



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO ACADÊMICO DO AGRESTE  
NÚCLEO DE FORMAÇÃO DOCENTE  
CURSO DE FÍSICA-LICENCIATURA

FELIPE ALVES DOS SANTOS

**PROCESSAMENTO DE IMAGENS DIGITAIS DE TOMOGRAFIA POR COERÊNCIA  
ÓTICA PARA CARACTERIZAÇÃO DE MATERIAIS POROSOS**

CARUARU  
2025

FELIPE ALVES DOS SANTOS

**PROCESSAMENTO DE IMAGENS DIGITAIS DE TOMOGRAFIA POR COERÊNCIA  
ÓTICA PARA CARACTERIZAÇÃO DE MATERIAIS POROSOS**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Física-Licenciatura do Centro Acadêmico Agreste (CAA) da Universidade Federal de Pernambuco (UFPE), na modalidade de monografia, como requisito parcial para a obtenção do grau de Licenciado em Física.

**Área de concentração:** Física Aplicada

**Orientador:** Prof. Dr. Sérgio de Lemos Campello

CARUARU

2025

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Santos, Felipe Alves dos .

Processamento de imagens digitais de tomografia por coerência ótica para  
caracterização de materiais porosos / Felipe Alves dos Santos. - Caruaru, 2025.  
45p : il.

Orientador(a): Sergio de Lemos Campello  
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de  
Pernambuco, Centro Acadêmico do Agreste, Física - Licenciatura, 2025.  
Inclui referências, apêndices.

1. processamento de imagens digitais. 2. materiais porosos. 3. porosidade. I.  
Campello, Sergio de Lemos . (Orientação). II. Título.

530 CDD (22.ed.)

FELIPE ALVES DOS SANTOS

**PROCESSAMENTO DE IMAGENS DIGITAIS DE TOMOGRAFIA POR COERÊNCIA  
ÓTICA PARA CARACTERIZAÇÃO DE MATERIAIS POROSOS**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Física-Licenciatura do Centro Acadêmico Agreste (CAA) da Universidade Federal de Pernambuco (UFPE), na modalidade de monografia, como requisito parcial para a obtenção do grau de Licenciado em Física.

Aprovado em: 15/08/2025

**BANCA EXAMINADORA**

---

Prof. Dr. Sérgio de Lemos Campello  
Orientador(a) - CAA/UFPE

---

Prof. Dr. João Francisco Liberato de Freitas  
Examinador(a) interno(a) - CAA/UFPE

---

Prof. Dr. Marcos Luiz Henrique  
Examinador(a) interno(a) - CAA/UFPE

## **AGRADECIMENTOS**

A Deus, fonte suprema da vida, que me concedeu a graça de uma nova existência e a oportunidade de contemplar, com renovado olhar, a beleza de cada amanhecer.

À minha esposa, companheira de todas as jornadas, cujo amor constante e apoio inabalável foram o alicerce sobre o qual ergui minhas forças e esperanças.

Aos meus pais, exemplos de dedicação e sacrifício, que não mediram esforços, fazendo do possível e até do impossível, para que eu pudesse trilhar este caminho.

Aos mestres que, com paciência e firmeza, compartilharam seu saber e, sobretudo, àqueles que, com sua luz e inspiração, marcaram de forma indelével minha formação: Anderson Rodrigo, Diana Patrícia, Gustavo Camelo Neto, Luiz Gustavo e Sergio Campello.

Ao professor Ernesto, que me confiou o acesso ao Laboratório de Ensino de Física (LAEF), permitindo-me pôr em dia meus estudos e aprofundar meu compromisso com a docência.

Aos colegas de curso, cuja amizade e parceria suavizaram os desafios da caminhada acadêmica e encheram de alegria meus dias no CAA. Em especial, minha sincera gratidão a Alex Pedro e Vagner Diego, pela amizade fraterna e presença constante.

## RESUMO

Os materiais porosos são caracterizados pela presença de espaços vazios no seu interior. A depender da região de interesse, todo material pode ser considerado poroso. Uma maneira de identificar a porosidade é através do processamento de imagens digitais, o qual busca, de forma não invasiva, determinar a porosidade do objeto em estudo. Este processamento de imagens refere-se ao conjunto de técnicas e métodos utilizados para manipular e analisar imagens digitais em duas ou mais dimensões, utilizando computadores e algoritmos específicos. Deste modo, este trabalho tem como objetivo analisar imagens de rochas fontes de petróleo para extrair um valor para a porosidade deste material. As imagens do material poroso foram obtidas por meio da técnica de OCT (Optical Coherence Tomography) e a análise foi feita através da aplicação de algoritmo específico para a determinação dos poros, utilizando a linguagem MATLAB<sup>®</sup> como ferramenta. Para o processamento das imagens, inicialmente foi aplicado um filtro de redução de ruído, obtendo uma melhor definição da região da rocha, seguido do processo de binarização por meio do método de Otsu. Após a binarização, aplicou-se a segmentação da região de interesse, resultando em uma nova imagem, a qual separa a informação proveniente da rocha e do poro. Em seguida, realizou-se a contagem dos pixels onde foram detectados os poros e, finalmente, a determinação da porosidade. Após o processamento das imagens, identificou-se a presença de porosidade no material rochoso, que em uma imagem binária é representada por pixels de valor zero (pixels pretos). Assim, foi calculado o valor da porosidade através da razão de pixels pretos pelo total de pixels das imagens, resultando no valor de 25,33%, o qual corresponde a uma boa aproximação da porosidade real do material analisado, que é de 17,45%. Também foi determinada a distribuição dos tamanhos dos poros. Os resultados mostram que o algoritmo é uma ferramenta funcional na determinação de porosidade de imagens de rochas fonte.

**Palavras-chave:** processamento de imagens digitais; materiais porosos; porosidade.

## ABSTRACT

Porous materials are characterized by the presence of void spaces within their structure. Depending on the region of interest, any material can be considered porous. One method to identify porosity is through digital image processing, which seeks to non-invasively determine the porosity of the object under study. This image processing refers to the set of techniques and methods used to manipulate and analyze digital images in two or more dimensions, using computers and specific algorithms. Thus, this work aims to analyze images of petroleum source rocks to extract a value for the porosity of this material. The images of the porous material was obtained using Optical Coherence Tomography (OCT) technique, and analysis was performed by applying a specific algorithm for pore determination, using MATLAB<sup>®</sup> language. For image processing, a noise reduction filter was initially applied to obtain a better definition of the rock region, followed by binarization using Otsu's method. After binarization, the region of interest was segmented, resulting in a new image that separates the information from the rock and the pore. The pixels, where pores were detected, were counted and finally the porosity was determined. After processing, porous was identified in the rock material, represented by zero-value pixels (black pixels) in a binary image. Thus, the porosity value was calculated by the ratio of black pixels to the total number of pixels in the images, resulting in a value of 25,33%, which corresponds to a good approximation of the real porosity of the material analyzed, which is 17,45%. The pore size distribution was also determined. The results show that the algorithm is a functional tool in determining the porosity of source rock images.

**Keywords:** digital image processing; porous materials; porosity.

## LISTA DE FIGURAS

Figura 1 – Diferentes tonalidades de cor para uma mesma imagem. . . . .	12
Figura 2 – Conversão de um pixel RGB para escala de cinza. . . . .	13
Figura 3 – Processo de filtragem por convolução de matrizes . . . . .	14
Figura 4 – Imagem com presença de ruído e resultado da filtragem por filtro de média. . . . .	15
Figura 5 – Imagem em escala de cinza formada pela matriz $M$ . . . . .	17
Figura 6 – Imagem binarizada da matriz $M$ . . . . .	19
Figura 7 – Imagem em escala de cinza e sua binarização para diferentes valores de $l$ . . . . .	19
Figura 8 – Imagem da amostra e representação da região analisada. . . . .	24
Figura 9 – Imagem em escala da rocha 266. . . . .	25
Figura 10 – Representação esquemática do algoritmo de processamento de ima- gens para análise de porosidade e distribuição dos poros. . . . .	29
Figura 11 – Imagem original da rocha obtida por OCT. . . . .	30
Figura 12 – Aplicação do filtro de suavização sobre imagem original da rocha. . .	31
Figura 13 – Imagem binarizada com destaque para as regiões sólidas (branco). . .	31
Figura 14 – Máscara de poros (preto: rocha; branco: poros detectados). . . . .	32
Figura 15 – Região final considerada para o cálculo da porosidade. . . . .	33
Figura 16 – Granulometria acumulada dos poros ao longo de 160 imagens. . . .	34

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>2</b>	<b>INTRODUÇÃO AO PROCESSAMENTO DE IMAGENS DIGITAIS</b>	<b>11</b>
2.1	ETAPAS DO PROCESSAMENTO DE IMAGENS	11
2.2	TIPOS DE IMAGENS	12
2.3	CONVERSÃO PARA ESCALA DE CINZA	12
2.4	FILTRAGEM E REALCE	13
2.5	SEGMENTAÇÃO E EXTRAÇÃO DE REGIÕES DE INTERESSE	15
<b>2.5.1</b>	<b>Segmentação por limiarização</b>	<b>15</b>
<b>2.5.2</b>	<b>Segmentação por região de interesse</b>	<b>19</b>
<b>3</b>	<b>MATLAB<sup>®</sup> COMO FERRAMENTA PARA ANÁLISE E PROCESSAMENTO DE DADOS</b>	<b>20</b>
3.1	FUNCIONALIDADES DO AMBIENTE MATLAB <sup>®</sup>	20
3.2	TRABALHANDO COM MATRIZES E IMAGENS	21
3.3	VANTAGENS E LIMITAÇÕES DO MATLAB <sup>®</sup>	22
<b>4</b>	<b>METODOLOGIA</b>	<b>24</b>
4.1	ANÁLISE DAS IMAGENS E CARACTERIZAÇÃO DA AMOSTRA	24
4.2	NOME E CAMINHO DAS IMAGENS	25
4.3	ACUMULADORES E MATRIZ DE ARMAZENAMENTO	25
4.4	MATRIZ DE FILTRO PONDERADA	26
4.5	LEITURA E PROCESSAMENTO DAS IMAGENS	26
<b>4.5.1</b>	<b>Leitura das imagens</b>	<b>26</b>
<b>4.5.2</b>	<b>Conversão em Escala de cinza</b>	<b>26</b>
<b>4.5.3</b>	<b>Filtragem e binarização</b>	<b>27</b>
<b>4.5.4</b>	<b>Medianas dos primeiros pixels brancos</b>	<b>27</b>
<b>4.5.5</b>	<b>Máscara de poros</b>	<b>27</b>
<b>4.5.6</b>	<b>Porosidade acumulada</b>	<b>28</b>
<b>4.5.7</b>	<b>Granulometria</b>	<b>28</b>
4.6	POROSIDADE FINAL	28
4.7	GRANULOMETRIA TOTAL	28
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>30</b>
5.1	IMAGEM ORIGINAL E SUAVIZAÇÃO	30
5.2	IMAGEM BINARIZADA	31
5.3	MÁSCARA DE POROS DETECTADOS	32
5.4	POROSIDADE CONSIDERADA	32
5.5	POROSIDADE CALCULADA	33
5.6	GRANULOMETRIA MORFOLÓGICA ACUMULADA	34

<b>6</b>	<b>CONCLUSÕES E PERSPECTIVAS . . . . .</b>	<b>36</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>37</b>
	<b>APÊNDICE A – ALGORITMO UTILIZADO NO PROCESSAMENTO DAS IMAGENS . . . . .</b>	<b>40</b>
	<b>APÊNDICE B – CERTIFICADO: XXXVIII ENCONTRO DE FÍSICA DO NORTE E NORDESTE . . . . .</b>	<b>44</b>
	<b>APÊNDICE C – CERTIFICADO: 32º CONGRESSO DE INICIAÇÃO CIENTÍFICA . . . . .</b>	<b>45</b>

## 1 INTRODUÇÃO

O processamento digital de imagens envolve a manipulação de imagens digitais por computadores para melhorar a qualidade visual, apoiar sistemas autônomos em controle de qualidade e automação, e otimizar a transmissão e armazenamento de imagens. A tecnologia é aplicada em diversos dispositivos, como sensores de satélite e câmeras digitais (Jingar; Panchal; Oza, 2024). As principais tarefas de processamento de imagem são redução de ruído, aprimoramento de imagem, classificação de pixels e agrupamento de pixels (Kaestner; Lehmann; Stampanoni, 2008).

Os materiais que possuem espaços vazios em seu interior, são caracterizados como materiais porosos. A porosidade fornece uma fração da quantidade de poros em uma amostra de material em relação ao volume total da amostra (Nimmo, 2004). Há séculos, os cientistas buscam compreender como os fluidos caminham através de meios porosos devido às diversas aplicações possíveis em variadas áreas do conhecimento, como engenharia de materiais, mecânica, química, civil e biomédica (Vafai, 2005).

A tomografia de coerência óptica, do inglês **Optical Coherence Tomography**, é uma técnica de imagem emergente aplicada em diversas investigações biológicas, médicas e de materiais, que utiliza a medição óptica para detectar a luz retroespalhada com alta resolução e ampla faixa dinâmica, com base no atraso óptico. Diferentemente do ultrassom, a OCT se beneficia da alta velocidade da luz para obter suas medições (Boppart, 2003).

A OCT é uma técnica não invasiva que usa interferência de luz para criar imagens bidimensionais e tridimensionais de amostras, sejam elas transparentes ou não. Ao combinar feixes de luz de referência com luz dispersa pela amostra, a técnica gera padrões de interferência que são então transformados em imagens detalhadas com alta precisão (Campello et al., 2014). A aquisição de imagem não destrutiva, através de OCT, tem a vantagem sobre a imagem destrutiva de que a mesma amostra pode ser escaneada repetidamente sob diferentes condições iniciais ou um processo pode ser monitorado espacial e temporalmente (Kaestner; Lehmann; Stampanoni, 2008).

Por ser uma técnica de obtenção de imagens não invasiva e não destrutiva, a tomografia por coerência óptica se distingue por ser uma ferramenta poderosa para caracterização estrutural de vários materiais como polímeros, biomateriais e meios porosos orgânicos. Além disso, abre possibilidade para caracterização de materiais porosos não-orgânicos, entre eles, as rochas fonte de petróleo (Campello et al., 2014).

As técnicas de processamento de imagens surgem como uma solução não destrutiva para análises de imagens de OCT, podendo ser utilizadas para extrair informações qualitativas e quantitativas das amostras.

## 2 INTRODUÇÃO AO PROCESSAMENTO DE IMAGENS DIGITAIS

Uma definição de imagem pode ser dada como uma função que varia em duas direções (altura e largura) onde cada ponto possui um valor que representa a sua "cor" ou intensidade. Esses pontos são identificados por coordenadas  $(x, y)$ , e o valor associado a cada um deles indica o quão claro ou escuro ele é, algo que chamamos de nível de cinza ou intensidade. Quando essas posições  $(x, y)$  e os valores de intensidade são contáveis, ou seja, têm uma quantidade limitada e bem definida, estamos lidando com uma imagem digital. O processamento de imagens digitais é justamente a área que estuda como tratar essas imagens com a ajuda de computadores. Cada um desses pequenos pontos que compõem a imagem digital pode ser chamado de elemento da imagem ou pixel, que é o nome mais comum. Em resumo, um pixel é a menor parte de uma imagem digital, contendo tanto a localização quanto a informação visual daquele ponto específico (Gonzalez; Woods, 2006).

A palavra pixel surge como abreviação de *picture element*, ou seja, elemento de imagem. É a menor unidade utilizada para representar uma imagem digital (Sonka, 2015). Dizemos que uma imagem está em escala de cinza quando os dados da imagem consistem em apenas um canal de cor, que representa o brilho, a intensidade ou a densidade da imagem (Burger; Burge, 2016).

### 2.1 ETAPAS DO PROCESSAMENTO DE IMAGENS

Ainda, segundo Gonzalez e Woods (2006), as etapas clássicas do processamento de imagens incluem:

1. **Aquisição:** captura da imagem por sensores digitais, como câmeras fotográficas, microscópios digitais ou equipamentos de tomografia computadorizada;
2. **Pré-processamento:** aplicação de filtros para correção de ruídos, como o filtro mediana para remover interferências em imagens médicas, ou a normalização de brilho e contraste em fotos subexpostas;
3. **Segmentação:** divisão da imagem em regiões significativas para identificação de objetos ou padrões, como separar áreas de vegetação e áreas urbanas em imagens de satélite, ou isolar células em imagens microscópicas;
4. **Extração de características:** medição de atributos como forma (circularidade de grãos), tamanho (diâmetro de partículas), borda (detecção de contornos de peças mecânicas) e textura (rugosidade de uma superfície metálica);
5. **Pós-processamento e interpretação:** visualização de resultados, como realce de regiões de interesse em exames de ressonância magnética, tomada de decisão

baseada em reconhecimento de padrões para controle de qualidade industrial, ou integração das informações extraídas em modelos computacionais para previsão ou simulação.

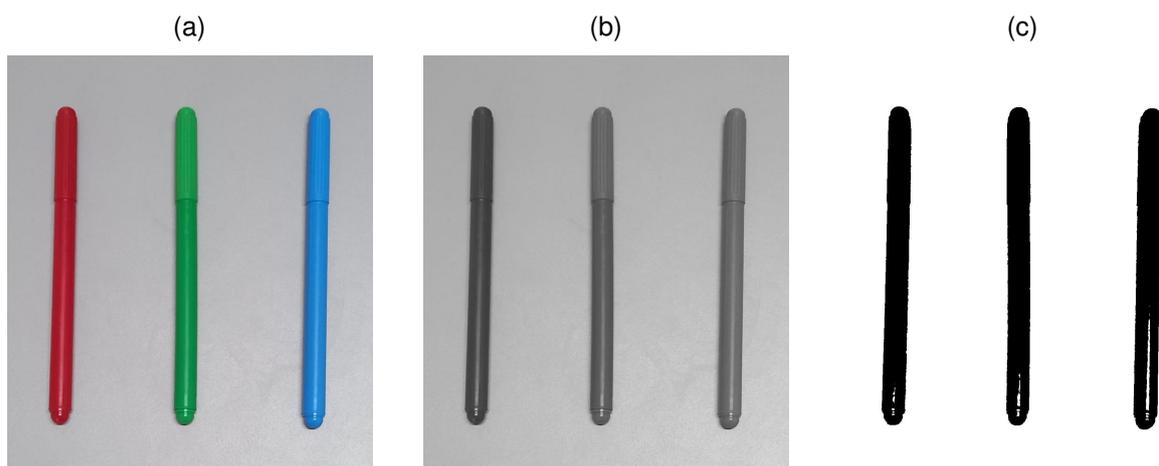
## 2.2 TIPOS DE IMAGENS

As imagens digitais são classificadas com o tipo de informação que contêm:

- **Binárias:** possuem apenas dois valores (0 e 1), representando preto e branco. São ideais para análise de contornos e formas após a segmentação;
- **Escala de cinza:** cada pixel possui um valor entre 0 e 255, permitindo uma representação tonal da intensidade luminosa;
- **Coloridas:** também conhecida como imagem RGB, que é um tipo de imagem digital formada pela combinação de três cores básicas: vermelho (Red), verde (Green) e azul (Blue), daí vem a sigla RGB.

Cada tipo de imagem possui vantagens e aplicações específicas de acordo com a complexidade do fenômeno analisado. A figura 1 mostra um exemplo de uma mesma imagem em diferentes tonalidades de cor: (a) colorida (combinação de cores vermelha, verde e azul, totalizando  $(2^8)^3 = 16.777.216$  cores; (b) escala de cinza (tonalidade entre 0 e 255 tons de cinza); (c) binária (apenas duas cores: preto e branco)

Figura 1 – Diferentes tonalidades de cor para uma mesma imagem.



Fonte: O autor (2025)

## 2.3 CONVERSÃO PARA ESCALA DE CINZA

A conversão para escala de cinza é uma etapa comum no pré-processamento de imagens, especialmente quando a informação de cor não é essencial para a análise.

Existem diversas maneiras de realizar essa conversão, uma delas é dada por:

$$I = 0,2990R + 0,5870G + 0,1140B \quad (2.1)$$

Essa transformação é aplicada pelo MATLAB<sup>®</sup> por meio da função `rgb2gray` (The MathWorks, 2025).

Por exemplo, uma imagem colorida que possui um determinado pixel com cores RGB (120, 170, 240), ao ser convertido para escala de cinza ficaria com o valor do pixel igual a:

$$I = 0,2990 \cdot 120 + 0,5870 \cdot 170 + 0,1140 \cdot 80 \approx 145$$

A figura 2 mostra: (a) a imagem colorida e (b) a imagem transformada em escala de cinza.

Figura 2 – Conversão de um pixel RGB para escala de cinza.



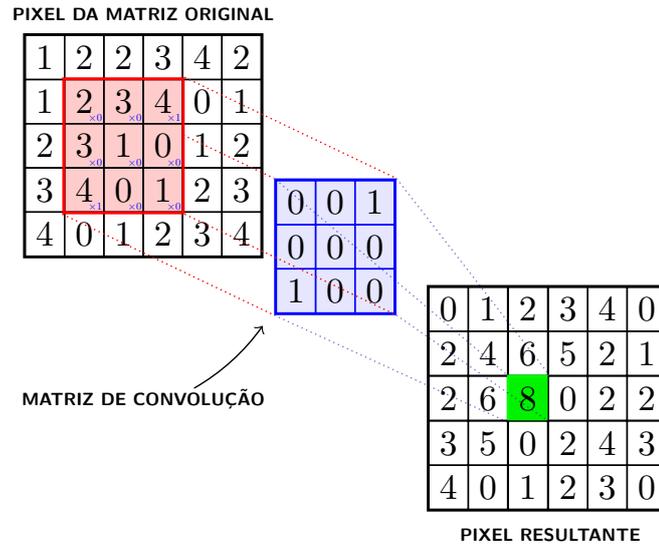
Fonte: O autor (2025)

## 2.4 FILTRAGEM E REALCE

As técnicas de filtragem de imagens digitais fazem parte do processamento de imagens e buscam obter transformações com o objetivo de fazer alterações necessárias para determinado propósito.

Essas técnicas são geralmente utilizadas em imagens em níveis de cinza. O processo de filtragem no domínio espacial atua sobre cada pixel e é similar a uma operação matemática chamada convolução (Backes; Sá Junior, 2016). A figura 3 mostra um exemplo de aplicação de convolução de duas matrizes.

Figura 3 – Processo de filtragem por convolução de matrizes



Fonte: O autor (2025)

A convolução é uma operação fundamental na abordagem linear para análise de imagens. Ela é representada por uma integral que mede o grau de sobreposição entre duas funções,  $f(t)$  e  $h(t)$  à medida que uma delas é deslocada sobre a outra (Sonka, 2015).

A convolução unidimensional de uma imagem  $f$  por uma máscara  $h$  é dada por:

$$h(x, y) * f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b h(s, t) f(x-s, y-t) \quad (2.2)$$

em que  $a$  e  $b$  se referem à metade da altura e da largura, respectivamente, da máscara  $h$ .

A filtragem no domínio do espaço age diretamente sobre os pixels da imagem. Por exemplo, o filtro de média é um filtro linear que utiliza uma máscara (ou janela) que percorre pixel a pixel da imagem. Para cada posição da máscara, calcula-se a média dos valores dos pixels vizinhos, gerando um novo valor para o pixel central:

$$F = \frac{1}{N} \sum_{i=1}^N x_i$$

onde  $x_i$  representa os valores dos pixels sob a máscara e  $N$  é o número total de pixels, considerados na vizinhança.

É possível ver na figura 4 a diferença entre: (a) uma imagem com ruído e (b) o resultado da aplicação do filtro de média para eliminação desse ruído.

Figura 4 – Imagem com presença de ruído e resultado da filtragem por filtro de média.



Fonte: (Edge AI and Vision Alliance, 2025)

## 2.5 SEGMENTAÇÃO E EXTRAÇÃO DE REGIÕES DE INTERESSE

A segmentação visa isolar objetos ou padrões de interesse dentro da imagem. Pode ser baseada em limiarização, contornos, regiões conectadas ou morfologia matemática. Essa etapa é essencial em aplicações como reconhecimento de padrões, diagnóstico médico e controle de qualidade (Sonka, 2015).

### 2.5.1 Segmentação por limiarização

Uma imagem em níveis de cinza possui tonalidades que vão de 0 a 255, onde 0 representa a cor preta e 255 a cor branca. Imagens binárias têm uma interpretação visual simples: um pixel com valor 0 aparece preto, enquanto um pixel com valor 1 aparece branco (Jingar; Panchal; Oza, 2024). Um exemplo de imagem binária já foi mostrado na figura 1c.

Para binarizar uma imagem é preciso determinar um limiar  $l$  e a partir do valor desse limiar, todos os pixels maiores que  $l$  assumirão o valor 1, enquanto o restante dos pixels se tornarão 0. Matematicamente essa regra pode ser escrita como:

$$f(x, y) = \begin{cases} 1, & f(x, y) > l \\ 0, & f(x, y) \leq l \end{cases}$$

sendo  $f(x, y)$  a função que representa a imagem e  $l$  é o valor do limiar estabelecido.

Para que a binarização seja eficiente, é necessário escolher um valor de  $l$  que separe bem o objeto do restante da imagem. Caso contrário, ou a imagem fica muito escura (excesso de pixels pretos) ou fica muito clara (excesso de pixels brancos).

Um método eficaz de encontrar esse limiar é conhecido como Método de Otsu, que é uma técnica bastante usada para separar uma imagem em duas partes diferentes, com base nos tons de cinza dos pixels. A ideia é encontrar um valor de corte, chamado

de  $l$ , que divida a imagem em dois grupos de pixels da forma mais eficiente possível. Esse valor é escolhido de modo que os pixels dentro de cada grupo sejam parecidos entre si (ou seja, com pouca variação), enquanto os dois grupos sejam bem diferentes um do outro (com bastante contraste entre eles). De forma simples, o primeiro grupo inclui os pixels com intensidades que vão de 0 até  $l$ , e o segundo grupo inclui os que vão de  $l+1$  até o valor máximo possível de intensidade, representado por  $G$  (Backes; Sá Junior, 2016).

Os grupos, as médias e a variância são calculados por:

$$a_1 = \sum_{i=0}^l P(i) \quad (2.3a)$$

$$a_2 = \sum_{i=l+1}^G P(i) \quad (2.3b)$$

$$\mu_1 = \sum_{i=0}^l iP(i)/a_1 \quad (2.3c)$$

$$\mu_2 = \sum_{i=l+1}^G iP(i)/a_2 \quad (2.3d)$$

$$\mu_t = a_1\mu_1 + a_2\mu_2 \quad (2.3e)$$

$$\sigma^2 = a_1(\mu_1 - \mu_t)^2 + a_2(\mu_2 - \mu_t)^2 \quad (2.3f)$$

As variáveis e símbolos presentes nas equações (2.3) são definidos como segue:

- $P(i)$  — probabilidade de ocorrência do nível de cinza  $i$ , obtida a partir do histograma da imagem, de modo que  $\sum_{i=0}^G P(i) = 1$ ;
- $l$  — limiar (*threshold*) que separa os dois grupos de pixels;
- $G$  — nível máximo de cinza da imagem (255 para imagens de 8 bits);
- $a_1$  — probabilidade cumulativa dos pixels pertencentes ao primeiro grupo (níveis 0 até  $l$ );
- $a_2$  — probabilidade cumulativa dos pixels pertencentes ao segundo grupo (níveis  $l+1$  até  $G$ );
- $\mu_1$  — média dos níveis de cinza do primeiro grupo;
- $\mu_2$  — média dos níveis de cinza do segundo grupo;
- $\mu_t$  — média global dos níveis de cinza da imagem;

- $\sigma^2$  — variância entre classes, que o método de Otsu busca maximizar para determinar o limiar ideal.

Como exemplo de aplicação do método, considere a matriz  $M$  dada por:

$$M = \begin{bmatrix} 45 & 33 & 145 & 22 & 152 & 41 & 18 & 153 & 27 & 190 \\ 150 & 28 & 39 & 31 & 178 & 44 & 47 & 200 & 194 & 36 \\ 29 & 23 & 41 & 136 & 181 & 48 & 160 & 34 & 142 & 27 \\ 19 & 37 & 49 & 201 & 189 & 43 & 199 & 32 & 144 & 25 \\ 46 & 20 & 35 & 42 & 31 & 172 & 193 & 155 & 11 & 39 \\ 147 & 30 & 14 & 48 & 17 & 50 & 21 & 188 & 26 & 138 \\ 24 & 40 & 19 & 133 & 176 & 23 & 182 & 46 & 170 & 28 \\ 12 & 139 & 36 & 191 & 45 & 200 & 33 & 196 & 174 & 163 \\ 185 & 29 & 37 & 201 & 156 & 16 & 47 & 178 & 15 & 137 \\ 142 & 21 & 22 & 32 & 43 & 40 & 153 & 13 & 38 & 149 \end{bmatrix}.$$

Essa matriz gera a imagem mostrada na figura 5:

Figura 5 – Imagem em escala de cinza formada pela matriz  $M$



Fonte: O autor (2025)

Por construção, os pixels “escuras” estão no intervalo  $[0,50]$  e os “claros” em  $[130,255]$ . Não há pixels com níveis entre 51 e 129.

Dado um limiar  $l$ , calculam-se, usando as equações 2.3:

$$a_1 = \sum_{i=0}^l P(i) = \frac{60}{100} = 0.6$$

$$a_2 = \sum_{i=l+1}^G P(i) = \frac{40}{100} = 0.4$$

$$\mu_1 = \frac{\sum_{i=0}^l i P(i)}{a_1} = \frac{1907}{60} \approx 31.7833$$

$$\mu_2 = \frac{\sum_{i=l+1}^G i P(i)}{a_2} = \frac{6732}{40} = 168.3000$$

$$\mu_t = a_1 \mu_1 + a_2 \mu_2 = 19,06998 + 67,32 = 86.39$$

$$\begin{aligned} \sigma^2 &= a_1 (\mu_1 - \mu_t)^2 + a_2 (\mu_2 - \mu_t)^2 \\ &= 0,6(31,7833 - 86,39)^2 + 0,4(168,3 - 86,39)^2 \\ &= 4472,8321 \end{aligned}$$

Como não existem pixels no intervalo  $[51, 129]$ , qualquer limiar

$$l \in \{50, 51, \dots, 129, 130, 132\}$$

separa exatamente os mesmos conjuntos de pixels. Por consequência, os valores calculados para a variância nesse intervalo serão os maiores possíveis e idênticos entre si, ou seja, todos terão o valor dado por  $\sigma^2 = 4472,8321$ .

Calculando explicitamente:

$$\sigma^2(49) \approx 4339.6713, \quad \sigma^2(50) \approx 4472.8321, \quad \sigma^2(132) \approx 4472.8321, \quad \sigma^2(133) \approx 4384.8438.$$

Assim, o valor máximo é

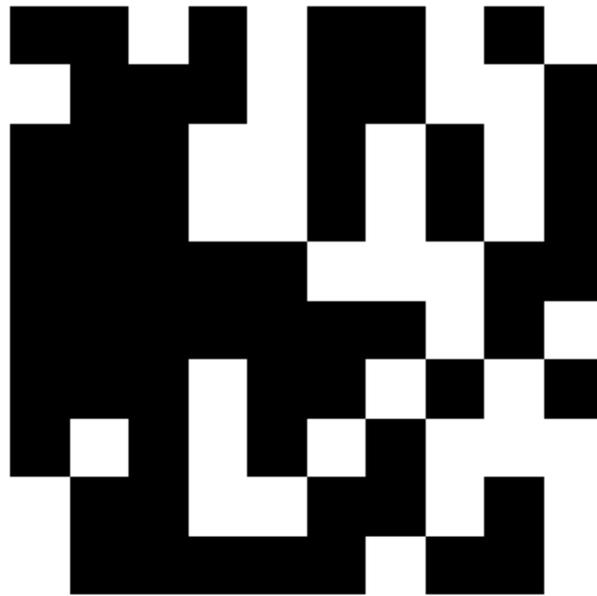
$$\sigma^2(l) = 4472.8321 \text{ para todo } l \in [50, 132].$$

Em implementações que varrem  $l$  de 0 a 255 e escolhem o primeiro  $l$  que atinge o máximo, o limiar ótimo retornado é

$$l = 50.$$

Dessa forma, aplicando o valor  $l = 50$  como limiar de binarização na imagem da figura 5 o resultado é uma imagem binária, como mostrado na figura 6

Figura 6 – Imagem binarizada da matriz  $M$ .



Fonte: O autor (2025)

O método de Otsu é amplamente utilizado para determinar esse limiar de forma automática, maximizando a variância entre classes. Na figura 7 é possível ver como se comporta uma imagem binarizada pela escolha de diferentes valores de  $I$ : (a) imagem original em escala de cinza; (b) limiar abaixo do ideal ( $I = 80$ ); (c) limiar obtido pelo método de Otsu ( $I = 134$ ); (d) limiar acima do ideal ( $I = 200$ ).

Figura 7 – Imagem em escala de cinza e sua binarização para diferentes valores de  $I$ .



Fonte: O autor (2025)

### 2.5.2 Segmentação por região de interesse

É possível criar regiões de interesse, conhecidas como ROIs (region of interest) que podem possuir vários formatos, desde elipses, linhas e polígonos até mesmo formatos livres, baseados nas características da imagem (The MathWorks, 2025).

### 3 MATLAB® COMO FERRAMENTA PARA ANÁLISE E PROCESSAMENTO DE DADOS

O MATLAB® é o principal software da empresa The MathWorks, Inc. Como o próprio nome sugere, o MATLAB® foi originalmente criado como um laboratório de matrizes (Matrix Laboratory). Até hoje, as matrizes continuam sendo o componente central da linguagem. O software permite realizar operações matemáticas com matrizes de forma prática, em um ambiente interativo e fácil de usar. Com o tempo, o MATLAB® incorporou também funções numéricas, cálculos simbólicos e ferramentas de visualização, ampliando seu uso em diversas áreas da ciência e engenharia (Chapra; Canale, 2014).

O MATLAB® é um software poderoso que possui funções integradas capazes de realizar uma ampla variedade de tarefas, desde operações matemáticas básicas até visualizações em três dimensões. Além disso, o MATLAB® oferece um conjunto completo de estruturas de programação, permitindo que os usuários desenvolvam códigos personalizados de acordo com suas necessidades específicas (Attaway, 2019).

#### 3.1 FUNCIONALIDADES DO AMBIENTE MATLAB®

Independentemente da finalidade do trabalho o uso do MATLAB® como ferramenta de computação técnica contribui significativamente para tornar o processo mais eficiente e produtivo, seja desenvolver um algoritmo, realizar uma análise, criar gráficos, elaborar um relatório ou construir uma simulação. Isso se deve, em grande parte, ao seu ambiente flexível, que permite realizar análises avançadas, visualizar dados com clareza e desenvolver algoritmos com o apoio de uma ampla variedade de recursos. Com mais de mil funções voltadas para áreas como matemática, estatística, ciência e engenharia, o MATLAB® oferece acesso imediato a um sistema robusto de cálculo numérico de alto desempenho. Esse poder computacional é complementado por ferramentas gráficas interativas que facilitam a criação de representações visuais como gráficos, imagens, superfícies e volumes. Além disso, o MATLAB® conta com os chamados *toolboxes* — conjuntos de algoritmos desenvolvidos por especialistas, voltados para aplicações específicas. Essas "caixas de ferramentas" permitem explorar diferentes técnicas numéricas, analíticas e gráficas de forma prática, sem a necessidade de programar tudo do zero. Com os *toolboxes* mais avançados, é possível expandir ainda mais as possibilidades do MATLAB® em áreas como processamento de sinais e imagens, análise de dados, estatística, modelagem matemática e controle de sistemas. Tanto as funções dos *toolboxes* quanto as do próprio MATLAB® podem ser adaptadas e otimizadas conforme as necessidades do projeto (Hahn; Valentine, 2019).

## 3.2 TRABALHANDO COM MATRIZES E IMAGENS

Existem funções no MATLAB<sup>®</sup> que servem para trabalhar diretamente com imagens. A seguir são listadas as principais funções do software que foram utilizadas no algoritmo do apêndice A.

- $A = \text{imread}(\text{filename})$ : Lê a imagem do arquivo especificado por `filename`, inferindo o formato do arquivo a partir do seu conteúdo. Se `filename` for um arquivo com várias imagens, lê a primeira imagem do arquivo `imread` (The MathWorks, 2025).
- $A = \text{imagesc}(C)$ : Exibe os dados da matriz `C` como uma imagem que utiliza toda a gama de cores do mapa de cores. Cada elemento de `C` especifica a cor de um pixel da imagem. A imagem resultante é uma grade `m`-por-`n` de pixels, onde `m` é o número de linhas e `n` é o número de colunas em `C`. Os índices de linha e coluna dos elementos determinam os centros dos pixels correspondentes (The MathWorks, 2025).
- $I = \text{rgb2gray}(RGB)$ : Converte a imagem `truecolor` RGB para uma imagem em tons de cinza `I`. A função converte imagens RGB para tons de cinza, eliminando as informações de matiz e saturação, mantendo a luminância. Se você tiver o *Parallel Computing Toolbox* instalado, `rgb2gray` poderá realizar essa conversão em uma GPU (The MathWorks, 2025).
- $B = \text{imfilter}(A,h,options,...)$  Executa filtragem multidimensional de acordo com uma ou mais opções especificadas. Uma das opções é `replicate`: Os valores da matriz de entrada fora dos limites da matriz são considerados iguais ao valor da borda da matriz mais próximo. (The MathWorks, 2025).
- $T = \text{graythresh}(I)$ : Calcula um limiar global `T` a partir de uma imagem em tons de cinza `I`, usando o método de Otsu (Equações 2.3). O método de Otsu escolhe um limiar que minimiza a variância intraclasse dos pixels pretos e brancos limiarizados e maximiza a variância entre as classes. O limiar global `T` pode ser usado com `imbinarize` para converter uma imagem em tons de cinza em uma imagem binária (The MathWorks, 2025).
- $BW = \text{im2bw}(I,level)$ : Converte a imagem em tons de cinza `I` em uma imagem binária `BW`, substituindo todos os pixels na imagem de entrada com luminância maior que `level` valor 1 (branco) e substituindo todos os outros pixels pelo valor 0 (preto). A função atual para binarização de imagens no MATLAB<sup>®</sup> é `imbinarize`, porém, essa função foi introduzida na versão r2016a, enquanto que a versão utilizada neste estudo é a versão r2015b (The MathWorks, 2025).

- $X = \text{zeros}(n)$ : retorna uma matriz n-por-n de zeros (The MathWorks, 2025).
- $k = \text{find}(X, n)$ : retorna os primeiros  $n$  índices correspondentes aos elementos diferentes de zero em  $X$  (The MathWorks, 2025).
- $TF = \text{isempty}(A)$ : retorna valor lógico 1 (true) se  $A$  estiver vazio e valor lógico 0 (false) caso contrário (The MathWorks, 2025).
- $M = \text{median}(A)$  - retorna o valor mediano de  $A$  (The MathWorks, 2025).
- $SE = \text{strel}(\text{"disk"}, r)$ : cria um elemento estruturante em forma de disco, onde  $r$  especifica o raio (The MathWorks, 2025).
- $J = \text{imopen}(I, SE)$ : realiza a abertura morfológica na imagem em tons de cinza ou binária  $I$  usando o elemento estruturante  $SE$ . A operação de abertura morfológica é uma erosão seguida de uma dilatação, usando o mesmo elemento estruturante para ambas as operações (The MathWorks, 2025).
- $\text{total} = \text{bwarea}(BW)$  estima a área dos objetos na imagem binária  $BW$ .  $\text{total}$  é um escalar cujo valor corresponde aproximadamente ao número total de pixels na imagem, mas pode não ser exatamente o mesmo porque diferentes padrões de pixels são ponderados de forma diferente (The MathWorks, 2025).

Essas funções fazem parte da Image Processing Toolbox, uma extensão do MATLAB<sup>®</sup> dedicada ao processamento de imagens digitais.

### 3.3 VANTAGENS E LIMITAÇÕES DO MATLAB<sup>®</sup>

Entre as principais vantagens do MATLAB<sup>®</sup> estão:

- Facilidade de uso e curva de aprendizagem suave;
- Poderosa visualização gráfica;
- Documentação extensa e comunidade ativa;
- Ferramentas específicas para diversas áreas.

Por outro lado, suas limitações incluem:

- Custo elevado de licenciamento, especialmente para uso fora do meio acadêmico;
- Desempenho inferior em comparação com linguagens compiladas como C ou Fortran, especialmente para tarefas computacionalmente intensivas;
- Dependência de toolboxes pagos para funcionalidades avançadas.

Apesar dessas limitações, o MATLAB<sup>®</sup> continua sendo uma das ferramentas preferidas em ambientes de pesquisa e desenvolvimento devido à sua eficiência e confiabilidade (Chapman, 2008).

## 4 METODOLOGIA

Este estudo adotou uma abordagem metodológica de natureza aplicada, exploratória e quantitativa, com foco no desenvolvimento e aplicação de algoritmos computacionais para a análise de porosidade e distribuição do tamanho dos poros de rochas porosas a partir de imagens binarizadas. A pesquisa envolveu programação em MATLAB® e análise estatística.

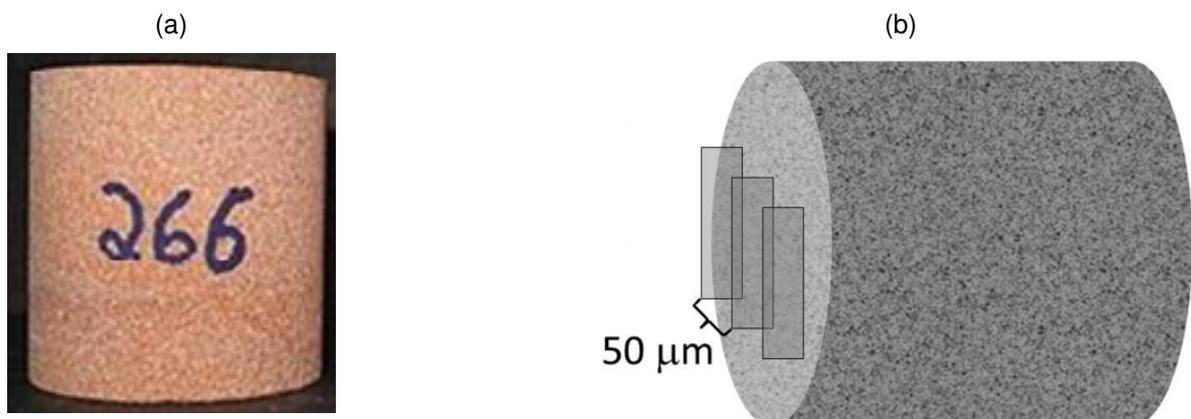
O estudo foi realizado a partir do processamento de imagens de rocha fonte de petróleo obtidas através da técnica de OCT. Inicialmente, o MATLAB® foi programado para realizar a leitura dos arquivos no computador e interpretar essas imagens como matrizes e todo o tratamento feito é baseado em álgebra matricial.

### 4.1 ANÁLISE DAS IMAGENS E CARACTERIZAÇÃO DA AMOSTRA

Para o desenvolvimento deste estudo, foram utilizadas imagens da amostra de uma rocha denominada Rocha 266, adquiridas por meio da técnica de Tomografia de Coerência Óptica (OCT). Essa técnica permite a obtenção de imagens em alta resolução de estruturas internas, sem a necessidade de destruição do material analisado.

A Figura 8 apresenta a imagem da amostra da rocha física utilizada (8a) e uma ilustração esquemática indicando a região interna de onde foram obtidas as imagens (8b). Essa área corresponde a uma seção transversal da rocha, posicionada perpendicularmente à superfície analisada, permitindo investigar a distribuição vertical dos poros ao longo da profundidade da amostra.

Figura 8 – Imagem da amostra e representação da região analisada.

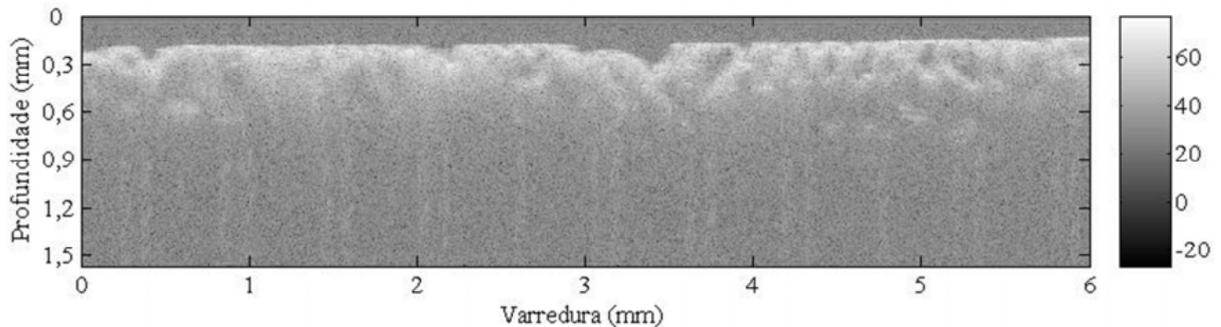


Fonte: (Campello, 2012)

Além disso, a figura 9 apresenta uma das imagens obtidas diretamente da amostra por meio do equipamento de OCT, já em escala real, onde a varredura das imagens é de 6,0 mm e a profundidade máxima atingida é de 1,5 mm. Essa imagem

serve como exemplo representativo das 160 imagens processadas ao longo deste trabalho.

Figura 9 – Imagem em escala da rocha 266.



Fonte: (Campello, 2012)

#### 4.2 NOME E CAMINHO DAS IMAGENS

Para trabalhar com as imagens, inicialmente foi necessário indicar ao MATLAB® em que diretório se encontram as imagens no computador. Isso foi feito definindo a variável `Diretorio`, que retorna os arquivos de imagem contidos no computador pelo caminho `C:\Users\felipe\Documents\MATLAB\MATLAB\Rocha266\`. Cada imagem está nomeada de forma sequencial, com a primeira sendo `rocha266_0001_BScan.bmp`, até a última, que é `rocha266_0160_BScan.bmp`. Dessa forma, para facilitar a leitura das imagens durante a análise do processamento, foi definido um prefixo e um sufixo com o retorno `rocha266_` e `_BScan.bmp`, respectivamente. Essas variáveis percorrerão o diretório em busca dos arquivos de imagem, cuja totalidade é definida por `TotalImagens`.

#### 4.3 ACUMULADORES E MATRIZ DE ARMAZENAMENTO

Ao final do processamento das imagens, é feito um cálculo de porosidade total da rocha analisada, assim como um gráfico representando sua granulometria.

Para a porosidade, o comando `TotalPixels` representa a quantidade total de pixels restantes na imagem já processada e `TotalPoros` indica quantos desses pixels serão considerados poros.

Em relação à granulometria, uma matriz de dimensões  $m \times n$  é criada, onde  $m$  representa o total de imagens selecionadas para análise e  $n$  representa o tamanho do Raio do elemento estruturante. Essa matriz é definida pela função `zeros` e todos os seus elementos serão compostos inicialmente por valores iguais a zero. Posteriormente, ao aplicar os procedimentos necessários, os valores obtidos são inseridos nessa matriz, substituindo os elementos de valor zero pelos dados encontrados.

#### 4.4 MATRIZ DE FILTRO PONDERADA

A aplicação de filtragem para eliminação de ruído presente na imagem se deu pela escolha de um filtro de média ponderada  $P$ , que é uma matriz de convolução definida por “um filtro ponderado ( $P$ ) onde o peso do elemento central seria nulo para que o elemento a ser substituído na matriz não tivesse influência no valor final. Os primeiros vizinhos teriam peso igual a 1 e os segundos vizinhos teriam peso igual  $1/\sqrt{2}$ ” (Campello, 2012, p. 53) :

#### 4.5 LEITURA E PROCESSAMENTO DAS IMAGENS

Nas etapas seguintes é feito um loop para englobar os dados de todas as imagens lidas pelo MATLAB<sup>®</sup>, de maneira que as seções 4.5.1 - 4.5.7 estão contidas nessa estrutura de repetição.

##### 4.5.1 Leitura das imagens

Após a indicação das imagens, é preciso fazer a leitura desses arquivos. A função `sprintf` retorna uma cadeia de texto como resultado, nesse caso, o nome das imagens, e reúne-os na variável `NomeImagem`, que servirá, junto com a variável `Diretorio`, como um caminho completo para ser aplicado à função `fullfile`, definindo a variável `CaminhoImagem`.

Portanto, ao aplicar a função `imread` ao `CaminhoImagem` com as especificações acima descritas, o MATLAB<sup>®</sup> já será capaz de reconhecer todas as imagens do diretório como matrizes e estará pronto para aplicar as funções e comandos necessários para o processamento.

##### 4.5.2 Conversão em Escala de cinza

As imagens obtidas por OCT para análise de processamento de imagens já se encontram em escala de cinza e podem ser trabalhadas desta maneira. Porém, para garantir que o programa funcione adequadamente para todas as imagens no caso de haver algum arquivo de imagem em, por exemplo, canais de cor RGB, torna-se necessário fazer a verificação dos arquivos e conversão.

A conversão para escala de cinza é feita iniciando uma condicional `if` que verifica se a imagem está ou não em escala de cinza. Se já estiver, o arquivo permanece inalterado. Caso esteja em formato RGB, a função `rgb2gray` converte essa imagem para o canal de cor desejado.

### 4.5.3 Filtragem e binarização

Computacionalmente, a precisão da filtragem foi obtida calculando-se o limiar de binarização e comparando este valor com uma segunda aplicação do filtro na imagem filtrada. O comando `imfilter` aplica o processo de convolução da matriz da imagem original pela matriz `P` definida na seção 4.4.

Uma vez obtida a filtragem, o passo seguinte é determinar o valor do limiar ótimo para binarização. A função `graythresh` aplica nas imagens os conceitos vistos na seção 2.5.1. Cada imagem recebe um valor, que é, então, usado na função `im2bw` para aplicar a binarização em todas elas.

### 4.5.4 Medianas dos primeiros pixels brancos

Feita a binarização, foi preciso diferenciar na imagem a região da rocha do restante da imagem: a parte superior da imagem não contém o material, enquanto a parte inferior contém o material, mas não há informação precisa sobre o sinal de imagem. Essa diferenciação é realizada por meio de uma seleção de pixels brancos e pretos dentro de um intervalo específico onde está concentrada a maior quantidade de pixels brancos, que representam a rocha.

A partir da imagem binária, onde os pixels brancos representam o material da rocha, obteve-se o valor da mediana das linhas que iniciam os primeiros brancos, coluna por coluna, nos dois sentidos verticais das imagens. Dessa forma, cada imagem recebe um intervalo de corte, onde os pixels pretos podem ser interpretados, quase que em sua totalidade, como sendo parte dos poros.

Afim de determinar essa região de interesse onde prevalecem as informações sobre a rocha e seus poros, um segundo loop é iniciado. Uma varredura com a função `find` é feita em busca dos primeiros pixels brancos nas imagens binárias, tanto da região superior quanto da inferior e a verificação é realizada pela função `isempty`, retornando duas matrizes: `PrimeirosBrancos` e `UltimasLinhas`. A mediana é aplicada por `median` nessas duas matrizes e assim define o intervalo de recorte nas imagens.

### 4.5.5 Máscara de poros

É feita uma varredura coluna por coluna na imagem binária para construir uma matriz de máscara `MascaraPoros` associada às regiões porosas. Para cada coluna, determina-se inicialmente a posição do primeiro pixel branco. A partir dessa linha, o código verifica linha por linha até alcançar o índice `MedianaLinhas2`, em busca do primeiro pixel preto. Quando encontrado, todas as linhas desta coluna, desde essa posição até `MedianaLinhas2`, são marcadas como `true` em `MascaraPoros`, representando assim a região porosa a ser considerada para análises posteriores.

#### 4.5.6 Porosidade acumulada

O algoritmo realiza a quantificação de porosidade considerando a região definida pela máscara construída anteriormente. Primeiramente, a matriz `PorosConsiderados` é obtida pela combinação lógica entre as áreas não preenchidas (`~Binarizacao`) e a `MascaraPoros`, isolando todas as regiões porosas detectadas. Em seguida, extrai-se uma sub-matriz (`PorosidadeConsiderada`), delimitada verticalmente entre `MedianaLinhas1` e `MedianaLinhas2` e horizontalmente em todas as colunas, para focar a análise numa área definida da imagem. Dessa região são contabilizados o total de pixels (`PixelsTotais`) e o número de pixels brancos (`PixelsBrancos`), correspondentes à porosidade. Por fim, esses valores são acumulados nas variáveis `TotalPixels` e `TotalPoros`, permitindo obter uma noção geral da porosidade ao longo das iterações das imagens processadas.

#### 4.5.7 Granulometria

É realizada uma análise de granulometria morfológica por aberturas sucessivas com elementos estruturantes circulares. Para cada raio especificado no vetor `Raio`, é gerado um elemento estruturante do tipo disco pela função `strel`, utilizado para realizar uma operação de abertura morfológica (`imopen`) sobre a matriz de poros (`PorosConsiderados`). Em seguida, determina-se a área total dos poros restantes após a operação por meio de `bwarea`, e o resultado é armazenado no vetor `Granulo` para cada raio avaliado. Por fim, todas as medidas obtidas para a amostra atual são armazenadas na matriz `GranulometriaTotal`, permitindo representar a distribuição de tamanhos de poros em função do raio do elemento estruturante utilizado.

### 4.6 POROSIDADE FINAL

Obtida a região de recorte, para eliminar a contagem dos pixels pretos que não fazem parte da amostra, uma varredura foi feita em busca desses pixels, coluna por coluna, aceitando como parte dos poros somente aqueles pixels que surgem imediatamente após um pixel branco. Logo, a porosidade pode ser calculada por meio da relação  $P = 100 * P_p / (P_b + P_p)$ , onde  $P_b$  e  $P_p$  representam, respectivamente, os pixels brancos e pretos das imagens. O índice de porosidade total considerada é calculado pela razão entre o número de pixels classificados como poros (`TotalPoros`) e o total de pixels analisados (`TotalPixels`), multiplicada por 100 para representar o resultado em percentual.

### 4.7 GRANULOMETRIA TOTAL

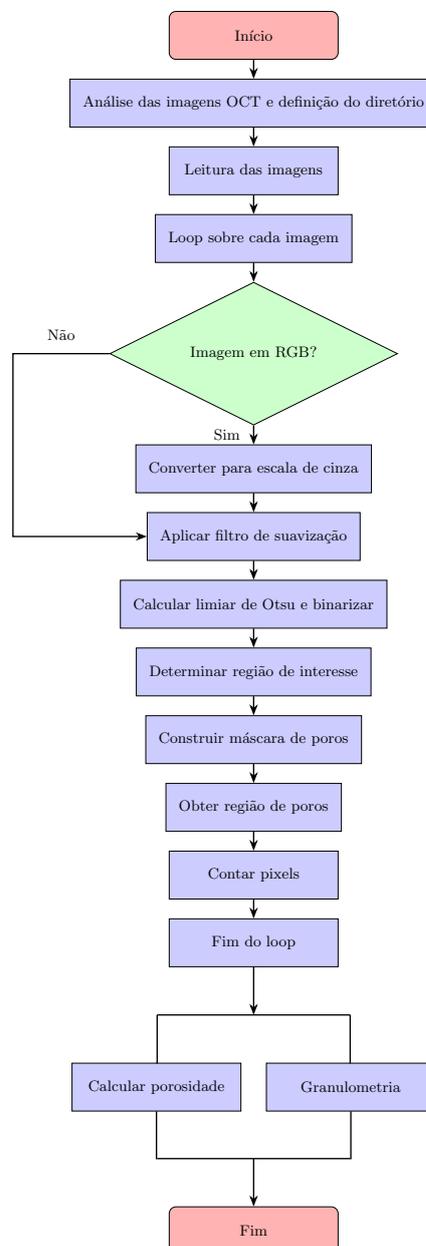
O procedimento final foi determinar a área das regiões porosas na imagem da rocha. Com isso, foi utilizado um processo de granulometria, que calcula a quantidade

de áreas de mesmo tamanho em toda a imagem para um raio que varia de 1 a 40 pixels.

Calcula-se, então, a curva de granulometria acumulada (*CurvaGranuloAcumulada*) a partir dos dados armazenados em *GranulometriaTotal*, utilizando a soma (*sum*) ao longo de todas as amostras. Dessa forma, obtém-se uma representação da distribuição de poros para todas as imagens analisadas, permitindo verificar como a área porosa varia em função do raio do elemento estruturante.

O fluxograma da figura 10 representa as etapas desenvolvidas desde o início até o fim do processo metodológico, tornando mais clara a sequência das etapas adotadas.

Figura 10 – Representação esquemática do algoritmo de processamento de imagens para análise de porosidade e distribuição dos poros.



Fonte: O autor (2025)

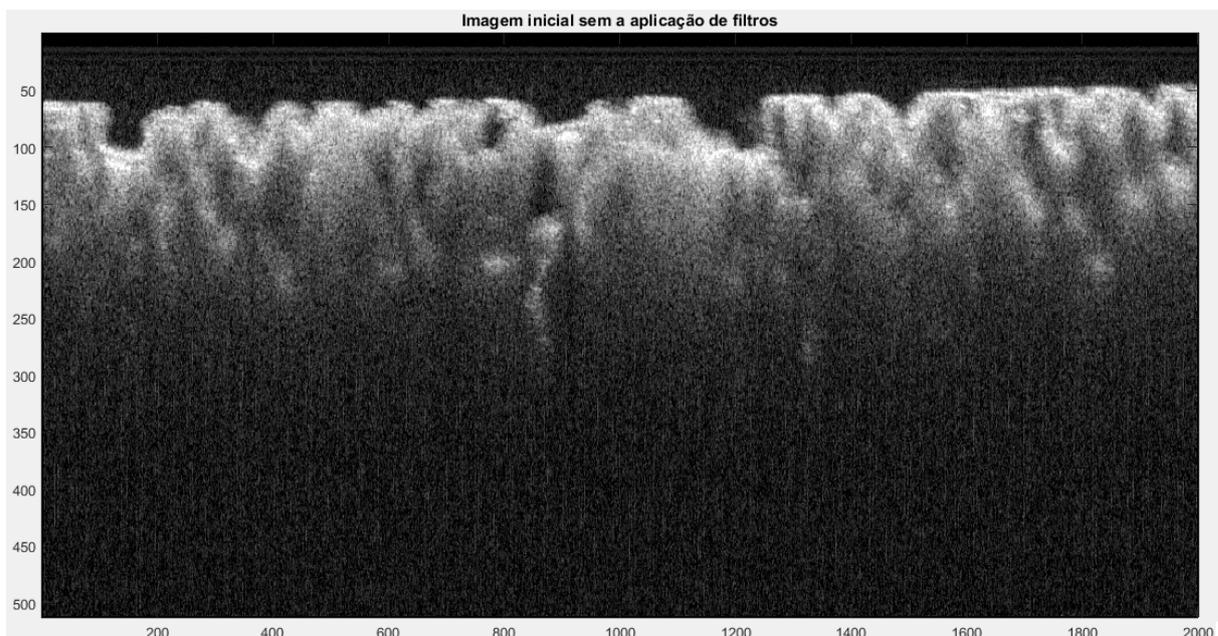
## 5 RESULTADOS E DISCUSSÃO

Foram utilizadas 160 imagens sequenciais obtidas por tomografia de coerência óptica (OCT), que representam seções transversais de uma amostra de rocha porosa. As imagens passaram por etapas sucessivas de pré-processamento, binarização, segmentação e análise morfológica, com o intuito de identificar e quantificar regiões porosas.

### 5.1 IMAGEM ORIGINAL E SUAVIZAÇÃO

A Figura 11 apresenta uma das imagens obtidas diretamente por tomografia (OCT), ou seja, a imagem sem aplicação de filtros.

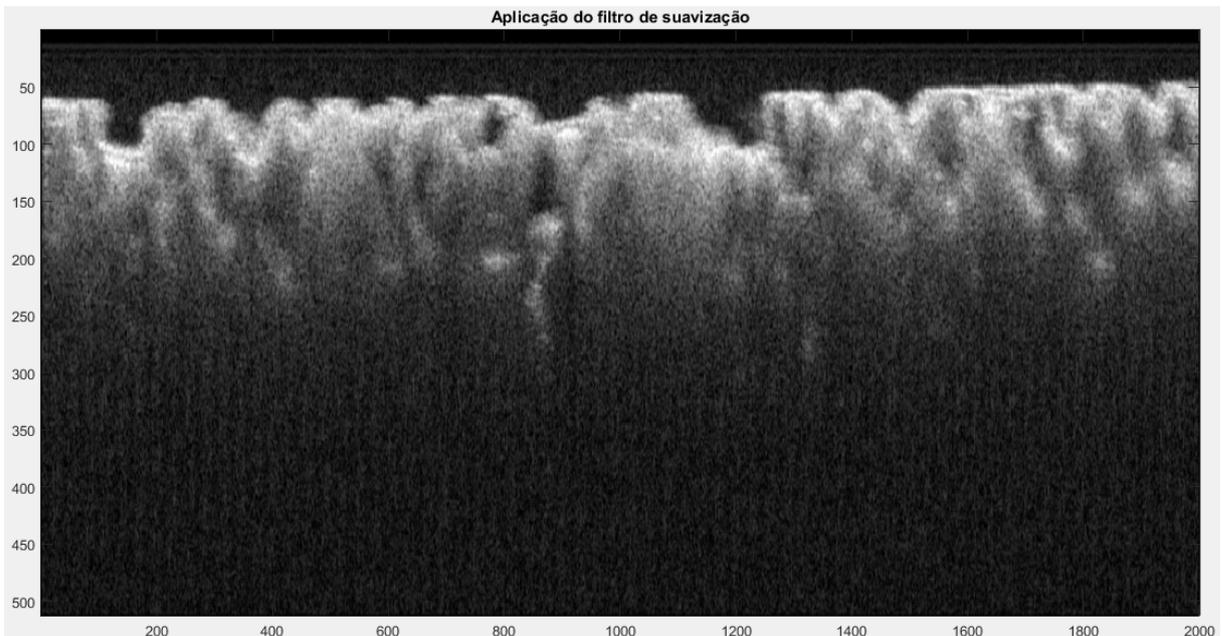
Figura 11 – Imagem original da rocha obtida por OCT.



Fonte: O autor (2025)

Já a figura 12 mostra uma das imagens originais utilizadas no processamento após a aplicação do filtro de média ponderada descrito na seção 4.4 da metodologia. É possível observar uma melhoria na definição das regiões sólidas da rocha (zonas mais claras), com significativa atenuação do ruído presente nas regiões mais profundas da imagem.

Figura 12 – Aplicação do filtro de suavização sobre imagem original da rocha.



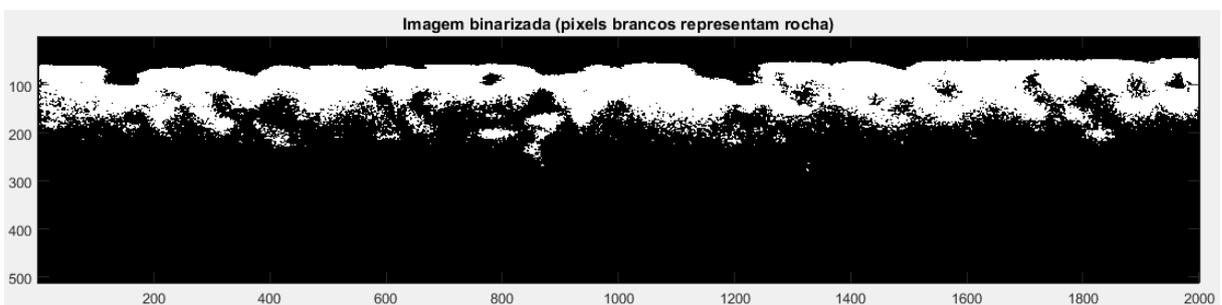
Fonte: O autor (2025)

Esse processo é fundamental para garantir maior precisão nas etapas seguintes de limiarização e segmentação.

## 5.2 IMAGEM BINARIZADA

A Figura 13 exibe o resultado da binarização realizada com limiar adaptativo, onde os pixels brancos correspondem às regiões sólidas da rocha e os pixels pretos representam espaços vazios ou porosos.

Figura 13 – Imagem binarizada com destaque para as regiões sólidas (branco).



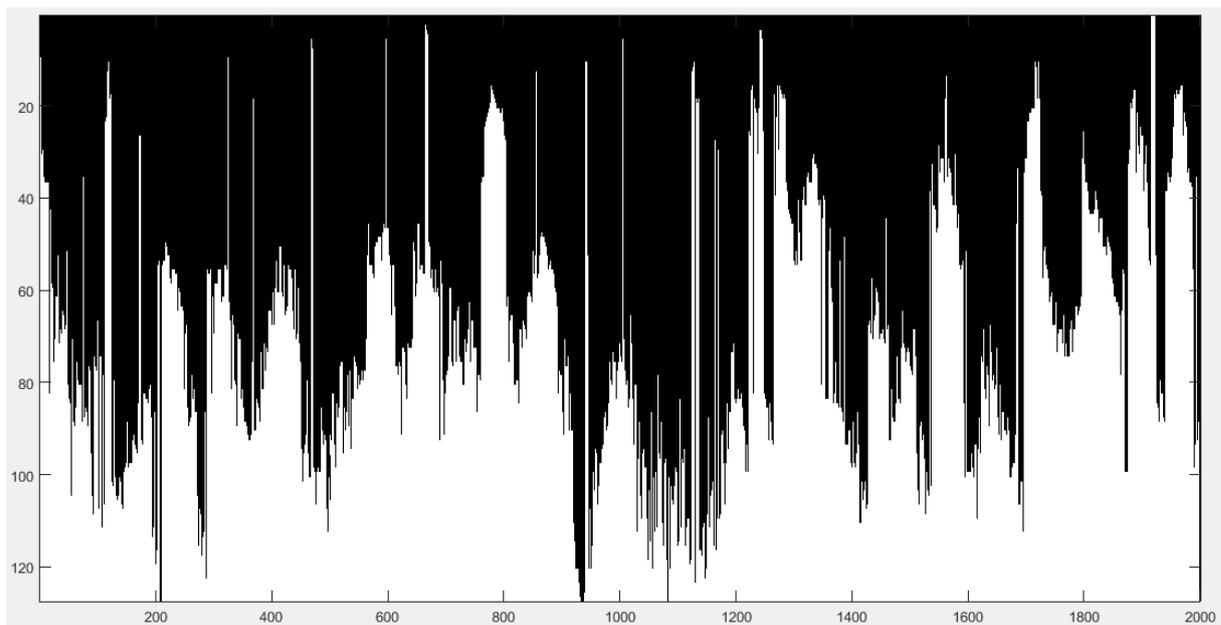
Fonte: O autor (2025)

Nota-se que a binarização preserva com boa fidelidade as estruturas principais da rocha, servindo de base para o mapeamento dos poros.

### 5.3 MÁSCARA DE POROS DETECTADOS

Com base nas medições das linhas onde predominam os primeiros pixels brancos (4.5.4), foi possível construir uma máscara lógica que isola as regiões efetivamente porosas. A Figura 14 mostra a imagem binária já tratada com essa máscara, onde os poros foram destacados ao longo das colunas a partir do limiar inferior estabelecido pelas medianas. Neste processo de identificação dos poros o esquema de cores é invertido: a cor branca representa os poros e a cor preta representa a rocha.

Figura 14 – Máscara de poros (preto: rocha; branco: poros detectados).



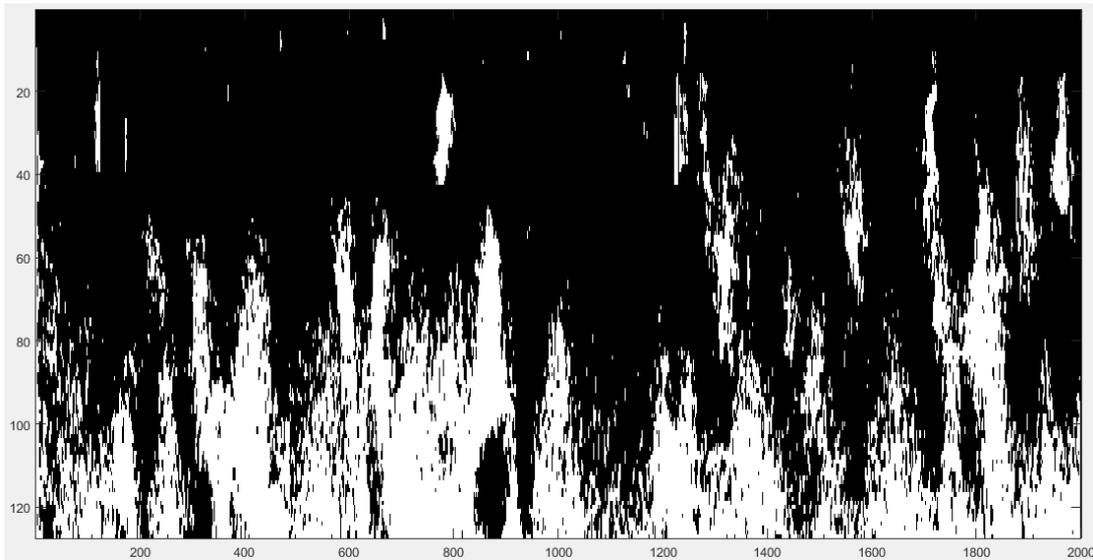
Fonte: O autor (2025)

A varredura vertical por coluna permitiu identificar com precisão os limites superiores e inferiores das zonas de interesse, descartando regiões externas não pertencentes à amostra rochosa.

### 5.4 POROSIDADE CONSIDERADA

A Figura 15 mostra a imagem final usada para o cálculo da porosidade, resultante da sobreposição da máscara com a imagem binarizada. Somente os pixels pretos dentro da área recortada (definida entre as medianas superiores e inferiores) foram considerados como poros válidos. Isso assegura que apenas regiões internas da amostra sejam contabilizadas, eliminando interferências externas. Essa matriz serviu de base para a extração dos valores totais de porosidade por imagem e para o cálculo global ao longo da série de 160 imagens.

Figura 15 – Região final considerada para o cálculo da porosidade.



Fonte: O autor (2025)

## 5.5 POROSIDADE CALCULADA

Com base na imagem final processada (Figura 15), que representa a matriz de porosidade considerada, foi possível calcular a porosidade total da amostra ao longo das 160 imagens analisadas. O cálculo foi realizado a partir da razão entre a quantidade de pixels identificados como poros e o número total de pixels analisados dentro da região de interesse. A fórmula utilizada foi:

$$\text{Porosidade (\%)} = \frac{\text{Pixels brancos da figura 15}}{\text{Total de pixels da figura 15}} \cdot 100 \quad (5.1)$$

Os valores de *TotalPoros* e *TotalPixels* foram acumulados iterativamente a cada imagem, durante a execução do algoritmo. A região considerada em cada imagem foi previamente delimitada pelas medianas verticais, garantindo que apenas áreas pertencentes à estrutura real da rocha fossem contabilizadas. Dessa forma, o valor obtido para a porosidade total da rocha utilizando a equação 5.1 foi de 25,33%.

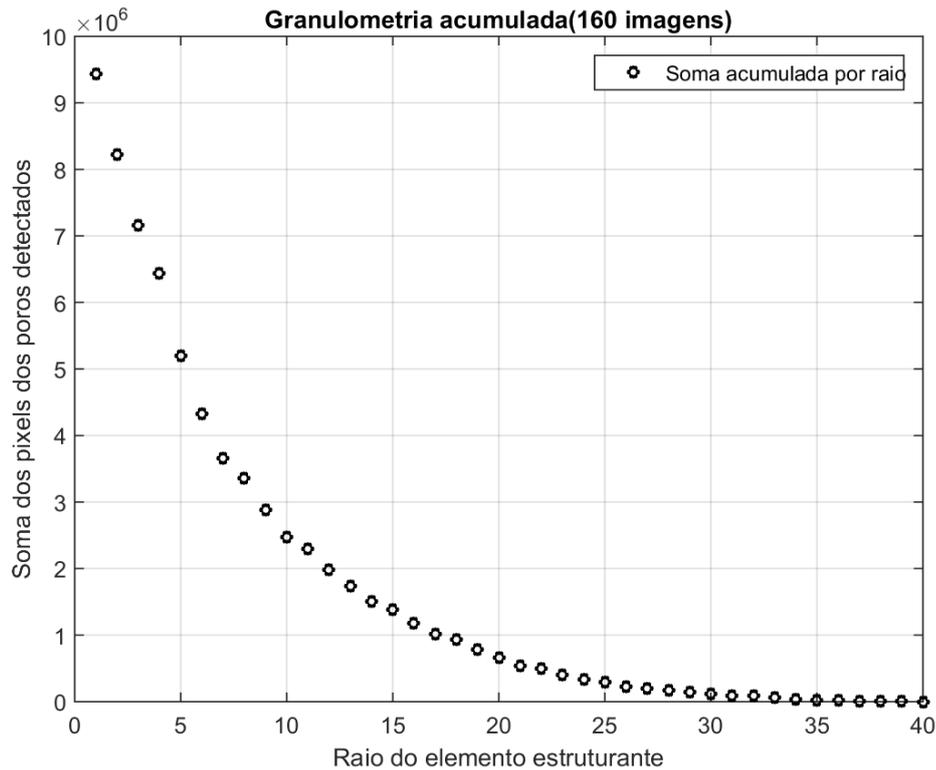
Esse valor pode ser comparado ao encontrado em (Campello, 2012), na qual a mesma amostra foi analisada utilizando técnicas complementares e de maior refinamento computacional. Nesse estudo, a porosidade encontrada foi de 17,45%. A diferença entre os valores obtidos pode ser atribuída a diversos fatores, como:

- o uso de filtros e máscaras diferentes para segmentação das regiões porosas;
- recuperação de sinal da amostra;
- parâmetros de limiarização e na definição da região útil de interesse.

## 5.6 GRANULOMETRIA MORFOLÓGICA ACUMULADA

A Figura 16 apresenta a curva de granulometria acumulada obtida após a aplicação sucessiva da operação de abertura morfológica para elementos estruturantes circulares com raio variando de 1 a 40 pixels.

Figura 16 – Granulometria acumulada dos poros ao longo de 160 imagens.



Fonte: O autor (2025)

Observa-se que a maior parte dos poros possui tamanho reduzido, com áreas totais decrescendo à medida que o raio do elemento estruturante aumenta. Isso indica que a estrutura da rocha apresenta uma distribuição predominantemente de microporos, o que pode influenciar diretamente sua permeabilidade e capacidade de armazenamento de fluidos.

Para complementar a visualização gráfica, a Tabela 5.1 apresenta os valores numéricos da área total dos poros detectados para cada raio utilizado. Os dados reforçam a tendência decrescente da área acumulada à medida que o raio aumenta, evidenciando a predominância de microporos na amostra analisada.

Tabela 5.1 – Soma dos pixels dos poros detectados para diferentes raios.

<b>Raio</b>	<b>Soma</b>	<b>Raio</b>	<b>Soma</b>	<b>Raio</b>	<b>Soma</b>	<b>Raio</b>	<b>Soma</b>
1	9433338	11	2295329	21	538841	31	91479
2	8216739	12	1980301	22	494709	32	83611
3	7154405	13	1730998	23	405530	33	56747
4	6441517	14	1507085	24	329962	34	39701
5	5202852	15	1381263	25	290115	35	26568
6	4320298	16	1185038	26	224134	36	26269
7	3658837	17	1018910	27	192294	37	12548
8	3355026	18	939563	28	164701	38	10510
9	2880288	19	787190	29	149214	39	5099
10	2469260	20	662145	30	109508	40	0

Fonte: O autor (2025)

## 6 CONCLUSÕES E PERSPECTIVAS

O resultado obtido se aproxima do valor real da porosidade da amostra dado o grau de simplificação metodológica aplicado. O fato do valor de 25,33% estar dentro da ordem de grandeza esperada reforça a validade da abordagem e demonstra que o algoritmo desenvolvido é eficaz em quantificar a porosidade de forma automatizada e com bom grau de confiabilidade.

Como continuidade deste estudo, pretende-se explorar a aplicação de outros filtros e técnicas de processamento de imagem que possam refinar ainda mais a segmentação dos poros, com o objetivo de reduzir a diferença em relação ao valor real da porosidade e aumentar a precisão da análise.

## REFERÊNCIAS

ATTAWAY, S. **MATLAB: A Practical Introduction to Programming and Problem Solving**. 5 ed.: Butterworth-Heinemann, 2019.

BACKES, A. R.; SÁ JUNIOR, J. J. d. M. **Introdução à visão computacional usando Matlab**. 1 ed. Brasil: ALTA BOOKS, 2016.

BOPPART, S. A. Optical coherence tomography: Technology and applications for neuroimaging. **EBSCOhost Academic Search Premier**, v. 40, n. 4, p. 529–541, 2003.

BURGER, W.; BURGE, M. J. **Digital Image Processing: An Algorithmic Introduction Using Java**. 2 ed.: Springer Publishing Company, Incorporated, 2016.

CAMPELLO, S. L. **Um método para estudo de meios porosos utilizando OCT**. 2012. Universidade Federal de Pernambuco.

CAMPELLO, S. L. et al. Micro-structural information of porous materials by optical coherence tomography. **Microporous and Mesoporous Materials**, v. 198, n. 4, p. 50–54, 2014.

CHAPMAN, S. J. **MATLAB Programming for Engineers**. 4 ed. Toronto, Canada: Thomson Learning, 2008.

CHAPRA, S. C.; CANALE, R. P. **Numerical Methods for Engineers**. 7 ed.: McGraw Hill, 2014.

EDGE AI AND VISION ALLIANCE. **Understanding Spatial Noise and Its Reduction Methods Using Convolution**. [S. l.: s. n.], 2025. Accessed on: 20 July 2025.

Disponível em:

Available%20at:%20<https://www.edge-ai-vision.com/2024/06/understanding-spatial-noise-and-its-reduction-methods-using-convolution/>.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3 ed. USA: Prentice-Hall, Inc., 2006.

HAHN, B. D.; VALENTINE, D. T. **Essential MATLAB for Engineers and Scientists**. 7 ed.: Academic Press, 2019.

JINGAR, M. P.; PANCHAL, M. K.; OZA, M. P. A Review on Image Processing. **International Journal of Intelligent Systems and Applications in Engineering**, p. 515–520, 2024.

KAESTNER, A.; LEHMANN, E.; STAMPANONI, M. Imaging and image processing in porous media research. **Advances in Water Resources**, p. 1174–1187, 2008.

NIMMO, J. R. Porosity and Pore Size Distributions. **Encyclopedia of soils in the environment**, p. 295–303, 2004.

SONKA, M. e. **Image Processing, Analysis, and Machine Vision**. 4 ed.: Cengage, 2015.

THE MATHWORKS, I. **bwarea - Area of objects in binary image**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/images/ref/bwarea.html](https://www.mathworks.com/help/images/ref/bwarea.html).

THE MATHWORKS, I. **Create ROI Shapes**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/images/roi-creation-overview.html](https://www.mathworks.com/help/images/roi-creation-overview.html).

THE MATHWORKS, I. **find - Find indices and values of nonzero elements**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/matlab/ref/find.html](https://www.mathworks.com/help/matlab/ref/find.html).

THE MATHWORKS, I. **graythresh — Global image threshold using Otsu’s method**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/images/ref/graythresh.html?searchHighlight=graythresh&s\\_tid=srchtitle\\_support\\_results\\_1\\_graythresh](https://www.mathworks.com/help/images/ref/graythresh.html?searchHighlight=graythresh&s_tid=srchtitle_support_results_1_graythresh).

THE MATHWORKS, I. **im2bw — (Not recommended) Convert image to binary image, based on threshold**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/images/ref/im2bw.html](https://www.mathworks.com/help/images/ref/im2bw.html).

THE MATHWORKS, I. **imagesc - Display image with scaled colors**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/matlab/ref/imagesc.html](https://www.mathworks.com/help/matlab/ref/imagesc.html).

THE MATHWORKS, I. **imfilter — N-D filtering of multidimensional images**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [https://www.mathworks.com/help/images/ref/imfilter.html?searchHighlight=imfilter&s\\_tid=srchtitle\\_support\\_results\\_1\\_imfilter](https://www.mathworks.com/help/images/ref/imfilter.html?searchHighlight=imfilter&s_tid=srchtitle_support_results_1_imfilter).

THE MATHWORKS, I. **imopen - Morphologically open image**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/images/ref/imopen.html?searchHighlight=imopen+&s\\_tid=srchtitle\\_support\\_results\\_1\\_imopen+](https://www.mathworks.com/help/images/ref/imopen.html?searchHighlight=imopen+&s_tid=srchtitle_support_results_1_imopen+).

THE MATHWORKS, I. **imread — Read image from graphics file**. [S. l.: s. n.], 2025. Accessed on: 19 July 2025. Disponível em: [Available%20at:%20https://www.mathworks.com/help/matlab/ref/imread.html](https://www.mathworks.com/help/matlab/ref/imread.html).

THE MATHWORKS, I. **isempty - Determine whether array is empty.** [*S. l.: s. n.*], 2025. Accessed on: 19 July 2025. Disponível em: Available%20at:%20https://www.mathworks.com/help/matlab/ref/double.isempty.html.

THE MATHWORKS, I. **median - Median value of array.** [*S. l.: s. n.*], 2025. Accessed on: 19 July 2025. Disponível em: Available%20at:%20https://www.mathworks.com/help/matlab/ref/double.median.html.

THE MATHWORKS, I. **rgb2gray.** [*S. l.: s. n.*], 2025. Accessed on: 19 July 2025. Disponível em: Available%20at:%20https://www.mathworks.com/help/matlab/ref/rgb2gray.html?searchHighlight=rgb2gray&s\_tid=srchtitle\_support\_results\_1\_rgb2gray#buiz8mj-1-I.

THE MATHWORKS, I. **strel - Morphological structuring element.** [*S. l.: s. n.*], 2025. Accessed on: 19 July 2025. Disponível em: Available%20at:%20https://www.mathworks.com/help/images/ref/strel.html.

THE MATHWORKS, I. **zeros - Create array of all zeros.** [*S. l.: s. n.*], 2025. Accessed on: 19 July 2025. Disponível em: Available%20at:%20https://www.mathworks.com/help/matlab/ref/zeros.html.

VAFAl, K. **Handbook of Porous Media.** 2 ed. USA: CRC Press, 2005.

## APÊNDICE A – ALGORITMO UTILIZADO NO PROCESSAMENTO DAS IMAGENS

```

1
2 %----- Diretorio das imagens -----
3 Diretorio = 'C:\Users\felipe\Documents\MATLAB\MATLAB\Rocha266
   \';
4 prefixo = 'rocha266_';
5 sufixo = '_BScan.bmp';
6 TotalImagens = 160;
7
8 %----- Acumuladores de porosidade -----
9 TotalPixels = 0;
10 TotalPoros = 0;
11
12 %----- Dados para Granulometria -----
13 Raio = 1:40;
14 GranulometriaTotal = zeros(TotalImagens, length(Raio));
15
16 %----- Matriz de filtro e normalizacao -----
17 P = [0.7071 1.0000 0.7071;
18      1.0000 0.0000 1.0000;
19      0.7071 1.0000 0.7071];
20
21 FiltroNormalizado = (1 / sum(P)) * P;
22
23 %----- Processamento -----
24 for k = 1:TotalImagens
25     %----- Nome e leitura da imagem -----
26     NomeImagem = sprintf('%s%04d%s', prefixo, k, sufixo);
27     CaminhoImagem = fullfile(Diretorio, NomeImagem);
28     Imagens = imread(CaminhoImagem);
29
30     %----- Conversao em Escala de cinza -----
31     if size(Imagens, 3) == 3
32         EscalaCinza = rgb2gray(Imagens);
33     else
34         EscalaCinza = Imagens;
35     end
36

```

```

37  %----- Filtragem e binarizacao -----
38  EscalaCinzaFiltrada = imfilter(EscalaCinza,
    FiltroNormalizado, 'replicate');
39  Limiar = graythresh(EscalaCinzaFiltrada);
40  Binarizacao = im2bw(EscalaCinzaFiltrada, Limiar);
41  [nLinhas, nColunas] = size(Binarizacao);
42
43  %----- Medianas dos primeiros pixels brancos -----
44  PrimeirosBranco = zeros(1, nColunas);
45  UltimasLinhas = zeros(1, nColunas);
46
47  for col = 1:nColunas
48      idxBranco = find(Binarizacao(:, col), 1); % de cima
        para baixo
49      if ~isempty(idxBranco)
50          PrimeirosBranco(col) = idxBranco;
51      end
52      idxUltimoBranco = find(Binarizacao(end:-1:1, col) ==
        1, 1);
53      if ~isempty(idxUltimoBranco)
54          UltimasLinhas(col) = nLinhas - idxUltimoBranco +
            1;
55      end
56  end
57
58  %----- Intervalo de corte -----
59  MedianaLinhas1 = median(PrimeirosBranco(PrimeirosBranco
        > 0));
60  MedianaLinhas2 = median(UltimasLinhas(UltimasLinhas > 0))
        ;
61  LinhasdeCorte = floor(MedianaLinhas2 - MedianaLinhas1);
62
63  %----- Regiao recortada -----
64  Recorte = zeros(LinhasdeCorte, nColunas);
65  for col = 1:nColunas
66      E1 = find(Binarizacao(:, col), 1);
67      if ~isempty(E1)
68          fim = min(E1 + LinhasdeCorte - 1, nLinhas);

```

```

69         Recorte(1:(fim - E1 + 1), col) = Binarizacao(E1:
           fim, col);
70     end
71 end
72
73 %----- Mascara de poros -----
74 MascaraPoros = false(nLinhas, nColunas);
75 for col = 1:nColunas
76     idx_branco = find(Binarizacao(:, col) == 1, 1);
77     if isempty(idx_branco), continue; end
78     for linha = floor(idx_branco)+1 : floor(
           MedianaLinhas2)
79         if Binarizacao(linha, col) == 0
80             MascaraPoros(linha:MedianaLinhas2, col) =
               true;
81             break;
82         end
83     end
84 end
85
86 PorosConsiderados = ~Binarizacao & MascaraPoros;
87
88 %----- Porosidade acumulada -----
89 PorosidadeConsiderada = ~Binarizacao(MedianaLinhas1:
           MedianaLinhas2, 1:2000) & MascaraPoros(MedianaLinhas1:
           MedianaLinhas2, 1:2000);
90
91 PixelsTotais = numel(PorosidadeConsiderada);
92 PixelsBrancos = nnz(PorosidadeConsiderada);
93
94 TotalPixels = TotalPixels + PixelsTotais;
95 TotalPoros = TotalPoros + PixelsBrancos;
96
97 %----- Granulometria -----
98 Granulo = zeros(1, length(Raio));
99 for r = Raio
100     se = strel('disk', r);
101     PorosAnalisados = imopen(PorosConsiderados, se);
102     Granulo(r) = bwarea(PorosAnalisados);

```

```
103     end
104     GranulometriaTotal(k, :) = Granulo;
105
106 end
107
108 %----- Porosidade Final -----
109 PorosidadeTotal = 100 * TotalPoros / TotalPixels;
110 fprintf('Porosidade total considerada: %.2f%%\n',
        PorosidadeTotal);
111
112 %----- Granulometria total -----
113 CurvaGranuloAcumulada = sum(GranulometriaTotal, 1); % soma
        das granulometrias
114
115 fig = figure;
116 plot(Raio, CurvaGranuloAcumulada, 'k o', 'LineWidth', 2, '
        MarkerSize', 5);
117 grid on;
118 xlabel('Raio do elemento estruturante');
119 ylabel('Area total dos poros detectados');
120 title('Granulometria acumulada(160 imagens)');
121 legend({'Soma acumulada por raio'}, 'Location', 'northeast');
122 set(gca, 'FontSize', 12);
123 saveas(fig, fullfile(Diretorio, 'Granulometria_Total_Rocha.
        png'));
```

## APÊNDICE B – CERTIFICADO: XXXVIII ENCONTRO DE FÍSICA DO NORTE E NORDESTE



O Comitê Organizador certifica que o trabalho **Determination of Porosity in Rocks Using Digital Image Processing** de autoria de **Felipe Alves dos Santos e Sérgio de Lemos Campello** foi apresentado na sessão **Física Aplicada** no XXXVIII Encontro de Física do Norte e Nordeste no período de 25 a 27 de novembro de 2024, em Aracaju, SE, na Universidade Tiradentes. Promovido pela Sociedade Brasileira de Física.



**APÊNDICE C – CERTIFICADO: 32º CONGRESSO DE INICIAÇÃO CIENTÍFICA**