

Filtro gaussiano

O filtro Gaussiano pode ser usado como um filtro **passa-baixa**. Usando a função Gaussiana para obter valores de uma máscara a ser definida digitalmente.

O Filtro de Gaussiano em 1-D tem a forma:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$$

onde σ é o desvio padrão. É assumido que a distribuição tem média zero (i.e. está centrada em $x=0$). A distribuição é ilustrada na Figura 1.

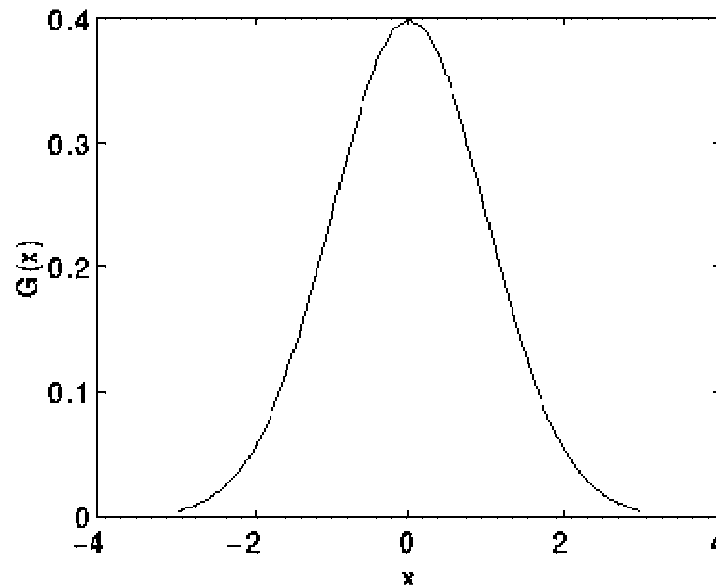


Figura 1- Forma 1D da distribuição Gaussiana com média zero e desvio padrão um, $\sigma = 1$ na equação da função Gaussiana, o valor máximo da curva é $\approx 0,3989$.

Em 2-D, sua forma isotrópica (i.e. circularmente simétrica) é dada pela equação:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Mostrada na figura 2.

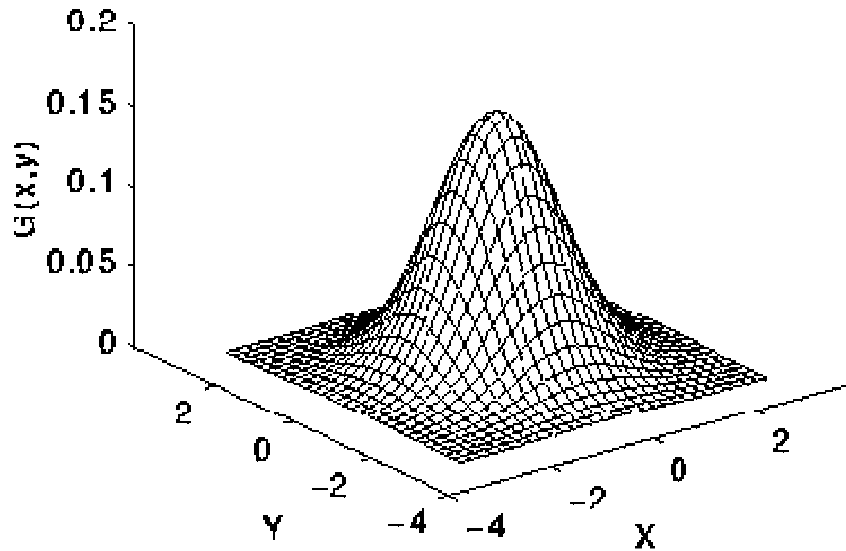


Figura 2 – Representação da função Gaussiana em 2D com média em (0,0) e desvio padrão $\sigma=1$, o valor máximo da curva é $\approx 0,1592$

Considerando que a imagem é armazenada como uma coleção de pixels discretos que nós precisamos produzir uma **forma discreta da distribuição** Gaussiana para obter um núcleo de convolução (máscara ou kernel), este seria teoricamente infinitamente grande, mas na prática se zera aproximadamente a uma distância de três unidades (ou pixels) do centro.

A Figura 3 mostra a forma inteira 5x5 do kernel da Gaussiana para $\sigma= 1.0$.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Figure 3 Forma discreta aproximada da função Gaussiana com $\sigma=1.0$, o valor máximo em $x=0$ é $1/(2\pi)\approx 0,1591\approx 41/273= 0,15$. É possível encontrar aproximações com valor central de $42/273$, $43/273$ e até $44/273=0,1611$

Um forma mais rápida é, ao invés da convolução em 2-D fazer primeiro uma convolução em 1-D na direção x e depois usar a mesma máscara 1-D da Gaussiana na direção y . A Figura 4 mostra uma forma de reproduzir o kernel 2D usando 7 pixels, em duas convoluções em 1-D. Neste uso os valores depois deverão ser multiplicados por 273 arredondados para cima, sempre e finalmente truncados para inteiros. O componente em y é o mesmo, mas orientado verticalmente.

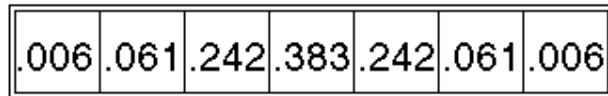


Figura 4 Um dos pares 1-D para convolução que tem o mesmo efeito do kernel da Figura 3.

As figuras 5 e 6 abaixo, para uma aproximação da Gaussiana com desvio padrão $\sigma^2 = 2/\pi \approx 0,6366$, ou seja $\sigma \approx 0.8$.

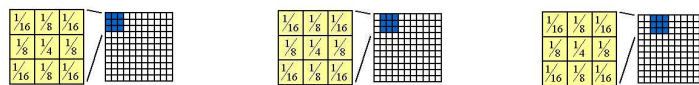


Figura 5 - Passagem do filtro Gaussiano 2D com $\sigma^2 = 2/\pi \approx 0,6366$, ou seja $\sigma \approx 0.8$.

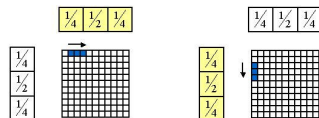


Figura 6 – Mesmo resultadp usando dois filtros Gaussianos em 1D com $\sigma^2 = 2/\pi \approx 0,6366$, ou seja $\sigma \approx 0.8$.

Para filtros Gaussianos com maiores desvios padrões podem-se montar outra máscara, mas o mais usual é passar a mesma máscara mais de uma vez na imagem.

Os filtros Gaussianos são muito úteis também em Biologia Computacional e Visão, pois no processo de visão algumas células têm forma de resposta aproximadamente Gaussiana.

O uso do filtro Gaussiano é suavizar (*smoothing*, *blur*) a imagem, quase da mesma forma que o filtro de média (mean filter). O resultado será tão mais suave quanto maior o valor de sigma, σ , desvio padrão (*standard deviation*) da Gaussian usada. Desvios padrões maiores também fazem com que a máscara (*convolution kernels*) deva ser maior para uma representação adequada. Teste esse desempenho usando alguns softwares, como o tutorial “HIPR2”: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm#2>

Por não ter todos os pesos iguais como o **filtro de média**, o filtro Gaussiano faz uma suavização mais delicada (*gentler smoothing*) preservando mais os contornos que o de média.

Mas as melhores justificativas para seu uso e seu melhor desempenho se relaciona com sua resposta no domínio das frequências (*frequency response*). A maioria dos filtros de suavização (o de mediana é uma exceção) são baseados na convolução agindo como passa baixas no domínio das frequências ([lowpass frequency filters](#)). Isso significa que removem frequências espaciais depois de um certo valor. Suas diferenças assim são mais bem entendidas se **olharmos para suas formas depois de transformados por Fourier** ([Fourier transform](#)). A Figura 7 mostra como o filtro de média 1-D de largura 7 e como o Gaussiano com $\sigma=3$ ficam depois de transformados para o domínio da frequência usando a expressão contínua da transformada de Fourier 1D.

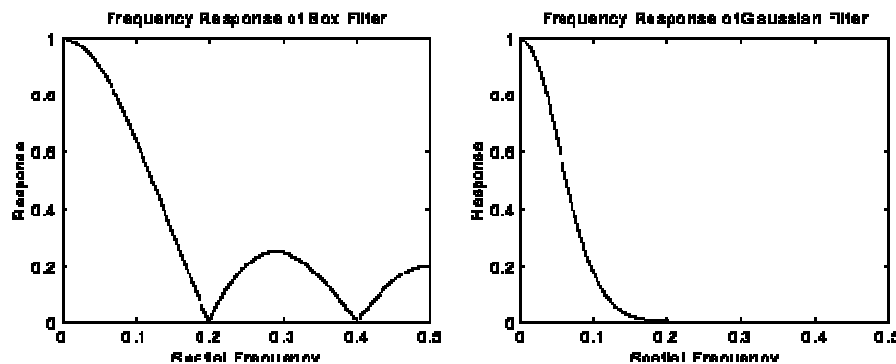


Figura 7 Resposta em Frequência do filtro de média (*i.e. mean filter*) com 7 pixels e do Gaussiano com $\sigma=3$. O eixo horizontal das frequências (*frequency axis*) está representado em ciclos por pixel (vai até 0,5 apenas pois valores maiores são insignificantes).

Os dois filtros atenuam altas frequências, mas o de média tem caráter oscilatório. O próprio formato da curva Gaussiana é preservado no domínio das frequências, isto é continua sendo uma Gaussiana. Isto faz com que ele seja adequado em combinações com outros em técnicas de detecção de bordas como a [zero crossings](#) e seja usado para derivar outros como o que veremos a seguir, o filtro de Gabor (e também o [Canny edge detector](#)).

Filtro Laplaciano de Gaussiano

O Laplaciano , $L(x,y)$, de uma imagem é a derivada segunda do valor da função que descreve a intensidade dos seus pixels , $I(x,y)$, sendo representado por:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Considerando que a imagem é armazenada como uma coleção de pixels discretos, o que nós precisamos é produzir uma **forma discreta da derivada segunda** que é representada por um núcleo de convolução (máscara ou kernel), por exemplo, como as da Figura 8.

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

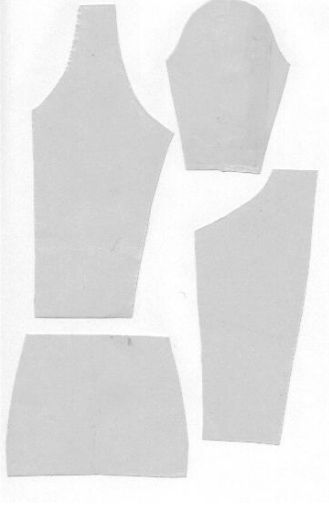
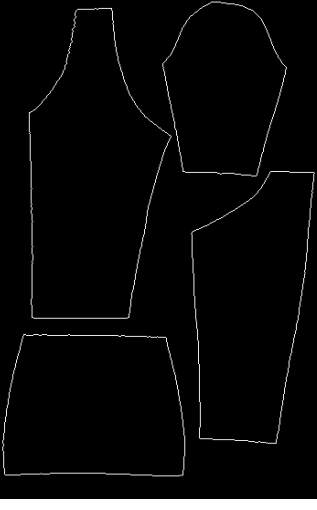
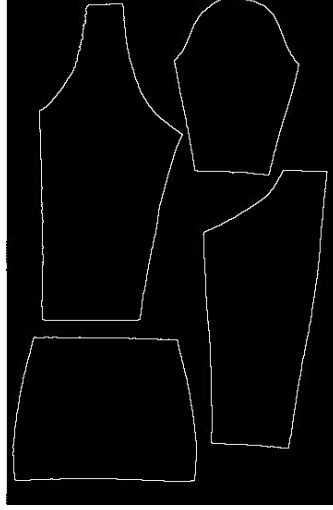
Figura 8 - Aproximações do filtro Laplaciano. (Note, que aqui o valor central é positivo, mas pode ocorrer deste ser negativo, ou seja toda a máscara estar multiplicada por -1).

Algumas vezes pode ser interessante aumentar a detecção das bordas, isto pode ser feito aumentando os pesos das máscaras usadas, como mostrado abaixo:

$$\nabla^2 \approx$$

0	4	0
4	-20	4
0	4	0

Veja na figura a seguir o resultado da aplicação da máscara com mais peso e depois o obtido aplicando a primeira máscara da figura 8 na imagem Moldes Binarizados usadas em (D. Do B Oliveria, 2003) . Compare os resultados mostrados na figura abaixo.

		
<p>Molde Digitalizado</p>	<p>Filtro de Laplaciano da figura 8</p>	<p>Filtro Laplaciano <i>com uma máscara com mais peso</i></p>

Figuras obtidas de “Uma Proposta de Metodologia para Vetorização Controlada de Moldes para a Indústria do Vestuário”, Dissertação de Mestrado de Drucilla do Bem Oliveira, defendida em 07/2003.

Feliz Aniversario!

Como a imagem processada pelo filtro Laplaciano é muito sensível a ruídos, ela geralmente deve ser processada por um passa baixa como o filtro Gaussiano antes de aplicada na imagem.

No entanto como estas duas operações são lineares e associativas, podem ser aplicadas no kernel do filtro e depois na imagem, reduzindo bem o tempo de processamento, pois o tamanho de ambos é muito menor do que o da imagem.

Assim, surge o filtro que processa a derivada segunda da imagem processada pelo filtro de Gaussiano em uma só operação, usando a derivada segunda da máscara Gaussiana a ser usada. Este processo resulta no *Marr Edge Detector* ou LoG ('*Laplacian of Gaussian*') ou seja no **Filtro Laplaciano do Gaussiano**, que é representa por:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

E que é representado em 2D na figura abaixo.

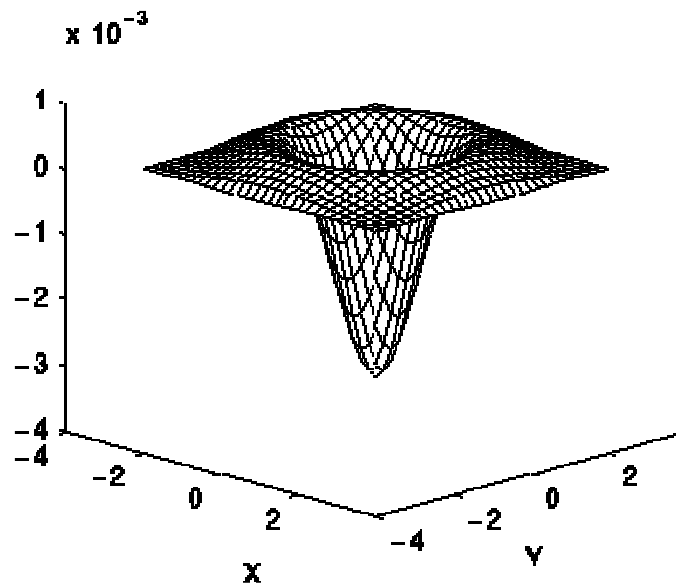


Figura 9 - Forma em 2-D do filtro Laplaciana do Gaussiano (LoG). Os eixos x e y estão em unidades de desvio padrão (σ).

Para uma Gaussiana com $\sigma = 1.4$ esta função pode ser representada no forma digital da Figura 10.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Figura 10 Aproximação Discreta do LoG para uma Gaussiana com $\sigma= 1.4$

Máscaras maiores ou mesmo menores podem ser utilizada como a da figura 11 , seguinte.

Note que o LoG tem também o formato de uma Gaussiana, mas multiplicada por um **fator que a deixa mais fina**, e inverte seu sinal de acordo com os valores de x e y. Mas tem o mesmo aspecto de inversão de sinais dos kernels simples Laplacianos. Devido a isto, se forem usadas Gaussianas muito finas ($\sigma < 0.5$ pixels) a suavização praticamente não ocorre.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Figura 11 - Máscara do Laplaciano do Gaussiano

Como o LoG calcula as derivadas segundas de uma imagem, áreas com intensidades constantes (*i.e.* com gradiente = zero), terão valores de LoG também zero. Na vizinhança da mudança da intensity, a resposta do LoG sera positiva no lado mais escuro e negativa no lado mais claro. Fazendo um aumento de contraste entre duas regiões, assim os efeitos do LoG para um função degrau como a da figura 12 são:

- zero a alguma distancia do ponto de variação das intensidade ,
- positivo no lado escuro que muda para claro,
- negative no lado claro que muda para escuro,
- cruzando o zero no ponto de degrau.

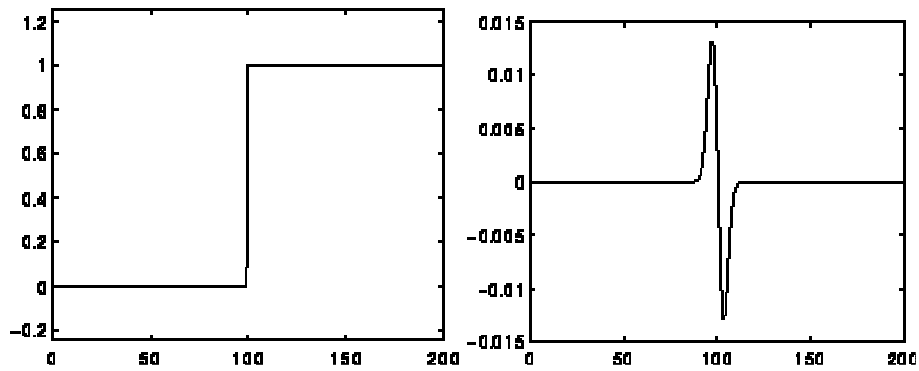


Figure 12 Resposta (à esquerda) 1-D do filtro LoG (com uma Gaussiana com $\sigma = e$ em 3 pixels) para uma função degrau (*step edge*) representada por uma imagem 1-D image com 200 (à direita).

O efeito deste filtro é salientar (*highlight*) os contornos de objetos presentes na imagem. Veja em <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm> experiências com estes filtros.

O fato de este filtro fazer os lados cruzarem o zero é usado para detectar lados no métodos chamados de [zero crossing edge detection](#). Com este tipo de método, no entanto a direção dos lados não pode ser calculada, já que a máscara do Laplaciano é simétrica.

É preciso prestar atenção nos valores dos pixels de saída, pois como estes podem ser negativos podem simplesmente deixar de ser representados ou serem representados por valores reais de forma inadequada, provocando *overflow* ou saturação nos limites.

É possível aproximar o LoG com um filtro que é apenas a diferença de duas Gaussianas de tamanhos diferentes. Este tipo de filtro é chamado de *DoG filter* (abreviatura de '*Difference of Gaussians*').

Os filtros **LoG** e **DoG** são considerados importantes no processo biológico da visão (Marr 1982). Uma aproximação dos LoG, muito mais rápidas são conhecidas como filtros **DoB** ('*Difference of Boxes*'). Que são simplesmente a diferença de 2 filtros de média de tamanhos diferentes, o que produz uma versão mais reta dos LoG.

Referências:

R. Haralick and L. Shapiro *Computer and Robot Vision*, Vol. 1, Addison-Wesley Publishing Company, 1992, pp 346 - 351.

B. Horn *Robot Vision*, MIT Press, 1986, Chap. 8.

D. Marr *Vision*, Freeman, 1982, Chap. 2, pp 54 - 78.

D. Vernon *Machine Vision*, Prentice-Hall, 1991, pp 98 - 99, 214.