

# SUAVIZAÇÃO DE IMAGENS – IMAGE SMOOTHING

## Ruídos :

### *Salt and Pepper Noise*

Causado por erros na transmissão de dados. Os pixels corrompidos ou são alterados para o valor máximo, ou tem alguns bits alterados, causando uma diferença brusca de tons entre este pixels e seus vizinhos. Quando os pixels são alternadamente modificados para 0 ou o máximo, este ruído é chamado de ruído salt and pepper, devido a sua aparência. Para este tipo de ruído, as melhores técnicas são de filtragem em passa-baixa, ou seja filtros de suavização como o de média e gaussiano são relativamente mal sucedidos porque o pixel que foi alterado pode variar significativamente do valor original, e assim a média pode dar um valor diferente do valor original. Um filtro mais eficiente nesse caso seria o filtro de mediana, que remove este tipo de ruído mais eficientemente e preserva o contorno e pequenos detalhes da imagem.

Ex.



**Imagem Original**



**Imagem Com Ruído Salt and Pepper**

### *Gaussian noise*

Ruído gaussiano pode ser reduzido usando um filtro espacial. De qualquer forma deve-se ter em mente que quando suavizando a imagem nos reduzimos não apenas o ruído, mas também detalhes de pequena escala contidos na imagem .

Ex.



**Imagem Original**



**Imagem Com Ruído Gaussiano**

## Filtros :

Filtros de passa baixa também conhecidos como filtros de suavização são empregados na remoção de ruídos de alta frequência espacial em imagens digitais. Ruídos que geralmente são introduzidos durante o processo de conversão analógica-digital.

Entre as técnicas de suavização conhecidas existem as de suavização conservativa, técnicas de redução de ruídos que tem seu nome derivado do fato de empregar um algoritmo simples e rápido de filtragem que elimina o ruído de forma a manter os detalhes de altas frequências como os contornos da imagem. Essa técnica é especialmente desenvolvida para remover picos de ruído como ruídos salt and pepper, porém não é muito eficiente na redução de ruído aditivo como o ruído gaussiano por exemplo.

As técnicas de suavização não conservativas, como os filtros de média por exemplo, geralmente eliminam detalhes como linhas finas e curvas agudas, causando o efeito de blurring.

### *Filtros de média*

O filtro de média é implementado da seguinte maneira, temos uma janela que percorrerá toda a imagem, o elemento central dessa janela receberá a média de todos os elementos da janela. Quanto maior for a janela, mais influencia dos vizinhos este pixel sofrerá e maior será o efeito de blurring, visto que levaremos em consideração um número maior de pixels.

Por exemplo vejamos a janela 3x3 abaixo :

$$\begin{bmatrix} 244 & 247 & 245 \\ 252 & 12 & 238 \\ 244 & 245 & 250 \end{bmatrix}$$

A média dos valores será :

$$(12 + 238 + 244 + 244 + 245 + 245 + 247 + 250 + 252) / 9 = 219$$

Assim o valor desse pixel que era 12 será de 219

Agora imagine o exemplo em que tenhamos uma linha preta cortando a imagem, e a janela pegue esta linha, da seguinte forma :

$$\begin{bmatrix} 244 & 12 & 245 \\ 252 & 11 & 238 \\ 244 & 08 & 250 \end{bmatrix}$$

A média dos valores será :

$$(12 + 238 + 244 + 244 + 08 + 245 + 11 + 250 + 252) / 9 = 169$$

E como resultado teremos uma linha cinza médio que muitas vezes será imperceptível no contexto. Perdendo assim este detalhe.

Averaging é um caso especial de convolução discreta. Por exemplo o caso abaixo onde temos uma janela 3x3 e a máscara de convolução é a seguinte:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Veja o seguinte exemplo :

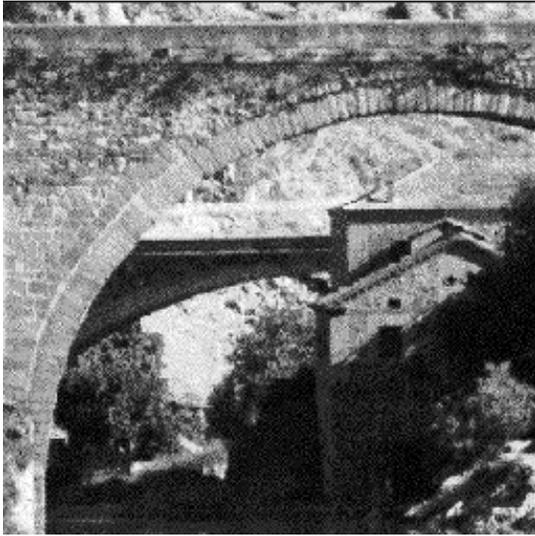


Imagem Original



Imagem com Ruído Salt and Pepper



Imagem após filtro de media 3x3

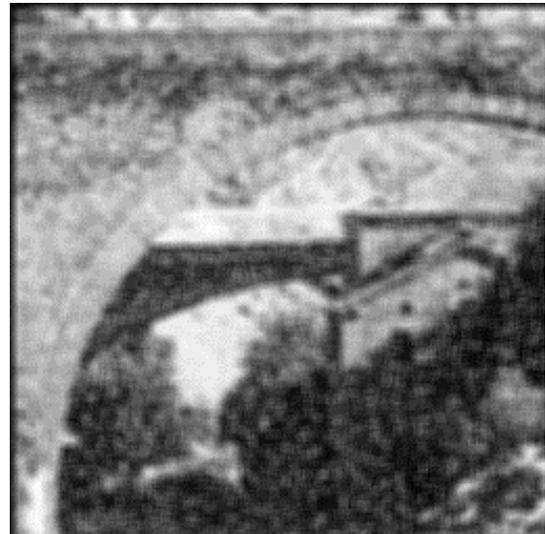


Imagem após filtro de media 5x5

### ***Filtros de mediana***

Os filtros de mediana reduzem o blurring e preserva a edging coloca o valor da mediana no elemento do meio o tamanho da janela importa.

A desvantagem principal do filtro de mediana em uma vizinhança retangular é o dano causado nas linhas finas e curvas agudas.

O pixel é substituído pelo valor médio de seus vizinhos , caso o tamanho da sua janela seja par , então o valor da mediana será a media dos dois valores centrais. Esse filtro é um dos melhores filtros de suavização que preserva o contorno. Nos pegamos a mediana ordenando em ordem crescente ou decrescente, os valores dos pixels e pegamos o valor médio .

Por exemplo :

Dada a seguinte janela 3x3

$$\begin{bmatrix} 244 & 247 & 245 \\ 252 & 12 & 238 \\ 244 & 245 & 250 \end{bmatrix}$$

Ordenando os valores teremos :

12 238 244 244 **245** 245 247 250 252

O quinto valor será a mediana ou seja **245**

Em geral filtros de mediana permitem que grande partes dos detalhes que possuem altas frequências espaciais da imagem que passem, ao mesmo tempo que é muito efetivo na remoção de ruídos, onde menos da metade dos vizinhos tenham sido modificados pelo ruído. Como consequência disso o filtro de mediana pode não ser tão eficaz na remoção de ruídos gaussianos.

Um dos maiores problemas do filtro de mediana é o seu custo relativamente complexo e caro . Para encontrar a mediana é necessário que você ordene todos os valores da vizinhança, o que é relativamente lento. performance.

Veja o seguinte exemplo :

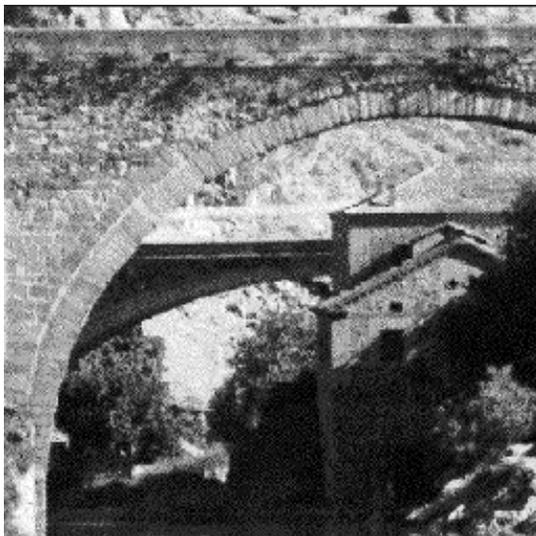


Imagem Original

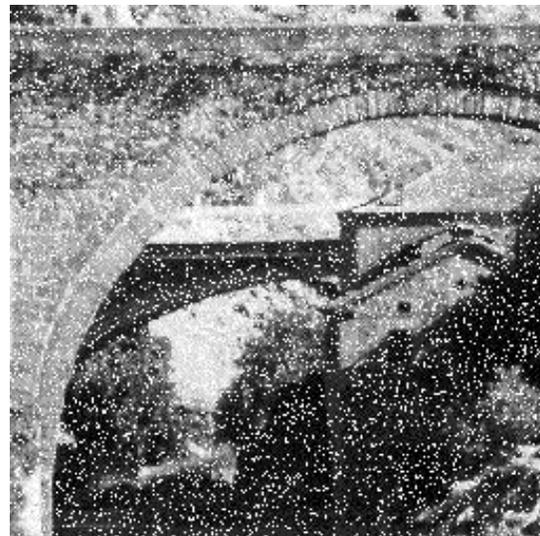


Imagem com Ruído Salt and Pepper



Imagem após filtro de media 3x3

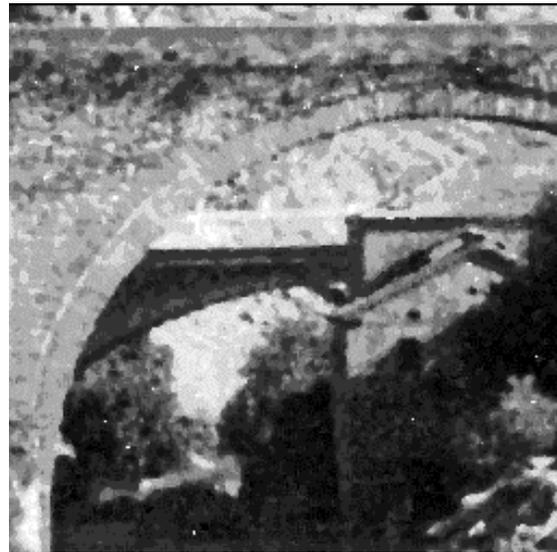


Imagem após filtro de mediana 3x3

### **Referências :**

*Notas de aula.*

***Livros:***

Alan H. Watt, [Fabio Policarpo - The Computer Image](#), Addison-Wesley Pub Co (Net); 1998- ISBN: 0201422980

***Sites :***

<http://www.dai.ed.ac.uk/HIPR2/csmooth.htm>

<http://www.nasatech.com/Briefs/Sept99/NPO20475.html>

<http://www.icaen.uiowa.edu/~dip/LECTURE/contents.html>

<http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Smoothin.html>

[http://imaging.utk.edu/publications/papers/Yiyong\\_ICIP2000.pdf](http://imaging.utk.edu/publications/papers/Yiyong_ICIP2000.pdf)

<http://www.worldscinet.com/journals/ijprai/15/1504/S0218001401001076.html>

<http://ct.radiology.uiowa.edu/~jiangm/courses/dip/html/node70.html>

<http://www.lems.brown.edu/~tek/research/morphology/image-smoothing.html>

<http://citeseer.nj.nec.com/context/478769/0>

<http://citeseer.nj.nec.com/cs>

<http://www.dai.ed.ac.uk/CVonline/transf.htm>

<http://www.cs.rochester.edu/~kyros/Courses/CS290C/Lectures/lecture-5/sld017.htm>

<http://mvc.man.ac.uk/research/visual/medical/courses/SNOTES/studentnotes-5.html>

[http://www.math.purdue.edu/~lucier/692/related\\_papers\\_summaries.html](http://www.math.purdue.edu/~lucier/692/related_papers_summaries.html)

<http://rkb.home.cern.ch/rkb/AN16pp/node268.html>

<http://www.umiacs.umd.edu/users/vasi/JavaWorkshop/node4.htm>

<http://www.cs.uni-duesseldorf.de/aurich/nlg/papers.html>

<http://www.students.ncl.ac.uk/ganesh.vengadasalam/exciting/tutorial/chapter3/chapter3.html>

<http://www.cs.sunysb.edu/~mueller/teaching/cse332/imgProc.pdf>

<http://www.ecn.purdue.edu/VISE/ee438L/lab10/pdf/lab10a.pdf>