

Relatório do Trabalho

Alunos: Luis Henrique Santos Nascimento; João Victor Figueiredo Fernandes Santana; Rivaldo Leite Dos Santos Neto.

1) Parte Um:

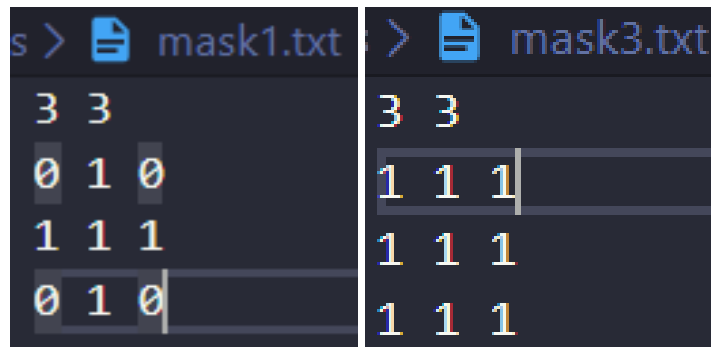
Para a primeira parte do trabalho da matéria, tivemos como tarefa fazer um programa que recebe como entrada uma imagem com um texto no formato PBM e tem a função de remover o ruído dessa imagem para criar outro arquivo PBM que possui o conteúdo da imagem de entrada sem o ruído.

Inicialmente pensamos em remover o ruído usando abertura para remover os ruídos externos e fechamento para tirar os pontinhos brancos das letras. Como os ruídos externos eram formados por apenas 1 pixel ou 2 formando uma linha na vertical ou horizontal foram utilizadas as máscaras: mask2.txt e a mask4.txt que são máscaras 3x3 com 2 pixels na horizontal e vertical respectivamente. Então fizemos 2 aberturas uma com cada uma dessas 2 máscaras, posteriormente usamos 2 fechamentos cada um com uma das mesmas máscaras usadas na abertura pra remover os pontos brancos nas letras, fechar os espaços em branco dentro das letras.

Como depois dessas operações algumas letras ficaram com uma fratura, resolvemos inverter as operações, primeiro os fechamentos para fechar os espaços em branco das letras e depois a abertura para remover o ruído externo. Essa inversão nas operações sanou o problema numa imagem que estávamos testando, porém ao usar essa mesma operação em outra imagem algumas inconsistências permaneceram em algumas letras.

Então resolvemos mudar de abordagem, usamos a mediana com a máscara mask3.txt de tamanho 3x3 com 9 pixels para remover o ruído. Como algumas bordas das letras ficaram mal formadas, com alguns espaços em branco entre os pixels, decidimos fazer uma dilatação com a máscara mask1.txt de tamanho 3x3 com formato de cruz para melhorar o formato das letras consertando as bordas, em seguida usamos 2 aberturas uma com a mask4.txt e outra com a mask2.txt para romper com as pequenas pontes entre as letras, tanto as horizontais quanto as verticais, porém algumas pontes restaram, mas essas são pontes que em média possuem 4 pixels e se usássemos uma nova abertura com uma máscara maior para rompê-las notamos que algumas letras seriam muito prejudicadas ficando com grandes fraturas, então decidimos manter essas pontes.

No processo várias outras tentativas foram feitas para ver qual seria a melhor combinação de operações e máscaras para tirar os ruídos e deixar as letras bem formadas e sem pontes, ao final percebemos que apenas precisávamos tirar o ruído sem se importar com o formato das letras ou se existia pontes entre elas então fizemos uma última alteração no código mantendo as operações de mediana e dilatação e removendo as duas operações de abertura.



mask1.txt

mask3.txt

rat accumsan nec. Phasellus arcu libero
an tincidunt eget dui eu condimentum. N
is lectus. Suspendisse potenti.

do vitae condimentum id, condimentum
massa, aliquet ut nisl quis, vestibulum b
ctetur, bibendum varius dui ornare. Ut i
d ante sit amet efficitur. In dapibus sapie
accumsan ligula quis est bibendum porttit
vel ipsum. Nunc feugiat, leo ut males

Imagem com ruído

**rat accumsan nec. Phasellus arcu libero
an tincidunt eget dui eu condimentum. N
is lectus. Suspendisse potenti.**

**do vitae condimentum id, condimentum
nassa, aliquet ut nisl quis, vestibulum b
ctetur, bibendum varius dui ornare. Ut i
d ante sit amet efficitur. In dapibus sapie
accumsan ligula quis est bibendum porttit
vel ipsum. Nunc feugiat, leo ut males**

Imagem de saída do programa.

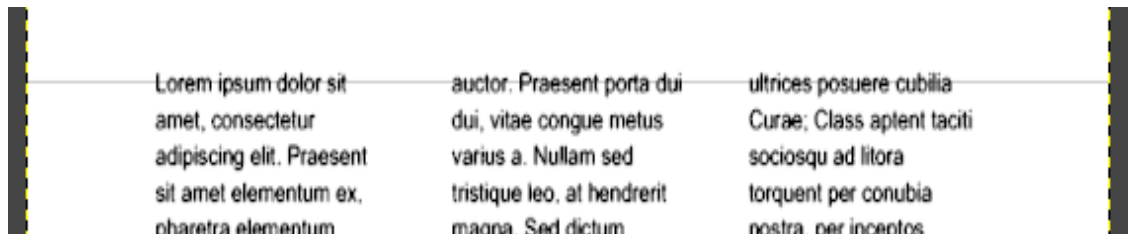
2) Parte Dois:

Para a parte dois do projeto tivemos como objetivo acrescentar no programa funções que contassem quantas colunas, linhas e palavras existiam na imagem, além de circular as palavras.

Começamos fazendo a função para contar as colunas, a ideia inicial foi posicionar uma lista no centro da primeira linha da imagem, sendo que essa lista possui tamanho menor do que o espaço que há entre uma coluna e outra, assim a lista inicia com 0 em todas as posições e conforme ela se movimenta horizontalmente e chega nas palavras onde existem pixels pretos, de valor 1, ela coloca dentro dela esses pixels sobre os quais “passa por cima” durante sua trajetória.

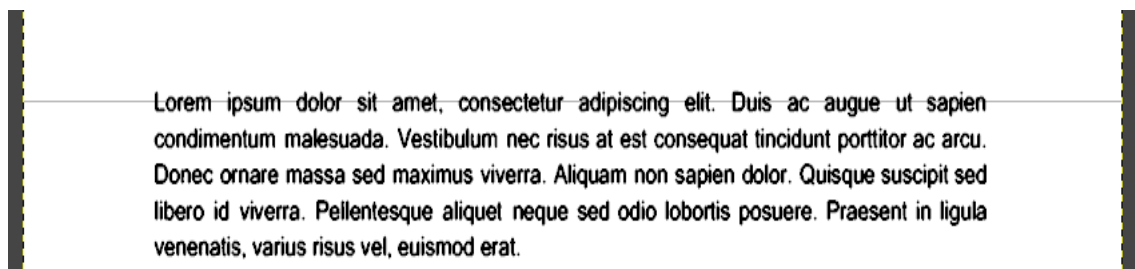
Durante sua trajetória, seu peso (a soma de todos os valores nela presentes) é verificado, caso o peso era 0 e foi alterado significa que o leitor (a lista) chegou a uma palavra, como ele é grande seu peso nunca volta a ser zero dentro de uma coluna. Porém ele é menor do que o espaço entre as colunas, assim, se o leitor tinha

peso maior que 0 e agora é 0 significa que ela acabou de sair de uma coluna, logo o valor de colunas é incrementado. Nós também aproveitamos a parte da função em que o leitor detecta uma coluna, seu peso era 0 e agora é maior que 0, para guardar o ponto onde as colunas começam para utilizar nas funções de contar linhas e palavras.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet elementum ex, rharetra elementum	auctor. Praesent porta dui dui, vitae congue metus varius a. Nullam sed tristique leo, at hendrerit maagna. Sed dictum	ultrices posuere cubilia Curae; Class aptent taciti sociosqu ad litora torquent per conubia nostra. per inceptos
---	--	--

Trajetória do leitor usado para contar colunas.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ac augue ut sapien condimentum malesuada. Vestibulum nec risus at est consequat tincidunt porttitor ac arcu. Donec ornare massa sed maximus viverra. Aliquam non sapien dolor. Quisque suscipit sed libero id viverra. Pellentesque aliquet neque sed odio lobortis posuere. Praesent in ligula venenatis, varius risus vel, euismod erat.
--

Trajetória do leitor usado para contar colunas.

Usando a mesma ideia usada para contar colunas, usando peso atual e peso anterior do leitor, fizemos a função de contar linhas. Nesse caso usamos um leitor menor, visto que o espaço entre as linhas é bem menor do que o espaço entre as colunas e esse se move na vertical.

Usamos os pontos conseguidos na função para contar colunas no momento de posicionar o leitor. O leitor era posicionado acima da primeira linha de cada coluna, uma coluna por vez, ao descer até o final do texto ele era reposicionado no início da próxima coluna para repetir a leitura nessa outra coluna. O objetivo era achar a coluna que possuía mais linhas. Um problema encontrado nessa função foi por conta dos parágrafos, que pulavam uma linha mais outra coluna poderia ter uma linha no lugar do parágrafo, assim o programa não considerava essa linha no número total de linhas da imagem.

Para resolver isso usamos uma *string* para verificar o quanto o leitor desceu sem ler nada, no caso dele detectar uma linha vazia do parágrafo, assim ao detectar a linha vazia o leitor pula horizontalmente para o ponto onde está a próxima coluna

para averiguar se nessa outra coluna há uma linha exatamente onde existe um espaço em branco na coluna anterior, se não houver linha ele pula horizontalmente para próxima coluna até que todas as colunas sejam verificadas naquele ponto. Depois da verificação ele retorna a posição onde ele estava (espaço em branco) e continua o processo, sendo que se na verificação alguma das colunas vizinha tinha uma linha naquele ponto ele agora considera aquela linha na conta de linhas totais.

<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet elementum ex, pharetra elementum tortor. Morbi ac feugiat nibh. Nulla ac lectus a sapien semper pulvinar. In facilisis nunc vel facilisis cursus. Mauris vulputate eget sapien eget volutpat. Suspendisse dui quam, vulputate nec magna vitae, placerat pharetra mauris. Suspendisse eu </p>	<p> auctor. Praesent porta dui dui, vitae congue metus varius a. Nullam sed tristique leo, at hendrerit magna. Sed dictum mattis ante pharetra fermentum. Aenean id bibendum nunc. Duis vestibulum nulla vel maximus viverra. Sed hendrerit cursus justo, a tincidunt libero pretium nec. Mauris ac justo vestibulum, ornare metus ac, tincidunt sem. Vivamus nec sapien </p>	<p> ultrices per Curae; Cl sociosqu torquent p nostra, pe himenaec sodales li vestibulur interdum imperdiet vehicula e Duis aliqu risus, vel consectetur ornare ris </p>
--	--	--

Imagem do teste para conferir a trajetória do leitor utilizado para contar linhas.

Continuando com a linha de ideia das funções anteriores montamos a função para contar palavras, porém dessa vez não utilizamos uma lista como leitor, mas sim uma matriz, pois precisaríamos da altura e da largura da palavra.

Da mesma forma que a função que conta as colunas guardava a posição das colunas, nós usamos a função de contar linhas para armazenar numa lista os pontos onde as linhas estavam para assim ter a posição central de cada linha de cada coluna e com isso posicionar o leitor exatamente em cima das palavras para percorrer cada uma das linhas das colunas.

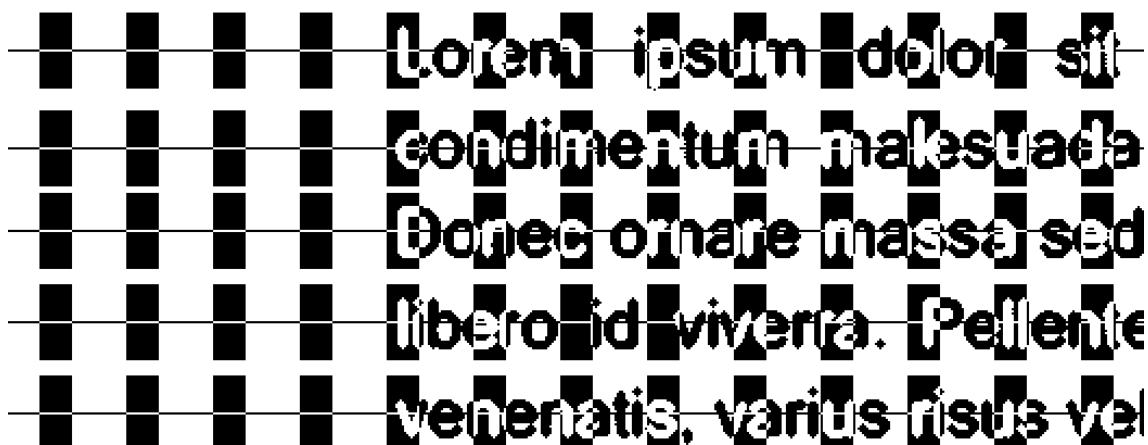
Assim, com o leitor corretamente posicionado ele percorre todas as linhas da primeira coluna e vai para a próxima coluna e repete o procedimento até verificar

todas as linhas. O processo de contar as palavras, assim como as funções anteriores se baseia no peso atual e anterior do leitor. Se era 0 e agora é maior que 0 o leitor chegou a uma palavra. Se antes era maior que 0 e agora é 0, o leitor acabou de sair da palavra, então o número de palavras é incrementado em 1. Com isso, nós também conseguimos os pontos x e y iniciais e finais das palavras que foram armazenados para serem usados pela função que circula as palavras.

Lorem ipsum dolor sit amet,
 condimentum malesuada. Vesti
 Donec ornare massa sed maxim
 libero id viverra. Pellentesque
 venenatis, varius risus vel, euism

 In id mauris sit amet lectus co
 porta dictum imperdiet, orci qua
 ..

Tentativa de posicionar o leitor para pegar as palavras de forma completa.

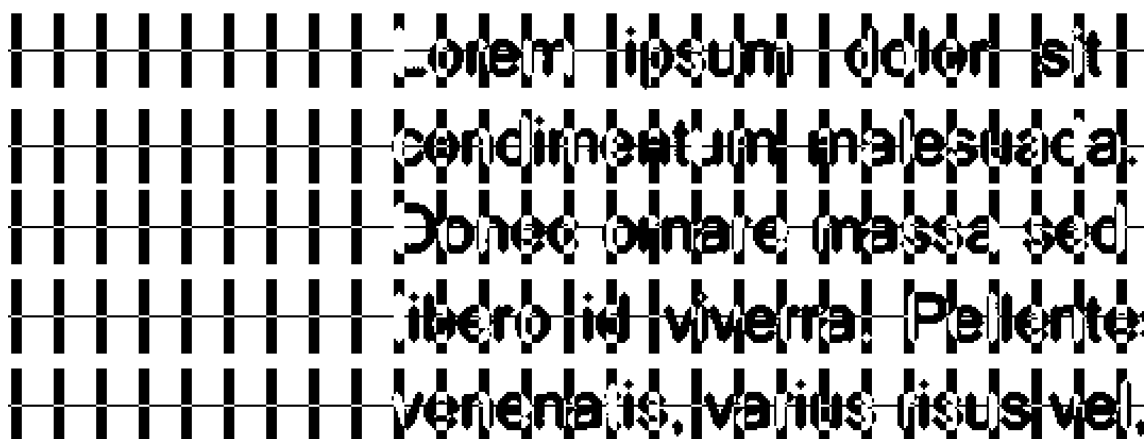


Lorem ipsum dolor sit
 condimentum malesuada
 Donec ornare massa sed
 libero id viverra. Pellente
 venenatis, varius risus vel




In id mauris sit amet lect
 porta dictum imperdiet, or

Utilizando os pontos centrais das linhas para posicionar o leitor.



Lorem ipsum dolor sit
 condimentum malesuada
 Donec ornare massa sed
 libero id viverra. Pellente
 venenatis, varius risus vel,



In id mauris sit amet lect
 porta dictum imperdiet, or

Leitor ajustado para caber nos espaços entre as palavras e fazer a contagem.

Por fim, fizemos a função que circula as palavras usando o conjunto de pontos x e y conseguidos pela função que conta as palavras.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ac augue
condimentum malesuada. Vestibulum nec risus at est consequat tincidunt portti
Donec ornare massa sed maximus viverra. Aliquam non sapien dolor. Quisque
libero id viverra. Pellentesque aliquet neque sed odio lobortis posuere. Praesi
venenatis, varius risus vel, euismod erat.

In id mauris sit amet lectus consectetur commodo eget at erat. Mauris vulpul
porta dictum imperdiet, orci quam congue leo, et faucibus erat enim sed diam. I
ullamcorper augue, et venenatis ligula. Sed tincidunt vehicula libero eu gra
condimentum nisl sed lacus vulputate tempus in in eros. Suspendisse enim nunc,
enim eget, pellentesque convallis massa. Cras non viverra est. Pellentesque dui
eget turpis ac, ornare euismod ligula. Morbi lacinia, leo in rutrum ultrices,
elementum ligula, vel pulvinar nulla nisl a nisi. In erat neque, varius ac volutpat vi
quam. Suspendisse nec orci sit amet quam pulvinar varius. Vestibulum tristique
pretium ultrices, lectus ipsum congue magna, ac scelerisque magna mi dapibu
luctus, sem eu gravida volutpat, purus felis tempus sapien, quis pulvinar elit elit e

Morbi diam libero, gravida at metus ac, hendrerit pharetra ligula. In vestibulum nu
viverra, a interdum lorem lacinia. Duis porttitor venenatis malesuada. Vestibuli
eros quam, eget mattis erat accumsan nec. Phasellus arcu libero, laoreet ut so

Imagem 01 Final.

Lorem ipsum dolor sit
amet, consectetur
adipiscing elit. Praesent
sit amet elementum ex,
pharetra elementum
tortor. Morbi ac feugiat
nibh. Nulla ac lectus a
sapien semper pulvinar.
In facilisis nunc vel
facilisis cursus. Mauris
vulputate eget sapien
eget volutpat.
Suspendisse dui quam,
vulputate nec magna
vitae, placerat pharetra
mauris. Suspendisse eu
magna sit amet sem
malesuada sodales
laoreet at purus. Nunc
fermentum volutpat

auctor. Praesent porta dui
dui, vitae congue metus
varius a. Nullam sed
tristique leo, at hendrerit
magna. Sed dictum
mattis ante pharetra
fermentum. Aenean id
bibendum nunc. Duis
vestibulum nulla vel
maximus viverra. Sed
hendrerit cursus justo, a
tincidunt libero pretium
nec. Mauris ac justo
vestibulum, ornare metus
ac, tincidunt sem.
Vivamus nec sapien
elementum, euismod
diam id, porttitor nulla.
Etiam id iaculis lacus,
lobortis luctus sem.

ultrices posuer
Curae. Class a
sociosqu ad lito
torquent per co
nostra, per ince
himenaeos. Ut
sodales ligula u
vestibulum. Pra
interdum ultrici
imperdiet finibu
vehicula eu.

Duis aliquet dig
risus, vel imper
consectetur vel
ornare risus ac
accumsan pulv
Aliquam ornare
condimentum t
porta. Vestibulu

Imagem 02 Final.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque pellentesque nec diam at bibendum. Sed pulvinar et dolor bibendum sollicitudin. Sed eu aliquet erat, sed ultrices justo. Donec quis mollis libero, sed porttitor nulla. Proin sed magna congue, accumsan purus non, vestibulum metus. Vestibulum purus libero, sollicitudin nec scelerisque ut, tristique vel augue. Ut condimentum velit nec quam condimentum, id mattis urna rhoncus. Fusce tincidunt dictum sagittis.

Etiam rutrum pretium erat, eu efficitur sem hendrerit vel. Nam egestas est eu nibh porttitor commodo. Quisque ac metus ut mauris ultricies bibendum. Sed semper cursus mattis. Sed semper mauris id odio pharetra, vel eleifend odio varius. In at tristique lacus, quis tincidunt lectus. Nam mattis ex et nisi rhoncus convallis. Suspendisse pulvinar lorem purus, eu aliquet velit pulvinar id. Mauris porttitor, arcu consequat mattis interdum, felis augue finibus risus, in feugiat turpis mauris at arcu. Vestibulum eleifend convallis pretium. Aenean quis mi nec leo facilisis rutrum in eu libero. Donec laoreet eu magna sed laoreet.

Integer egestas facilisis ipsum a aliquam. Morbi lacus nibh, tincidunt vel risus nec, gravida mattis tortor. Vestibulum consequat urna non ullamcorper faucibus. In libero libero, condimentum quis ultricies nec, semper eget nibh. Praesent vel cursus elit. Aliquam ut pharetra sapien. Praesent scelerisque efficitur ornare. Nunc vel egestas diam.

Integer ligula metus, fermentum ut iaculis

at. Donec eu tellus sit amet nunc finibus facilisis. Ut sit amet pretium ante, in bibendum velit. Pellentesque arcu risus, tempus ac libero eget, malesuada pretium nunc.

Cras eget magna vel sapien pretium semper eget rhoncus nisi. Ut tincidunt volutpat maximus. Nulla fringilla, metus eget accumsan aliquam, dui sapien egestas leo, pulvinar eleifend orci risus in est. Phasellus hendrerit, metus vitae consequat efficitur, felis urna iaculis quam, eu mattis augue nunc vel sem. Sed venenatis diam ac blandit ultrices. Nulla id lobortis nibh, sed dapibus neque. Quisque in egestas velit. Mauris efficitur id elit sit amet congue. Nullam ullamcorper suscipit odio, et faucibus diam malesuada elementum. Integer vestibulum velit magna, non iaculis nulla sollicitudin ac. Donec rhoncus, felis a lobortis ultricies, nisl nunc malesuada risus, eu molestie enim eros a est. Pellentesque luctus finibus nibh. Curabitur lorem est, congue non rutrum et, ornare eget quam.

Sed lacinia, magna eu placerat imperdiet, nulla mauris pulvinar ante, nec facilisis magna erat in ipsum. Fusce dictum porta consequat. Proin egestas odio id massa pellentesque, sed posuere odio vulputate. Curabitur ut sollicitudin nisl. Aliquam erat volutpat. Etiam ex nisi, porttitor at risus a, posuere dignissim eros. Maecenas eu cursus libero. Mauris vulputate gravida ullamcorper. Morbi accumsan eu turpis eget egestas.

Imagem 03 Final.