

Unidade de processamento gráfico no diagnóstico auxiliado por computador

Graphics processing unit on computer-aided diagnosis

Unidad de procesamiento gráfico en diagnóstico asistido por computador

José Raniery Ferreira Junior¹, Marcelo Costa Oliveira²

RESUMO

Descritores: Diagnóstico por Imagem; Interpretação de Imagem Assistida por Computador; Análise e Desempenho de Tarefas

A interpretação de imagens é uma tarefa complexa na rotina clínica devido a diversos fatores. Logo, é importante integrar ferramentas computacionais ao diagnóstico por imagem, conhecidas como CAD (*computer-aided diagnosis*), pois elas possuem capacidade de auxiliar o especialista fornecendo uma segunda opinião na interpretação do caso. Contudo, sistemas CAD devem fornecer uma resposta rápida ao especialista para não levar risco ao paciente. O objetivo deste trabalho é realizar um levantamento bibliográfico para demonstrar a efetividade da unidade de processamento gráfico (GPU – *graphics processing unit*) na otimização de ferramentas CAD e de diversas aplicações de imagens médicas. Diversos trabalhos foram encontrados na literatura nas mais diversas áreas, como recuperação de imagens baseada em conteúdo, classificação de imagens, detecção de objetos, reconstrução de imagens e renderização de regiões. Com isso foi demonstrada a efetividade do uso do paralelismo massivo realizado por GPUs em sistemas de auxílio computadorizado à medicina.

ABSTRACT

Keywords: Diagnostic Imaging; Image Interpretation Computer-Assisted; Task Performance and Analysis

The image interpretation process is a complex task in the clinical routine due to several aspects. Therefore, it is important to integrate computational tools to diagnostics imaging, known as computer-aided diagnosis (CAD), because they have the ability to aid specialists by providing a second opinion on interpreting the case. However, CAD systems must provide a quick response to the specialists in order to not put the patient in risk. The goal of this work is to perform a bibliographic survey to demonstrate the efficiency on graphics processing unit (GPU) optimization of CAD tools and several medical image applications. Several works were found in literature on wide range of fields, like content-based image retrieval, image classification, object detection, image reconstruction, and region rendering. Therefore, GPU massive paralelism was demonstrated to be effective on systems for computer-aided medicine.

RESUMEN

Descriptores: Diagnóstico por Imagen; Interpretación de Imagen Asistida por Computador; Análisis y Desempeño de Tareas

La interpretación de imágenes es complejo en la práctica clínica debido a varios factores. Por tanto, es importante integrar herramientas computacionales para diagnóstico por imagen, conocido como CAD (*computer-aided diagnosis*) porque tienen la capacidad de ayudar al especialista proporcionando una segunda opinión en la interpretación del caso. Todavía, sistemas CAD deben proporcionar una respuesta rápida a especialista para no tomar riesgos para el paciente. El objetivo de este trabajo es revisar la literatura para demostrar la efectividad de la optimización por la unidad de procesamiento gráfico (GPU - *graphics processing unit*) de herramientas CAD y varias aplicaciones de imagen médica. Se han encontrado varios trabajos en la literatura en las más diversas áreas, como recuperación de imágenes basada en contenido, clasificación de imágenes, detección de objetos, reconstrucción de imágenes y renderización de regiones. Con ello se demostró la efectividad del uso del paralelismo masivo realizado por GPUs en sistemas de auxilio computerizado a la medicina.

¹ Mestre em Informática pela Universidade Federal de Alagoas - UFAL, Maceió (AL), Brasil.

² Professor Associado do Instituto de Computação da Universidade Federal de Alagoas - UFAL, Maceió (AL), Brasil.

INTRODUÇÃO

O diagnóstico por imagem é uma tarefa complexa na rotina clínica, e o processo de interpretação de imagens é um desafio reconhecido mesmo para especialistas experientes⁽¹⁾. Radiologistas são limitados pela capacidade humana de análise, sofrendo influências de fatores externos (e.g. ruído e luminosidade) e internos, como o nível de treinamento e condições psicológicas (e.g. fadiga e pressão). Portanto, o especialista está sujeito a erros de detecção (reconhecimento de um padrão não-normal) ou de diagnóstico (identificação da natureza do achado radiológico), resultando em variabilidade intra e inter-observador⁽²⁾.

Além disto, radiologistas enfrentam outro grande desafio para manter a alta precisão na interpretação das imagens: o grande crescimento do volume de imagens médicas geradas. Um hospital de grande porte produz milhões de imagens todos os anos, levando a produção total de grandes centros à casa dos milhares de *petabytes*⁽³⁾.

Diante destes desafios no diagnóstico por imagem, é importante integrar ferramentas computacionais ao processo de interpretação de imagens. O objetivo do CAD (*computer-aided diagnosis*) é melhorar a acurácia diagnóstica e aprimorar a consistência na interpretação do diagnóstico por imagem, mediante o uso da sugestão de resposta fornecida por um computador⁽⁴⁾. Sistemas CAD de detecção e diagnóstico diferencial aumentam a capacidade do radiologista na interpretação das imagens e na diferenciação entre regiões anatômicas e tecidos. Logo, a adoção de ferramentas computacionais no auxílio ao diagnóstico não é mais uma opção, mas sim uma necessidade.

Em face do crescimento do volume de dados gerados atualmente e da relevância das ferramentas CAD, é importante utilizar tecnologias que façam uso de paralelismo para garantir o alto desempenho na execução das tarefas. A alta performance em termos de tempo de execução no auxílio computadorizado ao diagnóstico é fundamental em diversas aplicações clínicas, pois pode acarretar diretamente na saúde do paciente. Por exemplo, em um acidente vascular cerebral, o especialista tem poucos minutos para a tomada de decisão emergencial do caso. Logo, uma ferramenta CAD para estes tipos de doença deve necessariamente oferecer uma resposta rápida ao especialista para não oferecer mais riscos ao paciente. Além disso, o baixo tempo de processamento do *software* é importante porque, caso contrário, pode acarretar no desinteresse do especialista em utilizar a ferramenta.

Algumas infraestruturas paralelas foram empregadas na literatura para garantir o alto desempenho das aplicações médicas, e.g. computação em nuvem e grades computacionais⁽⁴⁻⁵⁾. Contudo, estas soluções dependem de uma conexão disponível, estável e de alto desempenho para transmissão dos dados, o que nem sempre é possível. Além disso, elas são suscetíveis a ataques externos, comprometendo a privacidade dos dados do paciente. Apesar de sua natureza gráfica, as unidades de processamento gráfico (GPU - *graphics processing unit*)

desenvolveram-se e passaram a possuir características de propósitos gerais de programação e não somente de aplicações gráficas. Diversos algoritmos têm feito uso do paralelismo em GPU devido a sua alta performance na execução das tarefas, facilidade em implementação e grande variedade de dispositivos, arquiteturas e plataformas. Além disso, no domínio médico, a GPU é uma abordagem confiável por questões de segurança e privacidade, visto que os dados do paciente não precisam ser trafegados para outros computadores e ambientes remotos⁽³⁾.

Neste contexto, este artigo apresenta um levantamento de trabalhos que fizeram uso do paralelismo em GPUs no diagnóstico auxiliado por computador e em diversas aplicações clínicas. Este levantamento visa apresentar a viabilidade e efetividade do desenvolvimento paralelo em GPU em algoritmos CAD.

FUNDAMENTAÇÃO TEÓRICA

O termo “computação heterogênea” tem sido empregado para designar o uso de diferentes arquiteturas em um mesmo dispositivo para se obter melhores desempenhos das aplicações que executarão neste dispositivo⁽⁶⁾. Para fazer uso do máximo poder computacional de uma arquitetura heterogênea, é necessário aplicar o conceito de paralelismo. Uma aplicação é paralela quando pode ser dividida em várias tarefas, chamadas de *threads*, que são executadas simultaneamente em diferentes núcleos processadores.

A GPU desenvolveu-se e tornou-se um dispositivo com dezenas, centenas ou milhares de núcleos com maior poder de processamento paralelo do que a CPU (*central processing unit*)⁽⁷⁾. Este poder de processamento paralelo evoluiu e passou a atingir computações de propósito geral e não apenas do escopo gráfico. O paradigma de programação paralela realizado nas unidades de processamento gráfico é chamado de Computação de Propósito Geral em GPU (GPGPU - *general purpose computing on graphics processing unit*).

GPUs são dispositivos paralelos que atuam em conjunto com as CPUs⁽⁶⁾. Contudo, as GPUs possuem sua própria hierarquia de memória e os dados devem ser transferidos através de um barramento entre GPU e CPU. Existem arquiteturas mais recentes que conseguem integrar GPU e CPU em um mesmo barramento⁽⁸⁾, e.g. AMD APU (*accelerated processing unit*), Intel HD Graphics e Nvidia Denver Project, porém elas não são abordadas neste artigo.

As principais plataformas de programação para GPU são CUDA (*compute unified device architecture*) e OpenCL (*open computing language*). Enquanto CUDA pode ser utilizada apenas em GPUs da fabricante Nvidia, o padrão aberto OpenCL atinge diversos modelos e fabricantes tanto de GPUs quanto de CPUs, através de diferentes ambientes de desenvolvimento, e.g. *frameworks*, bibliotecas ou diretivas. Além disso, ele possui uma grande comunidade de desenvolvimento, material, etc, que a torna uma das plataformas paralelas mais utilizadas. Porém, um dos seus principais problemas é manter o alto desempenho

enquanto preserva a portabilidade entre os diferentes dispositivos.

Normalmente, a melhor forma de conseguir o melhor desempenho de uma aplicação é dividir a carga de trabalho igualmente entre p núcleos⁽⁹⁾, que pode ser representado pela equação $T_p = T_s / p$, onde T_p é o tempo da execução paralela e T_s é o tempo da execução sequencial. Nestes casos, o programa paralelo possui ganho de desempenho linear. Contudo, o ganho linear é muito difícil de se obter devido ao problema do *overhead*, e.g. a transferência dos dados da memória da CPU para GPU e vice-versa. Desta forma, o ganho de desempenho de uma aplicação paralela em relação a sua execução sequencial pode ser medido pela métrica *Speedup*⁽⁹⁾, em que T_s / T_p , onde T_s é o tempo da execução sequencial, e T_p é o tempo da execução paralela.

MÉTODOS

As bases de dados utilizadas no levantamento bibliográfico foram Web of Science, Science Direct, PUB-MED, IEEE Xplore e Scholar. O período estudado compreendeu 8 anos, entre 2007 e 2015. A busca utilizou os seguintes descritores em português e inglês: *GPU*, *graphics processing unit*, *unidade de processamento gráfico*, *CAD*, *computer-aided diagnosis*, *diagnóstico auxiliado por computador*.

RESULTADOS

Recuperação de Imagens Baseada em Conteúdo

A técnica de recuperação de imagens baseada em conteúdo (CBIR - *content-based image retrieval*) tem sido descrita como uma das ferramentas CAD de diagnóstico diferencial mais promissoras, pois ela auxilia o processo de decisão clínica, recuperando em grandes bases de dados, casos similares já diagnosticados pertencentes à mesma modalidade, região anatômica e com alterações estruturais provocadas por determinadas doenças⁽⁵⁾.

Um algoritmo de CBIR otimizada por GPU para o auxílio computadorizado ao diagnóstico do câncer de pulmão baseado na recuperação de nódulos pulmonares similares foi proposto em⁽¹⁰⁾. Apesar da performance (em termos de velocidade) não ser um fator crítico no diagnóstico do câncer, o autor optou pelo aumento de desempenho para não acarretar no desinteresse do usuário em utilizar o algoritmo de CBIR com baixo rendimento em velocidade. O autor paralelizou em uma GPU modelo GeForce GT 640 a análise de similaridade dos nódulos através da função de distância Euclidiana. Para os experimentos foram utilizadas imagens de tomografia computadorizada (CT - *computed tomography*) provenientes do projeto público *Lung Image Database Consortium* e obteve *Speedups* de 20x com a plataforma OpenCL.

A recuperação otimizada de imagens similares de CT provenientes do *Lung Image Database Consortium* como ferramenta de apoio ao CAD foi realizada⁽³⁾. Uma ferramenta otimizada de apoio ao CAD é importante na rotina clínica de um hospital para que o especialista não perca tempo que poderia ser gasto na tomada de decisão diagnóstica de outro caso. Os autores utilizaram a distância

Euclidiana como métrica de similaridade das imagens e obtiveram *Speedup* de 64x, com a plataforma OpenCL para o desenvolvimento paralelo da função de distância em uma GPU modelo GeForce GT 640.

Já em outro estudo foi realizada a recuperação de imagens similares em um algoritmo de CBIR voltado para medicina, com diferentes configurações de desenvolvimento e testes. Os autores utilizaram a função de distância Quadratic (diferença no histograma entre duas imagens) para mensurar a similaridade de imagens de CT, ressonância magnética (MRI - *magnetic resonance imaging*) e radiografia provenientes de um repositório local. Os autores obtiveram ganho de desempenho em uma GPU modelo GeForce 9500 GT de 25x utilizando a plataforma paralela CUDA⁽¹¹⁾.

Classificação de Imagens

Sistemas CAD podem auxiliar o diagnóstico através da quantificação de características da imagem e sua classificação como correspondendo a padrões normais ou anormais². Alguns sistemas otimizados por GPU para o auxílio ao diagnóstico diferencial e/ou prognóstico foram propostos na literatura para a classificação de lesões radiológicas ou células cancerígenas em imagens histopatológicas⁽¹²⁻¹⁴⁾.

Outros autores propuseram um algoritmo para auxiliar o diagnóstico precoce da doença de Alzheimer baseado na classificação de imagens de MRI do cérebro. A classificação foi otimizada em uma GPU modelo Geforce GTX 480 na dispendiosa etapa de registro de imagens, sem perda significativa na qualidade da imagem resultante. A otimização se deu através do cálculo da pirâmide Gaussiana, com *Speedup* de 2x, e redimensionamento das imagens de ressonância magnética, com *Speedup* de 60x. Os exames de MRI foram obtidos pelo projeto público *Alzheimer's Disease Neuroimaging Initiative* e a plataforma de desenvolvimento paralelo empregada foi OpenCL⁽¹²⁾.

Um sistema CAD voltado para o câncer de mama utilizando imagens de mamografia e ultrassonografia foi desenvolvido em outro estudo⁽¹³⁾. A ferramenta realiza a diferenciação das lesões mamárias de acordo com a malignidade (malignas ou benignas). Os autores otimizaron as etapas de seleção das características mais relevantes das imagens através de busca exaustiva e da validação cruzada externa, etapas reutilizáveis em outras aplicações CAD. As imagens de mamografia e ultrassonografia foram obtidas em um repositório local, e CUDA foi utilizada no desenvolvimento paralelo em uma GPU modelo GeForce 580GTX obtendo um ganho de desempenho de até 150x.

Um sistema para o auxílio ao prognóstico do neuroblastoma para patologistas foi otimizado por outros pesquisadores. A ferramenta CAP (*computer-aided prognosis*) classifica as células dos tumores em pobre ou rico em estroma de Schwann. O diagnóstico das células tumorais demanda muito tempo ao especialista devido a grande quantidade de imagens histopatológicas microscópicas de WSI (*whole slide images*) geradas. Logo, um sistema CAP otimizado mostra-se de grande importância na rotina laboratorial. As imagens de WSI utilizadas nos

experimentos foram provenientes de um repositório local. A biblioteca OpenGL foi utilizada para a otimização em uma GPU modelo GeForce 8800GTX no processo de extração de características para o classificador k -nn (*k-nearest neighbors*) e foi obtido um *Speedup* de 45x⁽¹³⁾.

Detecção de Objetos e Regiões

Sistemas CAD de detecção auxiliam a localização de padrões anormais através da varredura da imagem pelo computador². Além da localização (semi-)automática de lesões radiológicas, sistemas CAD de detecção podem auxiliar na identificação de regiões anatômicas e histológicas ou objetos cirúrgicos para fins de segmentação e acompanhamento (*tracking*)⁽¹⁵⁻¹⁷⁾.

Outros estudiosos detectaram regiões potencialmente invasivas de câncer de mama para um sistema CAP com o objetivo de otimizar o trabalho de patologistas. As regiões foram identificadas em imagens histopatológicas de WSI provenientes de um repositório local e classificadas para eliminação de falso-positivos, ou seja, eliminar regiões não-invasivas que foram identificadas como invasivas. A otimização foi realizada durante as etapas de *sparse coding* da detecção e de treinamento do classificador SVM (*support vector machine*) através da biblioteca *cuSVM* e da plataforma CUDA. A solução paralela em uma GPU modelo GeForce 9400M obteve um ganho de desempenho de 25x para o *sparse coding* e 17x para o SVM em relação à solução sequencial da CPU⁽¹⁵⁾.

Um algoritmo otimizado de detecção de vasos sanguíneos e vias aéreas em diferentes modalidades de imagens médicas foi desenvolvido. O algoritmo pode ser utilizado em diversos procedimentos pré-cirúrgicos, e.g. broncoscopia, e com isso, acaba-se reduzindo o imprescindível tempo requerido durante as operações cirúrgicas. A otimização foi realizada na segmentação multi-tarefa dos tubos estruturais através da plataforma OpenCL em uma GPU modelo AMD HD7970. Os experimentos foram realizados em imagens de CT, MRI e de ultrassonografia de um repositório local. Cada modalidade de imagem obteve um *Speedup* diferente devido às diferentes resoluções espaciais e amplitudes da escala de cinza. Os ganhos de desempenho com as imagens de CT, MRI e ultrassom foram de 35x, 40x e 45x, respectivamente⁽¹⁶⁾.

A segmentação da vértebra em imagens de radiografia para a detecção de anormalidades no sistema músculo-esquelético foi realizada. A etapa de extração recursiva do contorno da vértebra foi otimizada em uma GPU modelo Tesla C1060 pela plataforma CUDA e pela biblioteca OpenGL. As imagens radiográficas foram obtidas pelo projeto NHANES II da agência americana de medicina *National Library of Medicine*. Os autores obtiveram um *Speedup* de 6x com o paralelismo da recursividade na GPU⁽¹⁷⁾.

Reconstrução de Imagens

Alguns trabalhos foram encontrados na literatura em que os autores focaram na reconstrução de imagens médicas de diferentes modalidades⁽¹⁸⁻²⁰⁾. A reconstrução

das imagens pode ser útil na visualização de tecidos sobrepostos (e.g. mamas densas) e assim auxiliar o processo de diagnóstico por imagem de diversas doenças, como por exemplo, o câncer de mama⁽²⁰⁾. Contudo, a reconstrução de imagens demanda um alto custo computacional, o que pode dificultar a etapa de interpretação das imagens no diagnóstico clínico pelo elevado tempo de processamento. Desta forma, é necessária a utilização de paralelismo em GPU para aumentar o desempenho das funções.

A reconstrução de imagens de CT através do algoritmo MART (*multiplicative algebraic reconstruction technique*) foi realizada em outro estudo. A implementação paralela do algoritmo foi realizada pela plataforma CUDA e um repositório local de imagens de CT foi utilizado nos experimentos. O ganho de desempenho de uma GPU modelo Tesla M2070 em relação à CPU foi de 28x⁽¹⁸⁾. Já outros, realizaram a reconstrução de imagens de CT de feixe cônico (CBCT - *cone beam computed tomography*), bastante utilizadas no domínio odontológico. O algoritmo de *weighted backprojection* foi paralelizado pelas plataformas CUDA e OpenCL com o objetivo de avaliar o desempenho das duas. CUDA obteve melhor desempenho com *Speedup* de 130x em uma GPU modelo Tesla C2070, porém o OpenCL obteve alto ganho de desempenho com *Speedup* de 80x e ainda permitiu a portabilidade do código-fonte e interoperabilidade de diferentes dispositivos⁽¹⁹⁾.

Imagens de tomossíntese mamária digital (DBT - *digital breast tomosynthesis*) foram reconstruídas utilizando dois algoritmos iterativos diferentes: MLEM (*maximum likelihood - expectation maximization*) e OSEM (*ordered subsets - expectation maximization*)⁽²⁰⁾. O paralelismo foi executado em ambos os algoritmos pela plataforma CUDA e as imagens de DBT foram obtidas de um repositório local. O *Speedup* de uma GPU modelo Quadro FX3800 obtido foi de 6x em relação à CPU, com diferenças pouco significativas nas imagens resultantes das duas abordagens.

Renderização de Regiões

A renderização de regiões anatômicas são importantes, pois eles podem auxiliar procedimentos pré e intra-cirúrgicos, e.g. neurocirurgia para remoção de lesões⁽²⁰⁾. Alguns trabalhos foram encontrados na literatura que realizam o rendering de regiões anatômicas⁽²¹⁻²²⁾. Os autores realizaram a reconstrução de fibras nervosas neurais por propagação de linhas. Foram utilizadas imagens de ressonância magnética de tensores de difusão (DTMRI - *diffusion tensor magnetic resonance imaging*) provenientes de um repositório local. A reconstrução otimizada das fibras nervosas neurais foi realizada pela plataforma CUDA. O *Speedup* obtido foi de 38x com uma GPU modelo 8800 GTS⁽²¹⁾. Já em outro estudo os autores obtiveram melhor ganho de desempenho com *Speedup* de 40x com uma GPU modelo GeForce GTX 470 em relação às implementações em CPU⁽²²⁾.

Também foi realizada a otimização da reconstrução de fibras nervosas neurais por propagação de linhas em imagens de DTMRI⁽²³⁾. Contudo, os autores utilizaram a biblioteca gráfica OpenGL para o desenvolvimento

paralelo em uma GPU modelo GeForce GTX 280. O ganho de desempenho neste trabalho foi de 44x em relação à solução sequencial.

A Tabela 1 apresenta um resumo das aplicações de CAD de detecção, CAD de diagnóstico diferencial, CAP, entre outras.

DISCUSSÃO

O CAD continua evoluindo e desenvolvendo novos métodos para diminuir suas limitações. Neste trabalho, foi destacado o problema do tempo de execução/processamento de alguns algoritmos que exigem um dispendioso custo computacional. Na rotina clínica de um hospital ou laboratório, é importante a resposta rápida de um *software* CAD para que o usuário do sistema não perca seu tempo que poderia ser gasto em outro caso, resulte no desinteresse do usuário pelo *software* ou ofereça risco ao paciente em casos de urgência.

As soluções paralelas em GPU conseguiram otimizar o tempo de execução nos mais diversos casos apresentados e apresentaram um *Speedup* mínimo de 2x. Porém, a integração da GPU e da CPU em um mesmo barramento pode aumentar ainda mais a performance do paralelismo visto que os dados de entrada não precisam ser copiados para a memória global da GPU (Seção II). A GPU poderá acessar diretamente os dados de entrada na memória RAM da CPU, logo, não há transferência de dados CPU-GPU. O uso da computação heterogênea, ou seja, de várias arquiteturas e dispositivos diferentes integrados por uma plataforma comum (e.g. OpenCL), também pode fornecer maiores ganhos no desempenho dos algoritmos CAD, devido ao aumento do poder computacional paralelo.

Com o surgimento das características radiômicas (*radiomics*, i.e. grande número de atributos de imagens aliados com outras informações do paciente para

caracterizar lesões), além do processamento de imagens radiológicas e histopatológicas, também é esperado o uso de GPU na análise de imagens moleculares (genoma, proteína e metabolismo) e fisiológicas. Uma das características da análise radiômica é a grande quantidade de atributos extraídos das imagens e do material genético. Logo, leva-se um tempo considerável na extração dessas características, o que pode interferir no desempenho de um sistema CAD.

Por fim, espera-se a avaliação dos sistemas CAD otimizados por GPU em ambientes reais de serviços de radiologia, patologia e cirurgia. Dessa forma, após a validação clínica dos algoritmos na rotina clínica de um hospital ou laboratório biomédico, a GPU poderá ser integrada de vez no diagnóstico auxiliado por computador e no processamento de imagens biomédicas.

CONCLUSÃO

Este trabalho apresentou um levantamento bibliográfico que demonstrou a efetividade do uso do paralelismo massivo realizado por GPUs em sistemas de auxílio computadorizado ao diagnóstico e prognóstico de doenças, a partir do processamento e análise de imagens radiológicas e histopatológicas.

Com o grande volume de dados criados ultimamente, a GPU ainda possui grande aplicabilidade em ser utilizada no contexto em *Big Data* (rápida criação de grandes volumes de dados heterogêneos), com a integração de outros dados em saúde, como o prontuário eletrônico do paciente. Dessa forma, a GPU torna-se o paradigma de desenvolvimento paralelo mais apropriado no processamento de dados médicos, devido ao seu alto desempenho, facilidade de implementação, diversidade de plataformas de desenvolvimento, segurança ao paciente e processamento em múltiplos domínios computacionais.

Tabela 1- Aplicações da medicina por imagem otimizadas por GPU.

Categoria	Aplicação	Plataforma
Recuperação	CBIR como ferramenta de apoio ao diagnóstico clínico	CUDA ¹¹ , OpenCL ^{3,10}
Classificação	Sistemas de apoio ao diagnóstico diferencial e prognóstico	CUDA ¹³ , OpenCL ¹² , OpenGL ¹⁴
Detecção	Sistemas de apoio à detecção e segmentação	CUDA ^{15,17} , OpenCL ¹⁶ , OpenGL ¹⁷
Reconstrução	Visualização e rendering de imagens	CUDA ^{18,19,20} , OpenCL ¹⁹
Rendeização	Acompanhamento (<i>tracking</i>) de regiões anatômicas	CUDA ^{21,22} , OpenGL ²³

REFERÊNCIAS

- Doi K. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Comput Med Imaging Graph*. 2007;31(4-5):198-211.
- Azevedo-Marques P. Diagnóstico auxiliado por computador na radiologia. *Radiol Bras*. 2001;34(5):285-93.
- Ferreira Jr J, Oliveira M, Freitas A. Performance evaluation of medical image similarity analysis in a heterogeneous architecture. *Proceeding of the 2014 IEEE 27th International Symposium on Computer-Based Medical Systems* [Internet]; 2014 May 27-29. New York, NY. p.159-64. Available from: dx.doi.org/10.1109/CBMS.2014
- Xue H, Inati S, Sørensen T, Kellman P, Hansen M. Distributed MRI reconstruction using gadgetron-based cloud computing. *Magn Reson Med* [Internet]. *Magn Reson Med*. 2015 Mar;73(3):1015-25. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24687458>
- Oliveira M, Cirne W, Azevedo PM Towards applying content-based image retrieval in the clinical routine. *Futur Gener Comput Syst* [Internet]. 2007;23:466-74. Available from: dl.acm.org/citation.cfm?id=1221902
- Stringhini D, Gonçalves R, Goldman A. Introdução à computação heterogênea. *Anais da XXXI Jornada de Atualização em Informática do XXXII Congresso da Sociedade Brasileira de Computação* [Internet]; 2012 Jul 16-

19. Curitiba, PR. Disponível em: www.lbd.dcc.ufmg.br/colecoes/jai/2012/007.pdf
7. Liu L. Computing infrastructure for big data processing. *Front Comput Sci* [Internet]. 2013;7(2):165-70. Available from: dx.doi.org/10.1007/s11704-013-3900-x
 8. Marowka A. Extending Gustafson-Barsis's law for dual-architecture computing. *Proceeding of the 11th International Conference Parallel Processing and Applied Mathematics*. 2015 Set 6-9. Krakow, Poland. 2016 Part I. Available from: dx.doi.org/10.1007/978-3-319-32152-3_12
 9. Pacheco PS. *An introduction to parallel programming*. Burlington: Elsevier; 2011.
 10. Ferreira Jr JR. *Auxílio computadorizado ao diagnóstico do câncer de pulmão otimizado por GPU* [dissertação]. Maceió: Universidade Federal de Alagoas. Instituto da Computação. Programa de Pós-Graduação em Informática; 2015.
 11. Yadav K, Srivastava A, Ansari M. GPU-parallel implementation of color based medical image retrieval in compressed domain. *Int J Comput Applic* [Internet]. 2011;8-14. Available from: research.ijcaonline.org/dia/number1/SPE314T
 12. Shamonin D, Bron E, Lelieveldt B, Smits M, Klein S, Staring M. Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. *Front Neuroinform* [Internet]. 2013;7:50. Available from: www.ncbi.nlm.nih.gov/pubmed/24474917
 13. Sidiropoulos K, Kostopoulos S, Glotsos D, Athanasiadis E, Dimitropoulos N, Stonham J, et al. Multimodality GPU-based computer-assisted diagnosis of breast cancer using ultrasound and digital mammography images. *Int J Comput Assist Radiol Surg* [Internet]. 2013;8(4):547-60. Available from: www.ncbi.nlm.nih.gov/pubmed/23354971
 14. Ruiz A, Sertel O, Ujaldon M, Catalyurek U, Saltz J, Gurcan M. Pathological image analysis using the GPU: Stroma classification for neuroblastoma. *Proceeding of the 2007 IEEE Int Conf Bioinforma Biomed BIBM* [Internet]. 2007 Nov 2-7; San Jose, CA. EUA. Available from: dx.doi.org/10.1109/BIBM.2007.15
 15. Huang C, Veillard A, Roux L, Loménie N, Racoceanu D. Time-efficient sparse analysis of histopathological whole slide images. *Comput Med Imaging Graph* [Internet]. 2011;35(7-8):579-91. Available from: www.ncbi.nlm.nih.gov/pubmed/21145705
 16. Smistad E, Elster A, Lindseth F. GPU accelerated segmentation and centerline extraction of tubular structures from medical images. *Int J Comput Assist Radiol Surg* [Internet]. 2014;9(4):561-75. Available from: www.ncbi.nlm.nih.gov/pubmed/24177985
 17. Lecron F, Mahmoudi S, Benjelloun M, Mahmoudi S, Manneback P. Heterogeneous computing for vertebra detection and segmentation in x-ray images. *Int J Biomed Imaging* [Internet]. 2011. Available from: dx.doi.org/10.1155/2011/640208
 18. Bajpai M, Gupta P, Munshi P. Fast multi-processor multi-GPU based algorithm of tomographic inversion for 3D image reconstruction. *Int J H Perform Comput Applic* [Internet]. 2015;29(1):64-72. Available from: dl.acm.org/citation.cfm?id=2733371
 19. Mukherjee S, Moore N, Brock J, Leeser M. CUDA and OpenCL implementations of 3D CT reconstruction for biomedical imaging. *2012 IEEE Conf High Perform Extrem Comput* [Internet]. 2012;1-6. Available from: ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6408674
 20. Azevedo B. *Reconstrução/Processamento de imagem médica com GPU em tomossíntese* [dissertação]. Lisboa: Universidade Nova de Lisboa. Faculdade de Ciências e Tecnologia; 2011.
 21. Mittmann A. *Tractografia em tempo real através de unidades de processamento gráfico* [dissertação]. Florianópolis: Universidade Federal de Santa Catarina. Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação; 2009.
 22. Aart E, Sepasian N, Jalba A, Vilanova A. CUDA-Accelerated Geodesic Ray-Tracing for Fiber Tracking. *Int J Biomed Imaging* [Internet]. 2011. Available from: dl.acm.org/citation.cfm?id=2070256
 23. Köhn A, Klein J, Weiler F, Peitgen H. A GPU-based fiber tracking framework using geometry shaders. *Proceeding of the Society of Photo-Optical Instrumentation Engineers* [Internet]. 2009 Feb 8-9; Lake Buena Vista, Florida, USA. Available from: dx.doi.org/10.1117/12.812219