

Curso Bacharelado em Ciência da Computação

Disciplina: Algoritmos

Aluno: Erick Scur Padilha

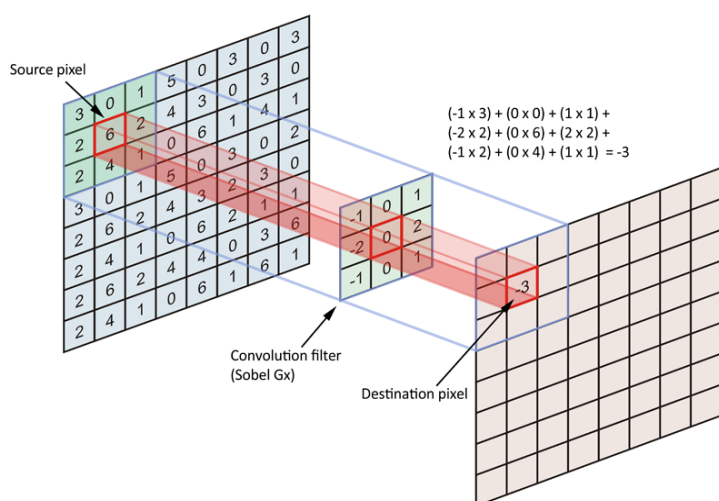
Docente: Manassés Ribeiro

RELATÓRIO DE DESENVOLVIMENTO DO TRABALHO FINAL - PROCESSAMENTO DE ARRAY 2D

1. INTRODUÇÃO

Este relatório tem como objetivo apresentar a lógica e a linha de pensamento para implementar em linguagem C, um código de processamento de Array de duas dimensões ou também chamado de filtro de convolução.

O algoritmo consiste em duas estruturas bidimensionais(matriz), uma sendo a fonte e outra sendo o filtro ou kernel, onde serão realizados cálculos sobre cada pequeno conjuntos de elementos, para assim criar uma terceira matriz com os resultados, como exemplifica a figura abaixo:



Um exemplo de aplicação prática do processamento de array, seria uma convolução 2D onde um filtro de convolução é aplicado sobre uma imagem para gerar uma imagem filtrada.

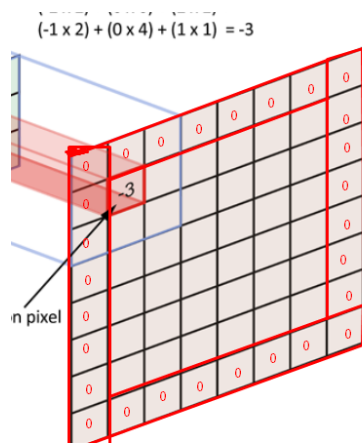
2. METODOLOGIA

Primeiramente é necessário criar uma função para alocar a memória necessária para as matrizes, a função receberá um valor inteiro como parâmetro, que será o tamanho da matriz (usuário irá fornecer este valor) e irá retornar um ponteiro para um ponteiro que aponta para a matriz. Essa função primeiramente vai alocar um vetor (matriz = (int*) malloc(sizeof(int) * tam);) e depois para cada posição desse vetor vai alocar mais um vetor de mesmo tamanho:

```
for(i=0 ; i<tamanho ; i++){  
    matriz[i]=(int*) malloc(sizeof(int) * tam);  
}
```

Será necessário criar dois ponteiros, um para a matriz “fonte” e outro para a matriz “resultado” e também uma matriz 3x3 que será o filtro de convolução ou kernel. Após isso, implementar uma função que irá popular a matriz “fonte” com números aleatórios, isso pode ser feito utilizando a função “rand” e definindo sua semente como a função “time”, para que assim não sejam sempre os mesmos números gerados.

É necessário delimitar as bordas da matriz “resultado” como 0, como ilustra a imagem abaixo:



Finalizado isso, implementar uma função que irá pegar a matriz fonte e realizar o cálculo de um “quadrado de 3x3” de seus valores com a matriz filtro e guardar isso em uma variável acumuladora.

Isso pode ser feito utilizando 4 estruturas de repetição “for”, duas para percorrer a matriz fonte, utilizando as variáveis i e j(iniciando em 0 e indo até o tamanho máximo - 2, para que assim respeite as “bordas” da matriz) e duas para percorrer a matriz filtro, utilizando as variáveis k e l(começando em 0 e indo até 2). Para encontrar o valor de soma, é necessário multiplicar o valor de fonte na posição [i+k] [j+l] pelo valor de filtro[k][l] e adicionar na variável soma, após isso, é necessário adicionar esse valor na matriz resultado, na posição[i+1][j+1],como mostra o trecho de código abaixo:

```

for(i=0;i<tam-2;i++){
    for(j=0;j<tam-2;j++){
        soma = 0;
        for(k=0;k<3;k++){
            for(l=0;l<3;l++){
                soma = soma + (fonte[i+k][j+l] * filtro[k][l]);
            }
        }
        resultado[i+1][j+1]=soma;
    }
}

```

Inserido os valores na matriz resultado, imprimir ambas as matrizes fonte e resultado, para o usuário visualizá-las, assim finalizando o funcionamento do algoritmo. A seguir, está um exemplo do funcionamento do programa com uma matriz de tamanho 8x8:

```

Digite o tamanho da matriz: 8
-----
Matriz Fonte(gerada aleatoriamente):
7| 7| 3| 1| 6| 4| 1| 7|
2| 0| 9| 5| 1| 0| 3| 4|
7| 4| 0| 4| 2| 4| 4| 5|
1| 8| 1| 6| 4| 3| 3| 7|
3| 8| 8| 2| 5| 2| 0| 7|
4| 4| 4| 8| 9| 8| 4| 0|
4| 3| 4| 0| 1| 4| 5| 4|
5| 2| 1| 2| 4| 9| 4| 5|
-----
Matriz Resultado:
0| 0| 0| 0| 0| 0| 0| 0|
0| 3| 4| -11| -7| 1| 12| 0|
0| -7| 3| -1| -8| 5| 10| 0|
0| -2| -10| 5| -6| -5| 14| 0|
0| 10| -10| 2| -3| -16| 6| 0|
0| 5| -1| 4| 4| -11| -11| 0|
0| -4| -2| 2| 15| 3| -12| 0|
0| 0| 0| 0| 0| 0| 0| 0|

```