

Departamento da Área de Informática

**Curso:** Bacharelado em Engenharia da Computação **Semestre:** 9

**Curso:** Bacharelado em Engenharia de Controle e Automação **Semestre:** Optativa

**Disciplina:** Processamento Digital de Imagens.

**Professor:** Esp. Giuliano Robledo Zucoloto Moreira.

Cuiabá-MT, 28 de julho de 2022.

## **NOTA EXPLICATIVA: Transformada discreta de Fourier**

Esta nota trata da operação de processamento digital de imagens no domínio da frequência por meio da aplicação da Transformada Discreta de Fourier (*DFT - Discrete Fourier Transform*). A ênfase é na aplicação, para maiores detalhes da *DFT* é necessária a consulta a nota específica sobre o assunto. A nota visa promover o aprendizado do desenvolvimento de algoritmos relacionados à *DFT* como também utilizar funções providas pela linguagem de programação. Ao final é apresentado um exercício que consiste em comparar o desempenho do processamento de uma imagem utilizando as funções desenvolvidas pelo(a) estudante e as funções da linguagem e explicar a diferença com base na teoria apresentada na nota específica da *DFT*.

# **1 Transformada Discreta de Fourier**

Está fora do escopo desta nota esgotar o assunto da *DFT*, o objetivo desta seção é rememorar as bases da *DFT* necessárias à sua aplicação no processamento digital de imagens e também . A ênfase está no desenvolvimento de algoritmos básicos: um para calcular a *DFT* de uma imagem digital, um para realizar a convolução entre uma imagem e um filtro e uma para calcular a *DFT* inversa da imagem.

## **1.1 Transformada Discreta de Fourier em duas dimensões (*DFT*-2D)**

Rememoremos a definição da *DFT*-2D que é expressa pela equação 1:

$$Y[k, l] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) e^{-j \frac{2\pi}{M} km} e^{-j \frac{2\pi}{N} ln}; \quad (1)$$

Onde:

$Y[k, l] \rightarrow$  é a  $k$ -ésima amostra no domínio da frequência;

$x(n, m) \rightarrow$  é a  $n$ -ésima amostra no domínio do tempo;

$j \rightarrow$  indicador de número complexo;

$M \rightarrow$  dimensão vertical do número de amostras por intervalo de espaço;  
 $N \rightarrow$  dimensão horizontal do número de amostras por intervalo de espaço;  
 $k \rightarrow$  índice da coordenada vertical da amostra no domínio da frequência e  $k = \{k \in \mathbb{Z} \mid 0 \leq k \leq M - 1\}$ ;  
 $l \rightarrow$  índice coordenada horizontal da amostra no domínio da frequência e  $l = \{l \in \mathbb{Z} \mid 0 \leq l \leq N - 1\}$ ;  
 $n \rightarrow$  índice da amostra no domínio do tempo.

Na forma apresentada a equação 1 flexibiliza o processamento de imagens com dimensões de largura e altura diferentes.

É possível compactar a apresentação da equação 1 agrupando os termos que multiplicam o sinal espacial de entrada  $x(m, n)$  em um único termo, conforme a equação 2:

$$W_{MN} = \frac{1}{\sqrt{MN}} e^{(-j\frac{2\pi}{M}km)} e^{(-j\frac{2\pi}{N}ln)}; \quad (2)$$

O agrupamento apresentado na equação 2 é muito útil na aceleração do processamento pois é dependente apenas das dimensões da imagem, o que é muito útil na construção de um conjunto de arquivos  $W_{MN}$  previamente calculados para diversas dimensões que possa ser acessado no processamento para reduzir cálculos.

Aplicando-se o termo  $W_{MN}$  da equação 2 à equação 1 tem-se uma forma de apresentação compactada da  $DFT$ -2D, forma apresentada na equação 3:

$$y[k, l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) e^{(W_{MN})}; \quad (3)$$

## 1.2 Algoritmos da $DFT$ -2D

Esta subseção trata o desenvolvimento dos algoritmos para as equações 1, 2 e 3 apresentadas na subseção 1.1, os quais são nomeados respectivamente como `dft2d`, `wmn` e `dft2dwmn`.

Lembre-se que os códigos-fonte dos algoritmos são redigidos em linguagem *m* e considera-se os desenvolvimentos no software *GNU Octave* versão 6.1.

Os códigos-fonte que serão apresentados nesta seção tem estritamente o papel de demonstrar uma forma de implementação das equações matemáticas, sendo desprezadas diversas verificações como se o arquivo selecionado existe, se ele realmente é uma imagem, se a imagem tem apenas um canal entre outras.

### 1.2.1 Algoritmo `dft2d`

O algoritmo trata da importação direta dos dados de uma imagem e da implementação da equação. Segue o código-fonte do algoritmo `dft2d` que implementa a equação 1:

## Código-fonte do algoritmo dft2d

```

[arq, cam] = uigetfile();
x = imread(cam, arq);
[M, N] = size(x);
y=zeros(M,N);
for (k=0:1:M-1)
    for (l=0:1:N-1)
        for (m=0:1:M-1)
            for (n=0:1:N-1)
                y(k+1,l+1) += x(m+1,n+1)*
                    exp(complex(0, ((2*pi)/M)*k*m))*
                    exp(complex(0, ((2*pi)/N)*l*n));
            endfor
        endfor
    endfor
endfor

```

No código-fonte do **algoritmo dft2d**, linha de comando 9 (nove), foi necessário alterar o modelo original da equação devido aos limites dos índices da imagem na linguagem de programação *m*, onde o índice do primeiro píxel da imagem não é 0 (zero), mas sim 1 (um) e a apresentação dos termos complexos foi modelada através da função adequada da linguagem.

### 1.2.2 Algoritmo wmn

Segue o código-fonte do algoritmo wmn que implementa a equação 2:

## Código-fonte do algoritmo wmn

```

[arq, cam] = uigetfile();
x = imread(cam, arq);
[M, N] = size(x);
mn = (M*N)^2;
wmn=zeros(M,N);
for (k=0:1:M-1)
    for (l=0:1:N-1)
        for (m=0:1:M-1)
            for (n=0:1:N-1)
                wmn(k+1,l+1)=exp(complex(0, -((2*pi)/M)*k*m))*
                    exp(complex(0, -((2*pi)/N)*l*n));
            endfor
        endfor
    endfor
endfor
wmn = (1/sqrt(mn))*wmn;
save arquivowmn.txt wmn;

```

Como no mesmo caso do algoritmo anterior, **algoritmo dft2d**, foi necessário alterar o modelo original da equação 2 no **algoritmo wmn** devido aos limites dos índices da imagem na linguagem de programação *m* e da função para processamento números complexos.

Para fins de simplificação computacional, sem prejuízos dada a propriedade da linearidade, optou-se por realizar a multiplicação dos termos por  $w_{mn}$  fora dos laços de repetição do algoritmo (penúltima linha do algoritmo).

### 1.2.3 idft2d

Segue o código-fonte do algoritmo  $w_{mn}$  que implementa a equação 2:

#### Código-fonte do algoritmo idft2d

```
[arq, cam] = uigetfile();  
x = imread(cam, arq);  
[M, N] = size(x);  
mn = M*N;  
wmn=zeros(M,N);  
for (k=0:1:M-1)  
    for (l=0:1:N-1)  
        for (m=0:1:M-1)  
            for (n=0:1:N-1)  
                wmn(k, l) = exp(-i((2*pi)/M)*k*m)*  
                    exp(-i((2*pi)/N)*l*n);  
            endfor  
        endfor  
    endfor  
endfor
```

## Referências