

Davi Siqueira Correa Galati - 21905069

Eric Vinício Rocha França - 21902089

Pedro Rafael Faria Ferreira - 21907590

## **Sistemas distribuídos aplicados à compressão e recuperação de imagens**

Este trabalho destina-se a cumprir requisito de  
avaliação da disciplina Arquitetura de Sistemas  
Distribuídos, do UnICEUB.

## **SUMÁRIO**

1. Contextualização
2. Introdução
3. Transformada Discreta
4. Sistemas Multiprocessados
5. Message Passing Interface (MPI)
6. Material e Métodos
7. Compressão Distribuída de Imagens
8. Resultado e Discussão
9. Bibliografia

## **RESUMO**

O processamento de imagens requer muita velocidade de processamento, com isso em mente, é interessante utilizar ferramentas como distribuição de processamento em vários nós ou computadores, para isso, são usados algoritmos distribuídos de compressão e expansão de imagens que utilizam a imagem transformada discreta de cosseno (DTC).

Os resultados mostram que a economia de tempo conseguida com os algoritmos paralelos, em relação aos seus equivalentes sequenciais, é função da resolução da imagem e da complexidade dos cálculos envolvidos, ou seja, quanto maior a eficiência quanto maior o tempo de processamento em relação ao tempo de comunicação entre os nós.

## **PALAVRAS-CHAVE**

Compressão e expansão de imagens, Discret Transform Cossine (DTC), Distribuição de processamento

## **ABSTRACT. Distributed systems applied to image compression and recovery.**

Image processing requires a lot of processing speed; with that in mind, it is useful to use tools such as distributing processing across multiple nodes or computers. For this, distributed image compression and expansion algorithms using discrete transformed image are used of cosine.

The results show that the time savings achieved with parallel algorithms, in relation to their sequential equivalents, is a function of the image resolution and the complexity of the calculations, that is, the greater the efficiency, the greater the processing time in relation to time. of communication between the nodes.

## **Key words**

Image Compression and Expansion, discrete transformed image are used of cosine (DTC), distributing processing

## CONTEXTUALIZAÇÃO

Este artigo feito por alunos do Uniceub (Centro Universitário de Brasília) de Ciência da Computação, tem como finalidade compreender e sintetizar as informações contidas no artigo “Sistemas distribuídos aplicados à compressão e recuperação de imagens” de Fabio Jorge Assad Gostaldon e Ailton Akira Shinoda, publicado em 2008. Onde podemos compreender melhor a forma como acontece o processamento digital de imagens utilizando softwares que estão baseados na distribuição de processamento.

## INTRODUÇÃO

Processamento digital de imagens é uma área que demanda grande capacidade de processamento. Sendo assim, torna-se interessante a implementação de softwares que estejam baseados na distribuição do processamento em vários nós ou máquinas da rede de computadores.

## TRANSFORMADA DISCRETA

A maioria das aplicações utilizam transformadas bidimensionais no processamento de imagens.

No artigo foi abordado compreensão de imagens, ao utilizar a transformada apenas alguns resultados são aproveitados e reescritos com menos bits, o que torna possível o armazenamento em menor espaço e a transmissão em menos tempo.

A amostra da imagem é submetida a transformada discreta de cosseno e alguns resultados são selecionados estrategicamente e cada um é reescrito com uma quantidade de bits proporcionando a compreensão de dados.

Figura 01: Fórmula da transformada discreta de cosseno

$$F[u, v] = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f[m, n] X_{m,n}(u, v)$$

Fonte:

Onde:

Figura 02: Elementos da fórmula discreta de cosseno

$$\begin{aligned} u, v &= \text{posições discretas } (0, 1, 2, \dots, N-1); \\ f[m, n] &= N \times N \text{ pixels da imagem } (0, 1, 2, \dots, N-1); \\ X_{m,n}(u, v) &= \cos\left[\frac{(2m+1)u\pi}{2N}\right] \cos\left[\frac{(2n+1)v\pi}{2N}\right]; \\ F[u, v] &= \text{resultado da DTC.} \end{aligned}$$

Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

E a DTC inversa é:

Figura 03: Elementos da fórmula discreta de cosseno

$$\hat{f}[m, n] = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} c[u]c[v]X_{m,n}(u, v)$$

Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

Onde:

Figura 04: Elementos da fórmula discreta de cosseno

$$\begin{aligned} m, n &= \text{índices resultantes dos pixels da imagem}; \\ \hat{f}[m, n] &= \text{resultado da DTC inversa.} \end{aligned}$$

Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

## SISTEMAS MULTIPROCESSADOS

Uma forma de aumentar a velocidade dos computadores é a utilização de sistemas multiprocessados, máquinas constituídas de muitas CPUs cada uma independente da outra, mas quando trabalhando coletivamente, possuem poder computacional muito grande, os sistemas multiprocessados classificam-se de três maneiras: multiprocessadores de memória compartilhada (supercomputadores) que se comunicam por meio dela, multicomputadores, constituídos por pares CPU-memória e não compartilham um espaço de memória (Clusters), comunicam-se por meio de troca de mensagens em uma rede dedicada de alta performance, baixa latência e banda passante alta e sistemas distribuídos constituídos de várias estações de trabalho completa com CPU, memória e disco rígido local, monitor, teclado e mouse, interconectadas por uma rede comercial-padrão e comunicam-se por meio de troca de mensagens..

Os custos desses sistema distribuídos também são reduzidos, pois pode ser utilizada a rede de computadores já disponível, com o avanço exponencial da Internet, os computadores ligados a ela também podem integrar um sistema distribuído.

### **Message Passing Interface (MPI)**

Consiste em uma biblioteca criada para padronizar a troca de dados e sincronização, auxiliando na comunicação entre processos de ambientes de memória não compartilhada através de uma coleção de rotinas. O MPI possui cerca de 125 funções para as linguagens de programação C, C++, Fortran 77/90.

O padrão MPI também é mencionado como Multiple Program Multiple Data (MPMD) já que os processos muitas vezes podem executar diferentes programas. A comunicação de dados além do balanceamento da carga são fundamentais em implementações paralelas.

Comunicação assíncrona e programação modular são suportados pelo MPI através de comunicadores que possibilitam a definição de módulos de encapsulamento de estruturas. Nesse artigo será abordada a LAM (Local Area Network) que consiste em uma implementação MPI criada para elaboração de sistemas em uma rede local.

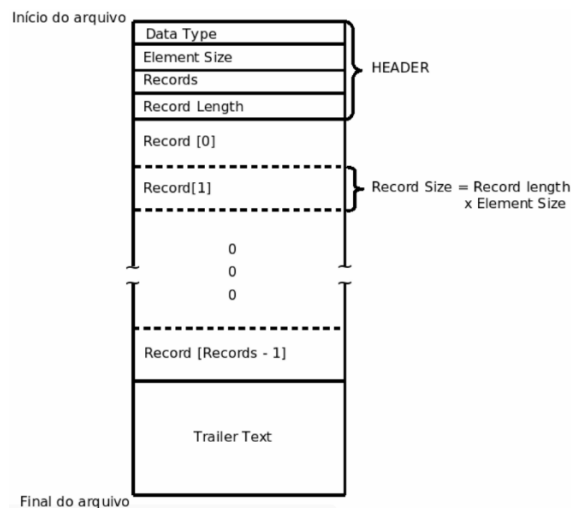
## **MATERIAL E MÉTODOS**

O Laboratório Linux de Processamento Paralelo localizado no Departamento de Engenharia Elétrica da Unesp/Ilha Solteira foi a sede para a elaboração do trabalho abordado.

Precedendo a etapa de implantação dos algoritmos foi definida a estrutura de arquivos para o armazenamento das imagens.

Este trabalho foi baseado em Embree e Danieli (1998) e tem a sua forma ilustrada na Figura 5.

Figura 5. Estrutura de um arquivo de dados do tipo DSP.



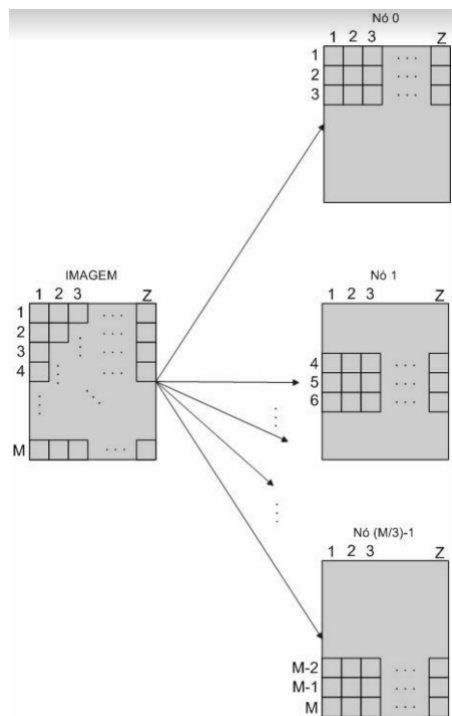
Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

Uma matriz é o que representa a imagem digital monocromática e a intensidade da luz e o nível de cinza de cada pixel é definido no valor de cada uma das posições

O algoritmo paralelo de compreensão implementado em C lê a imagem e realiza uma técnica de para diminuir o tempo de transformação de uma imagem chamada código de bloco. O código de bloco promove uma divisão da imagem em várias imagens menores. Nesse ponto ocorre o paralelismo de forma que o número de imagens divididas é separado em conjuntos e quando novamente agrupadas formam a imagem original, os nós recebem esses conjuntos.

Os blocos são distribuídos seguindo uma divisão de bloco de linha que é representada na figura 6.

Figura 06: Exemplificação da distribuição dos blocos entre nós.



Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

A partir de cada nó se aplica a DTC nas imagens divididas, definida pela equação da Figura 7.

Figura 07: Fórmula DTC

$$DTC\{A\} = \frac{CAC^T}{N^2}$$

Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

Onde a matriz A é a sub-imagem, N é a ordem da matriz A e C a matriz de cossenos constantes assim cada elemento é definido pela equação da Figura 8.

Figura 08: Equação de definição de elementos.



$$c[u][n] = \cos\left[\frac{(2n+1)u\pi}{2N}\right]$$

Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

Os resultados da DTC são reescritos com menos bits, e com isso há perda de informação, danificando a qualidade da imagem recuperada.

O resultado da DTC forma uma matriz 8x8 e é por meio da posição do coeficiente na matriz, como é apresentado na Figura 9, que é determinado a quantidade de bits que ele será reescrito.

Figura 09: Elementos da fórmula discreta de cosseno

8	8	4	4	4	4	4	4
8	8	4	4	4	4	0	0
4	4	4	4	0	0	0	0
4	4	4	4	0	0	0	0
4	4	0	0	0	0	0	0
4	4	0	0	0	0	0	0
4	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0

Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

O código de bloco faz com que ocorra mais degradação da qualidade da imagem. Quanto menor o bloco mais intensa a degradação entre os blocos.

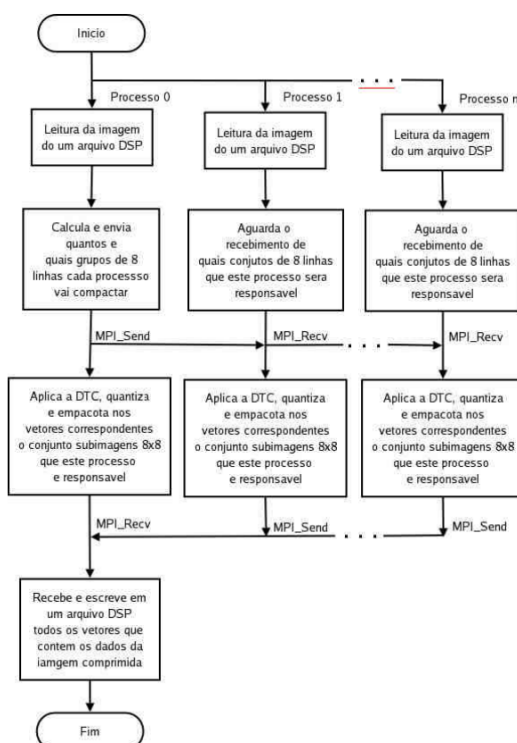
O tamanho ideal do bloco é determinado de acordo com os blocos de pixels adjacentes.

## COMPRESSÃO DISTRIBUÍDA DE IMAGENS

A figura 09 representa um fluxograma do algoritmo paralelo de compressão de imagens, nele é possível entender as etapas do processo em questão.

Inicialmente é feita a leitura do arquivo DPS, após a leitura o primeiro nó calcula quantos processos cada computador vai compactar e repassa a informação, enquanto isso os outros aguardam essa informação para iniciarem a compactação, logo depois é aplicada a DTC, quantiza e empacota os dados nos vetores correspondentes o conjunto de sub imagens que o processo é responsável e por fim um nó principal recebe as informações e escreve em um arquivo DSP todos os vetores todos os vetores da imagem definida.

Figura 10: Fluxograma do algoritmo paralelo de compressão de imagens



Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

A forma como os arquivos são expandidos é similar à forma como foram comprimidos, inicia-se com a leitura do arquivo DSP (arquivo C ++ 6 projecto Visual), em seguida, a quantidade de *record* é repartido entre os nós, que descompacta os coeficientes quantizados os copiando para uma matriz 8x8, onde é aplicada a DTC inversa, resultando em sub imagens, estas são enviadas ao nó principal onde são adequadamente agrupadas e lá formam uma nova imagem (com qualidade inferior).

Figura 11: Exemplo de Imagem após compressão

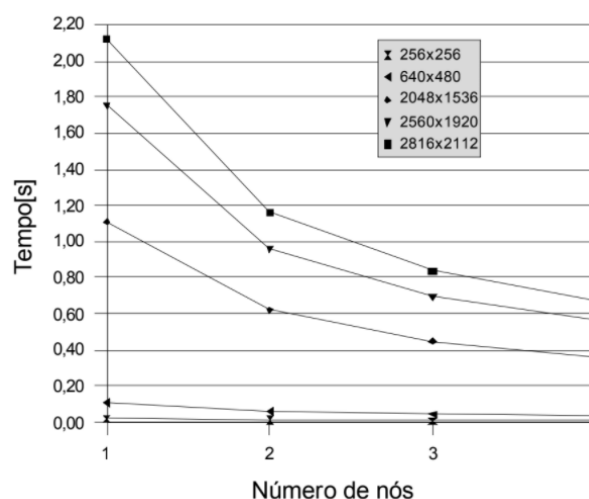


Fonte: ryukamitimassa, wordpress.

## RESULTADO E DISCUSSÃO

Pode-se perceber que o tempo de processamento das imagens aumenta com o aumento no número de pixels, porém, a grande quantidade de blocos (imagens de alta resolução) tendem a ser mais eficientes quando utilizados o algoritmo paralelo, a quantidade de nós também ajuda na redução do tempo de compressão das imagens.

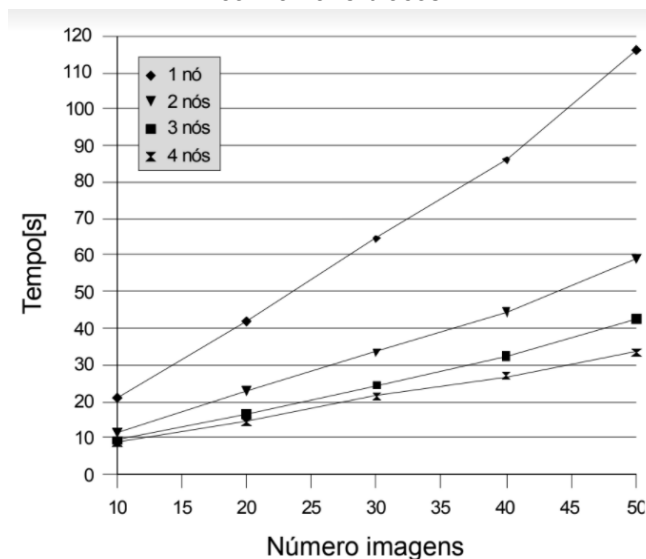
Figura 12: Tempo de compressão para imagens de diferentes resoluções.



Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

O uso do algoritmo paralelo é mais eficiente em quantidade maiores de fotos e de maior resolução.

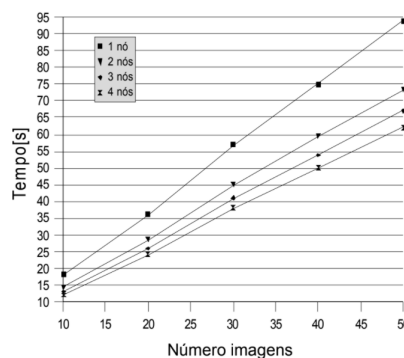
Figura 13: Tempos de execução para comprimir séries de imagens com 92.928 blocos.



Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

Os resultados obtidos na pesquisa do artigo, mostram que a técnica de compressão de imagens é mais vantajosa para manter qualidade do que a técnica de paralelização. Isso se dá pela maior simplicidade na quantidade de operações e complexidade dos cálculos realizados, fazendo com que o tempo de processamento dos dados consiga ser menor que o tempo gasto entre a comunicação entre os processos. Com isso, nota-se que o processamento paralelo é vantajoso em situações de processamento intenso.

Figura 14: Tempo de execução para expandir a série de imagens comprimidas anteriormente.



Fonte: Sistemas distribuídos aplicados à compressão e recuperação de imagens, 2008.

Vale ressaltar que os autores do artigo original, ao fazerem testes para verificar tempo e velocidade com que os dados são transferidos e as imagens compactadas utilizavam um composto de 50 imagens para que pudessem medir melhor, já que a velocidade de processamento de uma única imagem era muito rápida e difícil de medir com exatidão.

## **CONCLUSÃO**

A utilização das técnicas de paralelização dos algoritmos de compressão e expansão não diminui sua eficiência com relação à qualidade da imagem resultante, isso devido à paralelização estar baseada no código de bloco já bem fundamentado teoricamente.

A economia de tempo foi significativa, chegando até a 3,5 vezes com quatro computadores para se comprimir 50 imagens de 92.928 blocos. Isso é resultado da alta quantidade de cálculos complexos a serem realizados em cada parcela da imagem que os processos efetuam.

A complexidade dos cálculos não se mostrou tão alta e, por isso, o ganho de tempo com o processamento distribuído aplicado na expansão não foi muito significativo.

## BIBLIOGRAFIA

FILHO, Antônio; SILVA, Luis. ANÁLISE DA COMPRESSÃO JPEG NA GENERALIZAÇÃO DE MODELOS DIGITAIS. **Seção de Engenharia Cartográfica**, Rio de Janeiro, 2013.

Compressão de imagens. Diário a Bordo de Oficina de Multimídia\_B. Disponível em: <<https://ryukamitimasa.wordpress.com/2012/11/16/compressao-de-imagens/>>. Acesso em: 17 Aug. 2021.

GOSTALDON, Fabio Jorge Assad; SHINODA, Ailton Akira. Sistemas distribuídos aplicados à compressão e recuperação de imagens, [S. l.], p. 1-7, 6 jul. 2007. Disponível em: <<https://drive.google.com/file/d/1jPcTQMvQnuRbIIUiox-i1oG-HmCJteC5/view>>. Acesso em: 18 ago. 2021.