

**UNIVERSIDADE DE BRASÍLIA – UNB
FACULDADE DE TECNOLOGIA - FT
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL - ENC**

**CARACTERIZAÇÃO MORFOLÓGICA-GEOTÉCNICA UTILIZANDO
IMAGENS DE MICROTOMOGRAFIA COMPUTADORIZADA DE
RAIO-X**

THAIS KOGUI DE MOURA

ORIENTADOR: MICHÉLE DAL TOÉ CASAGRANDE, D.Sc

**COORIENTADOR: LUAN CARLOS DE SENA MONTEIRO OZELIM,
D.Sc**

DISSERTAÇÃO DE MESTRADO EM GEOTECNIA

PUBLICAÇÃO: G.DM-409/2024

BRASÍLIA, DF: JUNHO - 2024

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**CARACTERIZAÇÃO MORFOLÓGICA-GEOTÉCNICA UTILIZANDO IMAGENS
DE MICROTOMOGRAFIA COMPUTADORIZADA DE RAIO-X**

THAIS KOGUI DE MOURA

**DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA CIVIL E AMBIENTAL DA UNIVERSIDADE DE BRASÍLIA COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE.**

APROVADA POR:

MICHÉLE DAL TOÉ CASAGRANDE, D.Sc.
(Universidade de Brasília)
(ORIENTADORA)

LUAN CARLOS DE SENA MONTEIRO OZELIM, D.Sc.
(Universidade de Brasília)
(COORIENTADOR)

ANDRÉ LUÍS BRASIL CAVALCANTE, D.Sc.
(Universidade de Brasília)
(EXAMINADOR INTERNO)

GIOVANNA MONIQUE ALELVAN, D.Sc.
(Universidade Federal de Minas Gerais)
(EXAMINADORA EXTERNA)

DATA: BRASÍLIA/DF, 13 de JUNHO de 2024

FICHA CATALOGRÁFICA

MOURA, THAIS KOGUI DE

Caracterização Morfológica-Geotécnica Utilizando Imagens de Microtomografia Computadorizada de Raio-X. Distrito Federal, 2024. 122 p., 210x297 mm (ENC/FT/UnB, Mestre, Geotecnia, 2024).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Civil.

- | | |
|-------------------------------|----------------------------|
| 1. Microtomografia | 2. Processamento de imagem |
| 3. Caracterização Morfológica | 4. Python |
| I. ENC/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

MOURA, T. K. (2024). Caracterização Morfológica-Geotécnica Utilizando Imagens de Microtomografia Computadorizada de Raio-X. Dissertação de Mestrado, Publicação G.DM-409/2024, Departamento de Engenharia Civil, Universidade de Brasília, Brasília, DF, 122 p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Thais Kogui de Moura

TÍTULO DA DISSERTAÇÃO DE MESTRADO: Caracterização Morfológica-Geotécnica Utilizando Imagens de Microtomografia Computadorizada de Raio-X. Distrito Federal. GRAU / ANO: Mestre / 2024

É concedida à Universidade de Brasília a permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

Thais Kogui de Moura

Campus Universitário Darcy Ribeiro,
Prédio SG-12, Asa Norte, Brasília-DF
CEP 70.910-900 - Brasília/DF – Brasil

AGRADECIMENTOS

Agradeço a Deus, que é amor, cuidado e ombro amigo.

Agradeço à minha obachan e ojichan, Mieko Aoki Kogui e Saburo Kogui, pelos ensinamentos nessa vida. Meus pais, Ivete e Israel, e meus irmãos Erika e Thomas. Ao meu companheiro Thiago e ao nosso maior presente, nosso filho Davi. Aos meus amigos que tenho no coração.

À minha orientadora Michéle e ao meu coorientador Luan por todo apoio acadêmico e amigo.

Agradeço à vida e reconheço o privilégio de oportunidades.

*Do outro lado, o que mais se temia
era o desconhecido que a aurora trazia
O medo é realmente um cara intrometido
Entra sem ser chamado e chega convencido
Sussurra nos ouvidos
mil histórias com monstros e bandidos
E às vezes pega até os adultos desprevenidos*

*Ele não é mau
é só um cara preocupado
que vem lembrar: “É importante ter cuidado”
Só vira problema
se ao contrário de um poema
em vez de nos fazer voar nos prende como algema*

*E pra todo sim existe um não
Também pra todo problema tem uma solução
E ela chega pequena
Como uma bobagem
Ganha voltagem
Cresce, Cresce!
É a coragem!*

*Ela diz pro medo: Calma!
Lembra que a gente tem fibra, a gente tem alma!
Então de repente, num rompante
Grita num alto falante e se faz gigante!*

*A coragem nos convence de prima
Nos dá força e nos aproxima
E aquele monstro feroz
De garra forte, veloz
Na verdade, nem existe
O outro é como nóiz*

*E o senhor medo vai embora contente
Quando vê a coragem triunfar lindamente
E provar que tudo é uma questão de olhar
Às vezes o que nos assusta tem muito da gente
(Só é um pouco diferente)*

(E foi assim que eu e a escuridão ficamos amigas – Emicida)

CARACTERIZAÇÃO MORFOLÓGICA-GEOTÉCNICA UTILIZANDO IMAGENS DE MICROTOMOGRÁFIA COMPUTADORIZADA DE RAIOS-X

RESUMO

A engenharia geotécnica tem utilizado os avanços tecnológicos e análises numéricas para minimizar erros nas medições de propriedades do solo e obter resultados mais precisos. Métodos da ciência dos materiais têm contribuído para a análise das propriedades do solo em escala dos grãos, permitindo caracterizações morfológicas, como a microtomografia computadorizada de raios-X (μ CT). Neste sentido, esta dissertação tem como objetivo a implementação de algoritmos e desenvolvimento de scripts em Python para processar imagens bidimensionais geradas por microtomografia computadorizada de raio-X e a comparação dos resultados obtidos de caracterização morfológica-geotécnica com os resultados por métodos experimentais e por outros softwares (Mathematica e DragonFly). As amostras analisadas incluíram areia, rejeito de minério de ferro e um compósito de rejeito de minério de ferro com polímero, visando analisar parâmetros morfológicos em mesoescala como esfericidade, granulometria, porosidade e índice de vazios. A pesquisa destacou que a configuração das imagens no microtomógrafo e o pré-processamento das imagens podem influenciar significativamente os resultados. Foram utilizadas as bibliotecas Python OpenCV, NumPy, Matplotlib, PIL, Scikit-image, SciPy e Pandas para a análise de parâmetros morfológicos de amostras geotécnicas, demonstrando-se uma alternativa relativamente viável, desde que sejam consideradas as especificidades das imagens microtomografadas da amostra estudada. Os resultados das análises pelo código Python para as amostras de areia e rejeito de minério de ferro tiveram boas correlações quando comparados com resultados de trabalhos anteriores. Já para o compósito de rejeito de minério de ferro estabilizado com polímero, o resultado obtido pelo código Python não obteve correlação próxima aos valores encontrados pelo software de processamento do microtomógrafo ZEISS Xradia 510 Versa.

PALAVRAS-CHAVE: Microtomografia, Processamento de Imagem, Caracterização Morfológica, Python.

ABSTRACT

Geotechnical engineering has been leveraging technological advances and numerical analyses to minimize errors in the measurement of soil properties and to achieve more precise results. Methods from material science have contributed to the analysis of soil properties at the grain scale, enabling morphological characterizations, such as X-ray computed microtomography (μ CT). In this context, this dissertation aims to implement algorithms and develop Python scripts to process two-dimensional images generated by X-ray computed microtomography. The samples analyzed included sand, iron ore tailings, and a composite of iron ore tailings with polymer, aiming to analyze meso-scale morphological parameters such as sphericity, granulometry, porosity, and void ratio. The research highlighted that the configuration of the images in the microtomograph and the image pre-processing can significantly influence the results. Python libraries such as OpenCV, NumPy, Matplotlib, PIL, Scikit-image, SciPy, and Pandas were used for the morphological parameter analysis of geotechnical samples, demonstrating a relatively viable alternative, provided the specificities of the microtomographed images of the studied samples are considered. The results of the analyses using Python code for the geotechnical parameters through image processing for the sand sample showed that the obtained sphericity matched the value found using Mathematica software, the granulometry exhibited a distribution similar to the experimental curve, maintaining its classification as medium sand, and the void ratio and porosity found were significantly close; for the iron ore tailings sample, the sphericity had variability in values depending on the chosen methods, the granulometry showed a tendency for larger particles compared to the experimental curve, and the void ratio and porosity found by the Python code exactly matched the values obtained by DragonFly software; however, for the composite of iron ore tailings stabilized with polymer, the results obtained by the Python code did not closely correlate with the values found by DragonFly software.

KEYWORDS: Microtomography, Image Processing, Morphological Characterization, Python.

SUMÁRIO

1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO GERAL.....	1
1.2. PROBLEMA E RELEVÂNCIA DA PESQUISA.....	3
1.3. JUSTIFICATIVA	4
1.4. HIPÓTESE	5
1.5. OBJETIVOS.....	5
1.5.1. OBJETIVOS ESPECÍFICOS.....	5
1.6. ESTRUTURA DA DISSERTAÇÃO	6
2. REVISÃO BIBLIOGRÁFICA.....	7
2.1. MICROTOMOGRÁFIA COMPUTADORIZADA DE RAIOS-X.....	7
2.2. PROCESSOS COMPUTACIONAIS COM IMAGENS.....	11
2.2.1. PRÉ-PROCESSAMENTO E PROCESSAMENTO EM IMAGENS.....	13
2.3. ANÁLISE POR IMAGENS: ESFERICIDADE.....	18
2.4. ANÁLISE POR IMAGENS: GRANULOMETRIA	22
2.5. ANÁLISE POR IMAGENS: ÍNDICE DE VAZIOS E POROSIDADE.....	27
3. MATERIAIS E MÉTODOS	29
3.1. MATERIAIS	29
3.1.1. AREIA.....	29
3.1.2. REJEITO DE MINÉRIO DE FERRO	31
3.1.3. MICROTOMÓGRAFO COMPUTADORIZADO DE RAIOS-X.....	35
3.2. MÉTODOS.....	39
3.2.1. ESFERICIDADE	39
3.2.2. GRANULOMETRIA	47
3.2.3. ÍNDICE DE VAZIOS E POROSIDADE.....	54
4. RESULTADOS E DISCUSSÕES	65
4.1. ESFERICIDADE DA AREIA.....	65
4.2. ESFERICIDADE DO REJEITO DE MINÉRIO DE FERRO.....	68
4.3. GRANULOMETRIA DA AREIA	70
4.4. GRANULOMETRIA DO REJEITO DE MINÉRIO DE FERRO	73
4.5. ÍNDICE DE VAZIOS E POROSIDADE DA AREIA.....	75
4.6. ÍNDICE DE VAZIOS E POROSIDADE DO REJEITO DE MINÉRIO DE FERRO	75
4.7. ÍNDICE DE VAZIOS E POROSIDADE DO COMPÓSITO DE REJEITO DE	
MINÉRIO DE FERRO E POLÍMERO.....	77
5. CONSIDERAÇÕES FINAIS	80
6. SUGESTÕES PARA PESQUISAS FUTURAS	82
APÊNDICES.....	83

CAPÍTULO 1

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO GERAL

Buscando-se solucionar problemas e aprimorar resultados, a engenharia tem evoluído e se adequado ao ambiente globalizado da economia de mercado e da tecnologia em busca de melhoramento e otimização de processos. Nesse contexto, a engenharia geotécnica pode se deparar com situações de incerteza e variabilidade em investigações laboratoriais, tornando-se necessária a buscar pela minimização ou eliminação de erros em medições de propriedades dos solos. Esses erros seriam, conforme indicado por Vanmarcke (1977a), uma das três fontes de incertezas na caracterização de estruturas dos solos, a qual se soma à carência de conhecimento, como elencado por Lacasse e Nadim (1996); e, conforme Morgenstern (1995), à incerteza de parâmetros devido a erros humanos ou deficiências no aparelho onde se determina a propriedade do solo.

Com o advento de avanços na tecnologia e evolução em análises numéricas, formas de minimização ou eliminação das fontes de incertezas podem ser utilizadas para se obter resultados melhores e mais acurados, otimizando esforço e tempo. Tal fato é evidenciado pela utilização de equipamentos com acurácia em escala de poros para caracterização geométrica das partículas, os quais podem revelar a estrutura do solo por meio de processos computacionais.

Introduzidos na geomecânica para avaliar solos em microescala, métodos da ciência dos materiais têm contribuído para a análise de propriedades dos solos na escala de grãos permitindo caracterizações geométrica e morfológica por meio de ensaios, como a microtomografia computadorizada (μ CT), porosimetria por injeção de mercúrio (MIP), microscopia eletrônica de varredura (SEM) e microscopia ótica. O interesse da engenharia geotécnica pela obtenção dessas caracterizações tem destaque em estudos relevantes como a previsão da capacidade de infiltração dos solos, sendo determinante para soluções de engenharia como a destinação de águas pluviais em centros urbanos, redução na vazão de pico e estudos de plumas de contaminação (Durand *et al.*, 2012) e no estudo do comportamento mecânico em microescala, como esmagamento e rearranjo de partículas (Cheng *et al.* 2019).

Em relação à utilização de equipamentos de raio X, suas aplicações tiveram início na área médica para a caracterização da estrutura óssea e de órgãos, expandindo para outros ramos

da ciência devido à possibilidade de análises não destrutivas. Um desses ramos é a ciência dos solos, dentro da qual, segundo Al-Raoush e Willson (2005), figura como pioneiro Petrovic et al. (1982) e Crestana et al. (1985). Os trabalhos pioneiros trataram da utilização de imagens na análise de densidade do meio poroso de solo e esferas de vidro usando técnicas de escâner de raio X e na análise das variações no teor de água de meios porosos compostos por areia e areia fina silte-argilosa por meio de tomografia computadorizada (CT), respectivamente.

Com os avanços computacionais na capacidade de processamento de dados e imagens associados às metodologias de escala microestrutural e mesoestrutural, Al-Raoush & Willson (2005) e Zubeldia (2013) contextualizam a utilização de imagens em análises de meios porosos:

Com o aumento do poder computacional permitindo uma modelagem mais realista em escala de poros e os avanços nas técnicas de imagem com a tomografia, pesquisadores podem usufruir de ferramentas e dados para obter avanços fundamentais na compreensão e modelagem de processos e fenômenos na escala de poros (Al-Raoush & Willson, 2005).

Com os avanços tecnológicos e o desenvolvimento das técnicas de processamento de imagens, é possível reconstruir imagens bi e tridimensionais de amostras de solo, que podem ser utilizadas na geração do domínio (o meio poroso), para a aplicação dos métodos numéricos em mesoescala (Zubeldia, 2013).

Desde o pioneirismo na década de 80 até hodiernamente, a geotecnia tem se beneficiado dos métodos numéricos em mesoescala para simulações numéricas em diversos temas, como para as definições do arranjo de partículas que caracteriza a “geometria” do meio poroso, conforme demonstrado em trabalhos mais recentes como Ozelim & Cavalcante (2019), assim como para os estudos dos efeitos independentes do tamanho e formato dos grãos em comportamentos de cisalhamento pré-liquefação, liquefação e pós-liquefação, como demonstrado por Lakkimsetti & Latha (2023).

Seguindo os estudos citados, a geometria do meio poroso e tamanho e formato dos grãos podem ser obtidos com o processamento digital de microtomografias do solo ou a partir do uso de procedimentos numéricos que geram artificialmente geometrias que representam o meio poroso, sem que se entre em detalhes sobre microporosidade e característica de escala compatível.

Dentre as diversas alternativas para o processamento digital de imagens, algoritmos em diferentes linguagens podem ser utilizados, como a linguagem Python, a qual, segundo pesquisa da SlashData divulgada em 2021, é uma das linguagens mais populares de programação, sendo a segunda mais amplamente adotada, logo atrás de Javascript. Essa popularidade se deve à ascensão da ciência de dados e machine learning (ML), haja vista ter sido constatado que mais de 70% de desenvolvedores de ML e cientistas de dados utilizam linguagem Python.

Nesse cenário em constante evolução, esta pesquisa se dedicou a desenvolver um algoritmo alternativo para estimar parâmetros geotécnicos morfológicos através da linguagem de programação Python, reconhecendo sua popularidade e versatilidade impulsionadas pelo crescimento da ciência de dados e da aprendizagem de máquina. Essa abordagem oferece uma nova perspectiva para análises de amostras, visando minimizar ou eliminar erros em suas medições. A metodologia proposta busca inferir alguns parâmetros que caracterizam o solo, apresentando potencial para otimizar os procedimentos operacionais relacionados aos ensaios laboratoriais.

1.2. PROBLEMA E RELEVÂNCIA DA PESQUISA

É usual que métodos semianalíticos, analíticos, empíricos e/ou numéricos analisem o fenômeno de fluxo de fluidos considerando o solo como meio contínuo. Nesses casos, parâmetros como permeabilidade e sucção são utilizados para analisar o fluxo multifásico à luz da mecânica dos solos não saturados. Contudo, esses modelos macroscópicos não consideram a física microscópica do fluxo nos vazios dos solos, impedindo que se reflita a heterogeneidade do processo e não levando em conta a distribuição dos poros, a qual é de grande relevância nos solos profundamente intemperizados (Durand *et al.*, 2012).

Para os compósitos contendo rejeito de mineração, esse problema de fluxo se torna ainda mais relevante pois, segundo Yueqi *et al.* (2020), há uma relação funcional entre a microestrutura de rejeitos arenosos, sua a micromecânica e seu índice de permeabilidade - esses característicos das estruturas microscópicas. Tais informações se tornam importantes para estudos de falhas de barragens de rejeito por *piping* durante as infiltrações, sendo os estudos sobre características e comportamentos mecânico dos rejeitos ainda escassos.

Nesse contexto, de crescente investigação das características microscópicas dos solos e seus impactos no comportamento geomecânico observado, tem-se a seguinte questão de pesquisa:

Considerando a bibliografia existente de estudos de imagens para solos, é possível elaborar uma ferramenta de análise de imagens em código aberto para auxiliar na avaliação das propriedades microscópicas/ em escala de grão de compósitos com rejeitos de mineração?

1.3. JUSTIFICATIVA

Segundo Birmopilis *et al.* (2022), a resposta hidromecânica de argilas naturais em escala de engenharia (macroescala) é governada pela escala de ciência de materiais (micro), levando a uma resposta em macroescala complexa, devido aos efeitos microestruturais decorrentes da organização das partículas e suas interações dentro do solo. Esse entendimento destaca a importância do estudo da estrutura dos solos em microescala, existindo um consenso de que os meios porosos de granulometria mais fina são fortemente influenciados por sua microestrutura interna, a qual precisa ser adequadamente descrita para determinar seu comportamento macroscópico (Pusch & Schomburg, 1999; Pusch & Weston, 2003; Romero & Simms, 2008 *apud* Zubeldia, 2013).

Em consonância com essa perspectiva, foi evidenciado que a tomografia de raio-x é um método confiável e não destrutivo para quantificar informações essenciais de escala de poros, auxiliando na compreensão avançada do comportamento hidrológico e mecânico de solos não saturados em macroescala (Willson *et al.*, 2012).

Durand *et al.* (2012) ressaltam a relevância das características físicas dos solos, como granulometria, textura do material e o teor de umidade inicial, em simulações em mesoescala (escala dos grãos), buscando avaliar seus impactos não só na velocidade e na capacidade de infiltração, como também no potencial de erosão de acordo com a velocidade do fluxo. A compreensão dos processos de infiltração e erosão, por meio de análises em mesoescala, pode beneficiar a elaboração de obras de infiltração, influenciando na escolha de materiais, dimensionamento e técnica de construção, assim como no melhoramento da drenabilidade de solos e aterros existentes.

Zubeldia (2013) destaca que, embora a análise em mesoescala seja mais custosa computacionalmente, é possível tomografar domínios relativamente pequenos onde se pode realizar um estudo mais detalhado do processo de transporte de fluidos, apresentando como vantagem se tratar de um método não destrutivo para o estudo do material. Os avanços tecnológicos e o desenvolvimento das técnicas de processamento de imagens tornam possíveis a reconstrução de imagens bi e tridimensionais de amostras de solo, que podem ser utilizadas

na geração do domínio (o meio poroso) para a aplicação dos métodos numéricos em mesoescala.

Desta maneira, com os avanços computacionais e variabilidade de recursos para processamento de imagens, esta dissertação se justifica no contexto do estudo de ferramentas existentes em código livre (Python) para a segmentação e obtenção de propriedades geométricas de meios porosos, adaptando-as e dando-se ênfase à possibilidade de sua aplicação a materiais advindos da combinação de solos e rejeitos de mineração. Assim, busca-se propor uma possível otimização para o processo de aferição de parâmetros de caracterização física tradicionais por meio de ferramenta de análise de imagens em linguagem Python, utilizando imagens de alta resolução geradas em microtomógrafos de raio X.

A predileção pela linguagem Python respalda-se nas vantagens de ser uma linguagem aberta, dinâmica, interpretada, orientada a objetos com herança múltipla, com tratamento de erros, exceções e gerenciamento automático de memória, que implementam conceitos modernos de linguagem de programação. Além disso, há uma comunidade extremamente ativa, que disponibiliza um vasto conjunto de bibliotecas (Tavares Neto & Silva, 2022).

1.4.HIPÓTESE

Ao adaptar e traduzir em código aberto metodologias de processamento de imagens para solos, será possível definir pré-requisitos, vantagens e desvantagens no seu uso para o estudo de amostras de solo e rejeito de mineração, sendo possível inferir caracterizações geométricas das partículas e estrutura do material, otimizando esforços e tempo dispendidos pelo método convencional de ensaios.

1.5.OBJETIVOS

O objetivo geral deste trabalho é definir proposta de algoritmo e desenvolvimento de script em linguagem Python para processamento de imagens geradas a partir da microtomografia computadorizada de raio-x de amostras geotécnicas e sua análise para obtenção de parâmetros em mesoescala.

1.5.1. OBJETIVOS ESPECÍFICOS

- Compreender, de forma geral, definições e condições utilizadas na geração de imagens por microtomografia computadorizada de raio-x;
- Identificar fatores que possam influenciar na obtenção dos parâmetros dos solos;
- Propor algoritmo de processamento digital de imagens;
- Desenvolver script utilizando, e adaptando quando for o caso, bibliotecas disponíveis para processamento de imagens 2D;
- Validar e comparar resultados de parâmetros morfológicos obtidos computacionalmente com resultados de métodos laboratoriais convencionais ou outros métodos de visão computacional de trabalhos anteriores; e
- Propor soluções gerais de análise computacional a materiais geotécnicos, compósitos ou não, especificando limitações do μ CT na análise de parâmetros obtidos a partir das imagens geradas, otimizando a obtenção de caracterização física do solo.

1.6. ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está estruturada em cinco capítulos: introdução, revisão bibliográfica, materiais e métodos, resultados e discussões e considerações finais. Após a introdução ao tema, o segundo capítulo de revisão bibliográfica cria a base referencial para entendimento do assunto aqui tratado.

No terceiro capítulo são apresentados os materiais e métodos empregados para conduzir a pesquisa. No quarto capítulo os resultados são apresentados e analisados teoricamente. No quinto capítulo as considerações finais são sintetizadas aos principais pontos encontrados e discutidos. E por fim, indicam-se sugestões para pesquisas futuras. Nos apêndices são apresentados os códigos de programação utilizados no trabalho.

CAPÍTULO 2

2. REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma compilação abrangente e concisa de conceitos derivados de uma revisão detalhada da literatura nacional e internacional. Seu propósito é estabelecer uma base sólida de conhecimento sobre o tema de pesquisa, situando o trabalho no contexto acadêmico e profissional.

Para essa dissertação, busca-se fundamentar a pesquisa para a elaboração de uma metodologia de processamento de imagens obtidas por microtomografia de raio-x e análise de parâmetros geotécnicos. Essa metodologia visa inferir parâmetros em mesoescala e otimizar sua obtenção.

A revisão bibliográfica inclui uma breve explicação sobre os equipamentos de microtomografia de raio-x, além de abordar estudos que empregaram metodologias de análise de imagens em diversos materiais na engenharia geotécnica, com ênfase em diferentes parâmetros e análises.

2.1. MICROTOMOGRAFIA COMPUTADORIZADA DE RAIOS-X

A Tomografia Computadorizada (TC) é uma técnica que permite a visualização de seções transversais (cortes internos) de um objeto de forma não destrutiva, sendo uma técnica de caracterização e medida de propriedades físicas de amostras e materiais. O princípio de funcionamento do microtomógrafo de raio-x baseia-se na propriedade dos materiais absorverem esta radiação de forma diferenciada dependendo de sua composição química e densidade.

Derivada da Tomografia Computadorizada (CT), a microtomografia computadorizada (microCT ou μ CT) é uma técnica baseada nos mesmos princípios da CT, produzindo imagens a partir da incidência de raios X, porém com a diferença de ter sido projetado para gerar imagens de alta resolução (Clark & Badea, 2014), com tamanho de pixel (*pixel size*) mais acurado, na faixa de um micrômetro (Landis; Keane, 2010). As imagens são geradas em microtomógrafos compostos por emissor microfocal de raio-X, colimador, suporte para amostra e detector de raios X constituído por detector de cintilação e dispositivo de carga acoplada (CCD) (Boerckel *et al.*, 2014).

O funcionamento do equipamento fundamenta-se na geração de raios X pelo emissor microfocal que incidem sobre o colimador, focando e direcionando a radiação emitida, e sobre pelo corpo de prova, que atenua a radiação a ser medida pelo CCD, convertendo-a em luz visível (fótons) e gerando um sinal eletrônico digitalizado para processamento dos dados de imagem (Hao et al., 2008).

Em geral, as amostras a serem analisadas requerem pouca ou nenhuma preparação (Augusto, 2016), sendo posicionadas em um suporte que é rotacionado em 180° ou 360°, em pequenos intervalos angulares com feixes de raio X atravessando a amostra, passando pelo cintilador e atingindo o detector, que registra a intensidade em forma de projeção. À medida que as projeções são geradas, algoritmos computacionais realizam o imageamento de uma série de seções transversais 2D (*slices*), que possibilitam a reconstrução da imagem tridimensional da amostra (Enríquez-León, 2021). A Figura 2.1 ilustra simplificada os processos de aquisição e reconstrução das imagens.

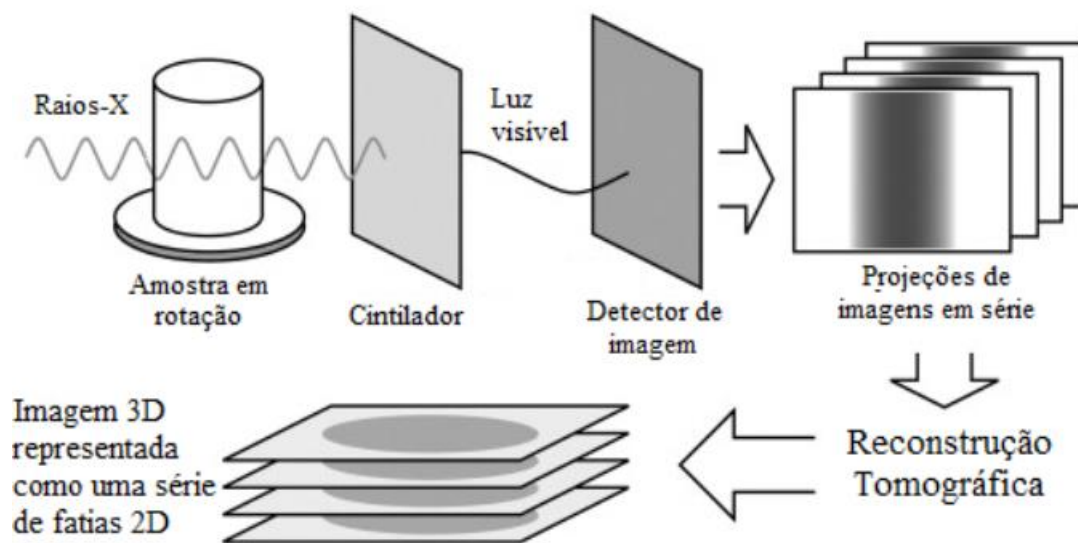


Figura 2.1. Esquematização da aquisição de imagens 2D e 3D por tomografia de raio X. (Fonte: Landis; Keane, 2010 - Adaptado)

Quando um feixe de raios-x atinge um material com uma determinada espessura, parte desse feixe é dispersa, parte é absorvida e uma fração atravessa o material sem interação. A intensidade I do feixe resultante está relacionada à intensidade I_0 do feixe incidente pela Lei de Lambert-Beer Eq. (1), que indica que a intensidade de um feixe diminui exponencialmente à medida que a espessura do meio absorvente aumenta de forma aritmética (Augusto, 2016).

$$I = I_0 e^{-\mu x} \quad (1)$$

onde:

I_0 : Intensidade do feixe incidente, em $W\ m^{-2}$;

I : Intensidade do feixe emergente, em $W\ m^{-2}$;

μ : Coeficiente de atenuação linear total, em cm^{-1} ; e

x : Espessura do material, em cm .

A fonte de raios X emite um feixe de fótons com intensidade I_0 . Esses raios-X atravessam o objeto e são reduzidos devido às interações dos fótons no feixe de raios-X com os elementos constituintes do objeto. O feixe emerge do objeto com uma intensidade reduzida, I , que é então medida pelo detector. O coeficiente de atenuação (μ) depende da densidade, número atômico e energia dos raios X do objeto escaneado, como mencionado por Palombo (2017).

A aquisição das imagens 3D geradas no microtomógrafo é derivada do modelo de matriz bidimensional, representado pelo conjunto de pixels, os quais são substituídos pelo modelo de matriz volumétrica por meio de algoritmos de reconstrução da amostra, onde os pixels ganham dimensão de profundidade espacial, recebendo o nome de voxel (Figura 2.2).

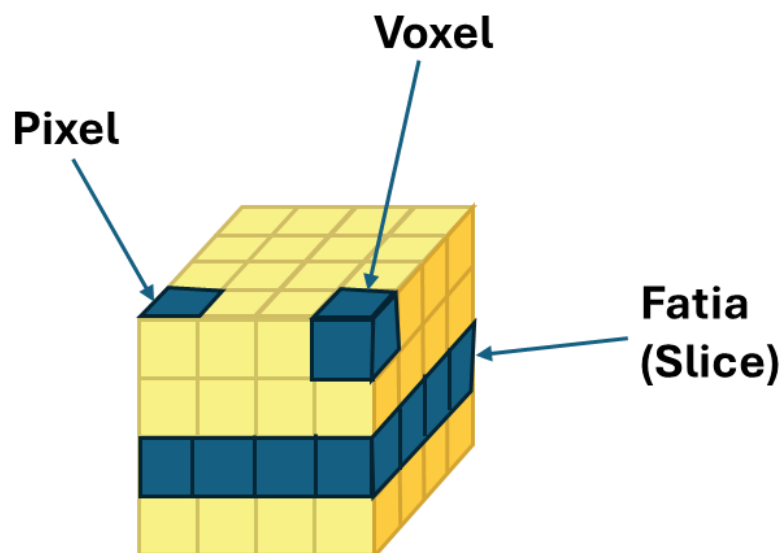


Figura 2.2. Representação gráfica dos elementos pixel, voxel e fatia (*slice*).

Por meio das projeções 2D e da reconstrução tridimensional do sólido, é possível extrair informações tanto externas quanto internas da estrutura escaneada, transformando o campo da mecânica experimental devido a essa capacidade de visualizar o arranjo das partículas em materiais opacos e possibilitando a avaliação de propriedades comumente disponíveis (como porosidade, distribuição do tamanho dos grãos e teor de água), bem como aquelas não disponíveis usando técnicas laboratoriais padrão (como orientação de partículas, variação local de porosidade, teor de água e grau de saturação) (Moscariello *et al.*, 2016)

Apesar das vantagens oferecidas pelas microtomografia computadorizada de raio-x para extrair informações de estruturas escaneadas, é importante considerar as limitações impostas pela resolução. Safari (2021) cita que, embora poros e grãos maiores possam ser facilmente detectados a partir de imagens de microtomografia computadorizada de raio-x, a quantificação de heterogeneidades em pequena escala é severamente influenciada pelos seus limites de resolução.

A resolução limitada pode afetar a precisão e a qualidade das informações obtidas, especialmente em detalhes finos ou áreas de alta complexidade, sendo crucial reconhecer sua influência nos resultados. Portanto, ao realizar análises baseadas em imagens, é essencial levar em consideração as limitações de resolução e implementar estratégias adequadas para lidar com esses desafios. Zubeldia (2016) cita que meios porosos com diâmetros de partículas e/ou poros inferiores a duas vezes a resolução do equipamento, em termos de tamanho de pixel, não devem ser fotografados, considerando que a microtomografia não é capaz de identificar partículas menores ou iguais a duas vezes o tamanho do pixel.

Além dessa restrição quanto ao tamanho da menor unidade que se quer identificar na amostra em relação à resolução do equipamento, outra limitação é a identificação diferenciada de fluidos em amostras geotécnicas, como os vazios de ar e a água. Essa limitação pode ser atribuída ao fato de que a presença de líquidos ou materiais com densidades semelhantes na amostra tornam difíceis a distinção clara dos limites entre os materiais.

Willson *et al.* (2012) realizou uma avaliação quantitativa das propriedades multifásicas da areia fina de Ottawa em diferentes escalas, abrangendo tanto a escala de partículas quanto a escala de poros. Para isso, empregaram imagens de tomografia computadorizada de raios-X para quantificar uma variedade de parâmetros, incluindo tamanhos e áreas de partículas individuais, distribuição de tamanho de grãos e poros, forma e tamanho de poros, distribuição de meniscos de água e áreas de superfície sólida, líquida e gasosa. Além disso, a fase aquosa presente nessas amostras foi dopada com Cs (na forma de CsCl), permitindo a obtenção de dados de alta qualidade sobre o conteúdo e a morfologia da fase aquosa em sistemas de solo

não saturados com duas fases fluidas. Isso possibilitou uma diferenciação de contraste entre os vazios de ar e a fase aquosa, contribuindo para uma compreensão mais detalhada da estrutura e composição desses sistemas.

Evan *et al.* (2015) explicam que, assim como ocorre em demais técnicas de medição, a tomografia de raios X tem vantagens e desvantagens quando comparada com outros métodos, citando que uma das limitações é a sua capacidade de discriminar materiais em função da atenuação dos raios-X: embora a atenuação dos raios X de muitos minerais comuns seja suficientemente diferente para que a tomografia por raios X os distinga, existem alguns nos quais a sobreposição na atenuação dos raios X torna a discriminação impossível com base apenas nessa característica, a exemplo da calcopirita e a magnetita. Modelos atuais de sistemas de tomografia por raios X, como o Zeiss Xradia 520 Versa, superam essa limitação ao medir automaticamente uma amostra em duas energias selecionadas para otimizar a diferença na atenuação dos raios X entre os minerais. A capacidade de discriminar entre minerais em medições 3D também será melhorada à medida que sistemas que combinam tomografia por raios X e fluorescência por raios X, como o Skyscan 2140, se desenvolvam nos próximos anos.

Moscariello *et al.* (2016) citam que a seleção dos parâmetros adequados da microtomografia para realização da varredura é uma etapa importante para se obter um bom contraste entre as fases de uma amostra não saturada, explicando que o contraste é governado pela velocidade de rotação da placa e pelo tempo de espera antes da aquisição de cada radiografia; já o tempo de aquisição e a boa qualidade da imagem resultante são governados pela energia e pela intensidade da fonte de raios X, pelos filtros e pelo número de radiografias.

2.2. PROCESSOS COMPUTACIONAIS COM IMAGENS

O processo computacional com imagens envolve um conjunto de áreas distintas que buscam simular a capacidade visual humana por meio de sistemas autônomos controlados por computadores: Processamento Digital de Imagens (PDI), a Visão Computacional (VC) e a Computação Gráfica (CG).

Embora essas áreas sejam inter-relacionadas, seus objetivos e métodos são distintos, o que ressalta a importância de diferenciá-las claramente: diferente da Visão Computacional (VC) e da Computação Gráfica (CG), o PDI busca realizar o pré-processamento das imagens utilizando técnicas de tratamento, como a correção da iluminação, eliminação de ruído e a segmentação, tendo-se geralmente imagens tanto na entrada quanto na saída do processo. Enquanto isso, o foco da Visão Computacional (VC) é a análise das imagens, identificando seus

componentes e obtendo informações de modelos gerados. Já na Computação Gráfica (CG), que se difere da VC, em que as imagens representam o objeto de estudo, as imagens são o resultado do processo. Na CG são geradas representações visuais seguindo descrições e especificações geométricas de modelos de entrada. Essas distinções, conforme resumidas na Tabela 2.1, são cruciais para uma compreensão clara e precisa das diferentes disciplinas dentro dos processos computacionais com imagens (COVAP, 2021).

Tabela 2.1. Diferenciação dos processos computacionais com imagens. Fonte: COVAP, 2021.

	Computação Gráfica (CG)	Visão Computacional (VC)	Processamento Digital de Imagens (PDI)
Descrição	Cria e altera imagens a partir de dados.	Análise de imagem para criação de modelos.	Transformação de imagem (tratamento).
Entrada → Saída	Modelo → Imagem	Imagem → Modelo	Imagem → Imagem

O tratamento de imagens em PDI e a Visão Computacional, onde o produto de saída daquele é comumente utilizado como matéria prima desse último, frequentemente tem as suas as fronteiras confundidas, levando à confusão sobre onde uma termina e a outra começa. Embora não haja uma demarcação clara entre essas duas áreas, segundo COVAP (2021) é possível adotar um paradigma que considere três níveis de processamento:

- I. **Baixo nível:** A nível de pixel, realiza operações de pré-processamento, sendo utilizada, por exemplo, na redução de ruído, aumento de contraste e restauração. Nesta etapa tanto a entrada quanto a saída são imagens.
- II. **Médio nível:** Operações mais complexas, como segmentação, partição e reconhecimento de objetos individuais. A entrada é uma imagem, mas a saída pode ser um conjunto contendo os atributos extraídos das imagens, como formas, bordas e objetos individuais.
- III. **Alto nível:** Interpretação do conteúdo da imagem e análise, muitas vezes processos associados com as funções da visão, como a classificação, o reconhecimento e o rastreamento de objetos.

Baseado nesses níveis, pode-se considerar que o processamento de imagem atua nos primeiros dois níveis, ou seja, envolve o pré-processamento e processos de extração de elementos de imagens até o reconhecimento de componentes individuais. O campo da Visão Computacional, mesmo associado com os níveis mais baixos, se torna mais evidente a partir das técnicas de alto nível de processamento, que utilizam informações extraídas das imagens para processos de inferências ou aprendizados, resultando em aplicações em diferentes áreas (COVAP, 2021).

2.2.1. PRÉ-PROCESSAMENTO E PROCESSAMENTO EM IMAGENS

Nos últimos anos, o emprego de técnicas avançadas de pré-processamento e processamento de imagens tem assumido uma importância significativa na análise e caracterização de materiais. Diversos estudos recentes têm explorado essas técnicas para aprimorar as análises dos materiais empregados na engenharia geotécnica, observando-se um padrão frequentemente adotado nas fases de pré-processamento e processamento de imagens, com variações nas escolhas de suas técnicas: eliminação de ruídos, segmentação por binarização e aplicação do algoritmo *Watershed*. É importante ressaltar que, embora essas sejam abordagens comuns, existe uma variedade de escolhas dessas técnicas disponíveis, refletindo a diversidade e a adaptação necessárias para lidar com as particularidades de cada estudo.

A eliminação de ruídos, representada no exemplo da Figura 2.3, tem a função principal de melhorar a qualidade da imagem ao preservar detalhes importantes, reduzindo o consumo de recursos computacionais e possibilitando uma melhor precisão em análises e medições.



Figura 2.3. (a) Imagem original (b) Imagem com filtro *Gaussian* com Kernel 17x17

A segmentação por binarização é um processo que divide uma imagem em duas regiões distintas, geralmente preto e branco, com base em um limiar de intensidade definido. Essa técnica é comumente utilizada quando se deseja separar objetos de interesse do fundo da imagem. Ao aplicar a binarização, os pixels da imagem são classificados como pertencentes à região preta (ou valor zero) se sua intensidade for menor que o limiar, e como pertencentes à região branca (ou valor máximo) se sua intensidade for maior ou igual ao limiar. Essa abordagem simplifica a imagem e destaca os objetos de interesse, facilitando análises subsequentes. A Figura 2.4 ilustra o processo de segmentação por binarização considerando a limiarização global da imagem.

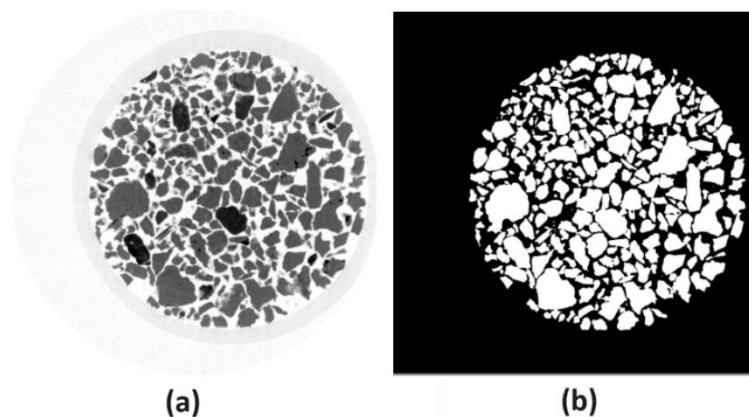


Figura 2.4. (a) Imagem original com inversão de cores (b) Imagem binarizada

A aplicação do algoritmo de *Watershed* em processamento de imagens é uma técnica utilizada especialmente em casos em que existem regiões com limites mal definidos ou complexos, como objetos que estão muito próximos, encostados uns aos outros ou sobrepostos. A função principal desse algoritmo é segmentar uma imagem em regiões ou objetos com base em critérios de similaridade visual, como intensidade de cor ou textura, auxiliada pela operação matemática “transformada de distância”, que calcula a distância de cada pixel até o pixel mais próximo pertencente a um conjunto de pixels de interesse. Essa transformada atribui a cada pixel um valor que representa a sua proximidade ao objeto de interesse, sendo útil para identificar contornos e realizar análises de forma, gerando marcadores iniciais para o algoritmo *Watershed*.

O algoritmo *Watershed* usa a ideia de simular um processo de enchente sobre uma paisagem topográfica, metaforicamente referente a uma bacia hidrográfica geológica ou divisão de drenagem que separa as bacias de drenagem, onde os vales são preenchidos até que as águas se encontrem nos picos de montanhas.

Pensando-se analogamente a uma imagem em escala de cinza como um mapa topográfico, os pixels de alta intensidade representariam os picos, enquanto os valores de baixa intensidade representariam os vales, ou mínimos locais. A Figura 2.5 ilustra um exemplo de aplicação do algoritmo *Watershed*.

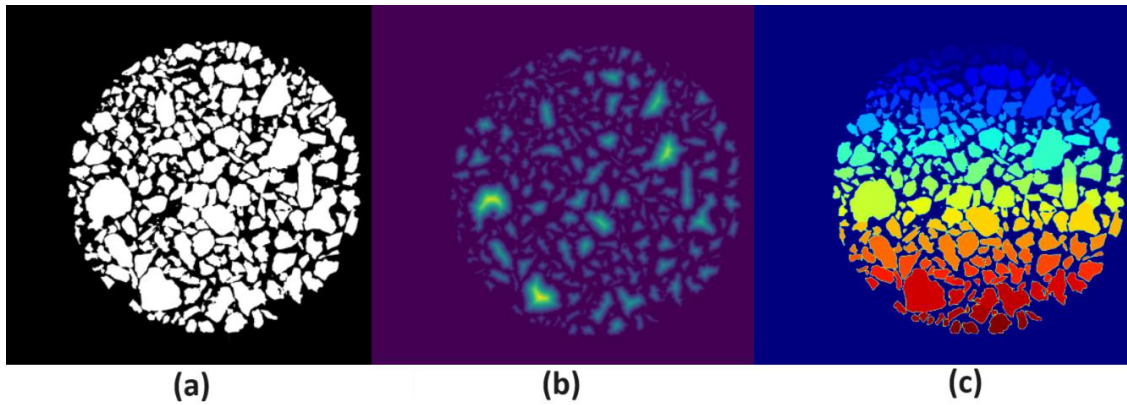


Figura 2.5. Aplicação do algoritmo *Watershed*: (a) Imagem binarizada (b) Aplicação da transformada de distância (c) Rotulação por *Watershed*

A aplicação dessas técnicas é evidenciada em estudos recentes, exemplificados pelo trabalho de Birmipilis *et al.* (2022), que investigaram a estrutura de uma amostra indeformada de argila sensível natural de Utby na Suécia, utilizando nano e microtomografias de alta resolução. A investigação foi baseada na obtenção de parâmetros usados para caracterizar a estrutura do solo a partir dos dados obtidos nas imagens 2D e 3D: tamanho das partículas, formato das partículas e orientação das partículas. Para isso, foi utilizado o software de processamento de dados de código-fonte aberto Fiji (ImageJ) e scripts baseados em Python para realização do pré-processamento e o processamento de imagens, os quais envolveram:

- I. Redimensionamento numérico e a conversão em matrizes 3D das imagens em escala de cinza obtidas, retirando-se um subconjunto dos dados originais com dimensões de 1.000 x 1.000 x 1.000 pixels para análise quantitativa, com o objetivo de se reduzir o esforço computacional e manter resultados estatisticamente significativos;
- II. Aplicação do filtro *Gaussian Blur* para redução dos ruídos e atenuação dos artefatos ou imperfeições que surgem durante a reconstrução da imagem tridimensional a partir de dados bidimensionais;

- III. Segmentação para identificação dos grupos de objetos para obtenção de propriedades geométricas, utilizando binarização com escolha manual de valor limite;
- IV. Aplicação do algoritmo *Watershed* para auxiliar na criação de rótulos únicos para cada grão individual a partir do conceito de máximos locais;
- V. Correção de “super-segmentações” geradas na aplicação do algoritmo *Watershed* para evitar superestimação de máximas locais em um único grão, que geralmente resulta em múltiplos rótulos em grãos muito grandes; e
- VI. Aplicação de um filtro de triagem para remover todos os rótulos com volume de partícula menores de 200 pixels, pois essas pequenas partículas são sensíveis a artefatos de segmentação que distorcem as análises subsequentes.

O resultado após esses passos de pré-processamento e processamento de imagem é apresentado na Figura 2.6, que representa um recorte de uma imagem 2D obtida na nanotomografia, removendo-se os rótulos com uma imagem abaixo de 200 pixels.

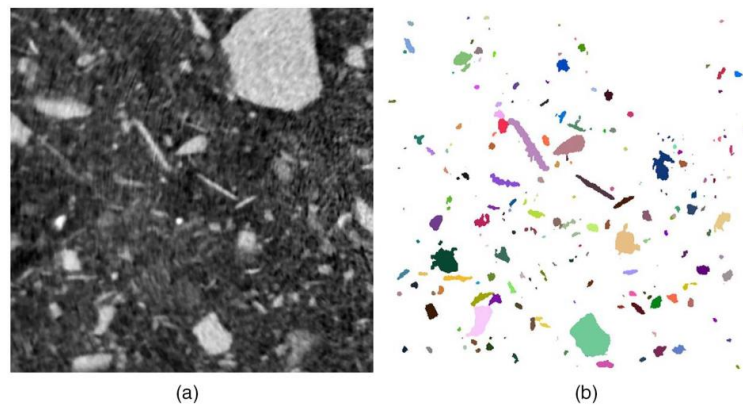


Figura 2.6. (a) Imagem original (b) Imagem rotulada após pré-processamento e processamento. (Fonte: Birmipilis *et al.*, 2022)

Assim como Birmipilis *et al.* (2022), outros trabalhos também trazem o pré-processamento e processamento das imagens como etapas fundamentais para melhorar a qualidade, relevância e usabilidade dos dados da imagem a serem utilizados em análises posteriores:

Safari *et al.* (2021) elaboraram um fluxo de trabalho para avaliar a distribuição de poros e tamanhos de grãos em amostras geológicas porosas a partir de imagens obtidas por microtomografia computadorizada de raios-x, sendo utilizado: filtro *NL-means* nas imagens

2D, em vez do volume 3D, devido ao baixo desempenho computacional do filtro 3D no Avizo; segmentação da imagem pela técnica de binarização, considerando-se essa etapa crucial devido à sua influência em toda análise quantitativa subsequente e verificando-se que algoritmos de limiar automático, como o método de Otsu, tenderam a segmentar excessivamente as imagens em escala de cinza em quase todas as amostras, sendo então adotado a limiarização manual, apesar ser trabalhosa e poder gerar uma tendência do operador; transformação de distância euclidiana e aplicação do algoritmo *Watershed* para fornecer imagens rotuladas para volumes de poros e grãos.

Pae & Moon (2020) buscaram investigar quantitativamente a tortuosidade da areia cimentada com diferentes teores de cimento, baseada em imagens de microtomografia computadorizada convertidas em espaço poroso 3D, utilizando-se os filtros *Gaussian Blur* e difusão anisotrópica, a segmentação pelo método de Otsu e o algoritmo *Watershed* para rotulação de cada grão pela técnica de clusterização.

Cheng *et al.* (2020) desenvolveram um aparelho triaxial em miniatura em conjunto com um microtomógrafo de raio X para investigação *in situ* do comportamento mecânico em microescala de solos granulares sob cisalhamento, tendo-se como um dos objetivos a análise de porosidade a partir de imagens tratadas por meio de filtro de difusão anisotrópica, binarização com determinação de limiar global e rotulação de cada grão com o auxílio de *Watershed* e de técnica de controle de supersegmentação.

Borges *et al.* (2019) apresentaram algoritmos de análises de granulometria, esfericidade dos grãos, índice de vazios e tortuosidade dos vazios, utilizando-se uma areia seca mal graduada retirada do Rio Tietê em São Paulo, peneirada e selecionada à fração entre 1,19 e 0,25 mm para análise no microtomógrafo. Com as imagens geradas no microtomógrafo, processaram-se as imagens em três etapas com o uso de um algoritmo desenvolvido no *Mathematica v. 10*: binarização pelo método de Otsu, identificação da área central de cada grão, determinação da área central para cada grão gerado na etapa anterior, individualização ou identificação morfológica de cada grão utilizando o algoritmo *Watershed*.

Taylor *et al.* (2015) buscaram descrever um método para identificar constrições de vazios em imagens de microtomografia computadorizada de um material granular com o objetivo de avaliar o desempenho de filtros de areia. Por meio do software de código aberto *ImageJ*, as imagens foram tratadas utilizando-se o filtro *median filtering*, binarização baseada no limiar de Otsu, aplicação da transformada de distância e do algoritmo *Watershed*.

Dessa forma, é notado que há um padrão frequentemente adotado nas fases de pré-processamento e processamento de imagens, com algumas variações nas escolhas de técnicas

de filtros de eliminação de ruídos, escolhas de limiares para segmentação binária e aplicação adicional de técnicas de controle da supersegmentação do algoritmo *Watershed*, refletindo a diversidade e a adaptação necessárias para lidar com as particularidades de cada estudo.

Realizadas essas etapas de pré-processamento e processamento das imagens que resultaram na individualização das partículas, torna-se possível iniciar as análises que extraem informações por processos de inferência e interpretação do conteúdo da imagem.

2.3. ANÁLISE POR IMAGENS: ESFERICIDADE

A esfericidade dos grãos é um parâmetro utilizado na caracterização de partículas ou grãos em diversas áreas, como geologia, geotecnia e ciências dos materiais. Esse parâmetro desempenha um papel importante na caracterização de materiais granulares, podendo ser utilizado como entrada para simplificar a calibração de modelos pelo método de elementos discretos (Rorato *et al.*, 2018), influenciando em parâmetros de deformação (Payan *et al.*, 2017), no grau de anisotropia (Oboudi *et al.*, 2016), no coeficiente de arraste no sistema partícula-fluido (Mendes & Melo, 2011) e na resistência mecânica verificada experimentalmente (Casagrande, 2005).

A definição de esfericidade pode ser entendida como a descrição do quão próximo um grão ou partícula se assemelha a uma esfera, sendo um índice de forma que relaciona o volume da partícula e um volume de uma esfera. Quanto mais próximo o valor da esfericidade estiver de 1, mais esférica é considerada a partícula.

Ribeiro & Bonetti (2013) destacam que, devido à necessidade de instrumentação ótica mais específica e ao processamento de imagens mais demorado associados à análise tridimensional de partículas, foram desenvolvidas equações para calcular a esfericidade no contexto bidimensional. Embora alguns autores tenham denominado esse parâmetro como "esfericidade" para análises de imagens bidimensionais (Riley, 1941), outros utilizaram o termo "circularidade", que foi considerado por Ribeiro & Bonetti (2013) como o mais apropriado. Apesar disso, autores recentes ainda utilizam o termo "esfericidade" ao se analisar imagens bidimensionais, como Borges *et al.* (2019).

Rorato *et al.* (2019) contextualiza que a esfericidade foi proposta originalmente por Wadell (1932), exigindo-se uma medição desafiadora da área superficial da partícula. Antes disso a esfericidade só poderia ser avaliada utilizando métodos de medidas mais simplificadas, baseadas em projeções bidimensionais. Ainda segundo Rorato *et al.* (2019), hoje estão em uso

essas medidas aproximadas por serem mais econômicas, sendo objeto de sua pesquisa o estudo de qual é a melhor representação aproximada mais se correlaciona com os valores 3D.

Existem diversas metodologias para calcular a esfericidade de grãos ou partículas, dependendo do contexto e das características específicas do material em estudo. Esses métodos podem variar desde abordagens simples, como a razão entre o volume da partícula e o volume de uma esfera de mesma área superficial, até técnicas mais complexas que levam em consideração aspectos geométricos detalhados da partícula. A escolha do método adequado geralmente depende da precisão desejada e das limitações práticas associadas à análise das imagens das partículas.

Wadell (1935), reconhecendo a complexidade de se obter a medição da área superficial das partículas, propôs uma alternativa prática bidimensional chamada esfericidade do diâmetro: obtém-se uma projeção plana da partícula, calcula-se sua área e, em seguida, calcula-se o diâmetro de um círculo com igual área à projeção; com esse valor, calcula-se a esfericidade pela relação entre esse diâmetro e o diâmetro do círculo circunscrito mínimo para a área projetada.

De acordo com Riley (1941), a esfericidade (ou circularidade em termos de imagens bidimensionais) pode ser determinada pela raiz quadrada da razão entre o diâmetro do maior círculo inscrito na partícula pelo diâmetro do menor círculo circunscrito na mesma partícula.

D'Agostinho (2004) propôs o cálculo de circularidade a partir da relação entre a área e o perímetro da partícula, conforme Eq. (2).

$$Circularidade = \frac{4 \pi \text{Área}}{(\text{Perímetro})^2} \quad (2)$$

Segundo Cho *et al.* (2006), a esfericidade pode ser quantificada como o diâmetro da maior esfera inscrita em relação ao diâmetro da menor esfera circunscrita.

Araújo *et al.* (2015) analisaram imagens para a determinação da forma e textura de areias, citando que há diversas definições para a esfericidade: Mendes (1972), que definiu a esfericidade a partir de sólidos 3D como a relação entre a área superficial da partícula e o seu volume; Mora & Kwan (2000), que definiram a esfericidade 3D como a razão da área superficial de uma esfera de mesmo volume da partícula e a sua área real; e Rittenhouse (1943), que definiu esfericidade 2D como a relação entre o diâmetro do círculo com uma área igual à projeção da partícula e ao diâmetro do menor círculo circunscrito à partícula Eq. (3), podendo variar de 0 a 1; na prática esses valores vão de 0,45 (alongada) a 0,97 (muito esférica).

$$Esfericidade = \frac{d_{eq}}{d_c} \quad (3)$$

onde:

d_{eq} : diâmetro do círculo com uma área igual à projeção da partícula na imagem 2D; e

d_c : diâmetro do menor círculo circunscrito à projeção da partícula na imagem 2D.

No estudo feito por Borges *et al.* (2019) de avaliação da esfericidade utilizando-se imagens 2D é citada a definição de esfericidade de Wadell (1932) como a razão entre o diâmetro do círculo de área igual à projeção quando projetada em sua maior face, e o diâmetro do menor círculo que circunscribe a projeção. Além disso, é citada também a esfericidade de Krumbein (Krumbein, 1941), que é a raiz cúbica da razão do produto das duas dimensões maiores menores pela maior dimensão maior. Apesar das citações, Borges *et al.* (2019) adotou a relação mostrada na Eq. (4) para avaliar a esfericidade amostra de areia pura.

$$Esfericidade = \frac{A_{sobreposta}}{A_{equivalente}} \times 100\% \quad (4)$$

onde:

$A_{sobreposta}$: é a área correspondente à intersecção da sobreposição concêntrica da área da projeção do grão sobre um círculo de igual área da projeção do grão; e

$A_{equivalente}$: é a área do círculo com o mesmo valor da área da projeção do grão. A Figura 2.7 traz a ilustração da esfericidade adotada por Borges *et al.* (2019) para melhor esclarecimento.

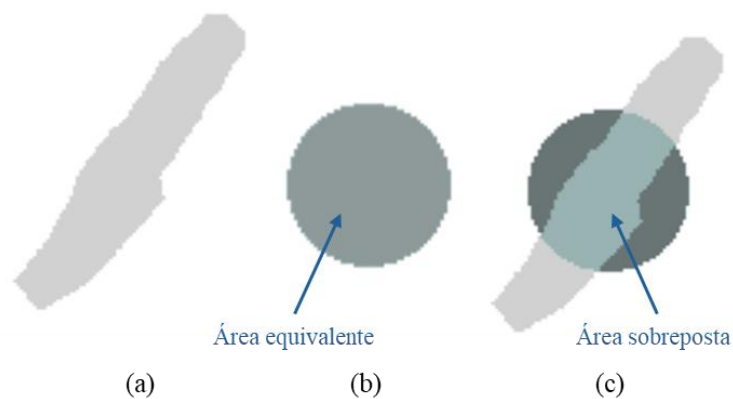


Figura 2.7. (a) Projeção do grão; (b) Círculo de mesma área da projeção do grão; e (c) Sobreposição concêntrica da projeção do grão sobre o círculo de área equivalente (Fonte: Borges *et al.*, 2019 – Adaptado)

De acordo com Rorato *et al.* (2019), dado que existem várias abordagens para determinar a esfericidade, é crucial considerar a direção da projeção aplicada nas análises 2D. Os autores ressaltam que nem sempre é possível identificar e definir essa direção, porém, mesmo quando as projeções das partículas são feitas de forma aleatória ou não, os parâmetros de forma em 2D e 3D podem ser comparados diretamente.

Além disso, de acordo com Rorato *et al.* (2019), a esfericidade 2D proposta por Krumbein & Sloss (1963), que utiliza uma referência gráfica de dupla entrada para inferir o arredondamento e a esfericidade por comparação visual, apresenta uma correlação baixa com a medida de esfericidade 3D. No entanto, a força da correlação para outras abordagens 2D aumenta significativamente quando as medidas são realizadas em projeções orientadas (ou seja, com área máxima). Em particular, a esfericidade do perímetro Eq. (5, representada pela Figura 2.8, é o descritor 2D que parece estar mais correlacionado com a esfericidade 3D, sendo essa correlação muito mais forte quando é utilizada a projeção máxima da partícula em comparação com a orientação aleatória, destacando a sensibilidade à orientação das partículas desse descritor. A esfericidade do perímetro é definida como a razão entre o perímetro do círculo que possui a mesma área que a partícula (p_n) e o perímetro real da partícula (p) (Sandeep *et al.*, 2023).

$$s_p = \frac{p_n}{p} \quad (5)$$

onde:

s_p : esfericidade do perímetro;

p_n : perímetro do círculo que possui a mesma área que a partícula; e

p : perímetro real da partícula na imagem 2D.

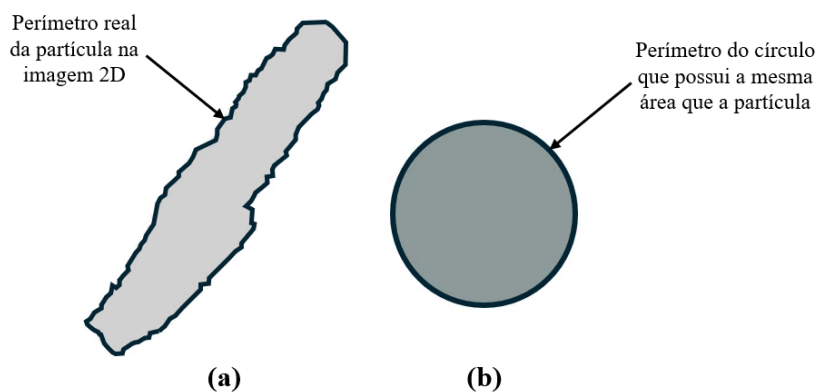


Figura 2.8. (a) Projeção do grão e (b) Círculo de área equivalente ao grão projetado (Fonte: Borges et al., 2019 – Adaptado)

2.4. ANÁLISE POR IMAGENS: GRANULOMETRIA

A granulometria é uma propriedade fundamental na mecânica dos solos que descreve a distribuição dos tamanhos das partículas presentes em uma amostra de solo. Essa distribuição é essencial para classificar e entender o comportamento e as propriedades do solo, como sua capacidade de drenagem, compactação e resistência.

Abrangente e convencionalmente, a análise granulométrica utiliza os métodos de peneiramento e sedimentação para determinar a distribuição dos tamanhos das partículas presentes em uma amostra de solo. O método de peneiramento envolve a passagem de uma amostra de solo por uma série de peneiras com aberturas de diferentes tamanhos, empilhadas em ordem decrescente de abertura, ficando retidas as partículas maiores nas peneiras de malha mais grossa e as menores, nas mais finas. Após a separação, as massas retidas em cada peneira são pesadas e expressas como porcentagem do peso total da amostra, permitindo a construção da curva granulométrica, que mostra a distribuição percentual nos diferentes diâmetros. Nesse método, algumas desvantagens podem ser evidenciadas: além de ser trabalhoso e demorado devido à preparação da amostra, à operação dos agitadores (ou manual) e à medição da massa em cada peneira, há a necessidade de se realizar manutenções nas balanças, agitadores e peneiras, que recorrentemente ficam obstruídas, além de ser um processo barulhento e que gera vibrações (Vangla *et al.* 2014).

Em relação ao método de sedimentação, este se destaca especialmente para partículas de tamanho reduzido, medindo a mudança na densidade em uma suspensão solo-água ao longo do tempo, à medida que as partículas do solo se depositam. Com o passar do tempo, as partículas mais pesadas sedimentam mais rapidamente, enquanto as mais leves permanecem em

suspensão por períodos mais longos. Medindo-se a altura das camadas de sedimentação em intervalos regulares, é possível calcular a taxa de sedimentação das partículas, fornecendo uma distribuição granulométrica da amostra. Embora seja um método amplamente utilizado, algumas desvantagens podem ser pontuadas: a execução do processo pode consumir muito tempo e mão de obra (Barman & Choudhury, 2020) e apresentam dependência do operador e do instrumento (Faroughi & Huber, 2016).

Pesquisadores têm utilizado técnicas de processamento de imagens para determinar distribuições de diâmetros de grãos como alternativa ao método de análise por peneira, que pode ser descrito como relativamente difícil e inconveniente (Aydın & Kurnaz, 2023).

Youwai & Makam (2023) propuseram um fluxo de trabalho de visão computacional para obter a análise granulométrica do solo, apresentando uma nova abordagem que utiliza visão computacional (CV) e a linguagem de programação Python para análise granulométrica de solos de granulação grossa, empregando uma câmera padrão de telefone celular. Sua metodologia envolve o uso da biblioteca OpenCV para detectar e medir partículas de solo em fotografias digitais tiradas sob condições normais de iluminação. Para uma determinação do tamanho das partículas, foi utilizado um marcador de calibração com dimensões conhecidas para ser usado como referência para encontrar a proporção entre pixels e comprimento e detectar o limite dos grãos nas imagens binárias para aplicação do menor retângulo que envolva os pixels referentes aos grãos (Figura 2.9), calculando-se então o tamanho da partícula como a média da largura e altura do retângulo.

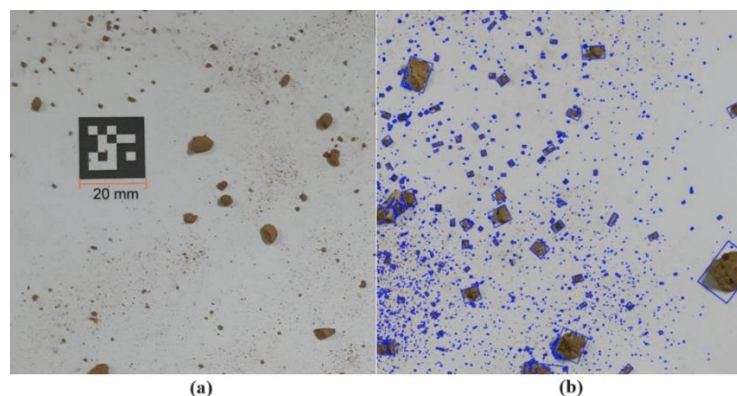


Figura 2.9. (a) Marcador ArUco em imagem dos grãos sobre papel A4 e (b) Aplicação da função *minAreaRect* (Fonte: Youwai & Makam, 2023)

Aydın & Kurnaz (2022) utilizaram a técnica de processamento digital de imagens como alternativa ao método de análise por peneira para determinação da distribuição de partículas em

solos granulares, comparando-se com os resultados do método tradicional de peneiramento. Para se chegar ao tamanho das partículas, foi necessária uma conversão de pixels para milímetros para obter os resultados no sistema métrico na análise feita na fotografia, utilizando-se como referência uma ferramenta de medição de comprimento (régua, fita métrica etc.) na mesa de luz. A conversão da distância da unidade para pixels foi realizada usando a ferramenta de medição da interface do programa Image Pro Plus.

Safari *et al.* (2021) propuseram a aplicação de um fluxo de trabalho de processamento de imagens para a extração das distribuições de tamanho de poros e grãos a partir de imagens de microtomografia computadorizada por raios-X e microscopia eletrônica de varredura em cinco diferentes arenitos e uma amostra de carbonato: os tamanhos dos grãos foram calculados considerando o chamado diâmetro equivalente, definido como o diâmetro de uma esfera com o mesmo volume que o de uma região de interesse segmentada.

Borges *et al.* (2019) apresentaram algoritmo de processamento de imagens e formulações para analisar um material arenoso a partir de imagens bidimensionais, gerando informações supervisionadas sobre o processo de caracterização da morfologia do solo, dentre elas a distribuição granulométrica: o cálculo dos diâmetros equivalentes dos grãos para distribuição granulométrica foi realizado considerando a menor projeção 1D de cada grão, obtida ajustando-se o menor retângulo delimitador a cada grão e escolhendo-se o menor tamanho desse polígono. Para possibilitar a plotagem no gráfico de distribuição granulométrica utilizando-se somente as imagens geradas, os autores adotaram que a massa específica é constante para todas as partículas, considerando que a areia estudada é um único material, assim como a altura referente à distância entre as fatias escaneadas. Isso faz com que seja simplificado o parâmetro a ser plotado no gráfico de granulometria no eixo das ordenadas, tornando a relação entre a massa dos grãos acumulada e a massa total dos grãos na amostra correspondente à área acumulada dos grãos e área total dos grãos na amostra, definindo-se os valores a serem plotados no eixo das ordenadas Eq. (6).

$$\%accum(m_{si}) = \frac{m_{si}}{m_t} = \frac{\sum(A_i h)\rho_{si}}{(A_t h)\rho_s} = \frac{\sum A_i}{A_t} \quad (6)$$

onde:

m_{si} : massa do grão de um grão i ;

m_t : massa total dos grãos;

A_i : área de um grão i ;

A_t : área total dos grãos;

h : distância entre as fatias;

ρ_{si} : é a massa específica de um grão i ; e

ρ_s : é a massa específica dos grãos, considerada a mesma de um grão i ;

Dipova (2017) determinou a distribuição granulométrica das areias por meio de técnicas de análise de imagens, utilizando aparelhos simples (câmeras não profissionais e software de código aberto) e uma técnica automatizada de contagem de pixels: os diâmetros equivalentes considerando imagens 2D foram calculados conforme similaridade de forma de uma partícula com uma elipse ou um retângulo, tendo como variáveis as dimensões de Ferret.

Evans *et al.* (2015) descreveram o desenvolvimento de um método que utiliza microtomografia de raios X para fornecer medição tridimensional da distribuição do tamanho dos grãos e localização espacial dos minerais em um minério de pirita contendo ouro, por meio do pacote do software *Trimble eCognition*, que foi utilizado para quantificar as características texturais que são insumos essenciais para modelar a liberação mineral neste minério durante a cominuição. No processo de análise de imagem, a segmentação em regiões de escala de cinza classifica os minerais previamente conhecidos do minério de acordo com suas semelhanças de escala de cor, conforme demonstrado na Figura 2.10, que identifica os minerais pirita, silicatos e carbonatos, sendo observado que os vários minerais de silicato neste minério (quartzo, albita, moscovita) não podem ser distinguidos individualmente porque seus valores de atenuação de raios X são muito semelhantes. A calibração da etapa de classificação para vincular o valor da escala de cinza na imagem às espécies minerais do minério foi confirmada utilizando um sistema de mineralogia automatizado baseado em Microscopia Eletrônica de Varredura.

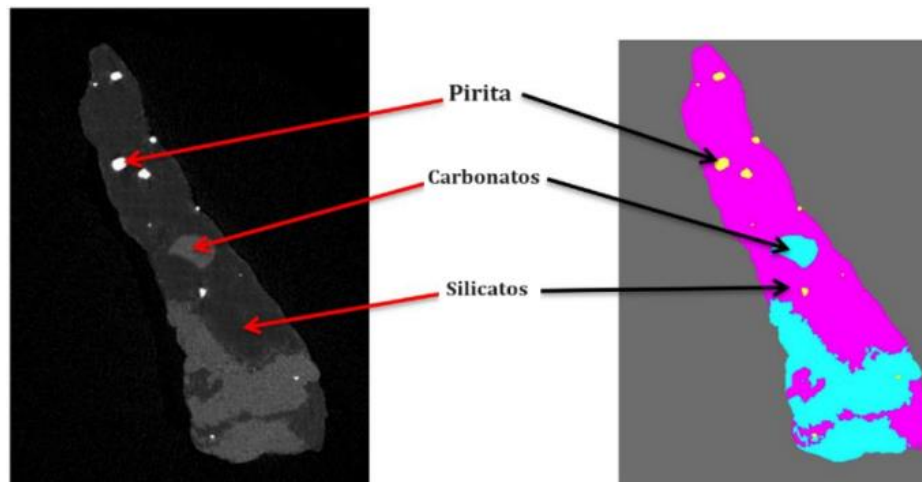


Figura 2.10. Imagem segmentada e classificada do software eCognition. (Fonte: Evans *et al.*, 2015- Adaptado)

Para casos de solos de granulação fina com tamanhos de partícula de 0,075 mm ou menor, a distribuição de tamanho dos grãos é difícil de prever com a análise típica de imagem digital individualizada de cada grãos devido ao tamanho extremamente pequeno das partículas e à dificuldade de separar partículas individuais, sendo proposta por Kim et al. (2023) uma outra forma de análise dos tamanhos dos grãos: os autores propuseram uma abordagem para fornecer uma análise eficiente e automatizada de solos de granulação fina a partir de imagens digitais adquiridas com uma câmera convencional do ensaio de sedimentação, com o objetivo de reduzir a dependência do operador e erros nas leitura, utilizando bibliotecas Python (como NumPy, Pandas, Pillow, OpenCV e Matplotlib) para realizar tarefas de processamento de imagem, incluindo recorte e extração de cor. Para isso, foi necessário adquirir uma série de imagens digitais da suspensão solo-água durante a sedimentação, substituindo-se o cilindro de vidro comumente utilizado nos ensaios de sedimentação por um recipiente de acrílico devido à possibilidade de interferência do reflexo de luz ao se obter as imagens. À medida que as partículas se depositavam, a distribuição de cores dentro da suspensão mudava e medidas de distância de sedimentação (L), no tempo T para uma partícula com diâmetro D (Figura 2.11), pode ser determinada usando a Lei de Stokes, assim como é no ensaio de sedimentação.

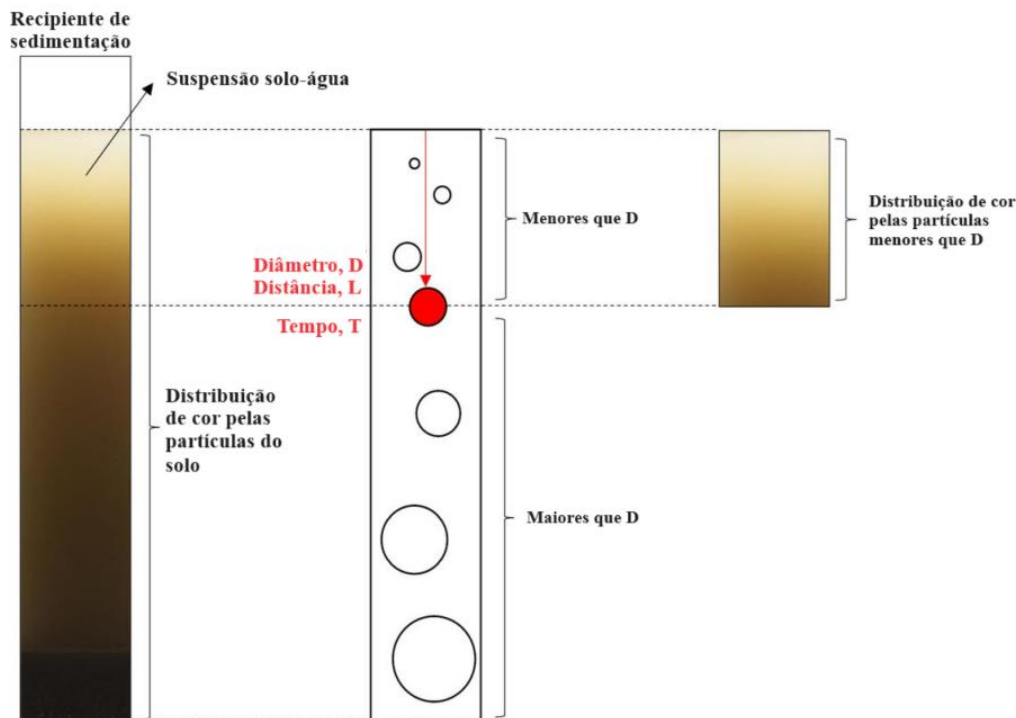


Figura 2.11. Diagrama conceitual de análise de partículas de tamanho de partículas finas (Fonte: Kim *et al.*, 2023 - Adaptado)

2.5. ANÁLISE POR IMAGENS: ÍNDICE DE VAZIOS E POROSIDADE

De forma clara e simples, Pinto (2006) define índice de vazios (e) como a relação entre o volume de vazios e o volume das partículas sólidas, não podendo ser determinado diretamente, mas calculado a partir de outros índices, e variando seus valores entre 0,5 e 1,5, podendo haver casos de argilas orgânicas com índices de vazios superiores a 3. O autor também define porosidade (n) como relação entre o volume de vazios e o volume total analisado, com valores variando entre 30 e 70%.

O estudo do índice de vazios e porosidade se faz relevante devido a sua influência em análises de fluxo e resistências de materiais porosos, interferindo em propriedades de transporte que são fortemente influenciadas pelo tamanho e arranjo de seus poros, sendo os estudos dos vazios dos materiais porosos um parâmetro-chave em muitos cálculos de engenharia para quantificar a geometria complexa do espaço poroso (Safari *et al.* 2021). Silva *et al.* (2023) cita que conceitos relacionados aos vazios, como tortuosidade, percolação e conectividade dos poros, influenciam no transporte de massa no concreto e podem influenciar propriedades mecânicas como resistência, retração e durabilidade.

Pesquisadores têm utilizado técnicas de processamento de imagens para determinar a relação dos vazios utilizando análise de imagens, buscando-se reduzir os esforços em laboratórios com ensaios tradicionais: Rozenbaum *et al.* (2012) que buscou avaliar a porosidade do solo resultante da associação de grãos de silte com uma fase argilosa por meio de microtomografia computadorizada de raio-x em volumes 3D; Silva *et al.* (2023) realizou uma revisão básica de técnicas (ressonância Magnética Nuclear, Porosimetria por Intrusão de Mercúrio e Microtomografia Computadorizada de Raios X) para análise da estrutura dos poros, buscando-se compreender a sua estrutura em materiais e assim adotar a técnica que melhore ou facilite a tomada de decisão onde a porosidade exerce influência; Fopa *et al.* (2023) apresentaram um estudo em escala de poros do transporte de nanopartículas em meios porosos arenosos, demonstrando a influência da injeção de nanopartículas no parâmetro porosidade.

No estudo realizado por Borges *et al.* (2019), foram propostos algoritmos de processamento de imagens e abordagens para a análise de materiais arenosos utilizando microtomografia computadorizada de raio-x. Uma das análises realizadas consistiu na determinação do índice de vazios a partir de imagens bidimensionais, considerando a razão baseada na área de vazios e sólidos, conforme Eq. (7).

$$e = \frac{A_v}{A_s} \quad (7)$$

onde:

e : índice de vazios;

A_v : área de vazios; e

A_s : área de sólidos.

CAPÍTULO 3

3. MATERIAIS E MÉTODOS

Neste capítulo, são detalhados os materiais e métodos empregados na presente pesquisa, abrangendo uma análise dos materiais selecionados para estudos de caso, os equipamentos utilizados para coleta e análise das imagens, bem como os métodos adotados para se chegar aos resultados. Dois estudos de caso foram analisados considerando materiais estudados em pesquisas anteriores, destacando-se os procedimentos específicos adotados para investigar as questões de pesquisa propostas.

Além disso, são apresentados os instrumentos e equipamentos utilizados durante a etapa de preparação do material, escaneamento e análise das imagens, juntamente com os procedimentos adotados na execução dos trabalhos. Esta seção tem como objetivo fornecer uma compreensão abrangente dos processos e técnicas empregados neste estudo para transparência e melhor entendimento dos resultados.

3.1. MATERIAIS

Nesta pesquisa, o foco investigativo é baseado em análise de imagens microtomografadas de três materiais estudados em pesquisas anteriores: areia pura, um dos objetos de pesquisa de Zubeldia (2013), rejeito de minério de ferro e rejeito de minério de ferro estabilizado com polímero, ambos estudados por Carneiro (2020). Para possibilitar uma análise visual dos grãos, foram selecionados esses materiais devido às suas características granulares e partículas de grãos maiores.

3.1.1. AREIA

Uma das análises de imagens microtomografadas realizada nesta pesquisa foi de uma areia seca classificada pelo Sistema Unificado de Classificação de Solos (SUCS) como areia pura mal graduada (SP), proveniente do Rio Tietê em São Paulo/SP, fornecida pelo Laboratório de Geotecnia da Universidade de Brasília – UnB, considerado um material de uso comum na construção civil no Distrito Federal (Figura 3.1) e objeto de estudo de Zubeldia (2013).



Figura 3.1. Grãos da areia pura mal graduada (Fonte: Zubeldia, 2013)

A areia pura mal graduada foi resultado do peneiramento que removeu as frações superiores a 1,19 mm (peneira 16) e inferiores a 0,25 mm (peneira 60), chegando-se à composição granulométrica apresentada na Figura 3.2.

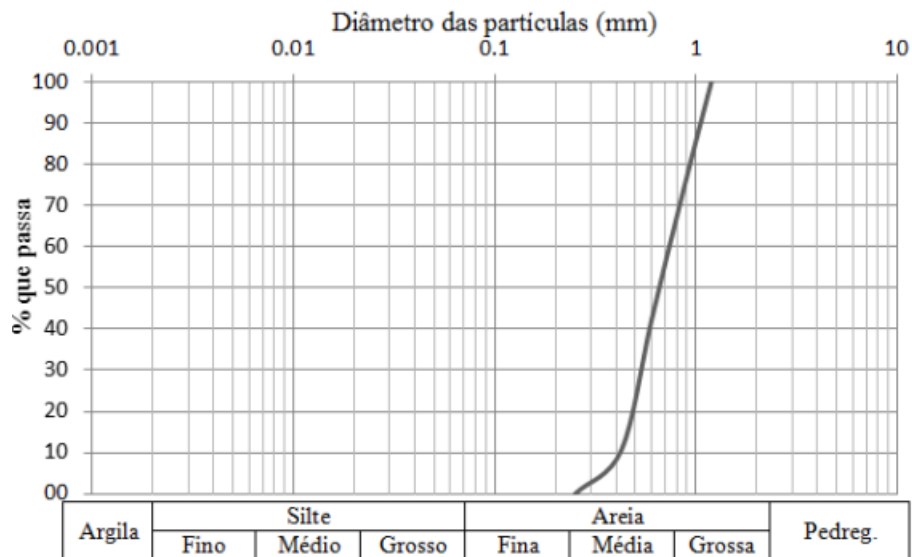


Figura 3.2. Distribuição granulométrica da areia pura mal graduada (Fonte: Zubeldia, 2013)

Para a aquisição das projeções e a reconstrução de imagens bidimensionais, a areia foi colocada dentro de um tubo de polipropileno de 1,5 ml, com paredes lisas e espessura uniforme, e escaneada utilizando-se microtomógrafo de raio-X de quarta geração, modelo SkyScan 1172, da Embrapa Instrumentação Agropecuária - São Carlos. As imagens foram adquiridas com uma

resolução de tamanho de pixel das imagens de $5,94 \mu\text{m}$, com dimensões de 2.000×2.000 pixels e em um total de 610 fatias de imagens bidimensionais. Ressalta-se que Borges *et al.* (2019) adotou o valor do tamanho de pixel das imagens de $11,32 \mu\text{m}$.

Como referência para os resultados que foram obtidos nesta pesquisa, o trabalho de Borges *et al.* (2019), que também analisou as imagens microtomografadas e geradas na pesquisa de Zubeldia (2013) utilizando um algoritmo desenvolvido no *Mathematica*, obteve a distribuição granulométrica (Figura 3.3), esfericidade média em torno de 79%, índice de vazios (e) de 0,76 e porosidade (n) de 43%.

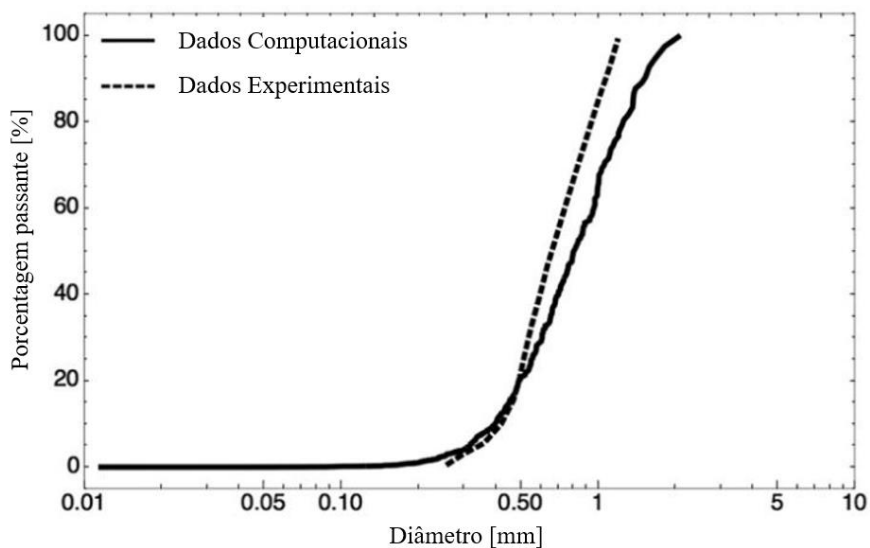


Figura 3.3. Distribuição granulométrica experimental e utilizando análise de imagens por Borges *et al.*, 2019 (Adaptado)

3.1.2. REJEITO DE MINÉRIO DE FERRO

Outro material objeto de estudo desta pesquisa foi o rejeito de minério de ferro (Figura 3.4) investigado por Carneiro (2020), sendo o mesmo utilizado na pesquisa desenvolvida por Sotomayor (2018), obtido dos diques de alteamento, compactado pela passagem dos equipamentos, sobre uma barragem de material de empréstimo.



Figura 3.4. Rejeito de minério de ferro

A distribuição granulométrica do rejeito de minério de ferro está apresentada na Figura 3.5.

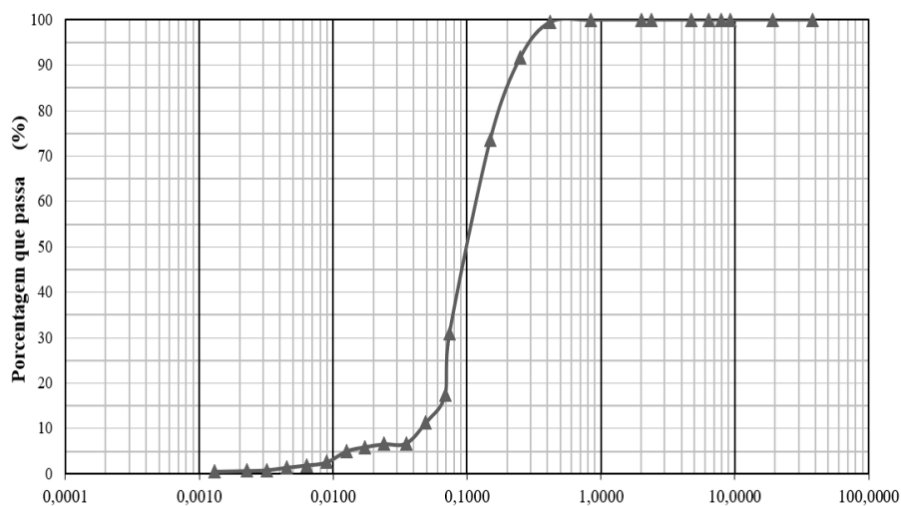


Figura 3.5. Curva granulométrica experimental do rejeito de minério de ferro (Fonte: Carneiro, 2020)

Além disso, Carneiro (2020) avaliou o comportamento do compósito de rejeito de minério de ferro estabilizado com polímero.

Carneiro (2020) realizou o ensaio de Difração de Raio-X (DRX) para verificar se a inserção da solução polimérica altera a composição dos minerais já existentes. A citada autora indicou que é possível constatar que o rejeito de minério de ferro estudado, independente da inserção do polímero, é composto basicamente por minerais de quartzo (SiO_2), goetita (FeO(OH)), hematita (Fe_2O_3), caulinita ($\text{Al}_2\text{Si}_2\text{O}_5(\text{OH})_4$), e outros minerais em proporções menores, conforme Figura 3.6.

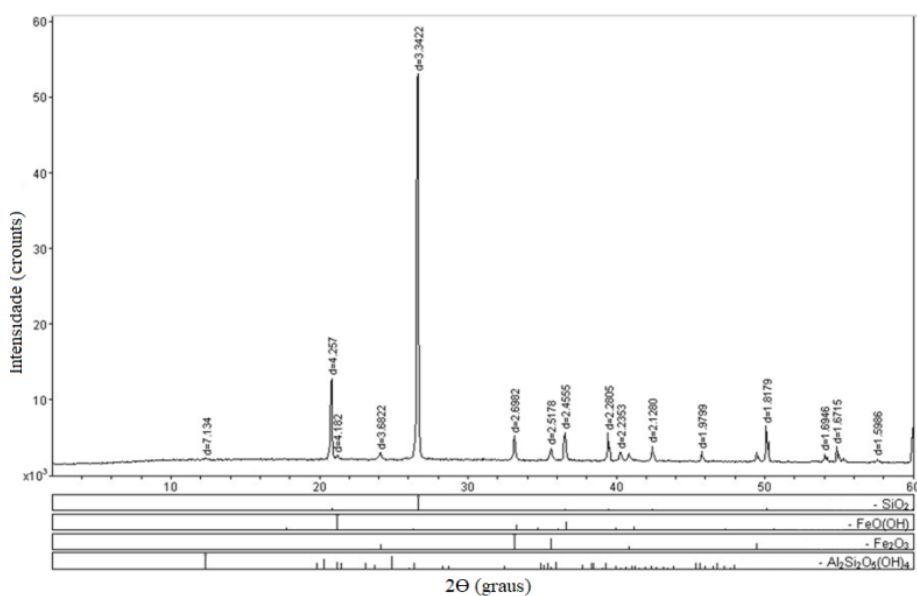


Figura 3.6. Difratoograma do rejeito de minério de ferro puro (Fonte: Carneiro, 2020)

Carneiro (2020) observa que no rejeito de minério de ferro puro, a sílica se destaca como o componente químico mais prevalente, sendo o quartzo o mineral característico mais abundante, originado das rochas que envolvem o minério de ferro. Em segundo lugar em termos de abundância, encontra-se o óxido férrico, com a hematita (Fe_2O_3) como seu mineral característico, indicando claramente a natureza do rejeito como sendo de minério de ferro.

Além disso, Carneiro (2020) conduziu sua pesquisa utilizando um compósito de rejeito de minério de ferro estabilizado com o polímero comercialmente conhecido como TERRAFIX 11, produzido pela empresa Waterflows. Este polímero é descrito como um copolímero acrílico estirenado, orgânico, sintetizado por meio de reações de polimerização aleatória em emulsão aquosa de caráter aniônico, apresentando densidade entre 0,98 e 1,04 g/cm^3 e total solubilidade em água.

Considerando a ausência de imagens de microtomografia computadorizada de raio-X para o rejeito de minério de ferro puro e para o compósito de rejeito de minério de ferro estabilizado com polímero, dois corpos de prova foram moldados para fins de escaneamento: um contendo apenas o rejeito de minério de ferro puro e outro contendo o rejeito de minério de ferro misturado com o polímero, na proporção de 1:2 do volume total (50% polímero e 50% água).

Para preparação das amostras, foram utilizadas seringas como moldes, onde o rejeito de minério de ferro puro foi inserido em uma seringa com diâmetro interno de aproximadamente 8,2 mm, enquanto o compósito de rejeito de minério de ferro estabilizado com polímero foi

colocado em uma seringa com diâmetro interno de aproximadamente 11,9 mm. A amostra de rejeito de minério puro foi previamente peneirada, separando-se para a moldagem o material retido na peneira de abertura de 0,25 mm (peneira 60). Já a amostra do compósito do rejeito de minério de ferro estabilizada com polímero foi feita considerando os mesmos parâmetros de moldagem de Carneiro (2020), sem qualquer seleção de material por peneiramento, com a dosagem da solução do polímero contendo 50% de polímero e 50% de água, ou seja, na proporção de 1:2, conforme descrito a seguir:

- I. Carneiro (2020) realizou a adição da solução em relação à massa do rejeito seco, na umidade ótima de 7,57%;
- II. Para a preparação da amostra desta pesquisa, foi adicionada uma solução de 22,71 g, compostas por 11,355 g de água e 11,355 g de polímero TERRAFIX11 aos 300,1 g de rejeito de minério de ferro disponíveis, totalizando 322,81 g de material para moldagem do corpo de prova para compactação;
- III. A compactação foi realizada em energia Proctor Normal em amostrador cilíndrico com de diâmetros médios de 50,1 mm e altura de 130 mm;
- IV. Para inserir a amostra dentro da seringa, cortou-se e descartou-se a ponta da seringa e posteriormente introduziu-se a seringa na amostra, como mostrado na Figura 3.7, passando pelo processo de cura durante 120 dias até ser enviada ao Laboratório de Fabricação Digital da Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio.



Figura 3.7. Preparação da amostra de compósito de rejeito de minério de ferro com polímero para microtomografia computadorizada de raio-x

As amostras de rejeito de minério de ferro puro (R) e de rejeito de minério de ferro estabilizada com polímero (RP) foram enviados ao laboratório mantidas nas próprias seringas fechadas com filme PVC (Figura 3.8) para escaneamento.



Figura 3.8. Acondicionamento das amostras de rejeito de minério de ferro - R (esquerda) e rejeito de minério de ferro estabilizado com polímero - RP (direita)

3.1.3. MICROTOMÓGRAFO COMPUTADORIZADO DE RAIOS-X

O Laboratório de Fabricação Digital da Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio possui o equipamento de microtomografia ZEISS Xradia 510 Versa (Figura 3.9) que apresenta um processo de duas etapas baseado em óptica de calibre de sincrotrão e possui capacidade multiescala que permite que sejam capturadas imagens da mesma amostra em uma ampla gama de ampliações, diferentemente dos tradicionais sistemas de tomografia computadorizada que dependem apenas de uma etapa de ampliação geométrica, dificultando a obtenção de alta resolução para amostras maiores devido às distâncias de trabalho mais longas necessárias (Carl Zeiss Microscopy, 2023). Em outras palavras, supera-se a limitação geométrica pelo uso de lentes, magnificando a resolução espacial quando atingido o limite máximo de afastamento do detector em relação à amostra e o limite máximo de proximidade da fonte de raio-x à amostra.



Figura 3.9. Microtomógrafo computadorizado de raio-X modelo ZEISS Xradia 510 Versa

As principais características técnicas do microtomógrafo ZEISS Xradia 510 Versa incluem: resolução espacial que atinge até $0,7 \mu\text{m}$, variedade de lentes objetivas que oferecem aumentos de 0,4x, 4x, 20x e 40x, operação em uma faixa de alta voltagem ajustável de 30 a 160 kV, potência máxima de 10 W e capacidade para acomodar amostras com diâmetro de até 300 mm e peso de até 15 kg.

O equipamento é composto por um sistema emissor de raio-x, suporte para a amostra e sistema detector de raios-x (Figura 3.10). O sistema emissor é o responsável por emitir feixes de elétrons que atingem uma placa de cobre, ou “alvo”, que gera a emissão de radiação na faixa de raios-x e que são colimados em feixe cônico. O suporte para a amostra é composto por uma mesa giratória para permitir a aquisição de imagens em diferentes ângulos (Figura 3.11). E o sistema detector é composto por lentes que possuem cintilador, o qual tem a função de transformar o raio-x em fótons de luz visíveis para a captura de fotos pela câmera CCD, responsável pela conversão de luz em carga elétrica e transformada em um sinal digital, que é lido para reconstruir a imagem.



Figura 3.10. Componentes do Microtomógrafo ZEISS Xradia 510 Versa

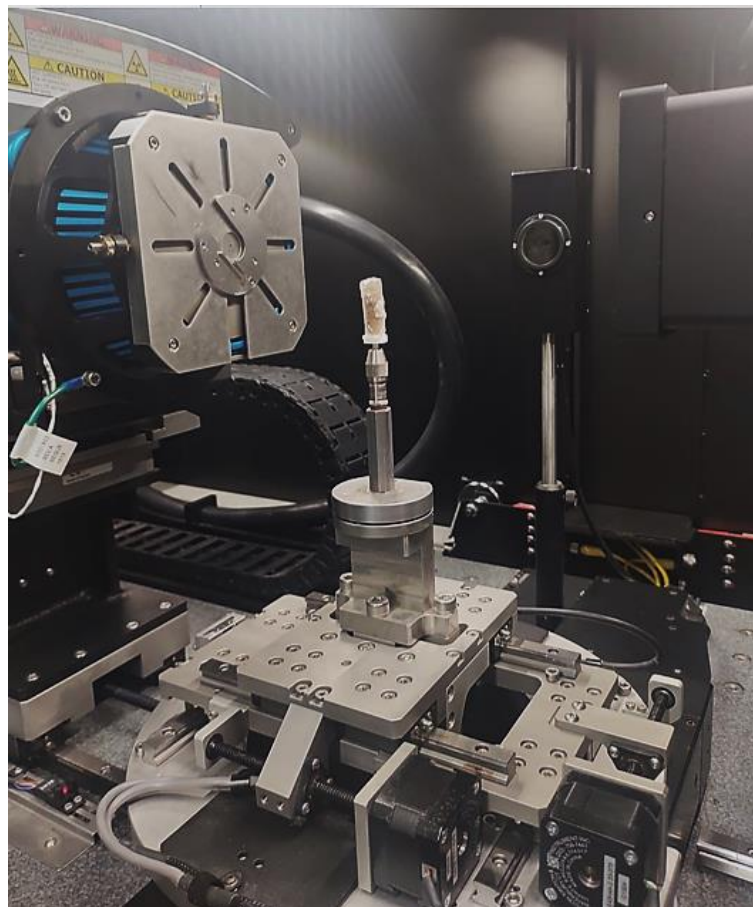


Figura 3.11. Detalhe da amostra sobre o suporte na mesa giratória

Em relação à definição da resolução espacial, os parâmetros de distância da fonte de raio-x até a amostra, distância do detector até a amostra, uso de lentes e escolha do *binning* (processo de combinação de vários pixels em apenas um pixel) são fundamentais para estabelecer a definição da resolução da amostra. Com os parâmetros utilizados para as amostras desta pesquisa, apresentados na Figura 3.12, foram adquiridas 1.016 fatias (*slices*) de imagens 2D para o rejeito de minério de ferro, com dimensões de 1.000 x 1.024 pixels, e 1.015 fatias (*slices*) de imagens 2D para o rejeito de minério de ferro estabilizado com polímero, com dimensões de 1.003 x 1.024 pixels. O tempo médio para a aquisição das imagens utilizadas neste trabalho foi de 1 hora e 30 minutos para a amostra de rejeito de minério de ferro e de 1 hora e 45 minutos para o composto de rejeito de minério de ferro estabilizado com polímero.

Imaging Mode:	Tomography	Imaging Mode:	Tomography
Camera Binning:	2	Camera Binning:	2
File Data Type:	ushort	File Data Type:	ushort
Display Data Type:	float	Display Data Type:	float
File Size:	3269.30 MB	File Size:	3269.30 MB
Source Filter:	LE3	Source Filter:	LE3
Source Setting:	70.17426 kV 85.37247 μ A	Source Setting:	80.14771 kV 87.3806 μ A
Source-RA Distance:	-20.0143 mm	Source-RA Distance:	-20.0236 mm
Det-RA Distance:	146.0069 mm	Det-RA Distance:	146.0069 mm
Pixel Size:	8.3129 μ m	Pixel Size:	8.3163 μ m
Optical Magnification:	0.3916 : 0.4X	Optical Magnification:	0.3916 : 0.4X
Exposure Time:	2.0000 sec	Exposure Time:	2.2000 sec
Sample Theta:	-179.996 deg	Sample Theta:	-179.995 deg
Date and Time:	10/20/2023 11:27:06.718	Date and Time:	10/20/2023 14:28:13.828

Figura 3.12. Parâmetros de configuração para as amostras de (a) rejeito de minério de ferro e (b) rejeito de minério de ferro estabilizado com polímero

Comparativamente ao microtomógrafo de bancada utilizado no escaneamento da amostra de areia no trabalho de Zubeldia (2013), o SkyScan 1172, a principal diferença entre os equipamentos está na possibilidade de uso de lentes no equipamento ZEISS Xradia 510 Versa, que magnifica a resolução espacial quando atingido o limite máximo das distâncias geométricas, não existindo essa possibilidade no microtomógrafo de bancada. O nível operacional de potência das fontes de raios x também possuem faixas diferenciadas: no microtomógrafo de banca há possibilidade de variação de 20 a 100 kV, enquanto para o ZEISS Xradia 510 Versa, de 30 a 160 kV. Além disso, há a diferença de limitação de tamanho do objeto a ser escaneado: para o microtomógrafo SkyScan 1172 há o limite máximo para objetos com 68 mm de diâmetro e para o ZEISS Xradia 510 Versa, 300 mm.

3.2. MÉTODOS

A partir das imagens 2D tomografadas de cada uma das amostras (areia, rejeito de minério de ferro e de rejeito de minério de ferro estabilizado com polímero), iniciaram-se as etapas de pré-processamento e de análise de imagens, buscando-se estabelecer uma proposta de algoritmo e de desenvolvimento de script em linguagem Python para obtenção de parâmetros morfológicos em escala de grãos, como granulometria, porosidade e esfericidade dos grãos.

O ambiente de desenvolvimento utilizado foi o Notebook Jupyter, que permite a criação e compartilhamento de documentos que contenham código executável, visualizações das imagens a cada processamento e textos explicativos, tudo em um único documento interativo.

O processamento digital das imagens foi realizado utilizando-se uma estação de trabalho com as seguintes configurações: processador Intel® Core™ i7-7500U 2.7 GHz com 16 GB de RAM e uma placa de vídeo NVIDIA® GeForce® 940MX 4 GB.

3.2.1. ESFERICIDADE

Considerando as diversas metodologias descritas no Capítulo 2 para se calcular a esfericidade 2D dos grãos, a esfericidade do perímetro da Eq. (5, verificada por Rorato *et al.* (2019) como o descritor 2D que parece estar mais correlacionado com a esfericidade 3D, foi uma das alternativas escolhida para a análise das imagens de areia e do rejeito de minério de ferro, além da esfericidade de Rittenhouse (1943). No entanto, é importante ressaltar que a análise da esfericidade não foi aplicada ao compósito de rejeito de minério de ferro com polímero, devido à limitação de resolução, que impossibilitou a identificação dos contornos das partículas mais finas, possivelmente devido aos conglomerados formados pelo polímero com as partículas mais finas.

3.2.1.1. ESFERICIDADE DA AREIA

Os códigos utilizados para inferir o parâmetro de esfericidade das 610 imagens 2D da areia estão apresentados no APÊNDICE A e a sua esquematização simplificada é apresentada na Figura 3.13.

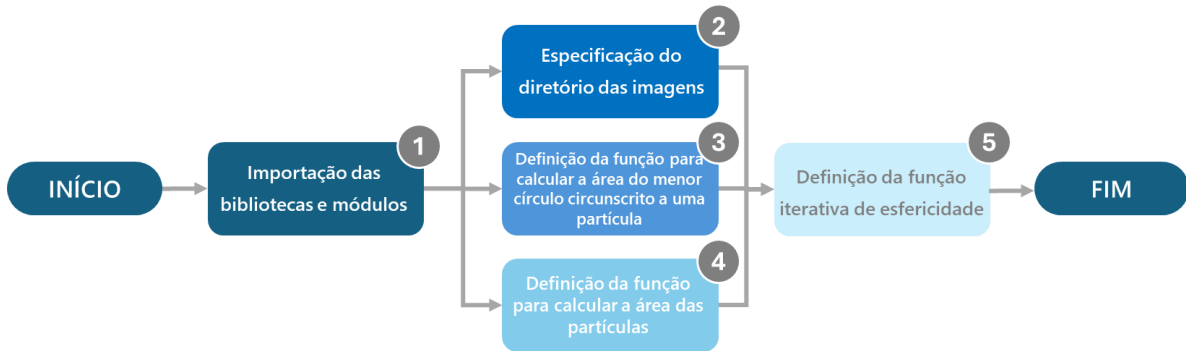


Figura 3.13. Esquematização simplificada do algoritmo de esfericidade para a areia

As descrições das etapas do esquema simplificado são:

1. Importação das bibliotecas e módulos necessários para serem utilizados ao longo do código para processamento e análise de imagens, bem como para manipulação de dados e visualização de resultados:
 - a. **“os”**: módulo que fornece funções para interagir com o sistema operacional, como manipular caminhos de arquivos e pastas;
 - b. **“cv2”**: biblioteca OpenCV, utilizada para processamento de imagens e visão computacional, como leitura, processamento e manipulação de imagens;
 - c. **“numpy as np”**: biblioteca Numpy, utilizada para suportar arrays e matrizes multidimensionais, juntamente com uma coleção de funções matemáticas para operar nesses arrays;
 - d. **“matplotlib.pyplot as plt”**: biblioteca Matplotlib, utilizada para criação de visualizações estáticas, como gráficos e plotagens;
 - e. **“PIL.Image”**: biblioteca PIL (Python Imaging Library), utilizada para processamento de imagens em Python. O módulo Image fornece classes e métodos para abrir, manipular e salvar imagens em vários formatos;
 - f. **“skimage.feature”**: parte do pacote scikit-image (versão 0.18.3), que é uma coleção de algoritmos para processamento de imagens. O módulo feature fornece funções para extração de características de imagens, como detecção de bordas e picos;
 - g. **“skimage.segmentation”**: Outro módulo do scikit-image que fornece algoritmos de segmentação de imagens, como o algoritmo de *Watershed*;
 - h. **“scipy.ndimage as ndi”**: parte do pacote SciPy, que é uma biblioteca para computação científica em Python. O módulo ndimage fornece

funções para processamento de imagens multidimensionais, como filtros e transformações;

- i. **“pandas as pd”**: biblioteca Pandas, utilizada para manipulação e análise de dados em Python, comumente usada para trabalhar com dados tabulares, como criar, manipular e analisar DataFrames; e
 - j. **“time”**: módulo que fornece várias funções relacionadas ao tempo, como medir a duração de operações.
2. Especificação do diretório que contém as imagens;
 3. Definição da função para calcular a área do menor círculo circunscrito a uma partícula em uma imagem binarizada e facilitar sua reutilização em outras partes do código;
 4. Definição da função para calcular a área das partículas individualmente, o perímetro da partícula e a área do círculo circunscrito à partícula, adicionando esses resultados a uma lista, e para facilitar sua reutilização em outras partes do código;
 5. Definição da função iterativa para cada arquivo de imagem no diretório, responsável pelo cálculo da esfericidade das partículas presentes nas imagens 2D:
 - a. Parâmetro de entrada será o caminho para a pasta que contém as imagens;
 - b. Inicializa-se uma lista vazia para armazenar os resultados;
 - c. O tratamento das imagens (Figura 3.14) são realizados iterativamente, cada imagem é aberta e convertida em uma matriz *numpy* para realizar os tratamentos das imagens: suas cores são invertidas e se aplica filtro Gaussiano; a imagem é binarizada pelo método de Otsu, que encontra limiar automaticamente a partir do histograma da imagem (Figura 3.15), e é calculada a transformada de distância, localizando-se seus máximos locais e rotulando esses máximos; segmenta-se utilizando *Watershed* para individualização das partículas;
 - d. Utiliza-se a função citada no item 4 para se ter o perímetro da partícula e calcula-se o perímetro equivalente do círculo de mesma área para encontrar a esfericidade do perímetro, conforme Sandeep *et al.* (2023);
 - e. Adiciona-se o resultado da esfericidade do perímetro à lista;
 - f. Converte-se a lista em DataFrame para exportar os dados em Excel.

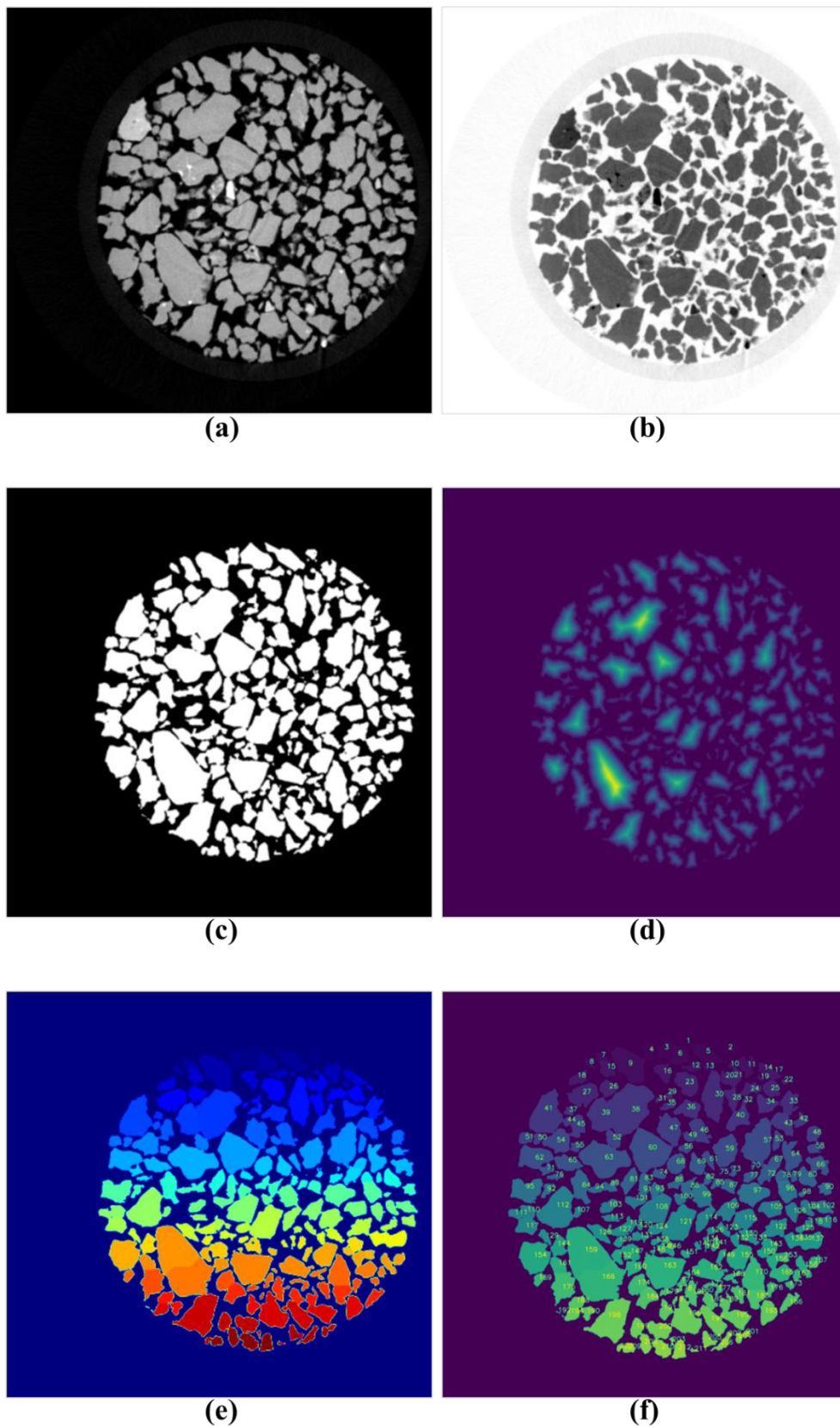


Figura 3.14. Pré-processamento e processamento das imagens 2D da areia: (a) Imagem original; (b) Inversão de cores e aplicação do filtro Gaussiano; (c) Binarização pelo método de

Otsu; (d) Transformação de distância euclidiana; (e) Separação das partículas com Watershed; e (f) Singularização atribuindo rótulos únicos para cada partícula

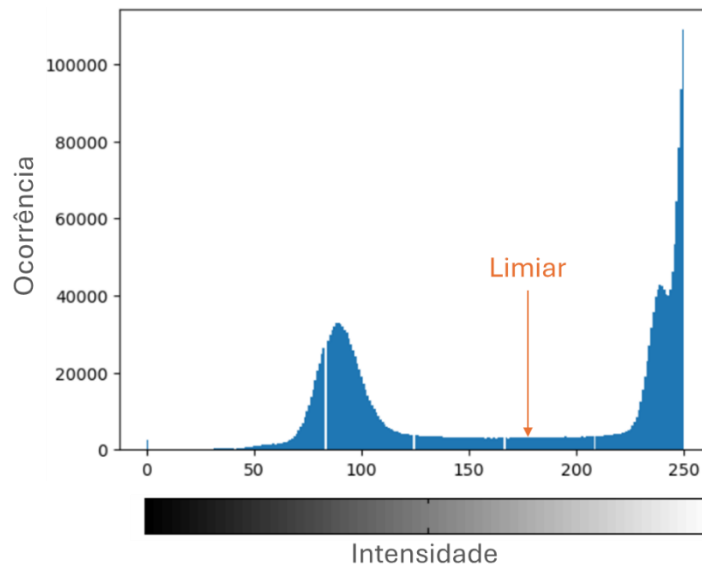


Figura 3.15. Histograma de nível de intensidade de uma fatia (*slice*) da areia

O algoritmo organiza os dados em Excel em colunas de “Nome do Arquivo” e “Número da Partícula”, que identificam a singularidade das partículas; de “Área” e “Área Circunscrita”, que representam respectivamente a área da partícula em pixels e a área do círculo que circunscribe a partícula também em pixels, utilizados no cálculo da esfericidade de Rittenhouse (1943); de “Raio Equivalente”, “Perímetro Equivalente” e “Perímetro da Partícula”, que são utilizados para o cálculo da coluna “Esfericidade do Perímetro”, sendo a coluna de “Raio Equivalente” uma coluna auxiliar para conferências.

3.2.1.2. ESFERICIDADE DO REJEITO DE MINÉRIO DE FERRO

Os códigos utilizados para inferir o parâmetro de esfericidade das 1.015 imagens 2D do rejeito de minério de ferro estão apresentados no APÊNDICE B e a sua esquematização simplificada é apresentada na Figura 3.16.

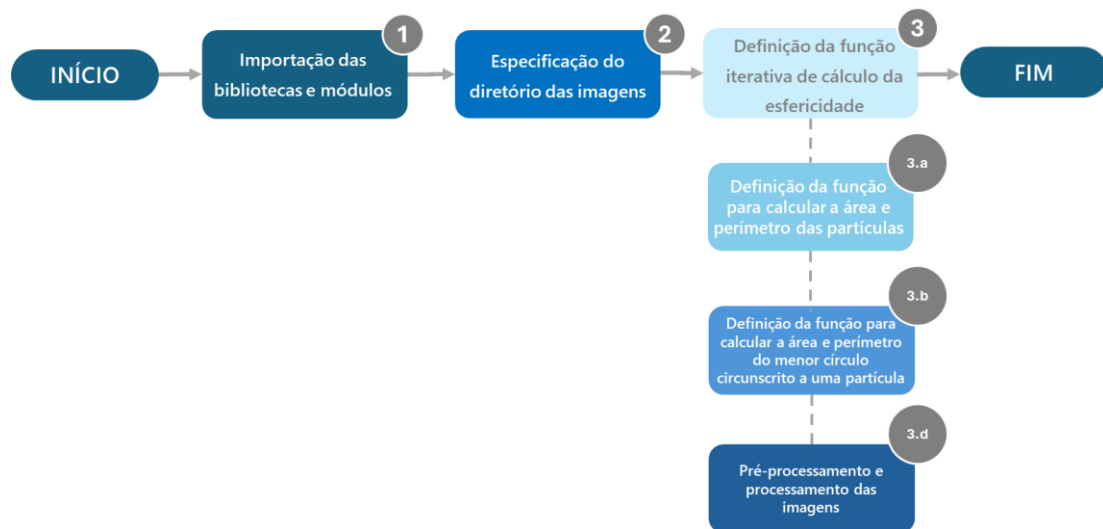


Figura 3.16. Esquematização simplificada do algoritmo de esfericidade para o rejeito de minério de ferro

As descrições das etapas do esquema simplificado são:

1. Importação das bibliotecas e módulos necessários para o processamento de imagem, manipulação de dados, visualização de resultados e medição de tempo, sendo as mesmas indicadas no código das areias no item 3.2.1.1;
2. Especificação do diretório que contém as imagens;
3. Definição da função iterativa de esfericidade para cada arquivo de imagem 2D no diretório;
 - a. Definição da função para calcular a área e o perímetro das partículas individualmente, adicionando esses resultados a uma lista;
 - b. Definição da função para calcular a área e perímetro do menor círculo circunscrito a uma partícula em uma imagem binarizada;
 - c. Inicialização de uma lista de dados para o DataFrame para armazenar os dados que serão usados;
 - d. Iteração sobre os arquivos de imagem no diretório para realização do pré-processamento e processamento das imagens (Figura 3.17 e Figura 3.18):
 - i. Inverte-se as cores da imagem;
 - ii. Aplica-se filtro de suavização Gaussiano;
 - iii. Arbitra-se um valor de limiar manualmente para operação de binarização da imagem;
 - iv. Inverte-se novamente as cores das imagens para tornar os grãos brancos (255) e o fundo preto (0); e

- v. Calcula-se a transformada de distância, encontra-se os picos locais na matriz de distância, rotula-se os picos locais e aplica-se a segmentação de *Watershed* para separar as partículas próximas.
- e. Cálculo de área projetada na imagem 2D de cada partícula, da área do menor círculo circunscrito à partícula, do raio equivalente referente ao círculo de mesma área da partícula e do perímetro equivalente referente ao círculo de mesma área da partícula;
- f. Criação de DataFrame com os dados encontrados;
- g. Criação das colunas “Esfericidade” e “Esfericidade do Perímetro” e cálculo dos seus valores; e
- h. Exportação do DataFrame para um arquivo Excel.

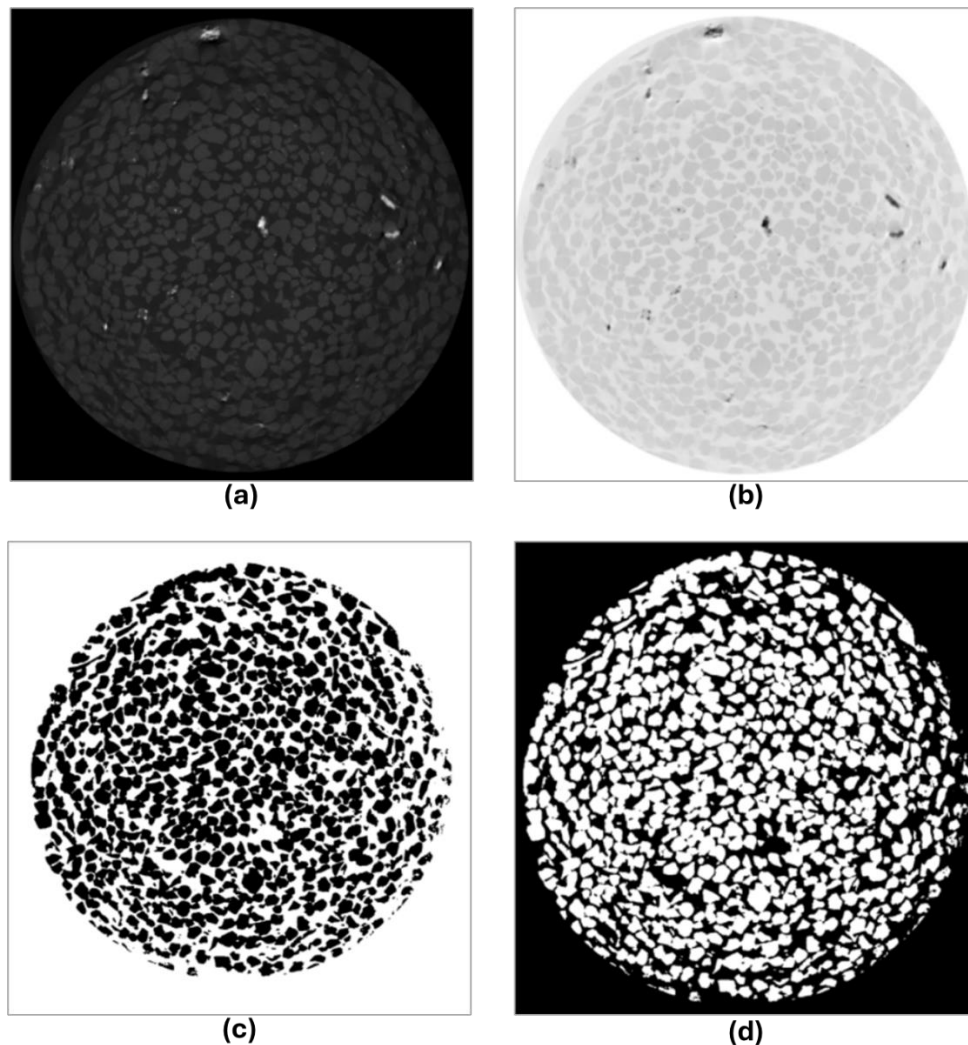


Figura 3.17. Pré-processamento das imagens 2D do rejeito de minério de ferro: (a) Imagem original; (b) Inversão de cores e aplicação do filtro Gaussiano; (c) Binarização com escolha manual de limiar; (d) Inversão de cores novamente

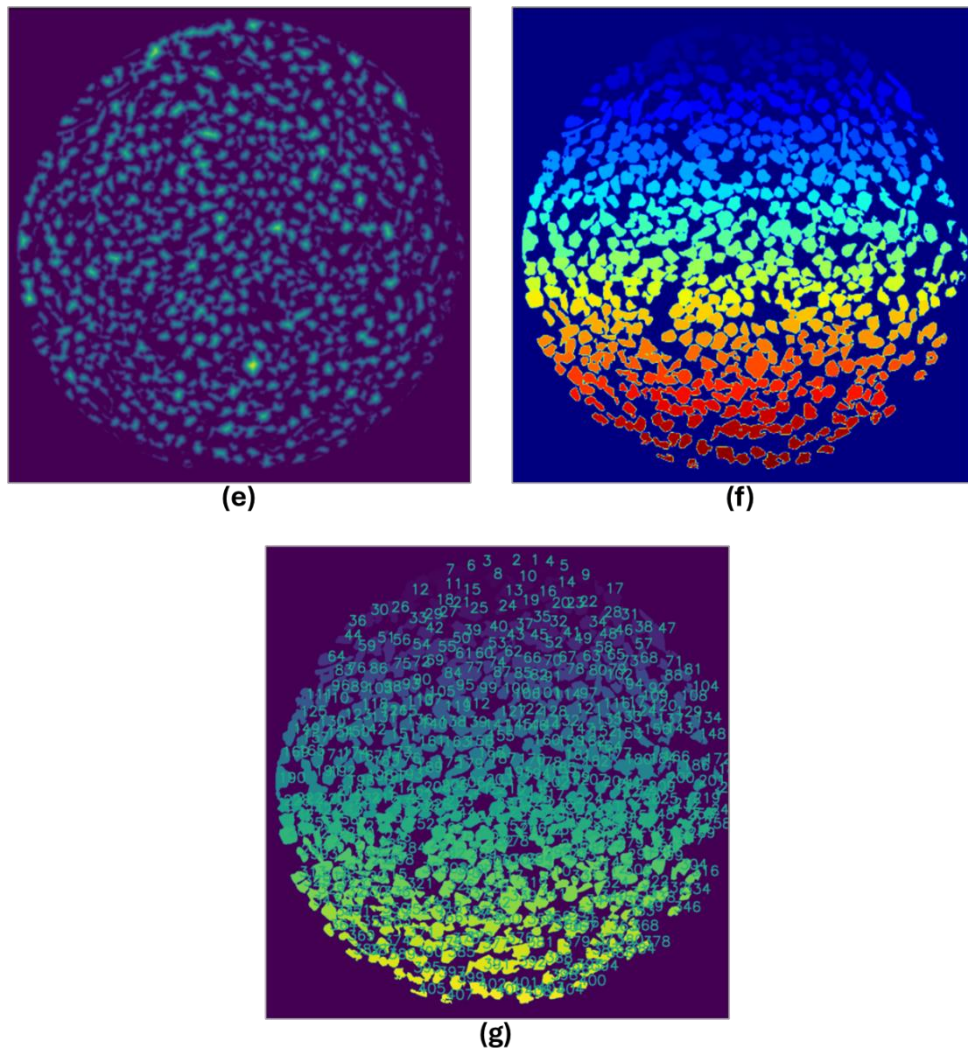


Figura 3.18. Processamento das imagens 2D do rejeito de minério de ferro: (e) Transformação de distância euclidiana; (f) Separação das partículas com Watershed; e (g) Singularização atribuindo rótulos únicos para cada partícula

Destaca-se que as imagens originais, como exemplificado na Figura 3.17.(a), foram modificadas mediante a aplicação de uma máscara preta sobre as imagens microtomografadas com o intuito de remover o fundo das imagens, na tentativa de se mitigar a influência do ruído. No entanto, embora efetivamente tenha ocultado o ruído de fundo, a máscara inadvertidamente suprimiu parte dos grãos localizados no canto inferior direito, conforme ilustrado na Figura 3.17.(b).

Outro aspecto significativo foi a necessidade de selecionar manualmente o limiar para a binarização: devido ao contraste próximo entre o fundo e os grãos, e considerando a aplicação da máscara, o histograma de intensidade exibiu um comportamento trimodal concentrado na faixa de intensidade mais clara (Figura 3.19), o que impactou consideravelmente a aplicação de

um limiar automático, como o método de *Otsu*, que depende da clara definição das regiões bimodais, tendo sido adotada uma escolha manual do limiar para a binarização no valor de 224.

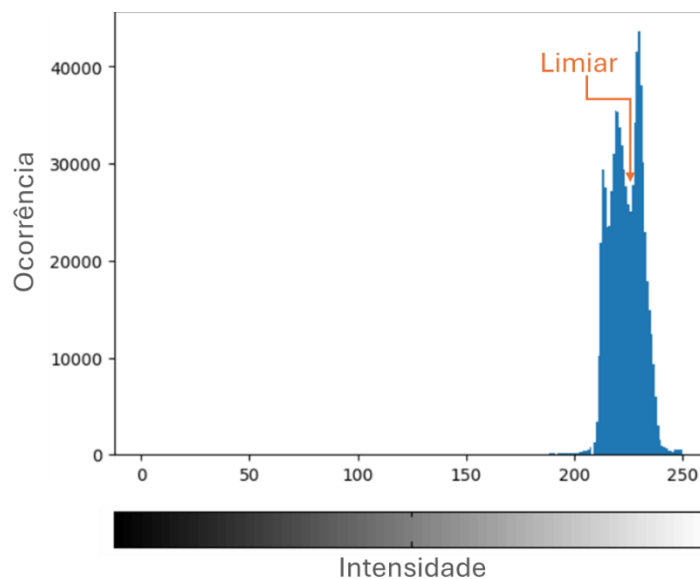


Figura 3.19. Histograma de nível de intensidade de uma fatia (*slice*) do rejeito de minério de ferro

O algoritmo organiza os dados em Excel em colunas de “Nome do Arquivo” e “Rótulo Único”, que identificam a singularidade das partículas; de “Área da partícula” e “Área do Círculo Circunscrito”, que representam respectivamente a área da partícula em pixels e a área do círculo que circunscribe a partícula também em pixels, utilizados no cálculo da esfericidade de Rittenhouse (1943); de “Perímetro Circunscrito”, utilizado como uma coluna auxiliar para conferência; de “Perímetro Equivalente” e “Perímetro da Partícula”, que são utilizados para o cálculo da coluna “Esfericidade do Perímetro”.

3.2.2. GRANULOMETRIA

Considerando as metodologias descritas no Capítulo 2 para se obter a distribuição granulométrica dos grãos a partir de imagens 2D, a metodologia utilizada por Borges *et al.* (2019) será a adotada nesta seção para avaliar a granulometria da areia e do rejeito de minério de ferro. É importante ressaltar que a análise da granulometria não foi aplicada ao compósito de rejeito de minério de ferro com polímero, devido à limitação de resolução, que impossibilitou a identificação dos contornos das partículas mais finas, possivelmente devido aos conglomerados formados pelo polímero com as partículas mais finas.

3.2.2.1. GRANULOMETRIA DA AREIA

Os códigos utilizados para inferir a distribuição granulométrica das 610 imagens 2D da areia estão apresentados no APÊNDICE C e a sua esquematização simplificada é apresentada na Figura 3.20.

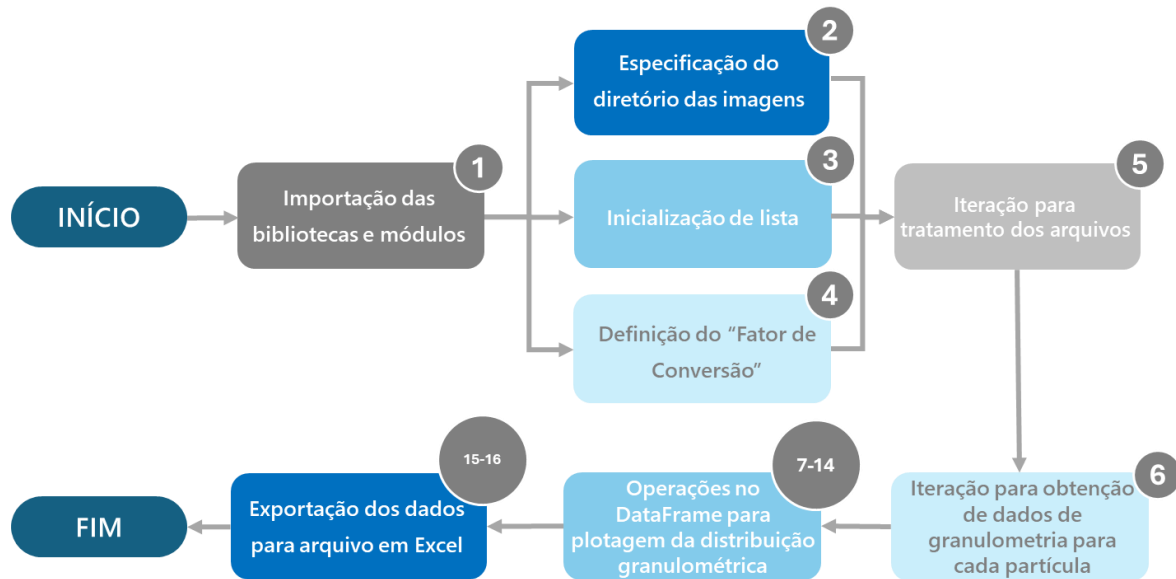


Figura 3.20. Esquematização simplificada do algoritmo de granulometria para a areia

As descrições das etapas do esquema simplificado são:

1. Importação das bibliotecas e módulos necessários para o processamento de imagem, manipulação de dados e visualização de resultados, sendo as mesmas indicadas no código das areias no item 3.2.1.1, com exceção do módulo “time”, que pode ser adicionado a critério do usuário;
2. Especificação do diretório que contém as imagens;
3. Inicialização uma lista vazia para armazenar os resultados;
4. Definição do “Fator de Conversão” para transformar as medidas de pixels para milímetros;
5. Iteração sobre os arquivos no diretório indicado para tratamento das imagens:
 - a. Cada imagem é aberta e convertida em uma matriz *numpy* para realizar os tratamentos das imagens;
 - b. Invertem-se as cores;
 - c. Aplica-se filtro de suavização Gaussiano;

- d. Binariza-se a imagem pelo limiar automático do método de Otsu;
 - e. Calcula-se a matriz de distância utilizando a transformação euclidiana;
 - f. Localizam-se os máximos locais na matriz de distância;
 - g. Os máximos locais são rotulados;
 - h. Aplica-se o algoritmo Watershed e obtém-se a individualização das partículas;
6. Para cada partícula individualizada encontrada no pré-processamento e processamento anterior, a inferência da granulometria é iniciada:
 - a. Calculam-se os momentos de cada partícula e seu respectivo centróide;
 - b. Aplica-se o menor retângulo delimitador rotacionado a cada uma das partículas e encontram-se a sua maior e menor aresta;
 - c. Convertem-se os tamanhos das arestas de pixel para milímetros;
 - d. Calcula-se a área de cada partícula em pixel e converte-se em milímetros quadrados;
 - e. Adicionam-se os dados à lista;
 - f. Opcionalmente e para fins de conferência visual, disponibiliza-se a plotagem do desenho do centróide de cada partícula, juntamente com o seu respectivo número de individualização, número de coordenada do centróide e desenho do menor retângulo delimitador rotacionada da partícula, recomendando-se que seja utilizado para uma pré-avaliação com poucas imagens e que seja colocado como comentário quando aplicado a muitas imagens;
 7. Criação de um DataFrame com os dados;
 8. Ordenação do DataFrame pela coluna que contém os valores da “menor aresta (mm)” para facilitar a plotagem da distribuição granulométrica;
 9. Cálculo da soma acumulada da coluna “área (mm²)”;
 10. Determinação do maior valor da coluna “área acumulada (mm²)”;
 11. Cálculo da porcentagem da área acumulada;
 12. Exibição do DataFrame para conferência;
 13. Ordenação dos dados do DataFrame pela coluna com os valores de “menor aresta” de cada partícula;
 14. Plotagem do gráfico para conferência visual;
 15. Especificação do caminho para salvar em um arquivo Excel; e
 16. Exportação do DataFrame para um arquivo Excel no caminho indicado.

No caso da granulometria, e diferentemente do parâmetro de esfericidade, que é uma medida adimensional, o diâmetro das partículas é expresso em milímetros, sendo imputado o valor de 5,94 μm para o tamanho do pixel da imagem na etapa 4, conforme configurado na microtomografia.

Seguindo-se a aplicação de Borges et al. (2019), a curva granulométrica dos solos é traçada a partir da distribuição cumulativa da massa das partículas (no eixo y) em relação aos seus diâmetros (no eixo x), sendo necessário obter esses parâmetros para cada grão. Para isso, considera-se que a massa equivalente de cada grão (m_{gi}) é representada genericamente pela multiplicação de seu volume e sua densidade Eq. (8).

$$m_{gi} = (A_{gi}h)\rho_{gi} \quad (8)$$

onde:

m_{gi} : é a massa do grão de um grão i ;

A_{gi} : é a área de um grão i ;

h : é a distância entre as fatias; e

ρ_{gi} : é a massa específica de um grão i .

Considerando que a areia estudada é um único material, a massa específica é constante para todas as partículas, assim como a altura referente à distância entre as fatias escaneadas. Isso faz com que seja simplificado o parâmetro a ser plotado no gráfico de granulometria no eixo das ordenadas, tornando a relação entre a massa dos grãos acumulada e a massa total dos grãos na amostra correspondente à área acumulada dos grãos e área total dos grãos na amostra, definindo-se os valores a serem plotados no eixo das ordenadas Eq. (9).

$$\%acumulada(m_{gi}) = \frac{\sum(A_{gi}h)\rho_{gi}}{(A_t h)\rho_g} = \frac{\sum A_{gi}}{A_t} \quad (9)$$

onde:

A_t : é a área total dos grãos na amostra; e

ρ_g : é a massa específica média dos grãos da amostra.

Dessa forma, a partir do cálculo das áreas em milímetros quadrados de cada partícula (etapa 6.a), encontra-se a área total dos grãos na amostra (etapa 10), utilizada como valor de referência (denominador) para as porcentagens do eixo y.

Assim, para plotagem do gráfico com a curva granulométrica, estruturaram-se os dados em uma tabela, singularizou-se cada partícula com suas respectivas fatias/arquivo de imagem, dimensão de menor aresta em milímetros e área da partícula em milímetros quadrados. Classificaram-se em ordem crescente os valores do parâmetro “menor aresta” e acumularam-se os valores de área das partículas em uma nova coluna de dados, chamada no código de “Área Acumulada em %”.

A partir da etapa 7, o algoritmo organiza os dados em Excel em colunas de “Nome do Arquivo” e “Número da Partícula”, que identificam a singularidade das partículas; de “Menor Aresta (mm)”, que traz os valores da menor aresta do retângulo delimitador rotacionado (Figura 3.21); de “Área (mm²)”, que traz os valores de área de cada partícula em milímetros quadrados; de “Área Acumulada (mm²)”, que traz os valores acumulados da coluna “Área (mm²)” após ordenar os dados da coluna “Menor Aresta (mm)”; e de “Área Acumulada em %”, que calcula a relação entre o valor da coluna “Área Acumulada (mm²)” e área total dos grãos na amostra (etapa 10).

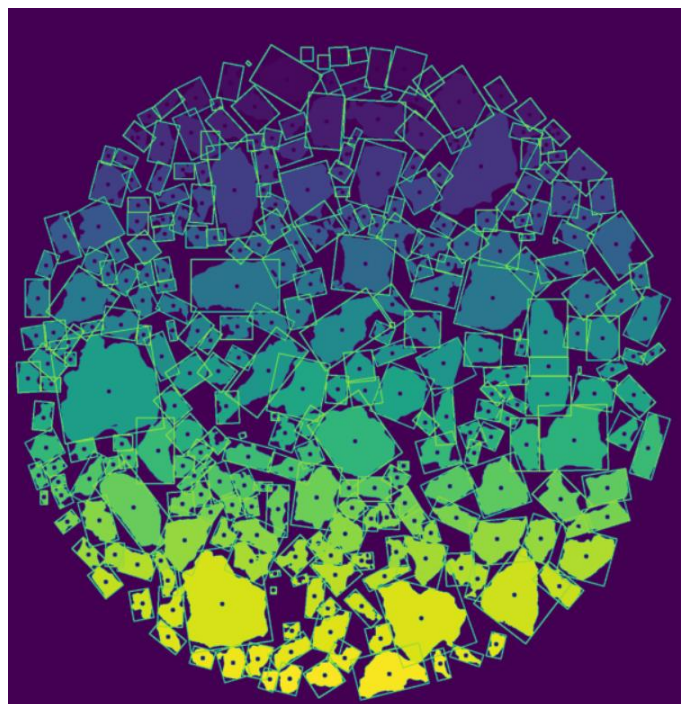


Figura 3.21. Aplicação do menor retângulo delimitador rotacionado a cada uma das partículas da amostra de areia

3.2.2.2. GRANULOMETRIA DO REJEITO DE MINÉRIO DE FERRO

Os códigos utilizados para inferir o parâmetro de esfericidade das 1.015 imagens 2D do rejeito de minério de ferro estão apresentados no APÊNDICE D e a sua esquematização simplificada é representada também pela Figura 3.20, tendo-se as mesmas etapas da descrição da granulometria da areia, porém com a diferença de escolha do limiar manual para o caso da granulometria do rejeito de minério de ferro:

1. Importação das bibliotecas e módulos necessários para o processamento de imagem, manipulação de dados, visualização de resultados e medição de tempo, sendo as mesmas indicadas no código das areias no item 3.2.1.1;
2. Especificação do diretório que contém as imagens;
3. Inicialização uma lista vazia para armazenar os resultados;
4. Definição do “Fator de Conversão” para transformar as medidas de pixels para milímetros;
5. Iteração sobre os arquivos no diretório indicado para tratamento das imagens:
 - a. Cada imagem é aberta e convertida em uma matriz *numpy* para realizar os tratamentos das imagens;
 - b. Invertem-se as cores;
 - c. Aplica-se filtro de suavização Gaussiano;
 - d. Arbitra-se um valor de limiar manualmente para operação de binarização da imagem;
 - e. Inverte-se novamente as cores das imagens para tornar os grãos brancos (255) e o fundo preto (0);
 - f. Calcula-se a matriz de distância utilizando a transformação euclidiana;
 - g. Localizam-se os máximos locais na matriz de distância;
 - h. Os máximos locais são rotulados;
 - i. Aplica-se o algoritmo Watershed e obtém-se a individualização das partículas;
6. Para cada partícula individualizada encontrada no pré-processamento e processamento anterior, a inferência da granulometria é iniciada:
 - a. Calculam-se os momentos de cada partícula e seu respectivo centróide;
 - b. Aplica-se o menor retângulo delimitador rotacionado a cada uma das partículas (Figura 3.22) e encontram-se a sua maior e menor aresta;

- c. Convertem-se os tamanhos das arestas de pixel para milímetros;
 - d. Calcula-se a área de cada partícula em pixel e converte-se em milímetros quadrados;
 - e. Adicionam-se os dados à lista;
 - f. Opcionalmente e para fins de conferência visual, disponibiliza-se a plotagem do desenho do centróide de cada partícula, juntamente com o seu respectivo número de individualização, número de coordenada do centróide e desenho do menor retângulo delimitador rotacionada da partícula, recomendando-se que seja utilizado para uma pré-avaliação com poucas imagens e que seja colocado como comentário quando aplicado a muitas imagens;
7. Criação de um *DataFrame* com os dados;
 8. Ordenação do *DataFrame* pela coluna que contém os valores da “Menor Aresta (mm)” para facilitar a plotagem da distribuição granulométrica;
 9. Cálculo da soma acumulada da coluna “Área (mm²)”;
 10. Determinação do maior valor da coluna “Área Acumulada (mm²)”;
 11. Cálculo da porcentagem da área acumulada;
 12. Exibição do *DataFrame* para conferência (opcional);
 13. Ordenação dos dados do *DataFrame* pela coluna com os valores de “menor aresta” de cada partícula;
 14. Plotagem do gráfico para conferência visual;
 15. Especificação do caminho para salvar em um arquivo Excel; e
 16. Exportação do *DataFrame* para um arquivo Excel no caminho indicado.

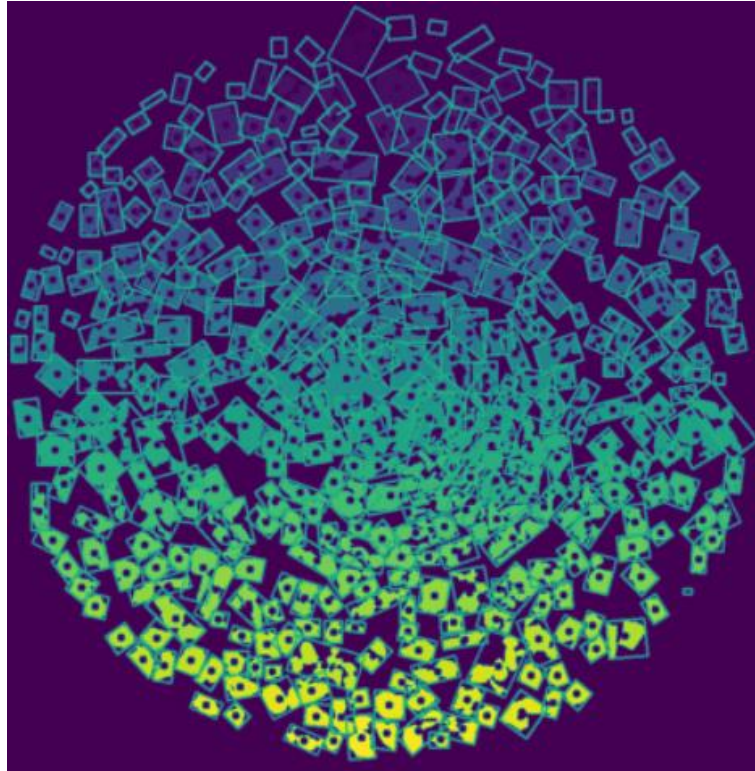


Figura 3.22. Aplicação do menor retângulo delimitador rotacionado a cada uma das partículas da amostra de rejeito de minério de ferro

3.2.3. ÍNDICE DE VAZIOS E POROSIDADE

Considerando as metodologias descritas no Capítulo 2 para se estudar os parâmetros de índice de vazios e porosidade dos materiais, a metodologia utilizada por Borges *et al.* (2019) será a adotada nesta seção para avaliar esses parâmetros para a areia, o rejeito de minério de ferro e o compósito de rejeito de minério de ferro com polímero.

3.2.3.1. ÍNDICE DE VAZIOS E POROSIDADE DA AREIA

Os códigos utilizados para inferir o índice de vazios e a porosidade das 610 imagens 2D da areia estão apresentados no APÊNDICE E e a sua esquematização simplificada é apresentada na Figura 3.23.



Figura 3.23. Esquematização simplificada do algoritmo de índice de vazios e porosidade para a areia

As descrições das etapas do esquema simplificado são:

1. Importação das bibliotecas e módulos necessários para o processamento de imagem, manipulação de dados, visualização de resultados e medição de tempo, sendo as mesmas indicadas no código das areias no item 3.2.1.1;
2. Especificação do diretório que contém as imagens;
3. Inicialização uma lista vazia para armazenar os resultados;
4. Definição do “Fator de Conversão” para transformar as medidas de pixels para milímetros;
5. Iteração sobre os arquivos no diretório indicado para tratamento das imagens:
 - a. Cada imagem é aberta e convertida em uma matriz *numpy* para realizar os tratamentos das imagens;
 - b. Invertem-se as cores;
 - c. Aplica-se filtro de suavização Gaussiano;
 - d. Binariza-se a imagem pelo limiar automático do método de Otsu;
 - e. Calcula-se a matriz de distância utilizando a transformação euclidiana;
 - f. Localizam-se os máximos locais na matriz de distância;
 - g. Os máximos locais são rotulados;
 - h. Aplica-se o algoritmo *Watershed* e obtém-se a individualização das partículas;
6. Cálculos para índice de vazios e porosidade por arquivo (ou *slice*):

- a. Cálculo da área de cada partícula em pixel e conversão em milímetros quadrados;
 - b. Adição das áreas das partículas à “área total de partículas do arquivo”;
 - c. Definição dos contornos das partículas encontradas e cálculo do menor círculo que circunscreve todos os contornos encontrados por arquivo (Figura 3.24), em milímetros quadrados;
 - d. Cálculo da área de vazios: as áreas de todas as partículas (“área total de partículas do arquivo”) são subtraídas da área do menor círculo circunscrito encontrada no item c;
 - e. Cálculo do índice de vazios: relação entre a área de vazios encontrada no item d e área total de partículas do arquivo encontrada no item b; e
 - f. Cálculo da porosidade: relação entre a área de vazios encontrada no item d e a área do menor círculo circunscrito aos contornos de todas as partículas encontrada no item c.
7. Armazenamento dos dados em lista e criação de um *DataFrame* com os dados;
 8. Cálculo do índice de vazios e porosidade da amostra: média dos todos os índices de vazios e porosidades no *DataFrame*.

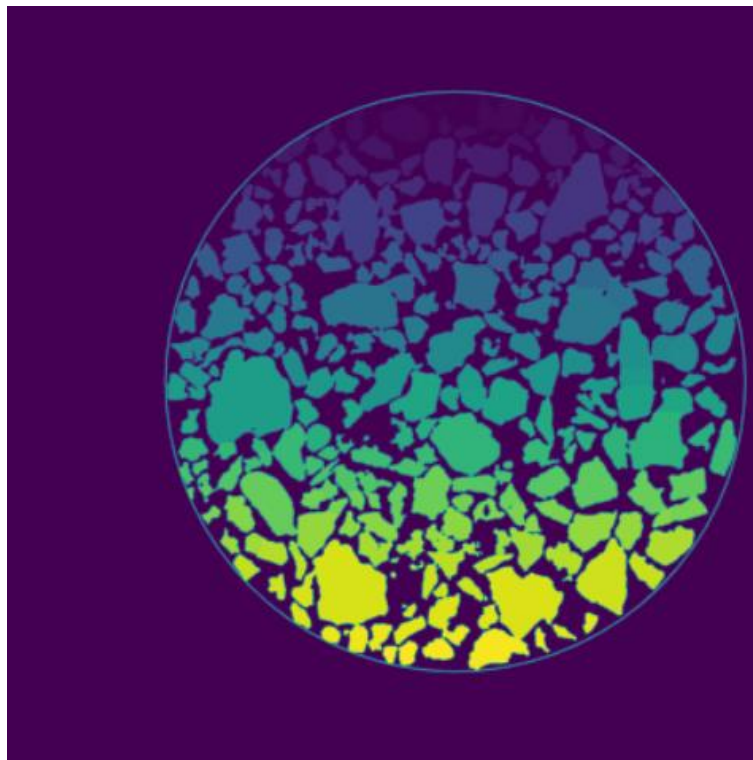


Figura 3.24. Menor círculo que circunscreve todos os contornos de partículas para determinar a área de interesse e realizar o cálculo da área de vazios da amostra

Foi imputado o valor de 5,94 μm por pixel na etapa 4 para o tamanho do pixel da imagem, conforme configurado na microtomografia, indicado na seção 3.1.1.

3.2.3.2. ÍNDICE DE VAZIOS E POROSIDADE DO REJEITO DE MINÉRIO DE FERRO

Os códigos utilizados para inferir o índice de vazios e porosidade das 1.015 imagens 2D do rejeito de minério de ferro estão apresentados no APÊNDICE F e a sua esquematização simplificada é representada também pela Figura 3.23, tendo-se as mesmas etapas da descrição do índice de vazios e porosidade da areia, porém com a diferença de escolha do limiar manual para o caso da granulometria do rejeito de minério de ferro:

1. Importação das bibliotecas e módulos necessários para o processamento de imagem, manipulação de dados, visualização de resultados e medição de tempo, sendo as mesmas indicadas no código das areias no item 3.2.1.1;
2. Especificação do diretório que contém as imagens;
3. Inicialização uma lista vazia para armazenar os resultados;
4. Definição do “Fator de Conversão” para transformar as medidas de pixels para milímetros;
5. Iteração sobre os arquivos no diretório indicado para tratamento das imagens:
 - a. Cada imagem é aberta e convertida em uma matriz *numpy* para realizar os tratamentos das imagens;
 - b. Invertem-se as cores;
 - c. Aplica-se filtro de suavização Gaussiano;
 - d. Arbitra-se um valor de limiar manualmente para operação de binarização da imagem;
 - e. Inverte-se novamente as cores das imagens para tornar os grãos brancos (255) e o fundo preto (0);
 - f. Calcula-se a matriz de distância utilizando a transformação euclidiana;
 - g. Localizam-se os máximos locais na matriz de distância;
 - h. Os máximos locais são rotulados;
 - i. Aplica-se o algoritmo Watershed e obtém-se a individualização das partículas;

- j. Opcionalmente e para fins de conferência visual, disponibiliza-se a plotagem do desenho da individualização de cada partícula após a aplicação das etapas anteriores (Figura 3.25), recomendando-se que seja utilizado para uma pré-avaliação com poucas imagens e que seja colocado como comentário quando aplicado a muitas imagens;
6. Para cada partícula individualizada encontrada no pré-processamento e processamento anterior, a inferência do índice de vazios e porosidade é iniciada:
 - a. Calcula-se a área de cada partícula em pixel e converte-se em milímetros quadrados, por arquivo de imagem;
 - b. Adicionam-se os dados à lista;
 - c. Encontra-se o contorno de cada partícula e o menor círculo que circunscreva todos os contornos das partículas encontrados para delimitar a área da seção da amostra a ser analisada;
 - d. Cálculo da área de vazios subtraindo-se, da área do círculo que circunscreve todas as partículas, a área total das partículas sólidas de rejeito de minério de ferro do item a.
 7. Armazenamento dos dados em lista e criação de um *DataFrame* com os dados;
 8. Cálculo do índice de vazios e porosidade da amostra: média dos todos os índices de vazios e porosidades no *DataFrame*.

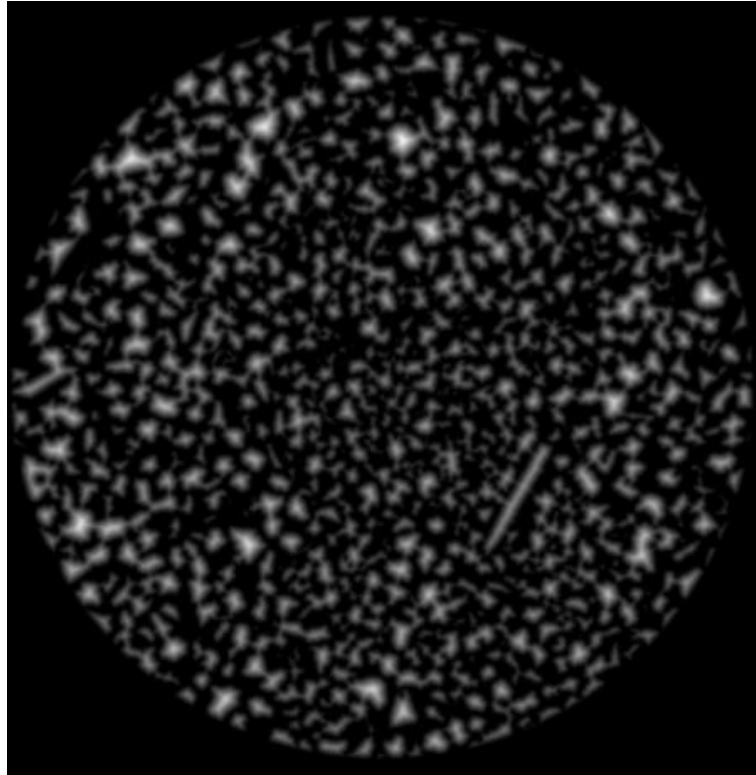


Figura 3.25. Plotagem para verificação do tratamento da imagem até a etapa 5

O fator de conversão indicado no passo 4 foi de $8,3129 \mu\text{m}$ por pixel, conforme tamanho do pixel demonstrado nos parâmetros de configuração da Figura 3.12.a.

3.2.3.3. ÍNDICE DE VAZIOS E POROSIDADE DO COMPÓSITO DE REJEITO DE MINÉRIO DE FERRO COM POLÍMERO

Os códigos utilizados para inferir o índice de vazios e porosidade das 1.015 imagens 2D do compósito de rejeito de minério de ferro com polímero estão apresentados no APÊNDICE G e a sua esquematização simplificada é representada também pela Figura 3.26.

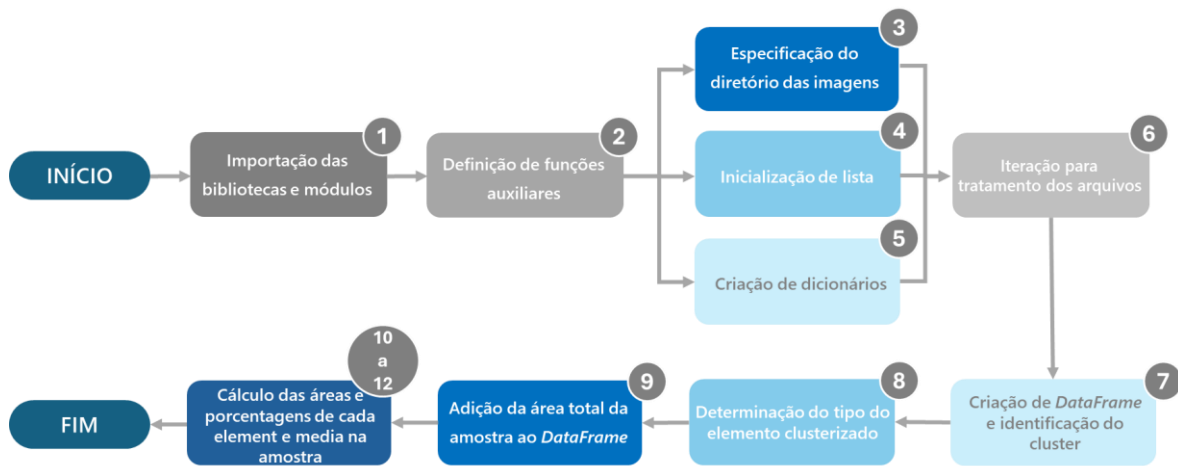


Figura 3.26. Esquematização simplificada do algoritmo de índice de vazios e porosidade para o compósito de rejeito de minério de ferro com polímero

As descrições das etapas do esquema simplificado são:

1. Importação das bibliotecas e módulos necessários para serem utilizados ao longo do código para processamento e análise de imagens, bem como para manipulação de dados e visualização de resultados:
 - a. **“os”**: módulo que fornece funções para interagir com o sistema operacional, como manipular caminhos de arquivos e pastas;
 - b. **“cv2”**: biblioteca OpenCV, utilizada para processamento de imagens e visão computacional, como leitura, processamento e manipulação de imagens;
 - c. **“numpy as np”**: biblioteca Numpy, utilizada para suportar arrays e matrizes multidimensionais, juntamente com uma coleção de funções matemáticas para operar nesses arrays;
 - d. **“pandas as pd”**: biblioteca Pandas, utilizada para manipulação e análise de dados em Python, comumente usada para trabalhar com dados tabulares, como criar, manipular e analisar DataFrames;
 - e. **“PIL.Image”**: biblioteca PIL (Python Imaging Library), utilizada para processamento de imagens em Python. O módulo Image fornece classes e métodos para abrir, manipular e salvar imagens em vários formatos;
 - f. **“sklearn.cluster.KMeans”**: módulo “sklearn.cluster” da biblioteca “scikit-learn”, utilizando a classe KMeans, que implementa o algoritmo de clustering K-Means, utilizado para agrupar dados em K clusters;

- g. **“time”**: módulo que fornece várias funções relacionadas ao tempo, como medir a duração de operações; e
 - h. **“matplotlib.pyplot as plt”**: biblioteca Matplotlib, utilizada para criação de visualizações estáticas, como gráficos e plotagens;
2. Definição de funções auxiliares:
 - a. Função **“calcular_menor_circulo_envolvente”**: calcula o menor círculo circunscrito a todas os pixels brancos (partículas) e calcula a área desse menor círculo;
 - b. Função **“aplicar_mascara”**: aplica uma máscara a uma imagem para facilitar os cálculos por visão computacional;
 3. Especificação do diretório que contém as imagens;
 4. Inicialização uma lista vazia para armazenar os resultados;
 5. Criação de dicionários para armazenar máscaras e área total da amostra por arquivo;
 6. Iteração sobre os arquivos no diretório indicado para tratamento das imagens:
 - a. Cada imagem é aberta e convertida em uma matriz *numpy* para realizar os tratamentos das imagens;
 - b. Invertem-se as cores;
 - c. Redimensionamento da imagem e aplicação de K-Means para segmentação em 3 clusters.;
 - d. Criação de imagens binárias para cada cluster;
 - e. Cálculo do menor círculo para cada cluster e cria uma máscara correspondente;
 - f. Atualização dos dicionários com imagens binárias e máscaras;
 - g. Encontra e armazena o menor círculo entre todos os clusters da imagem;
 7. Criação de um *DataFrame* a partir da lista de resultados e identificação do elemento evidenciado no cluster;
 8. Determinação do tipo de elemento (vazios, ferro, grãos) com base nas áreas;
 9. Adição da área total da amostra ao *DataFrame*;
 10. Cálculo das áreas de cada um dos elementos usando as máscaras e adiciona ao *DataFrame*;
 11. Cálculo da porcentagem da área de cada elemento em relação à área total da amostra; e

12. Cálculo e exibição da média em porcentagem na amostra para cada tipo de elemento.

A principal diferença no código descrito acima em relação aos demais está na maneira de classificar e separar os elementos presentes na amostra. Nos outros códigos, a binarização das imagens era suficiente para distinguir fase sólida (grãos de areia ou rejeito de minério de ferro) e vazios. No entanto, no caso do compósito de rejeito de minério de ferro com polímero, inicialmente tentou-se classificar mais de duas fases ou elementos: vazios, polímero, sílica e ferro. Essa tentativa não obteve sucesso porque não foi possível identificar o elemento "polímero" separadamente das demais fases a partir do histograma de intensidade de pixels. Isso se deve ao fato de que o polímero não possui uma densidade suficientemente contrastante em relação aos vazios, além de parte do material fino ter se misturado a ele.

Dessa forma, o código limita-se a classificar e separar três elementos: vazios, sílica (rejeito) e minério de ferro, considerando que o polímero estará com o contraste muito próximo ao da sílica por ter material fino misturado a ele. A técnica escolhida para realizar a segmentação dos elementos é a clusterização, que é uma técnica que utiliza algoritmos de aprendizagem de máquina para segmentar imagens. A clusterização, também conhecida como agrupamento, gera grupos com base na similaridade, que, neste trabalho, será tratada pela similaridade de intensidade de pixels encontrada no histograma.

Com a aplicação do algoritmo K-Means indicado no item 6.c, três imagens binárias foram criadas para cada arquivo de imagem (item 6.d), separando a área de interesse (elementos) em branco do fundo em preto (Figura 3.27).

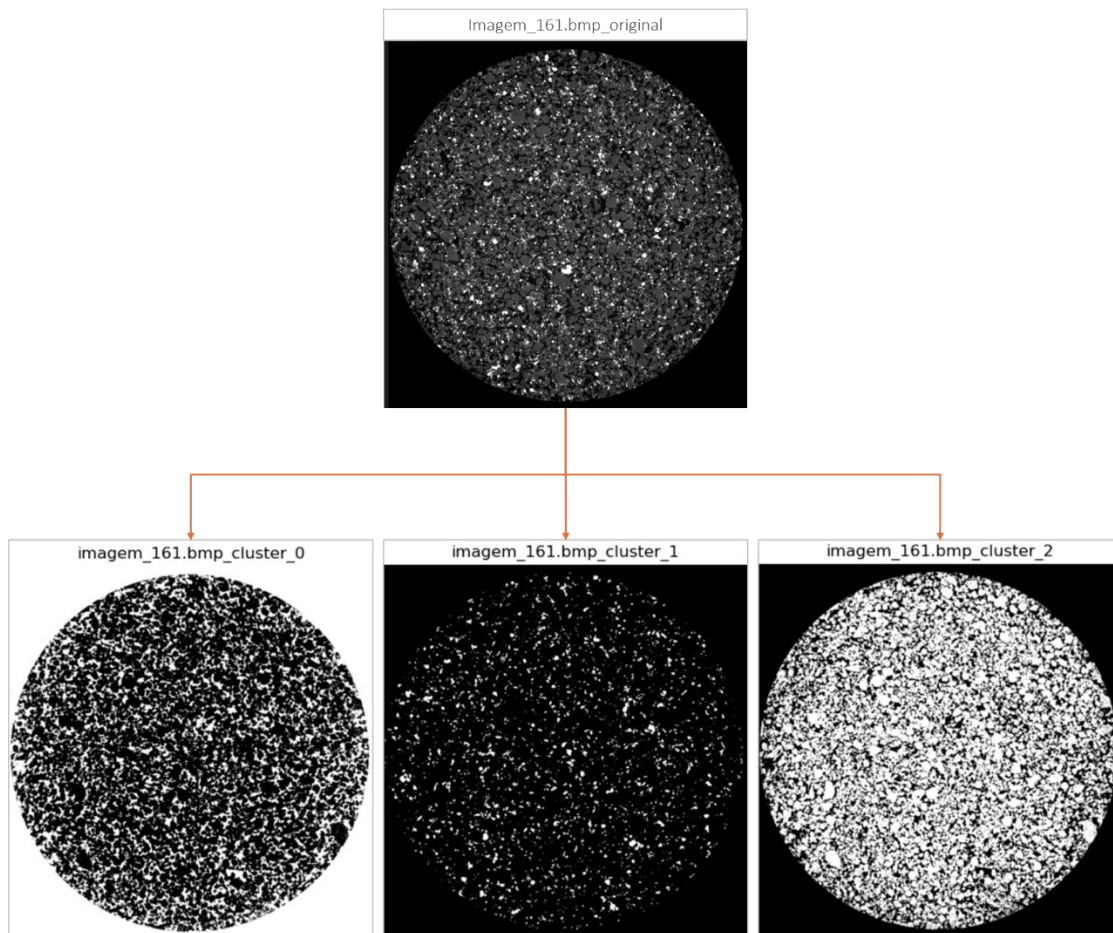


Figura 3.27. Clusterização de um arquivo de imagem 2D em três elementos

Para os cálculos de índice de vazios e porosidade é necessário encontrar a área da amostra, desconsiderando o seu fundo (item 6.g). Para isso, utilizou-se cada uma das três imagens binarizada com seu cluster em destaque: calculou-se o menor círculo que circunscreve os pixels brancos (elementos em destaque) de cada um dos clusters de cada um dos arquivos e escolheu-se, dentre os três círculos, o menor para ser considerada a área total da seção 2D da amostra contendo todos os elementos e desconsiderando o fundo. Apesar dessa escolha poder não considerar parte de elementos na borda da amostra, essa solução não tem influência relevante nos resultados, pois a contagem dos elementos também será restrita a essa área do menor círculo circunscrito.

O próximo passo foi a identificação de qual elemento está destacado em cada uma das três imagens clusterizadas. Partindo-se do conhecimento da composição do material (Figura 3.6), é característico da tomografia que, quanto mais denso o material é, maior é a sua intensidade do pixel, ou seja, mais claro, sendo possível inferir os elementos na imagem original

(Figura 3.27): minérios de ferro em cores mais claras, os grãos de sílica em cinza e os vazios em preto.

Utilizando-se a Figura 3.27 como exemplo, foram criadas três condições: imagens com o primeiro pixel da matriz da imagem na cor branca (255), como mostrado no cluster 0, são classificadas como uma imagem que está destacando os vazios; imagens com o menor total da área de pixels brancos entre as três imagens clusterizadas, como mostrado no cluster 1, são classificadas como uma imagem que está destacando minério de ferro; a imagem que restar é considerada a que destaca os grãos de sílica.

Dessa forma, encontrou-se a área correspondente a cada um dos três elementos clusterizados (vazios, sílica e minério de ferro) e calculou-se os valores de suas porcentagens em relação à área total da amostra (item 6.g).

CAPÍTULO 4

4. RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados alcançados pela dissertação. Conforme Capítulo 3, as análises estão divididas em: esfericidade para a areia e rejeito de minério de ferro, granulometria para a areia e rejeito de minério de ferro e índice de vazios e porosidade para a areia, rejeito de minério de ferro e compósito de rejeito de minério de ferro com polímero.

4.1. ESFERICIDADE DA AREIA

A partir dos dados gerados no código apresentado no APÊNDICE A, chegou-se ao valor médio das esfericidades por Rittenhouse (1943) e por Rorato *et al.* (2019) para a amostra da areia e comparou-se com o valor médio encontrado por Borges *et al.* (2019), conforme Tabela 4.1.

Tabela 4.1. Esfericidades médias encontradas para a areia

Esfericidade média por Rittenhouse (1943)	Esfericidade do perímetro média por Rorato (2019)	Esfericidade média por Borges <i>et al.</i> (2019)
53%	78%	79%

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 4.510 segundos, ou 75 minutos, para as 152.028 partículas encontradas.

A esfericidade por Rittenhouse (1943) foi calculada adaptando-se a Eq. (3) apresentada e chegando-se à Eq. (10), considerando que os dados gerados diretamente pelo código foram a área da partícula (A_{eq}) e a área do círculo circunscrito à partícula (A_c):

$$Esfericidade = \frac{d_{eq}}{d_c} = \frac{\sqrt{\frac{4A_{eq}}{\pi}}}{\sqrt{\frac{4A_c}{\pi}}} = \frac{A_{eq}}{A_c} \quad (10)$$

onde:

d_{eq} : diâmetro do círculo com uma área igual à projeção da partícula na imagem 2D; e

d_c : diâmetro do menor círculo circunscrito à projeção da partícula na imagem 2D.

A divergência do valor encontrado no cálculo da esfericidade média pelo método de Rittenhouse (1943) em relação aos demais métodos demonstra como diferentes abordagens resultam em valores distintos por considerarem aspectos diferentes em sua formulação. Já a proximidade de resultados de esfericidade pelo método utilizado por Borges *et al.* (2019) com a de Rorato *et al.* (2019) demonstram que a escolha de cálculo para a esfericidade de Borges *et al.* (2019) também pode ter uma boa correlação com a esfericidade 3D.

Considerando-se a proximidade de resultados da esfericidade do perímetro (Rorato *et al.*, 2019) e o resultado de esfericidade de Borges *et al.* (2019), os dados obtidos no código do APÊNDICE A foram tratados com o objetivo de serem retirados os *outliers* que pudessem ser indicativos de erros ou que pudessem distorcer a média para a amostra. Para isso os *outliers* foram identificados a partir dos limites inferior e superior com base no desvio padrão da amostra: média ± 2 vezes o valor do desvio padrão. O resultado da remoção dos *outliers* está demonstrado na Figura 4.1, obtendo-se a distribuição de frequência da esfericidade do perímetro conforme Figura 4.2 e a sua distribuição acumulada na Figura 4.3.

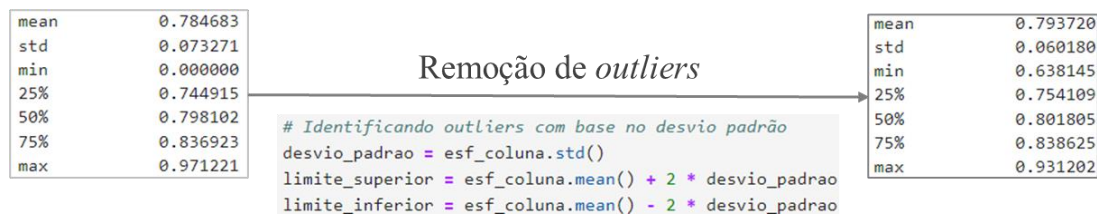


Figura 4.1. Remoção de outliers dos dados obtidos no código de esfericidade do perímetro para a areia

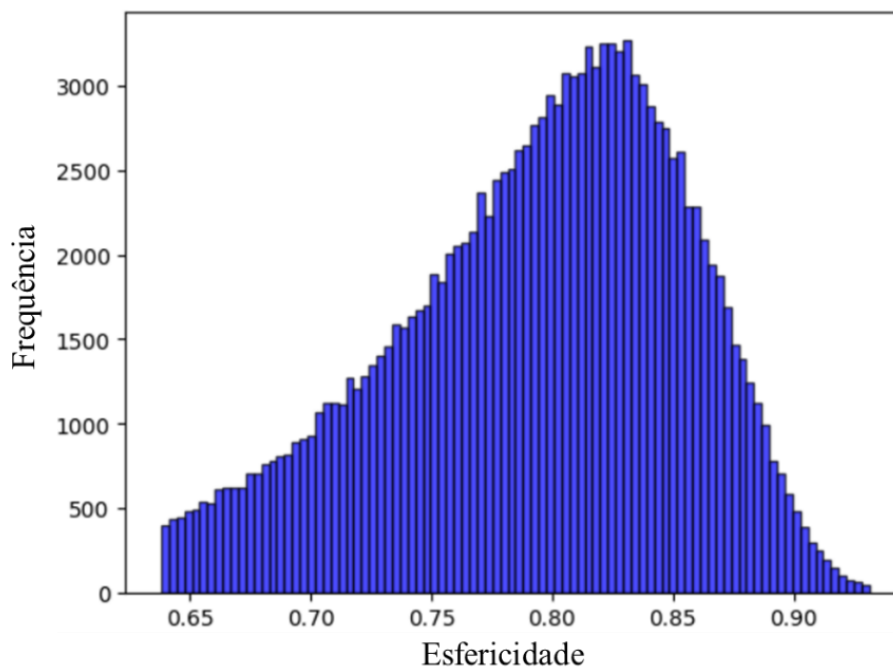


Figura 4.2. Distribuição de frequência da esfericidade do perímetro para a amostra de areia

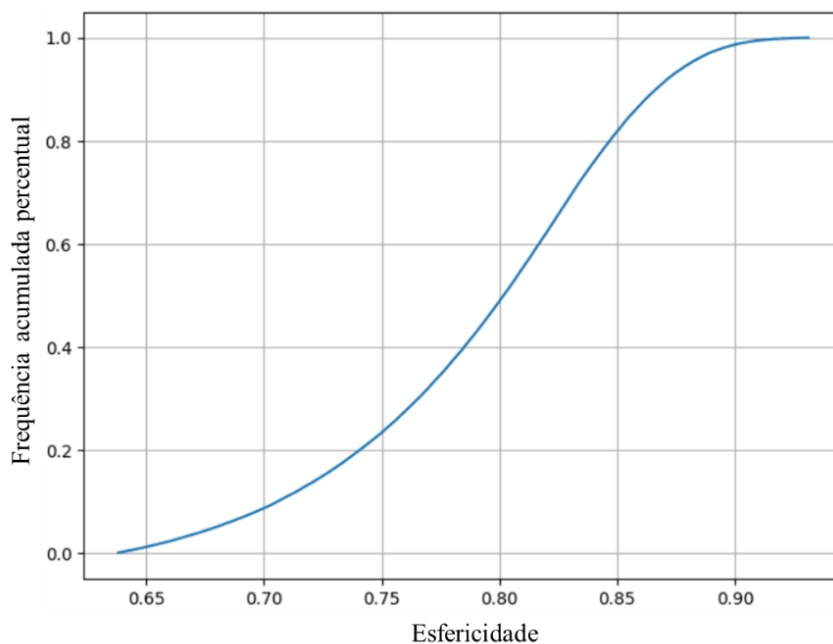


Figura 4.3. Distribuição acumulada percentual da esfericidade do perímetro para a amostra de areia

Dessa forma, o valor encontrado para a média da esfericidade do perímetro da amostra da areia após o tratamento dos dados com a remoção dos *outliers* (Figura 4.1) foi de 79%, mesmo valor encontrado por Borges *et al.* (2019).

4.2. ESFERICIDADE DO REJEITO DE MINÉRIO DE FERRO

A partir dos dados gerados no código apresentado no APÊNDICE B, chegou-se ao valor médio das esfericidades por Rittenhouse (1943) e por Rorato *et al.* (2019) para a amostra do rejeito de minério de ferro, conforme Tabela 4.2.

Tabela 4.2. Esfericidades médias encontradas para o rejeito de minério de ferro

Esfericidade média por Rittenhouse (1943)	Esfericidade do perímetro média por Rorato (2019)
52%	73%

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 2.177 segundos, ou 36 minutos, para as 420.928 partículas encontradas.

Considerando que o parâmetro esfericidade para o material de rejeito de minério de ferro não foi estudado em pesquisas anteriores, tomou-se como base para estudo de similaridade os resultados da areia, que também foi avaliada pelo método de Rittenhouse (1943) assim como pela esfericidade do perímetro. Os resultados para o rejeito de minério de ferro indicaram um valor menor para a esfericidade de Rittenhouse (1943) quando comparada à esfericidade do perímetro, assim como na comparação da amostra de areia (Tabela 4.1. Esfericidades médias encontradas para a areia), reforçando a ideia de que valores de esfericidade de Rittenhouse (1943) são menores do que a esfericidade do perímetro (assim como para a esfericidade adotada por Borges *et al.* (2019)).

As distribuições de frequência da esfericidade e a distribuição acumulada da esfericidade, por Rittenhouse (1943) e pelo método da esfericidade do perímetro, após a remoção de outliers, estão apresentadas na Figura 4.4 e Figura 4.5.

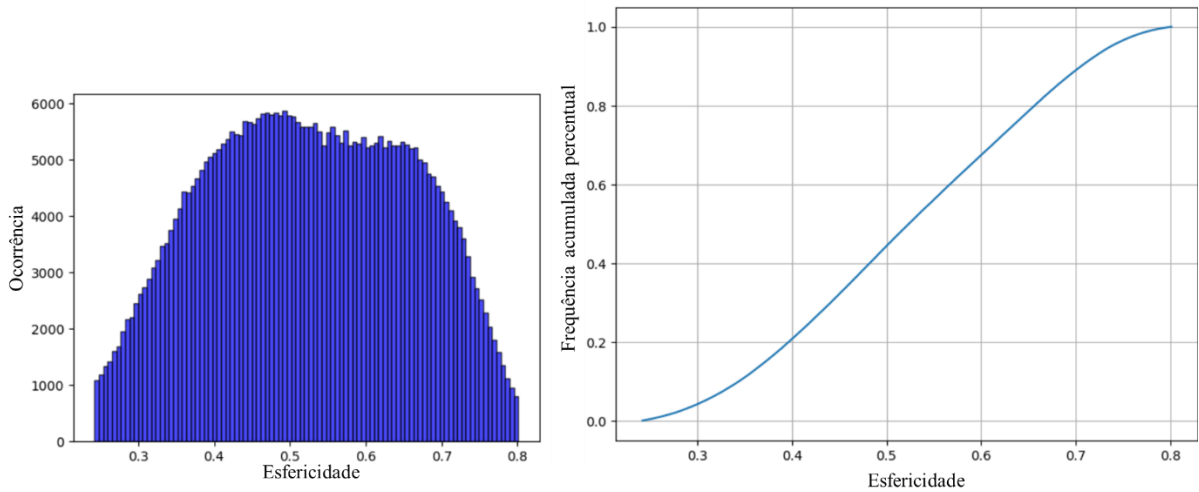


Figura 4.4. Distribuição de frequência da esfericidade e distribuição acumulada percentual da esfericidade do rejeito de minério de ferro pelo método de Rittenhouse (1943).

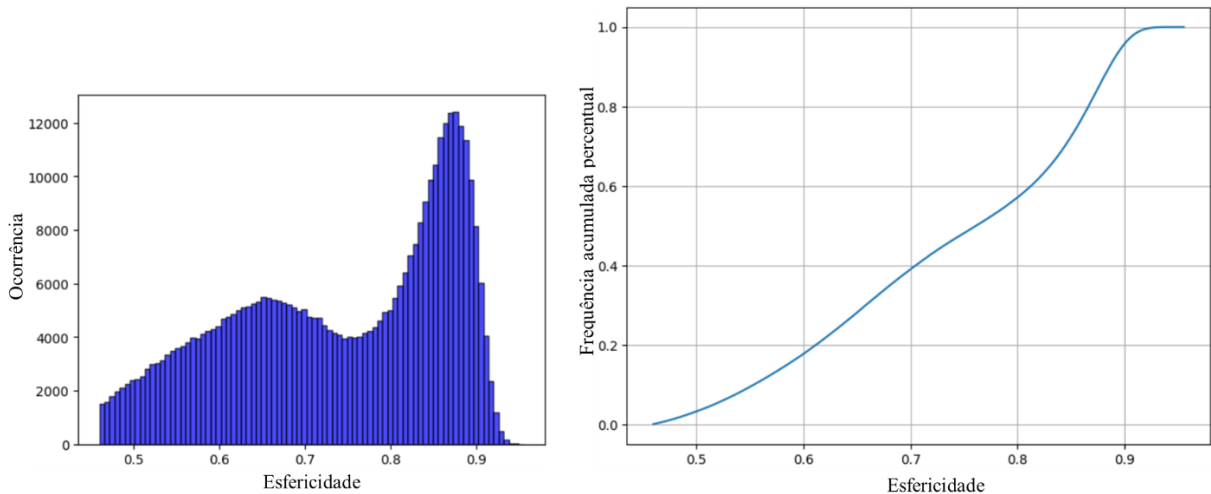


Figura 4.5. Distribuição de frequência da esfericidade e distribuição acumulada percentual da esfericidade do rejeito de minério de ferro pelo método da esfericidade do perímetro.

As estatísticas descritivas dessas distribuições estão apresentadas na Figura 4.6.

count	411711.000000
mean	0.524854
std	0.133406
min	0.242233
25%	0.419450
50%	0.523711
75%	0.633844
max	0.800957

(a)

count	408933.000000
mean	0.739739
std	0.126356
min	0.459787
25%	0.636019
50%	0.761503
75%	0.855929
max	0.955439

(b)

Figura 4.6. Estatísticas descritivas dos dados de esfericidade do rejeito de minério de ferro pelo (a) método de Rittenhouse (1943) e (b) método da esfericidade do perímetro.

É possível notar que a escolha do método de cálculo para a esfericidade do material tem grande influência nos resultados, com distribuições e estatísticas descritivas discrepantes.

4.3. GRANULOMETRIA DA AREIA

A partir dos dados gerados no código apresentado no APÊNDICE C, chegou-se à distribuição granulométrica para a amostra da areia, conforme Figura 4.7, utilizando-se o mesmo método e simplificações de Borges *et al.* (2019).

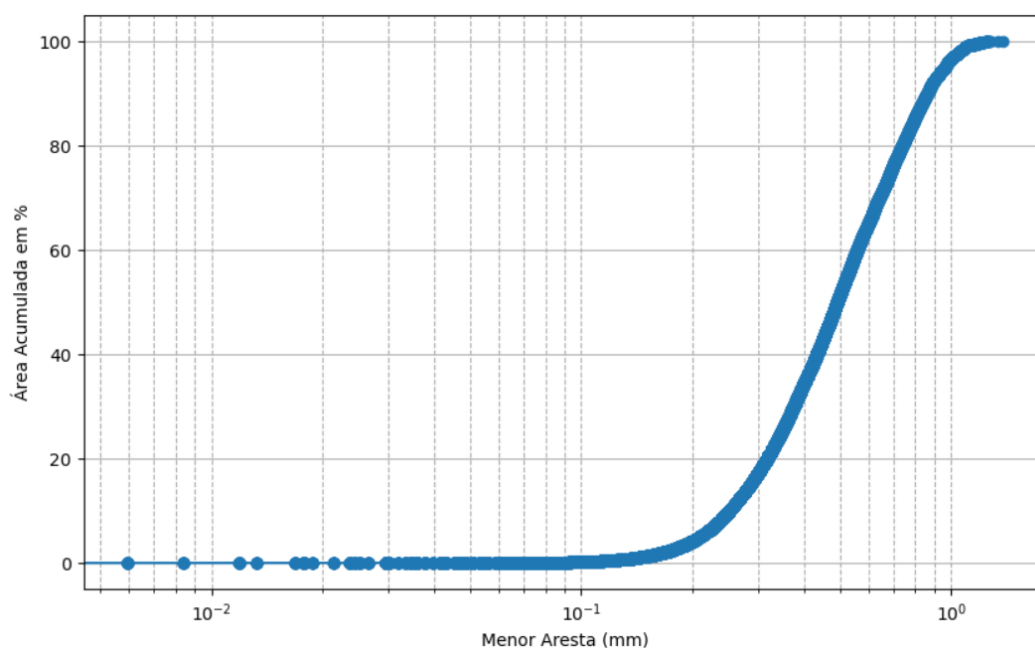


Figura 4.7. Distribuição granulométrica da areia em escala logarítmica: menor aresta do menor retângulo delimitador rotacionado vs. área acumulada em porcentagem

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 3.000 segundos, ou 50 minutos, para as 152.028 partículas encontradas.

Juntando-se o gráfico de distribuição granulométrica gerado pelo código do APÊNDICE C (Figura 4.7) com o gráfico experimental de Zubeldia (2013), apresentado na Figura 3.2, para comparação de resultados, obtém-se o gráfico da Figura 4.8.

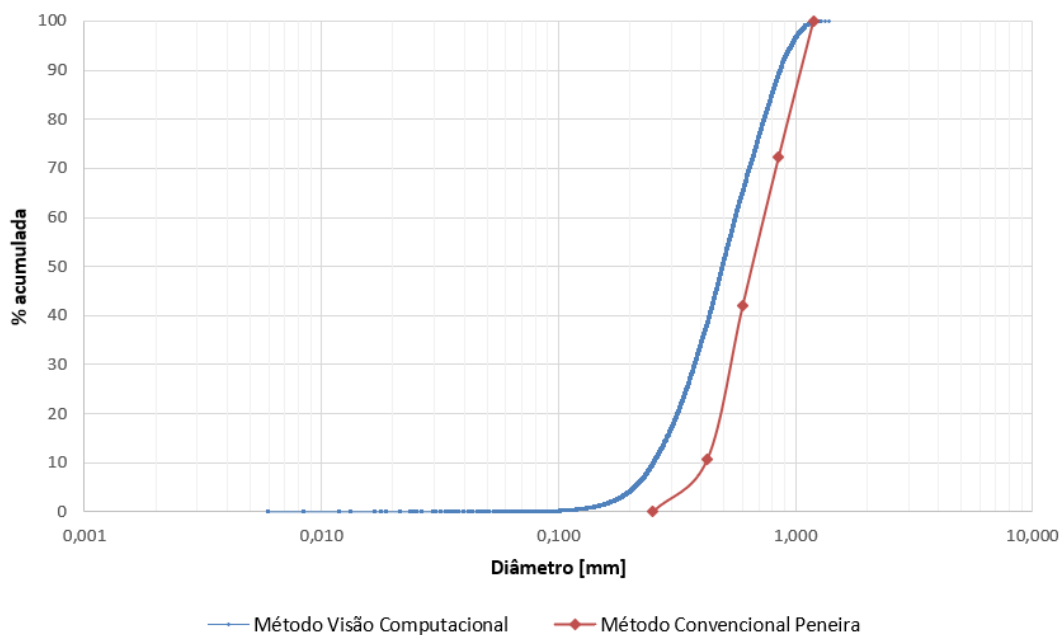


Figura 4.8. Comparativos das distribuições granulométricas da areia pelo método de visão computacional utilizando linguagem Python e pelo método convencional de peneiramento

A partir das curvas granulométricas, foi possível determinar os diâmetros D_{10} , D_{30} , D_{60} , o coeficiente de uniformidade (C_u) e o coeficiente de curvatura (C_c) (Tabela 4.3). Percebe-se que há uma diferença nos valores dos diâmetros aferidos experimental e computacionalmente, o que se revela quando observado o paralelismo à esquerda da curva pelo método computacional de análise de imagens em relação à curva pelo método de peneira. Porém, apesar dessa diferença, o material mantém-se classificado como um solo uniforme, devido ao seu coeficiente de uniformidade menor que 5, e mal graduado, por seu coeficiente de conformidade menor que 1.

Tabela 4.3 Comparação de resultados pelo método convencional de peneira e pelo método computacional de análise de imagens para a amostra de areia

Parâmetro	Método Convencional (Peneiramento)	Método Computacional (Python)
D_{10}	0,43	0,25
D_{30}	0,55	0,37
D_{60}	0,73	0,55
C_u	1,70	2,20
C_c	0,96368	0,99563

Nota-se que a curva granulométrica pelo método de visão computacional é mais suave quando comparada à curva pelo método de peneira, já que essa última forma se utiliza de intervalos de diâmetros, enquanto a primeira considera diâmetros contínuos. Isso permite que seja identificado mais precisamente os valores para análise dos coeficientes de uniformidade e curvatura. Além disso, verifica-se que a diferença dos diâmetros D10, D30 e D60 entre os métodos se mantém em 0,18 mm, demonstrando-se o paralelismo entre as curvas.

A discrepância observada entre a curva resultante do processamento de imagens 2D de microtomografias e a curva experimental obtida através do método de peneiramento, Figura 4.8, pode ser explicada pelo fato de que o peneiramento a seco possui menor precisão para grãos menores. Isso ocorre porque a aplicação da energia mecânica necessária para que as partículas passem pela abertura da malha aumenta a atração superficial entre as próprias partículas e entre as partículas e a tela, à medida que o tamanho das partículas diminui (Vangla *et al.* 2014), acumulando-se maior quantidade de material em peneiras de diâmetros maiores. Dessa forma, obtém-se uma curva granulométrica pelo método de peneira mais à direita quando comparado ao método computacional.

Outro fator que pode influenciar o resultado é a perda de materiais mais finos para o ambiente no momento do peneiramento e a execução de agitação com tempo e energia insuficientes para permitir a passagem dos grãos para peneiras de malhas menores, acumulando-se maior quantidade em peneiras de diâmetro maior. Além disso, há a questão da representatividade da amostra em equipamentos de microtomografia, que são quantitativamente menores quando comparada à determinação da granulometria pelo método convencional de peneira, podendo ter sido escolhida uma fração não representativa para escaneamento.

No que diz respeito ao código, foi notado que, durante o processamento das imagens e, conforme demonstrado pela Figura 4.9, pode ocorrer imprecisão na individualização das partículas pela chamada supersegmentação, que divide um mesmo grão em mais de um, resultando em um aumento de grãos menores na amostra. Entretanto, esse fenômeno não foi observado de forma extensiva ao longo das fatias. Portanto, entende-se que o impacto no deslocamento à esquerda da curva granulométrica obtida pelo método computacional não foi significativo.

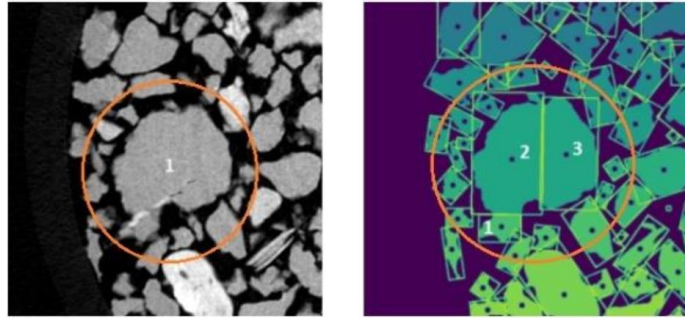


Figura 4.9. Exemplo de supersegmentação das partículas

Apesar das diferenças citadas, foi possível observar que o material apresentou uma distribuição com similaridade, sendo classificado dentro da mesma categoria, ou seja, areia média. Além disso, os coeficientes de uniformidade e de curvatura convergiram para o mesmo intervalo de classificação, o que demonstra a possibilidade de aplicação do algoritmo.

4.4. GRANULOMETRIA DO REJEITO DE MINÉRIO DE FERRO

A partir dos dados gerados no código apresentado no APÊNDICE D, chegou-se à distribuição granulométrica para a amostra de rejeito de minério de ferro peneirada (grãos retidos na peneira 60, de abertura de 0,25 mm), conforme Figura 4.10, utilizando-se o mesmo método e simplificações de Borges *et al.* (2019).

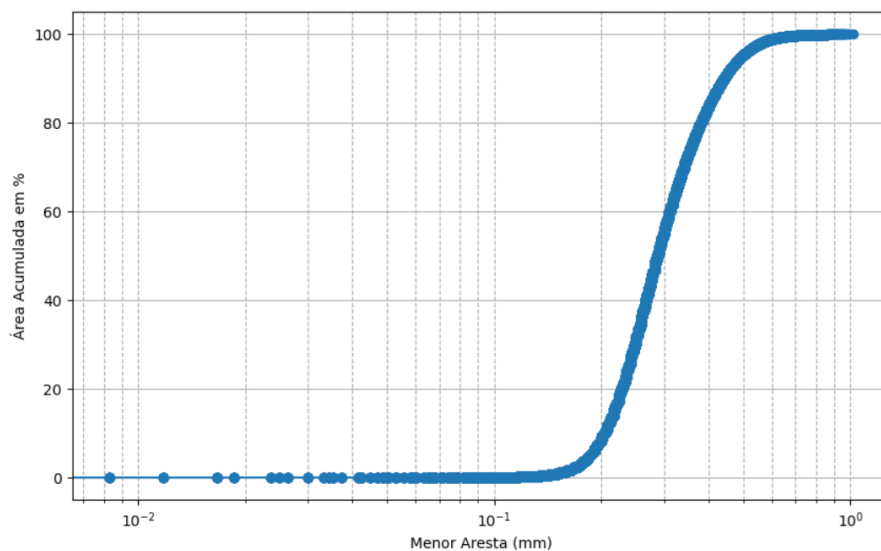


Figura 4.10. Distribuição granulométrica do rejeito de minério de ferro em escala logarítmica: menor aresta do menor retângulo delimitador rotacionado vs. área acumulada em porcentagem

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 1.611 segundos, ou 27 minutos, para as 419.146 partículas encontradas.

Juntando-se o gráfico de distribuição granulométrica gerado pelo código do APÊNDICE C (Figura 4.10) com o gráfico experimental de Carneiro (2020) somente para a fração de granulometria nas peneira de diâmetro de 0,149 mm a 2,00 mm para efeitos de comparação, conforme Figura 3.5, obteve-se o gráfico da Figura 4.11.

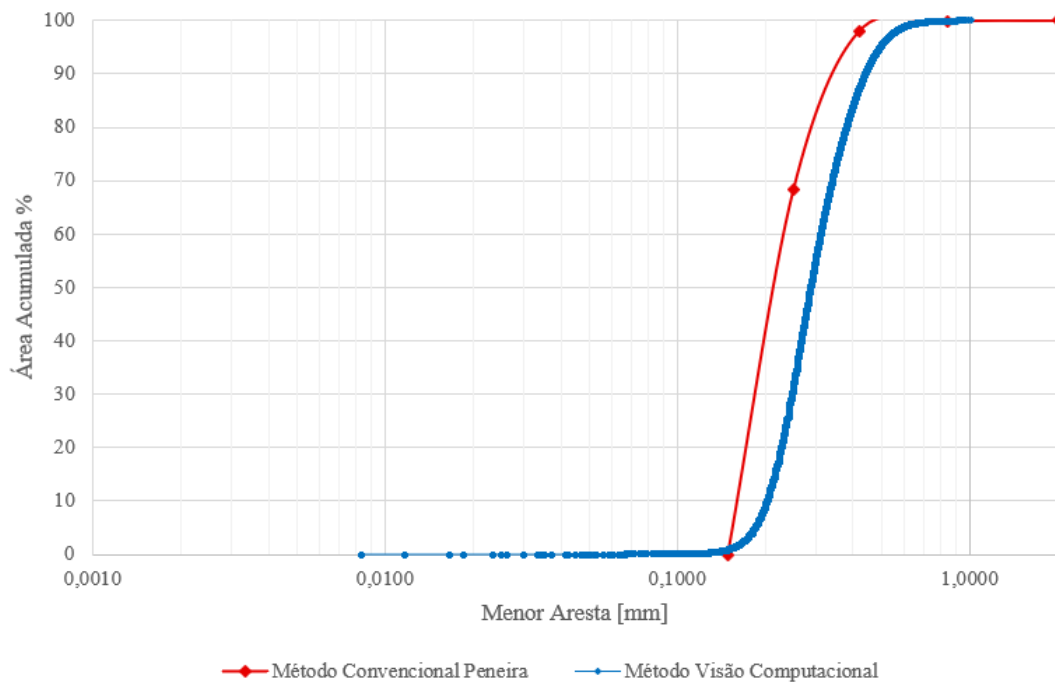


Figura 4.11. Comparativos das distribuições granulométricas do rejeito de minério de ferro pelo método de visão computacional utilizando linguagem Python e pelo método convencional de peneiramento (fração de 0,149 mm a 2,00 mm)

A análise dos parâmetros D_{10} , D_{30} , D_{60} , C_u e C_c não são aplicáveis neste caso pelo fato de o material para a análise computacional ter sido peneirado (grãos retidos na peneira de abertura de 0,25 mm) e comparado com uma fração da distribuição granulométrica pelo método convencional, o qual se restringiu ao intervalo de grãos retidos na fração de 0,149 mm a 2,00 mm.

Nota-se que a curva da análise computacional está deslocada à direita da curva experimental, indicando que o método pela visão computacional identificou maiores porcentagens para grãos de maior tamanho. Essa diferença pode ser atribuída à escolha do limiar manual, que possivelmente superestimou o tamanho dos grãos, e pela possibilidade da escolha da amostra não ser de uma fração representativa para escaneamento. Apesar dessa

diferença, foi possível observar que o material apresentou uma distribuição com similaridade, mantendo-se o paralelismo entre as curvas.

4.5. ÍNDICE DE VAZIOS E POROSIDADE DA AREIA

A partir dos dados gerados no código apresentado no APÊNDICE E, chegou-se à média de índice de vazios e à média da porosidade para a amostra da areia utilizando-se o mesmo método e simplificações por Borges *et al.* (2019) e comparou-se com o valor médio encontrado por Borges *et al.* (2019), conforme Tabela 4.4.

Tabela 4.4. Comparativo de índice de vazios e porosidade da areia calculados por código em Python e calculado pelo software *Mathematica* em Borges *et al.* (2019)

Índice de vazios com Python	Índice de vazios por Borges <i>et al.</i> (2019)	Porosidade com Python	Porosidade por Borges <i>et al.</i> (2019)
79%	76%	44%	43%

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 3.200 segundos, ou 54 minutos, para as 610 imagens 2D da areia.

Observa-se uma relevante proximidade entre os valores, evidenciando uma correlação entre os cálculos e a análise realizada pelo código em Python e pelo software *Mathematica*. Mesmo que Borges *et al.* (2019) tenham adotado um tamanho de pixel de 11,32 μm e o código em Python tenha utilizado 5,94 μm , a relação percentual não interfere no valor encontrado.

4.6. ÍNDICE DE VAZIOS E POROSIDADE DO REJEITO DE MINÉRIO DE FERRO

A partir dos dados gerados no código apresentado no APÊNDICE F, chegou-se à média de índice de vazios e à média da porosidade para a amostra do rejeito de minério de ferro utilizando-se o mesmo método e simplificações por Borges *et al.* (2019) e comparou-se com o valor médio encontrado pelo software *DragonFly* do Laboratório de Fabricação Digital da Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio, conforme Tabela 4.5.

Tabela 4.5. Comparativo de índice de vazios e porosidade do rejeito de minério de ferro calculados por código em Python e calculado pelo software *DragonFly*

Índice de vazios com Python	Índice de vazios com DragonFly	Porosidade com Python	Porosidade com DragonFly
84%	84%	45%	45%

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 1.569 segundos, ou 26 minutos, para as 1.015 imagens 2D do rejeito de minério de ferro.

Os valores calculados para o *DragonFly* consideraram as porcentagens obtidas do volume tridimensional da amostra microtomografadas e os valores calculados pelo código do APÊNDICE F considerou as imagens bidimensionais. Os valores obtidos no *DragonFly* foram originalmente: 54,3% de grãos de sílica, 0,2% de minério de ferro e 45,5% de vazios. Dessa forma, o valor de 45,5% de vazios representa a porosidade, já que é a porcentagem para todo o corpo de prova, e o índice de vazios considerou a porcentagem de vazios em relação à soma das porcentagens sólidas (54,3% de grãos de sílica somada à 0,2% do minério de ferro).

Nota-se que os valores convergem para os mesmos valores, demonstrando-se que a metodologia proposta por Borges *et al.* (2019) apresenta uma boa correlação entre os cálculos e análise pelo código em linguagem Python para as imagens bidimensional e pelo software *DragonFly* para o volume tridimensional.

Apesar da boa correlação demonstrada, é importante destacar como a escolha do limiar manual para a segmentação da imagem pode influenciar esse parâmetro (Figura 4.12):

LIMIAR DE 225		
	Índice de vazios na slice	Porosidade na slice
0	1.398244	0.583028
1	0.806459	0.446431
2	0.790449	0.441481
3	0.834461	0.454881
4	0.802520	0.445221
5	0.796283	0.443295
6	0.757450	0.430994
7	0.770631	0.435230
8	0.727877	0.421255
9	0.767647	0.434276
Índice de vazios da amostra: 0.8452021227979426		
Porosidade da amostra: 0.45360914385226864		
Tempo de execução: 26.086923837661743 segundos		

LIMIAR DE 224		
	Índice de vazios na slice	Porosidade na slice
0	1.999628	0.666625
1	0.889315	0.470708
2	0.875359	0.466769
3	0.909940	0.476423
4	0.897343	0.472947
5	0.875165	0.466714
6	0.836794	0.455573
7	0.854559	0.460788
8	0.807739	0.446823
9	0.858354	0.461889
Índice de vazios da amostra: 0.9804196917751453		
Porosidade da amostra: 0.4845259689725441		
Tempo de execução: 16.915174961090088 segundos		

LIMIAR DE 223		
	Índice de vazios na slice	Porosidade na slice
0	2.932544	0.745712
1	0.968491	0.491997
2	0.952499	0.487836
3	1.003039	0.500759
4	0.979774	0.494892
5	0.967309	0.491691
6	0.942228	0.485127
7	0.947134	0.486425
8	0.895376	0.472400
9	0.943038	0.485342
Índice de vazios da amostra: 1.1531431967853865		
Porosidade da amostra: 0.5142180346479045		
Tempo de execução: 17.229045629501343 segundos		

Figura 4.12. Comparativo da variação dos índices de vazios e porosidades pela alteração manual do limiar de segmentação

Ao comparar os resultados do código Python para 10 arquivos aleatórios de imagens de fatias bidimensionais, alterando-se manualmente o limiar em uma unidade, observa-se que os valores de índice de vazios e porosidade apresentam alta sensibilidade a essa alteração. Portanto, a escolha do limiar deve ser um ponto de atenção para o usuário do código.

4.7. ÍNDICE DE VAZIOS E POROSIDADE DO COMPÓSITO DE REJEITO DE MINÉRIO DE FERRO E POLÍMERO

A partir dos dados gerados no código apresentado no APÊNDICE G, chegou-se à média de porcentagem dos elementos de ferro, grãos de sílica para a amostra do compósito de rejeito de minério de ferro com polímero utilizando-se o método e simplificações análogas de Borges *et al.* (2019) e comparou-se com o valor médio encontrado pelo software *DragonFly* do

Tabela 4.6. Comparativo da porcentagem de elementos na amostra do compósito de rejeito de minério de ferro com polímero calculados por código em Python e calculado pelo software *DragonFly*

Cluster	Porcentagem de elementos com Python	Porcentagem de elementos com <i>DragonFly</i>
Grãos de sílica	64,2 %	56,5 %
Minério de Ferro	6,6 %	7,9%
Vazios	29,2 %	35,6 %

O tempo de execução do código na estação de trabalho citada foi de aproximadamente 4.018 segundos, ou 67 minutos, para as 1.015 imagens 2D do compósito de rejeito de minério de ferro com polímero.

A Figura 4.13 apresenta a segmentação no volume tridimensional utilizada para os cálculos das porcentagens dos elementos identificados na amostra do compósito de rejeito de minério de ferro com polímero.

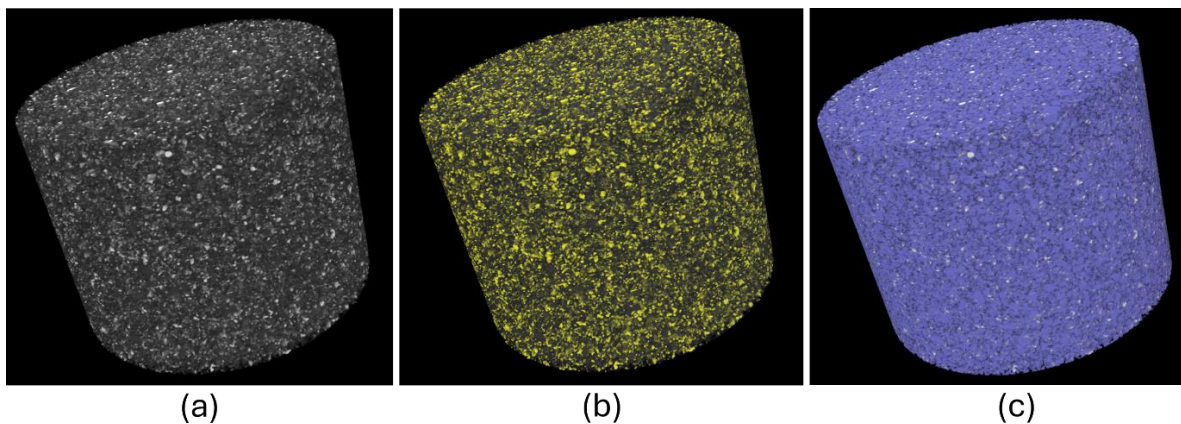


Figura 4.13. Segmentação pelo software *DragonFly*: (a) Imagem original; (b) Segmentação com destaque em amarelo para os minérios de ferro; (c) Segmentação com destaque em roxo para os grãos de sílica

Apesar de próximos, houve uma divergência considerável na inferência dos elementos pelos métodos, o que gerou resultados divergentes para o índice de vazios e porosidade estimados: pelo código Python, chegaram-se aos valores aproximados de 29% de porosidade e

41% de índices de vazios; já pelo software DragonFly, chegaram-se aos valores aproximados de 36% de porosidade e 55% de índice de vazios.

Esses valores divergentes podem ser atribuídos ao tipo de segmentação escolhida no código Python, o algoritmo K-Means, que agrupa os pixels a partir da sua categorização pela similaridade de suas intensidades, de forma não supervisionada, de modo a minimizar a variabilidade interna de cada grupo, podendo não corresponder a limites intuitivos. Isso pode ter influência significativa na segmentação a partir dos limiares determinados, como foi demonstrado na Figura 4.12.

Além disso, para que o algoritmo K-Means possa ser bem aplicado, é importante que os clusters de intensidade de pixel sejam bem separados, o que não é observado quando se plota o histograma da imagem original (Figura 4.14).

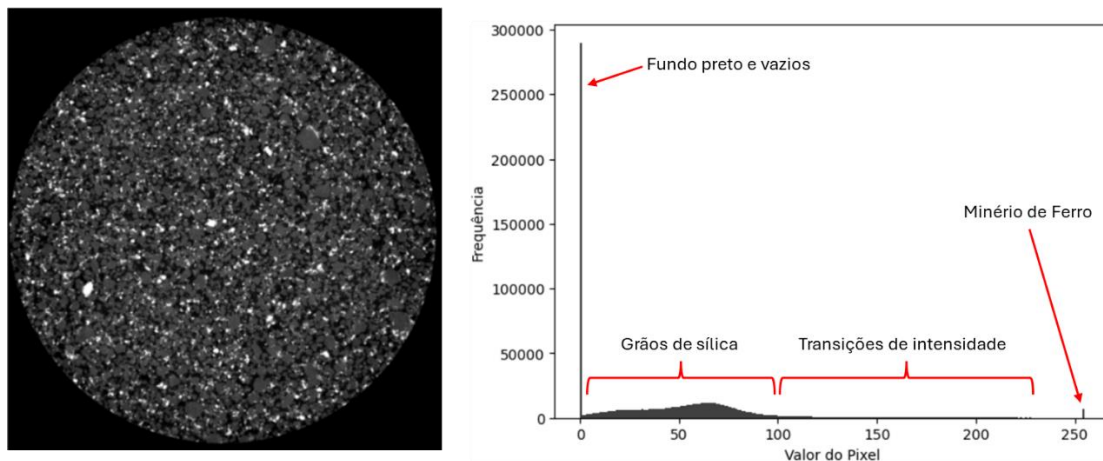


Figura 4.14. Histograma de imagem bidimensional do compósito de rejeito de minério de ferro com polímero.

CAPÍTULO 5

5. CONSIDERAÇÕES FINAIS

Esta pesquisa teve como objetivo a implementação de algoritmos e desenvolvimento de script em linguagem Python para processamento de imagens bidimensionais geradas a partir da microtomografia computadorizada de raio-x de amostras geotécnicas de areia, rejeito de minério de ferro e compósito de rejeito de minério de ferro com polímero, buscando-se analisar parâmetros morfológicos em mesoescala de esfericidade, granulometria, porosidade e índice de vazios.

As imagens bidimensionais geradas para a areia foram pré-processadas pela determinação de limiar global automático de *Otsu*, o que otimiza o tempo de execução do código aplicado. Em relação à análise de esfericidade, granulometria, índice de vazios e porosidade da areia, foi possível verificar correlação entre os valores obtidos por outros métodos e os valores obtidos pela proposta de algoritmo em linguagem Python: a esfericidade pelo código em Python obteve o mesmo valor encontrado pelo software *Mathematica*; a granulometria apresentou uma distribuição com similaridade à curva experimental, mantendo-se a sua classificação de areia média; e o índice de vazios e porosidade encontrados foram relevantemente próximos.

Para as imagens obtidas para o rejeito de minério de ferro, o pré-processamento utilizou a segmentação com limiar escolhido manualmente, que possui relevante influência nos resultados e demanda que o usuário do código Python analise visualmente a imagem para a escolha do limiar. Em relação à análise de esfericidade, granulometria, índice de vazios e porosidade do rejeito de minério de ferro, destacam-se:

- a comparação experimental ou comparação com outro método computacional para esfericidade não foi possível devido à inexistência de pesquisas precedentes com o mesmo parâmetro e material analisado;
- notou-se que os resultados dos métodos de Rittenhouse (1943) e esfericidade do perímetro citado por Rorato (2019) não convergem para um mesmo valor, sendo a esfericidade do perímetro relativamente mais alta quando comparada à esfericidade de Rittenhouse;
- a granulometria obtida computacionalmente para o rejeito demonstra uma tendência de existirem partículas maiores quando comparadas à curva experimental, possivelmente

devido à escolha de limiar manual, o qual possivelmente superestimou o tamanho dos grãos; e

- o índice de vazios e porosidade encontrados pelo código Python convergiram exatamente para os mesmos valores encontrados pelo software DragonFly, comprovando-se a efetividade do algoritmo e script Python.

Para o compósito de rejeito de minério de ferro com polímero, apesar da impossibilidade de segmentação do elemento polímero nas imagens, foi possível segmentar os elementos de grãos de sílica, minério de ferro e vazios utilizando-se a técnica de algoritmo K-Means. O resultado obtido pelo código Python não teve correlação próxima aos valores encontrados pelo software DragonFly, podendo ser atribuída a justificativa ao algoritmo K-Means: esse algoritmo só possui boa aplicabilidade em casos em que os clusters de intensidade de pixel estejam bem separados e definidos no histograma, o que não foi observado devido às partículas finas presentes na amostra, criando porções de transição de intensidade de pixel e confundindo a escolha automática dos limiares dos clusters.

Entende-se que as definições de configurações setadas no momento prévio à geração das imagens bidimensionais no microtomógrafo podem influenciar na resolução e contraste entre os elementos da amostra, dificultando ou facilitando o pré-processamento das imagens e consequentemente a obtenção dos parâmetros físicos da amostra. Outro fator que influenciou nas análises das amostras foi a aplicação de uma máscara para isolar a área de interesse (amostra dentro da seringa), fazendo com que, equivocadamente, mais um grupo de elementos fosse identificado na imagem, dificultando a aplicação de limiares automáticos.

Foi possível identificar também fatores que influenciam na obtenção dos parâmetros físicos propostos: a supersegmentação pelo algoritmo *Watershed* faz com que haja uma superestimação de partículas menores e a variação na escolha de limiares manuais podem influenciar significativamente nos resultados de índices de vazios ou porosidade.

Dessa forma, a utilização da linguagem Python para a análise de parâmetros morfológicos a partir de imagens bidimensionais de amostras geotécnicas é viável, desde que se considere a especificidade das imagens microtomografadas de cada amostra. É essencial respeitar o limite de resolução, buscar a melhor configuração de contraste para diferenciar os componentes da amostra e realizar um pré-processamento com segmentação representativa desses elementos. Sua aplicação tem potencial para reduzir esforços e tempo, sendo uma alternativa gratuita e facilitadora para pesquisadores, complementando e verificando

parâmetros obtidos por outras metodologias, sejam elas experimentais ou computacionais com outros softwares não-livres.

6. SUGESTÕES PARA PESQUISAS FUTURAS

- Investigar o uso de imagens tridimensionais para uma análise mais abrangente e precisa dos parâmetros morfológicos;
- Desenvolver e implementar algoritmos de segmentação mais avançados para melhorar a identificação e separação dos diferentes componentes das amostras, como técnicas de aprendizado de máquina e inteligência artificial, em especial para amostras de compósitos;
- Realizar estudos comparativos que validem os resultados obtidos com métodos computacionais em Python e técnicas tradicionais de laboratório;
- Estudar a influência nas configurações e resultados ao se considerar a densidade dos diferentes materiais componentes da amostra, suas composições químicas e as macroporosidades e microporosidades;
- Expandir a pesquisa para incluir amostras com diversas composições geotécnicas e condições de carregamento, comparando-se com resultados de outras técnicas não invasivas; e
- Estudar outros parâmetros morfológicos, como tortuosidade, grau de saturação, angularidade etc.

APÊNDICES

A – Código para esfericidade da areia - método por Rittenhouse (1943) e Rorato *et al.* (2019)

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
from scipy import ndimage as ndi
import pandas as pd
import time

# Diretório das imagens
diretorio_das_imagens = 'Informar caminho da pasta com as imagens'

# Função para calcular a área do menor círculo circunscrito a uma partícula
def calcular_area_circunscrita(mascara):
    cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
    if not cnts:
        return 0 # Retorna 0 se não houver contornos
    c = max(cnts, key=cv2.contourArea)
    # Encontrar o menor círculo circunscrito
    ((_, _), raio) = cv2.minEnclosingCircle(c)
    # Calcular a área do círculo
    area_circunscrita = np.pi * (raio ** 2)
    return area_circunscrita

# Função para calcular propriedades das partículas
```

```

def calcular_area_particulas(labels, otsu):
    resultados = []
    for label in np.unique(labels):
        if label == 0:
            continue
        mascara = np.zeros(otsu.shape, dtype='uint8')
        mascara[labels == label] = 255
        cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
        c = max(cnts, key=cv2.contourArea)
        area = cv2.contourArea(c)
        area_circunscrita = calcular_area_circunscrita(mascara)
        perimetro = cv2.arcLength(c, closed=True) # Calcula o perímetro da partícula
        resultados.append((label, area, area_circunscrita, perimetro))
    return resultados

# Função principal
def funcao_esfericidade_do_perimetro(diretorio_das_imagens):
    start_time = time.time() # Inicia o contador de tempo
    dados_resultados = []
    # Lista todos os arquivos no diretório
    arquivos_imagens = os.listdir(diretorio_das_imagens)
    # Itera sobre os arquivos e processa cada imagem
    for arquivo in arquivos_imagens:
        # Verifica se o arquivo é uma imagem (pode adicionar mais extensões se necessário)
        if arquivo.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp')):
            # Constrói o caminho completo para a imagem
            caminho_completo = os.path.join(diretorio_das_imagens, arquivo)
            # Abrindo a imagem
            imagem = Image.open(caminho_completo)
            # Conversão da imagem em "array numpy" para visualização e processamento
            imagem_array = np.array(imagem)
            # Inversão das cores

```

```

inv_imagem_array = 255 - imagem_array

# Desfoque gaussiano
desfoque = cv2.GaussianBlur(inv_imagem_array, (5, 5), 0)
# Método de Otsu - limiar global, ou seja, em toda a imagem, de forma automática
valor, otsu = cv2.threshold(desfoque, 0, 255, cv2.THRESH_BINARY_INV |
cv2.THRESH_OTSU)

# Plotando a matriz de distância
dist = ndi.distance_transform_edt(otsu)
# Localizando os máximos locais na matriz de distância
local_max = peak_local_max(dist, indices=False, min_distance=20, labels=otsu)
# Rotulando os máximos locais
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]
# Aplicando watershed
labels = watershed(-dist, markers, mask=otsu)
# Individualizando as partículas
img_final = labels.copy()
for label in np.unique(labels):
    if label == 0:
        continue
    mascara = np.zeros(otsu.shape, dtype='uint8')
    mascara[labels == label] = 255
    cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
    c = max(cnts, key=cv2.contourArea)
    ((x, y), r) = cv2.minEnclosingCircle(c)
    cv2.putText(img_final, "{}".format(label), (int(x) - 1, int(y)),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                (255, 255, 200), 2)
# Calcular áreas das partículas
resultados_areas = calcular_area_particulas(labels, otsu)
# Adicionar resultados à lista geral
for label, area, area_circunscrita, perimetro in resultados_areas:

```

```

    raio_equivalente = np.sqrt(area / np.pi) # Calcula o raio equivalente
    perimetro_equivalente = 2 * np.sqrt(np.pi * area) # Calcula o perímetro
equivalente
    esfericidade_perimetro = perimetro_equivalente / perimetro # Calcula a
esfericidade do perímetro
    dados_resultados.append((arquivo, label, area, area_circunscrita, raio_equivalente,
perimetro_equivalente, perimetro, esfericidade_perimetro))
    # Converter a lista de dados em um DataFrame do pandas
    df_resultados = pd.DataFrame(dados_resultados, columns=['Nome do Arquivo', 'Número
da Partícula', 'Área', 'Área Circunscrita', 'Raio Equivalente', 'Perímetro Equivalente',
'Perímetro da Partícula', 'Esfericidade do Perímetro'])
    # Especifica o caminho de salvamento do arquivo Excel
    caminho_excel = 'Informar caminho para salvar o arquivo Excel'
    # Exporta o DataFrame para um arquivo Excel
    df_resultados.to_excel(caminho_excel, index=False)
    print(f"O resultado foi salvo em: {caminho_excel}")
    end_time = time.time() # Finaliza o contador de tempo
    duration = end_time - start_time # Calcula o tempo de execução
    print(f"Tempo de execução: {duration:.2f} segundos")

# Exemplo de uso da função
funcao_esfericidade_do_perimetro(diretorio_das_imagens)

```

B – Código para esfericidade do rejeito de minério de ferro - método por Rittenhouse (1943) e Rorato *et al.* (2019)

```
from PIL import Image
import matplotlib.pyplot as plt
import cv2
import numpy as np
from scipy import ndimage as ndi
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
import os
import pandas as pd
import time

# Diretório das imagens
diretorio_das_imagens = 'Informar caminho da pasta com as imagens'
def Esfericidade_do_perimetro_RejeitoFe(diretorio_das_imagens):
    start_time = time.time() # Inicia o contador de tempo

    # Função para calcular propriedades das partículas
    def calcular_area_particulas(labels):
        resultados = []
        for label in np.unique(labels):
            if label == 0:
                continue
            mascara = np.zeros(labels.shape, dtype='uint8')
            mascara[labels == label] = 255
            cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
            c = max(cnts, key=cv2.contourArea)
            area = cv2.contourArea(c)
            perimetro = cv2.arcLength(c, closed=True) # Calcula o perímetro da partícula
            resultados.append((label, area, perimetro))
        return resultados
```

```

# Função para calcular a área e perímetro do menor círculo circunscrito a uma partícula
def calcular_area_circunscrita(mascara):
    cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
    if not cnts:
        return 0, 0 # Retorna 0 se não houver contornos
    c = max(cnts, key=cv2.contourArea)
    # Encontrar o menor círculo circunscrito
    ((_, _), raio) = cv2.minEnclosingCircle(c)
    # Calcular a área do círculo
    area_circunscrita = np.pi * (raio ** 2)
    # Calcular o perímetro do círculo
    perimetro_circunscrito = 2 * np.pi * raio
    return area_circunscrita, perimetro_circunscrito

# Lista para armazenar os dados do DataFrame
dados_dataframe = []

# Listar arquivos no diretório
for filename in os.listdir(diretorio_das_imagens):
    # Verificar se o arquivo é uma imagem (assumindo extensão .bmp)
    if filename.endswith(".bmp"):
        # Caminho completo para a imagem
        path = os.path.join(diretorio_das_imagens, filename)
        # Abrir imagem
        imagem = Image.open(path)
        imagem_array = np.array(imagem)
        # Inverter as cores da imagem
        inv_imagem_array = 255 - imagem_array
        # Aplicar o filtro gaussiano
        desfoque = cv2.GaussianBlur(inv_imagem_array, (5, 5), 0)
        # Aplicar a limiarização

```



```

limiar = 'informar limiarização'
_, thresh = cv2.threshold(desfoque, limiar, 255, cv2.THRESH_BINARY)
# Inverter a imagem binarizada novamente
inv_thresh = 255 - thresh

# Aplicar a matriz de distância
dist = ndi.distance_transform_edt(inv_thresh)
# Encontrar picos locais na matriz de distância
local_max = peak_local_max(dist, min_distance=20, indices=False,
labels=inv_thresh)
# Rotular os picos locais
markers, num_features = ndi.label(local_max)
# Aplicar segmentação de watershed
labels = watershed(-dist, markers, mask=inv_thresh)
# Calcular a área de cada partícula
resultados_areas = calcular_area_particulas(labels)
# Lista para armazenar as áreas dos círculos circunscritas
areas_circunscritas = []
for label in np.unique(labels):
    if label == 0:
        continue
    # Criar uma máscara para a partícula específica
    mascara_particula = np.zeros(inv_thresh.shape, dtype='uint8')
    mascara_particula[labels == label] = 255
    # Calcular a área do menor círculo circunscrito
    area_circunscrita, perimetro_circunscrito =
calcular_area_circunscrita(mascara_particula)
    # Calcular o raio equivalente com base na área da partícula
    raio_equivalente = np.sqrt(resultados_areas[label - 1][1] / np.pi)
    # Calcular o perímetro equivalente com base na área da partícula
    perimetro_equivalente = 2 * np.sqrt(np.pi * resultados_areas[label - 1][1])
    # Adicionar os dados à lista

```

```

        dados_dataframe.append((filename, label, resultados_areas[label - 1][1],
area_circunscrita,
                                perimetro_circunscrito,                                perimetro_equivalente,
resultados_areas[label - 1][2]))
# Criar DataFrame com os dados
df = pd.DataFrame(dados_dataframe, columns=['Nome do Arquivo', 'Rótulo Único', 'Área
da Partícula',
                                'Área do Círculo Circunscrito', 'Perímetro Circunscrito',
                                'Perímetro Equivalente', 'Perímetro da Partícula'])
# Calcular a esfericidade
df['Esfericidade'] = df['Área da Partícula'] / df['Área do Círculo Circunscrito']
# Calcular a esfericidade do perímetro
df['Esfericidade do Perímetro'] = df['Perímetro Equivalente'] / df['Perímetro da Partícula']
# Verificar se há valores zero no denominador
df.loc[df['Perímetro da Partícula'] == 0, 'Esfericidade do Perímetro'] = 'Ruído'
df.loc[df['Área do Círculo Circunscrito'] == 0, 'Esfericidade do Perímetro'] = 'Ruído'
# Especificar o caminho do arquivo Excel de saída
caminho_arquivo_excel = 'Informar caminho para salvar o arquivo Excel'
# Exportar o DataFrame para o arquivo Excel no caminho especificado
df.to_excel(caminho_arquivo_excel, index=False, engine='openpyxl')
end_time = time.time() # Finaliza o contador de tempo
duration = end_time - start_time # Calcula o tempo de execução
print(f"Tempo de execução: {duration:.2f} segundos")

Esfericidade_do_perimetro_RejeitoFe(diretorio_das_imagens)

```

C – Código para granulometria da areia - método por Borges *et al.* (2019)

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
from scipy import ndimage as ndi
import pandas as pd

# Diretório das imagens
diretorio_das_imagens = 'Informar caminho da pasta com as imagens'

# Lista todos os arquivos no diretório
arquivos_imagens = os.listdir(diretorio_das_imagens)

# Lista para armazenar os dados
dados = []

# Fator de conversão de micrômetros para milímetros
fator_conversao = 'Informar tamanho do pixel da imagem em micrômetros' / 1000

# Itera sobre os arquivos e processa cada imagem
for arquivo in arquivos_imagens:
    # Verifica se o arquivo é uma imagem (pode adicionar mais extensões se necessário)
    if arquivo.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp')):
        # Constrói o caminho completo para a imagem
        caminho_completo = os.path.join(diretorio_das_imagens, arquivo)
        # Abrindo a imagem
        imagem = Image.open(caminho_completo)
        # Conversão da imagem em "array numpy" para visualização e processamento
        imagem_array = np.array(imagem)
```

```

# Inversão das cores
inv_imagem_array = 255 - imagem_array
# Desfoque gaussiano
desfoque = cv2.GaussianBlur(inv_imagem_array, (5, 5), 0)
# Método de Otsu - limiar global, ou seja, em toda a imagem, de forma automática
valor, otsu = cv2.threshold(desfoque, 0, 255, cv2.THRESH_BINARY_INV |
cv2.THRESH_OTSU)
# Plotando a matriz de distância
dist = ndi.distance_transform_edt(otsu)
# Localizando os máximos locais na matriz de distância
local_max = peak_local_max(dist, indices=False, min_distance=20, labels=otsu)
# Rotulando os máximos locais
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]
# Aplicando watershed
labels = watershed(-dist, markers, mask=otsu)
# Individualizando as partículas
img_final = labels.copy()

for label in np.unique(labels):
    if label == 0:
        continue
    mascara = np.zeros(otsu.shape, dtype='uint8')
    mascara[labels == label] = 255
    cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
    c = max(cnts, key=cv2.contourArea)
    # Calcular momentos da imagem
    momentos = cv2.moments(c)
    if momentos["m00"] > 0:
        cx = int(momentos["m10"] / momentos["m00"])
        cy = int(momentos["m01"] / momentos["m00"])
    else:
        cx = cy = 0 # Defina cx e cy como 0 se a área for zero

```

```

# Calcular tamanho da menor e maior aresta do retângulo
rotated_rect = cv2.minAreaRect(c)
box = cv2.boxPoints(rotated_rect)
box = np.int0(box)
menor_aresta = min(np.linalg.norm(box[1] - box[0]), np.linalg.norm(box[2] -
box[1]))
maior_aresta = max(np.linalg.norm(box[1] - box[0]), np.linalg.norm(box[2] -
box[1]))

# Converter tamanho das arestas para milímetros
menor_aresta_mm = menor_aresta * fator_conversao
maior_aresta_mm = maior_aresta * fator_conversao

# Calcular área da partícula em pixels
area_pixel = cv2.contourArea(c)

# Converter área da partícula para milímetros quadrados
area_mm2 = area_pixel * (fator_conversao ** 2)

# Adicionar dados à lista
dados.append((arquivo, label, menor_aresta_mm, area_mm2))

# Desenhar o centróide na imagem final
cv2.circle(img_final, (cx, cy), 5, (0, 0, 0), -1) # Desenha um círculo no centróide
# Adicionar número da partícula e coordenadas do centróide
texto = "{} - Centróide: ({} , {})".format(label, cx, cy)
cv2.putText(img_final, texto, (int(cx) - 1, int(cy)),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 200), 2)

# Desenhar o retângulo delimitador rotacionado em vermelho
cv2.drawContours(img_final, [box], 0, (255, 0, 0), thickness=2)

# Exibir a imagem final
#plt.figure(figsize=(5, 5))
#plt.imshow(img_final, cmap='viridis') # Exibir a imagem final com cmap viridis
#plt.axis('off')
#plt.title(arquivo)
#plt.show()

```

```

# Criar DataFrame com os dados
df = pd.DataFrame(dados, columns=['Nome do Arquivo', 'Número da Partícula', 'Menor
Aresta (mm)', 'Área (mm²)'])
# Ordenar o DataFrame com base na coluna 'Menor Aresta (mm)'
df = df.sort_values(by='Menor Aresta (mm)')
# Calcular a soma acumulada da coluna 'Área (mm²)'
df['Área Acumulada (mm²)'] = df['Área (mm²)'].cumsum()
# Calcular o maior valor da coluna 'Área acumulada (mm²)'
max_area_acumulada = df['Área Acumulada (mm²)'].max()
# Calcular a porcentagem da área acumulada
df['Área Acumulada em %'] = (df['Área Acumulada (mm²)'] / max_area_acumulada) * 100
# Exibir DataFrame
print(df)

# Ordenar o DataFrame pela coluna 'Menor Aresta (mm)'
df_sorted = df.sort_values(by='Menor Aresta (mm)')
# Plotar o gráfico em escala logarítmica com grades verticais
plt.figure(figsize=(10, 6))
plt.plot(df_sorted['Menor Aresta (mm)'], df_sorted['Área Acumulada em %'], marker='o',
linestyle='--')
plt.xscale('log') # Definir escala logarítmica no eixo x
plt.xlabel('Menor Aresta (mm)')
plt.ylabel('Área Acumulada em %')
plt.title('Gráfico de Menor Aresta vs. Área Acumulada em % em Escala Logarítmica com
Grades Verticais')
plt.grid(True, which='both', axis='x', linestyle='--') # Adicionar grades verticais
plt.grid(True, axis='y') # Adicionar grades horizontais
plt.show()

# Especificar o caminho do arquivo Excel
caminho_arquivo_excel = 'Informar caminho para salvar o arquivo Excel'
# Exportar o DataFrame para o arquivo Excel
df.to_excel(caminho_arquivo_excel, index=False)

```

```
print(f'DataFrame exportado para {caminho_arquivo_excel} com sucesso.')
```

**D – Código para granulometria do rejeito de minério de ferro - método por
Borges *et al.* (2019)**

```
import time
# Iniciar contagem de tempo
start_time = time.time()

import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
from scipy import ndimage as ndi
import pandas as pd

# Diretório das imagens
diretorio_das_imagens = 'Informar caminho da pasta com as imagens'

# Lista todos os arquivos no diretório
arquivos_imagens = os.listdir(diretorio_das_imagens)

# Lista para armazenar os dados
dados = []

# Fator de conversão de micrômetros para milímetros
fator_conversao = 'Informar tamanho do pixel da imagem em micrômetros' / 1000

# Itera sobre os arquivos e processa cada imagem
for arquivo in arquivos_imagens:
    # Verifica se o arquivo é uma imagem (pode adicionar mais extensões se necessário)
    if arquivo.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp')):
        # Constrói o caminho completo para a imagem
```



```

caminho_completo = os.path.join(diretorio_das_imagens, arquivo)

# Abrindo a imagem
imagem = Image.open(caminho_completo)

# Conversão da imagem em "array numpy" para visualização e processamento
imagem_array = np.array(imagem)

# Inversão das cores
inv_imagem_array = 255 - imagem_array

# Desfoque gaussiano
desfoque = cv2.GaussianBlur(inv_imagem_array, (5, 5), 0)

# Aplicar a limiarização
limiar = 225
_, thresh = cv2.threshold(desfoque, limiar, 255, cv2.THRESH_BINARY)

# Inverter a imagem binarizada novamente
inv_thresh = 255 - thresh

# Plotando a matriz de distância
dist = ndi.distance_transform_edt(inv_thresh)

# Localizando os máximos locais na matriz de distância
local_max = peak_local_max(dist, indices=False, min_distance=20, labels=inv_thresh)

# Rotulando os máximos locais
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]

# Aplicando watershed
labels = watershed(-dist, markers, mask=inv_thresh)

# Individualizando as partículas
img_final = labels.copy()

for label in np.unique(labels):
    if label == 0:
        continue
    mascara = np.zeros(inv_thresh.shape, dtype='uint8')
    mascara[labels == label] = 255
    cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
    c = max(cnts, key=cv2.contourArea)

```

```

# Calcular momentos da imagem
momentos = cv2.moments(c)
# Calcular o centróide (x, y) apenas se a área da partícula for maior que zero
if momentos["m00"] > 0:
    cx = int(momentos["m10"] / momentos["m00"])
    cy = int(momentos["m01"] / momentos["m00"])
else:
    cx = cy = 0 # Defina cx e cy como 0 se a área for zero
# Calcular tamanho da menor e maior aresta do retângulo
rotated_rect = cv2.minAreaRect(c)
box = cv2.boxPoints(rotated_rect)
box = np.int0(box)
menor_aresta = min(np.linalg.norm(box[1] - box[0]), np.linalg.norm(box[2] -
box[1]))
maior_aresta = max(np.linalg.norm(box[1] - box[0]), np.linalg.norm(box[2] -
box[1]))
# Converter tamanho das arestas para milímetros
menor_aresta_mm = menor_aresta * fator_conversao
maior_aresta_mm = maior_aresta * fator_conversao
# Calcular área da partícula em pixels
area_pixel = cv2.contourArea(c)
# Converter área da partícula para milímetros quadrados
area_mm2 = area_pixel * (fator_conversao ** 2)
# Adicionar dados à lista
dados.append((arquivo, label, menor_aresta_mm, area_mm2))
# Desenhar o centróide na imagem final
cv2.circle(img_final, (cx, cy), 5, (0, 0, 0), -1) # Desenha um círculo no centróide
# Adicionar número da partícula e coordenadas do centróide
texto = "{} - Centróide: ( {}, {} )".format(label, cx, cy)
cv2.putText(img_final, texto, (int(cx) - 1, int(cy)),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 200), 2)
# Desenhar o retângulo delimitador rotacionado em vermelho

```

```

cv2.drawContours(img_final, [box], 0, (255, 0, 0), thickness=2)

# Exibir a imagem final
#plt.figure(figsize=(5, 5))
#plt.imshow(img_final, cmap='viridis') # Exibir a imagem final com cmap viridis
#plt.axis('off')
#plt.title(arquivo)
#plt.show()

# Criar DataFrame com os dados
df = pd.DataFrame(dados, columns=['Nome do Arquivo', 'Número da Partícula', 'Menor
Aresta (mm)', 'Área (mm²)'])
# Ordenar o DataFrame com base na coluna 'Menor Aresta (mm)'
df = df.sort_values(by='Menor Aresta (mm)')
# Calcular a soma acumulada da coluna 'Área (mm²)'
df['Área Acumulada (mm²)'] = df['Área (mm²)'].cumsum()
# Calcular o maior valor da coluna 'Área acumulada (mm²)'
max_area_acumulada = df['Área Acumulada (mm²)'].max()
# Calcular a porcentagem da área acumulada
df['Área Acumulada em %'] = (df['Área Acumulada (mm²)'] / max_area_acumulada) * 100
# Exibir DataFrame
#print(df)

# Finalizar contagem de tempo
end_time = time.time()

# Calcular tempo de execução
execution_time = end_time - start_time

# Exibir tempo de execução
print(f"Tempo de execução: {execution_time} segundos")

# Ordenar o DataFrame pela coluna 'Menor Aresta (mm)'
df_sorted = df.sort_values(by='Menor Aresta (mm)')

```

```
# Plotar o gráfico em escala logarítmica com grades verticais
plt.figure(figsize=(10, 6))
plt.plot(df_sorted['Menor Aresta (mm)'], df_sorted['Área Acumulada em %'], marker='o',
linestyle='-')
plt.xscale('log') # Definir escala logarítmica no eixo x
plt.xlabel('Menor Aresta (mm)')
plt.ylabel('Área Acumulada em %')
plt.title('Gráfico de Menor Aresta vs. Área Acumulada em % em Escala Logarítmica com
Grades Verticais')
plt.grid(True, which='both', axis='x', linestyle='--') # Adicionar grades verticais
plt.grid(True, axis='y') # Adicionar grades horizontais
plt.show()

# Especificar o caminho do arquivo Excel
caminho_arquivo_excel = 'Informar caminho para salvar o arquivo Excel'
# Exportar o DataFrame para o arquivo Excel
df.to_excel(caminho_arquivo_excel, index=False)
print(f'DataFrame exportado para {caminho_arquivo_excel} com sucesso.')
```

**E – Código para índice de vazios e porosidade da areia - método por Borges
et al. (2019)**

```
import time

# Iniciar contagem de tempo
start_time = time.time()

import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
from scipy import ndimage as ndi
import pandas as pd

# Diretório das imagens
diretorio_das_imagens = 'Informar caminho da pasta com as imagens'

# Lista todos os arquivos no diretório
arquivos_imagens = os.listdir(diretorio_das_imagens)

# Lista para armazenar os dados
dados = []

# Fator de conversão de micrômetros para milímetros
fator_conversao = 'Informar tamanho do pixel da imagem em micrômetros' / 1000

# Itera sobre os arquivos e processa cada imagem
for arquivo in arquivos_imagens:
    # Verifica se o arquivo é uma imagem (pode adicionar mais extensões se necessário)
```

```

if arquivo.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp')):      # Constrói o caminho
completo para a imagem
    caminho_completo = os.path.join(diretorio_das_imagens, arquivo)
    # Abrindo a imagem
    imagem = Image.open(caminho_completo)
    # Conversão da imagem em "array numpy" para visualização e processamento
    imagem_array = np.array(imagem)
    # Inversão das cores
    inv_imagem_array = 255 - imagem_array
    # Desfoque gaussiano
    desfoque = cv2.GaussianBlur(inv_imagem_array, (5, 5), 0)
    # Método de Otsu - limiar global, ou seja, em toda a imagem, de forma automática
    valor, otsu = cv2.threshold(desfoque, 0, 255, cv2.THRESH_BINARY_INV |
cv2.THRESH_OTSU)
    # Plotando a matriz de distância
    dist = ndi.distance_transform_edt(otsu)
    # Localizando os máximos locais na matriz de distância
    local_max = peak_local_max(dist, indices=False, min_distance=20, labels=otsu)
    # Rotulando os máximos locais
    markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]
    # Aplicando watershed
    labels = watershed(-dist, markers, mask=otsu)
    # Individualizando as partículas
    img_final = labels.copy()
    # Área total de todas as partículas do arquivo
    area_total_arquivo = 0

for label in np.unique(labels):
    if label == 0:
        continue
    mascara = np.zeros(otsu.shape, dtype='uint8')
    mascara[labels == label] = 255

```

```

cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]

for c in cnts:
    # Calcular área da partícula em pixels
    area_pixel = cv2.contourArea(c)
    # Calcular área da partícula em mm2
    area_mm2_particula = area_pixel * (fator_conversao ** 2)
    # Adicionar área da partícula à área total do arquivo
    area_total_arquivo += area_mm2_particula

# Converter a imagem final para encontrar contornos (CV_8UC1)
img_final_gray = (img_final.copy() * 255).astype(np.uint8)
# Encontrar os contornos das partículas na imagem final
contours, _ = cv2.findContours(img_final_gray, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

# Calcular o menor círculo que circunscreve todos os contornos
(x, y), radius = cv2.minEnclosingCircle(np.concatenate(contours))
# Calcular a área do menor círculo circunscrito em mm2
area_circulo_mm2 = np.pi * (radius * fator_conversao)**2
# Área de vazios
area_vazios_mm2 = area_circulo_mm2 - area_total_arquivo

# Índice de vazios
indice_vazios_na_slice = area_vazios_mm2 / area_total_arquivo if area_total_arquivo
!= 0 else 0

# Porosidade
porosidade_na_slice = area_vazios_mm2 / area_circulo_mm2

# Adicionar os dados à lista
dados.append({
    'Nome do arquivo': arquivo,
    'Área do menor círculo circunscrito (mm2): area_circulo_mm2,

```

```
'Área total de todas as partículas do arquivo (mm²)': area_total_arquivo,  
'Área de vazios (mm²)': area_vazios_mm2,  
'Índice de vazios na slice': indice_vazios_na_slice,  
'Porosidade na slice': porosidade_na_slice  
})  
  
# Criar DataFrame a partir dos dados  
df = pd.DataFrame(dados)  
# Exibir DataFrame  
print(df)  
  
# Calcular e imprimir o Índice de vazios da amostra  
indice_vazios_amostra = df['Índice de vazios na slice'].mean()  
print(f'Índice de vazios da amostra: {indice_vazios_amostra}')  
# Calcular e imprimir a Porosidade da amostra  
porosidade_amostra = df['Porosidade na slice'].mean()  
print(f'Porosidade da amostra: {porosidade_amostra}')  
  
# Finalizar contagem de tempo  
end_time = time.time()  
  
# Calcular tempo de execução  
execution_time = end_time - start_time  
  
# Exibir tempo de execução  
print(f"Tempo de execução: {execution_time} segundos")
```


**F – Código para índice de vazios e porosidade do rejeito de minério de ferro
- método por Borges *et al.* (2019)**

```
import time

# Iniciar contagem de tempo
start_time = time.time()

import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from skimage.feature import peak_local_max
from skimage.segmentation import watershed
from scipy import ndimage as ndi
import pandas as pd

# Diretório das imagens
diretorio_das_imagens = 'Informar caminho da pasta com as imagens'

# Lista todos os arquivos no diretório
arquivos_imagens = os.listdir(diretorio_das_imagens)

# Lista para armazenar os dados
dados = []

# Fator de conversão de micrômetros para milímetros
fator_conversao = 'Informar tamanho do pixel da imagem em micrômetros' / 1000

# Itera sobre os arquivos e processa cada imagem
for arquivo in arquivos_imagens:
    # Verifica se o arquivo é uma imagem (pode adicionar mais extensões se necessário)
    if arquivo.lower().endswith(('png', 'jpg', 'jpeg', 'bmp')):
```

```

# Constrói o caminho completo para a imagem
caminho_completo = os.path.join(diretorio_das_imagens, arquivo)
# Abrindo a imagem
imagem = Image.open(caminho_completo)
# Conversão da imagem em "array numpy" para visualização e processamento
imagem_array = np.array(imagem)
# Inversão das cores
inv_imagem_array = 255 - imagem_array
# Desfoque gaussiano
desfoque = cv2.GaussianBlur(inv_imagem_array, (5, 5), 0)
# Aplicar a limiarização
limiar = 225
_, thresh = cv2.threshold(desfoque, limiar, 255, cv2.THRESH_BINARY)
# Inverter a imagem binarizada novamente
inv_thresh = 255 - thresh
# Plotando a matriz de distância
dist = ndi.distance_transform_edt(inv_thresh)
# Localizando os máximos locais na matriz de distância
local_max = peak_local_max(dist, indices=False, min_distance=20, labels=inv_thresh)
# Rotulando os máximos locais
markers = ndi.label(local_max, structure=np.ones((3, 3)))[0]
# Aplicando watershed
labels = watershed(-dist, markers, mask=inv_thresh)
# Individualizando as partículas
img_final = labels.copy()

# Plotar a imagem final
#plt.imshow(dist, cmap='gray')
#plt.title('Imagem Final')
#plt.colorbar()
#plt.show()

# Área total de todas as partículas do arquivo

```

```

area_total_arquivo = 0
for label in np.unique(labels):
    if label == 0:
        continue
    mascara = np.zeros(inv_thresh.shape, dtype='uint8')
    mascara[labels == label] = 255
    cnts = cv2.findContours(mascara.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)[-2]
    for c in cnts:
        # Calcular área da partícula em pixels
        area_pixel = cv2.contourArea(c)
        # Calcular área da partícula em mm2
        area_mm2_particula = area_pixel * (fator_conversao ** 2)
        # Adicionar área da partícula à área total do arquivo
        area_total_arquivo += area_mm2_particula
    # Converter a imagem final para o formato correto (CV_8UC1)
    img_final_gray = (img_final.copy() * 255).astype(np.uint8)
    # Encontrar os contornos das partículas na imagem final
    contours, _ = cv2.findContours(img_final_gray, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    # Calcular o menor círculo que circunscreve todos os contornos
    (x, y), radius = cv2.minEnclosingCircle(np.concatenate(contours))
    # Calculando a área do menor círculo circunscrito em mm2 que englobe todas as
partículas
    area_circulo_mm2 = np.pi * (radius * fator_conversao)**2
    # Área de vazios
    area_vazios_mm2 = area_circulo_mm2 - area_total_arquivo
    # Índice de vazios
    indice_vazios_na_slice = area_vazios_mm2 / area_total_arquivo if area_total_arquivo
!= 0 else 0
    # Porosidade
    porosidade_na_slice = area_vazios_mm2 / area_circulo_mm2

```

```

# Adicionar os dados à lista
dados.append({
    'Nome do arquivo': arquivo,
    'Área do menor círculo circunscrito (mm2) a todas as partículas': area_circulo_mm2,
    'Área total de todas as partículas do arquivo (mm2)': area_total_arquivo,
    'Área de vazios (mm2)': area_vazios_mm2,
    'Índice de vazios na slice': indice_vazios_na_slice,
    'Porosidade na slice': porosidade_na_slice
})

# Criar DataFrame a partir dos dados
df = pd.DataFrame(dados)
# Exibir DataFrame
print(df)

# Calcular e imprimir o Índice de vazios da amostra
indice_vazios_amostra = df['Índice de vazios na slice'].mean()
print(f'Índice de vazios da amostra: {indice_vazios_amostra}')
# Calcular e imprimir a Porosidade da amostra
porosidade_amostra = df['Porosidade na slice'].mean()
print(f'Porosidade da amostra: {porosidade_amostra}')

# Finalizar contagem de tempo
end_time = time.time()
# Calcular tempo de execução
execution_time = end_time - start_time

# Exibir tempo de execução
print(f"Tempo de execução: {execution_time} segundos")

```

G – Código para índice de vazios e porosidade do compósito de rejeito de minério de ferro com polímero - método por Borges *et al.* (2019)

```
import os
import cv2
import numpy as np
import pandas as pd
from PIL import Image
from sklearn.cluster import KMeans
import time
import matplotlib.pyplot as plt

# Definir funções para calcular o menor círculo envolvente e aplicar a máscara
def calcular_menor_circulo_envolvente(cluster_image):
    # Encontrar os índices dos pixels brancos na imagem binária
    white_pixels_y, white_pixels_x = np.nonzero(cluster_image == 255)
    # Verificar se há pixels brancos na imagem binária
    if len(white_pixels_x) == 0:
        return None, None, None
    # Encontrar os pontos extremos (mínimo e máximo) em cada dimensão (x e y)
    min_x = np.min(white_pixels_x)
    max_x = np.max(white_pixels_x)
    min_y = np.min(white_pixels_y)
    max_y = np.max(white_pixels_y)
    # Calcular o centro e o raio do círculo envolvente
    center = ((min_x + max_x) // 2, (min_y + max_y) // 2)
    radius = max((max_x - min_x), (max_y - min_y)) // 2
    # Calcular a área do círculo
    area = np.pi * radius ** 2
    return center, radius, area

def aplicar_mascara(imagem, mascara):
    return cv2.bitwise_and(imagem, imagem, mask=mascara)
```

```

# Tempo inicial
start_time = time.time()

# Diretório onde estão armazenadas as imagens
diretorio_imagens = 'Informar caminho da pasta com as imagens'

# Lista para armazenar os resultados
resultados = []

# Dicionário para armazenar as imagens binárias
binary_images_dict = {}

# Dicionário para armazenar as máscaras
mascaras_dict = {}

# Dicionário para armazenar a área total da amostra por arquivo
area_total_amostra_dict = {}

# Iterar sobre as imagens no diretório
for imagem_nome in os.listdir(diretorio_imagens):
    # Abrir a imagem
    imagem = Image.open(os.path.join(diretorio_imagens, imagem_nome))
    # Converter a imagem em array numpy
    imagem_array = np.array(imagem)
    # Inverter a imagem
    inv_imagem_array = 255 - imagem_array
    # Redimensionar a imagem para uma matriz unidimensional
    reshaped_image = inv_imagem_array.reshape((-1, 1))
    # Aplicar o algoritmo k-means com 3 clusters e definir n_init explicitamente para suprimir
o aviso
    kmeans = KMeans(n_clusters=3, n_init=10)
    kmeans.fit(reshaped_image)
    cluster_labels = kmeans.labels_
    # Obter os rótulos únicos
    unique_labels = np.unique(cluster_labels)

```

```

# Lista para armazenar as áreas dos clusters
areas_menor_circulo = []

# Iterar sobre os rótulos únicos
for i in unique_labels:
    # Criar uma imagem binária para o cluster atual
    binary_image = np.where(cluster_labels == i, 255, 0).astype(np.uint8)
    # Remodelar a imagem binária para ter duas dimensões
    binary_image = binary_image.reshape(inv_imagem_array.shape)
    # Salvar a imagem binária no dicionário
    binary_images_dict[f"{imagem_nome}_cluster_{i}"] = binary_image
    # Calcular a área total em pixels
    total_area = binary_image.size
    # Calcular a área total dos pixels brancos
    area_branca = np.sum(binary_image == 255)
    # Adicionar resultados à lista
    resultados.append([imagem_nome, f"{imagem_nome}_cluster_{i}", total_area,
area_branca])
    # Calcular o menor círculo envolvente para o cluster atual
    center, radius, area = calcular_menor_circulo_envolvente(binary_image)
    if center is not None and radius is not None and area is not None:
        areas_menor_circulo.append(area)
    # Criar a máscara do menor círculo envolvente
    mask = np.zeros_like(inv_imagem_array)
    center = tuple(map(int, center)) # Converter os valores para int
    cv2.circle(mask, center, int(radius), (255, 255, 255), -1)
    # Salvar a máscara no dicionário
    mascaras_dict[f"{imagem_nome}_cluster_{i}"] = mask

# Encontrar o menor círculo envolvente entre os três clusters
menor_centro, menor_raio, menor_area =
calcular_menor_circulo_envolvente(binary_images_dict[f"{imagem_nome}_cluster_{np.ar
gmin(areas_menor_circulo)}"])

```

```

# Calcular a área total da amostra para o arquivo atual
area_total_amostra_dict[imagem_nome] = menor_area
# Criar a máscara do menor círculo envolvente
mask = np.zeros_like(inv_imagem_array)
menor_centro = tuple(map(int, menor_centro)) # Converter os valores para int
cv2.circle(mask, menor_centro, int(menor_raio), (255, 255, 255), -1)
# Salvar a máscara no dicionário para todos os clusters do arquivo atual
for i in unique_labels:
    mascaras_dict[f"{imagem_nome}_cluster_{i}"] = mask

# Exibir o DataFrame com os resultados
df_resultados = pd.DataFrame(resultados, columns=['Nome do Arquivo', 'Nome do Arquivo
+ Cluster', 'Área Total', 'Área Branca'])
for index, row in df_resultados.iterrows():
    imagem_nome = row['Nome do Arquivo']
    cluster_index = row['Nome do Arquivo + Cluster'][-1] # Obtém o índice do cluster do
nome
    binary_image = binary_images_dict[f"{imagem_nome}_cluster_{cluster_index}"]
    primeiro_pixel = binary_image[0][0] # Obtém o primeiro pixel da imagem binária
    if primeiro_pixel == 255:
        df_resultados.at[index, 'Elemento'] = 'Vazios'
    elif row['Área Branca'] == df_resultados[df_resultados['Nome do Arquivo'] ==
imagem_nome]['Área Branca'].min():
        df_resultados.at[index, 'Elemento'] = 'Ferro'
    else:
        df_resultados.at[index, 'Elemento'] = 'Grãos'

# Adicionar a coluna "Área Total Amostra" ao DataFrame
df_resultados['Área Total Amostra'] = df_resultados['Nome do
Arquivo'].map(area_total_amostra_dict)
# Adicionar a coluna "Área do Elemento" ao DataFrame
areas_elemento = []
for key, mask in mascaras_dict.items():

```



```

binary_image = binary_images_dict[key]
area_elemento = np.sum(cv2.bitwise_and(binary_image, binary_image, mask=mask) ==
255)
areas_elemento.append(area_elemento)
df_resultados['Área do Elemento'] = areas_elemento

# Adicionar a coluna "Porcentagem" ao DataFrame
df_resultados['Porcentagem'] = df_resultados['Área do Elemento'] / df_resultados['Área
Total Amostra'] * 100

# Plotar as imagens cluster e cluster com máscara
for key, binary_image in binary_images_dict.items():
    imagem_nome, cluster = key.split('_cluster_')
    cluster_index = int(cluster)

    # Plotar a imagem binária do cluster
    #plt.figure(figsize=(5, 5))
    #plt.imshow(binary_image, cmap='gray')
    #plt.title(f'{imagem_nome}_cluster_{cluster_index}') # Adicionar o nome do arquivo e
número do cluster ao título
    #plt.axis('off')
    #plt.show()

    # Plotar a imagem binária do cluster com a máscara
    #mask = mascaras_dict[key]
    #masked_image = cv2.bitwise_and(binary_image, binary_image, mask=mask)
    #plt.figure(figsize=(5, 5))
    #plt.imshow(masked_image, cmap='gray')
    #plt.title(f'{imagem_nome}_cluster_{cluster_index} com máscara') # Adicionar o nome
do arquivo e número do cluster ao título
    #plt.axis('off')
    #plt.show()

```

```
print(df_resultados)

# Calcular e imprimir a média da porcentagem para cada elemento
media_por_elemento = df_resultados.groupby('Elemento')['Porcentagem'].mean()
print("\nMédia da Porcentagem para cada Elemento:")
print(media_por_elemento)

# Tempo final
end_time = time.time()

# Exibir o tempo de execução
print(f"\nTempo de execução: {end_time - start_time} segundos")
```

REFERÊNCIAS BIBLIOGRÁFICAS

Al-Raoush R.I. & Willson C.S. (2005). *Extraction of physically realistic pore network properties from three-dimensional synchrotron X-ray microtomography images of unconsolidated porous media systems*. Journal of Hydrology, 300 (1–4): 44-64. <https://doi.org/10.1016/j.jhydrol.2004.05.005>

Araújo, G. S., Bicalho, K. V. & Tristão, F. A. (2015). *Análise de Imagens na Determinação da Forma e Textura de Areias*. Revista Brasileira de Ciência do Solo, 39(1):94-99. DOI: 10.1590/01000683rbc20150274.

Augusto, K. S. (2016). *Microtomografia Computadorizada de Raios X Aplicada à Caracterização de Porosidade em Pelotas de Minério de Ferro*. Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro, RJ, 156p.

Aydın, M. & Kurnaz, T.F. (2022). *An alternative method for the particle size distribution: Image processing*. Turkish Journal of Engineering, 7(2), 108-115. <https://doi.org/10.31127/tuje.1053462>

Barman, U. & Choudhury, R.D. (2020). *Soil texture classification using multi class support vector machine*. Information Processing in Agriculture, 7 (2): 318-332. <https://doi.org/10.1016/j.inpa.2019.08.001>

Birmpilis, G., Mohammadi, A.S., Villanova, J., Boller, E., Ando, E. & Dijkstra, J. (2022). *Fabric Investigation of Natural Sensitive Clay from 3D Nano- and Microtomography Data*. Journal of Engineering Mechanics, 148 (2). [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0002044](https://doi.org/10.1061/(ASCE)EM.1943-7889.0002044).

Birmpilis, G., Mohammadi, A.S., Villanova, J., Boller, E., Ando, E. & Dijkstra, J. (2022). *Fabric Investigation of Natural Sensitive Clay from 3D Nano- and Microtomography Data*. Journal of Engineering Mechanics, 48 (2). [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0002044](https://doi.org/10.1061/(ASCE)EM.1943-7889.0002044)

Boerckel, J. D., Mason, D. E., Mcdermott, A. M. & Alsberg, E. (2014) *Microcomputed tomography: approaches and applications in bioengineering*. StemCell Research & Therapy, 5 (6): 144, 2014. DOI: 10.1186/scrt534

Borges L.P.F., Moraes R. M., Crestana S. & Cavalcante A. L. B. (2019). *Geometric Features and Fractal Nature of Soil Analyzed by X-Ray Microtomography Image Processing*. International Journal of Geomechanics, 19 (8): 1–14. DOI: 10.1061/(ASCE)GM.1943-5622.0001464.

Carl Zeiss Microscopy, 2023. Zeiss Xradia 520 Versa. Acessado em: 15/11/2023. <<https://www.zeiss.com/microscopy/en/products/x-ray-microscopy/xradia-versa.html#overlay-benefits-2119365364>>.

Carneiro, A. A. (2020) *Comportamento Mecânico de um Rejeito de Minério de Ferro Estabilizado com Polímero e do Compósito Rejeito-Polímero Reforçado com Fibras de Polipropileno*. Tese de Doutorado, Programa de Pós-graduação em Geotecnia, Universidade de Brasília, Brasília, DF, 164 p.

Casagrande, M.D.T. (2005). *Comportamento de solos reforçados com fibras submetidos a grandes deformações*. Tese de Doutorado. Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 219 p.

Cheng, Z., Wang, J., Coop, M. R. & YE, G. (2020). *A miniature triaxial apparatus for investigating the micromechanics of granular soils with in situ X-ray micro-tomography scanning*. Frontiers of Structural and Civil Engineering, 14 (2): 357–373 <https://doi.org/10.1007/s11709-019-0599-2>

Cheng, Z., Wang, J., Coop, M.R. & Ye, G. (2019) *A miniature triaxial apparatus for investigating the micromechanics of granular soils with in situ X-ray micro-tomography scanning*. Frontiers of Structural and Civil Engineering, 14: 357-373. <https://doi.org/10.1007/s11709-019-0599-2>

Cho, G. C, Dodds, J. & Santamarina, J. C. (2006). *Particle Shape Effects on Packing Density, Stiffness, and Strength: Natural and Crushed Sands*. Journal of Geotechnical and Geoenvironmental Engineering, 132(5):591-602. DOI: 10.1061/(ASCE)1090-0241(2006)132:5(591)

Clark, D. P. & Badea, C. T. (2014) *Micro-CT of rodents: state-of-the-art and future perspectives*. Physica Medica, 30 (6): 619-634. <https://doi.org/10.1016/j.ejmp.2014.05.011>

Crestana, S., Mascarenhas, S. & Pozzi-Mucelli, R.S., (1985). *Static and Dynamic Three-Dimensional Studies of Water in Soil using Computed Tomographic Scanning*. Soil Science 140(1), 326–332.

Dipova, N. (2017). *Determining the Grain Size Distribution of Granular Soils Using Image Analysis*. Geotechnica Slovenica, 14: 29-37

Durand, R., Farias, M.M. & Carvalho, J. C. (2012). *Análise numérica de processos de infiltração em mesoescala. Tópicos sobre infiltração: teoria e prática aplicada a solos tropicais*. Faculdade de Tecnologia, Universidade de Brasília. Brasília, DF., 287-308.

Enríquez-León, A. J. (2021) *Caracterização Microestrutural de Matrizes Asfálticas de Agregados Finos a partir da Microtomografia Computadorizada*. Dissertação de mestrado. COPPE/UFRJ, Rio de Janeiro, RJ, 86 p.

Evans, C. L., Wightman, E. M. & Yuan, X. (2015). *Quantifying mineral grain size distributions for process modelling using X-ray micro-tomography*. Minerals Engineering, 82: 78-83. <http://dx.doi.org/10.1016/j.mineng.2015.03.026>

Faroughi, S. & Huber, C. (2016). *A theoretical hydrodynamic modification on the soil texture analyses obtained from the hydrometer test*. Géotechnique, 66 (5): 378–385. <https://doi.org/10.1680/jgeot.14.P.267>

Fopa, R.D., Bianco, C., Archilla, N. L., Moreira A.C. & Pak T. (2023). *A pore-scale investigation of the effect of nanoparticle injection on properties of sandy porous media*. Journal of Contaminant Hydrology, 253. <https://doi.org/10.1016/j.jconhyd.2022.104126>

Grupo Computação Visual Aplicada – COVAP (2021). *Material introdutório de Processamento Digital de Imagens e Visão Computacional*. Laboratório de Computação Visual Aplicada - Universidade Tecnológica Federal do Paraná. Disponível em: <https://covap-utfpr.github.io/pdi/index.html>

Hao, L., Hui, Z., Zhiwei, T. & Guangshu, H. (2008) *Micro-computed tomography for small animal imaging: Technological details*. Progress in Natural Science, 18 (5): 513-521. <https://doi.org/10.1016/j.pnsc.2008.01.002>

Kim, D.; Song, J. & Park, J. (2023). *Quantifying Soil Particle Settlement Characteristics through Machine Vision Analysis Utilizing an RGB Camera*. Agriculture, 13 (9):1674. <https://doi.org/10.3390/agriculture13091674>

Krumbein, W. C. (1941). *Measurement and geological significance of shape and roundness of sedimentary particles*. Journal of Sedimentary Research, 11 (2): 64–72. <https://doi.org/10.1306/D42690F3-2B26-11D7-8648000102C1865D>

Krumbein, W.C. & Sloss, L.L. (1963). *Stratigraphy and Sedimentation*, Segunda edição.

Lacasse, S. & Nadim, F. (1996). *Uncertainties in Characterizing Soil Properties* – Plenary paper. In Proceedings of ASCE Special Technical Publication No. 58: Uncertainty in the Geologic Environment - From Theory to Practice. Madison, Wisconsin, USA, 1: 49–75.

Lakkimsetti, B. & Latha G. M. (2023). *Role of grain size and shape on undrained monotonic shear, liquefaction, and post-liquefaction behaviour of granular ensembles*. Soil Dynamics and Earthquake Engineering, 173. <https://doi.org/10.1016/j.soildyn.2023.108086>.

Landis, E.N. & Keane, D.T. (2010) *X-ray microtomography*. Materials Characterization, 61 (12): 1305-1316. <https://doi.org/10.1016/j.matchar.2010.09.012>

Mendes J.C. (1972). *Estratigrafia e sedimentologia: Geologia estrutural e aerofotogeologia*. Brasília: Ministério da Educação e da Cultura/Instituto Nacional do Livro.

Mendes, L.V.R. & Melo, T.M. (2011). *Avaliação da Esfericidade como um Fator de Forma na Interação Partícula-Fluido*. Bacharelado em Engenharia Química. Universidade Federal do Rio de Janeiro.

Mora C. F. & Kwan A. K. H. (2000). *Sphericity, shape factor and convexity measurement of coarse aggregate for concrete using digital image processing*. Cement and Concrete Research, 30 (3): 351-358. [https://doi.org/10.1016/S0008-8846\(99\)00259-8](https://doi.org/10.1016/S0008-8846(99)00259-8)

Morgenstern, N. R. (1995). *Managing Risk in Geotechnical Engineering*. Proceeding of the 10th Pan American Conference on Soil Mechanics and Foundation Engineering, Guadalajara, Mexico, 4: 102-126.

Moscariello, M., Salager, S. & Cuomo, S. (2016). *X-ray Computed Tomography for Capillary Collapse of Loose Unsaturated Sand*. Procedia Engineering, 158: 33-38. <https://doi.org/10.1016/j.proeng.2016.08.401>

Oboudi, M., Pietruszczak, S. and Razaqpur, A.G. (2016). *Description of Inherent and Induced Anisotropy in Granular Media with Particles of High Sphericity*. International Journal of Geomechanics, 16 (4). [https://doi.org/10.1061/\(ASCE\)GM.1943-5622.0000635](https://doi.org/10.1061/(ASCE)GM.1943-5622.0000635)

Pae, J. & Moon, J. (2020) *Evaluation of Tortuosity in Cemented Sand Using X-Ray Computed Microtomography*. XV International Conference on Durability of Building Materials and Components. DOI: 10.23967/dbmc.2020.206

Palombo, L. A. (2017). *Microtomografia de raios X e a porosimetria por intrusão de mercúrio na determinação de porosidade e densidade de rochas reservatório*. Dissertação de Mestrado. Universidade de São Paulo, São Paulo, SP, 88p.

Payan, M., K. Senetakis, A. Khoshghalb, & N. Khalili (2017). “*Effect of gradation and particle shape on small-strain Young’s modulus and Poisson’s ratio of sands.*” International Journal of Geomechanics, 17 (5). [https://doi.org/10.1061/\(ASCE\)GM.1943-5622.0000811](https://doi.org/10.1061/(ASCE)GM.1943-5622.0000811)

Petrovic, A.M., Siebert, J.E., Rieke, P.E., (1982). *Soil Bulk Density in Three Dimensions by Computed Tomographic Scanning*. Soil Science Society of America Journal 46(3), 445–450.

Pinto, C. S. (2006). *Curso Básico de Mecânica dos Solos*. Oficina de textos, 3ª edição. São Paulo.

Ribeiro, S. & Bonetti, C. (2013). *Variabilidade Morfométrica de Sedimentos Arenosos: Revisão de Métodos e Uso de Software ImageJ na Diferenciação de Ambientes Depositionais na Ilha de Santa Catarina e Região Continental, SC, Brasil*. *Gravel*, 11 (1): 37-47.

Riley, N. A. (1941). *Projection sphericity*. *Journal of Sedimentary Research*, 11 (2): 94–95. doi: <https://doi.org/10.1306/D426910C-2B26-11D7-8648000102C1865D>

Rittenhouse, G. (1943). *A visual method of estimating two-dimensional sphericity*. *Journal of Sedimentary Petrology*, 13(2): 79-81.

Rorato, R., Arroyo, M., Andò, E. & Gens, A. (2019). *Sphericity measures of sand grains*. *Engineering Geology*, 254: 43-53. <https://doi.org/10.1016/j.enggeo.2019.04.006>

Rorato, R., Arroyo, M., Gens, A., Andò, E., Viggiani, G. (2018). *Particle Shape Distribution Effects on the Triaxial Response of Sands: A DEM Study*. *Micro to MACRO Mathematical Modelling in Soil Mechanics*. 277–286. https://doi.org/10.1007/978-3-319-99474-1_28

Rozenbaum, O., Bruand, A. & Le Trong, E. (2012). *Soil porosity resulting from the assemblage of silt grains with a clay phase: New perspectives related to utilization of X-ray synchrotron computed microtomography*. *Comptes Rendus Geoscience*, 344 (10): 516-525. <http://dx.doi.org/10.1016/j.crte.2012.09.004>

Safari, H., Balcom, B. J. & Afrough, A. (2021). *Characterization of pore and grain size distributions in porous geological samples – An image processing workflow*, *Computers & Geosciences*, 156. <https://doi.org/10.1016/j.cageo.2021.104895>

Sandeep, C. S., Hernandez, A., Stangeland, K. & Evans, T. M. (2023). *Shape characteristics of granular materials through realistic particle avatars*. *Computers and Geotechnics*, 157. <https://doi.org/10.1016/j.compgeo.2023.105352>

Silva, M. T. Q. S., Perretto, F., Cardoso, M. R. & Mazer, W. (2023). *Porosity: Some characterization techniques*. *Materials Today: Proceedings*.
<https://doi.org/10.1016/j.matpr.2023.03.716>

Sotomayor, J. M. G. (2018). *Avaliação do comportamento mecânico drenado e não drenado de rejeitos de minérios de ferro e de ouro reforçados com fibras de polipropileno*. Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, 184p.

Tavares Neto, R.F. & Silva, F.M. (2022). *Introdução à programação para engenharia: usando a linguagem Python*. LTC. Rio de Janeiro.

Taylor, H. F., O'Sullivan, C. & Sim, W. W. (2015). *A new method to identify void constrictions in micro-CT images of sand*. *Computers and Geotechnics*, 69: 279-290.
<https://doi.org/10.1016/j.compgeo.2015.05.012>

Vangla P., Roy N., Mendu K. & Latha G. M. (2014). *Digital image analysis for the determination of size and shape parameters of sand grains*. Golden Jubilee Conference of the IGS Bangalore, 1-9.

Vanmarcke, E. H. (1977). *Probabilistic Modeling of Soil Profiles*. *Journal of the Geotechnical Engineering Division, ASCE*, 103 (11): 1227-1246. <https://doi.org/10.1061/AJGEB6.0000517>
Wadell, H. (1932). *Volume, shape and roundness of rock particles*. *The Journal of Geology*, 40 (5): 443–451. <https://doi.org/10.1086/623964>

Willson, C. S., Lu, N. & Likos, W. J. (2012). *Quantification of Grain, Pore, and Fluid Microstructure of Unsaturated Sand from X-Ray Computed Tomography Images*. *Geotechnical Testing Journal*, 35 (6): 1–13. doi:10.1520/GTJ20120075.

Willson, C. S., Lu, N. & Likos, W. J.(2012). *Quantification of Grain, Pore, and Fluid Microstructure of Unsaturated Sand from X-Ray Computed Tomography Images*. *Geotechnical Testing Journal* 35 (6): 911–23. <https://doi.org/10.1520/GTJ20120075>

Youwai, S. & Makam, P. (2023). *Computer Vision for Particle Size Analysis of Coarse-Grained Soils*. 21st Southeast Asian Geotechnical Conference and 4th AGSSEA Conference. <https://doi.org/10.48550/arXiv.2311.06613>

Zubeldia, E. H. (2013). *Uso dos autômatos celulares bidimensionais e imagens tomográficas na geração de meios porosos artificiais*. Dissertação de Mestrado. Universidade de Brasília, Brasília, DF, 93 p.

Zubeldia, E. H., Ozelim, L. C. S. M., Cavalcante, A. L. B. & Crestana, S. (2016). *Cellular automata and x-ray microcomputed tomography images for generating artificial porous media*. International Journal of Geomechanics, 16 (2). [https://doi.org/10.1061/\(ASCE\)GM.1943-5622.0000527](https://doi.org/10.1061/(ASCE)GM.1943-5622.0000527)