Through The Eyes of Gabor Filter

Anuj shah (Exploring Neurons) · Follow

5 min read · Jun 17, 2018

(▶) Listen   (⬆) Share

The Gabor filter, named after <u>Dennis Gabor</u>, is a <u>linear filter</u> used in myriad image processing applications for edge detection, texture analysis, feature extraction, etc. The characteristics of certain cells in the visual cortex of some mammals can be approximated by these filters. These filters have been shown to possess optimal localization properties in both spatial and frequency domains and thus are well-suited for texture segmentation problems. Gabor filters are special classes of bandpass filters, i.e., they allow a certain 'band' of frequencies and reject the others. A Gabor filter can be viewed as a sinusoidal signal of particular frequency and orientation, modulated by a Gaussian wave. One such 2D Gabor filter is shown in Figure 1.
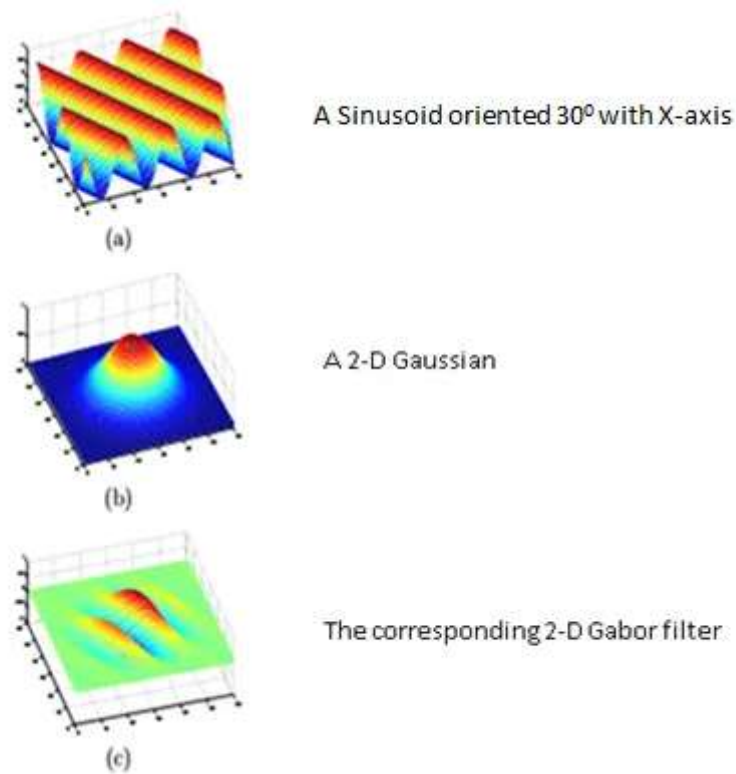
Figure 1: A 2-D Gabor filter obtained by modulating the sine wave with a Gaussian

From the above figure, we can notice that the sinusoid has been spatially localized. In practice to analyze texture or obtain feature from an image, a bank of Gabor filter with a number of different orientation are used.

Consider an example of an elephant that has patterns or stripes on its skin at different orientations. Now to highlight or extract out all those patterns we are going to use a bank of 16 Gabor filters at an orientation of 11.250 (i.e. if the first filter is at 00, then the second will be at 11.250, the third will be at 22.50, and so on.).Figure 2 shows all the filter banks of 16 filters
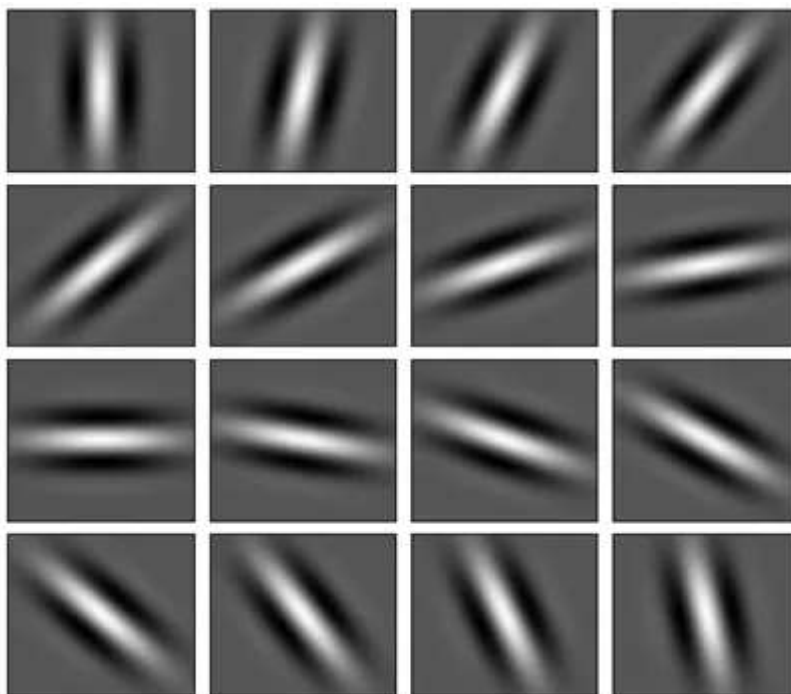
Figure 2: A bank of 16 Gabor filters oriented at an angle of 11.250 (i.e. if the first filter is at 00, then the second will be at 11.250, the third will be at 22.50, and so on.)

When the input image is convolved with all the Gabor filters the patterns are easily highlighted as shown in Figure 3. When a Gabor filter is applied to an image, it gives the highest response at edges and at points where texture changes. When we say that a filter responds to a particular feature, we mean that the filter has a distinguishing value at the spatial location of that feature.
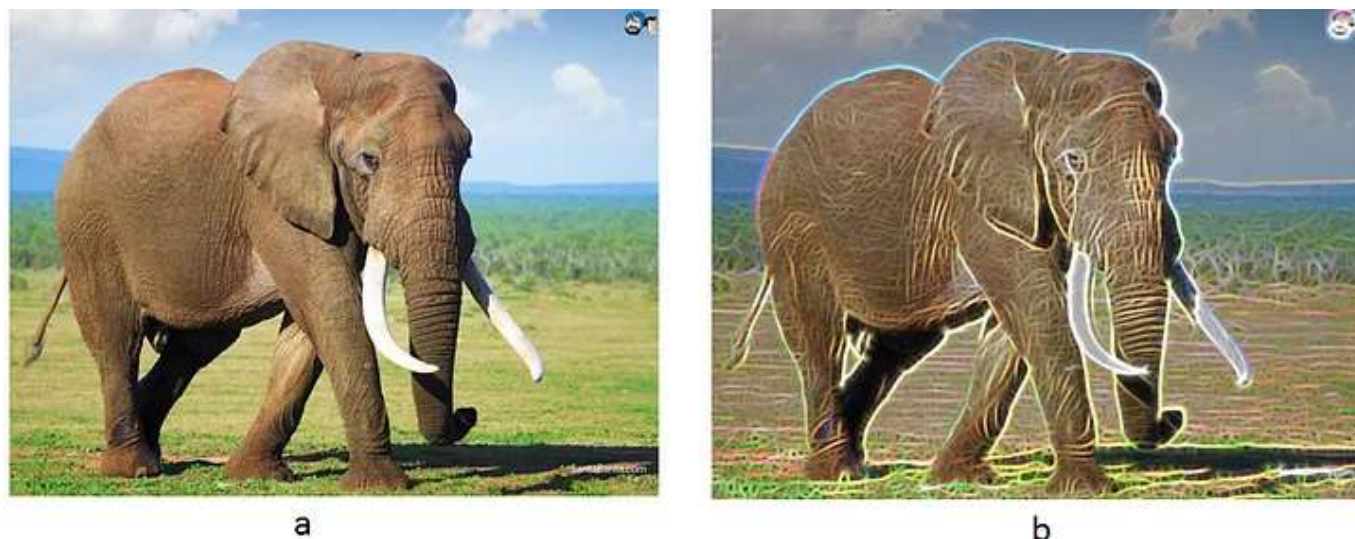

Figure 3: (a) The input image of an Elephant and (b) the output image after passing it through the Gabor filter bank

To better understand what each filter detects in the input image, consider a simple white circle on black background. When this image is passed through each filter in the filter bank, the edge of the circle which gets detected is the edge oriented at an angle at which the Gabor filter is oriented. Figure 4 clearly shows this.
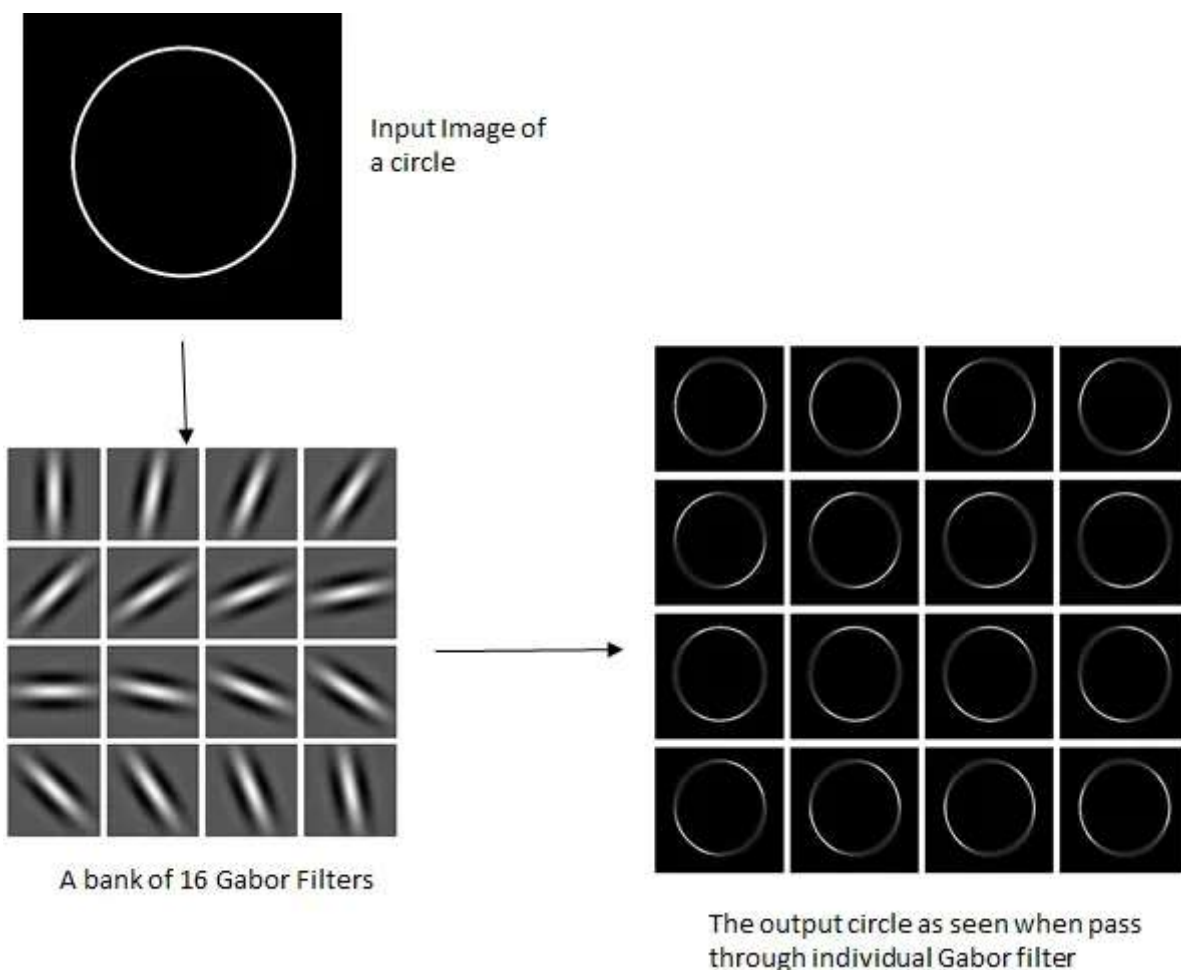

Input Image of a circle

A bank of 16 Gabor Filters

The output circle as seen when pass through individual Gabor filter

Figure 4: The corresponding oriented edge features being detected when passed through individual oriented Gabor filters

Different parameters that control the shape & size of the 2D Gabor filter:

There are certain parameters that control how the Gabor filter will be and which features will it respond to. A 2D Gabor filter can be viewed as a sinusoidal signal of particular frequency and orientation, modulated by a Gaussian wave. The filter has a real and an imaginary component representing orthogonal directions. The two components may be formed into a complex number or used individually. The equations are shown below:

Complex

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

Real

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

Imaginary

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

where

$$x' = x\cos\theta + y\sin\theta$$

and

$$y' = -x\sin\theta + y\cos\theta$$

In the above equation,

λ — Wavelength of the sinusoidal component.

θ — The orientation of the normal to the parallel stripes of the Gabor function.

Ψ — The phase offset of the sinusoidal function.

σ — The sigma/standard deviation of the Gaussian envelope

γ — The spatial aspect ratio and specifies the ellipticity of the support of the Gabor function.

The above-mentioned five parameters control the shape and size of the Gabor function. The role of each parameter is discussed in detail below. For an illustration of the effects of parameters, the following values are chosen as starting point:
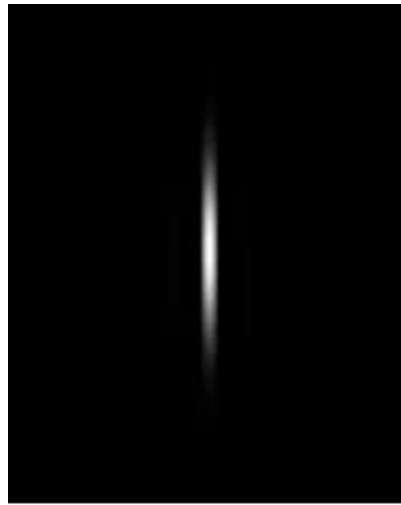
Lambda (λ) = 30; Theta (θ) = 00

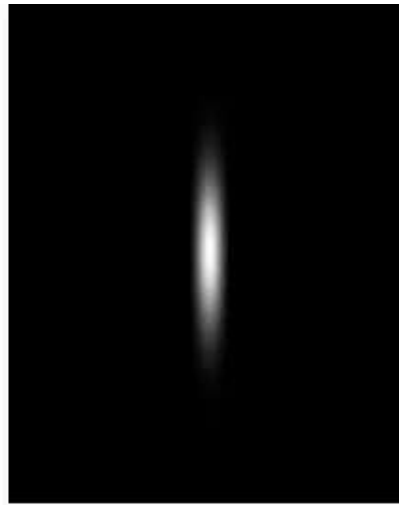Gamma (γ) = 0.25; Sigma (σ) = 10
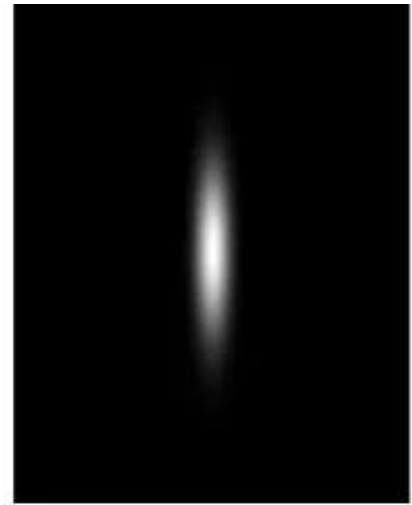
Psi (Ψ) = 0

Lambda (λ):

The wavelength governs the width of the strips of the Gabor function. Increasing the wavelength produces thicker stripes and decreasing the wavelength produces thinner stripes. Keeping other parameters unchanged and changing the lambda to 60 and 100, the stripes get thicker.

Lambda (λ) = 30   Lambda (λ) = 60   Lambda (λ) = 100

Figure 1: Keeping other parameters unchanged (Θ = 00, γ = 0.25, σ = 10, Ψ = 0), and on changing the lambda from 30 to 60 and 100 the Gabor function gets thicker

Theta (Θ):

The theta controls the orientation of the Gabor function. The zero-degree theta corresponds to the vertical position of the Gabor function.

Figure 2: Keeping other parameters unchanged (λ = 30, γ = 0.25, σ = 10, Ψ = 0), and on changing the theta from 00 to 450 and 900 the Gabor function rotates.

Gamma (γ):

The aspect ratio or gamma controls the height of the Gabor function. For a very high aspect ratio the height becomes very small and for a very small gamma value the height becomes quite large. On increasing the value of gamma to 0.5 and 0.75, keeping other parameters unchanged, the height of the Gabor function reduces.
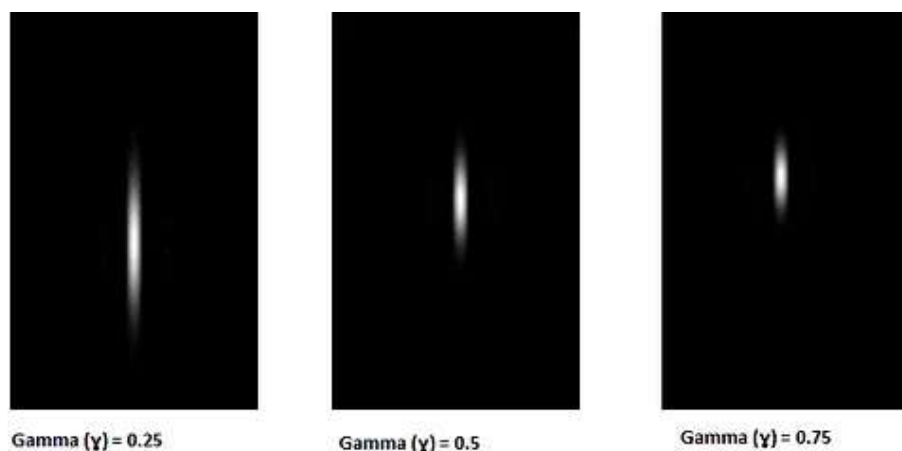

Figure 3: Keeping other parameters unchanged (λ = 30, Θ = 00, σ = 10, Ψ = 0), and on changing the gamma from 0.25 to 0.5 and 0.75, the Gabor function gets shorter.

Sigma (σ):

The bandwidth or sigma controls the overall size of the Gabor envelope. For larger bandwidth the envelope increase allowing more stripes and with small bandwidth

the envelope tightens. On increasing the sigma to 30 and 45, the number of stripes in the Gabor function increases.
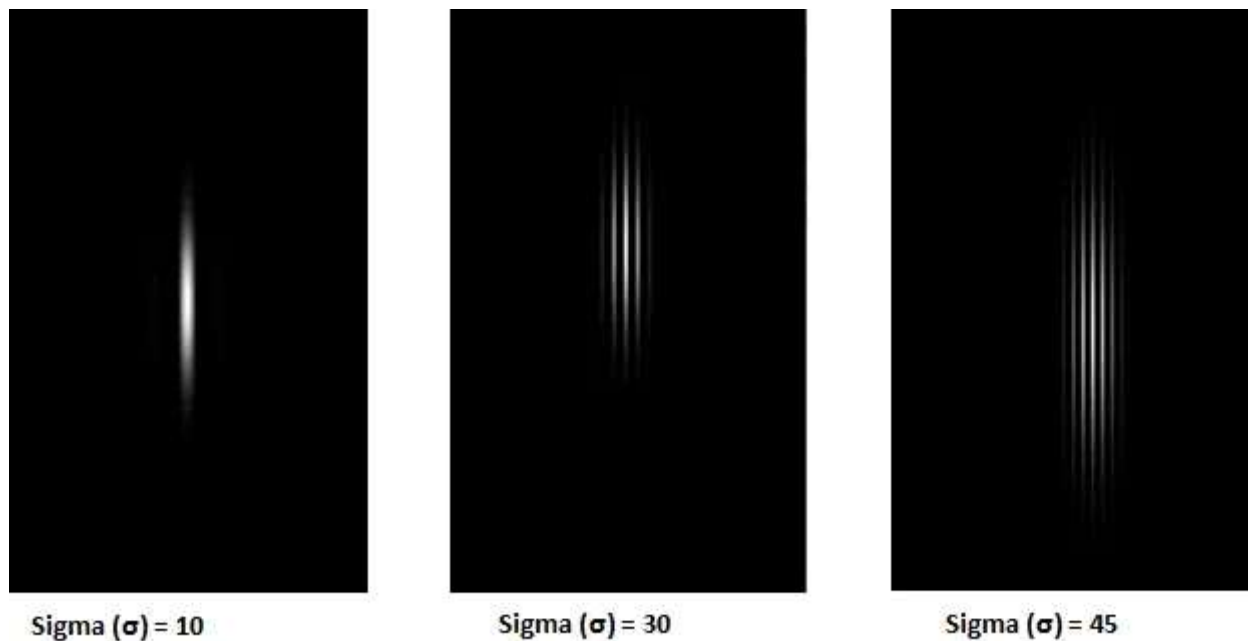

Figure 4: Keeping other parameters unchanged (λ = 30, Θ = 00 γ = 0.25, Ψ = 0), and on changing the sigma from 10 to 30 and 45 the number of stripes in the Gabor function increases.

In OpenCV Python, the following is the structure of the function that is used to create a Gabor kernel.

```
cv2.getGaborKernel(ksize, sigma, theta, lambda, gamma, psi, ktype)
```

ksize is the size of the Gabor kernel. If ksize = (a, b), we then have a Gabor kernel of size a x b pixels. As with many other convolution kernels, ksize is preferably odd and the kernel is a square (just for the sake of uniformity).

sigma is the standard deviation of the Gaussian function used in the Gabor filter.

theta is the orientation of the normal to the parallel stripes of the Gabor function.

lambda is the wavelength of the sinusoidal factor in the above equation.

gamma is the spatial aspect ratio.

psi is the phase offset.

ktype indicates the type and range of values that each pixel in the Gabor kernel can hold.

Theta(Orientation): governs the rotation of the Gabor envelope

Gamma(Aspect Ratio): controls the height of the Gabor filter. If the gamma value is the height of Gabor reduces and if the gamma value is small the height of Gabor increases.

Sigma (Bandwidth): It controls the overall size of the Gabor envelope. for larger bandwidth the envelope increase allowing more stripes and with smaller bandwidth the envelope tightens.

Psi: is the phase offset of the sinusoid.

That's it for now. If you find my articles helpful and wish to support them — Buy me a Coffee

Machine Learning   Gabor Filter   Iris Recognition   Texture Recognition

Gaussian Kernel

Follow   ◯

Written by Anuj shah (Exploring Neurons)