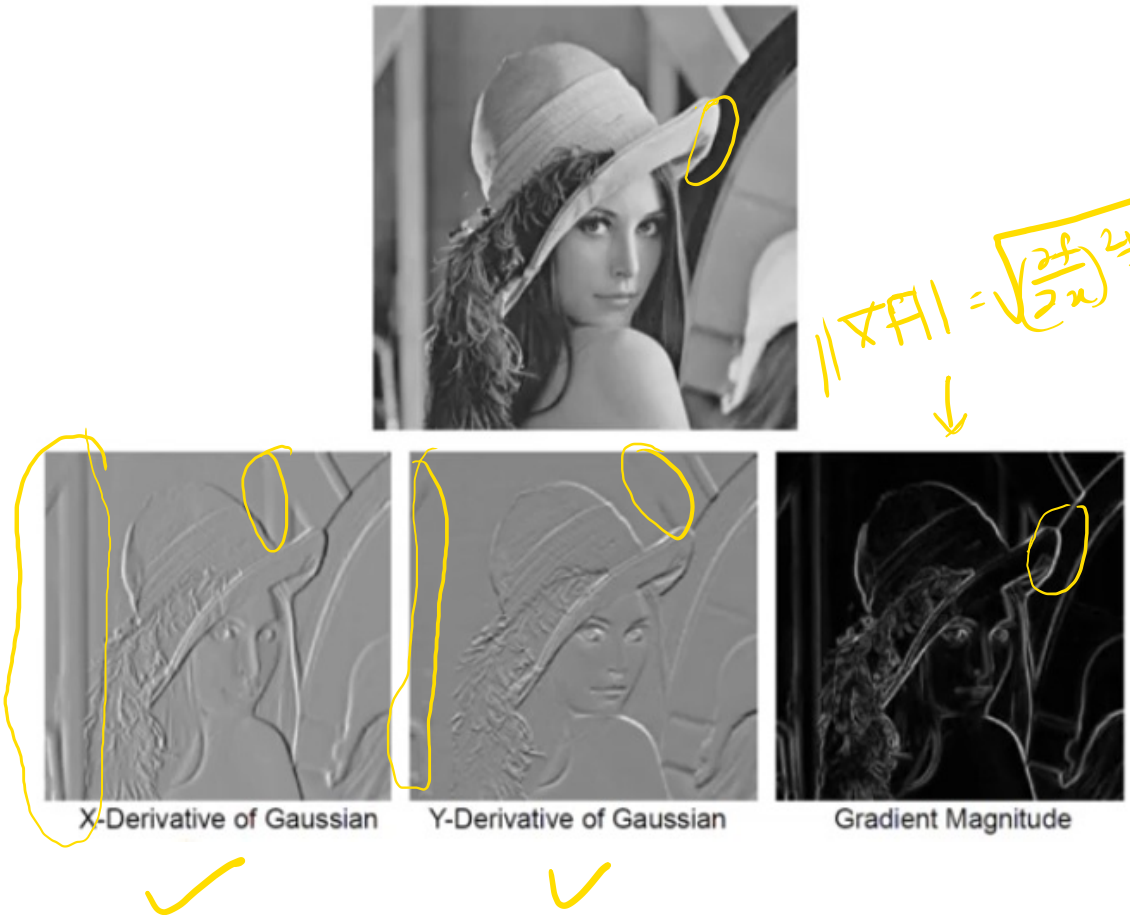
Compute of Gradients

Lina (image processing)



You can compute the x derivative of the Gaussian

You can compute the y derivative of the Gaussian

Now take the gradient magnitude, which is root of x derivative square plus y derivative square and you will find that the gradient magnitude gives you a set of edges

Properties of an edge detector

When we take a closer look at one of these edges on the image that we just saw earlier. It is hard to find where is the edge really.

There seem to be so many pixels that are white here, that an edge could be anywhere on those pixels.

But when we say we want an edge, we expect certain properties to be met.

Properties of an edge detector

① Good Detection

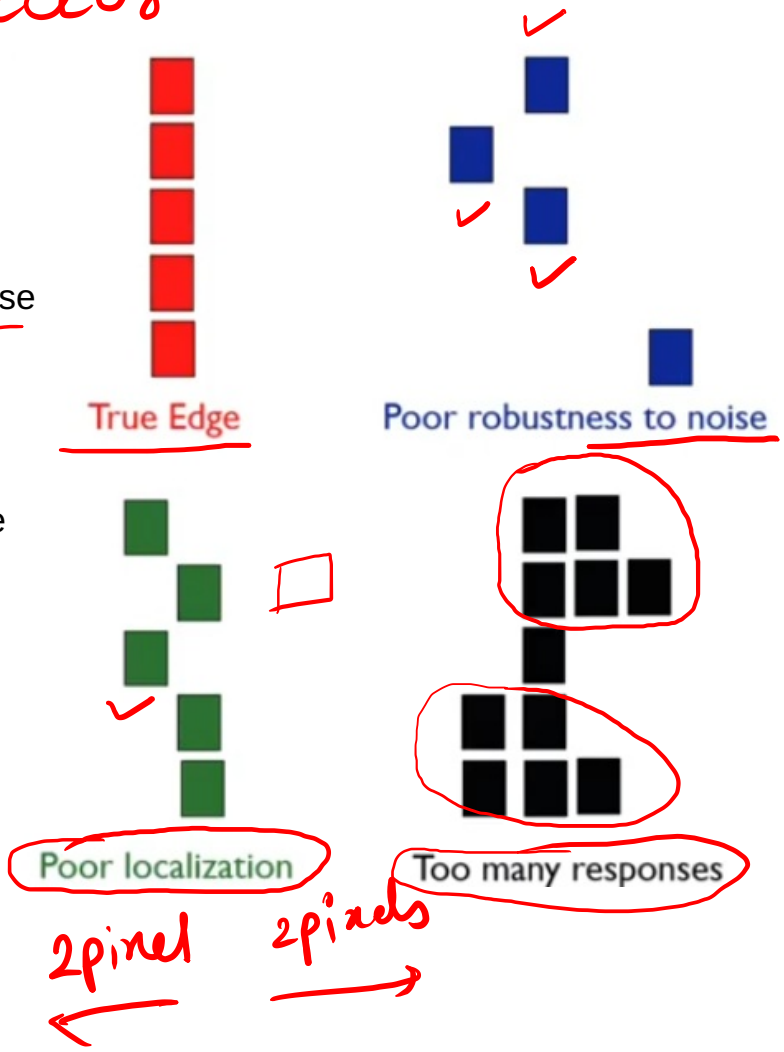
find all the real edges in the image and ignore noise or other artefacts. if there is an edge in the image, it should detect it. Noise points should not be detected as edges.

② Good Localization

if the edge is in this location, you also want to detect the edge in the same location. If you are to detect edges in the vicinity in the neighborhood and not exactly where the edge is, that will be called poor localization, but what we want is good localization.

③ Single Response

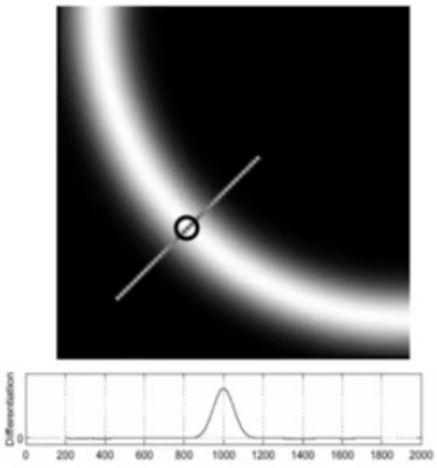
we only want a single response. We only want to return one point for a true edge point and not an entire region. If this was the true edge, this kind of a detection, which is spread out is not very useful and we want to be exactly at this location for a good edge detection method.

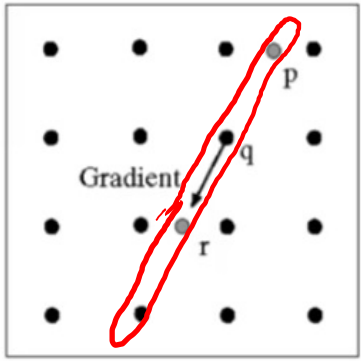


But so far, we only know how to use the gradient to detect the edges.

How do you solve problem such as good localization, single response, so on and so forth.

Non-Maxima Suppression





- Check if pixel is local maximum along gradient direction

A single response can be ensured in a particular region by doing Non-Maxima Suppression. Let us assume there was a circular kind of an edge in an image, that is how the edge looked.

Every white pixel would get detected as an edge, as we just saw, all of them could have high responses depending on what neighborhood you use to compute your gradient. So we may end up having many pixels corresponding to the edge, which you do not want. You want it to be isolated.

What do you do? You take a particular pixel, you compute the orientation of the gradient at that particular pixel as tan inverse of gradient along y direction by gradient along x direction, that gives you the orientation of the gradient at that particular pixel.

If you are at a pixel q and the orientation of the gradient is along a particular direction, then what you do is you go along the direction of the gradient and then try to see if this pixel is a maximum in that direction or not.

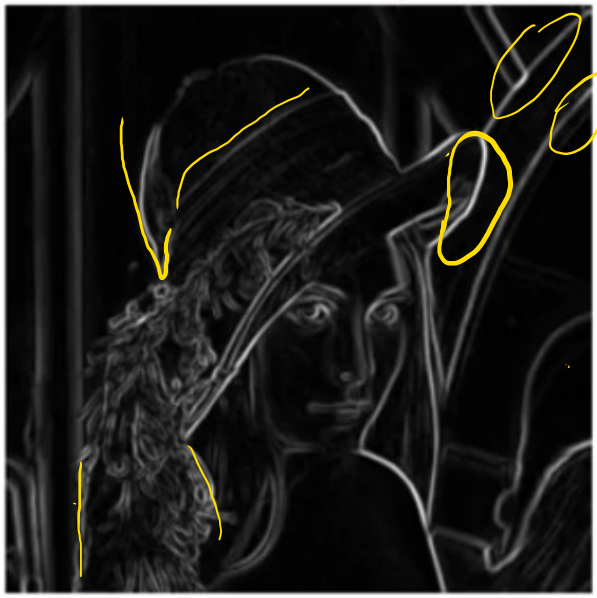
You want to retain only the pixel that has the highest gradient in value and make any other pixel that is not the maximum into zero. So you just check if the pixel is a local maximum along the gradient direction, if it is a maximum, retain its value, the edge magnitude value that you got, if it is not a maximum, make the edge magnitude zero and do not treat it as an edge.

upsampling

So in this process, one thing that you may have to remember, when you go along the direction of the gradient orientation, you may end up going to a location which is not defined on the image. It could be between two pixels. In those cases, you may have to interpolate, you can use simple linear interpolation or any simple interpolation method to be able to get what the value of the pixel of the gradient is at that location and you can then continue to do more maximum suppression.

Before and after Non-maxima Suppression

Lina Before



After Lina

fairly thinned
fairly localized

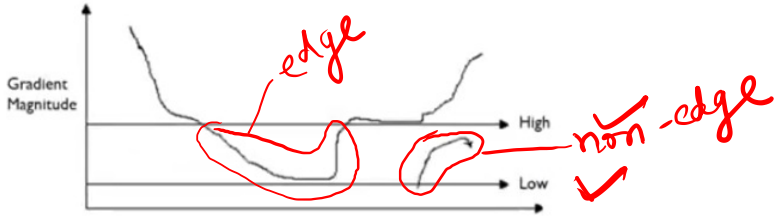
So here is an example of the same image after applying non-maximum suppression. And you can see now that the edges have fairly thinned out well and are fairly localized where the edge should be rather than have very thick edges.

One more problem that we find in the set of edges here is that, there are many discontinuities.

We would have ideally wanted the hat on her head to be one large edge, unfortunately, there seem to be multiple pieces.

In fact, there seem to be pieces even missing in between where we would have expected an edge.

Hysteresis Thresholding



Hysteresis Thresholding means that you would have to threshold your gradient magnitude

You would have two thresholds, a high threshold and a low threshold.

High gradient means it is an edge. If the gradient at a pixel is above the high threshold, it is definitely an edge pixel. Similarly, if the gradient at that pixel is less than a low value, it is definitely a non-edge pixel.

If you have the gradient at a particular pixel to be lying between these two thresholds. Then a particular pixel to be an edge pixel, if and only if, that pixel is connected to an edge pixel directly or via other pixels between low and high.

If you take one particular pixel there, you check if this pixel was connected directly or through a set of values to any edge pixel. If so, you would call that an edge pixel. On the other hand, if you take this pixel, if you now try to connect it, it does not connect to any edge pixel that is greater than high and hence, this entire set of pixels would get classified also as non-edge pixels.

Instead of relying on a threshold to be able to say whether you have an edge or not, you are now having two thresholds to ensure that even if you were on the boundary region, you could now use this approach to decide whether it should be an edge or not and that now gives more complete looking edges, where the edges are completely covering the object rather than those broken lines that we saw on the earlier.

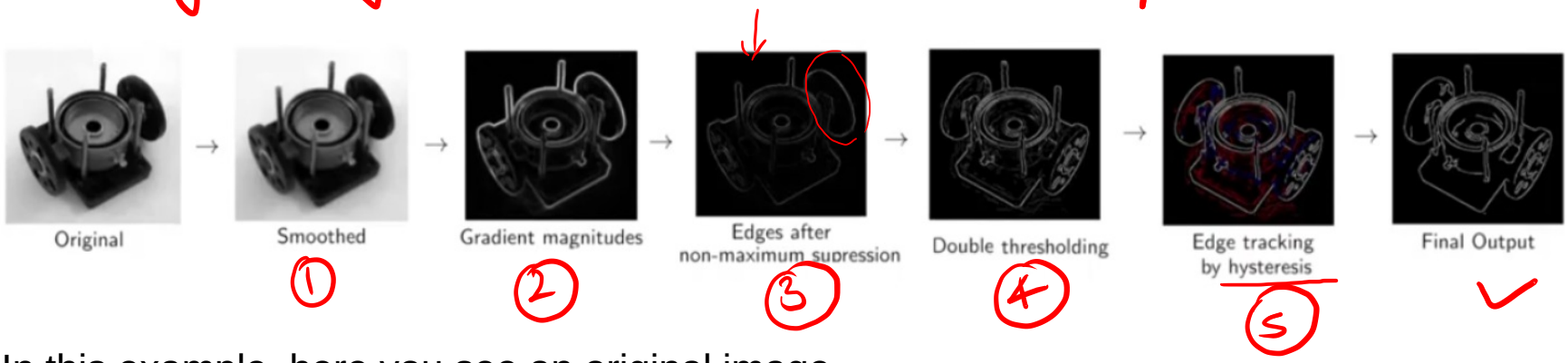
Canny Edge Detector

- Probably the most widely used edge detector in computer vision (J. Canny, 1986)
- Algorithm
  1. Filter image with derivative of Gaussian
  2. Find magnitude and orientation of gradient
  3. Non-maximum Suppression
  4. Linking and hysteresis Thresholding
     - ↳ Use the high threshold to start edge curves and the low threshold to continue them.

So you define two thresholds, low and high.

You use the high threshold to start edge curves and you use the low threshold to ensure that those edge curves can be continued until a little bit lower to be able to get a sense of more connected edges in the image.

Canny Edge pipeline and Examples



In this example, here you see an original image,
1) it smoothen using a Gaussian filter
2) and here are the gradient magnitudes after applying derivative, and
3) then you have the edges after doing the non-maximum suppression, you can see that the edges have thinner.
4) Then you do the double thresholding, which is the high thresholding. So you keep only the edges, edge pixels that are greater than the high threshold and you remove anything lower than the low threshold.
5) Then you do the hysteresis to give the continuity of the edges in between regions between high and low and here you have your final output after applying all of these processes.

other examples of canny edge detector

ex1 ex2 obj

Here are a few more examples of applying Canny edges, and you can actually see that for these kinds of images, they do a fairly good job, in a fairly cohesive way across various artefacts in the image.

Effect of σ in Canny Edge Detector

original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$

The choices of σ (Gaussian kernel size) depends on desired behaviour

(i) large σ detects large scale edges (ii) small σ detects fine edges

So one of the parameters that you have with the Canny Edge Detector is the size of the Gaussian kernel.

You could have various sigmas, you could have the Gaussian filter to be 3 cross 3, 5 cross 5, so on and so forth. So let us try to see what happens if you change the sigma in your Gaussian.

You will notice that a large sigma detects large scale edges and small sigma detects finer edges, it is very straightforward. So if you had a canny with sigma is equal to 1, you would have lots of small edges, whereas if you now increase sigma, then you would end up finding a canny filter, which has only large scale edges, not finer details, mainly because it increased the gradients of the Gaussian.