

# Comparação de métodos de suavização de ruídos localizados para imagens destinadas a biometria de digitais.

Luiz Gabriel Zeferino Duarte  
CC6P28  
SJRIOPRETOII  
N454CD8

Lucas de Oliveira Brandolezi  
CC6P28  
SJRIOPRETOII  
D9380G9

**RESUMO:** A biometria é um método de autenticação muito utilizado hoje em dia e, por envolver análise de imagens, o pré processamento é muito importante. por isso, nesse artigo abordaremos três tipos de algoritmos de suavização (média, mediana e Gaussiana) e veremos como eles se comportam dado um determinado tipo de ruído.

## 1. INTRODUÇÃO:

O uso de autenticações biométricas tem se tornado cada vez mais presente em nossas vidas, possuindo uma clara vantagem em relação às senhas convencionais, elas não podem ser esquecidas, quase nunca perdidas e, dependendo do método utilizado, extremamente únicas para cada indivíduo. Porém esse tipo de autenticação requer um sistema bem mais robusto uma vez que trabalha com imagens. Utilizando a biometria de digitais como exemplo, é preciso primeiro tirar uma “foto” da sua digital através de um aparelho específico e então guardá-la em um banco de dados. Assim, quando você precisar efetuar uma autenticação, você novamente tirara uma foto da sua digital, porém essa será usada

apenas para comparação com a imagem original, assim, se as duas imagens forem satisfatoriamente iguais, o sistema reconhece que se trata da mesma pessoa tentando acessar o serviço e libera o acesso.

porém, durante a etapa da comparação é muito comum que digitais iguais não batam devido a presença de pequenos ruídos localizados causados por falhas na leitura e transporte dessa imagem. por isso é muito importante realizar uma etapa de pré-processamento, adicionando filtros que buscam suavizar esses ruídos que atrapalham a autenticação.

## 2. METODOLOGIA:

O intuito deste artigo é comparar os três principais algoritmos de suavização de ruídos: média, mediana, gaussiana e analisar qual deles apresenta melhor resultado ao ser aplicado em uma imagem com um ruído localizado.

### 2.1. especificações:

A coleta de dados e análise dos resultados bem como o código fonte foram feitos utilizando o software GNU Octave versão 6.3.0 para Windows.

A imagem base utilizada para todos os processos foi a [Im.1.1] e, para fins de melhor visualização do resultado, ela foi ampliada em 3 vezes resultando na imagem [Im.1.2] vista a seguir:



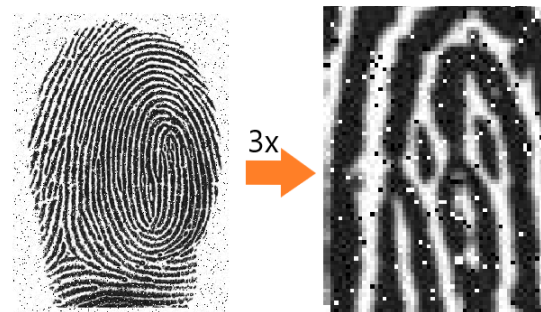
Im.1.1

Im.1.2

Além disso, todas as máscaras utilizadas nos algoritmos são de matrizes 3x3 (9 pixels).

### 2.2. Geração do ruído:

A fim de emular de forma contundente uma imagem com ruídos representados por pequenos pontos espalhados causados por perdas de informação durante a leitura e transporte da imagem, concluímos que a melhor forma de fazê-lo era submeter nossa imagem original [Im.1.1] a um ruído conhecido como “salt and pepper” que tem como característica pequenos ruídos localizados. o resultado pode ser visto na imagem [Im.1.3] e na sua amplificação [Im.1.4] a seguir:



Im.1.3

Im.1.4

### 2.3. Os três tipo de suavização:

Os três tipos de suavização utilizados neste estudo foram por média, mediana e gaussiana.

A suavização por média tem como característica aplicar uma máscara de convolução [Im.2.1] que altera o pixel central para uma média aritmética de todos os seus vizinhos.

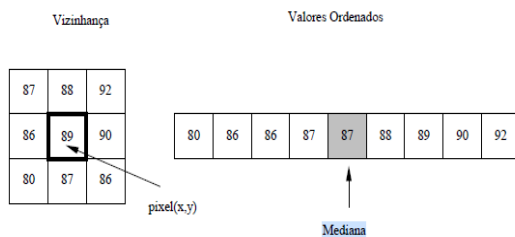
Já suavização por mediana tem como característica substituir o pixel central pela mediana dos valores de seus vizinhos[Im.2.2].

E por fim a suavização por gauss tem como característica aplicar uma máscara convolucional [Im.2.3] que substitui o pixel central de acordo com a distribuição normal dos seus pixels vizinhos.

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Im.2.1 (média)



Im.2.2(mediana)

$$1/16$$

1	2	1
2	4	2
1	2	1

Im.2.3(gaussiana)

No octave, as implementações usadas para efetuar as suavizações foram, respectivamente:

```
I = imread('imagem.jpg');
```

```
[l c] = size(I);
```

```
for x=2:l-1
```

```
    for y=2:c-1
```

```
        Imedia(x,y)=(double(I(x-1,y-1)) +
double(I(x,y-1)) + double(I(x+1,y-1)) +
double(I(x-1,y)) + double(I(x,y)) +
double(I(x+1,y)) + double(I(x-1,y+1)) +
double(I(x,y+1)) + double(I(x+1,y+1))) /9;
```

```
    endfor
```

```
endfor
```

---

```
I = imread('imagem.jpg');
```

```
Imediana = I;
```

```
[l c] = size(I);
```

```
for x=2:L-1
```

```
    for y=2:C-1
```

```
        vetor=[I(x-1,y-1),I(x-1,y),I(x-1,y+1),I(x,y-
1),I(x,y),I(x,y+1),I(x+1,y-1),I(x+1,y),I(x+
1,y+1)];
```

```
        for i=1:8
```

```
            for j=1:8
```

```
                if(vetor(j)>vetor(j+1))
```

```
                    aux = vetor(j);
```

```
                    vetor(j) = vetor(j+1);
```

```

        vetor(j+1) = aux;

    endif

endfor

endfor

Imediana(x,y) = vetor(5);

endfor

endfor
-----

I = imread('imagem.jpg');

[l c] = size(I);

for x=2:l-1

    for y=2:c-1

        Igaussiana(x,y) =(1*double(I(x-1,y-1)) +
        2*double(I(x,y-1)) + 1*double(I(x+1,y-1))
        + 2*double(I(x-1,y)) + 4*double(I(x,y)) +
        2*double(I(x+1,y)) + 1*double(I(x-1,y+1))
        + 2*double(I(x,y+1)) +
        1*double(I(x+1,y+1))) /16;

    endfor

endfor

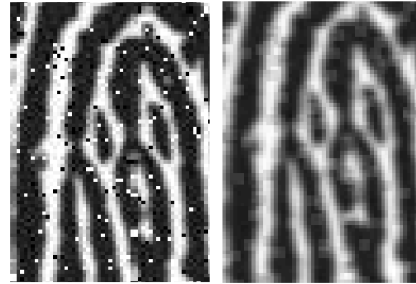
```

### 3. Resultados:

Agora serão apresentados os resultados dos testes previamente descritos através da comparação entre as imagens, dos histogramas das imagens e os tempos de execução.

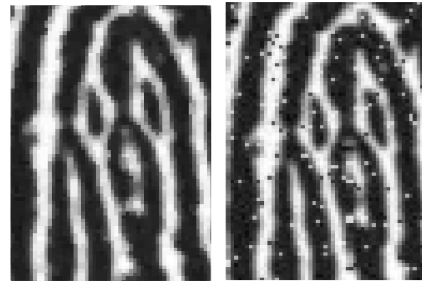
#### 3.1. Resultado da comparação entre os filtros:

A seguir estão representadas o ruído original [Im.3.1] e as imagens já suavizadas pelo método de média [Im.3.2], mediana[Im.3.3] e gaussiana[Im.3.4].



Im.3.1(original)

Im.3.2(média)

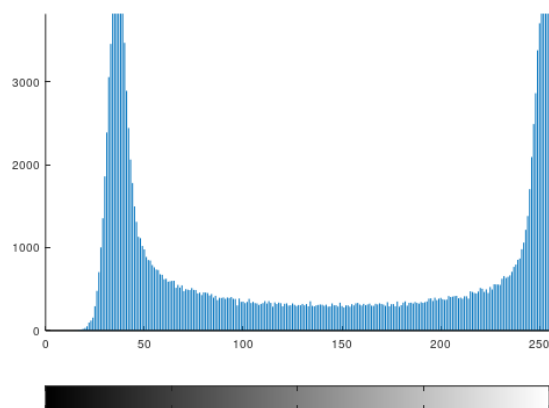


Im.3.3(mediana)

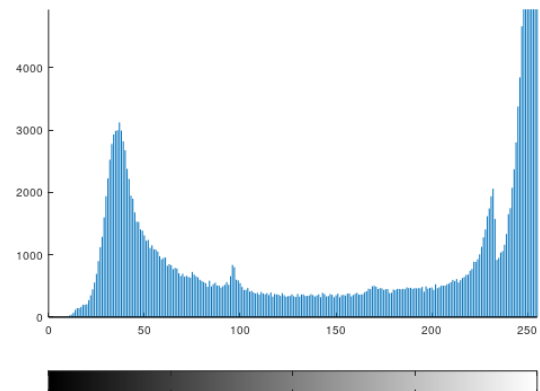
Im.3.4(gaussiana)

#### 3.2. Resultado da comparação dos histogramas entre os filtros:

A seguir estão representados os histogramas da imagem original sem ruído [Im.4.1], suavizada pela média[Im.4.2], mediana[Im.4.3] e gaussiana[Im.4.4] a fim de analisar qual das imagens suavizadas mais se assemelha a imagem naturalmente sem ruído.



Im.4.1(original)



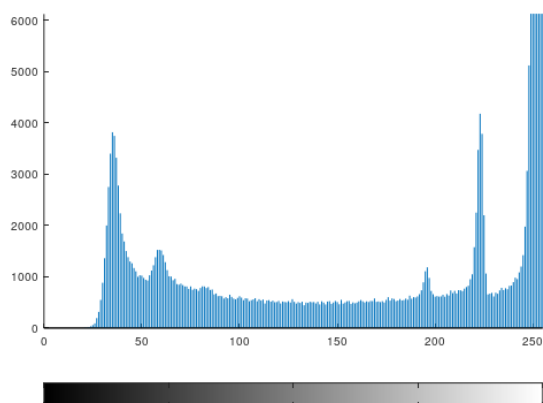
Im.4.4(gaussiana)

### 3.3. Resultado da comparação entre os tempos de execução:

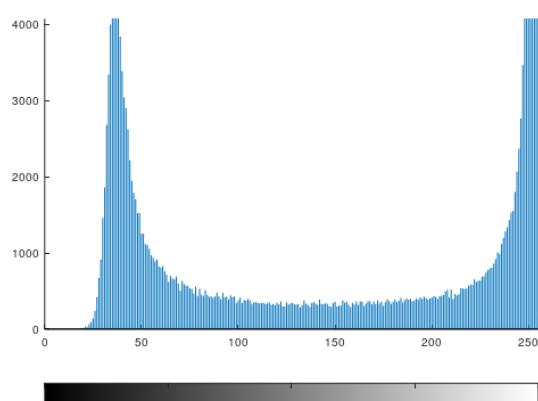
A seguir está representado o tempo de execução de cada método de suavização.

Tabela 1. Tempos de execução

método:	tempo(segundos)
média	15.50
mediana	70.12
gaussiana	15.47



Im.4.2(média)



Im.4.3(mediana)

### 3. CONCLUSÕES:

Comparando o resultado das imagens e os histogramas de cada algoritmos com o histograma da imagem original [Im 4.1], percebemos que o método da mediana [Im 4.3] foi o que obteve uma maior semelhança com o histograma da imagem original [Im 4.1], seguido pelo método da média [Im 4.2] e depois o gaussiano [Im 4.3]. Já com relação ao tempo de execução de cada algoritmo, percebemos uma grande semelhança entre o tempo do método de média e do método Gaussiana e grande distância com relação ao método de mediana. Com as comparações que

fizemos nesse artigo podemos concluir que o melhor resultado nas comparações entre os histogramas e o tempo de execução foi o método de suavização pela média.

#### 4.BIBLIOGRAFIA:

[1] SEGURANÇA DA INFORMAÇÃO:  
BIOMETRIA, UMA MANEIRA DE  
AUTENTICAÇÃO EFICIENTEWillian  
dos Santos Alécio<sup>1</sup>, Júlio Cesar Pereira<sup>1</sup>,  
Késsia Rita da Costa Marchi<sup>1</sup>  
Universidade Paranaense  
(UNIPAR)Paranavaí – PR – Brasil.

[2] Filtros compostos e adaptativos: o  
filtro de gaussiano, laplaciano do  
gaussiano e de  
gabor(harmônico-gaussiano)aluno:  
Luciano Lucas de Oliveira Junior e myself.

[3]EA978 Sistemas de Informações  
Gráficas - Prof. J. Mario De Martino.

[4]Filtros espaciais(suavização)Prof. Luiz  
Otavio Murta Jr.Informática Biomédica  
Depto. de Física e Matemática  
(FFCLRP/USP).