

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CURSO DE CIÊNCIA DA COMPUTAÇÃO

THYAGO PEREIRA DA SILVA - 117110164
NATHAN FERNANDES PEDROZA - 118110342

**Uma revisão sistemática de algoritmos de sensoriamento remoto para
monitoramento ambiental - NDVI**

Trabalho para a disciplina de programação
concorrente apresentado ao Curso Bacharelado
em Ciência da Computação da Universidade
Federal de Campina Grande.

Orientador: Prof. Thiago Emmanuel Pereira

Campina Grande
2023

RESUMO

A crescente preocupação com questões ambientais, torna o sensoriamento remoto uma ferramenta indispensável para monitoramento e compreensão das mudanças ocorridas no ambiente. Este estudo visa aprofundar a análise crítica de algoritmos chave no sensoriamento remoto, com foco no algoritmo NDVI (Índice de vegetação por diferença normalizada) .

À medida que o mundo enfrenta desafios ambientais cada vez mais complexos, a eficácia dos algoritmos de sensoriamento remoto desempenha um papel crucial na coleta de dados precisos e interpretação das mudanças ambientais.

Este estudo se propõe a fornecer uma visão abrangente das pesquisas realizadas até o momento, identificando áreas em que o NDVI pode ser otimizado, com ênfase nas possibilidades oferecidas por técnicas de programação concorrente com GPUs.

SUMÁRIO

1. INTRODUÇÃO.....	4
2. FUNDAMENTAÇÃO TEÓRICA.....	5
3. REVISÃO DA LITERATURA.....	6
4. REFERÊNCIAS BIBLIOGRÁFICAS.....	14

1. INTRODUÇÃO

A preocupação global com questões ambientais e o monitoramento das mudanças no ecossistema terrestre nunca esteve tão presente quanto nos dias atuais.

O sensoriamento remoto é o processo de aquisição de informações sobre a superfície da terra sem a presença física no local de estudo. Ele envolve coleta de dados, como imagens e informações espectrais, usando sensores especializados. Estes dados são posteriormente processados e analisados para obter uma visão detalhada do ambiente, que vai desde a detecção de alterações na cobertura vegetal até o monitoramento de mudanças climáticas e impactos ambientais.

O NDVI (Índice de vegetação por diferença normalizada) é uma métrica que avalia a saúde e a densidade da vegetação de determinada área pela análise da diferença entre a luz visível que as plantas absorvem para a fotossíntese e a luz infravermelha próxima que elas refletem. Por meio desta diferença é gerado um índice que varia de -1 a +1.

Nesse sentido, o papel do NDVI é fornecer uma janela para compreender a saúde das plantas, a produtividade da vegetação e as mudanças ambientais. É uma ferramenta importante para monitorar ecossistemas, terras agrícolas e florestas. Entretanto, é importante destacar que o cálculo do NDVI não é desprovido de desafios, em particular, desafios de natureza computacional. A análise de grandes volumes de dados de sensoriamento remoto requer considerável poder de processamento e eficiência computacional.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. PROGRAMAÇÃO EM GPU

As GPUs, embora sejam originalmente projetadas para manipular gráficos e renderização 3D, demonstram grande capacidade computacional em cálculos de propósito geral, incluindo processamento de imagens, e análise de dados.

Uma das características mais marcantes de uma GPU é sua capacidade de processamento paralelo. Em vez de executar tarefas sequencialmente, GPUs são projetadas para processar muitos dados em paralelo. Para tomar vantagem dessa feature, é preciso que o problema seja dividido em pequenos subproblemas independentes. O que é particularmente benéfico no contexto de cálculo do NDVI, onde a análise de cada pixel de uma imagem, pode ser realizada separadamente.

2.2. CUDA E OPENCL

Cuda desenvolvida pela NVIDIA, e o OpenCL um padrão de código aberto suportado por várias empresas, são as principais plataformas de programação utilizando GPU.

2.2.1. CUDA

Segundo o artigo por Yang, Zhu [3] o CUDA (Compute Unified Device Architecture) é uma tecnologia desenvolvida pela NVIDIA que torna possível o desenvolvimento de aplicações capazes de utilizar as vantagens presentes na GPU por meio de um ambiente de desenvolvimento Clike que faz uso a linguagem de programação C e seu compilador.

2.2.2. OPENCL

De acordo com a especificação do OpenCL [4] o OpenCL (Open Computing Language) é um padrão aberto para uso de computação paralela, em CPUs, GPUs e outro processadores, fornecendo aos desenvolvedores acesso portátil e eficiente ao poder dessas plataformas heterogêneas.

3. REVISÃO DA LITERATURA

3.1. CÁLCULO DO ÍNDICE

A fórmula do NDVI é a seguinte:

$$NDVI = (\rho_{NIR} - \rho_R) / (\rho_{NIR} + \rho_R) [1]$$

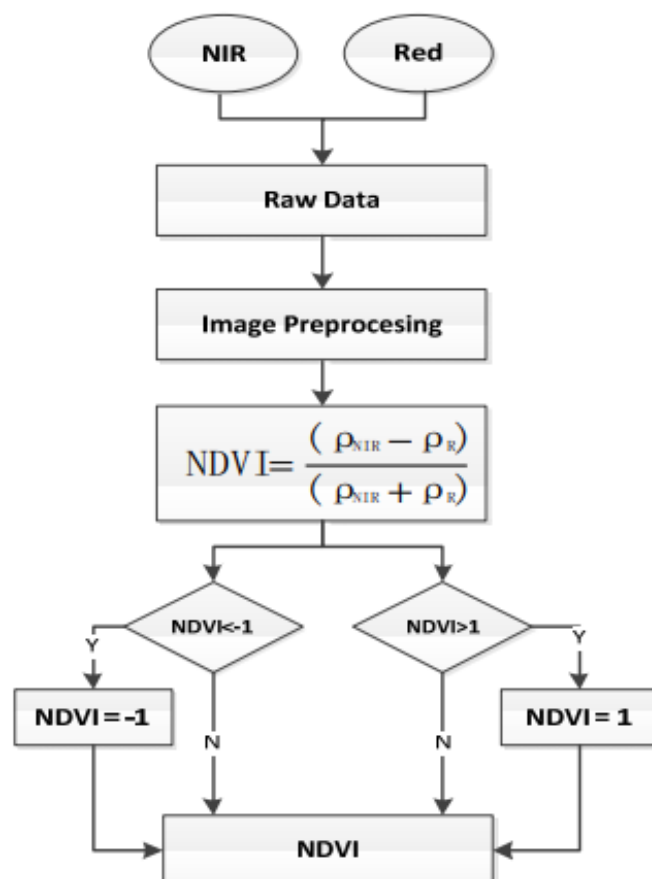
Onde ρ_{NIR} é a banda infravermelha próxima, e ρ_R é a banda de luz vermelha. Entretanto a abordagem computacional necessária, para chegar ao valor do índice de acordo como artigo de Juan e JianChao(2013) [1] segue os seguintes passos:

Passo 1 - Leitura dos valores observados para ρ_{NIR} e ρ_R .

Passo 2 - Uso da fórmula do NDVI para calcular o valor do índice.

Passo 3 - Interpretação do valor calculado para o NDVI.

Passo 4 - Geração de uma nova imagem, após o cálculo do índice.



Fonte: Juan & JianChao [1]

3.2. INTERPRETAÇÃO

Os valores do NDVI estão contidos no intervalo de -1 à +1 onde cada faixa de valor tem implicações específicas sobre a situação da cobertura vegetal na área analisada. De acordo com o departamento de agricultura dos estados unidos [2], o valor do NDVI pode ser interpretado da seguinte forma:

Valor do NDVI	Interpretação
$\text{NDVI} < 0.0$	Nuvens ou água.
$0.0 \leq \text{NDVI} \leq 0.1$	Solo desprovido de cobertura vegetal
$0.1 < \text{NDVI} < 0.6$	Vegetação esparsa
$0.6 \leq \text{NDVI}$	Vegetação densa

Tabela 1 - Interpretação do Índice.

3.3. APLICAÇÕES DO NDVI

3.3.1. AGRICULTURA DE PRECISÃO

O NDVI é utilizado pela agricultura moderna para monitorar o desenvolvimento das culturas, avaliar a necessidade de irrigação e identificar áreas com estresse hídrico ou doenças. De acordo com um estudo de caso [5] em uma plantação de batatas na Alemanha, o NDVI pode oferecer considerável benefício aos agricultores, na medida que permite a tomada de decisões relacionadas à irrigação, aplicação de fertilizantes e pesticidas, direcionadas ao índice.

3.3.2. MONITORAMENTO AMBIENTAL

Devido a correlação direta entre o NDVI e a saúde da

vegetação [6] existem inúmeras aplicações possíveis deste índice para fins ecológicos. O NDVI fornece informações sobre a distribuição espacial e temporal de comunidades vegetais, biomassa vegetal, fluxos de CO₂, qualidade da vegetação para herbívoros (porque a taxa de verdejamento pode se correlacionar a qualidade da comida) e a extensão da degradação da terra em diversos ecossistemas.

As aplicações do NDVI no monitoramento ambiental, não se limitam ao monitoramento da vegetação. Porque o NDVI é capaz de identificar corpos d'água e nuvens, o índice pode ser empregado eficazmente no monitoramento de rios e lagos [8].

3.3.3. GESTÃO DE DESASTRES

O NDVI faz parte de um leque de algoritmos que desempenham papel significativo no monitoramento de inundações e queimadas. Sua capacidade de distinguir entre superfícies aquáticas e terrestres é particularmente valiosa nesse contexto. Quando ocorrem inundações, as áreas alagadas refletem mais luz na banda do infravermelho próximo, resultando em valores de NDVI mais baixos. Este declínio nos valores de NDVI é um indicador da presença de inundações, permitindo a identificação de áreas afetadas e a tomada de decisões dirigidas.

Do mesmo modo, o NDVI é uma ferramenta eficaz no monitoramento de queimadas e incêndios florestais. As áreas queimadas exigem valores de NDVI significativamente mais baixos com relação a vegetação saudável. Durante um incêndio, a vegetação é danificada, resultando em uma diminuição acentuada nos valores de NDVI. Esse declínio nos valores de NDVI é um indicativo direto de áreas afetadas pelo fogo. A capacidade de monitorar essas mudanças rapidamente permite uma resposta mais eficaz no combate a incêndios e no gerenciamento de recursos.

3.4. IMPLEMENTAÇÕES PARALELAS DE NDVI EM GPU

A capacidade de processamento paralelo das GPUs oferece uma oportunidade de otimização do cálculo do NDVI e de gerenciamento eficaz de grandes volumes de dados provenientes de sensoriamento remoto. Uma abordagem que capitaliza essa tecnologia é discutida na literatura, como evidenciado pelo artigo intitulado “Parallel Processing of Satellite Images using CUDA Library: Focused on NDVI Calculation” [10], que evidencia como técnicas de processamento paralelo utilizando CUDA podem melhorar a velocidade de processamento do cálculo de NDVI. A implementação que leva em consideração os recursos de paralelismo oferecidos pela GPU é mostrada nas figuras 2, 3 e 4.

3.4.1. PRÉ-PROCESSAMENTO

Para calcular o NDVI é necessário utilizar os canais *Rband* e *Nband* de uma imagem. De modo a extrair essas informações de uma imagem bruta, o algoritmo proposto realiza a rotina de pré-processamento de forma sequencial, utilizando exclusivamente os recursos presentes na CPU. Através da função *TIFFReadRawStrip* da biblioteca *LIBtiff* os dados dos pixels da imagem são carregados e armazenados na variável chamada *data*, uma linha de cada vez. Em um laço subsequente, tirando proveito da ordenação dos canais (*Bband*, *Gband*, *Rband* e *Nband*), a extração e armazenamento dos dados de interesse para o cálculo do NDVI (*Rband* e *Nband*) é realizado nas respectivas variáveis.

```
//위성영상의 pixel data를 접근하는 방법
for(int count=0;count<imageheight/height;count++){
    uint32 col =0;
    for(int h=count*height;h<(count+1)*height;h++){
        strip = h;
        TIFFReadRawStrip(image,strip, buff, bc[strip]);
        data = (uint16*)buff;
        gettimeofday = clock();
        for(int w=0;w<width*4;w+=4){
            *(Rband + ((col*width)+w/4)) = data[w+1];
            *(Nband + ((col*width)+w/4)) = data[w+3];
        }
        col++;
    }
}
```

Figura 2 - Rotina de pré-processamento Fonte [10]

3.4.2. PROCESSAMENTO EM GPU

```
//GPU공간에 메모리 생성
uint16 *dev_G, *dev_N, *dev_ndwi;
//GPU공간에 메모리 크기 설정
cudaMalloc((void**)&dev_G,width*height*sizeof(uint16));
cudaMalloc((void**)&dev_N,width*height*sizeof(uint16));
cudaMalloc((void**)&dev_ndwi,width*height*sizeof(uint16));
//CPU 버퍼를 GPU버퍼로 COPY
cudaMemcpy(dev_G,
            Gband,
            width*height*sizeof(uint16),
            cudaMemcpyHostToDevice);
cudaMemcpy(dev_N,Nband,
            width*height*sizeof(uint16),
            cudaMemcpyHostToDevice);
//GPU상에서 NDWI연산 함수
add<<<grids,threads>>>>(dev_G,dev_N,dev_ndwi);
//GPU상에서 연산된 결과값을 CPU공간으로 COPY
cudaMemcpy(ndwi,
            dev_ndwi,width*height*sizeof(uint16),
            cudaMemcpyDeviceToHost);
```

Figura 3 - Controle e transferência GPU Fonte : [10]

```
//커널을 이용한 NDVI구현
__global__ void ndvi(int *g,int *n,int *ndvi){
    int tid = blockIdx.x * blockDim.x + threadIdx.x;

    while(tid<(width * height)){
        if(g[tid] == n[tid]){ndvi[tid] = -1;}
        else if(g[tid] < n[tid]) ndvi[tid] = 0;}
        else if(g[tid] > n[tid]) ndvi[tid] = 1;}

        tid += blockDim.x * gridDim.x;
    }
}
```

Figura 4 - Kernel de Cálculo do NDVI Fonte : [10]

O algoritmo baseado em GPU para cálculo do NDVI, requer alguns passos adicionais [Figura 3]. Após o pré-processamento, as informações Rband e Nband, estão armazenadas na memória da CPU, é preciso que esses dados sejam transferidos para a memória da placa gráfica. Por meio da função *cudaMalloc*, é alocado espaço de memória na GPU para 3 variáveis (*dev_G* , *dev_N* e *dev_NDVI*). Por meio da função *cudaMemory*, os dados correspondentes às bandas (*Rband* e *Nband*) são enviados para a memória da placa gráfica. As funções que serão executadas na GPU são

chamadas kernels, após a transferência dos dados da CPU para a GPU, o cálculo do NDVI é feito como mostrado no kernel [Figura 4], onde o algoritmo realiza aproximações do valor do NDVI baseado nos valores das bandas. O único aspecto adicional em comparação com um cálculo que ocorreu em uma CPU é o uso dos comandos dimensionais, para determinar o número total de threads, com o objetivo de garantir que o programa seja executado dentro do número de threads. Por fim, os resultados armazenados na variável *dev_ndvi* são transferidos da GPU de volta para a CPU.

3.4.3. PERFORMANCE CPU VS GPU

De acordo com os resultados encontrados em [10]. A análise comparativa dos tempos de processamento em CPU e GPU revela que o tempo de processamento em GPU superou o tempo de processamento em CPU. Esta vantagem em tempo de processamento, se torna cada vez mais evidente na medida que o volume de dados aumenta.

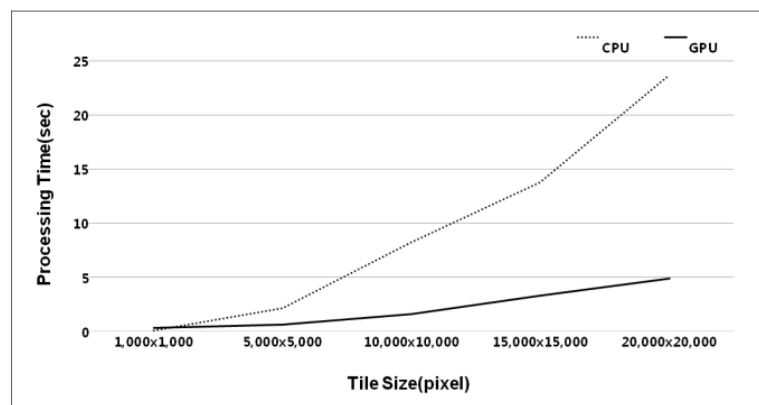


Figura 5 - Tempo de processamento CPU vs GPU Fonte: [10]

É interessante pontuar que o ganho na velocidade ocorre apenas quando o tamanho da imagem é suficientemente grande, para imagens pequenas, o tempo de processamento da CPU é superior. Isso ocorre porque, para pequenas quantidades de dados, não se justifica o custo de deslocamento dos dados da CPU para a GPU. Vários experimentos, presentes na literatura [1], [11], [12] chegam a valores diferentes para em qual tamanho de imagem (em pixels), passa a ser mais interessante utilizar a GPU.

3.4.4. PONTOS DE MELHORIA

Precisão	Pré-Processamento
O algoritmo faz aproximações grosseiras de NDVI de modo a simplificar o cálculo. Valores precisos interessam aos pesquisadores que fazem uso do índice.	A rotina de pré-processamento proposta não explora técnicas de programação concorrente, optando pela implementação puramente sequencial.

3.5. CONCLUSÕES

Neste trabalho, exploramos a importância do sensoriamento remoto e do Índice de Vegetação por Diferença Normalizada (NDVI) como ferramentas cruciais para monitorar o ambiente e compreender as mudanças que ocorrem nele. À medida que as preocupações ambientais se tornam cada vez mais prementes em nosso mundo, a necessidade de algoritmos eficazes de sensoriamento remoto, como o NDVI, se torna evidente. O NDVI desempenha um papel vital na avaliação da saúde da vegetação, monitoramento de ecossistemas, terras agrícolas e florestas, bem como na gestão de desastres, como inundações e incêndios florestais.

No entanto, também destacamos os desafios computacionais enfrentados na aplicação do NDVI em larga escala. O processamento de grandes volumes de dados de sensoriamento remoto via imagens de satélite requer recursos significativos de processamento e eficiência computacional. É nesse ponto que as Unidades de Processamento Gráfico (GPUs) e técnicas de programação concorrente se tornam valiosas. A capacidade de processamento paralelo das GPUs, quando utilizada adequadamente, pode otimizar significativamente o cálculo do NDVI e o gerenciamento de grandes conjuntos de dados.

Nossa revisão da literatura destacou implementações paralelas do NDVI em GPUs, evidenciando como essa abordagem pode melhorar a velocidade de processamento em comparação com a execução sequencial em CPUs. A análise comparativa demonstrou que, à medida que o tamanho do conjunto de dados aumenta, a GPU se torna mais vantajosa em termos de desempenho. No entanto, é importante observar que o ganho de desempenho é mais notável para conjuntos de dados consideravelmente grandes, uma vez que a transferência de dados entre a CPU e a GPU pode se tornar um gargalo em conjuntos de dados menores.

No entanto, há pontos de melhoria a serem considerados. Um deles é a busca por maior precisão nas aproximações usadas no cálculo do NDVI,

especialmente para pesquisadores que necessitam de valores mais precisos. Além disso, o pré-processamento dos dados, conforme proposto, ainda é realizado de forma sequencial, sem explorar completamente o potencial de programação concorrente.

Em resumo, este estudo ressalta a importância do NDVI e do sensoriamento remoto para monitoramento ambiental. A implementação paralela em GPUs demonstra um grande potencial para acelerar o cálculo do NDVI, especialmente em cenários de grande escala. À medida que a tecnologia avança, é crucial continuar aprimorando essas técnicas para atender às crescentes demandas por informações precisas e oportunas sobre nosso ambiente. A pesquisa futura pode se concentrar na otimização da precisão, na exploração de técnicas de programação concorrente e na adaptação contínua às mudanças tecnológicas.

4. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Juan, W., & Jianchao, S. (2013). A New Type of NDVI Algorithm Based on GPU Dividing Block Technology. 2013 International Conference on Computational and Information Sciences. doi:10.1109/iccis.2013.192
- [2] United States department of agriculture. Normalized Difference Vegetation Index (NDVI) . Disponível em:
[https://ipad.fas.usda.gov/cropexplorer/Definitions/spotveg.htm#:~:text=Normalized%20Difference%20Vegetation%20Index%20\(NDVI\)%3A&text=In%20general%2C%20NDVI%20values%20range,vegetation%20\(0.6%20and%20above\).](https://ipad.fas.usda.gov/cropexplorer/Definitions/spotveg.htm#:~:text=Normalized%20Difference%20Vegetation%20Index%20(NDVI)%3A&text=In%20general%2C%20NDVI%20values%20range,vegetation%20(0.6%20and%20above).)
- [3] Yang, Z., Zhu, Y., & Pu, Y. (2008). Parallel Image Processing Based on CUDA. 2008 International
- [4] Munshi, A. (2009). The OpenCL specification. 2009 IEEE Hot Chips 21 Symposium (HCS).
- [5] Heuvelink, G. B. M., & van Egmond, F. M. (2010). Space–Time Geostatistics for Precision Agriculture: A Case Study of NDVI Mapping
- [6] Pettorelli, N., Vik, J. O., Mysterud, A., Gaillard, J.-M., Tucker, C. J., & Stenseth, N. C. (2005). Using the satellite-derived NDVI to assess ecological responses to environmental change. *Trends in Ecology & Evolution*, 20(9), 503–510. doi:10.1016/j.tree.2005.05.011
- [7] Jiang, L., Liu, Y., Wu, S., & Yang, C. (2021). Analyzing ecological environment change and associated driving factors in China based on NDVI time series data. *Ecological Indicators*, 129, 107933. doi:10.1016/j.ecolind.2021.10793
- [8] Omute, P., Corner, R., & Awange, J. L. (2012). The use of NDVI and its Derivatives for Monitoring Lake Victoria's Water Level and Drought Conditions. *Water Resources Management*, 26(6), 1591–1613. doi:10.1007/s11269-011-9974-z
- [9] FERNANDEZ, A. (1997). Automatic mapping of surfaces affected by forest fires in Spain using AVHRR NDVI composite image data*1. *Remote Sensing of Environment*, 60(2), 153–162. doi:10.1016/s0034-4257(96)00178-2
- [10] Parallel Processing of Satellite Images using CUDA Library: Focused on NDVI Calculation *Journal of the Korean Association of Geographic Information Studies*, 19(3) 2016, pp. 29-42 Disponível em :
<http://dx.doi.org/10.11108/kagis.2016.19.3.029>

[11] Alvarez-Cedillo, J., Herrera-Lozada, J., & Rivera-Zarate, I. (2014). Implementation Strategy of NDVI Algorithm with Nvidia Thrust. Lecture Notes in Computer Science, 184–193. doi:10.1007/978-3-642-53842-1_16

[12] GUERROUJ, F. Z., LATIF, R., & SADDIK, A. (2020). Evaluation of NDVI and NDWI parameters in CPU-GPU Heterogeneous Platforms based CUDA. 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech). doi:10.1109/cloudtech49835.2020.