Limiarização Adaptativa com Imagem Integral

Felipe Adonai de Moraes

Sumário

- Objetivos
- Fundamentação Teórica
- Aplicações Práticas
- Algoritmo
- Exemplos
- Resultados

Objetivo

Implementar Técnica de Limiarização Adaptativa com Imagem Integral e comparar com outras técnicas de limiarização

- Linguagem de programação: Python

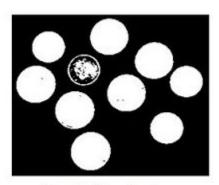
Fundamentação Teórica

Limiarização

- Classificar pixels em duas categorias: claros ou escuros com base em um limiar
- Problema com iluminação
- Método de Otsu



Input Image



threshold output

Limiarização adaptativa

- Forma de limiarização onde cada pixel possui um limiar diferente com base em características locais, como a média dos pixels vizinhos



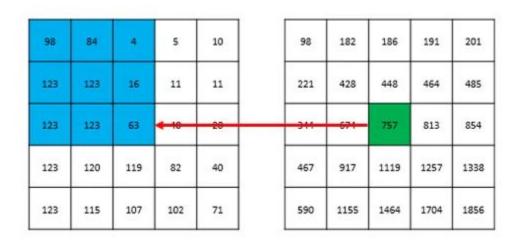




Imagem Integral (Tabela de Área Somada)

- Forma eficiente de calcular a área de uma parte de uma imagem

Custo linear



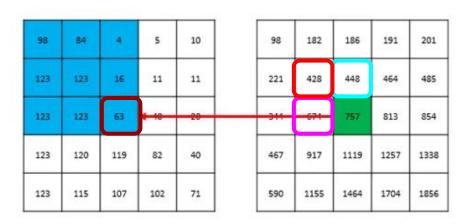
(left) An original grayscale image and (right) an integral image

Fonte: https://miro.medium.com/v2/resize:fit:640/format:webp/1*Bj5dt4XsTfwp8w81hal8yQ.png

Cálculo Imagem Integral

$$I(x,y) = f(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1).$$

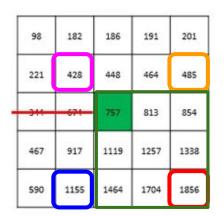
I é o valor da integral nos pontos (x,y) f é a intensidade nos pontos (x,y)

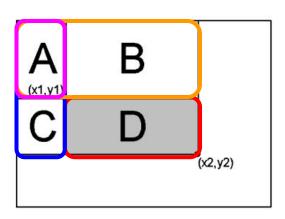


(left) An original grayscale image and (right) an integral image

Cálculo Área de um retângulo

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x,y) = I(x_2, y_2) - I(x_2, y_1 - 1) - I(x_1 - 1, y_2) + I(x_1 - 1, y_1 - 1).$$
D
B
C
A





$$(A+B+C+D)-(A+B)-(A+C)+A.$$

Limiarização Adaptativa com Imagem Integral

- Maior robustez quanto a mudanças fortes de luminosidade
- Algoritmo de performance real-time
 - limiarização normalmente faz parte de um processo maior
- Pontos positivos:
 - fácil de implementar; produz mesma saída independente da ordem de processamento dos pixels; competitivo real time; ordem de complexidade linear
- Pontos negativos:
 - usa uma iteração a mais na imagem para gerar a saída
 - Menos flexível para definir a forma da região de vizinhança, se limita a regiões retangulares.

Aplicações práticas

- Limiarização normalmente faz parte de um processo maior
- Detecção de bordas
- Real-time video streams
- Realidade aumentada
 - Detecção de marcadores



Algoritmo

- 1. Calcula integral da imagem
- 2. Efetua limiarização:
 - a. Parâmetros: Tamanho do bloco S; valor do limiar em porcentagem T
 - b. Para cada pixel, realiza a média dos valores dentro do bloco, e efetua a comparação para limiarização: se o valor do pixel for T por cento menor que a média, é preto, senão é branco

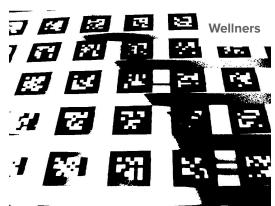
Calcula integral da imagem

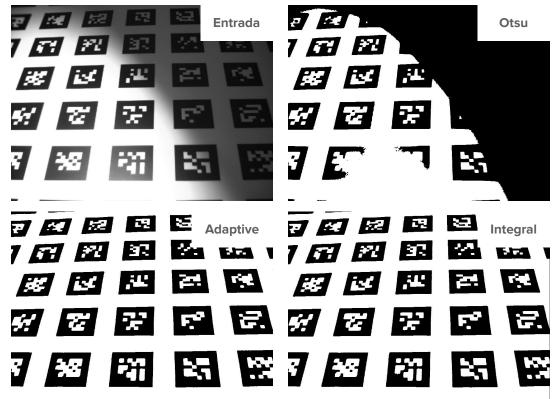
```
intImg = np.zeros((w, h))
 for i in range(w):
   sum = 0
   for j in range(h):
     sum += in_img[i, j]
     if i == 0:
       intImg[i, j] = sum
     else:
        intImg[i, j] = intImg[i-1, j] + sum
```

Efetua limiarização

```
out img = np.zeros((w, h))
  # Itera sobre cada pixel da imagem.
 for i in range(w - 1):
   for j in range(h - 1):
      # Define as coordenadas do bloco.
     x1 = max(0, i - block size // 2)
     x2 = min(w - 1, i + block size // 2)
     y1 = max(0, j - block size // 2)
     y2 = min(h - 1, j + block size // 2)
      # Calcula a soma dos pixels no bloco.
      block_sum = intImg[x2, y2] - intImg[x1, y2] - intImg[x2, y1] + intImg[x1, y1]
      block count = (x2 - x1) * (y2 - y1)
      # Compara o pixel com a média do bloco.
      if in img[i, j] * block count <= (block sum * (100 - threshold) / 100):</pre>
        out img[i, j] = 0
     else:
        out img[i, j] = 255
```

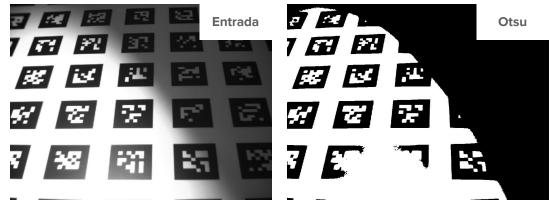
w = 713, h = 954 block_size = 89 threshold = 15

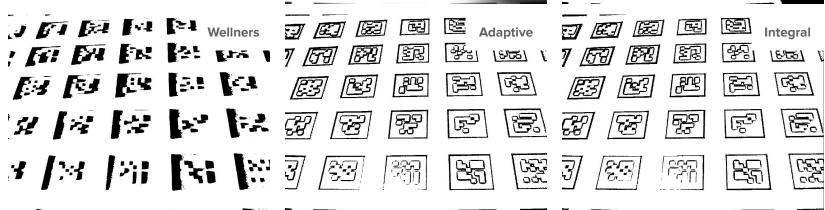




Algoritmo	Tempo	Num. Iterações	
Otsu	1.5153 s	1360659	
Wellners	0.6405 s	680202	
Medium	11.2971 s	4981626810	
Integral	3.0514 s	1358738	

w = 713, h = 954 block_size = 10 threshold = 15

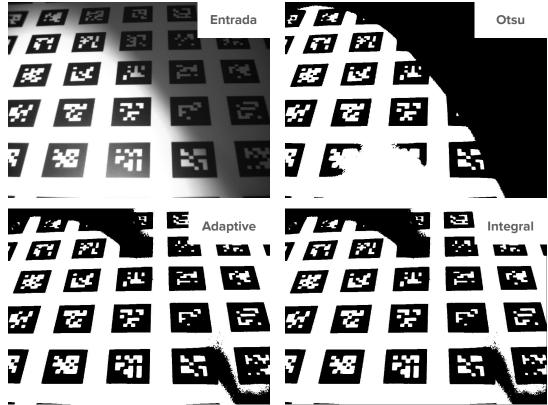




Algoritmo	Tempo	Num. Iterações	
Otsu	1.6077	1360659	
Wellners	0.6271	680202	
Medium	11.0732	4981626810	
Integral	3.0770	1358738	

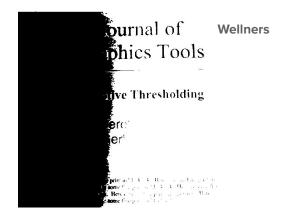
w = 713, h = 954 block_size = 500 threshold = 15

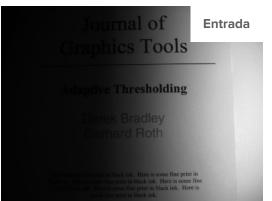


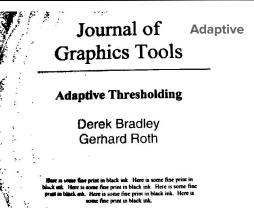


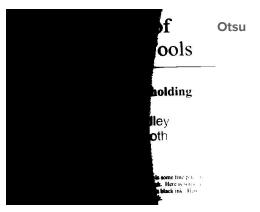
Algoritmo	Tempo	Num. Iterações	
Otsu	1.5840	1360659	
Wellners	0.6949	680202	
Medium	110.6432	121686617702	
Integral	3.3581	1358738	

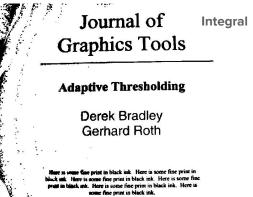
w = 624, h = 841 block_size = 78 threshold = 15





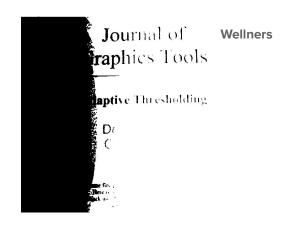


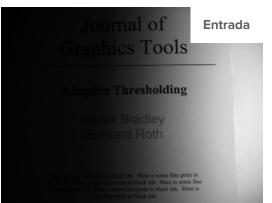




Algoritmo	Tempo	Num. Iterações	
Otsu	1.1603	1049823	
Wellners	0.4600	524784	
Medium	8.0946	3017483040	
Integral	2.4127	1048104	

w = 624, h = 841 block_size = 78 threshold = 50

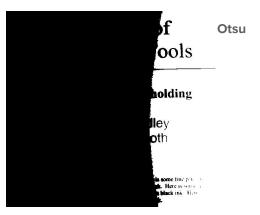




Journal of Adaptive
Graphics Tools

Adaptive Thresholding

de a stra pur melaktik Here samatta jaktim Program a pur melaktik Here a arabat a samat a pun melaktik Here a arabat a samata pun melaktik



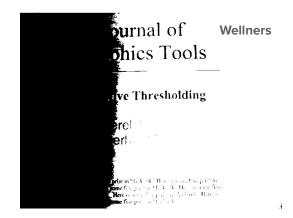
Journal of Integral
Graphics Tools

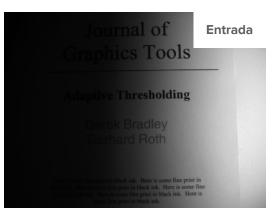
Adaptive Thresholding

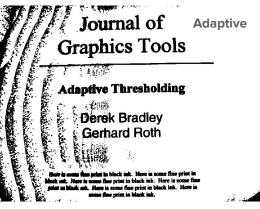
de se diseptir melok dik Hare sesanaha petenta.
 de se de se parametrisk dik Hares se de se de

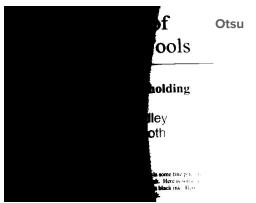
Algoritmo	Tempo	Num. Iterações	
Otsu	1.2193	1049823	
Wellners	0.5313	524784	
Medium	9.6168	3017483040	
Integral	2.6812	1048104	

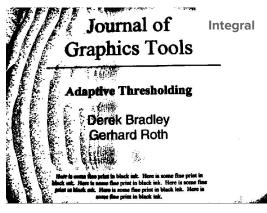
w = 624, h = 841 block_size = 78 threshold = 5







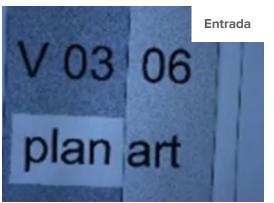


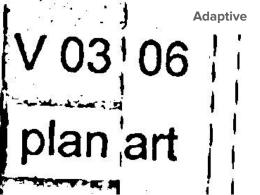


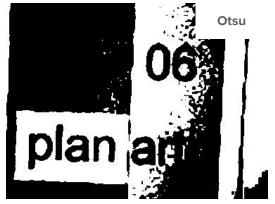
Algoritmo	Tempo	Num. Iterações	
Otsu	1.2652	1049823	
Wellners	0.4999	524784	
Medium	7.6402	3017483040	
Integral	2.4057	1048104	

w = 278, h = 420 block_size = 34 threshold = 15











Algoritmo	Tempo	Num. Iterações	
Otsu	0.3239	233775	
Wellners	0.1099	116760	
Medium	1.1802	127922964	
Integral	0.6074	232823	

w = 773, h = 1378block_size = 96 threshold = 15

Filtering in the Spatial Doma Wellners

e imodifying or enhancing an image. For example, you can liner an image to emphasize ** Satures. Image processing operations implemented with filtering include smoothing, s

operation, in which the value of any given pixel in the output image is determined by ues of the pixels in the neighborhood of the corresponding input pixel. A pixel's neight by their locations relative to that pixel. (SeeNeighborhood or Block Processing: An C **Geighborho**od operations.) Linear filtering is filtering in which the value of an output pixes of the pixels in the input pixel's neighborhood.

one is accomplished through an operation called convolution. Convolution is a neighbor noutput pixel is the weighted sum of neighboring input pixels. The matrix of weights is ca A known as the filter. A convolution kernel is a correlation kernel that has been rotated 13

a ne image is

mage Filtering in the Spatial Doma

Entrada

ing or enhancing an image. For example, you can liner an image to emphasize res. Image processing operations implemented with filtering include smoothing,

ion, in which the value of any given pixel in the output image is determined b s of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood neir locations relative to that pixel. (SeeNeighborhood or Block Processing: An O hood operations.) Linear filtering is filtering in which the value of an output pix es of the pixels in the input pixel's neighborhood.

age is accomplished through an operation called convolution. Convolution is a neighbo output pixel is the weighted sum of neighboring input pixels. The matrix of weights is ca iso known as the filter. A convolution kernel is a correlation kernel that has been rotated 1

Is Image Filtering in the Spatial Don Adaptive

a technique for modifying or enhancing an image. For example, you can inter an image to emphasiz remove other features. Image processing operations implemented with filtering include smoothing, nhancement.

a neighborhood operation, in which the value of any given pixel in the output image is determined b Ithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighi pixels, defined by their locations relative to that pixel. (SeeNeighborhood or Block Processing: An (scussion of neighborhood operations.) Linear filtering is filtering in which the value of an output pix of the values of the pixels in the input pixel's neighborhood.

ng of an image is accomplished through an operation called convolution. Convolution is a neighbo which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is ca temel, also known as the filter. A convolution kernel is a correlation kernel that has been rotated 18

suppose the image is

- 24 1 8 15
- 5 7 14 16
- 6 13 20 22
- 12 19 21 3

Is Image Filtering in the Spatial Doma

s neighborhood.

Doma

Otsu

Integral

or example, you can liner arrimage to emphasiz

ns implemented with filtering include smoothing.

of any given pixel in the output image is determined by

orhood of the corresponding input pixel. A pixel's neigh

that pixel. (SeeNeighborhood or Block Processing: An (

er filtering is filtering in which the value of an output pix

operation called convolution. Convolution is a neighbor

of neighboring input pixels. The matrix of we gots is a

kernel is a correlation kernel that has been rotated it

a technique for modifying or enhancing an image. For example, you can liner an image to emphasiz remove other features. Image processing operations implemented with filtering include smoothing, inhancement.

a neighborhood operation, in which the value of any given pixel in the output image is determined b ithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neigh pixels, defined by their locations relative to that pixel. (SeeNeighborhood or Block Processing: An C iscussion of neighborhood operations.) Linear filtering is filtering in which the value of an output pix of the values of the pixels in the input pixel's neighborhood.

ng of an image is accomplished through an operation called convolution. Convolution is a neighbo which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is ca kernel, also known as the filter. A convolution kernel is a correlation kernel that has been rotated 18

suppose the image is

- 24 1 8 15 5 7 14 16
- 6 13 20 22

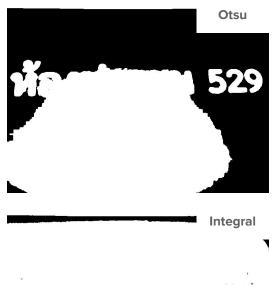
12 19 21 3

Algoritmo	Tempo	Num. Iterações	
Otsu	2.2678	2130643	
Wellners	0.9188	1065194	
Medium	17.2660	9337746410	
Integral	4.5984	2128238	

Imagem com alta intensidade w = 599, h = 919 block_size = 74 threshold = 15

Wellners





Algoritmo	Tempo	Num. Iterações
Otsu	1.2851	1101217
Wellners	0.4971	550481
Medium	9.8246	2859022481
Integral	2.4825	1099445

Imagem com baixa intensidade w = 599, h = 919block_size = 74 threshold = 15

Wellners

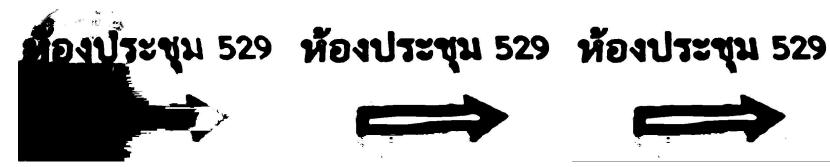


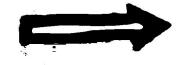
Adaptive

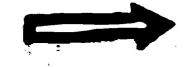


Integral

Otsu







Resultados

Algoritmo	Vantagens	Desvantagens	Quando Usar
Otsu	Fácil implementação Baixo custo	Problemas ao lidar com iluminação	Histogramas com distribuição bimodal de intensidade Fundo uniforme
Wellners	Fácil implementação Adapta-se melhor com variação da iluminação	Depende da ordem de leitura dos pixels Baixa representação dos vizinhos	Pequenas variações de iluminação
Medium	Grande representação dos vizinhos Robusto quanto a variação de iluminação Grande parametrização	Custo computacional	Imagens com grande variação de iluminação
Integral	Eficiência computacional	Menos flexível Necessita de uma iteração a mais	Necessidade de alta performance: performance real-time

Limitações

- Baixa taxa de recall quando objeto de interesse está em áreas de alta intensidade
- Caso objeto de interesse seja maior do que S x S, ele pode ser classificado como fundo. Para resolver, basta aumentar o tamanho de S, porém pode perder detalhes da imagem
- Ajustes dos parâmetros Block size e Threshold interferem, porém percebeu-se que utilizar block_size = 1/8 da largura da imagem e threshold = 15 apresentou bons resultados
- Menos eficiente para imagens grandes:
 - Cálculo linear da imagem integral
 - Acesso a memória

Ponto de melhoria

- Computar métricas de qualidade da segmentação (taxa recall)
- Otimização do algoritmo
 - Podemos realizar a limiarização simultaneamente ao cálculo da imagem integral, pois a limiarização de um pixel, não exige que toda a matriz da imagem integral esteja completa

Referências

Bradley, D., & Roth, G. (2004). Adaptive thresholding using the integral image. Journal of Graphics Tools, 9(1), 11-17.

Pintaric, T. (2023, 14 de junho). Concept of Modified Adaptive Thresholding Using Integral Image to Decompose Text Under Illumination. Kittipop-P. https://kittipop-p.medium.com/concept-of-modified-adaptive-thresholding-using-integral-image-to-decompose-text-under-illumination-f8ced99e271