# Pipeline - Distance Map for Ablation and Tumor Segmentation (Python)

1. mainDistanceMetrics.py – main file from which the workflow should be started
   * *rootdir* variable: a folderpath containing ablation and tumor segmentation masks must be given
   * the script iterates through the files and folders looking for files entitle “*ablationSegm”* and “*tumorSegm”* (current name convention for mask, the script can be edited if other names used) and stores the filepaths into a *DataFrame s*tructure
2. distancesv2.py
   * a function which calculates the Distance Metrics using mauerer algorithm from ITK (for Euclidean distances)
   * def \_\_init\_\_(self, maskfile , referencefile, flag\_symmetric, flag\_mask2reference, flag\_reference2mask)
   * class SurfaceDistanceMeasuresITK(Enum):
   * hausdorff\_distance, max\_distance, min\_surface\_distance, mean\_surface\_distance, median\_surface\_distance, std\_surface\_distance, rms\_surface\_distance = range(7)

Mauerer Distance Map for the Reference Object (deemed as "tumor in this particular case")

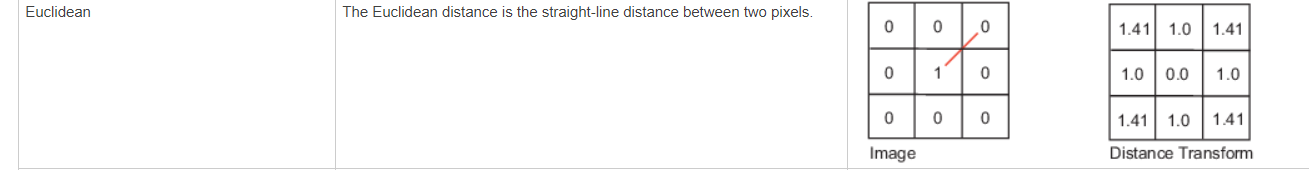
* + Algorithm Pipeline :
  + 1. compute the contour surface of the object (6-pixel-connectivity neighborhood)
  + 2. convert from SimpleITK format to Numpy Array Img
  + 3. remove the zeros from the contour of the object, NOT from the distance map
  + 4. compute the number of 1's pixels in the contour
  + 5. instantiate the Signed Mauerer Distance map for the object (negative numbers also)
  + **Current implementation**: compute distances from tumor surface (contour) to ablation
  + Calculated the distance map for the Ablation Segmentation using Mauerer Algorithm Implementation from SimpleITK
  + mask\_distance\_map = sitk.SignedMaurerDistanceMap(segmentation, squaredDistance=False, useImageSpacing=True)
  + Multiply the distance map of the ablation with the contour of the tumor (the tumor surface)
  + Remove the zeros from the contour of the tumor surface
  + Divide by 255 (so we have values of 1’s=object and 0’s=no-object)
  + Change the sign convention by multiplying with -1 (so that inside distances are negative and outside distances are positive)

1. Other Functions used for plotting:
   * *plotHistDistances.py*: cols\_val, bins = pm.plotHistDistances(pats[idx], pat\_id, rootdir, distanceMap\_seg2ref, n2, title)
   * *plotBoxplotSurface :* *plotBinsSurfacePercentage(bins\_4intervals, rootdir, flag\_all\_ranges=False)*
   * *DistancesVolumes\_twinAxes.py*

<https://ch.mathworks.com/help/images/ref/bwdist.html>

For each pixel in BW, the distance transform assigns a number that is the distance between that pixel and the nearest nonzero pixel of BW.





Maurer Algorithm implements the Euclidean distance. The problem with this metric is that for certain ablation/tumor shapes is misleading. Since the Euclidean distance is a metric of the closest distance, the shortest distance will be returned when computing the distances from Ablation to Tumor. Given the figure below, the maximum distance returned will be dAT. Perhaps it will be more intuitive if as maximum distance dTA is returned instead, which is the largest distance between the ablation and tumor surface in the picture depicted. However, as a surgeon I am interested In how small are my minimal distances from the tumor contour.

**Tumor**

Ablation

**+dAT**

**-d**

**d=0**

**+dTA**

## Why did we compute distances from Ablation to Tumor and not Tumor to Ablation:

????

## Next Step: Compute Surface Distances between Ablation Predicted and Ablation Resulted

* Compute the distance map for the ablation predicted
* Compute the surface of the ablation resulted
* Plot the histograms
* Extract the position of the ablation probe
* Project the ablation probe at different angles (circular path) and compute distances from the probe tip to the contour of the ablation (predicted and resulted). Compare the results.

## Next Step: Approximate the contour of the tumor and depending on the shape (circularity, bifurcations) select the suitable option for computing the metrics

## Principal Component Analysis for the contour of the object’s volume?

## Point to Surface Distance?

<https://ch.mathworks.com/matlabcentral/fileexchange/52882-point2trimesh------distance%C2%A0between-point-and-triangulated-surface>

<https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_plane>