

# Image Formation and Analysis (Formation et Analyse d'Images)

James L. Crowley

ENSIMAG 3 - MMIS Option MIRV

First Semester 2010/2011

Lesson 7

8 December 2010

## Description of Images with Gaussian Derivatives

### Lesson Outline:

1	Image Description Using Gaussian Derivatives.....	2
	Gaussian Derivatives Operators.....	2
	The Gaussian function in 1 and 2 D.....	2
	Gaussian Digital Filters .....	4
	Properties of Gaussian Receptive Fields (or Wavelets).....	6
	Using the Gaussian to compute image derivatives .....	8
	Steerability of Gaussian Derivatives.....	8

# 1 Image Description Using Gaussian Derivatives

## Gaussian Derivatives Operators

The Gaussian Function is  $G(x, \sigma) = e^{-\frac{x^2}{2\sigma^2}}$

The Gaussian function is invariant to affine (linear) transformations.

$$T_a\{G(x, \sigma)\} = G(T_a\{x\}, T_a\{\sigma\})$$

Recall from lesson 2 we saw that  $x_r = x_c \frac{F}{z_c}$

The apparent size of an object is inversely proportional to its distance

A change in size (or scale) is a special case of an affine transform:

$$T_s\{G(x, \sigma)\} = G(T_s\{x\}, T_s\{\sigma\}) = G(sx, s\sigma)$$

This is just one of the many interesting properties of the Gaussian function.

## The Gaussian function in 1 and 2 D.

The Gaussian function is:  $G(x, \sigma) = e^{-\frac{x^2}{2\sigma^2}}$

2D Gaussian Kernel:  $G(x, y, \sigma) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} * e^{-\frac{y^2}{2\sigma^2}}$

The Gaussian derivatives are:

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

$$G_x(x, \sigma) = -\frac{x}{\sigma^2} G(x, \sigma) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

$$G_{xx}(x, \sigma) = \frac{x^2 - \sigma^2}{\sigma^4} G(x, \sigma) = \frac{x^2 - \sigma^2}{\sigma^4} e^{-\frac{x^2}{2\sigma^2}}$$

$$G_{xxx}(x, \sigma) = -\frac{x^3 - x\sigma^2}{\sigma^6} G(x, \sigma) = -\frac{x^3 - x\sigma^2}{\sigma^6} e^{-\frac{x^2}{2\sigma^2}}$$

En 2D: Derivative Filters:

$$G_x(x, y, \sigma) = -\frac{x}{\sigma^2} G(x, y, \sigma)$$

$$G_{xx}(x, y, \sigma) = \frac{x - \sigma^2}{\sigma^4} G(x, y, \sigma)$$

$$G_{xy}(x, y, \sigma) = \frac{xy}{\sigma^4} G(x, y, \sigma)$$

$$G_{xxx}(x, y, \sigma) = -\frac{x^3 - x\sigma^2}{\sigma^6} G(x, y, \sigma)$$

$$\nabla^2 G(x, y, \sigma) = G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma)$$

Diffusion Equation:  $\nabla^2 G_x(x, y, \sigma) = G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) = \frac{\partial G(y, y, \sigma)}{\partial \sigma}$

Scale property:  $G(x, y, \sqrt{2}\sigma) = G(x, y, \sigma) * G(x, y, \sigma)$

As a consequence:  $\nabla^2 G(x, y, \sigma) \approx G(x, y, \sigma_1) - G(x, y, \sigma_2)$

This typically requires  $\sigma_2 \geq 1.4 \sigma_1$  :  $\nabla^2 G(x, y, \sigma) \approx G(x, y, \sigma) - G(x, y, \sqrt{2}\sigma)$

Thus it is common to use:  $\nabla^2 G(x, y, \sigma) \approx G(x, y, \sigma) - G(x, y, \sigma) * G(x, y, \sigma)$

## Gaussian Digital Filters

Computers represent image as 2D sampled digitized signals. Because they are sampled, processing requires convolution with a sampled filter.

To obtain a 2D digital Gaussian filter we must sample the function at a rate of  $\Delta x$ , and  $\Delta y$  over some finite support of radius  $R$ .

$$G(x, \sigma)$$

Because we can control scale with  $\sigma$ , we can set the sample size to  $\Delta x = \Delta y = 1$ .

This gives a sampled function

$$G(n, \sigma) = e^{-\frac{n^2}{2\sigma^2}}$$

To represent this in a computer we must also specify the finite support (number of samples),  $N$ .

We set  $N = 2R + 1$ .

This gives us 2 parameters to control:

- 1) The scale of the Gaussian  $\sigma/\Delta x$
- 2) the size of the support  $N = 2R+1$

Truncating a function to a finite support is equivalent to multiply by a window  $W_N(n)$

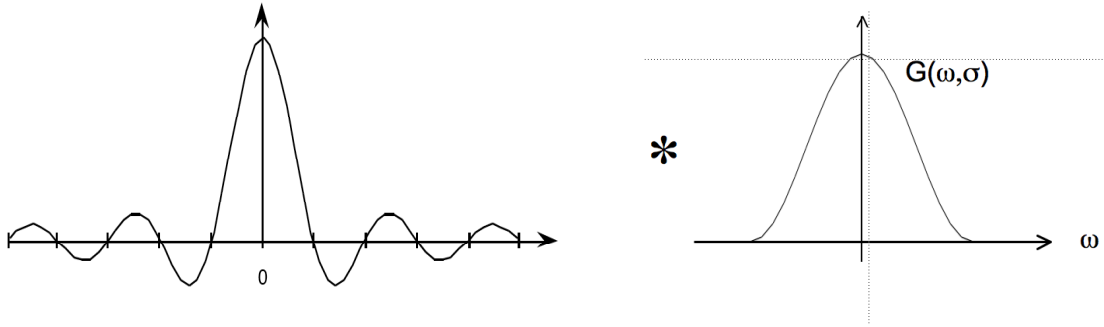
When we limit  $G(x, \sigma)$  to a finite support, we multiply by a window

$$G(n, \sigma) = G(n, \sigma) \cdot w_N(n) \text{ where } w_N(n) = \begin{cases} 1 & \text{for } -R \leq n \leq R \\ 0 & \text{otherwise} \end{cases}$$

Multiplying by a finite window is equivalent to convolving with the Fourier transform of the finite window:

$$F\{G(n, \sigma) \cdot w_N(n)\} = G(\omega, \sigma) * W_N(\omega)$$

$$\text{where } W_N(\omega) = \frac{\sin(\omega N/2)}{\sin(\omega/2)} \quad \text{and} \quad G(\omega, \sigma) = \sqrt{2\pi} \sigma e^{-\frac{1}{2}\sigma^2\omega^2}$$



For  $N < 7$ , the ripples in  $WN(\omega)$  dominate the spectrum and corrupt the resulting Gaussian.

At  $N=7$  the effect is tolerable but significant.

At  $N \geq 9$  the effect becomes minimal

In addition for  $\sigma/\Delta x < 1$ , the phenomenon of aliasing folds a significant amount of energy at the nyquist frequency, corrupting the quality (and the invariance) of the Gaussian function.

Finally, it is necessary to assure that the "gain" of the Gaussian filter is 1. This can be assured by normalizing so that the sum of the coefficients is 1. If the Gaussian were infinite in extent, then

$$\sum_{x=-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}} = \sqrt{2\pi}\sigma$$

However, because we truncate the Gaussian to an extent  $n \pm R$ , we must calculate the sum of the coefficients, A:

$$A = \sum_{n=-R}^R e^{-\frac{n^2}{2\sigma^2}}$$

For the 2D Sampled Gaussian  $G(i, j, \sigma) = e^{-\frac{i^2+j^2}{2\sigma^2}}$

(Attention:  $i, j$  are INTEGERS, not complex numbers!)

2D Gaussian is separable:  $G(i, j, \sigma) = \frac{1}{A} e^{-\frac{(i^2+j^2)}{2\sigma^2}} = e^{-\frac{i^2}{2\sigma^2}} * e^{-\frac{j^2}{2\sigma^2}}$

The normalization factor  $A = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} e^{-\frac{(i^2+j^2)}{2\sigma^2}} \approx 2\pi\sigma$

This factor will be slightly smaller if the Gaussian is evaluated over a finite window  $W_N(i,j)$  of size  $N \times N$  pixels.

This factor can be determined from the filter, and can be ignored in discussing the properties of the Gaussian.

The Gaussian filter is thus normalized by dividing by  $A$  to give a unit gain Receptive Field.

$$G(i, j, \sigma) = \frac{1}{A} e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

### Properties of Gaussian Receptive Fields (or Wavelets)

Separability:

Continuous Gaussian Kernel:  $G(x, y, \sigma) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} * e^{-\frac{y^2}{2\sigma^2}}$

Sampled Gaussian:  $W_N(i, j) \cdot G(i, j, \sigma) = W_N(i, j) \cdot e^{-\frac{(i^2+j^2)}{2\sigma^2}} = (W_N(i) \cdot e^{-\frac{i^2}{2\sigma^2}}) * (W_N(j) \cdot e^{-\frac{j^2}{2\sigma^2}})$

Scale property:  $G(x, y, \sqrt{2}\sigma) = G(x, y, \sigma) * G(x, y, \sigma)$

Discrete form:  $G(i, j, \sqrt{2}\sigma) = G(i, j, \sigma) * G(i, j, \sigma)$

Derivative Filters:

$$G_x(i, j, \sigma) = -\frac{i}{\sigma^2} G(i, j, \sigma)$$

$$G_{xx}(i, j, \sigma) = \frac{i - \sigma^2}{\sigma^4} G(i, j, \sigma)$$

$$G_{xy}(i, j, \sigma) = \frac{ij}{\sigma^4} G(i, j, \sigma)$$

$$G_{xxx}(i, j, \sigma) = -\frac{i^3 - i\sigma^2}{\sigma^6} G(i, j, \sigma)$$

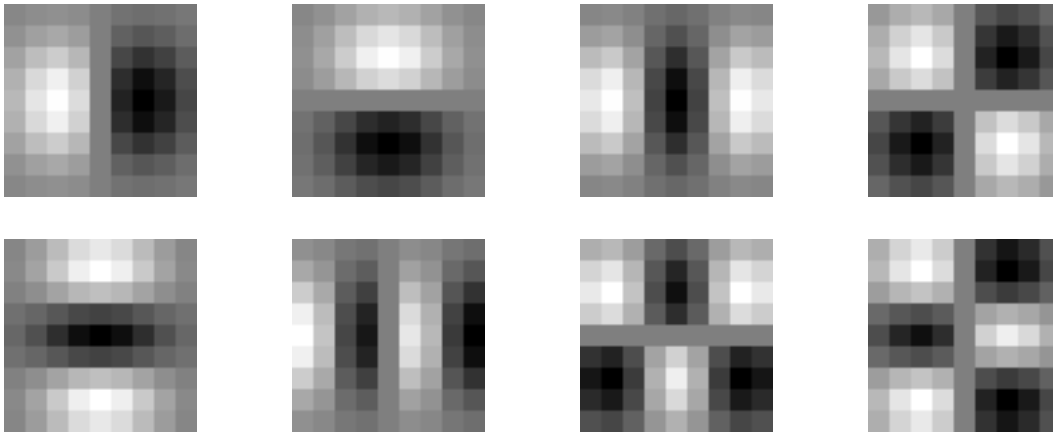
$$\nabla^2 G_x(i, j, \sigma) = G_{xx}(i, j, \sigma) + G_{yy}(i, j, \sigma)$$

Scale property:  $G(i, j, \sqrt{2}\sigma) = G(i, j, \sigma) * G(i, j, \sigma)$

Diffusion Equation:  $\nabla^2 G_x(i, j, \sigma) = G_{xx}(i, j, \sigma) + G_{yy}(i, j, \sigma) = \frac{\partial G(i, j, \sigma)}{\partial \sigma}$

Thus it is common to use:  $\nabla^2 G(i, j, \sigma) \approx G(i, j, \sigma) - G(i, j, \sigma) * G(i, j, \sigma)$

We can use these sampled functions to create a basis set of receptive fields:



The Gaussian receptive fields  $G_x, G_y, G_{xx}, G_{xy}, G_{yy}, G_{xxx}, G_{xxy}, G_{xyy}, G_{yyy}$ .

Note that there is only one parameter:  $\sigma$ . This determines the limit of the resolution for the position of a contrast point.

These can be used to compute a local description known as the "local jet".

$$\vec{L}(i, j, \sigma) = P * \vec{G}(i, j, \sigma)$$

The Local jet can be steered to a normalization angle  $\theta$ .

Steerability:

$$G_{\theta}^1(x, y, \sigma) = \cos(\theta) \cdot G_x(x, y, \sigma) + \sin(\theta) \cdot G_y(x, y, \sigma)$$

$$G_{\theta}^2(x, y, \sigma) = \cos(\theta)^2 G_{xx}(x, y, \sigma) + \cos(\theta)\sin(\theta)G_{xy}(x, y, \sigma) + \sin(\theta)^2 G_{yy}(x, y, \sigma)$$

$$G_{\theta}^3(x, y, \sigma) = \cos(\theta)^3 G_{xxx}(x, y, \sigma) + \cos(\theta)^2 \sin(\theta)G_{xxy}(x, y, \sigma) + \cos(\theta)\sin(\theta)^2 G_{xyy}(x, y, \sigma) + \sin(\theta)^3 G_{yyy}(x, y, \sigma)$$

Note the scale parameter  $\sigma$  determines the "resolution" of the derivatives.

You MUST specify  $\sigma$ . The smallest  $\sigma$  is not always the best.

Many computer vision algorithms give unpredictable results because the researchers forget to specify the scale  $\sigma$  at which the algorithm was validated.

## Using the Gaussian to compute image derivatives

Consider an image:  $P(i, j)$ ,

The Gradient  $\vec{\nabla}P(i, j)$  is calculated by  $P(i, j) * \vec{\nabla}G(i, j, \sigma)$

where 
$$\vec{\nabla}G(i, j, \sigma) = \begin{pmatrix} G_x(i, j, \sigma) \\ G_y(i, j, \sigma) \end{pmatrix}$$

Gradient: 
$$\vec{\nabla}P(i, j) \approx \vec{\nabla}(P * G(i, j, \sigma)) = P * \vec{\nabla}G(i, j, \sigma) = \begin{pmatrix} P * G_x(i, j, \sigma) \\ P * G_y(i, j, \sigma) \end{pmatrix}$$

Laplacien: 
$$\nabla^2 P(i, j) = P * \nabla^2 G(i, j, \sigma) = \begin{pmatrix} P * \nabla^2 G(i, j, \sigma) \\ P * \nabla^2 G(i, j, \sigma) \end{pmatrix} \approx P * \nabla^2 G(i, j, \sigma_1) - P * \nabla^2 G(i, j, \sigma_2)$$

where typically 
$$\frac{\sigma_1}{\sigma_2} \geq \sqrt{2}$$

### Steerability of Gaussian Derivatives.

For each pixel, one can calculate the orientation of maximal gradient. This orientation is equivariant with rotation. One can use this as an "intrinsic" orientation to normalize the receptive fields at any point in the image.

Local orientation: 
$$\theta_i(x, y, \sigma) = \text{Tan}^{-1}\left(\frac{G_y \cdot P(x, y, \sigma)}{G_x \cdot P(x, y, \sigma)}\right)$$

It is possible to synthesize an oriented derivative at any point as a weighted sum of derivatives in perpendicular directions. The weights are given by sine and cosine functions.

$$G_1^\theta(x, y, \sigma) = \cos(\theta) \cdot G_x(x, y, \sigma) + \sin(\theta) \cdot G_y(x, y, \sigma)$$

By steering the Gaussian response to the local orientation, we obtain an "invariant" measure of local contrast:

Thus:

$$P_1(x, y; \theta, \sigma) = \text{Cos}(\theta) \langle G_x(x, y; \sigma) \cdot P(x, y) \rangle + \text{Sin}(\theta) \langle G_y(x, y; \sigma) \cdot P(x, y) \rangle$$

a similar measure can be obtained for the second and third derivatives:

$$G_2^\theta(x, y, \sigma) = \text{Cos}(\theta)^2 G_{xx}(x, y, \sigma) + \text{Cos}(\theta)\text{Sin}(\theta) G_{xy}(x, y, \sigma) + \text{Sin}(\theta)^2 G_{yy}(x, y, \sigma)$$

$$G_3^\theta(x, y, \sigma) = \text{Cos}(\theta)^3 G_{xxx}(x, y, \sigma) + \text{Cos}(\theta)^2 \text{Sin}(\theta) G_{xxy}(x, y, \sigma) + \text{Cos}(\theta)\text{Sin}(\theta)^2 G_{xyy}(x, y, \sigma) + \text{Sin}(\theta)^3 G_{yyy}(x, y, \sigma)$$