

Trabalho 3

MC920 João Vitor Araki Gonçalves (176353)

Introdução

O objetivo desse trabalho é analisar a aplicação de operadores morfológicos para identificação de regiões de texto e não texto numa imagem, e posteriormente identificar as palavras individuais e calcular o número de palavras numa imagem.

O Programa

O programa foi implementado em python 3.7.3 utilizando as seguintes bibliotecas:

- numpy: Calculos vetorizados
- opencv2: Leitura e escrita da imagens.
- scikit-image: Aplicação de operadores morfológicos

Execução

Primeiramente instalar as dependencias do projeto:

```
pip install -r requirements.txt
```

O programa pode ser executado pela linha de comando:

```
python main.py
```

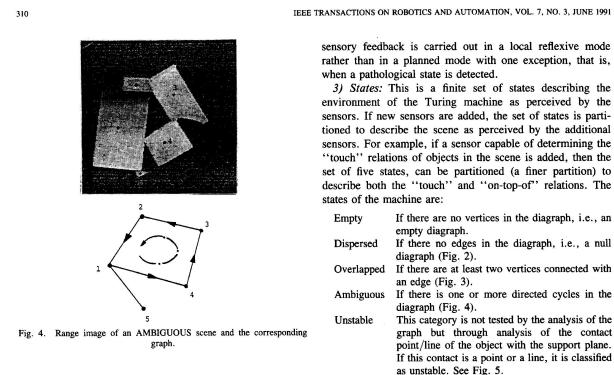
A imagem bitmap.pbm será utilizada como entrada para o programa.

Para cada uma das etapas do processo, será gravado uma imagem correspondente, no formato step1.pbm , step2.pbm , etc. As únicas exceções para essa regra é a etapa 7, que gera duas imagens step7.pbm e step7_original.pbm , e a etapa 8, que não possui imagem resultante.

Processo

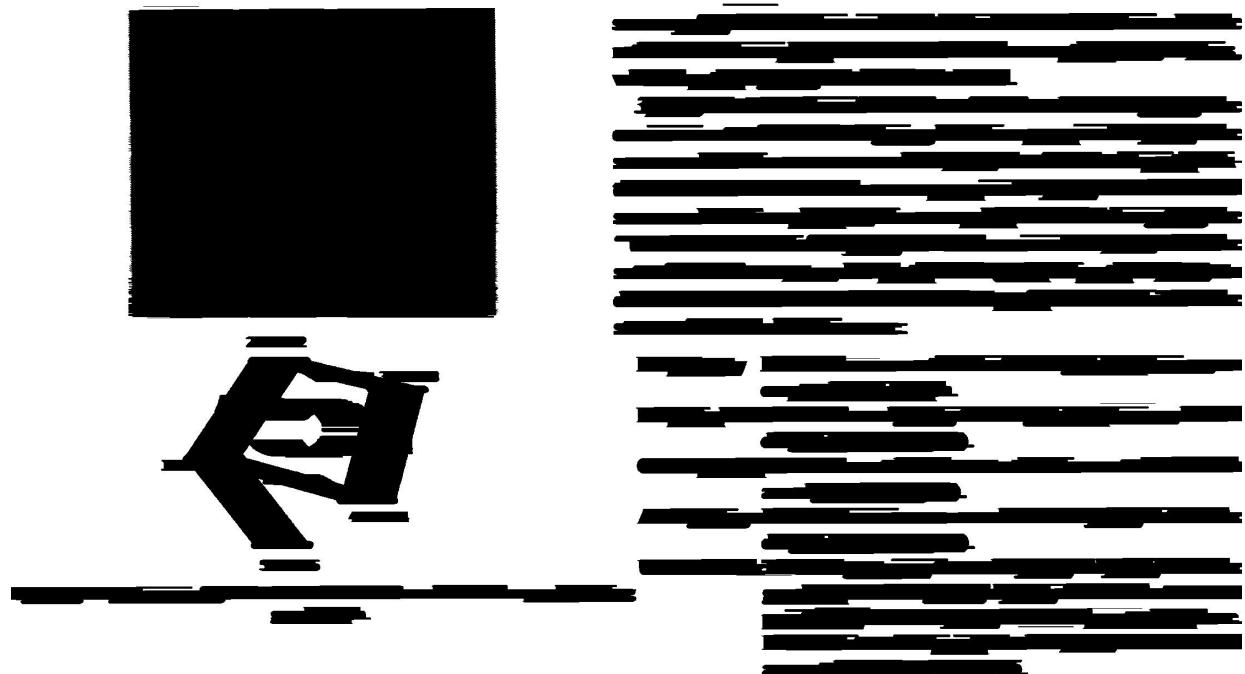
Leitura da Imagem

A imagem é lida pelo método cv2.imread(filename, 0) que lê a imagem e à guarda uma matriz numpy com valores de 0 à 255 para os níveis de cinza, e então a imagem é convertida para uma imagem binária (com valores booleanos como valor dos pixels) e pegamos o negativo da imagem, pois os elementos estruturantes utilizados esperam uma imagem branca num fundo preto (o contrário do que temos no caso de um texto).



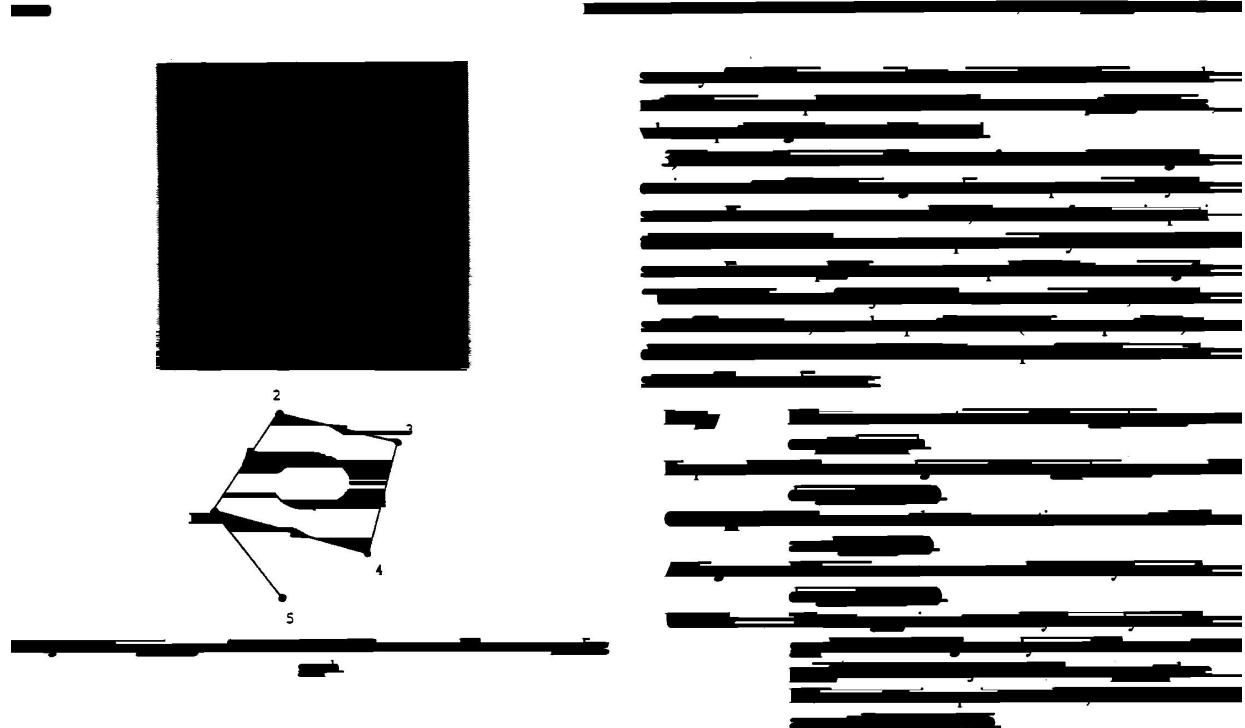
Passo 1: Dilatação 1x100

Na primeira etapa, aplicamos uma dilatação com um elemento estruturante de altura 1 e largura 100 composto do valor 1 no negativo da imagem original.



Passo 2: Erosão 1x100

Utilizando o mesmo elemento estruturante da primeira etapa, realizamos uma erosão na imagem resultante da dilatação.



Passo 3: Dilatação 200x1

Em seguida, realizamos o mesmo processo que as etapas 1 e 2, só que para um elemento estruturante de 200 de altura por 1 de largura. Primeiro aplicamos uma dilatação com esse elemento.



Passo 4: Erosão 200x1

Utilizando o mesmo elemento estruturante, realizamos uma erosão na imagem resultante da dilatação.

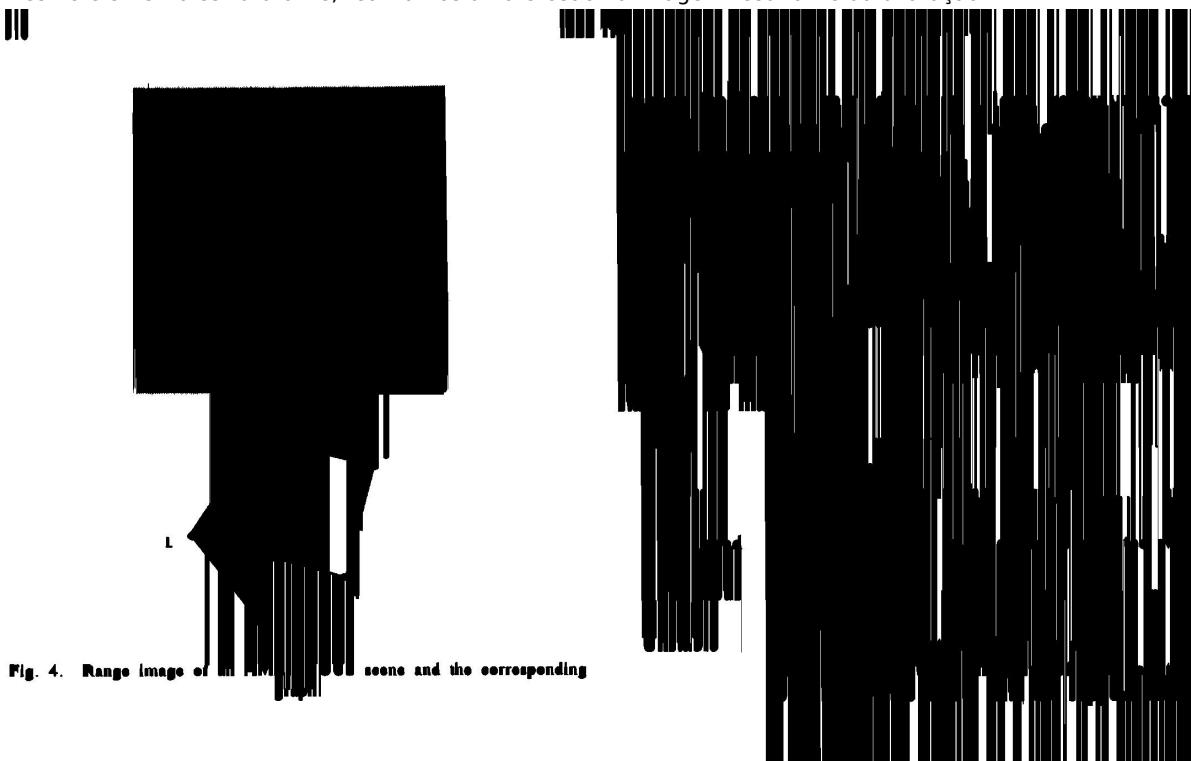


Fig. 4. Range image of a scene and the corresponding

Como é possível observar nas imagens, as operações com elemento estruturante 1x100 preencheu horizontalmente os vãos entre as palavras, e o com o elemento estruturante 200x1 preencheu verticalmente como esperado. Portanto esperamos que a intersecção das duas operações sejam blocos de texto onde palavras tinham uma distância menor e foram unidas durante o processo de dilatação e erosão, ou seja, blocos onde se encontravam linhas de texto.

Passo 5: Intersecção

Fazendo uma multiplicação dos dois resultados, temos o resultado de uma operação de and entre as duas, ou seja sua intersecção.

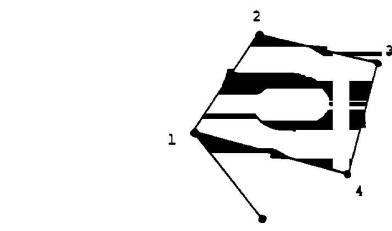
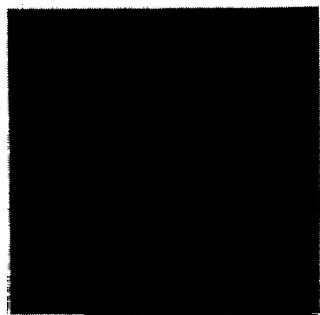
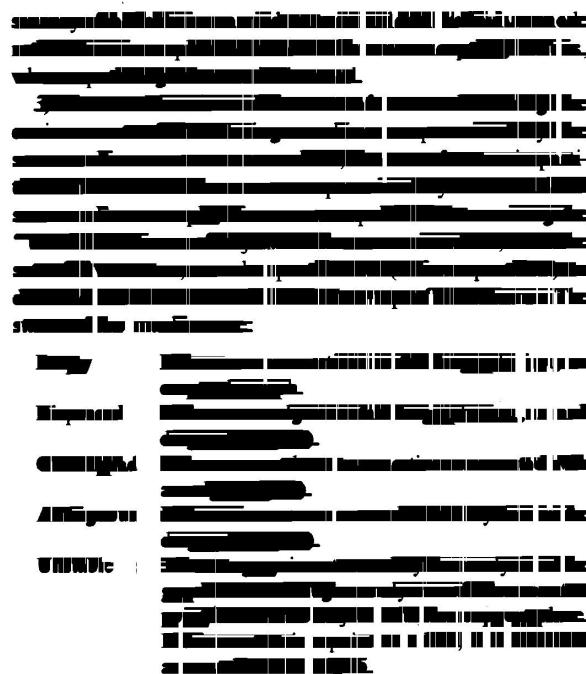
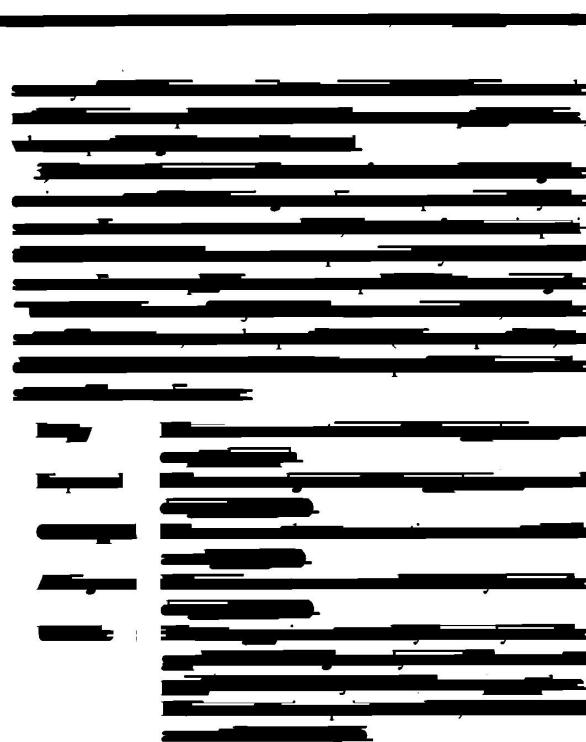
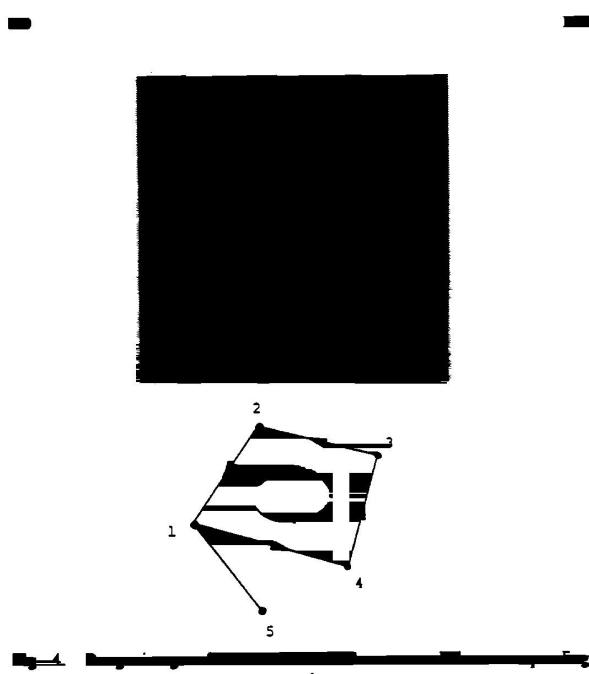


Fig. 4. Range image of **THE PENTAGRAM** scene and the corresponding graph.



Passo 6: Fechamento

No resultado dessa intersecção, ainda é possível identificar alguns blocos de palavras únicas e até mesmo algumas letras, como queremos identificar as linhas de texto aplicamos o fechamento com elemento estruturante de 1 de altura e 30 de largura]

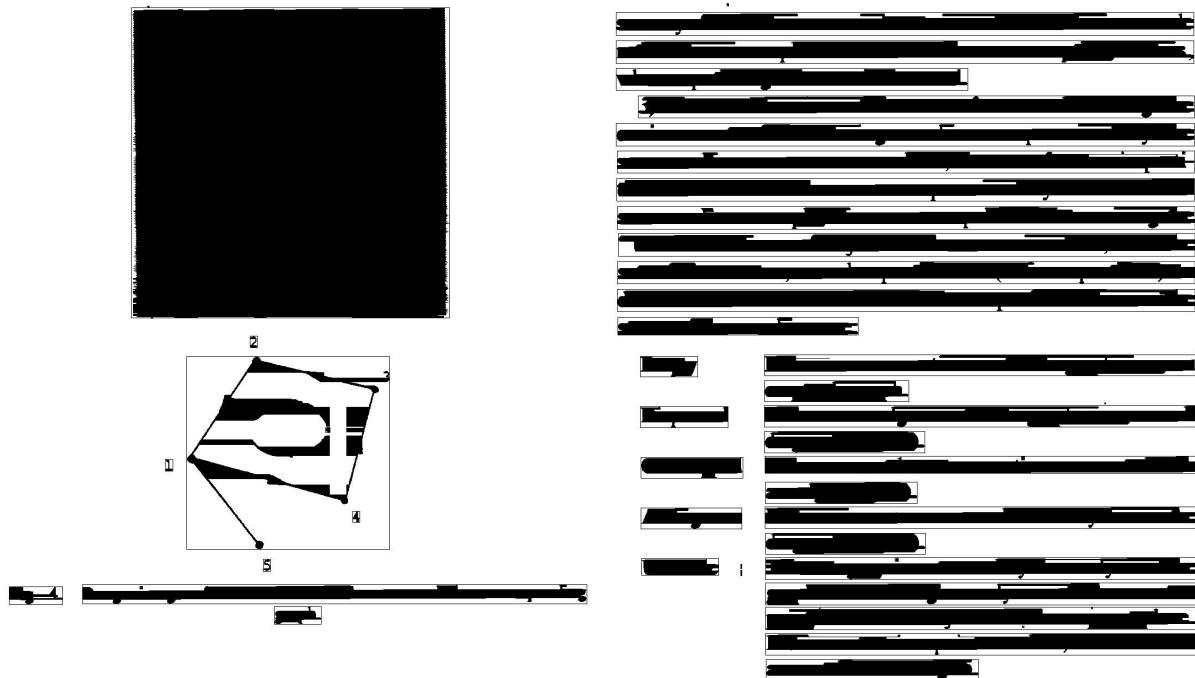


Após esse processo, temos componentes conexos na imagem bem definidos onde se encontravam as linhas de texto e o restante do conteúdo da imagem.

Passo 7: Identificando componentes conexos

Para encontrar as componentes conexos e seus bounding boxes, foi utilizado o programa em c fornecido `comp_conexos.c`, com uma modificação para ele retornar os bounding boxes dos componentes encontrados, e não apenas o número de componentes

Output do programa em c:



Aplicando os bounding boxes retornados sobre a imagem original

310

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

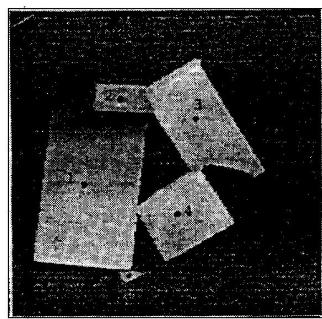


Fig. 4 Range image of an AMBIGUOUS scene and the corresponding graph.

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.

Passo 8: Calculando métricas

Agora que temos os elementos na imagem, precisamos determinar quais são blocos de texto e quais não são. Para isso vamos calcular algumas métricas sobre cada elemento.

1. A razão entre o número de pixels pretos e o número total de pixels (altura × largura)
2. A razao entre o número de transições verticais e horizontais branco para preto e o número total de pixels pretos

Essas métricas foram calculadas para todas as caixas encontradas no método anterior, realizando os cálculos sobre a imagem original.

Passo 9: Identificando blocos de texto

Após o cálculo das métricas, e análise de que valores cada bloco possuía, foi determinado que os blocos de texto possuem os valores:

Relação de pretos e brancos no intervalo [0.2, 0.5] (o que faz sentido, já que boa parte do texto é espaço em branco). Relação de transições horizontais e verticais de branco para preto no intervalo [0.03, 0.1]

Esses valores foram determinados analisando caixa por caixa, verificando seus valores e era um bloco de texto ou não.

Resultado de apenas blocos de texto marcados.

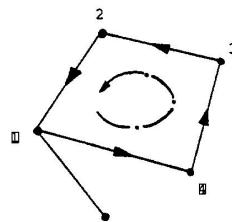
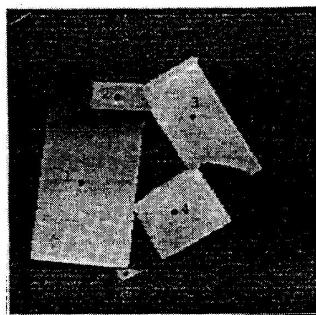


Fig. 4] Range image of an AMBIGUOUS scene and the corresponding graph.

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.

Passo 10: Encontrando palavras

Para identificar palavras, foi aplicado um processo similar ao de encontrar os blocos de texto, só que em escala menor, realizando o processo para cada pequeno bloco de texto por vez. A grande diferença entre essa etapa e a anterior é a escala, na anterior queríamos que cada bloco de texto se tornasse um componente conexo, agora queremos que cada palavra seja um componente conexo, portanto vamos utilizar elemento estruturantes bem menores e vamos pular a etapa do fechamento.

Os elementos estruturantes utilizados foram: um com 6 de altura e 10 de largura outro com 10 de altura e 5 de largura

Essas dimensões foram as que tiveram melhor resultado final, elementos menores encontrava-se o problema de pingos nos ís ficarem isolados, e portanto serem considerados componentes conexos independentes.

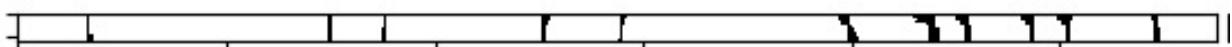
Para cada um deles elementos estruturantes, realizou-se a dilatação seguida da erosão, e posteriormente a operação de intersecção entre os resultados.

Então, utilizamos novamente o código fornecido para identificação de componentes conexos para encontrar os componentes do resultado desse processo, que esperamos que sejam as palavras individuais do texto.

Bloco de texto original

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

Resultado das dilatações e erosões



Resultado do identificador de componentes conexos



Então para cada bounding box dessa pequena região da imagem, calculamos as coordenadas relativas à imagem total, e temos o seguinte resultado:

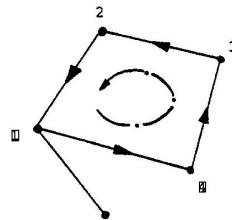
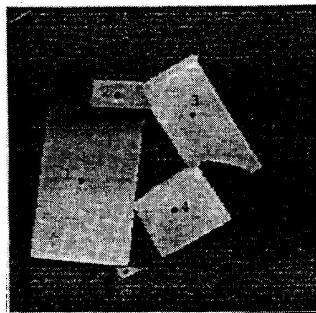


Fig. 4 Range image of an AMBIGUOUS scene and the corresponding graph.

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

Como podemos observar foi possível identificar as palavras individuais com uma boa precisão. Porém temos alguns pontos que o algoritmo falhou, que é o caso dos números próximos da figura do grafo na imagem, o algoritmo conseguiu identificar apenas 2 deles.

Finalmente o algoritmo identificou 240 palavras e 37 blocos de texto no total.

Foi feito o mesmo processo, só que utilizando a primeira página desse relatório como input, esse foi o resultado da identificação das palavras:

May 20, 2019

II Trabalho 3

MC920 João Vitor Araki Gonçalves (176353)

2.1 Introdução

O objetivo desse trabalho é analisar a aplicação de operadores morfológicos para identificação de regiões de texto e não texto numa imagem, e posteriormente identificar as palavras individuais e calcular o número de palavras numa imagem.

2.2 O Programa

O programa foi implementado em python 3.7.3 utilizando as seguintes bibliotecas: * numpy: Calculos vetoriais * opencv2: Leitura e escrita da imagens. * scikit-image: Aplicação de operadores morfológicos

2.3 Execução

Primeiramente instalar as dependencias do projeto:

```
pip install -r requirements.txt
O programa pode ser executado pela linha de comando
python main.py
A imagem bitmap.pbm será utilizada como entrada para o programa
Para cada uma das etapas do processo, será gravado uma imagem correspondente, no formato
step1.pbm, step2.pbm, etc. As únicas exceções para essa regras é a etapa 7, que gera duas imagens
step7.pbm e step7_original.pbm, e a etapa 8, que não possui imagem resultante.
```

1.3 Processo

1.3.1 Leitura da Imagem

A imagem é lida pelo método cv2.imread(filename,) que le a imagem e a guarda uma matriz numpy com valores de 0 à 255 para os níveis de cinza, e então a imagem é convertida para uma imagem binária (com valores booleanos como valor dos pixels) e pegamos o negativo da imagem, pois os elementos estruturantes utilizados esperam uma imagem branca num fundo preto (o contrário do que temos no caso de um texto).

1

Como é possível ver, ele falhou para identificar as palavras individualmente, encontrou apenas blocos de texto, e falhou para catalogar alguns blocos como texto. Esse é um problema desse método, ele possui grande sensibilidade à fonte utilizada e tamanho da mesma, talvez com um tuning maior dos parâmetros, e utilizando o tamanho dos componentes conexos para determinar o tamanho dos elementos estruturantes, seria possível obter um método mais flexível à outras imagens.