

An Investigation and Implementation of Biometrics Image Processing Techniques

Deepak Srinivasan

TJHSST Computer Systems Lab 2005 - 2006

Methodology (of Gabor Filter)

The Gabor filter is based on the equation $\exp(-0.5((x')^2/(\delta_x)^2 + (y')^2/(\delta_y)^2)) * \cos(2\pi * f * x')$ where $x' = x * \sin(\theta) + y * \cos(\theta)$ and $y' = x * \cos(\theta) - y * \sin(\theta)$, δ_x = standard deviation along the x-axis, and δ_y = standard deviation along the y-axis. The orientation of this filter changes what ridges and valleys actually are enhanced. By finding the angle of every pixel, the equation makes sure that all ridges of all angles are made clearer.

Using the equation above, one must create a Gabor kernel (its graph should look like similar to Figure 1), and then perform a convolution on that matrix. To find the standard deviations along the axes, one must find and use the deviation for a small specified area (around a 7x7). The frequency is the number of fingerprints between two points of some unit distance x. This unit distance was assumed to be about 10 pixels. Lastly, the angle (theta) was determined using the Sobel Edge Detector technique, which is also based on convolutions but is much simpler.

The goal of the Gabor filter is to use a matrix such as the one in Figure 1, and perform a convolution so that there is more contrast between the ridges and valleys (so that it can be seen easier) and noise can be removed.

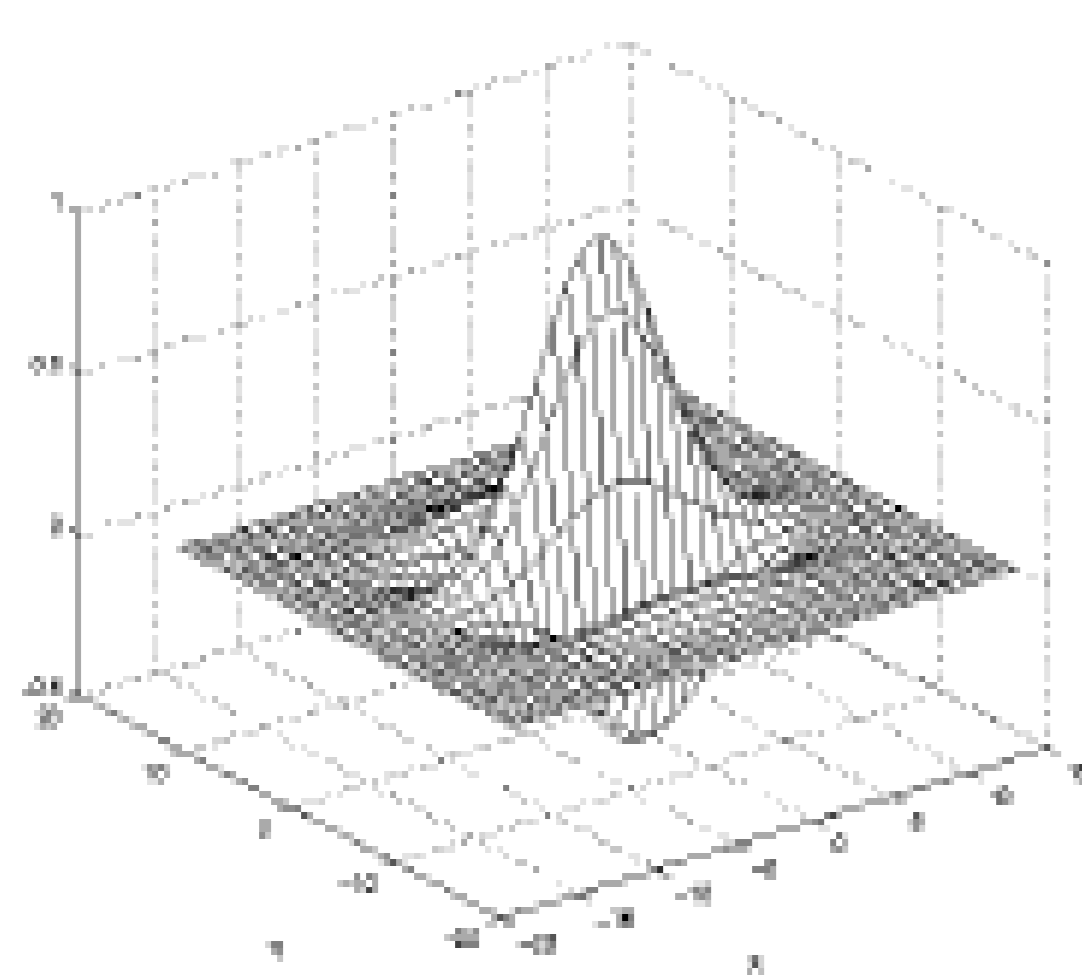


Figure 1

Methodology (of my Code)

The Gabor filter method was created using the programming language C. Most of the work dealt with performing calculations with various arrays, and the method malloc proved to be essential for the project.

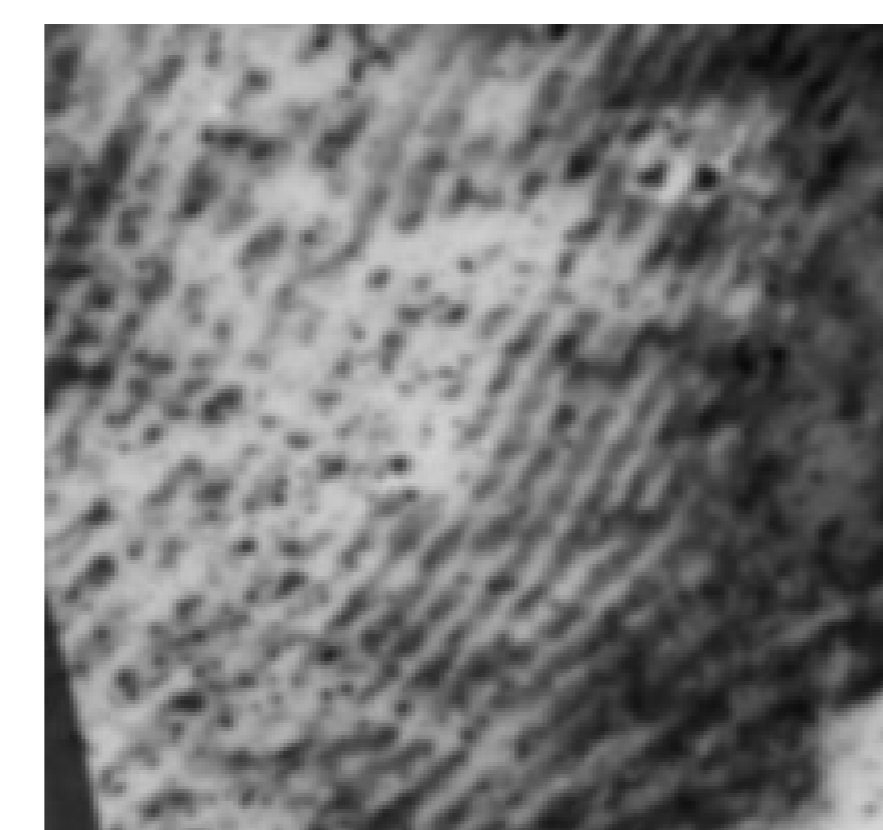
The testing interface used the programming languages of Visual Basic and C. The visual interface included two sliders, one for standard deviation and the other for orientation. The sliders were used to test various deviations and orientations, see which ones worked best, and see whether the product was working correctly.

Abstract

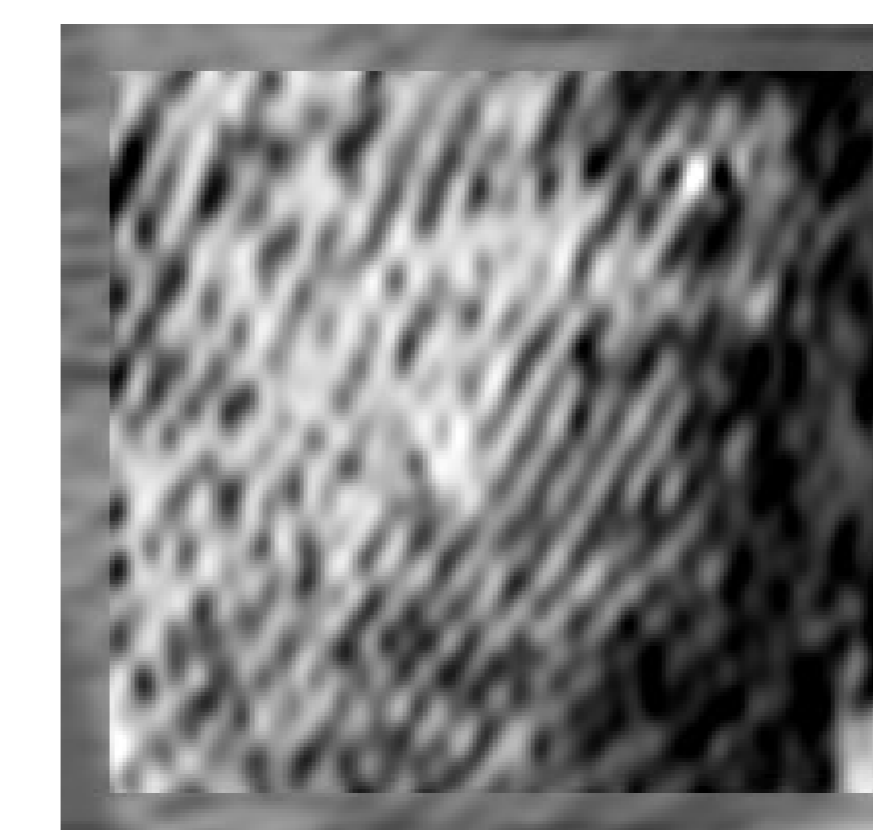
The goal of this project is to implement the image processing technique, the Gabor filter, with the programming languages of Visual Basic and C. First, a project interface was created to be able to quickly test the developing Gabor filter method. In the method, the Gabor filter equation was used to create a kernel matrix. After inputting constants for the many variables, a convolution was performed on the kernel matrix and an image was made. After several weeks, the Gabor filter was implemented properly and the project was successful.

Background

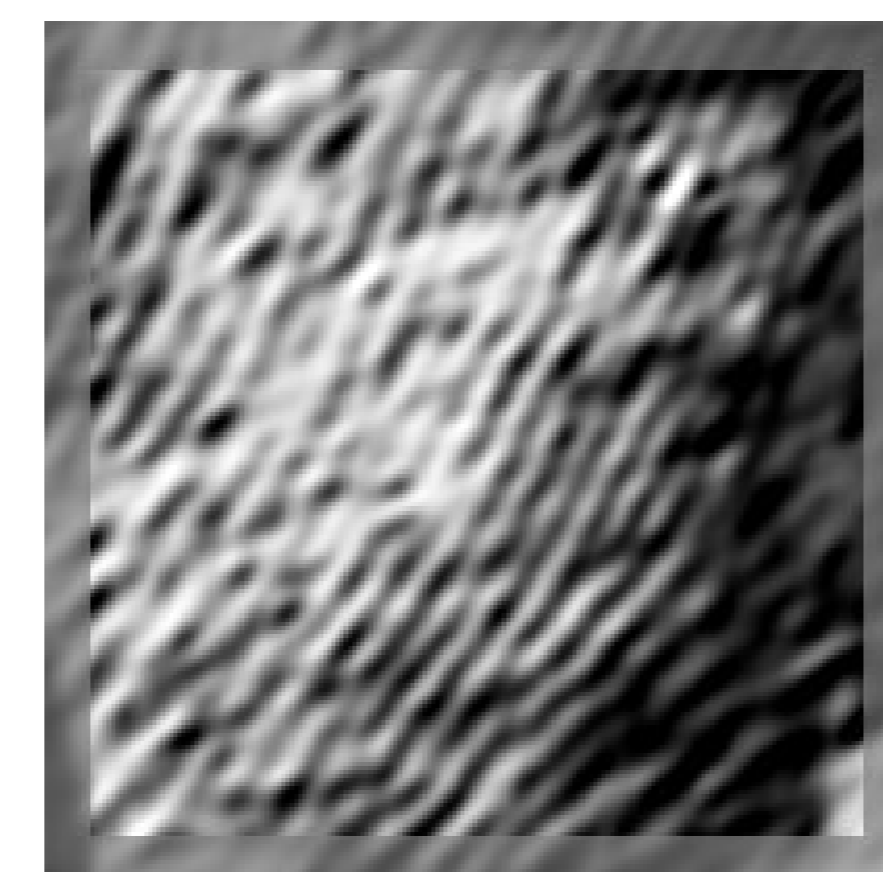
Fingerprint matches are often the critical piece of evidence that puts a criminal behind bars. So the identification's accuracy is important. In this modern age, fingerprint identification software supplements the examiner in this matching process. Many image processing techniques have not been created and work theoretically, but have not been tested in real world setting. That's why I eventually implemented the complex, but effective Gabor filter.



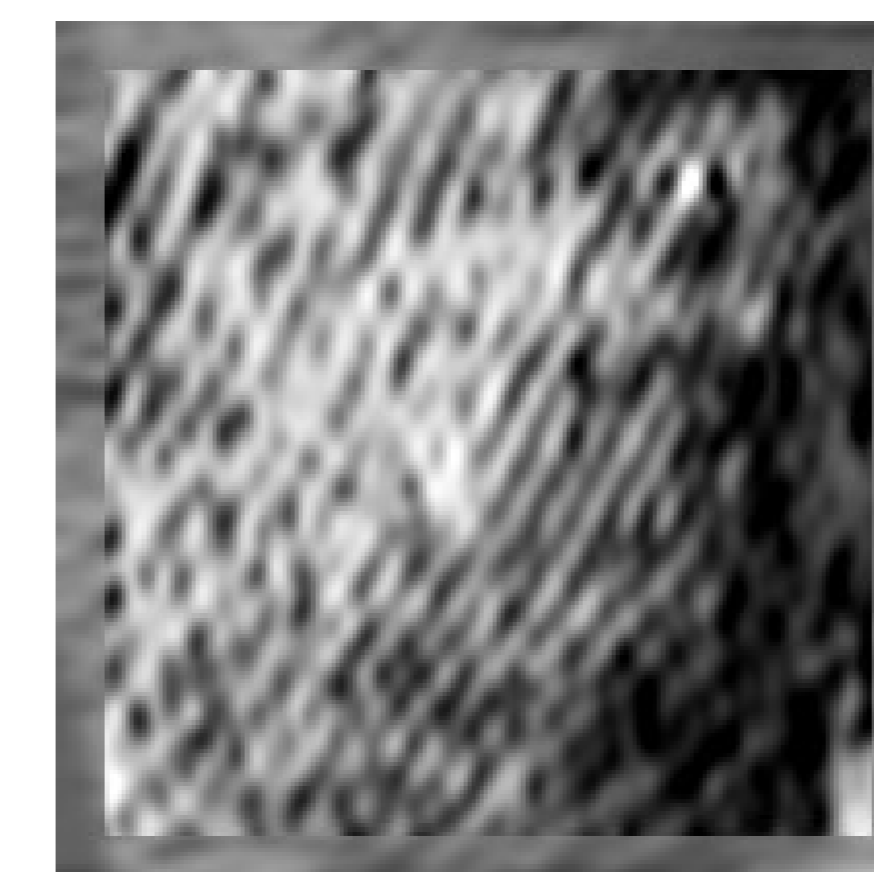
Left - Original Image



Right - Image after Gabor filter with Deviation 3 and an Orientation of 0°



Left - Image with a Deviation of 3 and an Orientation of 135°, Right - Image with a Deviation of 3 and an Orientation of 135°



Results

The project's result image did look like the proper Gabor filter image at the end of this project. The fingerprints' ridges and valleys were accentuated by creating a larger contrast. Also, the sliders for deviation and orientation worked correctly, changing the contrast of the ridges, and changing which ridges are accentuated.

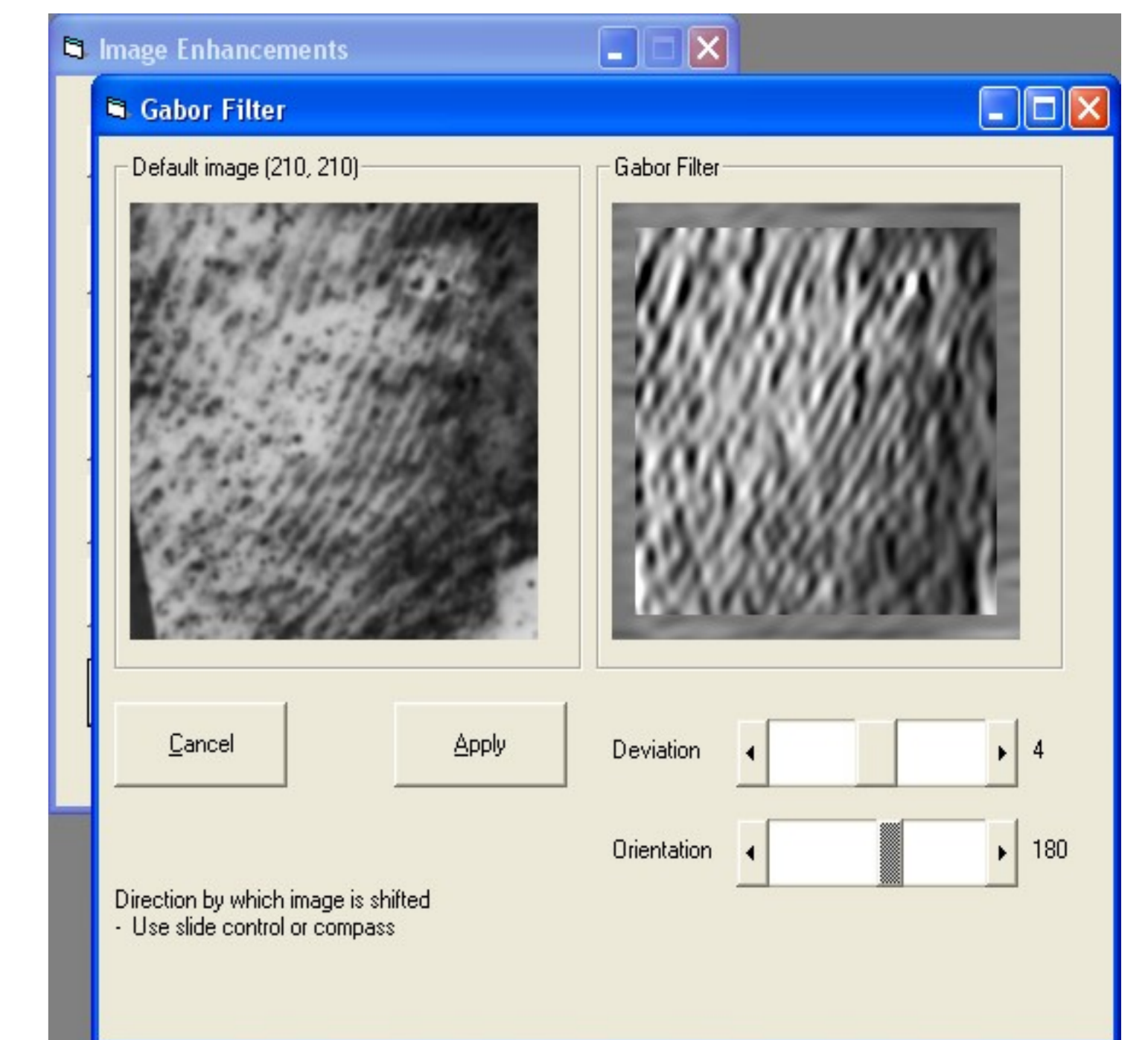


Figure 3

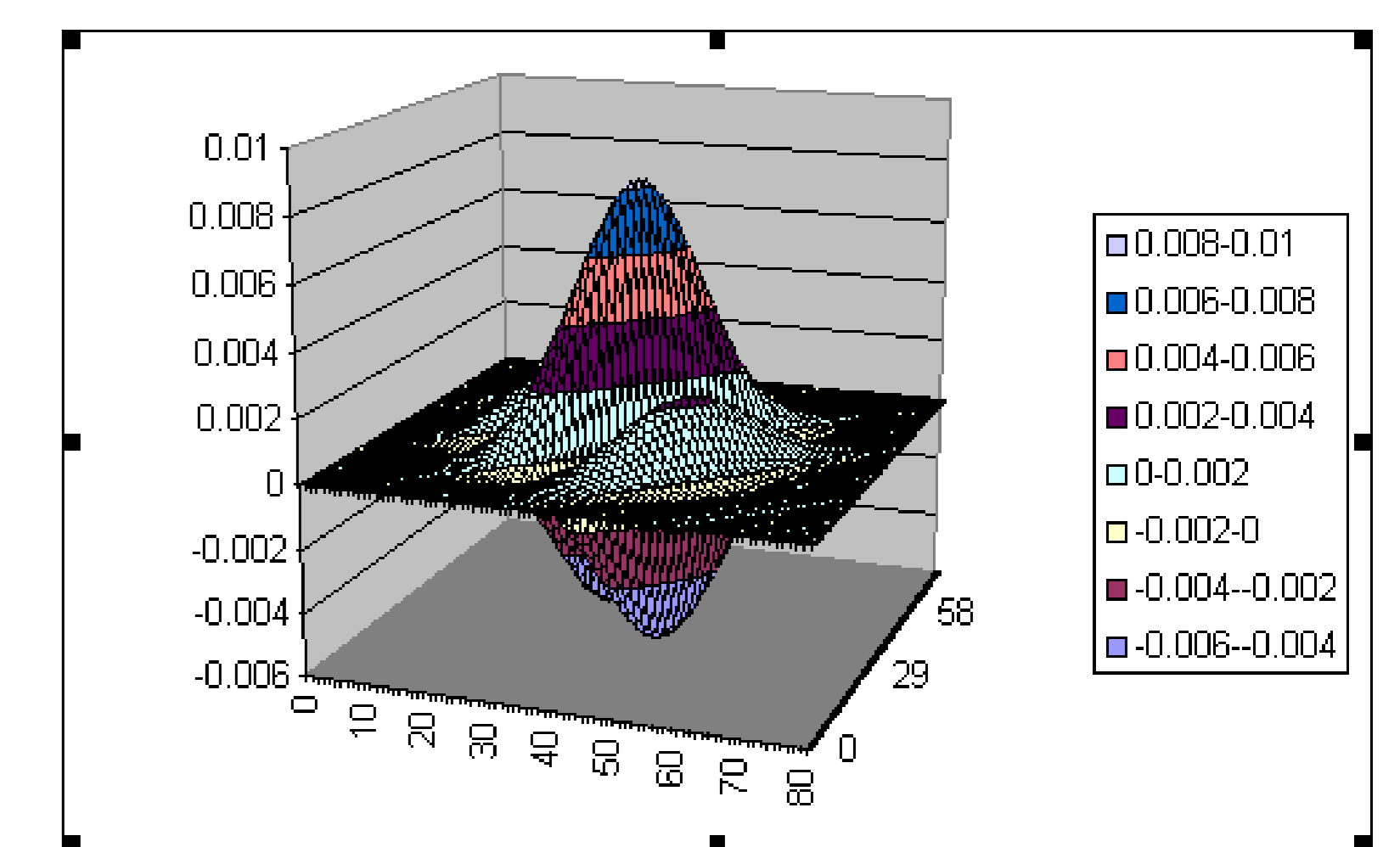


Figure 2

Conclusion

In the end, the Gabor filter algorithm did work successfully. The goal of implementing the theoretical concept of the Gabor filter into computer code was successful and this fact was confirmed by the creators of the Gabor filter technique.