

Universidade Federal Fluminense

ANA BEATRIZ RODRIGUES DE ANDRADE GRAÇA

Desenvolvimento de Ferramenta Computacional para  
Criação de Domínios Discretos em 3D via Imagens 2D

VOLTA REDONDA

2019

ANA BEATRIZ RODRIGUES DE ANDRADE GRAÇA

# Desenvolvimento de Ferramenta Computacional para Criação de Domínios Discretos em 3D via Imagens 2D

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Métodos Matemáticos e Computacionais Aplicados à Engenharia e Ciência.

Orientador:

Wesley Luiz da Silva Assis

Coorientador:

Tiago Araújo Neves  
Yoisell Rodríguez Núñez  
Paulo Rangel Rios

UNIVERSIDADE FEDERAL FLUMINENSE

VOLTA REDONDA

2019

Ficha catalográfica automática - SDC/BEM  
Gerada com informações fornecidas pelo autor

G729d Graça, Ana Beatriz Rodrigues de Andrade  
Desenvolvimento de Ferramenta Computacional para Criação  
de Domínios Discretos em 3D via Imagens 2D / Ana Beatriz  
Rodrigues de Andrade Graça ; Wesley Luiz da Silva Assis,  
orientador ; Yoisell Rodríguez Núñez, coorientador. Volta  
Redonda, 2019.  
161 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense,  
Volta Redonda, 2019.

DOI: <http://dx.doi.org/10.22409/PPG-MCCT.2019.m.39414584854>

1. Computação gráfica. 2. Processamento de imagem. 3.  
Reconstrução tridimensional. 4. Seccionamento em série;  
técnica. 5. Produção intelectual. I. Assis, Wesley Luiz da  
Silva, orientador. II. Núñez, Yoisell Rodríguez,  
coorientador. III. Universidade Federal Fluminense. Escola de  
Engenharia Industrial e Metalúrgica de Volta Redonda. IV.  
Título.

CDD -

Desenvolvimento de Ferramenta Computacional para Criação de Domínios  
Discretos em 3D via Imagens 2D

Ana Beatriz Rodrigues de Andrade Graça

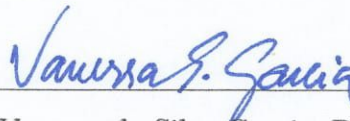
Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Métodos Matemáticos e Computacionais Aplicados à Engenharia e Ciência.

Aprovada por:



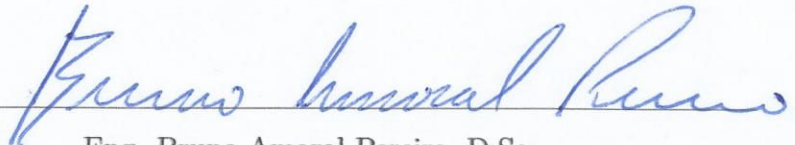
---

Prof. Wesley Luiz da Silva Assis, D.Sc. / MCCT-UFF  
(Presidente)



---

Prof<sup>a</sup>. Vanessa da Silva Garcia, D.Sc. / MCCT-UFF



---

Eng. Bruno Amaral Pereira, D.Sc.

Volta Redonda, 29 de Março de 2019.

*Aos diversos professores da UFF  
e aos que participaram da banca da minha dissertação,  
principalmente aos meus professores e orientadores:  
Wesley Assis, Tiago Neves, Yoissell Núñez e Paulo Rangel,  
que fizeram a grande diferença para a realização deste trabalho.  
A eles, a minha homenagem.*

# Agradecimentos

Os agradecimentos principais são direcionados a Deus, que iluminou o meu trajeto durante esse percurso. Deu-me força e coragem para seguir em frente e pisar fundo nas subidas mais íngrimes. Propôs um novo mundo de possibilidades. Ele permitiu que tudo isso acontecesse, não somente nesses dois anos de curso de mestrado, mas em todos os momentos de minha vida.

À minha família, por sua capacidade de acreditar e investir em mim. À minha mãe, Simone Rodrigues de Andrade, obrigada pelos seus cuidados, sua imensa dedicação e incentivo. Nas horas de desânimo e cansaço, agradeço pelos conselhos que me deram um gás para continuar trilhando por este caminho sinuoso. Ao meu pai, Ernani Graça Neto, apesar de todas as dificuldades e bloqueios diferenciais, sua presença significou segurança e certeza de que não estou sozinha nessa jornada. Aos dois, obrigada pelo carinho, a paciência e por sua capacidade de me trazerem paz na correria acelerada de cada semestre.

Ao meu namorado, Matheus Gorito de Paula, por me aguentar na correria do dia a dia e me dar forças, suporte e incentivos durante essa jornada. Fico grata pela amizade e companheirismo que nunca faltou entre nós.

Em especial, agradeço aos membros da banca: meu orientador, professor e doutor Wesley Luiz da Silva Assis que dedicou-me horas do seu tempo emprestando-me seus ouvidos com paciência, deu-me seu suporte no tempo que lhe coube, agradeço por contagiar a todos com sua alegria e bom humor. Fico grata pelas orientações e apoio. Agradeço também pela ajuda com questões de horários e pelo vínculo de amizade que criamos durante o curso de mestrado. À professora e doutora Vanessa da Silva Garcia que, além de professora, foi e está sendo minha conselheira com relação as questões do mestrado e experiências de vida. Ao membro externo da banca, engenheiro e doutor Bruno Amaral Pereira, por comparecer a UFF no dia de minha apresentação de defesa de dissertação e prestigiá-la.

Aos meus outros orientadores: professor e doutor Tiago Araújo Neves por sempre estar disposto a me ouvir e me aconselhar de forma sincera. Agradeço pelo convívio, amizade, apoio, suporte e pela gentileza de doar este template. Ao professor e doutor Yoissell Rodríguez Núñez por seu bom humor contagiante, incentivos a todo instante, vontade de ajudar os alunos com maior prazer, dedicação e laço de amizade de foi criado durante o curso de mestrado. Ao professor e doutor Paulo Rangel Rios pelas suas correções, dicas, incentivos e amizade. Não tenho palavras para expressar o quanto vocês fizeram a diferença para realização deste trabalho.

Agradeço a todos os professores que estiveram presentes na apresentação da minha dissertação de mestrado, por terem feito elogios aos quais fiquei emocionada. Agradeço também a presença de todos os amigos e colegas que prestigiaram a apresentação.

Aos professores do MCCT <sup>1</sup> que fizeram a diferença para meu desempenho: André Gusso, Cleyton Senior Stampa, Diomar Cesar Lobão, Gustavo Benitez Alvarez, Jorge Ferreira, Panters Rodríguez Bermúdez, Paulo Rangel Rios, Tiago Araújo Neves, Vanessa da Silva Garcia, Wellington Gomes Dantas, Wesley Luiz da Silva Assis e Yoissell Rodríguez Núñez, por ministrarem muito bem as aulas que estive presente, por estarem disponíveis para tirar dúvidas dos alunos, por horas de conversa, conselhos e dicas sobre qual área seguir e de como ser um profissional de qualidade.

Aos meus amigos e colegas do MCCT, Isamara Landim, Larissa Miguez e João Paulo Andrade da Silva pelo grupo incrível que formamos e todas discussões produtivas que tivemos. Ao André Luís Gonçalves Ferreira, Danilo Pinto Moreira de Souza, Gerardo Amado Pelén S., Juliano Vieira de Oliveira, Magna Amorim, Otávio Xavier Barbosa, Pedro Martins Menezes, Rafael Martins de Oliveira e Silva, Renan Franco Damasceno, Viviane Tavares Maia e a todos que criaram um laço de amizade nesse período do curso de mestrado. Nunca poderei me esquecer de agradecer o meu amigo Jonathas Luis Groetares Ferreira que dedicou horas de seu tempo tirando minhas dúvidas sobre os assuntos referentes a dissertação de mestrado. Aos meus amigos e colegas integrantes do grupo NMM <sup>2</sup> coordenado pelo professor Paulo Rangel Rios: Aline Cristine Lemos Duarte, André Luiz Moraes Alves, Felipe da Silva Siqueira, Gabriella Maria Silveira de Sá, Guilherme Dias da Fonseca, Harison da Silva Ventura, Mariana Sizenando Lyrio, Nathan Fernandes Ignácio e Talita Fonseca dos Prazeres. As amigas e colegas do programa de doutorado em Engenharia Metalúrgica: Daniele Aparecida Nogueira e Késsia Gomes

---

<sup>1</sup>Mestrado em Modelagem Computacional em Ciência e Tecnologia

<sup>2</sup>Núcleo de Modelamento Microestrutural

Paradela. Ao meu amigo do ICEX<sup>3</sup>, que agora está se formando na EEIMVR<sup>4</sup>, Douglas Oliveira por ter iniciado esta pesquisa em forma de iniciação científica. Sem ele, este trabalho não teria evoluído. Não menos importante, agradeço ao Diego Magalhães Baía pela cortesia de ter me passado as imagens originais de Aço Inoxidável Duplex contidas neste trabalho.

Quero agradecer também a minha família do coração: À Dona Teresinha Andrade que me acolheu na casa dela, onde morei por um ano, mas já frequentava seu restaurante desde 2012. Considerei e sempre vou considerar a família dela como se fosse minha: Seu Raimundo, Marcos Andrade e Vanuza Rodrigues. Além disso, sou grata pelo destino ter me colocado em uma casa tão boa onde tive a oportunidade de morar com 10 meninas que foram e serão como irmãs pra mim: Clara Maria Lima, Caroline Pires, Júlia Furtado, Denimara Dias, Letícia Carpenter, Sara Menezes, Heloisa Saletti, Larissa Lima, Tainara Albuquerque e Julinha Borges. Além delas, não posso deixar de agradecer a amizade que fiz com a Dâmaris Ribeiro Sá, graças as meninas que estão matriculadas no curso de química da UFF<sup>5</sup>. Quero muito que todas saibam o quanto foram e são importantes pra mim. Espero que essa amizade fique eternizada. Em especial, agradeço a Tia Flauzina, mãe da Clara Maria, por também ser como uma mãe maravilhosa para as meninas da casa. Obrigada por ter separado um tempinho para escutar tudo sobre meu trabalho de dissertação e ainda me dar várias dicas, presentes, almoço e janta.

Agradeço a equipe de professores do ICEX, por terem me recebido de volta com tanto carinho nesse último ano, onde ministrei as disciplinas de Geometria Analítica e Introdução à Matemática Superior, como professora de magistério superior. Tive uma enorme experiência profissional que colaborou para o meu aprendizado. Além disso, foi muito gratificante trabalhar no mesmo lugar que me formei como Bacharel em Matemática com ênfase em Matemática Computacional, em 2016. Tenho certeza que os laços de amizade que eu tinha com os professores na graduação só se fortaleceram, pois passaram a ser meus colegas de trabalho. Aos docentes do departamento da Matemática: Alan Prata de Paula, Alessandro Gaio Chimenton, André Ebling Brondani, Carlos Henrique Pereira do Nascimento, Edilaine Ervilha Nobili, Francisca Andrea Macedo França, Gilmar Garbugio, Honório Joaquim Fernando, Ivan Wilber Aguilar Maron, Jacqueline Bernardo Pereira Oliveira, Luiz Felipe Nobili França, Marina Ribeiro Barros Dias, Marina Sequeiros Dias, Miguel Adriano Koiler Schnoor e Rosemary Miguel Pires. Aos docentes do

---

<sup>3</sup>Instituto de Ciências Exatas

<sup>4</sup>Escola de Engenharia Industrial Metalúrgica de Volta Redonda

<sup>5</sup>Universidade Federal Fluminense

departamento da Física: Adriano de Oliveira Caminha, Adriano de Souza Martins, Aquino Lauri de Espíndola, Armando Flavio Rodrigues, Carlos Eduardo Fellows, Gustavo Luis Furtado Vicente, Jose Augusto Oliveira Huguenin, Ladário da Silva, Luiz Telmo da Silva Auler, Marcos Veríssimo Alves, Rodrigo Garcia Amorim, Thadeu Josino Pereira Penna e Vera Lucia Prudencia dos Santos Caminha. Aos docentes do departamento de Química: Alessandra Rodrigues Rufino, Denise de Castro Bertagnolli, Diego Pereira Sangi, Julliane Yoneda Huguenin, Ligia Maria Mendonça Vieira e Renata Luz Martins.

Aos seguranças e funcionários da EEIMVR e do ICEX, que sempre me cumprimentavam de bom humor e me deram atenção com relação as chaves, documentos e esclarecimento de dúvidas. Em especial, a Camila Souza, pelo forte laço de amizade que criamos e pela ajuda com os documentos, inscrições nas disciplinas e paciência com as minhas dúvidas.

O espaço aqui é pouco para agradecer a cada um em particular. Por isso, no geral, agradeço a todos os professores, amigos e colegas que me proporcionaram o conhecimento e, além disso, manifestaram caráter e afetividade. A todos aqueles que, de alguma forma, estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena. Até mesmo os que estão longe, abasteceram e ainda abastecem meu coração de sentimentos bons com certos aditivos.

Agradeço ao apoio da PROAES <sup>6</sup> da UFF, ao Programa de Pós-graduação Acadêmico, Stricto Sensu, em Modelagem Computacional em Ciência e Tecnologia da EEIMVR da UFF e ao NMM. Agradeço também a CAPES <sup>7</sup> pela oportunidade e auxílio financeiro durante o período do mestrado e a FAPERJ <sup>8</sup>.

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financiamento 001 e apoio do CNPq <sup>9</sup>.

Por fim, agradeço a todos que, de forma direta ou indiretamente, contribuíram para o meu desempenho.

---

<sup>6</sup>Pró-Reitoria de Assuntos Estudantis

<sup>7</sup>Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

<sup>8</sup>Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro

<sup>9</sup>Conselho Nacional de Desenvolvimento Científico e Tecnológico

# Resumo

A estereologia quantitativa é uma ciência que estuda características de campos amostrais em 3D a partir de características em 2D de uma determinada amostra. Todavia, a técnica de práticas estereológicas são apenas aproximações. Diversas características tridimensionais de uma amostra só podem ser determinadas com precisão a partir de estudos em três dimensões. Atualmente, o método de seccionamento em série (*serial sectioning*), seguido de reconstrução em 3D por renderização, é um dos mais empregados para estudar amostras tridimensionais. Este trabalho tem como objetivo, desenvolver um software no qual seja possível criar um volume a partir de imagens sequenciais em 2D de uma amostra. A sequência de imagens pode ser obtida por distintos métodos, sendo os mais usuais: seccionamento em série e a tomografia computadorizada. Na técnica de *serial sectioning*, o material é desbastado em uma determinada direção por diversas vezes e, em cada etapa de desbaste, é retirada uma foto do plano seguinte ao desbastado (normal à direção de desbaste). Ao fim da etapa de desbastes, todas as fotos são reunidas no programa proposto neste trabalho e transformadas em matrizes bidimensionais. Essas matrizes são sobrepostas formando uma matriz tridimensional. Este trabalho propõe o desenvolvimento de técnicas computacionais para usar a sintaxe de uma imagem digital e criar um domínio computacional com referências ao espaço discreto com coordenadas cartesianas em três dimensões. Conseqüentemente, a amostra é reconstruída numericamente. A amostra virtual agora reconstruída no computador, é um domínio computacional. Algumas aplicações em reconstrução 3D foram indicadas neste trabalho. O programa foi validado a partir de uma comparação com um resultado visualizado através do software ImageJ, já validado e aplicado em trabalhos encontrados na literatura. Comparações qualitativas foram realizadas com trabalhos de outros autores, e uma boa corroboração foi observada. Ao fim do trabalho, notou-se que o programa é capaz de reconstruir, com fidelidade, a amostra real. O código fonte aberto proporciona futuras melhorias e adaptações para novas situações, além de possibilitar escrever a saída de dados de acordo com o programa de renderização que o usuário optar por escolher.

# Abstract

Quantitative stereology is a science that studies the characteristics of 3D sample fields from the 2D characteristics of a given sample. However, the technique of stereological practices are only approximations. Several three-dimensional characteristics of a sample can only be accurately determined from three-dimensional studies. Nowadays, the serial sectioning method, followed by 3D rendering reconstruction, is one of the most used to study three-dimensional samples. This work aims to develop software in which it is possible to create a volume from sequential 2D images of a sample. The sequence of images can be obtained by different methods, being the most usual: serial sectioning and computed tomography. In the serial sectioning technique, the material is thinned in a certain direction for several times and, in each thinning step, a photo is taken from the next plane to the roughing (normal to the thinning direction). At the end of the thinning step, all the photos are gathered in the program proposed in this work and transformed into two-dimensional arrays. These arrays are superimposed into a three-dimensional matrix. This work proposes the development of computational techniques to use the syntax of a digital image and to create a computational domain with references to the discrete space with cartesian coordinates in three dimensions. Consequently, the sample is reconstructed numerically. The virtual sample now rebuilt on the computer, is a computational domain. Some applications in 3D reconstruction were indicated in this work. The program was validated in comparison to an image visualized through ImageJ software, already validated and applied in works found in the literature. Qualitative comparisons were performed with other authors, and a good corroboration was observed. At the end of the work, it was noticed that the program is able to reconstruct, with fidelity, the real sample. Open source provides future improvements and adaptations for new situations, as well as making it possible to write data output according to the rendering program that user could choose to select.

# Palavras-chave

1. Computação Gráfica
2. Domínio Computacional
3. Processamento de Imagens
4. Reconstrução 3D
5. Seccionamento em Série

# Glossário

CAPES	:	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNPq	:	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DMR	:	<i>Digital Material Representation</i>
EEIMVR	:	Escola de Engenharia Industrial Metalúrgica de Volta Redonda
FAPERJ	:	Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro
FIB	:	<i>Focused Ion Beam</i>
ICEx	:	Instituto de Ciências Exatas
MCCT	:	Modelagem Computacional em Ciência e Tecnologia
MEV	:	Microscópio Eletrônico de Varredura
MRI	:	<i>Magnetic Resonance Imaging</i>
NMM	:	Núcleo de Modelamento Microestrutural
OOP	:	<i>Object Oriented Programming</i>
OpenCV	:	<i>Open Source Computer Vision</i>
POO	:	Programação Orientada a Objetos
PROAES	:	Pró-Reitoria de Assuntos Estudantis
RM	:	Ressonância Magnética
SEM	:	<i>Scanning Electron Microscope</i>
TC	:	Tomografia Computadorizada ou Tomografia Computorizada
UFF	:	Universidade Federal Fluminense
UML	:	<i>Unified Modeling Language</i>

# Sumário

<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xx</b>
<b>1 Introdução</b>	<b>21</b>
1.1 Objetivos . . . . .	26
1.1.1 Objetivo Geral . . . . .	26
1.1.2 Objetivos Específicos . . . . .	26
<b>2 Fundamentação Teórica</b>	<b>27</b>
2.1 Imagem Digital . . . . .	27
2.1.1 Paradigma dos Quatro Universos . . . . .	28
2.1.2 Contínuo $\times$ Discreto . . . . .	30
2.1.3 Representação de uma Imagem . . . . .	33
2.1.3.1 Representação Vetorial e Matricial . . . . .	34
2.1.3.2 Representação Espacial . . . . .	36
2.1.3.3 Representação de Cor . . . . .	39
2.1.4 Cores e Modelo RGB . . . . .	41
2.2 Seccionamento em Série . . . . .	44
<b>3 Revisão Bibliográfica</b>	<b>50</b>
3.1 Estado da Arte . . . . .	50
3.2 Vantagens e Desvantagens da Técnica de Seccionamento em Série Convencional . . . . .	57

---

3.2.1	Vantagens . . . . .	58
3.2.2	Desvantagens . . . . .	58
3.3	Materiais . . . . .	60
<b>4</b>	<b>Metodologia</b>	<b>62</b>
4.1	Construção do Código Computacional . . . . .	63
4.1.1	Estrutura do Código . . . . .	64
4.1.2	Classe: Dados.cpp . . . . .	64
4.1.3	Classe: Imagem.cpp . . . . .	66
4.1.4	Classe: Matriz.cpp . . . . .	67
4.1.5	Arquivo principal: Principal.cpp . . . . .	67
4.1.5.1	Repetição de Planos Adjacentes . . . . .	67
4.1.5.2	Planos Intermediários Obtidos por meio da Média das Cores	69
4.1.6	Classe: Saída.cpp . . . . .	71
4.1.7	Algoritmo . . . . .	72
4.2	Binarização das Imagens . . . . .	75
4.3	Ferramentas Gráficas . . . . .	79
4.4	Informações do Computador Pessoal . . . . .	81
<b>5</b>	<b>Resultados e Discussão</b>	<b>82</b>
5.1	Ferro Fundido Nodular . . . . .	82
5.2	Aço Inoxidável Duplex . . . . .	86
5.3	Comparação entre as Técnicas de Repetição de Planos e Média das Cores .	89
5.4	Outras Possíveis Aplicações . . . . .	93
5.4.1	Tomografia Computadorizada . . . . .	93
5.4.1.1	Validação do Código . . . . .	101
5.4.1.2	Estudo Comparativo . . . . .	101

---

<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>103</b>
6.1	Conclusões . . . . .	103
6.2	Trabalhos Futuros . . . . .	105
<b>7</b>	<b>Produções Acadêmicas</b>	<b>106</b>
7.1	Resumos Aceitos em Simpósios e Encontros . . . . .	106
7.2	Resumos Publicados em Anais de Congressos . . . . .	107
7.3	Artigo Completo Publicado . . . . .	107
7.4	Capítulo de Livro . . . . .	107
7.5	Pôsteres Apresentados . . . . .	108
7.5.1	Votação de Pôsteres . . . . .	109
	<b>Referências</b>	<b>110</b>
	<b>Apêndice A - Como abrir um arquivo de extensão .dat no Tecplot 360<sup>®</sup></b>	<b>117</b>
	<b>Apêndice B - Como abrir um arquivo de extensão .dat no ParaView</b>	<b>127</b>
	<b>Apêndice C - Como abrir um arquivo de extensão .dat no VisIt</b>	<b>137</b>
	<b>Anexo A - Imagens de Ferro Fundido Nodular</b>	<b>149</b>
	<b>Anexo B - Imagens de Aço Inoxidável Duplex</b>	<b>153</b>

# Lista de Figuras

2.1	Paradigma dos Quatro Universos. . . . .	29
2.2	(a) Função contínua. (b) Função descontínua. . . . .	30
2.3	Discretização de Imagens. . . . .	32
2.4	Reta Orientada. . . . .	34
2.5	Segmento Orientado. . . . .	34
2.6	Representação do vetor $\vec{v}$ no plano cartesiano. . . . .	35
2.7	Arranjo de elementos em duas direções (matriz). . . . .	35
2.8	Descrição de imagens matriciais por conjunto de pixels. . . . .	36
2.9	Conjunto suporte da imagem. . . . .	36
2.10	Reticulado de discretização $F_{\Delta}$ . . . . .	37
2.11	Imagens de uma igreja com resoluções espaciais diferentes. . . . .	38
2.12	Figura original do carro. . . . .	40
2.13	Processo de amostragem e quantização de uma imagem. . . . .	40
2.14	Processo aditivo das cores primárias. . . . .	41
2.15	Processo de subtração de cores secundárias. . . . .	42
2.16	Modelo RGB a partir dos eixos XYZ. . . . .	43
2.17	Exemplo de duas pilhas de 10 seções em série. . . . .	45
2.18	(a) Amostra posicionada na Politriz Automática juntamente com o painel de controle. (b) Processo de polimento em andamento. . . . .	47
2.19	Fluxograma do processo de seccionamento em série. . . . .	48
3.1	Linha do tempo de alguns artigos relacionados ao tema. . . . .	50

3.2	(a) Pequeno volume microestrutural construído por seções bidimensionais resultantes da técnica de seccionamento em série. (b) Grande volume microestrutural obtido por pequenos volumes ao redor do volume apresentado em (a). . . . .	52
3.3	Principais componentes do <i>Robo-Met.3D</i> : polidor automático (esquerda), braço do robô (centro), estação de gravação automática (centro inferior) e microscópio invertido motorizado (direita). . . . .	53
3.4	(a) Desenho do equipamento: MEV acoplado com o FIB. (b) Aparência do sistema. . . . .	55
3.5	Microestrutura simulada pelo programa Autômato Celular. . . . .	56
3.6	Microscopia de amostra de Ferro Fundido Nodular. . . . .	60
3.7	Microscopia de amostra de Aço Inoxidável Duplex. . . . .	61
4.1	UML das classes contidas no código computacional. . . . .	64
4.2	A partir de duas cores principais escolheu-se, como exemplo, cinco cores intermediárias para serem calculados os valores de R, de G e de B, utilizando a média de cores. . . . .	71
4.3	Código do cabeçalho do arquivo de saída. . . . .	72
4.4	Exemplo de um cabeçalho de um arquivo de saída. . . . .	72
4.5	Exemplo da aplicação do método de binarização em uma imagem. . . . .	76
4.6	Aplicação do processo de binarização em uma seção bidimensional de Ferro Fundido Nodular. . . . .	76
4.7	Aplicação do processo de binarização em uma seção bidimensional de Aço Inoxidável Duplex. . . . .	77
4.8	Erro oriundo do processo de seccionamento em série. . . . .	78
4.9	Erro derivado do processo de binarização. . . . .	79
4.10	Ícones dos aplicativos. . . . .	79
5.1	Reconstrução 3D da microestrutura do Ferro Fundido Nodular. . . . .	82
5.2	Reconstrução 3D do Ferro Fundido Nodular visualizada pelo Tecplot 360 <sup>®</sup> . . . . .	83
5.3	Reconstrução 3D do Ferro Fundido Nodular visualizada pelo ParaView. . . . .	83

---

5.4	Reconstrução 3D do Ferro Fundido Nodular visualizada pelo VisIt. . . . .	84
5.5	Reconstrução 3D de Ferro Fundido Nodular com 57 seções bidimensionais. Visualização da reconstrução feita através do Tecplot 360 <sup>®</sup> . . . . .	85
5.6	Reconstrução 3D do Aço Inoxidável Duplex visualizada pelo Tecplot 360 <sup>®</sup> . . . . .	86
5.7	Reconstrução 3D do Aço Inoxidável Duplex visualizada pelo ParaView. . . . .	87
5.8	Reconstrução 3D do Aço Inoxidável Duplex visualizada pelo VisIt. . . . .	87
5.9	Reconstrução 3D de Aço Inoxidável Duplex com 57 seções bidimensionais. Visualização da reconstrução feita através do Tecplot 360 <sup>®</sup> . . . . .	88
5.10	Teste das abordagens implementadas no código desenvolvido, utilizando cores sólidas. . . . .	89
5.11	Resultados com Ferro Fundido Nodular e Aço Inoxidável Duplex. . . . .	90
5.12	Resultado da técnica de repetição de planos adjacentes. . . . .	91
5.13	Resultado da técnica de obtenção de planos intermediários por meio da médias das cores. . . . .	91
5.14	Área de interesse ampliada. . . . .	91
5.15	Área de interesse mais ampliada. . . . .	92
5.16	10 planos principais e 6 intermediários preenchidos com a Técnica da Repetição. . . . .	93
5.17	Imagens de TC em série. . . . .	94
5.18	Imagens de TC. . . . .	95
5.19	Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo $x$ e visualizadas através do Tecplot 360 <sup>®</sup> . . . . .	96
5.20	Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo $z$ e visualizadas através do Tecplot 360 <sup>®</sup> . . . . .	97
5.21	Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo $x$ e visualizadas através do ParaView. . . . .	98
5.22	Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo $z$ e visualizadas através do ParaView. . . . .	99
5.23	Aproximação da imagem para visualização dos detalhes. . . . .	100

---

5.24	Aproximação da imagem para visualização dos detalhes. . . . .	100
5.25	Visualização do resultado através do software ImageJ. . . . .	101
5.26	Imagens de TC toraco-abdomino-pélvica. . . . .	102
5.27	Reconstrução 3D feita a partir de imagens de TC toraco-abdomino-pélvica. . . . .	102
7.1	Pôsteres apresentados em congressos e simpósios. . . . .	108
7.2	Resultado da votação de pôsteres de pesquisa. . . . .	109
A.1	Tecplot 360 <sup>®</sup> 2010. . . . .	117
A.2	Abrindo arquivo no Tecplot 360 <sup>®</sup> . . . . .	118
A.3	Selecionando o formato do arquivo. . . . .	119
A.4	Selecionando arquivo para abrir no Tecplot 360 <sup>®</sup> . . . . .	120
A.5	Janela “Select Inicial Plot”. . . . .	120
A.6	Área em que os dados aparecerão. . . . .	121
A.7	Ativando o sistema de cores RGB. . . . .	121
A.8	Janela “RGB Coloring Options”. . . . .	122
A.9	Janela “RGB Coloring Options”. . . . .	122
A.10	Ativando a opção “Scatter”. . . . .	123
A.11	Configurando o RGB da opção “Scatter”. . . . .	123
A.12	Janela “Select Color”. . . . .	124
A.13	Configurando o formato dos pontos da malha. . . . .	124
A.14	Selecionando a opção “Sphere”. . . . .	125
A.15	Resultado: Microestrutura completa. . . . .	125
A.16	Resultado: Visualização da microestrutura de Ferro Fundido Nodular. . . . .	126
B.1	ParaView 5.4.1-RC3. . . . .	127
B.2	Abrindo arquivo no ParaView. . . . .	128
B.3	Selecionando arquivo para abrir no ParaView. . . . .	129
B.4	Selecionando o formato do arquivo. . . . .	130

---

B.5	Arquivo aberto no canto esquerdo do ParaView. . . . .	131
B.6	Área em que os dados aparecerão. . . . .	132
B.7	“Counter”. . . . .	132
B.8	Arquivo “Contour1”. . . . .	133
B.9	Mostrar/ocultar arquivos. . . . .	133
B.10	Visualização das esferas de grafite da microestrutura. . . . .	134
B.11	Rotação da imagem. . . . .	134
B.12	Modificando a cor das esferas de grafite. . . . .	135
B.13	Janela “Pick Solid Color”. . . . .	135
B.14	Resultado: Visualização da microestrutura de Ferro Fundido Nodular. . . .	136
C.1	VisIt 2.13.1. . . . .	137
C.2	Abrindo arquivo no VisIt. . . . .	138
C.3	Selecionando o tipo de arquivo. . . . .	139
C.4	Selecionando arquivo para abrir no VisIt. . . . .	140
C.5	Selecionando modo de abertura do arquivo. . . . .	141
C.6	Arquivo aberto na seção “Sources” em frente “Active source”. . . . .	142
C.7	Adicionando opções para visualização. . . . .	143
C.8	Ativando a visualização no VisIt. . . . .	144
C.9	Resultado: esferas de grafite da microestrutura de Ferro Fundido Nodular. .	145
C.10	Rotação da imagem. . . . .	145
C.11	Modificando a cor das esferas de grafite. . . . .	146
C.12	Modificando a cor das esferas de grafite. . . . .	147
C.13	Resultado: Visualização da microestrutura de Ferro Fundido Nodular.. . .	148
A.1	Imagens de Ferro Fundido Nodular obtidas através da técnica de Seccionamento em Série. . . . .	152
B.1	Imagens de Aço Inoxidável Duplex obtidas através da técnica de Seccionamento em Série. . . . .	159

# Lista de Tabelas

4.1	Informações referentes ao computador utilizado neste trabalho. . . . .	81
5.1	Informações da reconstrução 3D apresentada nas Figuras 5.2, 5.3 e 5.4. . .	84
5.2	Informações da reconstrução 3D apresentada na Figura 5.5. . . . .	85
5.3	Informações da reconstrução 3D apresentada nas Figuras 5.6, 5.7 e 5.8. . .	87
5.4	Informações da reconstrução 3D apresentada na Figura 5.9. . . . .	88

# Capítulo 1

## Introdução

A proposta deste trabalho é motivada pela dificuldade de recriar domínios computacionais com fidelidade ao meio natural [22]. Domínios computacionais (ou volumes de controle) são regiões nas quais se deseja determinar alterações em variáveis específicas que evoluem de maneira dinâmica e são, geralmente, modelados a partir de equações algébricas, representação de computação gráfica, criação de malhas computacionais a partir de eixos cartesianos, dentre outros sistemas. A representação computacional de detalhes de domínios com geometrias complexas tem sido um problema na maioria dos casos nos quais a geometria põe em risco a integridade dos resultados gerados via análise numérica. Com o advento da computação, novas técnicas de representação gráfica foram criadas e com elas novas formas de expressar modelos físicos por meio de técnicas computacionais. Todavia, as representações geométricas complexas permaneciam com deficiência devido à dificuldade na reprodução computacional de determinadas formas ou padrões irregulares [21, 23]. Modelar regiões de geometrias complexas como um cérebro com todos detalhes dos sulcos cingulados laterais e centrais, o corpo caldoso e o cerebelo, não é uma tarefa nada simples. Entretanto, a fidelidade na geometria é de extrema importância, pois milímetros de diferença do original podem causar relevantes erros nos resultados finais em determinados casos, como em cálculos de concentrações de células cancerosas, por exemplo. A partir da evolução computacional e do surgimento de novas tecnologias, surgiram também novas maneiras de armazenamento de informações geométricas, de peças, regiões, áreas e etc. A fotografia na sua forma analógica foi uma das primeiras formas de se eternizar uma cena e foi criada em 1826 por Joseph Nicéphore Niépce [8]. Contudo, apenas em 1975 um engenheiro da Kodak criou a primeira câmera digital, que foi comercializada em 1990. Sendo assim, somente a partir da década de 90, com este recurso em mãos, a foto digital passou a ser salva em computadores

nos formatos que são conhecidos atualmente. As fotos capturadas são salvas em arquivos digitais de diferentes formatos. Estes arquivos são portadores de algumas informações, tais como: tamanho da imagem, densidade de pixels, níveis de vermelho, verde e azul em cada pixel e etc. Além desses atributos, o mais importante é a organização dos pixels que forma o sentido sintático da fotografia. Mensagens visuais costumam ser estudadas em termos de forma e conteúdo. Nesse caso, pode-se considerar três níveis: morfologia (a configuração da fotografia), sintaxe (a organização visual dos elementos) e semântica (o significado da imagem). A organização visual dos elementos é responsável por posicionar cada pixel em seu devido lugar. Deste modo, a representação geométrica da cena é mantida a rigor, e formas geométricas complexas podem ser representadas com fidelidade.

A microscopia óptica, a microscopia eletrônica, a tomografia computadorizada (TC) e a ressonância magnética (RM) são técnicas que fornecem apenas imagens bidimensionais (2D), sem fornecer informações quantitativas diretas sobre a estrutura interna tridimensional (3D) de um determinado material, objeto de interesse ou de um corpo humano. Uma microestrutura pode ser definida como a estrutura detalhada de um material, obtida através do microscópio. Cada microestrutura apresenta características particulares, tais como o tamanho, informações geométricas, distribuição de fases e morfologia. E, para entender o comportamento do material envolvido, é necessário estabelecer um vínculo entre a microestrutura e suas propriedades [81, 82]. Do mesmo modo, existem diferentes objetos com suas respectivas particularidades, assim como, cada ser humano possui um conteúdo corporal com característica própria. Logo, tais elementos, que possuem uma natureza tridimensional, não devem ser analisados apenas com imagens bidimensionais, em razão de que existem casos em que essa análise se torna insatisfatória. Em particular, certas informações geométricas e morfológicas sobre uma microestrutura tridimensional não podem ser deduzidas devido a limitações bidimensionais, pois não possibilita a observação de um volume microestrutural da amostra. Nesse caso, a aplicação de técnicas de modelagem numérica pode apoiar a investigação experimental tridimensional realizada através da reconstrução 3D [24, 71].

Aplicado a área metalúrgica, o método utilizado neste trabalho permite a reconstrução das estruturas internas de materiais de modo que o resultado final seja um conjunto de dados 3D que torne possível visualizar e identificar elementos tais como poros, grãos, fases, precipitados, dentre outros. Além disso, foram feitas reconstruções tridimensionais utilizando imagens de tomografia computadorizada (TC) de abdômen, cuja aplicação é voltada para área da medicina.

Na metalurgia, inicialmente é necessário aplicar uma técnica, conhecida como *Seccionamento em Série*, no material de interesse. Esta técnica é empregada para adquirir seções bidimensionais de uma determinada microestrutura [28, 29]. Como resultado desse processo, obtém-se uma sequência de imagens metalográficas. Na área da medicina, pode-se escolher uma série de imagens de tomografia computadorizada (TC) ou de imagens de ressonância magnética (MRI). Estes tipos de sucessões de imagens ordenadas servem como dados de entrada para o código computacional que foi desenvolvido neste trabalho. O modo como o programa está sendo implementado fornece um arquivo de saída que contém os dados necessários para gerar uma malha discreta que representa a reconstrução tridimensional e é aplicada em diversas áreas da ciência de maneira geral, através de softwares de visualização 3D. Com este programa é possível realizar novos estudos com a matriz reconstruída, pois os dados não são descartados.

A profundidade de desbastamento, de materiais metálicos, entre uma imagem bidimensional e sua imagem subsequente influência nos resultados da reconstrução. Em trabalhos anteriores observou-se que, de acordo com o aumento da distância entre as camadas, aumenta também o erro relativo de representação do volume integrado (imagem em 3D) [5]. Para tentar minimizar esse erro utilizou-se duas abordagens distintas, implementadas no código computacional, que superam a perda de informações que o processo de seccionamento em série pode causar na hora do desbastamento de uma amostra real [23].

Uma viabilidade interessante é que, com a aplicação do processo de seccionamento em série em uma amostra real de um determinado tipo de material, apenas esta amostra é destruída. E após a reconstrução da mesma, diversos testes podem ser realizados computacionalmente. Caso os testes fossem feitos no mundo real, haveriam muitas amostras sendo destruídas e, com isso, custos mais elevados.

O paradigma da Programação Orientada a Objetos (POO) foi utilizado no decorrer do desenvolvimento do código computacional. Os objetos são partes reutilizáveis de um programa que modelam o mundo real. Tem-se percebido que a utilização de uma abordagem de implementação de códigos orientados a objetos torna os grupos de desenvolvimento mais produtivos do que quando utilizadas somente técnicas de programação estruturada. A orientação a objetos permite a criação de programas mais fáceis de entender, corrigir e modificar [27, 31].

O restante deste trabalho foi organizado da seguinte maneira: o Capítulo 2 traz a fundamentação teórica realizada para iniciar o trabalho. Nesse sentido, foi feita uma revisão de textos, livros e artigos sobre o assunto estudado. De início, preparou-se uma breve exposição sobre conceitos centrais de Imagem Digital. Isto envolve teorias de Processamento de Imagem [1, 61] e inclui o Paradigma dos Quatro Universos [36, 37], mostrando a diferença entre o universo contínuo e o universo discreto. Este capítulo também abrange o modo como pode-se representar uma imagem através do computador, utilizando as técnicas de amostragem e quantização. Além disso, apresenta-se, brevemente, o sistema de cores RGB [7] que foi utilizado como base no decorrer do trabalho. Na sequência, o capítulo relata os conceitos e aplicações de uma técnica para reconstrução tridimensional chamada *Seccionamento em Série*, que possui uma grande importância para a realização deste trabalho. É a partir desta técnica que consegue-se dar continuidade a presente pesquisa, pois seus resultados servem de dados de entrada para a ferramenta computacional principal que foi desenvolvida por este trabalho. O Capítulo 3 faz uma revisão bibliográfica, na qual começa por um breve resumo das referências [10, 12, 14, 41, 47, 51, 63, 64]. A partir disso, foram relatadas diferentes técnicas de seccionamento serial, seguida de reconstrução 3D, com base no que os autores disseram e escreveram sobre o assunto. Também foi mostrado o que esses autores utilizaram em seus trabalhos. Outras referências, sobre seccionamento em série, foram expostas na última seção do capítulo mencionado anteriormente e no decorrer do texto. Posteriormente, fala-se sobre as vantagens e desvantagens da a técnica convencional de seccionamento em série e expõe-se a opinião de alguns autores. Ademais, o capítulo salienta um pouco sobre dois materiais metálicos nos quais a técnica foi aplicada e testada: Ferro Fundido Nodular e Aço Inoxidável Duplex. O Capítulo 4 aduz a metodologia efetuada para a progressão do trabalho. A primeira seção descreve como o código computacional foi construído e estabelece suas divisões. Para isto, as subseções divulgam detalhes de cada classe criada no código, além de expor uma discussão a respeito de duas abordagens elaboradas para resolver o problema de determinação de valor entre dois planos adjacentes. A segunda seção descreve a ideia de binarização de imagens e mostra exemplos de como essa técnica foi empregada. Sequencialmente, a penúltima seção fornece um resumo sobre as três ferramentas gráficas utilizadas para visualização dos resultados obtidos por meio do código computacional desenvolvido. E a última seção contém as informações da máquina utilizada para realizar a pesquisa, desenvolver o código computacional, gerar os resultados envolvidos e visualizá-los através de tais ferramentas gráficas. O Capítulo 5 expõe resultados de reconstruções tridimensionais

das microestruturas de Ferro Fundido Nodular e Aço Inoxidável Duplex apenas com o empilhamento das imagens principais obtidas através da técnica de seccionamento em série. Na sequência, exibe-se também outras reconstruções 3D, das duas microestruturas citadas, que foram realizadas utilizando as duas abordagens propostas: a repetição de planos adjacentes e o preenchimento de planos intermediários através da média das cores. Posteriormente, exibe uma aplicação na medicina onde foi feita uma reconstrução 3D, a partir de imagens bidimensionais de tomografia computadorizada (TC), de um abdômen. Para isto, incluiu-se uma breve definição de TC seguida de tais resultados. A fim de comparar os resultados de autoria própria com os resultados da literatura, também foi exposto resultados produzidos por outros autores. Todos os resultados de autoria própria, contidos no capítulo em questão, foram obtidos através de ferramentas gráficas: Tecplot 360 [68], ParaView [6] e/ou o VisIt [18], bem como uma análise de tais resultados e uma comparação entre essas três ferramentas gráficas utilizadas com o propósito de visualizar as reconstruções tridimensionais. O Capítulo 6 apresenta as conclusões do trabalho e das análises dos resultados, assim como descrever algumas propostas para trabalhos futuros. O último capítulo (Capítulo 7) ficou reservado para expor as produções acadêmicas realizadas decorrente da atual pesquisa. Para mais, nas últimas páginas deste arquivo, encontram-se três apêndices onde se explica o passo a passo de como abrir arquivos com formato \*.dat em cada uma das três ferramentas de visualização gráfica utilizadas no trabalho: Tecplot 360<sup>®</sup>, ParaView e VisIt. Além disso, encontram-se também dois anexos que contém todas as imagens de Ferro Fundido Nodular e Aço Inoxidável Duplex que foram obtidas por outros autores através do seccionamento serial.

Assumi-se que o leitor tenha conhecimento prévio de alguns cursos básicos de graduação da área de ciências exatas. Dentre eles, destacam-se os cursos de Álgebra Linear, Geometria Analítica e Cálculo de uma e várias variáveis. É suposto também, que o leitor tenha conhecimentos de estrutura de dados, teoria de algoritmos e esteja familiarizado com a linguagem de programação C++.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

O objetivo geral é desenvolver uma ferramenta computacional para criar domínios discretos em 3D com fidelidade a ambientes ou volumes de controle contínuos, extraídos da natureza, por meio de fotos, através da técnica de seccionamento em série ou tomografia computadorizada.

### 1.1.2 Objetivos Específicos

- Realizar reconstrução 3D a partir de uma série de imagens em 2D utilizando o código computacional desenvolvido;
- Desenvolver código computacional utilizando o paradigma da linguagem de programação conhecido como Programação Orientada a Objetos (POO);
- Aplicar métodos computacionais/numéricos para resolver o problema de determinação de valor entre dois planos discretos;
- Visualizar o domínio construído com auxílio de alguma ferramenta gráfica já utilizada em outras referências, tais como Tecplot 360<sup>®</sup> [68], ParaView [6], VisIt [18], Mathematica, Matlab<sup>®</sup> ou outros;
- Permitir que a malha possa ser trabalhada após a reconstrução;
- Analisar os resultados obtidos.

Como a maioria dos programas de reconstrução 3D são de alto custo, não são escritos para alto desempenho e não possibilitam continuar trabalhando com a malha reconstruída ou após a reconstrução, a meta do projeto é suprir essas necessidades.

# Capítulo 2

## Fundamentação Teórica

### 2.1 Imagem Digital

O que é uma imagem? Computacionalmente falando, é possível pensar em uma imagem como se fosse um plano. E um plano possui pontos. No entanto, sabe-se que no mundo real existem infinitos pontos. Logo, uma imagem pode ser considerada como uma relação de um par de números reais que aponta para outro número real. Essa ideia pode ser representada, simbolicamente, como:  $\mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ . Um par de números reais  $(x, y)$  é um ponto do plano da imagem. O número real que é apontado é a cor. Assim, a cor pode ser representada como um número. Porém, o computador não trabalha com números reais. Trabalha apenas com zeros e uns. A ideia básica que está por trás do computador é poder contar os pontos, diferentemente do mundo real em que não é possível contá-los. Para isto, é preciso realizar um processo que denomina-se *Amostragem*. Realizar o processo de amostragem significa selecionar alguns pontos do mundo real, mesmo que sejam muitos, e passar esses pontos para o computador de modo que seja possível contá-los, isto é, de forma que este conjunto de pontos seja enumerável <sup>1</sup>. Além disso, existem infinitas cores. Mas, da mesma maneira, o computador não consegue representar todas elas. Por isso, é preciso conseguir contá-las também. E com isso, existe o processo que denomina-se *Quantização*.

Neste capítulo, foram apresentados alguns conceitos sobre processamento digital de imagens [36, 37, 38] que foram utilizados ao longo deste trabalho. Isso inclui o Paradigma dos Quatro Universos, a diferença entre o meio contínuo e o meio discreto, conceitos de amostragem e quantização de imagens e, por fim, foi apresentado o modelo de cores

---

<sup>1</sup>Um conjunto  $X$  qualquer é *enumerável* quando é finito ou quando existe uma bijeção  $f : \mathbb{N} \mapsto X$ , onde  $\mathbb{N}$  é o conjunto dos números naturais.

RGB que foi de primordial importância na implementação de abordagens discutidas nos próximos capítulos.

### 2.1.1 Paradigma dos Quatro Universos

Antes de começar a falar sobre representação e manipulação de imagens no computador, é necessário apresentar um paradigma de abstração na qual denomina-se de *Paradigma dos Quatro Universos*. Com ele, é possível ter uma ideia melhor de como é sair do mundo real, onde tudo é contínuo, e entrar para o mundo computacional, onde tudo é discreto. Esse paradigma é importante para dar sequência nas próximas discussões, visto que é necessário modelar diversos objetos em estudo. Ele ajudou a compreender todos os conceitos que foram descritos adiante. Para descrevê-lo, foi necessário estabelecer quatro conjuntos que compõe esse paradigma: o universo físico  $\mathcal{F}$ , o universo matemático  $\mathcal{M}$ , o universo de representação  $\mathcal{R}$  e o universo de implementação  $\mathcal{I}$  [37].

1. Universo Físico  $\mathcal{F}$ : é o conjunto no qual se encontra os objetos ou fenômenos do mundo real, isto é, grandezas analógicas não digitalizadas, que pretende-se estudar e representar. Como exemplo, pode-se citar: voz humana, tipos de solo, cadastro urbano e rural, dados geofísicos e topográficos.
2. Universo Matemático  $\mathcal{M}$ : é o conjunto que contém as descrições abstratas dos objetos encontrados no universo físico, ou seja, é o universo no qual se equaciona as grandezas analógicas para, posteriormente, representá-las na forma digital. O que se quer dizer é que nesse universo está incluso uma definição matemática (formal) das entidades a serem representadas. Por exemplo: dados temáticos e cadastrais, modelos numéricos de terreno, dados de sensoriamento remoto.
3. Universo de Representação  $\mathcal{R}$ : é o conjunto que permite trazer as descrições abstratas do universo matemático para o mundo digital. Nesse universo, as grandezas analógicas são representadas por descrições simbólicas e finitas associadas a objetos do universo matemático. É aqui que utiliza-se os processos de discretização (amostragem e quantização) que foram definidos com auxílio da Figura 2.3 e discutidos com mais detalhes na seção 2.1.3. Portanto, é nesse mesmo universo que ocorre a discretização dos objetos contínuos.
4. Universo de Implementação  $\mathcal{I}$ : é o conjunto no qual utiliza-se estruturas de dados e algoritmos que são associadas às descrições do universo de representação. Nesse

universo representa-se, especifica-se e codifica-se os formatos que representarão a grandeza analógica de forma digital. É importante lembrar que as estruturas de dados são escolhidas levando em consideração o desempenho, capacidade do equipamento e tamanho da massa de dados.

Com o auxílio do Gliffy <sup>2</sup> e softwares voltados para edição de fotos, PhotoScape <sup>3</sup> e Picasa <sup>4</sup>, foi esquematizado o paradigma dos quatro universos na Figura 2.1.

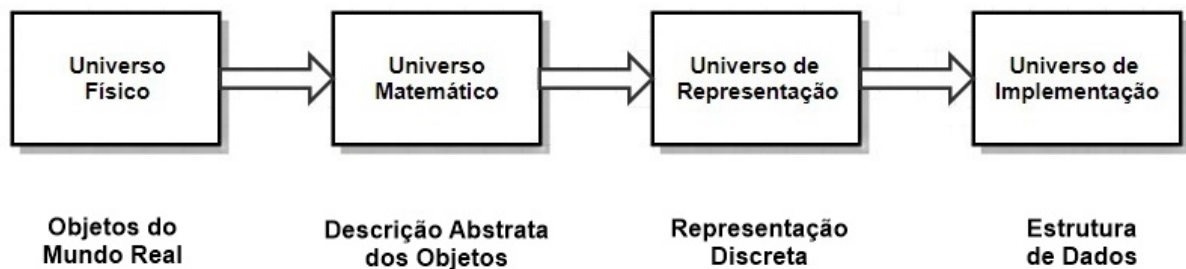


Figura 2.1: Paradigma dos Quatro Universos.  
Fonte: Adaptada de [37].

Assim, o paradigma de abstração, encontrado na Figura 2.1, se baseia no fato de que para estudar um determinado fenômeno ou objeto do mundo real (universo  $\mathcal{F}$ ), computacionalmente, é preciso associá-lo a um modelo matemático (universo  $\mathcal{M}$ ) e então encontrar uma representação discreta (universo  $\mathcal{R}$ ) para esse modelo que poderá ser implementado no computador (universo  $\mathcal{I}$ ).

No Exemplo 1 é possível ver como uma família de esferas pode se enquadrar no Paradigma dos Quatro Universos.

**Exemplo 1.** Bolas de Gude.

- Universo  $\mathcal{F}$  - Neste exemplo, considere os objetos do mundo real sendo superfícies esféricas de bolas de gude.
- Universo  $\mathcal{M}$  - O modelo matemático é a descrição na forma de função. Neste caso, existem duas formas:

a) Forma implícita:  $x^2 + y^2 + z^2 = r^2$ ,  $r > 0$ , onde  $r$  é o raio da esfera.

<sup>2</sup><https://www.gliffy.com/>

<sup>3</sup><https://photoscape.br.uptodown.com/windows>

<sup>4</sup><https://picasa.google.com.br/>

b) Forma paramétrica:  $(x, y, z) = r(\cos u \cos v, \operatorname{sen} u \cos v, \operatorname{sen} v)$ ,  $r > 0$ ,  $0 \leq u \leq \pi$ ,  $0 \leq v \leq 2\pi$ .

- Universo  $\mathcal{R}$  - A representação são os parâmetros da descrição funcional acima. Neste caso, nome e raio  $(id, r)$ .
- Universo  $\mathcal{I}$  - Para implementação utiliza-se estrutura de dados. Mais especificamente, pode-se escolher uma lista associativa que, neste caso, tenha a seguinte forma:  $id_k \mapsto r_k$ , tal que  $k = 1, \dots, M$ , onde  $M$  é a quantidade de bolas de gude na qual pretende-se estudar ou representar computacionalmente.

### 2.1.2 Contínuo $\times$ Discreto

O conceito matemático de função é largamente utilizado e é de fundamental importância para o entendimento de algumas noções durante as discussões deste capítulo. A seguir, é introduzida uma ideia delicada do Cálculo que diz respeito ao estudo de funções, o conceito de continuidade de uma função. Assim, foi possível entender o que são funções contínuas e funções não contínuas (ou funções discretas). Nesta seção foi descrito os aspectos básicos desse conceito.

Sejam  $g$  e  $h$  funções ilustradas nas Figuras 2.2a e 2.2b, respectivamente. Foi possível representá-las com o auxílio do software Geogebra <sup>5</sup> e do PhotoScape <sup>6</sup>.

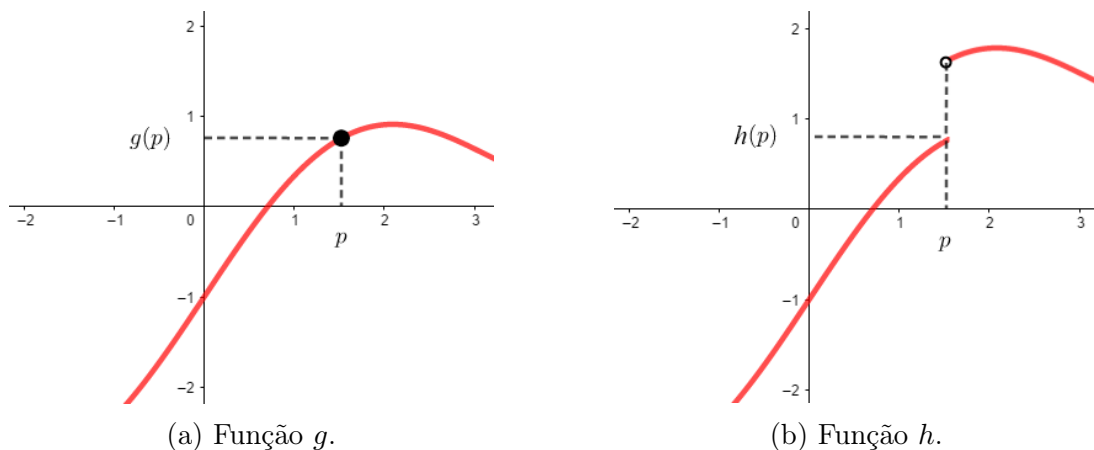


Figura 2.2: (a) Função contínua. (b) Função descontínua.

Fonte: Adaptada de [40].

Note que as funções  $g$  e  $h$ , contidas na Figura 2.2, se comportam de modo diferente no ponto  $p$ . O gráfico de  $g$  não apresenta “salto” em  $p$ , ao passo que o gráfico de  $h$ , apresenta.

<sup>5</sup><https://www.geogebra.org/>

<sup>6</sup><https://photoscape.br.uptodown.com/windows>

Essa é a noção intuitiva de uma *função contínua em um ponto  $p$  de seu domínio*.

Considere o termo “possuir salto” como uma propriedade. Logo,  $g$  não satisfaz esta propriedade no ponto  $p$ . Porém, a função  $h$  satisfaz. Pode-se dizer, então, que uma função é *contínua no ponto  $p$*  se ela não satisfaz a propriedade definida acima. Caso contrário, considera-se que a função é *discreta* ou *descontínua no ponto  $p$* .

Matematicamente, é permitido definir, a rigor, o conceito de função contínua de uma única variável. Atente-se a Definição 1.

**Definição 1.** Sejam  $g$  uma função unidimensional e  $p$  um ponto de seu domínio  $D_g$ .

$g$  é *contínua em  $p$*   $\Leftrightarrow$  Para todo  $\epsilon > 0$  dado, existe  $\delta > 0$  ( $\delta$  dependendo de  $\epsilon$ ), tal que, para todo  $x \in D_g$ ,

$$p - \delta < x < p + \delta \Rightarrow g(p) - \epsilon < g(x) < g(p) + \epsilon.$$

Equivalentemente,

$g$  é *contínua em  $p$*   $\Leftrightarrow \forall \epsilon > 0$  dado,  $\exists \delta = \delta(\epsilon) > 0$ , tal que,  $\forall x \in D_g$ ,

$$|x - p| < \delta \Rightarrow |g(x) - g(p)| < \epsilon.$$

Portanto, uma função é dita *contínua em seu domínio* ou simplesmente *contínua* se ela for contínua em todos os pontos pertencentes ao seu domínio. Mais detalhes sobre as noções de Cálculo podem ser encontradas em [40] e [50].

A partir das ideias anteriores, é necessário definir mais dois conceitos que foram úteis no decorrer do texto. Definem-se como *dados contínuos* os dados quantitativos que podem tomar todos os valores numéricos compreendidos no seu intervalo de variação. Denominam-se *dados discretos* os dados quantitativos que só podem tomar um número finito, ou infinito enumerável, de valores distintos.

Como os computadores trabalham apenas com números inteiros, ou no máximo com aproximações de números reais (chamados pontos flutuantes), não é possível representar funções contínuas computacionalmente. Logo, são feitas aproximações. É possível, apenas, simular a continuidade de uma função através do computador. E para isso utiliza-se o processo de discretização. Isto que dizer que basta tomar valores pontuais ao longo do plano  $xy$  e guardar o valor da função bidimensional aplicada nas coordenadas  $(x, y)$ . É necessário discretizar tanto o domínio como o contra-domínio da função imagem que está definida na seção 2.1.3.

O processo de discretização do domínio é chamado de *Amostragem*. O processo de discretização do contra-domínio é chamado de *Quantização*. Mais detalhes sobre estes dois processos também podem ser vistos na seção 2.1.3.

A Figura 2.3 foi elaborada através da utilização do TikZ, um pacote feito para criar gráficos de forma programática. O TikZ permite que a criação de gráficos sofisticados de forma bastante intuitiva e fácil. Com isso, ao olhar para Figura 2.3 é possível absorver uma noção do que é o processo de amostragem e o processo de quantização, quando o domínio da função imagem está contido em  $\mathbb{R}^2$  e o contra-domínio está contido em  $\mathbb{R}^3$ .

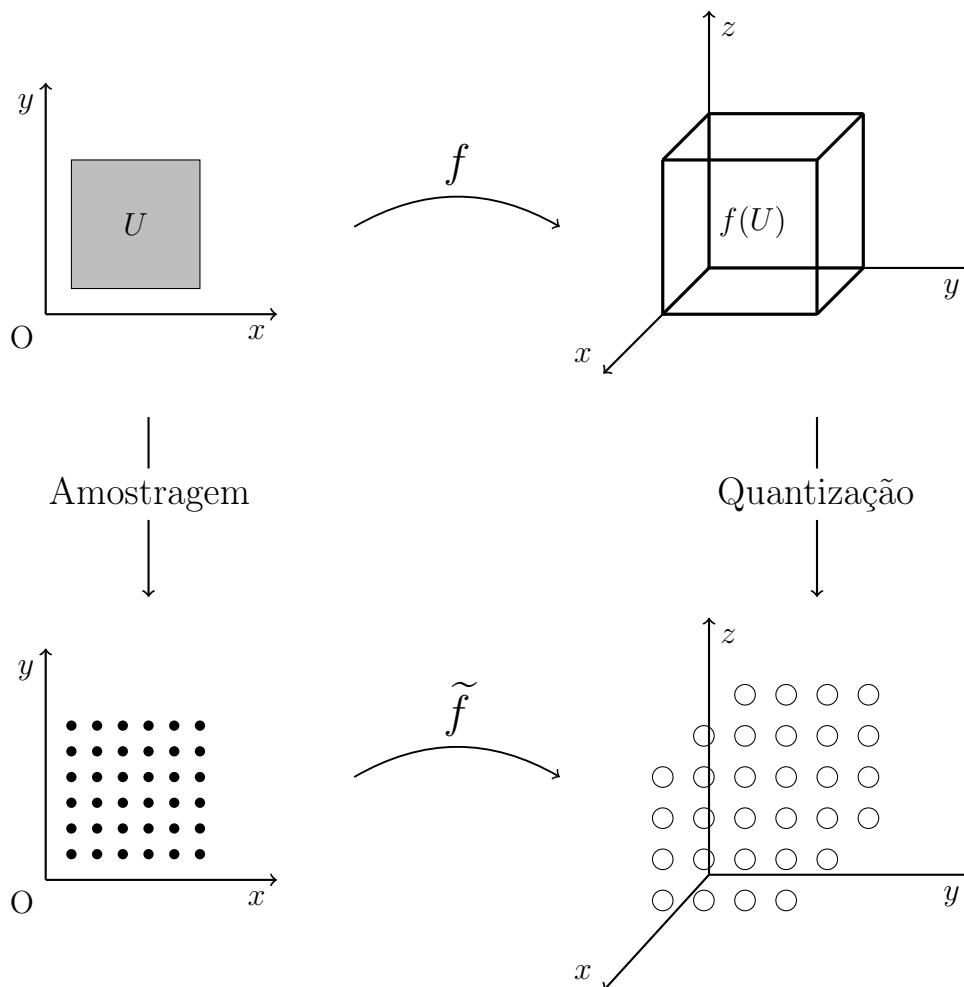


Figura 2.3: Discretização de Imagens.

Fonte: Adaptada de [36].

A função  $\tilde{f}$  possui a mesma regra da função  $f$ , porém seu domínio e sua imagem são “menores” do que o domínio e imagem da função  $f$ . Escolheu-se esta notação para enfatizar que o processo de amostragem e quantização foi aplicado em  $f$  e o resultado é a função  $\tilde{f}$ .

### 2.1.3 Representação de uma Imagem

Um modelo matemático que pode ser usado para descrever uma imagem é o de uma função. Logo, uma imagem pode ser definida como uma aplicação  $f : U \subset \mathbb{R}^2 \rightarrow C$ , onde  $U$  é o conjunto *suporte da imagem* e  $C$  é o *espaço de cor* [36, 37], conforme a Figura 2.3. Isto é, a função imagem é uma função bidimensional  $f(x, y)$ , onde  $x$  e  $y$  são as coordenadas espaciais do plano.

Dá-se o nome de *intensidade*, *amplitude* ou *nível de cinza* (no ponto  $x, y$ ) a função  $f$  aplicada em qualquer par de coordenadas  $(x, y)$ , ou seja, os valores de intensidade são os valores dados pela altura  $z = f(x, y)$ . Se  $x$ ,  $y$  e  $z$  forem quantidades finitas e discretas, pode-se dizer que essa imagem é uma *imagem digital*. Isto significa que uma imagem digital é composta por um número finito de elementos. Caso contrário, a imagem é definida como uma *imagem analógica* (representação real da cena).

Para criar uma imagem digital, é necessário converter dados contínuos para dados discretos. Isso envolve dois processos que, na seção 2.1.2, foram definidos brevemente: *amostragem* e *quantização*. A discussão sobre cada um deles se encontra nas subseções 2.1.3.2 e 2.1.3.3.

O conjunto de valores  $f(U) \subset C$  é chamado de *cores da imagem* ou *gamute de cores* da imagem. E em geral, tem-se que  $C = \mathbb{R}^n$ . No caso de uma imagem monocromática, considera-se  $n = 1$ . Quando uma imagem é composta por vários tons de uma única cor (pode ser qualquer cor), ela recebe o nome de *imagem monocromática*. Um exemplo que pode ser citado, é o efeito *sépia*, que dá o tom mais amarronzado em uma fotografia. Já uma imagem tricromática é uma imagem que é composta por três diferentes tons de cor. Em geral, utiliza-se o modelo RGB, que está melhor comentado na seção 2.1.4 e representa as três cores primárias: *Red* (Vermelho), *Blue* (Azul) e *Green* (Verde). A maioria das cores que um olho humano consegue enxergar são representadas por combinações dessas três cores. Assim, uma representação comum para uma imagem colorida utiliza três bandas: R, G, e B.

As próximas subseções descrevem alguns tipos de representações de imagens que existem na literatura estudada.

### 2.1.3.1 Representação Vetorial e Matricial

Para entender como se representa uma imagem através de vetores, é necessário relembrar os conceitos de reta e segmento orientado. Com o propósito de garantir clareza ao leitor, sugere-se as referências [66, 79].

Seja  $s$  uma reta pertencente ao espaço bidimensional, vide Figura 2.4. Diz-se que  $s$  é *orientada* quando se escolhe um sentido de percurso e uma seta (ou flecha) que caracteriza visualmente o seu sentido, considerando este sentido como sendo positivo.

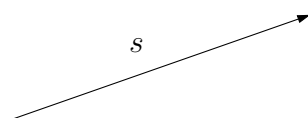


Figura 2.4: Reta Orientada.  
Fonte: Autoria própria (2018).

Assim, é possível determinar um *segmento orientado* (Figura 2.5) por um par ordenado de pontos  $A(x_0, y_0)$  e  $B(x_1, y_1)$ , onde o primeiro é chamado de *origem* e o segundo é a *extremidade*. Em geral, considera-se  $x_0 = y_0 = 0$  como sendo a origem do sistema de coordenadas em questão.

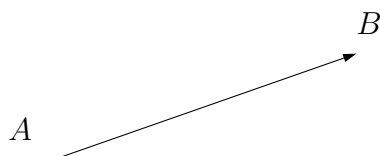


Figura 2.5: Segmento Orientado.  
Fonte: Autoria própria (2018).

Portando, a Geometria Analítica nos permite representar um *vetor* como um segmento de reta orientado que é caracterizado por *direção*, *sentido* e *módulo* (ou comprimento ou intensidade).

Seja  $\vec{v} = (a, b)$  um vetor do espaço bidimensional, conforme a Figura 2.6. Pelo teorema de Pitágoras, pode-se concluir que o módulo do vetor  $\vec{v}$  é dado pela raiz da soma das coordenadas ao quadrado, ou seja,

$$|\vec{v}| = \sqrt{a^2 + b^2}$$

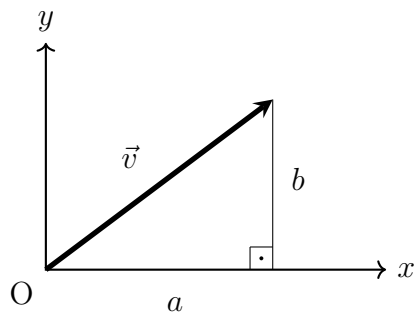


Figura 2.6: Representação do vetor  $\vec{v}$  no plano cartesiano.  
Fonte: Autoria própria (2018).

Se essa ideia for estendida para o espaço tridimensional, tem-se:

$$|\vec{v}| = \sqrt{a^2 + b^2 + c^2}, \quad \text{onde } \vec{v} = (a, b, c).$$

Os elementos básicos que são utilizados na representação vetorial são chamados de *primitivas vetoriais* da imagem. Eles podem ser pontos, linhas, curvas e superfícies tridimensionais. Como exemplo, escolha um elemento pertencente a um conjunto de pontos. Pode-se associar este elemento a uma determinada posição que é representada por coordenadas. E uma cor associada a essa posição aparecerá na tela do computador [7].

Para comentar sobre a representação matricial, é necessário definir o que é um arranjo (*array*) de elementos em duas direções. Neste tipo de representação a imagem é descrita por um conjunto de células em um arranjo espacial bidimensional que é chamado de matriz, vide Figura 2.7. O primeiro passo é definir quantos elementos vão existir em cada direção. Cada célula representa um pixel da imagem. Com isso, pode-se formar objetos através de um conjunto de pixels [7]. Por exemplo, considere uma matriz de tamanho  $4 \times 4$ .

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Figura 2.7: Arranjo de elementos em duas direções (matriz).  
Fonte: [7].

Para formar os objetos a partir deste conceito, necessita-se utilizar adequadamente os pixels. Na Figura 2.8 é ilustrado a forma como a imagem fica salva na memória do

computador e indica como a imagem é vista através da tela do computador.

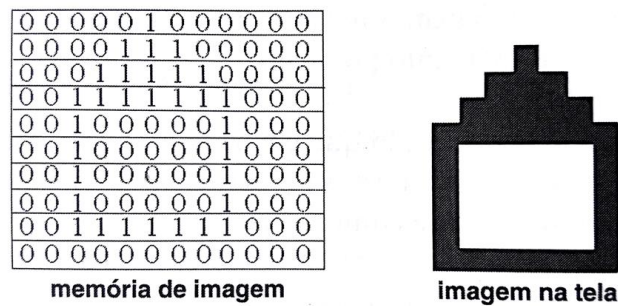


Figura 2.8: Descrição de imagens matriciais por conjunto de pixels.

Fonte: [7].

No decorrer deste trabalho foi utilizado uma biblioteca de Processamento de Imagem para o desenvolvimento da ferramenta proposta. Esta biblioteca é a OpenCV [42]. E ela lida com a representação matricial de imagens.

### 2.1.3.2 Representação Espacial

A representação espacial nada mais é do que a representação do conjunto suporte da imagem, que foi definido anteriormente e denominado como conjunto  $U$ . A partir de agora, considere o conjunto suporte da imagem (domínio da função) como sendo um retângulo semelhante ao da Figura 2.9.

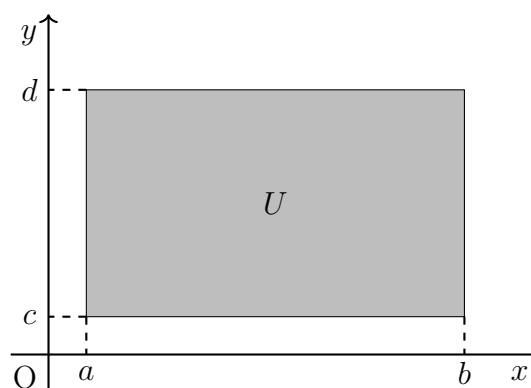


Figura 2.9: Conjunto suporte da imagem.

Fonte: Autoria própria (2018).

Matematicamente, pode-se representar o retângulo  $U$  como

$$U = [a, b] \times [c, d] = \{(x, y) \in \mathbb{R}^2 / a \leq x \leq b \text{ e } c \leq y \leq d\}.$$

A *amostragem matricial uniforme* é o método mais utilizado para fazer a discretização espacial de uma imagem. Com esse método, a amostragem do conjunto  $U$  normalmente é baseada em um reticulado uniforme ilustrado na Figura 2.10 e é definido como

$$F_{\Delta} = \{(x_i, y_j) \in \mathbb{R}^2 / 0 \leq i \leq m, 0 \leq j \leq n\},$$

onde os pontos  $(x_i, y_j)$  são chamados de *pontos amostrais* e as equações (2.1) e (2.2) são válidas.

$$x_i = a + i\Delta x \quad ; \quad \Delta x = (b - a)/m \quad ; \quad i = 0, \dots, m. \quad (2.1)$$

$$y_j = c + j\Delta y \quad ; \quad \Delta y = (d - c)/n \quad ; \quad j = 0, \dots, n. \quad (2.2)$$

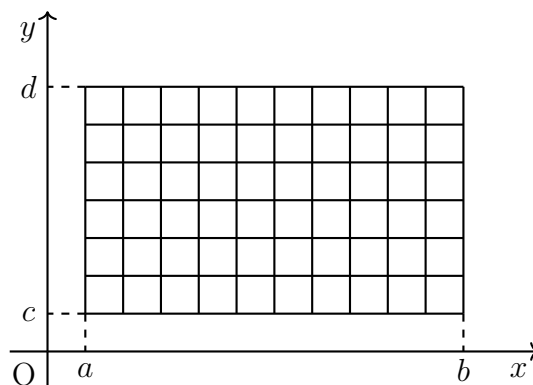


Figura 2.10: Reticulado de discretização  $F_{\Delta}$ .

Fonte: Adaptada de [36].

As variáveis  $m$  e  $n$  indicam as dimensões (largura e altura) do reticulado e estão comentadas no decorrer do texto.

A amostragem é a discretização espacial de  $f(x, y)$  nas direções  $x$  e  $y$ , que gera uma matriz de amostras, ou seja, é uma divisão do plano  $xy$  em uma grade, onde  $x$  e  $y$  passam a ser números inteiros não negativos. E para definir o processo de amostragem é preciso selecionar uma quantidade finita de pontos de uma imagem com infinitos pontos. Isso significa que é possível selecionar pequenos conjuntos de espaços que se tornam uma imagem discreta. Esse processo está diretamente ligado com a quantidade de informação que se deseja guardar. Quanto maior a amostragem, mais detalhes podem ser encontrados e, conseqüentemente, maior é o espaço necessário para o armazenamento.

Considere agora uma *imagem discreta* sendo uma matriz bidimensional, denotada por  $A$ , formada por  $m$  linhas e  $n$  colunas, sendo que cada elemento de imagem discreta  $a_{ij} = f(x_i, y_j)$  (com  $i = 0, \dots, m$  e  $j = 0, \dots, n$ ) é chamado de *elemento de imagem*,

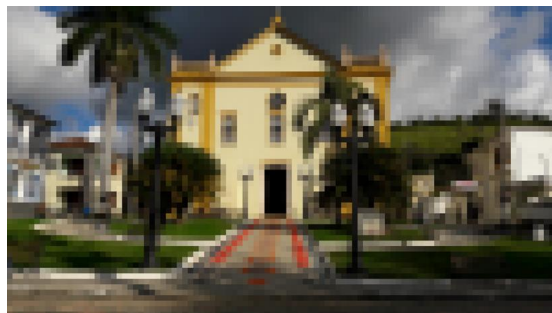
*elemento pictórico, pels* ou *pixel*. A título de curiosidade, *Pixel* é uma abreviação da palavra inglesa “picture element” [7]. Como já foi visto, esse é o termo mais comum para se referir a cada elemento de uma imagem.

Uma imagem formada por um conjunto de pixel é chamada de *imagem digital*. Esta pode ser considerada uma outra definição para imagem digital.

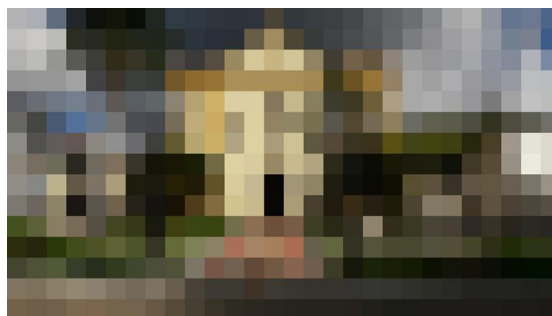
A resolução de uma imagem digital pode ser descrita pelo número de pixels apresentados na tela de um monitor e, por sua vez, também descreve o nível de detalhe que uma imagem comporta. Cada pixel é uma unidade básica da imagem e contém informações sobre cores e tonalidades [7].

A quantidade de linhas  $m$  da matriz de pixels  $A$  é chamada de *resolução vertical* da imagem. O número de colunas  $n$  é a *resolução horizontal*. Logo, a *resolução espacial*, ou *resolução geométrica*, da imagem é dada pelo número de amostras  $m \times n$ .

O exemplo apresentado na Figura 2.11 foi baseado nas teorias encontradas em [36] e faz-se possível observar uma mesma imagem com resoluções espaciais diferentes. Na Figura 2.11b o tamanho da célula de cada pixel é bem maior do que o tamanho das células na Figura 2.11a.



(a) Imagem com maior resolução.



(b) Imagem com menor resolução.

Figura 2.11: Imagens de uma igreja com resoluções espaciais diferentes.

Fonte: Autoria própria (2017).

### 2.1.3.3 Representação de Cor

A representação de cor é a representação do espaço de cor da função imagem  $f$ . Em outras palavras, é a representação do gamute de cores da imagem, que foi definido anteriormente como o conjunto  $f(U)$ .

O processo de discretização de cor é chamado de *quantização*. Esse processo faz com que cada pixel  $f(x, y)$  assumira um valor inteiro não negativo de intensidade luminosa (nível de cinza). Isto significa que a quantização de uma imagem é o processo que reduz o gamute de cores  $f(U)$  e faz com que a imagem seja exibida com um menor número de cores. Inicialmente, uma imagem contém um conjunto contínuo de cores. Após a aplicação do processo de quantização a imagem terá um conjunto discreto de cores.

Formalmente, a quantização é uma transformação dada por

$$q : \mathbb{R}^n \mapsto M_k, \quad (2.3)$$

onde  $M_k = \{p_1, p_2, \dots, p_k\} \subset \mathbb{R}^n$  e cada elemento  $p_i \in M_k$  é chamado de *nível de quantização*. Note que  $M_k$  é um subconjunto finito de  $\mathbb{R}^n$  e pode ser chamado de *mapa de cor* ou *codebook* da transformação de quantização. Se  $k = 2^m$ , então é possível codificar o conjunto  $M_k$  no computador utilizando  $m$  bits. Desse modo, a transformação (2.3) é chamada de quantização de  $m$  bits [37].

Também é possível ter uma quantização entre subconjuntos finitos de cor, contidos em  $\mathbb{R}^n$ . Considere, agora, a transformação

$$\tilde{q} : M_j \mapsto M_k. \quad (2.4)$$

onde  $M_j$  é construído do mesmo modo que  $M_k$ . Suponha que  $j > k$ ,  $j = 2^n$  e  $k = 2^m$ . Pode-se dizer que a transformação (2.4) é uma quantização de  $n$  para  $m$  bits. Portanto, o processo de quantização altera a resolução de cor da imagem.

As imagens contidas no Exemplo 2, a seguir, foram desenvolvidas com o auxílio do PhotoScape <sup>7</sup>, Picasa <sup>8</sup> e PhotoFiltre <sup>9</sup> apenas para exemplificar e ilustrar para o leitor, de maneira didática, a ideia de quantização e amostragem.

<sup>7</sup><https://photoscape.br.uptodown.com/windows>

<sup>8</sup><https://picasa.google.com.br/>

<sup>9</sup><http://www.photofiltre-studio.com/download-en.htm>

**Exemplo 2** (Amostragem e quantização de uma imagem). Ao olhar a Figura 2.12 é possível contemplar um carro passando por uma poça d'água em uma estrada estreita de serra. Suponha que nessa figura existam infinitos pontos.



Figura 2.12: Figura original do carro.  
Fonte: Autoria própria (2016).

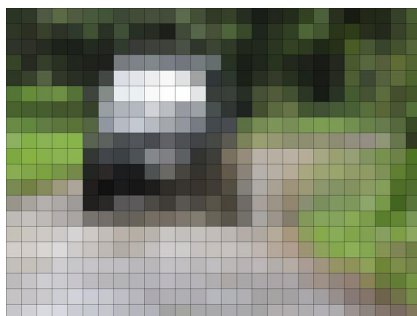
Primeiro é preciso definir uma grade de amostragem, conforme o exemplo indicado pela Figura 2.13a. Essa grade é aplicada sobre a Figura 2.12 de forma que cada célula da grade contenha uma sub-imagem, segundo observa-se na Figura 2.13b. Dentro das sub-imagens, é necessário escolher uma única cor predominante, como na Figura 2.13c. Por fim, retira-se a grade de amostragem e o resultado é apresentado na Figura 2.13d.



(a) Grade de amostragem  $20 \times 27$ .



(b) Grade aplicada na figura original.



(c) Imagem quantizada ainda com a grade de amostragem.



(d) Imagem final após os processos de amostragem e quantização.

Figura 2.13: Processo de amostragem e quantização de uma imagem.  
Fonte: Autoria própria (2016).

### 2.1.4 Cores e Modelo RGB

Atualmente, as cores são vistas em aparelhos digitais, tais como a TV, a tela de um celular, o monitor de um computador e uma câmera fotográfica. Estes aparelhos geram a cor através de luz. Essa formação de cores é composta pelas três cores primárias: vermelho, verde e azul que no inglês são red, green e blue que formam a sigla RGB para esse sistema de cores.

O motivo do uso dessas três cores primárias se deve ao fato de que os olhos humanos possuem três tipos de sensores coloridos diferentes que denominam-se *fotopigmentos* [7]. De acordo com [3], as cores mudam sob diferentes condições de iluminação, ou seja, não existe uma cor fixa. Alguns objetos, como o sol e uma brasa, irradiam luz e se dizem luminosos. Mas, a maioria dos objetos que podem ser vistos não são luminosos. Os olhos humanos conseguem ver estes objetos porque a luz de alguma outra fonte refletiu-se neles.

Segundo [7], os sistemas de cores podem ser aditivos ou subtrativos. No processo aditivos de cores, ilustrado na Figura 2.14, o preto simboliza a ausência de cores e o branco é gerado pela soma de todas elas. Este processo é utilizado nos monitores de vídeo e televisão, no qual a cor é gerada por comprimento de ondas luminosas. Logo, é com este processo que optou-se lidar do decorrer do trabalho.

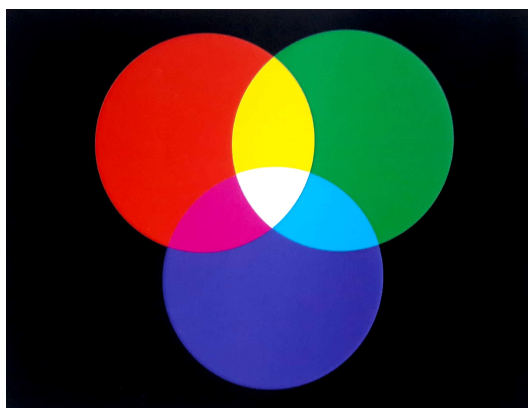


Figura 2.14: Processo aditivo das cores primárias.

Fonte: [7].

Além das cores primárias, a Figura 2.14 contém outras cores resultantes da soma entre elas:

- Amarelo = vermelho + verde;
- Magenta = vermelho + azul;
- Ciano = azul + verde;
- Branco = vermelho + azul + verde.

Um pixel é formado pelos níveis de intensidade das três cores primárias em conjunto. Ele pode variar entre todas elas, desde o preto até o branco. Matematicamente, a representação da cor de cada pixel de uma imagem colorida pode ser obtida pela expressão:

$$C = \tilde{r} \cdot R + \tilde{g} \cdot G + \tilde{b} \cdot B, \quad (2.5)$$

onde  $\tilde{r}$ ,  $\tilde{g}$  e  $\tilde{b}$  são os coeficientes de mistura correspondentes as intensidades de R, de G e de B, separadamente. Desse modo, a cor  $C$  de cada pixel da imagem pode ser representada como um ponto no espaço tridimensional por meio da tripla coordenada  $(\tilde{r}, \tilde{g}, \tilde{b})$  [3, 7].

Já o processo de subtração de cores, que pode ser visto na Figura 2.15, é utilizado nas impressoras e pinturas. Neste processo acontece o inverso: o branco é a ausência de cores e o preto indica a presença de todas cores juntas.

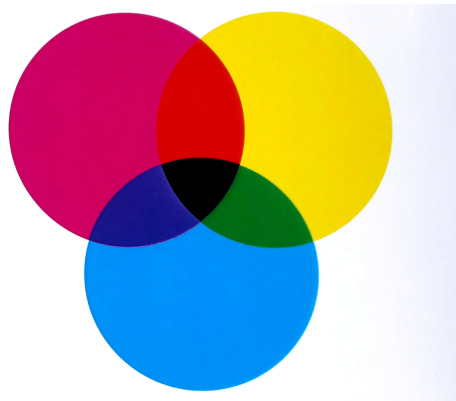


Figura 2.15: Processo de subtração de cores secundárias.  
Fonte: [7].

O propósito desta seção não envolve discutir detalhes sobre o processo de subtração. Basta saber que operações desse tipo podem ser feitas com as cores. Para mais informações sugere-se a referência [7].

Ao olhar a Figura 2.16, na próxima página, pode-se observar um cubo unitário, onde sua diagonal principal vai do preto ao branco e possui quantidades iguais de cores primárias que representa a escala cinza. Cada ponto interno ao cubo pode ser representado pela tripla  $(\tilde{r}, \tilde{g}, \tilde{b})$ , onde as coordenadas estão presentes na equação (2.5).

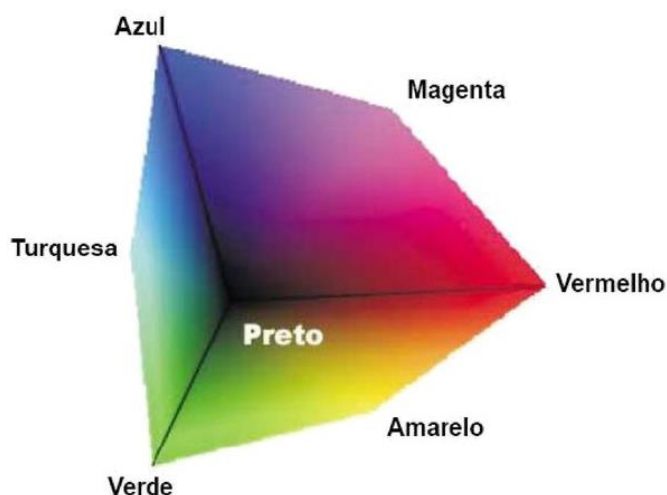


Figura 2.16: Modelo RGB a partir dos eixos XYZ.  
Fonte: [7].

O RGB é um espaço contínuo e, por conseguinte, em trabalhos computacionais procura-se aplicar o processo de discretização (amostragem e quantização) nas imagens, como visto nas seções anteriores. Neste trabalho, pretende-se trabalhar com imagens discretizadas em 256 níveis de cores para R, 256 níveis para G e mais 256 níveis para B. Totalizando, assim, aproximadamente 16,7 milhões de cores distintas.

Deve-se ressaltar que existem diversos sistemas de cores que não foram apresentados aqui, pois foge do foco do trabalho. O sistema escolhido para implementação das técnicas contidas no código da ferramenta computacional deste trabalho é o sistema de cores RGB. Esta escolha se deve ao uso da biblioteca OpenCV, de processamento de imagens, utilizada no código computacional desenvolvido neste trabalho.

O sistema RGB foi abordado novamente no capítulo 4 para relacionar o desenvolvimento do código computacional com tal sistema.

Cores é um assunto que envolve o mecanismo da visão colorida, sensação de cores e muitos outros detalhes [7] que fogem do foco deste trabalho. Devido a isto, esta seção apenas fornece uma breve introdução sobre cores e o modelo RGB e suas operações, que foram utilizadas na implementação computacional das técnicas de obtenção de planos adjacentes. Na computação gráfica, as cores e diversos outros aspectos são amplamente necessários e utilizados, sendo assim, um vasto campo de pesquisas. Como leitura complementar, os autores de [7] sugerem as referências [56] e [78].

## 2.2 Seccionamento em Série

Microestruturas <sup>10</sup> de materiais geralmente são de natureza tridimensional e, portanto, a visualização de microestruturas tridimensionais (3D) é de interesse primário. Uma seção metalográfica bidimensional (2D) não contém todas as informações relativas à verdadeira geometria microestrutural 3D, já que em um plano metalográfico, os volumes <sup>11</sup> aparecem como áreas, e as superfícies <sup>12</sup> aparecem como linhas [63]. Sendo assim, observações de seções bidimensionais nos permitem fazer certas suposições sobre as formas das características 3D que estão enterradas dentro de um material. No entanto, os tamanhos, distribuições espaciais, formas e interconectividade de características microestruturais complexas só podem ser caracterizadas com análises 3D [47] apud [26].

Destarte, nessa seção foi apresentado um método aplicado a reconstrução de microestruturas tridimensionais chamado *Seccionamento em Série* ou *Serial Sectioning* (em inglês), que foi estabelecido para gerar dados microestruturais através da repetição controlada de cortes bidimensionais [64]. Conforme os autores de [47], a palavra “serial” equivale a ideia de corte repetido. Este método tem se desenvolvido nos últimos anos e muitos novos e excelentes resultados foram relatados em um campo de análise de microestruturas de materiais metálicos [41] apud [2, 14]. Um dos problemas comentados por [47] era que, até a década de 80, não havia software de computador disponível para obter medições em 3D ou para exibir imagens 3D com precisão. Na maioria dos casos, os resultados 3D foram representados por esboços desenhados à mão, gráficos de comprimento  $\times$  profundidade ou filmes. Mas os autores de [47] também previram o rápido desenvolvimento de software e hardware de computador que ocorreu nos últimos anos. Desde então, os recursos de processamento de imagens e visualização 3D melhoraram a um ponto em que armazenar e representar imagens 3D não é mais uma limitação severa. Na pesquisa de [64] revelou-se que devido a melhorias rápidas e contínuas em hardware e software para manipular e analisar dados microestruturais em 3D, houve uma forte revitalização da técnica na comunidade de ciência de materiais.

O presente trabalho se inicia, de fato, após a execução do processo de seccionamento em série sobre uma amostra real de microestrutura, visto que os resultados referentes ao método são utilizados como dados de entrada para a ferramenta computacional desenvolvida. Por isso, esta seção tem por objetivo explicar o processo de reconstrução 3D via seccionamento em série expondo detalhes deste processo que foi empregado e validado

---

<sup>10</sup>Estrutura detalhada de um material, obtida através do microscópio.

<sup>11</sup>Por exemplo: grãos, vazios e partículas.

<sup>12</sup>Por exemplo: limites de grão e interfaces precipitadas.

através de pesquisas anteriores, como em [10, 12, 13, 15, 16, 19, 39, 41, 47, 51, 60, 73, 75, 76]. No próximo capítulo, foi citado o que alguns autores propuseram com suas pesquisas e foram apresentadas, brevemente, diferentes técnicas utilizadas por eles.

“A técnica de seccionamento em série desenvolvida em 1970 possibilita reconstruir um volume da microestrutura em 3D a partir de seções planas da mesma.” [10].

Geralmente, a técnica mencionada por [10] é aplicada em diversos materiais, tais como Aço, Ferro Fundido, Alumínio e outros. No entanto, nada impede que a técnica seja aplicada a outros tipos de materiais e/ou objetos de interesse.

O processo da técnica de seccionamento serial consiste em remover camadas sequenciais de um determinado material. A cada camada removida mediante a lixamento e/ou polimento metalográfico da superfície da amostra é necessário corrigir a amostra e fotografar sua superfície usando microscopia óptica [63, 65, 71]. Através desse processo, obtém-se imagens 2D de cada camada, que são chamadas de *seções planas*, *seções bidimensionais* ou *fatias em série* da microestrutura. Após a obtenção das imagens ocorre o empilhamento das mesmas, conforme foi ilustrado na Figura 2.17, que são utilizadas, posteriormente, para fazer a reconstrução tridimensional.

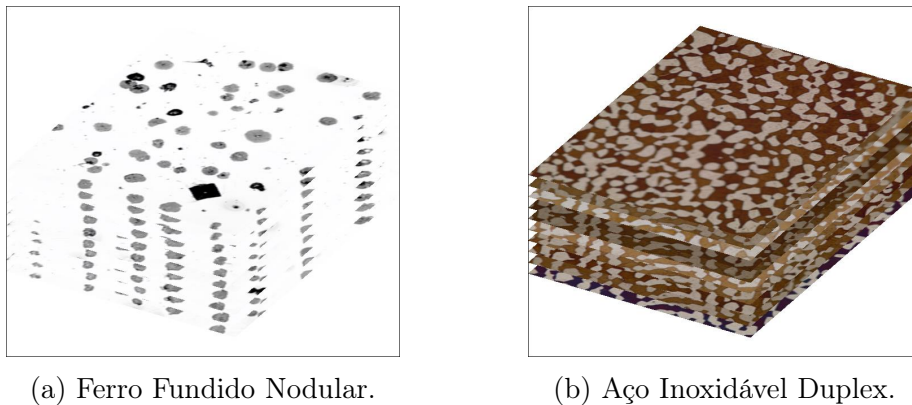


Figura 2.17: Exemplo de duas pilhas de 10 seções em série.  
Fonte: [23].

Segundo [63], uma pilha de seções seriais alinhadas, essencialmente, constituem um conjunto de dados de imagens de volume semelhantes aos encontrados na tomografia computadorizada de raios X e nas imagens de ressonância magnética (MRI).

O processo utilizado para obter as seções bidimensionais de materiais metálicos neste trabalho foi o processo metalográfico convencional, baseado em trabalho manual. Essa abordagem convencional para seccionamento serial é extremamente demorada e requer-se

um esforço considerável para fazer a preparação da amostra. Em detalhes, esse processo é realizado pela seguinte sequência de passos básicos:

1. A preparação da amostra é feita através de ataque químico adequado ao que se deseja analisar na microestrutura. O objetivo do ataque é obter um bom contraste com relação as fases presentes no material metálico de modo que as fases necessitam estar bem distintas em relação à matriz;
2. Escolhe-se qual a região da amostra que será analisada;
3. Microimpressões Vickers são feitas sobre a região de interesse escolhida no segundo passo. Elas auxiliam no alinhamento das seções e servem para identificar a exata região que foi selecionada após sucessivos lixamentos e/ou polimentos. Além disso, as Microimpressões Vickers também possibilitam medir precisamente a distância entre as seções consecutivas logo após cada lixamento metalográfico, pois o tamanho da microimpressão diminui conforme o desbastamento ocorre [47, 63, 71, 80, 81];
4. Somente neste passo do processo adquire-se a primeira imagem de uma das seções bidimensionais em série. E, para isto, com o auxílio do microscópio óptico, ocorre uma tomada de foto da região da amostra que foi escolhida para a análise no segundo passo.
5. Aplicam-se dois processos laboratoriais sobre a microestrutura em uma determinada direção. O primeiro é o *lixamento*, na qual utiliza-se uma lixa grossa para retirar uma camada da amostra. O segundo processo é o *polimento*, onde é usado um pano fino com o intuito de proporcionar brilho à superfície, caso seja necessário. Esses dois processos podem ser feitos com auxílio de uma *Politriz Metalográfica* ou *Polidora Metalográfica* (Figura 2.18) que é um equipamento que faz o desbastamento ou polimento de diversos tipos de amostras com precisão e regularidade. Esse é considerado o passo mais crítico da técnica de seccionamento em série, pois existem alguns fatores que interferem na qualidade final da superfície e na profundidade de seccionamento. Ressalta-se que, dependendo da estrutura do material, aplicar apenas o polimento metalográfico é suficiente para este passo. Porém, alguns materiais passam apenas pelo processo de lixamento. Isso acontece porque cada material tem suas limitações e, com isso, a preparação laboratorial é diferente para cada caso. Conforme observado por [47] apud [26], cada novo sistema de liga requer diferentes técnicas de seccionamento, especialmente se os objetivos do estudo forem diferentes.



(a)



(b)

Figura 2.18: (a) Amostra posicionada na Politriz Automática juntamente com o painel de controle. (b) Processo de polimento em andamento.

Fonte: [54].

Por fim, o polimento e/ou lixamento metalográfico (descrito no passo 4) e o processo de tomada de foto (descrito no passo 5) se repetem várias vezes até obter um número de seções suficientemente bom para reconstruir o objeto em estudo [24, 49, 71, 81].

Na próxima página, veja um fluxograma ilustrado na Figura 2.19, que foi feito com o propósito de generalizar e ilustrar o passo a passo descrito.

O objetivo desse fluxograma é mostrar, de forma descomplicada, o fluxo das informações e elementos evidenciando a sequência operacional que caracteriza o trabalho que está sendo executado [25]. Neste caso, criou-se uma estrutura visual que ilustra uma sequência de operações com a finalidade de simplificar, compreender e enxergar melhor o processo de seccionamento serial. O fluxograma foi construído através do site *Draw.io* (disponível em [43]), conforme a Figura 2.19.

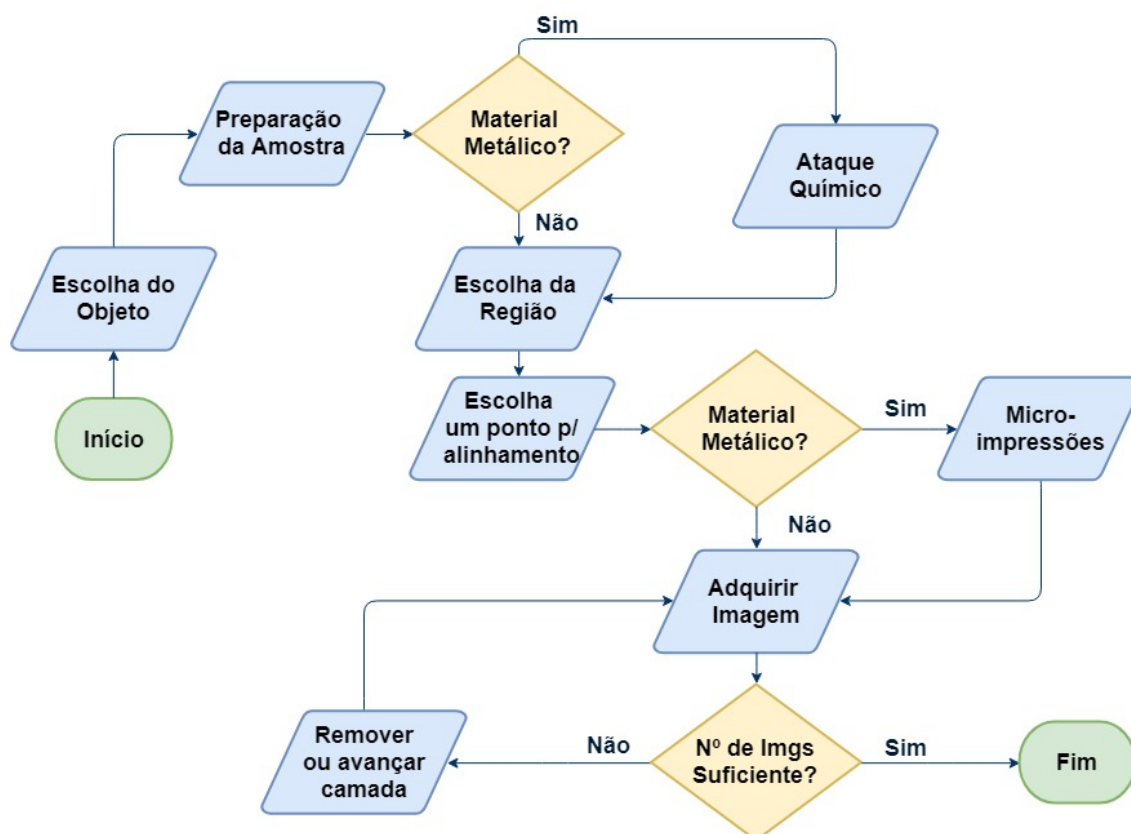


Figura 2.19: Fluxograma do processo de seccionamento em série.

Fonte: Autoria própria (2019).

Ao final da aplicação deste método obtém-se uma quantidade satisfatória de seções planas (ou imagens em série). A cada passo de remoção de pequenas camadas da amostra ou apenas avanço de camada, a próxima imagem a ser obtida é uma fotografia do plano que está localizado exatamente em baixo do plano anterior e assim sucessivamente.

Entretanto, de acordo com [10] encontram-se algumas dificuldades na aplicação desta técnica. Como exemplo, pode-se citar a medida ideal da distância entre as séries de seções que se obtém como resultado da aplicação deste método. Quando a distância entre cada duas seções é considerada muito grande, ocorre uma perda de informação para reconstrução do material escolhido. Porém, quando ela é relativamente pequena, o trabalho experimental, a ser feito, tende a aumentar. Os autores de [10] também escreveram:

*“A parte experimental da reconstrução requer várias tentativas até atingir as melhores condições de polimento, ocasionando perda de tempo e material. Com a simulação de todas as partes do processo de reconstrução 3D, será possível prever, por exemplo, a quantidade exata de seções que devem ser retiradas da amostra em estudo, evitando testes experimentais aleatórios.”* [10].

Por isso, um dos fatores relevantes a ser considerado na técnica de seccionamento em série é a espessura a ser retirada da amostra. Como o desbastamento da amostra real gera grandes perdas de informações, procura-se aplicar métodos computacionais para determinar a microestrutura entre dois planos discretos. Busca-se encontrar os planos intermediários que representem de modo fidedigno as informações que foram perdidas no processo de desbaste da microestrutura. Também é preciso ressaltar que as fotografias que são obtidas por meio de microscopia óptica necessitam de alinhamento para que, na hora da reconstrução, os objetos de interesse possam ser vistos e analisados corretamente e com maior clareza [33, 51].

Outro fator importante é o tratamento das imagens (aquisição, arquivamento, manejo, etc), que este trabalho tenta aprimorar, uma vez que falhas neste processo podem causar perda de dados da microestrutura. As fotos capturadas são salvas em arquivos digitais de diferentes formatos. Estes arquivos são portadores de algumas informações, tais como: tamanho da imagem, data, densidade de pixels, níveis de vermelho, verde e azul em cada pixel e etc. A organização visual dos elementos é responsável por posicionar cada pixel em seu devido lugar. Deste modo, a representação geométrica da cena é mantida a rigor, e formas geométricas complexas podem ser representadas com maior fidelidade. Cada fotografia é transformada em uma matriz bidimensional e essas matrizes são sobrepostas formando uma matriz tridimensional.

A estrutura interna do material, geralmente referida como microestrutura, é conhecida por desempenhar um papel importante no controle das propriedades e características de desempenho do material [74]. De acordo com os autores de [74], anos recentes testemunharam avanços que envolveram o uso de seccionamento em série automatizado combinado com o escaneamento de superfície 2D usando técnicas de microscopia adequadas que podem ser encontradas em [45, 64] ou raios X de alta energia combinados com tomografia computadorizada [57, 77].

Existem alguns métodos mais elaborados de seccionamento em série para investigação e representação 3D da estrutura interna de materiais. No próximo capítulo, foram descritos alguns desses métodos, como por exemplo o FIB, cuja sigla significa *Focused Ion Beam* (do inglês) ou Feixe de Íons Focalizado, que fornece resultados desbastando a amostra do material com um feixe de íons focalizado [82]. Muitas dessas técnicas de caracterização de estruturas de materiais 3D mencionadas requerem o uso de equipamentos altamente especializados e incorrem em alto custo e esforço.

# Capítulo 3

## Revisão Bibliográfica

### 3.1 Estado da Arte

Ao olhar a Figura 3.1, construída através do site *Draw.io* [43], é possível ver uma linha do tempo que contém alguns artigos, de outros autores, relacionados ao tema em questão. Esses e outros trabalhos foram relevantes para a pesquisa atual. Os trabalhos que não estão inclusos na linha do tempo, já foram introduzidos no decorrer do texto para complementar esta pesquisa.

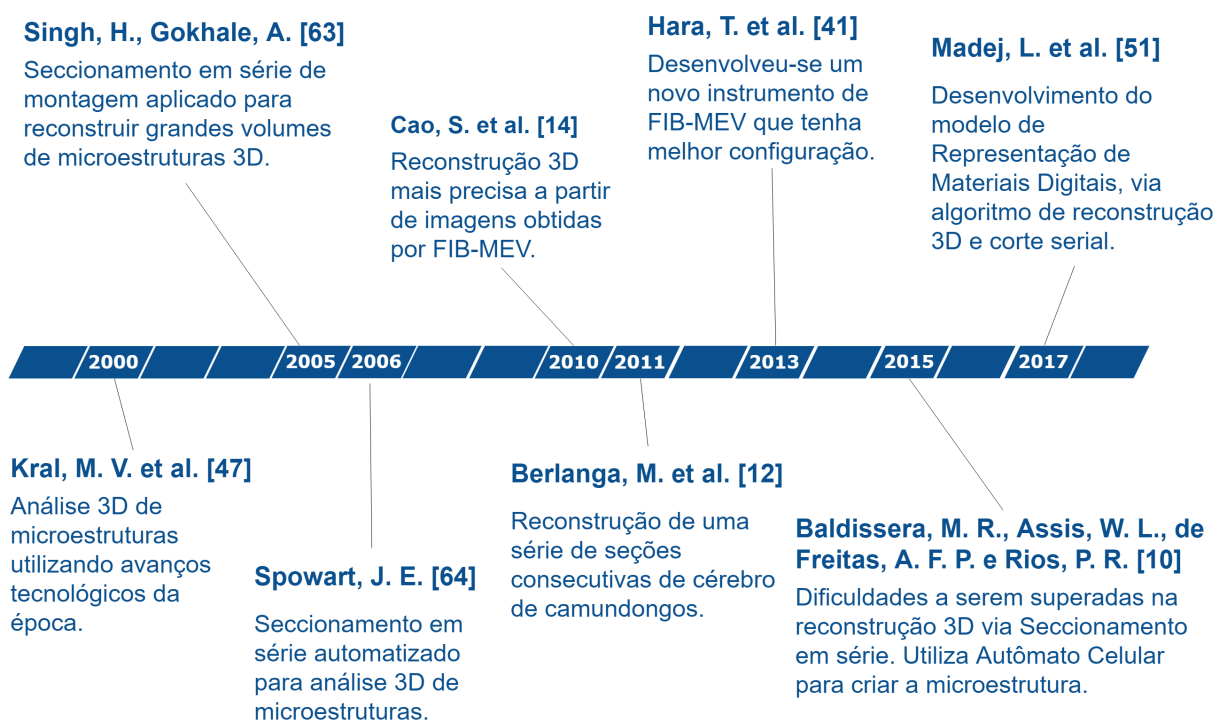


Figura 3.1: Linha do tempo de alguns artigos relacionados ao tema.

Fonte: Autoria própria (2019).

No mês de outubro do ano de 1999, M. V. Kral, M. A. Mangan, G. Spanos e R. O. Rosenberg enviaram um artigo intitulado “*Three-dimensional analysis of microstructures*” [47] para a revista *Materials Characterization*. O artigo foi aceito em janeiro de 2000 e trata de uma abordagem para a análise 3D de microestruturas que é mostrada em três diferentes ligas de aço. Os autores descrevem várias técnicas experimentais, bem como resultados recentes da época e avanços na reconstrução e visualização assistida por computador. Aproveitou-se dos avanços tecnológicos de software e hardware para descrever as melhorias na análise tridimensional de microestruturas. Foram feitos recortes de microdureza com a intenção de marcar a área de interesse, servir como marcas de referência para o alinhamento posterior da imagem e para calibrar a profundidade da remoção do material. O uso de dois softwares foi indispensável: o *PhotoShop* foi muito útil para suas funções de edição e o *NIH Image* foi indispensável para manipulações de “pilhas” de imagens. A transformação da pilha de imagens digitais individuais foi transformada em imagens 3D por meio do software de visualização AVS versão 5.3. Grãos inteiros, bem como precipitados individuais, foram reconstruídos por corte e edição de imagens para conter apenas as áreas de interesse. Mais detalhes dessas técnicas podem ser encontrados em [46].

Em setembro de 2004, Harpreet Singh e Arun M. Gokhale enviaram um artigo com o título “*Visualization of three-dimensional microstructures*” [63] para a revista *Materials Characterization*. O artigo foi aceito em outubro de 2004 e diz que

“*A visualização, caracterização e representação de microestruturas tridimensionais (3D) são de interesse significativo para o entendimento e modelagem de relações entre processamento e propriedade da microestrutura.*” [63].

Os autores de [63] utilizaram uma técnica eficiente de corte em série de montagem que foi aplicada para a reconstrução de microestruturas tridimensionais de grande volume. No ano em que o artigo foi publicado, observou-se que a técnica de seccionamento serial clássica tinha sido utilizada em muitos estudos de microestruturas 3D de materiais opacos. Foi relatado que pode-se utilizar essa técnica clássica de seccionamento em série, desenvolvida na década de 1970 [58], para um volume microestrutural relativamente pequeno, pois é bastante útil para visualização de partículas 3D, características morfológicas e detalhes microestruturais de curto alcance em resolução suficientemente alta. No entanto, o seccionamento serial clássico não é útil para a caracterização quantitativa de atributos topológicos e também não é útil para uma estimativa verdadeiramente não viesada de distribuições de tamanho de grão e/ou

partícula 3D (veja mais detalhes em [70, 72]). Além disso, os autores também disseram que a técnica clássica de seccionamento serial não pode ser usada para reconstruir um grande volume de microestrutura 3D em resolução suficientemente alta.

Segundo [54] apud [48] apud [63, 70] é possível gerar um grande volume microestrutural 3D com alta resolução. Para isto, reconstrói-se, primeiro, um pequeno volume de microestrutura 3D, como indicado na Figura 3.2 (a). Em seguida, basta reconstruir pequenos volumes microestruturais que estão na vizinhança do pequeno volume citado inicialmente de modo que sejam reunidos para gerar o grande volume microestrutural como na Figura 3.2 (b). Portanto, pode-se definir uma montagem como sendo uma imagem microestrutural de uma grande área com alta resolução [48].

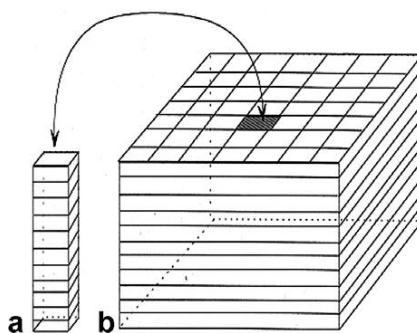


Figura 3.2: (a) Pequeno volume microestrutural construído por seções bidimensionais resultantes da técnica de seccionamento em série. (b) Grande volume microestrutural obtido por pequenos volumes ao redor do volume apresentado em (a).

Fonte: [48] apud [63].

Nos trabalhos de [48] e [63] as seções bidimensionais resultantes da técnica do seccionamento em série foram alinhadas com o auxílio do software de análise de imagens 3D VoxBlast 3.10, no qual as imagens foram giradas digitalmente até que fossem alinhadas exatamente com as respectivas seções anteriores. A estratégia empregada foi localizar três pontos comuns a cada duas seções consecutivas em série. Uma das imagens é posta alinhada com o primeiro ponto. Em seguida, a imagem é rotacionada em torno deste ponto até que o segundo ponto comum também esteja alinhado. Já o terceiro ponto é utilizado para inclinação da imagem. Com essa estratégia pode-se provar a unicidade de um plano que, neste caso, é uma seção bidimensional. No trabalho de [63] foram utilizados recuos de microdureza para o propósito de facilitar o alinhamento.

Em 31 de outubro de 2005, Jonathan E. Spowart enviou um artigo para revista *Scripta Materialia*. O título do artigo é “*Automated serial sectioning for 3-D analysis of microstructures*” [64] e foi aceito em 10 de janeiro de 2006. O autor descreve um técnica

de seccionamento em série automatizado e diz que é uma técnica flexível e de baixo custo para obter rapidamente dados microestruturais de uma ampla variedade de materiais. Ele também cita algumas vantagens sobre a técnica proposta comparado com métodos manuais e apresenta exemplos relevantes identificando as principais áreas tecnológicas para o desenvolvimento e aprimoramento da técnica de seccionamento automatizada. De acordo com [64] apud [4, 65], a manipulação mecânica automática da amostra durante cada uma das etapas do processo (incluindo o posicionamento da amostra no microscópio para geração de imagens) diminui drasticamente a variabilidade entre as seções, comparada com a técnica manual. Uma das técnicas automáticas que o autor mostra é o *Robo-Met.3D*, que é exibido na Figura 3.3. Ele diz que o robô é totalmente automatizado: “É um dispositivo de seccionamento em série que usa polimento metalográfico para obter a remoção controlada da superfície da amostra”.



Figura 3.3: Principais componentes do *Robo-Met.3D*: polidor automático (esquerda), braço do robô (centro), estação de gravação automática (centro inferior) e microscópio invertido motorizado (direita).

Fonte: [64].

O autor também faz uma comparação do *Robo-Met.3D* com outras técnicas disponíveis na literatura: *3-D Atom probe tomography*, *Focused ion beam (FIB/ SEM)*, *X-ray diffraction Microscopy*, *X-ray computed micro-tomography* e *Micro-miller*.

No mês de setembro de 2009, os autores Shanshan Cao, Christoph Somsen, Mihail Croitoru, Dominique Schryvers e Gunther Eggelerb enviaram um artigo intitulado “*Focused ion beam/scanning electron microscopy tomography and conventional transmission electron microscopy assessment of Ni<sub>4</sub>Ti<sub>3</sub> morphology in compression-aged Ni-rich Ni-Ti single crystals*” [14] para revista *Scripta Materialia*. Este artigo foi revisado em 25 de novembro de 2009 e aceito em 27 de novembro de 2009. Trata-se de um estudo

que fornece uma descrição mais precisa dos tamanhos, formas e arranjos espaciais de precipitados. Os autores de [14] investigaram o tamanho, a morfologia e a configuração dos precipitados de  $\text{Ni}_4\text{Ti}_3$  em uma liga Ni-Ti de monocristal por meio de análise de imagens bidimensionais baseadas em microscopia eletrônica de transmissão e reconstrução tridimensional a partir de imagens obtidas em um feixe de íons focalizado juntamente com sistema de feixe duplo de microscopia eletrônica de varredura (FIB-MEV) com o objetivo de fornecer uma descrição mais precisa de tais características. O seccionamento serial através de microscópio eletrônico de varredura de feixe de íon com foco de feixe duplo (FIB-MEV) foi capaz de executar de forma mais eficiente e precisa a documentação dos vários planos metalográficos <sup>1</sup>, que neste caso, são as seções bidimensionais obtidas através da técnica de seccionamento em série. Além disso, o FIB-MEV permite obter imagens de alta resolução [15].

Foi publicado, no ano de 2011, um artigo intitulado “*Three-dimensional reconstruction of serial mouse brain sections: solution for flattening high-resolution large-scale mosaics*” [12] cujos autores são Monica L. Berlanga, Sébastien Phan, Eric A. Bushong, Stephanie Wu, Ohkyung Kwon, Binh S. Phung, Steve Lamont, Masako Terada, Tolga Tasdizen, Maryann E. Martone e Mark H. Ellisman. Neste artigo, publicado pela revista *Frontiers in Neuroanatomy*, os autores descrevem cuidadosamente as etapas usadas para reconstruir digitalmente uma série de seções consecutivas do cérebro de camundongos marcadas com três manchas, uma mancha para vasos sanguíneos, uma coloração nuclear e uma mancha de mielina. O interessante é que estas manchas marcam marcos neuroanatômicos importantes que são usados para empilhar as seções em série durante a reconstrução. Além disso, os autores também mostraram que o uso de dois softwares (*ir-Tweak* e *Mogrifier*) permitem que os cientistas trabalhem com volumes digitalizados apesar das variações de espessura e rupturas nas seções bidimensionais. Este artigo se concentra na montagem de fatias em série de tecido cerebral de camundongo, seccionadas em um vibratome (instrumento usado para cortar fatias finas de material) e fotografadas usando microscopia de laser 3D em mosaico em larga escala. Para esta experiência, utilizou-se três ratinhos macho, cada um com aproximadamente 2 meses de idade. Todas as seções foram achatadas e depois registradas entre si usando ferramentas de software já desenvolvidas e disponíveis gratuitamente. Para mais detalhes veja [12].

Submetido em outubro de 2011 e aceito em fevereiro de 2012 pela revista *Journal of Alloys and Compounds*, o artigo cujo título é “*Application of orthogonally arranged*

---

<sup>1</sup>Metalografia é a área que estuda as características estruturais de materiais metálicos e suas ligas para relacioná-los com suas propriedades físicas, químicas e mecânicas.

*FIB-SEM for precise microstructure analysis of materials*” [41], trata-se de um novo instrumento de feixe duplo FIB-MEV que tem a melhor configuração para observação de microestrutura em termos de boa resolução espacial e alto contraste para investigar as distribuições de dois precipitados <sup>2</sup> ( $M_{23}C_6$  and MX) em uma liga de ferro, especialmente na fronteira entre as fases de martensita delta-ferrita e temperada. Os autores são Toru Hara, Koichi Tsuchiya, Kaneaki Tsuzaki, Xin Man, Tatsuya Asahata e Atsushi Uemoto. O método utilizado pelos autores se desenvolveu nos últimos anos e muitos novos resultados excelentes foram relatados em um campo de análise de microestruturas de materiais metálicos [2, 14]. O foco do artigo é melhorar a qualidade das imagens MEV através da modificação das configurações do MEV e FIB. No sistema elaborado por eles, as colunas MEV e FIB são organizadas ortogonalmente. Também é descrito algumas vantagens desse arranjo para uma observação precisa da microestrutura, e um exemplo de aplicação para aço martensítico temperado é mostrado. Na configuração proposta pelos autores, uma amostra grande não pode ser usada. No entanto, há muitas vantagens em termos de contraste, foco e resolução espacial. O tamanho máximo da amostra é um quadrado de 4 mm x 4 mm com uma espessura de 1,5 mm para a amostra global. Os instrumentos de feixe duplo FIB-MEV também fazem análises 3D pelo método de seccionamento em série. Um FIB é combinado com o MEV para permitir a obtenção de imagens de reconstrução 3D do tipo de corte em série. Nesse processo, repetições de fatiamento são executadas pelo FIB e observações são realizadas usando o MEV. A reconstrução 3D é feita empilhando as imagens dos planos fatiados que são produzidos pelo FIB [41]. A Figura 3.4 mostra uma representação conceitual do sistema ortogonal FIB-MEV proposto por [41]. Foram obtidas 260 imagens através da aplicação do método.

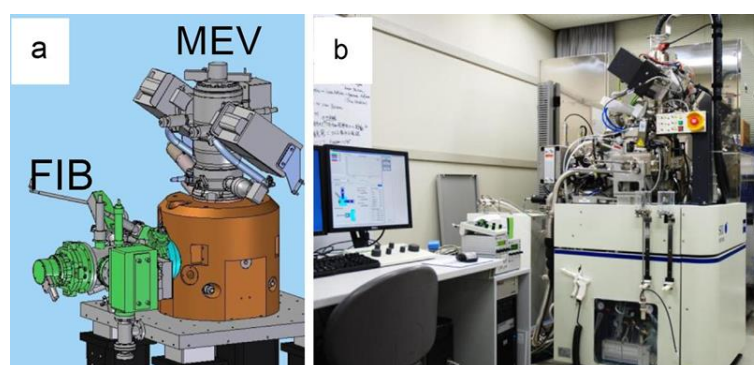


Figura 3.4: (a) Desenho do equipamento: MEV acoplado com o FIB. (b) Aparência do sistema.

Fonte: Adaptada de [41].

<sup>2</sup>Uma suspensão de pequenas partículas sólidas produzidas num líquido por reação química.

Em setembro de 2015, os autores Márcia Regina Baldissera, Wesley Luiz da Silva Assis, Artiane Fernanda Pereira de Freitas e Paulo Rangel Rios, escreveram um artigo intitulado “Simulação do processo de reconstrução em três dimensões” [10] que cita dificuldades existentes na aplicação da técnica de reconstrução tridimensional via seccionamento em série. Foi feita uma simulação de uma microestrutura tridimensional de Ferro Fundido Nodular através do método Autômato Celular com o objetivo de obter informações que auxiliem a parte experimental da reconstrução 3D. A utilização da simulação computacional fornece dados relevantes para o trabalho, como por exemplo, o tamanho adequado da espessura que é retirada do material no processo de seccionamento em série, a quantidade de seções necessárias para fazer a reconstrução do volume microestrutural, dentre outras. A reconstrução 3D foi feita com o auxílio do programa “*Reconstruct*” e utilizou-se 50 seções bidimensionais da microestrutura simulada. Este artigo já foi mencionado anteriormente neste capítulo com intuito de deixar o texto coerente, de acordo com os assuntos abordados.

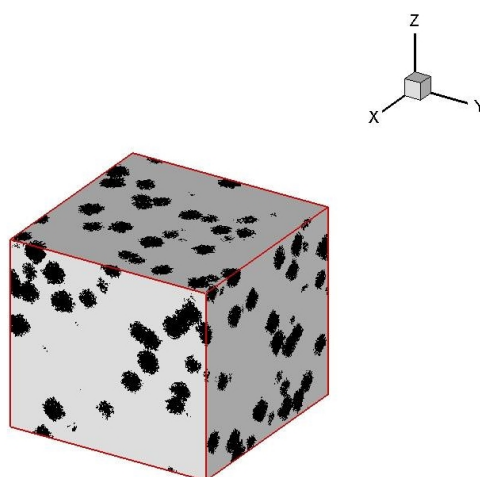


Figura 3.5: Microestrutura simulada pelo programa Autômato Celular.

Fonte: [10].

Os autores de “Simulação do processo de reconstrução em três dimensões” também identificaram que a microestrutura simulada através do programa Autômato Celular desenvolvido por eles, apresentada na Figura 3.5, ficou bastante semelhante à microestrutura real de uma amostra de Ferro Fundido Nodular. Essas e outras informações podem ser encontradas em [10].

Recentemente, no ano de 2017, os autores L. Madej, M. Mojzeszko, J. Chraponski, S. Roskosz e J. Cwajna escreveram um artigo cujo título é “*Digital material representation model of porous microstructure bases on 3D reconstruction algorithm*” [51] que foi

publicado na revista *Archives of Metallurgy and Materials*. Este artigo faz um estudo sobre materiais porosos e apresenta uma abordagem para o desenvolvimento de um modelo 3D de Representação de Materiais Digitais (DMR), onde assumi-se que cada característica importante da microestrutura é explicitamente considerada durante as simulações numéricas. Primeiramente, o artigo expõe detalhes sobre os algoritmos de seccionamento serial e processamento de imagens. O tamanho dos poros na microestrutura investigada determinou uma espessura de camada, que foi removida durante a operação de polimento simples. A área de interesse foi marcada por dois recortes de microdureza para identificar a região investigada e avaliar a espessura das camadas removidas. Com o mesmo intuito desta dissertação, o corte serial manual feito no trabalho referente ao artigo, foi realizado para fornecer dados de entrada para o algoritmo de reconstrução 3D. Uma pilha de 21 imagens de seções de microestrutura porosa foi obtida e usada como dados de entrada para processamento posterior. As imagens passaram pelo processo de binarização, assim como as imagens utilizadas nessa dissertação.

Logo depois, ainda no artigo supramencionado, são apresentados conceitos de reconstrução 3D e os algoritmos desenvolvidos. O desenvolvimento do algoritmo de reconstrução 3D mencionado, dedicado às microestruturas porosas, foi o principal objetivo do artigo. Os autores propuseram um algoritmo eficiente para reconstruir a microestrutura do material poroso. Além disso, duas técnicas diferentes para o algoritmo foram desenvolvidas para identificar, automaticamente, as características correspondentes nas imagens 2D subsequentes: os poros. O algoritmo desenvolvido pelos autores deve ser aplicado separadamente a cada par de imagens 2D, diferentemente do algoritmo implementado na dissertação atual. Para mais detalhes sobre o algoritmo e os métodos utilizados no artigo, veja [51].

## 3.2 Vantagens e Desvantagens da Técnica de Seccionamento em Série Convencional

A técnica de seccionamento serial, exposta neste trabalho, oferece algumas vantagens e desvantagens durante e após a aplicação de seu processo, assim como qualquer metodologia experimental [75]. As vantagens e desvantagens podem variar dependendo do material escolhido para execução da técnica. Isso acontece porque, para cada tipo de material, é possível utilizar diferentes técnicas existentes que geram sequências de imagens bidimensionais. Em particular, apresentar-se-á as vantagens e desvantagens da técnica de seccionamento em série convencional.

### 3.2.1 Vantagens

Para investigações microestruturais em materiais que não carecem de serem destruídos, o seccionamento serial oferece uma série de vantagens.

Conforme [64], a vantagem mais significativa é a flexibilidade da técnica. O custo do equipamento necessário para realização do corte serial é relativamente baixo, dependendo do escopo da investigação e o modo de operação necessário. Esses equipamentos de seccionamento são comumente encontrados na maioria dos laboratórios de materiais, portanto, a implementação manual dessa metodologia pode ser prontamente iniciada [75].

A técnica também possibilita a obtenção de dados microestruturais tridimensionais em uma ampla variedade de materiais [64]. Tanto que outra vantagem descrita pelo autor de [75] é que:

“O tipo de dados coletados durante um experimento de seccionamento serial pode ter um impacto profundo na facilidade de identificar e classificar os recursos microestruturais.” [75].

A reconstrução tridimensional de microestruturas por meio do seccionamento serial também fornece informações mais precisas sobre morfologias dos microconstituintes <sup>3</sup> do material. Desse modo, existe um controle maior sobre as propriedades físicas e mecânicas do material [15]. Além disso, é possível escolher a profundidade da camada retirada de acordo com o tamanho das particularidades da microestrutura elegida para análise [46, 81].

Ademais, os resultados obtidos em reconstruções 3D, provindos de imagens obtidas pela técnica de seccionamento serial, são bastante razoáveis [81].

### 3.2.2 Desvantagens

Uma das principais desvantagens da técnica de seccionamento em série manual é a sua natureza destrutiva. O volume da amostra é inevitavelmente consumido durante o processo de coleta de dados, o que impede a reutilização do material após o processo [75]. Sendo que a obtenção do sólido tridimensional se dá aplicando o próprio processo de seccionamento serial [19, 41]. Logo, pode-se dizer que a técnica de seccionamento em série é um método destrutivo. Por isso, é improvável que algum dia essa técnica possa

---

<sup>3</sup>Um indivíduo dentro de uma microestrutura.

competir diretamente com técnicas não destrutivas, como por exemplo a tomografia de raios-X de alta resolução [64] apud [9, 52].

Por volta de 1980, o autor de [26] comentava que a prática do método de seccionamento em série é tedioso. De fato, a autora de [54], diz que existe um outro fator que influencia negativamente na utilização dessa técnica: a quantidade de trabalho experimental necessário para realizar uma reconstrução satisfatória. A principal barreira para a aplicação desta técnica é o longo tempo despendido para obter as seções bidimensionais e realizar o procedimento de reconstrução 3D [15]. Quanto mais imagens capturadas, maior o esforço manual e computacional [10].

Historicamente, este método foi realizado por meio de cuidadosos lixamentos e/ou polimentos manuais juntamente com o uso de marcas de referência ótica, como por exemplo, microimpressões, para medir a taxa de remoção da camada [64] apud [47]. Porém, considerou-se mais duas principais limitações com relação ao processo manual: a taxa na qual as seções em série podem ser produzidas e a precisão com a qual a remoção da camada pode ser medida e controlada. As taxas de seccionamento podem ser tão lentas quanto 10 seções obtidas por dia [64] apud [4].

Atualmente, existem máquinas fotográficas modernas disponíveis para capturar as imagens bidimensionais dos materiais que são submetidos ao processo de seccionamento em série. Porém, mesmo assim a câmera é responsável por limitar a resolução das imagens bidimensionais. Isso gera perda de informações, pois a quantidade de pixel de cada imagem é finita. Com isso, nem todos os dados microestruturais são registrados. Portanto, ocorrem perdas de informações da microestrutura original para as imagens da câmera [81]. Conseqüentemente, a resolução das representações 3D finais realizadas a partir de imagens de seccionamento serial são limitadas pela espessura de cada camada de seção serial [46].

A retirada de camadas da amostra também gera perda de informações. Devido a isto, esse trabalho tem por objetivo minimizar essa perda de informações através da técnica de repetição de planos adjacentes ou da técnica de obtenção de planos intermediários por meio da média das cores, que é descrita no capítulo 4.

### 3.3 Materiais

Um foco maior é dado à reconstrução de amostras de materiais metálicos, em escala microestrutural. Porém, nada impede que a técnica seja aplicada a outros materiais e objetos de interesse. Neste trabalho foram utilizados dois materiais metálicos: Ferro Fundido Nodular e Aço Inoxidável Duplex.

O Ferro Fundido Nodular é uma liga composta de carbono e silício com teor de carbono acima de 2% e silício entre 1 e 3%, podendo conter outros elementos químicos tais como manganês, fósforo, zircônio, estanho, e etc. As propriedades dos Ferros Fundidos dependem da forma de precipitação da grafita e da estrutura da matriz. A principal característica do Ferro Fundido Nodular é a ductilidade, tenacidade e resistência mecânica devido a sua grafita que apresenta uma forma esferoidal que não interrompe a continuidade da matriz, resultando na sua melhor ductilidade e tenacidade. A presença das esferas ou nódulos de grafite mantém as características de boa usinabilidade e razoável estabilidade dimensional. Geralmente é utilizado na indústria para a confecção de peças que necessitem de maior resistência ao impacto em relação aos ferros fundidos cinzentos, além de maior resistência à tração, resistência ao alongamento e escoamento [17, 24]. A Figura 3.6 apresenta duas imagens microscópicas, de uma amostra de Ferro Fundido Nodular, que foram obtidas através do processo de seccionamento em série. Os recortes de microdureza em forma de losango são visíveis em ambas imagens e foram usados para marcar a área de interesse, para servir como marcas de referência para o alinhamento posterior da imagem e para calibrar a profundidade da remoção do material.

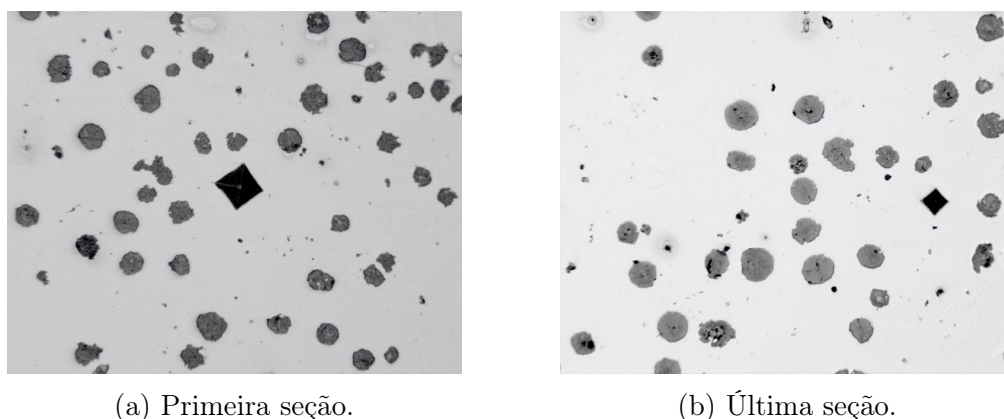


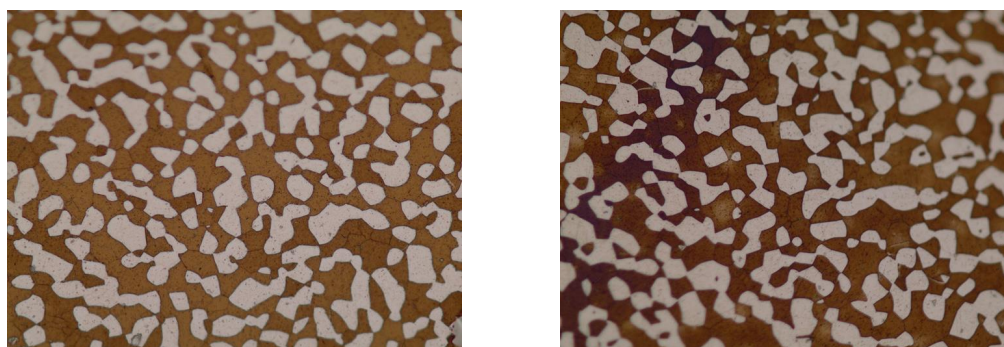
Figura 3.6: Microscopia de amostra de Ferro Fundido Nodular.

Fonte: [24].

A autora de [24] obteve 57 seções bidimensionais de Ferro Fundido Nodular, por meio da técnica de seccionamento em série. A Figura 3.6 apresenta a primeira e a última seção

arquivadas no computador. As demais imagens podem ser encontradas no Anexo A.

Já o Aço Duplex é um aço inoxidável composto por dois tipos de microestruturas: Ferrita e Austenita, onde a parcela volumétrica de ambas fases são aproximadamente iguais. Este aço é caracterizado pela resistência alta mecânica e resistência a corrosão [20]. A Figura 3.7 apresenta duas imagens microscópicas de uma amostra de Aço Inoxidável Duplex que foram obtidas através do processo de seccionamento em série. Em ambas imagens é possível ver as fases do material: Ferrita (em tons mais escuros) e Austenita (em tons mais claros). As microimpressões feitas na amostra de Aço Inoxidável Duplex não são visíveis nestas imagens, pois o autor revelou que elas foram feitas ao lado da área apresentada nas fotos.



(a) Primeira seção.

(b) Última seção.

Figura 3.7: Microscopia de amostra de Aço Inoxidável Duplex.

Fonte: [11].

O trabalho de [11] gerou 90 seções bidimensionais de Aço Inoxidável Duplex, por meio da técnica de seccionamento em série. Ao olhar a Figura 3.7 pode-se ver a primeira e a última seção bidimensional arquivadas no computador, as demais podem ser vistas no Anexo B.

Essas e outras informações sobre os materiais citados nestas seção podem ser encontradas em [17] e em [20].

Os processos de seccionamento em série do Ferro Fundido Nodular e do Aço Inoxidável Duplex foram feitos no laboratório da Universidade Federal Fluminense - Escola de Engenharia Industrial e Metalúrgica de Volta Redonda.

# Capítulo 4

## Metodologia

A ferramenta principal deste trabalho foi desenvolvida com auxílio do software Microsoft Visual Studio<sup>®</sup> escrevendo-a na linguagem de programação orientada a objetos C++. Esta é uma linguagem de programação de primeira classe para desenvolver aplicativos de computador de capacidade industrial e de alto desempenho [27, 62]. De acordo com [27], o C++ desenvolveu-se a partir do C, foi criado por Bjarne Stroustrup no início da década de 1980 na Bell Laboratories.

A escolha desta linguagem se deve a compatibilidade com diversos sistemas operacionais e a possibilidade de novas funcionalidades serem facilmente incorporadas no código. Além disso, é uma linguagem amplamente utilizada e possui vasta documentação e elevada contribuição de usuários de todo o mundo [31].

Uma das características deste desenvolvimento é o emprego do paradigma da Programação Orientada a Objetos, POO (ou do inglês, *Object Oriented Programming*, OOP), que possibilita melhorias quanto à reutilização do código e quanto ao trabalho em equipe.

“**Objetos** são essencialmente componentes de software reutilizáveis que modelam itens do mundo real.” [27].

Também utilizou-se algumas funções da biblioteca OpenCV [34, 44], para o processamento de imagens. Ela foi desenvolvida pela Intel, e está disponível em [42].

Ao fim do trabalho obteve-se a primeira versão de uma ferramenta computacional que reconstrói em 3D um material seccionado em série e, posteriormente, salva todas as características geométricas deste material para futuras análises e novos estudos. A matriz

em três dimensões fica salva em um arquivo de saída e pode ser utilizada por outras ferramentas numéricas e/ou programas de simulação computacional.

A possibilidade de adquirir um material reconstruído em 3D e reaproveitá-lo para realizar simulações de transformação de fase, crescimento de tumores, troca de calor, análise de deformação, dentre outros, torna o programa inédito no mercado e de grande aplicabilidade no meio científico. É uma ferramenta de código aberto que cria um conjunto de dados tridimensional que permite a visualização 3D através de outras ferramentas gráficas: Tecplot 360<sup>®</sup> [68], ParaView [6], VisIt [18] ou equivalentes.

Vale ressaltar que esta pesquisa não se compromete com a geração de ferramentas de visualização dos resultados, sendo assim, as representações gráficas dos domínios podem ser apresentadas com auxílio dos programas mencionados no parágrafo anterior.

Para esta primeira versão do código, foram implementadas duas abordagens distintas que prometem recuperar algumas informações, através de aproximações, que foram perdidas na hora do desbastamento da amostra real que ocorre no processo de seccionamento em série. Essa perda se deve a profundidade de lixamento e/ou polimento entre uma imagem bidimensional e sua imagem subsequente. Com isso, é necessário acrescentar imagens intermediárias a cada duas originárias da técnica de seccionamento em série. E para essa determinação, apresenta-se as duas abordagens descritas nas seções 4.1.5.1 e 4.1.5.2: a repetição de planos adjacentes e uma aproximação do que seriam os planos intermediários por meio da média das cores de planos reais vizinhos.

## 4.1 Construção do Código Computacional

Devido a linguagem de programação escolhida, a construção do código computacional é formado por objetos, atributos, classes e métodos. Os objetos são dispositivos responsáveis por seu próprio conteúdo e capazes de armazenar informações, executar uma tarefa ou dar acesso a outro objeto. Os atributos são os dados dos objetos, que podem ser constantes ou variáveis. Já as classes são modelos para a criação de objetos [30, 69, 83]. Neste caso, as classes definidas pelo programador buscam realizar a reconstrução tridimensional de microestruturas e trabalham em conjunto por meio de métodos. Os métodos são rotinas que acessam os dados das classes e são equivalentes as funções na linguagem de programação C.

### 4.1.1 Estrutura do Código

Um diagrama de classes é ilustrado conforme a Figura 4.1. A elaboração de tal diagrama foi realizada com auxílio do site *Draw.io* disponível em [43] e utilizou-se alguns princípios da Linguagem Unificada de Modelagem (UML) para construí-lo.

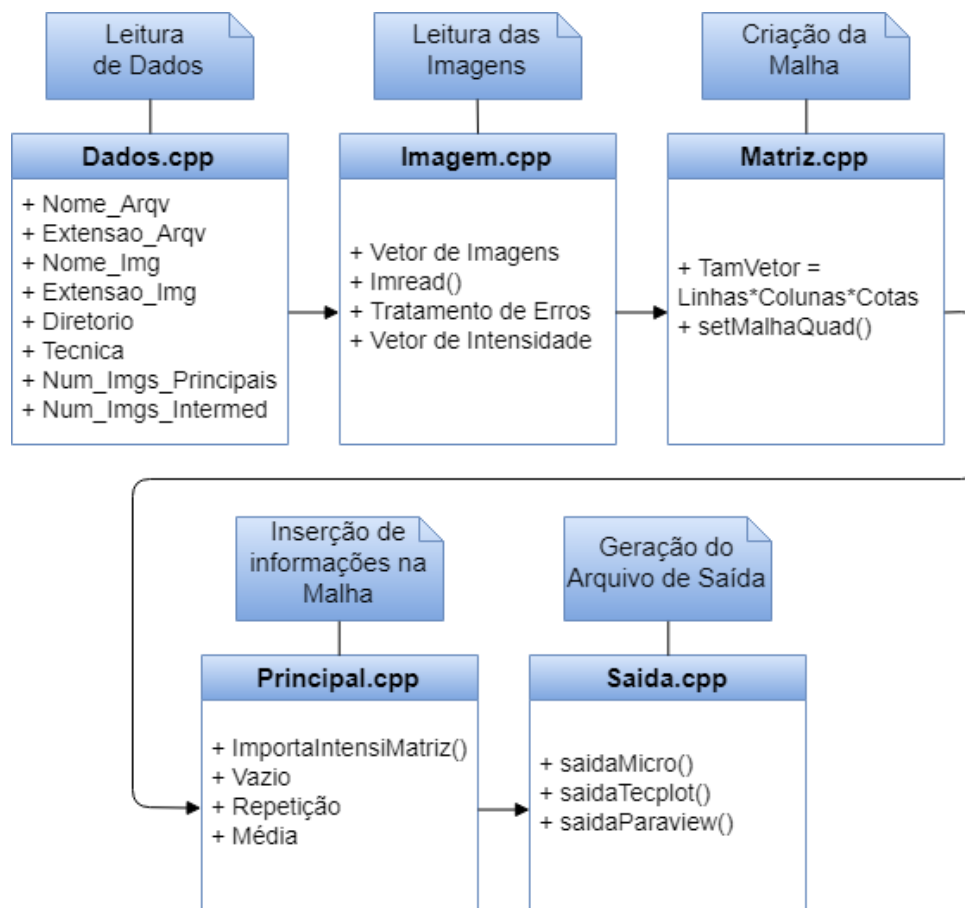


Figura 4.1: UML das classes contidas no código computacional.

Fonte: Autoria própria (2019).

As principais classes contidas no programa são: `Dados.cpp`, `Imagem.cpp`, `Matriz.cpp` e `Saída.cpp`, além do arquivo `Principal.cpp`. Elas foram escolhidas de acordo com as tarefas necessárias para se gerar a malha discreta da reconstrução tridimensional de uma microestrutura. As seções a seguir descrevem qual o papel de cada uma delas.

### 4.1.2 Classe: `Dados.cpp`

Esta classe está encarregada de coletar e armazenar dados previamente definidos em um arquivo de entrada padronizado. Neste arquivo estão localizados os dados de entrada necessários para o programa prosseguir com sua execução. O usuário pode alterar estes

dados para gerar o arquivo final com as informações da reconstrução 3D que se pretende fazer. Os principais dados de entrada são:

1. **Extensão do arquivo de saída:** são as possíveis extensões que o arquivo de saída pode ter para servir como dado de entrada para outros programas de visualização gráfica depois de gerado. Por enquanto, pode-se escolher a extensão \*.dat para, posteriormente, abrir o arquivo no Tecplot 360<sup>®</sup> [68], ParaView [6] ou VisIt [18]. Em outras versões futuras do programa pretende-se implementar novos tipos de arquivos de saída para outros programas de análise;
2. **Nome do arquivo de saída:** fica a critério do usuário escolher o nome do arquivo final que contém a malha tridimensional. Sugere-se “Saida” ou “Final”;
3. **Nome padrão das seções bidimensionais (imagens) principais** da amostra que foram salvas no computador. O código foi implementado de maneira que todas as imagens principais estejam com o mesmo nome seguido de números cardinais, na sequencia, indicando os elementos constituintes desse conjunto de seções bidimensionais. Ressalta-se que é necessário começar do número zero;
4. **Extensão das seções bidimensionais principais:** as mais utilizadas foram \*.tif, \*.png e \*.jpg. Porém é possível escolher outras extensões que sejam suportadas pelo programa devido a utilização da função *imread()* da biblioteca OpenCV [42]. Por exemplo, \*.bmp, \*.dib, \*.pbm, \*.pgm, \*.ppm, \*.sr e \*.ras;
5. **Diretório:** caminho que contém as seções bidimensionais principais salvas no computador;
6. **Parâmetro da escolha da técnica:** é a variável que indica qual abordagem vai ser utilizada durante a execução do programa. Escolhe-se “V” para quando se deseja que os planos intermediários fiquem vazios, “R” para quando a abordagem desejada é a repetição de planos adjacentes e “M” para obter planos intermediários por meio da média das cores. Ambas abordagens estão detalhadas nas seções 4.1.5.1 e 4.1.5.2.
7. **Número inicial de imagens:** é a quantidade de seções bidimensionais principais que se quer utilizar para fazer a reconstrução;
8. **Número de planos intermediários:** é a quantidade de seções bidimensionais intermediárias que se deseja inserir a cada duas seções bidimensionais principais.

Todos esses dados são inicializados com zeros ou *strings* vazias, dependendo do tipo da variável.

Com os dados digitados corretamente, cria-se um objeto que armazena todos esses dados e os torna acessíveis em qualquer parte do código. Desse modo, não é preciso fazer uma releitura do arquivo de entrada. Isto significa que é possível economizar tempo computacional se o reuso dos dados de entrada forem necessários.

### 4.1.3 Classe: **Imagem.cpp**

A classe **Imagem.cpp** está incumbida de fazer a leitura das imagens principais com o objetivo de coletar algumas informações sobre cada uma delas. Dentre as principais informações pode-se citar as dimensões (número de linhas e número de colunas) e o valores de R, de B e de G em cada pixel. E para obter o valor de intensidade de pixel, precisa-se saber o tipo de imagem e o número de canais [55]. Neste caso, são três canais: R, G e B.

Todas essas informações foram coletadas a partir da função *imread()* da biblioteca OpenCV [42, 44] que foi de extrema importância para implementação do código. A função *imread()* carrega uma imagem do arquivo especificado e a retorna. Se a imagem não puder ser lida (devido a arquivos ausentes, permissões incorretas, formato não suportado ou inválido), a função retorna uma matriz vazia [55]. O único parâmetro, que foi necessário utilizar, para chamar a função *imread()* foi o nome das imagens principais que foram dadas como dados de entrada e armazenadas em um objeto da classe **Dados.cpp**. Existem outras opções de parâmetros, porém não foram utilizados, por enquanto.

Para utilizar a função *imread()* foi necessário aprender um pouco mais sobre a classe **Mat** e usá-la também. A **Mat** é uma classe da biblioteca OpenCV que mantém as características da matriz/imagem (número de linhas e colunas, tipo de dados, etc) e um ponteiro para dados. Pode-se ter várias instâncias de **Mat** correspondentes aos mesmos dados [55]. Essa classe transforma a imagem em uma matriz de dados. Com isso, torna-se possível coletar a informação de cada pixel da imagem. Basta percorrer as linhas e as colunas dessa imagem e ir coletando os valores de intensidade de RGB de cada pixel.

Na classe **Imagem.cpp** também foram implementados alguns tratamentos de erro com relação as imagens, visto que todas as imagens devem estar contidas na mesma pasta e devem possuir a mesma dimensão. Caso pelo menos uma das imagens não estiver contida no diretório indicado no arquivo de entrada, o programa para sua execução e uma mensagem de erro aparece na tela. Caso haja divergência entre as dimensões das imagens

principais o programa também para sua execução e mostra outro tipo de mensagem na tela.

Mais detalhes sobre a biblioteca OpenCv e o funcionamento da função *imread()* e da classe **Mat** podem ser encontrados em [44, 55] e processamento de imagens com Opencv podem ser vistos em [34].

#### 4.1.4 Classe: **Matriz.cpp**

No início deste capítulo falou-se que o paradigma da POO possibilita a reutilização do código e melhora o trabalho em equipe. A classe **Matriz.cpp** é um dos exemplos de reutilização do código. Esta classe foi implementada por um colega do laboratório NMM<sup>1</sup> e também pode ser encontrada nas descrições de seu trabalho, vide [30]. Foi necessário fazer pequenas adaptações e nem todos os métodos foram utilizados. Porém essa classe continuou sendo responsável por criar o domínio discreto.

Dessa forma, uma malha tridimensional cúbica foi implementada. De acordo com [30], a menor unidade desta malha é uma célula que possui formato semelhante a de um cubo cuja união ordenada forma a matriz que representa o espaço discreto. A classe **Matriz.cpp** cria esta malha usando um vetor unidimensional como estrutura de dados.

#### 4.1.5 Arquivo principal: **Principal.cpp**

É na classe principal que foram implementadas as duas abordagens de interesse deste trabalho: a repetição de planos adjacentes e planos intermediários obtidos por meio da média das cores.

##### 4.1.5.1 Repetição de Planos Adjacentes

Considere um plano como sendo uma imagem bidimensional que foi obtida através do processo de seccionamento em série descrito no capítulo 3.

Como o domínio computacional que pretende-se gerar é um domínio discreto, então é possível pensar em um número finito de pontos que estão contidos em cada plano/imagem utilizado. Para obtê-los, juntamente com suas propriedades e informações, utiliza-se uma função chamada *imread()* da biblioteca OpenCV [42, 55].

---

<sup>1</sup>Núcleo de Modelamento Microestrutural.

O algoritmo começa percorrendo os planos mais baixos e passa para os planos superiores conforme a quantidade de pontos que cada plano contém. Com isso, o primeiro ponto do primeiro plano está fixo na origem do sistema tridimensional de coordenadas. O próximo ponto vai ser o ponto cuja a coordenada  $x$  avança uma unidade no sentido positivo de seu eixo, ou seja, prossegue para o próximo pixel da imagem. Isso ocorre até que a quantidade de pixels no eixo das abcissas se acabe, pois tem-se um número finito de pixels por imagem. Após essa etapa, avança-se uma unidade de medida no eixo das ordenadas, ou seja, avança-se um pixel na direção  $y$  e segue-se este raciocínio para o eixo das cotas até que o código computacional atinga o número que é calculado previamente pela multiplicação entre a altura e a largura da seção bidimensional vezes a quantidade de seções bidimensionais que são consideradas para fazer o empilhamento que, posteriormente, resulta na reconstrução 3D.

Enquanto a função *imread()* coleta os valores de R (vermelho), G (verde) e B (azul) de cada pixel, o código calcula um valor que definiu-se como *intensidade*, cuja a conta é feita pela seguinte expressão:

$$\text{Intensidade} = R + (G * 256) + (B * 256 * 256), \quad (4.1)$$

visto que tem-se 256 possíveis tons de R, de G e de B. Totalizando 16.777.216 possíveis cores diferentes de pixel. Durante esse processo, as cores são salvas em um vetor de tamanho  $x \times y \times z$ , onde  $x$  corresponde a largura da imagem,  $y$  é a altura e  $z$  é a quantidade total de planos que é escolhido para usar na reconstrução.

Como dados de entrada, precisa-se do número inicial de imagens principais que foi escolhido de acordo com o processo de seccionamento em série e do número de planos intermediários que deseja-se repetir a cada duas imagens.

Assim, pode-se calcular o tamanho do vetor intensidade através de uma função da biblioteca vector chamada *size()*. Tendo o tamanho do vetor de intensidade, pode-se obter a quantidade de pontos que cada plano possui. Para isto, basta dividir o tamanho do vetor intensidade pela quantidade inicial de planos que é um dado de entrada.

Também consegue-se calcular a quantidade total de planos que se almeja visualizar no final do código. Soma-se o número inicial de planos com o número de planos que existem entre cada dois planos multiplicado pela quantidade de planos iniciais menos um. Desse modo, a quantidade total de pontos que deve-se ter é a multiplicação entre a quantidade total de planos vezes a quantidade de pontos de cada plano. É nessa parte que o código para repetição de planos passa a ser desenvolvido cuidadosamente, sendo

necessário percorrer todos os pontos que estão contidos no volume de controle.

O preenchimento dos planos intermediários é feito com o auxílio de comandos condicionais e comandos de repetição padrões da linguagem de programação C++. A ideia principal é que a cor do plano de baixo vai se repetir nos planos intermediários mais próximos que estão em cima, de modo que as coordenadas  $x$  e  $y$  desse ponto não se alterem. Cada ponto/pixel da imagem possui uma cor. Considere que esses valores são os estados da matriz em um determinado ponto. É preciso repetir os estados dos planos inferiores nos planos superiores, onde os planos intermediários estão vazios. A solução para esse problema foi identificar onde cada plano acaba de ser preenchido, isto é, foi necessário saber em qual ponto a coordenada  $z$  se altera. Isto foi necessário porque o código está vetorizado. Para esta identificação, utilizou-se a ideia de contadores. O contador principal serve para contar quantos pontos já foram percorridos através do comando de repetição. E quando esse contador for divisível pela quantidade total de pontos contidos em um plano, isto é, esta divisão possui resto igual zero, sabe-se que um plano acabou de ser preenchido com seus respectivos estados. Assim, repete-se o estado desse plano no plano intermediário acima, através do segundo contador que volta o vetor que contém os valores dos estados dos pontos na posição inicial do plano que pretende-se repetir. E um terceiro contador vai verificar quantos planos foram repetidos. Se esse terceiro contador for igual ao número de planos que deseja-se repetir a cada dois planos, então o primeiro contador não é subtraído pela quantidade total de pontos que contém em um plano: ele apenas continua a preencher o próximo plano que não vai ser repetido dessa vez. Caso contrário, o primeiro contador é subtraído pela quantidade de pontos que um plano contém, com o objetivo de voltar no início dos estados. Essa ideia se repete conforme a quantidade de planos que coloca-se como dado de entrada até o último plano ser totalmente preenchido com seu devido estado.

#### 4.1.5.2 Planos Intermediários Obtidos por meio da Média das Cores

Uma outra alternativa para preencher os planos intermediários é utilizar a média entre as cores. Para obter a média entre duas cores, basta somar os valores de R, os valores de G e os valores de B e dividir cada um por dois. O resultado é uma nova cor que é a média entre duas cores. Vale ressaltar que o resultado dessa divisão pode ser um número não inteiro. Então, tais valores são arredondados para o inteiro mais próximo.

O primeiro passo desta técnica é empilhar as seções bidimensionais, que são os dados de entrada para o software principal deste trabalho, do mesmo modo que foi feito para

construir a repetição de planos. Depois que todos os planos principais são empilhados juntamente com os planos intermediários vazios, os vazios passam a ser preenchidos utilizando uma proporção vinculada com a equação (4.2) que é a taxa de variação de uma função, visto que nem sempre tem-se apenas um plano intermediário no meio dos principais.

$$T_{V_{\text{canal}}} = \frac{f_{\text{canal}}(x_k, y_k, z_{k+1}) - f_{\text{canal}}(x_k, y_k, z_k)}{z_{k+1} - z_k} \quad (4.2)$$

Considerou-se uma função  $f(x, y, z)$  de três variáveis que representa os níveis de R, de G e de B de cada pixel de cada imagem principal. A variável “canal”, contida na equação (4.2), precisa ser substituída por R, G ou B, separadamente. Isto significa que existem três valores, distintos ou não, de taxa de variação para cada valor de  $k$ , (uma taxa para cada canal) onde  $k = 1, 2, \dots, n - 1$  e  $n$  representa a quantidade de planos principais que foram dados como dado de entrada. Além disso, o cálculo do valor da taxa de variação  $T_V$ , apresentada na equação (4.2), se repete conforme a quantidade de pixels (ou pontos) que cada plano contém e, portanto, de acordo também com a quantidade de planos principais utilizados para fazer a reconstrução 3D.

A taxa de variação  $T_V$  simboliza o quanto o R, o G e o B “estão distantes” um dos outros através de uma proporção que está relacionada à quantidade de planos intermediários que se aspira preencher. As coordenadas  $x$  e  $y$  são as mesmas para um dado ponto fixo do plano que o algoritmo percorre. O que muda nessa posição é a informação da coordenada  $z$  que indica em qual plano esse ponto de coordenada  $x$  e  $y$  está. Isto significa que as variáveis  $x_k$  e  $y_k$  são as mesmas para pontos escolhidos na mesma direção (em cima e em baixo). Já as variáveis  $z_k$  e  $z_{k+1}$  correspondem aos primeiro e segundo planos principais quando  $k = 1$ , depois correspondem ao segundo e terceiro planos principais quando  $k = 2$  e assim por diante até o algoritmo acabar de percorrer todos os planos desejados. Observou-se que o valor da diferença entre  $z_{k+1}$  e  $z_k$  é o mesmo valor da quantidade total de planos intermediários mais um.

Resumindo, para canal=R e  $k = 1$ , o valor de  $f_R(x_1, y_1, z_1)$  representa o valor de R no primeiro ponto do primeiro plano principal e o valor de  $f_R(x_1, y_1, z_2)$  representa o valor de R no primeiro ponto do segundo plano principal. Esse raciocínio se propaga analogamente para o G, representado por  $f_G$ , e para B, representado por  $f_B$ .

O código preenche todos os planos vazios por blocos com o objetivo de tentar deixar o código paralelizável.

Dessa forma, conseguiu-se obter os planos intermediários utilizando a média das cores. Na Figura 4.2 é possível ver três exemplos de média de cores de um único pixel por tabela. As cores intermediárias são calculadas de modo que os pontos intermediários que estão mais perto de um ponto que contém um estado específico fiquem mais próximos desse estado e os pontos mais perto de um segundo estado fiquem mais próximos desse segundo estado.

R	G	B	Cor
255	255	255	
213	213	213	
170	170	170	
128	128	128	
85	85	85	
43	43	43	
0	0	0	

(a) Branco e preto.

R	G	B	Cor
127	127	127	
129	106	109	
130	85	92	
132	64	74	
133	42	56	
135	21	39	
136	0	21	

(b) Cinza e vinho.

R	G	B	Cor
237	28	36	
240	64	30	
243	99	24	
246	135	18	
249	171	12	
252	206	6	
255	242	0	

(c) Vermelho e amarelo.

Figura 4.2: A partir de duas cores principais escolheu-se, como exemplo, cinco cores intermediárias para serem calculados os valores de R, de G e de B, utilizando a média de cores.

Fonte: Autoria própria (2018).

#### 4.1.6 Classe: Saída.cpp

A classe **Saida.cpp**, como o próprio nome já diz, é a classe responsável por salvar os dados de saída que são gerados enquanto o programa está sendo executado. A implementação dessa classe tem por objetivo criar e montar um arquivo de texto que contém os pontos espaciais com suas respectivas intensidades e cores identificadas pelo sistema RGB.

Por enquanto, as opções de extensões para gerar o arquivo de saída são \*.dat e \*.vtk. Vários arquivos de extensão \*.dat foram utilizados como dado de entrada em programas de visualização gráfica, já citados anteriormente: Tecplot 360<sup>®</sup> [68], ParaView [6], VisIt [18] ou equivalentes.

Para o arquivo do tipo \*.dat tem-se as três primeiras colunas compostas por um ponto do espaço  $\mathbb{R}^3$ , a quarta coluna contém os valores de intensidade que foi definida na equação (4.1) e as últimas três colunas tem-se os valores de R, G e B, de cada ponto, respectivamente. O cabeçalho é construído de acordo com a Figura 4.3. E ao olhar a Figura 4.4 é possível ver um exemplo cuja as dimensões das imagens utilizadas para construir a malha são 150x112 e a quantidade total de imagens empilhadas é 10.

```
//Cabeçalho para o o arquivo de extensão .dat:
arquivo << "Title = 'ImagePorcess'" << endl;
arquivo << "variables= X, Y, Z, Intensity, Red, Green, Blue" << endl;
arquivo << "zone" << endl;

//Dados a serem inseridos no arquivo .dat:
arquivo << "T=\\" \" << \" \" << "i=" << colunas << \" \" << "j=" << linhas << \" \" <<
    "k=" << cotas << \" \" << "F=point" << \" \" << endl;
arquivo << \" \" << endl;
```

Figura 4.3: Código do cabeçalho do arquivo de saída.  
Fonte: Aatoria própria (2019).

```
1 Title = 'ImagePorcess'
2 variables= X, Y, Z, Intensity, Red, Green, Blue
3 zone
4 T=" " i=150 j=112 k=10 F=point
5
```

Figura 4.4: Exemplo de um cabeçalho de um arquivo de saída.  
Fonte: Aatoria própria (2019).

É sabido que um arquivo de saída com extensão \*.vtk também pode ser visualizado no ParaView [6], mas no decorrer do trabalho não foi utilizado porque tem-se apenas um semi cabeçalho montado para tal extensão.

Espera-se que outras extensões para o arquivo de saída estejam disponíveis em versões futuras do programa.

#### 4.1.7 Algoritmo

As tarefas realizadas pelo código computacional podem ser resumidas através dos seguintes passos:

1. Leitura dos dados de entrada descritos na seção 4.1.2;
2. Leitura das seções bidimensionais principais e coleta das devidas informações, através da classe **Imagem.cpp**;
3. Criação de uma malha tridimensional (classe **Matriz.cpp**) de acordo com a quantidade de seções bidimensionais principais e intermediárias lidas no passo 1. e de acordo também com as informações coletadas no passo 2.;
4. Inserção das informações das seções bidimensionais principais e intermediárias na malha tridimensional;
5. Geração do arquivo de saída;

O Algoritmo 1 mostra a estrutura da função *main()* contida no arquivo **Principal.cpp**. Primeiramente, ocorre a criação de um objeto com o nome “*dados*”. A seguir, o método *setDados()* é aplicado neste objeto com o propósito de fazer a leitura do arquivo “Entrada.dat”. No próximo passo ocorre a leitura das imagens através do método *imagem()*. As informações coletadas na hora da leitura são salvas nas variáveis *vetorIntensi*, *tamLin* e *tamCol*. Feita a coleta das informações das imagens, o código cria a malha tridimensional com a chamada do método *imagem()*. Agora o objeto *dados* possui as informações necessárias para criação da malha e é passado como argumento. Ademais, o método *importaIntensiParaMatriz()* fica responsável por fazer a inserção das informações na malha 3D. Por fim, cria-se um objeto com o nome de “*saida*” e aplica-se o método *saidaMicro()* sobre ele. Esse método testa se o arquivo de saída vai possuir extensão \*.vtk ou \*.dat (para ParaView [6] ou Tecplot [68], respectivamente).

---

**Algoritmo 1** *main()*

---

- 1: Criar objeto *dados*
  - 2: *dados.setDados("Entrada.dat")*
  - 3: *imagem(dados, vetorIntensi, tamLin, tamCol)*
  - 4: *matriz(dados)*
  - 5: *importaIntensiParaMatriz(dados, vetorIntensi, matriz, tamLin, tamCol)*
  - 6: Criar objeto *saida*
  - 7: *saida.saidaMicro(dados, matriz, dados.getNomeArquivoSaida())*
- 

O método *importaIntensiParaMatriz()* é descrito pelo Algoritmo 2. Ele testa qual é o valor do parâmetro de escolha da técnica que foi deixado como dado de entrada.

---

**Algoritmo 2** *importaIntensiParaMatriz(dados, vetorIntensi, matriz, tamLin, tamCol)*

---

- 1: **se** Parâmetro da escolha da técnica == "V" **então**
  - 2:   Preencha a malha somente com os planos principais. Os planos intermediários ficam vazios (todos pixels na cor preta)
  - 3: **fim se**
  - 4: **se** Parâmetro da escolha da técnica == "R" **então**
  - 5:   Execute técnica da Repetição de Planos Adjacentes
  - 6: **fim se**
  - 7: **se** Parâmetro da escolha da técnica == "M" **então**
  - 8:   Execute técnica da Obtenção de Planos Intermediários por meio da Média das Cores
  - 9: **fim se**
-

O Algoritmo 3 descreve uma opção de deixar os planos intermediários vazios (na cor preta) com intuito visualizá-los sem a utilização de uma das técnicas abordadas no trabalho.

---

**Algoritmo 3** Preencher malha somente com os planos principais
 

---

```

1: contador = 0
2: para i=0 até tamVetorIntensi faça
3:   matrix->setState0(contador, vetorIntensi[i])
4:   se (contador % tamPlano) == 0 então
5:     contador += (dados->getNumPlanosIntermedi()*tamPlano);
6:   fim se
7: fim para

```

---

Já os Algoritmos 4 e 5 descrevem as técnicas utilizadas no código computacional.

---

**Algoritmo 4** Técnica da Repetição de Planos Adjacentes
 

---

```

1: contador = 0
2: contIntensi = 0
3: contPreenchi = 0
4: para i=0 até (tamPlano*qntTotalPlanos) faça
5:   matrix->setState0(contador, vetorIntensi[contIntensi])
6:   contador++
7:   contIntensi++
8:   se (contador % tamPlano) == 0 então
9:     contPreenchi++
10:    se (contPreenchi % (dados->getNumPlanosIntermedi()+1) ) == 0 então
11:      contIntensi = contIntensi
12:    senão
13:      contIntensi -= tamPlano
14:    fim se
15:  fim se
16: fim para

```

---

---

**Algoritmo 5** Técnica da Obtenção de Planos Intermediários por meio da Média das Cores
 

---

```

1: Aplique o Algoritmo 3.
2: auxiliar = 0
3: para k=0 até (numInicialPlanos-1) faça
4:   para i=1 até planosIntermedi faça
5:     para j=0 até tamPlano faça
6:       intensiPonto1 = (*vetorIntensi)[j + (tamPlano)*k]
7:       intensiPonto2 = (*vetorIntensi)[j + (tamPlano)*(k+1)]
8:       R1 = convIntensiParaR(intensiPonto1, 256)
9:       G1 = convIntensiParaG(intensiPonto1, 256)
10:      B1 = convIntensiParaB(intensiPonto1, 256)
11:      R2 = convIntensiParaR(intensiPonto2, 256)
12:      G2 = convIntensiParaG(intensiPonto2, 256)
13:      B2 = convIntensiParaB(intensiPonto2, 256)
14:      parametroR = (R2 - R1) / (planosIntermedi + 1)
15:      parametroG = (G2 - G1) / (planosIntermedi + 1)
16:      parametroB = (B2 - B1) / (planosIntermedi + 1)
17:      R = R1 + parametroR*(i)
18:      G = G1 + parametroG*(i)
19:      B = B1 + parametroB*(i)
20:      intensidadeMedia = R + (G * 256) + (B * 256 * 256)
21:      matrix->setState0(j + (tamPlano)*(i) + (auxiliar), intensidadeMedia)
22:     fim para
23:   fim para
24:   auxiliar += (tamPlano*(planosIntermedi+1));
25: fim para

```

---

## 4.2 Binarização das Imagens

Em geral, antes de efetuar a reconstrução 3D é feito um tratamento nas imagens originais que reduzem a quantidade de cores que as mesmas possuem, passando essa quantidade para apenas duas cores: preto e branco. Esse processo é chamado *binarização (de imagens)* devido a quantidade de cores que elas passam a apresentar após sua aplicação.

A binarização é o método mais simples de segmentação de imagens. A segmentação consiste em identificar os objetos e/ou materiais que se deseja analisar na imagem [53]. A princípio, as imagens originais originárias da técnica de seccionamento em série são coloridas, ou seja, podem ser representadas no sistema de cores RGB que foi descrito na seção 2.1.4. E para aplicar o processo de binarização em uma imagem, o primeiro passo

é transformar uma imagem original colorida (Figura 4.5a) em uma imagem que contenha apenas tons de cinza (Figura 4.5b). A imagem digital na escala cinza pode ter duzentos e cinquenta e seis (256) ou mais níveis de tons de cinza, e após a aplicação do processo de binarização a imagem digital final, passa a ter seus níveis reduzidos à somente dois: preto e branco (Figura 4.5c).



(a) Foto original colorida.

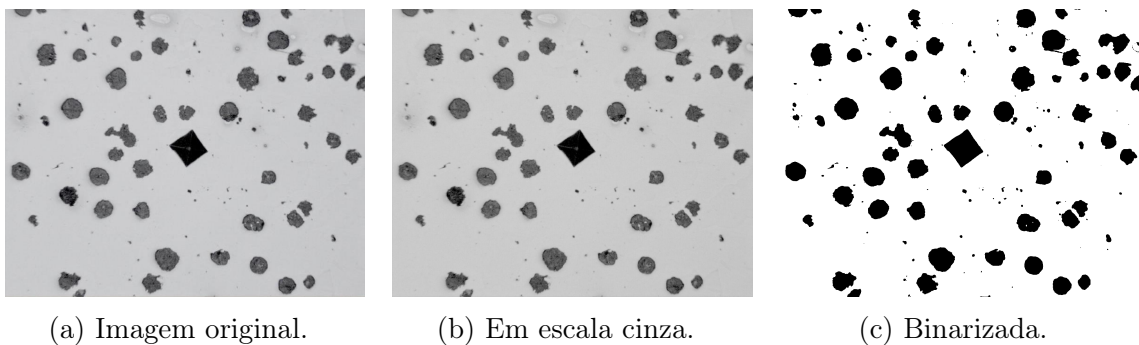
(b) Em escala cinza.

(c) Binarizada.

Figura 4.5: Exemplo da aplicação do método de binarização em uma imagem.

Fonte: Autoria própria (2018).

A conversão de uma imagem com níveis de cinza para uma imagem com representação binária é importante para identificar objetos que se almeja analisar e separá-los do fundo da imagem, como é o caso do Ferro Fundido Nodular, onde se espera segmentar os nódulos dos demais componentes da imagem, vide Figura 4.6.



(a) Imagem original.

(b) Em escala cinza.

(c) Binarizada.

Figura 4.6: Aplicação do processo de binarização em uma seção bidimensional de Ferro Fundido Nodular.

Fonte: (a) [24]. (b) e (c) Autoria própria (2018).

As imagens do Ferro Fundido Nodular originárias do processo de Seccionamento em Série já possuem muitos tons de cinza. Com a transformação da imagem original (Figura 4.6a) para escala cinza (Figura 4.6b) pode-se observar pouca modificação. Ao final do processo de binarização, obteve-se uma imagem com duas cores (preto e branco), onde os nódulos são pretos e o fundo da imagem é branco (Figura 4.6c).

Já no Aço Inoxidável Duplex deseja-se segmentar a austenita (em branco) e a ferrita (em preto), conforme pode ser visto na Figura 4.7. Analogamente, tem-se a imagem original (Figura 4.7a). Posteriormente, obteve-se a mesma imagem em escala cinza (Figura 4.7b) e, por fim, a imagem binarizada (Figura 4.7c).

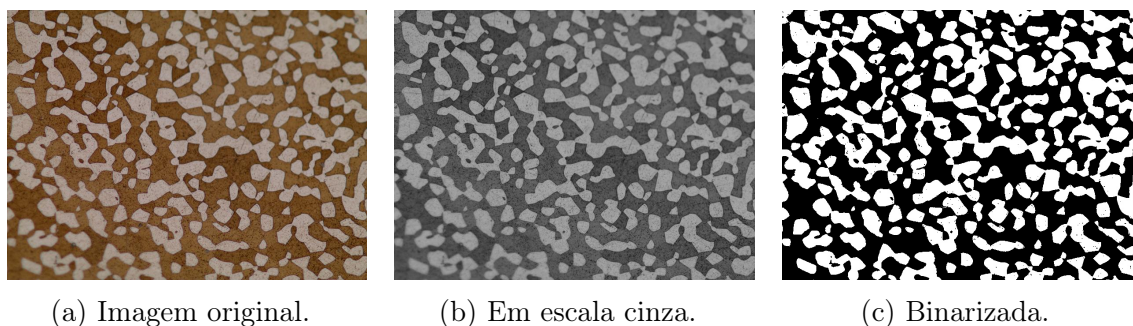


Figura 4.7: Aplicação do processo de binarização em uma seção bidimensional de Aço Inoxidável Duplex.

Fonte: (a) [11]. (b) e (c) Autoria própria (2018).

A separação dos objetos contidos na imagem é baseada na variação de intensidade entre os pixels do objeto e os pixels do plano de fundo. Para diferenciar os pixels de interesse dos pixels não relevantes (os quais são eventualmente rejeitados), realiza-se uma comparação de cada valor de intensidade de pixel em relação a um limiar determinado de acordo com o problema a ser resolvido. Depois de separar corretamente os pixels de interesse, pode-se definir com um valor determinado para identificá-los, ou seja, pode-se atribuir a eles um valor de 0 (preto), 255 (branco) ou qualquer valor que atenda às suas necessidades.

O código implementado para fazer a transformação das imagens coloridas para escala cinza e posteriormente binarizá-las também foi desenvolvido com o auxílio da biblioteca OpenCV [42]. Utilizou-se a classe **Mat** e a função *imread()* (já vistas na seção 4.1.3). Outras funções interessantes como *cvtColor()*, *threshold()* e *Imwrite()* foram utilizadas de maneira bem simples e eficiente.

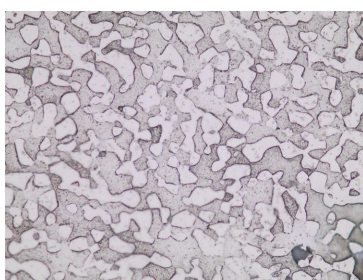
- A função *cvtColor()* converte um espaço de cores de uma imagem de para outro espaço de cores. Neste caso, a conversão de uma imagem colorida para escala cinza é feita com o parâmetro **CV\_BGR2GRAY**.
- A função *threshold()* modifica o valor de cada pixel. Se o valor do pixel for maior que um valor limite, ele é atribuído a um valor (pode ser branco), caso contrário, é atribuído outro valor (pode ser preto). O primeiro argumento dessa função é a

imagem de origem, que deve ser uma imagem em tons de cinza, visto que utilizou-se *cvtColor()* para obtê-la. O segundo argumento é o valor limite usado para classificar os valores de pixel. O terceiro argumento é o **maxVal** que representa o valor a ser dado se o valor do pixel for maior que (às vezes menor que) o valor do limite [55].

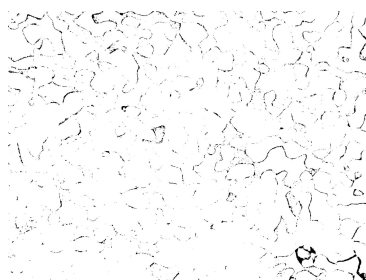
- A função *Imwrite()* salva uma imagem em um arquivo especificado. O formato da imagem é escolhido com base na extensão do nome do arquivo.

Durante os estudos para realização deste trabalho encontrou-se duas possibilidades de erro que podem ocorrer com as seções bidimensionais obtidas através da técnica de seccionamento em série manual:

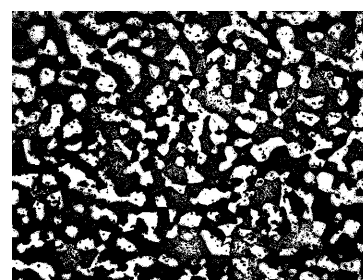
1. A própria técnica, que propõe que o material seja seccionado, pode acarretar erro no instante em que ocorre a tomada de foto ou até mesmo os produtos químicos utilizados no processo podem causar danos ao material em questão e proporcionar um erro evidente em algumas imagens provenientes da técnica. Como por exemplo, a Figura 4.8a. O processo de binarização aplicado a esta imagem não atingiu seu objetivo de fazer a segmentação dos materiais desejados (Figura 4.8b). Então, a proposta é corrigir computacionalmente as imagens que contém esses tipo de erros. E uma alternativa encontrada para solucionar esse problema foi aplicar o processo de binarização de imagem com outro valor para o segundo argumento da função *threshold()* descrita anteriormente. Na Figura 4.8b o valor do parâmetro utilizado foi 100. E na Figura 4.8c utilizou o valor 205. Note que o erro foi minimizado.



(a) Imagem original obtida através do Seccionamento em Série.



(b) Aplicação do processo de binarização com valor de *threshold* igual a 100.



(c) Aplicação do processo de binarização com valor de *threshold* igual a 205.

Figura 4.8: Erro oriundo do processo de seccionamento em série.

Fonte: (a) [11]. (b) e (c) Autoria própria (2018).

2. Outro erro que pode ocorrer é um erro causado apenas na hora do tratamento da imagem, ou seja, a imagem original (Figura 4.9a) não contém erros derivados do processo de seccionamento em série. O processo de binarização aplicado nesta imagem que insere

um erro não esperado. A solução sugerida para consertar este segundo erro é a mesma sugerida no erro anterior. Na Figura 4.9b foi aplicado o processo de binarização com o valor do segundo argumento da função *threshold()* sendo 100. E na Figura 4.9c o valor foi 140. Observe que o erro também foi minimizado.

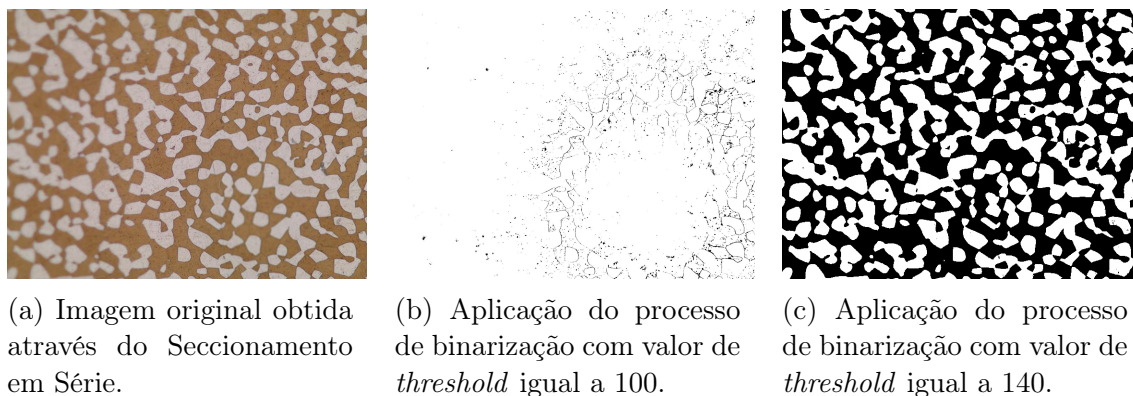


Figura 4.9: Erro derivado do processo de binarização.

Fonte: (a) [11]. (b) e (c) Autoria própria (2018).

Portanto, ambos erros podem ser solucionados com a alteração do valor do segundo parâmetro da função *threshold()*. Os valores modificados foram determinados através de testes por meio de tentativa e erro, com intuito de se aproximar da imagem desejada. Este ajuste manual foi feito em cada imagem, separadamente.

Uma outra alternativa para corrigir esses dois tipos erros é motivada pela utilização das técnicas descritas nas seções 4.1.5.1 e 4.1.5.2. Isto significa que existe a opção de descartar as imagens que não tem possibilidades de correções e substituí-las por outros planos oriundos das abordagens implementadas para encontrar planos intermediários.

### 4.3 Ferramentas Gráficas

Esta seção foi reservada para fazer uma breve descrição sobre as três ferramentas gráficas utilizadas neste trabalho. Os ícones desses softwares podem ser vistos na Figura 4.10.



Figura 4.10: Ícones dos aplicativos.

Fonte: (a) [68], (b) [6] (c) [18].

O Tecplot 360<sup>®</sup> é um software comercial de simulação numérica que ajuda engenheiros e cientistas a descobrirem, analisarem e entenderem informações provindas de dados complexos. Ele pode ser usado para visualizar a saída de linguagens de programação, como o C++, e pode manipular grandes conjuntos de dados, automatizar fluxos de trabalho e visualizar resultados paramétricos [68]. No entanto, o Tecplot 360<sup>®</sup> é um software pago. Para mais detalhes veja <https://www.tecplot.com/>.

O ParaView é um aplicativo de código aberto, multiplataforma, na qual é possível fazer visualização gráfica e análise dados [59]. Ele permite que usuários possam, rapidamente, criar visualizações com seus dados, usando técnicas qualitativas e quantitativas. Os criadores do ParaView desenvolveram-o de modo que fosse lícito analisar conjuntos de dados extremamente grandes usando recursos de computação de memória distribuída. Ele pode ser executado em supercomputadores para analisar conjuntos de dados na faixa de petabytes de tamanho ou pode ser executado em computadores pessoais onde se tem conjuntos de dados menores [6]. Já ganhou vários prêmios relacionados à computação de alto desempenho e tornou-se uma ferramenta muito utilizada em laboratórios nacionais, universidade e indústrias. Um dos principais objetivos da equipe do ParaView foi criar uma interface flexível e de fácil entendimento. Mais detalhes sobre este software estão disponíveis em <https://www.paraview.org/>.

O VisIt também é uma ferramenta de código aberto. Nele é permitido visualizar, interagir, fazer animações e análise gráfica com dados científicos. Ele é compatível com os sistemas operacionais Windows, Linux e Mac, podendo ser usado para visualizar campos escalares e vetoriais definidos em malhas estruturadas e não estruturadas 2D e 3D. O VisIt também foi projetado para lidar com tamanhos de conjuntos de dados muito grandes (em escala de terabytes) e ainda pode manipular pequenos conjuntos de dados (em escala de kilobytes). Devido ao seu design de plugin personalizável, o VisIt é capaz de visualizar dados de mais de 120 diferentes formatos de dados científicos [18]. No site <https://wci.llnl.gov/simulation/computer-codes/visit/> é possível encontrar mais detalhes.

Para as três ferramentas gráficas utilizadas nesse trabalho, fez-se o uso de arquivos no formato \*.dat. Para saber como abrir um arquivo desse tipo no Tecplot 360<sup>®</sup> veja o APÊNDICE A. O passo a passo de como abrir esse tipo de arquivo no ParaView pode ser visto no APÊNDICE B e, por fim, também foi descrito um passo a passo de como abrir um arquivo desse tipo no VisIt, vide APÊNDICE C. Futuramente pretende-se utilizar outros formatos de arquivos e outros softwares equivalentes.

## 4.4 Informações do Computador Pessoal

O computador utilizado para pesquisa, criação do código computacional, geração dos arquivos de saída e obtenção da visualização dos resultados com auxílio de ferramentas gráficas foi um computador pessoal (notebook) da marca Dell<sup>®</sup>, modelo Inspiron 3421. As especificações do modelo estão contidas na Tabela 4.1.

<b>Processador</b>	Intel <sup>®</sup> Core <sup>™</sup> i5
<b>Velocidade do relógio</b>	1.8GHz
<b>Memória RAM</b>	4,00GB
<b>Placa de vídeo</b>	Integrada
<b>Sistema Operacional</b>	Windows 10 Pro

Tabela 4.1: Informações referentes ao computador utilizado neste trabalho.

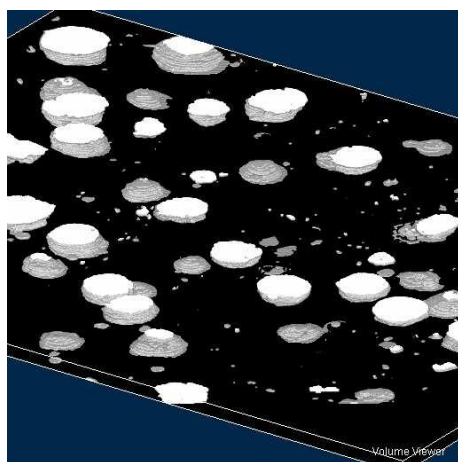
# Capítulo 5

## Resultados e Discussão

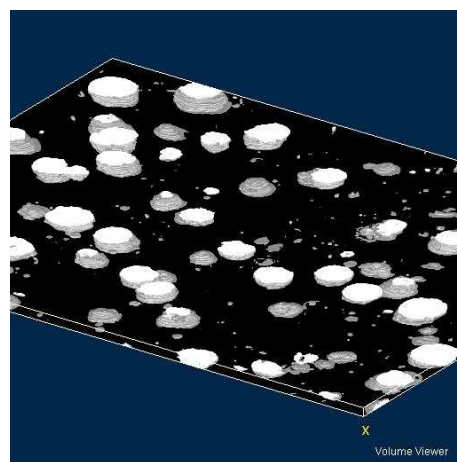
Este capítulo é destinado a apresentar e analisar os testes e resultados obtidos através da ferramenta computacional desenvolvida neste trabalho, descrita detalhadamente no Capítulo 4. Os resultados de autoria própria podem ser visualizados com auxílio de ferramentas gráficas supramencionadas com minuciosidade na seção 4.3. Em particular, foram utilizados os softwares Tecplot 360<sup>®</sup> [68], ParaView [6] e VisIt [18].

### 5.1 Ferro Fundido Nodular

A reconstrução 3D da microestrutura de Ferro Fundido Nodular feita por [24] sugere esferas de grafite na matriz ferrítica (Figura 5.1), onde utilizou-se 10 e 8 seções bidimensionais obtidas através da técnica de seccionamento em série.



(a) Com 10 seções bidimensionais.



(b) Com 8 seções bidimensionais.

Figura 5.1: Reconstrução 3D da microestrutura do Ferro Fundido Nodular.

Fonte: [24].

Ao olhar os resultados apresentados na Figura 5.1, optou-se por utilizar a mesma quantidade de planos principais para verificação do código computacional desenvolvido. Logo, nas Figuras 5.2, 5.3 e 5.4 é possível ver resultados da reconstrução 3D da amostra de Ferro Fundido Nodular visualizadas por meio do Tecplot 360<sup>®</sup>, do ParaView e do VisIt, respectivamente. A reconstrução 3D apresentada nas Figuras 5.2a, 5.3a e 5.4a possuem 10 seções bidimensionais, selecionadas a cada 6 imagens originais obtidas por [24]. A reconstrução 3D apresentada nas Figuras 5.2b, 5.3b e 5.4b possuem 8 seções bidimensionais, selecionadas a cada 8 imagens originais obtidas por [24]. Conseqüentemente, esses resultados podem ser comparados com a Figura 5.1 de autoria de [24] e validados através da visualização das esferas de grafite contidas na microestrutura de Ferro Fundido Nodular.

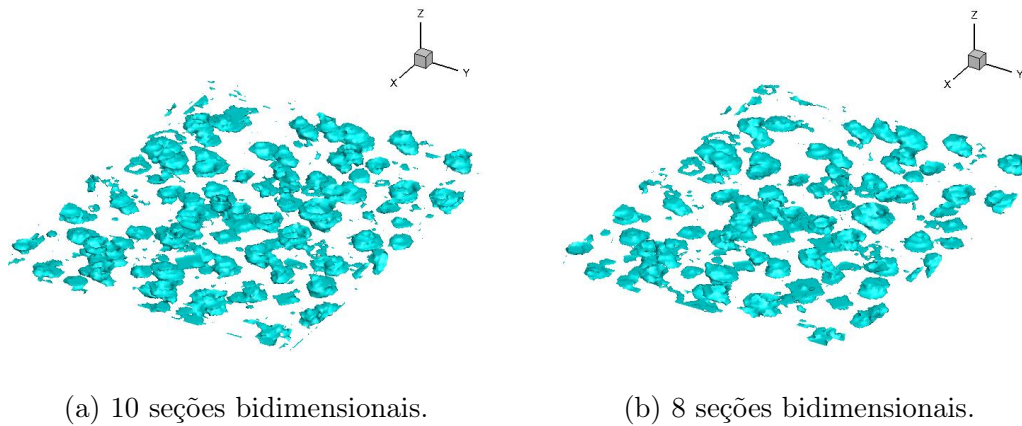


Figura 5.2: Reconstrução 3D do Ferro Fundido Nodular visualizada pelo Tecplot 360<sup>®</sup>.  
Fonte: Autoria própria (2019).

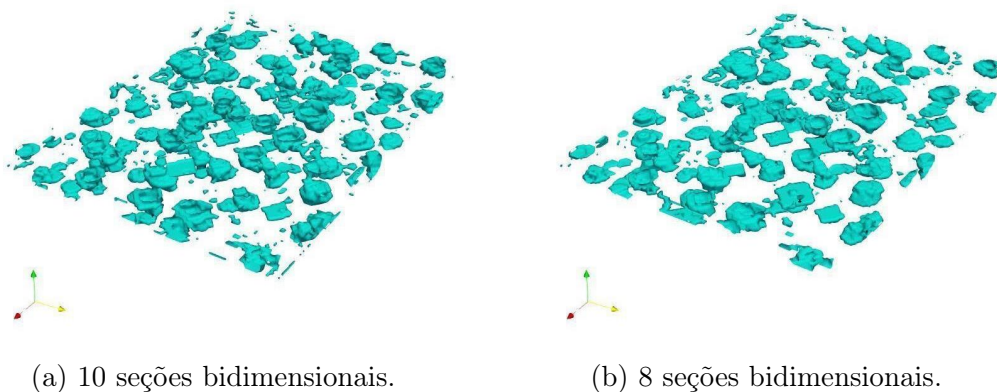


Figura 5.3: Reconstrução 3D do Ferro Fundido Nodular visualizada pelo ParaView.  
Fonte: Autoria própria (2019).

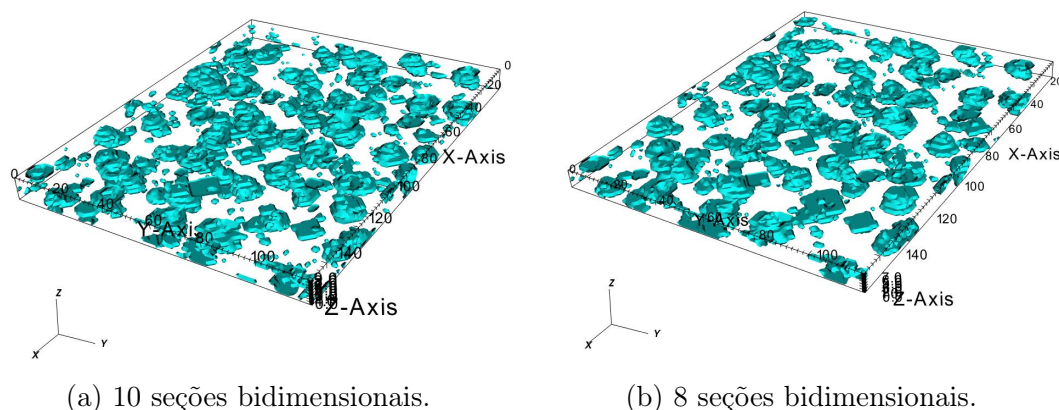


Figura 5.4: Reconstrução 3D do Ferro Fundido Nodular visualizada pelo VisIt.  
Fonte: Autoria própria (2019).

Pode-se observar que a visualização dos resultados, obtidos pelo código computacional, apresentados nas Figuras 5.2, 5.3 e 5.4 foram semelhantes aos resultados da Figura 5.1. Além disso, é possível comparar a visualização, dos resultados, feita pelas ferramentas gráficas utilizadas para conceber a reconstrução 3D. Os três resultados, vistos nas Figuras 5.2, 5.3 e 5.4, são iguais. Porém, notou-se que a visualização de cada ferramenta gráfica é diferente, pois cada uma delas usa um algoritmo diferente para fazer a renderização.

Todos os planos principais utilizados nas reconstruções ilustradas nas Figuras 5.2, 5.3 e 5.4 passaram pelo processo de binarização descrito na seção 4.2 do capítulo anterior. Posteriormente, a leitura das imagens bidimensionais foi feita, suas respectivas informações foram coletadas e inseridas na malha tridimensional implementada, por meio da ferramenta computacional principal desenvolvida no trabalho. Sendo assim, o arquivo de saída de extensão \*.dat foi gerado e aberto com os programas de visualização gráfica Tecplot 360<sup>®</sup>, ParaView e VisIt.

As informações sobre a resolução das imagens utilizadas nas reconstruções tridimensionais apresentadas nas Figuras 5.2, 5.3 e 5.4, o tempo computacional<sup>1</sup> gasto aproximado para gerar o arquivo de saída através do código computacional desenvolvido e o tamanho final do arquivo de saída com extensão \*.dat podem ser encontrados na Tabela 5.1.

	<b>Dimensões</b>	<b>Tempo computacional</b>	<b>Tamanho do arquivo</b>
<b>10 seções</b>	150 x 120	2.56 segundos	4,92 MB
<b>8 seções</b>	150 x 120	2.31 segundos	3,94 MB

Tabela 5.1: Informações da reconstrução 3D apresentada nas Figuras 5.2, 5.3 e 5.4.

<sup>1</sup>As informações de tempo computacional presentes nas tabelas não incluem o tempo computacional gasto na abertura do arquivo de saída através de ferramentas gráficas.

Outros testes com Ferro Fundido Nodular foram feitos com seções bidimensionais de resolução 1280 x 1024 e os resultados podem ser vistos na Figura 5.5, onde utilizou-se todas as 57 seções bidimensionais obtidas por [24]. Analogamente, todas as imagens passaram pelo processo de binarização.

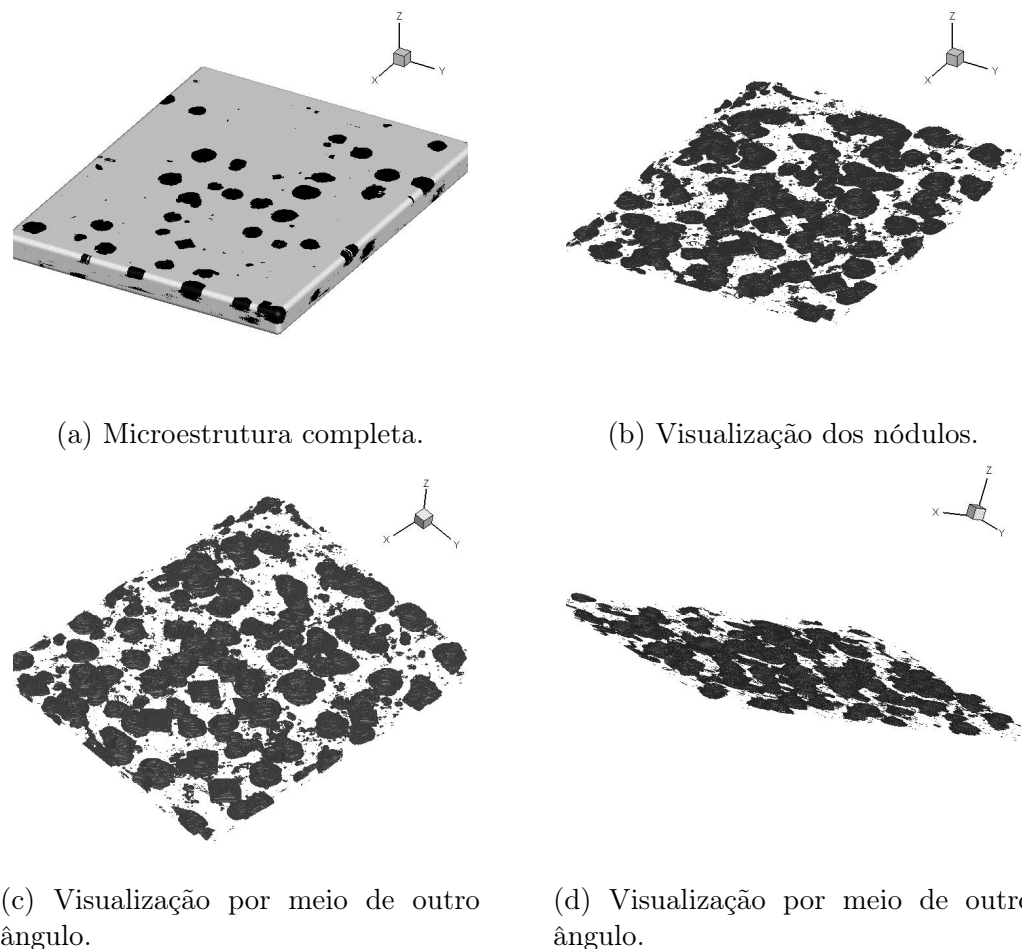


Figura 5.5: Reconstrução 3D de Ferro Fundido Nodular com 57 seções bidimensionais. Visualização da reconstrução feita através do Tecplot 360<sup>®</sup>.

Fonte: Autoria própria (2019).

Ao olhar a Tabela 5.2 é possível ver informações sobre a reconstrução tridimensional apresentada na Figura 5.5, tais como a resolução das imagens, o tempo computacional<sup>2</sup> gasto aproximado para gerar o arquivo de saída através do código computacional desenvolvido e o tamanho final do arquivo de saída com extensão \*.dat.

	Dimensões	Tempo computacional	Tamanho do arquivo
<b>57 seções</b>	1280 x 1024	13.09 minutos	2,18 GB

Tabela 5.2: Informações da reconstrução 3D apresentada na Figura 5.5.

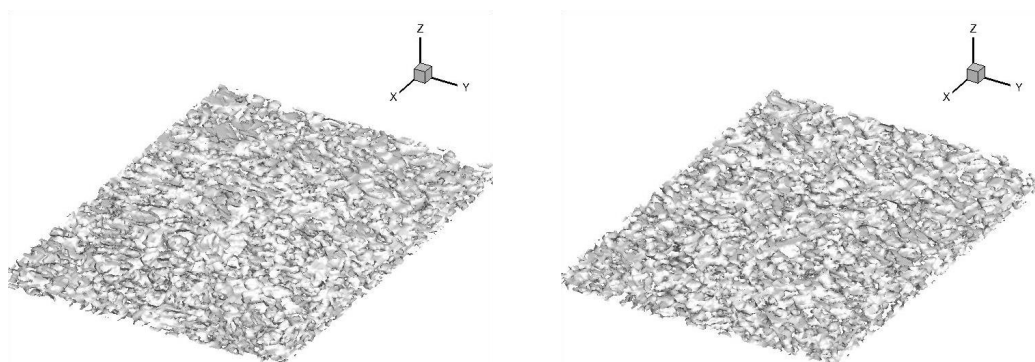
<sup>2</sup>As informações de tempo computacional presentes nas tabelas não incluem o tempo computacional gasto na abertura do arquivo de saída através de ferramentas gráficas.

Por um lado, utilizar mais seções bidimensionais, para fazer uma reconstrução 3D, traz uma grande vantagem: o domínio computacional ganha mais detalhes e informações que podem vir a ser úteis e/ou necessárias. Por outro lado, utilizar muitas seções bidimensionais acarreta a desvantagem de se ter maior gasto em tempo computacional, tanto para gerar o arquivo de saída quanto para abri-lo em um das ferramentas gráficas sugeridas neste trabalho.

## 5.2 Aço Inoxidável Duplex

A seguir, foram apresentados resultados computacionais com o material Aço Inoxidável Duplex.

Semelhantemente, foram feitos testes primários com o Aço Inoxidável Duplex, nas mesmas condições em que foram feitos para o Ferro Fundido Nodular, na seção 5.1 deste capítulo. Os resultados obtidos podem ser vistos nas Figuras 5.6, 5.7 e 5.8, onde, novamente, foram utilizados três softwares de visualização gráfica: Tecplot 360<sup>®</sup>, do ParaView e do VisIt, respectivamente. A reconstrução 3D apresentada nas Figuras 5.6a, 5.7a e 5.8a possuem 10 seções bidimensionais, selecionadas a cada 10 imagens originais obtidas por [11]. A reconstrução 3D apresentada nas Figuras 5.6b, 5.7b e 5.8b possuem 8 seções bidimensionais, selecionadas a cada 12 imagens originais obtidas por [11].



(a) 10 seções bidimensionais.

(b) 8 seções bidimensionais.

Figura 5.6: Reconstrução 3D do Aço Inoxidável Duplex visualizada pelo Tecplot 360<sup>®</sup>.  
Fonte: Autoria própria (2019).

Os dados inseridos como dados de entrada, em cada um dos três softwares utilizados para visualização, são iguais. Mas, com relação as ferramentas gráficas utilizadas para visualização das reconstruções 3D, pode-se observar que a visualização dos resultados

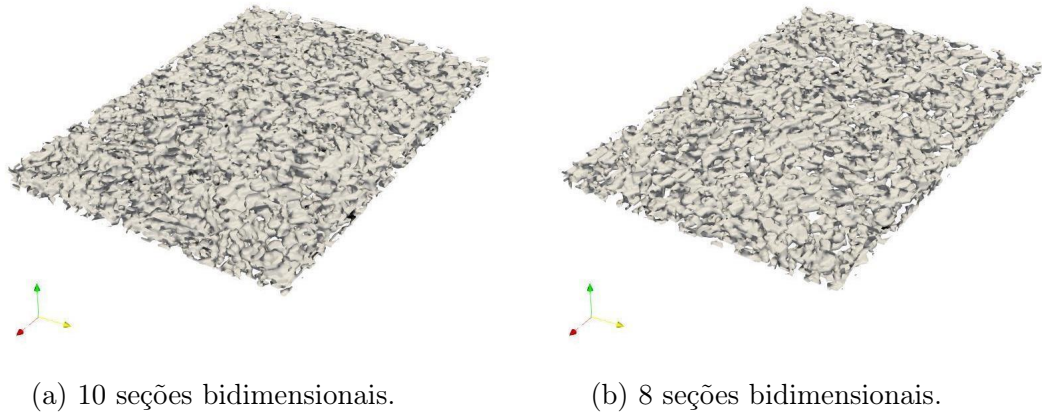


Figura 5.7: Reconstrução 3D do Aço Inoxidável Duplex visualizada pelo ParaView.  
Fonte: Autoria própria (2019).

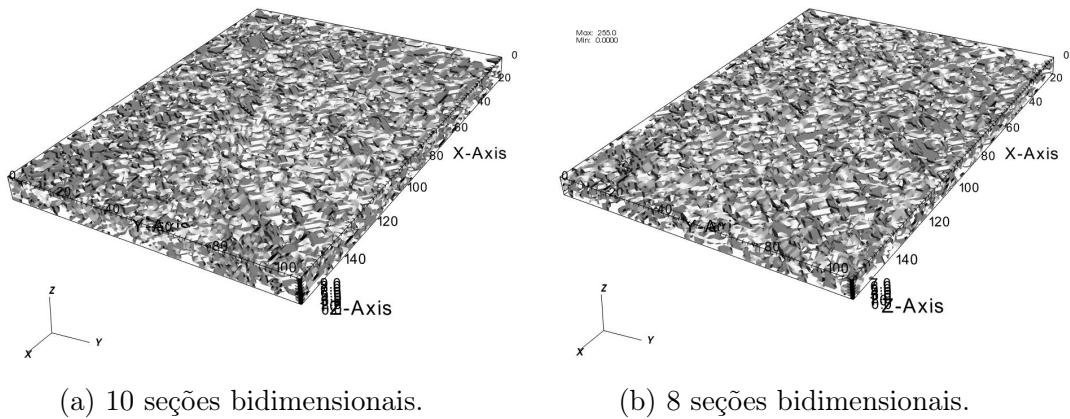


Figura 5.8: Reconstrução 3D do Aço Inoxidável Duplex visualizada pelo VisIt.  
Fonte: Autoria própria (2019).

apresentados nas Figuras 5.6, 5.7 e 5.8 são semelhantes entre si, pois cada ferramenta utiliza um tipo de algoritmo para renderização.

As informações sobre a resolução das imagens utilizadas na reconstrução tridimensional apresentada nas Figuras 5.6, 5.7 e 5.8, o tempo computacional <sup>3</sup> gasto para gerar o arquivo de saída através do código computacional desenvolvido e o tamanho final do arquivo de saída com extensão \*.dat podem ser vistos na Tabela 5.3.

	<b>Dimensões</b>	<b>Tempo computacional</b>	<b>Tamanho do arquivo</b>
<b>10 seções</b>	150 x 112	2.05 segundos	3,58 MB
<b>8 seções</b>	150 x 112	1.72 segundos	2,81 MB

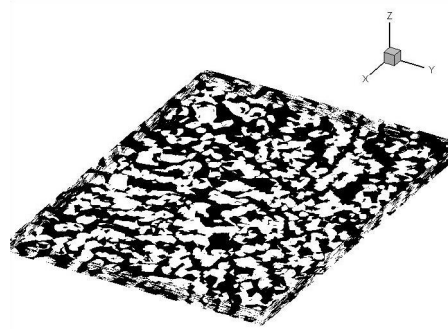
Tabela 5.3: Informações da reconstrução 3D apresentada nas Figuras 5.6, 5.7 e 5.8.

<sup>3</sup>As informações de tempo computacional presentes nas tabelas não incluem o tempo computacional gasto na abertura do arquivo de saída através de ferramentas gráficas.

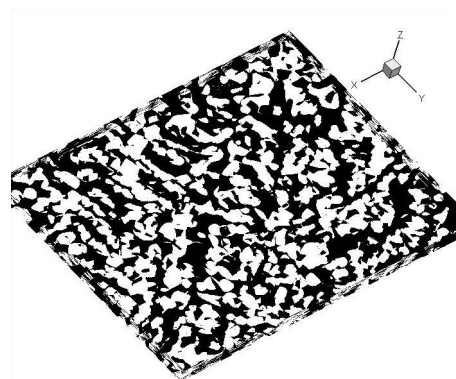
Assim como na seção 5.1, outros testes com Aço Inoxidável Duplex foram feitos com seções bidimensionais de resolução 1280 x 958, conforme os resultados apresentados na Figura 5.9, onde também utilizou-se todas as 57 seções bidimensionais obtidas por [11]. Do mesmo modo, todas as imagens passaram pelo processo de binarização.



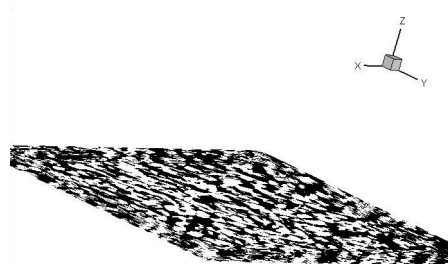
(a) Microestrutura completa.



(b) Visualização de uma fase.



(c) Visualização por meio de outro ângulo.



(d) Visualização por meio de outro ângulo.

Figura 5.9: Reconstrução 3D de Aço Inoxidável Duplex com 57 seções bidimensionais. Visualização da reconstrução feita através do Tecplot 360<sup>®</sup>.

Fonte: Autoria própria (2019).

Ao olhar a Tabela 5.4 é possível ver informações sobre a reconstrução tridimensional apresentada na Figura 5.9, tais como a resolução das imagens, o tempo computacional<sup>4</sup> gasto aproximado para gerar o arquivo de saída através do código computacional desenvolvido e o tamanho final do arquivo de saída com extensão \*.dat.

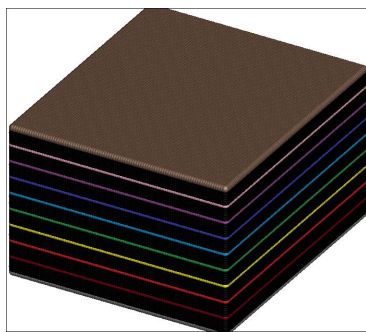
	<b>Dimensões</b>	<b>Tempo computacional</b>	<b>Tamanho do arquivo</b>
<b>57 seções</b>	1280 x 958	12.45 minutos	1,65 GB

Tabela 5.4: Informações da reconstrução 3D apresentada na Figura 5.9.

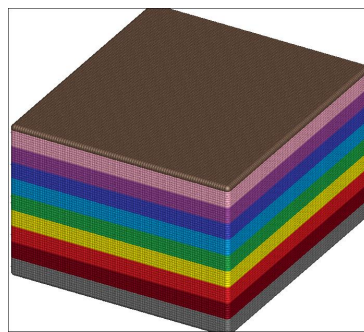
<sup>4</sup>As informações de tempo computacional presentes nas tabelas não incluem o tempo computacional gasto na abertura do arquivo de saída através de ferramentas gráficas.

## 5.3 Comparação entre as Técnicas de Repetição de Planos e Média das Cores

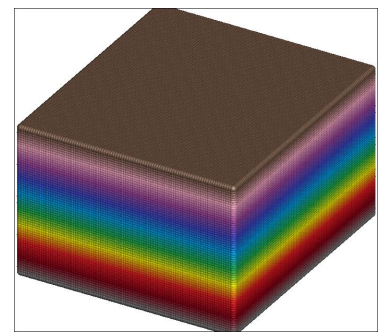
Testes foram feitos com cores sólidas: cinza, vinho, vermelho, amarelo, verde, azul claro, azul escuro, roxo, rosa e marrom escolhidas ao acaso, porém de modo que o primeiro plano principal possui todos os pixels com o mesmo tom de cinza  $\text{rgb}(127, 127, 127)$ , o segundo plano principal possui o mesmo tom de vinho  $\text{rgb}(136, 0, 21)$  em todos seus pixels e assim, o raciocínio se propaga, analogamente, para os outros oito planos principais restantes. Optou-se por fazer uma reconstrução tridimensional nessas condições e obteve-se os resultados apresentados na Figura 5.10 apenas como teste inicial. É possível observar que com a mesma quantidade de planos principais (dez) e de planos intermediários (seis), obteve-se diferentes resultados com as técnicas propostas.



(a) Seis planos vazios preenchidos pela cor preta cuja as coordenadas no sistema RGB é a tripla  $\text{rgb}(0,0,0)$ .



(b) Seis planos intermediários preenchidos com repetição.



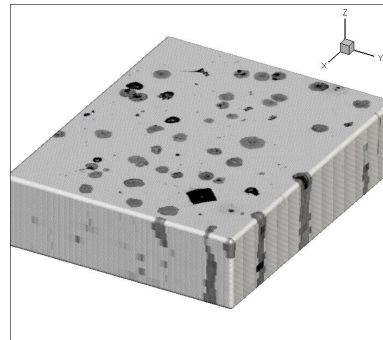
(c) Seis planos intermediários preenchidos com a técnica da média de cores.

Figura 5.10: Teste das abordagens implementadas no código desenvolvido, utilizando cores sólidas.

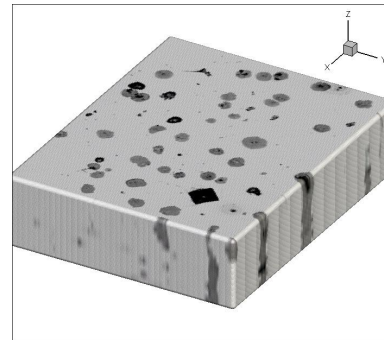
Fonte: Autoria própria (2018).

Ao atentar-se ao teste feito e mostrado na Figura 5.10, pode-se observar que a reconstrução ficou mais suave com a utilização da técnica de obtenção de planos intermediários por meio da média das cores (Figura 5.10c) do que com a utilização da técnica de repetição de planos (Figura 5.10b).

Nas mesmas condições, testes foram feitos com as imagens de Ferro Fundido Nodular e Aço Inoxidável Duplex. Os resultados obtidos, respectivamente, podem ser vistos na Figura 5.11.



(a) Técnica da repetição de planos.



(b) Técnica da média entre as cores.



(c) Técnica da repetição de planos.



(d) Técnica da média entre as cores.

Figura 5.11: Resultados com Ferro Fundido Nodular e Aço Inoxidável Duplex.

Fonte: Autoria própria (2018).

Na Figura 5.12 e na Figura 5.13 a reconstrução da microestrutura completa de Ferro Fundido Nodular é realizada por meio das duas abordagens apresentadas neste trabalho e, após este procedimento, a área de interesse é escolhida.

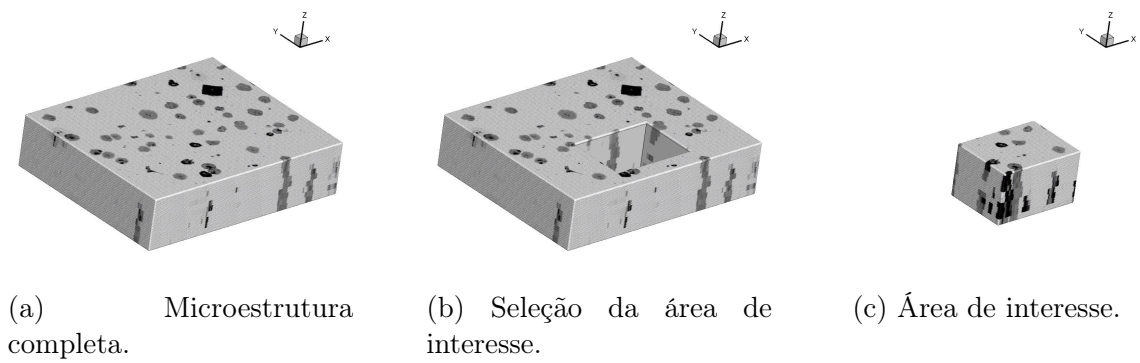


Figura 5.12: Resultado da técnica de repetição de planos adjacentes.  
Fonte: Autoria própria (2018).

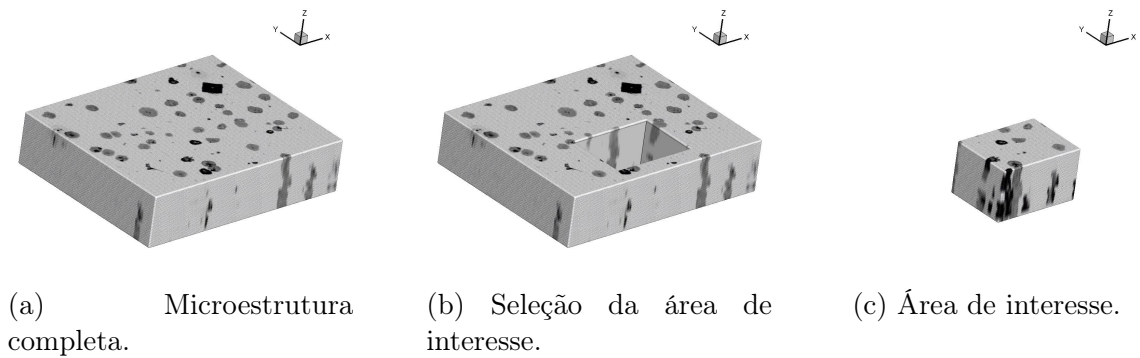


Figura 5.13: Resultado da técnica de obtenção de planos intermediários por meio da médias das cores.

Fonte: Autoria própria (2018).

Através da Figura 5.14, pode-se observar os detalhes ampliados da imagem que mostram a diferença da borda da microestrutura quando utiliza-se as diferentes técnicas de obtenção de planos intermediários.

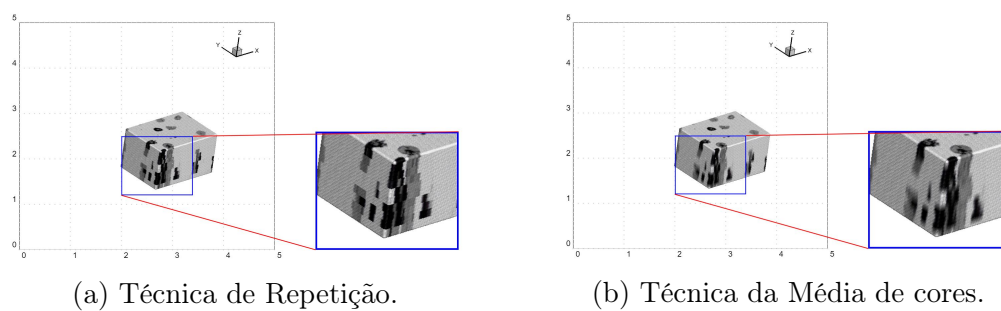
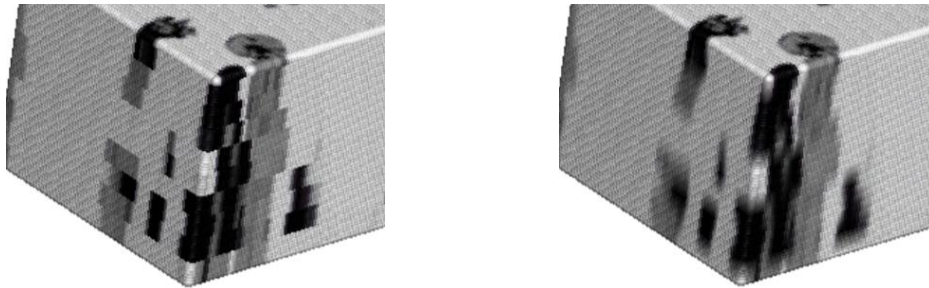


Figura 5.14: Área de interesse ampliada.  
Fonte: Autoria própria (2018).

Uma maior ampliação da área de interesse é exibida na Figura 5.15 para melhorar a exibição da diferença entre as duas abordagens.



(a) Técnica de Repetição.

(b) Técnica da Média de cores.

Figura 5.15: Área de interesse mais ampliada.

Fonte: Autoria própria (2018).

Note que, agora, a reconstrução tridimensional do Ferro Fundido Nodular sugere colunas de grafite, conforme visto nas Figuras 5.11a, 5.11b, 5.12, 5.13 e 5.14. Isso ocorreu devido a enumeração dos planos principais e intermediários. Inicialmente, tem-se dez planos principais e seis planos intermediários a serem preenchidos por uma das duas abordagens supramencionadas. A contagem desses planos foi feita considerando o total de planos utilizados na reconstrução: planos principais + planos intermediários. Isto significa que os dezesseis planos (no total) foram enumerados na sequência. Porém, a contagem de planos deve ser feita apenas nos planos principais para evitar deformação na reconstrução do objeto de interesse.

A solução para arrumar este problema foi enumerar os planos intermediários de modo que a distância entre o primeiro plano principal e o último seja mantida. Desse modo, a reconstrução não aparenta esticamento nas esferas de grafite. Os resultados apresentados na Figura 5.16 testemunham que a enumeração dos planos é de crucial importância para preservar a estrutura e a configuração da reconstrução 3D de uma microestrutura, por exemplo.

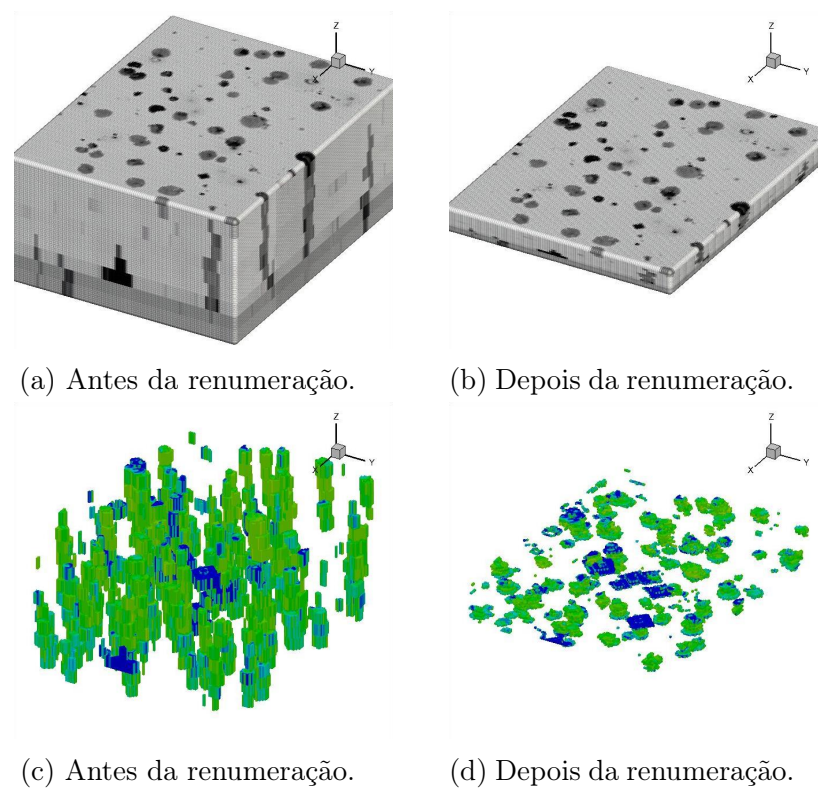


Figura 5.16: 10 planos principais e 6 intermediários preenchidos com a Técnica da Repetição.

Fonte: Autoria própria (2019).

## 5.4 Outras Possíveis Aplicações

Segundo [63], uma microestrutura 3D pode ser processada usando várias técnicas, incluindo tomografia computadorizada de raios X, ressonância magnética e seccionamento serial. Em particular, esta seção foi reservada para mostrar resultados de testes de reconstrução 3D de um abdômen com imagens 2D obtidas por meio de tomografia computadorizada.

### 5.4.1 Tomografia Computadorizada

A *Tomografia Computadorizada* ou *Tomografia Computadorizada* (TC) pode ser definida como um exame (ou método diagnóstico) que produz imagens dos órgãos internos do corpo humano e utiliza a radiação x para obtê-las. Cada uma das imagens produzidas por esta técnica representa uma seção ou “fatia” da estrutura de interesse do conteúdo corporal em um dos três planos do espaço tridimensional. As imagens das estruturas são geradas

como se nelas tivessem sido realizado uma série de cortes finos [32, 35]. Posteriormente, são reconstruídas em um plano bidimensional na tela do computador.

A reconstrução tridimensional apresentada nas Figuras 5.19, 5.20, 5.21 e 5.22 foram obtidas a partir 341 imagens de um exame familiar de tomografia computadorizada. Algumas seções, em série, podem ser vistas na Figura 5.17, onde o fundo das imagens está na cor preta.

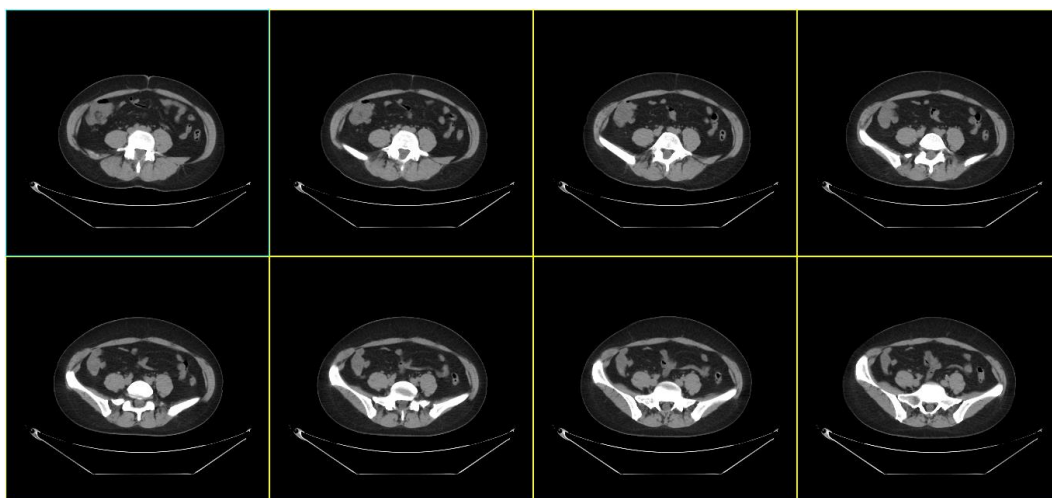


Figura 5.17: Imagens de TC em série.  
Fonte: Arquivo de exame familiar (2017).

A escolha da quantidade de imagens contidas na Figura 5.17 e a escolhas das imagens em si, foi uma tentativa de aproximação da Figura 5.26 do artigo [67].

Outras seções, intercaladas, podem ser vistas na Figura 5.18, onde o fundo das imagens está na cor branca e a resolução de cada sub-figura é de 520x380.

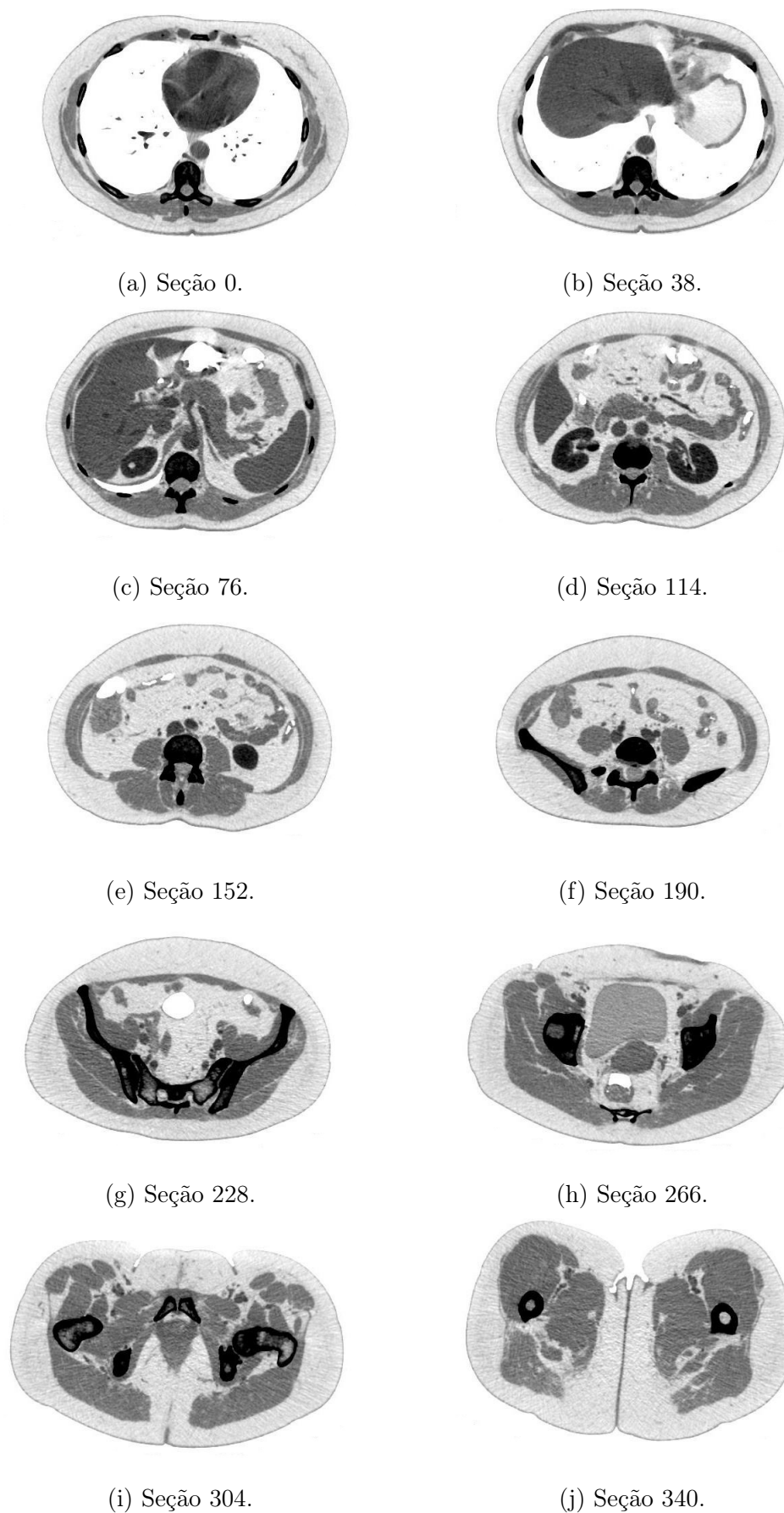


Figura 5.18: Imagens de TC.  
Fonte: Arquivo de exame familiar (2017).

A ferramenta computacional principal levou, aproximadamente, 22 minutos para gerar o arquivo de saída, de tamanho 2,04 GB, com 341 imagens principais e suas respectivas informações. Porém, o tempo gasto para abrir/carregar o arquivo de saída para visualização não está incluso nesses mesmos 22 minutos. A visualização da reconstrução 3D apresentada nas Figuras 5.19 e 5.20 foi realizada através da ferramenta de visualização gráfica Tecplot 360<sup>®</sup>.

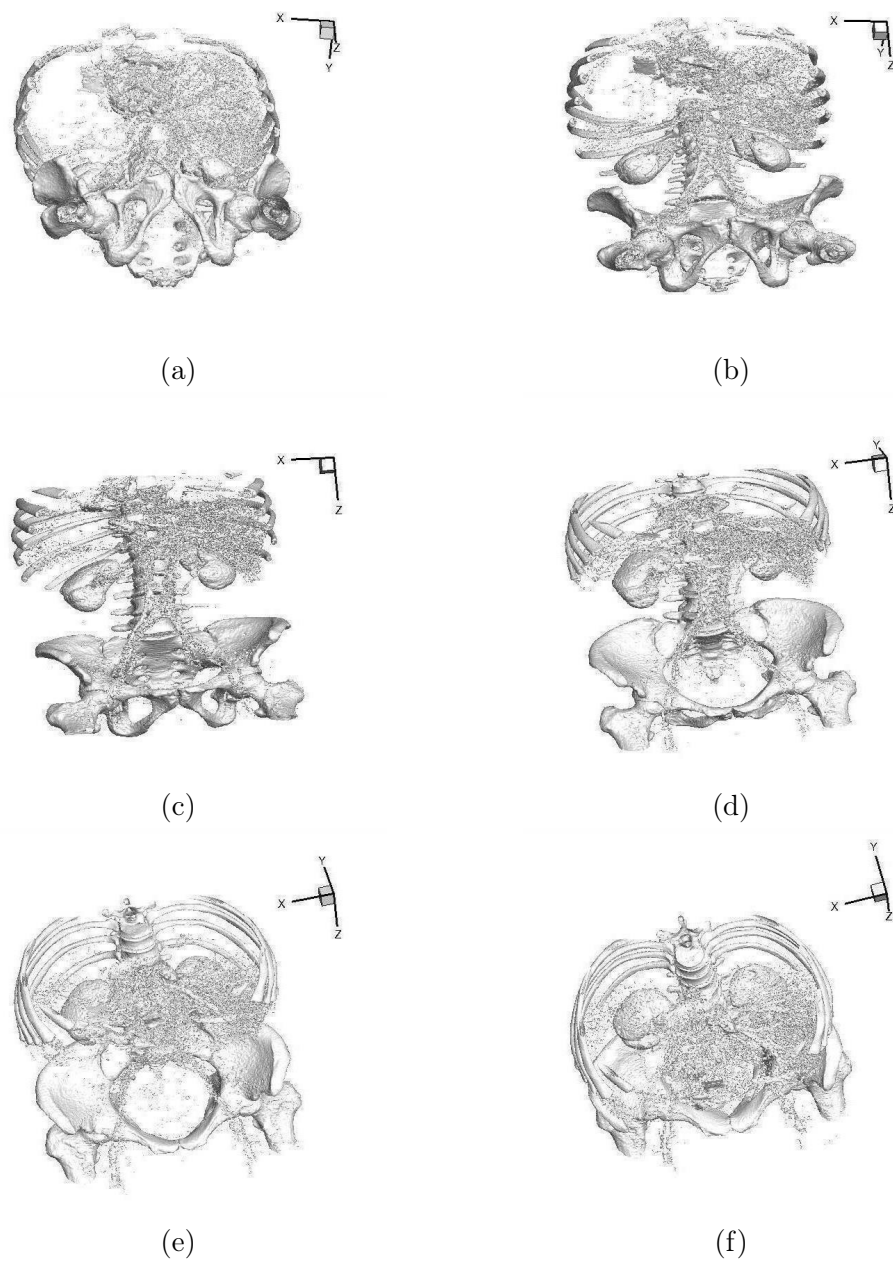


Figura 5.19: Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo  $x$  e visualizadas através do Tecplot 360<sup>®</sup>.

Fonte: Autoria própria (2019).

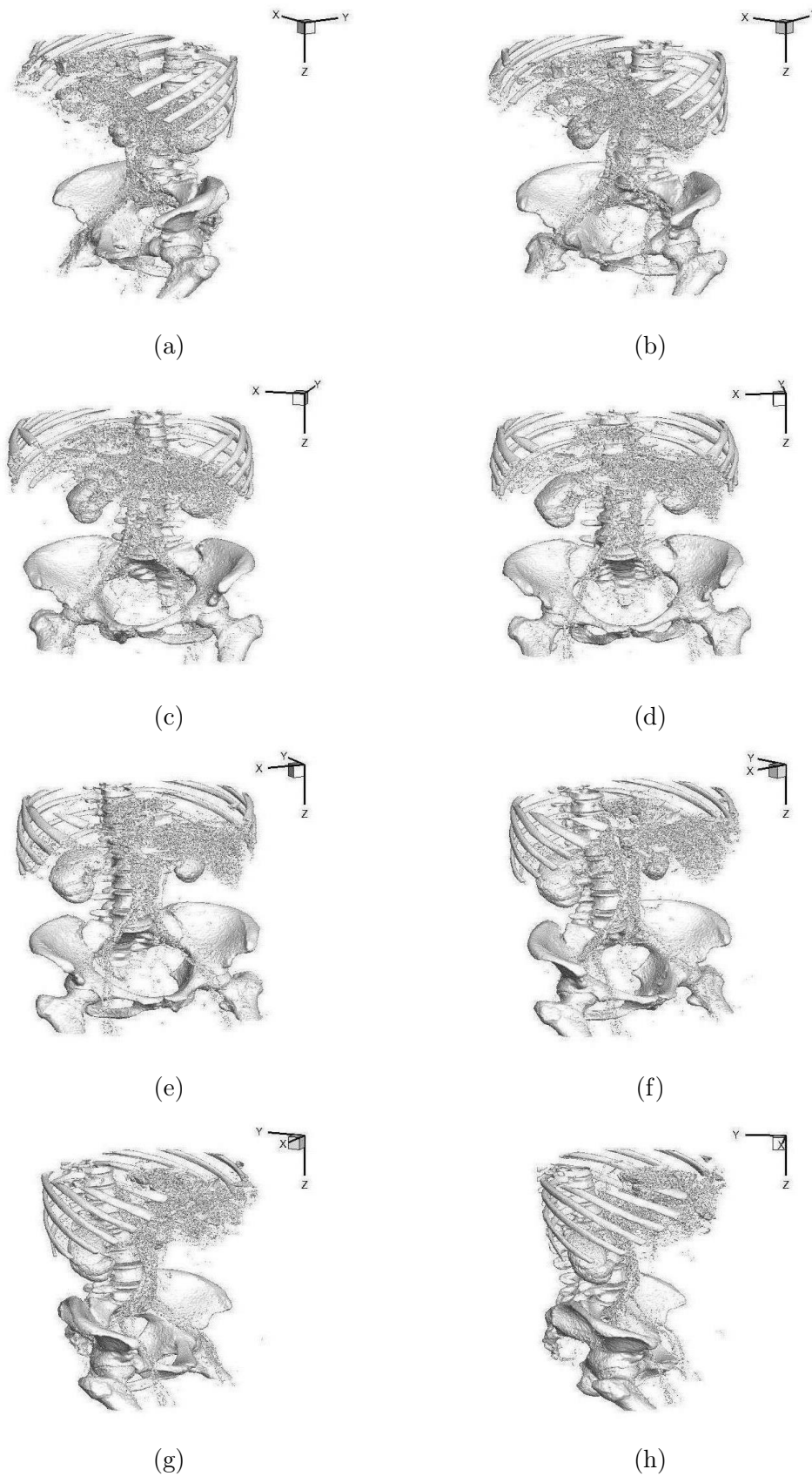


Figura 5.20: Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo  $z$  e visualizadas através do Tecplot 360<sup>®</sup>.

Fonte: Autoria própria (2019).

A visualização da reconstrução 3D apresentada nas Figuras 5.21 e 5.22 foi realizada através da ferramenta de visualização gráfica ParaView.

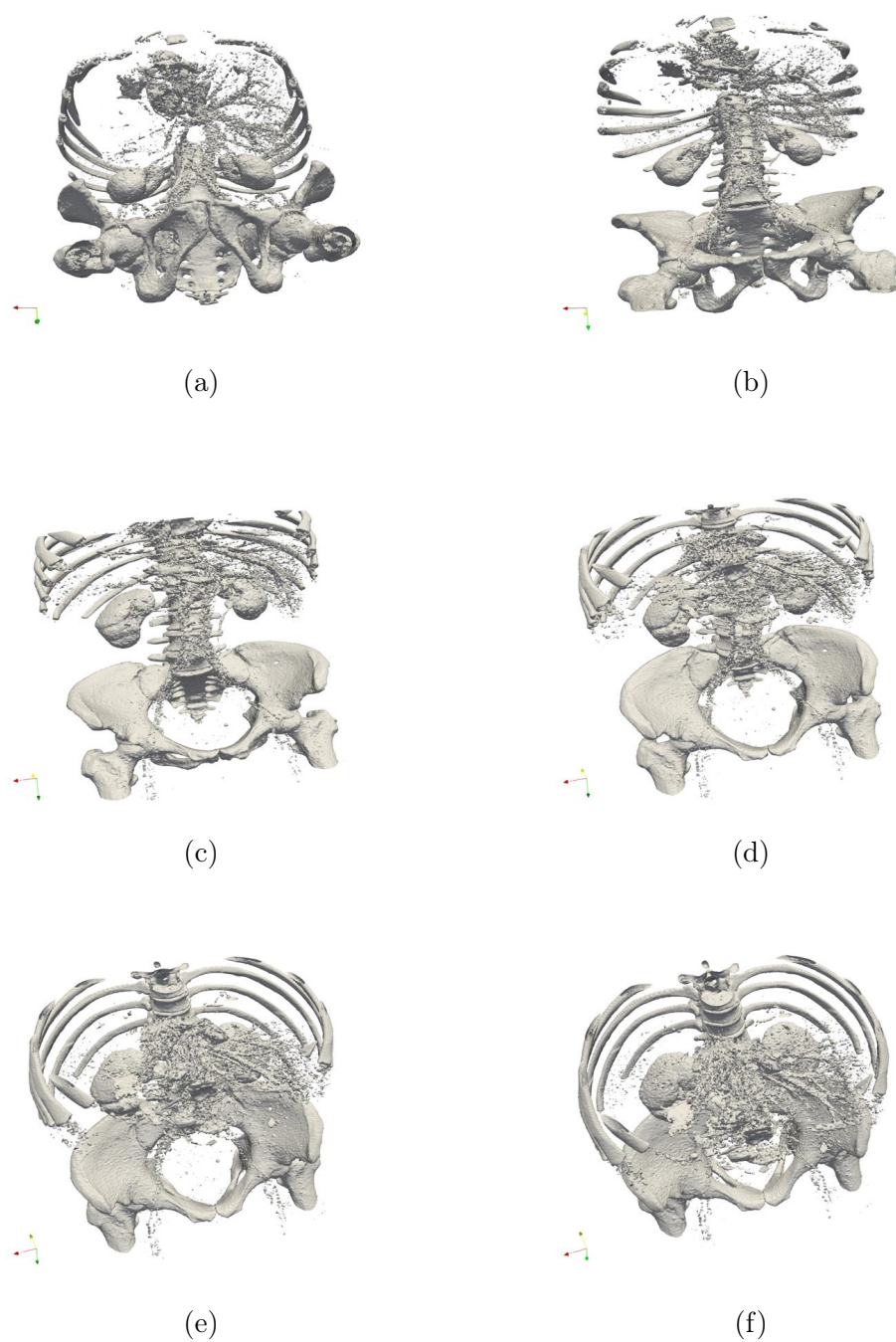


Figura 5.21: Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo  $x$  e visualizadas através do ParaView.

Fonte: Autoria própria (2019).

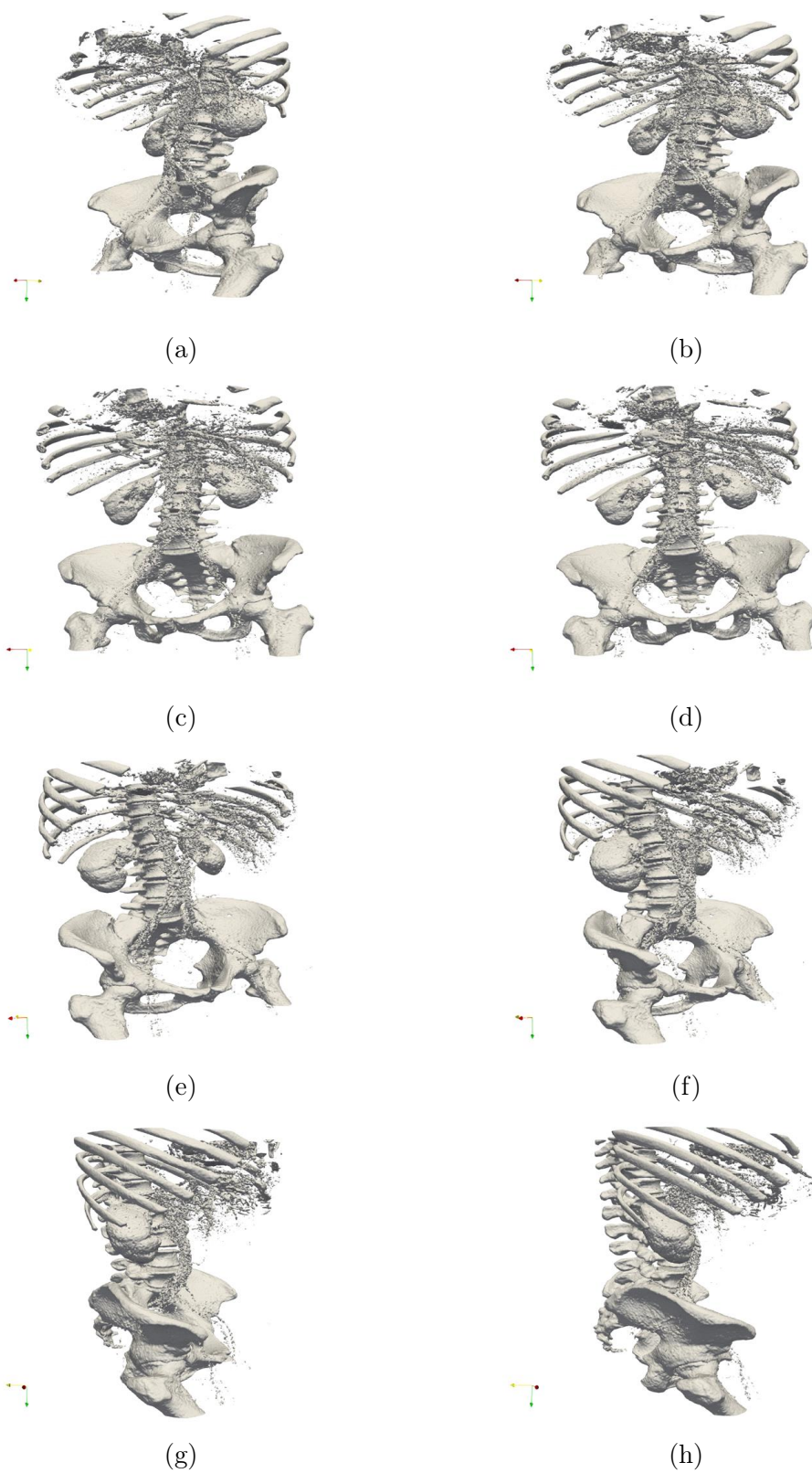


Figura 5.22: Reconstrução 3D a partir de imagens 2D de TC. Figuras rotacionadas com relação ao eixo  $z$  e visualizadas através do ParaView.

Fonte: Autoria própria (2019).

Mediante ao zoom da imagem apresentada na Figura 5.23, é possível ver melhor os rins, as cinco vértebras da região lombar e algumas vértebras torácicas. Além disso, pode-se ver também uma parte da bacia óssea.

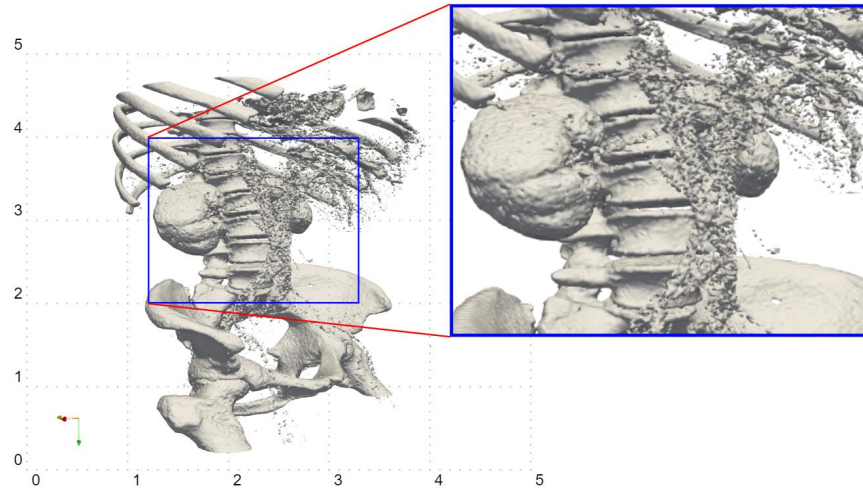


Figura 5.23: Aproximação da imagem para visualização dos detalhes.  
Fonte: Autoria própria (2019).

Ao observar a Figura 5.24, é possível ver os detalhes laterais do lado direito da coluna vertebral. No zoom dessa reconstrução 3D pode-se ver duas das quatro curvaturas da coluna vertebral: Curvatura Lombar e Curvatura Sacral. Mais abaixo pode-se enxergar o Sacro e o Cóccix. Ademais, nota-se rugosidade do tecido ósseo.

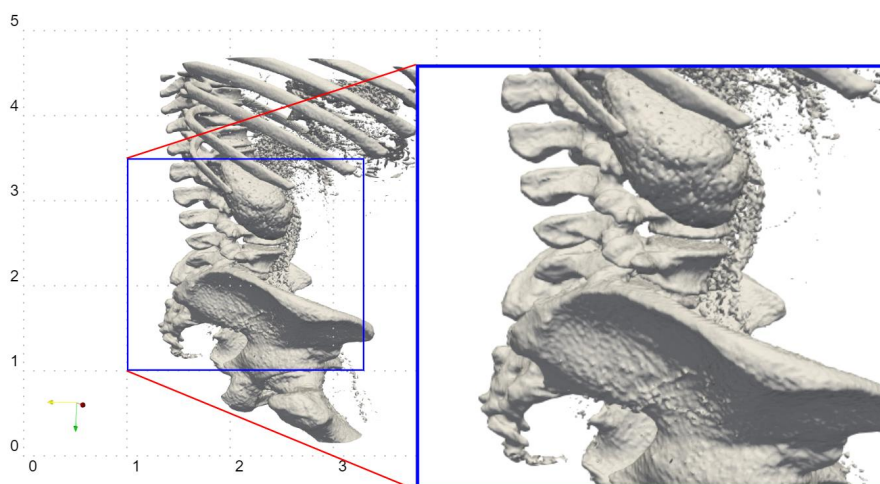


Figura 5.24: Aproximação da imagem para visualização dos detalhes.  
Fonte: Autoria própria (2019).

#### 5.4.1.1 Validação do Código

Optou-se por fazer a validação do código computacional, de autoria própria, por meio do software ImageJ. As imagens bidimensionais usadas como dados de entrada pro ImageJ são as mesmas imagens vistas nas Figuras 5.17 e 5.18. Do mesmo modo, para esta reconstrução tridimensional, utilizou-se 341 imagens de TC. Como pode ser visto na Figura 5.25, os resultados apresentados pelo ImageJ são iguais aos resultados de autoria própria apresentados nas Figuras 5.19, 5.20, 5.21 e 5.21. Além disso, a visualização do resultado é semelhante a visualização observada nas três ferramentas de visualização gráfica utilizadas no trabalho.

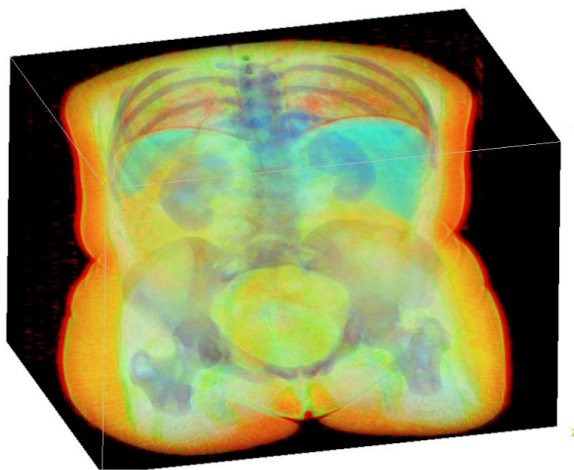


Figura 5.25: Visualização do resultado através do software ImageJ.

Fonte: Autoria própria (2019).

#### 5.4.1.2 Estudo Comparativo

Visando reforçar a validação do código desenvolvido, a seguir foi feito um estudo comparativo com o resultado de outros autores. Foi visto que, a partir das imagens bidimensionais de TC, apresentadas na Figura 5.26, os autores de [67] fizeram uma reconstrução 3D (Figura 5.27) da mesma região corporal que foi feita no presente trabalho, vide Figuras 5.19, 5.20, 5.21 e 5.22 anteriores. Retrata-se o caso de um doente do sexo masculino de 71 anos, hipertenso, fumador, com história de Enfarte Agudo do Miocárdio dois meses antes.

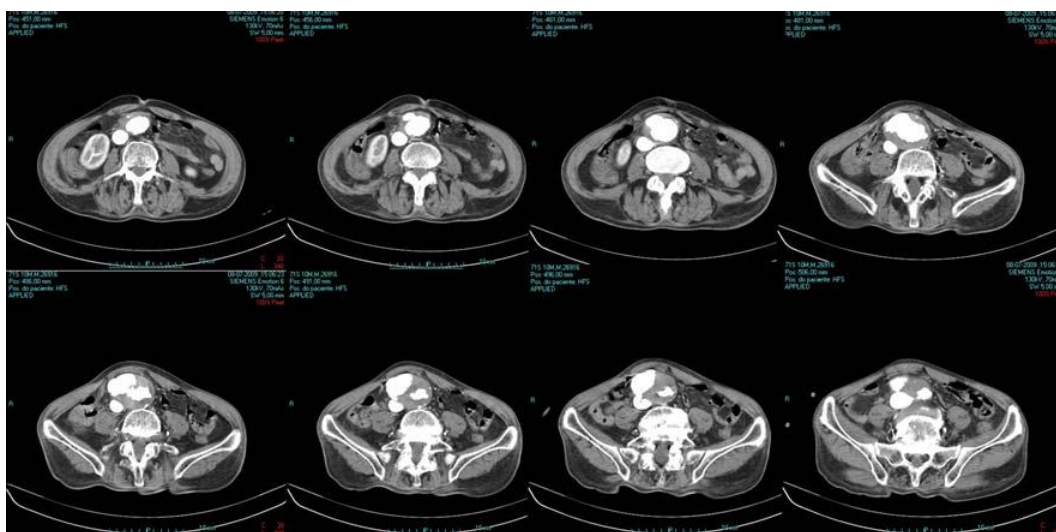


Figura 5.26: Imagens de TC toraco-abdomino-pélvica.  
Fonte: [67].

Note que o resultado da reconstrução tridimensional obtida por [67], na Figura 5.27, é semelhante aos resultados das reconstruções tridimensionais exibidas nas Figuras 5.19, 5.20, 5.21 e 5.22.



Figura 5.27: Reconstrução 3D feita a partir de imagens de TC toraco-abdomino-pélvica.  
Fonte: [67].

# Capítulo 6

## Conclusões e Trabalhos Futuros

### 6.1 Conclusões

O objetivo geral deste trabalho foi cumprido. Isto significa que uma primeira versão de uma ferramenta computacional que cria domínios computacionais discretos em 3D a partir de imagens 2D foi desenvolvida. Notou-se que os resultados, gerados por esta ferramenta computacional e visualizados por softwares de visualização gráfica, possuem fidelidade a ambientes ou volumes de controle contínuos, extraídos da natureza. Efetivamente, foi possível utilizar imagens bidimensionais obtidas pela técnica de seccionamento em série e imagens bidimensionais de tomografia computadorizada.

Além do objetivo geral, os objetivos específicos também foram atendidos. Naturalmente, conseguiu-se realizar reconstruções 3D a partir de uma série de imagens 2D utilizando o código computacional desenvolvido. Foi evidente que a escolha de uma linguagem de programação que possibilita empregar o paradigma da Programação Orientada a Objetos permitiu que códigos de outros integrantes do grupo NMM<sup>1</sup> fossem reutilizados para dar sequência ao desenvolvimento deste software. Logo, isso permitirá que outros autores prossigam neste trabalho com maior facilidade quando for preciso incorporar novas funcionalidades ao programa. Posteriormente, foi visto que, com a aplicação de técnicas computacionais, o problema de determinação de valor entre dois planos discretos foi reparado através das duas abordagens descritas no Capítulo 4. E, com isso, verificou-se que a reconstrução de uma microestrutura fica mais suave com a utilização da técnica de obtenção de planos intermediários por meio da média das cores do que com a utilização da técnica de repetição de planos. Ademais, foi possível utilizar três ferramentas gráficas distintas para visualizar o domínio gerado pelo código computacional:

---

<sup>1</sup>Núcleo de Modelamento Microestrutural.

Tecplot 360<sup>®</sup>, ParaView e VisIt. Ainda com base nos resultados obtidos, a malha do domínio computacional gerado está pronta para ser trabalhada após a reconstrução e já pode ser aplicada em simulações computacionais tais como Transformação de Fase, Transferência de Calor e Deformação a partir dos métodos Autômato Celular, Cone Causal, Monte Carlo, Elementos Finitos, Diferenças Finitas, dentre outros. De resto, a análise dos resultados obtidos foi satisfatória.

A necessidade de ter um programa de baixo custo, escrito para alto desempenho e que possibilite continuar trabalhando com a malha reconstruída, foi suprida. Além disso, a verificação do código computacional foi feita a partir de uma comparação entre a visualização dos resultados obtidos pelo código computacional desenvolvido e a visualização dos resultados obtidos pelo software ImageJ. Outrossim, foram feitas comparações com os trabalhos anteriores, de outros autores, e também obteve-se sucesso. Pôde-se observar também que a resolução das imagens bidimensionais e a quantidade de planos utilizados na reconstrução 3D influenciam nos resultados.

Os resultados deste trabalho mostraram a possibilidade de modelar o processo de reconstrução 3D através da técnica de seccionamento em série, assim como por meio de imagens de tomografia computadorizada.

## 6.2 Trabalhos Futuros

A seguir, encontram-se sugestões de trabalhos futuros que podem ser feitos em uma próxima etapa desta dissertação:

- Implementar a técnica de interpolação entre dois planos e comparar com as outras duas abordagens descritas no Capítulo 4;
- Incluir código que binariza as imagens dentro do código principal deste trabalho;
- Paralelizar o código computacional que foi desenvolvido, pois foi implementado com o intuito de deixá-lo paralelizável devido o tamanho da malha que se pretende gerar a cada entrada de dado escolhida para reconstruir determinado objeto;
- Implementar uma interface gráfica para facilitar o uso desta ferramenta computacional por outros usuários;
- Tentar melhorar o modo de implementação do código computacional adequando-o melhor ao paradigma da Programação Orientada a Objetos, visto que existem algumas dependências de dados entre as classes.
- Utilizar as malhas geradas pelo programa e empregá-las em modelos numéricos como Volumes Finitos, Diferenças Finitas, Elementos Finitos, Cone Causal, Autômato Celular, Monte Carlo, dentre outros.

# Capítulo 7

## Produções Acadêmicas

### 7.1 Resumos Aceitos em Simpósios e Encontros

- **GRAÇA, A. B. R. A.**; ASSIS, W. L. S.; NEVES, T. A.; NUNEZ, Y. R.; RIOS, P. R.; FERREIRA, J. L. G.; *Estudo para Criação de Domínios Discretos Tridimensionais a partir de Imagens Digitais*. IV SIMMA - Simpósio de Matemática da Região Sul Fluminense, UFF. 2018.
- **GRAÇA, A. B. R. A.**; ASSIS, W. L. S.; NEVES, T. A.; NUNEZ, Y. R.; RIOS, P. R.; FERREIRA, J. L. G. *Implementação De Software Para Reconstrução 3D Utilizando Imagens 2D De Microestruturas Obtidas Através Técnica De Seccionamento Em Série*. XII Encontro Acadêmico de Modelagem Computacional, LNCC. 2018.
- **GRAÇA, A. B. R. A.**; ASSIS, W. L. S.; NEVES, T. A.; NUNEZ, Y. R.; RIOS, P. R.; FERREIRA, J. L. G.; PRAZERES, T. F.; OLIVEIRA, D. *Simulação Do Processo De Reconstrução 3D De Um Ferro Nodular Via Open Source*. XII Encontro Acadêmico de Modelagem Computacional, LNCC. 2018.

## 7.2 Resumos Publicados em Anais de Congressos

- ALVES, A. L. M.; ASSIS, W. L. S.; PRAZERES, T. F.; **Andrade A. B. R.**; COSTA, M. F. B.; FONSECA, G. D.; FERREIRA, J. L. G.; OLIVEIRA, D. *Study of the Influence of the Distance Between Layers in the 3D Reconstruction Technique via Serial Section*. XVI Brazilian MRS Meeting, Gramado-RS. 2017. p. 1473-1473.
- **GRAÇA, A. B. R. A.**; ASSIS, W. L. S.; NEVES, T. A.; NUNEZ, Y. R.; RIOS, P. R.; PRAZERES, T. F.; FERREIRA, J. L. G. *Desenvolvimento de Ferramenta Computacional para Criação de Domínios Discretos em 3D*. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics. v.6, n. 2. CNMAC 2018.
- PRAZERES, T. F.; ASSIS, W. L. S.; **GRAÇA, A. B. R. A.** *Influência da Resolução da Imagem na Reconstrução Microestrutural em 3D por Seccionamento em Série*. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics. v.6, n. 2. CNMAC 2018.

## 7.3 Artigo Completo Publicado

- **GRAÇA, A. B. R. A.**; ASSIS, W. L. S.; NEVES, T. A.; NUNEZ, Y. R.; RIOS, P. R.; PRAZERES, T. F.; FERREIRA, J. L. G.; BAÍA, D. M. *Implementação de software para reconstrução 3D utilizando imagens 2D de microestruturas obtidas através da técnica de seccionamento em série*. XII Encontro Acadêmico de Modelagem Computacional - EAMC 2019. Laboratório Nacional de Computação Científica - LNCC/MCTIC.

## 7.4 Capítulo de Livro

- ASSIS, W. L. S.; PRAZERES, T. F.; **GRAÇA, A. B. R. A.** *Study of the Influence of the Distance Between Layers in the 3D Reconstruction Technique via Serial Section*. Convite da Editora Atena. 2019.

# 7.5 Pôsteres Apresentados

**STUDY OF THE INFLUENCE OF THE DISTANCE BETWEEN LAYERS IN THE RECONSTRUCTION TECHNIQUE BY SERIAL SECTION**  
 Ana Beatriz R. de Andrade Graça, Jonathan Luis G. Ferreira, Wesley Luiz da Silva Assis, Yossiel Rodriguez Núñez, Tiago Araújo Neves, Paulo Rangeli Rios

**1. INTRODUCTION**  
 The serial section technique is a powerful tool for the reconstruction of 3D structures from 2D images. This study aims to investigate the influence of the distance between layers on the reconstruction quality. The results show that the distance between layers significantly affects the accuracy of the reconstruction, with smaller distances leading to more detailed and accurate 3D models.

**2. BIBLIOGRAPHIC REVIEW**  
 The literature on serial section techniques is extensive, covering various applications in materials science, biology, and geology. This review highlights the challenges associated with this technique, such as image registration and the choice of reconstruction algorithms.

**3. COMPUTATIONAL PROCEDURE**  
 The computational procedure involves the acquisition of serial sections, image registration, and the application of a reconstruction algorithm. The distance between layers is varied to assess its impact on the final 3D reconstruction.

**4. RESULTS AND DISCUSSION**  
 The results demonstrate that the distance between layers is a critical parameter in the reconstruction process. Smaller distances result in higher resolution and more accurate 3D reconstructions, but also increase the computational cost and the risk of artifacts.

**5. CONCLUSIONS**  
 The study concludes that the distance between layers has a significant influence on the quality of the 3D reconstruction. Optimizing this parameter is essential for achieving accurate and detailed results.

(a) SBPMat 2017.

**Estudo para Criação de Domínios Discretos Tridimensionais a partir de Imagens Digitais**  
 Ana Beatriz R. de Andrade Graça, Jonathan Luis G. Ferreira, Wesley Luiz da Silva Assis, Yossiel Rodriguez Núñez, Tiago Araújo Neves, Paulo Rangeli Rios

**Introdução**  
 A criação de domínios discretos tridimensionais a partir de imagens digitais é um processo essencial para a simulação de materiais. Este estudo investiga a influência da resolução da imagem na reconstrução microestrutural em 3D.

**Objetivos**  
 O objetivo principal é avaliar a influência da resolução da imagem na qualidade da reconstrução microestrutural em 3D, considerando diferentes parâmetros de processamento.

**Metodologia**  
 A metodologia envolve a aquisição de imagens digitais, o processamento das mesmas e a aplicação de algoritmos de reconstrução para gerar domínios discretos tridimensionais.

**Resultados**  
 Os resultados mostram que a resolução da imagem tem um impacto significativo na qualidade da reconstrução, com resoluções mais altas resultando em domínios mais detalhados e precisos.

**Conclusões**  
 Conclui-se que a escolha da resolução da imagem é crucial para a obtenção de domínios discretos tridimensionais de alta qualidade, sendo necessário otimizar este parâmetro para cada aplicação específica.

(b) SIMMA 2017.

**Desenvolvimento de Ferramenta Computacional para Criação de Domínios Discretos em 3D**  
 Ana Beatriz R. de Andrade Graça, Wesley Luiz da Silva Assis, Jonathan Luis G. Ferreira, Yossiel Rodriguez Núñez, Tiago Araújo Neves, Paulo Rangeli Rios

**Introdução**  
 O desenvolvimento de uma ferramenta computacional para a criação de domínios discretos em 3D é fundamental para a simulação de materiais. Esta ferramenta permite a geração automática de domínios discretos a partir de imagens digitais.

**Objetivos**  
 O objetivo principal é desenvolver uma ferramenta computacional que permita a criação automática de domínios discretos em 3D a partir de imagens digitais, otimizando o processo de simulação.

**Metodologia**  
 A metodologia envolve a aquisição de imagens digitais, o processamento das mesmas e a aplicação de algoritmos de reconstrução para gerar domínios discretos em 3D.

**Resultados**  
 Os resultados mostram que a ferramenta desenvolvida é capaz de gerar domínios discretos em 3D de forma automática e eficiente, permitindo a simulação de materiais com maior precisão.

**Conclusões**  
 Conclui-se que a ferramenta desenvolvida é uma solução eficaz para a criação de domínios discretos em 3D, facilitando o processo de simulação de materiais e melhorando a qualidade dos resultados.

(c) CNMAC 2018.

**Influência da Resolução da Imagem na Reconstrução Microestrutural em 3D por Seccionamento em Série**  
 Taitia F. dos Prazeres, Wesley Luiz da Silva Assis, Paulo Rangeli Rios, Ana Beatriz R. de Andrade Graça

**Introdução**  
 A reconstrução microestrutural em 3D por seccionamento em série é uma técnica amplamente utilizada para a análise de materiais. Este estudo investiga a influência da resolução da imagem na qualidade da reconstrução microestrutural em 3D.

**Objetivos**  
 O objetivo principal é avaliar a influência da resolução da imagem na qualidade da reconstrução microestrutural em 3D, considerando diferentes parâmetros de processamento.

**Metodologia**  
 A metodologia envolve a aquisição de imagens digitais, o processamento das mesmas e a aplicação de algoritmos de reconstrução para gerar domínios discretos em 3D.

**Resultados**  
 Os resultados mostram que a resolução da imagem tem um impacto significativo na qualidade da reconstrução, com resoluções mais altas resultando em domínios mais detalhados e precisos.

**Conclusões**  
 Conclui-se que a escolha da resolução da imagem é crucial para a obtenção de domínios discretos em 3D de alta qualidade, sendo necessário otimizar este parâmetro para cada aplicação específica.

(d) CNMAC 2018.

Figura 7.1: Pôsteres apresentados em congressos e simpósios. Fonte: (a) [5], (b), (c) Autoria própria e (d) [29]

### 7.5.1 Votação de Pôsteres

#### CNMAC 2018

#### Quatro Pôsteres Gerais mais votados pelo público

Sessão	Título	Autores	Votos
ST12	Desenvolvimento de Ferramenta Computacional para Criação de Domínios Discretos em 3D	Ana Beatriz Graça* Wesley Assis Tiago Neves, et. al	74
ST02	Uma Abordagem Fuzzy para Auxiliar no Diagnóstico Médico das Faringotonsilites	Mariana Pissini Magda Peixoto*	18
ST10	Computational Model of a Heart Chamber through Navier-Stokes equations	Laryssa Abdala* Maicon Correa Carlos Mady	18
ST17	Uma Experiência de Sala de Aula Invertida Como Metodologia de Ensino-Aprendizagem de Cálculo I: desafios e possibilidades	Rutyele Moreira*	16

Figura 7.2: Resultado da votação de pôsteres de pesquisa.

Fonte: <<http://www.cnmac.org.br/novo/downloads/9f280520b1e5739019ce5cb1e12f1cac.pdf>>.

# Referências

- [1] ACHARYA, T., RAY, A. K. *Image Processing: Principles and Applications*. Wiley-Interscience, 2005.
- [2] ADACHI, Y., MOROOKA, S., NAKAJIMA, K., SUGIMOTO, Y. Computer-aided three-dimensional visualization of twisted cementite lamellae in eutectoid steel. *Acta Materialia* 56, 20 (2008), 5995–6002.
- [3] AGOSTON, M. K. *Computer Graphics and Geometric Modelling: Implementation & Algorithms*, vol. 1. Springer, Berlin, Heidelberg, 2005.
- [4] ALKEMPER, J., VOORHEES, P. Quantitative serial sectioning analysis. *Journal of microscopy* 201, 3 (2001), 388–394.
- [5] ALVES, A. L. M., DA SILVA ASSIS, W. L., DOS PRAZERES, T. F., DE ANDRADE, A. B. R., DA COSTA, M. F. B., FONSECA, G. D., FERREIRA, J. L. G., DE OLIVEIRA, D. Study of the influence of the distance between layers in the 3d reconstruction technique via serial section. Em *Proceeding Series of the XVI Brazil MRS Meeting* (Gramado - RS, 2017), SBPMAT, p. 1473.
- [6] AYACHIT, U. *The ParaView Guide: A Parallel Visualization Application*, community ed. Kitware, Inc., USA, 2015.
- [7] AZEVEDO, E., CONCI, A. *Computação Gráfica: Geração de Imagens*. Elsevier, 2003.
- [8] BAATZ, W. Photography: An illustrated historical overview. *New York: Barron's* (1997).
- [9] BABOUT, L., MAIRE, E., BUFFIERE, J.-Y., FOUGERES, R. Characterization by x-ray computed tomography of decohesion, porosity growth and coalescence in model metal matrix composites. *Acta Materialia* 49, 11 (2001), 2055–2063.
- [10] BALDISSERA, M. R., ASSIS, W. L. D. S., DE FREITAS, A. F. P., RIOS, P. R. Simulação do processo de reconstrução em três dimensões, 2015.
- [11] BAÍA, D. M. Análise da variação das propriedades mecânicas e resistência à corrosão, do aço inoxidável superduplex uns s32550 após tratamento térmico de envelhecimento, 2018. Trabalho de Conclusão de Curso em Engenharia Metalúrgica da Escola de Engenharia Industrial Metalúrgica de Volta Redonda da Universidade Federal Fluminense.
- [12] BERLANGA, M. L., PHAN, S., BUSHONG, E. A., WU, S., KWON, O., PHUNG, B. S., LAMONT, S., TERADA, M., TASHIZEN, T., MARTONE, M. E., ELLISMAN,

- M. H. Three-Dimensional Reconstruction of Serial Mouse Brain Sections: Solution for Flattening High-Resolution Large-Scale Mosaics. *Frontiers in Neuroanatomy* 5, March (2011), 1–8.
- [13] BHANDARI, Y., SARKAR, S., GROEBER, M., UCHIC, M. D., DIMIDUK, D. M., GHOSH, S. 3D polycrystalline microstructure reconstruction from FIB generated serial sections for FE analysis. *Computational Materials Science* 41, 2 (2007), 222–235.
- [14] CAO, S., SOMSEN, C., CROITORU, M., SCHRYVERS, D., EGGELER, G. Focused ion beam/scanning electron microscopy tomography and conventional transmission electron microscopy assessment of Ni<sub>4</sub>Ti<sub>3</sub> morphology in compression-aged Ni-rich Ni-Ti single crystals. *Scripta Materialia* 62, 6 (2010), 399–402.
- [15] CERQUEIRA, F. A., NISHIKAWA, A. S., GUESSER, W. L., AZEVEDO, C. R. D. F. Three-dimensional reconstruction of compacted graphite in vermicular cast iron by manual serial sectioning. *Metallurgy and materials* 68, 1 (2015), 307–312.
- [16] CHAWLA, N., GANESH, V. V. Three-dimensional (3D) Microstructure Visualization and Finite Element Modeling of the Mechanical Behavior of Heterogeneous Materials. *Proceedings of the 11th Annual Conference on Fracture* (2005).
- [17] CHIAVERINI, V. *Aços e Ferros Fundidos*, 6 ed. Publicação da Associação Brasileira de Metalurgia e Materiais, São Paulo, 1990.
- [18] CHILDS, H., BRUGGER, E., WHITLOCK, B., MEREDITH, J., AHERN, S., PUGMIRE, D., BIAGAS, K., MILLER, M., HARRISON, C., WEBER, G. H., KRISHNAN, H., FOGAL, T., SANDERSON, A., GARTH, C., BETHEL, E. W., CAMP, D., RÜBEL, O., DURANT, M., FAVRE, J. M., NAVRÁTIL, P. VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*. Oct 2012, p. 357–372.
- [19] CONVERSE, M. I., FULLWOOD, D. T. Enhancing nanoscale SEM image segmentation and reconstruction with crystallographic orientation data and machine learning. *Materials Characterization* 83 (2013), 109–122.
- [20] COPALERT, H. *Metalografia dos Produtos Siderúrgicos Comuns*, 4 ed. Edgard Blücher, São Paulo, 2008.
- [21] DE ANDRADE GRAÇA, A. B. R., DA SILVA ASSIS, W. L., NEVES, T. A., NÚÑEZ, Y. R., RIOS, P. R., DOS PRAZERES, T. F., FERREIRA, J. L. G. Desenvolvimento de ferramenta computacional para criação de domínios discretos em 3d. Em *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics* (IMECC/UNICAMP - Campinas - SP, 2018), SBMAC.
- [22] DE ANDRADE GRAÇA, A. B. R., DA SILVA ASSIS, W. L., NEVES, T. A., NÚÑEZ, Y. R., RIOS, P. R., FERREIRA, J. L. G. Estudo para criação de domínios discretos tridimensionais a partir de imagens digitais. IV SIMMA - Simpósio de Matemática da Região Sul Fluminense - 2018.

- [23] DE ANDRADE GRAÇA, A. B. R., DA SILVA ASSIS, W. L., NEVES, T. A., NÚÑEZ, Y. R., RIOS, P. R., FERREIRA, J. L. G., BAÍA, D. M. Implementação de software para reconstrução 3d utilizando imagens 2d de microestruturas obtidas através da técnica de seccionamento em série. Em *XII Encontro Acadêmico de Modelagem Computacional - EAMC* (2019), Laboratório Nacional de Computação Científica - LNCC/MCTIC.
- [24] DE FREITAS, A. F. P. Metalografia quantitativa tridimensional dos ferros fundidos nodular e cinzento. Dissertação de Mestrado, 2008.
- [25] DEBASTIANI, C. A. *Definindo escopo em projetos de software*. Novatec Editora, 2016.
- [26] DEHOFF, R. Quantitative serial sectioning analysis: preview. *Journal of Microscopy* 131, 3 (1983), 259–263.
- [27] DEITEL, H. M., DEITEL, P. J. *C++ how to program*, 5 ed. Pearson Education, Inc., 2006.
- [28] DOS PRAZERES, T. F. Análise de erros na reconstrução em 3d por seccionamento em série: um estudo em microestrutura simulada. Dissertação de Mestrado, Universidade Federal Fluminense, 2019.
- [29] DOS PRAZERES, T. F., DA SILVA ASSIS, W. L., DE ANDRADE GRAÇA, A. B. R. Influência da resolução da imagem na reconstrução microestrutural em 3d por seccionamento em série. Em *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics* (IMECC/UNICAMP - Campinas - SP, 2018), SBMAC.
- [30] FERREIRA, J. L. G. Simulação orientada a objetos de transformações de fase em materiais cristalinos. Dissertação de Mestrado, Universidade Federal Fluminense, 2018.
- [31] FERREIRA, J. L. G., ASSIS, W. L. S., NEVES, T. A., DE OLIVEIRA, G. C. Simulação orientada a objetos da nucleação e crescimento via autômato celular. Em *Anais de congresso do X Encontro Acadêmico de Modelagem Computacional* (2017).
- [32] FIALHO RODRIGUES, A., FARINAZZO VITRAL, R. W. Aplicações da tomografia computadorizada na odontologia. *Pesquisa Brasileira em Odontopediatria e Clínica Integrada* 7, 3 (2007).
- [33] GANTI, S., TURNER, B., KIRSCH, M., ANTHONY, D., MCCOY, B., TRIVEDI, H., SUNDAR, V. Three-dimensional (3d) analysis of white etching bands (webs) in aisi m50 bearing steel using automated serial sectioning. *Materials Characterization* 138 (2018), 11–18.
- [34] GARCIA, G. B., SUAREZ, O. D., ARANDA, J. L. E., TERCERO, J. S., GRACIA, I. S. *Learning Image Processing with OpenCV*. Packt Publishing, 2015.
- [35] GARIB, D. G., RAYMUNDO JR, R., RAYMUNDO, M. V., RAYMUNDO, D. V., FERREIRA, S. N. Tomografia computadorizada de feixe cônico (cone beam): entendendo este novo método de diagnóstico por imagem com promissora

- aplicabilidade na ortodontia. *Rev Dental Press Ortod Ortop Facial* 12, 2 (2007), 139–56.
- [36] GOMES, J., VELHO, L. *Sistemas Gráficos 3D*, 2 ed. Série de Computação e Matemática. IMPA, 2007.
- [37] GOMES, J., VELHO, L. *Fundamentos da Computação Gráfica*, 1 ed. Série de Computação e Matemática. IMPA, 2015.
- [38] GONZALES, R. C., WOODS, R. E. *Processamento Digital de Imagens*, 3 ed. Pearson Education do Brasil, 2010.
- [39] GROEBER, M. A., HALEY, B. K., UCHIC, M. D., DIMIDUK, D. M., GHOSH, S. 3D reconstruction and characterization of polycrystalline microstructures using a FIB-SEM system. *Materials Characterization* 57, 4-5 (2006), 259–273.
- [40] GUIDORIZZI, H. L. *Um Curso de Cálculo*, 5 ed., vol. 1. LTC, 2001.
- [41] HARA, T., TSUCHIYA, K., TSUZAKI, K., MAN, X., ASAHATA, T., UEMOTO, A. Application of orthogonally arranged FIB-SEM for precise microstructure analysis of materials. *Journal of Alloys and Compounds* 577, SUPPL. 1 (2013), S717–S721.
- [42] INTEL CORPORATION, WILLOW GARAGE, I. Opencv (open source computer vision library), Junho 2000. Acessado em 08/2017.
- [43] IO, D. Disponível em: < <https://www.draw.io/> >. Acesso em Janeiro de 2019 15 (2017).
- [44] KAEHLER, A., BRADSKI, G. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*, 1 ed. O'Reilly Media, 2017.
- [45] KOTULA, P. G., ROHRER, G. S., MARSH, M. P. Focused ion beam and scanning electron microscopy for 3d materials characterization. *MRS Bulletin* 39, 4 (2014), 361–365.
- [46] KRAL, M., SPANOS, G. Three-dimensional analysis of proeutectoid cementite precipitates. *Acta Materialia* 47, 2 (1999), 711–724.
- [47] KRAL, M. V., MANGAN, M. A., SPANOS, G., ROSENBERG, R. O. Three-dimensional analysis of microstructures. *Materials Characterization* 45 (2000), 17–23.
- [48] LEE, S. G., GOKHALE, A. M. Visualization of three-dimensional pore morphologies in a high-pressure die-cast Mg-Al-RE alloy. *Scripta Materialia* 56, 6 (2007), 501–504.
- [49] LI, M., GHOSH, S., ROUNS, T. N., RICHMOND, H. W. O., HUNT, W. Serial sectioning method in the construction of 3d microstructures for particlereinforced mms. *Materials Characterization* 41 (1998), 81–95.
- [50] LIMA, E. L. *Análise real, Funções de Uma Variável*, 12 ed., vol. 1 em *Coleção Matemática Universitária*. IMPA, 2014.

- [51] MADEJ, L., MOJZESZKO, M., CHRAPONSKI, J., ROSKOSZ, S., CWAJNA, J. Digital Material Representation Model of Porous Microstructure Based on 3D Reconstruction Algorithm. *Archives of Metallurgy and Materials* 62, 2 (2017), 563–569.
- [52] MARROW, T., BUFFIERE, J.-Y., WITHERS, P., JOHNSON, G., ENGELBERG, D. High resolution x-ray tomography of short fatigue crack nucleation in austempered ductile cast iron. *International journal of fatigue* 26, 7 (2004), 717–725.
- [53] MONTEIRO, L. H. Binarização por otsu e outras técnicas usadas na defecção de placas.
- [54] NOGUEIRA, D. A. Reconstrução tridimensional por seccionamento em série dos poros presentes no aço 316l sinterizado. Dissertação de Mestrado, Universidade Federal Fluminense, 2016.
- [55] OPEN, C. Opencv documentation, 2013.
- [56] PHAM, B., PRINGLE, G. Color correction for an image sequence. *IEEE Computer Graphics and Applications* 15, 3 (1995), 38–42.
- [57] POKHAREL, R., LIND, J., KANJARLA, A. K., LEBENSOHN, R. A., LI, S. F., KENESEI, P., SUTER, R. M., ROLLETT, A. D. Polycrystal plasticity: Comparison between grain - scale observations of deformation and simulations. *Annual Review of Condensed Matter Physics* 5, 1 (2014), 317–346.
- [58] RHINES, F., CRAIG, K., ROUSSE, D. Measurement of average grain volume and certain topological parameters by serial section analysis. *Metallurgical Transactions A* 7, 11 (1976), 1729–1734.
- [59] ROBER, N. Paraview tutorial: for the visualization of earth-and climate science data. *Deutsches Klimarechenzentrum, Bundesstrasse 45a 20146* (2014).
- [60] RODRIGUES, M. R. B. *Reconstrução Microestrutural 3D das Ligas Ti-6Al-4V e Nb-TiO<sub>2</sub> por seccionamento em série*. PhD thesis, Universidade Federal Fluminense, 2009.
- [61] RUSS, J. C. *The Image Processing Cookbook*, 4 ed. CreateSpace Independent Publishing Platform, 2017.
- [62] SINGER, A. B. *Practical C++ Design: from programming to architecture*. Apress, 2017.
- [63] SINGH, H., GOKHALE, A. Visualization of three-dimensional microstructures. *Materials Characterization* 54 (2005), 21–29.
- [64] SPOWART, J. E. Automated serial sectioning for 3-D analysis of microstructures. *Scripta Materialia* 55, 1 SPEC. ISS. (2006), 5–10.
- [65] SPOWART, J. E., MULLENS, H. E., PUCHALA, B. T. Collecting and analyzing microstructures in three dimensions: a fully automated approach. *JOM* 55, 10 (2003), 35–37.

- [66] STEINBRUCH, A., WINTERLE, P. *Geometria Analítica*. Pearson Makron Books, 2004.
- [67] TABORDA, L., PEREIRA, L., AMONA, E., PINTO, E. G., RODRIGUES, J. Abdominal aortic aneurysm: an uncommon presentation. *Acta medica portuguesa* 24, 5 (2011), 857–62.
- [68] TECPLOT, I. *User's Manual*, 2010.
- [69] TEN DYKE, R. P., KUNZ, J. C. Object-oriented programming. *IBM Systems Journal* 28, 3 (1989), 465–478.
- [70] TEWARI, A., GOKHALE, A., GERMAN, R. Effect of gravity on three-dimensional coordination number distribution in liquid phase sintered microstructures. *Acta Materialia* 47, 13 (1999), 3721–3734.
- [71] TEWARI, A., GOKHALE, A. M. Application of three-dimensional digital image processing for reconstruction of microstructural volume from serial sections. *Materials Characterization* 44 (2000), 259–269.
- [72] TEWARI, A., GOKHALE, A. M. Estimation of three-dimensional grain size distribution from microstructural serial sections. *Materials Characterization* 46, 4 (2001), 329–335.
- [73] THORNE-TJOMSLAND, G., DUMONTIER, M., JAMIESON, J. C. 3D topography of noncompact zone Golgi tubules in rat spermatids: A computer-assisted serial section reconstruction study. *Anatomical Record* 250, 4 (1998), 381–396.
- [74] TURNER, D. M., KALIDINDI, S. R. Statistical construction of 3-D microstructures from 2-D exemplars collected on oblique sections. *Acta Materialia* 102 (2016), 136–148.
- [75] UCHIC, M. D. Serial Sectioning Methods for Generating 3D Characterization Data of Grain- and Precipitate-Scale Microstructures. *Computational Methods for Microstructure-Property Relationships* (2011).
- [76] UCHIC, M. D., GROEBER, M. A., DIMIDUK, D. M., SIMMONS, J. P. 3D microstructural characterization of nickel superalloys via serial-sectioning using a dual beam FIB-SEM. *Scripta Materialia* 55, 1 SPEC. ISS. (2006), 23–28.
- [77] VILLANOVA, J., CLOETENS, P., SUHONEN, H., LAURENCIN, J., USSEGLIO-VIRETTA, F., LAY, E., DELETTE, G., BLEUET, P., JAUFFRÈS, D., ROUSSEL, D., LICHTNER, A. Z., MARTIN, C. L. Multi-scale 3D imaging of absorbing porous materials for solid oxide fuel cells. *Journal of Materials Science* 49 (agosto de 2014), 5626–5634.
- [78] WILSON, M. H., BROCKLEBANK, R. Colour and perception: The work of edwin land in the light of current concepts. *Contemporary Physics* 3, 2 (1961), 91–111.
- [79] WINTERLE, P. *Vetores e Geometria Analítica*. Pearson Makron Books, 2000.

- 
- [80] WU, K. M., INAGAWA, Y., ENOMOTO, M. Three-dimensional morphology of ferrite formed in association with inclusions in low-carbon steel. *Materials Characterization* 52 (2004), 121–127.
- [81] WUNSCH, B., CHAWLA, N. Serial sectioning for 3d visualization and modeling of sic particle reinforced aluminum composites. *TMS Outstanding Student* (2003), 1–16.
- [82] ZANKEL, A., WAGNER, J., POELT, P. Serial sectioning methods for 3d investigations in materials science. *Micron* 62 (2014), 66–78.
- [83] ZIMMERMANN, T., DUBOIS-PÈLERIN, Y., BOMME, P. Object-oriented finite element programming: I. governing principles. *Computer methods in applied mechanics and engineering* 98, 2 (1992), 291–303.

## APÊNDICE A - Como abrir um arquivo de extensão .dat no Tecplot 360<sup>®</sup>

Abra o programa Tecplot 360<sup>®</sup>, de acordo com a Figura A.1.

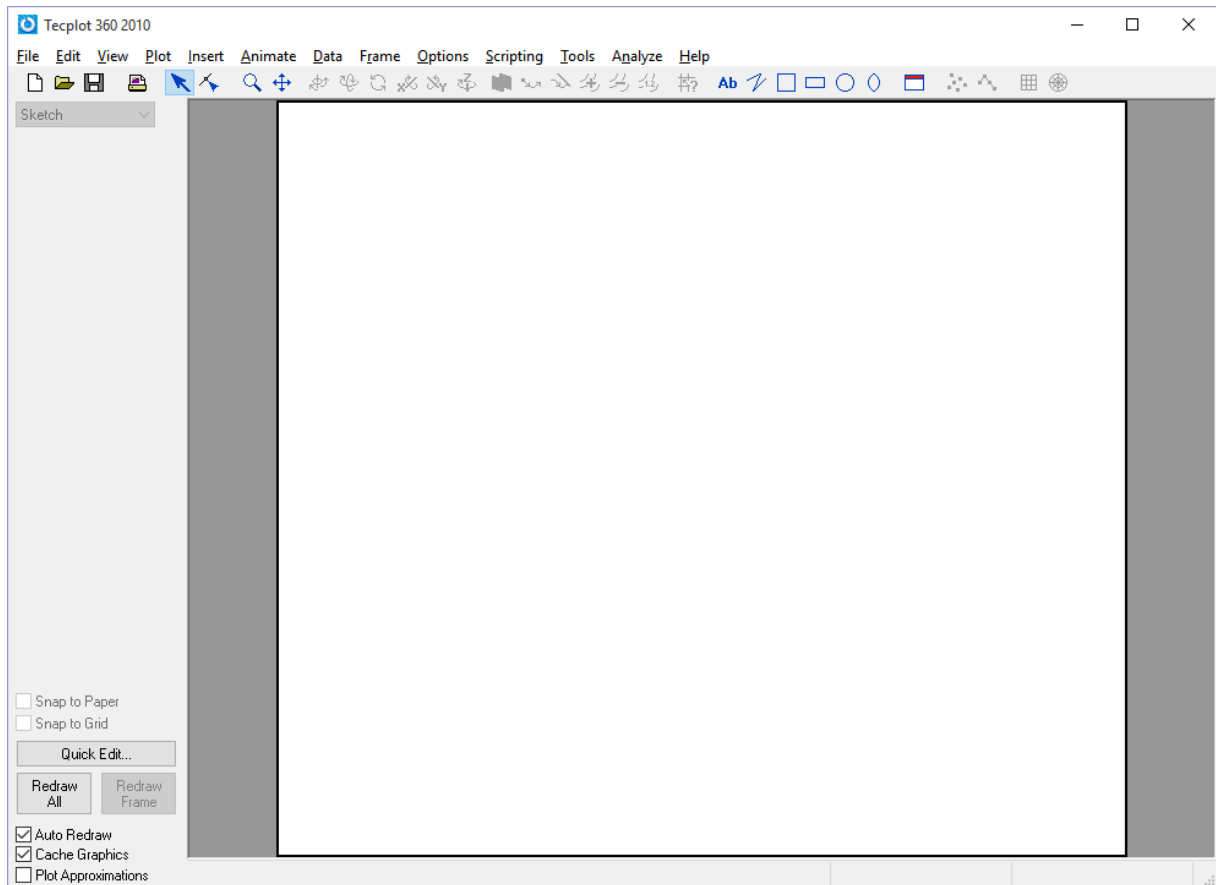


Figura A.1: Tecplot 360<sup>®</sup> 2010.

Clique em “File” no canto superior esquerdo da tela. Em seguida, clique na opção “Load Data File(s)...”, conforme a Figura A.2

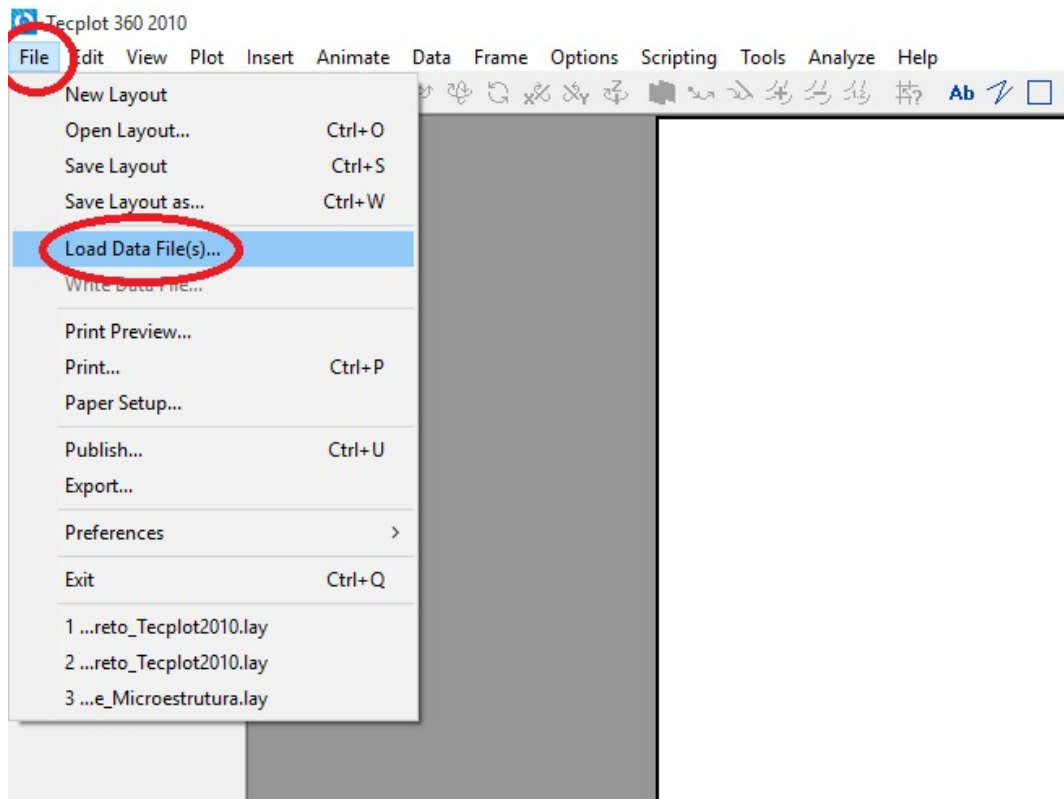


Figura A.2: Abrindo arquivo no Tecplot 360 ®.

Uma nova janela vai abrir (Figura A.3) com opções para seleção de formato de arquivo. Selecione “Tecplot Data Loader” e clique em “OK”.

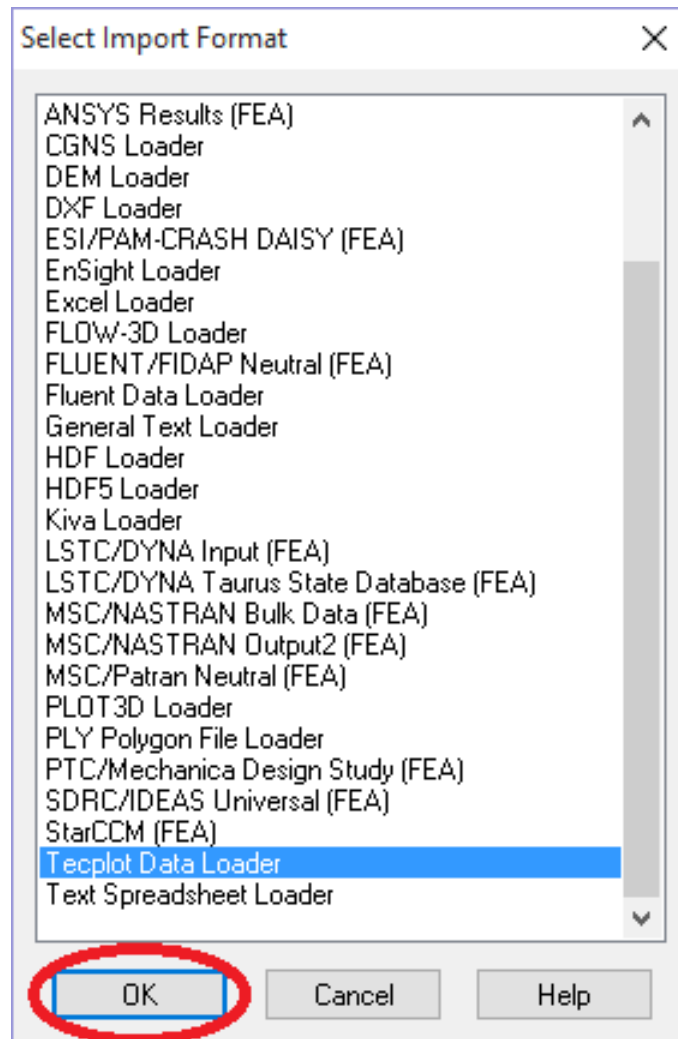


Figura A.3: Selecionando o formato do arquivo.

Quando a janela “Tecplot Data Loader” abrir, navegue por suas pastas e procure o arquivo desejado em seu diretório.

Selecione o arquivo de interesse e clique no botão “Abrir”, conforme a Figura A.4.

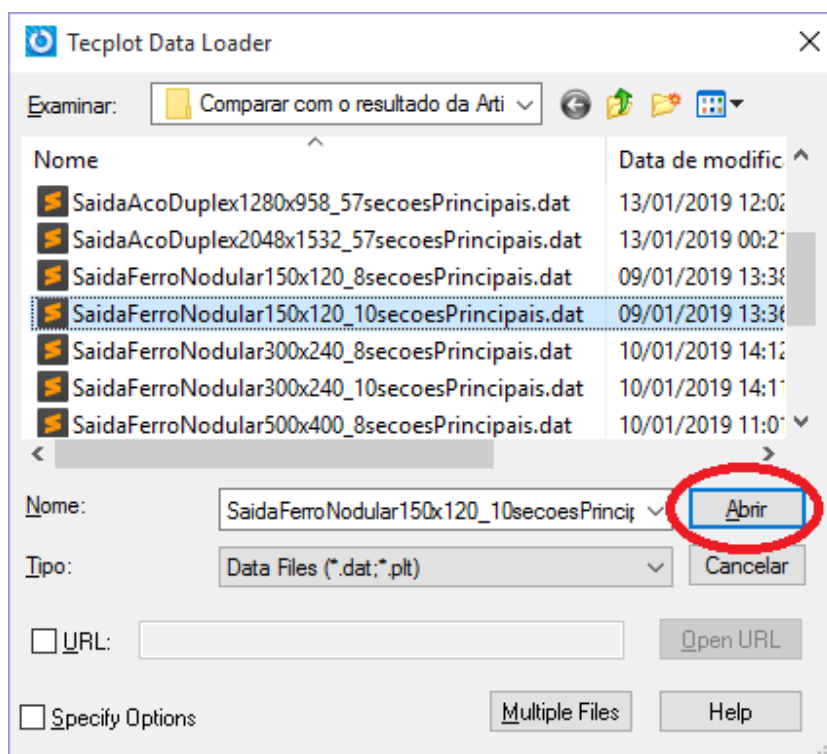


Figura A.4: Selecionando arquivo para abrir no Tecplot 360<sup>®</sup>.

Espera o Tecplot 360<sup>®</sup> carregar o arquivo escolhido.

Assim que o arquivo for carregado, a janela “Select Initial Plot” (Figura A.5) vai abrir. Verifique se as opções estão de acordo com seu objetivo. Então, clique em “OK”.

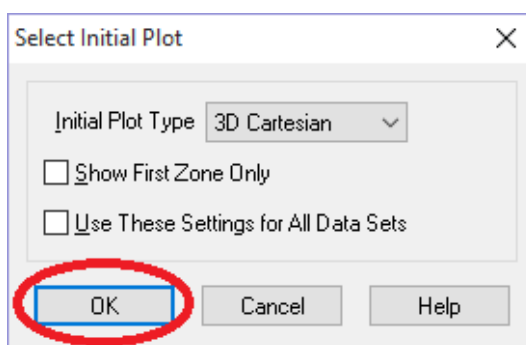


Figura A.5: Janela “Select Inicial Plot”.

O local em que os dados aparecerem na tela será carregado, conforme a Figura A.6.

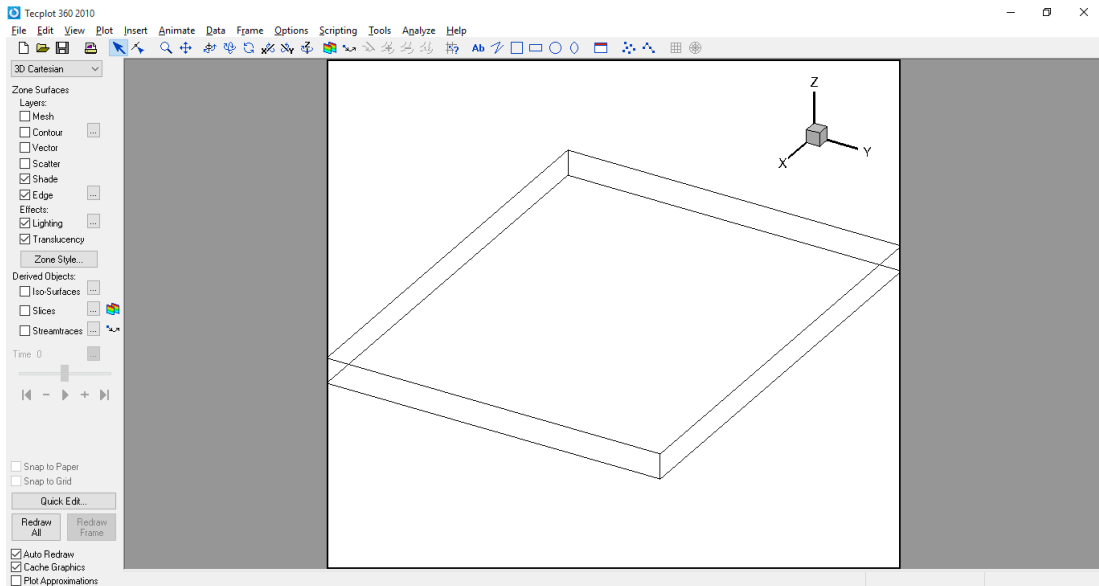


Figura A.6: Área em que os dados aparecerão.

No canto superior esquerdo, clique em “Plot”, procure por “RGB Coloring” e clique em “Variables/Range...”, assim como apresentado na Figura A.7.

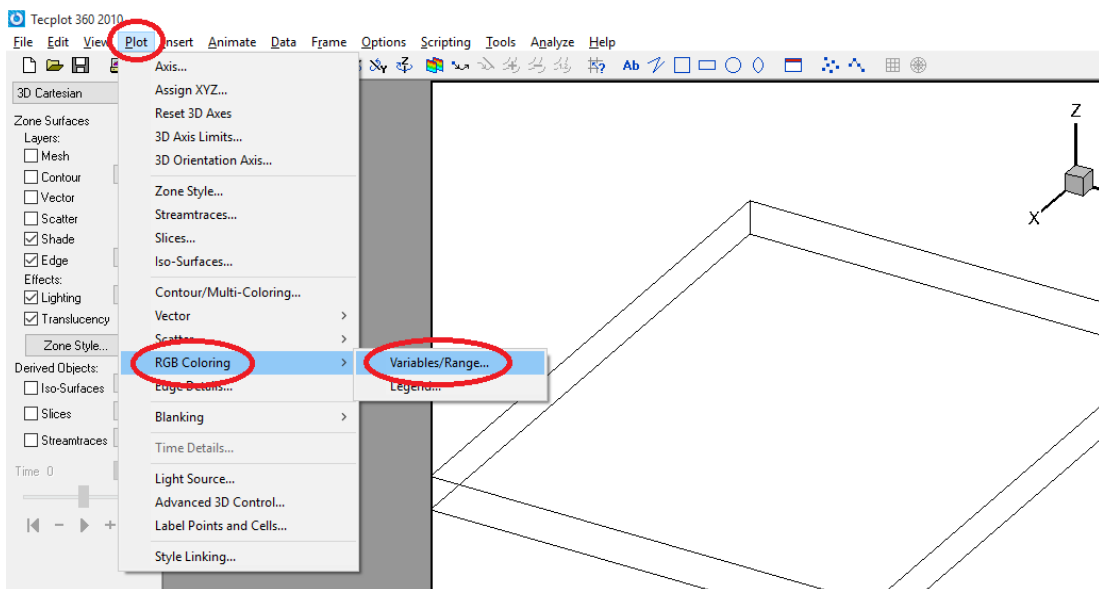


Figura A.7: Ativando o sistema de cores RGB.

A janela “RGB Coloring Options” vai abrir, de acordo com a Figura A.8.

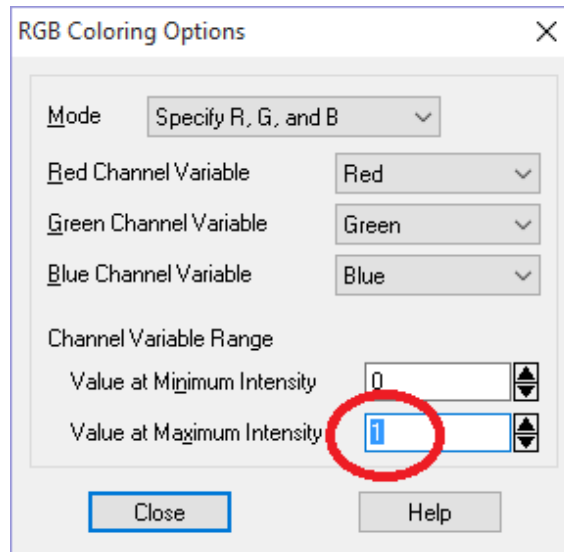


Figura A.8: Janela “RGB Coloring Options”.

Troque o “Value at Maximum Intensity” de 1 para 255, conforme a Figura A.9 e clique em “Close”.

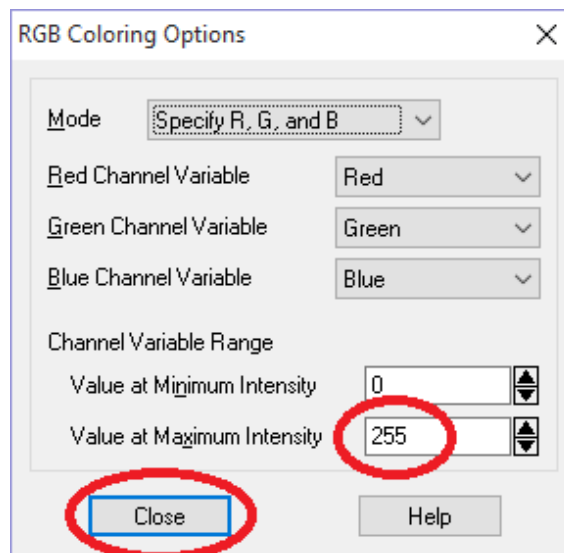


Figura A.9: Janela “RGB Coloring Options”.

A seguir, selecione a opção “Scatter” no canto esquerdo da tela. Uma mensagem aparecerá na tela (Figura A.10). Clique em “Sim”.

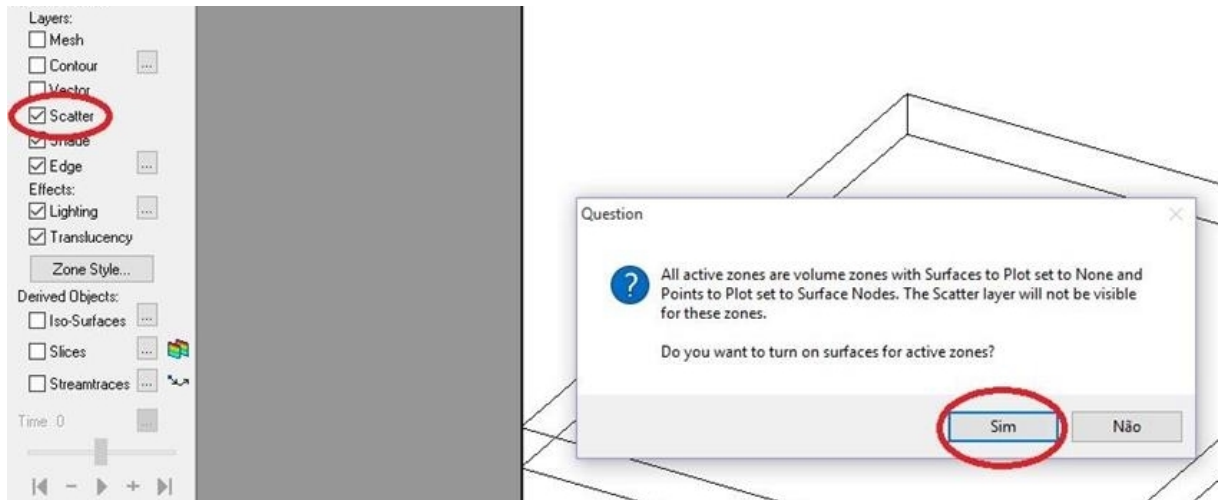


Figura A.10: Ativando a opção “Scatter”.

Clique no botão “Zone Style...”. Quando a janela abrir, selecione a aba “Scatter” e clique em “Outline Color”, conforme a Figura A.11.

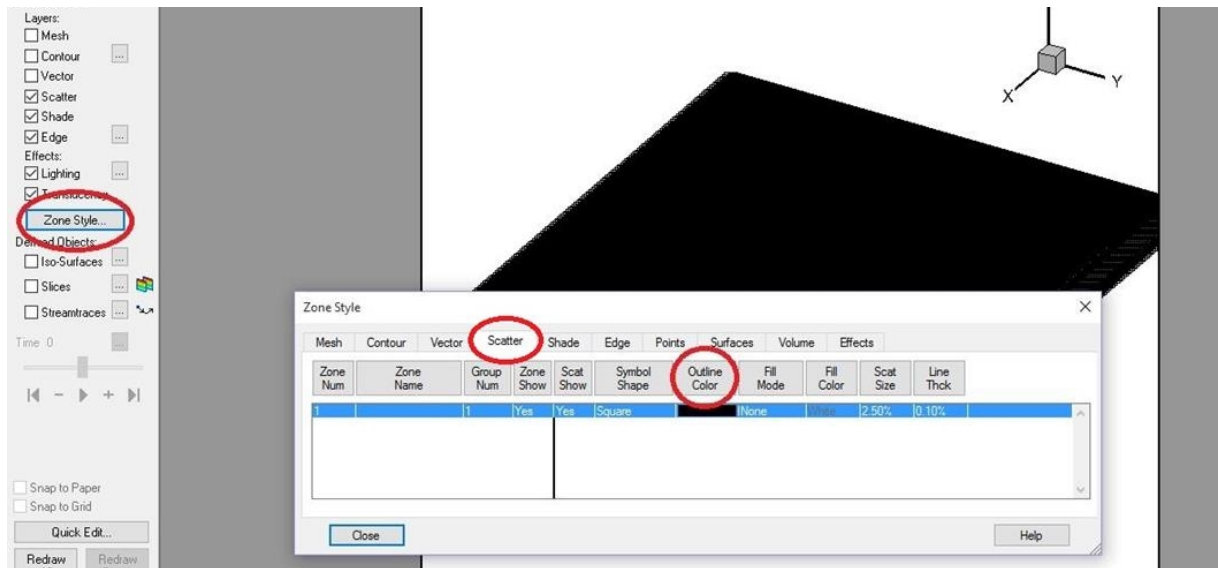


Figura A.11: Configurando o RGB da opção “Scatter”.

Selecione a opção “RGB” na janela “Select Color”, conforme consta na Figura A.12.



Figura A.12: Janela “Select Color”.

Em seguida, para melhorar a aparência da microestrutura, clique em “Symbol Shape”, como apresentado na Figura A.13.

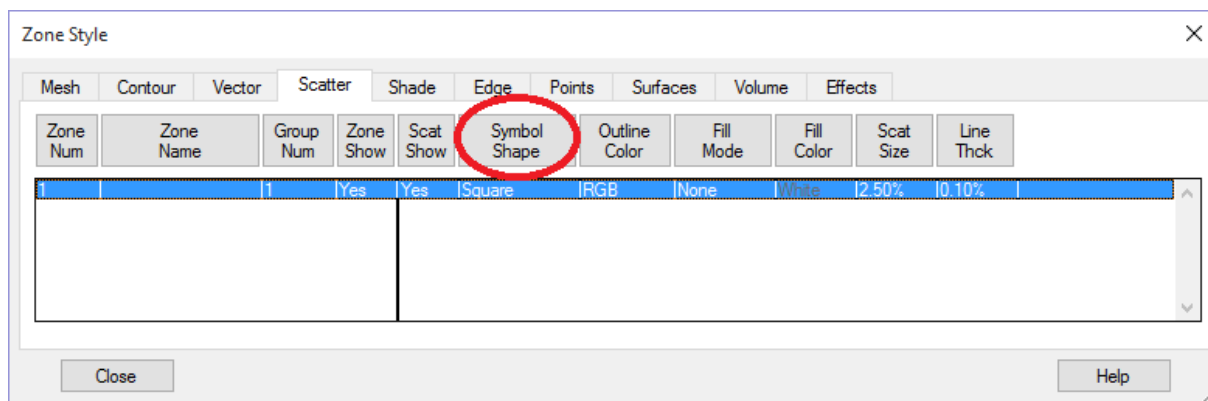


Figura A.13: Configurando o formato dos pontos da malha.

Escolha a opção “Sphere”, conforme a Figura A.14, e feche a janela “Zone Style”.

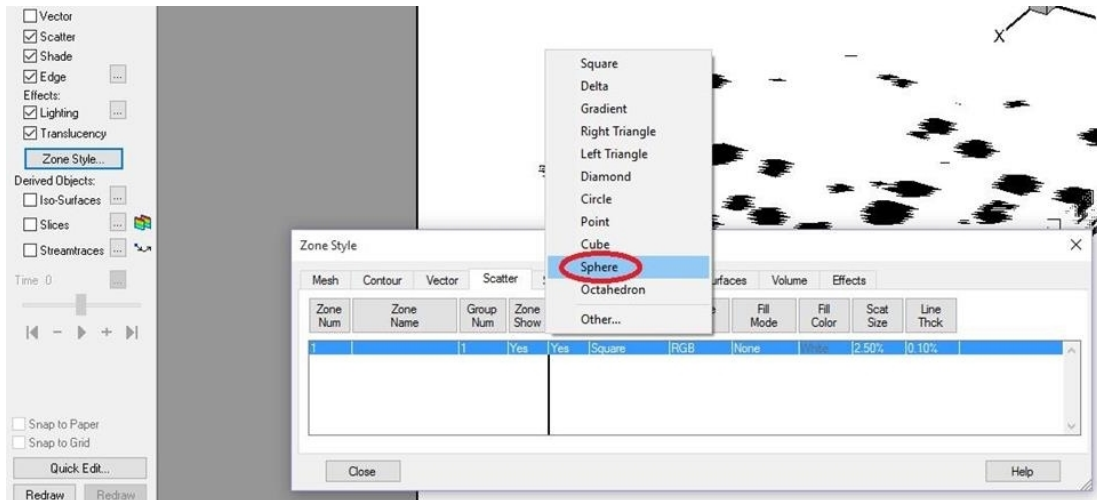


Figura A.14: Selecionando a opção “Sphere”.

O resultado pode ser visto na Figura A.15.

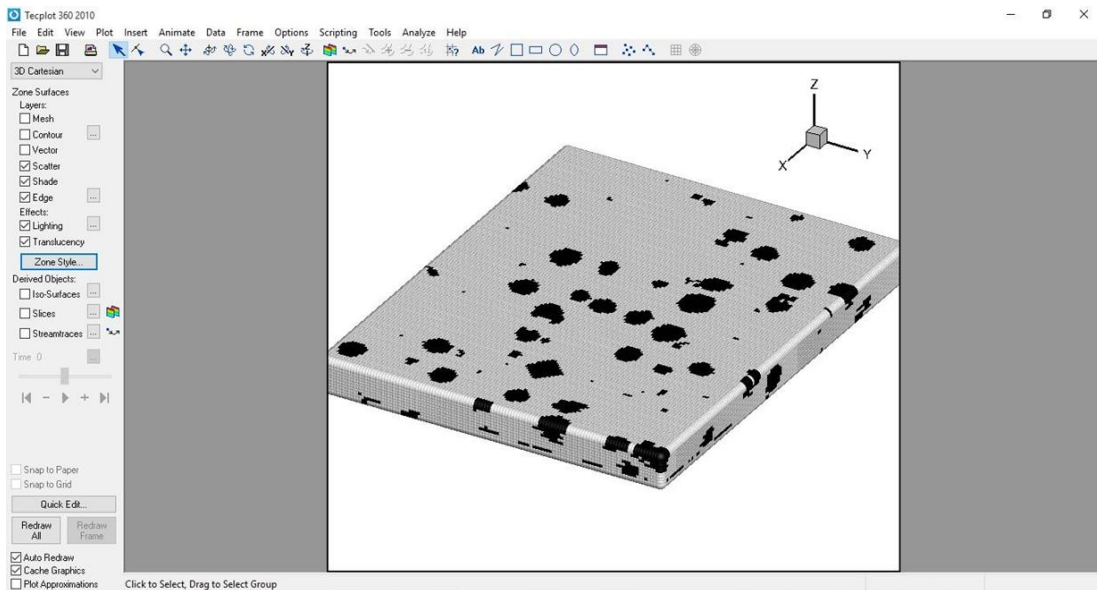


Figura A.15: Resultado: Microestrutura completa.

Para ver os nódulos de grafite da microestrutura de Ferro Fundido Nodular, assim como os resultados apresentados na seção 5.1 do capítulo 5, desmarque as opções “Scatter” e “Shade” e, posteriormente, selecione a opção “Iso-Surfaces”. O resultado ficará semelhante a Figura A.16.

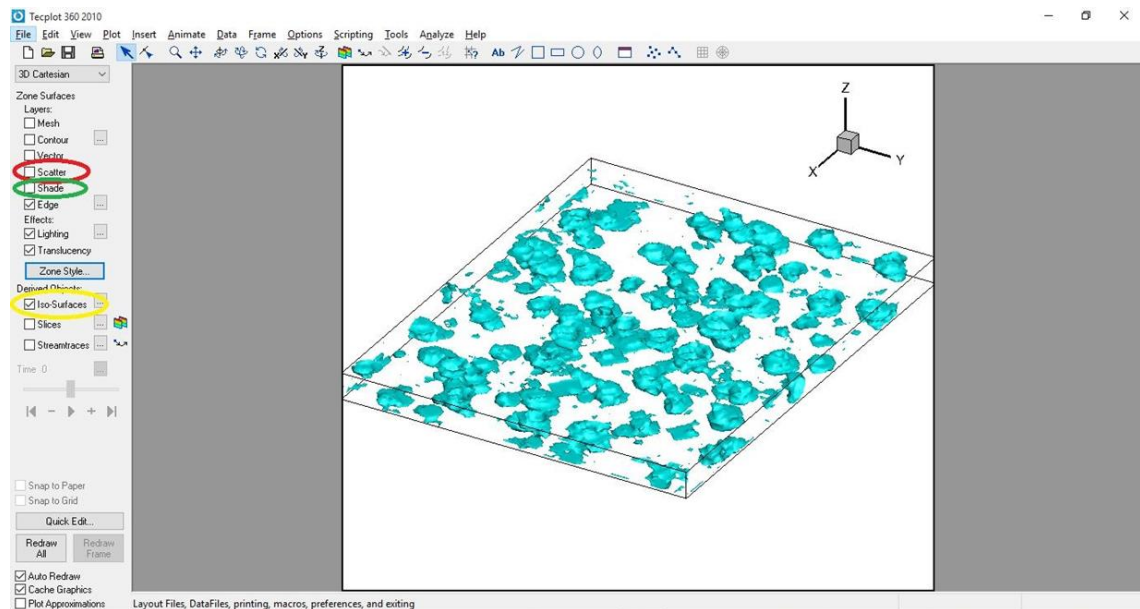


Figura A.16: Resultado: Visualização da microestrutura de Ferro Fundido Nodular.

## APÊNDICE B - Como abrir um arquivo de extensão .dat no ParaView

Abra o programa ParaView, conforme a Figura B.1.

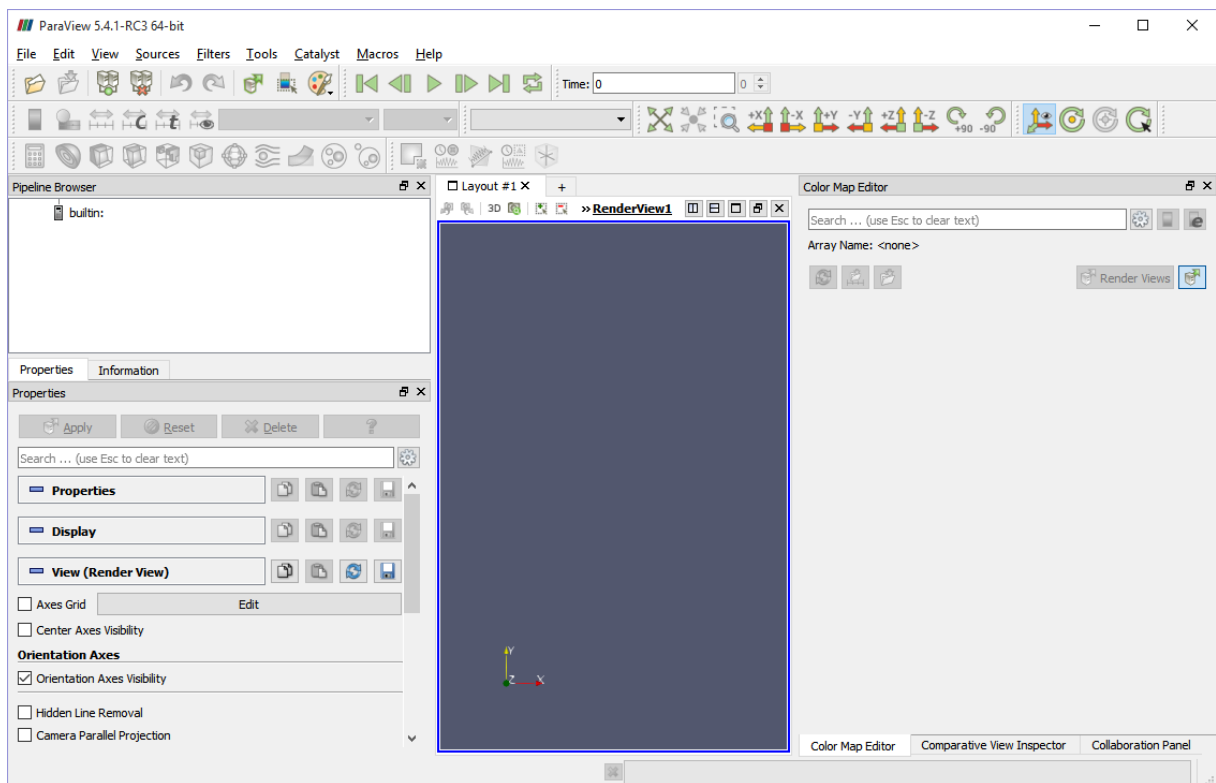


Figura B.1: ParaView 5.4.1-RC3.

Clique em “File” no canto superior esquerdo da tela. Em seguida clique na opção “Open...”, como pode ser visto na Figura B.2. Ou, se preferir, apenas use o atalho “Ctrl+O”.

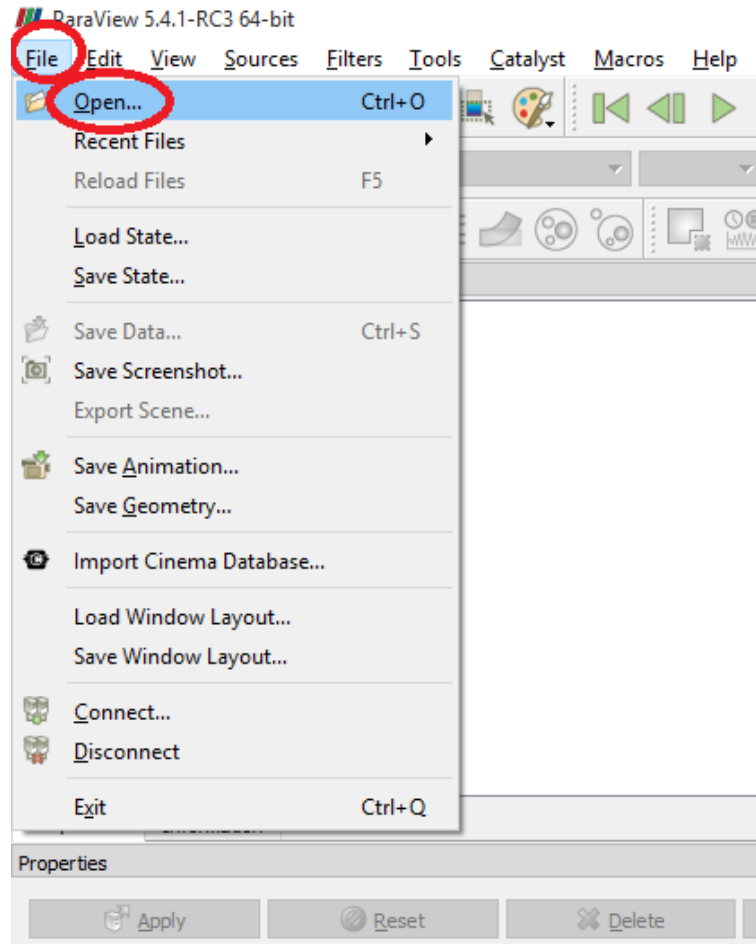


Figura B.2: Abrindo arquivo no ParaView.

Na janela “Open File”, navegue por suas pastas e procure o(s) arquivo(s) desejado(s) em seu diretório.

Selecione o arquivo de interesse e clique no botão “OK”, conforme a Figura B.3.

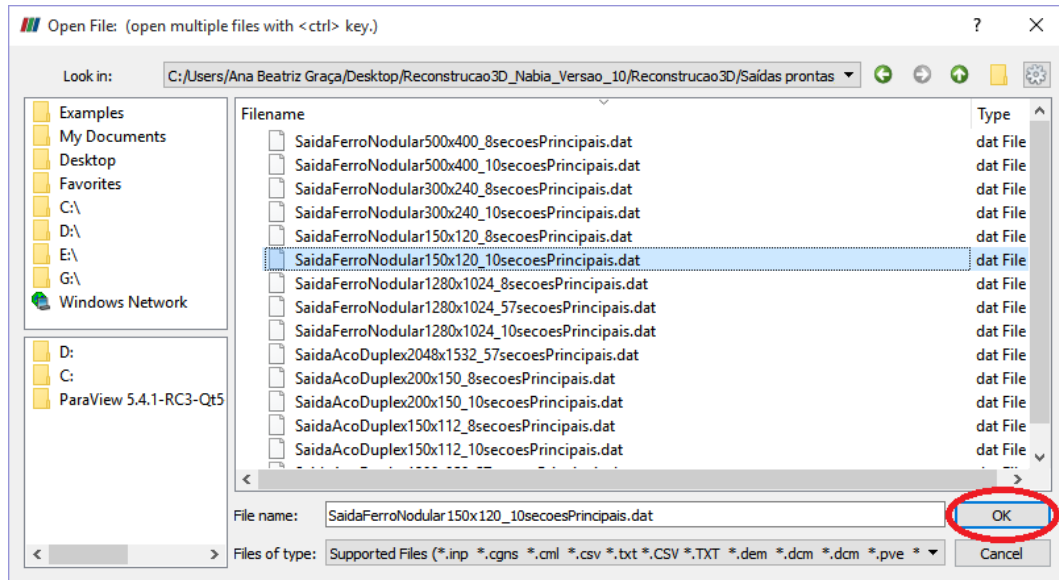


Figura B.3: Selecionando arquivo para abrir no ParaView.

Uma nova janela vai abrir (Figura B.4) com opções para seleção de formato de arquivo. Selecione “Tecplot Files” e clique em “OK”.

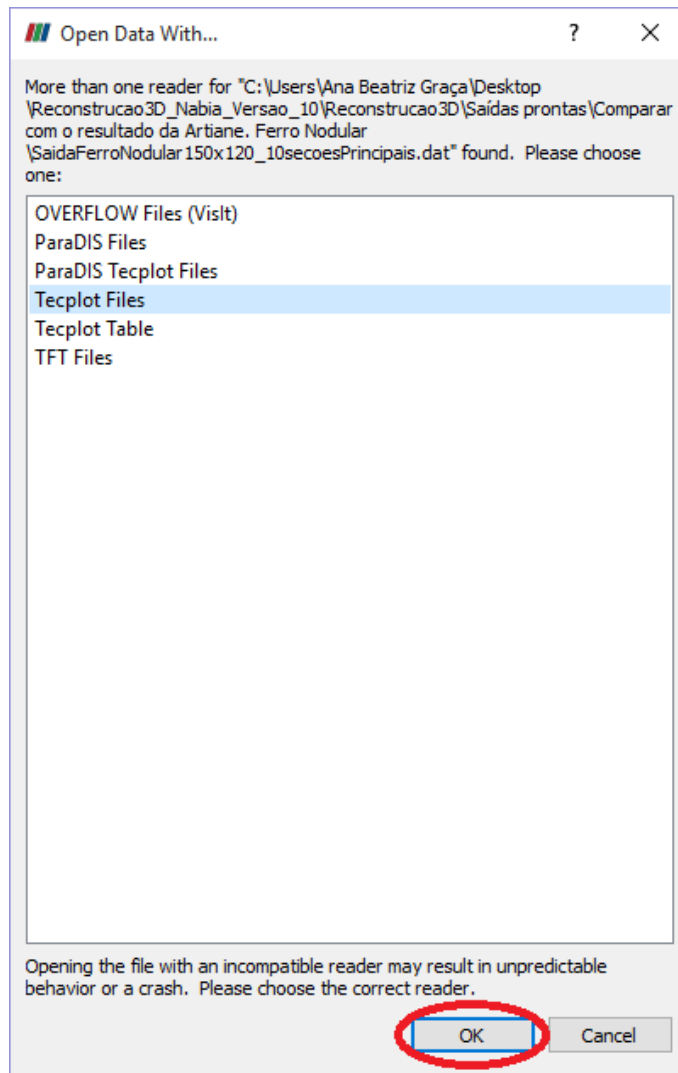


Figura B.4: Selecionando o formato do arquivo.

O arquivo aparecerá aberto no canto esquerdo da tela, como consta na Figura B.5. Abaixo, na aba “Properties”, clique em “Apply”.

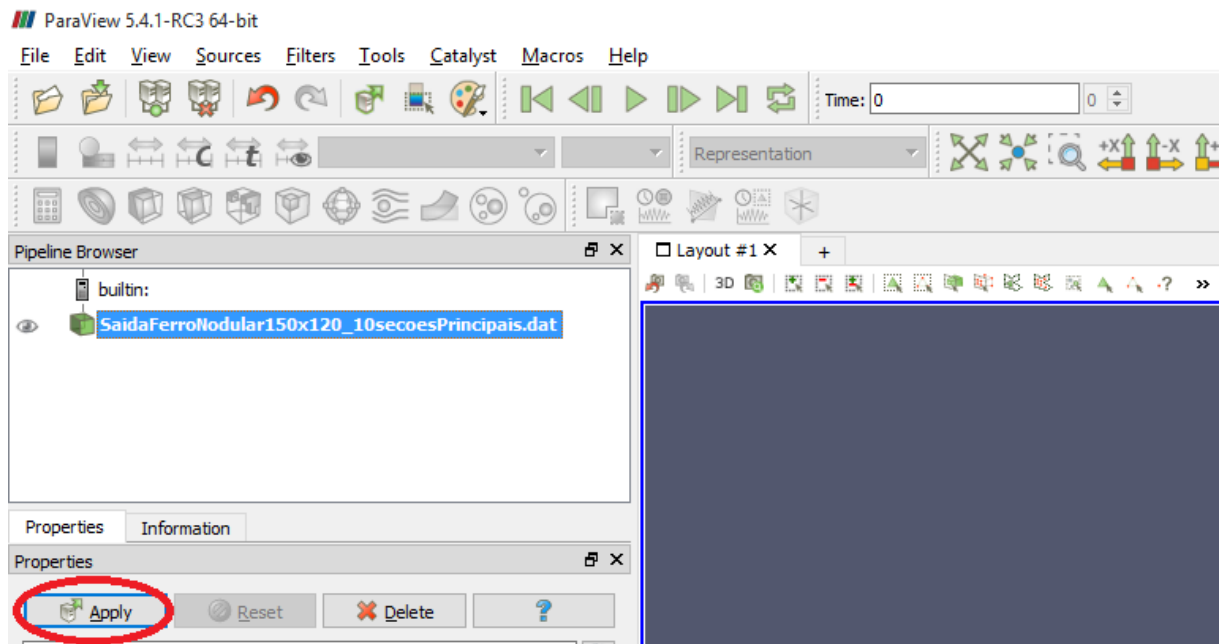


Figura B.5: Arquivo aberto no canto esquerdo do ParaView.

Espera o ParaView aplicar as devidas opções no arquivo escolhido.

Lembre-se que é possível ajustar as dimensões das telas internas do ParaView.

O local em que os dados aparecerem na tela será carregado, conforme a Figura B.6.

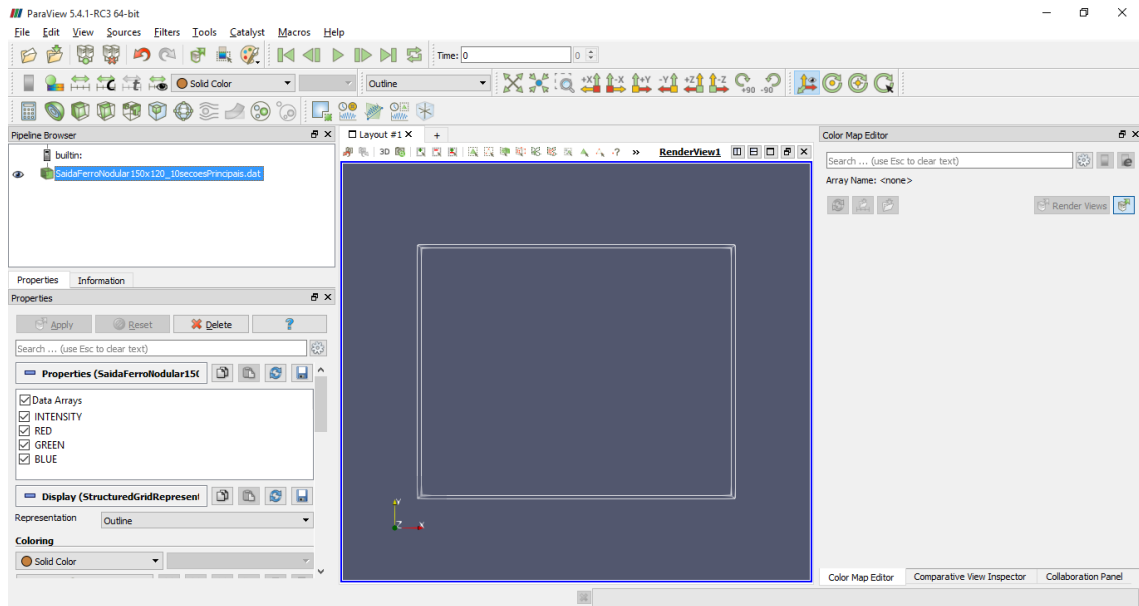


Figura B.6: Área em que os dados aparecerão.

O arquivo que foi aberto logo abaixo do “builtin”, dentro do “Pipeline Browser”, já estará selecionado. Caso não esteja, selecione-o. Desse modo, as opções logo acima estarão disponíveis (Figura B.7). Passe a seta em cima dessas opções para ler seus respectivos nomes. Clique em “Counter”.

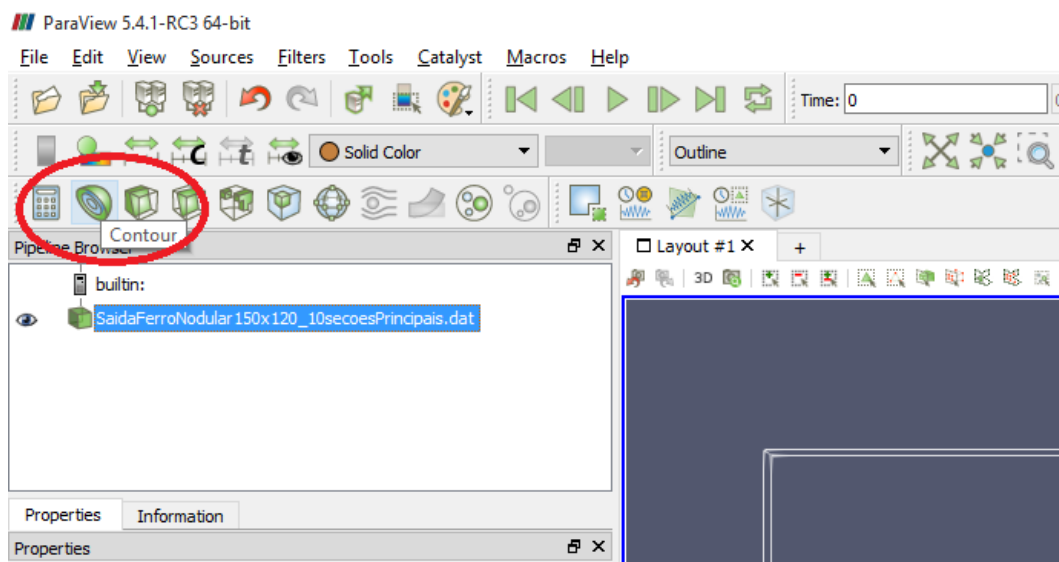


Figura B.7: “Counter”.

O ParaView criará um arquivo chamado “Contour1”, conforme a Figura B.8.

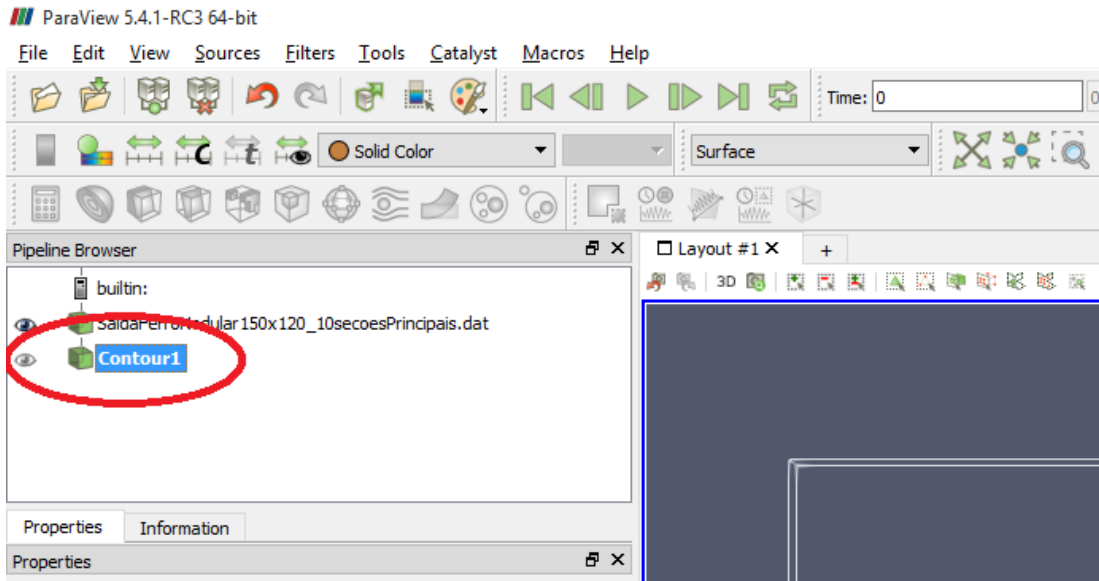


Figura B.8: Arquivo “Contour1”.

Clique na imagem do olho que aparece ao lado esquerdo dos respectivos arquivos abertos, como apresentado na Figura B.9.

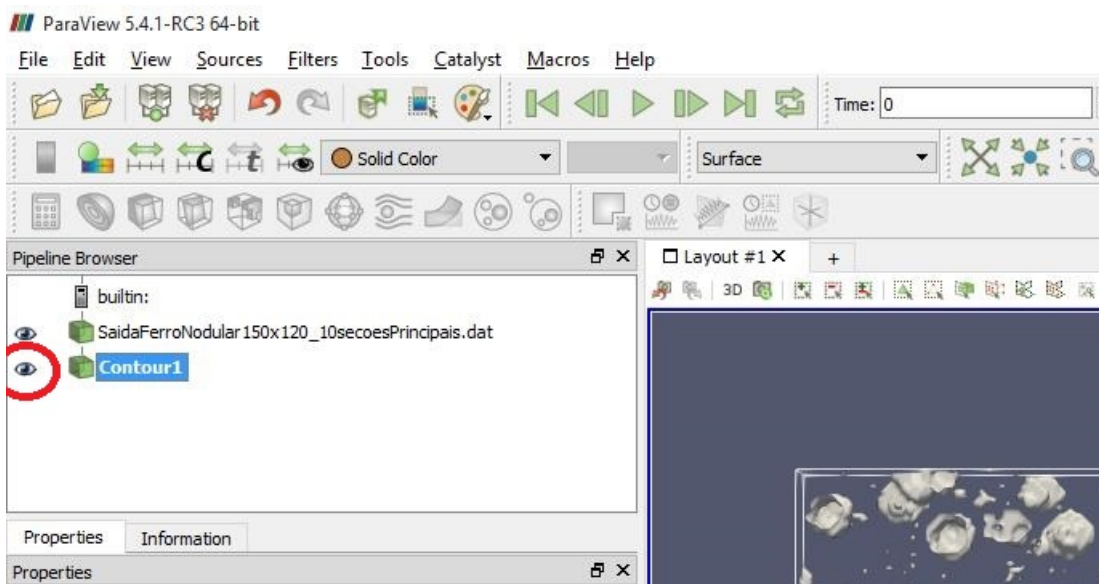


Figura B.9: Mostrar/ocultar arquivos.

Agora é possível visualizar a reconstrução tridimensional da microestrutura de Ferro Fundido Nodular (Figura B.10).

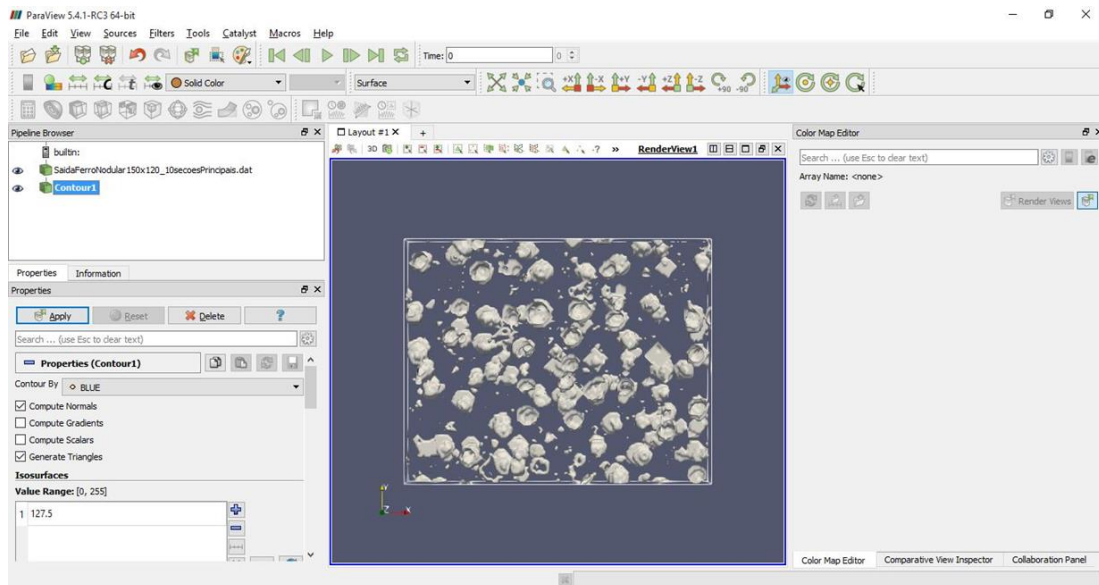


Figura B.10: Visualização das esferas de grafite da microestrutura.

Também é possível rotacionar os eixos, com o mouse, para escolher outro ângulo para visualização, conforme a Figura B.11. Segure o botão esquerdo do mouse e mexa-o.

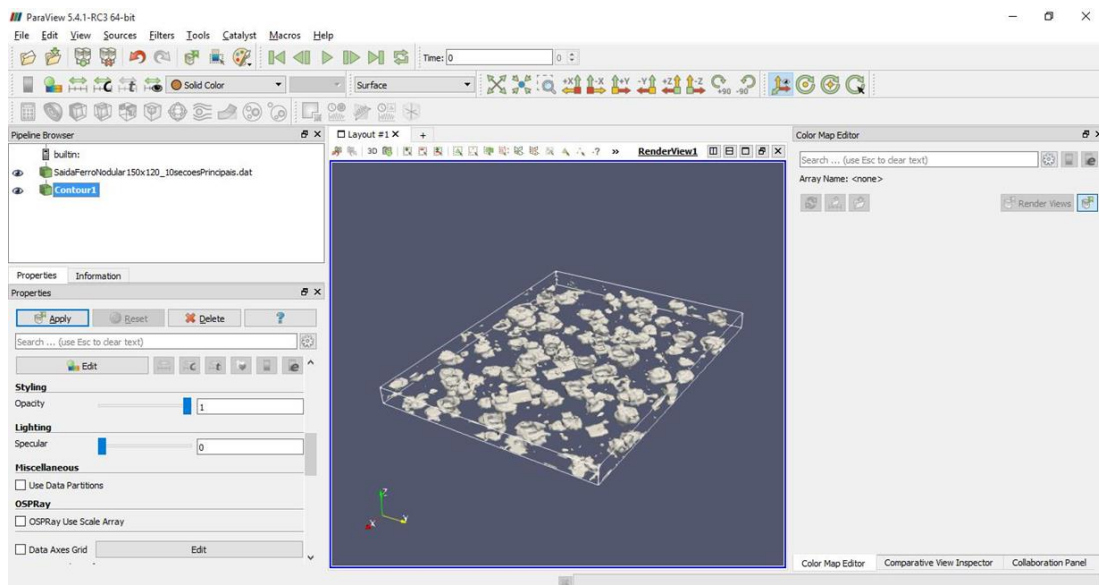


Figura B.11: Rotação da imagem.

Para mudar a cor dos nódulos, procure por “Coloring” dentro da aba “Properties” e clique no botão “Edit”, conforme é mostrado na Figura B.12.

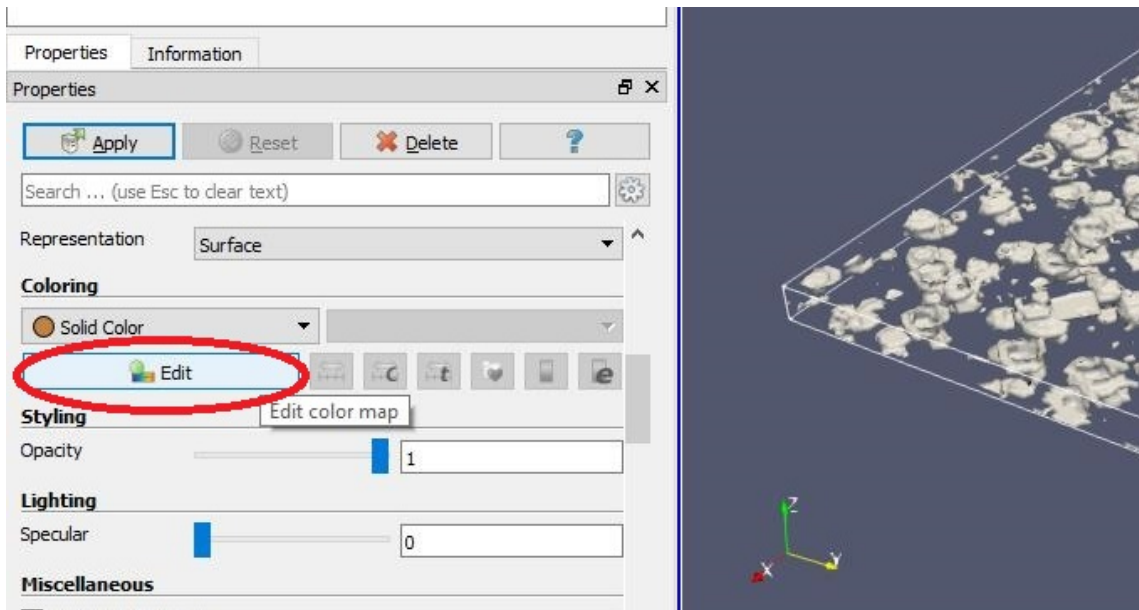


Figura B.12: Modificando a cor das esferas de grafite.

Na janela “Pick Solid Color” (Figura B.13) escolha a cor desejada e clique em “OK”.

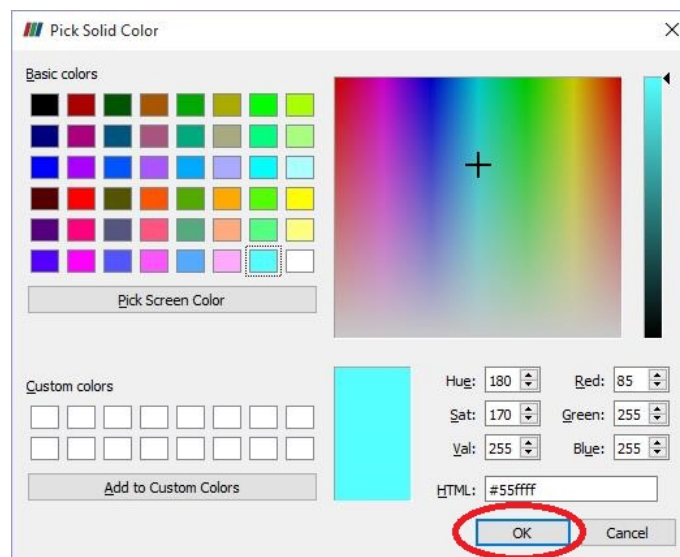


Figura B.13: Janela “Pick Solid Color”.

O resultado pode ser visto na Figura B.14.

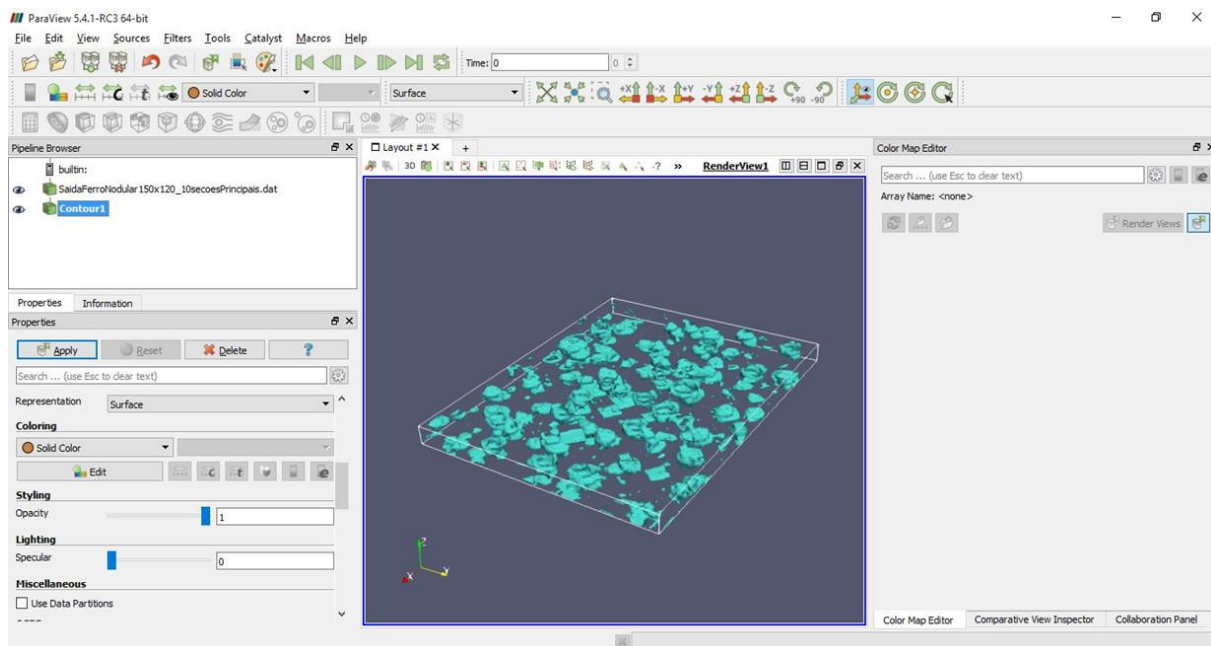


Figura B.14: Resultado: Visualização da microestrutura de Ferro Fundido Nodular.

## APÊNDICE C - Como abrir um arquivo de extensão .dat no VisIt

Abra o programa VisIt, conforme a Figura C.1.

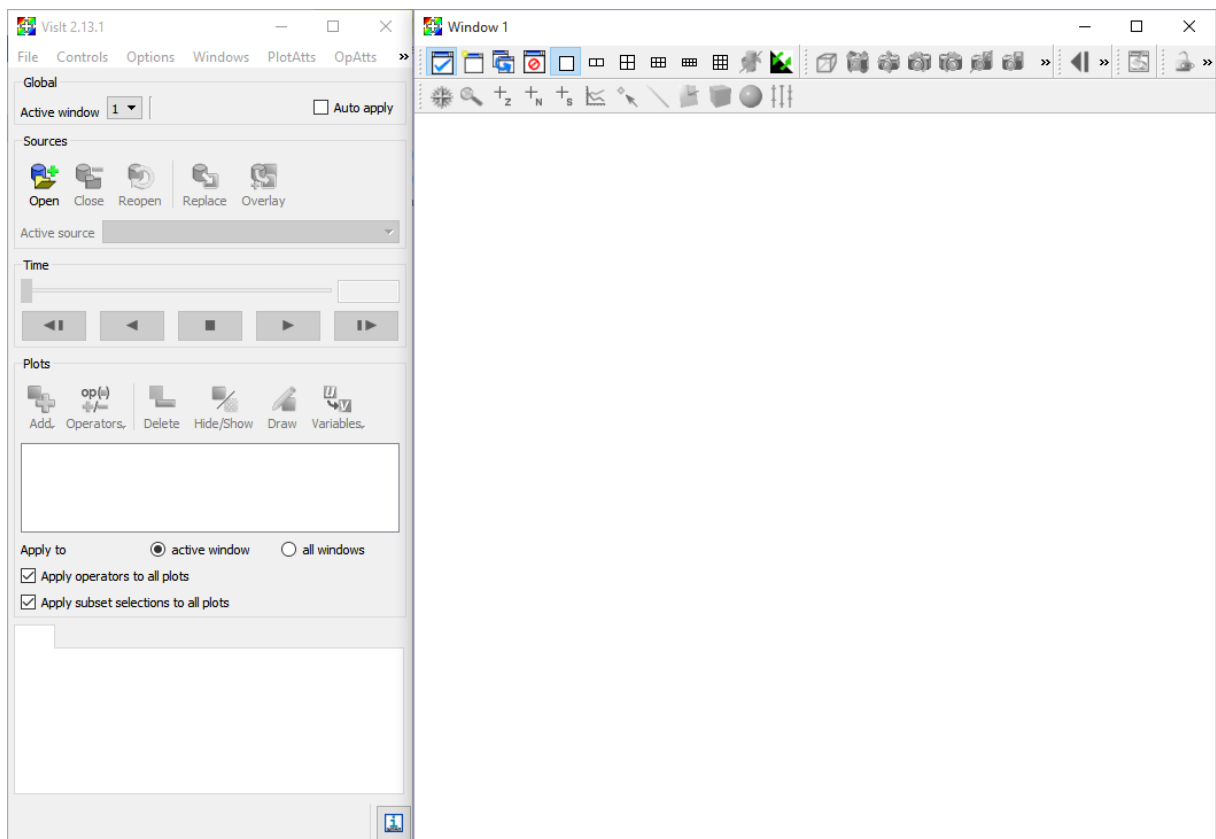


Figura C.1: VisIt 2.13.1.

As configurações são feitas na janela esquerda e a visualização dos dados é feita na janela direita.

Clique em “Open”, localizado na seção “Sources”, como pode ser visto na Figura C.2. Ou, se preferir, apenas use o atalho “Ctrl+O”.

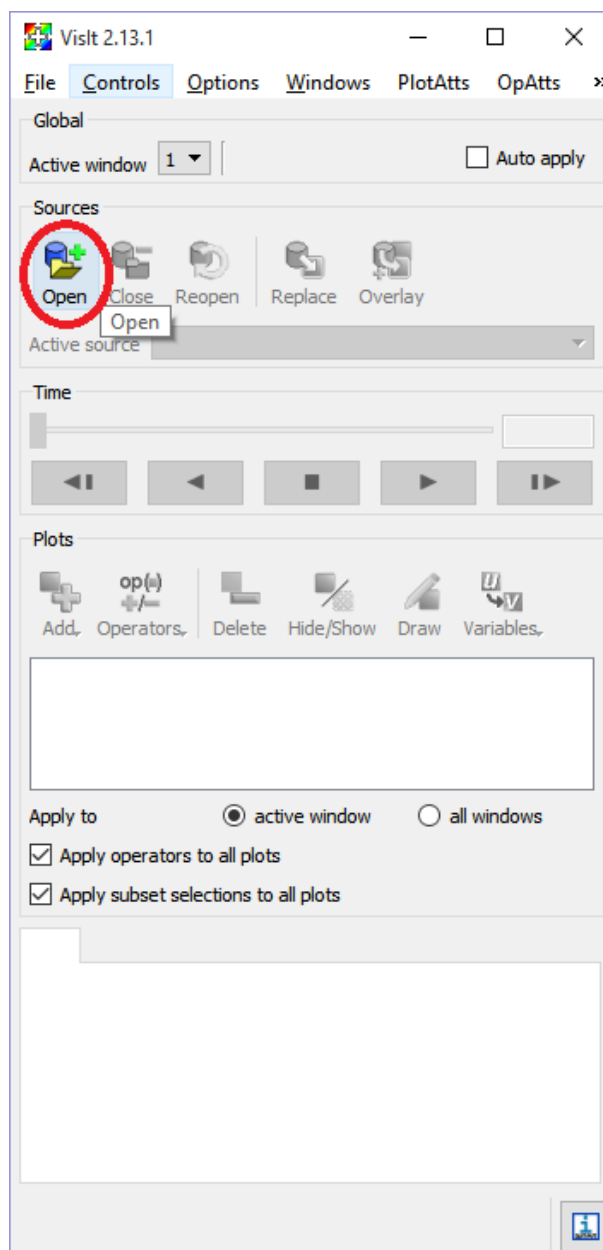


Figura C.2: Abrindo arquivo no VisIt.

Na janela “File open”, navegue por suas pastas e procure o arquivo desejado em seu diretório.

Selecione o tipo de arquivo que deseja abrir, do mesmo modo que está ilustrado na Figura C.3.

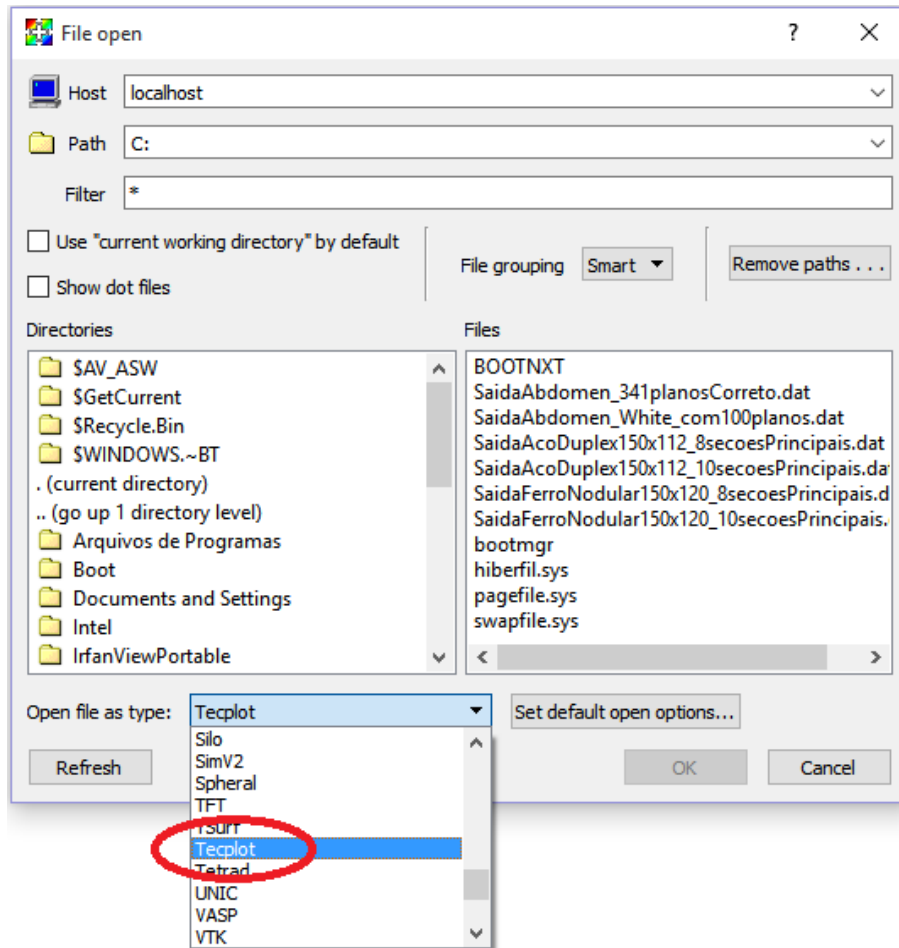


Figura C.3: Selecionando o tipo de arquivo.

Selecione o arquivo de interesse e clique no botão “OK”, conforme a Figura C.4.

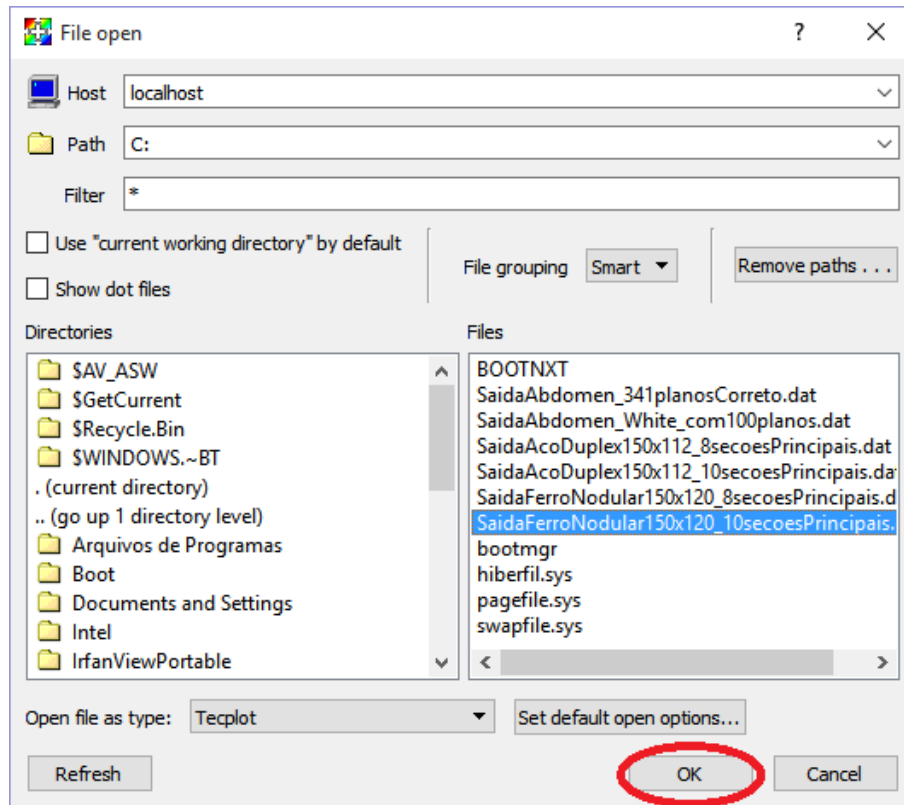


Figura C.4: Selecionando arquivo para abrir no VisIt.

Uma nova janela vai abrir (Figura C.5) com opções de abrir o arquivo desejado em modo “serial” ou em modo “parallel”.

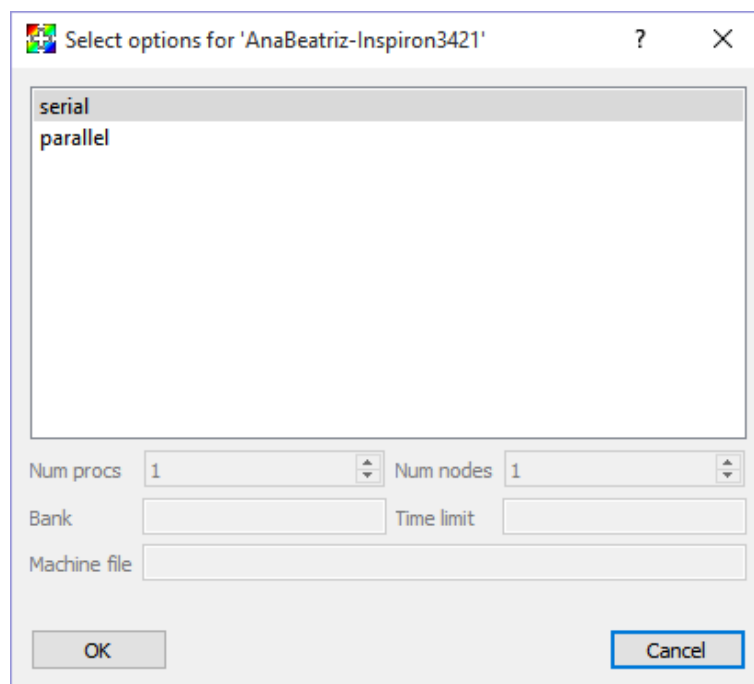


Figura C.5: Selecionando modo de abertura do arquivo.

O arquivo aparecerá aberto na tela, como consta na Figura C.6. Abaixo, na seção “Plots”, clique em “Add”.

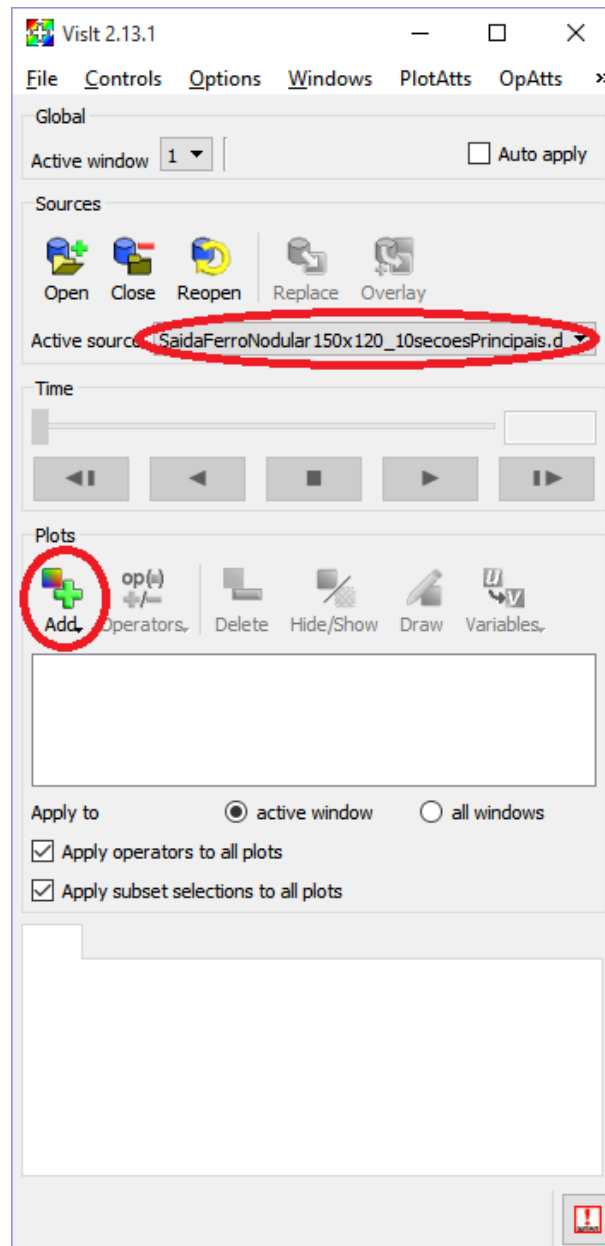


Figura C.6: Arquivo aberto na seção “Sources” em frente “Active source”.

Após clicar no botão “Add”, selecione “Contour” e, em seguida clique em “BLUE”, “GREEN” ou “RED”, conforme apresentado na Figura C.7.

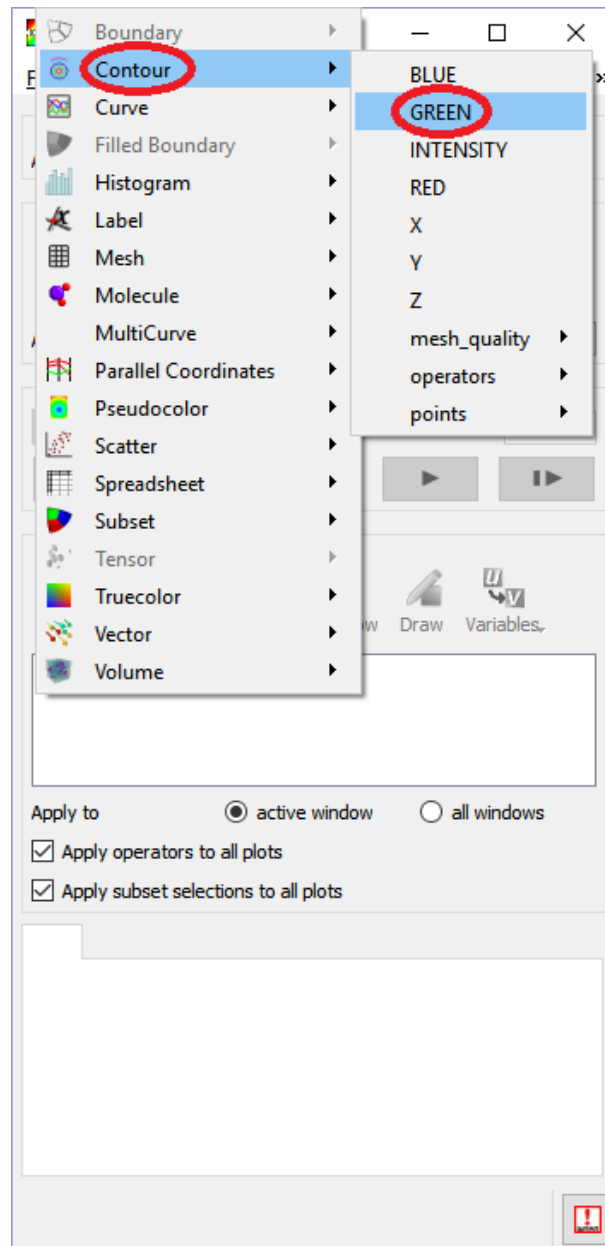


Figura C.7: Adicionando opções para visualização.

A opção de visualização aparecerá na seção “Plots”. Selecione-a e clique em “Draw” para visualizar, seguindo a Figura C.8.

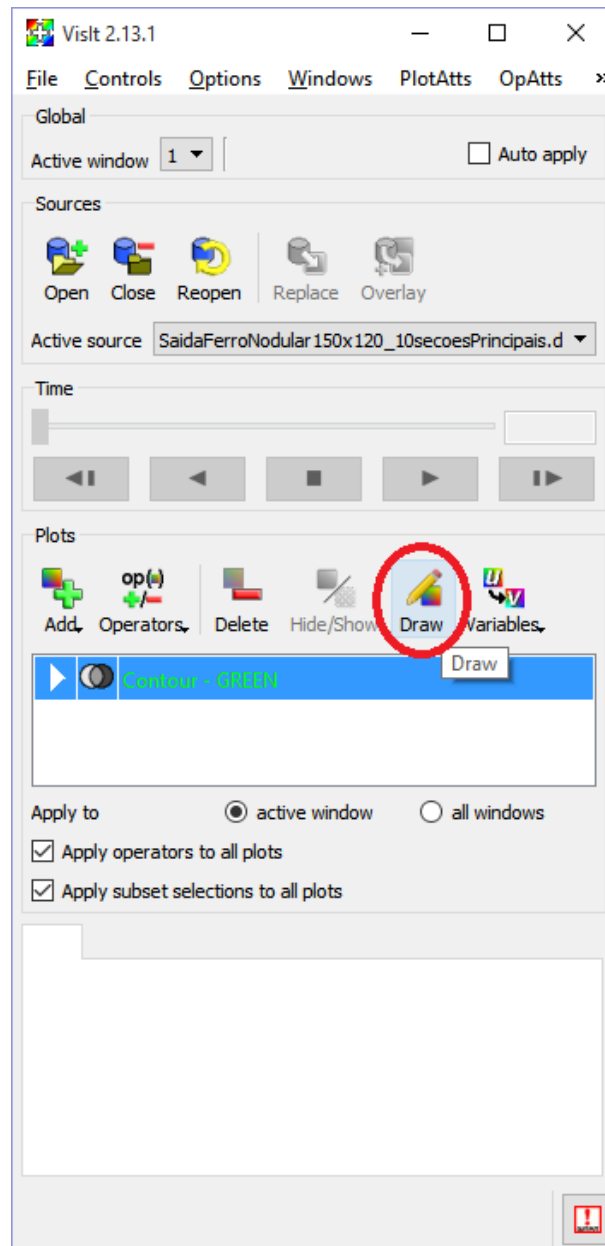


Figura C.8: Ativando a visualização no VisIt.

O primeiro resultado é apresentado na Figura C.9.

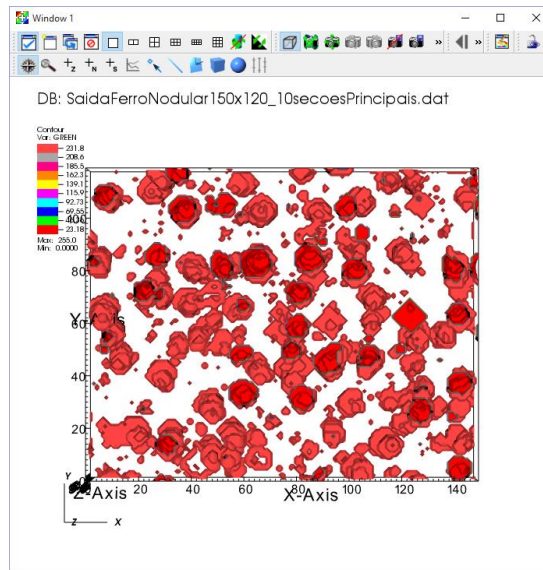


Figura C.9: Resultado: esferas de grafite da microestrutura de Ferro Fundido Nodular.

Também é possível rotacionar os eixos, com o mouse, para escolher outro ângulo para visualização, conforme a Figura C.10. Segure o botão esquerdo do mouse e mexa-o.

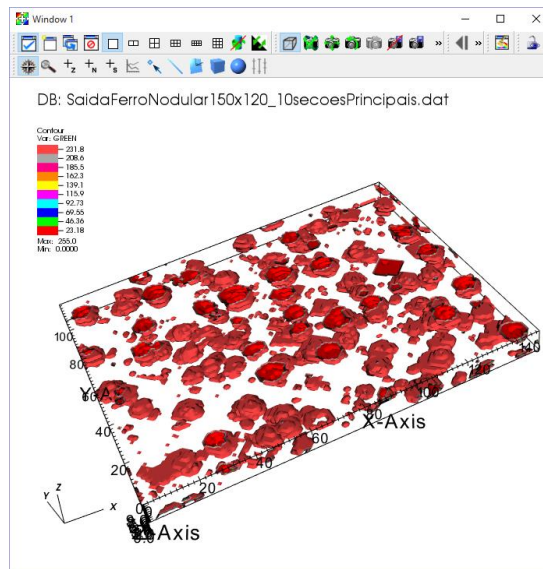


Figura C.10: Rotação da imagem.

Para mudar a cor dos nódulos, clique duas vezes em “Contour - GREEN” que foi criado anteriormente. Uma janela de configurações abrirá, conforme é mostrado na Figura C.11.

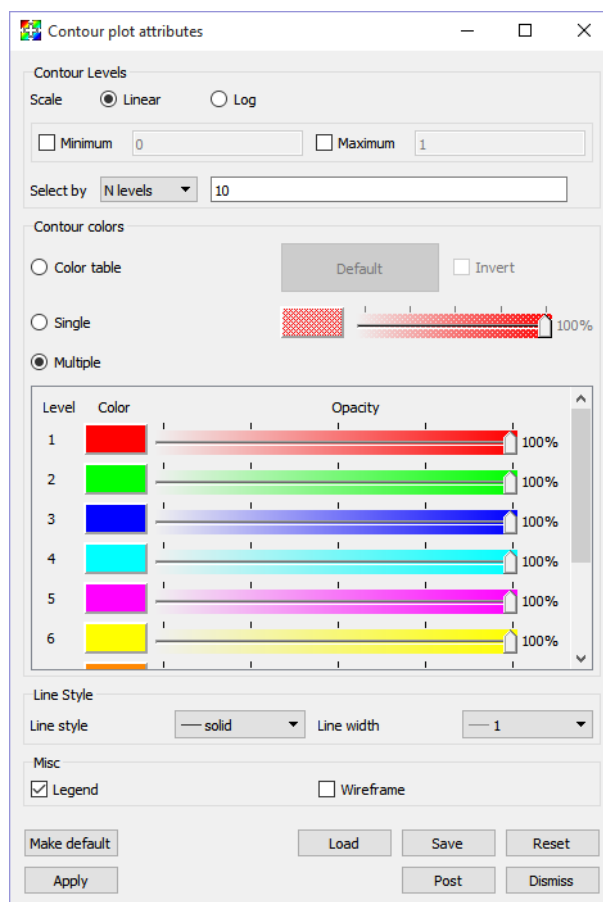


Figura C.11: Modificando a cor das esferas de grafite.

Selecione a opção “Single” e depois clique na cor vermelha que está em frente esta opção. Uma janela menor com outras cores aparecerá. Selecione a cor desejada e clique em “Apply” no final da janela, conforme mostrado na Figura C.12.

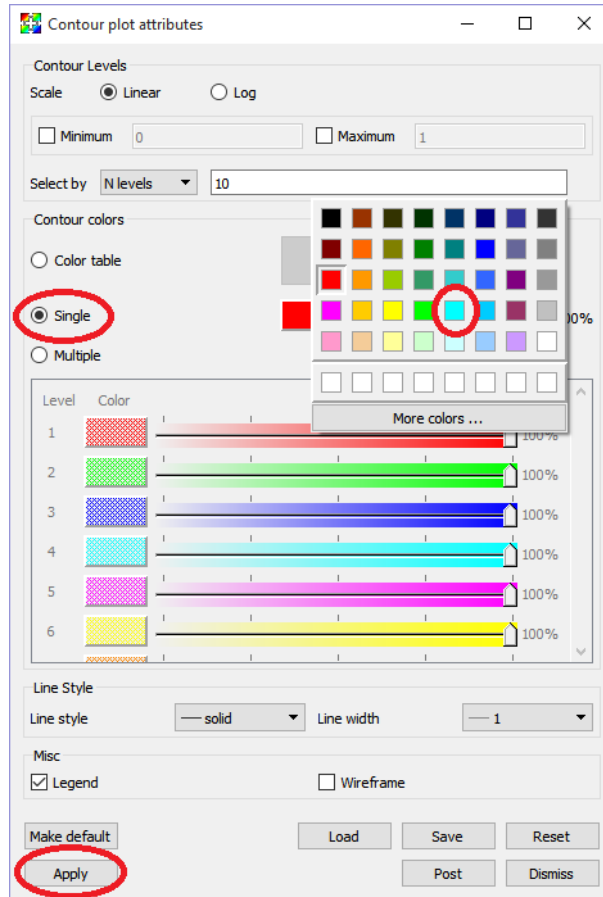


Figura C.12: Modificando a cor das esferas de grafite.

Por fim, tem-se um resultado próximo aos resultados apresentados na seção 5.1 do capítulo 5, vide Figura C.13.

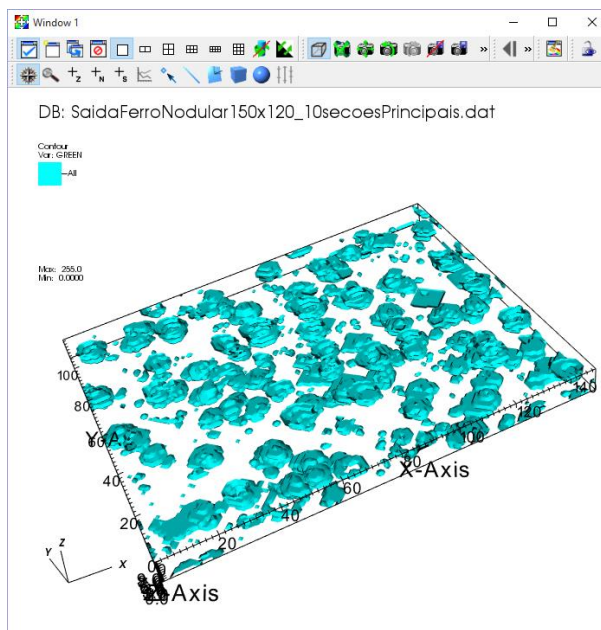
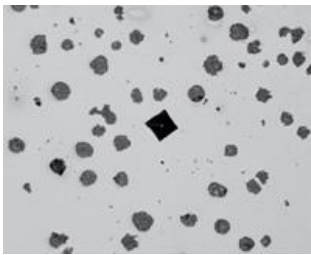
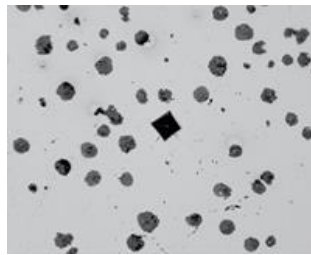


Figura C.13: Resultado: Visualização da microestrutura de Ferro Fundido Nodular..

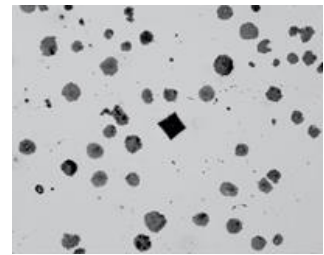
## ANEXO A - Imagens de Ferro Fundido Nodular



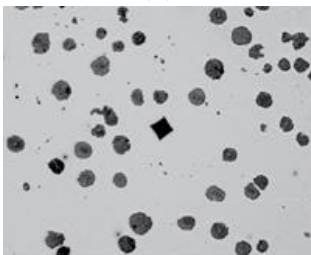
(1)



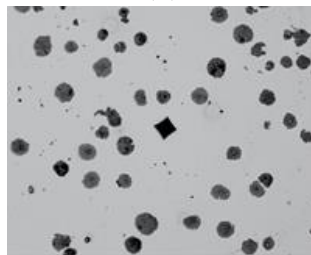
(2)



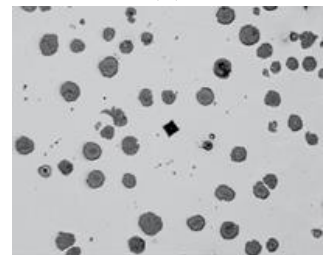
(3)



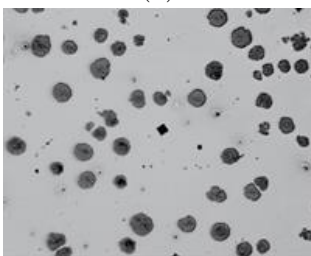
(4)



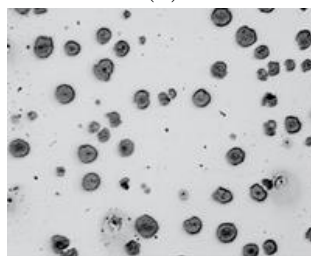
(5)



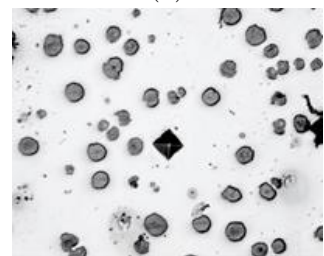
(6)



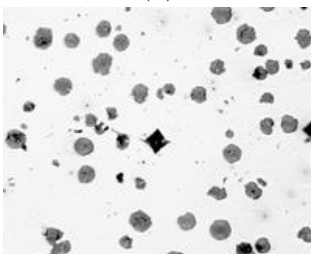
(7)



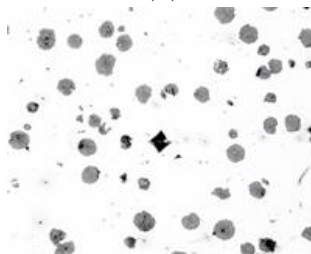
(8)



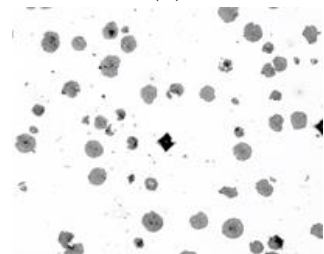
(9)



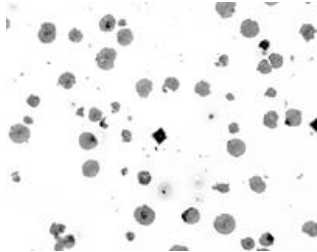
(10)



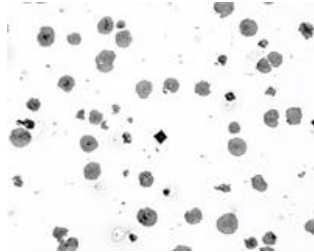
(11)



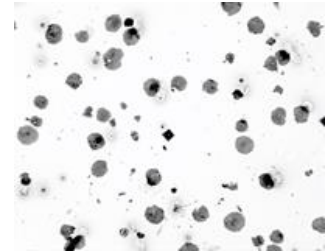
(12)



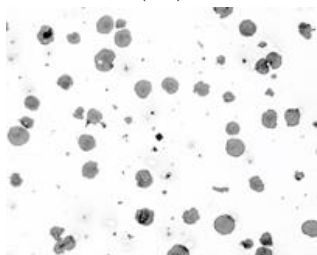
(13)



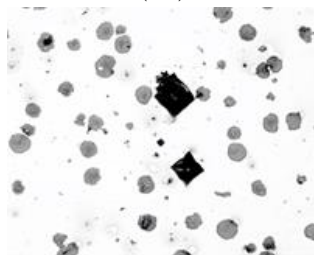
(14)



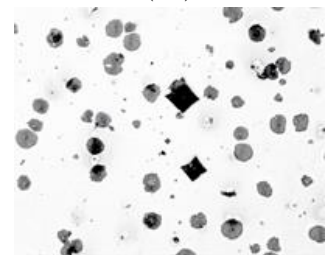
(15)



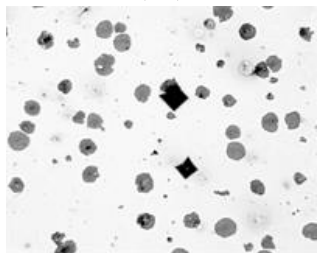
(16)



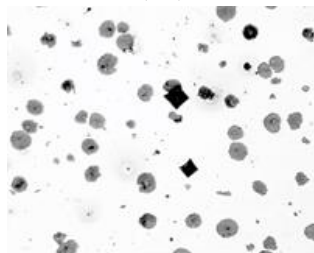
(17)



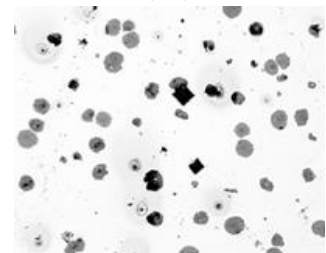
(18)



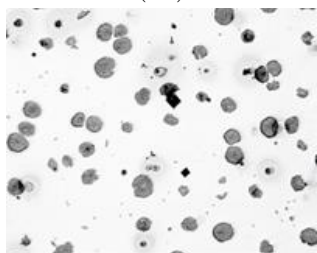
(19)



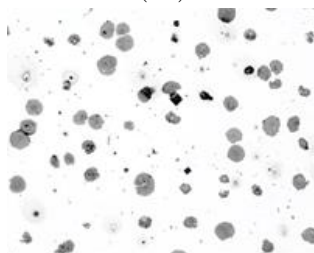
(20)



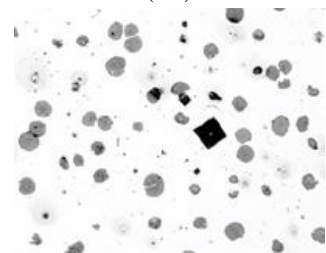
(21)



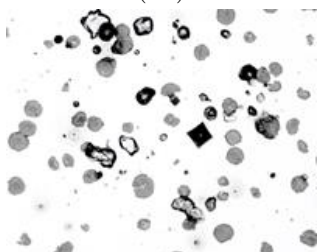
(22)



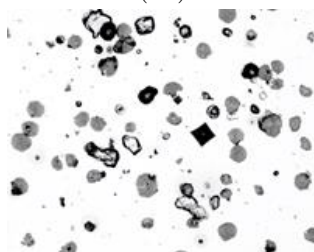
(23)



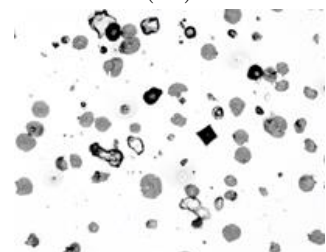
(24)



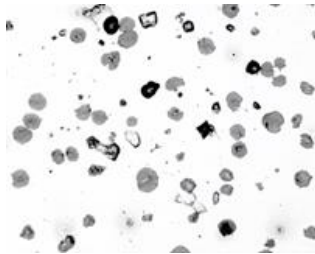
(25)



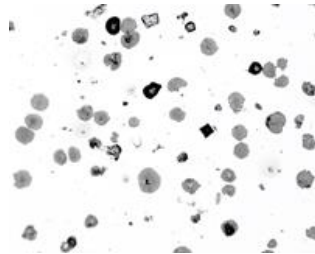
(26)



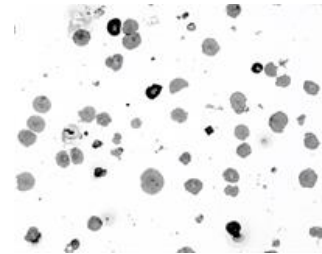
(27)



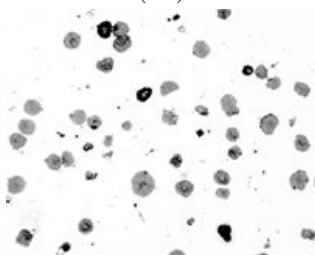
(28)



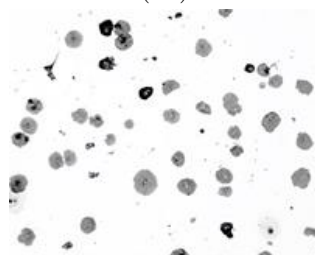
(29)



(30)



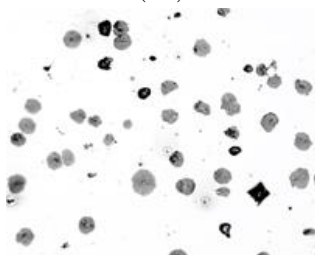
(31)



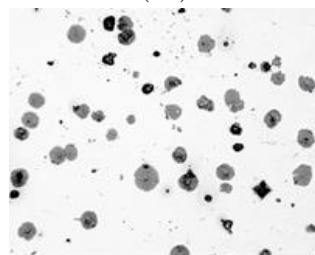
(32)



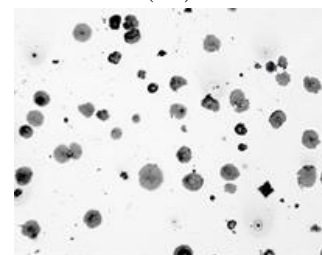
(33)



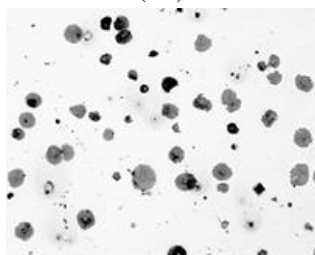
(34)



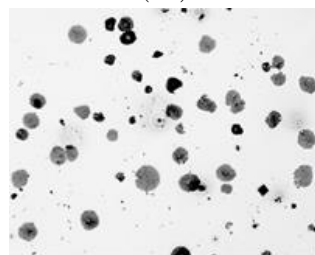
(35)



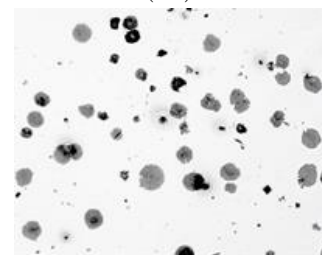
(36)



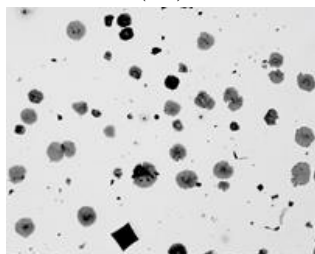
(37)



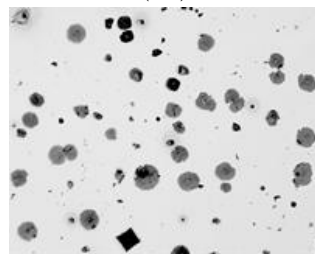
(38)



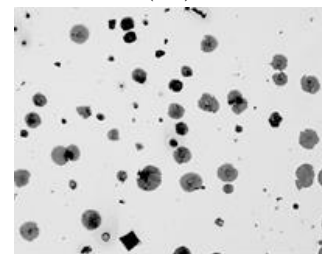
(39)



(40)



(41)



(42)

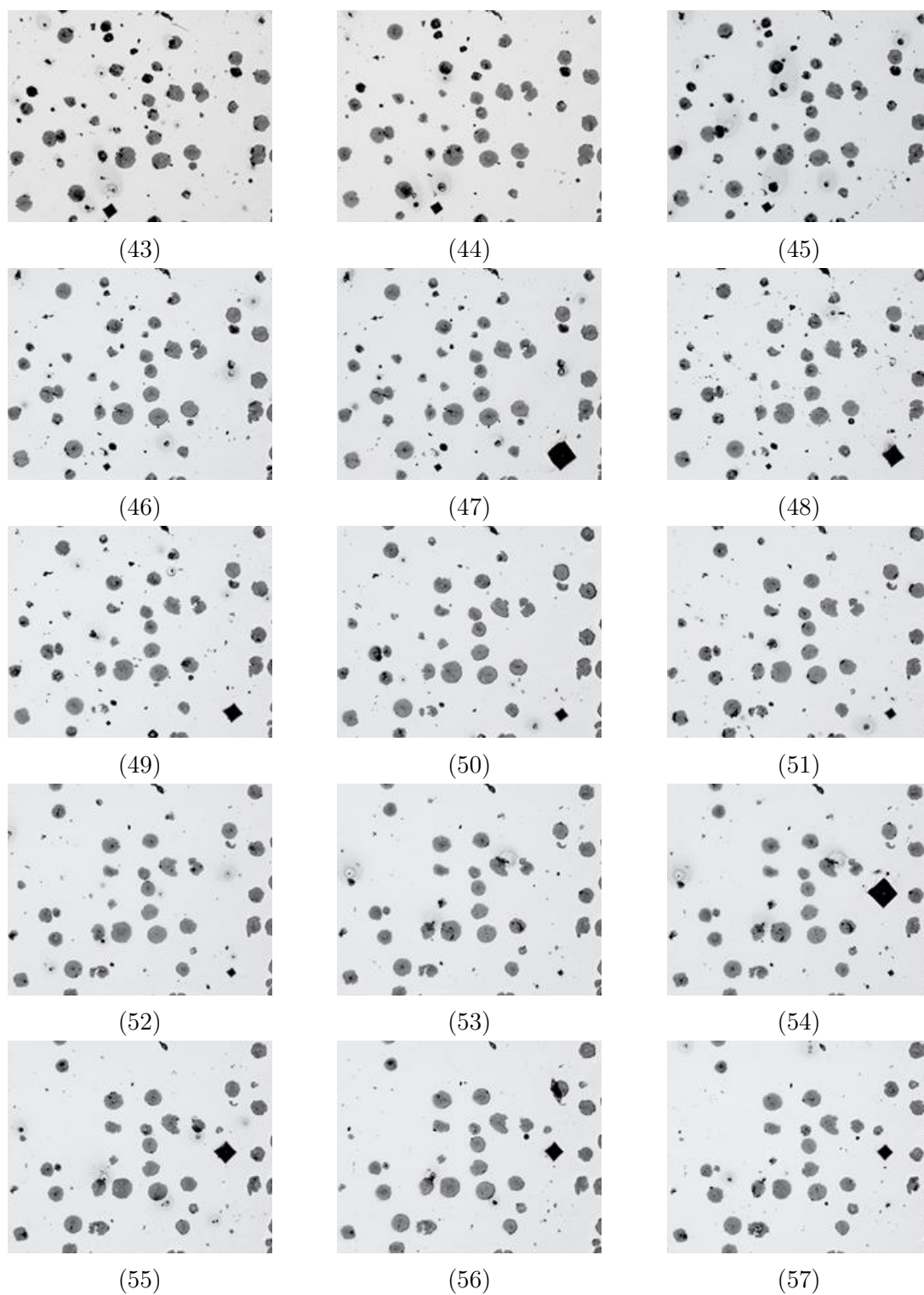


Figura A.1: Imagens de Ferro Fundido Nodular obtidas através da técnica de Seccionamento em Série.

Fonte: Autoria de [24].

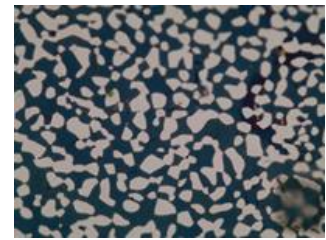
## ANEXO B - Imagens de Aço Inoxidável Duplex



(1)



(2)



(3)



(4)



(5)



(6)



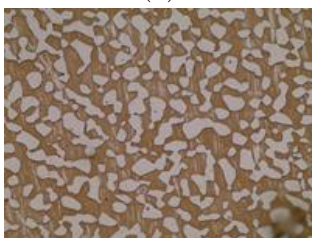
(7)



(8)



(9)



(10)



(11)



(12)



(13)



(14)



(15)



(16)



(17)



(18)



(19)



(20)



(21)



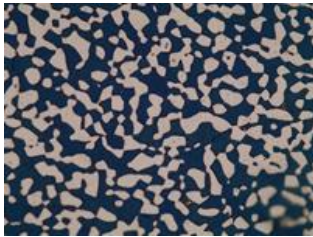
(22)



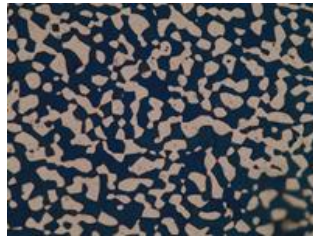
(23)



(24)



(25)



(26)



(27)



(28)



(29)



(30)



(31)



(32)



(33)



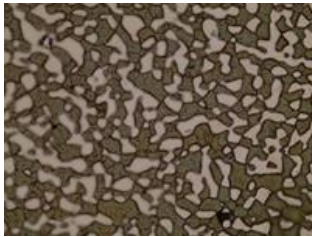
(34)



(35)



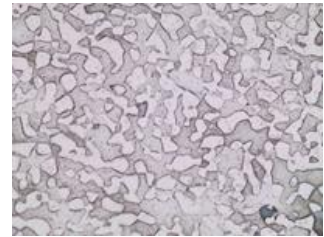
(36)



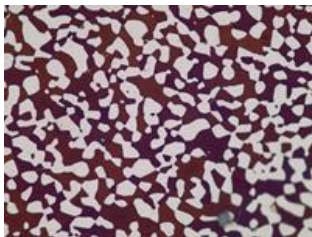
(37)



(38)



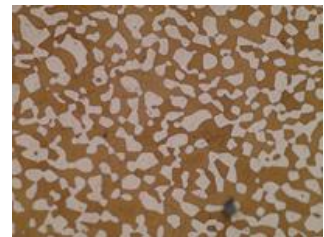
(39)



(40)



(41)



(42)



(43)



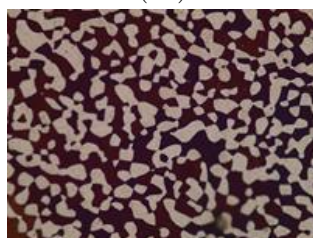
(44)



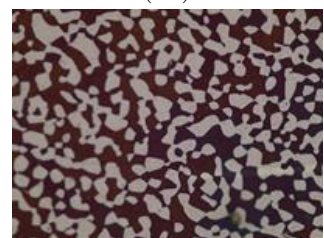
(45)



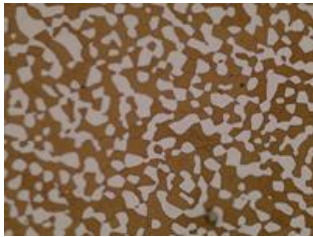
(46)



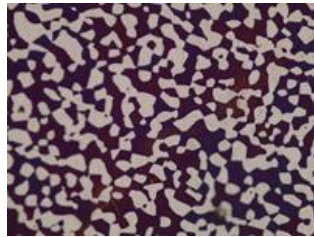
(47)



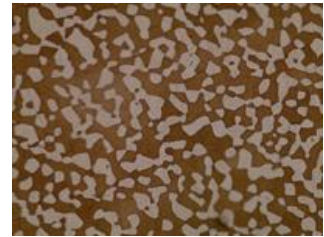
(48)



(49)



(50)



(51)



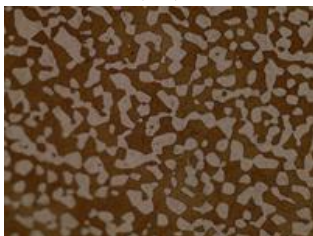
(52)



(53)



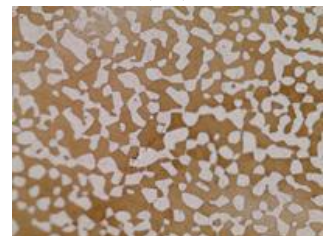
(54)



(55)



(56)



(57)



(58)



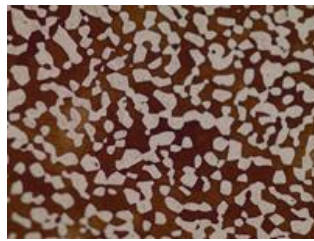
(59)



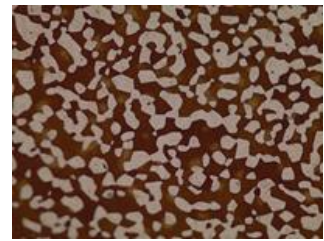
(60)



(61)



(62)



(63)



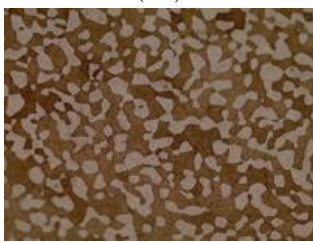
(64)



(65)



(66)



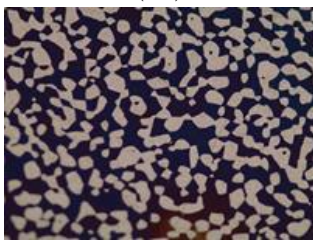
(67)



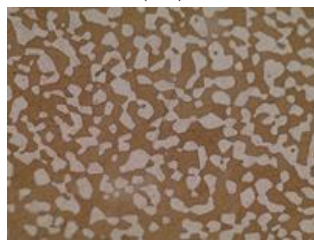
(68)



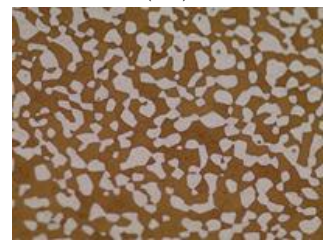
(69)



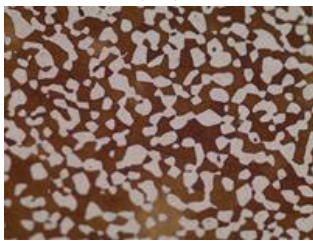
(70)



(71)



(72)



(73)



(74)



(75)

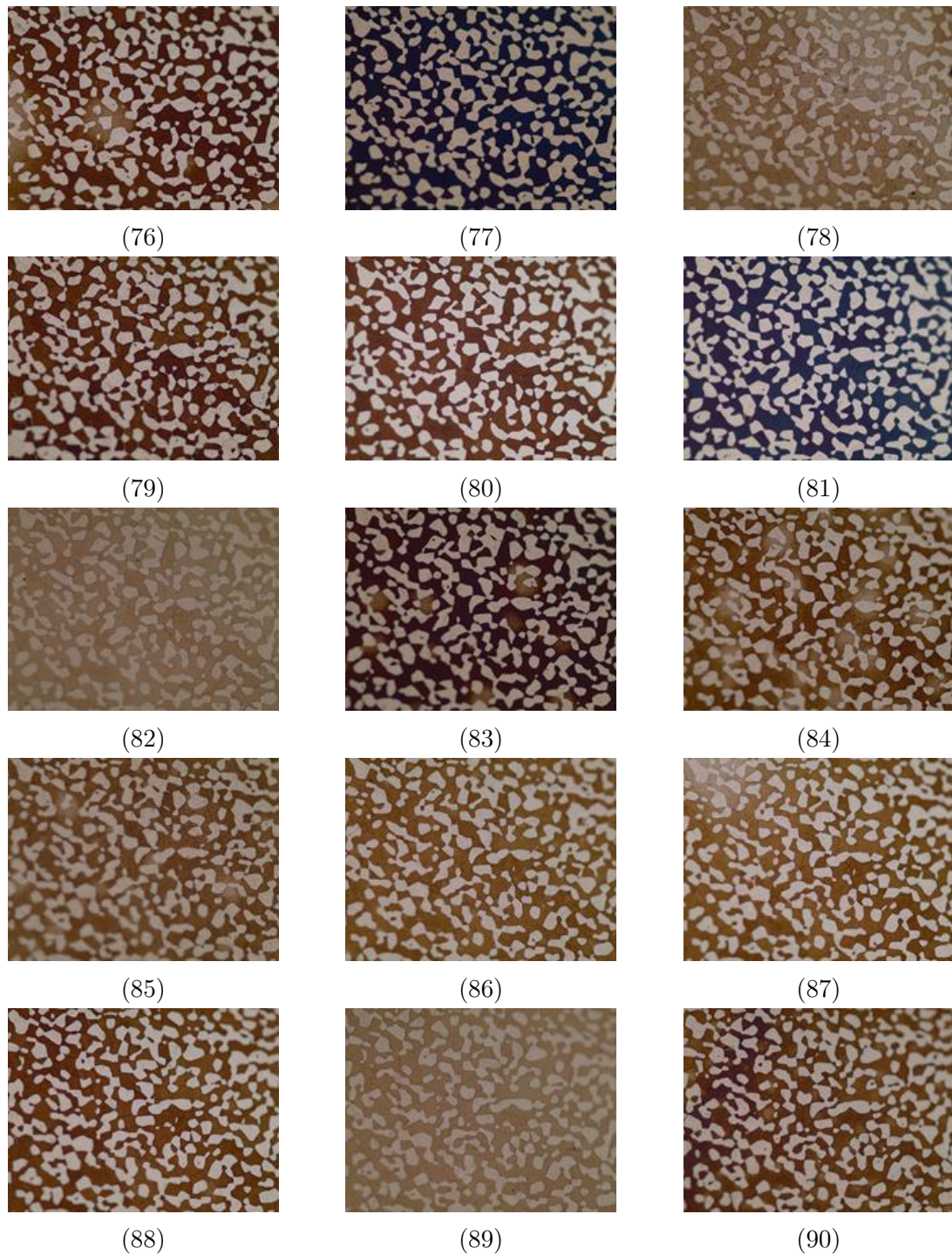


Figura B.1: Imagens de Aço Inoxidável Duplex obtidas através da técnica de Seccionamento em Série.

Fonte: Autoria de [11].