

# CANNY FILTER APPLICATIONS AND IMPLEMENTATION WITH GRASS

Francesco PIROTTI \*, Antonio VETTORE \*

(1) CIRGEO – Interdepartment Center of Cartography, Photogrammetry, Remote Sensing and GIS -  
University of Padua – Viale dell'Università 16 – 35020 Legnaro (PD) – Email. [Cirgeo@unipd.it](mailto:Cirgeo@unipd.it)  
– [www.cirgeo.unipd.it](http://www.cirgeo.unipd.it)

**KEY WORDS:** Canny, Image Filtering, Image analysis, GRASS GIS

## **Summary:**

In this work an edge detecting filter using the Canny algorithm is implemented and applied to test images in order to assess its potential in detecting borders of elements. The filter will also be used on a real application in order to detect changes in image geometry. The role played by GRASS GIS is crucial because of its open source written in C language, its portability on different platforms and the ease of having new code added to implement new modules in a GIS environment.

In this publication we present therefore how the implementation of the filter is carried out and how it was used on the test images to check the accuracy of results. Finally it is also applied to a digital image of the side of Stromboli vulcano to check its deformations. From such image the Canny filter can extract the profile of the conical shape of the mountain and compare it in real time with the same profiles taken previously, detecting changes due to erosion or landslides. The result can be sent using internet technology to operators which interpret the activities recorded.

## INTRODUCTION

In image processing it is common to preprocess the image to permit the enhancement of certain parts in which the operator is interested. Edge extraction is often used to identify borders of elements, especially to segment the image in different zones which are well differentiated as grey values are concerned. If the elements in which one is interested have a well defined border it is good practice to apply an edge detecting filter to delimit such elements. An example offhand can be to detect different fields growing crop, or roads or desert dunes' movement. Edge detecting is especially helpful in cases where the different elements we want to separate have similar digital signatures but have a well defined border. In such cases the classical supervised image classification can have problems due to the similarity in their digital signatures whereas a border filter separates easily the elements from their neighbors.

The Canny Filter is different from the other classic edge detecting filters because it considers the directionality of the element and permits the personalization of the behavior of the filter by giving the value of parameters of the algorithm which permit changes in direction of the polarization of the filter and sensibility of the edge extraction. This becomes helpful to adapt the filter to the image characteristic, and permits therefore to optimize its use. On the other hand the filter is more complex in respect to the others which are used, and requires an operator which has knowledge on its application and which can optimize its use.

The following paper explains how a new module for inserting a Canny Filter module inside the GRASS structure in order to be able to use it as a tool. Both the programming phase and the application and testing phase will be described. The final application will be on a digital image of the side of Stromboli volcano and it is part of a project on monitoring deformations of volcanoes using images from commercial digital cameras. The filter in this case is used to extract the profile of the volcano identifying the skyline margin (figure 1) and it

is used therefore to compare it to previous profiles to check for changes in the skyline which might mean that deformation events are going on. Operators can be kept informed in real time of the changes of the skyline detected by the system, in order to take decisions regarding security.

## METHODS

First step was to program the module in ANSI C language, making use of the GRASS libraries for managing raster elements. Thanks to the existence of this code the user could concentrate on the implementation of the algorithm without having to deal with the problem of coding for raster procedures within GRASS.

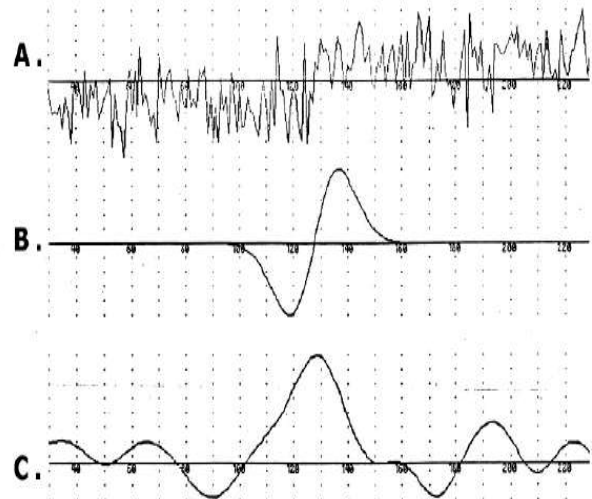


Image 1: A. Grey levels in an image. B. - The filtering kernel. C. - The resulting grey level transformation

Canny introduced an analytical approach to edge detecting in digital images. He made the hypothesis that a margin in one dimension can be represented like a step in a signal characterized by gaussian white noise with a certain standard variation ( $\sigma$ ). Canny filter assigns as edge the pixels which have a maximum gradient of grey value in an image after it has been processed with a gaussian filter with an appropriate  $\sigma$  value. This permits theoretically to model the white noise and thus to eliminate it before finding the derivative of the image. The maximum values of the first derivative or the zero points of the second derivative the operator

can spot margins. Calibrating the parameters of the Canny filter one can minimize the uncertainty of two characteristics: the directionality of the margin and the white noise of the signal. The axis of the direction of the margin can be chosen to focus in the directionality of the elements, while the white noise can be modeled by changing the standard deviation of the Gaussian curve and its shape [1]. In figure 1 the representation of how the filter works is shown: **a** – one dimensional representation of a signal with a margin **b** – an image edge detection filter **c** – the result of a convolved with b; the other formulas represent details of how the filter works. The Canny filter, compared to other edge detecting filters, is considered the most appropriate in the case of margins with a definite directionality and with a regular Gaussian white noise (Dmitrij 1999). In case the margin has different directionalities but still well defined ones, it can be applied more than once, and the results merged together for getting the best results. In our case in fact, the profile of the mountain in the image has one direction for the right slope and one for the left. There is also a noise in the edge between the slope and the background sky, as it is clearly seen in image 5.

To let the user give its input parameters to the filter, a dialog window has been set up for GRASS (image 2); it asks for the direction of the two polarities in degrees and for the standard deviation of the Gauss model for the white noise as well as a threshold for the edge/no-edge decision. This information is necessary to calibrate the filter for optimizing its effects on the image which will be filtered. The parameters which can be changed from their default value are: standard deviation of the two axes of the filter (X e Y), direction of polarity of the two axes (in degrees) and threshold for the decision edge / not-edge. It is also possible to change the dimension of the filter window, even if, as will be explained later, this will usually bring a worse result in the filter performance. The advantage in the Canny filter is the possibility to make the algorithm optimal for the image which is being processed, trying the results and changing the parameters to find the best combination. To find how the filter works on

edges, two artificial images and “Lena” will be processed with different parameters, and the results will be shown and commented. Finally the filter will also be applied to a real image taken of the side of the Stromboli Vulcano (image 5) also with different parameters to find the best combination for producing a continuous margin of the slopes. Comparing the edge extracted with the actual edge will permit and estimation of accuracy.

GRASS 5.3 version was used for applying the Canny Filter module and for visualization and accuracy estimation. Module implementation was done in Linux environment with the support of Qt designer. The application of the Canny Filter to the edge of the Stromboli Vulcano is part of a bigger project to monitor the behavior of the side of this vulcano in order to set up a real time alarm system helpful in case of eruptions and landslides, which so often happen in this mountain.



Image 2: GRASS module's dialog window

## RESULTS

The first step was to apply the filter to an image with a well defined shape, which in this case was an arrow. This shape was chosen because it contains elements with an horizontal, vertical and diagonal direction. The result will therefore bring to light the difference on how the filter works direction-wise. Another characteristic of the artificial image is the fuzzy edges, which will help to

assess the behavior of the filter in cases of not-so-well defined edges.

The other image used is the classic Lena image (see image 4) which is normally used to test image-processing algorithms. This image contains a lot of interesting elements which put to test the algorithm, such as zones with very fuzzy edges, or shapes with complex edges, and a lot shapes with different directions.

As is seen in all the images processed, the filter produces three intermediate images before the final product with edges. The first two products are the edge results of the two polarities, the third product is the normalized average of the first two and the final image applies a threshold to the third, defining a two bit image with edge / not-edge.

Using these two images the sensibility of the filter is tested. With the first image the directional behavior of the filter is tested. The original image (image 3a) clearly has horizontal, vertical and 45° elements with blurry edges. In image 3b it is clear that the horizontal component was absolutely unrecognized, whereas image 3c shows that the vertical component was unrecognized. These results were obtained using the default values of the filter: 0° of polarity for the X and Y axis and 1 for the standard deviation.

In image 4 a different image was used for assessing the results of the filter changing the default parameters as is described below each result. Decreasing the standard deviation to 0.3 diminishes the ability of the filter to find edges, but it also diminishes “false edges” and noise. As was seen with the artificial image, changing the polarity of the axes enhances the sensibility for finding edges in different directions.

## CONCLUSIONS

Many different kinds of edge filters are available for the end-user. Their optimal utilization requires profound knowledge on its mechanisms. With this paper the Canny Algorithm was applied to images to test its functions, as well as an explanation of its mathematical “behind the scene” mechanism.

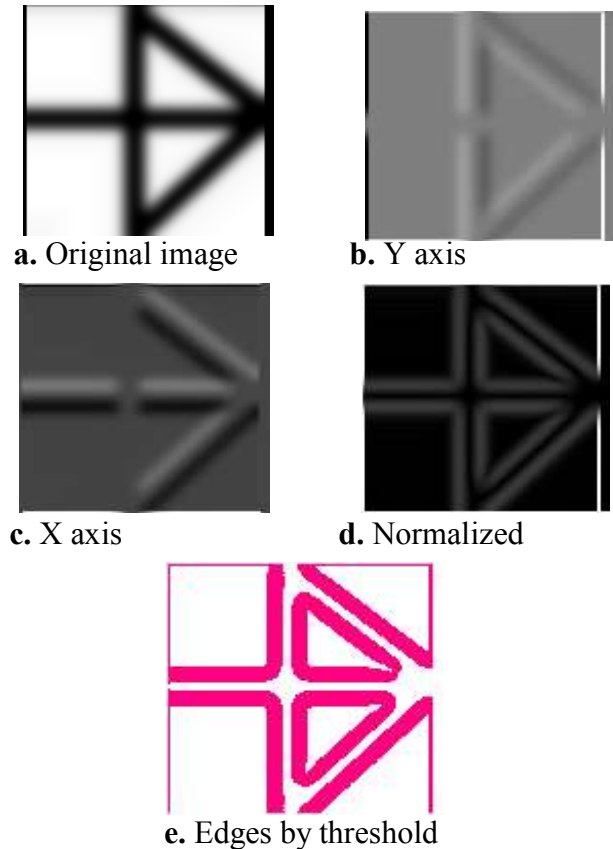


Image 3 – Canny filter applied to artificial image

The optimal performance can be obtained by carefully selecting the ideal parameters according to the elements which the user is interested in. This selectivity can be used to apply this filter to particular tasks, like it was done for Stromboli.

The Canny filter is considered an optimal choice for any task which requires edge extraction or edge enhancement of elements with different shapes and characteristics thanks to its “programmability” and its ability to be adapted to different circumstances.

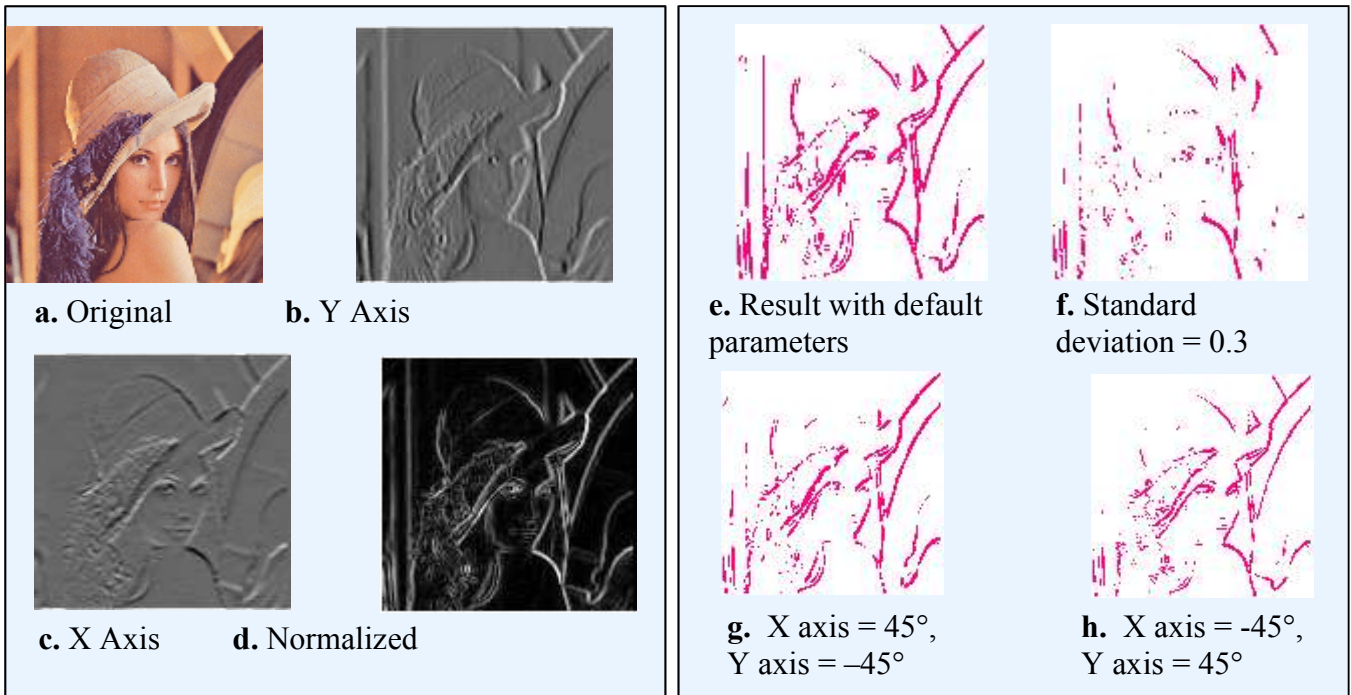


Image 4 – Filter applied to test images

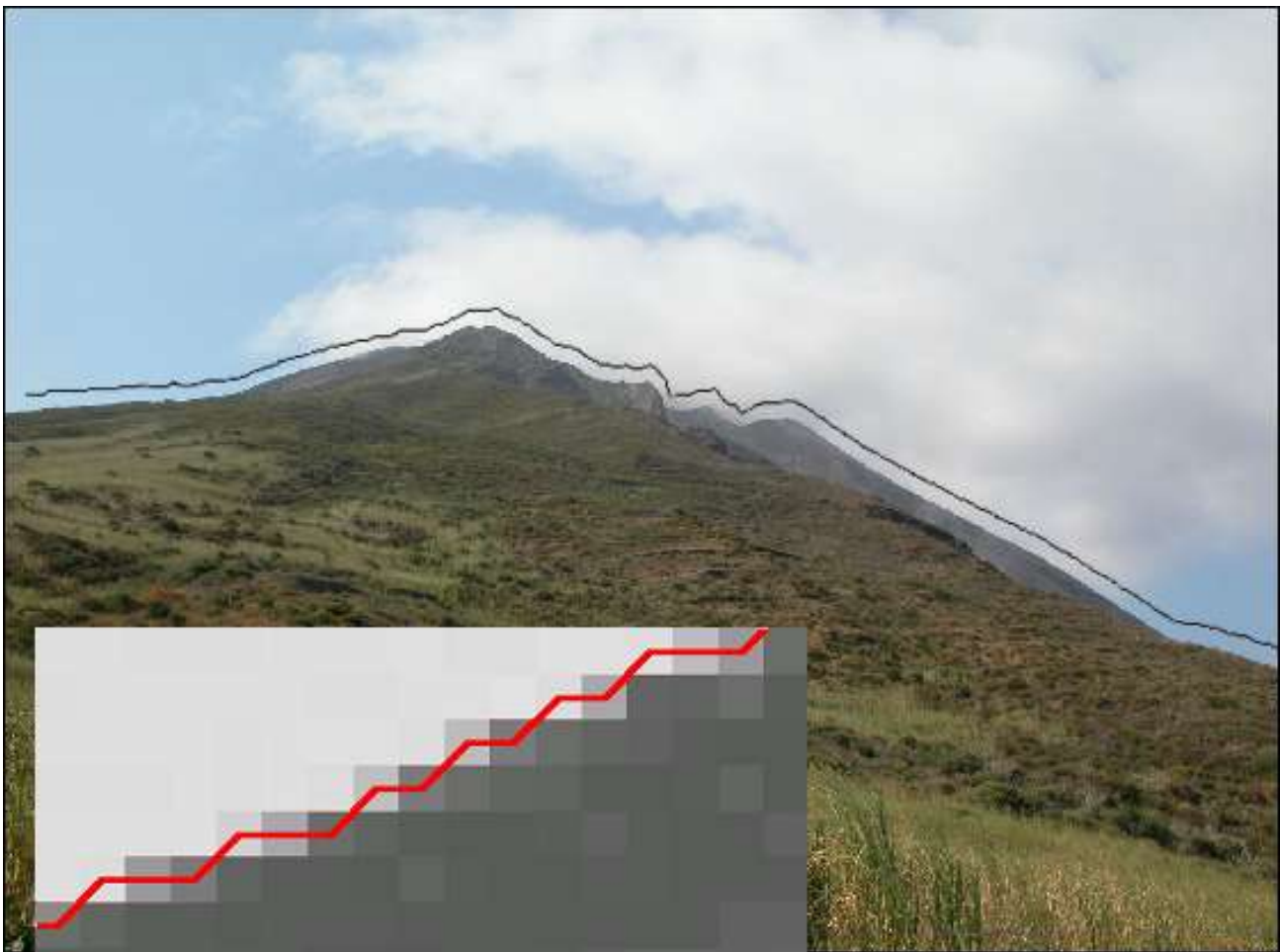


Image 5: The side of the Stromboli Vulcano: its profile has been extracted with the Canny filter. A detail of this extraction can be seen in the zoomed grey-level image of the side.

## REFERENCES

- [1] Canny J. (1986) "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, NO. 6. November 1986
- [2] Csetverikov D. (1999) "Basic Algorithms for Digital Image Analysis – a course", Institute of Informatics, Budapest, Hungary
- [3] Jain Anil K. (1989) *Fundamentals of Digital Image Processing*, Prentice Hall, 1989
- [4] Singh, A. (1989) "Digital Change Detection Techniques Using Remotely Sensed Data" *Remote Sensing* 10(6): 989 – 1003