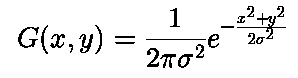
**Segmentation Implementation**

1. Gaussian Smoothing

The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is like the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. In 2-D, an isotropic (i.e. circularly symmetric) Gaussian has the form:



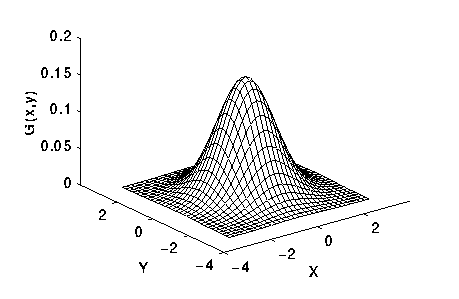
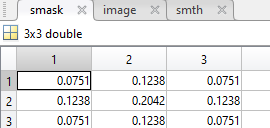


Figure 1: 2-D Gaussian distribution with mean (0,0) and sigma=1

The Gaussian kernel generated looks like:



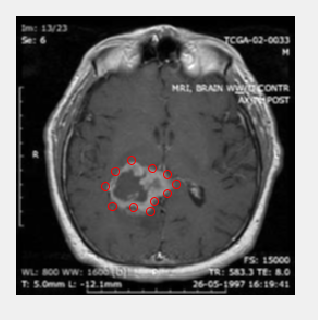
The effect of Gaussian smoothing is to blur an image, in a similar fashion to the mean filter. The degree of smoothing is determined by the standard deviation of the Gaussian. (Larger standard deviation Gaussians, of course, require larger convolution kernels in order to be accurately represented.)

The Gaussian outputs a `weighted average' of each pixel's neighborhood, with the average weighted more towards the value of the central pixels. This contrasts with the mean filter's uniformly weighted average. Because of this, a Gaussian provides gentler smoothing and preserves edges better than a similarly sized mean filter.

1. Segmentation (Active Contours: Snakes)

A snake is a spline which minimizes energy and is influenced by external and image forces which in turn pull it towards the edges and lines. Snakes find nearby edges and lock onto them, thus accurately localizing them [10]. The algorithm utilizes certain energy terms which helps the model to push out from a local minimum and towards the desired solution.

* We begin the process by providing the preprocessed MRI scan as an input to the algorithm. On this image, we give an initial boost by providing an initial set of points as input to the algorithm. A spline will be created from these points and then the snake will slither towards a nearby contour to get the result.



The external constraint forces and internal image forces play a very important role in determining the salient image contours. If we represent the position of snake parametrically by v(s) = (x(s), y(s)), then the energy term can be written as

Esnake = 0∫1 Esnake (v(s))ds (1)

= 0∫1 Eint(v(s)) + Eimage(v(s)) + Econ(v(s)) ds (2)

In equation (2), *Eint* is the internal energy of the spline, Eimage is the image forces and Econ is the external constraint forces [10].

The equation for internal energy (*Eint*) is:

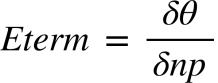
Eint = (α(s)|vs(s)|² + β(s)|vss(s)|²)/2            (3)

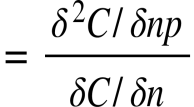
It contains the first order derivative which is controlled by alpha(α) and second order derivative which is controlled by beta(β). The snake behaves like a membrane because of the first order term and like a thin plate because of the second order term.

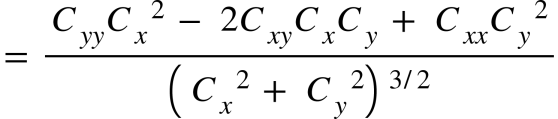
We also need an energy function to attract the snake towards the important image features. The weighted combination of three energy terms forms the total image energy [10].

Eimage = wlineEline + wedgeEedge + wtermEterm    (4)

The image intensity is the simplest image functional. Eline = I(x,y). Depending upon the sign of wline, the snake will slither towards light or dark lines in the image. The edges in the image can be found with simple energy functional. Eedge= -|∇I(x,y)|². This will attract the snake towards large image gradients. The terminations of line segments and corners can be found using another energy functional. If C(x,y)= Gσ(x,y)\* I(x,y) is the smoothed image, θ= tan-1(Cy/Cx) and n= (cosθ,sinθ) and np= (-sinθ,cosθ) are unit vectors then the curvature of the level contour in C(x,y) is

 (5)

 (6)

(7)

Using equations (3) and (4), the overall energy can be computed which helps the snake lock on to contours in the neighboring region of the provided input. Once the tumor boundaries are caught by the snake, the required region can be extracted from the image.

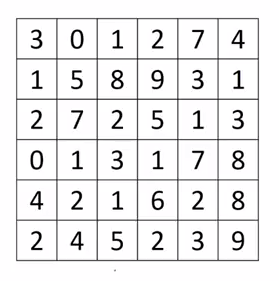
* The algorithm will calculate the 3 independent energy terms to compute the total energy of the image. It does this by using the following MATLAB functions:

1. Gradient

The gradient of a function of two variables, F(x,y) , is defined as

https://edoras.sdsu.edu/doc/matlab/techdoc/ref/math_g5a.gif

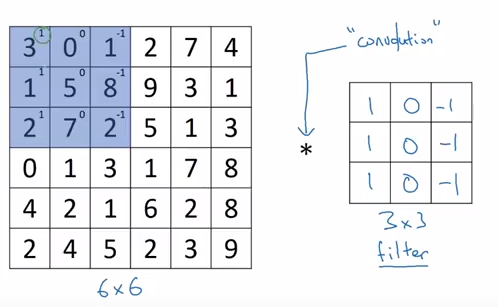
and can be thought of as a collection of vectors pointing in the direction of increasing values of F. In the figure below a matrix is given. Gradient is simple the difference between the adjacent pixels in horizontal and vertical direction.



1. Conv2

C = conv2(A,B) computes the two-dimensional convolution of matrices A and B. If one of these matrices describes a two-dimensional finite impulse response (FIR) filter, the other matrix is filtered in two dimensions.

The size of C in each dimension is equal to the sum of the corresponding dimensions of the input matrices, minus one. That is, if the size of A is [ma,na] and the size of B is [mb,nb], then the size of C is [ma+mb-1,na+nb-1]. ‘same’ returns the central part of the convolution of the same size as A.



The masks used for convolution are:-

m1 = [-1 1];

m2 = [-1;1];

m3 = [1 -2 1];

m4 = [1;-2;1];

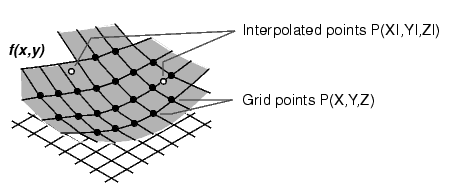
m5 = [1 -1;-1 1];

Thus, the derivatives are calculated in such a manner.

1. Interp2

ZI = interp2(X,Y,Z,XI,YI) returns matrix ZI containing elements corresponding to the elements of XI and YI and determined by interpolation within the two-dimensional function specified by matrices X, Y, and Z. X and Y must be monotonic, and have the same format ("plaid") as if they were produced by meshgrid. Matrices X and Y specify the points at which the data Z is given. Out of range values are returned as NaNs.

The interp2 command interpolates between data points. It finds values of a two-dimensional function F(x,y) underlying the data at intermediate points. Interpolation is the same operation as table lookup. Described in table lookup terms, the table is tab = [NaN,Y; X,Z] and interp2 looks up the elements of XI in X, YI in Y, and, based upon their location, returns values ZI interpolated within the elements of Z.



Given this set of employee data,

years = 1950:10:1990;

service = 10:10:30;

wage = [150.697 199.592 187.625

179.323 195.072 250.287

203.212 179.092 322.767

226.505 153.706 426.730

249.633 120.281 598.243];

it is possible to interpolate to find the wage earned in 1975 by an employee with 15 years' service:

w = interp2(service,years,wage,15,1975)

w =

190.6287

* After the energy terms are computed. The new snake position is interpolated and fixed for each iteration. The snake will thus, slither slowly towards the contour in each iteration.