# Examen/proyecto final

> Miembros
>
> **0226594 Sara Carolina Gómez Delgado**
> **0224969 Luis Eduardo Robles Jiménez**

| I. | Valor: 10% | |
|---|---|---|

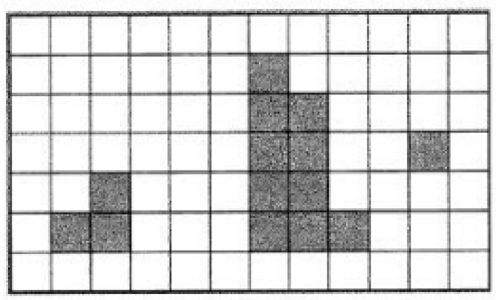
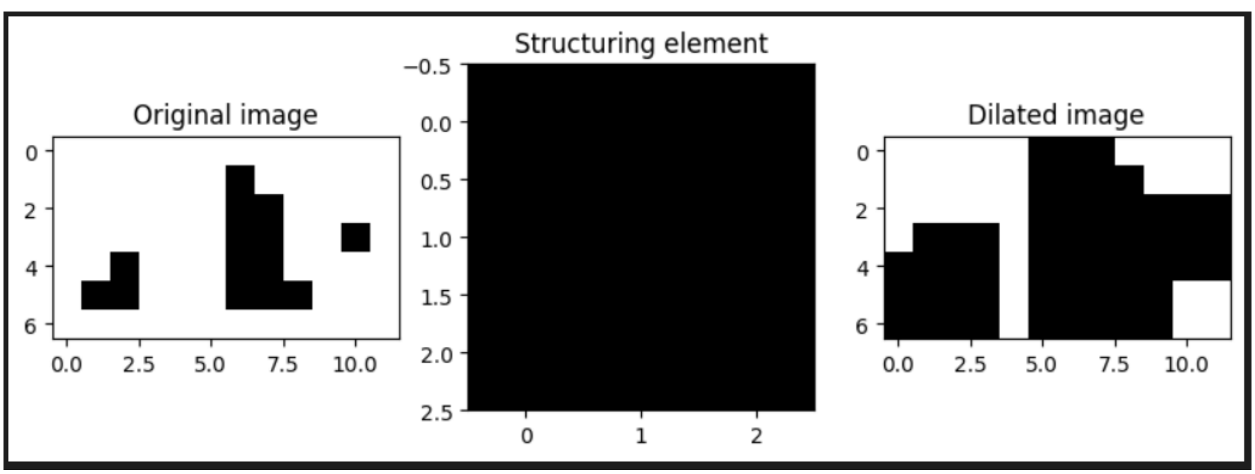
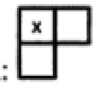


**Figure $I_1$.** Binary image of size 12x7 pixels, where gray == on, white == off.

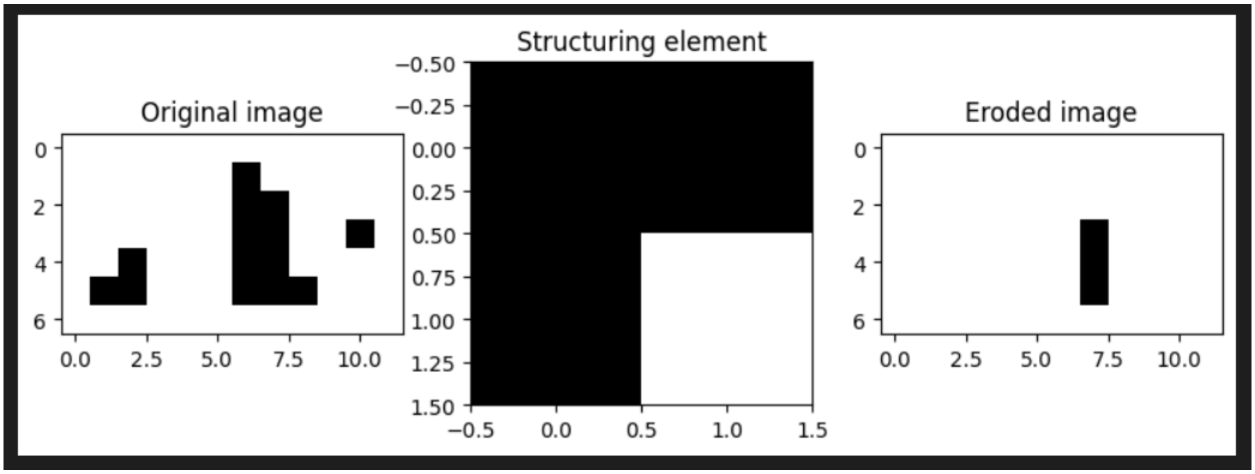
**(a)** By drawing in the graphic below, show the result of dilating $I_1$ using a 3x3 square structuring element (SE) whose origin is at the middle pixel.



Dilating is setting the pixel under the *origin pixel* as 1 if there’s a hit of the structuring element with the pixels of the original image below and 0 otherwise.

**(b)** (i) Show the result of eroding $I_1$ with this SE where the “x” marks its origin pixel: 







The idea is simple, by iterating the structuring element over the image, the pixel where *origin pixel* lies on is 1 if there’s a fit of the structuring element with the pixels below and 0 otherwise.

| | | |
|---|---|---|
| III. | Valor: 50% | |

a. The Prewitt kernels are given by $k_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ and $k_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$.

For an unknown image $I$ of size $M \times N$ you are given the edge values (computed using $k_x$ and $k_y$, respectively) as matrices $I_x$ and $I_y$, i.e., all the $M \times N$ values are given for both $I_x$ and $I_y$. Can you uniquely recover the original image $I$ based on $I_x$ and $I_y$? If so, how? If not, what (minimal) additional information is required?

b. On what principles does the Harris corner detector work? Given a grayscale image $I$, describe how would you use the Harris corner detector to decide if the pixel at $(i, j)$ is on a corner or not?

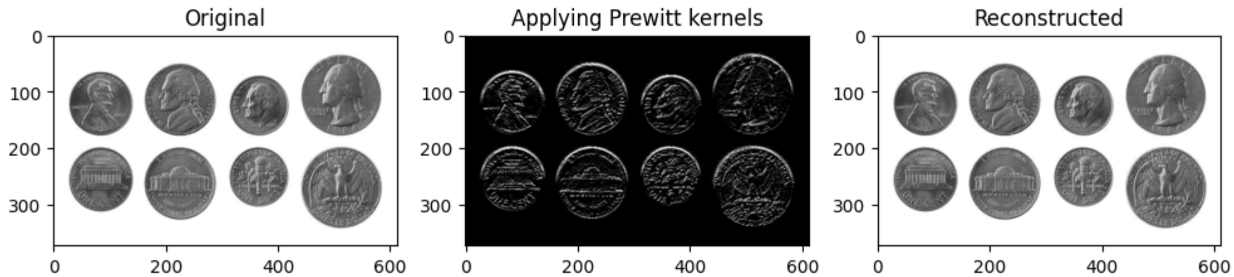
c. Given a moderately noisy image, would you use the USAN (Univalue Segment Assimilating Nucleus) or the Harris corner detector to extract corners? Why?

d. We use an analog-to-digital (A/D) convertor to digitize an analog signal, and then a digital-to-analog (D/A) to convert the signal back to an analog signal. In each of the two steps, typically a low pass filter is used. What are the purpose(s) of the two low-pass filters? Please provide appropriate explanatory figures (in the spatial domain or the frequency domain).

e. In the Canny edge detector, can we switch the order of non-maximal suppression and double thresholding without leading to performance loss? If so, why? If not, why not?

**a)**

Prewitt operator, as Scharr and Sobel Operator, is a derivative technique used for edge detection. As it is derivative, it is possible to recover the original image I based on Ix and Iy because we can apply the inverse operation (integrals) as we can see in the following image:




**b)**

Principles

The Harris corner detector works taking information on the corners in order to infer data from the image.

Determine if (i,j) is on corner or not

Given a grayscale image:

1. Find the difference in intensity for a displacement $(u, v)$ in every direction:

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \, [\underbrace{I(x+u, y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}}]^2$$

2. Maximize the function $E(u, v)$ applying Taylor Expansion to the last equation. The result is the following:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

Where

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

$I_x$ and $I_y$ are image derivatives and $x$ and $y$ directions.

3. This is the step that makes Harris Corner different to others. Here we use the following equation to determine if a window can contain a corner or not.

$$R = det(M) - k(trace(M))^2$$

where

- $det(M) = \lambda_1 \lambda_2$
- $trace(M) = \lambda_1 + \lambda_2$
- $\lambda_1$ and $\lambda_2$ are eigenvalues of $M$

There are 3 possibilitites:

a) $|R|$ is small: this means that **the region is flat.**

b) $R < 0$: this means that **the region is edge.**

c) $R$ is large: this means that **the region is corner.**

**c)**

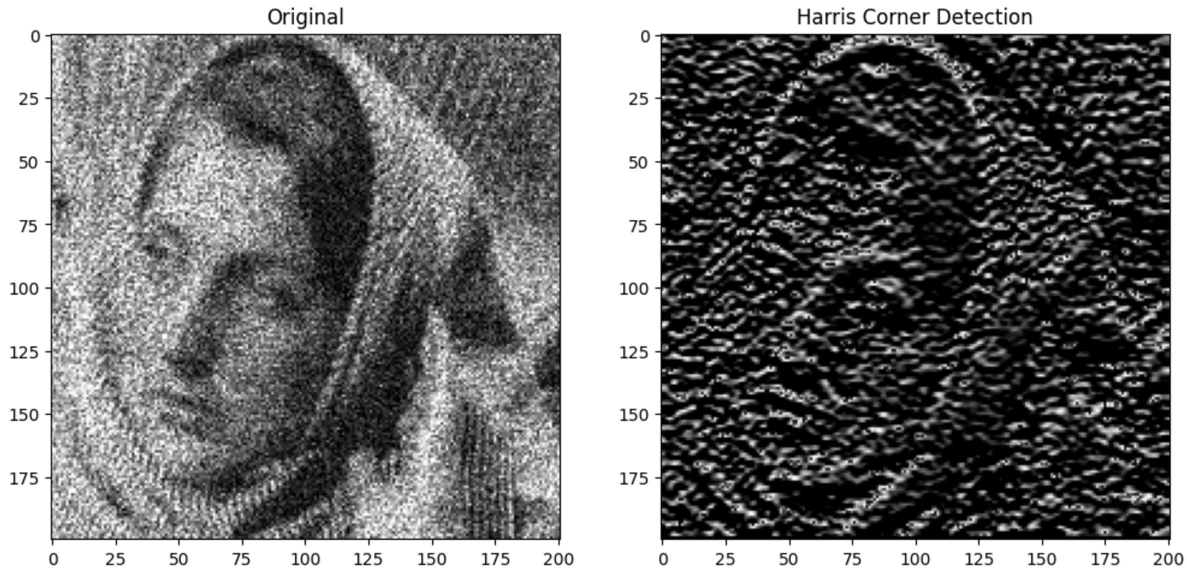
USAN (Univalue Segment Assimilating Nucleus) is a low level of image processing that applies a circular mask of fixed radius to each image pixel. The nucleus (central pixel) is compared to every pixel under this mask to see if the intensity value is different or similar.

The function in charge of comparing both pixels is the following:

$$C(r, r_0) = \begin{cases} 1, & \text{If } |I(r) - I(r_0)| <= T, \\ 0, & \text{Otherwise} \end{cases}$$

USAN does not use derivatives, so noise reduction is not necessary.

On the other hand, Harris is very vulnerable to fail with a noisy image since it used derivatives and any different threshold as we can see in the following image:



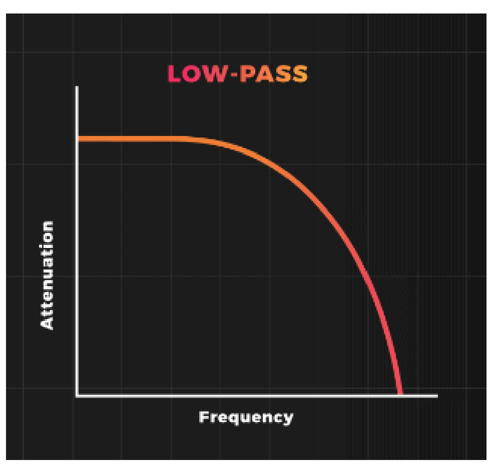

As a conclusion, for a noisy image it would be better to use USAN rather than Harris Corner Detection

**d)**

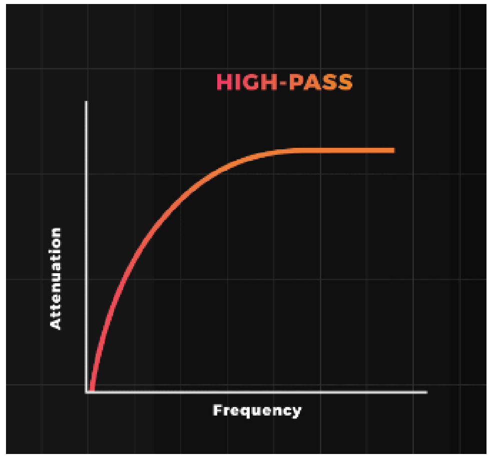
The purpose of the low pass filters is to avoid high frequencies.

In Music

When we want to manipulate the frequency in a song to increase the bass we need a low pass filter, this will try to block the high frequencies.




And, otherwise, if we want to emphasize the higher notes, then we need a high pass filter.




In Images

In images is the same situation, when we want to reduce high spatial frequency noise, we apply a low pass filter which will smooth the image (blur).

Otherwise, if we want to reduce low spatial frequency noise, we need to apply a high pass filter (**Figure 5**).

*Figure 5: Gaussian high pass filter*

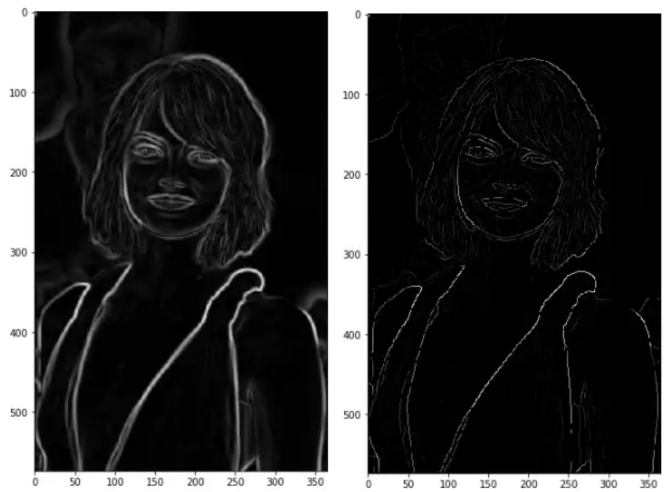
**e)**

Given a grayscale image, the Canny edge detector consists of 5 steps:

1. Noise reduction
2. Gradient calculation
3. Non-maximum suppression
4. Double threshold
5. Edge Tracking by Hysteresis

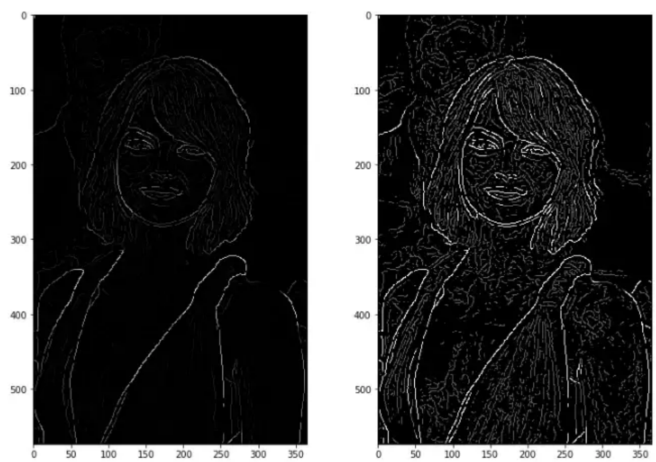
After applying these steps, we get the following result:

The third step (non-maximum suppression), helps to thin the detected edges:



Gradient intensity (left) || non-max suppression (left)

While the fourth step (Double threshold), decide if the intensity of each pixel is relevant or not:



non-max suppression (left) || double threshold (right)

After understanding how each of these steps work, we can conclude that by switching the order of them would affect the behavior of the algorithm since the borders would be extremely thin or maybe they wouldn't even be able to distinguish each other well.

| | | |
|---|---|---|
| V. | Valor: 50% | |

The Mars Rover uses a stereo system to create depth maps of the surface of Mars. The figure below shows a stereo pair of images captured by the Rover. The stereo algorithm is currently attempting to match the pixel at the centre of the yellow window in Image 1 against Image 2 using a template matching approach.

Let $I(x,y)$ be Image 2, $t(x,y)$ the template and $(u,v)$ the location at which we position the template in Image 2 to evaluate the matching score.

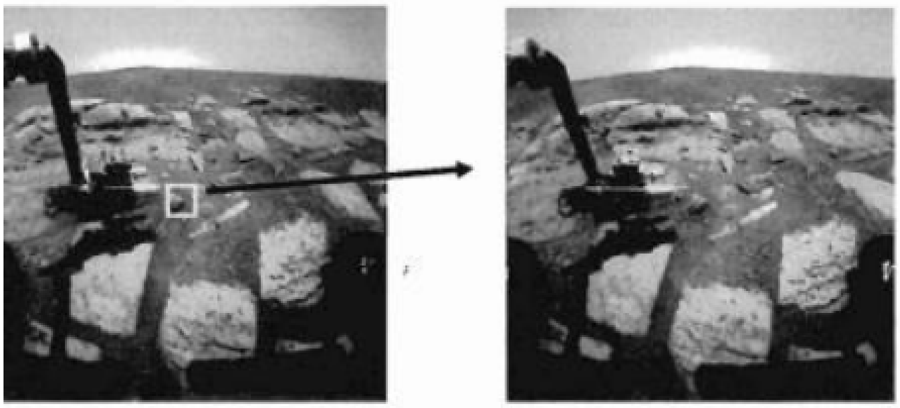


Image 1 Image 2

Give the formula for the template matching score obtained using each of the following measures:

i. Cross correlation (CC)

The cross correlation of a template against an image is defined as:

$$C_{fg} = \sum_{[i,j]\in R} f(i+x, j+y)g(i,j)$$

Where:

- $C$ is the correlation value
- $i, j$ are the indices of each pixel within the template
- $x, y$ are an offset in the original image, so the template can be moved around

- $f$ is the image where the template is being searched in
- $g$ is the template that we're looking for

### ii. Normalized cross correlation (NCC)

This measure comes to solve the problem of having a higher correlation value when comparing to brighter images than to darker images even when neither one is similar to the template.

And it is defined as:

$$NCC = C_{fg}(\hat{f}, \hat{g})$$

Where:

- $\hat{f} = \frac{f - \overline{f}}{\sqrt{\left(\sum f - \overline{f}\right)^2}}$
- $\hat{g} = \frac{g - \overline{g}}{\sqrt{\left(\sum g - \overline{g}\right)^2}}$
- $\overline{f}, \overline{g}$ are the mean value of the image and template respectively

### iii. Sum of squared differences (SDD)

Another measure that tells how different two chunks of data (pixels in this case) are.

It is defined by the following formula:

$$SSD = \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2$$

Where:

- $f$ is the image where the template is being searched in
- $g$ is the template that we're looking for

iv. For each of the above measures, describe if it is a measure of similarity or dis-similarity. What is the relationship between cross correlation (CC) and sum of the squared differences (SSD)? What are the main advantages of using normalized cross correlation (NCC) versus unnormalized cross correlation (CC)?

a. **Measures of similarity or dis-similarity?**

- *Cross-correlation (CC) and Normalized cross-correlation (NCC)*

  (Normalized) cross-correlation are a measure of similarity since the product of the values is the greatest when both sources of data have high values, and when one of them is smaller so the product is.

- *Sum of squared differences (SDD)*

  SDD is a dis-similarity measure because as it is a sum of the differences, the more apart both compared values are, the greater the result will be.

b. **Relationship between CC and SSD?**

- Intuitively, we can say that as SSD is a dis-simalirity measure, we might find something like a negative similarity measure within it.
- Mathematically, it can be proven by re-writing the SSD formula.

$$SSD = \sum_{[i,j]\in R} (f(i,j) - g(i,j))^2$$

$$SSD = \sum_{[i,j]\in R} (f(i,j)^2 + g(i,j)^2 - 2f(i,j)g(i,j))$$

$$SSD = \sum_{[i,j]\in R} (f(i,j)^2 + g(i,j)^2) - 2\sum_{[i,j]\in R} f(i,j)g(i,j)$$

$$SSD = \sum_{[i,j]\in R} (f(i,j)^2 + g(i,j)^2) - 2C(i,j)$$

- As said, two times negative cross-correlation is within the sum squared differences formula. The robustness of the latter shows up by considering the squared values of the images.

c. **NCC vs CC?**

The advantages of a normalized cross-correlation over the simple one are robustness. More specifically:

- The score of the normalized one is in the range [-1, 1]. Where 1 is a perfect match and -1 is completely anti-correlated. And the other one varies depending on the input.
- Simple cross-correlation is sensible to brightness, and that's an unreliable behavior.