Processamento Digital de Imagens – Trabalho 1
Filtros de Suavização Não Lineares

Guilherme L. Salomão, João Victor M. Freire,
Martin Heckmann, Renan D. Pasquantonio

23 de Setembro de 2020

1 Introdução

A Filtragem Espacial é uma técnica de processamento de imagens que, diferentemente das transformações pontuais de intensidade, determinam a nova intensidade de um pixel baseado nas intensidades dos pixels em uma determinada vizinhança $m \times n$.

Existem duas categorias de filtros espaciais: lineares e não lineares. Os primeiros são caracterizados por algoritmos eficientes, além de resultados analíticos importantes que melhoram sua performance. Podem ser aplicados à imagem através das operações de correlação-cruzada ou de convolução. Já os filtros não-lineares são caracterizados por um alto custo computacional, porém em certos casos é possível obter resultados melhores do que com os lineares.

2 Motivação

Um dos principais tipos de filtragem espacial é a suavização. Seus principais usos são para tornar objetos de uma imagem mais uniformes ou para remoção de ruídos. Dentre os filtros lineares de suavização, estudamos em aula o filtro de média simples e o filtro gaussiano. Agora, neste trabalho, implementamos dois filtros de suavização usando média geométrica e mediana, que nos permitem obter resultados diferentes daqueles obtidos usando os filtros lineares de suavização.

3 Explicação do Método Implementado

3.1 Média Geométrica

A média aritmética comum consiste em dividir a soma de n valores por n, ou seja $\frac{1}{n}\sum_{j=1}^{n} a_j$. Já a chamada média geométrica é descrita como o produto de n valores elevado à $\frac{1}{n}$, ou seja, $[\prod_{j=1}^{n} a_j]^{\frac{1}{n}}$.

No contexto de processamento de imagens, e em particular de filtros espaciais não-lineares, podemos definir o conjunto dos pixels que devem ser incuídos no produtório como o conjunto S_{xy}, que é um retângulo de tamanho $m \times n$ centrado em

cada pixel (x, y) da imagem. Assim, definimos a transformação da imagem como

$$\hat{f}(x, y) = [\prod_{(s,t)\in S_{xy}} f(s, t)]^{\frac{1}{mn}},$$

ou então dizemos que a nova intensidade de cada pixel (x, y) é a média geométrica dos pixels à sua volta.

Em alguns casos, a média geométrica pode ajudar a preservar detalhes. É especialmente útil no processo de remoção de ruído Gaussiano e é melhor que a média aritmética na preservação de bordas, mas na prática acaba sendo mais utilizada quando os valores de intensidade estão em escala logarítmica ou com intensidades com grandes intervalos de valores (não costuma ser o caso de imagens).

3.2 Mediana

A mediana divide o conjunto de valores exatamente na metade, o que é feito ordenando o conjunto e selecionando o elemento do meio.

Em processamento de imagens, pode ser usada para criação de um filtro espacial não-linear de suavização. Consiste na aplicação da fórmula

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\text{mediana}}[f(s, t)]$$

na imagem. Assim, a intensidade de cada pixel (x, y) é a mediana das intensidades dos pixels no retângulo $m \times n$ centrado no ponto.

O filtro de mediana é bastante útil no pré-processamento de imagens. É um filtro muito efetivo na remoção de ruídos e texturas, principalmente em ruídos impulsivos, mas que consegue preservar as bordas de uma imagem.

4 Explicação do Código

4.1 Média Geométrica

No bloco abaixo, podemos ver a definição da função `geometric_mean` (comentários da função no notebook jupyter):

```
def geometric_mean(img, m, n):

    num_rows, num_cols = img.shape

    half_row_size = m//2
    half_col_size = n//2
    img_padded = np.pad(img, ((half_row_size,half_row_size),
                              (half_col_size,half_col_size)),
                        mode='symmetric')
    img_filtered = np.zeros((num_rows, num_cols))
    for row in range(num_rows):
        for col in range(num_cols):
            product_region = 1
            for s in range(m):
                for t in range(n):
                    product_region *=
                    (img_padded[row+s, col+t])**(1./(m*n))
            img_filtered[row, col] = product_region

    return img_filtered
```

A função `geometric_mean` recebe como parâmetros, `img` sendo a imagem a ser filtrada, `m` e `n` sendo respectivamente a largura e a altura do filtro. A utilização de dois parâmetros responsáveis pela dimensão do filtro permite a utilização de filtros retangulares que proporcionam diferentes resultados de filtragem.

Obtendo as dimensões da imagem com `img.shape` podemos então gerar uma nova imagem chamada `img_padded`, que utilizaremos para a aplicação do filtro. A `img_padded` será uma imagem semelhante à original, mas com algumas adaptações. Nela serão acrescentadas bordas laterais de tamanho `m/2`, isto é, metade da largura do filtro, e bordas na parte inferior e superior da imagem de tamanho `n/2`, que é a metade da altura do filtro. Essas novas posições serão preenchidas com os valores espelhados da imagem original, não repetindo apenas a borda ou uma outra constante.

Esse processo é chamado espelhamento. Isso é feito para possibilitar a aplicação do filtro em *pixels* próximos à borda da imagem com resultados melhores ao reutilizar valores que já estão dentro da região do filtro. Caso utilizássemos o valor padrão zero no preenchimento, a imagem resultante acabaria com bordas pretas.

Em seguida, iremos acessar cada *pixel*, ou seja, cada elemento da matriz e aplicar o filtro em cada um. A aplicação do filtro consiste na aplicação da fórmula da média geométrica mostrada anteriormente nos valores de dentro do filtro, no entanto, foi feita uma pequena alteração na fórmula durante a implementação para impedir *overflow* durante a multiplicação. A modificação utiliza a propriedade da exponencial que diz que $(a \times b)^n = a^n \times b^n$. Assim, a nova transformação será definida como

$$\hat{f}(x,y) = [\prod_{(s,t)\in S_{xy}} f(s,t)^{\frac{1}{mn}}].$$

O valor final dessa operação é atribuída ao elemento correspondente na imagem final filtrada `img_filtered`, repetindo esse processo para todos os pixels da imagem. E por fim retornando a nova imagem.

4.2 Mediana

No bloco abaixo, podemos ver a definição da função `median` (comentários da função no notebook jupyter):

```
def median(img,m,n):

    num_row, num_col = img.shape

    half_row_size = m//2
    half_col_size = n//2
    img_padded = np.pad(img, ((half_row_size,half_row_size),
                              (half_col_size,half_col_size)),
                        mode='symmetric')

    img_filtered = np.zeros((num_row,num_col))
    for row in range(num_row-1):
        for col in range(num_col-1):

            med_array = []
            for k in range(m):
                for y in range(n):
                    med_array.append(img_padded[row+k][col+y])
            med_array = np.sort(med_array)
            median = len(med_array)//2
            if(len(med_array)%2 == 0):
                img_filtered[row][col] = ((med_array[median-1]*0.5)
                                          +(med_array[median]*0.5)
                                         ).astype('uint8')
            else:
                img_filtered[row][col] = med_array[median]
    return img_filtered
```

Para o filtro de mediana, recebemos os mesmos parâmetros que o filtro da média geométrica, isto é, `img` sendo a imagem a ser filtrada, a largura `m`, e a altura `n` do filtro. Com isso também geramos uma imagem `img_padded` com espelhamento da mesma forma que foi implementada em `geometric_mean`.

Após a geração do `img_padded`, será acessado cada pixel da imagem, ou seja, cada elemento da matriz, e para cada um iremos coletar todos os valores ao redor dele, obedecendo o tamanho do filtro, e armazená-los no vetor chamado `med_array`.

Esse vetor será ordenado para obtermos a mediana, que será o valor central do vetor caso ele tenha um tamanho ímpar, ou a média aritmética dos dois valores centrais caso seja um vetor de tamanho par.

O valor da mediana será atribuído ao elemento correspondente ao atual na imagem filtrada `img_filtered`, e o processo será repetido para todos os pixels da imagem. E por fim retornando a imagem filtrada.

Referências

[1] Comin, C. H. Processamento Digital de Imagens – Aula 8: Filtragem Espacial I, 2020.

[2] Jr., M. P. Image Restoration – Image Processing scc0251. http://wiki.icmc.usp.br/images/7/78/Dip08_restoration.pdf, 2011.

[3] Wikipedia. Geometric Mean. https://en.wikipedia.org/wiki/Geometric_mean.

[4] Wikipedia. Median. https://en.wikipedia.org/wiki/Median.