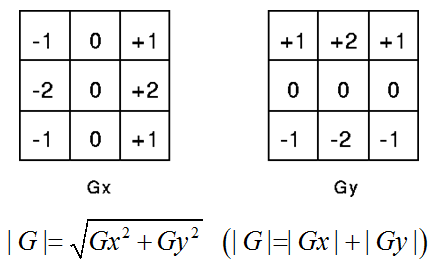
**Edge detection**

In Vision, an edge is the boundary between an object and its background. If edge detection is executed with precision, all the objects can be located and identified in the picture. Objects such as an area, perimeter, shape, etc. The purpose of detecting sharp changes is to capture important objects and reduce the amount of data to be processed that is irrelevant.

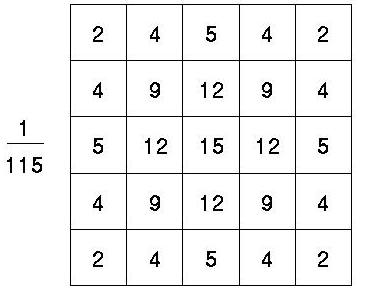
The figure above shows the well-known Lena. On the left is the original picture whereas the right shows edge detection. As said before the information that may be regarded as less relevant is filtered out, while preserving the important structural properties of Luna. Edge detection is one of the fundamental steps of computer vision techniques. A lot of edge detection algorithms are out there. Two of those chosen randomly will be discussed next. These are the Sobel Edge Detection and the Canny Edge Detection.

**Sobel edge detection**The Sobel operator performs edge detection by looking at the image gradient. An image gradient is a change in intensity (or color) of an image. An edge in an image occurs when the gradient is greatest and the Sobel operator makes use of this fact to find the edges in an image. It calculates the approximate image gradient of each pixel by convolving the image with a pair of 3x3 filters. These filters estimate the gradients in the horizontal (x) and vertical (y) directions. The magnitude of the gradient is simply the sum of these two gradients. The most common masks to use for the Sobel operator are:  
The values of the mask can be amplified to detect even more edges. The important part of these masks is that the middle row(x) or column(y) must be zero. A normalized gradient picture is the result of each of these masks. The results for applying these masks is shown below.

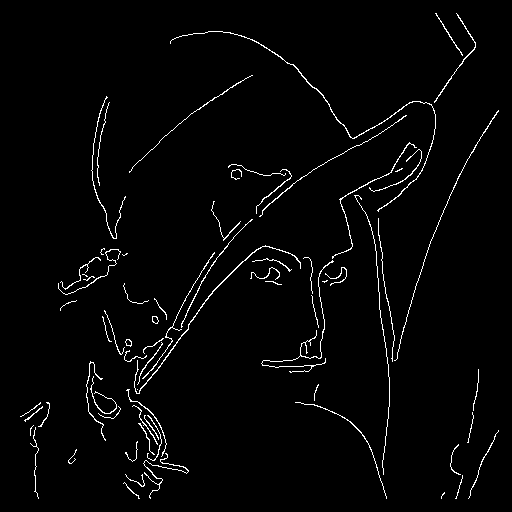
The result for the Gy is shown in the middle whereas the result for the Gx is shown on the right. Note that Gaussian blur was used for the original image to reduce noise. If both of these gradient are put together then it will show the edges as the picture on the right. An advantage of using the Sobel operator is that it gives an estimate of edge direction as well as edge magnitude at a point which is more informative and is of considerable use n later processing. A disadvantage of the Sobel edge detector is that the response to a step edge is present over either two or three pixels width, dependent on the precise position of the step relative to the pixel grid.

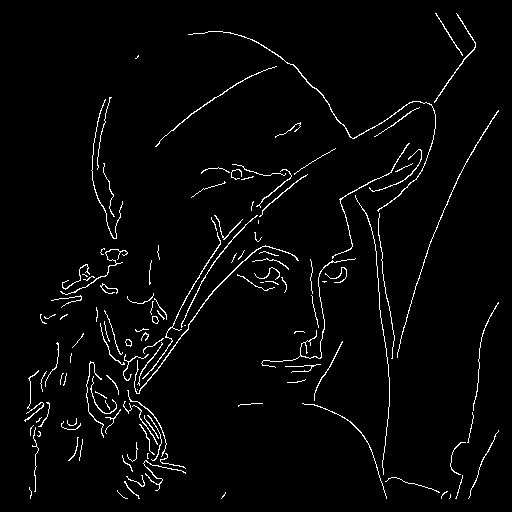


**Canny edge detection**John Canny came up with a theory about optimizing an edge detector. It should satisfy these three conditions:  
- The edge detector should respond only to edges and should find all of them: no edges should be missed.  
- The distance between the edge pixels as found by the edge detector and the actual edge should be as small as possible.  
- The edge detector should not identify multiple edge pixels where only a single edge exits.  
To meet these three conditions, John went step by step to simplify and execute the three conditions.

The first step was to add a Gaussian mask. Because the Canny edge detector is susceptible to noise present in raw unprocessed image data, the noise should be removed first. The result of using a Gaussian filter is a blurred version of the original image. However not a single noisy pixel will be present. How large the Gaussian mask should be depends on the size of an image. Using a larger mask will cause details to be lost forever while minimalizing the possibility of noisy pixels. On the right is an example of a gaussian 5x5 mask.


 \operatorname{atan2} (y, x)=2 \arctan \frac{\sqrt{x^2+y^2}-x}{y}.
The second step is to search for the edge strength. This will be done by using the mentioned Sobel operator. The canny edge detection is not satisfied with the two filters (x,y) the Sobel had just created. It also needs to know the direction of the edge. This will be done by using the atan2 function (using Gx and Gy as the x,y). Now we have four angles representing vertical, horizontal and the two diagonals.

The third step is to use non-maximal suppression. This means the edges will be thinned out. This process results in one pixel wide ridges from thick edges. It requires horizontal and vertical gradients and the magnitude of gradients (Sobel). The basic idea of non-maximal suppression is that: If a pixel value is not greater than its neighbored pixels, then the pixel is not the edge(the value is set to zero). During this process, it also needs to know the direction of the gradient vector in order to find 2 neighbor pixels on the same direction, then to compare them with the current pixel. That is why the 4 filters were needed in the second step. On the left we see Luna with one pixel edges. Using non-maximal suppression has put the basic edges.

The last step is to perform hysteresis threshold. It is needed to remove the weak edges and connect the splitted edges. To connect the edges it compares each pixel with the 8 surrounding neightbours. If the neighbor is greater than the low threshold that has been input, then it will also become an edge. The range of threshold is: 0 < tLow < tHigh < 1. Choosing a tLow and tHigh is easier since non-maximal suppression was applied.  
On the right we see Luna after the Canny edge detection. The advantages for using the Canny algorithm are that there is a better detection than most algorithms even when in noise conditions. The disadvantage of using this algorithm is that there are so many steps to take that it will cost a lot of performance if executed on a computer.