**ENHANCEMENT ON FINGERPRINT AND OTHER IMAGES USING GABOR FILTER**

BARATH ARAVIND K (36821010), YOGESHWARAN A (36821026)

**ABSTRACT**

This study provides an overview of general Gabor filter and its applications. Gabor filters have succeeded in many applications, e.g., face detection and recognition, iris recognition and fingerprint matching where Gabor feature-based methods are always considered among the top performers.

**1. Introduction**

In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for texture analysis, which essentially means that it analyses whether there is any specific frequency content in the image in specific directions in a localized region around the point or region of analysis. Frequency and orientation representations of Gabor filters are claimed by many contemporary vision scientists to be similar to those of the human visual system. They have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2-D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave (see Gabor transform)

**2. Definiton**

There are certain parameters that controls how 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.



In OpenCV Python, 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(σ)** is the standard deviation of the Gaussian function used in the Gabor filter.

**Theta(Ө)** is the orientation of the normal to the parallel stripes of the Gabor function.

**Lambda**(**λ)** is the wavelength of the sinusoidal factor in the above equation.

**Gamma(ɣ)** is the spatial aspect ratio.

**Psi(Ψ)** is the phase offset.

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

**2.1 Ksize**

On varying ksize, the size of the convolution kernel varies. In the code above we modify the parameter ksize, while keeping the kernel square and of an odd size. We observe that there is no effect of the size of the convolution kernel on the output image. This also implies that the convolution kernel is scale invariant, since scaling the kernel’s size is analogous to scaling the size of the image. Here are a few results with varying ksize. For all the following images, sigma = 4.0, theta = 0, lambda = 10.0, gamma = 0.5, psi = 0, and ktype = cv2.CV\_32F (i.e., each pixel of the convolution kernel holds a weight which is a 32-bit floating point number).

 Input image ksize = 31x31

ksize = 51x51

**2.2. Sigma**

 This parameter controls the width of the Gaussian envelope used in the Gabor kernel. Here are a few results obtained by varying this parameter.

 Sigma = 2 sigma = 4

sigma = 6

**2.3. Theta**

This is perhaps one of the most important parameters of the Gabor filter. This parameter decides what kind of features the filter responds to. For example, giving theta a value of zero means that the filter is responsive only to horizontal features only. So, in order to obtain features at various angles in an image, we divide the interval between 0 and 180 into 16 equal parts, and compute a Gabor kernel for each value of theta thus obtained. Note that we’ve chosen 16 just because it was the default value in the OpenCV implementation. These parameter values could be modified to suit specific purposes. Following are the results of varying theta on the above input image.

 theta = 45 theta = 135

**2.4. Lambda**

 Here’s the variation with lambda (theta is set to zero).

 lambda = 8 lambda = 10

lambda = 12

**2.5. Gamma**

Gamma controls the ellipticity of the gaussian. When gamma = 1, the gaussian envelope is circular.

gamma = 0.3 gamma = 1.0

**2.6. Psi**

 This parameter controls the phase offset.

 psi = 0 psi = 90

**3. Create filter function as Code:**

def create\_gaborfilter():

filters = []

num\_filters = 16

ksize = 35 # The local area to evaluate

sigma = 3.0 # Larger Values produce more edges

lambda = 10.0

gamma = 2.0

psi = 0 # Offset value - lower generates cleaner results

for theta in np.arange(0, np.pi, np.pi / num\_filters): # Theta is the orientation for edge detection

kern = cv2.getGaborKernel((ksize, ksize), sigma, theta, lambd, gamma, psi, ktype=cv2.CV\_64F)

kern /= 1.0 \* kern.sum() # Brightness normalization

filters.append(kern)

return filters

**4. Sample input and output images:**

Output of enhanced img program.



Image after applying gabor filter.

**** Input for gabor filter.



Output after applying gabor filter

**5. Applications of 2-D Gabor filters in image processing:**

1. For document image processing, Gabor features are ideal for identifying the script of a word in a multilingual document.

2. Gabor filters with different frequencies and with orientations in different directions have been used to localize and extract text-only regions from complex document images (both grey and colour), since text is rich in high frequency components, whereas pictures are relatively smooth in nature.

3. It has also been applied for facial expression recognition.

4. Gabor filters have also been widely used in pattern analysis applications. For example, it has been used to study the directionality distribution inside the porous spongy trabecular bone in the spine.

5. The Gabor space is very useful in image processing applications such as optical character recognition, iris recognition and fingerprint recognition.

6. Relations between activations for a specific spatial location are very distinctive between objects in an image.

7. Important activations can be extracted from the Gabor space in order to create a sparse object representation.

**6. Conclusion**

The key focus of this work is on the enhancement of fingerprint images and other images. By using enhancement technique Gabor filter, the quality of input images greatly increased and highlights specific areas like drawing an outline which pave the way to extract fingerprint features i.e the minutiae points more accurately.

**7. References**

* A Literature survey of Gabor filter and its application by ASMAA KAFUO, SAMERA DIAF, AISHA GONIFEDA, ZAMZAM ALSHATOURI.
* Fingerprint Image Enhancement And It‟s Feature Extraction For Recognition by Pankaj Bhowmik, Kishore Bhowmik, Mohammad Nurul Azam, Mohammed Wahiduzzaman Rony.
* Python.org
* Digital Image Processing by Gonzalez Rafael C.
* Principles of Digital Image Processing by Wilhelm Burger and Mark J. Burge.