locais, são registados quatro valores no detector ((1), (2), (3) e (4)).

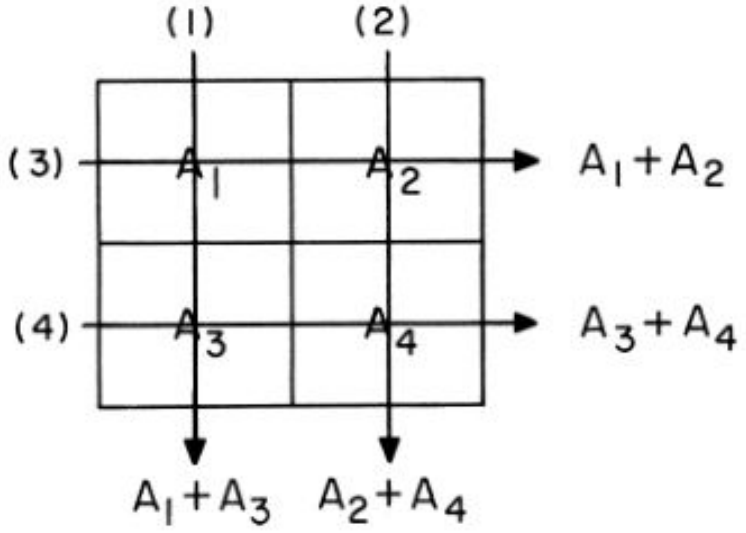

FIGURA 3.5: Esquema simplificado do processo de estimação dos valores do coeficiente de atenuação do objecto atravessado pelos feixes de raio-X, a partir das projecções tomográficas. [63].

Os valores registados são conhecidos e representam a atenuação sofrida por cada um dos feixes de raio-X, ao atravessarem o objecto. Os valores A1, A2, A3 e A4 são desconhecidos e representam a contribuição de cada um dos elementos do objecto para a atenuação total do feixe. O seguinte conjunto de equações representa as relações entre eles:

$$\begin{aligned} (1) &= A_1 + A_3 \\ (2) &= A_2 + A_4 \\ (3) &= A_1 + A_2 \\ (4) &= A_3 + A_4 \end{aligned} \quad . \tag{3.4}$$

Uma vez que estamos na presença de um sistema de quatro equações independentes com quatro incógnitas, é possível calcular os valores exactos das contribuições para a atenuação de cada um dos elementos do objecto.

Este exemplo aqui representado ilustra a reconstrução de uma imagem com quatro elementos (ou píxeis), necessitando por isso do registo de quatro medições de atenuação independentes. Generalizando, a reconstrução de uma imagem com n píxeis necessita do registo de n medições de atenuação independentes. Isto também significa que quanto maior o número de registos independentes detectados, maior a resolução da imagem reconstruída.

A generalização deste exemplo para três dimensões acontece quando lidamos com a reconstrução de um volume tridimensional a partir de projecções bidimensionais. Logo é necessário adicionar uma terceira dimensão ao problema, mas o princípio mantém-se.

Este exemplo ilustra de uma forma muito simples o conceito do processo de reconstrução de imagens em técnicas tomográficas. A técnica ilustrada neste exemplo intitula-se de retroprojecção: a informação contida numa medição é projectada no sentido retrógrado ao longo do percurso percorrido pelo feixe do qual foi obtida. Este método relativamente simples apresenta no entanto algumas desvantagens, como a criação de imagens difusas (com o efeito *blur*). Este efeito diminui naturalmente com o aumento do número de projecções.

Esta implementação é considerada analítica, uma vez que obtém os dados reconstruídos com uma aplicação única de operações analíticas a cada medição. Existe no entanto outra abordagem ao problema da reconstrução, a implementação iterativa, que modifica sucessivamente as estimativas que são retroprojectadas até à obtenção de uma imagem satisfatória.

Independentemente do método escolhido, a reconstrução deve ser encarada matematicamente como a resolução de um problema inverso, ilustrado na seguinte equação linear:

$$Y = Ax + \eta, \tag{3.5}$$

onde Y representa o conjunto dos dados adquiridos, A a matriz de sistema (secção 3.2.4) que representa o modelo geométrico da transmissão e detecção da radiação, x o conjunto dos dados tridimensionais a serem estimados e η o ruído (ruído associado à dispersão da

radiação X e ruído electrónico). No entanto, como estamos a lidar com várias projecções, a equação 3.5 passa a ser expressa da seguinte forma:

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_n \\ \vdots \\ A_N \end{bmatrix} x, \tag{3.6}$$

onde N representa o número total de projecções e n uma determinada projecção, sendo que $1 < n < N$. De modo a simplificar o problema, η não é contabilizado. Considerando agora todos os píxeis I de uma dada projecção Y_n, e o número de vóxeis J da imagem estimada x, a equação 3.6 passa a ter a seguinte apresentação:

$$\begin{bmatrix} y_1 & \cdots & y_i & \cdots & y_I \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1} & \cdots & a_{Ij} & \cdots & a_{IJ} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_J \end{bmatrix}. \tag{3.7}$$

Na equação 3.7, y_i representa o i-ésimo píxel da projecção Y_n, x_j o j-ésimo vóxel de x, e a_{ij} o elemento da matriz de sistema correspondente à combinação do píxel i com o vóxel j. O valor de y_i é então dado por:

$$y_i = \sum_j a_{ij} x_j \ . \tag{3.8}$$

Para que seja possível a obtenção do coeficiente de atenuação correspondente a cada vóxel, é necessário recorrer à lei de Lambert-Beer (equação 3.1):

$$\ln\left(\frac{I_0}{I}\right)_i = \sum_j \mu_j x_{ij}, \tag{3.9}$$

onde $\ln\left(\frac{I_0}{I}\right)_i$ representa a atenuação sofrida pelo feixe de raio-X correspondente à direcção i (R_i), que é definida, para cada posição da fonte (ou projecção), e pelo *i-ésimo* elemento (bin) do detector; μ_j é o coeficiente de atenuação no vóxel j; e x_{ij} representa a probabilidade de um fotão R_i ser absorvido no vóxel j.

É de fácil conclusão que o valor de μ_j não é calculável apenas com uma medição, dada o elevado número de parâmetros desconhecidos na equação 3.9, mas sim com a contribuição de múltiplas projecções.

A partir da atribuição de diferentes tons de cinzento a diferentes intervalos de valores de μ, é possível criar o contraste no volume reconstruído e assim distinguir as várias estruturas que o constituem.

O algoritmo de reconstrução deve então ter como objectivo a resolução da equação 3.5, que se traduz para o contexto de DBT na equação 3.9.

3.2.3.1 Algoritmos de Reconstrução

Como já foi referido anteriormente, os algoritmos de reconstrução podem ser divididos em duas grandes categorias: analíticos e iterativos.

Os algoritmos analíticos baseiam-se em modelos de dados matematicamente idealizados, tornando demasiado simples a física dos processos subjacentes, limitando assim o detalhe das imagens reconstruídas. Existem dois algoritmos analíticos muito utilizados em tomossíntese: *Shift-and-Add* (SSA) e *Filtered BackProjection* (FBP) [64].

Os algoritmos iterativos podem ser divididos em dois grandes grupos: algébricos e estatísticos. Geralmente os algoritmos iterativos têm uma estrutura comum e seguem um determinado conjunto de passos (figura 3.6):

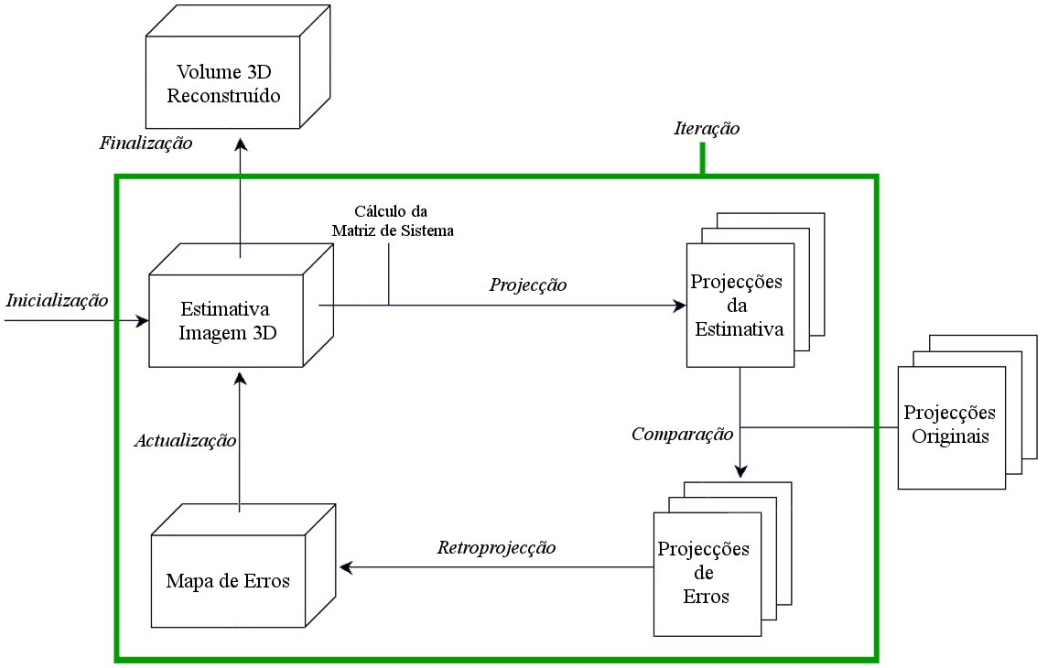

FIGURA 3.6: Diagrama esquemático que representa um algoritmo de reconstrução iterativo.

- *Inicialização* - o processo iterativo começa com uma imagem inicial estimada, normalmente uma constante;
- *Projecção* - é realizada uma operação de projecção à estimativa actual, de modo a obter um conjunto de projecções estimadas;
- *Comparação* - depois é feita uma comparação entre as projecções estimadas e as projecções originais (obtidas na aquisição), obtendo-se as projecções de erros;
- *Retroprojecção* - é aplicado a estas projecções uma operação de retroprojecção de modo a obter um mapa de erros;
- *Actualização* - o mapa de erros é utilizado para actualizar a estimativa da iteração anterior;
- Este processo é repetido iterativamente até ser obtida uma imagem tão próxima das projecções originais quanto desejável;

Os algoritmos utilizados neste projecto são o algoritmo algébrico SART (*Simultaneous Algebraic Reconstruction Technique*), e os algoritmos estatísticos ML-EM (*Maximum*

Likelihood – Expectation Maximization) e OS-EM (*Ordered Subsets – Expectation Maximization*).

SART O algoritmo SART (*Simultaneous Algebraic Reconstruction Technique*) é um caso particular do algoritmo ART clássico (*Algebraic Reconstruction Technique*). Estes algoritmos abordam o problema da reconstrução como um conjunto de equações lineares (equações 3.5 a 3.8). O algoritmo ART clássico é bastante rápido a convergir para uma solução, no entanto produz muito ruído e não se apresenta como implementação adequada a GPU, uma vez que cada iteração apenas processa uma linha de projecção de cada vez [65]. O algoritmo SART, por outro lado, permite o cálculo de múltiplas linhas de projecção em simultâneo. A seguinte equação representa o modelo iterativo do algoritmo SART, e como as estimativas da iteração seguinte x_j^{k+1} são calculadas a partir das estimativas da iteração actual x_j^k:

$$x_j^{k+1} = x_j^k + \lambda \frac{\sum\limits_{y_i \in Y_n} a_{ij} \left(\frac{y_i - \sum\limits_j a_{ij} x_j^k}{\sum\limits_j a_{ij}} \right)}{\sum\limits_{y_i \in Y_n} a_{ij}}, \tag{3.10}$$

onde y_i representa o i-ésimo píxel da projecção original Y_n e λ representa o factor de relaxamento. É possível realizar uma associação entre os vários elementos desta equação e o esquema 3.6:

- $\sum\limits_j a_{ij} x_j^k$ - *projecção*;
- $y_i - \sum\limits_j a_{ij} x_j^k$ - *comparação*;
- $\frac{\sum\limits_{y_i \in Y_n} a_{ij} \left(\frac{y_i - \sum\limits_j a_{ij} x_j^k}{\sum\limits_j a_{ij}} \right)}{\sum\limits_{y_i \in Y_n} a_{ij}}$ - *retroprojecção*;
- $x_j^k + \lambda \frac{\sum\limits_{y_i \in Y_n} a_{ij} \left(\frac{y_i - \sum\limits_j a_{ij} x_j^k}{\sum\limits_j a_{ij}} \right)}{\sum\limits_{y_i \in Y_n} a_{ij}}$ - *Actualização*.

OS-EM e ML-EM Os algoritmos OS-ML e ML-EM são bastante semelhantes, na medida em que o primeiro é uma derivação do segundo. O algoritmo OS-EM agrupa as projecções em subgrupos, de modo a optimizar o tempo de cálculo [66]. O OS-EM é mais rápido a convergir, no entanto permite o aparecimento de ruído nas reconstruções mais cedo do que o ML-EM. O modelo iterativo destes algoritmos encontra-se representado nas seguintes equações:

ML-EM

$$x_j^{k+1} = \frac{x_j^k}{\sum\limits_i a_{ij}} \sum_i a_{ij} \frac{Y_i}{\sum\limits_t a_{it} x_t^k}, \tag{3.11}$$

OS-EM

$$x_j^{k+1} = \frac{x_j^k}{\sum\limits_{i \in S_n} a_{ij}} \sum_{i \in S_n} a_{ij} \frac{Y_i}{\sum\limits_t a_{it} x_t^k}, \tag{3.12}$$

onde também x_j^k e x_j^{k+1} representam as estimativas da iteração actual e seguinte, respectivamente. Repare-se que a única diferença entre as duas equações encontra-se nos *subsets* de projecções S_n criados no algoritmo OS-EM com o objectivo de diminuir o tempo de convergência. De forma análoga ao que foi descrito anteriormente para o algoritmo SART, é também possível fazer uma associação entre as equações 3.11-3.12 e o esquema da figura 3.6.

A implementação de qualquer um destes algoritmos requer uma etapa fundamental - o cálculo da matriz de sistema. Esta matriz representa o modelo geométrico dos processos de transmissão e detecção, e contém informação sobre cada projecção.

3.2.4 Cálculo da Matriz de Sistema

A matriz de sistema (MS) é um elemento fundamental no processo de reconstrução de qualquer algoritmo iterativo pois descreve o modelo geométrico da transmissão e detecção dos feixes de radiação. O cálculo da MS não está dependente do tipo de algoritmo, mas sim da geometria do sistema DBT.

Neste trabalho, o cálculo da MS foi implementado tendo por base a aproximação *ray driven* [67, 68]. A matriz é construída então, para cada posição da fonte de raio-X (ou projecção), da seguinte forma:

1. São considerados feixes de radiação individuais (R_i). Cada feixe é definido como tendo origem na fonte de raio-X e término no elemento (ou *bin*) do detector. O número de feixes considerados corresponde, por isso, ao número de *bins* do detector;
2. Para cada R_i:
   (a) São calculadas as coordenadas dos pontos onde o feixe R_i intersecta a grelha tridimensional da FOV (figura 3.7);
   (b) São calculadas as distâncias euclidianas no espaço entre cada duas intersecções sucessivas. O número de distâncias resultante será igual ao número de intersecções menos um.
   (c) As distancias são normalizadas relativamente à distância total do feixe R_i na FOV (ou seja, cada distância é reduzida à porção, entre 0 e 1, da sua contribuição para a atenuação total daquele feixe);
   (d) Cada distância é atribuída ao vóxel da FOV correspondente (explicação mais à frente);
3. Estes valores vão preenchendo a MS da seguinte forma: cada linha da matriz corresponde ao feixe R_i e cada elemento (a_{ij}) corresponde à contribuição do vóxel j para a atenuação do feixe R_i.

A figura 3.7 representa de forma simplificada a interacção de dois feixes de radiação (R_1 e R_2) com a FOV. Os vóxeis destacados a verde e azul são aqueles atravessados por R_1 e R_2, respectivamente.

A figura 3.8 representa uma simplificação a duas dimensões deste procedimento.

O segmento $\bar{AB}$ representa um determinado feixe de radiação R_i. Os pontos de X_0 a X_5 e de Y_0 a Y_3 representam as 10 interacções entre R_i e a grelha da FOV. "Seguindo"o caminho do feixe do ponto $A(A_x, A_y)$ para o ponto $B(B_x, B_y)$, as intersecções são calculadas sempre que uma das coordenadas toma valores múltiplos do *binsize* (dimensão

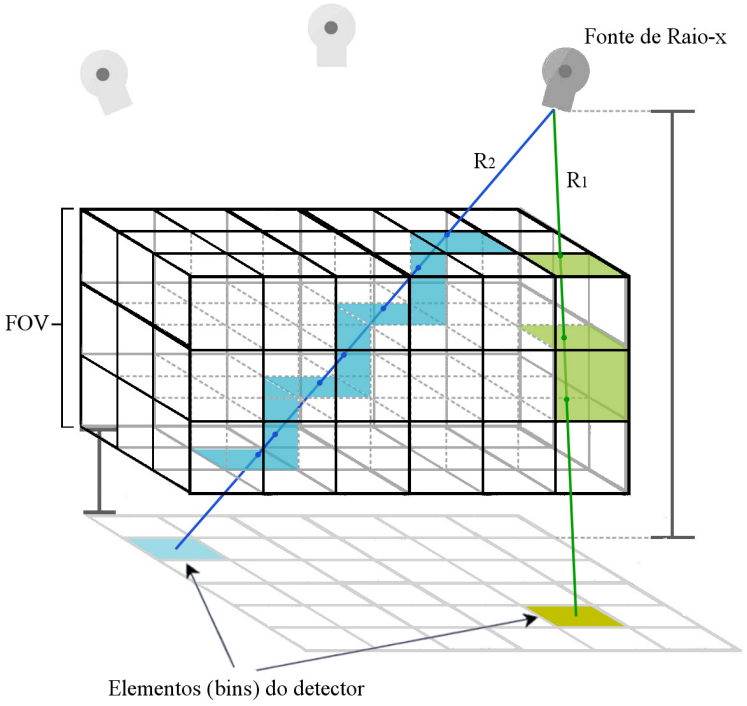

FIGURA 3.7: Representação esquemática simplificada da interacção entre dois feixes de radiação (R_1 e R_2) com a FOV. Os quadrados coloridos de azul e verde representam os vóxeis atravessados por cada um dos feixes correspondentes.

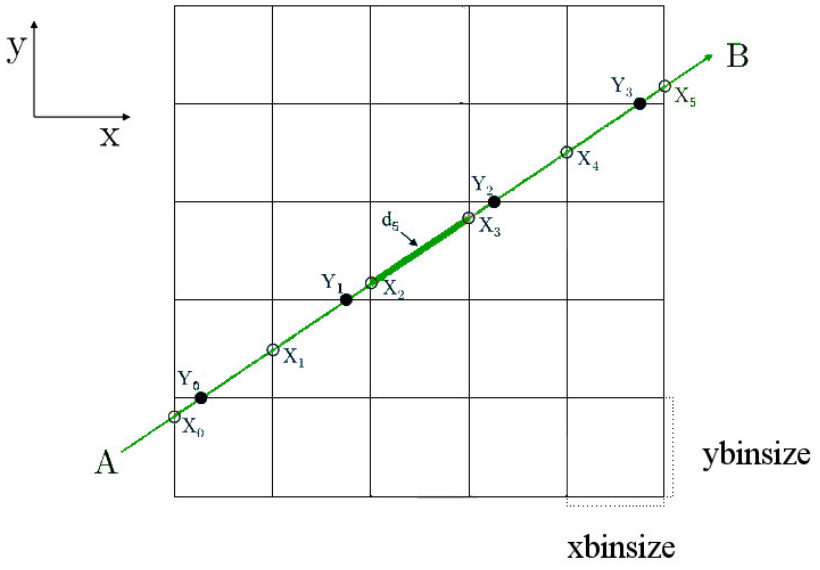

FIGURA 3.8: Esquema a duas dimensões das intersecções entre um feixe de radiação ($\bar{AB}$) e a FOV. A distância d_5 representa a distancia euclidiana entre duas intersecções consecutivas.

do bin). É calculado depois a distância entre cada duas intersecções consecutivas. Na figura encontra-se representada d_5, a 5ª distância entre intersecções do feixe R_i, que corresponde à distância euclidiana entre X_2 e X_3. Depois de normalizadas relativamente ao segmento $\bar{AB}$, cada uma das distâncias representa o quanto aquele vóxel contribuí para a atenuação do feixe R_i considerado.

Considerando agora a generalização para três dimensões, temos que o feixe R_i, que corresponde ao troço entre o bin i (A_x, A_y, A_z) e a fonte de raio-X (B_x, B_y, B_z), pode ser representado da seguinte forma:

$$\begin{aligned} X(\alpha) &= A_x + \alpha(B_x - A_x) \\ Y(\alpha) &= A_y + \alpha(B_y - A_y) \ , \\ Z(\alpha) &= A_z + \alpha(B_z - A_z) \end{aligned} \tag{3.13}$$

onde X, Y e Z correspondem às coordenadas do raio R_i para cada α, com $0 \leqslant \alpha \leqslant 1$, sendo que $\alpha = 0$ no ponto A e $\alpha = 1$ no ponto B. Como as coordenadas de A e B são valores conhecidos, é possível encontrar as coordenadas das intersecções. O feixe intersecta um eixo sempre que uma das suas coordenadas toma o valor múltiplo do *binsize* correspondente à direcção do eixo considerado. Para cada coordenada calculada, por exemplo $X(\alpha)$, as duas coordenadas restantes, $Y(\alpha)$ e $Z(\alpha)$, podem ser obtidas através da seguinte relação:

$$\begin{aligned} \alpha &= \frac{X(\alpha) - A_x}{B_x - A_x} \\ Y(\alpha) &= A_y + \alpha(B_y - A_y) \ \cdot \\ Z(\alpha) &= A_z + \alpha(B_z - A_z) \end{aligned} \tag{3.14}$$

Depois de as distâncias serem calculadas e normalizadas, são atribuídas aos vóxeis correspondentes. Para este efeito é considerado que:

- os feixes R_i deslocam-se sempre no sentido *bin*-fonte;

- uma determinada distância corresponde sempre ao caminho entre dois pontos do vóxel correspondente (um ponto de entrada e um ponto de saída).

A atribuição das distâncias aos vóxeis é realizada então considerando as intersecções de saída: as coordenadas do vóxel são obtidas a partir das coordenadas da intersecção de saída da distância correspondente. No entanto, o declive do feixe interfere com esta atribuição, e como o seu valor pode ser tanto positivo como negativo, as duas situações devem ser consideradas separadamente:

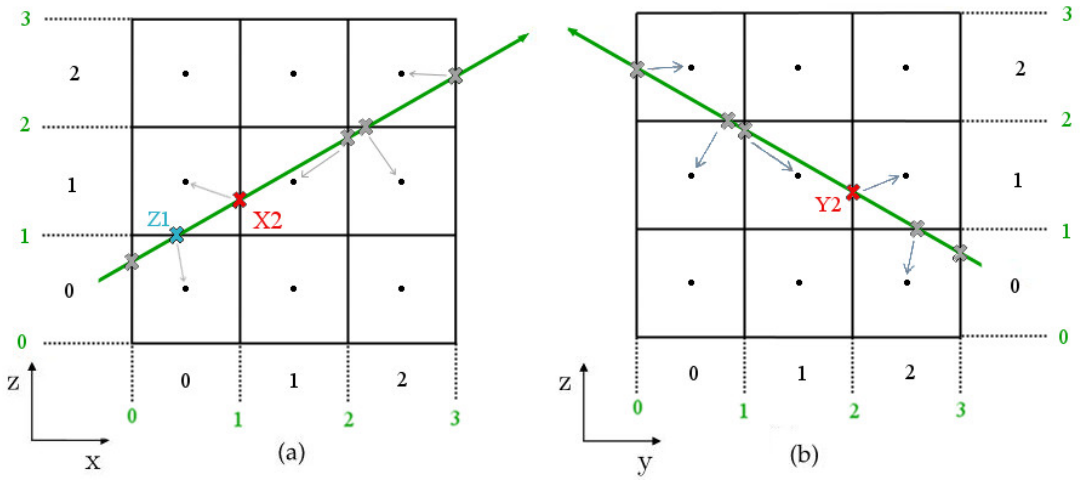

FIGURA 3.9: Esquema simplificado da atribuição das distâncias aos vóxeis correspondentes. As duas imagens representam as duas possibilidades: declive positivo (a) e negativo (b). Os números verdes representam os múltiplos do *binsize* referentes às coordenadas das intersecções, enquanto que as escalas pretas são referentes às coordenadas dos vóxeis.

- declive positivo (figura 3.9(a)):
  - a coordenada da intersecção de saída é múltipla do *binsize*: a coordenada do vóxel é obtida subtraindo 1 unidade à coordenada da intersecção de saída;
  - a coordenada da intersecção se saída não é múltipla do *binsize*: a coordenada do vóxel é obtida arredondando a coordenada da intersecção de saída;
- declive negativo (figura 3.9(b)) - arredondar todas as coordenadas.

Considere-se os exemplos representados da figura 3.9:

- na situação (a), onde o declive é positivo, encontram-se realçadas duas intersecções (uma com o eixo dos xx (X2) e outra com o eixo dos zz (Z1)). As suas coordenadas, representadas em termos dos múltiplos de *binsize*, são as seguintes: Z1(0.5;1) e X2(1;1.4). As coordenadas dos vóxeis são então as seguintes: Z1(0;0) e X2(0;1). No que diz respeito à intersecção Z1:
  - a coordenada x (0.5) é arredondada porque não é múltipla inteira de *xbinsize*, tomando assim o valor 0;
  - à coordenada z (1) é subtraída uma unidade porque é múltipla inteira do *zbinzise*, tomando assim o valor 0;

  No que diz respeito à intersecção X2:
  - à coordenada x (1) é subtraída uma unidade porque é múltipla inteira do *xbinsize*, tomando assim o valor 0;
  - a coordenada z (1.4) é arredondada porque não é múltipla inteira do *zbinsize*, tomando assim o valor de 1;
- na situação (b), onde o declive é negativo, encontra-se realçada uma intersecção com o eixo yy (Y2). As suas coordenadas, representadas em termos dos múltiplos de *binsize*, são as seguintes: Y2(2;1.3). As coordenadas dos vóxeis são então as seguintes: Y2(2;1). Ambas as coordenadas (2;1.3) são arredondadas, independentemente de serem múltiplas inteiras do *binzise*, tomando assim o valor (2;1).

Capítulo 4

Computação em GPU

Há mais de uma década que se vem registando um interesse crescente na utilização das unidades de processamento gráfico (GPU - *Graphical Processing Units*) em aplicações não-gráficas. Desde a sua introdução em contexto académico, por volta do ano 2000, aquando da publicação de alguns artigos científicos, até aos dias de hoje, onde é possível encontrar incontáveis aplicações comerciais e industriais, a computação em GPU tem evoluído e amadurecido a uma velocidade impressionante. Esta expansão tem sido acompanhada pelo surgimento de inúmeras linguagens e ferramentas de programação, que são também cada vez mais sofisticadas, tornando assim a computação em GPU cada vez mais acessível e de fácil aprendizagem. O facto de a introdução à programação em GPU estar hoje mais facilitada não significa no entanto que tirar total partido das capacidades do seu hardware seja uma tarefa simples, bem pelo contrário, é um processo complexo que requer treino e dedicação. Neste contexto, o presente capítulo pretende introduzir o leitor aos conceitos básicos da computação em GPU, mais concretamente à programação em CUDA.

4.1 Contextualização Histórica

Os microprocessadores baseados numa unidade central de processamento (CPU - *Central Processing Unit*) foram os responsáveis pelo grande aumento da performance dos computadores, e pela diminuição dos seus custos de produção, principalmente durante os últimos anos do século XX [69]. Actualmente são capazes de executar milhares de

milhões (10^9) de operações por segundo. Este avanço tem vindo a permitir às aplicações de software uma maior funcionalidade, uma interface com os utilizadores melhorada e uma melhor capacidade para obter resultados úteis.

As unidades CPU têm a sua génese da arquitectura de *von Neumann* e são por isso de origem sequencial, optimizados para executar um conjunto de operações numa determinada ordem. Um dos principais parâmetros de performance dos CPUs tem sido tradicionalmente a sua frequência de relógio: número de operações executadas por unidade de tempo. Ao duplicar a frequência de relógio, duplicar-se-ia a performance do processador. Esta regra dita a tendência de crescimento exponencial da frequência de relógio durante vários anos: o número de transístores que se consegue integrar num *chip* duplica a cada dois anos [70]. No entanto, em 2000 este crescimento sofre uma interrupção abrupta ao atingir a chamada "barreira de potência"(*power wall*) [71]: a potência (ou consumo de energia) de um CPU é proporcional à terceira potência da sua frequência de relógio [72], e começava a aproximar-se da potência equivalente à de uma célula de energia nuclear [73]. Sendo impossível refrigerar circuitos integrados com este tipo de potência, a frequência de relógio acabou por estabilizar em pouco menos de 4 GHz. Pouco tempo depois surgem também barreiras de memória e de ILP (*Instruction Level Parallelism*)[71], e a computação sequencial atinge o seu limiar de performance. Os fabricantes de CPUs iniciam então uma estratégia de aumentar a performance a partir da criação de múltiplos núcleos e implementação de intrusões paralelas.

Esta estagnação da performance dos CPUs acontece praticamente em simultâneo com o crescimento exponencial da performance dos GPUs, consequência da sua intensiva computação paralela. As GPUs são unidades de processamento inicialmente desenvolvidas com o objectivo de realizar a renderização gráfica de objectos tridimensionais numa matriz bidimensional (píxeis num ecrã). Estas unidades têm origem nos aceleradores gráficos 2D utilizados nas décadas de 80 e 90, que surgem como auxiliares de computação no visionamento e interacção dos recentemente lançados sistemas operativos gráficos [74]. Como o cálculo e actualização da cor de um determinado píxel é completamente independente dos restantes, a computação paralela é uma tarefa bastante intuitiva nas GPUs.

Esta mudança de paradigma no que diz respeito ao aperfeiçoamento da performance computacional dá-se então no sentido da incrementação das instruções paralelas em

múltiplos núcleos, em vez do aumento da frequência de relógio. As unidades GPU encontram-se muito mais adaptadas a estas condições do que as unidades CPU, e têm sido por isso bastante procuradas: actualmente o **"gap"** de performances entre as duas unidades é cerca de 7 vezes quando comparados os picos quer da largura de banda quer de *gigaflops* (1 *gigaflop* = 10^9 ***fl****oting-point* ***op****erations*).

Esta discrepância de performances deve-se essencialmente às diferenças no design das arquitecturas de ambas as unidades (figura 4.1). O *hardware* paralelo está tão presente na arquitectura CPU, sobretudo devido às sucessivas escolhas dos fabricantes em dedicarem mais transístores e espaço de *chip* a *hardware* de controlo, como por exemplo *branch prediction* e *out-of-order execution*. Por outro lado, a pressão realizada pela indústria de jogos de vídeo em processar gráficos com dimensões cada vez maiores, cada vez mais rápido, impulsionaram os fabricantes das unidades GPU a dedicarem mais espaço de *chip* e mais transístores ao cálculo paralelo intensivo [75].

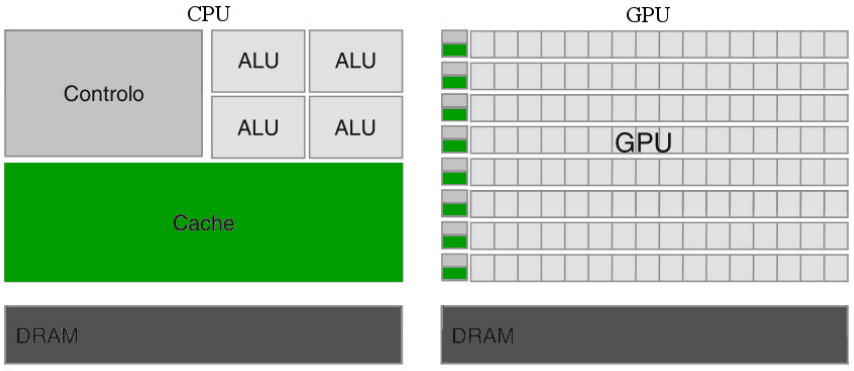

FIGURA 4.1: Esquema representativo das principais diferenças entre as arquitecturas das unidades CPU e GPU. **ALU** (*Arithmetic Logic Unit*) representa a unidade de aritmética e lógica. Adaptado de [69].

A maioria do software de âmbito geral é desenvolvido de raiz para ser executado de um modo sequencial, ou seja, num determinado momento no tempo, o processador apenas executa uma instrução, e a instrução seguinte só é executada quando a actual termina a sua execução. Esta execução é interpretada pelo programador seguindo os vários passos do código de modo sequencial. Estas aplicações estão limitadas às capacidades de processamento dos CPU's. Por outro lado, as aplicações que são programadas de raiz para correrem em múltiplos núcleos, vão continuar a tirar partido dos avanços de hardware. Estes programas correm várias tarefas em paralelo que trabalham em conjunto para atingirem um objectivo comum. Se uma determinada tarefa puder ser

dividida em várias sub-tarefas independentes, que possam ser processadas em paralelo (ao mesmo tempo), distribuídas pelos vários núcleos de um processador, o seu tempo de cálculo pode ser reduzido relativamente à abordagem sequencial [76].

4.2 Programação em GPU

Os GPUs do início do século XX foram concebidos basicamente para produzir uma cor para cada píxel do ecrã utilizando unidades aritméticas programáveis conhecidas como *pixel shaders*. De modo geral, cada *pixel shader* usa a sua posição (x,y) no ecrã, juntamente com informações adicional, para computar uma cor final e actualiza-la no ecrã em tempo real. A aritmética realizada é programável, e foram vários os investigados que se começaram a aperceber que poderiam processar nestas unidades dados mais gerais para além de apenas "cores"[74].

A possibilidade de programar unidades GPU para a computação de tarefas que não simplesmente a renderização gráfica despertou então o interesse de vários investigadores por volta do ano 2000. Numa primeira fase, quando os investigadores pioneiros dão os primeiros passos nesta área, a computação em GPU era realizada indirectamente: as operações tinham que ser mapeadas primeiro para um ambiente gráfico e depois manipuladas através de aplicações (APIs - *Application Programming Interface*) como OpenGL [77] ou DirectX [78]. Esta abordagem passou a ser conhecida como GPGPU (*General-Purpose Computing on GPU*). Apesar da metodologia pouco intuitiva e dos vários problemas de compilação, estas aplicações permitiram perceber o elevado potencial da computação GPU na resolução de problemas de cálculo aritmético mais geral [79].

Esta metodologia foi no entanto ficando obsoleta e não conseguia satisfazer as necessidades de problemas mais complexos e com maiores exigências, uma vez que o código de baixo nível era sempre limitado pelas APIs. Este facto tornou a utilização da GPGPU muito limitada a programadores com experiência em renderização gráfica.

Este panorama não é alterado até 2007, com o lançamento da linguagem de programação da NVIDIA dedicada exclusivamente à computação não-gráfica das GPUs: CUDA (*Compute Unified Device Architecture*).

4.3 CUDA

CUDA é um modelo de programação que permite a programação das GPU a partir de um paradigma semelhante à programação em C. Para além da melhoria da interface em termos de *software*, a NVIDIA realizou também algumas alterações à arquitectura das GPUs, adicionando *hardware* facilitador da programação em paralelo [69]. A programação em CUDA não necessita da interface gráfica das aplicações anteriores, em vez disso, é criada toda uma nova interface de programação em paralelo aplicada à computação de âmbito-geral.

O modelo de computação CUDA é considerado heterogéneo: utiliza tanto a CPU como a GPU tirando partido das capacidades de ambos. O objectivo dos programas assim desenvolvidos é sempre paralelizar a maior quantidade de tarefas possível, com o objectivo de minimizar o tempo de execução [80]. A optimização máxima conseguida a partir deste modelo é dada pela lei de Amdahl:

$$\begin{aligned} speedup &= \frac{1}{r_s + \frac{r_p}{n}}, \\ \lim_{n\to\infty} speedup &= \frac{1}{1 - r_p} \end{aligned} \tag{4.1}$$

onde n representa o número de processadores que executam a mesma porção do código, r_s e r_p representam as porções de código sequencial e paralelo, respectivamente, de tal modo que $r_s + r_p = 1$. Esta lei demonstra que quanto maior a porção de código paralelizada, maior a taxa de optimização (*speedup*).

Neste modelo de programação o sistema consiste num *host*, que corresponde ao CPU tradicional e num ou mais *devices*, que são co-processadores dedicados ao processamento paralelo intensivo de dados. Cada CUDA *device* suporta o modelo de execução SPMD (*Single-Program Multiple-Data*)[81] onde os múltiplos programas que são executados em simultâneo, processam o mesmo grupo de dados [82].

A sintaxe da linguagem CUDA é baseada fundamentalmente no standard ANSI C, estendido com algumas *keywords* que definem as funções paralelas, chamadas de *kernels*.

Estes *kernels* descrevem as tarefas executadas por um determinado *thread*, e são tipicamente invocados em milhares de *threads* simultaneamente. Os *threads* apresentam-se como a unidade básica de execução num determinado *device* e podem ser agrupados em *blocks*, que por sua vez podem ser agrupados em *grids*, dentro dos limites definidos pelo fabricante, formando assim uma hierarquia abstracta de execução paralela (figura 4.2). Os *threads* pertencentes ao mesmo *block* partilham memória e podem ainda sincronizar as suas acções a partir de funções integradas (*buit-in*).

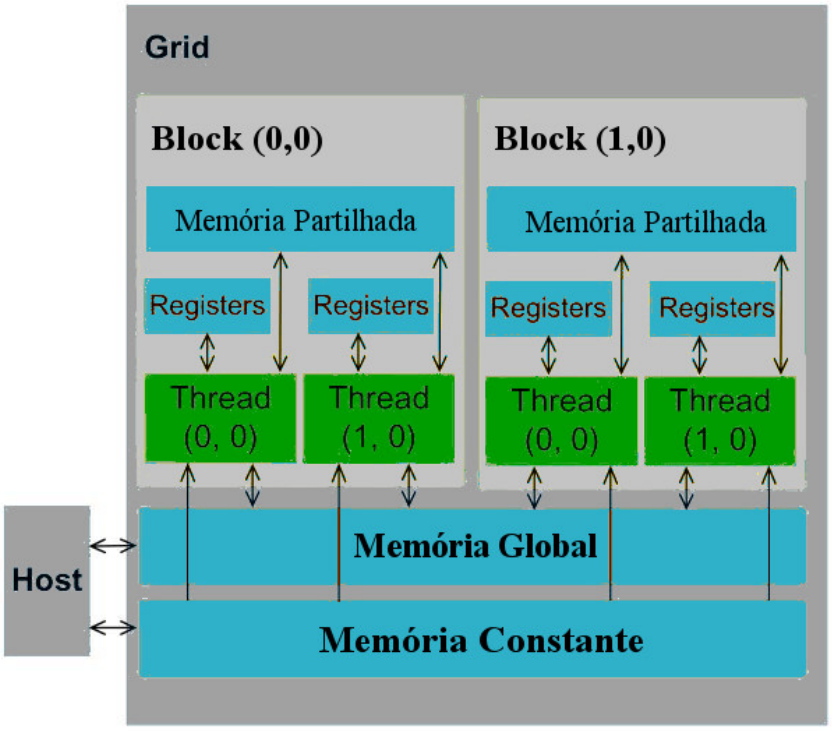

FIGURA 4.2: Esquema representativo da hierarquia de execução em CUDA. Adaptado de [69].

A estrutura representada na figura 4.2 reflecte a integração da execução entre o *host* e o *device*: o *host* invoca o *kernel*, que é executado nos múltiplos *threads* do *device*. O *kernel* é declarado usando a seguinte expressão:

```
__global__ void kernel_name(argument_declaration)
```

Para a invocação do *kernel* é necessário também a especificação da organização da hierarquia, nomeadamente o número de *blocks* (**`number_blocks`**) e o número que *threads* por *block* (**`number_threads`**):

kernel_name <<<number_blocks,number_threads>>>(arguments)

As variáveis **number_blocks** e **number_threads** representam então a dimensão da hierarquia de *threads* para uma determinada execução. Estas variáveis podem ser to tipo **int**, mas também podem ser do tipo **dim3**, um tipo de variável com 3 dimensões, no caso de se escolher construir uma hierarquia tridimensional. O número total de *threads* é então dado por: **number_blocks** × **number_threads**.

Cada *thread* que executa o *kernel* possui uma combinação única do seu índice dentro do *block* e do índice do *block* a que pertence dentro da *grid*. Estes índices são acessíveis dentro do próprio *kernel* a partir das variáveis *built-in* **threadIdx** e **blockId**, respectivamente. Ambas as variáveis são constituídas por três componentes cada, correspondentes às coordenadas no espaço:

- **threadIdx** - (**threadIdx.x**,**threadIdx.y**,**threadIdx.z**);
- **blockId** - (**blockId.x**,**blockId.y**,**blockId.z**).

Para além disso existem também outras variáveis *built-in* acessíveis dentro do *kernel* que representam o número total de *threads* num determinado *block* (**blockDim**) e o número de *blocks* no *grid* (**gridDim**). Tal como as variáveis anteriores, também estas são constituídas por três componentes.

O ambiente de programação CUDA possui funções específicas para a alocação (**cudaMalloc**) e libertação (**cudaFree**) de memória e ainda para a transferência de dados do *host* para o *device* e vice-versa (**cudaMemcpy**).

O Toolkit CUDA é uma ferramenta disponibilizada pela NVIDIA que permite a utilização de várias bibliotecas de funções optimizadas para GPU que implementam várias rotinas populares em computação de dados, como por exemplo processamento de imagem e sinal (NPP), transformada de Fourier (cuFFT), geração de números aleatórios (cuRAND), rotinas matemáticas (CUDA Math Library), entre outras [83].

Uma das bibliotecas com maior utilidade no cálculo da matriz de sistema é sem dúvida a biblioteca *Thrust*.

4.3.1 Biblioteca Thrust

Thrust [84] é uma biblioteca de funções compatíveis com CUDA, baseada na biblioteca de *templates* em C++ - STL (*Standard Template Library*) [85]. Rotinas como *scan*, *sort* ou *transform* são alguns dos exemplos das funcionalidades desta biblioteca.

De modo a utilizar as capacidades desta biblioteca é necessário primeiro entender alguns conceitos básicos de STL, como *containers* e *iterators*. Os *containers* são estruturas dinâmicas abstractas comparáveis a vectores, mas que podem armazenar elementos de vários tipos, enquanto que os *iterators* funcionam como ponteiros para os elementos dos *containers*. Em *Thrust* estão disponíveis dois tipos de *containers*: **`host_vector`** e **`device_vector`**, cada um deles contendo um par de *iterators* (**`begin()`** e **`end()`**), apontando respectivamente para o inicio e fim de cada *container*. Considere-se o seguinte exemplo, onde se cria um *container* H com 4 elementos inteiros:

```
thrust::device_vector<int> H(4);
```

O *iterator* **`H.begin()`** aponta para a posição do primeiro elemento de H (H[0]), enquanto que **`H.end()`** aponta para a posição logo após o último elemento de H (H[5]). Assim, este par de *iterators* define as posições entre as quais se encontra a informação relevante, permitindo assim uma forma intuitiva de aceder aos dados dos *containers*.

Dois conceitos também muito importantes na contextualização deste trabalho são a criação de *tuples* e o conceito de **`zip_iterator`**. A criação de *tuples* permite a aglomeração de múltiplas variáveis de modo a serem processadas em conjunto, como um todo. A rotina responsável por isto é a **`thrust::make_tuple`**. O **`zip_iterator`** permite a formação de novos *iterators* consoante as necessidades do utilizador. Funciona como um ponteiro personalizado, com enorme versatilidade, sendo possível criar *iterators* de múltiplas variáveis, que podem também elas ser de tipos diferentes.

Os algoritmos STL são então rotinas que implementam determinadas operações aos dados analisados. Os exemplos incluem:

- **`thrust::transform`** - aplica uma determinada operação a cada elemento de um conjunto de dados;
- **`thrust::reduce_by_key`** - é um exemplo de uma redução onde uma determinada operação é realizada, reduzindo um conjunto de dados recebidos e devolvendo

um único valor, consoante o operador especificado. Neste caso ("by_key") significa que essa redução vai ser selectiva, seguindo a lógica do vector *keys* considerado;

- **`thrust::copy_if`** e **`thrust::remove_if`** - são exemplos de rotinas que implementam reordenações. Os dados recebidos são reorganizados adicionando, removendo ou movendo elementos;

- **`thrust::sort_by_key`** é um exemplo de uma rotina utilizada para a ordenação de um determinado conjunto de dados, consoante a condição pretendida;

Estes algoritmos apresentam implementações quer para o *host*, quer para o *device*.

Finalmente, é importante também entender o conceito de operador (*operator*). Estas entidades funcionam como indicadores das operações que são possíveis realizar nos algoritmos. Existem alguns *operators* pré-definidos que correspondem às operações mais comuns: **`thrust::greater<T>`** que corresponde ao operador $>$ e **`thrust::plus<T>`** que corresponde ao operador $+$. Mas também existe a opção de criação de *operators* personalizados, que permitem realizar uma infinidade de operações à escolha do utilizador.

Thrust é uma biblioteca de *templates*, o que permite a declaração de rotinas que aceitem variáveis do tipo genérico T. Assim, as mesmas rotinas podem ser utilizadas para processar vários tipos de dados distintos.

Capítulo 5

Metodologia

Neste capítulo são abordadas as várias estratégias e procedimentos utilizados ao longo deste trabalho, desde a sua fase inicial de planeamento, até aos métodos mais concretos de programação e *debugging*, passando por uma breve contextualização do trabalho realizado anteriormente por outros autores. Na primeira secção, 5.1, são descritas as especificações técnicas do sistema DBT utilizado na aquisição das projecções. Depois, na secção 5.2, é apresentado, de forma sucinta, a implementação puramente sequencial (CPU) do processo de reconstrução, anteriormente desenvolvida. Posteriormente, na secção 5.3, é descrita a criação da implementação heterogénea (CPU+GPU), também desenvolvida anteriormente, que incluí o programa de cálculo da matriz de sistema (MS), escrito em CUDA. Na secção 5.4 são descritas as alterações efectuadas a esta implementação do cálculo da MS, que permitiram reduzir os artefactos observados nas imagens reconstruídas.

5.1 Sistema DBT

Os dados utilizados neste trabalho foram disponibilizados pelo departamento de Imagiologia do Hospital da Luz, obtidos num dispositivo Siemens MAMMOMAT Inspiration, na sua versão com dupla modalidade: tem a capacidade de realizar tanto exames de tomossíntese como de mamografia digital. As principais características do equipamento são idênticas à sua versão 2D anterior (que apenas permite a realização de mamografia

digital), incluindo a fonte de raio-X, o braço mecânico, o detector e electrónica associada [86].

A geometria do equipamento e o processo de aquisição encontram-se esquematizados na figura 5.1. A fonte de raio-X descreve um movimento contínuo de arco em volta da mama, com uma amplitude de 50º (de -25º até +25º). As 25 projecções são adquiridas com um intervalo de 2º, sem interrupção do movimento da fonte. O centro de rotação localiza-se a 4,7 cm da superfície do detector e a mama é comprimida entre este e a placa de compressão até aos 6 cm. O raio de rotação da fonte, ou seja, a distância da fonte de raio-X ao centro de rotação é constante e mede 62,5 cm. O detector é constituído por um painel de selénio amorfo com uma matriz de 2 816 × 3 584 elementos com 85 µm × 85 µm cada, que perfazem uma área total de aproximadamente 24 cm × 30 cm. O processo de aquisição tem a duração de 20 segundos [86].

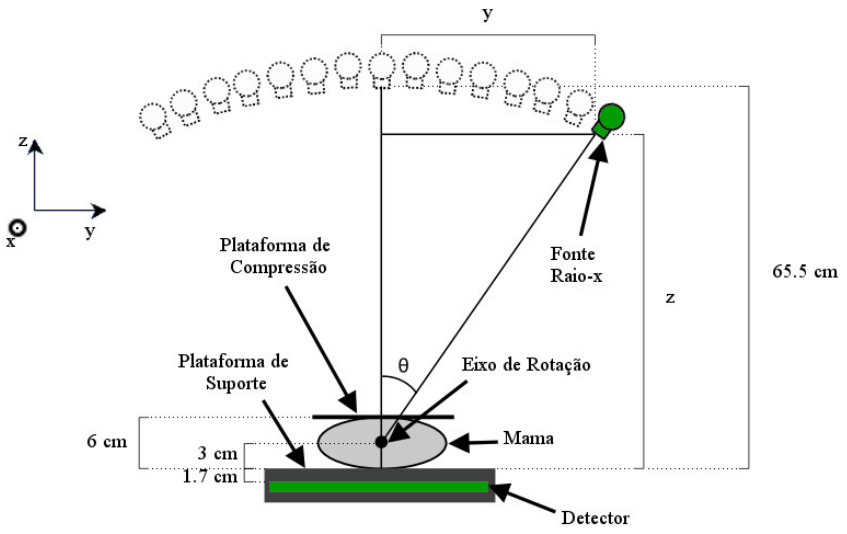

FIGURA 5.1: Representação esquemática da geometria do dispositivo e especificações técnicas do processo de aquisição. Adaptado de [87].

Em contexto clínico, as projecções são depois utilizadas para reconstruir o volume tridimensional utilizando o algoritmo analítico FBP. Este processo é realizado na estação de trabalho (*workstation*) acoplada ao equipamento, demora cerca de 60s e cria imagens paralelas com 1 mm de espessura, num ficheiro final com 2,1 GB de dimensão em memória [86].

No contexto da realização deste trabalho, foram utilizadas projecções de um exame realizado a uma mulher saudável, devidamente anonimizadas. Uma vez que durante a

aquisição desse exame foi utilizada uma compressão de 6 cm, e cada imagem reconstruída tem 1 mm de espessura, foi reconstruído um total de 60 imagens. Este valor corresponde então ao número de cortes reconstruídos na direcção perpendicular ao detector (N_{Slices}). Assim, as 3 dimensões da FOV são: as duas dimensões do detector, $Detector_x \times Detector_y$; e a dimensão "$z$", que corresponde ao número de cortes (N_{Slices}):

$$Detector_x \times Detector_y \times N_{Slices} = 2\,816 \times 3\,584 \times 60 = 605\,552\,640 \tag{5.1}$$

A FOV (volume irradiado pela fonte de raio-X que contribuí para a formação da imagem) é então formada por cerca de 600 milhões de elementos individuais (vóxeis), cada um dos quais com as seguintes dimensões: 85 µm × 85 µm × 1 mm.

5.2 Implementação Sequencial

A implementação puramente sequencial do processo de reconstrução foi realizado em IDL (*Interactive Data Language*), uma linguagem de programação interpretada orientada a vectores. É muito utilizada como ferramenta de processamento de grandes quantidades de dados, uma vez que permite a generalização de operações, que normalmente são aplicadas a escalares, a vectores ou matrizes de dados. Esta linguagem é muito popular na área da imagem médica uma vez que, para alem de apresentar relativa facilidade em processar grandes quantidades de dados, apresenta uma sintaxe acessível mesmo a utilizadores pouco experientes em ciências da computação [88].

Este tipo de linguagem apresenta no entanto algumas limitações:

- não permite ao programador conhecer os detalhes das estruturas internas que estão a ser executadas no computador, podendo este facto ser um obstáculo no que diz respeito à optimização temporal dos processos de computação;
- o programa não é compilado. Ao simplificar alguns dos aspectos mais problemáticos das linguagens de baixo nível (e.g. alocação de memória), perde velocidade de computação - as linguagens compiladas (como o Fortran, C ou CUDA) têm várias vantagens sobre as linguagens interpretadas, uma vez que o compilador traduz os

comandos para a linguagem nativa da máquina e consegue muitas vezes optimizar a ordem de execução e tornar o programa mais rápido.

Esta implementação foi desenvolvida no grupo de trabalho do IBEB e executa a reconstrução das imagens DBT a partir das projecções fornecidas.

O programa utiliza um algoritmo de reconstrução iterativo (ART, ML-EM ou OS-EM) e contém a informação das especificações técnicas do equipamento e da geometria da aquisição referidas na secção anterior. O código encontra-se escrito segundo o paradigma de programação orientada a objectos e apresenta uma divisão em vários procedimentos (*procedures*) e funções (*functions*), que funcionam como rotinas com permissões e tarefas distintas. O início do programa encarrega-se da criação de um objecto abstracto que simboliza todo o processo de reconstrução e cujos membros representam os vários componentes desse mesmo processo (e.g. o volume estimado, as projecções originais, as projecções estimadas, os ângulos correspondentes às várias posições da fonte de raio-X, o raio de rotação, etc). É possível ao utilizador, aquando da criação deste objecto, inicializar alguns destes membros, que são passadas ao programa como parâmetros, de entre as quais: qual o algoritmo que pretende utilizar (ART, ML-EM ou OS-EM); qual o factor de escala, se for o caso, que prefere (16, 8, 4, 2, 1[pré-definido]); se pretende ou não aplicar filtro de pós-processamento.

Depois da criação do objecto segue-se a execução das várias rotinas que realizam o processo de reconstrução. Independentemente de todas as variantes que o processo possa assumir, a sequencia de acções segue, de modo geral, o esquema de um algoritmo iterativo (figura 3.6). A figura 5.2 corresponde à adaptação desse esquema à execução em IDL:

1. Inicialização da estimativa do volume 3D - é construída uma matriz tridimensional com as dimensões da FOV e preenchida com zeros;
2. A cada iteração:
   (a) Realização das projecções do volume estimado (*byLORReprojection*) - para a criação de cada uma das projecções é necessário o cálculo da matriz de sistema (MS) correspondente (*MakeLorCalculation*);

(b) Comparação das projecções do volume estimado com as projecções originais - aqui são obtidas as projecções de erros (imagens que contêm as diferenças ponto-a-ponto entre os dois conjuntos de projecções);

(c) Realização da retroprojecção das projecções de erros - aqui é construído o mapa de erros (volume 3D que contém as diferenças entre a estimativa e os valores reais);

(d) Actualização da estimativa da iteração anterior com os dados do mapa de erros (*Update*);

3. No final de todas as iterações, a estimativa do volume 3D irá conter o resultado da reconstrução das imagens contidas nas projecções originais.

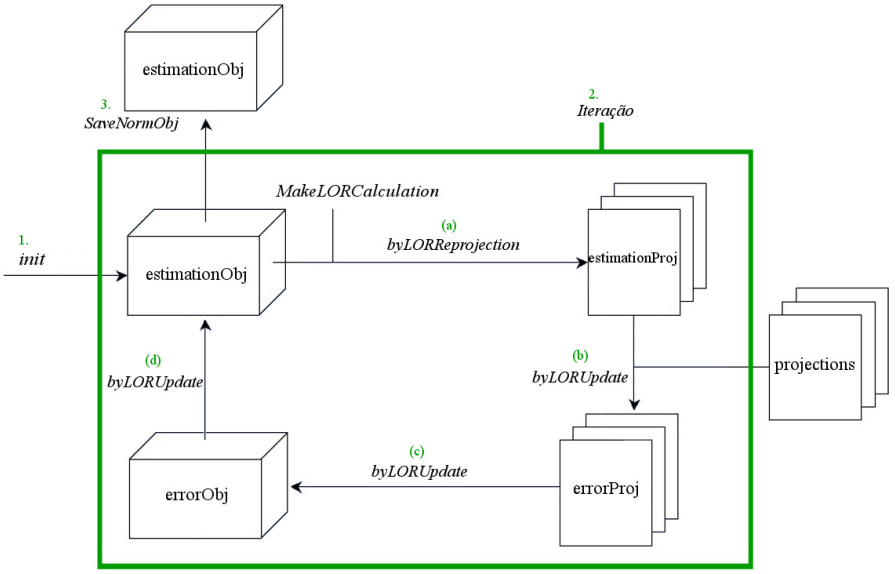

FIGURA 5.2: Diagrama esquemático das várias fases do processo de reconstrução de um algoritmo iterativo. Os vários números [1-3] e letras [(a)-(d)] apresentados encontram-se associados aos itens da lista anterior.

Esta sequência aplica-se a qualquer um dos três algoritmos (ART, ML-EM e OS-EM) que o utilizador possa escolher, logo existem procedimentos que são comuns a todos eles, como por exemplo *byLORReprojection*, *DownScale* ou *MakeLorCalculation*. No entanto, procedimentos como o *Update* são específicos para cada um deles: *byLORARTUpdate*, *byLORMLEMUpdate* e *byLOROSEMUpdate*.

Para a obtenção de cada uma das projecções em (a) é necessário o cálculo de uma nova MS. Isto acontece pois a cada projecção corresponde uma diferente posição da fonte de

raio-X e, consequentemente, uma diferente disposição dos feixes que atravessam a FOV e suas interacções com a matéria.

Ainda assim, considerando apenas uma determinada projecção, o cálculo da MS não é executado todo de uma vez, mas sim "LOR a LOR"(*byLOR*), ou seja, para uma determinada LOR:

- são calculadas as coordenadas das intersecções com a FOV e as distâncias correspondentes;
- as distâncias são normalizadas e atribuídas aos vóxeis correspondentes;
- é realizada a projecção da LOR na projecção em questão, tendo em conta a contribuição de cada vóxel atravessado;
- as coordenadas e as distâncias são apagadas.

Este processo é repetido para cada LOR daquela projecção. O cálculo da MS é realizado deste modo **"parcial"** devido a restrições de memória.

A memória necessária para guardar todos os elementos da MS de uma projecção apenas é igual ao número de estimativas dos coeficientes de atenuação, que é igual ao número de elementos da FOV, (605 552 640) a multiplicar pelo espaço que um valor *float* ocupa em memória (4 *byte*). Este valor (2 422 210 560) corresponde a cerca de 2.26 GB. Considerando as 25 projecções, o cálculo das matrizes de sistema iria ocupar mais de 56 GB de memória.

Nesta implementação, o cálculo da MS é assim executado para cada uma das LORs de cada vez, de um modo sequencial: o cálculo para a LOR seguinte só é iniciado quando o cálculo da LOR actual terminar. Estes vários cálculos são completamente independentes e podem ser paralelizados. Esta situação apresenta enorme potencial de optimização, para além de que o cálculo da MS representa a grande maioria do esforço computacional exigido durante todo o processo de reconstrução. Tendo em conta todo este panorama, optou-se por implementar uma alternativa mais adequada: o cálculo da MS em GPU.

Para além do mais, a linguagem IDL não se apresenta como uma solução muito viável no que diz respeito à implementação da reconstrução de imagens DBT, uma vez que não permite a optimização que apenas as linguagens de mais baixo nível permitem.

5.3 Implementação Heterogénea

A implementação heterogénea (CPU+GPU) realiza o processo de reconstrução das imagens DBT aproveitando os melhores atributos de ambas as arquitecturas. As tarefas são executadas na unidade que se apresenta mais adequada, sendo que as computacionalmente mais exigentes, como é o caso do cálculo da matriz de sistema (MS), são executadas na GPU. Esta integração foi desenvolvida por Ferreira [1] utilizando a linguagem de programação CUDA. A figura 5.3 representa esquematicamente o princípio desta integração.

O programa IDL invoca o programa CUDA quando necessita da MS. Sempre que este ponto é atingido, o programa IDL suspende a sua execução e o programa CUDA é executado no CPU também de forma sequencial, gerindo os recursos da GPU sempre que necessita realizar computação paralela. No final da execução do programa CUDA, a MS é devolvida ao programa IDL.

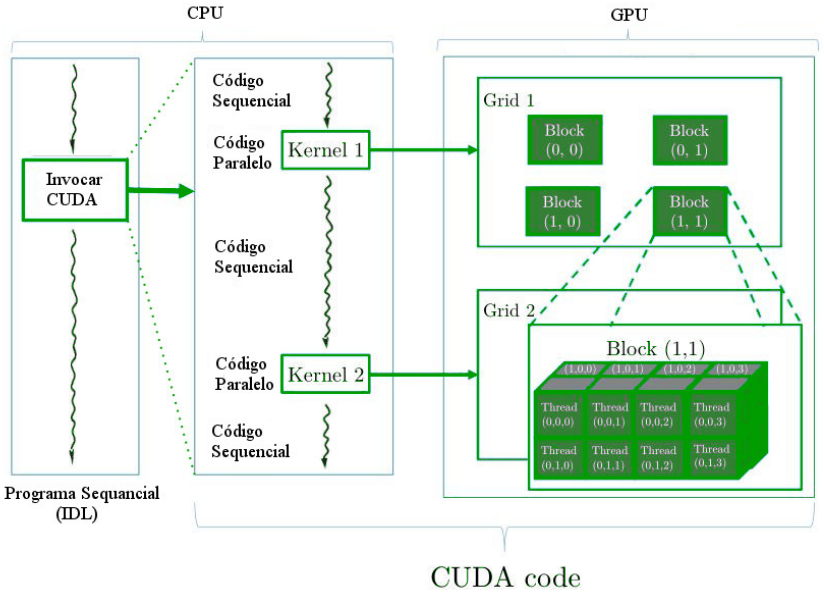

FIGURA 5.3: Representação da integração na implementação heterogénea. Adaptado de [1]

O programa foi escrito na linguagem de programação CUDA, recorrendo à utilização de várias funções e rotinas da biblioteca *Thrust*, entre as quais **`thrust::sort_by_key`**, **`thrust::make_tuple`** e **`thrust::remove_copy_if`**.

O programa CUDA, com os respectivos nomes utilizados nos cabeçalhos das funções, apresenta a seguinte estrutura:

1. Cálculo das intersecções;
   - Intersecções com o eixo xx (`__global__ sysmatx`);
   - Intersecções com o eixo yy (`__global__ sysmaty`);
   - Intersecções com o eixo zz (`__global__ sysmatz`);
2. Remoção das intersecções que ocorrem fora da FOV (**`compact`**);
3. Reagrupamento das intersecções (**`group`**);
4. Ordenação das intersecções (**`sort_x_bin`**);
5. (A) cálculo das distâncias e (B) atribuição das distâncias aos vóxeis da FOV (`__global__ distances`);
6. Normalização das distâncias (**`normalizacaosysmat`**);

O cálculo das intersecções (passo 1) é realizado de forma equivalente para os três eixos, recorrendo às equações 3.14. As especificações geométricas do sistema no que diz respeito à posição do *bin* do detector são as seguintes:

$$\begin{aligned} A_x &= (n_x + 0.5) \times xbinsize \\ A_y &= (n_y + 0.5) \times ybinsize \ , \\ A_z &= -17\text{mm} \end{aligned} \tag{5.2}$$

onde $xbinsize = ybinsize = 85\,\mu\text{m}$ e $n_x, n_y \in N$ tal que $0 \leqslant n_x < Detector_x$ e $0 \leqslant n_y < Detector_y$; no que diz respeito à posição da fonte de raio-X:

$$\begin{aligned} B_x &= \frac{Detector_x}{2} \times xbinsize \\ &= \frac{2816}{2} \times 0,085\text{mm} = 119,68\text{mm} \\ B_y &= d_1 \times \sin\theta + \frac{Detector_y}{2} \times ybinsize \\ &= 625\text{mm} \times \sin\theta + \frac{3584}{2} \times 0,085\text{mm} = \\ &= 625\text{mm} \times \sin\theta + 152,32\text{mm} \\ B_z &= d_1 \times \cos\theta + d_2 \\ &= 625\text{mm} \times \cos\theta + 30\text{mm} \end{aligned} \tag{5.3}$$

onde d_1 representa a distância entre a fonte e o eixo de rotação, d_2 a distância entre a plataforma de suporte e o eixo de rotação (figura 5.1). B_x é sempre constante ao longo de todo o movimento da fonte em torno da mama. Isto significa que a MS é simétrica relativamente ao eixo dos xx, bastando apenas realizar o cálculo de uma das metades.

As intersecções são calculadas de modo paralelo na GPU (`__global__`): cada *thread* é responsável pelo cálculo das intersecções de uma determinada LOR (*bin*). Cada intersecção necessita de quatro valores de modo a ser identificada: três valores *float* correspondentes às coordenadas (x, y e z) que definem a sua posição no espaço; e um valor *int*, que identifica o *bin* do detector correspondente à LOR a que a intersecção pertence. À medida que as intersecções são calculadas, quatro *arrays* vão sendo preenchidos - um para cada uma das coordenadas das intersecções (x, y e z) e um quarto para a identificação do *bin* correspondente. No final do cálculo de todas as intersecções são criados os seguintes *arrays*:

- eixo xx - intersectionsXx, intersectionsXy, intersectionsXz, binX;
- eixo yy - intersectionsYx, intersectionsYy, intersectionsYz, binY;
- eixo zz - intersectionsZx, intersectionsZy, intersectionsZz, binZ;

No passo 2 são removidas as intersecções que se encontram fora dos limites da FOV. Estes limites são:

$$\begin{aligned} 0 < & X < \frac{Detector_x}{2} \\ 0 < & Y < Detector_y \\ 0 < & Z < Nslices \end{aligned} , \tag{5.4}$$

onde X, Y e Z representam as coordenadas das intersecções. Basta que uma coordenada não satisfaça estas condições para que a intersecção seja removida. Esta remoção é realizada recorrendo ao *template* **`thrust::remove_copy_if`**, que remove as intersecções dos *arrays* se estas não satisfizerem as condições definidas em 5.4.

Seguidamente (passo 3) as intersecções são reagrupadas. Esta reorganização consiste na colocação de todas as intersecções (referentes aos três eixos) em quatro novos *arrays*: três correspondentes às coordenadas X, Y e Z e um correspondente ao *bin*. A função responsável por esta cópia é a **`thrust::copy`**.

O passo seguinte (4) consiste na ordenação das intersecções. É necessário garantir que as intersecções se encontrem por ordem, para que o cálculo das distâncias possa ser executado correctamente (entre intersecções consecutivas). A reordenando das intersecções é realizada primeiro por ordem crescente da coordenada X e depois por ordem crescente de bin. Isto é conseguido a partir da formação de um *zip iterator*[X,Y,Z,*bin*] que irá permitir manter as coordenadas e o *bin* de cada intersecção juntos durante as ordenações, evitando perda da informação. As rotinas responsáveis pela criação dos *zip iterators* e pela ordenação propriamente dita são o **`thrust::zip_iterator`** e o **`thrust::sort_by_key`**, respectivamente.

Depois (passo 5(A)) são calculadas as distâncias entre cada duas intersecções consecutivas. Aqui é novamente recrutada a GPU (**`__global__`**): cada *thread* é responsável por calcular a distância entre a intersecção a que corresponde e a seguinte: o *thread* (n) é responsável por calcular a distância entre a intersecção (n) e (n+1). Logo de seguida (passo 5(B)), ainda na GPU, dentro do mesmo *kernel*, cada distância é atribuída ao vóxel da FOV correspondente. Esta atribuição é realizada de acordo com a descrição da secção 3.2.4.

Por fim, no passo 6, é realizada a normalização das distâncias. Esta normalização é feita relativamente à distância total do trajecto que o feixe considerado (correspondente ao *bin*) descreve ao atravessar a FOV. Basicamente é realizada uma divisão de cada distância pela soma de todas as distâncias do *bin* correspondente. Esta operação é de elevada complexidade, uma vez que as distancias de todos os *bins* se encontram guardadas na mesma variável, no entanto cada uma destas distâncias deve apenas ser normalizada relativamente ao *bin* a que pertence. Esta complexa tarefa é realizada recorrendo às rotinas **`thrust::inclusive_scan`**, **`thrust::reduce_by_key`** e **`thrust::transform`**.

De todas estas funções, apenas os que apresentam a declaração (**`__global__`**) são executadas na GPU, como *kernels*. As restantes são executadas na CPU como funções normais.

Devido à limitação de memória na GPU, o cálculo da MS não foi executado para todos os *bins* do detector de uma só vez, mas sim para um determinado bloco de *bins*: o detector (2 816 × 3 584 *bins*) foi dividido em vários blocos de 88 × 56 *bins* (figura 5.4).

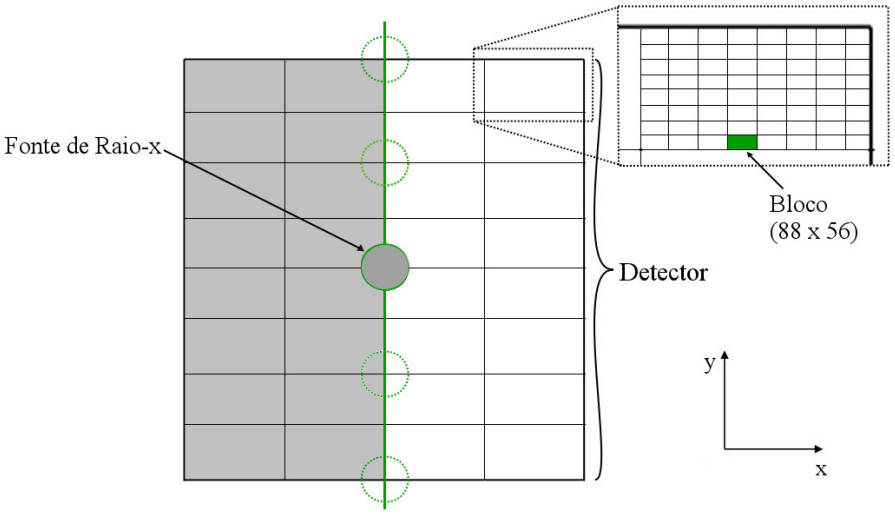

FIGURA 5.4: Representação da divisão do detector em blocos de *bins*.

O programa CUDA está assim concebido para receber do programa IDL um bloco de *bins*, e devolver a MS para esses *bins*, dada uma determinada posição da fonte. Esta informação é trocada na forma de oito variáveis: x(int), y(int), *angle*(float), *radius*(float), *matX*(int), *matY*(int), *matZ*(int), *matdist*(float). As quatro primeiras são passadas do IDL para o CUDA: x e y são índices que representam as coordenadas do bloco de *bins*

que se pretende processar, relativamente à sua posição no detector; *angle* e *radius* dão informação sobre a posição da fonte de raio-X. As quatro últimas são preenchidas durante a execução do programa CUDA e são passadas para o IDL. Contêm a informação da MS: *matX*, *matY* e *matZ* correspondem às coordenadas dos vóxeis e *matdist* às distâncias normalizadas.

Esta informação é enviada "em bruto", ou seja, as coordenadas dos vóxeis e as distâncias de todos os *bins* do bloco analisado são todas agrupadas nas mesmas variáveis (*matX*, *matY*, *matZ* e *matdist*). O IDL consegue processar a informação e associar as coordenadas e distâncias aos *bins* correspondentes uma vez que estas se encontram ordenadas por ordem crescente de bin, e dentro de cada bin por ordem crescente de coordenada Z. Assim, a mudança de *bin* ocorre quando dois elementos consecutivos da variável da coordenada Z (*matZ*) decrescem.

Nesta implementação, a integração entre o programa em CUDA e IDL foi feita a partir do procedimento LINKIMAGE do IDL. Esta ferramenta permite ao IDL interagir com programas externos escritos noutra linguagem, e executar estes programas externos como se fossem rotinas incorporadas (*built in*). Na integração entre CUDA e IDL, os principais aspectos a reter sobre do LINKIMAGE são os seguintes:

- Permite a passagem de variáveis do programa CUDA para IDL e vice-versa. A variável no IDL corresponde a uma estrutura em CUDA (ou C) chamada IDL_VARIABLE;
- O acesso à informação contida numa variável IDL, cujo conteúdo depende da forma como essa variável foi criada, deve ser feito, em CUDA, através de um apontador chamado IDL_VPTR;
- O programador deve ter em consideração que o tipo da variável pode ser alterado aquando da sua transferência de CUDA para IDL (e.g. um *int* em IDL é um *short int* em CUDA);
- O programa compilado em CUDA necessita ser executado pelo LINKIMAGE antes da execução de qualquer rotina na sessão actual de IDL que dele dependa. Quando essa sessão é fechada, o LINKIMAGE tem que ser novamente executado.

Uma das características mais importantes desta abordagem é a integração gradual do processamento paralelo num programa sequencial já implementado. Não só permite a construção de blocos de código individuais e a sua integração no algoritmo sequencial à medida que se vão tornando disponíveis, mas também a comparação dos resultados entre as duas implementações, quer ao nível da qualidade das imagens, quer ao nível do tempo de reconstrução.

5.4 Correcção da Implementação Heterogénea

A implementação heterogénea apresentada na secção anterior produz imagens ligeiramente diferentes das que são obtidas pela implementação puramente sequencial. Estas diferenças manifestam-se quando são comparadas duas imagens (uma reconstruída pela implementação heterogénea e outra pela implementação puramente sequencial) correspondentes ao mesmo corte do volume reconstruído.

A imagem apresentada na figura 5.5 corresponde à subtracção ponto-a-ponto entre dois cortes (z=27) obtidos na implementação heterogénea e puramente sequencial. Na imagem (a) foi utilizado o algoritmo SART, e na imagem (b), o algoritmo ML-EM.

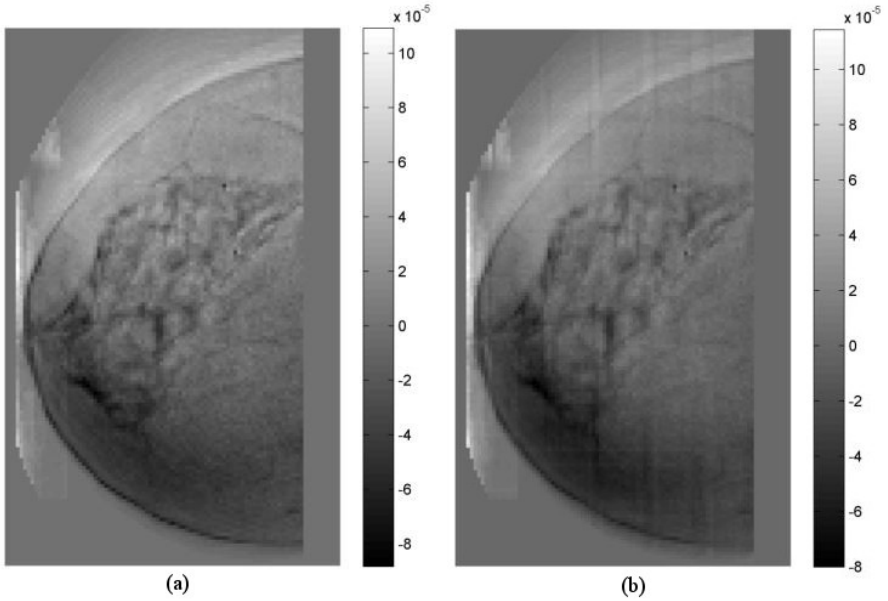

FIGURA 5.5: Diferença entre as imagens reconstruídas pela implementação heterogénea e puramente sequencial, correspondentes ao corte 27 (z=27), reconstruídas com o algoritmo SART **(a)** e ML-EM **(b)** [1].

Estes resultados não só sugerem a presença de uma irregularidade na implementação do cálculo da matriz de sistema (MS) em GPU, mas também uma possível propagação desse erro.

Numa primeira fase pensou-se que talvez as diferenças nas imagens reconstruídas se devessem simplesmente ao facto de estarem a ser implementadas na GPU. De modo a descartar esta possibilidade foi desenvolvido um código em C++ que implementou precisamente as mesmas funções que o código CUDA, mas que foi executado no CPU. As imagens reconstruídas com esta implementação não só mantiveram as alterações observadas na implementação CUDA como também a intensidade dessas alterações aumentou. Conclui-se assim que o problema não residia no facto de o cálculo ser executado na GPU.

De modo a simplificar a apresentação dos resultados, nas próximas secções vão ser utilizadas as seguintes expressões:

- CUDA_I - refere-se à implementação heterogénea original, desenvolvida por Ferreira;
- CUDA_II - refere-se à implementação heterogénea corrigida, desenvolvida durante esta dissertação.

A partir da análise dos códigos IDL e CUDA_I, recorrendo a técnicas de *debug*, foram identificadas algumas diferenças nas matrizes de sistema calculadas por ambas as implementações:

- Discordância no número de interacções de alguns *bins* específicos;
- As coordenadas de algumas intersecções não coincidiam.

Foram realizadas 4 alterações principais ao código CUDA_I:

1. na função que calcula as intersecções com o eixo dos yy (**`sysmaty`**), foi alterada a condição do ciclo **`for`** que percorre as várias coordenadas Y múltiplas do *ybinsize*. A condição **`<=`** foi alterada para **`<`** (linha 115);
2. na função que reordena as intersecções (**`sort_x_bin`**), foi alterado o bloco de instruções que realizava a reordenação por ordem crescente da coordenada X para

passar a realizar por ordem crescente da coordenada Z. A reordenação por *bin* não foi alterada (linha 353);

3. na função que calcula as distâncias (**__global__ distances**), quando é realizada a atribuição das distâncias aos vóxeis, foi reduzido o limiar a partir do qual se considera que uma coordenada Z é múltipla do *binzise*. O valor anterior era 0.0001 e foi alterado para 0.00001 (linha 428);

As alterações realizadas são de seguida apresentadas em excertos de código CUDA, apresentando primeiro o código antes da alteração e em segundo o código depois da alteração. A alteração 4 é a excepção uma vez que não é uma alteração propriamente dita mas sim uma inclusão.

Listing 5.1: Alteração 1 - Antes

```
1 for(yint=yint; yint<=floorf(yfocus/(float)SCALE) && yint<=detectorYDim;
    yint++)
```

Listing 5.2: Alteração 1 - Depois

```
2 for(yint=yint; yint<floorf(yfocus/(float)SCALE) && yint<detectorYDim;yint
    ++)
```

Listing 5.3: Alteração 2 - Antes

```
//SORT BY X
```

```
4   typedef thrust::device_vector<float>::iterator          Iterator;
    typedef thrust::device_vector<int>::iterator            IteratorInt;
6   typedef thrust::tuple<Iterator, Iterator, IteratorInt> IteratorTuple;
    typedef thrust::zip_iterator<IteratorTuple>             ZipIterator;
8
    ZipIterator begin (thrust::make_tuple(y.begin(), z.begin(), binx.begin())
     );
10  ZipIterator end   (thrust::make_tuple(y.end(),   z.end(),   binx.end()));

12  thrust::sort_by_key(x.begin(), x.end(), begin);
```

Listing 5.4: Alteração 2 - Depois

```
//SORT BY Z
14  typedef thrust::device_vector<float>::iterator          Iterator;
    typedef thrust::device_vector<int>::iterator            IteratorInt;
16  typedef thrust::tuple<Iterator, Iterator, IteratorInt> IteratorTuple;
    typedef thrust::zip_iterator<IteratorTuple>             ZipIterator;
18
    ZipIterator begin (thrust::make_tuple(x.begin(), y.begin(), binx.begin())
     );
20  ZipIterator end   (thrust::make_tuple(x.end(),   y.end(),   binx.end()));

22  thrust::sort_by_key(z.begin(), z.end(), begin);
```

Listing 5.5: Alteração 3 - Antes

```
    if(abs(indez1-round(indez1)) < 0.0001)
```

Listing 5.6: Alteração 3 - Depois

```
24  if(abs(indez1-round(indez1)) < 0.00001)
```

Listing 5.7: Alteração 4 - Depois

```
template <typename T>
```

```
26 struct remove_repeated
   {
28   __host__ __device__ bool operator()(const thrust::tuple<T,unsigned long
       int,int>& t)
     {
30     return (thrust::get<0>(t) < 0.0001);// return 1 if distance is < 0.0001
     }
32 };

34 int delzeros(float*intxc,unsigned long int*intyc,int*bin,int dim){

36     (...) // CODE MISSING HERE

38 //-- TEST -- to remove the repeated intersections (distances smaler than
       0.0001)

40   thrust::device_vector<float> x_out_2(dim);
     thrust::device_vector<unsigned long int> y_out_2(dim);
42   thrust::device_vector<int> bin_out_2(dim);

44   ZipIterator output_begin_2 (thrust::make_tuple(x_out_2.begin(),
                                 y_out_2.begin(),
46                               bin_out_2.begin()
                                 )
48                 );
     ZipIterator output_end_2 (thrust::remove_copy_if(output_begin,
50                           output_end,
                             output_begin_2,
52                           remove_repeated<float>()));

54   size_t ArraySizet = output_end_2 - output_begin_2;

56   thrust::copy(x_out_2.begin(),x_out_2.begin()+(int)ArraySizet,dev_ptrx);
     thrust::copy(y_out_2.begin(),y_out_2.begin()+(int)ArraySizet,dev_ptry);
58   thrust::copy(bin_out_2.begin(),bin_out_2.begin()+(int)ArraySizet,dev_bin)
       ;
   }
```

Capítulo 6

Resultados e Discussão

O presente capítulo pretende apresentar os resultados obtidos com a realização deste trabalho realizando uma análise crítica através da exposição de alguns comentários. Encontra-se dividido em duas secções: a primeira (6.1) apresenta as imagens reconstruídas e realiza uma avaliação da redução dos artefactos; a segunda secção (6.2) apresenta os resultados relativos à optimização temporal do processo de reconstrução, e comenta a contribuição das alterações efectuadas.

6.1 Avaliação das Imagens Reconstruídas

A avaliação das correcções foi realizada a partir da comparação entre as imagens reconstruídas com a implementação sequencial (IDL) e cada uma das implementações heterogéneas (CUDA_I e CUDA_II).

As imagens comparadas correspondem à vista superior (plano xy) no corte z=27 dos volumes reconstruídos com o algoritmo SART. A figura 6.1 apresenta os resultados desta comparação.

As duas imagens da esquerda (a e d) são iguais e representam a reconstrução em IDL. As duas imagens da coluna central correspondem às reconstruções heterogéneas: a imagem superior (b) corresponde à reconstrução CUDA_I e a imagem inferior (e) à reconstrução CUDA_II. A diferença entre as imagens CUDA_I e IDL encontra-se representada na imagem superior à direita (c) e a diferença entre as imagens CUDA_II e IDL na imagem inferior à direita (f).

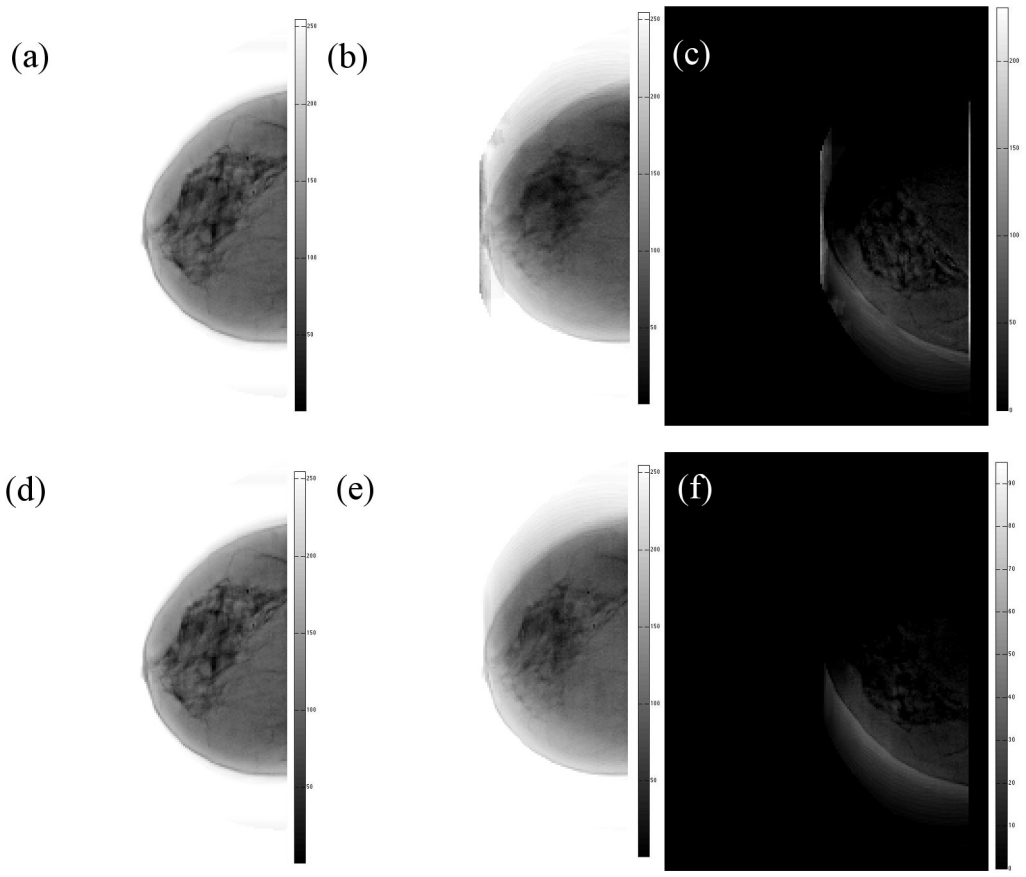

FIGURA 6.1: Comparação entre as imagens reconstruídas pela implementação sequencial (IDL) e cada uma das implementações heterogéneas: CUDA_I (em cima) e CUDA_II (em baixo).

Como se pode observar nas imagens das diferenças, nenhuma das imagens obtidas com as implementações heterogéneas (CUDA_I e CUDA_II) é igual à imagem IDL. No entanto observa-se uma ligeira melhoria de CUDA_I para CUDA_II. Este resultado é sugestivo do sucesso das correcções realizadas.

Os resultados apresentados nestas imagens são apenas referentes a um corte (z=27) e a um algoritmo (SART) mas são demonstrativos do que acontece em todo o volume reconstruído, para todos os algoritmos.

6.2 Avaliação do Tempo de Reconstrução

O cálculo da matriz de sistema (MS) foi também integrado noutros blocos do algoritmo com o objectivo de diminuir o tempo total de reconstrução. Os dados apresentados na tabela 6.1 mostram os resultados obtidos.

Tabela 6.1: Resultados da optimização temporal alcançada nos algoritmos ART e ML-EM, com factores de escala (FE) 16, 8 e 4. Os valores das colunas IDL e CUDA encontram-se expressos em segundos.

FE	**ART**			**ML-EM**		
	IDL	**CUDA**	**speedup**	**IDL**	**CUDA**	**speedup**
16	86,66	18,41	**4,71**	88,61	18,12	**4,89**
8	416,24	85,82	**4,85**	423,51	85,04	**4,98**
4	2392,43	483,32	**4,95**	2414,84	484,02	**5,00**

Estes valores foram registados para os algoritmos SART e ML-EM, com três factores de escala (FE) diferentes: 16, 8 e 4. Os tempos apresentados nas colunas IDL e CUDA encontram-se expressos em segundos e representam a média aritmética de três medições cada. Os valores de *speedup* são adimensionais e correspondem ao rácio $IDL/CUDA$. Os tempos correspondentes aos FE 2 e 1 não foram calculados devido a restrições de tempo.

Capítulo 7

Considerações Finais

O cancro representa hoje uma das principais causas de morte a nível mundial. A pouca informação que ainda se possui sobre os processos subjacentes à divisão descontrolada das células tumorais e aos fenómenos associados à malignidade e invasão de tecidos saudáveis, levam a que a maior parte do investimento e do esforço financeiro realizado pelos países ocidentais seja no sentido de melhorar as técnicas de diagnóstico e improvisar o tratamento e a qualidade de vida dos doentes.

O cancro da mama surge como um exemplo representativo desta situação. Estima-se que seja o segundo tipo de cancro mais comum em todo o mundo e o quinto mais mortífero, sendo que, se apenas considerarmos os dados referentes à população feminina, o panorama é bastante mais grave: não só apresenta a maior taxa de incidência, contabilizando um quarto de todos os cancros presentes nas mulheres em todo o mundo, como também a maior taxa de mortalidade, cerca de meio milhão de mortes anuais. É de facto uma das doenças mais prevalentes da nossa era, no entanto apresenta um prognóstico relativamente bom, tendo em conta outros tipos de tumores. Se detectado com antecedência, ou seja, numa fase mais precoce do seu desenvolvimento, o cancro da mama apresenta, na maioria das vezes, um diagnóstico benigno sem quaisquer complicações ou sintomas para a paciente. Esta detecção precoce só é no entanto possível recorrendo a técnicas de imagem médica.

Quando se fala em imagem médica no cancro da mama, está-se a falar de mamografia, pelo menos, por enquanto. A mamografia é de facto a técnica imagiológica universalmente aceite como a mais indicada para o rastreio e diagnóstico do cancro da mama,

sendo essencialmente eficaz na detecção de lesões numa fase inicial do desenvolvimento. Tem como vantagens o custo reduzido e a boa resolução espacial, permitindo uma boa detecção de estruturas de pequenas dimensões. Contudo apresenta também desvantagens associadas à sobreposição de estruturas na imagem devido ao facto de ser uma técnica bidimensional. Uma elevada taxa de falsos-negativos, mas sobretudo de falsos-positivos têm causado alguma polémica nos últimos anos e são várias as vozes que se levantam ainda hoje contra o uso da mamografia para o rastreio do cancro da mama. O problema que muitos autores levantam tem a ver com o sobrediagnóstico e com o rastreio desnecessário de um número demasiado elevado (centenas) de mulheres saudáveis, para que uma mulher doente possa ser diagnosticada. No entanto, independentemente das polémicas, a mamografia continua a ser nos dias de hoje a técnica de imagem referência para o rastreio e diagnóstico do cancro da mama.

O avanço das tecnologias de computação e processamento electrónico que vieram mais tarde dar origem à mamografia digital, tornaram também possível o desenvolvimento de outras técnicas ainda mais inovadoras. A DBT (*Digital Breast Tomosynthesis*) é um desses exemplos: consiste numa técnica de imagem médica que permite a realização de uma imagem tridimensional da mama. As imagens obtidas correspondem a cortes com diferentes profundidades no tecido mamário, deixando de existir qualquer interferência relacionada com a sobreposição de tecidos. O volume tridimensional formado pelos conjuntos destes cortes é conseguido a partir das projecções adquiridas à medida que a fonte de raio-X descreve um movimento de arco em torno da mama. As vantagens da DBT sobre a mamografia (maior sensibilidade, mas sobretudo maior especificidade) não apresentam evidência suficiente que possam compensar os gastos adicionais, a não ser talvez em mulheres mais jovens, com o tecido mamário mais denso. Estes gastos não estão só relacionados com o investimento financeiro, mas também com um investimento humano. O tempo que um clínico demora a analisar 60 imagens de DBT não é compatível com certas situações de maior urgência. Além disso, o processo de reconstrução implementado actualmente nos equipamentos hospitalares não é a melhor opção em termos de qualidade de imagem.

O processo de reconstrução das imagens em DBT é realizado com recurso a algoritmos computacionais. A sua principal divisão é feita segundo critérios concepcionais: os algoritmos analíticos representam um modelo matemático bastante simplificado e não entram em conta com uma quantidade de fenómenos físicos que acontecem durante o processo

de aquisição; por outro lado, os algoritmos iterativos são bastante mais robustos neste aspecto. Consideram o modelo da transmissão da radiação X pelos tecidos biológicos e implementam uma representação da geometria da aquisição muito mais próxima da realidade. Como era de esperar, todo este trabalho adicional tinha que ser pago de alguma forma: no caso dos algoritmos iterativos, o tempo de reconstrução é o parâmetro compensatório. Os algoritmos iterativos são bastante mais pesados do que os analíticos e consequentemente demoram muito mais tempo a reconstruir o volume de imagens. É então esta a principal razão da escolha dos algoritmos analíticos como mais adequados à reconstrução DBT em ambiente clínico.

A construção da matriz de sistema contribui de forma maioritária para a carga computacional dos algoritmos iterativos. O seu cálculo é fundamental no processo de reconstrução, uma vez que contém a informação detalhada da contribuição de cada elemento de volume (vóxel) para a atenuação dos feixes de raio-X. No entanto, como requer a resolução de centenas de milhares de operações aritméticas lineares, a sua computação nas unidades de processamento convencionais (CPU) torna-se obsoleta. Esta constatação é ainda mais importante quando se sabe que muitas destas equações lineares são completamente independentes entre si.

O potencial de paralelização do cálculo da matriz de sistema, nomeadamente os cálculos das intersecções e posteriormente, o cálculo e atribuição das distâncias aos vóxeis correspondentes, tornam este bloco de operações ideal para a computação em GPU.

As unidades de processamento gráfico (GPU) são hoje uma realidade completamente distinta do contexto de onde surgiram há cerca de 20 anos. Olhando para a sua evolução desde simples aceleradores gráficos de um ambiente 2D até às arquitecturas multifacetadas dedicadas quase exclusivamente à computação paralela intensiva, facilmente se entende o peso que hoje representam não só nas áreas das ciências da computação, mas um pouco por todos os campos da ciência e da indústria mundial. O surgimento de linguagens estritamente dedicadas à programação das GPUs para fins não gráficos, como é o caso da linguagem da NVIDIA (CUDA), permitiram uma expansão especialmente acentuada, e o campo das imagens médicas não foi excepção a este contágio.

A implementação do cálculo da matriz de sistema em CPU, a partir da criação de um programa heterogéneo (CPU+GPU), realizada por Ferreira no IBEB, demonstrou que é possível alcançar uma optimização significativa (1.6x) do tempo total de reconstrução.

A sua implementação apresenta no entanto alguns problemas de concepção: as imagens reconstruídas não são precisamente idênticas àquelas obtidas com a implementação puramente sequencial (CPU).

É neste contexto que se insere o trabalho realizado na presente dissertação, cujo objectivo inicial consistia na eliminação destes erros e a maior optimização do processo, dando continuidade ao trabalho iniciado anteriormente.

A temática da reconstrução iterativa de imagens associada à programação em GPU não foi de todo um tema fácil de pegar, no entanto senti-me motivado e com vontade de trabalhar. Os resultados obtidos não coincidiram com o esperado inicialmente, uma vez que não consegui eliminar por completo os erros responsáveis pelos artefactos das imagens, no entanto foram alcançadas algumas melhorias relativamente à implementação anterior: foi conseguido uma redução dos artefactos a partir da realização de quatro alterações ao código CUDA, e foi também conseguido uma maior optimização do tempo de reconstrução, aplicando o cálculo da matriz de sistema também à retroprojecção.

De modo geral, foram obtidos resultados razoáveis com este trabalho. Senti-me bastante motivado a estudar e a programar durante todo o período de duração do estágio e tenho a certeza que vou continuar a sentir interesse por estes temas, mesmo agora com o fechar deste ciclo.

Referências

[1] P. Ferreira, "Optimization of Breast Tomosynthesis Image Reconstruction using Parallel Computing," Tese de Mestrado, Universidade Nova de Lisboa, 2014.

[2] C. Pritsivelis e R. H. S. Machado, "Embriologia, anatomia e fisiologia da mama," em *Ginecologia Fundamental*, J. Conceição, Ed. Atheneu, 2006, cap. 4.

[3] A. Adam, A. K. Dixon, J. H. Gillard *et al.*, *Grainger & Allison's Diagnostic Radiology*. Churchill Livingstone, 2014.

[4] S. Mader e P. Galliart, *Understanding Human Anatomy and Physiology*. McGraw-Hill Higher Education, 2005.

[5] R. Seeley, T. Stephens, e P. Tate, *Anatomy and Physiology*. McGraw-Hill Higher Education, 2003.

[6] V. Harmer, *Breast Cancer Nursing Care and Management*. Blackwell Publishing, 2011.

[7] G. M. Cooper, *The Cancer Book: A Guide to Understanding the Causes, Prevention, and Treatment of Cancer*. Jones & Bartlett Learning, 1993.

[8] D. J. Winchester e D. P. Winchester, *Atlas of Clinical Oncology: Breast Cancer*, A. C. Society, Ed. B.C. Decker, 2000.

[9] K. I. Bland e E. M. Copeland III, *The Breast: Comprehensive Management of Benign and Malignant Diseases*. Elsevier Health Sciences, 2009, vol. 2.

[10] W. A. Berg, "Imaging the local extent of disease." em *Seminars in Breast Disease*, W. A. Berg, S. A. Feig, e M. D. Lagios, Eds. Saunders, 2001, vol. 4, pp. 153–173.

[11] S. Edge, D. Byrd, C. Compton *et al.*, "Cancer staging manual," *American Joint Committee on Cancer (AJCC). 7th ed. New York: Springer*, 2010.

[12] G. J. Whitman, D. G. Sheppard, M. J. Phelps *et al.*, "Breast cancer staging," *Seminars in roentgenology*, vol. 41, no. 2, pp. 91–104, 2006.

[13] J. Ferlay, I. Soerjomataram, M. Ervik *et al.*, "Globocan 2012 - cancer incidence and mortality worldwide," *International Agency for Research on Cancer*, 2014.

[14] L. Dossus e P. R. Benusiglio, "Lobular breast cancer: incidence and genetic and non-genetic risk factors." *BCR: Breast cancer research*, vol. 17, 2015.

[15] N. F. Gant e F. G. Cunningham, *Basic Gynecology and Obstetrics.* Appleton and Lange, 1993.

[16] D. Mitchell e D. Gordon, *Breast Health the Natural Way.* John Wiley & Sons, 2002.

[17] K. D. Paulsen e P. M. Meaney, *Alternative Breast Imaging: Four Model-based Approaches.* Springer, 2005.

[18] V. A. Loving, W. B. DeMartini, P. R. Eby *et al.*, "Targeted ultrasound in women younger than 30 years with focal breast signs or symptoms: outcomes analyses and management implications," *American Journal of Roentgenology*, vol. 195, no. 6, pp. 1472–1477, 2010.

[19] J. Robbins, D. Jeffries, M. Roubidoux *et al.*, "Accuracy of diagnostic mammography and breast ultrasound during pregnancy and lactation." *AJR. American journal of roentgenology*, vol. 196, no. 3, pp. 716–22, Mar. 2011.

[20] P. B. Gordon e S. L. Goldenberg, "Malignant breast masses detected only by ultrasound. a retrospective review," *Cancer*, vol. 76, no. 4, pp. 626–630, 1995.

[21] P. Ricci, E. Maggini, E. Mancuso *et al.*, "Clinical application of breast elastography: state of the art." *European journal of radiology*, vol. 83, no. 3, pp. 429–37, Mar. 2014.

[22] W. DeMartini e C. Lehman, "A review of current evidence-based clinical applications for breast magnetic resonance imaging," *Topics in magnetic resonance imaging*, vol. 19, no. 3, pp. 143–150, 2008.

[23] M. D. Pickles, P. Gibbs, M. Lowry *et al.*, "Diffusion changes precede size reduction in neoadjuvant treatment of breast cancer," *Magnetic resonance imaging*, vol. 24, no. 7, pp. 843–847, 2006.

[24] L. Martincich, F. Montemurro, G. De Rosa *et al.*, "Monitoring response to primary chemotherapy in breast cancer using dynamic contrast-enhanced magnetic resonance imaging," *Breast cancer research and treatment*, vol. 83, no. 1, pp. 67–76, 2004.

[25] J. Seely, E. Nguyen, e J. Jaffey, "Breast mri in the evaluation of locally recurrent or new breast cancer in the postoperative patient: correlation of morphology and enhancement features with the bi-rads category," *Acta radiologica*, vol. 48, no. 8, pp. 838–845, 2007.

[26] E. A. Morris e L. Liberman, *Breast MRI: Diagnostic and Intervention.* Springer Science, 2005.

[27] S. Weinstein e M. Rosen, "Breast mr imaging: current indications and advanced imaging techniques," *Radiologic Clinics of North America*, vol. 48, no. 5, pp. 1013–1042, 2010.

[28] F. Sardanelli, A. Fausto, G. Di Leo *et al.*, "In vivo proton mr spectroscopy of the breast using the total choline peak integral as a marker of malignancy," *American journal of roentgenology*, vol. 192, no. 6, pp. 1608–1617, 2009.

[29] C. Mountford, S. Ramadan, P. Stanwell *et al.*, "Proton mrs of the breast in the clinical setting," *NMR in Biomedicine*, vol. 22, no. 1, pp. 54–64, 2009.

[30] I. Khalkhali, J. Villanueva-Meyer, S. L. Edell *et al.*, "Diagnostic accuracy of 99mtc-sestamibi breast imaging: multicenter trial results," *Journal of Nuclear Medicine*, vol. 41, no. 12, pp. 1973–1979, 2000.

[31] L. Weir, D. Worsley, e V. Bernstein, "The value of fdg positron emission tomography in the management of patients with breast cancer," *The breast journal*, vol. 11, no. 3, pp. 204–209, 2005.

[32] R. R. Raylman, S. Majewski, M. F. Smith *et al.*, "The positron emission mammography/tomography breast imaging and biopsy system (pem/pet): design, construction and phantom-based measurements," *Physics in medicine and biology*, vol. 53, no. 3, p. 637, 2008.

[33] G. Gennaro, "Physics and radiation dose of digital breast tomosynthesis," em *Digital Breast Tomosynthesis: A Pratical Approach*, A. Tagliafico, N. Houssami, e M. Calabrese, Eds. Springer-Verlag GmbH, 2016, cap. 1, pp. 1–10.

[34] H. Vainio e F. Bianchini, *Breast cancer screening*, ser. IARC Handbooks of Cancer Prevention. Lyon: IARC Press, 2002, vol. 7.

[35] A. Salomon, "Beitrage zur pathologie und klinik der mammacarcinome," *Arch Klin Chir*, vol. 101, pp. 573–668, 1913.

[36] W. Vogel, "Die roentgendarstellung der mammatumoren," *Arch Klin Chir*, vol. 171, pp. 618–626, 1932.

[37] J. Gershon-Cohen e A. Strickler, "Roentgenologic examination of the normal breast: its evaluation in demonstrating early neoplastic changes," *Am J Roentgenol Radium Ther*, vol. 40, pp. 189–201, 1938.

[38] R. L. Egan, "Experience with mammography in a tumor institution: Evaluation of 1,000 studies 1," *Radiology*, vol. 75, no. 6, pp. 894–900, 1960.

[39] S. Shapiro, P. Strax, e L. Venet, "Periodic breast cancer screening in reducing mortality from breast cancer," *Jama*, vol. 215, no. 11, pp. 1777–1785, 1971.

[40] B. H. Lerner, ""to see today with the eyes of tomorrow": A history of screening mammography." *Canadian Bulletin of Medical History/Bulletin canadien d'histoire de la médecine*, vol. 20, no. 1, pp. 299–321, 2003.

[41] P. C. Gøtzsche e K. J. Jørgensen, "Screening for breast cancer with mammography," *Cochrane Database Syst Rev*, vol. 6, p. CD001877, 2013.

[42] L. L. Humphrey, "Breast cancer screening: Summary of the evidence for the u.s. preventive services task force," *Ann. Intern . Med*, vol. 137, p. 347–360, 2002.

[43] X. Wu, A. E. Deans, e H. Liu, "X-ray diagnostic techniques," em *Biomedical Photonics Handbook: Biomedical Diagnostics*, T. Vo-Dinh, Ed. CRC Press, 2014, cap. 26, pp. 655–689.

[44] J. D. Bronzino, *Biomedical engineering handbook*. CRC press, 1999, vol. 2.

[45] P. C. Stomper, D. B. Kopans, N. L. Sadowsky *et al.*, "Is mammography painful?: A multicenter patient survey," *Archives of internal medicine*, vol. 148, no. 3, pp. 521–524, 1988.

[46] A. Poulos, D. McLean, M. Rickard *et al.*, "Breast compression in mammography: how much is enough?" *Australasian radiology*, vol. 47, no. 2, pp. 121–126, 2003.

[47] R. H. Gold, L. W. Bassett, e B. E. Widoff, "Highlights from the history of mammography," *Radiographics*, vol. 10, no. 6, pp. 1111–1131, 1990.

[48] K. L. Bontrager e J. Lampignano, *Textbook of Radiographic Positioning and Related Anatomy*. MOSBY, 2009.

[49] C. J. Rose, "Statistical models of mammographic textures and appearance," Dissertação de Doutoramento, Faculty of Medical and Human Sciences of University of Manchester, 2005.

[50] C. J. D'Orsi, A. C. of Radiology, B.-R. Committee *et al.*, *ACR BI-RADS Atlas: Breast Imaging Reporting and Data System*. American Colege of Radiology, 2013.

[51] M. Chevalier, P. Morán, J. I. Ten *et al.*, "Patient dose in digital mammography," *Medical physics*, vol. 31, no. 9, pp. 2471–2479, 2004.

[52] B. Heddson, K. Rönnow, M. Olsson *et al.*, "Digital versus screen-film mammography: a retrospective comparison in a population-based screening program," *European journal of radiology*, vol. 64, no. 3, pp. 419–425, 2007.

[53] P. Skaane, K. Young, e A. Skjennald, "Population-based Mammography Screening : Comparison of Screen-Film and Full-Field Digital Mammography with Soft-Copy Reading — Oslo I Study 1," *Radiology*, vol. 229, pp. 877–884, Dez. 2003.

[54] J. M. Park, E. A. Franken, M. Garg *et al.*, "Breast Tomosynthesis : Present Considerations and Future Applications," *Radio Graphics*, vol. 27, pp. 231–241, 2007.

[55] F. J. Gilbert, L. Tucker, e K. C. Young, "Digital breast tomosynthesis (DBT): a review of the evidence for use as a screening tool," *Clinical Radiology*, vol. 71, no. 2, pp. 141–150, Fev. 2016.

[56] I. Reiser e S. Glick, *Tomosynthesis Imaging*. Taylor & Francis, 2014.

[57] J. A. Baker e J. Y. Lo, "Breast tomosynthesis: state-of-the-art and review of the literature." *Academic radiology*, vol. 18, no. 10, pp. 1298–310, Out. 2011.

[58] S. P. Poplack, T. D. Tosteson, C. A. Kogel *et al.*, "Digital breast tomosynthesis: initial experience in 98 women with abnormal digital screening mammography." *AJR. American journal of roentgenology*, vol. 189, no. 3, pp. 616–23, Set. 2007.

[59] Z. des Plantes B.G., "Eine neue methode zur diffenzierung in der rontgenographie (planigraphie) (in german)," *Acta Radiol.*, vol. 13, p. 182–92, 1932.

[60] J. T. D. III e D. J. Godfrey, "Digital x-ray tomosynthesis: current state of the art and clinical potential," *Physics in Medicine and Biology*, vol. 48, no. 19, p. R65, 2003.

[61] I. Hologic. (2011, Fev.) Summary of safety and effectiveness data (ssed) - digital breast tomosynthesis - selenia dimensions 3d system c-view software module. Food and Drug Administration. [Online] Disponível em: http://www.fda.gov/downloads/advisorycommittees/committeesmeetingmaterials/medicaldevices/medicaldevicesadvisorycommittee/radiologicaldevicespanel/ucm324866.pdf [Acedido a 20 Ago 2015].

[62] J. A. Baker, E. L. Rosen, J. Y. Lo *et al.*, "Computer-aided detection (cad) in screening mammography: sensitivity of commercial cad systems for detecting architectural distortion," *American Journal of Roentgenology*, vol. 181, no. 4, pp. 1083–1088, 2003.

[63] J. D. Enderle, S. M. Blanchard, e J. D. Bronzino, *Introduction to Biomedical Engineering*. Elsevier Academic Press, 2005.

[64] T. Gomi, "A Comparison of Reconstruction Algorithms Regarding Exposure Dose Reductions during Digital Breast Tomosynthesis," *J. Biomedical Science and Engineering*, vol. 7, pp. 516–525, Jun. 2014.

[65] R. Gordon, R. Bender, e G. T. Herman, "Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography," *Journal of Theoretical Biology*, vol. 29, no. 3, pp. 471–481, Dez. 1970.

[66] H. M. Hudson e R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data." *IEEE transactions on medical imaging*, vol. 13, no. 4, pp. 601–9, Jan. 1994.

[67] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional ct array," *Medical physics*, vol. 12, no. 2, pp. 252–255, 1985.

[68] F. Jacobs, E. Sundermann, B. D. Sutter *et al.*, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *Journal of Computing and Information Technology*, vol. 6, no. 1, pp. 89–94, Mar. 1998.

[69] D. B. Kirk e W. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2010.

[70] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[71] K. Asanovic, R. Bodik, B. Catanzaro *et al.*, "The landscape of parallel computing research: a view from berkeley, technical report," 12 2006.

[72] A. R. Brodtkorb, C. Dyken, T. R. Hagen *et al.*, "State-of-the-art in heterogeneous computing," vol. 1, no. 18, p. 1–33, 2010.

[73] G. Taylor, "Energy efficient circuit design and the future of power delivery," 10 2009.

[74] J. Sanders e E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison Wesley, 2010.

[75] J. Owens, "Streaming architectures and technology trends," em *Gpu Gems 2: Programming Techniques for High-performance Graphics and General-purpose Computation*, 1st ed., M. Pharr e R. Fernando, Eds. Addison-Wesley Professional, 2005, cap. 29.

[76] H. Sutter, "The Free Lunch Is Over A Fundamental Turn Toward Concurrency in Software," *Dr. Dobb's Journal*, 2005.

[77] R. J. Rost, B. Licea-Kane, D. Ginsburg *et al.*, *OpenGL shading language*. Pearson Education, 2009.

[78] Microsoft. Directx 11. [Online] Disponível em: https://www.microsoft.com/en-us/download/details.aspx?id=17431 [Acedido a 2 Mar 2016].

[79] M. Papadrakakis, G. Stavroulakis, e A. Karatarakis, "A new era in scientific computing: Domain decomposition methods in hybrid cpu–gpu architectures," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 13, pp. 1490–1508, 2011.

[80] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," *Proceedings of the April 18-20, 1967, spring joint computer conference on - AFIPS '67 (Spring)*, p. 483, 1967.

[81] M. J. Atallah e M. Blanton, *Algorithms and Theory of Computation Handbook, Volume 2: Special Topics and Techniques*. CRC press, 2009.

[82] S. Ryoo, C. I. Rodrigues, S. S. Baghsorkhi *et al.*, "Optimization principles and application performance evaluation of a multithreaded gpu using cuda," em *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*. ACM, 2008, pp. 73–82.

[83] Gpu-accelerated libraries, nvidia. [Online] Disponível em: https://developer.nvidia.com/gpu-accelerated-libraries [Acedido a 10 Jul 2016].

[84] Thrust library. [Online] Disponível em: https://developer.nvidia.com/thrust [Acedido a 10 Jul 2016].

[85] N. M. Josuttis, *The C++ Standard Library*. Addison Wesley, 2012.

[86] Siemens mammomat inspiration - data sheet. [Online] Disponível em: http://www.deltamedicalsystems.com/DeltaMedicalSystems/media/Product-Details/Tomo-Data-Sheet.pdf [Acedido a 15 Jun 2016].

[87] D. Schaa, B. Brown, B. Jang *et al.*, "Gpu acceleration of iterative digital breast tomosynthesis," em *GPU Computing Gems Emerald Edition*, ser. Applications of GPU Computing Series, W. H. Wen-Mei, Ed. Morgan Kaufmann, 2011, pp. 647–657.

[88] D. W. Fanning, *IDL programming techniques*. Fanning software consulting, 2000.