

DESENVOLVIMENTO DE PROGRAMA COMPUTACIONAL PARA ANÁLISE DE IMAGENS DE MICROTOMOGRAFIA DE RAIOS-X

TIAGO YPARRAGUIRRE VIEGAS

Aluno de Graduação em Eng^a Química, 6^o período, UFRJ
Período PIBIC/CETEM: julho de 2012 a julho de 2013,
tviegas@cetem.gov.br

CLAUDIO L. SCHNEIDER

Orientador, Engenheiro de Minas, Ph.D.
cschneid@cetem.gov.br

1. INTRODUÇÃO

A análise por microtomografia de raios-X é utilizada para reconstruir e criar seções transversais de um objeto em 3D sem destruí-lo. Imagens de alta resolução podem ser geradas de materiais muito densos de até 8,0 g/cm³ (Lin e Miller, 2001, 2004). A captação de imagens é feita com a emissão de raios-X em formato cônico e sua captação por uma placa detectora enquanto a amostra gira lentamente. As projeções são processadas e por análise estereológica pode-se reconstruir seções horizontais reais da amostra. Esta técnica é aplicada para análise de partículas, o que permite a obtenção de uma série de dados que vão desde tamanho de partícula até sua liberação. Se comparado à microscopia eletrônica, a microtomografia possui vantagens pois não é destrutiva e obtém uma reconstrução em 3D real da amostra e sua estrutura interna.

2. OBJETIVOS

Neste trabalho objetivou-se desenvolver um programa nas linguagens C e C++ para analisar imagens de microtomografia de raios-X visando à obtenção de dados como tamanho e volume de partícula, composição e principalmente liberação.

3. METODOLOGIA

3.1 Imagens

As imagens de tomografia de raios-X foram obtidas de minérios de ferro e de cobre em um tomógrafo XRadia MicroXCT do Departamento de Engenharia Metalúrgica da Universidade de Utah. A aquisição foi feita com as seguintes definições: 80V, 125^a e temperatura da câmara de -60°C. Foram geradas cerca de 900 imagens com cerca de 1000x1000 pixels para cada tamanho de partícula de cada minério. Os tempos de aquisição foram longos e de cerca de 8h.

3.2 Software desenvolvido

Um *software* foi desenvolvido para manipular as imagens e extrair dados mencionados no objetivo. Este foi desenvolvido utilizando as linguagens C e C++ e funciona apenas com imagens em escala de cinza e com pixels de 8bits. Sua interface está em língua inglesa para facilitar sua internacionalização, como também devido a forte influência desta língua na criação de programas e na programação em si.

Para o desenvolvimento da interface do programa utilizamos um *software* chamado Fluid que faz parte do conjunto de ferramentas denominado FLTK. Para utilizá-lo e também para o desenvolvimento do nosso programa, simulamos um ambiente GNU/Linux com o uso do MinGW Shell.

3.3 Passo-a-passo do tratamento de imagens

Para a extração de dados, primeiramente, a imagem sofre o tratamento de um filtro mediano que atenua o ruído e permite que os dados sejam extraídos com maior confiança.

Após isto, o próximo passo é realizar um processo de delimitação para melhorar o contorno das partículas e ajudar a diferenciar entre as partículas e o fundo. Após isto, realiza-se a separação

das fases presentes das partículas. Neste processo, separa-se o fundo (preto), mineral menos denso (cinza) e mineral mais denso (branco). Com isso, se gera uma imagem com apenas três cores e a partir daí pode-se medir as áreas superficiais, volumes e liberação de cada uma das fases. A delimitação é um processo para melhorar o contorno das partículas e ajudar a diferenciar partículas do fundo.

O quarto passo é separar as partículas e rotulá-las em cada imagem. Logo após, as imagens são analisadas em conjunto e as partículas são identificadas em 3 dimensões. Resta apenas então contar os pixels e definir a área superficial, volume e liberação das partículas.

4. RESULTADOS E DISCUSSÃO

4.1 Vantagem da microtomografia de raios-X sobre microscopia eletrônica

A microscopia eletrônica foi e ainda é muito usada para realização de análises de propriedades de partículas. Porém, esta técnica se utiliza apenas de seções transversais obtidas em duas dimensões. Portanto, propriedades como diâmetro de partícula, volume, composição e liberação podem ser medidas apenas em uma ou duas dimensões. Para a obtenção das mesmas propriedades em três dimensões, é necessário realizar conversões estereológicas baseadas em uma série de dados estatísticos. Esta conversão possui erros que aumentam se os dados forem poucos ou não forem representativos. Ela é necessária porque nem todos os dados obtidos em 2D não podem ser diretamente extrapolados para o 3D. Para melhor compreensão, temos como exemplo o que ocorre com dados de liberação. Seções transversais de partículas puras revelam sempre 100% de liberação. Porém, ao se obter seções transversais de uma partícula mista, algumas seções podem ainda assim nos fornecer uma liberação em 2D de 100%, enquanto outras seções darão liberações diferentes de 100%. Portanto, percebe-se que a liberação 2D é sempre maior do que a 3D, uma vez que além das partículas puras, aquelas mistas também podem revelar liberação de 100% em alguns casos.

4.2 Apresentação do programa desenvolvido

O *software* desenvolvido está representado na Figura 1. Nesta, pode-se ver a janela principal, a de visualização da imagem e a do histograma. Também está em destaque para ilustração o submenu de filtros, que possui maior quantidade de opções. A janela de visualização de imagem pode ter seu tamanho ajustado, o que leva a um redimensionamento da imagem em análise. No menu da janela principal estão alocadas todas as funções do programa. Em “File”, estão presentes as opções de carregar e salvar imagens e encerrar o programa. Ao lado está a opção que abre a janela de histograma. Esta mostra um gráfico em quantidade de pixels por cada valor na escala de cinza e permite a realização de algumas operações de segmentação de imagem. Como as imagens são em 8 bits, a escala de cinza varia de 0 (preto) até 255 (branco). A opção de menu “Binary” possui uma única opção para rotular as partículas. Este procedimento utiliza uma imagem binária (preta e branca) para contar a quantidade de partículas e numerá-las. Por último, tem-se a opção “Filters”. Entre os filtros disponíveis, pode-se observar algumas rotinas de detecção de bordas, remoção de ruídos, delimitação, ajuste de brilho e separação de fases.

4.3 Filtros implementados

Filtros medianos com o uso de uma matriz 3x3x3 foram desenvolvidos para remoção de ruído. Este tamanho de matriz é ideal para evitar flutuações sem grandes alterações nas bordas e nos dados (Videla et al, 2006). Este filtro consiste em substituir o valor de cada pixel pela média deste com os seus pixels vizinhos. Estes últimos, neste caso, não são somente os pixels em um quadrado 3x3 da imagem, mas também considera-se os quadrados 3x3 da imagem acima e abaixo que também estão próximos e são vizinhos do pixel em questão.

A delimitação, como já mencionado, aumenta o contraste entre as bordas das partículas do fundo. Nela, se usa primeiramente um filtro que identifica as bordas das partículas. Em seguida, pixels das bordas recebem valores (cores) mais próximos aos do background ou da partícula, dependendo da quantidade de cada um destes ao redor do pixel em questão.

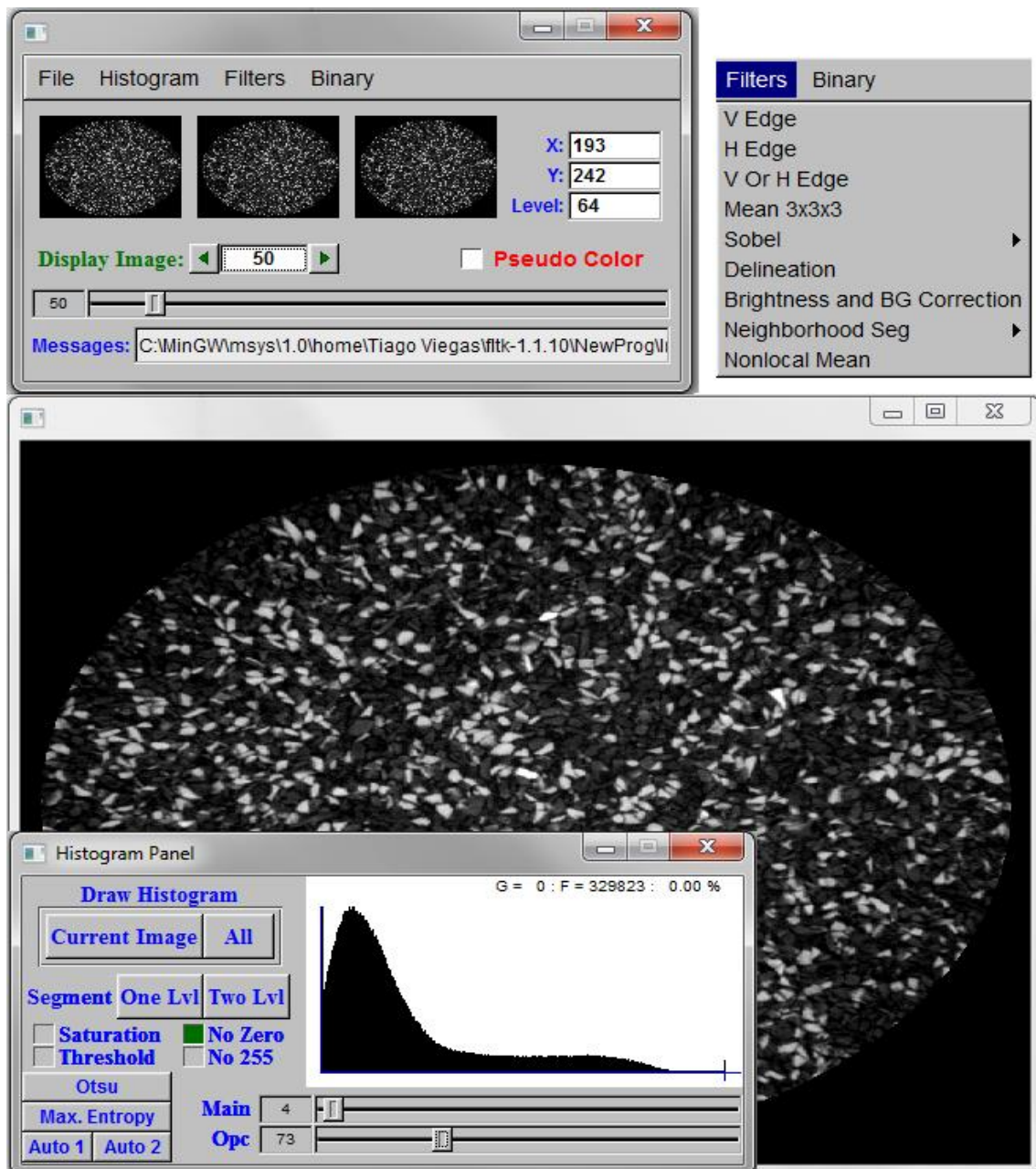


Figura 1. Programa desenvolvido mostrando a visualização de imagens, histograma e filtros.

Filtros de detecção de borda verificam as diferenças de tons de cinza entre pixels vizinhos para localizar as bordas das partículas. Uma diferença de cores muito grande indica que aquela região é uma borda de partícula e, portanto, este filtro de certa forma traça um contorno de todas as partículas. Ele é usado principalmente como etapa do processo de delimitação, pois é necessário identificar as bordas para poder realizar a melhoria de contorno das partículas.

Os filtros para separação de fases também foram implementados utilizando análises com matrizes 3x3 e 5x5. Porém, os resultados obtidos não foram expressivos devido a problemas com a má qualidade das imagens. Um filtro de correção de brilho e de fundo foi adicionado na tentativa de melhorar esta qualidade, porém sem sucesso uma vez que o baixo contraste era o que mais prejudicava a imagem. Portanto, foi necessária a utilização de um software auxiliar para a obtenção de melhores resultados. Recorreu-se então ao programa Fiji, que possui uma rotina de segmentação denominada Weka que permite a adaptação dos filtros ao tipo de imagem analisada e produz bons resultados. A Figura 2 mostra a mesma imagem tratada com diferentes filtros. As duas últimas imagens mostradas na figura revelam a comparação entre os resultados com o filtro desenvolvido e o filtro de segmentação Weka.

Fiji é um pacote de processamento de imagem utilizado no programa ImageJ que se baseia na linguagem Java. Weka é uma coleção de algoritmos utilizados para processamento mineral e está disponível como *plugin* de segmentação de imagens para o conjunto Fiji/ImageJ.

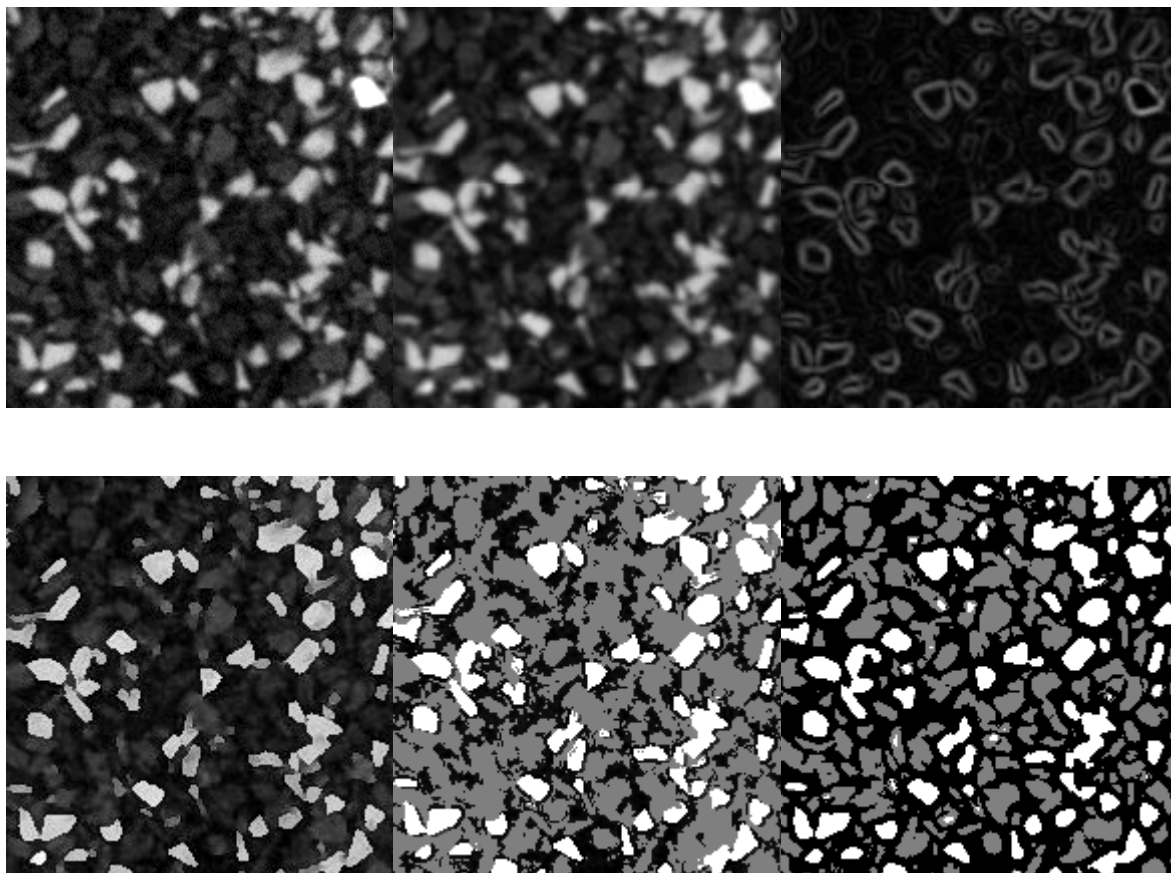


Figura 2. Seções transversais ampliadas da fração $-75 +53\mu\text{m}$ de minério de cobre. Da esquerda pra direita, de cima para baixo: Original, após filtro mediano, após detecção de bordas, após delineação, após filtro de segmentação desenvolvido e após filtro Weka.

Devido à complexidade da tarefa e principalmente a dificuldade gerada pela falta de contraste e qualidade da imagem, este trabalho não pode atingir todo seu objetivo no tempo estipulado. Porém, ótimos resultados já foram alcançados viabilizando ainda mais sua conclusão.

5. AGRADECIMENTOS

O autor agradece ao PIBIC/CNPq pelo apoio financeiro, ao CETEM pela oportunidade, a seu orientador Dr. Cláudio Schneider pela orientação e a todos que ajudaram de alguma forma.

6. REFERÊNCIAS BIBLIOGRÁFICAS

C. L. LIN, J. D. MILLER, A New Cone Beam X-Ray Microtomography Facility for 3D Analysis of Multiphase Materials. In: 2ND WORLD CONGRESS ON INDUSTRIAL PROCESS TOMOGRAPHY, Hannover, Germany, 2001, pp. 98–108.

J. D. MILLER, C. L. LIN, Three-dimensional analysis of particulates in mineral processing systems by cone beam X-Ray microtomography, **Minerals and Metallurgical Processing**, v.21, 2004, p. 113–124.

VIDELA, A.; LIN, C. L.; MILLER, J.D.; Watershed Functions Applied to a 3D Image Segmentation Problem for the Analysis of Packed Particle Beds. **Part. Part. Syst. Character.** v.23, p.237–245, 2006