

Lista de Implementação 2

1. Implemente o algoritmo de rotulação de componentes conexos em imagens binárias (8 ou 16 bits). Recebe uma imagem I e um inteiro $n = \{4, 8\}$.
2. Implemente um algoritmo que mapeia rótulos para cores RGB pseudoaleatórias.
 $(RGB) = (C * K_r \bmod 256, \dots, \dots)$
3. Implemente uma função que recebe uma imagem binária, um elemento estruturante B e um inteiro n de 1 a 3. retorna o gradiente morfológico em função de n:
 $n=1: \beta_d \ n=2: \beta_e \ n=3: \beta_f$
4. Implemente o algoritmo de dilatação condicional. Recebe uma imagem I, uma máscara M e um elemento estruturante B. Retorna $(I + B) \cap M$
5. Implemente uma função que recebe duas imagens A e B. Retorna true se $A=B$ e false se $A \neq B$.
6. Implemente o algoritmo de extração de componentes conexos. Recebe uma imagem A, um marcador Y (imagem com 1 pixel diferente de zero) e um elemento estruturante B.
7. Implemente uma função que recebe um elemento estruturante B e retorna seu complemento. Pixeis “dont care” são dados por -1 e seu complemento é -’.
8. Implemente uma função que recebe um elemento estruturante B, um inteiro n e retorna B rotacionado por $n * \frac{\pi}{4} rad$ no sentido horário.
9. Implemente uma função que retorna o resultado do hitmiss de uma imagem de entrada limiarizada pelo elemento estruturante B.
10. Implemente uma função que recebe uma imagem de entrada I, um elemento estruturante B e uma lista de inteiros n_i . Retorna a união dos hitmss de I por B rotacionado por $n_i * \frac{\pi}{4} rad$, onde n_i é o i-ésimo item da lista n_i .