

Projeto 1 de Introdução ao Processamento de Imagens

Terceira Parte

Gabriel Martins de Miranda
130111350
Universidade de Brasília
Email:gabrielmirandat@hotmail.com

Abstract—Trabalho baseado nos templates IEEE.

I. RESUMO

O algoritmo apresentado neste relatório apresenta um método para, dada uma imagem binária (uma imagem que apresenta apenas dois níveis de cinza, neste caso preto=0 e branco=255) contabilizar quantas manchas existem nela (conjunto de pixels pretos envoltos por pixels brancos) e se existem buracos nestas manchas (conjunto de pixels brancos envoltos por pixels pretos). Foram utilizadas quatro imagens de teste, mostradas na seção *Resultados*.

II. INTRODUÇÃO

Algumas considerações iniciais para o entendimento do leitor. Uma imagem pode ser vista como uma matriz, em que cada elemento recebe o nome de pixel e é a menor unidade da imagem. Se a imagem não é colorida, como é o caso, ela é representada em níveis de cinza, que vão de 0 a $2^n - 1$, onde n representa o número de bits que representam um pixel. O trabalho consiste em manipular estas representações. A resolução da imagem indica quantos pixels ela tem. No exemplo, $512 \times 512 = [n \text{ de linhas} \times n \text{ de colunas}]$ da matriz, totalizando em 262144 pixels.

III. METODOLOGIA

Dada a imagem original, $p = p(x, y)$, foi criada uma nova imagem $f = f(x, y)$ com uma coluna a mais no extremo esquerdo e uma linha a mais na parte superior. Esta adaptação fez-se necessária pois $\forall (x, y) \in \text{Dom}(p)$, foram comparados os pixels $(x-1, y)$ e $(x, y-1)$, sendo x o número de linhas e y o número de colunas. Criou-se uma matriz de rótulos, que chamaremos por $g = g(x, y)$, do mesmo tamanho de f , onde todo pixel branco $f(x, y) = 255$ foi mapeado para $g(x, y) = 1$ e nenhuma mudança foi feita onde $f(x, y) = 0$.

Percorreu-se f pixel a pixel, e cada vez que $f(x, y) = 0$ algumas comparações foram feitas, a saber:

Se $t = f(x-1, y)$, $r = f(x, y-1)$, $tr = g(x-1, y)$ e $rr = g(x, y-1)$

- Se $r == 255$ e $t == 255$, $g(x, y) = \text{novo rotulo}$.
- Se $r == 0$ e $t == 0$ e $tr == rr$, $g(x, y) = tr$.
- Se $r == 0$ e $t == 0$ e $tr \neq rr$, estabeleceu-se a equivalência entre tr e rr , sendo $\forall i \forall j \in [2..x] \times [2..x]$, se $g(i, j) == \max(tr, rr) \Rightarrow g(i, j) = \min(tr, rr)$.

- A matriz de rótulos foi 'normalizada' para apresentar apenas valores de 2 a $N_{de_manchas} + 1$ para os pixels que representam as manchas.
- A rotulação das manchas foi similar a dos buracos.

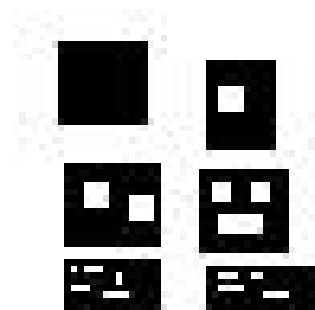
Funções utilizadas:

- 1) *preprocessamento.m*: Tem por objetivo binarizar a imagem original para que os pixels apresentem apenas $\text{branco} = 255$ e $\text{preto} = 0$.
- 2) *proc_manchas.m*: mapear a matriz de rótulos das manchas. A função retorna a matriz de rótulos das manchas e um vetor_linha cujos elementos representam os rótulos atribuídos as manchas e cujo *size* é igual ao número de manchas.
- 3) *proc_buracos.m*: mapear a matriz de rótulos dos buracos. Aqui a f foi invertida, assim como a g e o mesmo processo foi feito.
- 4) *juntador_mancha_buraco.m*: Aqui foi criada uma nova matriz $z = z(x, y)$ dada por $z(x, y) = \max(\text{rotulos_das_manchas}(x, y), \text{rotulos_dos_buracos}(x, y))$.
- 5) *detector.m*: realiza a quantificação de buracos para cada mancha. Aqui foi utilizado o mesmo vetor de manchas criado anteriormente. Foi criada uma nova linha para o vetor, tornando-o uma matriz $[2 \times n \text{ de manchas}]$. A cada verificação de que um buraco pertence a uma mancha, sendo $(1, n)$ o índice da mancha, $(2, n)$ é incrementado.
- 6) *resultado.m*: Mostra a interface com o usuário, mostrando a saída gráfica na tela e os resultados na janela de comando.

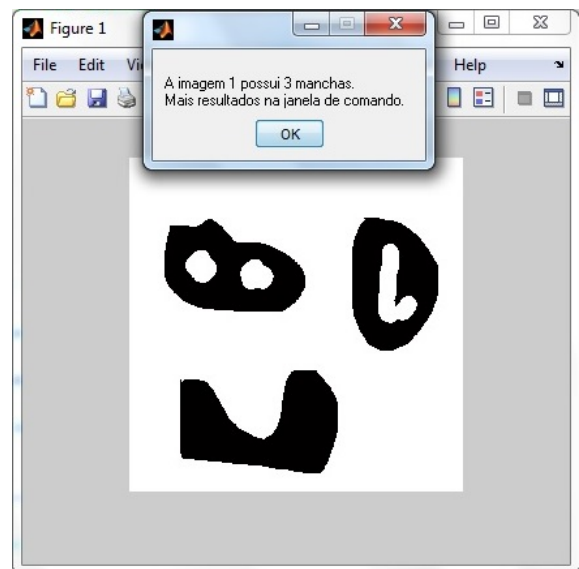
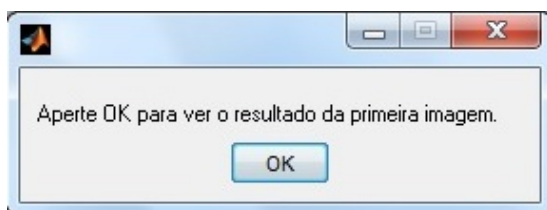
IV. RESULTADOS

Saídas gráficas:

- Imagens analisadas:



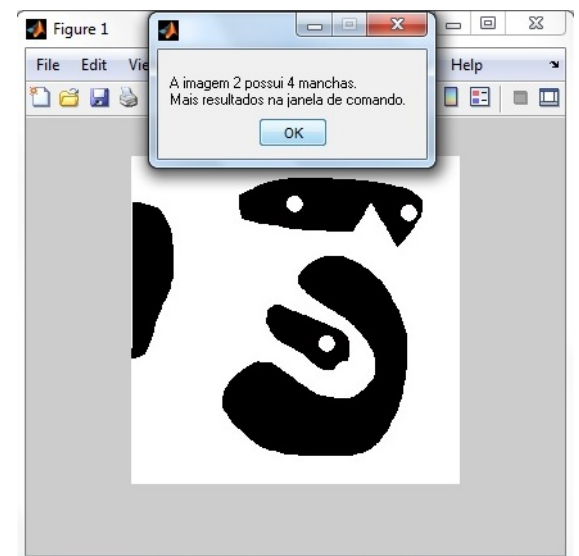
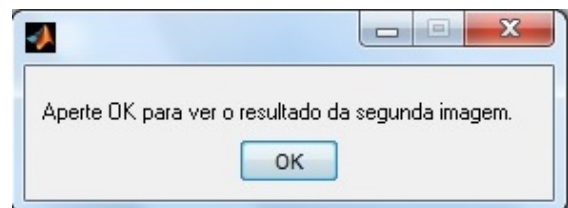
- Resultados da primeira imagem:



RESULTADO IMAGEM 1

A mancha 1 possui 1 buraco(s) .->TIPO B
A mancha 2 possui 2 buraco(s) .->TIPO B
A mancha 3 não possui buraco(s) .->TIPO A

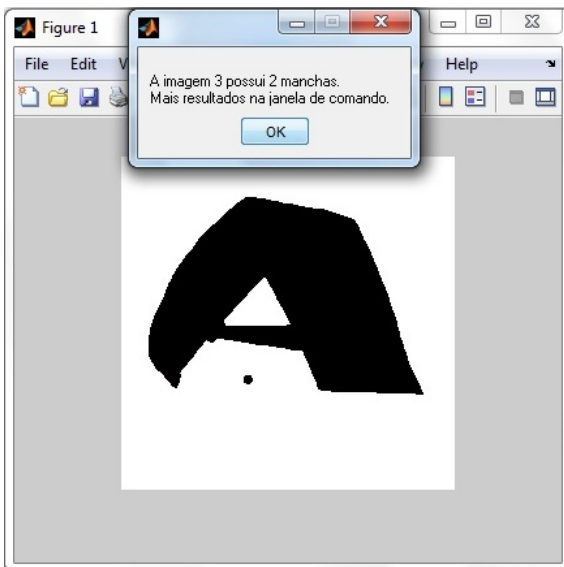
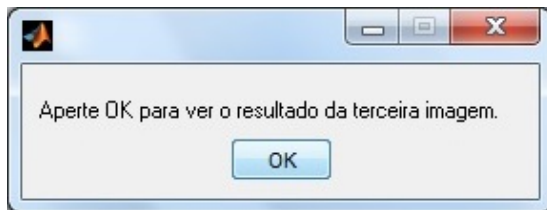
- Resultados da segunda imagem:



RESULTADO IMAGEM 2

A mancha 1 possui 2 buraco(s).->TIPO B
A mancha 2 não possui buraco(s).->TIPO A
A mancha 3 não possui buraco(s).->TIPO A
A mancha 4 possui 1 buraco(s).->TIPO B

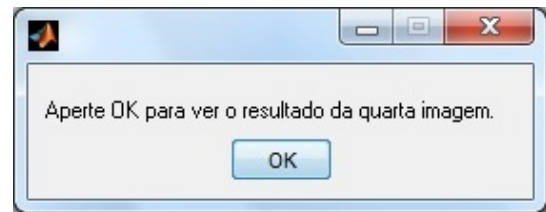
- Resultados da terceira imagem:



RESULTADO IMAGEM 3

A mancha 1 possui 1 buraco(s).->TIPO B
A mancha 2 não possui buraco(s).->TIPO A

- Resultados da quarta imagem:



RESULTADO IMAGEM 4

A mancha 1 não possui buraco(s).->TIPO A
A mancha 2 possui 1 buraco(s).->TIPO B
A mancha 3 possui 2 buraco(s).->TIPO B
A mancha 4 possui 3 buraco(s).->TIPO B
A mancha 5 possui 5 buraco(s).->TIPO B
A mancha 6 possui 4 buraco(s).->TIPO B

V. CONCLUSÃO

Foi possível observar a importância de algoritmos de detecção de padrões em imagens, que podem ser usados em diversas áreas, como por exemplo na área de saúde.