Trabalho 3 - MO443 - Introdução ao Processamento de Imagem Digital

Aluno: Gabriel Bianchin de Oliveira RA: 230217

Maio 2019

1 Introdução

O objetivo desse trabalho é implementar as técnicas de operadores morfológicos para segmentar regiões de texto e não texto de uma imagem de entrada.

Os operadores morfológicos utilizados foram dilatação e erosão. Foram utilizadas funções de identificação de componentes conexos para classificar se um determinado componente é um texto ou é não texto, baseado na razão entre o número de pixels pretos nos componentes.

2 Código

O código foi implementado em Python 3.6.7, com as bibliotecas OpenCV 4.0.0 e NumPy 1.13.3.

2.1 Como Executar

Para a execução do código, deve-se executar o script codigo.py. O script recebe como argumento uma imagem de entrada no formato PBM e o prefixo das imagens de saída. Um exemplo de execução é mostrado a seguir:

python3 codigo.py imagens/bitmap.pbm saida

Ao final, serão salvas as imagens de segmentação de linhas e de palavras da imagem de entrada. Utilizando como exemplo a entrada acima, as imagens serão salvas como saida-linhas.pbm e saida-palavras.pbm.

2.2 Entrada

A entrada de dados consiste em uma imagem no formato PBM. A imagem utilizada para testes está na pasta imagens, e foi obtida pelo seguinte diretório: http://www.ic.unicamp.br/helio/imagens_morfologia/

2.3 Saída

A saída de dados consiste em duas imagens resultantes da aplicação dos operadores morfológicos na imagem informada como entrada. Uma imagem será a segmentação de linhas da imagem e a outra imagem será da segmentação de palavras da imagem. O formato das imagens de saída será PBM. As imagens serão salvas na pasta saida.

3 Operadores Morfológicos

Os operadores morfológicos podem ser utilizados em extração de componentes conexos, delimitação de fecho convexo, extração de bordas, afinamento de bordas, entre outras aplicações [1].

A análise de imagens por operadores morfológicos pode beneficiar diversas áreas do conhecimento, como medicina, biologia, microscopia, metalurgia, entre outras.

Os operadores morfológicos utilizam a teoria dos conjuntos para representam os objetos de uma imagem. Por convenção, os objetos possuem valores pretos (valor 1) e o fundo possuem valores brancos (valor 0) [1].

3.1 Dilatação

A dilatação de uma imagem utiliza a adição de Minkowski, para que o elemento estruturante se desloque pela imagem. Quando o elemento estruturante está posicionado sobre um pixel da imagem, todos os pixels marcados pelo elemento estruturante recebem o valor 1 (se tornam pixels pretos).

Uma ilustração da dilatação de uma imagem pode ser vista na Figura 1.

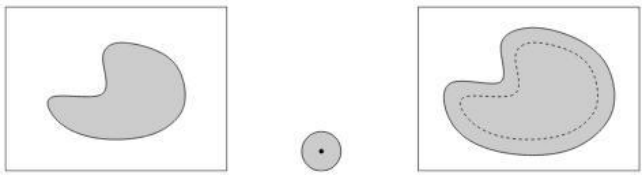

Figura 1: Imagem original (figura à esquerda) e elemento estruturante (figura central). Após a varredura do elemento estruturante na imagem original, ocorre a dilatação da imagem (figura à direita) [1]

3.2 Erosão

A erosão de uma imagem utiliza a subtração de Minkowski, para que o elemento estruturante se desloque pela imagem. Assim, pixels que não apresentarem o padrão do elemento estruturante não aparecerão na imagem.

Uma ilustração da erosão de uma imagem pode ser vista na Figura 2.

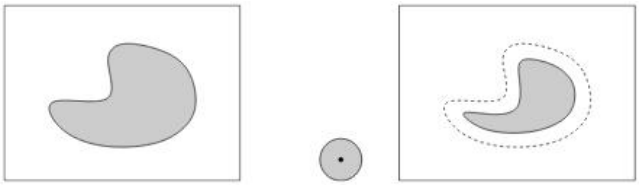

Figura 2: Imagem original (figura à esquerda) e elemento estruturante (figura central). Após a varredura do elemento estruturante na imagem original, ocorre a erosão da imagem (figura à direita) [1]

3.3 Fechamento

O fechamento de uma imagem corresponde a dilatação seguida de uma erosão. O fechamento é utilizado para a separação estreita de objetos, eliminar pequenos buracos e preencher lacunas no contorno [1].

4 Resultados

4.1 Leitura de Dados

A leitura da imagem de entrada é feita pela função **cv2.imread**, da biblioteca OpenCV[1], que carrega a imagem em níveis de cinza e transforma em uma matriz de dimensões m e n, onde m representa as linhas e n representa as colunas da matriz.

Como a leitura de imagem pelo OpenCV assume que o *background* é preto e o *forground* é branco, foi utilizada a função **cv2.bitwise_not** para que os textos e as imagens passem a ser *forground*, ou seja, branco, e o *background* seja preto.

Para exemplificar os resultados obtidos, será utilizada a imagem bitmap.pbm. A imagem original é mostrada na Figura 3. A Figura 4 mostra a aplicação da função **cv2.bitwise_not** citada anteriormente.

4.2 Segmentação de Linhas da Imagem

4.2.1 Aplicação do Operador Fechamento

Inicialmente, foi aplicado um fechamento (dilatação seguida de erosão) da Figura 4 com um elemento estruturante de 1 pixel de altura e 100 pixels de largura, como mostra a Figura 5. Para isso, foi utilizada a função **cv2.morphologyEx** com parâmetro **cv2.MORPH_CLOSE**, indicando o fechamento da imagem. Com isso, espaços pretos entre pixels brancos no sentido horizontal foram fechados.

Em seguida, foi aplicado o fechamento da Figura 4 com outro elemento estruturante, com 200 pixels de altura e 1 pixel de largura. O resultado é

[1] https://docs.opencv.org/3.0-beta/index.html

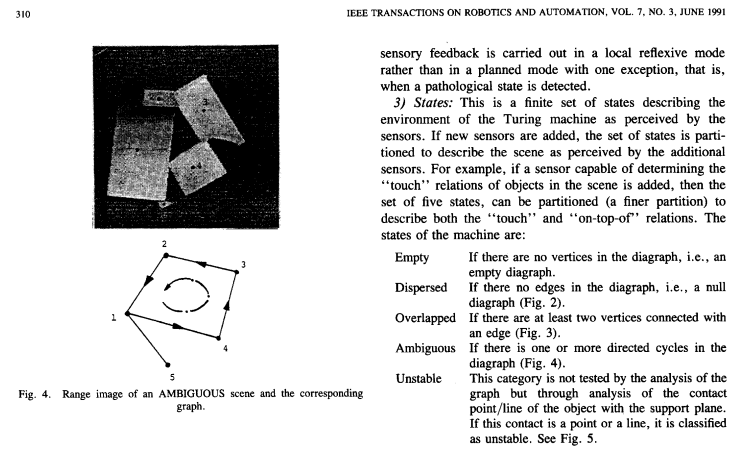

Figura 3: Imagem original

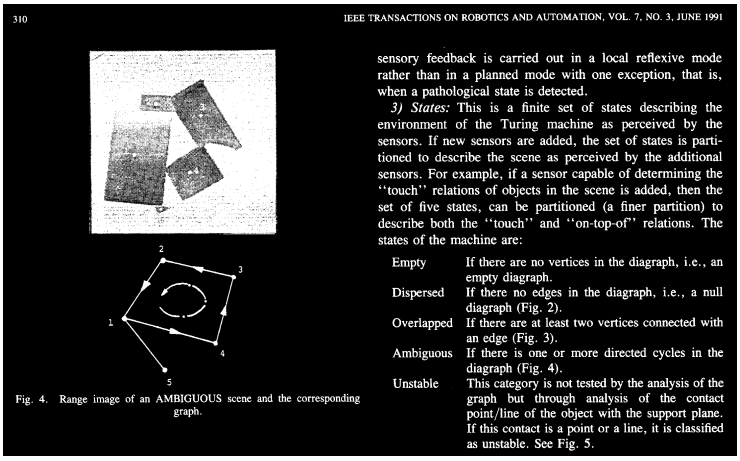

Figura 4: Imagem após a função **cv2.bitwise_not**

mostrado na Figura 6. Com isso, espaços pretos entre pixels brancos no sentido vertical foram fechados.

Após a realização do fechamento da imagem com dois elementos estruturantes gerando duas imagens resultantes, foi realizada a operação AND entre as Figuras 5 e 6, gerando a Figura 7.

Com a aplicação AND realizada, foi feito o fechamento do Figura 7 com o elemento estruturante de 1 pixel de altura e 30 pixels de largura, conforme mostra a Figura 8.

Figura 5: Fechamento da Figura 4 com elemento estruturante de 1 pixel de altura e 100 pixels de largura

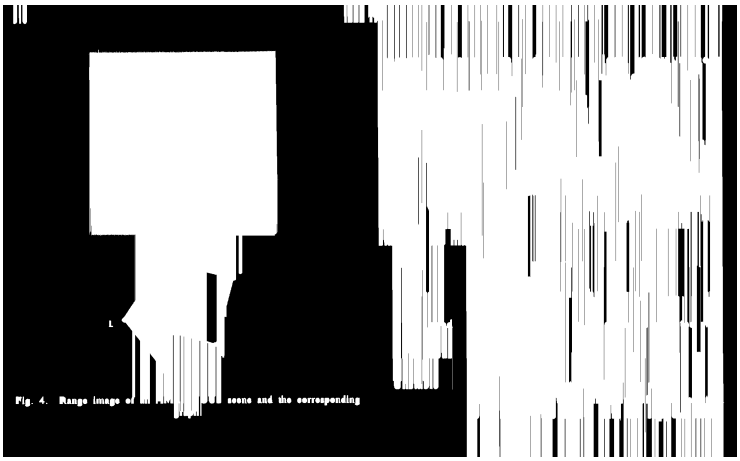

Figura 6: Fechamento da Figura 4 com elemento estruturante de 200 pixels de altura e 1 pixel de largura

4.2.2 Identificação de Componentes Conexos

Após a aplicação de operadores morfológicos na imagem original, foi feita a identificação de componentes conexos da imagem resultante (Figura 8). Para isso, foi utilizada a função **cv2.connectedComponentsWithStats**. Para melhor visualização dos retângulos envolventes, utilizou-se a imagem original (Figura 4). Ao todo, foram encontrados 54 componentes conexos na imagem original. A Figura 9 mostra os componentes conexos da imagem.

Dentre os componentes encontrados, alguns componentes não representavam

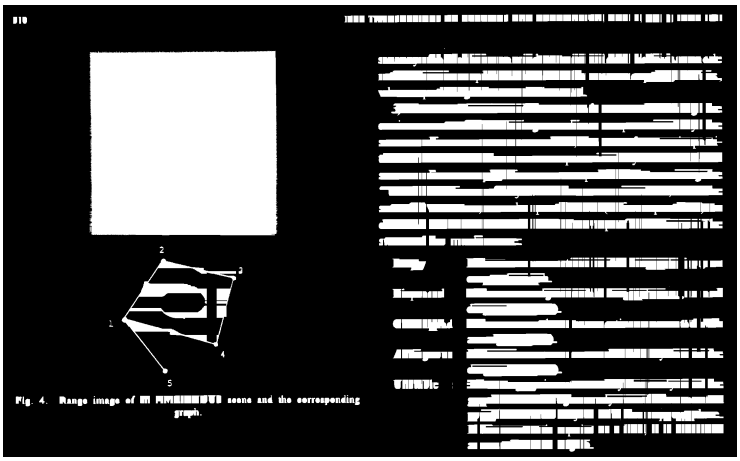

Figura 7: Resultado do operador AND nas Figuras 5 e 6

Figura 8: Fechamento da Figura 7 com um elemento estruturante de 1 pixel de altura e 30 pixels de largura

necessariamente texto, já que figuras e desenhos foram considerados componentes conexos. Portanto, foi necessário a classificação de textos e não textos para ser possível contabilizar as linhas de texto.

4.2.3 Classificação de Textos e Não Textos

Para realizar a classificação de textos e não textos, foi feito o cálculo de pixels pretos em cada componente, além da razão do número de pixels pretos pelo tamanho do componente e a razão das transições de pixels brancos para pixels pretos pelo número de pixels pretos. Para calcular a transição de pixels brancos

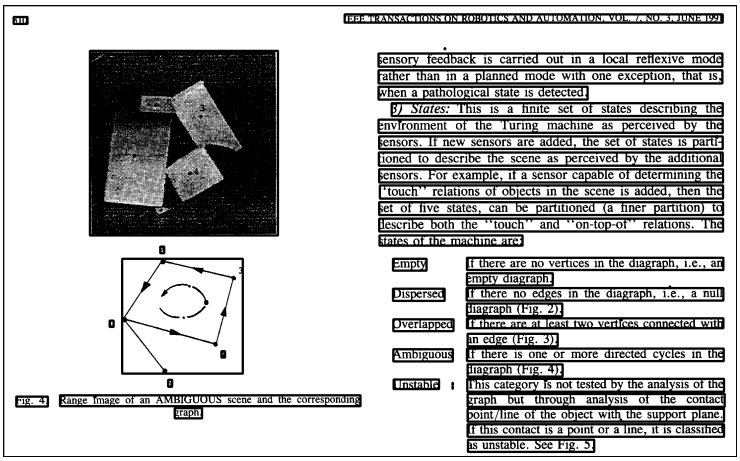

Figura 9: Componentes conexos encontrados na imagem

para pixels pretos, foi utiliza a vizinhança-4. A Figura 10 mostra o resultado obtido.

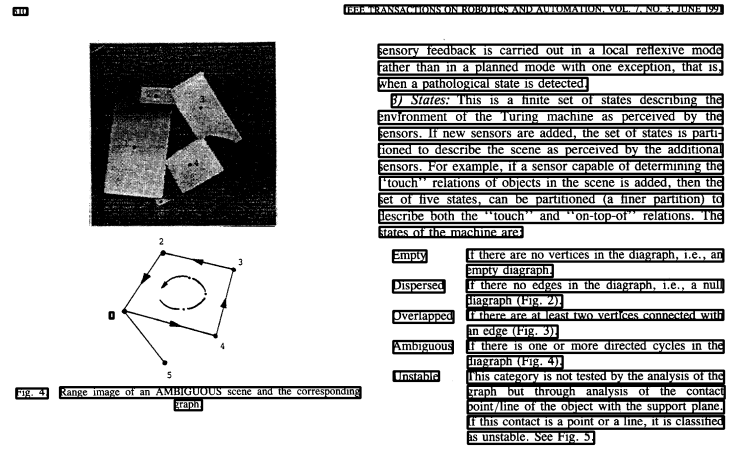

Figura 10: Segmentação de texto e não texto por linhas

Para diferenciar o texto de não texto, foram utilizados como parâmetros a razão de pixels pretos pelo tamanho do componente (valores abaixo de 0.4), quantidade de pixels pretos (valores acima de 0) e a razão de transições e pixels pretos (valores acima de 0.4). Na Figura 10, foram encontradas 38 linhas.

4.3 Segmentação de Palavras da Imagem

4.3.1 Aplicação do Operador Fechamento

Inicialmente, foi aplicado um fechamento (dilatação seguida de erosão) da Figura 4 com um elemento estruturante de 1 pixel de altura e 10 pixels de largura, como mostra a Figura 11. Com isso, espaços pretos entre pixels brancos no sentido horizontal foram fechados.

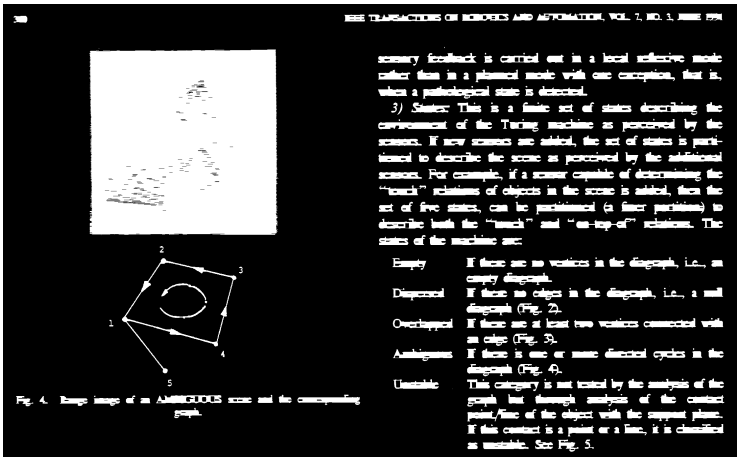

Figura 11: Fechamento da Figura 4 com elemento estruturante de 1 pixel de altura e 10 pixels de largura

Em seguida, foi aplicado o fechamento da Figura 4 com outro elemento estruturante, com 100 pixels de altura e 1 pixel de largura. O resultado é mostrado na Figura 12. Com isso, espaços pretos entre pixels brancos no sentido vertical foram fechados.

Após a realização do fechamento da imagem com dois elementos estruturantes gerando duas imagens resultantes, foi realizada a operação AND entre as Figuras 11 e 12, gerando a Figura 13.

Com a aplicação AND realizada, foi feito o fechamento do Figura 13 com o elemento estruturante de 1 pixel de altura e 10 pixels de largura, conforme mostra a Figura 14.

4.3.2 Identificação de Componentes Conexos

Após a aplicação de operadores morfológicos na imagem original, foi feita a identificação de componentes conexos da imagem resultante (Figura 14). Para isso, foi utilizada a função **cv2.connectedComponentsWithStats**. Para melhor visualização dos retângulos envolventes, utilizou-se a imagem original (Figura 4). Ao todo, foram encontrados 316 componentes conexos na imagem original. A Figura 15 mostra os componentes conexos da imagem.

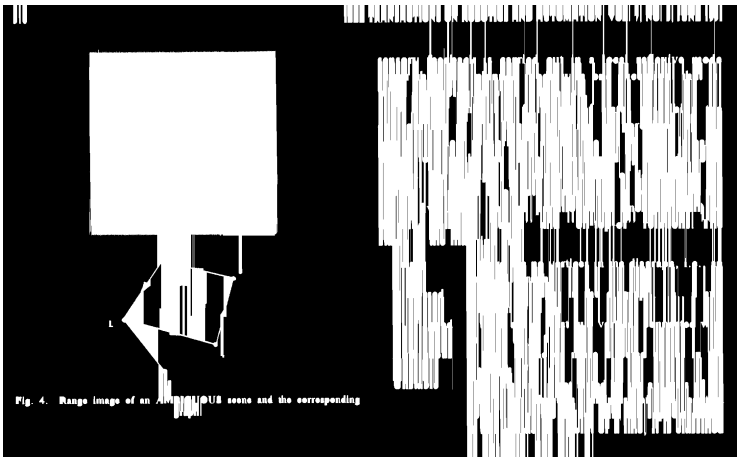

Figura 12: Fechamento da Figura 4 com elemento estruturante de 100 pixels de altura e 1 pixel de largura

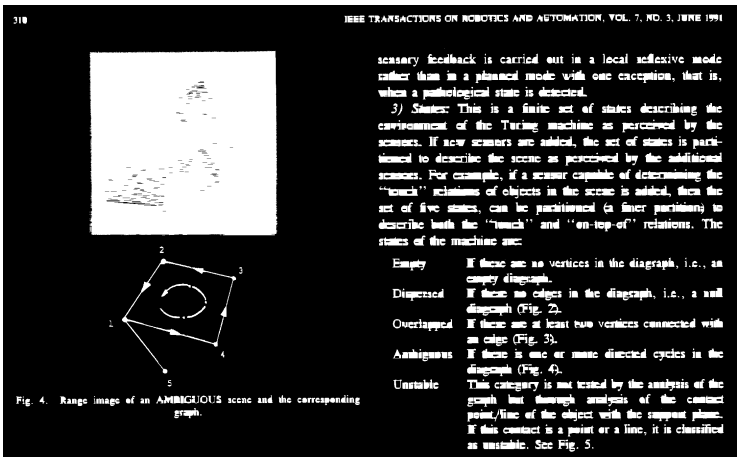

Figura 13: Resultado do operador AND nas Figuras 11 e 12

Dentre os componentes encontrados, alguns componentes não representavam necessariamente texto, já que figuras e desenhos foram considerados componentes conexos. Portanto, foi necessário a classificação de textos e não textos para ser possível contabilizar as palavras do texto.

4.3.3 Classificação de Textos e Não Textos

Para realizar a classificação de textos e não textos, foi feito o cálculo de pixels pretos em cada componente, além da razão do número de pixels pretos pelo tamanho do componente e a razão das transições de pixels brancos para pixels

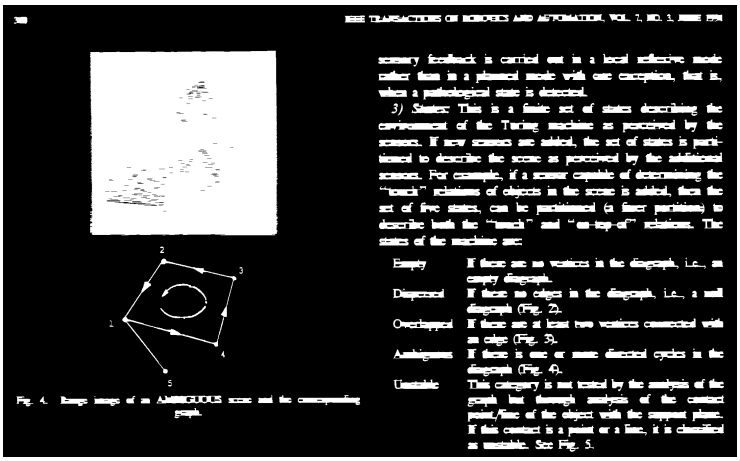

Figura 14: Fechamento da Figura 13 com um elemento estruturante de 1 pixel de altura e 10 pixels de largura

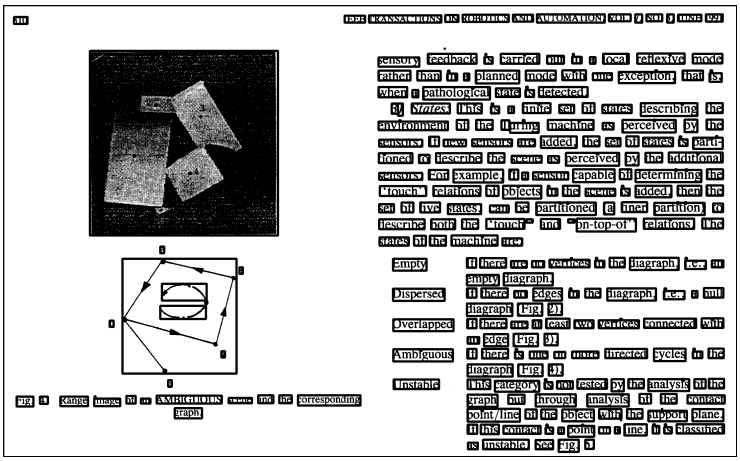

Figura 15: Componentes conexos encontrados na imagem

pretos pelo número de pixels pretos. Para calcular a transição de pixels brancos para pixels pretos, foi utiliza a vizinhança-4. A Figura 16 mostra o resultado obtido.

Para diferenciar o texto de não texto, foram utilizados como parâmetros a razão de pixels pretos pelo tamanho do componente (valores entre 0.15 e 0.6), quantidade de pixels pretos (valores acima de 0) e a razão de transições e pixels pretos (valores acima de 0.6). Na Figura 16, foram encontradas 271 palavras.

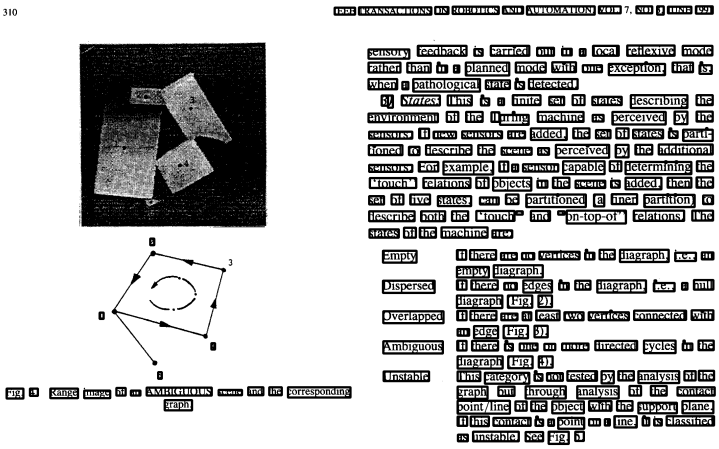

Figura 16: Segmentação de texto e não texto por palavras

5 Conclusão

Operadores morfológicos podem auxiliar em diversas tarefas. Dentre as possibilidades de operações com operadores morfológicos, a contagem de linhas e palavras de uma imagem apresenta um grande desafio, devido a dificuldade de separar textos de não textos. Esse trabalho apresentou um exemplo de aplicação de operadores morfológicos na contagem de linhas e palavras. Algumas dificuldades que apareceram foram na escolha de parâmetros para separar textos de não textos e na classificação de textos com diferentes fontes.

References

[1] Helio Pedrini http://www.ic.unicamp.br/ helio/. Morfologia matemática.