



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA

**RELATÓRIO DA ATIVIDADE 2 – INTRODUÇÃO AO
PROCESSAMENTO DIGITAL DE IMAGENS**

GABRIEL ALCÂNTARA
GIOVANNI BRUNO
HELTER YORDAN A. DA COSTA
SABRINA SILVA
WALLISSON DANTAS DA SILVA

JOÃO PESSOA, 2020.

GABRIEL ALCÂNTARA
GIOVANNI BRUNO
HELTER YORDAN A. DA COSTA
SABRINA SILVA
WALLISSON DANTAS DA SILVA

ATIVIDADE 2 - SISTEMA DE MANIPULAÇÃO DE IMAGENS

Relatório solicitado pelo professor
Leonardo Vidal Batista, da disciplina de
Processamento Digital de Imagens - PDI,
do curso de Engenharia de Computação.

JOÃO PESSOA, 2020.

RESUMO

O trabalho 02 tem por finalidade desenvolver um programa onde dada uma imagem I em níveis de cinza, de dimensões $R \times C$, realize tais demandas:

1. Rotacionar I por um ângulo especificado (parâmetro entre 0 e 360 graus), utilizando:

(a) mapeamento direto;

(b) mapeamento reverso com interpolação bilinear.

A imagem rotacionada deve preservar todo o conteúdo da imagem original.

2. Exibir o módulo da DCT de I , sem o nível DC, e o valor do nível DC

3. Encontrar e exibir uma aproximação de I obtida preservando o coeficiente DC e os n coeficientes AC mais importantes de I , e zerando os demais. O parâmetro n é um inteiro no intervalo $[0, R \times C - 1]$.

4. Encontrar a imagem resultante da filtragem de I por um filtro passa-baixas ideal quadrado, com frequência de corte f_c (parâmetro especificado pelo usuário) igual à aresta do quadrado, em pixels.

A rotação e a DCT devem ser desenvolvidas utilizando as equações estudadas em sala de aula, sem o uso de bibliotecas prontas para esse fim.

SUMÁRIO

RESUMO	2
SUMÁRIO	3
INTRODUÇÃO	4
DESENVOLVIMENTO	5
MATERIAIS E MÉTODOS	5
ROTAÇÃO	5
ROTAÇÃO POR MAPEAMENTO DIRETO	5
ROTAÇÃO POR MAPEAMENTO INVERSO COM INTERPOLAÇÃO BILINEAR	7
MÓDULO DCT	9
COMPRESSOR BASEADO EM COSSENOS	10
FILTRO PASSA-BAIXAS	11
CONCLUSÃO	11

INTRODUÇÃO

Nosso trabalho tem como finalidade o desenvolvimento de filtros e transformações em imagens, utilizando dos métodos e equações abordados em sala no decorrer das aulas. Iniciamos com o desenvolvimento do método de rotação, realizado por mapeamento direto e por mapeamento inverso. Realizamos a exibição do módulo DCT para imagem trabalhada com seus detalhamentos solicitados, o desenvolvimento da transformação para aproximação com preservação de coeficientes e do filtro passa-baixas com frequência de corte semelhante a aresta do quadrado.

DESENVOLVIMENTO

MATERIAIS E MÉTODOS

O trabalho de implementação foi realizado através da linguagem de programação Python e da plataforma de desenvolvimento Jupyter Notebook.

As etapas de desenvolvimento seguiram uma ordem pré-estabelecida na descrição do trabalho:

- Rotacionar uma imagem I por um ângulo específico através de mapeamento direto e mapeamento reverso;
- Exibir o módulo da DCT de I , sem o nível DC, e o valor do nível DC;
- Encontrar e exibir uma aproximação de I obtida preservando o coeficiente DC e os 'n' coeficientes AC mais importantes;
- Através de um filtro passa-baixas ideal quadrado, encontrar a imagem resultante da filtragem de I .

ROTAÇÃO

O trabalho consiste em rotacionar nossa imagem por um ângulo que será especificado no código mantendo o detalhamento da imagem original nas devidas proporções.

Para tal será utilizada uma família de equações já estudadas em sala de aula que utilizam ângulos, senos e cossenos para encontrar a posição dos pixels e realizar a rotação. Para isso temos dois métodos que podem ser utilizados: rotação por mapeamento direto e rotação por mapeamento reverso.

ROTAÇÃO POR MAPEAMENTO DIRETO

Abaixo vemos as fórmulas utilizadas para rotação por mapeamento direto. Estas fórmulas envolvem ponto flutuante, porém, por serem coordenadas, precisam ser números inteiros; para isto é feito o arredondamento. Este arredondamento pode causar problemas, pois mais de um valor pode cair no mesmo pixel, fazendo com que alguns pixels fiquem vazios deixando buracos na imagem. Por não garantir que a imagem é 100% preservada após a rotação, este método não é utilizado na prática.

Rotação por ângulo θ em torno de (i_c, j_c)

$$i' = \text{round}((i - i_c) \cos \theta - (j - j_c) \sin \theta + i_c)$$

$$j' = \text{round}((i - i_c) \sin \theta + (j - j_c) \cos \theta + j_c)$$

$(i_c, j_c) = \text{centro da imagem}$

$(i', j') = \text{coordenada rotacionada}$



Mapeamento direto ($\theta = 90, -90, 180$ e $360, 47$ e 194)

ROTAÇÃO POR MAPEAMENTO REVERSO COM INTERPOLAÇÃO BILINEAR

Neste método vemos o “reverso”. Partimos da imagem final rotacionada vazia e com ela voltamos na imagem original para coletar o valor do pixel e escrevê-lo na imagem final. A rotação da imagem é feita sem perda de dados, pois não é preciso arredondamento.

Mapeamento reverso:

rotação por ângulo $-\theta$ em torno de (i_c, j_c)

$$i = (i' - i_c) \cos \theta + (j' - j_c) \sin \theta + i_c$$

$$j = -(i' - i_c) \sin \theta + (j' - j_c) \cos \theta + j_c$$

(i_c, j_c) = centro da imagem rotacionada vazia

(i, j) = coordenada da imagem original

O valor encontrado no pixel $f(i, j)$ da imagem original é atribuído a um certo $g(i', j')$ na imagem rotacionada final. Esse método também sofre com valores de coordenadas fracionados, porém é possível utilizar as equações de reconstrução bilinear, a fim de encontrar o valor do pixel nestas coordenadas.

$$f(i, y) = f(i, j) + (y - j)[f(i, j+1) - f(i, j)]$$

$$f(i+1, y) = f(i+1, j) + (y - j)[f(i+1, j+1) - f(i+1, j)]$$

$$f(x, y) = f(i, y) + (x - i)[f(i+1, y) - f(i, y)]$$

$f(x, y)$ = valor na coordenada fracionada

$$x \in [i, i+1]$$

$$y \in [j, j+1]$$

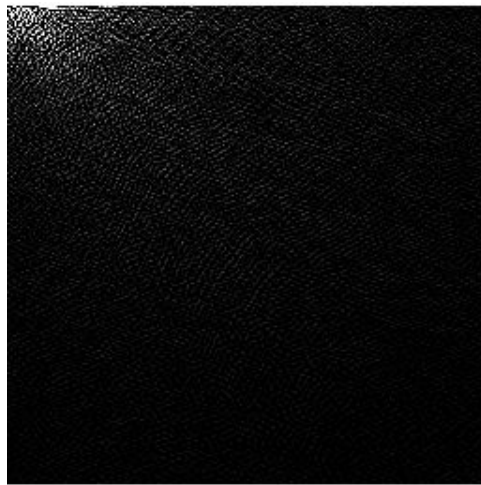


Mapamento reverso ($\theta = 90, -90, 180, 360, 57, -166$)

MÓDULO DCT

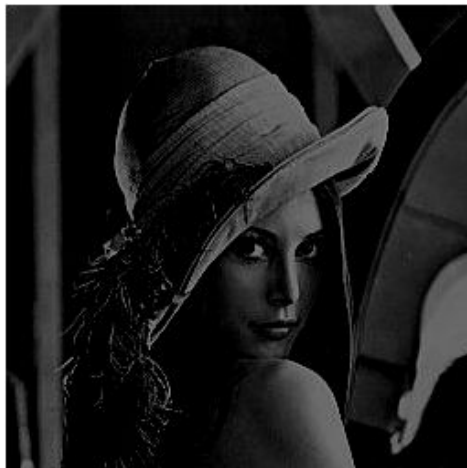
A Transformada Cosseno Discreta (DCT) expressa um sinal em termos de uma soma de senóides com diferentes frequências e amplitudes. A DCT pega o sinal no domínio normal e transforma para o domínio da frequência. A IDCT se trata da DCT inversa, ou seja, pega o sinal no domínio da frequência e transforma para o domínio normal.

Para nosso experimento, a fim de reduzir a complexidade do algoritmo 2D, foi feita a aplicação da DCT e IDCT de uma dimensão em cada linha e cada coluna de pixels da imagem. Para tornar a DCT mais visível foi feito seu módulo e retirada do nível DC.



Resultado da DCT

Nível DC encontrado: 30278,476562500004

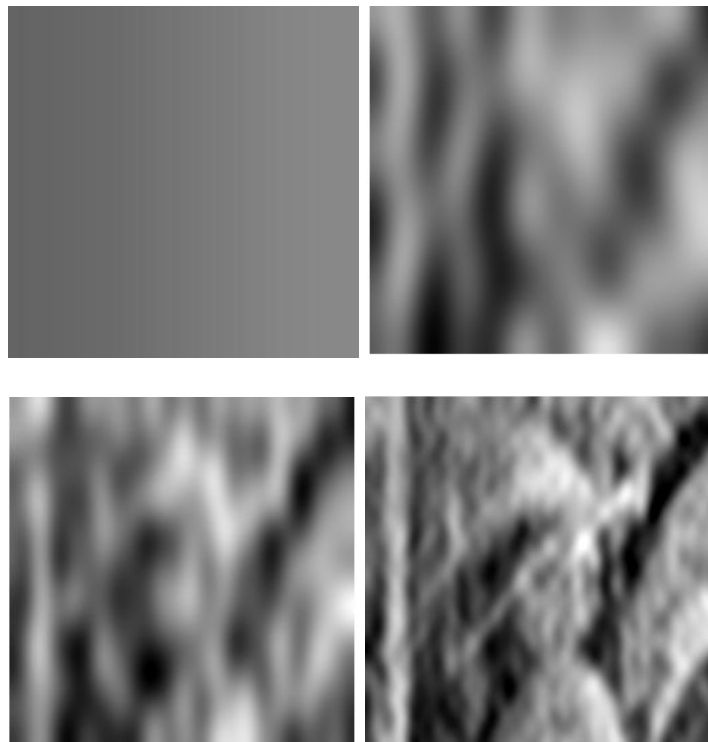


Resultado da IDCT após o módulo e retirada do nível DC

Observa-se que o resultado obtido da imagem foi bem mais escuro que o da imagem original. Tal efeito é resultado da normalização feita para exibição da DCT, onde foi realizado o módulo e retirado o nível DC. O valor do nível DC é muito alto e ao retirá-lo o novo valor mais alto, que será normalizado para 255, vai se aproximar dos outros valores. Portanto, esses novos valores irão ser mais próximos de 255 do que antes, causando o escurecimento da imagem. Ao retirar o nível DC é impossível voltar para imagem original.

COMPRESSOR BASEADO EM COSSENOS

Nesta etapa desenvolvemos métodos para encontrar e exibir uma aproximação da imagem preservando o coeficiente DC e os N coeficientes AC mais importantes. Com isso, zeramos os demais coeficientes, assim ressaltando os mais importantes e tirando valores menos úteis, comprimindo a imagem.



Aproximação usando n coeficientes (n = 1, 20, 50, 200)

Observa-se que ao aumentar o valor de n, mais cossenos entram na composição da imagem, fazendo com que sua visualização fique mais clara. Com $n = 1$ é possível ver quase que claramente o cosseno mais importante da imagem.

FILTRO PASSA-BAIXAS

Por último, foi realizada uma filtragem da imagem por um filtro passa-baixas ideal quadrado. O passa-baixas é um filtro que suaviza a imagem, desfocando-a, pois retira vários cossenos acima do valor da frequência de corte. O filtro é ideal pois as frequências são totalmente preservadas (abaixo de f_c) ou completamente descartadas (acima de f_c), contudo isso não significa que ele é bom, mas sim que é praticamente impossível de se fazer na prática.



Resultado filtro passa-baixas ($f_c = 45, 200$)

f_c = frequência de corte

A diferença do filtro passa-baixas e o compressor baseado em cossenos, é que o filtro deixa passar os valores de menor frequência, já o compressor preserva os valores mais importantes, que podem não necessariamente ser os de menor frequência.

CONCLUSÃO

Ao final do trabalho, foi possível perceber que o conhecimento adquirido em sala de aula foi aplicado com sucesso em todo o processo de construção dos tópicos referentes ao projeto. Além disso, foi possível agregar conhecimento referente a plataforma de programação Jupyter Notebook, utilizada no processo de criação e compilação do código.