

Projeto 4

MC920 - Introdução ao Processamento de Imagem Digital

Caio Augusto Alves Nolasco - RA:195181

1. Introdução e especificação do problema

O objetivo do trabalho 4 é, através de operadores morfológicos, segmentar em uma imagem regiões que contenham texto escrito, destacando com um retângulo as linhas de texto.

2. Estrutura geral do algoritmo

Para isso, a imagem original é submetida duas vezes à dilatação seguida de erosão: uma vez com um elemento estruturante de tamanho (1, 200), e outra vez com outro elemento estruturante de tamanho (200, 1). A intersecção das imagens resultantes das duas operações é salva em outra imagem, usando a operação AND bit a bit. Essa intersecção então é submetida ao operador de fechamento, com um elemento estruturante de tamanho (1,30).

Identifica-se as componentes conexas da imagem modificada pelos operadores morfológicos, assim como as coordenadas dos retângulos que melhor envolvem cada objeto encontrado. A partir da densidade de pixels pretos e o número de transições de pixels brancos para pixels pretos, o algoritmo decide para cada retângulo se o objeto é um bloco de texto, e caso sim, desenha o retângulo na imagem original em volta de uma linha de texto.

3. Pressupostos e decisões de design

O programa em Python recebe uma imagem de entrada para aplicar os operadores morfológicos e encontrar as partes com texto. Sobre essa imagem são feitos os seguintes pressupostos:

- A imagem é monocromática;
- A imagem está salva no disco do computador que executa o código;
- A imagem está em formato '.pbm';
- O local de origem do arquivo é dado pelo usuário (ex: /home/cnolasco/arqbitmap.pbm);
- O local de destino das imagens intermediárias do processo e a imagem final é dado pelo usuário (ex: /home/cnolasco/Outputs/);
- As imagens resultantes são salvas em formato '.pbm'.

4. Padrões de implementação

Os operadores morfológicos são realizadas por meio de chamadas de funções da biblioteca OpenCV2. Os elementos estruturantes, para operações que necessitam de um kernel, usam um *array* de uns da biblioteca Numpy. As operações de dilatação são realizadas pela função **cv2.dilate()**, similarmente aos operadores de erosão, fechamento e intersecção, respectivamente chamados por **cv2.erode()**, **cv2.morphologyEx(cv2.MORPH_CLOSE)** e **cv2.bitwise_and()**.

A identificação das regiões de componentes conexas é feita pela função **cv2.findContours()**, que retorna coordenadas dos limites das regiões conexas. Os limites são transformados em retângulos correspondentes pela função **cv2.boundingRect()**, que retorna as coordenadas do retângulo que envolve cada objeto identificado na imagem, assim como sua largura e altura. As regiões segmentadas são analisadas de acordo com a sua densidade de pixels pretos e número de transições de pixels brancos para pixels pretos, e quando é identificado um bloco de texto, o retângulo é desenhado na imagem original.

5. Saída

O programa salva as seguintes imagens: ambas as imagens resultantes da dilatação seguida por erosão, para cada um dos elementos estruturantes; a imagem resultante da operação de intersecção AND; a imagem resultante da operação de fechamento; e por fim, a imagem original com retângulos destacando as regiões que contêm texto.

6. Resultados dos operadores morfológicos

Dilatação e erosão

A primeira operação realizada é a dilatação da imagem original seguida pela operação de erosão, com um elemento estruturante de tamanho (1,100). A dilatação da imagem diminui o tamanho de espaços vazios entre os elementos principais da imagem, preenchendo espaços vazios entre regiões de interesse. A erosão então diminui os elementos dilatados, criando maior espaçamento entre regiões dilatadas. Devido ao tamanho do primeiro elemento estruturante, essa primeira sequência de transformações traz para o plano principal regiões de texto horizontais.

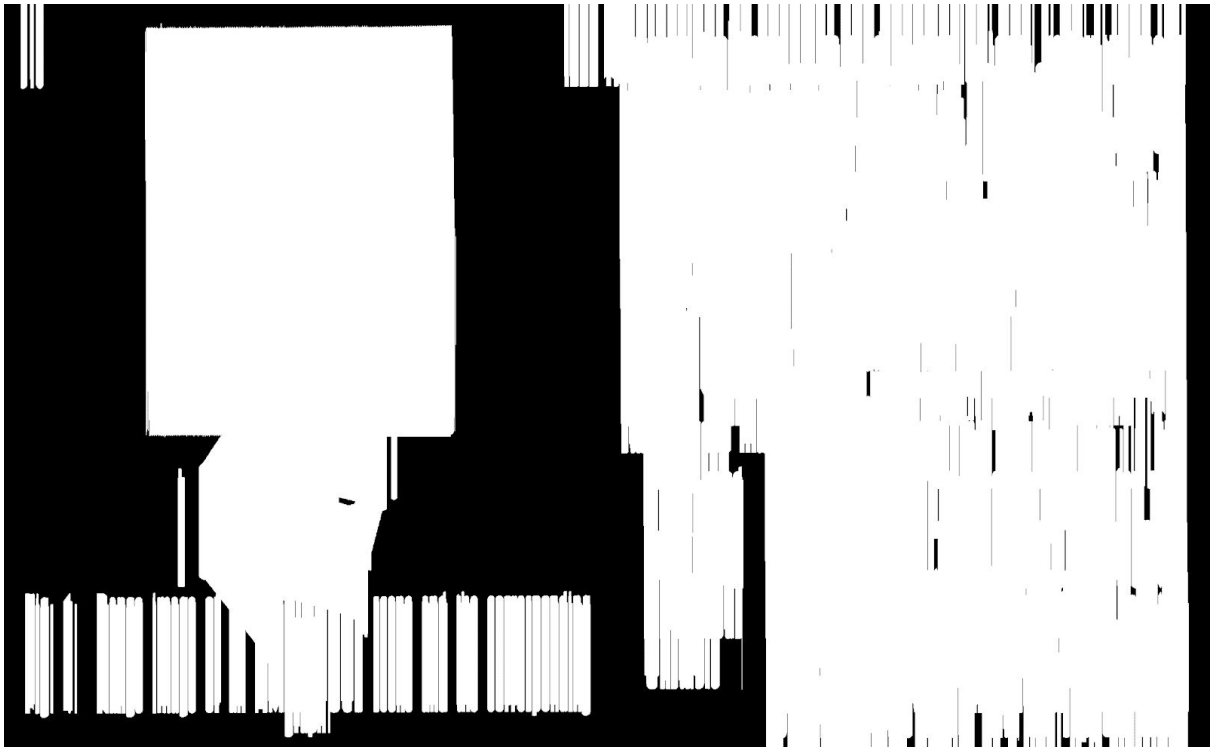


a) Dilatação com elemento estruturante (1,100)

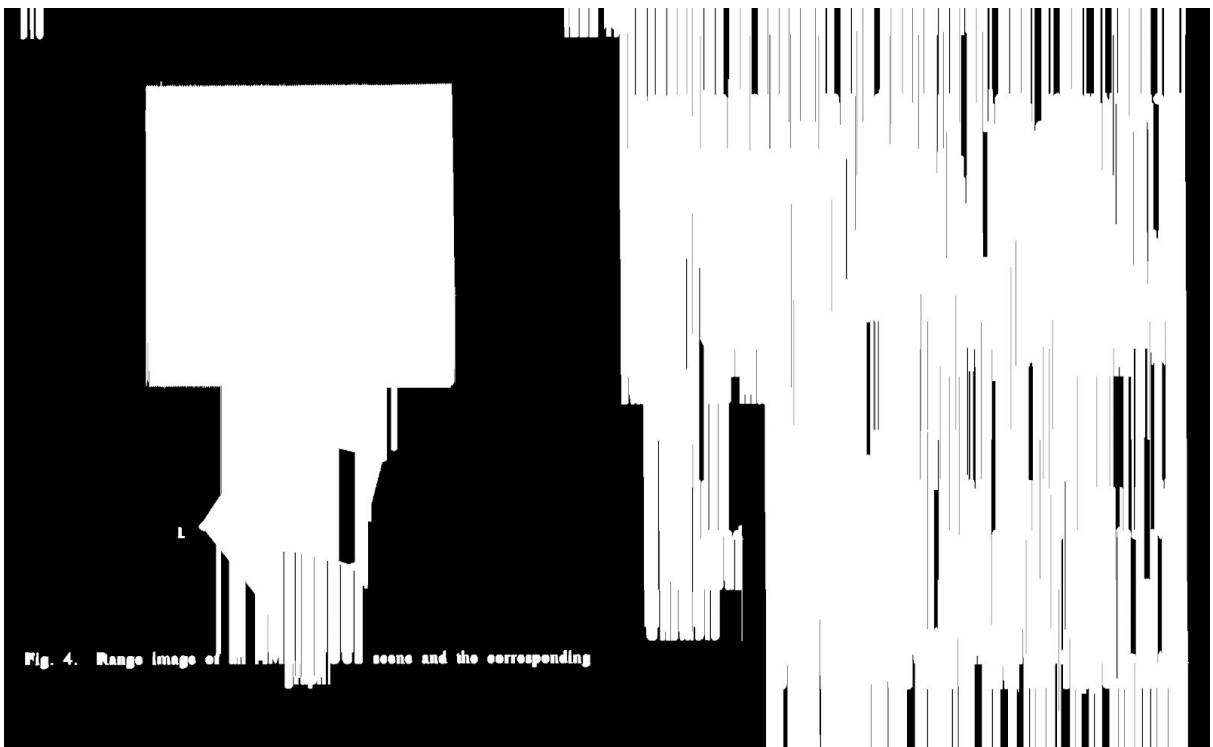


b) Erosão com elemento estruturante (1,100)

Os mesmo operadores morfológicos são aplicados, desta vez com um elemento estruturante de tamanho (200,1), afetando em maior intensidade regiões de interesse verticais.



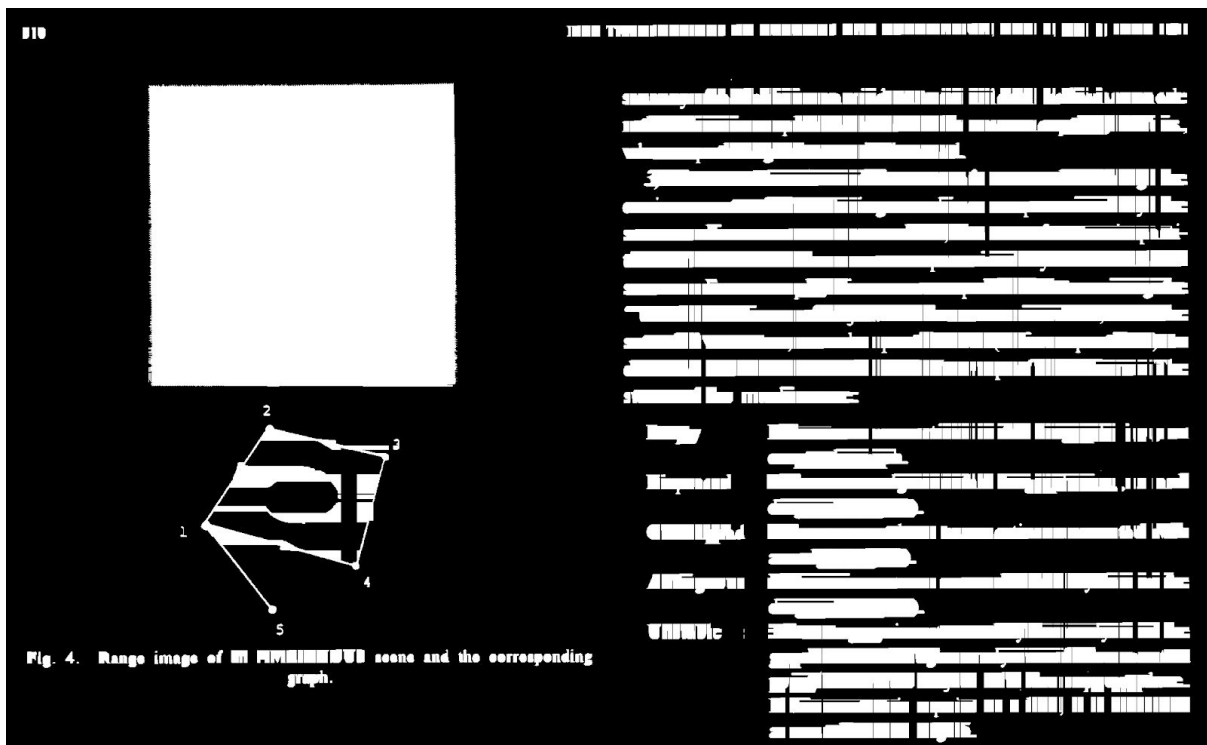
c) Dilatação com elemento estruturante (200,1)



d) Erosão com elemento estruturante (200,1)

Intersecção AND

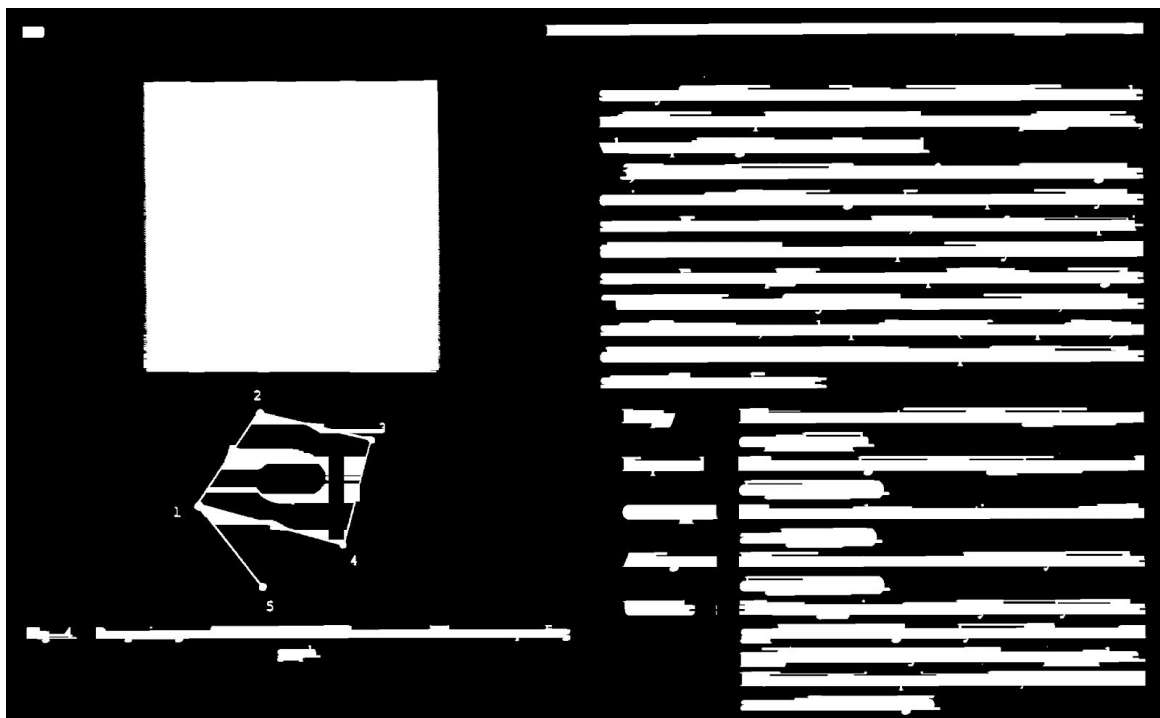
Os resultados exibidos acima são unidos por sua intersecção, realizando uma operação morfológica AND, bit a bit



e) Intersecção das imagens submetidas a dilatação-erosão

Fechamento

A imagem obtida da intersecção é submetida a operação de fechamento, para que as fendas e objetos pequenos da imagem processada sejam corrigidas e regiões de interesse na imagem sejam compactadas em objetos conexos em primeiro plano. O elemento estruturante utilizado é de tamanho (1,30).



f) Fechamento da imagem com elemento estruturante (1,30)

Componentes conexas e identificação de blocos de texto

As componentes conexas da imagem submetida ao fechamento são identificadas, cada uma correspondente a uma região de interesse. Estabelece-se um retângulo que melhor encobre a região para cada contorno encontrado, e o conteúdo compreendido pelo retângulo é analisado para a identificação de texto.

Dois parâmetros são calculados para cada componente: $R1$ = a densidade de pixels negros no retângulo; e $R2$ = a quantidade de transições de pixels brancos para pixels pretos dividida pela quantidade de pixels pretos totais. O valor das razões para cada componente foi analisado, e um intervalo estabelecido para separar regiões de texto foi $R1 \in [0,1, 0,7]$ e $R2 \in [0.03,0.3]$. Assim, regiões cujas razões estão dentro dos intervalos estabelecidos são consideradas texto, e um retângulo é desenhado em volta da região na imagem original.

7. Resultado final

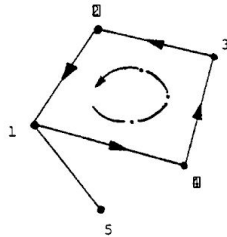
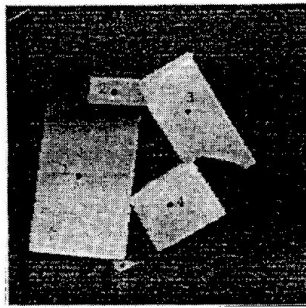


Fig. 4) Range image of an AMBIGUOUS scene and the corresponding graph

sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

Empty	If there are no vertices in the diagraph, i.e., an empty diagraph.
Dispersed	If there no edges in the diagraph, i.e., a null diagraph (Fig. 2).
Overlapped	If there are at least two vertices connected with an edge (Fig. 3).
Ambiguous	If there is one or more directed cycles in the diagraph (Fig. 4).
Unstable	This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point or a line, it is classified as unstable. See Fig. 5.

g) Resultado da segmentação de regiões com texto

O algoritmo destaca cada linha de texto da imagem com um retângulo envolvendo o bloco de texto. No final, foram encontrados 36 blocos de texto.