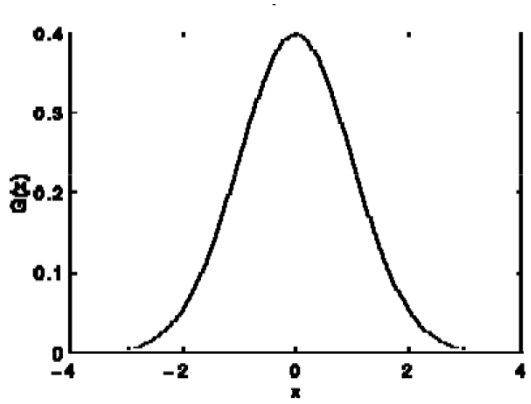
Digital Image Processing Gaussian Filtering

Gaussian filtering is used to blur images and remove noise and detail.
In one dimension, the Gaussian function is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Where σ is the standard deviation of the distribution. The distribution is assumed to have a mean of 0. Shown graphically, we see the familiar bell shaped Gaussian distribution.



Gaussian distribution with mean 0 and σ = 1

- **Significant Values In Gaussian filtering**

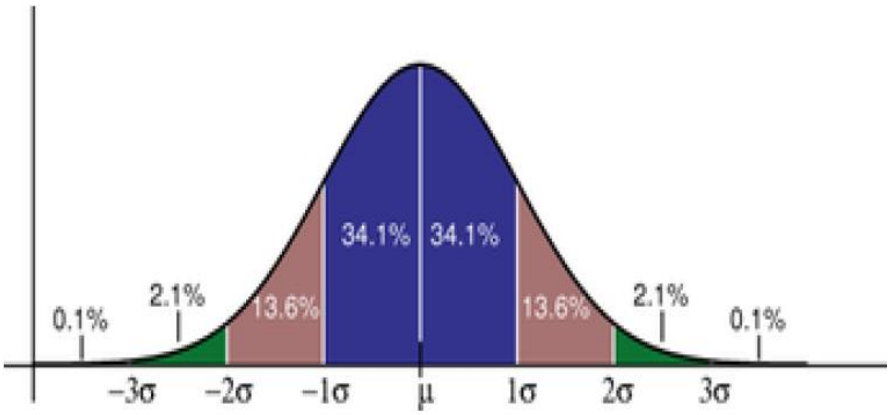
x	0	1	2	3	4
$\sigma * G(x) / 0.399$	1	$e^{-0.5/\sigma^2}$	e^{-2/σ^2}	$e^{-9/4\sigma^2}$	e^{-8/σ^2}
$G(x) / G(0)$	1	$e^{-0.5/\sigma^2}$	e^{-2/σ^2}	$e^{-9/4\sigma^2}$	e^{-8/σ^2}

For σ=1:

x	0	1	2
$G(x)$	0.399	0.242	0.05
$G(x) / G(0)$	1	0.6	0.125

• Standard Deviation

The Standard deviation of the Gaussian function plays an important role in its behaviour. The values located between +/- σ account for 68% of the set, while two standard deviations from the mean (blue and brown) account for 95%, and three standard deviations (blue, brown and green) account for 99.7%. This is very important when designing a Gaussian kernel of fixed length.
Therefore increasing the standard deviation increases the effect of the filter on the image.



Distribution of the Gaussian function values (Wikipedia)

• Integral Form of Gaussian Function

The Gaussian function has important properties, which are verified with respect to its integral:

$$I = \int_{-\infty}^{\infty} \exp\left(-\mathrm{x}^2\right) dx = \sqrt{\pi}$$

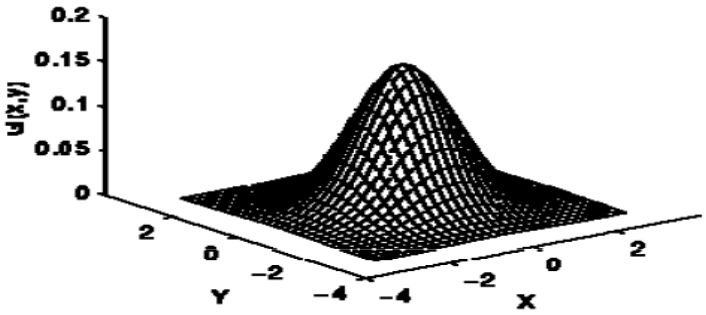
In probabilistic terms, it describes 100% of the possible values of any given space when varying from negative to positive values Gauss function is never equal to zero. It is a symmetric function.

• Two Dimensional Gaussian Function

When working with images we need to use the two dimensional Gaussian function. This is simply the product of two 1D Gaussian functions (one for each direction) and is given by:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

A graphical representation of the 2D Gaussian distribution with mean (0,0) and σ = 1 is shown to the right.



The Gaussian filter works by using the 2D distribution as a point-spread function. This is achieved by convolving the 2D Gaussian distribution function with the image. We need to produce a discrete approximation to the Gaussian function. This theoretically requires an infinitely large convolution (1) kernel (2) , as the Gaussian distribution is non-zero everywhere. Fortunately, the distribution has approached very close to zero at about three standard deviations from the mean. 99% of the distribution falls within 3 standard deviations.
This means we can normally limit the kernel size to contain only values within three standard deviations of the mean.

Convolution mentioned in (1) is an operation that takes 2 functions and produces a third one. So it is an operation which expresses relationship of input and output function with a third function.
Gaussian kernel coefficients are sampled from the 2D Gaussian function. Where σ is the standard deviation of the distribution.

(2) Kernel in mathematics is a set of vectors that lands on the origin (zero vector). Therefore when a vector is a zero vector kernel gives you the all possible solutions to the equation. However, in image processing an image kernel is a small matrix used to apply effects on the image itself. This is done by doing a convolution between a kernel and an image.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The distribution is assumed to have a mean of zero. We need to discretize the continuous Gaussian functions to store it as discrete pixels. An integer valued 5 by 5-convolution kernel approximating a Gaussian with a σ of 1 is shown to the right,

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

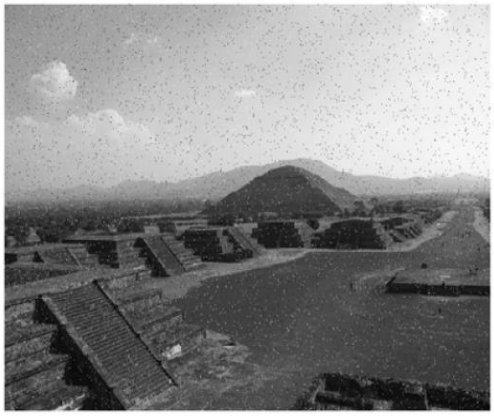
The Gaussian filter is a non-uniform low pass filter. The kernel coefficients diminish with increasing distance from the kernel's centre. Central pixels have a higher weighting than those on the periphery. Larger values of σ produce a wider peak (greater blurring). Kernel size must increase with increasing σ to maintain the Gaussian nature of the filter.
Gaussian kernel coefficients depend on the value of σ. At the edge of the mask, coefficients must be close to 0. The kernel is rotationally symmetric with no directional bias.
Gaussian kernel is separable which allows fast computation separable, computation.
Gaussian filters might not preserve image brightness.

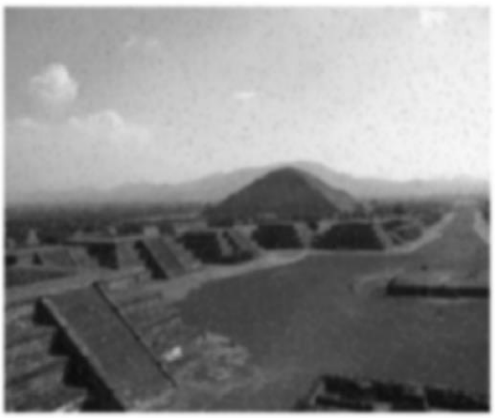
• Using Areas of Gaussian Function

The Gaussian function is used in numerous research areas:

1– It defines a probability distribution for noise or data.
2– It is a smoothing operator.
2– It is used in mathematics.

1-Gaussian filtering is used to remove noise and detail. It is not particularly effective at removing salt and pepper noise. Compare the results below with those achieved by the median filter.

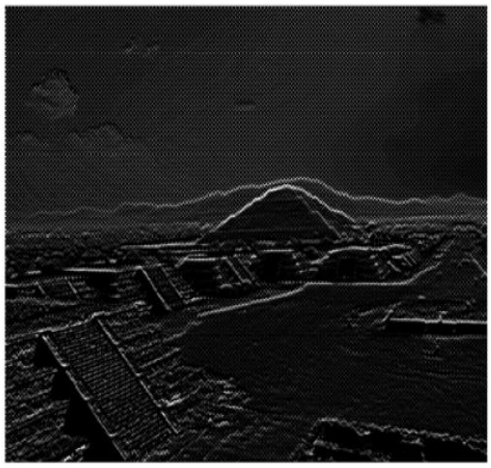
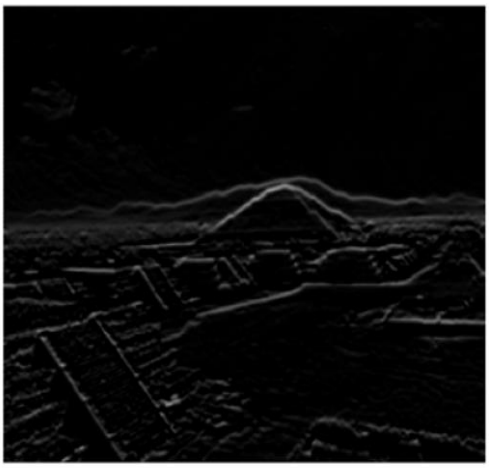




2-Gaussian filtering is more effective at smoothing images. It has its basis in the human visual perception system. It has been found that neurons create a similar filter

when processing visual images. The halftone image at left has been smoothed with a Gaussian filter and is displayed to the right.


3-This is a common first step in edge detection. The images below have been processed with a Sobel filter commonly used in edge detection applications. The image to the right has had a Gaussian filter applied prior to processing.

Code of some image processings on MATLAB:

```
I = imread('cameraman.tif');

Iblur1 = imgaussfilt(I,2);
Iblur2 = imgaussfilt(I,4);
Iblur3 = imgaussfilt(I,8);
 The value after I is used to change the value of sigma.

figure
imshow(Iblur1)
title('Smoothed image, \sigma = 2')

This code smootths the image by gaussian blur with the standard deviation of 2.We
can change the standard deviation by changing the valkue after the “sigma”
notation.
This specific code is only used for isotropic gaussian filters which means that
deviation is the same along all the dimensions.
```

```
IblurX1 = imgaussfilt(I,[4 1]);
IblurX2 = imgaussfilt(I,[8 1]);
IblurY1 = imgaussfilt(I,[1 4]);
IblurY2 = imgaussfilt(I,[1 8]);
 This is used to use different deviation values in different axis of dimensions.

figure
imshow(IblurX1)
title('Smoothed image, \sigma_x = 4, \sigma_y = 1')
```

Creating a 2-D filter in Matlab.

```
1) h = fspecial(type)
2) h = fspecial('average',hsize)
3) h = fspecial('disk',radius)
4) h = fspecial('gaussian',hsize,sigma)
5) h = fspecial('laplacian',alpha)
6) h = fspecial('log',hsize,sigma)
7) h = fspecial('motion',len,theta)
8) h = fspecial('prewitt')
9) h = fspecial('sobel')
```

1) `h = fspecial(type)` creates a two-dimensional filter `h` of the specified `type`. Some of the filter types have optional additional parameters, shown in the following syntaxes. `fspecial` returns `h` as a correlation kernel, which is the appropriate form to use with `imfilter`.
2) `h = fspecial('average',hsize)` returns an averaging filter `h` of size `hsize`.
3) `h = fspecial('disk',radius)` returns a circular averaging filter (pillbox) within the square matrix of size `2*radius+1`.
4) `h = fspecial('gaussian',hsize,sigma)` returns a rotationally symmetric Gaussian lowpass filter of size `hsize` with standard deviation `sigma`. Not recommended. Use `imgaussfilt` or `imgaussfilt3` instead.
5) `h = fspecial('laplacian',alpha)` returns a 3-by-3 filter approximating the shape of the two-dimensional Laplacian operator, `alpha` controls the shape of the Laplacian.
6) `h = fspecial('log',hsize,sigma)` returns a rotationally symmetric Laplacian of Gaussian filter of size `hsize` with standard deviation `sigma`.
7) `h = fspecial('motion',len,theta)` returns a filter to approximate, once convolved with an image, the linear motion of a camera. `len` specifies the length of the motion and `theta` specifies the angle of motion in degrees in a counter-clockwise direction. The filter becomes a vector for horizontal and vertical motions. The default `len` is `9` and the default `theta` is `0`, which corresponds to a horizontal motion of nine pixels.
8) `h = fspecial('prewitt')` returns a 3-by-3 filter that emphasizes horizontal edges by approximating a vertical gradient. To emphasize vertical edges, transpose the filter `h'`.

```
[ 1  1  1
  0  0  0
 -1 -1 -1 ]
```

9)`h = fspecial('sobel')` returns a 3-by-3 filter that emphasizes horizontal edges using the smoothing effect by approximating a vertical gradient. To emphasize vertical edges, transpose the filter `h'`.

```
[ 1  2  1
  0  0  0
```

```
 -1 -2 -1 ]
```

These are used to do some other filters on an image in MATLAB.And this is an example.

```
H = fspecial('disk',10);
blurred = imfilter(I,H,'replicate');
imshow(blurred);
```

-This is another form to do 2-D gaussian filter

Function Inputs:
- Filter_size: size of filter
- sigma: standard deviation
Function Output:
- 2D Gaussian filter matrix
Example to plot filter matrix in 3D:

```
g1=Gaussian_filter(50,2);
g2=Gaussian_filter(50,7);
g3=Gaussian_filter(50,11);
figure(1);
subplot(1,3,1);surf(g1);title('filter size = 50, sigma = 2');
subplot(1,3,2);surf(g2);title('filter size = 50, sigma = 7');
subplot(1,3,3);surf(g3);title('filter size = 50, sigma = 11');
```

References:

http://www.cse.psu.edu/~rtc12/CSE486/lecture04_6pp.pdf

https://web.cs.hacettepe.edu.tr/~erkut/bbm413.f13/w05-point-operations-4pp.pdf

https://www.cs.cornell.edu/courses/cs5670/2017sp/lectures/lec01_filter.pdf

http://pages.stat.wisc.edu/~mchung/teaching/MIA/reading/diffusion.gaussian.kernel.pdf.pdf

https://www.imageeprocessing.com/2011/04/matlab-code-for-linear-filtering.html#:~:text=Linear%20filtering%20technique%20is%20used,weighted%20sum%20of%20neighboring%20pixels.

https://www.imageeprocessing.com/2014/04/gaussian-filter-without-using-matlab.html

-SECOND ALTERNATİVE TOPİC-

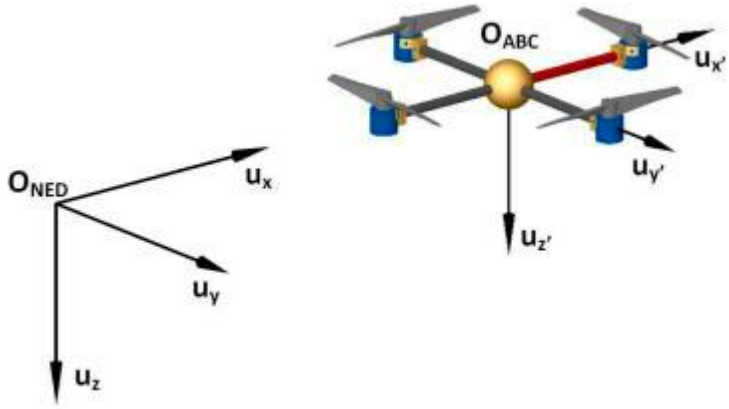
Quadcopters Dynamics:

Mathematical model describes multirotor movement and behaviour with the respect to the input values of the model and external influences on multirotor. Mathematical model can be considered as a function that is mapping inputs on outputs. By using mathematical model, it is possible to predict position and attitude of multirotor by knowing the angular velocities of propellers, i.e. it enables computer simulation of multirotor behaviour in different conditions.

In order to explain how a drone works we need to define coordinate systems of the drone.

1) Earth reference axis of inertia(body frame,O_{NED})

2) Quadcopter inertial reference axis((earthframe, O_{ABC})



The angular position of the quadcopter with respect to the axis of inertia of the axis fixed to the body is expressed by directions. While performing these operations, “pitch, roll and yaw, ”angles are used.

$$R(x, \emptyset) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\emptyset & -sin\emptyset \\ 0 & sin\emptyset & cos\emptyset \end{bmatrix}$$

$$R(y, \theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}$$

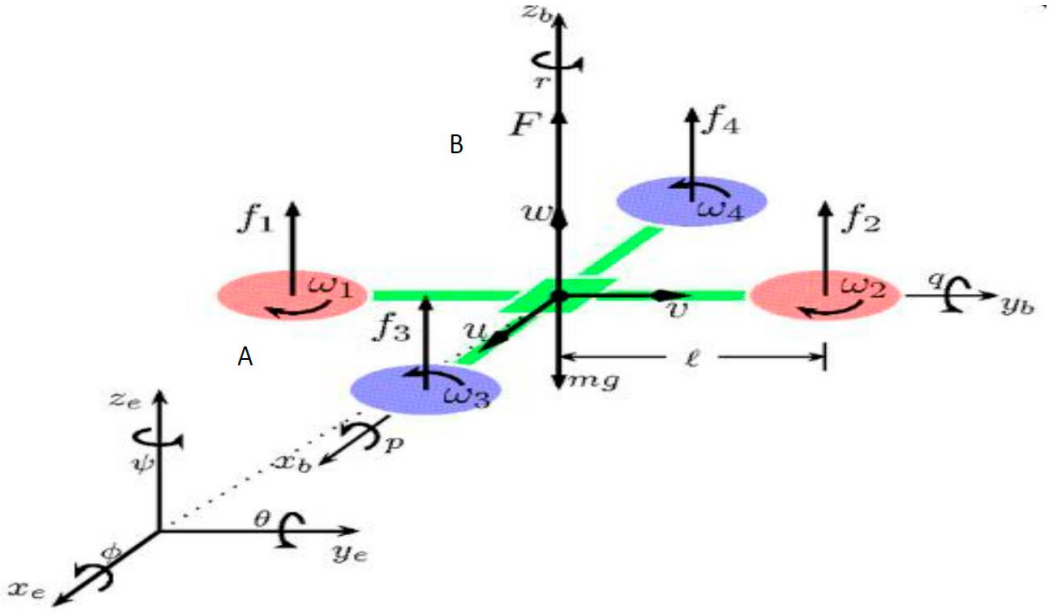
$$R(z, \Psi) = \begin{bmatrix} cos\Psi & -sin\Psi & 0 \\ sin\Psi & cos\Psi & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

These are rotaions according to x,y and z axis.
R is the rotation matrix that denotes the orientation of the OABC coordinate frame with respect to the ONED frame.

$$R = \begin{bmatrix} cos\Psi cos\theta & cos\Psi sin\theta sin\emptyset - sin\Psi cos\emptyset & cos\Psi sin\theta sin\emptyset + sin\Psi cos\emptyset \\ sin\Psi cos\emptyset & sin\Psi sin\theta sin\emptyset + cos\Psi cos\emptyset & sin\Psi sin\theta sin\emptyset + sin\emptyset cos\Psi \\ -sin\theta & cos\Psi sin\emptyset & cos\theta cos\Psi \end{bmatrix}$$

$$R(\phi, \theta, \psi) = R(x, \phi)\, R(y, \theta)\, R(z, \psi)'dir.$$

The picture below shows the places and the angles of the drones movement axis.



In the MATLAB Simulink program drones graphics and schemes can be drawn and the values of angles can be calculated.

REFERENCES:

https://openaccess.firat.edu.tr/xmlui/handle/11508/18372
https://bib.irb.hr/datoteka/886586.009-0035.pdf
https://www.researchgate.net/publication/292176923_Mathematical_Modelling_of_Unmanned_Aerial_Vehicles_with_Four_Rotors
https://journals.tubitak.gov.tr/elektrik/issues/elk-12-20-1/elk-20-1-12-1007-624.pdf
https://www.maxwell.vrac.puc-rio.br/30497/30497.PDF