

Avaliação de Desempenho de Atributos de Textura 3D e Nitidez de Borda 3D na Recuperação de Nódulos Pulmonares Similares

Lucas Lins Lima, José Raniery Ferreira Junior, Marcelo Costa Oliveira
Instituto de Computação (IC)
Universidade Federal de Alagoas (UFAL)
Laboratório de Telemedicina e Informática Médica (LaTIM)
Hospital Universitário Prof. Alberto Antunes (HUPAA/EBSERH/UFAL)

Resumo— O câncer de pulmão é o maior causador de câncer em todo o mundo, e aliado a isso a produção de imagens médicas tem sido cada vez maior no campo médico. Para combater esses dois desafios, a ferramenta de Recuperação de Imagens Baseada em Conteúdo (CBIR) pode fornecer apoio à decisão aos especialistas, permitindo-lhes encontrar nódulos similares num banco de dados a uma imagem de consulta. Porém, um passo crucial da CBIR é a etapa de Análise de Similaridade entre Imagens (ISA), e um dos desafios da ISA é o tempo de processamento para executar todas as comparações. Este artigo apresenta uma comparação de performance entre duas técnicas de atributos: Atributos de Textura 3D e Nitidez de Borda 3D, e também ambas concatenadas, tanto na CPU como na GPU com o uso do framework OpenCL, e a Distância Manhattan para fazer as comparações de similaridade.

Palavras-chaves—Recuperação de Imagens Baseada em Conteúdo, Unidade de Processamento Gráfico.

Abstract— The lung cancer is the major causer of cancer worldwide and allied to this, medical imaging production has been increasingly growing in the medical field. To combat these two challenges, the Content-Based Image Retrieval (CBIR) tool can provide decision support to experts, enabling them find similar images in a database of images from a reference image. However, a crucial step of CBIR is the stage of Image Similarity Analysis (ISA), and one of the ISA's challenge is the processing time to perform all comparisons. This article presents a performance comparison between two techniques of attributes: 3D Texture Attributes and 3D Margin Sharpness, and also both concatenated, both in CPU and in GPU with the use of the OpenCL framework, and the Manhattan Distance to do the comparisons of similarity.

Index Terms—Content-Based Image Retrieval, Graphics Processing Unit.

I. INTRODUÇÃO

O câncer de pulmão é o principal causador de morte por câncer em todo mundo e é considerado um dos principais problemas a ser enfrentado num país industrialmente desenvolvido [1]. Assim, a detecção precoce de nódulos pulmonares malignos é um passo importante no tratamento do câncer.

O volume de dados de imagens médicas gerado para auxiliar ao diagnóstico e terapia dos tratamentos nos grandes centros de radiologia está em crescimento constante, chegando a ser produzido cerca de 2 Terabytes de dados ao ano. Isto é motivado pela crescente importância dada a imagem no diagnóstico do paciente [2].

Sistemas computacionais de auxílio ao diagnóstico (CAD) tem se mostrado uma ferramenta importante com o objetivo principal de melhorar a precisão do diagnóstico, funcionando assim como uma ferramenta de auxílio ao diagnóstico mediante o uso da resposta do computador como referência [3]. E dentro de sistemas CAD, há ferramentas para auxiliarem os radiologistas, como é o caso da ferramenta de Recuperação de Imagens Baseada em Conteúdo (CBIR), que nos últimos anos tem sido uma das técnicas de visão computacional mais estudada [4].

A CBIR dá suporte a decisão do radiologista, permitindo que a partir de uma imagem de referência ele encontre imagens similares presentes num banco de dados de imagens. A razão para apresentar imagens similares se deve ao fato de que os radiologistas aprendem habilidades de diagnóstico através da observação de muitos casos clínicos durante suas rotinas hospitalar, e os seus conhecimentos adquiridos a partir da impressão visual de imagens com várias doenças constitui a base para os seus diagnósticos [5].

Um passo da CBIR e de extrema importância, é a Análise de Similaridade entre as Imagens (ISA), e um dos desafios da ISA é justamente o tempo de processamento levado para realizar a comparação entre as imagens [6]. Esta etapa de comparação entre uma imagem de referência e as imagens contidas num banco de dados utilizando apenas um computador pode exigir muito mais tempo do que é razoável para sistemas CAD.

Para subtrair o tempo de processamento levado pela ISA na CBIR, trabalhos baseados em Unidades de Processamento Gráfico (GPU) e de Computação de Propósito Geral na GPU (GPGPU) tornaram-se uma boa oportunidade para se explorar um novo paradigma de paralelismo devido às altas performances paralelas e baixos custos [7]. Além disso, GPGPU oferece segurança aos dados dos pacientes, visto que esses dados permanecem em máquina local sem necessidade de serem trafegados numa rede.

O objetivo deste trabalho é quantificar o ganho de performance usando a GPU na recuperação de nódulos pulmonares. Como objetivo secundário, iremos avaliar a performance entre Atributos de Textura 3D, Nitidez de Borda 3D e as duas concatenadas.

Este artigo está organizado da seguinte forma: seção II descreve o estado da arte da CBIR, dos Atributos de Textura 3D, da Nitidez de Borda 3D, da GPGPU e do OpenCL. A seção III descreve como foram implementados os algoritmos, os casos de experimentos avaliados, os materiais utilizados e os resultados obtidos discursados. E a seção IV termina o trabalho com a conclusão.

II. ESTADO DA ARTE

A. Recuperação de Imagens Baseada em Conteúdo

A Recuperação de Imagens Baseada em Conteúdo (CBIR) é uma técnica que, permite ao especialista submeter uma imagem de consulta ou referência ao sistema e a partir desta colher imagens similares, que estão contidas num banco de dados de imagens, e esta semelhança é calculada a partir de características extraídas das imagens, como textura e forma [8].

Na figura 1 é ilustrado um exemplo de um sistema simples da CBIR. Na etapa de “Consulta da Imagem”, uma imagem é provida pelo usuário ao sistema. A etapa de “Extração das Características” da imagem consiste em determinar características particularizadas por algumas propriedades da imagem submetida pelo usuário, armazenando em seguida em um vetor as características da imagem submetida. A etapa de “Correspondência de Similaridade” compara o vetor de características da imagem inserida no sistema aos vetores de características das imagens já inseridas no banco de dados, que representa o componente da Análise de Similitude de Imagens (ISA), onde esta comparação é feita utilizando alguma função de medição de distância. Finalmente, um grupo de imagens semelhantes à imagem de consulta inserida no sistema é retornado ao especialista auxiliando seu diagnóstico na etapa de “Recuperação das Imagens”.

A medida de distância utilizada na etapa de Correspondência de Similaridade pode ser a Distância Manhattan (DM), onde as distâncias menores representam maior semelhança entre as imagens. A DM conhecida como “city block” utiliza a soma das diferenças absolutas entre os vetores de características, como mostra a equação 1, e esta simplicidade de cálculo resulta em um ótimo desempenho computacional [10].

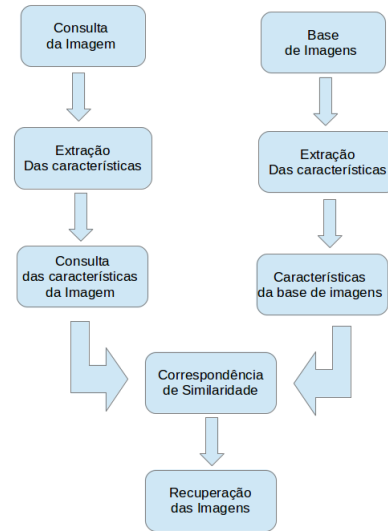


Figura 1. Arquitetura do sistema CBIR [9].

$$DM(x, y) = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \quad (1)$$

B. Atributos de Textura 3D

Atributos de textura ganharam um espaço de extrema importância na medicina por conseguirem destacar os detalhes finos presentes na estrutura de uma imagem, e assim reconhecer padrões de informação não visíveis aos especialistas [11] [12].

A estatística de segunda ordem proposta por Haralick em [13], utiliza matriz de co-ocorrência, que representa a probabilidade de ocorrência de um par de pixel com níveis de cinza i e j , em que i e j são os pixels vizinhos dada uma distância “ d ” entre os pixels e uma orientação “ θ ” entre os pixels nas dimensões x e y , respectivamente [14]. Depois, é calculada a função de probabilidade de pares de pixels entre fatias ao longo do eixo Z , em um volume de imagem. Para assim finalmente construir a Matriz de Co-ocorrência de Níveis de Cinza.

Em seguida, foram calculados as seguintes funções, propostas por Haralick, nas matrizes de co-ocorrência, para assim serem gerados os atributos de textura 3D: energia, entropia, momento da diferença inverso, matiz, contraste, proeminência, correlação, variância e homogeneidade.

Assim, 9 atributos podem ser calculados numa matriz de co-ocorrência, seguindo as 4 orientações angulares de θ (0° , 45° , 90° e 135° , no sentido anti-horário), gerando um vetor de atributos contendo 36 atributos para caracterizarem cada imagem.

C. Nitidez de Borda 3D

Um atributo de nitidez de borda é importante para separar lesões em questões de malignidade devido ao fato de que os nódulos malignos avançam na direção dos tecidos vizinhos [15].

Xu et al. definiram a nitidez de borda para nódulos pulmonares em duas características [16]. A primeira mede a diferença de intensidade entre os *pixels* do tecido do pulmão e do tecido do nódulo, enquanto que a segunda mede a transição de intensidade dos *pixels* sobre a borda do nódulo. Essas duas características são extraídas de uma função sigmóide aplicada em um vetor de intensidade composto pelos *pixels* presentes em linhas ortogonais traçadas de forma automática na borda do nódulo pulmonar.

Após a criação de linhas ortogonais na borda do nódulo, é extraído atributos estatísticos sobre as intensidades dos *pixels*. Estes atributos são 12: diferença entre os extremos, soma dos valores, soma dos quadrados, soma dos *logs*, média aritmética, média geométrica, variância da população, variância da amostra, desvio padrão, média de *kurtosis*, medida de *skewness* e segundo momento central. Gerando um vetor contendo 12 características para cada nódulo pulmonar.

Atributos estatísticos das intensidades dos *pixels* de linhas ortogonais sobre a borda dos nódulos nas imagens do volume apresentaram uma precisão média de 81% para os 10 primeiros casos recuperados [17].

D. Computação de Propósito Geral na GPU (GPGPU)

Unidades de Processamento Gráfico (GPUs) têm beneficiado a ciência e a alta performance em sistemas em geral com o paradigma de processamento em paralelo, viabilizando o uso da Computação de Propósito Geral na GPU (GPGPU) [18]. As GPUs são dispositivos paralelos que atuam em conjunto com as CPUs para assim gerarem a heterogeneidade entre as duas.

A GPU é organizada como um conjunto de multiprocessadores, onde cada multiprocessador contém um conjunto de processadores (núcleos ou *cores*), que executam o mesmo código através de centenas a milhares de *threads* concorrentemente por meio de uma memória compartilhada [19].

As linguagens de programação para GPUs mais conhecidas são CUDA (*Dispositivo de arquitetura computacional unificada*), da NVIDIA, e OpenCL (*Linguagem de Computação Aberta*). Enquanto CUDA pode ser utilizada exclusivamente em placas da NVIDIA, o OpenCL, que é *opensource*, pode ser utilizado em uma gama de dispositivos de computação que suportam paralelismo, como placas da NVIDIA e da AMD.

E. Linguagem de Computação Aberta (OpenCL)

Com o OpenCL é possível escrever um único programa que possa ser executado em uma ampla gama de sistemas, onde nenhum outro padrão de programação paralela consegue esse grande alcance. Esta é uma das razões pelas quais o OpenCL é tão importante e tem o potencial de transformar a indústria de softwares [20].

A habilidade principal do OpenCL é mapear eficientemente sistemas homogêneos e heterogêneos, por meio de dispositivos *single-core* e *multi-core*, arquiteturas e fabricantes diversos, em um único sistema de computação paralela.

A abstração efetuada pelo OpenCL pode ser compreendida pelos 4 seguintes passos: modelo de plataforma, modelo de

execução, modelo de memória e modelo de programação [20]. Em OpenCL, o modelo de execução baseia-se nos elementos chamados de itens de trabalho (*work-items*) e grupos de trabalho (*work-groups*), onde os *work-items* são agrupados em *work-groups*. Múltiplos *work-groups* podem ser alocados numa mesma Unidade Computacional (UC), porém estes estão dispostos em uma fila de execução. Um conjunto de UCs formam um dispositivo. *Work-items* de um mesmo *work-group* podem tanto se comunicarem como também se sincronizarem eficientemente, devido aos recursos compartilhados presentes nas UCs (Figura 2).

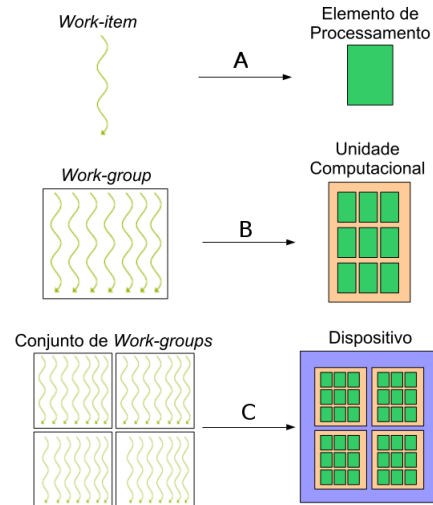


Figura 2. Modelo dos elementos do modelo de execução (à esquerda) ao do modelo de plataforma (à direita) do OpenCL. Em A, um *work-item* equivale a um Elemento de Processamento, assim como em B, um *work-group* é o mesmo que uma Unidade Computacional (UC), e em C um conjunto de *work-groups* equipara a um Dispositivo [21].

III. AVALIAÇÃO DE DESEMPENHO

A. Implementação

Primeiramente, todos os nódulos foram segmentados manualmente utilizando as marcações dos radiologistas do projeto LIDC [22]. Em seguida, um algoritmo da CBIR foi implementado possuindo 5 passos a fim de ser recuperar os nódulos pulmonares mais similares a partir de um nódulo de referência. No passo 1, é determinado um nódulo do próprio banco de imagens, para servir como caso de referência para a comparação com os outros nódulos do banco. No passo 2, foi gerado os vetores de características das imagens, criados a partir da extração dos atributos das imagens. No passo 3, é feita a Análise de Similaridade das Imagens (ISA), entre nódulo de referência e os nódulos do banco através de uma medida de distância, e a medida utilizada foi a Distância Manhattan. No passo 4 é feita a ordenação dos nódulos mais similares ao nódulo de referência. E por fim no passo 5, é retornado ao especialista as imagens mais similares a imagem submetida ao sistema da CBIR (Figura 3).

Para realizar a Análise de Textura 3D, após a segmentação de todos os nódulos da base, onde cada fatia do nódulo possui

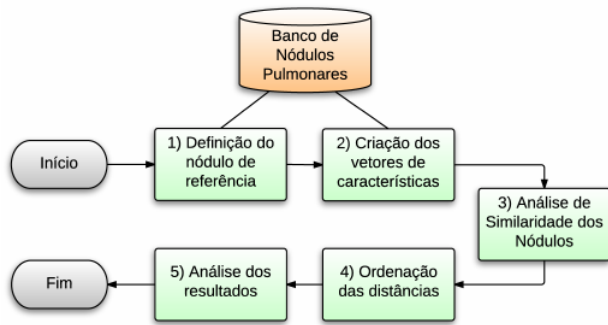


Figura 3. Fluxograma do algoritmo da CBIR implementado [23].

o número n de coordenadas cartesianas (x, y) que correspondem a marcação do nódulo. Em cada fatia do nódulo um polígono de n lados foi criado, sendo cada lado a união de duas coordenadas cartesianas. *Pixels* que não estavam inseridos no polígono foram marcados como preto. *Pixels* inseridos no polígono permaneceram com o tom de cinza da coordenada cartesiana na imagem original. Após a segmentação, Atributos de Textura foram extraídos a partir da matriz de co-ocorrência calculada sobre o volume de imagens. O cálculo da matriz de co-ocorrência utilizou nove atributos, quatro orientações angulares e distância entre os pixels igual a 1 (Seção II-B), permitindo a criação de um vetor de características para cada nódulo com 36 atributos. E por fim o vetor características de cada nódulo foi armazenado no banco de nódulos desenvolvido no MongoDB.

A Análise de Nitidez de Borda 3D foi realizada nas imagens originais de tomografia computadorizada, adotando a criação de linhas ortogonais na borda do nódulo, como proposto em [16], e extraíndo atributos estatísticos sobre as intensidades dos *pixels* das linhas ortogonais em todas as imagens do volume.

Na implementação da Análise de Nitidez de Borda 3D foram realizados os seguintes passos: vinte pontos de controle foram selecionados a partir do primeiro *pixel* marcado pelo radiologista, automaticamente na marcação da borda em cada fatia do nódulo. Se a borda contiver n *pixels*, então um ponto de controle é marcado a cada $n/20$ *pixels*. Depois, segmentos de reta foram desenhados em cada um dos 20 pontos de controle perpendicularmente a borda da lesão. Uma máscara foi criada para eliminar os segmentos de reta que cruzaram a parede do pulmão, e assim, incluir apenas *pixels* que pertenciam ao tecido do nódulo e do pulmão. A máscara foi gerada usando um algoritmo de segmentação por *threshold* com aplicação da operação morfológica matemática de dilatação, realizada na imagem de tomografia computadorizada. Após a exclusão dos segmentos de reta que não pertenciam ao pulmão, todos os *pixels* dos segmentos de reta contidos no volume do nódulo tiveram seus valores guardados num *array* de níveis de cinza, e este foi ordenado em ordem crescente. Em seguida, atributos estatísticos foram calculados sobre o *array* de *pixels* II-C. No final, os Atributos de Nitidez de Borda foram armazenados em um vetor de características

contendo 12 atributos para cada nódulo. E por fim, os vetores de característica de cada nódulo foram armazenados na base de nódulos do MongoDB.

Ainda foi criado um terceiro vetor contendo os Atributos Integrados (AI), que engloba os Atributos de textura 3D (AT) e os Atributos de Nitidez de Borda 3D (ANB) concatenados formando assim um vetor com 48 atributos. A junção dos AT e dos ANB foi aplicada pelo fato de que, descritores de imagens devem ser concatenados para oferecer uma melhor caracterização no processo de comparação [24].

O algoritmo da CBIR foi implementado para ser executado em duas arquiteturas diferentes: em uma CPU e numa GPU dedicada através do framework OpenCL. As mesmas imagens de referência e do banco de dados foram utilizadas nos três experimentos (seção III-C). O tempo de execução é a soma da transferência dos dados da memória RAM do hospedeiro a memória Global do dispositivo alvo, execução do *kernel* no dispositivo e a transferência de volta da memória Global do dispositivo para a memória RAM do hospedeiro.

B. Experimentos

A seguir são descritos os três casos de teste. Todos os casos foram testados em três maneiras de processamento: *single-core* é quando a execução é feita de modo sequencial na CPU, *multi-core* significa o uso de centenas de *cores* na CPU e *many-core* representa os *cores* na GPU. Nos três experimentos foram utilizados até 1171 nódulos para serem comparados, sendo x o número de nódulos, $x = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 1171$:

- Experimento 1: Foi verificado o desempenho da CBIR com o uso da abordagem da Análise dos Atributos de Textura 3D.
- Experimento 2: Foi verificado o desempenho da CBIR com o uso da abordagem da Análise de Nitidez de Borda 3D.
- Experimento 3: Foi verificado o desempenho da CBIR com a integração das duas abordagens dos experimentos 1 e 2, Análise dos Atributos de Textura 3D e Análise de Nitidez de Borda 3D.

A seguinte métrica foi usada nos experimentos, a fim de calcular a melhoria de performance da aplicação paralela sobre a implementação sequencial:

$$Speedup = \frac{T_1}{T_c}, \quad (2)$$

onde T_1 é o tempo de execução do melhor algoritmo sequencial e T_c é o tempo de execução do algoritmo paralelo usando c Unidades Computacionais (UC).

C. Materiais

O banco de nódulos desenvolvido em nosso laboratório possui 752 exames e 1.944 nódulos pulmonares nas 5 classificações de probabilidade de malignidade:

- Malignidade 1: probabilidade alta de ser benigno;
- Malignidade 2: probabilidade moderada de ser benigno;
- Malignidade 3: probabilidade indeterminada;

- Malignidade 4: probabilidade moderada de ser maligno;
- Malignidade 5: probabilidade alta de ser maligno.

Nódulos com probabilidade 3 foram descartados por apresentarem probabilidade indeterminada de malignidade. Assim neste trabalho, nódulos com probabilidade de serem malignidade 1 e 2 foram considerados benignos, e nódulos com probabilidade de malignidade 4 e 5 foram considerados malignos (Tabela III-C).

Probabilidade de Malignidade	Benignos		Malignos		TOTAL
	1	2	4	5	
Número de Nódulos	273	472	266	160	1.171
Soma no Número de Nódulos	745		426		

Tabela I
NÚMERO DE NÓDULOS UTILIZADOS [23].

Os dispositivos multiprocessadores utilizados nos experimentos foram uma GPU Nvidia GeForce GT 640 com 954 Mhz, 1 Gb de memória global e 384 núcleos de processamento, e uma CPU Intel Core i5-3350P com 3.10 GHz, 8 Gb de memória RAM e 4 núcleos de processamento. O ambiente de desenvolvimento para ambos os dispositivos foi um sistema operacional Linux Ubuntu v13.04 e linguagem de programação Java v1.8. A plataforma de desenvolvimento paralelo foi o OpenCL v1.2 para garantir a portabilidade de código-fonte entre as diferentes arquiteturas (Seção II-E), com *binding* para a linguagem Java, jOCL v0.1.9. A biblioteca gráfica utilizada nos algoritmos de extração de características dos nódulos pulmonares foi a OpenCV v2.4, com *binding* para Java, JavaCV v0.1.

D. Resultados e Discussão

Todos os tempos de processamento são a média de execução para cada número de nódulos pulmonares nos três experimentos. A figura 4 a seguir mostra os resultados do experimento 1 (Seção III-C).

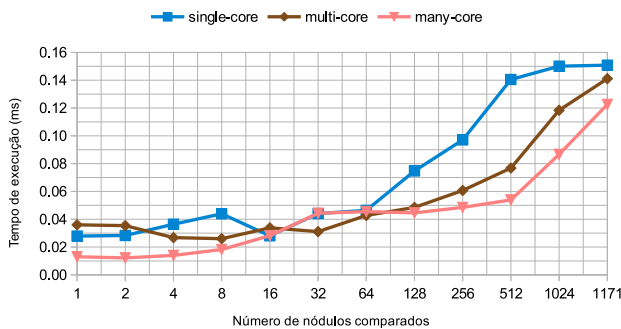


Figura 4. Avaliação de desempenho do cenário 1.

Os resultados mostram que em nenhum momento o tempo sequencial (*single-core*) foi melhor em comparação aos outros dois tempos. O tempo sequencial teve forte crescimento entre 64 e 512 nódulos. Entre 1 e 16, e 128 e 1171 nódulos o tempo paralelo na arquitetura *many-core* apresentou melhor

desempenho. O tempo de execução começou a crescer de forma mais visível no uso de *many-core* a partir de 512 nódulos. A arquitetura *multi-core* obteve melhor performance no uso de 32 e 64 nódulos, mas já em 128 nódulos começou a ter grande crescimento. Os speedups do tempo paralelo *multi-core* e *many-core* sobre o tempo sequencial foi de 1.06x e 1.23x, respectivamente neste cenário.

A precisão média dos nódulos pulmonares a partir dos atributos de textura 3D providas das funções de estatística de Haralick foi de 74% no trabalho de Oliveira e Ferreira [4].

A figura 5 a seguir mostra os resultados do experimento 2.

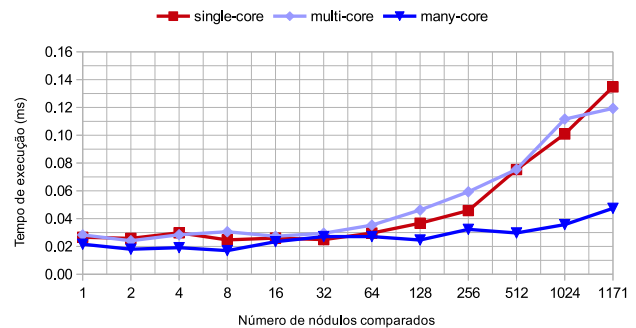


Figura 5. Avaliação de desempenho do cenário 2.

É possível observar que, o tempo de execução no modo *single-core* só começa a ser mais demorado nos casos de 2 e 4 nódulos, porém entre 1024 e 1171 nódulos já fica mais destacado que é o pior. Na arquitetura *multi-core* o tempo de processamento é às vezes pior do que na arquitetura *single-core*, como é nos casos de 8 e 1024 nódulos. Ao longo de todo o gráfico, o tempo de processamento na arquitetura *many-core* é melhor, com exceção apenas no caso de 32 nódulos, que perde um pouco em comparação ao processamento *single-core*. Os speedups do tempo paralelo *multi-core* e *many-core* sobre o tempo sequencial foi de 1.29x e 2.84x, respectivamente neste experimento.

A figura 6 seguinte mostra os resultados referentes ao experimento 3.

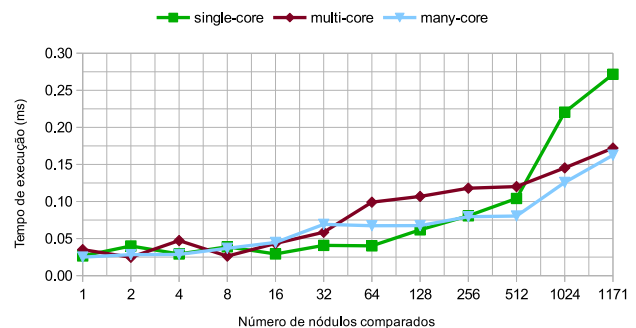


Figura 6. Avaliação de desempenho do cenário 3.

Na integração os Atributos de Textura 3D e de Nitidez de Borda 3D podemos ver que, o tempo de processamento na forma sequencial (*single-core*) é apenas melhor entre 16 e 128 núdulos, já entre 512 e 1024 núdulos começa a ficar pior gradativamente. No processamento *multi-core*, a vantagem é vista apenas para 8 núdulos, depois disso o tempo de execução só cresce, ficando pior do que a forma sequencial inclusiva entre 64 e 512 núdulos. No modo *many-core*, a execução é pior apenas e forma quase imperceptível entre 16 e 32 núdulos, a partir de 512 núdulos supera os tempos de execução de maneira *single-core* e *multi-core*. Os speedups do tempo paralelo *multi-core* e *many-core* sobre o tempo sequencial neste cenário foi de 1.57x e 1.67x, respectivamente.

IV. CONCLUSÃO

Este trabalho apresentou um comparativo de performance para análise de núdulos pulmonares tanto na CPU como também na GPU através do framework OpenCL para permitir a execução do código em diferentes máquinas, ou seja, a interoperabilidade de diferentes arquiteturas cada vez maiores. Também foi implementado um algoritmo do sistema da CBIR com as técnicas de Atributos de Textura 3D e de Nitidez de Borda 3D para a extração dos atributos, e a métrica utilizada foi a Manhattan. Os comparativos foram feitos em 3 tipos de arquitetura: *single-core*, *multi-core* e *many-core*.

Os resultados mostraram que o uso do processamento paralelo na GPU através do OpenCL é vantajoso a fim de diminuir o tempo de processamento da Análise de Similaridade das Imagens (ISA) dos núdulos pulmonares, apresentando speedup de até aproximadamente 3x. O uso da arquitetura *many-core* também se mostrou mais adequado do que a arquitetura *multi-core* na medida que o número de imagens foram aumentando, e teve situação em que o desempenho da arquitetura *single-core* até superou o tempo de execução da arquitetura *multi-core* como vimos nos três experimentos, chegando em até 0.59% de ganho de performance no experimento três. Os speedups foram relativamente pequenos devido ao fato do processamento da Distância Manhattan ser bastante rápido, caso fosse utilizado uma métrica mais robusta como a Mahalanobis ou utilizado um número de núdulos a serem comparados bem maior, o speedup seria mais expressivo [25].

A GPU também oferece segurança aos dados dos pacientes, já que os mesmos não precisam ser trafegados numa rede, como em *Grids* ou *Clusters* computacionais, correndo o risco de serem usados para fins que afetem a integridade do paciente, e são processados em máquina local.

Portanto, a GPU oferece ganho de performance e segurança a sistemas médicos CBIR.

REFERÊNCIAS

- [1] J. Ferlay, H.-R. Shin, F. Bray, D. Forman, C. Mathers, and D. M. Parkin, "Estimates of worldwide burden of cancer in 2008: Globocan 2008," *International Journal of Cancer*, 2010.
- [2] M. C. Oliveira, W. Cirne, and P. M. de Azevedo Marques, "Grades computacionais na otimização da recuperação de imagens médicas baseada em conteúdo," *Radio Bras*, vol. 40, no. 4, pp. 255–261, 2006.
- [3] J. R. Ferreira Junior, M. C. Oliveira, and A. L. Freitas, "Performance evaluation of medical image similarity analysis in a heterogeneous architecture," in *Proceedings of 27th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2014)*, 2014, pp. 159–164.
- [4] M. C. Oliveira and J. R. Ferreira, "A bag-of-tasks approach to speed up the lung nodules retrieval in the bigdata age," in *Healthcom'13*, 2013, pp. 632–636.
- [5] T. Deserno, S. Antani, and R. Long, "Ontology of gaps in content-based image retrieval," *Journal of Digital Imaging*, 2012.
- [6] M. Deserno, Thomas, "Challenges of medical image processing," *Computer Science-Research and Development*, vol. 26, no. 1-2, pp. 5–13, 2011.
- [7] F. Prochazka and M. C. Oliveira, "Aplicabilidade de gpus de baixo custo na otimização da análise de similaridade de imagens," *Future Generation Computer Systems*, 2012.
- [8] F. I. Alam and R. U. Faruqi, "Optimized calculations oh haralick texture for image classification," *IEEE Transactions on Systems on Systems, Man, and Cybernetics*, 2011.
- [9] H. Heirari, A. Chalechale, and A. A. Mohammadabadi, "Quality of dicom header information for image categorization," *Proceedings of the International Symposium on Medical Imaging*, 2002.
- [10] Animo-Blind, "Titulo da publicao animo," 2014.
- [11] C. B. Akgil, D. L. Rubin, S. Napel, C. F. Beaulieu, G. Hayit, and B. Acar, "Content-based image retrieval in radiology: current status and future directions," *Journal of Digital Images*, vol. 24, no. 2, pp. 208–222, 2011.
- [12] M. C. Oliveira, W. Cirne, and P. M. de Azevedo Marques, "Towards applying content-based image retrieval in the clinical routine," *Future Generation Computer Systems*, vol. 23, no. 3, pp. 466–474, 2007.
- [13] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 6, pp. 610–621, 1973.
- [14] J. M. P. da Costa, *Anlise de Imagem: Medio de rea Edificada*. Trabalho de Conclusão do Curso de Licenciatura em Engenharia Informática - Universidade da Beira Interior, 2010.
- [15] L. J. E. and M. A. L., "A margin sharpness measurement for the diagnosis of breast cancer from magnetic resonance imaging examinations," *Academic Radiology*, vol. 18, no. 12, pp. 1577–1581, 2011.
- [16] J. XU, S. Napel, H. Greenspan, F. Beaulieu, N. Agrawal, and D. Rubin, "Quantifying the margin sharpness of lesions on radiological images for content-based image retrieval," *Medical Physics*, vol. 39, pp. 5405–5418, 2012.
- [17] J. R. Ferreira Junior and M. C. Oliveira, "Evaluating margin sharpness analysis on similar pulmonary nodule retrieval," in *Proceedings of 28th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2014)*, 2015, pp. 1–6.
- [18] K. S. Lee, "Characterization and exploitation of GPU memory systems," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2012.
- [19] D. Stringhini, R. Goncalves, and A. Goldman, *Introdução à Computação Heterogênea*. XXXI Jornada de atualização em Informática (JAI), 2012.
- [20] A. Munshi, B. Gaster, T. Mattson, and J. Fung, *The OpenCL Programming Guide*. Pearson., 2012.
- [21] L. Lima, J. R. Ferreira Junior, and M. C. Oliveira, "Performance comparison of medical image similarity measures in a heterogeneous architecture," 2014.
- [22] G. Armato, Samuel, G. McLennan, L. Bidaut, and F. McNitt-Gray, Michal, "The lung image database consortium (lidc) and image database resource initiative (idri): A completed reference database of lung nodules on ct scans," *Medical physics*, vol. 38, no. 2, pp. 915–931, 2011.
- [23] J. R. F. Junior, *Auxílio Computadorizado ao Diagnóstico do Câncer de Pulmão Otimizado por GPU*. Dissertação de Mestrado - UFAL, 2015.
- [24] A. J. Traina, G. R. Balan, M. Bortolotti, and C. T. Jr, "Content-based image retrieval using approximate shape of objects," in *Proceedings of 17th IEEE International Symposium on Computer-Based Medical Systems (IEEE CBMS)*, 2004, pp. 91–96.
- [25] R. De Maesschalck, D. Jouan-Rimbaud, and L. Massart, D., "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, pp. 1–18, 2000.