Trabalho 4

Levy Gurgel Chaves – RA: 264958

MO443/MC920 - Introdução ao Processamento de Imagem Digital
Universidade Estadual de Campinas
Prof. Hélio Pedrini

24 de Junho de 2020

1 Introdução

O objetivo deste trabalho é implementar técnicas fundamentais de processamento de imagens, onde conceitos importantes como operadores morfológicos são abordados de maneira prática.

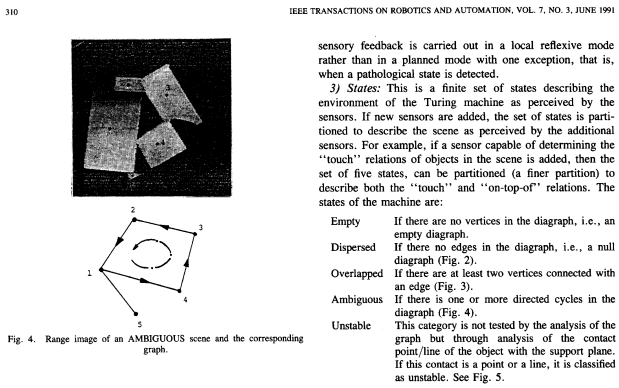
Juntamente com o relatório, está sendo enviado o arquivo LEVY_MO443T4_264958.zip, contendo todo o conteúdo apresentado neste relatório. A seção 2 comenta detalhadamente sobre a entrada e as saídas do programa, enquanto a seção 4 apresenta a explicação da solução dos problemas descritos no quarto trabalho, assim como o resultado dos experimentos realizados.

2 O programa

O programa foi desenvolvido em Python em sua versão 3.7.5 com o auxilio das seguintes bibliotecas nas suas respectivas versões: Numpy 1.18.2; Matplotlib 3.2.1; Scikit-image 0.16.2 e Scipy 1.4.1. O mesmo pode ser conferido com o auxílio do Jupyter notebook executando o arquivo *notebook-t4.ipynb*.

2.1 Entrada

O programa realiza a leitura de uma imagem binária a partir da pasta */images*. Por convenção, a Figura 1 representa a imagem de referência adotada para os experimentos.



310 IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 7, NO. 3, JUNE 1991

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.

Figura 1: Imagem de referência para experimentos

2.2 Saída

As imagens relativas aos resultados dos experimentos possuem formato *PNG* e são armazenadas na pasta */output*.

3 Definição do problema

Dada uma imagem binária como entrada, projetar um algoritmo que realiza a segmentação dos elementos contidos em texto e não-texto utilizando operadores morfológicos.

4 Soluções e decisões tomadas

4.1 Leitura das imagens

A leitura da imagem de referência foi realizada através da função **skimage.io.read** onde a mesma retorna um array Numpy possuindo dimensões MxNxC, onde MxN representam as dimensões da imagem lida e C indica a quantidade de diferentes canais.

4.2 Pré-processamento

Após a leitura da imagem referência, a função utilizada retorna uma matriz contendo apenas os valores 0 e 255 devido a natureza binária da imagem de referência. A imagem em questão possui fundo branco com conteúdo preto, contudo os operadores morfológicos esperam imagens brancas com fundo preto. Sendo assim, a função *cv2.bitwise_not* é aplicada à imagem de referência para ajustar o fundo e o conteúdo de acordo com os pre requisitos dos operadores morfológicos.

4.3 Dilatação e erosão 1 $\times$ 100

Como ponto de partida, vamos realizar a operação de fechamento (dilatação seguindo de erosão) com um elemento estruturante possuindo 1 pixel de altura e 100 pixels de largura com o negativo da imagem original. O resultado da operação de dilatação e erosão pode ser visualizado nas Figuras 2 e 3.

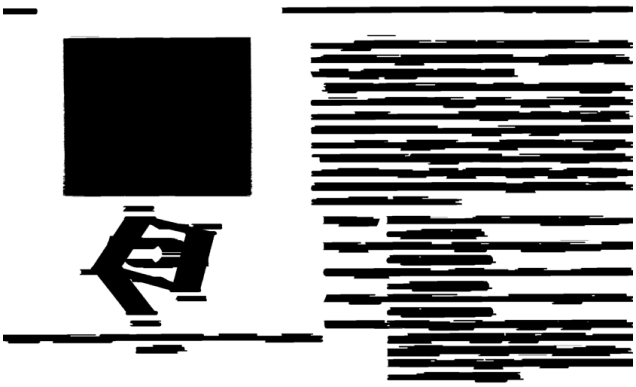


Figura 2: dilatação com elemento estruturante 1 × 100

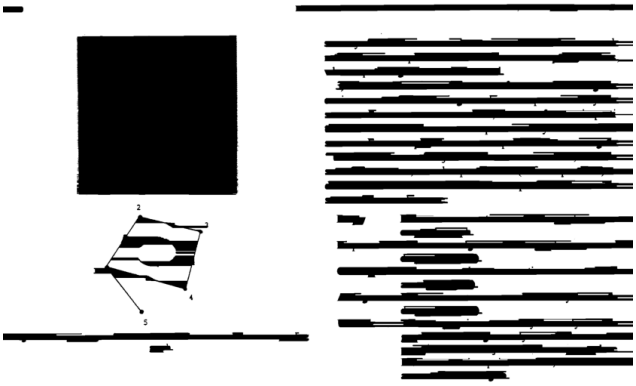


Figura 3: erosão da Figura 2 com elemento estruturante 1 × 100

4.4 Dilatação e erosão 200 × 1

Em seguida, realiza-se o mesmo procedimento descrito no subseção anterior, porém com um outro elemento estruturante retangular possuindo 200 pixels de largura e 1 pixel de altura. O resultado da dilatação e erosão, respectivamente, pode ser visualizado nas Figuras 4 e 5.

Figura 4: dilatação com elemento estruturante 200 × 1

Como é possível observar, a operação com elemento estruturante 1×100 preencheu horizontalmente espaços entre as palavras. Já com o elemento estruturante 200×1 aplicado, percebe-se que preencheu alguns espaços verticais entre palavras. Sendo assim, podemos

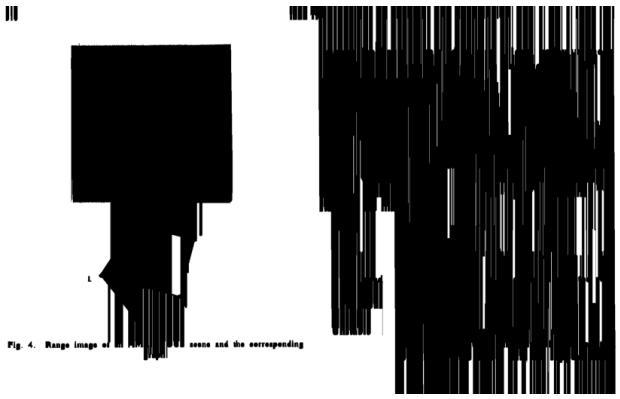


Figura 5: erosão da Figura 4 com elemento estruturante 200 × 1

pensar que os 2 passos anteriores estão tentando "esticar"horizontalmente e verticalmente a imagem dada como entrada. Portanto, podemos esperar que a intersecção entre esses dois resultados acentuem regiões de interesse onde, possivelmente, há grande possibilidade da existência de palavras nessas regiões.

4.5 Intersecção

Realizando a intersecção dos resultados finais obtidos através dos passos descritos nas subseções 4.3 e 4.4, obtemos o resultado mostrado na Figura 6.

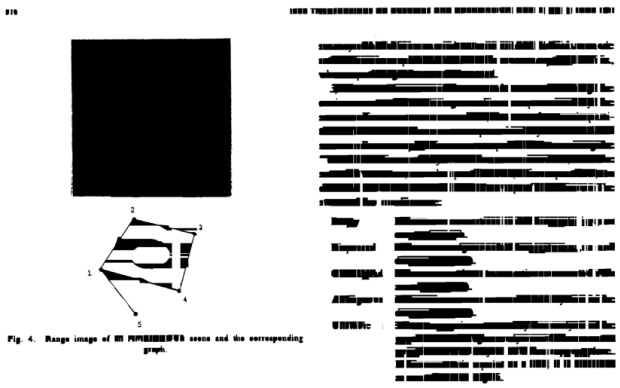


Figura 6: resultado da operação de intersecção

Assim como esperado, a operação de intersecção acentuou possíveis regiões de texto e não-texto. Sendo assim, temos uma primeira estimativa da localização das linhas e figuras da imagem.

4.6 Fechamento

Após a aplicação da operação de intersecção, percebe-se que algumas regiões ainda apresentam descontinuidades. Por exemplo, gostaríamos que uma linha inteira pertencesse a uma única componente conexa, formando assim um bloco de texto. Para remover essas espaços indesejáveis, aplica-se, novamente, a operação morfológica de fechamento com elemento estruturante de 1 pixel de altura e 30 pixels de largura. O resultado pode ser visto na Figura 7

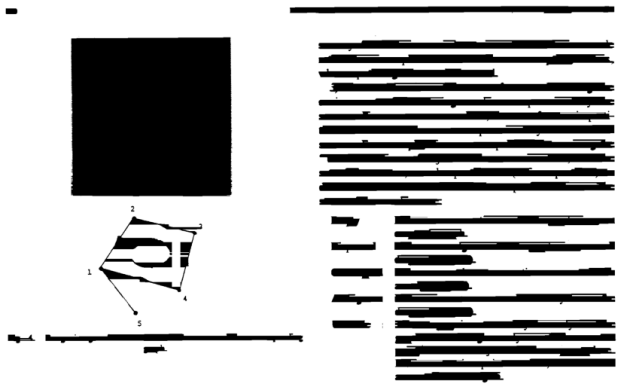


Figura 7: resultado do fechamento na imagem de intersecção

4.7 Componentes conexas e Blocos de texto

A partir do resultado obtido no passo anterior, observa-se que a imagem resultante agora apresenta blocos de cor preta bem definidos, agora sem as descontinuidades indesejadas em blocos de texto. Após o fim desse processamento inicial, podemos utilizar a função *connectedComponentsWithStats* do OpenCV para obter as coordenadas das componentes conexas mostras anteriormente.

Com as componentes obtidas a partir do algoritmo, deve-se definir quais destas contém texto ou não. Para tal, uma regra baseada em limiar foi adotada. Para cada componente conexa retornada pela função do OpenCV, duas razões foram calculadas:

- razão entre o numero de pixels pretos e o número total de pixels (RB)
- razão entre o número de transições verticais e horizontais branco para preto e o numero total de pixels pretos (RT)

De posse de cada razão para cada componente, pode-se então realizar a classificação entre texto e não-texto. Diversos valores de RB e RT foram testados com o intuito de achar a melhor combinação dessas variáveis. Por fim, para aceitar uma componente como bloco de texto, os valores que mais se adequaram à imagem de referência foram:

$$0.1 \leq RB \leq 0.5 \quad e \quad 0.1 \leq RT \leq 0.4$$

Os componentes selecionados tiverem o bouding box destacados na Figura 8, totalizando 39 linhas encontradas.

4.8 Segmentação de palavras

Com o objetivo de encontrar blocos de palavra, realizou-se processo semelhante ao usado para encontrar blocos de texto, todavia **este processamento foi aplicado apenas aos componentes classificados como bloco de texto**. Para cada componente encontrada na Figura 8, os seguintes passos foram realizados:

1. operação de *crop* na região em questão;
2. aplicação de duas operações de dilatação com elemento estruturante 5 pixels de altura e 10 pixels de largura;
3. aplicação de duas operações de dilatação com elemento estruturante 10 pixels de altura e 5 pixels de largura;
4. realiza a intersecção das saída dos passos (2) e (3);
5. obter os componentes conexos da imagem intersecção resultante.

Com a obtenção das componentes conexas de cada bloco de texto, não houve necessidade de aplicar nenhuma técnica de limiarização, visto que, para a imagem de referência, todos blocos de textos encontrados contém apenas palavras. Todavia isto pode não ser verdade para uma outra imagem de teste. Ao final deste processo, foram encontradas **245 palavras** – ver Figura 9.

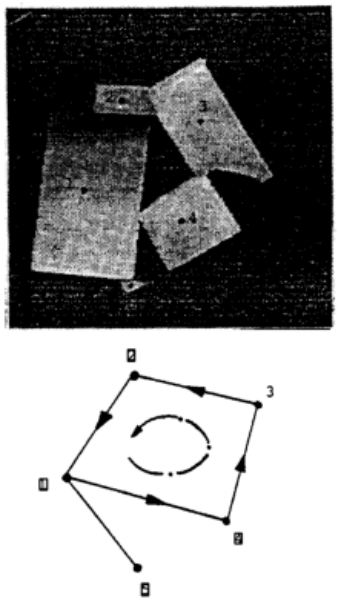


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

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.

Figura 8: componentes conexas classificadas como bloco de texto

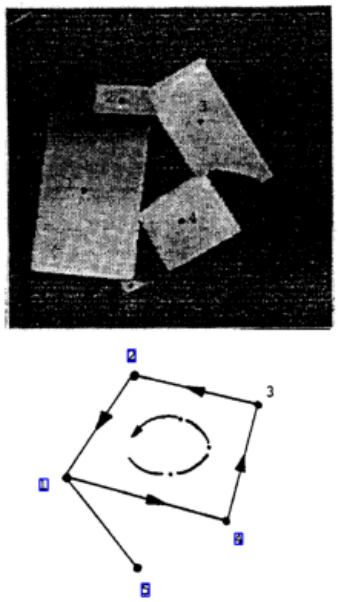


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

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.

Figura 9: blocos de palavra em destaque