



## **Segundo Trabalho**

**Objetivo:** Este trabalho tem como objetivo a verificação de aprendizagem da criação e utilização de recursos da linguagem de programação C envolvendo arquivos, bibliotecas e gerenciamento de memória por meio da implementação de uma aplicação de processamento de imagens.

### **Instruções:**

- O trabalho poderá ser feito em dupla.
- O trabalho deverá ser implementado na linguagem C.
- O trabalho vale de 0 a 10 e corresponde a segunda avaliação periódica.
- A avaliação do trabalho será feita nas datas combinadas na forma de uma apresentação do trabalho para os professores da disciplina

### **Descrição:**

- Deverá ser implementado o algoritmo de segmentação *watershed* para operar em imagens do tipo pgm.
- O trabalho deverá ser modularizado com pelo menos 1 biblioteca:
  - Biblioteca para uma fila dinâmica.
    - Definição do tipo de dado a ser armazenado na fila
    - Enfileirar elemento
    - Desenfileirar elemento
    - Verificar se a fila está vazia
    - Entre outras funções necessárias para a fila;
- O programa deverá receber uma imagem, aplicar o pré-processamento, aplicar o *watershed* e salvar a imagem resultante destacando os segmentos da imagem.
- A seguir será apresentada uma descrição do algoritmo watershed com alguns detalhes de como deverá ser feito no trabalho, para facilitar a implementação. Além disso, estão destacadas na parte de referências livros e artigos que podem auxiliar na compreensão do algoritmo.

### **Segmentação com *watershed***

O algoritmo watershed é um algoritmo de segmentação de imagens que faz a rotulação de regiões.



- Para a aplicação do watershed, é considerado um conjunto de coordenadas, os quais são chamados de marcadores. No contexto do trabalho, esses marcadores serão lidos de um arquivo de entrada.
- A aplicação do watershed deve ser feita sobre o gradiente da imagem pois os resultados se tornam mais interessantes.
  - As operações de gradiente morfológico, leitura e escrita em arquivos pgm/ppm, leitura dos marcadores, e outras funções auxiliares estão disponíveis para download no classroom. Além do código, uma imagem e arquivo com um conjunto de marcadores que pode ser usada para teste.
    - Caso a equipe deseje, pode utilizar outras imagens e outros marcadores para realizar mais testes.
- Para a implementação, é utilizada uma estrutura de dados chamada de fila hierárquica. A seguir será dada uma descrição de uma fila hierárquica.

### Fila Hierárquica

- Uma fila hierárquica é uma fila que respeita uma hierarquia em suas inserções e remoções;
- Pode-se ver uma fila hierárquica como uma coleção de N filas, onde cada uma delas segue a política de uma fila normal (FIFO).
- As operações de inserção e remoção pode ser descritas da seguinte forma:
  - **Inserção:** Sempre que um elemento é inserido na fila hierárquica, deve-se informar junto ao elemento qual o nível da hierarquia deste elemento. O elemento deverá ser inserido na fila correspondente a sua hierarquia.
  - **Remoção:** As remoções na fila hierárquica são feitas sempre no menor nível da hierarquia (Em alguns contextos podem ser feitas no maior). Sendo assim, sempre que se deseja remover um elemento, deve-se procurar pela fila de menor nível de hierarquia e remover seu primeiro elemento.
- A Figura 1 ilustra algumas operações em uma fila hierárquica com 5 níveis de hierarquia que armazena coordenadas cartesianas.
  - Observe que na Figura 1 o elemento de dado (5,0) foi inserido na fila hierárquica com hierarquia 0 (a hierarquia é um argumento da função de inserção). Sendo assim, ele será o último a sair da hierarquia 0, porém nenhum outro elemento com hierarquia maior poderá sair antes dele. Isso ocorre após duas remoções.



- Caso a próxima operação seja uma operação de desenfileirar, como a fila 0 está vazia, o elemento a ser removido seria o primeiro que foi inserido no nível 1 (ou seja, o elemento de dado (0,0)).

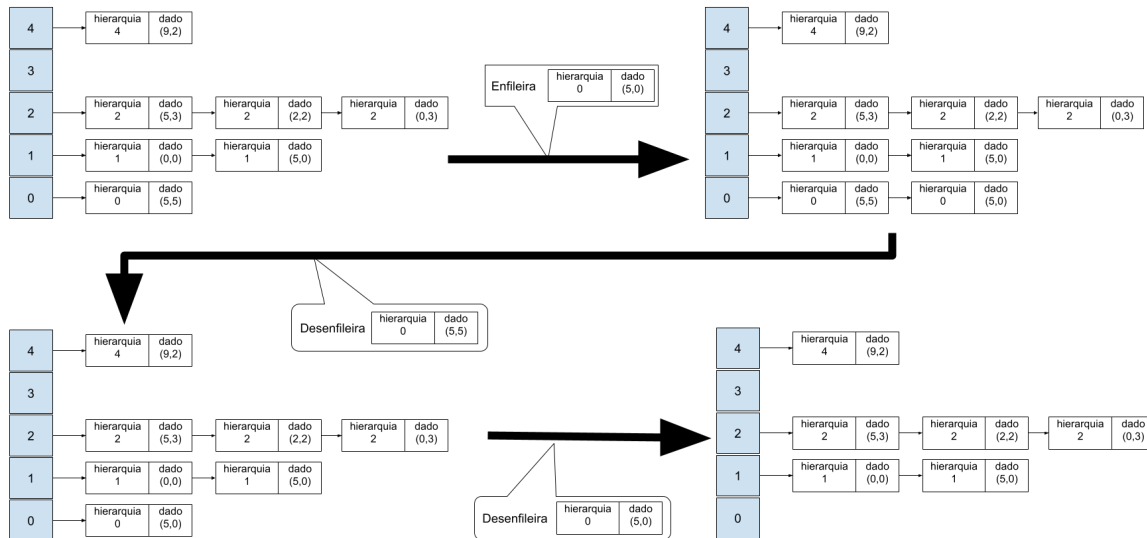


Figura 1: exemplo de operações de enfileirar e desenfileirar em uma fila hierárquica

- Com o conhecimento do funcionamento da fila hierárquica se torna possível a implementação do *watershed*.

#### Implementação do *watershed*

- O algoritmo *watershed* trabalha com uma fila hierárquica referente a quantidade de tons de cinza na imagem da imagem de entrada (neste trabalho, será utilizado o gradiente da imagem).
- Deve-se inserir na fila hierárquica as coordenadas do pixel e também seu rótulo;
- Toda inserção na fila hierárquica deve ser feita no nível correspondente ao tom de cinza da imagem (lembrando que a imagem aqui é o gradiente da imagem de interesse);
- Ao remover um elemento da fila, seus vizinhos que ainda não foram inseridos devem ser inseridos com o mesmo rótulo do pixel removido;
- Um pseudocódigo do algoritmo é apresentado no Algoritmo 1.



```
1.  /*
2.  Entradas:
3.      Imagem G: Gradiente que se deseja aplicar o watershed
4.      marcadores[]: Vetor de marcadores iniciais
5.  Saída:
6.      Imagem Ws: Grade com as mesmas dimensões da imagem G
           que será usada para computar o watershed
7.  */
8.  Watershed (Imagem G, Marcador marcadores[]){
9.      FilaHierarquica FH;
10.     Imagem Ws = ImagemDeZeros;
11.     para cada Marcador m  $\in$  marcadores {
12.         Enfileira(FH, m, G[m.linha][m.coluna]);
13.     }
14.     enquanto não Vazia(FH) {
15.         Marcador Pix = Desenfileira(FH);
16.         Ws[Pix.linha][Pix.coluna] = Pix.rotulo
17.         para cada Pixel v  $\in$  vizinhos(Pix){
18.             se v.rotulo == 0 {
19.                 mv = NovoMarcador(v.linha, v.coluna, Pix.rotulo);
20.                 v.rotulo = Pix.rotulo
21.                 Enfileira(FH, mv, G[v.linha][v.coluna]);
22.             }
23.         }
24.     }
25.     retorne Ws;
26. }
```

Algoritmo 1: Watershed

## Referências

**Processamento digital de imagens / Rafael C. Gonzalez, Richard E. Woods ; tradução: Cristina Yamagami, Leonardo Piamonte. Edição: 3. ed. Publicação São Paulo : Pearson/Prentice Hall, 2010. Capítulo 10.**



Universidade Estadual de Maringá  
Centro de Tecnologia  
Departamento de Informática  
Profs: Airton Marco Polidório  
Nilton Luiz Queiroz Junior



**Beucher, S. and Meyer, F. [1992]. "The Morphological Approach of Segmentation: The Watershed Transformation," in Mathematical Morphology in Image Processing, E. Dougherty (ed.), Marcel Dekker, New York.**