Universidade Tecnológica Federal do Paraná – UTFPR
Departamento Acadêmico de Eletrônica – DAELN
seu curso aqui
Disciplina: IF69D – Processamento Digital de Imagens
Semestre: 2020/1 ADNP
Prof: Gustavo B. Borba

RELATÓRIO
OMR para a correção de folhas de respostas de provas de múltipla escolha

Alunos:
Gabriel Carrico Guerrero / 1860216

11.2020

---

1 Objetivo

Implementar um sistema de Optical Mark Recognition (OMR) para ser aplicado na correção de provas. O sistema desenvolvido deve ser capaz de: adquirir as imagens das folhas de resposta (FR); identificar as MO e a resposta associada a cada questão; comparar as respostas com as de um gabarito; atribuir uma nota.

2 Fundamentação Teórica

Morfologia significa estudo da forma. Então as operações morfológicas tem como objetivo analisar e manipular a forma dos objetos na imagem. Sua base matemática vem da teoria dos conjuntos.

Um conjunto de pixels binários, denominados de 'elemento estruturante' varre a imagem. Com isso são realizadas as operações morfológicas. A forma do elemento estruturante depende da aplicação, neste caso estamos utilizando o formato de disco.

A descrição mais formal das operações morfológicas pode ser encontrada nas referências. [2].

Algumas definições:

1) O ponto de referência do elemento estruturante é o equivalente em morfologia ao ponto central das janelas de convolução quadradas. Este ponto de referência também é chamado de ponto central do elemento estruturante. Mas como este ponto central muitas vezes não é central, geometricamente falando, vamos usar o termo hot spot, como em [1]. Chamaremos o elemento estruturante de S. No Octave o hot spot é denominado origin e às vezes center.

2) Estamos aplicando morfologia em imagens binárias (bw). Vamos convencionar que os pixels dos objetos possuem sempre valor '1' (pixels brancos). Logo, os pixels do fundo possuem sempre valor '0' (pixels pretos). Vamos chamar a imagem de entrada de B. A mesma idéia pode ser usada para os pontos do elemento estruturante: um ponto ativo do elemento estruturante tem valor '1'. Logo, um ponto não ativo (irrelevante) do elemento estruturante tem valor '0'.

3) A vizinhança-4 (4-neighborhood) de um pixel é definida pelos dois pixels adjacentes à ele na

horizontal e pelos dois pixels adjacentes à ele na vertical. 4) A vizinhança-8 (8-neighborhood) de um pixel é definida pela vizinhança-4, mais os 4 pixels diagonais à ele.

5) Notação:

$$B \oplus S \tag{1}$$

- Dilatação da imagem B pelo elemento estruturante S.

$$B \ominus S \tag{2}$$

- Erosão da imagem B pelo elemento estruturante S.

$$B \bullet S \tag{3}$$

- Fechamento da imagem B pelo elemento estruturante S.

6)A saída das operações morfológicas deve ser escrita em uma nova imagem, inicialmente preenchida com zeros.

Dilatação 1 Cada vez que o hot spot de S encontra um pixel '1' de B, S é replicado na imagem de saída.

Erosão 2 O pixel '1' de B correpondente ao hot spot de S só é mantido na imagem de saída se todos os pixels '1' de S possuem pixel '1' correspondente em B.

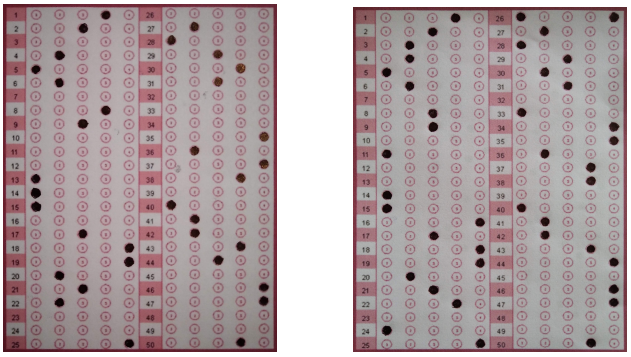
Fechamento 3 É a dilatação seguida da erosão, isto é, aplicar a dilatação na imagem B usando S e depois aplicar a erosão na imagem dilatada usando S. A operação de fechamento atua nos objetos da imagem principalmente das seguintes formas: suaviza contornos, funde quebras estreitas, elimina pequenos buracos, preenche espaços de contornos.

3 Implementação

Antes do início do algoritmo, as imagens são capturadas através de um smartphone. O modelo utilizado na captura foi um Asus X01BDA. Então foi utilizado um aplicativo externo para scanner, utilizando-se apenas o recurso de transformação geométrica. O aplicativo utilizado foi o Smart Doc Scanner.

O algoritmo começa obtendo as duas imagens que serão utilizadas. As respostas 1(b) do aluno são armazenadas na variável 'respostas' e o gabarito 1(a), na 'gabarito'. Então ambas as imagens são enviadas para a função 'cropa'.

Esta função tem como finalidade fazer os cortes iniciais da imagem, e prepará-la para a extração de informações posterior. Ela recebe como parâmetro uma imagem RGB e retorna uma imagem lógica.



(a) *(b)*

Figura 1: (a) Imagem do Gabarito. (b) Imagem das respostas.

3.1 Cropa

Primeiramente a imagem passa por um algoritmo de seleção de região de interesse baseado em cor. Como estamos interessados nas marcações feitas pelo aluno, o range escolhido foi entre 0 e 40. A função utilizada para a medida foi o 'roicolor'. Como o roicolor opera em cada canal separadamente, uma operação de max entre os três canais 2(a) 2(b) 2(c) é feita para formar a imagem lógica final, 'BWfinal' 2(d).

Em seguida, a imagem sofre duas operações de morfologia, 'imdilate' 3(a) 1 e 'imerode' 3(b) 2, ambas utilizando como parâmetro strel('disk',15,0), para efeitos de diminuir a quantidade de falsos positivos e agregar melhor as marcações. Também conhecido como fechamento da imagem, ou 'imclose'.

A terceira etapa é um tratamento das regiões espaciais da imagem. Para isso, um find encontra todos os pixels que estão marcados com 1. Em seguida, um novo corte é feito, numa região que vai do primeiro ao último pixel que foi assinalado com um 1.

Por último, uma remoção adicional é feita, buscando se livrar da borda colorida. Esta borda é útil para a obtenção e linearização da foto, mas tem pouca serventia no código. A imagem é então cortada ao meio e concatenada, a fim de formar uma tripa única 4(a) que poderá ser analisada posteriormente.

Essa imagem final é então retornada para a função 'main', que em seguida a enviará para a 'crialista'.

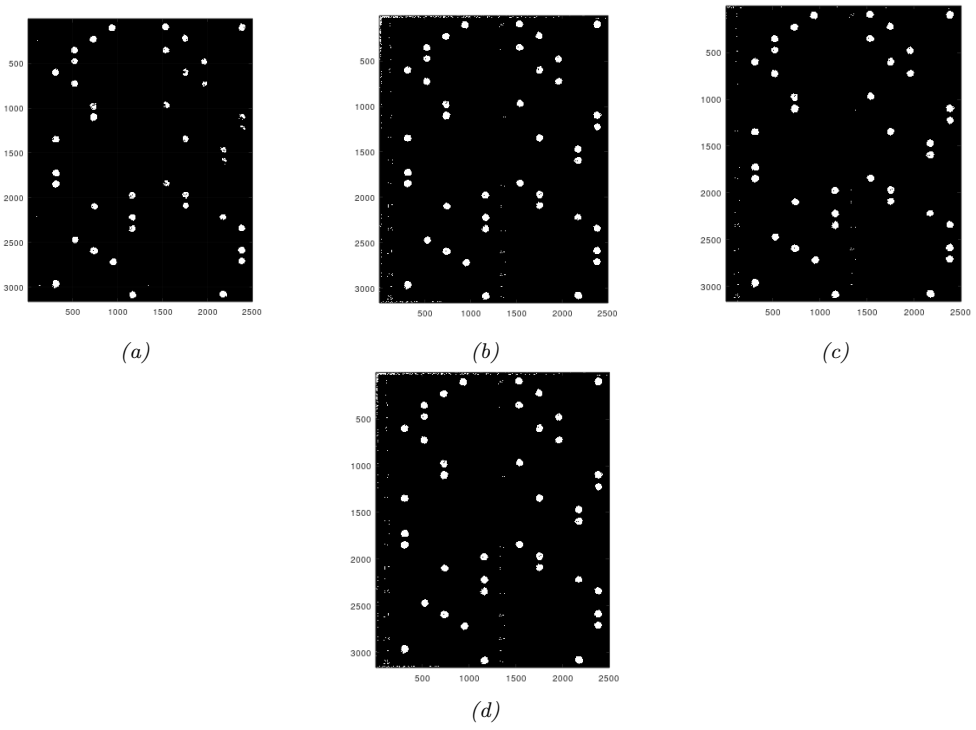
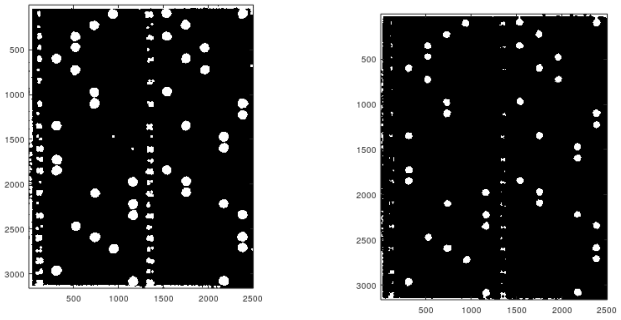


Figura 2: (a) roicolor canal vermelho. (b) roicolor canal verde. (c) roicolor canal azul. (d) MAX dos três canais.

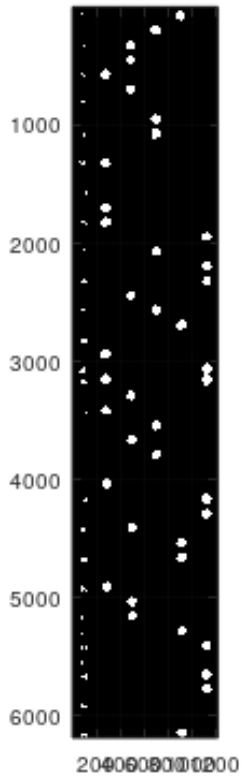
3.2 Cria Lista

A função 'crialista' tem seu funcionamento bastante simples. Ela recebe como parâmetro os resultados de 'cropa' e retorna um vetor com os valores correspondentes de



(a) *(b)*

Figura 3: (a) Respostas pós imdilate. (b) Respostas pós imerode.



(a)

Figura 4: (a) Resultado da concatenação numa faixa única de valores.

cada resposta. Para tal ela basicamente corta a imagem em 50 faixas iguais através de um laço for.

Esses cortes são então enviados para a função 'encontraValor' para serem processados um a um.

Dentro da função 'encontraValor' essas faixas são partidas em ainda mais pedaços menores. Para ser mais exato, são seis cortes. O primeiro representa o número da questão e é descartado. Os outros cinco são as alternativas 'a', 'b', 'c', 'd' e 'e'.

A função então passa a contabilizar o número de pixels de valor 1 dentro de cada subseção. Caso o valor ultrapasse uma base estipulada (no caso deste algoritmo, 2000),

ela assinala que aquela alternativa foi marcada.

O algoritmo por fim confere se alguma outra alternativa já havia sido assinalada para aquela questão. Caso sim, ele retorna um valor de invalidez para a mesma.

As respostas de cada questão são então retornadas da seguinte forma: '0' caso em branco 5(b), de '1' até '5' caso sejam as respostas de 'a' até 'e' e '6' em questões com mais de uma alternativa selecionada 5(a).

(a) *(b)*

Figura 5: (a) Exemplo de uma resposta nula, neste caso 'a' e 'e' foram marcados. (b) Exemplo de uma resposta branca, neste caso esse objeto detectado é justamente a numeração da questão.

3.3 Contagem final

Munido das duas listas, uma para o gabarito e outra para as respostas do aluno, o algoritmo passa a comparar ambas através de um laço for. Caso o valor da resposta for 0, incrementa-se o campo 'brancos'. Caso o valor da resposta for 6, incrementa-se o campo 'nulos'. Se nenhum dos dois casos, confere-se se a resposta é similar ao gabarito, em caso positivo, incrementa´se o campo 'acertos'.

Ao final, uma mensagem é exibida com o valor de acertos, brancos e nulos. Os resultados são então salvos num arquivo chamado 'resultados.mat'.

4 Resultados e conclusões

Para fins de demonstração, as duas imagens utilizadas para o funcionamento do algoritmo tiveram o seguinte resultado

Gabarito 1(a):

4 3 0 2 1 2 0 4 3 0 0 0 1 1 1 0 3 5 5 2 3 2 0 0 5
0 2 1 3 4 3 0 0 0 5 2 5 4 0 1 2 2 4 3 0 5 5 0 0 4

Respostas 1(b):

4 3 2 2 1 2 0 3 3 0 1 0 0 1 1 5 3 5 5 2 3 4 0 1 5
6 2 1 3 2 3 0 1 5 5 2 4 4 0 1 2 2 4 5 0 5 5 0 0 4

Os testes realizados demonstraram um funcionamento bastante eficiente do algoritmo. As fotos foram coletadas em diversas horas do dia, com iluminações diferentes e o resultado permaneceu condizente com o esperado.

O índice de acerto da leitura, com duas dezenas de testes, ultrapassou a casa dos 90%. Mais testes são necessários para garantir estes resultados.

Uma versão anterior do gabarito, sem a borda colorida foi discartada por não apresentar resultados tão precisos. A falta da borda prejudicava bastante a determinação de perspectiva da foto.

Outra limitação é a remoção da borda durante o algoritmo. Na versão atual, a remoção é feita na força bruta, recortando um número fixo de píxels da imagem. Um próximo passo interessante seria buscar uma automatização deste processo.

Um caminho posterior interessante para o desenvolvimento da implementação seria o recebimento e processamento de vários gabaritos por vez. Permitindo então um processamento estatístico posterior, podendo se definir quais questões tiveram maior indíce de

falha, quais tiveram maior indíce de acerto, média de uma turma, dentre outros dados relevantes.

Outra melhoria importante é habilitar o algoritmo para leitura do nome e outros dados do participante, uma vez que o atual unicamente faz a leitura das respostas.

Referências

[1] Wilhelm Burger and Mark Burge. *Digital image processing - an algorithmic introduction using Java.* Springer, 2010.

[2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley, 1992.