

Tomo-GPU: Um Ambiente de Resolução de Problemas Destinado à Análise de Dados Tomográficos Relativos à Caracterização Estrutural de Materiais

Paulo Jorge Quaresma, CITI-FCT/UNL,
pjq@di.fct.unl.pt
Adriano Lopes, CITI-FCT/UNL,
alopes@di.fct.unl.pt
Pedro Medeiros, CITI-FCT/UNL,
pm@di.fct.unl.pt

Bruno Preto, CITI-FCT/UNL,
bpreto@gmail.com
Alexandre Velhinho, CENIMAT-
FCT/UNL, ajv@fct.unl.pt

Resumo

A caracterização micro-estrutural de materiais compósitos utiliza a tomografia de raios X para capturar informação sobre as características internas das amostras, de forma a elucidar o investigador sobre as suas propriedades intrínsecas. Os dados tomográficos em bruto carecem de processamento mediante diversas operações computacionais, que começam pela eliminação de ruído e outros artefactos. A experiência adquirida pelos autores mostrou que, em alguns casos, o tempo de processamento é muito longo, não sendo igualmente fácil para um cientista de materiais interagir com o programa de forma a definir os parâmetros de computação mais adequados e uma sequência correta de operações. Este artigo descreve um ambiente de resolução de problemas chamado Tomo-GPU dedicado especificamente à área da tomografia e que será executado num computador de secretária que inclui um ou mais GPGPUs. O Tomo-GPU pretende explorar uma configuração hardware em que a um CPU *multi-core* tradicional se adicionam as capacidades de processamento de um ou mais GPUs para conseguir, mesmo com grandes volumes de dados, tempos de execução suficientemente curtos para permitir uma utilização interactiva. Uma das principais contribuições do trabalho será o suporte de uma hierarquia de memória com três níveis, em que além dos níveis disco e RAM tradicionalmente usados na chamada computação *out-of-core*, se acrescenta a memória do(s) GPU(s). O trabalho encontra-se em fase de desenvolvimento, tendo sido já implementado um conjunto de funcionalidades relacionadas com o tratamento dos dados e a sua visualização.

Palavras-chave: Micro-tomografia de materiais compósitos, GPGPU (General-Purpose Graphics Processing Unit), Visualização, Ambiente de Resolução de Problemas, Ambientes de execução heterogéneos.

Title: Tomo-GPU: a problem-solving environment for analysing tomographic data of structural characterization of materials

Abstract

The micro-structural characterization of composite materials uses X-ray tomography to collect information about the internal characteristics of the samples, in order to educate the researcher about their intrinsic properties. The raw tomographic data, need to go through several steps of computational processing, starting with the elimination of noise and other artifacts. The experience gained by the authors has shown that in some cases the required processing time is to long and therefore not easy for a materials scientist to interact with the

program in order to define the most adequate computing parameters and the correct sequence of operations.

This article describes a Problem Solving Environment (PSE), called *Tomo-GPU*, dedicated specifically to the field of tomography, and targeted to run on a desktop computer equipped with one or more GPGPUs. Tomo-GPU intends to exploit a hardware configuration in which is added to a traditional multi-core CPU, the processing capabilities of one or more GPUs to achieve, even with large volumes of data, execution times that are short enough to allow for an interactive use. One of the main contributions of this work is the support of a three-layer memory hierarchy where, in addition to the disk and RAM levels, traditionally used in the so called out-of-core memory computing, it is added the memory of the GPUs. The work is in a development stage with a number of features related to data processing and visualization already implemented.

Keywords: Micro-tomography of composite materials, GPGPUs (General Purpose Graphics Processing Units), Visualization, Problem Solving Environments, heterogeneous runtime environments

1. Objetivo

Os materiais compósitos [1] são formados a partir da dispersão de reforços descontínuos (usualmente fibras ou partículas cerâmicas) numa matriz metálica percolante¹. Estes materiais são objeto de interesse por parte de muitos domínios industriais, como as indústrias automóvel, naval, aeronáutica e aeroespacial, devido às suas propriedades otimizadas. Um controlo apropriado da distribuição destas partículas pode conduzir à produção de compósitos com gradiente funcional, nos quais a distribuição espacial dos reforços é objeto de uma variação controlada ao longo de pelo menos uma direção, facto do qual resultam propriedades potencialmente mais interessantes para certas aplicações.

Os especialistas de Materiais que trabalham nesta área concebem não só novos materiais compósitos, mas também novos métodos para o fabrico de compósitos já conhecidos. No ensaio destes métodos são produzidas amostras que servem para avaliar as propriedades do material. Neste processo de avaliação é muito importante a chamada caracterização micro-estrutural do compósito, nomeadamente no que à distribuição dos reforços diz respeito, através da obtenção de informação sobre, por exemplo, as suas dimensões, forma, orientação espacial, localização, concentração e relações de vizinhança ou de interação com a matriz.

Uma das técnicas usadas neste contexto consiste na análise de imagens tomográficas, em particular a micro-tomografia de raios X com radiação de sincrotrão (CT). Os volumes de dados produzidos por CT são muito grandes, correspondendo a uma matriz a três dimensões que contém a reconstrução virtual do material analisado. Neste contexto, é muito útil para o especialista de Materiais ter disponível uma aplicação informática que lhe permita:

- A visualização a três dimensões dos dados;

¹ Referindo-se esta designação ao facto de ser possível imaginar um número infinito de trajetos contínuos entre superfícies opostas do material exclusivamente através da matriz, i.e, sem atravessar quaisquer regiões ocupadas pelos reforços

- A execução de várias operações de processamento de forma a remover defeitos inerentes à técnica usada;
- Obtenção de informação referente à caracterização micro-estrutural do material, quer de natureza individual – estudo de um determinado elemento de reforço – quer com carácter estatístico;
- Uma adequada representação da informação obtida, de forma flexível e sintética. Neste aspeto estaria igualmente contemplada a possibilidade de enumeração seletiva de elementos de reforço, com base em critérios definidos pelo utilizador.

Com vista a uma mais fácil apreciação dos requisitos da aplicação informática em desenvolvimento, serão seguidamente detalhados alguns aspetos relacionados com a *génese das imagens tomográficas* e com o seu processamento posterior.

1.1 Geração de imagens tomográficas

O princípio da tomografia computadorizada consiste na medição, por intermédio da irradiação da amostra com uma radiação penetrante, da distribuição espacial de uma determinada grandeza física do objeto estudado, examinada a partir de diferentes direções (projeções), e no cálculo, a partir desses dados, de imagens virtuais livres de sobreposições. Em virtude do seu poder penetrante, resultante dos comprimentos de onda envolvidos, a radiação X constitui o vetor mais frequentemente utilizado na implementação da CT, correspondendo a grandeza física medida à sua intensidade após absorção parcial pelo objeto de estudo. Com esta técnica é possível reconstruir virtualmente o volume de material analisado, o que permite conhecer as suas propriedades internas, dentro dos limites de resolução espacial da técnica, os quais se situam, quando o feixe de raios-X é produzido por um sincrotrão, em torno de $1\mu\text{m}$.

A Figura 1 ilustra o procedimento adotado na micro-tomografia, no seu modo mais simples, designado como *contraste de fase*. A fonte de radiação de sincrotrão, representada na parte direita da imagem, emite um feixe paralelo de radiação policromática, da qual é extraída uma componente monocromática (i.e., dotada de uma dispersão mínima de comprimentos de onda) recorrendo a um monocromador. A amostra encontra-se no porta-amostras, e do lado esquerdo da figura está situada uma câmara que vai captar a imagem da amostra após esta ser atravessada pelo feixe de raios-X. A projeção bidimensional assim recolhida é depois enviada para um computador para processamento. Para aceder a uma visualização tridimensional torna-se necessário registar um conjunto alargado de projeções, cada uma referente a uma orientação distinta da amostra em torno do seu eixo, algo que na prática resulta da rotação do porta-amostras.

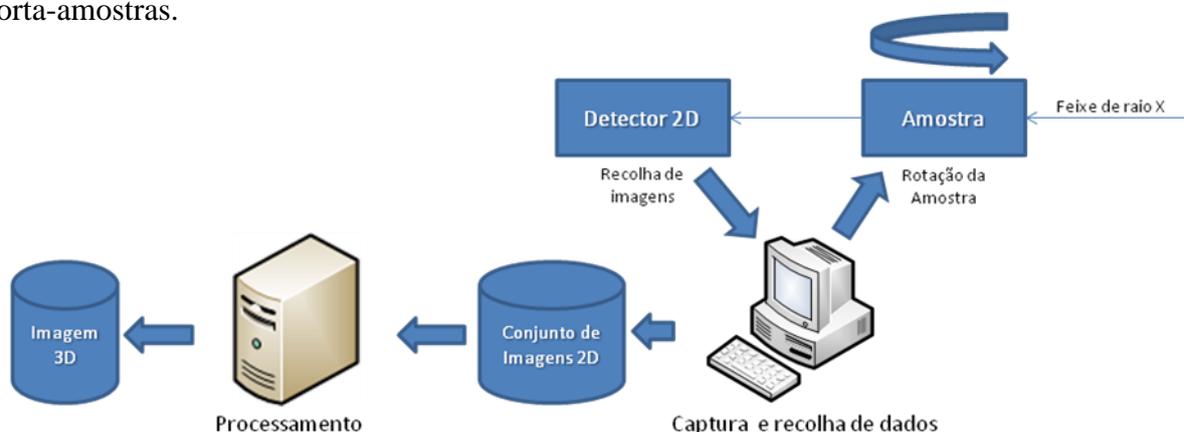


Figura 1- Montagem experimental de CT

Uma alteração deste procedimento, conhecida como *holotomografia*, recorre ao registo sequencial e sobreposição de projecções obtidas segundo diferentes valores da distância entre a amostra e a câmara de deteção da radiação transmitida. Este modo de operação traduz-se em ganhos significativos quanto ao contraste visual entre os constituintes do material, embora tal seja acompanhado de uma ligeira degradação da resolução espacial, bem como de um evidente acréscimo quanto aos tempos de aquisição.

1.2 Processamento das imagens tomográficas

Os volumes de dados produzidos por CT são de grande dimensão, correspondendo geralmente a *arrays* 3D, onde cada elemento, designado como *voxel*, pode ser um valor inteiro ou real. Esta informação requer várias operações de processamento, nomeadamente destinadas a:

- Remoção de artefactos (elementos da imagem sem existência real, associados aos fenómenos de interação entre a radiação e a matéria, bem como às interações entre diferentes componentes do feixe transmitido) e redução de outro ruído inerente ao processo de aquisição da imagem;
- Extração de informação útil, conducente à caracterização microestrutural pretendida dos compósitos. A principal condicionante desta tarefa consiste na incapacidade, por parte dos sistemas informáticos, de discriminar de forma imediata, a partir da gama cromática das imagens tomográficas, a presença de objetos de interesse (por exemplo, os reforços), sem recurso a operações intermédias de processamento da imagem.

De seguida, faremos uma breve referência ao conjunto de operações que um sistema deste tipo deve suportar:

- **Histograma e sua derivada:** Esta operação calcula a frequência de cada um dos níveis de cores, assim como o cálculo das diferenças de frequências entre cores consecutivas (derivada).
- **Segmentação:** Permite dividir a imagem em dois (preto e branco), ou três (preto, branco e cinzento) níveis de cores com o intuito de isolar populações de objetos cujo estudo seja particularmente interessante para o subsequente processamento.
- **Limpeza da imagem:** Permite a eliminação de ruído, nomeadamente pequenos conjuntos de voxels brancos em regiões pretas ou vice-versa.
- **Histerese:** Promove a dilatação da cor preta e branca dentro das regiões cinzentas. Estas estão normalmente na fronteira, a qual muitas vezes não é contínua ou bem definida.
- **Percolação:** Nesta operação pretende-se identificar claramente a fronteira das partículas, para tal é feita uma marcação da região interna e da respetiva fronteira (conjunto de voxels exteriores).
- **Aplicação de máscaras:** Permite evidenciar certos aspetos da imagem, nomeadamente a forma como os voxels cinzentos evoluíram para preto ou branco
- **Estatísticas de Granulometria:** Permite a identificação quanto ao número, tamanho, forma das partículas detetadas na matriz metálica.

Algumas dessas operações envolvem a análise individual de cada *voxel*, através de comparações de valores entre *voxels* vizinhos e alteração dos seus valores. Devido ao grande volume de *voxels* existente, estas operações podem requerer tempos de processamento de várias horas ou mesmo dias, numa máquina mono-processador.

O ponto de partida para este trabalho foi um programa denominado *Tritom* [2], desenvolvido por Gérard Vignoles, da Universidade de Bordéus. Tratava-se, na origem, de um programa

sequencial implementado em C, suportando várias operações de processamento tomográfico. No entanto, este programa apresentava algumas limitações:

- O tempo necessário para a realização de certas operações;
- A ausência de funcionalidades para visualização dos dados;
- A falta de uma interface gráfica;
- Ausência de várias funcionalidades como o reconhecimento de formas e orientação das partículas;
- Dificuldades no estabelecimento de uma sequência de operações.

Com o tempo, foram realizados esforços na solução de algumas das limitações enunciadas. Como exemplos, apontam-se a paralelização das operações mais dispendiosas, utilizando um esquema de paralelização dos dados [3], assim como o desenvolvimento de uma interface gráfica para não-especialistas [4].

De forma a auxiliar o especialista de Materiais na análise de dados tomográficos, qualquer sistema deve fornecer uma interface gráfica adequada. O conjunto de operações a ser disponibilizado inclui métodos eficientes para: (i) processamento de dados; (ii) visualização de resultados; (iii) interação mediante uma interface amigável para o utilizador. Um ponto-chave é a possibilidade do investigador poder definir os parâmetros da computação e verificar em tempo real os seus efeitos, incluindo a interatividade subjacente.

1.3 Visão geral do sistema a implementar

Neste trabalho, pretende-se construir um ambiente de resolução de problemas (PSE) para utilização por parte de um cientista de materiais. O cientista deverá utilizar um editor visual onde definirá a sequência de operações a aplicar aos dados. Um cenário típico é o indicado na Figura 2. A imagem é lida de um ficheiro, processada por intermédio da operação 1, após o que o utilizador dispõe de 3 opções: prosseguir para a operação 2, visualizar a imagem, ou guardar o ficheiro em disco. Cada operação é representada no PSE por uma caixa, e estas são ligadas entre si através de canais.

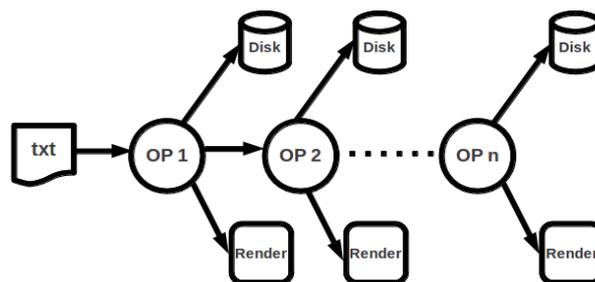


Figura 2 - Operações de processamento de uma amostra

Podemos definir um PSE como uma solução integrada que permite solucionar uma determinada classe de problemas num domínio particular de aplicação. Estes encapsulam os algoritmos e as estratégias de resolução dos problemas de forma a fornecer um acesso transparente ao utilizador final. De momento, não temos conhecimento de nenhum PSE específico para a área da tomografia. Sendo assim, os objetivos centram-se em 3 áreas fundamentais:

- Paralelização de algoritmos;
- Funcionalidades de visualização;
- Interatividade na definição da sequência de operações.

Do ponto de vista da funcionalidade do sistema, são várias as áreas disponíveis para intervenção. O sistema a desenvolver deve permitir a visualização de uma imagem bruta ou de partes selecionadas dessa imagem, devendo ser também possível a visualização de planos, segundo orientações definidas pelo utilizador. Nesta vertente de visualização de dados, será privilegiada a utilização das técnicas de extração de isosuperfícies e de *volume rendering*, de acordo com a sua adequação à tarefa a desempenhar.

Numa primeira fase, para a análise da imagem em bruto, devem ser desenvolvidos métodos para construção de histogramas, segmentação, e estatísticas de granulometria. Após esta última operação, é nossa intenção obter um conjunto de indicadores geométricos. Nestes podemos incluir indicadores individuais para cada objeto, tais como: centróides, volume e superfície do objeto, número de voxels internos e *bounding boxes*.

Numa segunda fase, é nossa intenção construir autocorrelogramas e algoritmos para separação de objetos. Relativamente aos objetos, pretendemos também obter informações sobre: raios e fatores de forma, momentos de inércia, direções principais, tamanho da zona de influência, e reconhecimento de faces, planos, e ângulos entre faces (morfologia cristalina).

É importante obter igualmente informações sobre a relação entre objetos, e nesta fase é importante conhecer a lista de primeiros vizinhos de um objeto selecionado, os comprimentos de matriz separando os objetos, e a construção de *clusters* com critério de distância entre objetos próximos.

A nível da distribuição de partículas, é importante obter dados sobre: distribuição de posições, direções principais, granulometria simples, fatores de forma, caixas de delimitação, distância média dos primeiros vizinhos, número de primeiros vizinhos, ângulos entre direções principais e taxa de associação entre grãos e poros.

Para a investigação visual dos dados é nossa intenção: o desenvolvimento de técnicas de *brushing* (selecionar e visualizar partículas de acordo com as distribuições de indicadores geométricos), a seleção de um objeto e do *cluster* associado, a coloração de partículas de acordo com uma distribuição particular, visualização dos esferoides equivalentes, e a visualização ilustrativa das direções principais.

O sistema a implementar deve poder ser executado num computador pessoal equipado com um CPU clássico com múltiplos núcleos e um ou mais GPU's.

A figura 3 apresenta uma visão do sistema. Numa primeira fase o especialista de Materiais define a sequência de operações a efetuar, bem como as situações em que existe interação com o utilizador (*steering*). Nesta fase estará disponível uma biblioteca de módulos específicos para o processamento de dados tomográficos e para a visualização desse volume de dados. Na segunda fase, com base na descrição da aplicação produzida anteriormente, um sistema de apoio à execução numa plataforma de hardware heterogénea instancia os vários módulos e estabelece os canais de comunicação entre eles. Uma vez iniciada a execução, o potencial computacional existente no conjunto CPU e GPU assegurará tempos de execução para as operações individuais que sejam compatíveis com uma utilização interativa da aplicação.

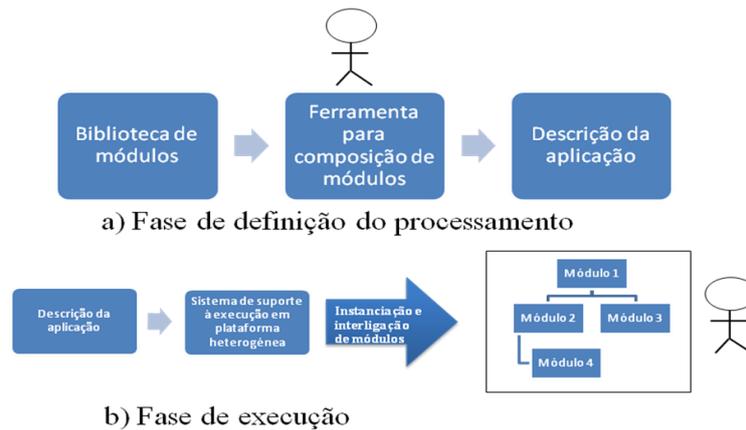


Figura 3 – Visão geral do sistema

É nosso objetivo também tirar partido da camada de software sobre a qual as versões paralelas dos algoritmos de processamento e visualização no interior do sistema podem ser construídas. Esta camada deve coordenar a computação paralela através de processadores heterogêneos e deve fornecer um mecanismo de sincronização das unidades de computação paralela.

2. Estado da Arte

Neste capítulo iremos apresentar algumas ferramentas para a construção de ambientes de resolução de problemas. De seguida exploraremos algumas técnicas para a visualização de grandes volumes de dados e terminaremos com algumas considerações sobre paralelização de algoritmos.

2.1 Ferramentas para a construção de PSEs

De um modo geral, o estudo de fenómenos físicos implica também a visualização dos respetivos dados científicos. Em termos gráficos, as técnicas utilizadas para o efeito inserem-se nas áreas de visualização científica, computação gráfica e processamento de imagem. Enquanto as duas primeiras operam sobretudo no modelo de dados intrínseco ao fenómeno físico, o processamento de imagem centraliza a sua ação na imagem adquirida pelo dispositivo que é utilizado durante a observação do fenómeno físico. De seguida, apresentam-se algumas ferramentas computacionais de índole gráfica que podem ser úteis na construção de PSEs.

2.1.1 Visual Toolkit

O VTK é um software de código aberto desenvolvido pela Kitware [7], com funcionalidades específicas para computação gráfica, processamento de imagem e visualização. Os objetos base e os algoritmos no VTK estão implementados em C++, embora suporte interfaces para outras linguagens como por exemplo Tcl/Tk, Java e Python.

2.1.2 SCIRun

Este pacote de software foi desenvolvido pela Universidade de Utah [9] e é considerado como um *Framework* clássico para a construção de PSEs segundo o modelo de fluxo de dados. Cada módulo realiza uma função específica sobre um conjunto de dados. Por exemplo, ler os

dados de um porto de entrada, aplicar-lhes uma transformação e enviar de seguida para um porto de saída. Os módulos são representados graficamente no editor do SCIRun através de caixas, e o fluxo de dados entre módulos é inerente às ligações existentes entre si. A Figura 4 apresenta um grupo de módulos interligados entre si, formando uma rede de dados.

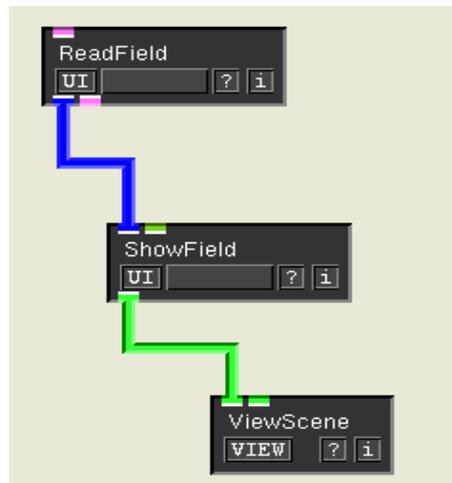


Figura 4 - Rede de módulos no ambiente SCIRun

O sistema suporta igualmente paralelismo intra e entre módulos. O paralelismo entre módulos é definido pelo programador quando pretende obter o maior desempenho para os seus algoritmos. Sendo assim, é da responsabilidade do programador a implementação de algoritmos paralelos para a exploração das arquiteturas multiprocessador.

2.1.3 OpenCV

O OpenCV é uma biblioteca de código aberto [12] para processamento de imagem e desenvolvida em C/C++. A biblioteca possui um conjunto de primitivas destinadas ao processamento de entidades geométricas (pontos, linhas, retângulos, etc.). A biblioteca disponibiliza igualmente um tipo especial de dados para codificação de imagens. O conjunto de operações sobre as imagens é vasto: alteração do tamanho, extração de canais individuais, sobreposição de duas imagens, segmentação de uma imagem, etc.~

2.1.4 ImageJ

O ImageJ é uma biblioteca Java de domínio público, desenvolvida no âmbito da área de processamento de imagem. Devido a ser de código aberto, é sempre possível implementar novos *plugins* e integrá-los na biblioteca.

As funcionalidades existentes são várias: visualização do histograma da imagem, aplicação de filtros à imagem, definição de regiões de interesse (ROI) e consequentemente manipulação destas, segmentação da imagem, análise de partículas, manipulação de camadas, etc. O sistema permite também obter informação importante em tarefas associadas a análise de partículas, como por exemplo o número e tamanho destas. Refira-se no entanto que não é possível proceder a uma análise de imagens a três dimensões.

2.2 Visualização de grandes volumes de dados

O processo de visualização de dados segue, de um modo geral, o esquema indicado na Figura 5. No contexto do PSE a desenvolver, a ordem de grandeza dos dados poderá alcançar os Gigabytes ou mesmo os Terabytes, o que significa que o nível de exigência computacional é elevado. Tal exigência é ainda mais premente em termos de software, a que não é alheio o facto de existir uma evolução desequilibrada entre hardware e software.

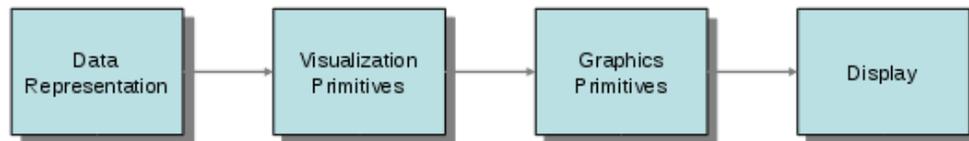


Figura 5 - Processo de visualização

De uma forma geral, os algoritmos para visualização científica são classificados de acordo com o tipo de dados que processam: (1) Algoritmos escalares (2) Algoritmos vetoriais (3) Algoritmos tensoriais.

No caso da tomografia em estudo, os dados são escalares, ou seja, associam um ponto ou célula 3D a um valor escalar. As técnicas bastante utilizadas na visualização deste tipo de dados são: extração de isosuperfícies e a renderização de volumes.

2.2.1 Extração de Isosuperfícies

Uma isosuperfície é um contorno tridimensional representando o conjunto de pontos no espaço 3D com um determinado valor escalar. O algoritmo clássico, denominado Marching-Cubes [11], tem sido melhorado desde a sua apresentação em 1987. O método percorre o volume de dados, composto por células 3D (cubos) com valores de dados nos respetivos vértices e, para cada uma das células, determina a superfície no seu interior com base nos valores dos vértices e no valor escalar da superfície a representar.

Constata-se na prática que uma grande percentagem de células do volume de dados não contém a superfície a representar. Assim sendo, podem ser utilizadas estratégias que visam acelerar o processo de extração de superfícies de modo a que só sejam processadas as células que realmente interessam. Tais estratégias assentam sobretudo num esquema de partição hierárquica do volume de dados, em que é possível seccionar recursivamente uma região em várias sub-regiões através de planos de corte, nomeadamente planos ortogonais.

2.2.2 Renderização de volumes

Esta técnica tem como objetivo principal a representação gráfica do volume 3D na sua globalidade. Essa representação assenta na entidade pixel, o que significa que não são geradas primitivas geométricas intermédias, como por exemplo seria o caso de triângulos. As metodologias mais importantes têm como base as seguintes estratégias:

- **Object order:** A visualização parte da decomposição do volume em elementos básicos, os quais são individualmente projetados no plano da imagem, criando assim a imagem final. O mapeamento de texturas plano a plano enquadra-se nesta filosofia.

- **Image order:** A visualização parte da projeção de raios a partir de cada pixel contido no plano da imagem, em direção ao volume de dados. O resultado final do pixel depende da composição da amostragem feita pelos vários elementos do volume de dados intersectados pelo raio. Por vezes, esta metodologia confunde-se com a técnica mais genérica de *ray casting* dada a semelhança existente.

Refira-se ainda que é usual adotar técnicas híbridas com base nas duas metodologias referidas.

2.3 Paralelização do processamento de dados tomográficos.

De forma a reduzir o tempo de execução dos algoritmos tomográficos, a utilização de um esquema de paralelização parece ser a opção mais viável. Na computação paralela, o termo *speedup* permite determinar a razão entre o tempo de execução de um algoritmo paralelo face ao tempo de execução da sua versão sequencial.

Foram conduzidas algumas experiências [3] utilizando um esquema de paralelização geométrica dos dados. Para mostrar as potencialidades do uso de SMP's foi desenvolvida uma versão da operação de limpeza da imagem, em que a imagem é dividida em várias zonas, sendo cada zona da responsabilidade de um processo leve. A solução recorreu á biblioteca Posix Threads [13]. Em alternativa podia-se ter utilizado também um esquema de memória partilhado baseado em OpenMP [14].

Os *clusters* de computadores revelaram-se um meio barato e fiável de aumentar o poder computacional de um sistema, bastando para o efeito dividir o trabalho a realizar por vários nós de computação. Na paralelização geométrica, os clusters de computadores são um recurso ideal para fazer a distribuição do trabalho de computação entre os vários processadores. Utilizando um esquema baseado em troca de mensagens recorrendo á biblioteca MPI [15] é possível implementar uma versão paralela dos algoritmos de processamento das imagens. Nesta situação, o volume total da imagem é dividido em vários sub-volumes, ficando cada processo responsável pelas operações de processamento dessa parte da imagem. Existe um processo *master* que coordena as operações.

A rápida evolução da tecnologia demonstrou que no futuro a computação de alto desempenho será híbrida. Sendo assim, é necessário integrar dois tipos de componentes: CPU's com múltiplos núcleos e os processadores gráficos (GPUs). O baixo custo destes últimos atraiu a atenção da comunidade científica, devido ao seu poder computacional. A sua capacidade para executar centenas de *threads*, simultaneamente torna-os bastante úteis para a execução de operações paralelas.

Nos tempos mais recentes apareceu uma série de novas arquiteturas, nas quais podemos enumerar:

- GPGPUs: É possível utilizar os processadores das placas gráficas para a realização de cálculos que normalmente eram feitos por CPU's. Esta situação tornou-se possível com a adição de novas funcionalidades ao *pipeline* gráfico.
- FPGAs: Estes são circuitos integrados que podem ser configurados pelo utilizador após a sua manufatura. Esta configuração é feita recorrendo a uma linguagem de descrição do hardware (HDL).
- Outros aceleradores como o processador Cell: Os aceleradores já são usados desde há bastante tempo, bastando mencionar para o efeito o coprocessador aritmético 8087 da Intel.

Nos tempos mais recentes o interesse por estes dispositivos aumentou, pois estes podem ser incorporados em computadores pessoais ou servidores, permitindo aumentar o *speedup* de determinado tipo de operações. Um exemplo de um acelerador é o processador Cell, que combina uma arquitetura de uso geral com elementos de coprocessamento, o que permite acelerar o processamento de várias aplicações.

Na programação destes aceleradores existem várias propostas, sendo o CUDA [8] e o OpenCL [6] as mais promissoras. O CUDA foi lançado pela NVIDIA em 2007, é um modelo de programação paralelo, que disponibiliza três abstrações: um grupo hierárquico de *threads*, memória partilhada, e um esquema de sincronização por barreira.

O OpenCL é uma norma para a programação paralela de uso geral em sistemas heterogéneos. Com esta norma é possível a utilização de todas as unidades computacionais de um sistema (CPU, GPU) como *peers*. Estas unidades podem executar código SIMD (*Single Instruction Multiple Data*) ou SPMD (*Single Process Multiple Data*).

Nesta área já foi feito algum trabalho, nomeadamente, sobre a reconstrução de imagens tomográficas, utilizando o processador Cell [19]. Neste trabalho foi desenvolvida uma versão otimizada do algoritmo de retroprojeção, o qual é muito utilizado na reconstrução de imagens médicas, por exemplo.

Outro trabalho realizado nesta área [20] envolvia a utilização do processador Cell e GPU's na reconstrução de imagens tomográficas. Para tal os autores implementaram uma versão do algoritmo SIRT em GPU's, utilizando CUDA. Esta implementação foi de seguida comparada com uma implementação normal deste algoritmo num CPU clássico. Os resultados obtidos na velocidade de computação e nos tempos de acesso à memória foram muito promissores.

3. Sistema em desenvolvimento

Como foi referido, neste trabalho pretendemos desenvolver um ambiente de resolução de problemas aplicado à área da tomografia. Este software deve poder ser executado num computador de secretária por um especialista em Engenharia de Materiais.

Na situação atual já temos disponível o programa Tritom que implementa várias operações de processamento de dados tomográficos. Este é executado em modo texto e não fornece funcionalidades de visualização. Algumas operações de processamento foram entretanto paralelizadas e otimizadas recorrendo a multiprocessadores e um esquema de memória distribuída baseado em trocas de mensagens [3].

Para a implementação do PSE iremos utilizar o SCIRun, onde pretendemos desenvolver um pacote específico de módulos aplicáveis à tomografia. Estes módulos tirarão partido de várias estratégias de paralelização com vista a aumentar o seu desempenho.

3.1 Módulos de visualização

A nível da visualização de dados existe um trabalho preliminar feito no âmbito de uma dissertação de Mestrado [4]. Foi utilizado o OpenDX [18] na implementação de um protótipo para um ambiente de resolução de problemas, o que significa que já possuía algumas funcionalidades para a visualização dos dados.

Os dados manuseados nas imagens tomográficas correspondem à cor de um voxel, logo tipicamente podemos utilizar vetores 3D para armazenamento. No entanto, verificamos que ao longo das operações de processamento os dados não variam muito, pelo menos em termos da percentagem total do volume da amostra. Sendo assim, esperamos ir para além da utilização de vetores para armazenamento de dados. É nossa intenção recorrer a estruturas de dados mais avançadas como por exemplo: *Octree*, *Kd-tree* e *Interval tree*. Estas estruturas de dados devem suportar de forma eficiente os algoritmos de visualização de dados e, por outro lado, é necessário pensar sempre na eficiência da paralelização dos dados.

No PSE em desenvolvimento, para a visualização dos dados preconizamos essencialmente duas estratégias: Extração de isosuperfícies e renderização de volumes. Em ambos os casos, a respetiva implementação tem como princípio orientador a utilização das potencialidades dos GPU's.

Nas funções de processamento de imagem esperamos utilizar: filtros em imagens 2D, tais como: Gaussiano, Média, Máximo, Mínimo, Variância, Mediana, etc.

Finalmente, esperamos também desenvolver um conjunto de ferramentas de uso geral, tais como: Histogramas de dados, extração de planos da imagem; gestão de parâmetros no GPU; estabelecimento de níveis de detalhe, etc.

Na Figura 6, podemos observar do lado direito uma amostra tomográfica com a dimensão de 400x400x400 voxéis depois de aplicada a operação de segmentação. Esta operação foi feita através de um módulo desenvolvido para o efeito e que considerou os voxéis com valores entre 118 e 140 como cinzentos nível 127. Todos os valores superiores a 140 ficaram brancos e todos os valores inferiores a 118 ficaram pretos. Do lado esquerdo da imagem temos o histograma da amostra antes de submetida a qualquer processamento, no qual são visíveis dois picos, um centrado no valor 140 e outro de muito menor dimensão centrado no valor 80.

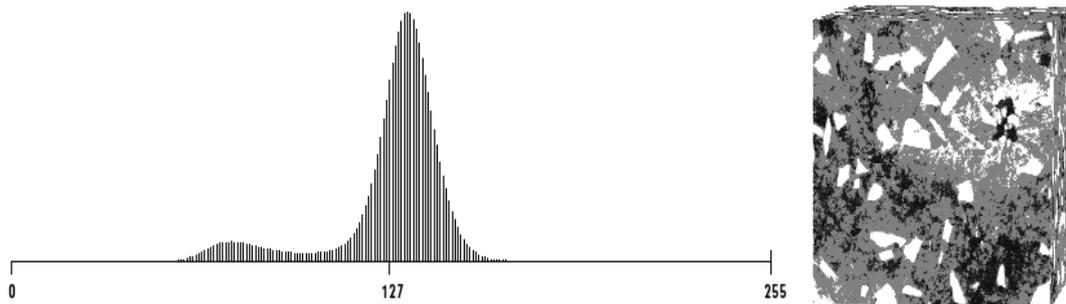


Figura 6 – Visualização em SCIRun de uma amostra tomográfica e respetivo histograma

3.2 Módulos dedicados à tomografia

Tomando como ponto de partida o programa desenvolvido Tritom, vamos desenvolver um conjunto de módulos para dar resposta às seguintes situações: Filtração de dados; regiões vizinhas conexas (retorno de todos os voxéis de um objeto individual); percolação de dados; segmentação de dados; estatísticas de granulometria; limpeza da imagem (eliminação de ruído e interferências).

Na primeira fase do trabalho foram já desenvolvidos alguns módulos englobados nesta categoria, dos quais se destacam: a segmentação, a granulometria e a estatística. O módulo de segmentação permite dividir a imagem em 2 ou 3 níveis de cor (branco, preto e cinzento) de acordo com a especificação do cientista. O módulo Estatística permite conhecer em cada momento o número de voxels pretos, brancos e cinzentos existentes na amostra. Finalmente, o módulo granulometria permite identificar o número de partículas existentes numa amostra, o seu volume e superfície, uma caixa de delimitação da partícula e o número de voxels internos. Para realizar esta operação é necessário que exista uma clara distinção entre as partículas (cor preta) e a matriz base (cor branca). Note-se que são necessárias operações precedentes para o estabelecimento deste contraste.

O módulo Granulometria (Figura 7) que foi implementado permite obter uma estimativa do número e tamanho das partículas encontradas na matriz metálica. Este módulo baseia-se numa estratégia de propagação a partir de um voxel preto inicial, tentando encontrar a maior região conexas. Neste módulo, e de forma a eliminar ruído, é possível definir o tamanho mínimo para que valor uma região possa ser considerada uma partícula. Por omissão este valor é 100 voxels, no entanto este valor pode ser alterado pelo cientista com base no tamanho médio esperado para as partículas.

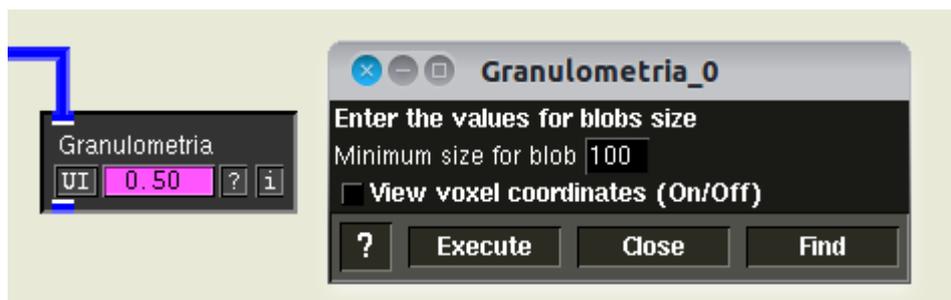


Figura 7 – Módulo Granulometria e interface gráfica respetiva

Para a codificação deste módulo utilizou-se também a biblioteca de funções geométricas *Geometric tools* [16]. O módulo fornece informação sobre o número, volume, área, número de voxels internos e a *bounding box* que delimita cada partícula.

Foi também desenvolvido um módulo que permite visualizar após cada operação de processamento o número de voxels pretos, brancos ou cinzentos na imagem.

3.3 Paralelização de módulos

Tal como referido em 2.3, a utilização de GPGPUs permite conciliar os requisitos de elevadas capacidades de processamento com o objetivo de disponibilizar uma estação de trabalho pessoal para o especialista em Materiais.

A arquitetura alvo para o PSE é um computador pessoal que, além do tradicional CPU *multicore*, tem também integrado um dispositivo gráfico que pode ser utilizado para computações não ligadas à renderização de imagens.

A atual versão do Tomo-GPU é construída sobre o SCIRun. Na secção 2.1.2 foi feita uma descrição do processo de adição de novos módulos. Desde que estas regras sejam seguidas, é

trivial adicionar ao Tomo-GPU módulos escritos em C/C++ que tirem partido de aceleradores como os GPGPUs. Na tese [23] são apresentados resultados que mostram que o uso de um acelerador como o Cell BE [22][21] podem conduzir a reduções consideráveis do tempo de execução de operações como a limpeza da imagem ou a granulometria face aos que são conseguidos num processador convencional.

Na secção 2.3 foram analisadas as alternativas para o desenvolvimento de aplicações para GPGPUs. Entre estas, foi escolhido o OpenCL pelos seguintes motivos:

- Permite a execução em ambientes heterogéneos (CPU + GPU);
- Foi desenvolvido para ser independente do fornecedor de hardware.

Estão em desenvolvimento módulos para o processamento de dados e visualização nos GPGPUs usando o OpenCL [24] ; em simultâneo ensaiam-se formas de representar a matriz a três dimensões que representa a imagem tomográfica que sejam adequados a diferentes operações de processamento – por exemplo usando Octrees [25]. O suporte destas estruturas de dados deve acomodar os três níveis de armazenamento já referidos – disco, RAM do CPU e RAM do GPU.

4. Conclusões e trabalho futuro

A construção de software para a caracterização de materiais compósitos que executa num computador pessoal equipado com um GPGPU é o objetivo principal do trabalho em curso. A disponibilidade deste software é importante para o especialista de materiais, uma vez que lhe permite evitar a aquisição de software dispendioso e/ou o acesso a infraestruturas computacionais remotas que possuem elevadas latências na sua utilização.

A disponibilização de um conjunto variado de funcionalidades para caracterização de materiais compósitos num computador pessoal equipado com GPGPUs, garantindo tempos de execução das operações muito curtos e com baixa latência é a característica original do Tomo-GPU.

O desenvolvimento em curso deste PSE é da responsabilidade de uma equipa multidisciplinar que inclui especialistas na paralelização de aplicações, computação gráfica e visualização e ciência de materiais. Um aspecto fundamental do projeto é a avaliação periódica da funcionalidade do sistema por especialistas em Engenharia de Materiais.

Bibliografia

[1] T. Cadavez, S. C. Ferreira, P. Medeiros, P. J. Quaresma, L.A. Rocha, A. Velhinho, and G. Vignoles. A graphical tool for tomographic characterization of microstructural features on metal matrix composites. *International Journal of Tomography and Statistics*, 2010.

[2] G.L. Vignoles. Image segmentation for phase-contrast hard X-ray cmt of c/c composites. *Carbon*, 43(2), 2000.

[3] P. J. Quaresma. Processamento paralelo aplicado a um problema de engenharia de materiais. Master's thesis, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2007.

- [4] T. Cadavez. Análise de imagens tomográficas: Visualização e paralelização do processamento. Master's thesis, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2008.
- [5] ImageJ. <http://rsbweb.nih.gov/ij>.
- [6] Ryoji Tsuchiyama, Takashi Nakamura, Takuro Lizuka, Akihiro Asahara, Satoshi Miki. The OpenCL Programming Book: Parallel Programming for MultiCore CPU and GPU. Fixstars Corporation, 2010.
- [7] VTK User's Guide Version 11. Kitware, inc, 2010.
- [8] D. Kirk, W. Hwu. Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, 2010.
- [9] Steven G. Parker, David M. Weinstein, and Christopher R. Johnson. The SCIRun computational steering software system. Technical report, Utah University, 2004.
- [10] Kitware's Software Developer's Quarterly. Kitware, Inc., 2006.
- [11] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH Computer Graphics, 21(4):163–169, 1987.
- [12] Gary Bradski and Adrian Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, 2008.
- [13] B. Chapman. Using OpenMP: Portable shared Memory Parallel Programming (Scientific and Engineering Computation). MIT Press, 2007
- [14] David Butenhof. Programming with POSIX Threads (Addison-Wesley Professional Computing). Addison-Wesley, 1997
- [15] William Gropp. Using MPI-2: Portable Parallel Programming with the Message-passing Interface (Scientific & Engineering Computation). MIT Press, 2000
- [16] P. Schneider, D. Eberly. Geometric tools for computer graphics. Morgan Kaufmann, 2003
- [17] Robert B. Haber and David A. McNabb, "Visualization Idioms: A Conceptual Model for Scientific Visualization Systems", *Visualization in Scientific Computing*, IEEE, pp 74-93, 1990.
- [18] IBM. IBM Visualization Data Explorer Programmer's guide, 7 edition, May 1997
- [19] Michael Knaup, Sven Steckmann, Olivier Bockenbach, Marc Kachelrie. Tomographic Image Reconstruction using the Cell Broadband Engine (CBE) General Purpose Hardware, Berlin Germany.

[20] Sander van der Maar, K. Joost Batenburg, Jan Sijbers. Experiences with Cell-BE and GPU for tomography, IBBT-Vision Lab, Department of Physics - University of Antwerp, Belgium

[21] A. Arevalo, R.M. Martinata, M. Pandian, E. Peri, K. Ruby, F. Thomas, C. Almond, Programming the Cell Broadband Engine™ Architecture: Examples and Best Practices, IBM Redbook, 2008

[22] J. Hill, Scientific Programming on the Cell using ALF, Relatório Técnico HPCxTR0708, Edinburgh Parallel Computing Centre, 2008

[23] P.Paiva, Utilização do processador Cell para o processamento de imagens tomográficas de materiais compósitos, Tese de Mestrado, Dep. Informática, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Abril de 2011

[24] B. Preto, Conceção e implementação de algoritmos de processamento de imagens tomográficas vocacionados para GPGPUs, Tese de Mestrado (a publicar), Dep. Informática, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Setembro de 2011

[25] S. Lefèvre, . S. Hornus, F. Neyret, Octree Textures in the GPU, cap. 37, GPU Gems 2 Programming Techniques for High-Performance Graphics and General-Purpose Computation, Addison-Wesley, 2005