

## BIBLIOTECA CIENTÍFICA DE PROCESSAMENTO DE IMAGENS DISTRIBUÍDA.

Fabio Jorge Assad Gostaldon, Ailton Akira Shinoda – Engenharia Elétrica – Departamento de Engenharia Elétrica – Faculdade de Engenharia – Campus de Ilha Solteira

Muito importante no processamento digital de imagens, a manipulação de vetores e matrizes é largamente utilizada.

A grande quantidade de dados a serem processados nessas manipulações aumenta o tempo de execução quando realizada de forma sequencial. Pensando nisso, a criação de programas livres que trabalhem de forma distribuída, aplicáveis ao processamento digital de imagem, se torna interessante.

Um exemplo de manipulação de vetores é a correlação, que é freqüentemente utilizada em processamento digital de sinal. A função correlação é maximizada quando dois sinais são similares no domínio da freqüência. A correlação discreta é um simples produto ponto a ponto entre dois vetores definida como segue:

$$R[d] = \sum_{i=0}^{N-1} x[i]y[i+d] \quad (1)$$

Então um programa seguindo a filosofia de software livre e que utiliza as funções da biblioteca MPI, chamado corP, foi criado para executar a correlação entre dois sinais. A Figura 1 ilustra o funcionamento geral do programa.

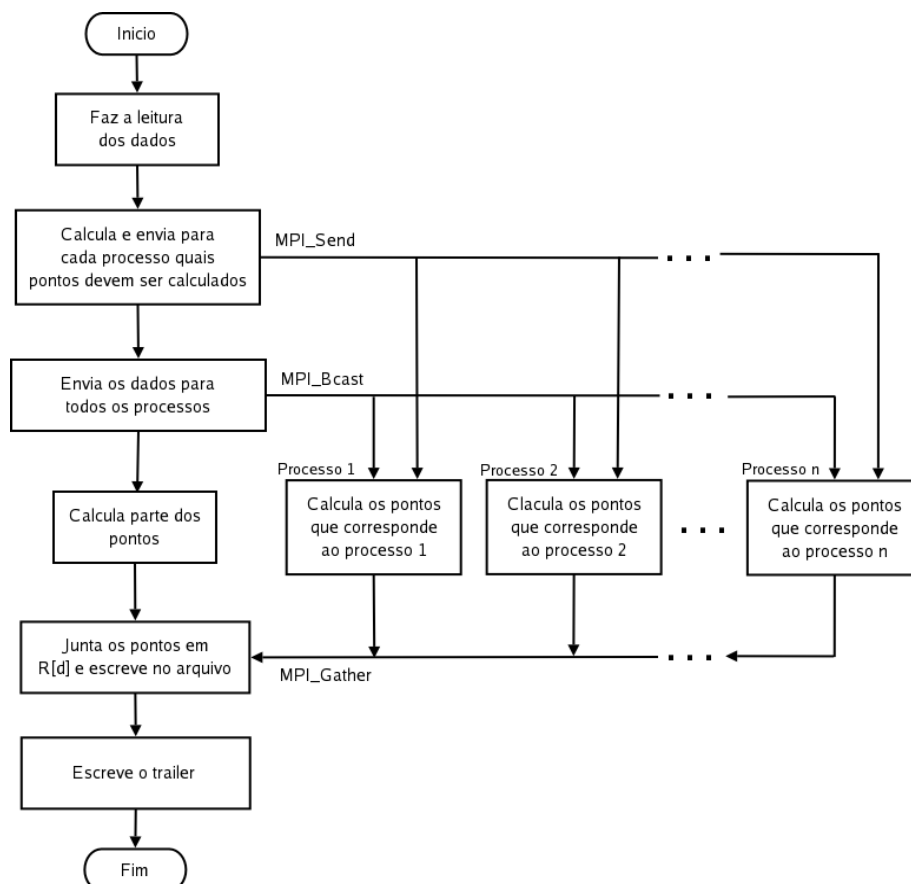


Figura 1: Fluxograma do programa corP

## BIBLIOTECA CIENTÍFICA DE PROCESSAMENTO DE IMAGENS DISTRIBUÍDA.

Fabio Jorge Assad Gostaldon, Ailton Akira Shinoda – Engenharia Elétrica – Departamento de Engenharia Elétrica – Faculdade de Engenharia – Campus de Ilha Solteira

O programa realiza a correlação entre a primeira gravação de um arquivo DSP\_FILE com a primeira gravação de outro arquivo DSP\_FILE. O programa abre os arquivos de dados DSP e faz a leitura da primeira gravação, esses dados são então enviados a todos os processos. Em seguida é enviado para cada processo quais pontos de  $R[d]$  cada um deve calcular, para isso foi implementada a rotina abaixo, que está no do código programa *corP*.

```
f = 0;
dest = 1;
w = 0;
while (dest < p) {
    d1 = min_lag + (size + rest - 1 + dest + w);
    d2 = min_lag + (2 * size + rest - 1 + dest + w);
    MPI_Send(&d1, 1, MPI_INT, dest, 0, MPI_COMM_WORLD);
    MPI_Send(&d2, 1, MPI_INT, dest, 0, MPI_COMM_WORLD);
    if (size != 1) {
        w = (size - 1) + f;
        f = f + (size - 1);
    }
    dest++;
}
```

Os arquivos DSP\_FILE tem estrutura definida conforme a Figura 2. Essa estrutura auxilia a manipulação dos dados.

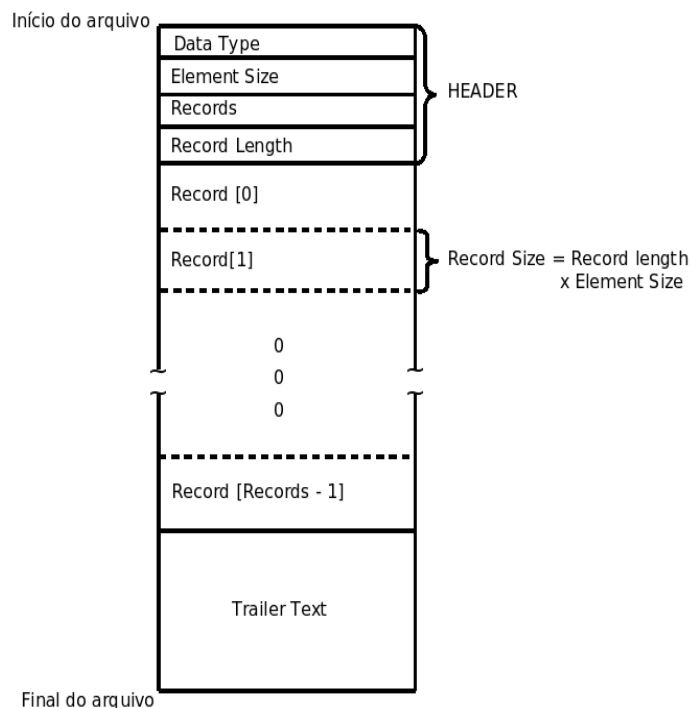
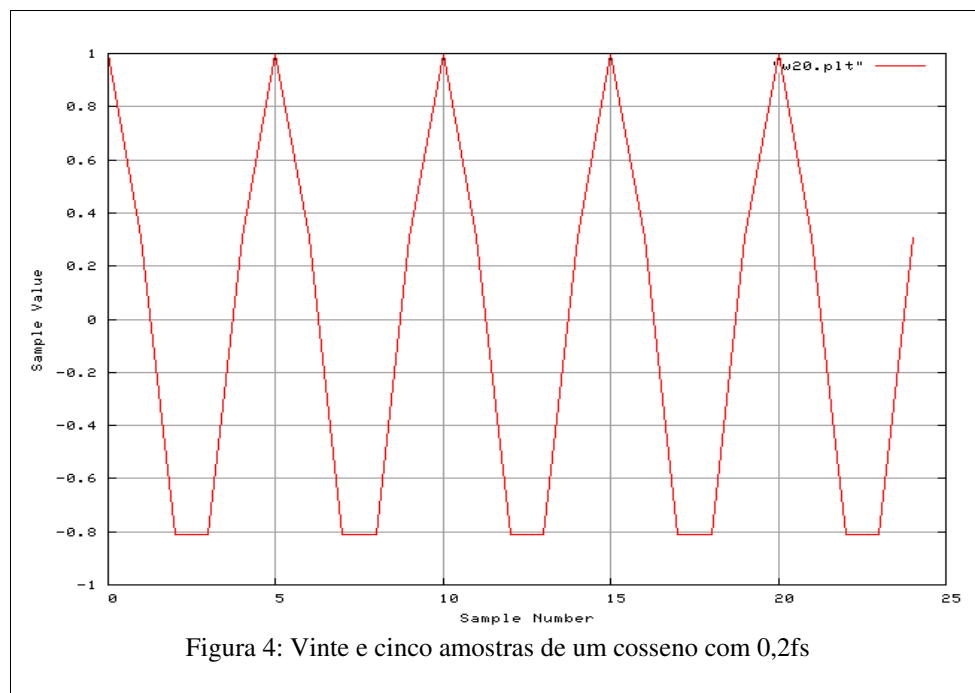
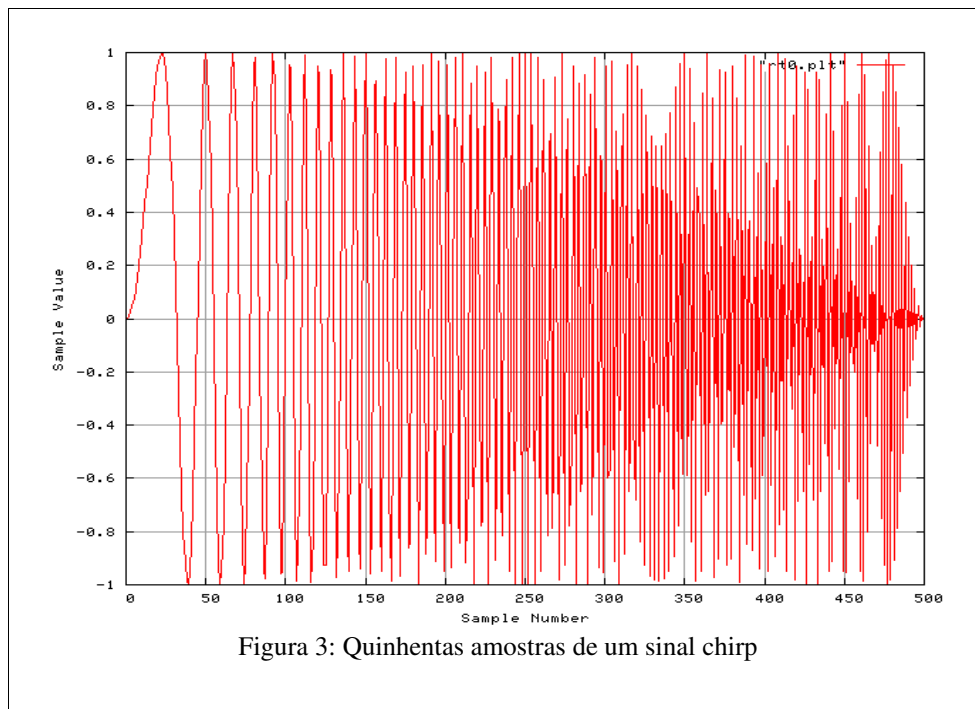


Figura 2: Estrutura de um arquivo de dados do tipo DSP

## BIBLIOTECA CIENTÍFICA DE PROCESSAMENTO DE IMAGENS DISTRIBUÍDA.

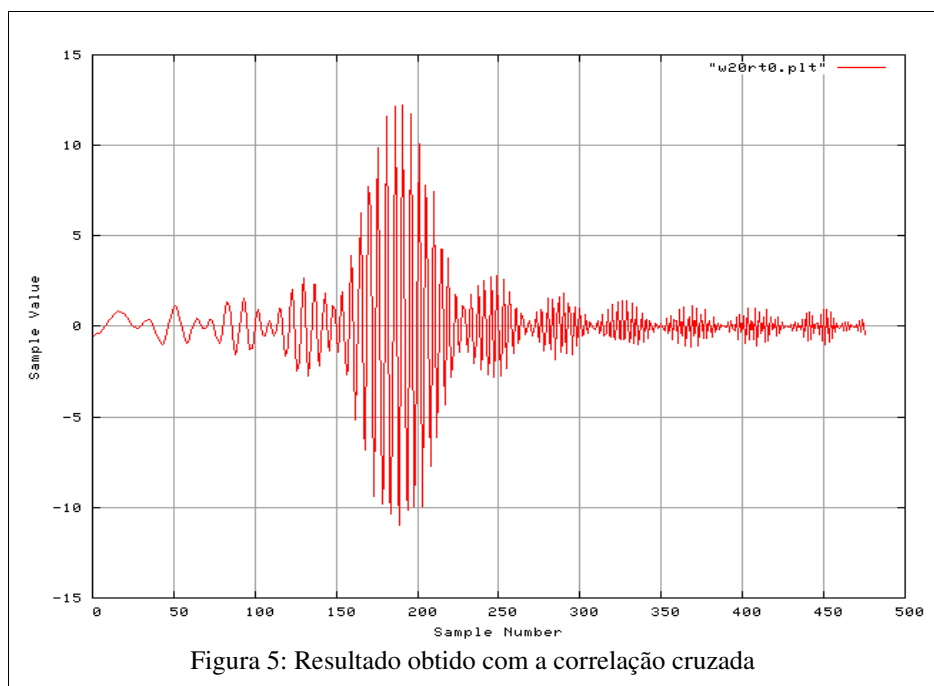
Fabio Jorge Assad Gostaldon, Ailton Akira Shinoda – Engenharia Elétrica – Departamento de Engenharia Elétrica – Faculdade de Engenharia – Campus de Ilha Solteira

Aplicando o programa em um sinal chirp – ilustrado na Figura 3 – e em uma sequência de 25 amostras de um cosseno com 20% da frequência de amostragem (Figura 4), obteve-se a saída da correlação cruzada mostrada na Figura 5.



## BIBLIOTECA CIENTÍFICA DE PROCESSAMENTO DE IMAGENS DISTRIBUÍDA.

Fabio Jorge Assad Gostaldon, Ailton Akira Shinoda – Engenharia Elétrica – Departamento de Engenharia Elétrica – Faculdade de Engenharia – Campus de Ilha Solteira



A execução dos programas distribuídos foi realizada no Laboratório Linux de Processamento Paralelo (LLPP), Departamento de Engenharia Elétrica, UNESP/FEIS. No trabalho em questão, a configuração do cluster de computadores foi de 4 computadores AMD 2,6 GHz, 512 Mb de memória RAM interligados com um switch da 3com operando a 100 Mb/s e com o sistema operacional linux Fedora Core 4. A versão empregada do LAM/MPI no cluster foi o 7.1.1.

A utilização da biblioteca MPI viabiliza o processamento paralelo, dando aos programas grande rendimento na manipulação da enorme quantidade de dados que envolve o processamento digital de imagem.

### Referências Bibliográficas

- [1] EMBREE, P.M. e DANIELI, D., *Algorithms for Digital Signal Processing*, 2nd ed., Hall PTR, New Jersey, 1998.
- [2] OPPENHEIM, A. V. e SCHAFER, S. W., *Discrete-Time Signal Processing*, 2nd ed, Prentice Hall, New Jersey, 1998.
- [3] PACHECO, P., *Parallel Programming with MPI*, Morgan Kaufmann, 1996.

**Bolsa:** FAPESP