Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Fisica Applicata

Region Growing and Fuzzy C-means algorithm segmentation for PET images of head-neck tumours

Relatore:
Prof.ssa Maria Pia Morigi

Presentata da:
Ilaria Cupparo

Correlatori:

Dott. Gabriele Guidi
Dott.ssa Raffaella Donnarumma
Dott. Nicola Maffei

Sessione III
Anno Accademico 2017/2018

Abstract

Scopo di questo lavoro di tesi, svolto presso l'Azienda Ospedaliero Universitaria di Modena, è l'implementazione e la validazione di due metodi di auto-segmentazione di immagini PET relativi ai tumori testa-collo. Questi processi di auto-segmentazione sono importanti soprattutto per superare i problemi relativi alla segmentazione manuale, effettuata dal medico radioterapista, quali il tempo di contornazione (che può raggiungere più di due ore) e la variabilità intra-osservatore e inter-osservatore. Gli algoritmi di Fuzzy C-means (FCM) e di Region Growing (RG) sono stati sviluppati in un' interfaccia grafica di MATLAB® in modo da poter scegliere iterativamente i diversi step necessari per ottenere una buona segmentazione. Inizialmente sono state effettuate delle operazioni di pre-processing per migliorare la qualità dell'immagine: filtro gaussiano per rimuovere il rumore ed operazione morfologica di opening per uniformare lo sfondo. Il fantoccio PET (NEMA IEC Body Phantom) è stato utilizzato per testare gli algoritmi in condizioni note. L'accuratezza dei processi é stata valutata considerando la variazione percentuale tra il volume calcolato ed il volume teorico, che risulta essere sempre nulla entro l'errore e raggiunge il valore più elevato per la sfera più piccola a causa dell'effetto di volume parziale, diminuendo, generalmente, all'aumentare della dimensione delle sfere. Per il test clinico sono state utilizzate 16 immagini PET di pazienti con tumore testa-collo. L'efficienza è stata stimata usando due indici quantitativi: il coefficiente di similarità di Dice (DSC) e la distanza media di Hausdorff (AHD). I valori di DSC ed AHD, ottenuti mediando su tutti i casi considerati, rientrano nel limite della letteratura (0.6 per DSC e circa 16 *mm* per AHD). Il tempo richiesto per segmentare tutte le fette di ogni immagine varia da qualche secondo per l'algoritmo FCM a qualche minuto per quello RG, rimanendo comunque sempre inferiore al tempo necessario per svolgere la segmentazione manuale. I risultati sono soddisfacenti, ma potrebbero essere migliorati aumentando il numero dei pazienti e testando la variabilità di contornazione tra più esperti per avere un parametro di confronto aggiuntivo. FCM potrebbe essere sperimentato su altre tipologie di tumori, ad esempio i linfomi, per testare l'efficienza nella segmentazione di regioni dislocate.

Abstract

The aim of this work, performed at Azienda Ospedialiero Universitaria in Modena, is the implementation and validation of autosegmentation methods of head and neck (H&N) tumor PET images. These autosegmentation processes are important mostly to overcome the problems of manual segmentation, performed by radiotherapist physician, regarding the contouring time (that can reach more than two hours) and the intra-observer and inter-observer variability. Fuzzy C-means (FCM) and Region Growing (RG) algorithms were developed in a MATLAB® GUI that allows to choice iteratively the different steps necessary for a good segmentation. Pre-processing operations were previously applied to improve image quality: a gaussian filter to remove noise and an opening morphological operation to uniform background. NEMA IEC body phantom, acquired with four hot spheres and two cold spheres, was firstly used to test the two methods in known condition. The accuracy of processes was evaluated considering the volume change between calculated and theoretical volume that is always null within error and reaches the highest value in the case of the smallest sphere because of partial volume effect, generally decreasing as sphere size increases. Afterwards, 16 PET images studies of H&N tumors were used for clinical test of algorithms. The efficiency was estimated using two quantitative coefficients: Dice Similarity Index (DSC) and Average Hausdorff Distance (AHD). Mean DSC and AHD values, obtained mediating on all cases, are within literature threshold (0.6 for DSC and about 16 mm for AHD). Contouring time, required to segment all slices of each case, changes from few seconds in FCM to some minutes in RG, always remaining inferior to manual segmentation time. The results are satisfactory, however, they could be improved increasing the number of patients and testing the variability between more experts to have a further comparison parameter. FCM could be also applied to other cases, e.g. lymphomas, to test the efficiency in the segmentation of displaced regions.

"Je suis de ceux qui pensent que la science est d'une grande beauté. Un scientifique dans son laboratoire est non seulement un technicien : il est aussi un enfant placé devant des phénomnès naturels qui l'impressionnent comme des contes de fés."

Maria Skłodowska

Contents

Introduction

In the last few years, Positron Emission Tomography (PET) has assumed an important role in radiation treatment planning. An accurate segmentation of disease region is very relevant to enable objective follow-up, optimize radiotherapy planning and evaluate the therapy. Principal segmentation modality is manual segmentation carried out by a radiotherapist physician; it presents some limitations correlated with time necessary to contour region and variability between the same expert and different experts. To overcame these problems, new automatic segmentation techniques have been developed.
This work, performed at Azienda Ospedaliero-Universitaria in Modena, deals with the implementation of two auto-segmentation algorithms, i.e. Fuzzy C-means and Region Growing on PET images (obtained by means of a GE Discovery STE scanner). They were set up in a Graphical Matlab Interface to allow the interactive selection of right segmentation parameters and to collect all actions in only one application. Before contour delineation, pre-processing operations were performed to partially remove image degradation mainly related to noise and non uniform background.
The accuracy of algorithms was initially tested on a sperimental PET phantom (NEMA IEC body phantom), which consists of six spheres, comparing the theoretical sphere volume with the same obtained after the segmentation process. Afterwards, 16 studies of head-neck (H&N) tumors were considered to perform a clinical testing on the two methods: quantitative indices were used to compare manual and automatic segmentation (Dice Similarity Coefficient and Average Hausdorff Distance). Some different parameters were taken into account to compare the efficiency of the two methods.

The thesis project is divided in four chapters:

- Chapter I gives a brief introduction on PET images and their degradation, followed by a comparison between manual and automatic segmentation and a mention on H&N tumor segmentation with some reference examples.

- Chapter II contains an explanation of algorithms, pre-processing operations and quantitative indices used to test methods accuracy (with related codes in the Appendices). A description of the developed MATLAB® GUI is also present.

- Chapter III reports in detail the results of sperimental and clinical tests, particularly a comparison between accuracy of the two methods.
- Chapter IV contains a discussion on the achieved results, possible algorithm improvement and future developments.

Chapter 1

Segmentation in Positron Emission Tomography

Positron Emission Tomography (PET) is a non-invasive functional imaging technique that plays a valuable role in oncological field. It allows to investigate a disease providing molecular information about metabolic body processes in addition to structural informations given also by other imaging techniques as Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) [1]. In the last few years, the research of optimal automatic segmentation methods of PET images has become increasingly important: it is necessary to outline lesions preserving all its informations and also overcome limitations of manual segmentation.

1.1 PET Images

PET images allow to obtain quantitative information about different diseases detecting photons emitted by a radiotracer, localized in more active regions (the abnormal cells)[1][2]. An accurate segmentation of images is necessary in order to discriminate abnormal tissue from surrounding regions and have a correctly detection of it. However, segmentation process is not very simple because of low spatial resolution and noise of images [3].

Spatial resolution is computed as Full Width at Half Maximum (FWHM) of Point Spread Function (PSF) and it represents the smallest distance at which two points can be considered separated [4] (Figure 1.1). It generally depends on geometry and size of detector elements (detector design) and also on reconstruction process. A direct consequence of low spatial resolution is the *Partial Volume Effect* (PVE)[5]: the counts of a small source are distribuited on a volume larger than the real one (Figure 1.2a) or radiotracer distribution is not correctly sampled on voxel grid (most voxels include different types of tissues)(Figure 1.2b).

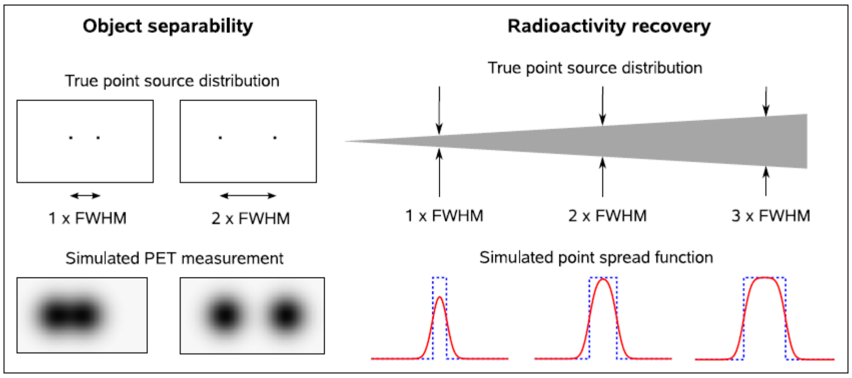

Object separability
True point source distribution
1 x FWHM
2 x FWHM
Simulated PET measurement
Radioactivity recovery
True point source distribution
1 x FWHM
2 x FWHM
3 x FWHM
Simulated point spread function

Figure 1.1: Objects separability in function of spatial resolution: two points are separable if their distance is greater than the spatial resolution(FWHM) [4].

This effect occurs when detail size is smaller than 2-3 FWHM and causes the decrease of region intensity and the "spilling out" of signal.

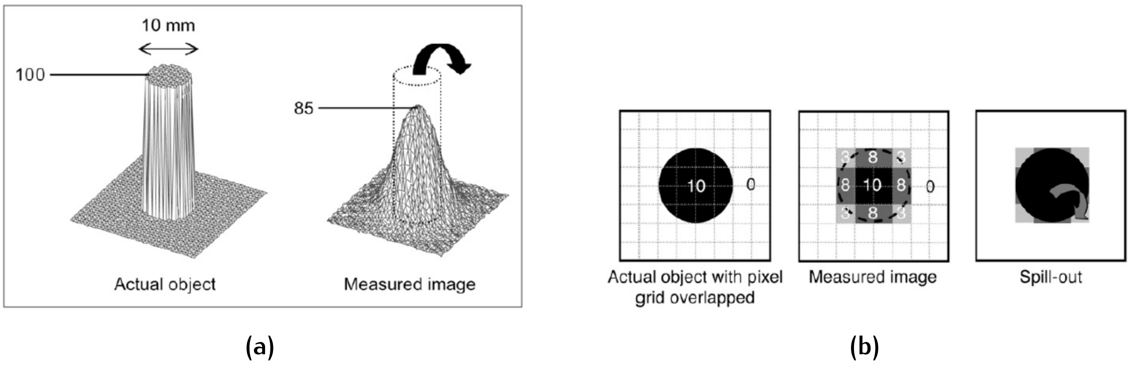


Figure 1.2: (a) Circular source of 10 *mm* and measured image: part of signal is seen outside actual source and maximum activity is reduced to 85. (b) Image sampling affected by PVE: also in this case the signal is outside the actual object producing "spilling out" [5].

PET noise is also correlated to detector, in particular to electronic system, and to random events [1]. It causes a reduction of contrast and makes image more blurred. Some studies have demonstrated that PET image noise generally presents a *gaussian distribution*. J.A. Fessler compares Poisson and Gaussian distribution to fit counts proving, evaluating χ^2, the most accuracy of gaussian one [6]. Mou, Huang and O´ Sullivan [7] prove the gaussian data distribution, that also affect noise, after FBP (Filtered Back Projection) image reconstruction as visible in Figure 1.3.

[1] Random events represent two photons detected simultaneously but that belog to different annihilation events.

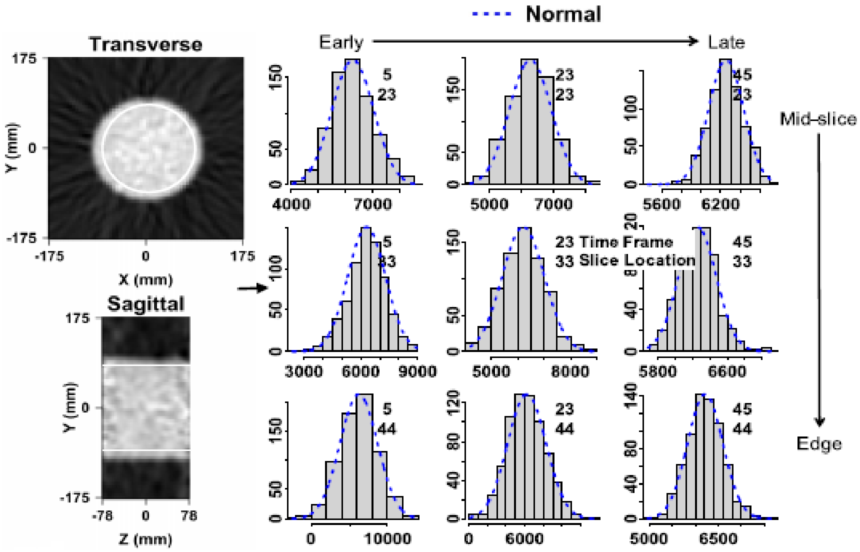


Figure 1.3: PET data fit with normal distribution referred to different time frames and slices location. Left: cross-section and sagittal image. Right: histograms generated from regions of interest[7].

Pre-processing operations, before segmentation, are very important to partially remove these effects. As described in chapter 2.2 a gaussian filter and a morpholofical operation were applied to adjust images. The first is used to remove noise; the second to uniform background illumination and so to sharpen intensity of object of interest restricted by PVE.

1.2 Manual and automatic segmentation

Segmentation process allows to repartition the image in different meaningful segments that correspond to the *regions of interest (ROIs)* namely the pathologies to investigate. The standard segmentation is *manual segmentation* defined as *"Gold Standard Method"* and performed by radiotherapist physician.
Generally during the lesion contours definition, three differents main volumes are distinguished [8]:

- *Gross Tumour Volume (GTV)* is the gross palpable or visible/demonstrable extent and location of malignant growth;
- *Clinical Target Volume (CTV)* is the tissue volume that contains a demonstrable *GTV* and sub-clinical microscopic malignant disease so the area directly surround-

ing the GTV that may contain microscopic disease and other areas considered to be at risk and require treatment;

- *Planning Target Volume (PTV)* is an extension of *CTV* and presents an additional margin to take into account setup uncertainties, machine tolerance and intra-treatment variations.

Figure 1.4a is a graphical representation of these three volumes while Figure 1.4b shows 3D visualization of them.

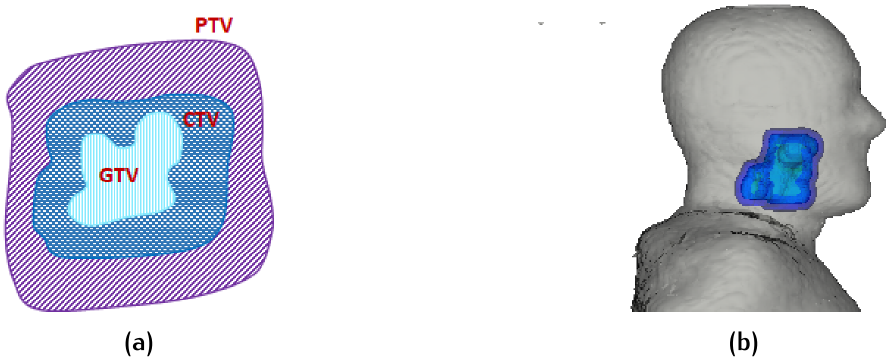


Figure 1.4: Graphic representation of *GTV*(light blue), *CTV* (blue), *PTV* (violet) volumes (a) and 3D visualization of the same volumes referred to one case of head-neck tumor (b).

Gold Standard segmentation method is usually performed on *CT* image and, after the PET-CT co-registration[2], it is extended to PET images too. This *manual segmentation*, required to outline the region to treat, presents some limitations:

1. The time needed to segment the ROI in all slices of an image is variable between 30-120 *min* and it depends on dimension and localization of lesion.

2. The contours can be different based on the operator that provides to segmentation.

Therefore, it is *time-consuming* and the results may *not be reproducible* or suffer from intra-observer and inter-observer variability [9].
During the last decades, new mathematical and physical algorithms have been developed to overcome these problems performing the *automatic segmentation* of images. Nowadays, differents segmentation algorithm exist [10] [11] based on:

[2] The co-registration technique allows to spatially alligned the images using the overlap of same specific points that generally correspond to specific features. The co-registration is very important to combine different information (morphological information of CT and metabolic information of PET) of different acquisition modalities to improve the diagnostic potentialities of each technique.

- the choice of intensity threshold value to discriminate background to foreground (e.g. thresholding, adaptive thresholding, etc.);
- image partition in different clusters (e.g. Fuzzy C-means, K-means, etc.);
- growing of image regions or image splitting(e.g. region growing, adaptive region growing, split and merge, etc.);
- edge detection (e.g. gradient or laplacian based methods, Canny algorithm, etc.).

The choice of one method or another depends on lesion to segment and on type of image.

1.3 Head-Neck tumors segmentation

PET has become an essential tool for clinical management of head-neck ($H\&N$) tumors [12]. Staging of these cancers is related to invasion of first cancer site (stage T), localization of regional nodes (stage N) and metastases identification (stage M) that is a spread outside $H\&N$ regions. Then, the accurate localization of malignant neoplasm and the eventually detection of lymph node metastases is very important first in the prognosis and, secondly, in the therapeutic procedure that generally requires a segmentation phase to outline disease regions [13]. Segmenting $H\&N$ lesions in PET images is demanding: each part of the lesion must be segmented and individually quantified.
Some studies have been developed to segment this type of disease. Schinagl, Vogel et al. [14] compare five different threshold methods used to segment $H\&N$ lesions related to 78 patients cases. Figure 1.5 shows CT (A) and PET (B) image with six different GTV

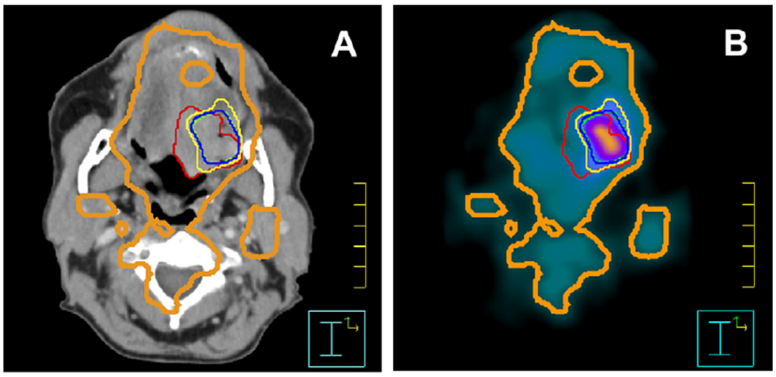


Figure 1.5: Segmented lesion on CT (A) and PET (B) image: GTV_{CT} in red, GTV_{vis} in light green, GTV_{SUV} in orange, $GTV_{40\%}$ in yellow, $GTV_{50\%}$ in blue, GTV_{SBR} in dark green. Note that $GTV_{50\%}$ and GTV_{SBR} are indistinguishable. GTV_{SUV} was failed including large areas of background tissue [14].

contours related to CT segmentation (GTV_{CT}), segmentation by visual interpretation (GTV_{vis}), isocontour using Standardized Uptake Value $SUV = 2.5$ (GTV_{SUV}), threshold at 40% ($GTV_{40\%}$) and 50% ($GTV_{50\%}$), threshold based on signal to background ratio (GTV_{SBR}).
Instead, Geets, Lee et al. [15] propose a gradient-based method that provides a more accurate volume estimate compared to threshold segmentation developed by Daisne et al.[16]: the results are shown in Figure 1.6.

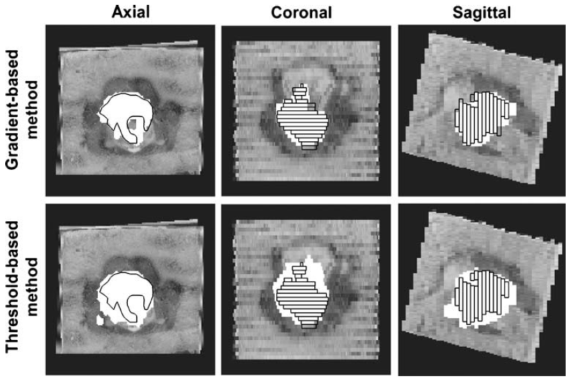


Figure 1.6: Comparison between gradient-based and threshold-based segmentation: black line identifies the standard volume of macroscopic specimen, while white area the calculated volume on PET images [15].

Recently, also a graph-based segmentation method has been developed to reduce intra and inter-operator variability as in Beichel et al. [12] study (Figure 1.7).

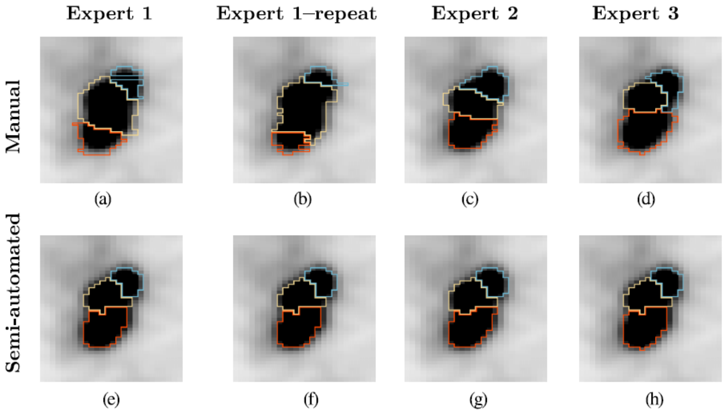


Figure 1.7: Comparison between manual slice-to-slice segmentation and graph based segmentation performed by three different experts and repeated two times by the same one expert [12].

In this work two different region-based algorithms were implemented to segment $H\&N$ lesion PET images of 16 patients and compare the results to those obtained by manual segmentation performed by an expert.

Chapter 2

Segmentation process and validation

In this thesis work, PET images segmentation was performed with two different methods: Fuzzy C-means algorithm and Region Growing algorithm. The accuracy of these two processes was estimated using quantitative methods: Dice Similarity Coefficient and Average Hausdorff Distance. All images were subjected to pre-processing operations to eliminate degradation effects: gaussian filter and morphological opening. Each step was implemented in a MATLAB® GUI described in section 2.4.

2.1 Algorithms description

2.1.1 Fuzzy C-means

Fuzzy C-means (FCM) is a clustering algorithm ([17], [18]) that allows to put each data point in a specific cluster. It is possible thanks to the minimization of an *objective function* that depends on distance between each point and each cluster center:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} ||x_i - c_j||^2 \tag{2.1}$$

where m is any real number greater than 1, u_{ij} is the membership degree of each point x_i in the cluster j, x_i is the i-th data point, c_j is the center of j-th cluster and $||\ ||$ is the norm that is expression of similarity between points and clusters center. N is the total number of points and C is the number of clusters [19].
The algorithm is iterative and each iteration provides to update the membership degree and the cluster center in order to optimize the Equation 2.1:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{||x_i - c_j||}{||x_i - c_k||})^{\frac{2}{m-1}}} \tag{2.2}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{2.3}$$

The process is stopped when the following condition is verified:

$$max_{ij}|u_{ij}^{k+1} - u_{ij}^k| < \epsilon \tag{2.4}$$

where ϵ is a termination criterion and its value is between 0 and 1, while k represents the number of steps.
Then the algorithm consists of four subsequent steps, outlined in Figure 2.1:

1. Random initialization of all membership values u_{ij} generally grouped in the membership matrix U;
2. Calculation of each cluster center c_j starting from u_{ij} values with equation 2.3;
3. At each k-th step:
   - updating of u_{ij} with equation 2.2;
   - updating of J_m with equation 2.1.
4. Checking equation 2.4: if the condition is verified the algorithm is stopped else the process returns to step 2.

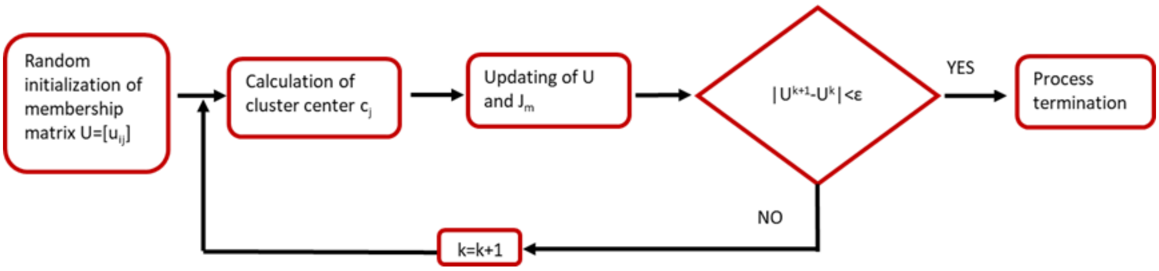

Random initialization of membership matrix U=[u_ij]
Calculation of cluster center c_j
Updating of U and J_m
|U^{k+1}-U^k|<ε
YES
Process termination
NO
k=k+1

Figure 2.1: Flow chart of FCM algorithm steps.

The FCM algorithm was implemented in a Matlab script and enclosed in a GUI: the description is reported in the Appendix B.1.

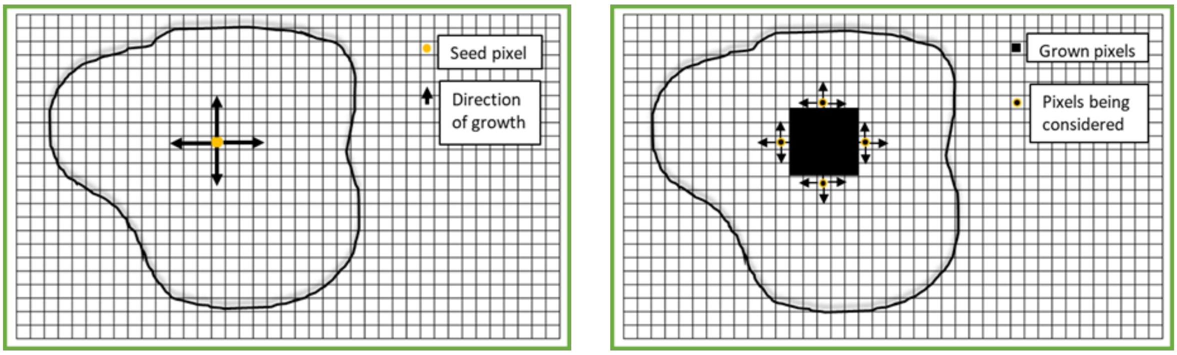
2.1.2 Region Growing

Region Growing (RG) is a segmentation algorithm that allows to partition image in some regions starting to one or more seed pixels [20]. Initial region consists of initial seed; then it grows including neighboring pixels (of seed) that satisfy a similarity criterion (that can be based on intensity, variance, color, shape, texture, size, etc.):

$$|z(x_i, y_i) - z(x_{seed}, y_{seed})| \leq \delta \tag{2.5}$$

where $z(x_i, y_i)$ is a specific feature of neighboring pixels, $z(x_{seed}, y_{seed})$ is the same feature of initial seed and δ is an upper bound to feature difference.
The growth can occur in every directions and it stops when the condition in equation 2.5 is not more satisfied by the new adjacent point. The final region will consist of neighboring pixels that are similar to seed.
Then the algorithm can be summarized in these following steps (schematization in Figure 2.2):

1. Choice of one or more initial seeds;
2. Choice of similarity criterion;
3. Addition of neighboring pixels to the region (or regions) if they satisfy the criterion expressed by equation 2.5;
4. Repeat point 3 until the condition in equation 2.5 is verified.



(a) Starting of growing a region.

(b) Growing process after a few iterations.

Figure 2.2: Schematization of RG process.

Each initial seed can be selected iteratively or automatically. If there are more regions, they must be disjoint because a single point cannot belong to different regions[20] [21]. In this work the chosen feature for growing region is pixel intensity, set in values between 0 and 1. The script is reported in Appendix B.2.

2.2 Pre-processing operation

2.2.1 Gaussian filter

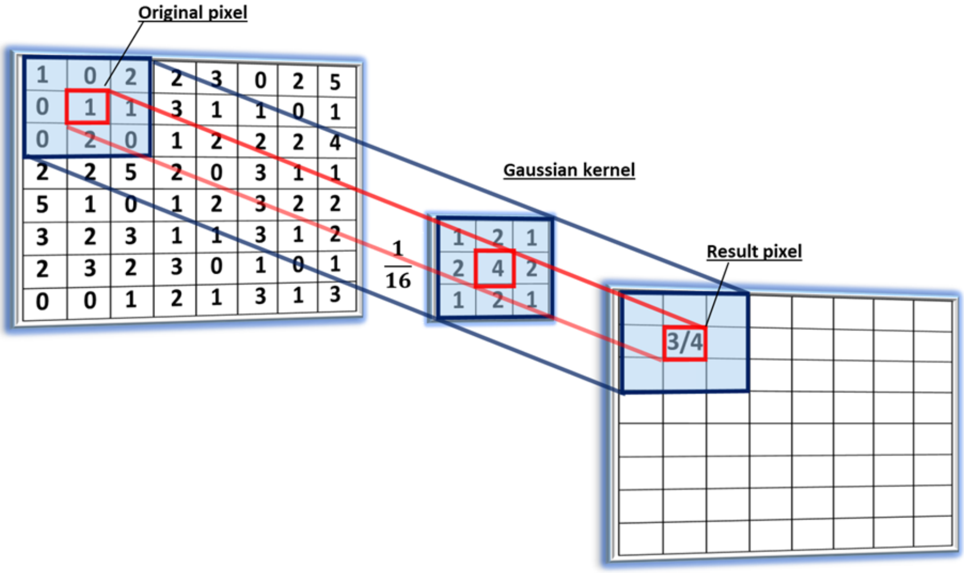
Gaussian filter plays an important role in image processing and it is generally used to remove noise [22].

Image filtering consists in a convolution between the same image and a kernel built starting from gaussian 2D function:

$$G(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.6}$$

where σ represents the standard deviation of distribution and the mean value is assumed as zero.

The convolution process [23] is visible in Figure 2.3: the new value of each pixel is calculated using a weighted average on neighboring pixels. Every time the pixel to change is at the center of kernel and it assumes the highest value given by the gaussian function.



1	0	2	2	3	0	2	5
0	1	1	3	1	1	0	1
0	2	0	1	2	2	2	4
2	2	5	2	0	3	1	1
5	1	0	1	2	3	2	2
3	2	3	1	1	3	1	2
2	3	2	3	0	1	0	1
0	0	1	2	1	3	1	3

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

Figure 2.3: Convolution process of image with a gaussian kernel.

The choice of σ is correlated with kernel dimension: mask size must increase with sigma value to preserve more image informations [24]. As visible in Figure 2.4, a mask radius of 3σ guarantees the conservation of over 99% of data information, a radius of 2σ over 95%, while a radius of σ around the 68%. The increase of kernel dimension allows to

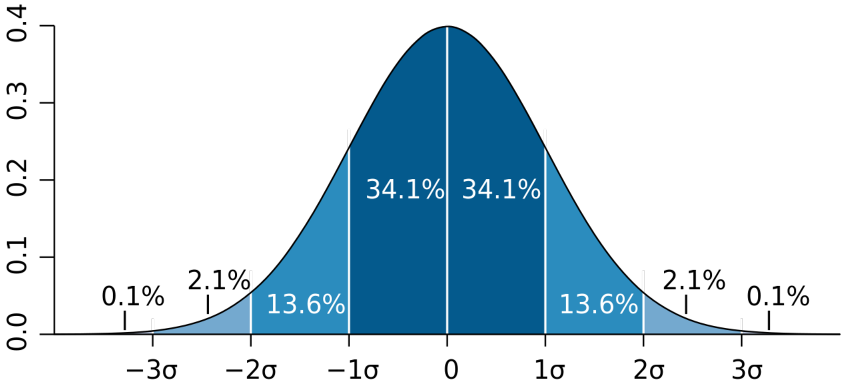
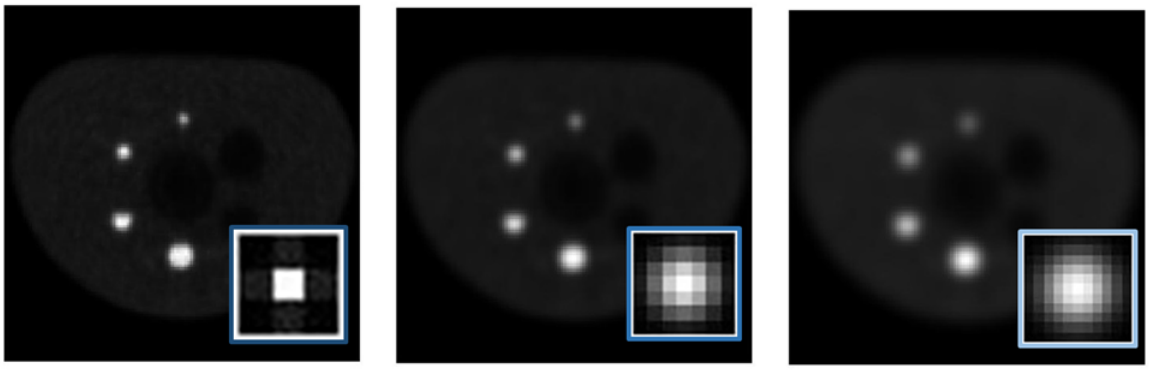


Figure 2.4: Gaussian distribution with different range in function of standard deviation value.

reduce more the noise, but also generates a more blurred image eliminating details: then a right selection of size is fundamental. A representation of a gaussian filter applied to PET phantom with three different sigma values is shown in Figure 2.5.



(a) $\sigma = 0.5$ **(b)** $\sigma = 1.5$ **(c)** $\sigma = 2.5$

Figure 2.5: Application of gaussian filter to PET phantom image with $\sigma = 0.5$ (a), $\sigma = 1.5$ (b) and $\sigma = 2.5$ (c).

In this work, a gaussian 2D filter was applied to PET images, that are generally affected by gaussian noise, changing σ value and, consequently, kernel dimension: $d = 2 \cdot (2\sigma) + 1$ (rounded to the nearest integer). In this way about 95% of information was preserved and both dimensions are considered (multiplication for the first 2). The addition of 1 guarantees the creation of an odd size kernel. The code, inserted to GUI and used to apply this pre-processing operation, is reported in Appendix B.3.

2.2.2 Morphological opening operation

Morphological opening is an operation that allows to preserve, in an image, the objects with the same shape of structuring element, smoothing the contours and eliminating the smaller details[25]. It consists in a combination of erosion and dilation operations:

$$A \circ B = (A \ominus B) \oplus B \tag{2.7}$$

where A is the image, B is the structuring element, $\circ$ is opening symbol, $\ominus$ is erosion symbol and $\oplus$ is dilation symbol. Opening process is visible in Figure 2.6: structuring element B is rolled inside the boundary of image A, the erosion process removes the details that are smaller than B, while the successive dilation process restores the shape of remaining objects.

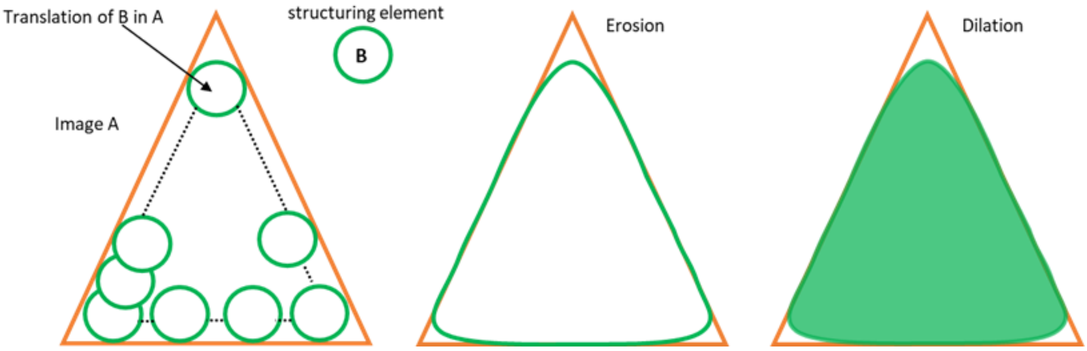
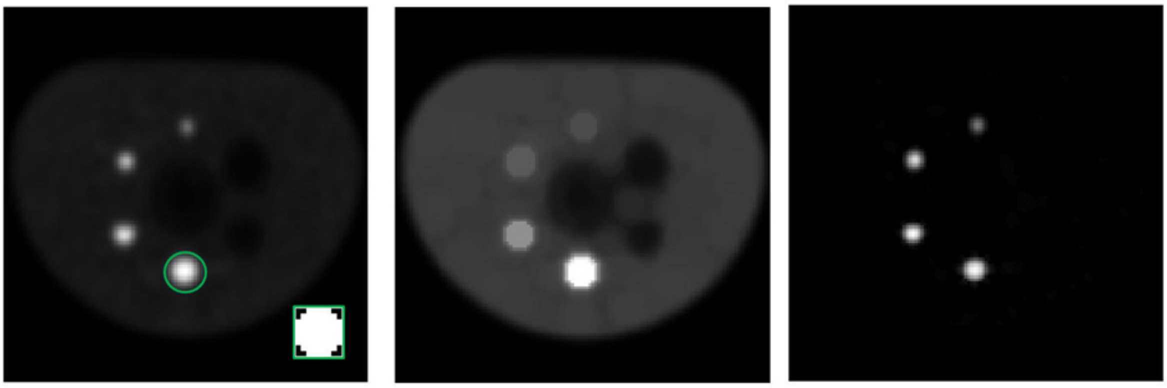


Figure 2.6: Opening operation of image A by structuring element B: translation of B in A, image A after erosion and image A after dilation (in green).

Opening operation can be used to correct non uniform background illumination and identify better foreground object. It occurs throught the following steps [26]:

1. Select a structuring element similar to or larger than the size of object to preserve;
2. Apply opening operation to original image by the structuring element previously built to obtain non uniform background;
3. Subtract the non uniform background from original image to obtain the original image with uniform background.

In Figure 2.7 there is a representation of removing non uniform background process applied to PET phantom image: the original image (Figure 2.7a) is restored subtracting from it the background (Figure 2.7b); the result is an image with uniform background(Figure 2.7c).



(a) Original Image. **(b)** Background. **(c)** Uniform Image.

Figure 2.7: Non uniform background removal process: (a) Original Image and structuring element in green of size as the circled sphere (9×9 pixels); (b) Background obtained after opening operation; (c) Original image with uniform background obtained after subtracting operation.

After checking the non uniformity of background (Appendix C) and applying gaussian filter, morphological opening operation was applied on PET images selecting the object of interest: the code is described in Appendix B.4.

2.3 Validation criteria

Dice Similarity Coefficient and Average Hausdorff distance were used to compare automatic segmentation (obtained by algorithms) with manual segmentation (gold standard) computing similarity parameters between the two masks.
Dice Similarity Coefficient (DSC) is a quantitative index used to measure the spatial overlap between two regions. It is defined as [27]:

$$DSC = 2\frac{|A \cap B|}{|A + B|} \tag{2.8}$$

where A and B represent the two regions to compare. DSC can take values between 0 and 1 that indicate no overlap and total overlap respectively. A graphical schematization is visible in Figure 2.8.

Hausdorff distance (HD) measures the spatial distance between two point sets. It is defined as [28]:

$$H(A, B) = max(\breve{H}(A, B), \breve{H}(B, A)) \tag{2.9}$$

where A and B are the two point sets. $\breve{H}(A, B)$ is the direct Hausdorff distance that represent the maximum of distances between each point $x \in A$ to its nearest neighbor

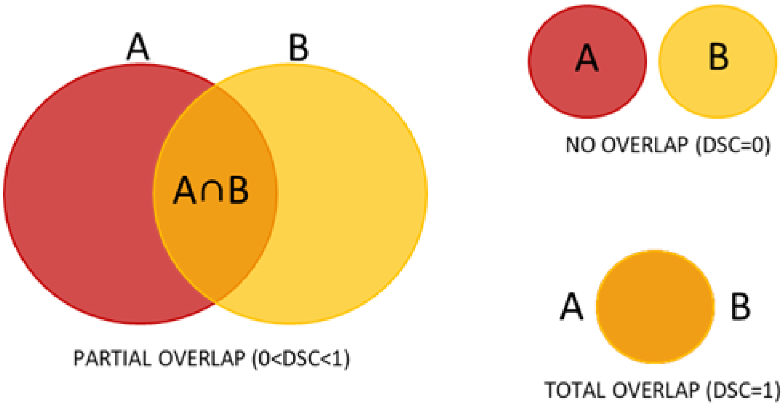


Figure 2.8: Schematization of DSC calculation basing on overlap between region A and B. Different overlap cases: no overlap (DSC=0), partial overlap (0<DSC<1), total overlap (DSC=1).

$y \in B$:

$$\breve{H}(A,B) = max_{x \in A} min_{y \in B} ||x, y|| \tag{2.10}$$

where || || is any norm, e.g. euclidean distance. A graphical schematization is visible in Figure 2.9.

Generally, in validation segmentation methods, HD is replaced by AHD (Average Hausdorff distance) obtained mediating on all slices to minimize edges segmentation errors. DSC and AHD were calculated using MATLAB codes reported in AppendixB.5.

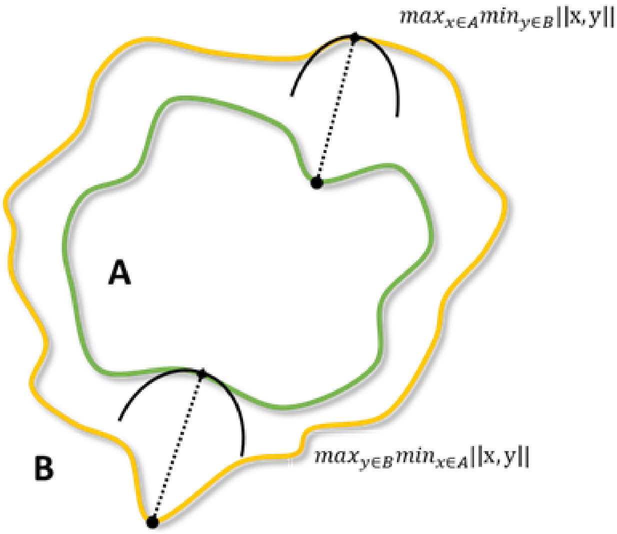

$max_{x \in A} min_{y \in B}||x,y||$
A
B
$max_{y \in B} min_{x \in A}||x,y||$

Figure 2.9: Schematization of Hausdorff distance calculation between two point sets A and B.

2.4 MATLAB® GUI

In this work, a MATLAB® GUI (graphical user interface) called *FeRS_GUI* was implemented to apply, choosing interactively, segmentation processes on PET images. The graphical interface is visible in Figure 2.10:

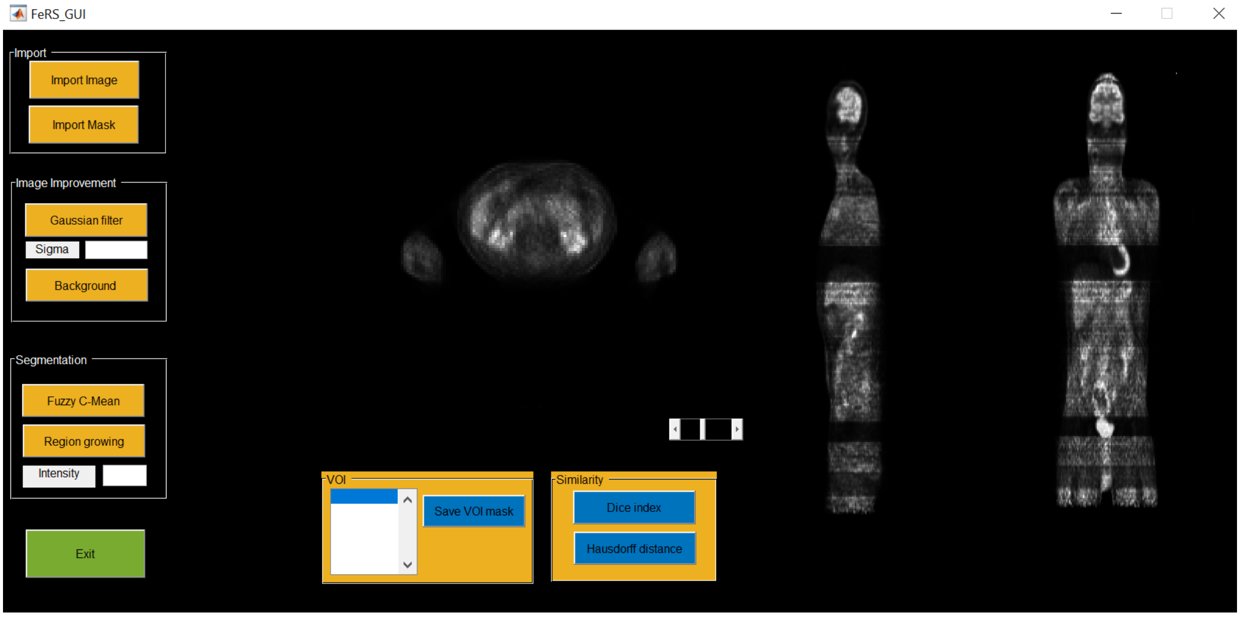


Figure 2.10: FeRS graphical interface for segmentation processes and method validation.

- *Import* panel allows to clicking on:
  - *Import image* to import a PET image that is displayed in the three projections: axial, sagittal and coronal;
  - *Import mask* to import the 3D mask obtained with manual segmentation and after used for comparison.
- *Image improvement* panel, that encloses pre-processing operations, allows to clicking on:
  - *Gaussian filter* to remove noise after the insertion of sigma parameter;
  - *Background* to uniform background using morphological opening operation after the selection of the object of interest.
- *Segmentation* panel allows to segment the image clicking on:
  - *Fuzzy C-means* that applies FCM algorithm after the selection of a cluster;

  - *Region growing* that applies RG algorithm with different intensity upper bound, after the selction of a seed.

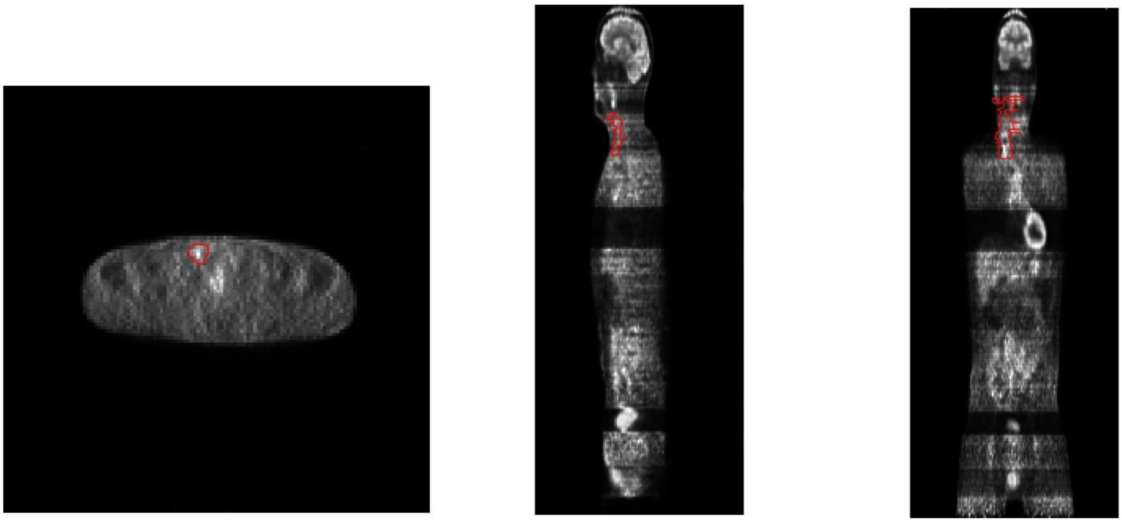
  In both buttons the selection of initial and final slice is also required to segment, but if the mask has already been loaded, the segmentation is carried out automatically on the same slices of mask.

- *ROI* panel presents a listbox that include the created VOI (Volume Of Interest) mask after segmentation and a push-button *Save VOI mask* that allows to save this mask in DICOM format. The name of VOI is composed of type of segmentation method used (FCM or RG) and ID case; e.g. *FCM_1* is the VOI mask referred to case 1 and obtained with FCM segmentation.

- *Similarity* panel allows to compare VOI mask obtained by algorithms with imported mask (gold standard) using DSC, *Dice Index* push-button, and AHD, *Hausdorff distance* push-button. The results are displayed on command window.

- *Exit* closes the GUI.

Some functions scripts are reported in Appendix B, as mentioned previously. The Figure 2.11 shows the result of a segmented slice of an H&N tumor PET image in the three different projections: axial, sagittal and coronal.



(a) Axial projection **(b)** Sagittal projection **(c)** Coronal projection

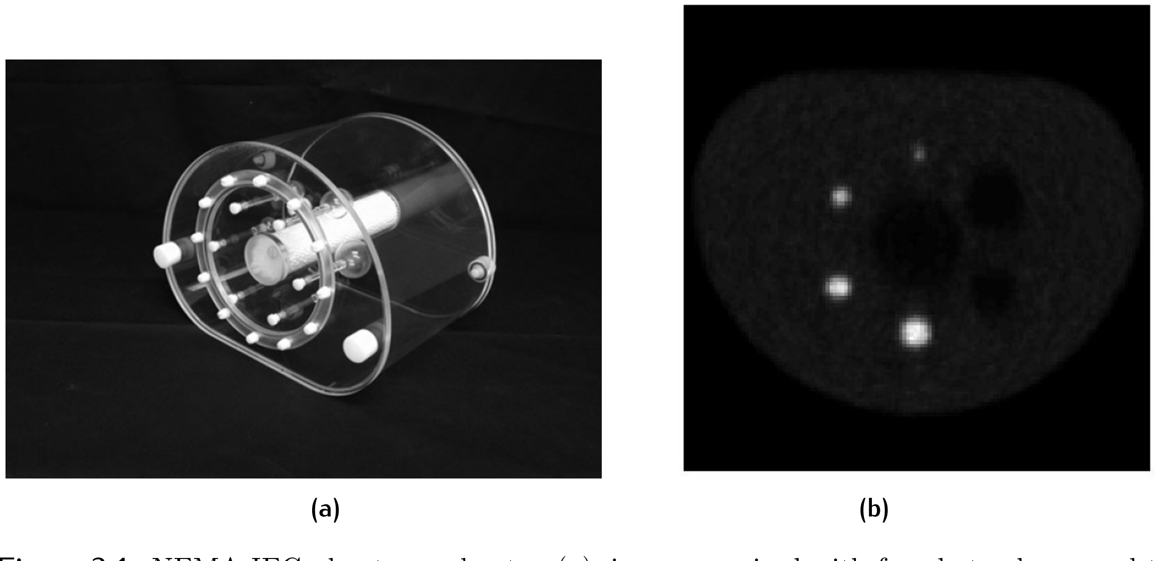
Figure 2.11: Slice segmentation of PET image in three different projections: axial(a), sagittal(b) and coronal(c).

Chapter 3

Data Analysis and Results

3.1 Sperimental test

Firstly, *NEMA IEC* body phantom (Figure 3.1a) was used to test algorithms in known conditions. It was previously acquired with four hot spheres (diameter between 10 and 22 mm) with an activity of 50 kBq/ml, and two cold spheres (diameter of 28 and 37 mm) (Figure 3.1b). The recorded background activity was about 4.7 kBq/ml that is 1/10 of foreground.


(a)

(b)

Figure 3.1: NEMA IEC phantom: phantom(a); image acquired with four hot spheres and two cold spheres (b).

The efficiency of both algorithms was proved comparing the theoretical volume V_{th} of spheres and the calculated volume V_{calc} obtained by segmentation contours. In Table

3.1, the values of V_{calc}, V_{th} and the *percentage change* between these two volumes[1] (V_{ch}) are shown for both methods (FCM and RG) with related error bars[2]. In Figure 3.2 is visible a graph of comparison between FCM and RG volume change.

$d_{th}(cm)$	$V_{th}(cc)$	FCM		RG	
		$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$
1.0	0.5	0.53±0.06	6±12	0.52±0.03	4±6
1.3	1.2	1.13±0.11	-6±9	1.17±0.05	-3±4
1.7	2.6	2.5±0.3	-4±12	2.55±0.07	-2±3
2.2	5.6	5.4±0.6	-4±11	5.5±0.2	-2±4
2.8	11.5	11.1±1.7	-3±15	11.4±1.4	-1±12
3.7	26.5	26.6±0.9	0±3	26.7±1.8	1±7

Table 3.1: Theoretical volume (V_{th}) and theoretical diameter (d_{th}) value, calculated volume (V_{calc}) and percentage change between calculated and theoretical volume for each sphere (V_{ch}). The errors on V_{calc} are calculated as maximum error repeating algorithms three times.

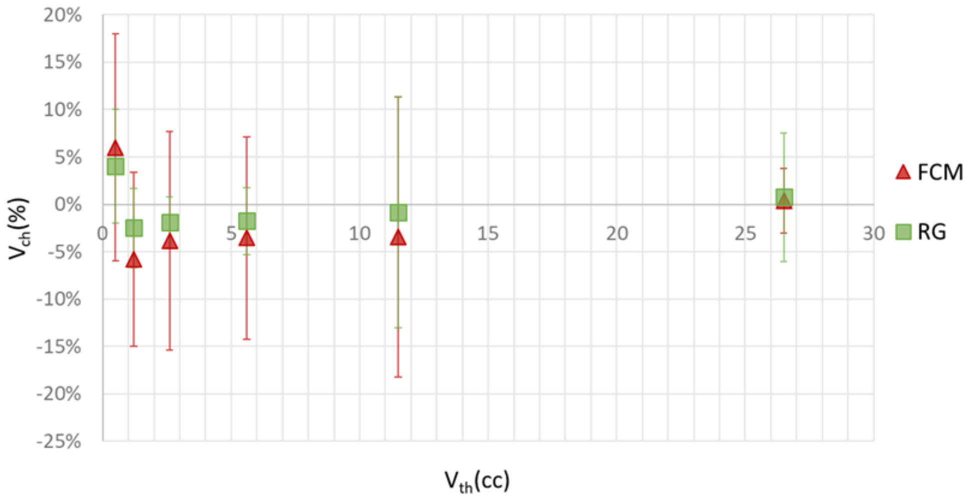


Figure 3.2: Percentage change between theoretical and calculated volume (V_{ch}) in function of theoretical volume of each spheres with error bars in red for FCM and in green for RG.

The volume change (in absolute value) takes values in the range 0-6% for FCM and 1-4% for RG; FCM method is characterized by more variability and its values are most distant from 0% that is the best estimation. Both methods present the biggest V_{ch}

[1] The percentage change is calculated with the equation: $V_{ch} = (V_{calc} - V_{th})/V_{th}$.

[2] The error on theoretical sphere diameters, and so on theoretical volumes, is not declared in the phantom manual (Appendix D) therefore it is considered negligible.

value for smallest sphere because of *partial volume effect* that causes an overestimation of calculated volume respect to theoretical one. Moreover, V_{ch} decreases as dimension of spheres increases: indeed, the segmentation of bigger regions is less difficult because their contours can be discriminated more easily. Furthermore, because MATLAB® GUI (as described in section 2.4) allows to insert different intensity upper bound in RG segmentation process, the best volume change values for RG were obtained testing three different intensity levels: $I = 0.3$, $I = 0.5$ and $I = 0.8$. The results are reported in Table 3.2 while the representation of $|V_{ch}|(\%)$ for three intensity values is shown in Figure 3.3.

V_{calc}	$\|V_{ch}\| \pm \Delta V_{ch}(\%)$		
	$I_{0.3}$	$I_{0.5}$	$I_{0.8}$
0.5	4±6	6±12	6±12
1.2	47±5	10±14	3±4
2.6	60±5	33±6	2±3
5.6	39±4	14±9	2±4
11.5	1±12	1±12	1±12
26.5	1±7	1±7	1±7

Table 3.2: Theoretical volume (V_{th}) and percentage change between theoretical and calculated volume in absolute value ($|V_{ch}|$) for three different intensity limits $I = 0.3$, $I = 0.5$, $I = 0.8$.

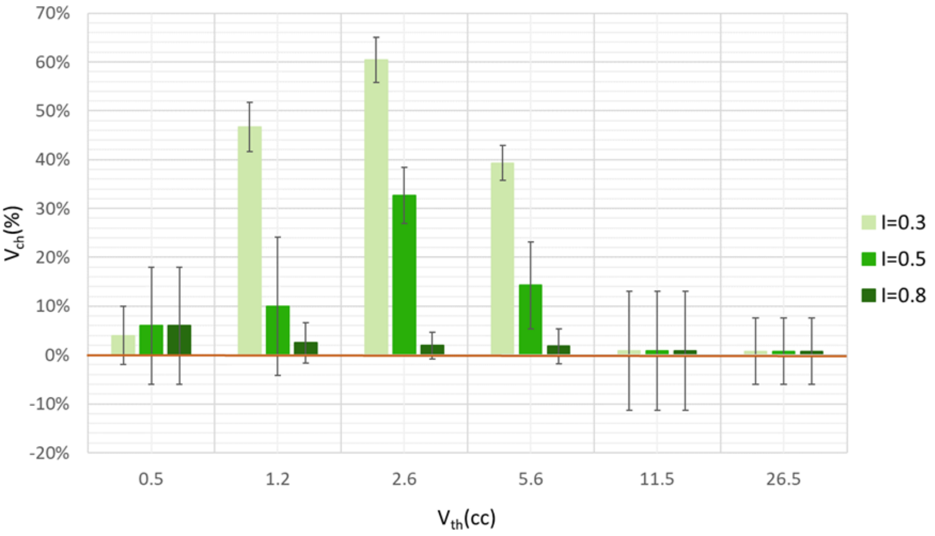


Figure 3.3: The volume change values for six spheres in RG method for three different intensity upper bounds: $I = 0.3$, $I = 0.5$, $I = 0.8$.

The volume change is better when it is close to 0. Medium-size spheres produce the best results with $I = 0.8$. The smallest one prefers an intensity equal to 0.3 and the two biggest spheres show the same results for each intensity value[3]. Generally the intensity values, $I = 0.3$ and $I = 0.5$, tend to generate a smaller VOI because lower intensity values define a region growth more limited and stopped before, so they are suitable for little regions. This does not happen for biggest spheres, that correspond to cold ones, due to the use of the complementary image to segment them which has low intensity values close to each other and therefore the resultant volume (and consequently the volume change) is the same. Comparing FCM and RG algorithm with t-test, applied to a confidence level of 95%, the two methods are not significantly different ($p \gg 0.05$) so they work in the same way; also testing the two methods with *one sample* t-test, assuming that each sample mean is equal to 0 (expectation value), the null hypothesis H_0 is confirmed ($p \gg 0.05$). The graph in Figure 3.4 shows a comparison between the mean volume change value that is $-0.006 \pm 0.028\%$ for FCM and $-0.013 \pm 0.017\%$ for RG.

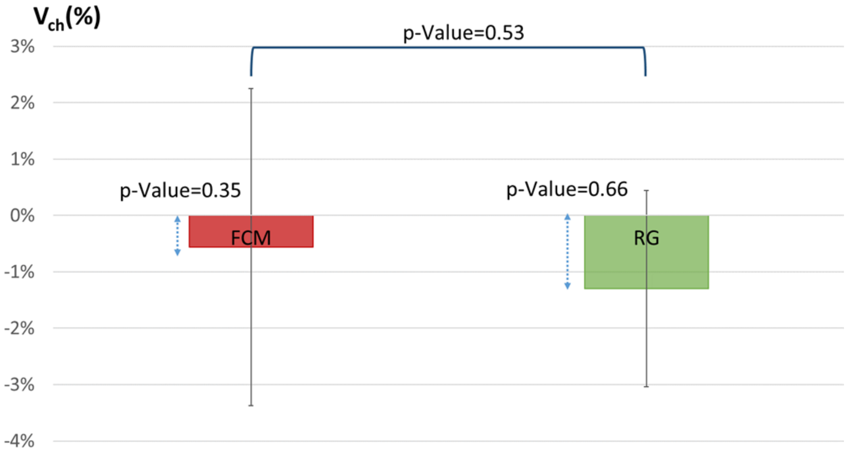


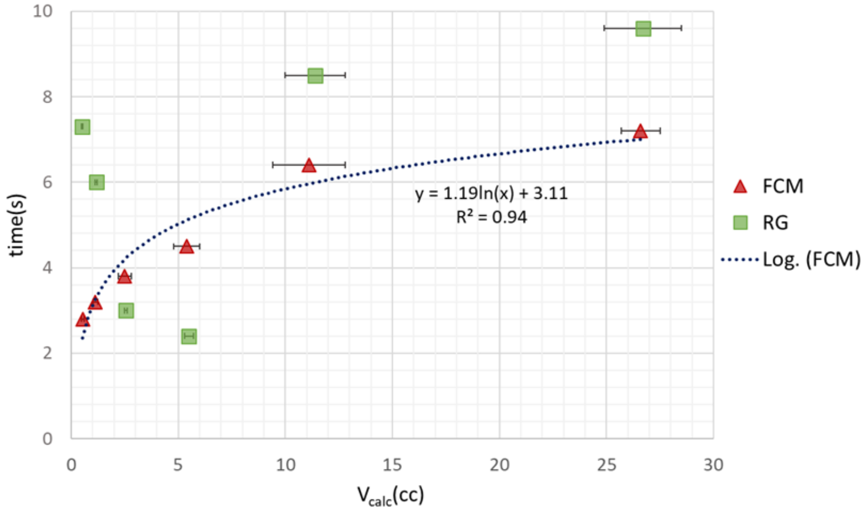
Figure 3.4: Representation of mean volume change value with error bars for both methods. FCM and RG are consistent and they are not significantly different assuming confidence level of 0.05, in fact p-value$= 0.53$. The expectation value, that is 0, is not very distant from two average value: also in this case, the null hypothesis is accepted. Comparing FCM with theoretical value 0, the p-value is 0.35 while for RG a p-value of 0.66 is obtained.

Two methods are consistent with p-value=0.53. The difference of each mean from 0, represented with light blue arrows, confirms the acceptance of H_0 by the one sample t-test with p-value=0.35 for FCM and p-value$= 0.66$ for RG.

The *contouring time* required for automatic segmentation is lower than manual segmentation time and it increases with the volume of region to outline. In Figure 3.5, is

[3] If also the biggest spheres had been hot, the discrimination between the three different intensity values would have been possible.

visible the trend of time as function of volume obtained with the values listed in the below table.



FCM

$V_{calc} \pm \Delta V_{calc}(cc)$	0.53 ± 0.06	1.13 ± 0.11	2.5 ± 0.3	5.4 ± 0.6	11.1 ± 1.7	26.6 ± 0.9
$t_{FCM}(s)$	2.8	3.2	3.8	4.5	6.4	7.2

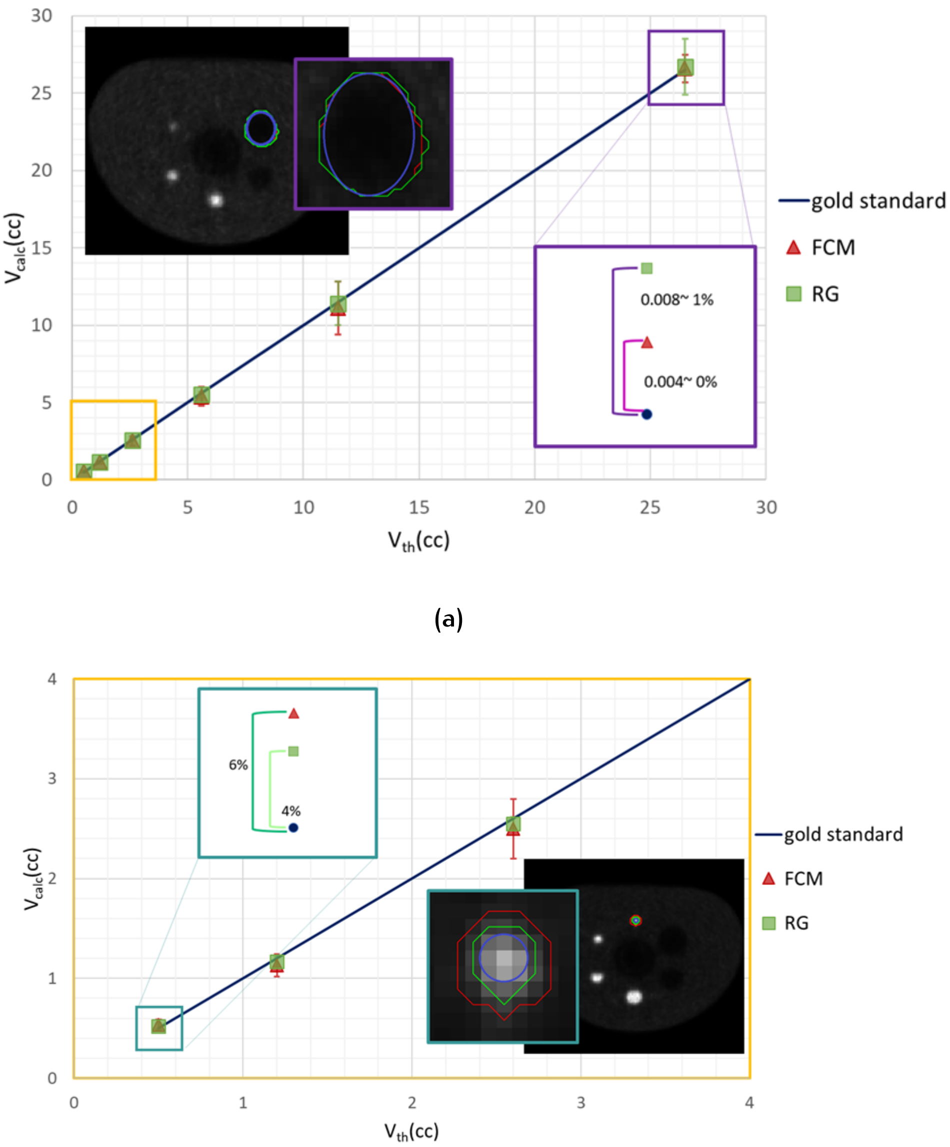
RG

$V_{calc} \pm \Delta V_{calc}(cc)$	0.52 ± 0.03	1.17 ± 0.05	2.55 ± 0.07	5.5 ± 0.2	11.4 ± 1.4	26.7 ± 1.8
$t_{RG}(s)$	7.3	6.0	3.0	2.4	8.5	9.6

Figure 3.5: Trend of time in function of calculated volume and table of values used in the graph. The time error is of the order of 10^{-6} s then it is considered negligible.

The trend is logarithmic($R^2 = 0.94$) for FCM method while the correlation between contouring time and volume is not the same for RG method: the time changes randomly and reaches the lowest values for hot spheres of medium dimension probably because they are high-contrasted.

The values of V_{th} and V_{calc} (Table 3.1) are also considered to represent the trend of calculated volume in function of theoretical one. The graph in Figure 3.6a shows this trend for FCM and RG methods compared to *gold standard* (the diagonal obtained placing V_{th} on x and y axes). Violet square represents the magnification of details for largest sphere of volume over than 26 cc: the distance between FCM and RG point from "gold" point is better visible and it attests an automatic segmentation close to manual one. The contour difference between two methods is outlined in red for FCM, in green for RG and in blue for manual segmentation and it is carried out on original image: it confirms an easier segmentation for large regions. Yellow square represents the magnification of details for volume size between 0-3 cc reported in Figure 3.6b. The greatest distance from gold



(a)

(b)

Figure 3.6: Trend of V_{calc} in function of V_{th} (a). Magnification of trend details for volume size between 0-3 *cc* corresponding to yellow square (b) and between 4-12 *cc* corresponding to light blue square (c).

diagonal and the contours of both methods, related to the smallest sphere, are shown respectively in the green water square and in the image on the graph. The major discrepancy of FCM point to gold point is confirmed and it is justifiable considering partial volume effect for small regions. In each case of intermediate volume value the behavior is the same for both methods and the calculated volume is generally underestimated respect to theoretical volume.

Before testing, *pre-processing* operations are performed to improve the images: *gaussian filter* and *opening* morphological operation. The *sigma* of gaussian function, important to reduce noise, changes also in function of spheres volume. Two algorithms were tested with three dfferent sigma values ($\sigma = 0.5, 1.5, 2.5$) to choose the best $V_{ch}(\%)$ for each sphere. The best value was chosen considering the error associated to volume change and estimating the distance from x-axis 0 value in function of this error (standard deviation). The results are reported in Table 3.3a, 3.3b and represented in Figure 3.7a, 3.7b.

V_{th}	$\sigma_{0.5}$		$\sigma_{1.5}$		$\sigma_{2.5}$	
	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$
0.5	0.53±0.06	6±12	0.54±0.07	8±14	0.70±0.13	40±26
1.2	1.43±0.12	19±10	1.13±0.11	-6±9	1.7±0.4	42±33
2.6	2.1±0.3	-19±12	2.5±0.3	-4±12	2.90±0.19	12±7
5.6	8.5±0.9	52±16	5.4±0.6	-4±11	8.1±0.7	45±13
11.5	/	/	15.2±1.1	32±10	11.1±1.7	-3±15
26.5	/	/	31±3	-17±11	26.6±0.9	0±3

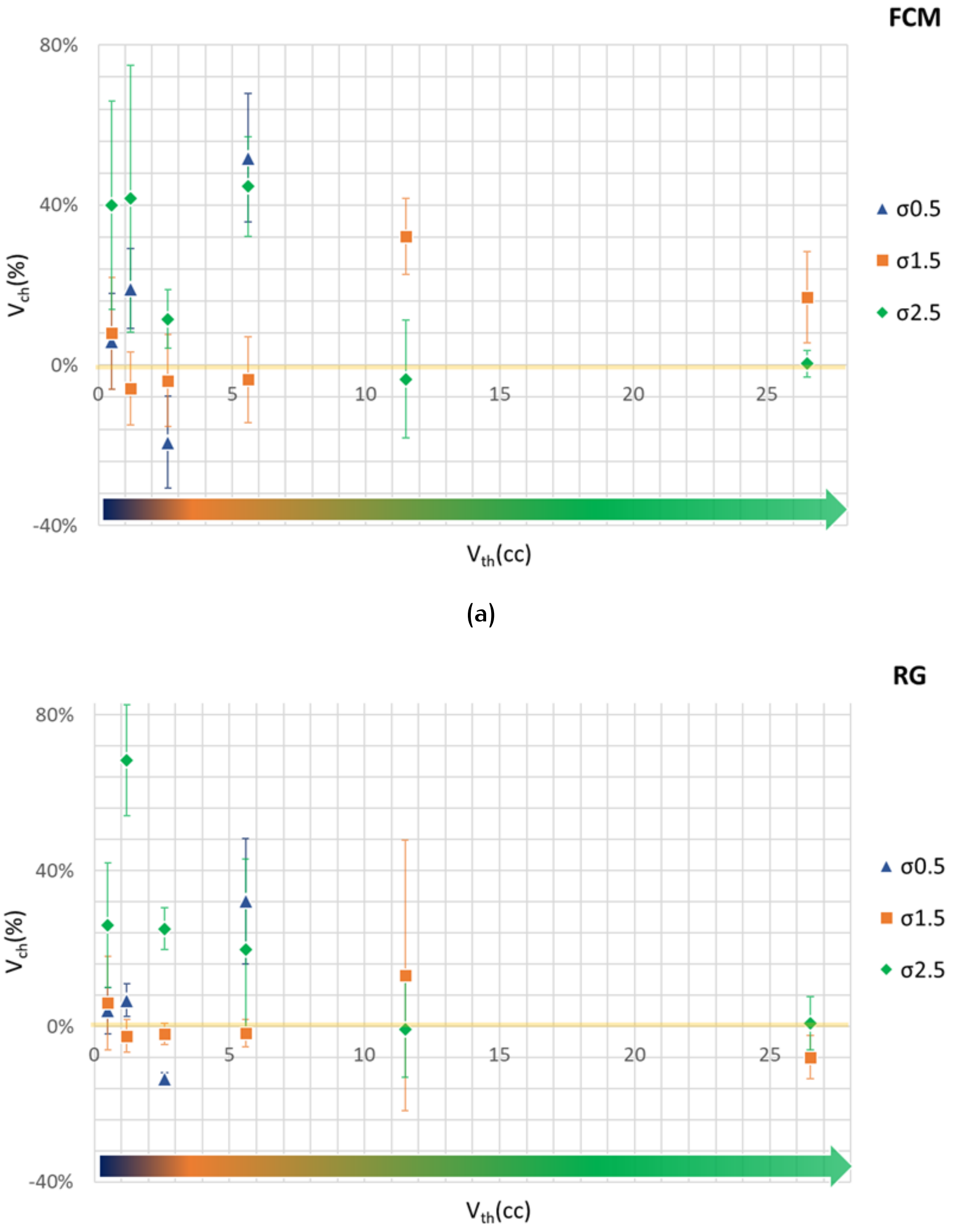
(a) FCM method. Slash bar indicates no sphere segmentation.

V_{th}	$\sigma_{0.5}$		$\sigma_{1.5}$		$\sigma_{2.5}$	
	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$	$V_{calc} \pm \Delta V_{calc}(cc)$	$V_{ch} \pm \Delta V_{ch}(\%)$
0.5	0.52±0.03	4±6	0.53±0.06	6±12	0.63±0.08	26±16
1.2	1.28±0.05	7±4	1.17±0.05	-3±4	2.02±0.17	68±14
2.6	2.25±0.04	-13±2	2.55±0.07	-2±3	3.25±0.14	25±5
5.6	7.4±0.9	32±16	5.5±0.2	-2±4	6.7±1.3	20±23
11.5	/	/	13±4	13±35	11.4±1.4	-1±12
26.5	/	/	24.4±1.5	-8±6	26.7±1.8	1±7

(b) RG method. Slash bar indicates no sphere segmentation.

Table 3.3: Volume difference for three different sigma values in FCM (a) and RG (b) method.

Highlighted values represent the best results: percentage volume change is minimized with sigma value of 1.5 for spheres with a volume in the range between $0.6 - 2$ *cc* instead a sigma of 2.5 is required for sphere volume dimensions over than 2 *cc*. The smallest sphere is rigthly segmented with $\sigma = 0.5$. The arrow, at the bottom of the graph, shows that, increasing the volume, the best σ passes from 0.5 to 1.5 and then to 2.5, indeed the color changes from blue to orange to dark green.



(a)

(b)

Figure 3.7: Calculated volume in function of theoretical volume for three different sigma value (0.5, 1.5 e 2.5) for FCM method (a) and RG method (b). Error bars related to $V_{ch}(\%)$ are shown in blue for $\sigma = 0.5$, orange for $\sigma = 1.5$ and dark green for $\sigma = 2.5$. The points related to biggest spheres for $\sigma = 0.5$ are not present because the algorithms did not segment them.

3.2 Clinical test

16 PET image studies [251-319 images for each study] of H&N tumors were used for clinical test of algorithms. The accuracy of the processes was estimated using quantitative methods as comparison between automatic and manual segmentation, DSC and AHD. Each DSC and AHD value was estimated mediating on most significant slices actually a percentage of slices correctly segmented was extrapolated excluding the slices with DSC value smaller than 0.01. In table 3.4, DSC and AHD values are shown for all studies with corresponding percentage of evaluable slices.

	FCM			**RG**		
ID	$DSC \pm \Delta DSC$	$AHD \pm \Delta AHD(cm)$	$slices_{evaluable}(\%)$	$DSC \pm \Delta DSC$	$AHD \pm \Delta AHD(cm)$	$slices_{evaluable}(\%)$
1	0.64±0.02	1.09±0.06	93	0.62±0.02	1.13±0.04	89
2	0.52±0.02	1.61±0.05	85	0.50±0.02	1.65±0.03	85
3	0.61±0.02	1.46±0.09	76	0.60±0.02	1.46±0.08	76
4	0.56±0.02	1.51±0.06	84	0.50±0.01	1.60±0.07	80
5	0.65±0.02	1.14±0.08	63	0.56±0.03	1.18±0.09	67
6	0.59±0.04	1.29±0.06	79	0.60±0.03	1.13±0.05	79
7	0.62±0.03	1.07±0.05	64	0.61±0.04	1.00±0.05	67
8	0.61±0.04	1.15±0.07	75	0.62±0.04	1.20±0.08	75
9	0.62±0.04	1.09±0.07	75	0.59±0.06	0.97±0.09	58
10	0.62±0.03	1.15±0.08	56	0.62±0.03	1.34±0.08	56
11	0.51±0.02	1.40±0.05	91	0.48±0.02	1.44±0.05	100
12	0.52±0.03	1.20±0.04	73	0.55±0.02	0.96±0.04	87
13	0.62±0.03	1.13±0.06	63	0.65±0.03	1.13±0.06	63
14	0.69±0.04	0.98±0.08	70	0.69±0.03	0.97±0.07	70
15	0.72±0.02	1.08±0.03	85	0.74±0.02	1.00±0.04	89
16	0.64±0.04	1.25±0.07	72	0.63±0.03	1.22±0.07	69

Table 3.4: DSC and AHD values for each case and corresponding percentage of slices correctly segmented.

The lower percentage values correspond to cases that present a very degraded image. However, in both algorithms, 75% of slices are rigthly segmented, as visible in Figure 3.8, and the variability between the two methods is comparable.
In RG segmentation, as was already done in the case of sperimental test, the best DSC value (and the corresponding AHD) was chosen computing it with three different intensity level: $I = 0.3$, $I = 0.5$, $I = 0.8$. The results are shown in Table 3.5 and collected in the histogram in Figure 3.9.

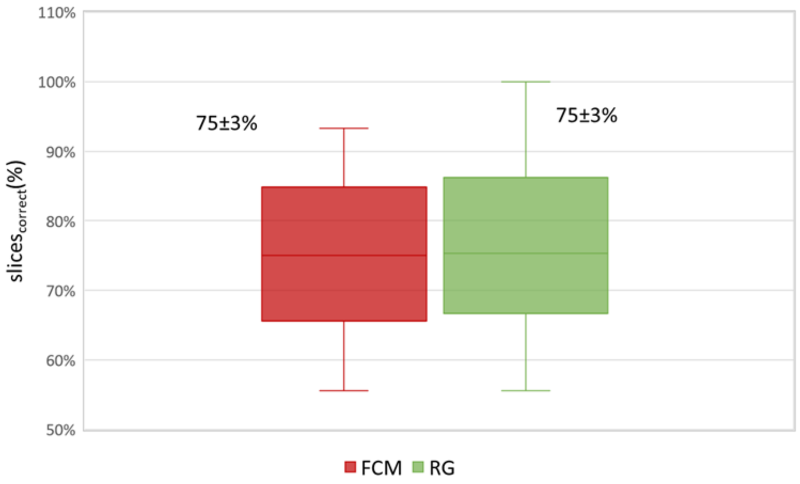


Figure 3.8: Visualization of box plot related to percentage of evaluable slices (values in Table 3.4) of FCM and RG method.

ID	$DSC \pm \Delta DSC$		
	$I_{0.3}$	$I_{0.5}$	$I_{0.8}$
1	0.58±0.02	0.58±0.02	0.62±0.02
2	0.49±0.02	0.43±0.01	0.50±0.02
3	0.59±0.02	0.57±0.02	0.60±0.02
4	0.49±0.02	0.48±0.01	0.50±0.01
5	0.44±0.04	0.54±0.03	0.56±0.03
6	0.59±0.03	0.56±0.02	0.60±0.03
7	0.48±0.02	0.48±0.02	0.61±0.04
8	0.49±0.03	0.60±0.04	0.62±0.04
9	0.59±0.06	0.57±0.03	0.57±0.03
10	0.54±0.03	0.51±0.03	0.62±0.03
11	0.47±0.01	0.47±0.01	0.48±0.02
12	0.53±0.02	0.55±0.02	0.52±0.01
13	0.61±0.02	0.61±0.02	0.65±0.03
14	0.60±0.03	0.69±0.03	0.69±0.03
15	0.50±0.03	0.60±0.02	0.74±0.02
16	0.51±0.03	0.57±0.02	0.63±0.03

Table 3.5: DSC value obtained with three different intensity level $I = 0.3$, $I = 0.5$, $I = 0.8$ in RG segmentation.

Every case presents the best Dice index for the intensity upper bound equal to 0.8; only two cases need a lower intensity level to rightly segment the image: they correspond

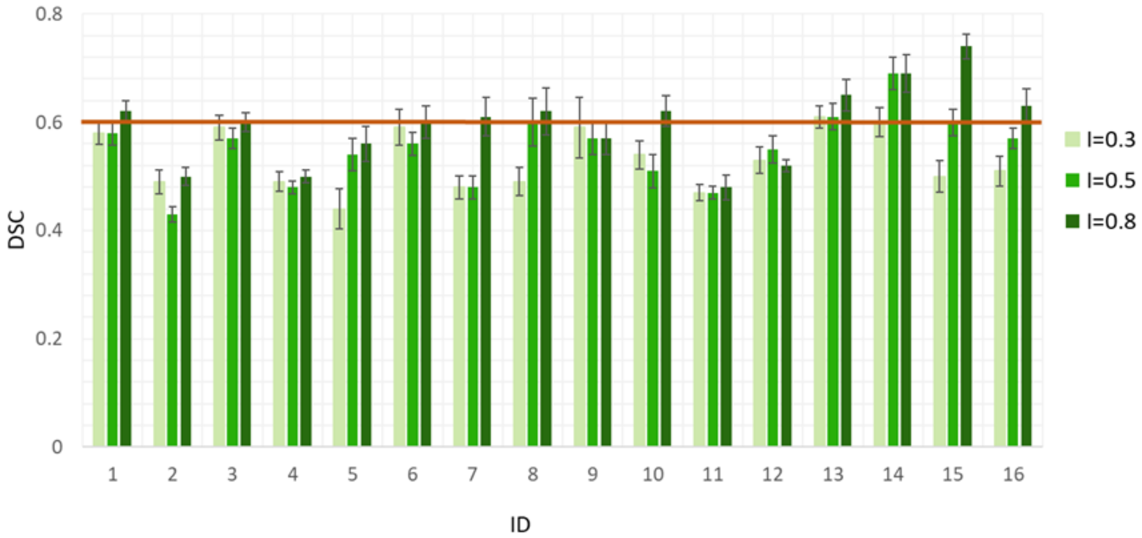


Figure 3.9: DSC value of each case for three different intansity values $I = 0.3$, $I = 0.5$, $I = 0.8$ in RG method.

to case $n°9$ and $n°12$ ($I = 0.3$ and $I = 0.5$ respectively), i.e. the smallest volumes at stake. It is reasonable because the smallest VOIs require less region growth.
Figure 3.10 gives an example of segmentation process on one slice for FCM (red curve) and RG (green curve) compared to gold standard that is manual segmentation (yellow curve) obtained considering the case $n°1$.

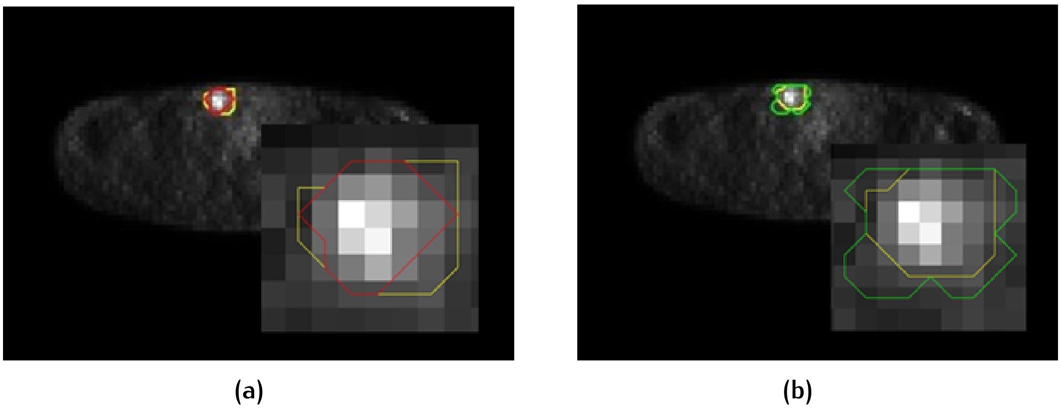


Figure 3.10: Representation of segmentation process of FCM in red (a) and RG method in green (b) compared to gold standard (yellow contour).

The automatic ROI overlaps most of manual ROI with an underestimation of volume in FCM and an overestimation in RG.
A representation of mean DSC and AHD values with bar errors, calculated mediating

on all cases, is visible in Figure 3.13 for both methods.

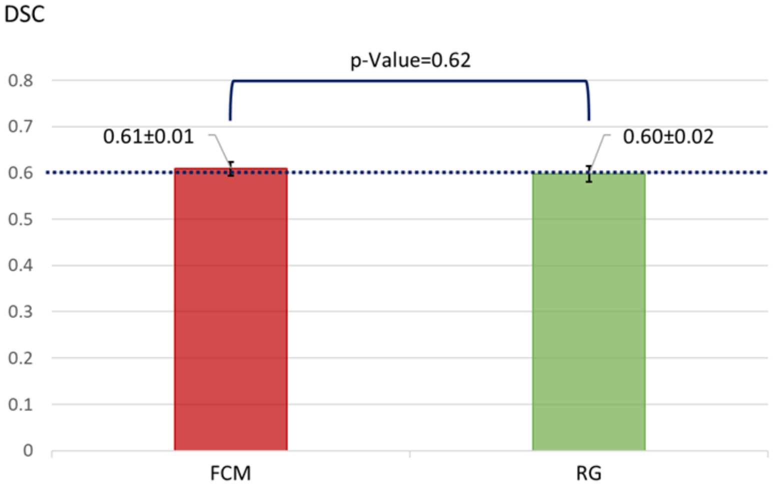


Figure 3.11: DSC average value on all cases.

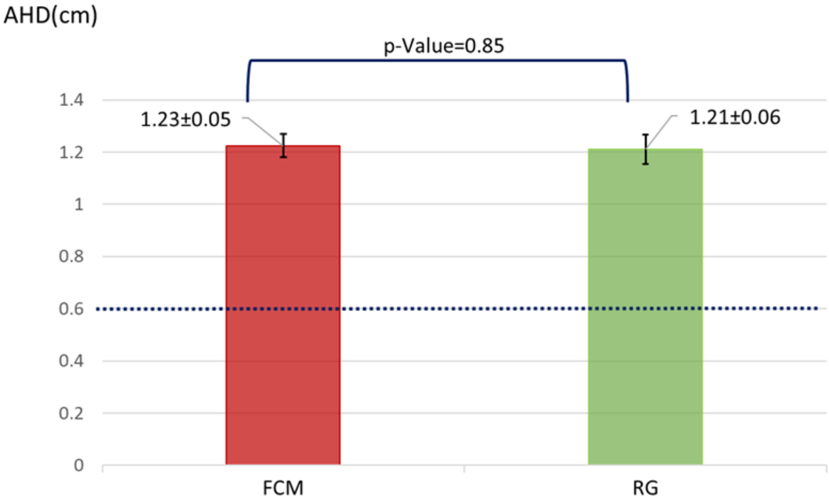


Figure 3.12: AHD average value on all cases.

Figure 3.13: DSC and AHD mean value: comparing two methods by these two indIces using *t*-test, they are not significantly different actually *p*-value is 0.62 and 0.85 respectively considering confidence interval of 95%.

The DSC mean value obtained mediating on all cases, is within the literature threshold of 0.6 [29] for FCM and RG method; AHD mean value is also acceptable considering literature threshold estimated by Yang of 16.3 ± 7.3 *mm* [30] and PET image resolution, equal to about 6 *mm*. Comparing the two methods by DSC and AHD value using t-*test*, they are not significantly different from each other ($p \gg 0.05$) within confidence interval of 95%.

DSC and AHD can be related to each other: generally when DSC value increases, AHD value decreases and vice versa. Figure 3.14 shows this correlation.

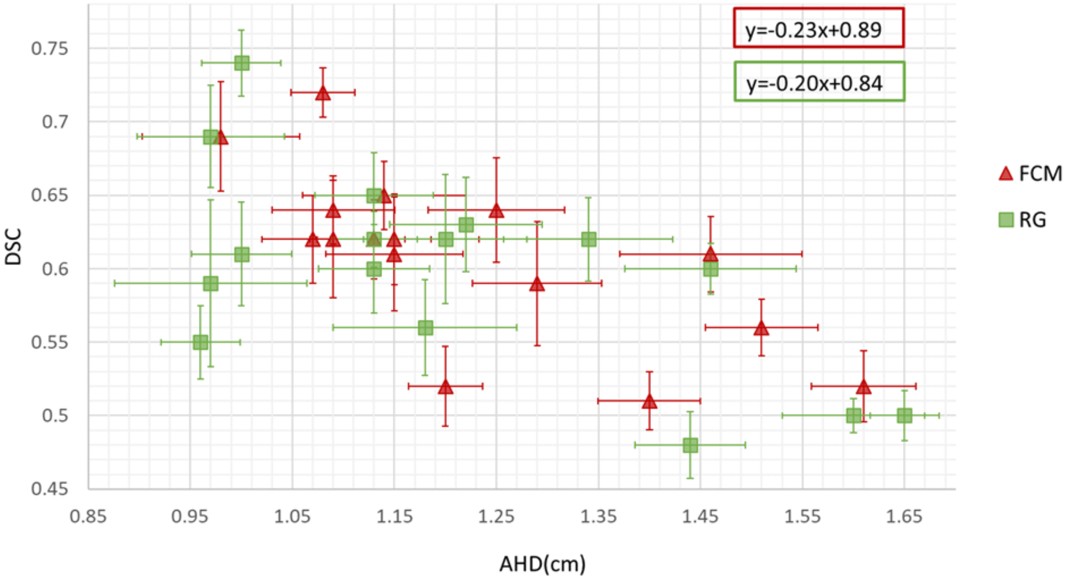


Figure 3.14: Trend of DSC in function of AHD for FCM and RG method.

Considering *linear correlation* coefficient r_0 and the percentage probability $P_N(|r| \geq r_0)$[4], the two quantities result related to each other, as visible in Table E.1 (Appendix E) with $N = 16$. Actually, in FCM, the estimation of r_0 is equal to 0.89 and the percentage probability is close to 0 with a value less than 0.05%: in this case the observed correlation r_0 is *highly significant*. In RG the linear correlation is also *significant* with $r_0 = 0.65$ and the probability near to 1%.

The correlation between contouring time and theoretical volume [5], obtained by manual segmentation, is shown in Figure 3.15 and the corresponding values are visible in Table 3.6. RG method still presents a random distribution of values while FCM partially confirms logarithmic trend with $R^2 = 0.68$. This trend is visible also plotting time

[4] $P_N(|r| \geq r_0)$ is the percentage probability that N measures of two variables not linearly correlated give a correlation coefficient $|r| \geq r_0$.

[5] The error on theoretical volume could not be estimate because the manual segmentation was performed only by one physician so it is a subjective measure. Anyway in literature there are some examples about gold standard error estimating as interobserver variability: Stapleford and al. in their study [31] evaluated an error of about 16.5% on theoretical volume.

ID	$V_{th}cc)$	$n^{\circ}\ slices_{tot}$	$t_{FCM}(s)$	$t_{RG}(s)$
1	219	45	25	129
2	204	20	14	36
3	436	45	31	82
4	412	45	26	93
5	119	24	18	45
6	23	14	6	137
7	75	39	23	58
8	47	20	11	86
9	15	12	6	39
10	286	27	37	45
11	148	23	14	101
12	18	15	10	88
13	118	32	21.2	30
14	51	20	13.4	84
15	145	27	18.0	9
16	141	32	18.3	23

Table 3.6: Value of contouring time related to different cases and associated with theoretical volume (gold standard volume) and number of slices for both methods.

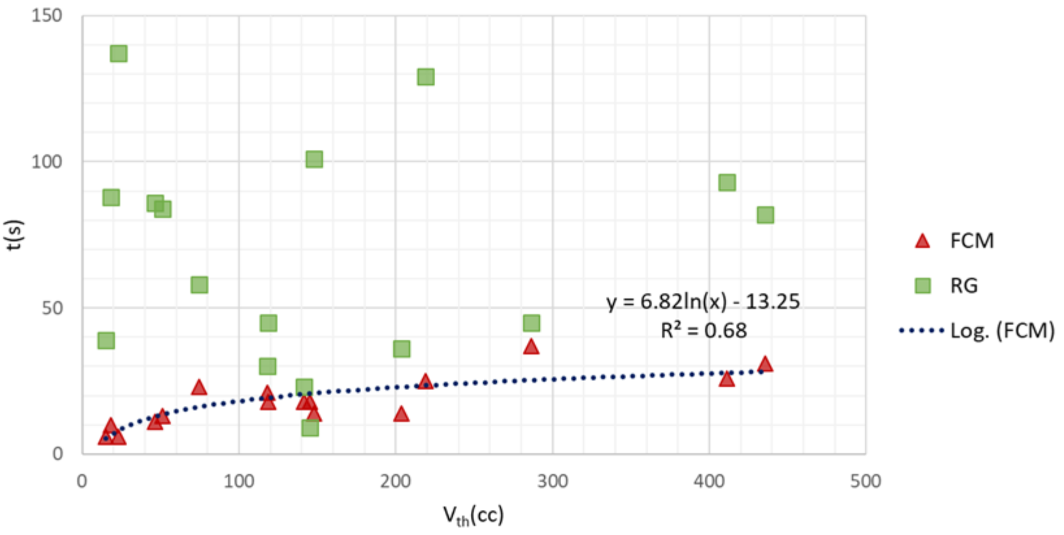


Figure 3.15: Trend of time in function of V_{th} for FCM and RG method.

value in function of slice number visible in Figure 3.16 only for FCM. The value farthest from the curve corresponds to case $n°10$ that presents an image very degraded then hard to segment. In this case, excluding the worst value, the trend is logarithmic with $R^2 = 0.94$. Box plot in Figure 3.17 shows time variability for FCM and RG methods.

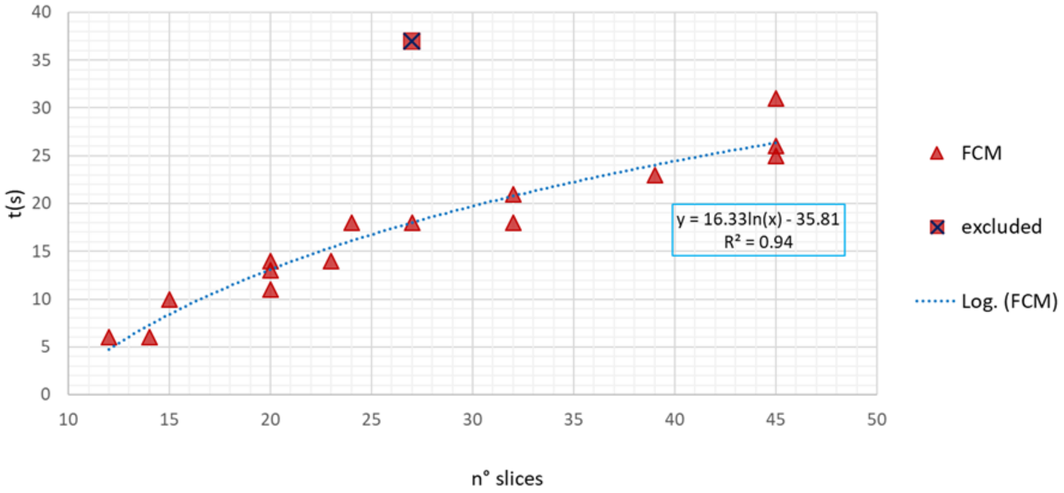


Figure 3.16: Trend of contouring time in function of slice number with all values and excluding the worst time value in FCM method.

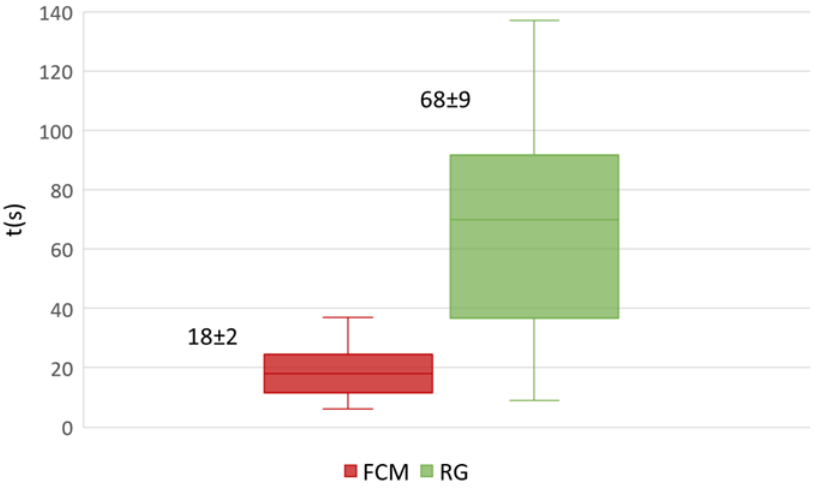


Figure 3.17: Box plot of contouring time (in seconds) for FCM and RG methods.

Contouring time for RG is longer than FCM one and, in some cases, it reaches values over 2 min. The mean time value is 18 ± 2 s for FCM and 68 ± 9 s for RG.

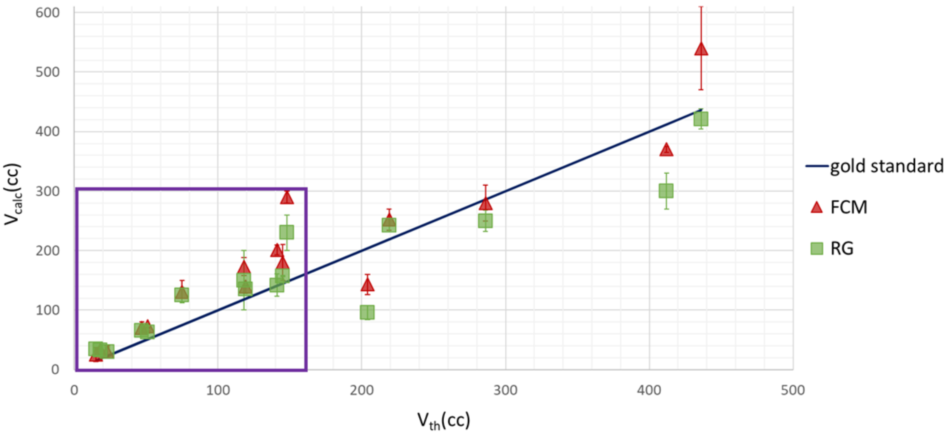
The VOIs obtained after segmentation (V_{calc}) can be related to those obtained with gold standard (V_{th}). The results are shown in Table 3.7 while the trend of calculate volume as function of theoretical one is visible in Figure 3.18.

$V_{th}(cc)$	$V_{calc} \pm \Delta V_{calc}(cc)$	
	FCM	RG
219	252±18	242±11
204	143±17	96±12
436	540±70	421±17
412	370±5	300±30
119	140±1	135±7
23	30±10	30±7
75	131±19	125±13
47	70±10	66±3
15	25±12	34±1
286	280±30	250±18
148	290±10	230±30
18	33±1	33±1
118	173±15	150±50
51	73±3	63±2
145	180±30	157±1
141	201±8	142±19

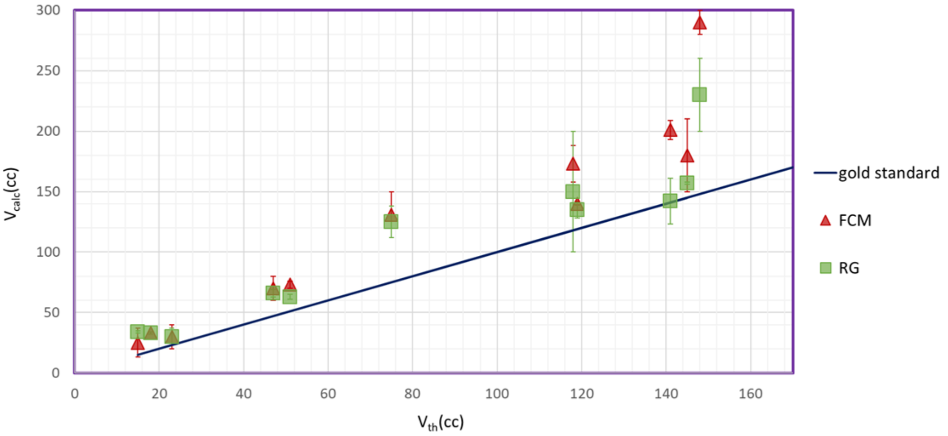
Table 3.7: Theoretical and calculated volume for each case in both methods. The errors on V_{calc} are calculated as maximum error repeating algorithms three times.

For smallest V_{th}, until 170 cc, calculated volume is overestimated for both methods because of PVE. Values larger than 170 cc can presented an overestimation or an underestimation of V_{calc} respect to V_{th}. Anyway the behavior of two methods is the same for all values except the biggest one that performs an overestimation for FCM and an underestimation for RG.

In pre-processing step, three different sigma values (0.5, 1.5 and 2.5) were chosen to determinate the best segmentation parameters based on DSC. The obtained values are shown in Table 3.8a, 3.8b while their representation is visible in Figure 3.19, 3.20. Considering the best sigma value (highlighted in Table 3.8a, 3.8b) for each V_{th}, two volume groups are created in both methods: one related to $\sigma = 1.5$ for volume values until 145 cc and another related to $\sigma = 2.5$ for volume value between 145-450 cc, represented by orange and green arrow respectively. There is an overlap between two groupes because, as visible in Figure 3.19, 3.20, two values present the best DSC for sigma value different from expectations. In fact, DSC related to $V_{th} = 219$ cc is better for $\sigma = 1.5$ (not 2.5, blue dash), while $V_{th} = 119$ cc has a better DSC value for $\sigma = 2.5$ (not 1.5, yellow dash).



(a)



(b)

Figure 3.18: Trend of calculated volume as function of theoretical one in FCM and RG method (a). Magnification of details in theoretical volume range between 0-170 cc(b). Blue line represents gold standard diagonal that is V_{th} vs V_{th}.

This anomaly could be due to the image degradation: an image poorly degraded requires a less smoothing to be well-contrasted and vice-versa.

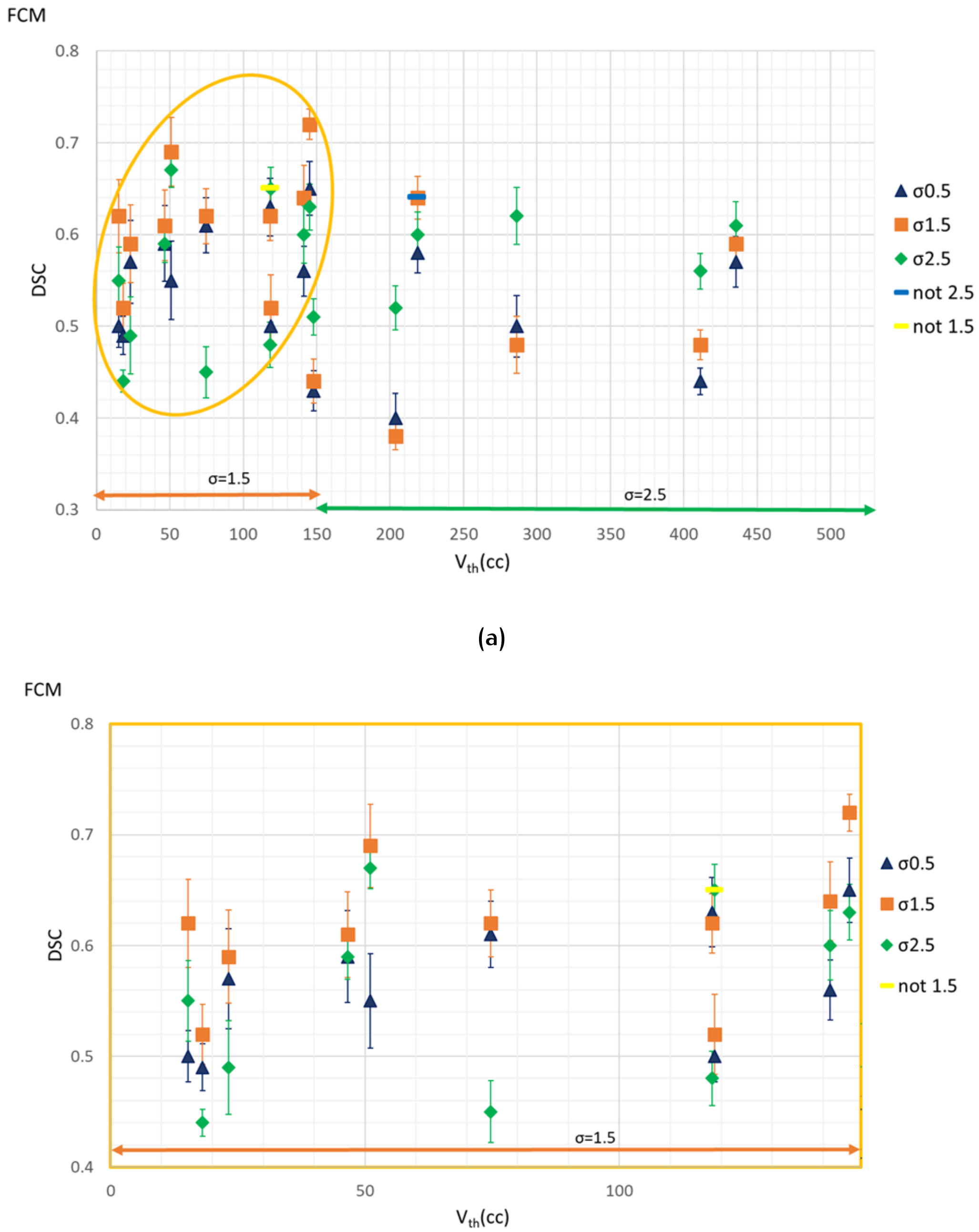
ID	V_{th})	$DSC \pm \Delta DSC$		
		$\sigma_{0.5}$	$\sigma_{1.5}$	$\sigma_{2.5}$
1	219	0.58±0.02	0.64±0.02	0.60±0.02
2	204	0.40±0.03	0.38±0.01	0.52±0.02
3	436	0.57±0.03	0.59±0.02	0.61±0.03
4	412	0.44±0.01	0.48±0.02	0.56±0.02
5	119	0.50±0.02	0.52±0.04	0.65±0.02
6	23	0.57±0.05	0.59±0.04	0.49±0.04
7	75	0.61±0.03	0.62±0.03	0.45±0.03
8	47	0.59±0.04	0.61±0.04	0.59±0.02
9	15	0.50±0.02	0.62±0.04	0.55±0.04
10	286	0.50±0.03	0.48±0.03	0.62±0.03
11	148	0.43±0.02	0.44±0.02	0.51±0.02
12	18	0.49±0.02	0.52±0.03	0.44±0.01
13	118	0.63±0.03	0.62±0.03	0.48±0.02
14	51	0.55±0.04	0.69±0.04	0.67±0.02
15	145	0.65±0.03	0.72±0.02	0.63±0.03
16	141	0.56±0.03	0.61±0.04	0.60±0.03

(a) FCM method.

ID	V_{th})	$DSC \pm \Delta DSC$		
		$\sigma_{0.5}$	$\sigma_{1.5}$	$\sigma_{2.5}$
1	219	0.53±0.02	0.62±0.02	0.61±0.02
2	204	0.43±0.03	0.39±0.02	0.50±0.02
3	436	0.58±0.02	0.56±0.02	0.60±0.02
4	412	0.43±0.02	0.47±0.02	0.50±0.01
5	119	0.47±0.02	0.53±0.03	0.56±0.03
6	23	0.56±0.05	0.60±0.03	0.54±0.02
7	75	0.54±0.02	0.61±0.04	0.57±0.03
8	47	0.58±0.04	0.62±0.04	0.60±0.04
9	15	0.47±0.03	0.59±0.06	0.48±0.01
10	286	0.50±0.03	0.46±0.03	0.62±0.03
11	148	0.45±0.02	0.47±0.02	0.48±0.02
12	18	0.47±0.01	0.55±0.02	0.51±0.01
13	118	0.63±0.04	0.65±0.03	0.48±0.02
14	51	0.60±0.03	0.69±0.03	0.67±0.03
15	145	0.68±0.03	0.74±0.02	0.73±0.03
16	141	0.57±0.03	0.63±0.03	0.58±0.03

(b) RG method.

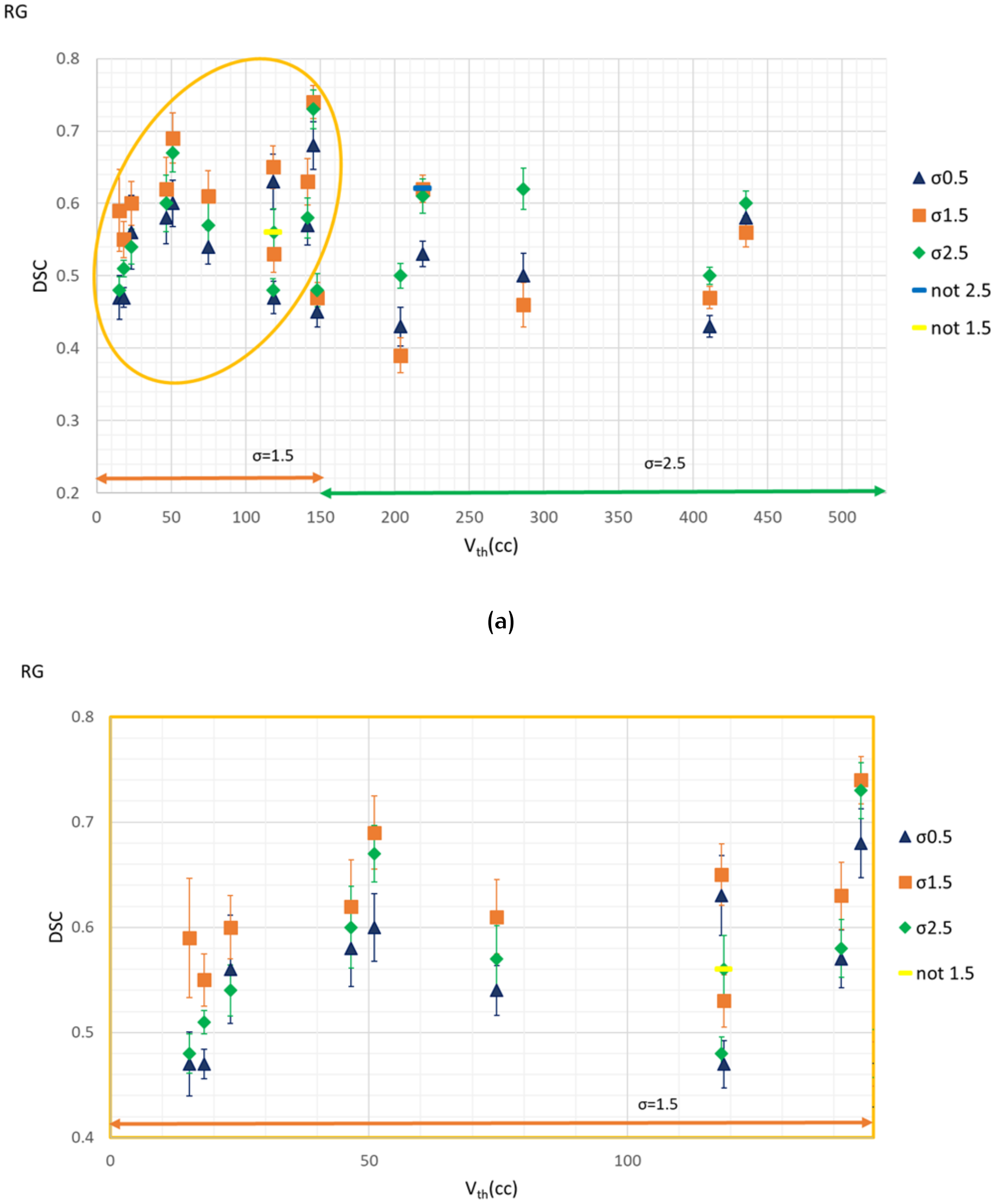
Table 3.8: DSC values for three different sigma values in FCM method (a) and RG method (b). Highlighted values are related to best sigma value for each case.



(a)

(b)

Figure 3.19: Representation of DSC values for three different sigma (0.5, 1.5 and 2.5) in FCM method (a). Magnification of details in theoretical volume range between 0-145 *cc* (slightly larger considering error bars)(b). Error bars are represented in the color of points: blue for $\sigma = 0.5$, orange for $\sigma = 1.5$ and green for $\sigma = 2.5$.



(a)

(b)

Figure 3.20: Representation of DSC values for three different sigma (0.5, 1.5 and 2.5) in RG method (a). Magnification of details in theoretical volume range between 0-145 *cc* (slightly larger considering error bars)(b). Error bars are represented in the color of points: blue for $\sigma = 0.5$, orange for $\sigma = 1.5$ and green for $\sigma = 2.5$.

Chapter 4

Discussion of results and future developments

The accuracy evaluation of the two algorithms shows acceptable results. Sperimental test on phantom, performed evaluating the percentage change between calculated and theoretical volume, has produced all volume change values that are null within the error so close to expectation value equal to 0. The largest $V_{ch}(\%)$ was obtained for smallest sphere because of the partial volume effect that overestimates the volume. Generally the V_{calc} is underestimated and it is closest to V_{th} in the case of bigger regions that, thanks to dimension, can be segmented easier.

Two methods work in the same way and actually they do not result significantly different, comparing them by t-test. RG process allows also to choice the best intensity parameter to obtain a better segmentation: only for smallest sphere a lower intensity was preferred to the standar $I = 0.8$ because it defines a region growth firstly stopped.

Sigma parameter of gaussian function used to remove noise is related to sphere size indeed the low value ($\sigma = 0.5$) is better for smallest sphere, the intermediate value ($\sigma = 1.5$) for medium-size spheres and the high value ($\sigma = 2.5$) for the biggest ones.

The time required to segment each sphere is a few seconds. It increases logarithmically with the sphere size in FCM algorithm, while it has a random distribution in RG because it is more sensitive to image contrast.

The tests performed on H&N tumors PET images have partially confirmed the results described above. The accuracy was estimated using DSC and AHD: the values found are, in both methods, within literature threshold [29] [30] and over of 75% of slices are correctly segmented. Computing p-value with t-test, the two methods are not significantly different from each other. Best DSC and AHD values are found with an intensity of $I = 0.8$ for each case except the two smallest ones that prefer a lower intensity to first stop the growth of region. Sigma parameter shows the same correlation with region size: smaller regions produce best values with lower sigma. There is no value that requires $\sigma = 0.5$ because there are not regions of size as the smallest phantom sphere.

Calculating the correlation coefficient, DSC and AHD present an *highly significant* correlation in FCM and a *significant* correlation in RG. Generally, as DSC increases moving toward 1, AHD decreases moving to lower values. Contouring time and theoretical volume show a logarithmic trend in FCM method. This trend is also visible considering countouring time as function of number of slices: it reaches a best value excluding the case related to image more degraded. In RG the distribution is still random. Both algorithms present a time much lower than that of manual segmentation that achieves better value in FCM.

FCM and RG automatic algorithms try to overcome the problems related to manual segmentation, in particular the time necessary to segment that both methods manage to improve. In fact, this time passes from hours, necessary to radiotherapist physician to manual segment the tumors, to seconds (at most minutes) required by these two automatic algorithms.

Moreover, these methods allow to segment well *H&N* tumors in most of the slice images: only a maximum percentage of 25% in the more degraded images should be segmented manually and this percentage could be improved also improving pre-processing operation. Anyway a check and an editing by a radiotherapist physician is always required.

The greatest difference in the two methods consists in the segmentation of more distinct regions: FCM is able to distinguish them, while RG could do it only selecting different initial seeds as starting points of different growing regions.

Future developments could be focused on the optimization of processes using a greater number of patients belonging to more homogeneous sets. Increasing the statistics, maybe, better results, regarding DSC and AHD value, could be obtained. In this way, also an estimation of intra-observer and inter-observer variability could be provided retesting manual and automatic segmentation by the same expert more than one time and by different experts. Important could be also the application on different tumors such as, lymphomas, to test in particular the efficiency of FCM to segment displaced regions.

Conclusions

The aim of this work is the development of automatic segmentation algorithms in PET images particularly to overcome the probelms related to manual segmentation and optimize time, decreasing the workload of radiotherapist physician. Fuzzy C-means and Region Growing algorithms were implemented in a MATLAB® GUI to choose iteratively the different steps required for a good segmentation. They were tested using sperimental PET phantom and 16 clinical PET image of H&N tumors.
Calculing the percentage change between theoretical and calculated volume of spheres in phantom, the results show a consistency with gold standard; also the value of DSC and AHD, obtained mediating on all studies, describe a volume overlap and a volume distance within literature threshold. The time required to segment all slices of each case reaches a maximum value of about 40 s in FCM and over 2 min in RG: in both processes this time is much less than time of manual segmentation that can achieve more than two hours. The slices correctly segmented are more than 75% in both methods; the others can be edited by radiotherapist physician always maintaing a time optimization.
Comparing two methods with t-test, p-value shows a not significantly difference between them: they work in the same way regarding DSC and also AHD. They also request the same pre-processing parameters to optimize the segmentation.
Definitely both methods allow to overcome the problem of manual segmentation related to contouring time. Future testing could ascertain or refute the optimization related to intra-observer and inter-observer variability repeating manual and automatic segmentation more than one time and by more experts.
Moreover, the use of FCM could be extended also to segmentation of more displaced regions that can be found in some clinical diseases as, for example, lymphomas.

Appendix A

Acronyms Index

AHD=Average Hausdorff Distance

CT=Computed Tomography

CTV=Clinical Target Volume

DSC=Dice Similarity Coefficient

FBP=Filtered Back Projection

FCM=Fuzzy C-means

FWHM=Full Width at Half Maximum

GTV=Gross Tumour Volume

GUI=Graphical User Interface

HD=Hausdorff Distance

H&N =Head and Neck

I=Intensity upper bound

MRI=Magnetic Resonance Imaging

RG=Region Growing

ROI=Region Of Interest

σ= *gaussian function parameter*

SUV= Standardized Uptake Value

PET=Positron Emission Tomography

PSF=Point Spread Function

PTV=Planning Target Volume

PVE=Partial Volume Effect

V_{calc}= *calculated Volume*

V_{ch}= *Volume change*

V_{th}= *theoretical Volume*

VOI=Volume Of Interest

Appendix B

MATLAB codes

The following scripts are reported as functions Callback of FeRS_GUI (section 2.4), then they include all actions associated to specific pressed button. The description of principal steps is commented with the symbol in green.

B.1 FCM code

```
function FCM_Callback(hObject, eventdata, handles)

K=[];
V=handles.V;
V0=handles.V0;
ax=handles.ax;
ay=handles.ay;
sx=handles.sx;
sy=handles.sy;
cx=handles.cx;
cy=handles.cy;
ROInew=handles.ROInew;

yf=handles.yf;
d3=handles.d3;
d1=handles.d1;
d2=handles.d2;
block=handles.block;

if yf==1
     %recall the dimension of interest object previously selected
    rp1=handles.rp1;
    l1r=round(rp1(1)):round(rp1(1)+rp1(3));
    l2r=round(rp1(2)):round(rp1(2)+rp1(4));
    handles.l1r=l1r;
    handles.l2r=l2r;
    uiwait(msgbox('Select the cluster'));
    waitforbuttonpress;
    [l1,l2]=ginput(1);
    handles.l1=l1;
    handles.l2=l2;
    data=get(handles.slider1,'UserData');
    currentval=data.val;
    z=round(currentval*(d3-1)+1);
    Imax=abs(V(round(l2),round(l1),z));
    handles.Imax=Imax;
    if block==0          %if mask has not already imported
        uiwait(msgbox('Select initial and final slice'));
        waitforbuttonpress;
        [x,y]=ginput(1);
        plot(x,y,'c*');
        data=get(handles.slider1,'UserData');
        currentval=data.val;
        k1=round(currentval*(d3-1)+1);
        handles.k1=k1;

        waitforbuttonpress;
        [x,y]=ginput(1);
        plot(x,y,'c*');
        data=get(handles.slider1,'UserData');
        currentval=data.val;
        k2=round(currentval*(d3-1)+1);
        handles.k2=k2;

        guidata(hObject,'Value');
        k1=handles.k1;
        k2=handles.k2;
    else
        k1=handles.k1;   %number of initial and final slice of mask
        k2=handles.k2;
    end
else
    k1=handles.k1;
    k2=handles.k2;
    l1=handles.l1;
    l2=handles.l2;
    l1r=handles.l1r;
    l2r=handles.l2r;
end

diametro=[];
A=[];
A_mm=[];
lev=[];
jb=0;
p_mm=handles.p_mm;
In=zeros((k2-k1+1));
k=k1-1;

```

```
tic
while k<k2
    k=k+1;
    handles.K=K;
    rp1=handles.rp1;
    img=V(:,:,k);
    Imax=handles.Imax;
    [n,m]=size(img);

   %approximative volume to set dimension
    volume=(4/3*3.14*rp1(3)*rp1(4)*(k2-k1)*p_mm(1))/1000;
    diam=rp1(3)*p_mm(1);

    I=abs(img(round(l2),round(l1))); %cluster intensity

    axes(handles.axial);
    imshow(V0(:,:,k),[]);

    %cluster number selected in function of region dimension
    if (diam>=16 && diam<=28) %MEDIE
        se2=strel('disk',1);
        if I<Imax
            cluster=3;
        else
            cluster=2;
        end
    elseif diam>=2 && diam<=16  %PICCOLE
        se2=strel('disk',0);

        if I<Imax
            cluster=3;
        else
            cluster=2;
        end
    elseif diam>=28  %GRANDI
        if diam>=28 && diam<35
            se2=strel('disk',2);
        else
            se2=strel('disk',3);
        end

        if I<Imax
            cluster=3;
        else
            cluster=2;
        end

    data=img(:);

    %Matlab fcm function returns an indexed matrix basing on cluster number
    [center,member]=fcm(data,cluster);
    [~,~]=sort(center);
    member=member';
    [~,label]=max(member,[],2);
    segmROI=reshape(label,size(img));
    px=[];
    py=[];
    hold on
    in=segmROI(round(l2),round(l1));
    ROImask=segmROI==in;

    A1=n*m;
    s2=regionprops(ROImask(ROImask==1),'ConvexArea');
    A2=s2.ConvexArea;
     if A2>0.8*A1 %select complementary image to better visualization
        ROImask=imcomplement(ROImask);
    end
    [L,nobj]=bwlabel(ROImask,4);
    c=L;

    %allow to give the more likely mask also when the image is very degraded
    if nobj>3
        in1=L(round(l2),round(l1));
        if in1==0
           [~,idx]=bwdist(L~=0);
           d=sub2ind(size(L),round(l2),round(l1));
           ddx=idx(d);
           [xc, yc] = ind2sub(size(L), ddx);
           in1=L(xc,yc);
        end
       c=L==in1;
     end
    hold on
    c=imdilate(c,se2);

     %contours realization
     B0=bwboundaries(c);
     for b=1:length(B0)
         boundary=B0{b};
         P=plot(boundary(:,2), boundary(:,1),'r','LineWidth', 0.8);
         px=[px P.XData];
         py=[py P.YData];
     end
     %area calculation
     stats=regionprops(c,'Area');
     area=stats.Area;
     A=[A area];
     A_mm=[A_mm area*p_mm(1)*p_mm(2)];

     ROInew(:,:,k)=c;
     handles.ROInew=ROInew;
     ax{k}=px;
     ay{k}=py;
     handles.ax=ax;
     handles.ay=ay;
     yf=yf+1;
     handles.yf=yf;
     K=[K,k];
     handles.K=K;

     guidata(hObject,handles);
end
toc
%disp(mean(A));
%disp(mean(A_mm));

%mask and contours in every projection
for k=1:(k1-1)
    ROInew(:,:,k)=zeros(size(img));
end
for k=(k2+1):d3
    ROInew(:,:,k)=zeros(size(img));
end

Rs=permute(ROInew,[3 1 2]);
[~,~,r3]=size(Rs);
for ks=1:r3
         px_s=[];
         py_s=[];
         segm_Rs=imrotate(Rs(:,:,ks)',90);
         BOs=bwboundaries(segm_Rs);
     for bs=1:length(BOs)
         boundarys=BOs{bs};
         P_s=[boundarys(:,2), boundarys(:,1)];
         px_s=[px_s P_s(:,1)'];
         py_s=[py_s P_s(:,2)'];
     end
     sx{ks}=px_s;
     sy{ks}=py_s;
     handles.sx=sx;
     handles.sy=sy;
     guidata(hObject,handles);
end

Rc=permute(ROInew,[2 3 1]);
[~,~,c3]=size(Rc);
for kc=1:c3
         px_c=[];
         py_c=[];
         segm_Rc=imrotate(Rc(:,:,kc),90);
         BOc=bwboundaries(segm_Rc);
     for bc=1:length(BOc)
         boundaryc=BOc{bc};
         P_c=[boundaryc(:,2), boundaryc(:,1)];
         px_c=[px_s P_c(:,1)'];
         py_c=[py_s P_c(:,2)'];
     end
     cx{kc}=px_c;
     cy{kc}=py_c;
    handles.cx=cx;
    handles.cy=cy;
    guidata(hObject,handles);
end

ROInew=squeeze(ROInew);
ROInew=im2uint8(ROInew);

%save and list mask

p2=handles.p2;
s1='_FCM_';
s2=p2;
image=ROInew;
BOname=[s1 s2];
folder=handles.folder;

dicomwrite(reshape(image,[d1,d2,1,d3]),BOname);

%store the selected folder in handles.folder;
files = dir(fullfile(folder,'_*')); %get all dcm files
handles.files=files;
for i=1:length(files)
    piclist{i} = files(i).name; %lists each of them in a cell
end

set(handles.listbox1,'String',piclist)
guidata(hObject,handles);
```

B.2 RG code

```
function RG_Callback(hObject, eventdata, handles)

K=[];
V=handles.V;
V0=handles.V0;
ax=handles.ax;
ay=handles.ay;
ROInew=handles.ROInew;

%insert intensity value
intensity = str2num(get(handles.rg2, 'String'));
handles.intensity=intensity;
if isempty(intensity)
    uiwait(msgbox('The intensity number is empty'));
    return;
end

y=handles.y;
```

```
d3=handles.d3;
d1=handles.d1;
d2=handles.d2;
block=handles.block;

if y==1
    uiwait(msgbox(’Select the seed’));
    waitforbuttonpress;
    [xa,ya]=ginput(1);
    seedI=round(xa);
    seedJ=round(ya);
    handles.seedI=seedI;
    handles.seedJ=seedJ;
    data=get(handles.slider1,’UserData’);
    currentval=data.val;
    z=round(currentval*(d3-1)+1);
    Imax=abs(V(round(seedJ),round(seedI),z));
    handles.Imax=Imax;
    if block==0
        uiwait(msgbox(’Select initial and final slice’));
        waitforbuttonpress;
        [x,y]=ginput(1);
        plot(x,y,’c*’);
        data=get(handles.slider1,’UserData’);
        currentval=data.val;
        k1=round(currentval*(d3-1)+1);
        handles.k1=k1;

        waitforbuttonpress;
        [x,y]=ginput(1);
        plot(x,y,’c*’);
        data=get(handles.slider1,’UserData’);
        currentval=data.val;
        k2=round(currentval*(d3-1)+1);
        handles.k2=k2;

        guidata(hObject,’Value’);
        k1=handles.k1;
        k2=handles.k2;
    else
        k1=handles.k1;
        k2=handles.k2;
    end
else
    k1=handles.k1;
    k2=handles.k2;
    seedI=handles.seedI;
    seedJ=handles.seedJ;
end

p_mm=handles.p_mm;
diametro=[];
A=[];
```

```
A_mm=[];
k=k1-1;

tic
while k<k2
    k=k+1;
    handles.K=K;
    img=V(:,:,k);
    Imax=handles.Imax;
    rp1=handles.rp1;
    diam=rp1(3)*p_mm(1);
    I=abs(img(round(seedJ),round(seedI)));
    axes(handles.axial);
    imshow(V0(:,:,k),[]);
    hold on
    px=[];
    py=[];
    %dimension discrimination for following dilation operation
    if  diam>=16 %MEDIE/GRANDI(diam>=16 && diam<=29 )
        se2=strel('disk',1);
        if I<0.3*Imax
            continue
        end
    elseif diam>=2 && diam<=16  %PICCOLE
        se2=strel('disk',0);
        if I<0.1*Imax
            continue
        end

     %region_growing function implemented in a separate matlab code
    [~,ROI]=region_growing(mat2gray(V),seedI,seedJ,k,intensity);

    segmROI=imbinarize(ROI);
    L=bwlabel(segmROI);
    in=L(seedJ,seedI);
    if in==0
        continue
    end

    c=L==in;
    c=imdilate(c,se2);
    %contours delineation
    BO=bwboundaries(c);
    hold on
    for b=1:length(BO)
        boundary=BO{b};
        P=plot(boundary(:,2), boundary(:,1),'r', 'LineWidth', 1);
        px=[px P.XData];
        py=[py P.YData];
    end

    %area calculation
    stats=regionprops(c,'Area');
    area=stats.Area;
    A=[A area];
    A_mm=[A_mm area*p_mm(1)*p_mm(2)];

    ROInew(:,:,k)=c;
    handles.ROInew=ROInew;

    ax{k}=px;
    ay{k}=py;
    handles.ax=ax;
    handles.ay=ay;

    y=y+1;
    handles.y=y;
    K=[K,k];
    handles.K=K;
    guidata(hObject,handles);
end
toc

%mask and contours in every projection
for k=1:(k1-1)
    ROInew(:,:,k)=zeros(size(V(:,:,k)));
end
for k=(k2+1):d3
    ROInew(:,:,k)=zeros(size(V(:,:,k)));
end


Rs=permute(ROInew,[3 1 2]);
[~,~,r3]=size(Rs);
for ks=1:r3
         px_s=[];
         py_s=[];
         segm_Rs=imrotate(Rs(:,:,ks)',90);
         BOs=bwboundaries(segm_Rs);
     for bs=1:length(BOs)
         boundarys=BOs{bs};
         P_s=[boundarys(:,2), boundarys(:,1)];
         px_s=[px_s P_s(:,1)'];
         py_s=[py_s P_s(:,2)'];
     end
     sx{ks}=px_s;
     sy{ks}=py_s;
     handles.sx=sx;
     handles.sy=sy;
     guidata(hObject,handles);
end

Rc=permute(ROInew,[2 3 1]);
[~,~,c3]=size(Rc);
for kc=1:c3
         px_c=[];
         py_c=[];
         segm_Rc=imrotate(Rc(:,:,kc),90);
         BOc=bwboundaries(segm_Rc);
     for bc=1:length(BOc)
         boundaryc=BOc{bc};
         P_c=[boundaryc(:,2), boundaryc(:,1)];
         px_c=[px_s P_c(:,1)'];
         py_c=[py_s P_c(:,2)'];
     end
     cx{kc}=px_c;
     cy{kc}=py_c;
     handles.cx=cx;
     handles.cy=cy;
     guidata(hObject,handles);
end
ROInew=squeeze(ROInew);
ROInew=im2uint8(ROInew);

%save and list mask
p2=handles.p2;
s1='_RG_';
s2=num2str(intensity);
s3=p2;
s4='_';
BOname=[s1 s2 s4 s3];
image=ROInew;
folder=handles.folder;
dicomwrite(reshape(image,[d1,d2,1,d3]),BOname);

%store the selected folder in handles.folder;
files = dir(fullfile(folder,'_*')); %get all dcm files
handles.files=files;
for i=1:length(files)
    piclist{i} = files(i).name; %lists each of them in a cell
end


set(handles.listbox1,'String',piclist)
guidata(hObject,handles);
```

Region_ Growing function recalled in line 101 is implemented in separated script named *region_ growing.m*. The code is shown below:

```
 function [slice,roi]=region_growing(V,seed_I,seed_J,seed_K,intensity)

slice=V(:,:,seed_K);
[r,~]=size(slice);
obj=[seed_J,seed_I];
int_obj=slice(seed_J,seed_I);
h=0;
for m=1:r-seed_J
          %define neighboring pixels
```

```
        neigh=[seed_I-m seed_J-m;seed_I-m seed_J+m;seed_I+m seed_J+m;seed_I+m seed_J-m];
        l1=[neigh(1,2):neigh(2,2)]';
        a1(1:2*m+1,1)=neigh(1,1);
        surr1=[a1 l1];
        a2=[neigh(2,1):neigh(3,1)]';
        l2(1,1:2*m+1)=neigh(2,2);
        surr2=[a2 l2'];
        l3=[neigh(3,2):-1:neigh(4,2)]';
        a3(1:2*m+1,1)=neigh(3,1);
        surr3=[a3 l3];
        a4=[neigh(4,1):-1:neigh(1,1)]';
        l4(1,1:2*m+1)=neigh(4,2);
        surr4=[a4 l4'];
        surr_tot=abs([surr1;surr2;surr3;surr4]);
        surr=unique(surr_tot,'rows');
        [s1,~]=find(surr(:,1)==0);
        [s2,~]=find(surr(:,2)==0);
        surr(s2,:)=[];
        surr(s1,:)=[];
        [s3,~]=find(surr(:,1)>128);
        [s4,~]=find(surr(:,2)>128);
        surr(s3,:)=[];
        surr(s4,:)=[];
        for n=1:length(surr)
               %similarity intensity criterion to satisfy
                if abs(slice(round(surr(n,2)),round(surr(n,1)))-int_obj)<=intensity
                    region=[surr(n,2) surr(n,1)];
                    int_region=slice(round(surr(n,2)),round(surr(n,1)));
                    obj=[obj;region];
                    int_obj=[int_obj;int_region];
                    roi=zeros(size(slice));
                    %roi creation
                    for i=1:length(int_obj)
                        roi(round(obj(i,1)),round(obj(i,2)))=int_obj(i);
                    end
                else
                    h=3;
                end
        end
                if h==3
                        break
                end
    end
end
```

B.3 Gaussian filter code

```
function Gaussian_Callback(hObject, eventdata, handles)

V=handles.V;
d3=handles.d3;

%insert sigma value
sigma = str2num(get(handles.SigmaValue, 'String'));
handles.sigma=sigma;
if isempty(sigma)
    uiwait(msgbox('The sigma number is empty'));
    return;
end

%define dimension
dimension=2*ceil(2*sigma)+1;

%create 2D gaussian kernel
h=fspecial('gaussian',dimension,sigma);
%convolution between image and kernel
V=convn(V,h,'same');

%updating of volume and displayed image
handles.V=V;
[d1,d2,d3]=size(V);
handles.V=V;
Vs=permute(V,[3 1 2]);
Vc=permute(V,[2 3 1]);
handles.d3=d3;
handles.Vs=Vs;
handles.Vc=Vc;
ROInew=zeros(d1,d2);
ROInew_s=zeros(d3,d2);

for i=1:d3
    A{i}=(V(:,:,i));
    ax{i}=[];
    ay{i}=[];
    ROInew(:,:,i)=zeros(d1,d2);
    similarity{i}=[];
end
imshow(A{i},[],'Parent',handles.axial);
colormap gray;
handles.A=A;
handles.d3=d3;
handles.ax=ax;
handles.ay=ay;
handles.ROInew=ROInew;
handles.similarity=similarity;

```

```
for j=1:d1
  S{j}=imrotate(Vs(:,:,j)',90);
  ROInew_s(:,:,j)=zeros(d3,d2);
  sx{j}=[];
  sy{j}=[];
end
imshow(S{j},[],'Parent',handles.sagittal);
colormap gray;
handles.S=S;
handles.d1=d1;
handles.sx=sx;
handles.sy=sy;

for k=1:d2
    C{k}=imrotate(Vc(:,:,k),90);
    cx{k}=[];
    cy{k}=[];
end
imshow(C{k},[],'Parent',handles.coronal);
colormap gray;
handles.C=C;
handles.d2=d2;
handles.cx=cx;
handles.cy=cy;

yf=1;
y=1;
handles.yf=yf;
handles.y=y;
set(handles.slider1,'Visible','on');
set(handles.Background,'Enable','on')
set(handles.Background,'Visible','on')

guidata(hObject,handles);
```

B.4 Opening code

```
function Background_Callback(hObject, eventdata, handles)

%select object of interest
uiwait(msgbox('Select the principal object'));
waitforbuttonpress;
rect1=imellipse;
rp1=getPosition(rect1);
handles.rp1=rp1;
d3=handles.d3;
V=handles.V;
V0=handles.V0;

```

```
%create structuring element of size as object of interest
se=strel('disk',round(rp1(4)/2));
%apply opening operation to obtain background
background=imopen(V,se);
%subtract backgroung from image
V2=imsubtract(V,background);
V=V2;
handles.V=V;

%updating volume and displayed image
[d1,d2,d3]=size(V);
handles.V=V;
Vs=permute(V,[3 1 2]);
Vc=permute(V, [2 3 1]);
handles.d3=d3;
handles.Vs=Vs;
handles.Vc=Vc;
ROInew=zeros(d1,d2);
ROInew_s=zeros(d3,d2);

for i=1:d3
    A{i}=(V(:,:,i));
    ax{i}=[];
    ay{i}=[];
    ROInew(:,:,i)=zeros(d1,d2);
    similarity{i}=[];
end
imshow(A{i},[],'Parent',handles.axial);
colormap gray;
handles.A=A;
handles.d3=d3;
handles.ax=ax;
handles.ay=ay;
handles.ROInew=ROInew;
handles.similarity=similarity;

for j=1:d1
  S{j}=imrotate(Vs(:,:,j)',90);
  ROInew_s(:,:,j)=zeros(d3,d2);
  sx{j}=[];
  sy{j}=[];
end
imshow(S{j},[],'Parent',handles.sagittal);
colormap gray;
handles.S=S;
handles.d1=d1;
handles.sx=sx;
handles.sy=sy;

for k=1:d2
    C{k}=imrotate(Vc(:,:,k),90);
    cx{k}=[];
    cy{k}=[];
```

```
end
imshow(C{k},[],'Parent',handles.coronal);
colormap gray;
handles.C=C;
handles.d2=d2;
handles.cx=cx;
handles.cy=cy;

yf=1;
y=1;
handles.yf=yf;
handles.y=y;
set(handles.slider1,'Visible','on');

guidata(hObject,handles);
```

B.5 DSC and HD code

```
function Dice_Callback(hObject, eventdata, handles)

p_mm=handles.p_mm;
d3=handles.d3;
k1=handles.k1;
k2=handles.k2;
V0=handles.V0;
V=handles.V;
folder=handles.folder;
mask=handles.mask;
similarity=handles.similarity;

list= get(handles.listbox1,'string'); %get the picture list
selected = get(handles.listbox1,'value'); % get which one is selected.
I= dicomread(fullfile(folder,cell2mat(list(selected)))); %open the picture

%recall imported mask as mask1 and obtained mask as mask2
mask1=double(mask);
I=squeeze(I);
mask2=rescale(I,0,1);

S=[];
K=[];
%calculate DSC with dice matlab function slice by slice
for k=1:d3
    similarity=dice(mask1(:,:,k),mask2(:,:,k));
    S=[S similarity];
    if similarity<0.3
        K=[K k];
    end
end
```

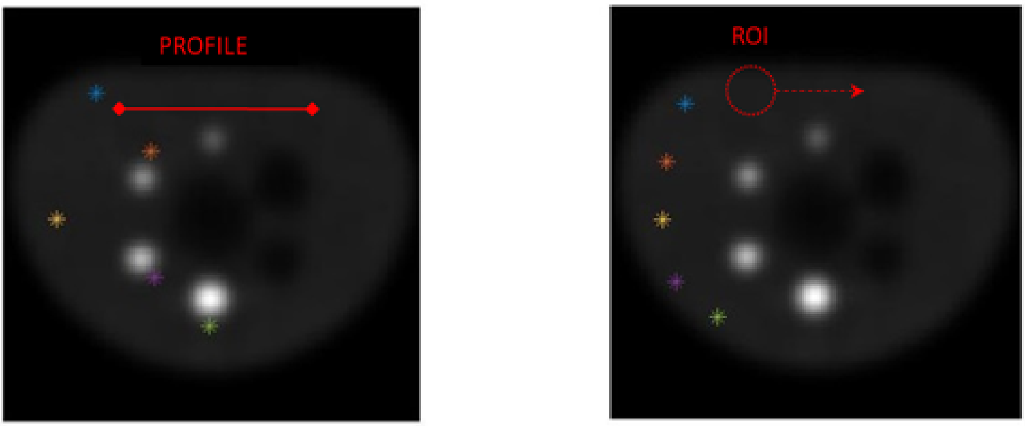
```
%exclude the DSC ideal value less than 0.3 that correspond to real value less than 0.01
mean(S(S>0.3));
handles.K=K;
set(handles.Hausdorff,'Enable','on')
```

```
function Hausdorff_Callback(hObject, eventdata, handles)


d1=handles.d1;
d2=handles.d2;
d3=handles.d3;
K=handles.K;
p_mm=handles.p_mm;
folder=handles.folder;
mask=handles.mask;

list= get(handles.listbox1,'string'); %get the picture list
selected = get(handles.listbox1,'value'); % get which one is selected.
I= dicomread(fullfile(folder,cell2mat(list(selected)))); %open the picture

%recall imported mask as mask1 and obtained mask as mask2
mask1=double(mask);
I=squeeze(I);
mask2=rescale(I,0,1);

HD=[];
%calculate HD using hausdorff function introduced by Zachary Danziger
for k=1:d3
    Hd=hausdorff(mask1(:,:,k),mask2(:,:,k));
    HD=[HD Hd];
end
%exclude the slices defined before
HD(K)=0;
%turn the distance in mm
HD=HD*p_mm(1);
disp(HD);
disp(mean(HD(HD>0)));

guidata(hObject,handles);
```

Appendix C

Non uniform background

The non uniformity of PET images background was verified on PET phantom implementing a Matlab code. It allows to draw a profile on image to identify the more intensity pixel value along it or a ROI, on each side of image, to identify the brigther one. Graphical results, visible in Figure C.1, confirm that the background is not uniformly illuminated and the most brightness is associated to hot spheres side. Figure C.1a shows results obtained by profile while Figure C.1b those obtained with ROI shift: each star point indicates pixel with most intensity value considering some on right side and some on left one.



(a) Profile results

(b) ROI results

Figure C.1: Graphical results obtained checking non uniformity of background. Figure (a) shows results obtained drawing profile, while Figure (b) show the result obtained drawing ROI on one side and moving it on the other side.

In Figure C.2 is visible the intensity profile associated with image in FigureC.1a.

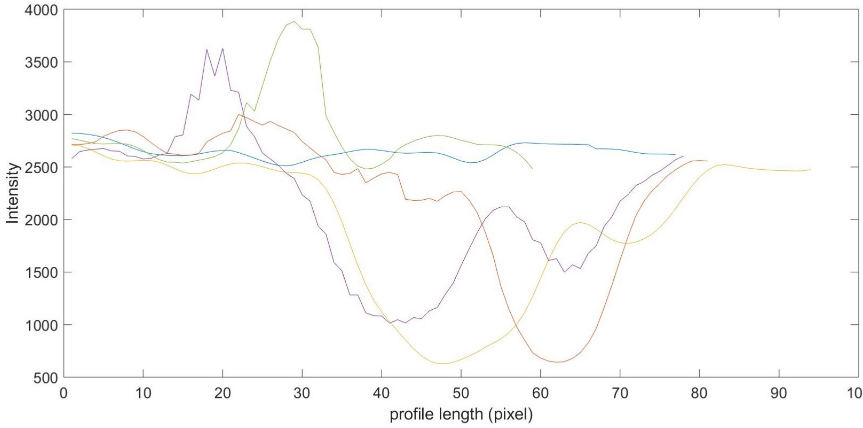
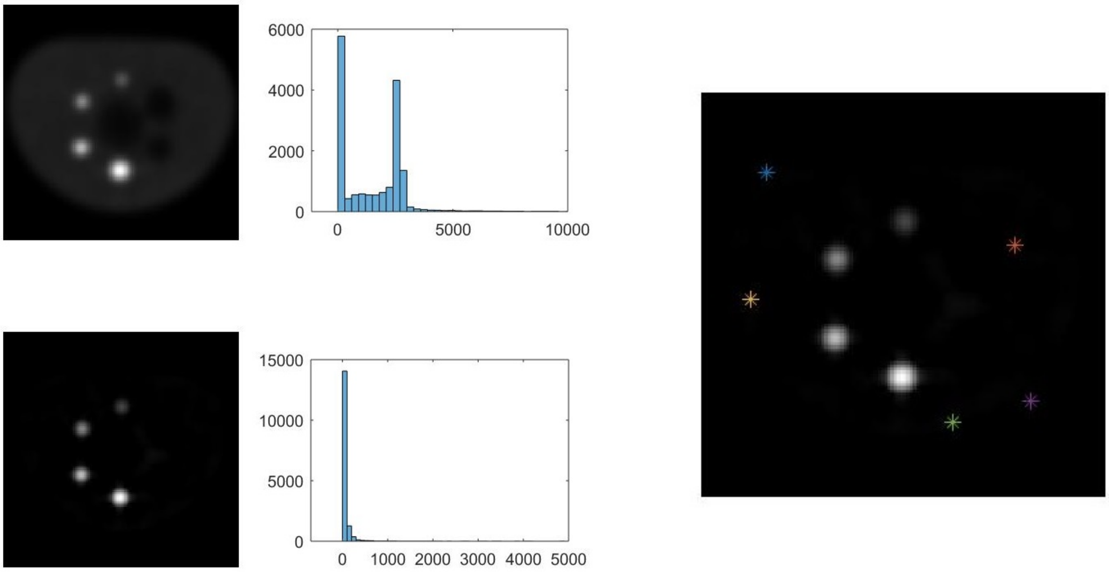


Figure C.2: Intensity profile obtained drawing different profiles on image.

After background correction with opening operation, the star points, related to brighter pixels, are distribuited more uniformly. In Figure C.3a is visible a comparison between PET image before and after correction with the associated histograms: the range is compressed and most of the values move to higher intensity. Instead, the distribution of star points in the new image is visible in Figure C.3b.



(a) Original and final image with histograms.

(b) Distribution of new star points.

Figure C.3: Graphical results after opening morphological application to non uniform background removal.

Matlab code used is replaced in the following lines:

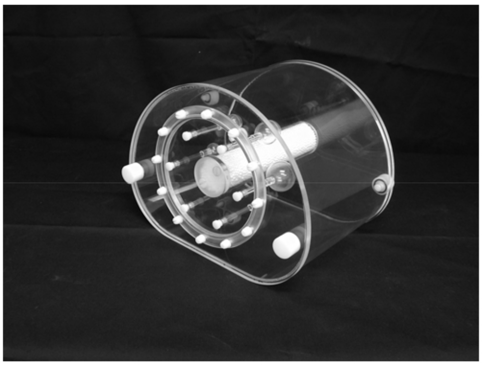
```matlab
%% Discrimination of background on right and on left(where there are hot spheres)
%It is uniform?

close all
clear all

%Image loading
V=dicomreadVolume(uigetdir);
%selection of one slice
img=V(:,:,18);
%application of gaussian filter
h=fspecial('gaussian',15,2.5);
img=conv2(img,h,'same');
figure(1)
imshow(img,[]);

set(gcf,'position',[0 115 560 420]);
[n,m]=size(img);

%check using profile (improfile)
uiwait(msgbox('Draw an horizontal profile'));
str='Yes';
answer='Yes';
p=0;
while strcmp(str,answer)==1
    [cx,cy,c]=improfile;
    figure(2)
    set(gcf,'position',[700 115 560 420]);
    plot(c)
    hold on
    %identification of most intensity pixel
    max_profile=max(c);
    index=find(c==max_profile);
    row=round(cx(index));
    column=round(cy(index));
    figure(1)
    hold on
    plot(row,column,'*');
    p=p+1;
    answer=questdlg('Another profile?');
    pause(0.1);
end

close(2)
figure(3)
imshow(img,[]);
set(gcf,'position',[0 115 560 420]);
uiwait(msgbox('Draw a ROI'));
str='Yes';
answer='Yes';

%check with ROI
```

```
while strcmp(str,answer)==1
    h=imellipse;
    mask1=createMask(h);
    meanValue1=mean(img(mask1));
    uiwait(msgbox(’Shift roi on the other side’));
    wait(h);
    mask2=createMask(h);
    meanValue2=mean(img(mask2));
    maxValue=max(meanValue1,meanValue2);
    %identification of most intensity center ROI value
    if maxValue==meanValue1
        figure(3)
        hold on
        [c,r]=find(mask1);
        plot(r(1),c(1),’*’);
    elseif maxValue==meanValue2
        figure(3)
        hold on
        [c,r]=find(mask2);
        plot(r(1),c(1),’*’);
    end
    delete(h)
    answer=questdlg(’Another ROI?’);
    pause(0.1);
end
set(gcf,’position’,[700 115 560 420]);
answer1=questdlg(’See the two image. Are the points uniformly distributed? Is the backgrou

%% Non uniform background removal with opening operation
close(figure(2));
close(figure(3));
if strcmp(str,answer1)==0
    figure(1)
    imshow(img,[]);
    uiwait(msgbox(’Select the principal object’));
    el=imellipse;
    c=getPosition(el);
    se=strel(’disk’,round(c(4)/2));
    background=imopen(img,se);
    img2=imsubtract(img,background);
    close(figure(1));
    figure
    subplot(2,2,1)
    imshow(img,[]);
    subplot(2,2,2)
    histogram(img);
    subplot(2,2,3)
    imshow(img2,[]);
    subplot(2,2,4);
    histogram(img2);
else
    close(figure(1));
    figure
    subplot(2,1,1)
    imshow(img,[]);
    subplot(2,1,2)
    histogram(img);
    img2=img;
end
set(gcf,’position’,[700 115 560 420]);

%% test the new image
if strcmp(str,answer1)==0
    img=img2;
    figure(2)
    imshow(img,[]);
    set(gcf,’position’,[0 115 560 420]);
    uiwait(msgbox(’Draw a ROI’));
    str=’Yes’;
    answer=’Yes’;
    while strcmp(str,answer)==1
        h=imellipse;
        mask1=createMask(h);
        meanValue1=mean(img(mask1));
        uiwait(msgbox(’Shift roi on the other side’));
        wait(h);
        mask2=createMask(h);
        meanValue2=mean(img(mask2));
        maxValue=max(abs(meanValue1),abs(meanValue2));
        if maxValue==abs(meanValue1)
            figure(2)
            hold on
            [c,r]=find(mask1);
            plot(r(1),c(1),’*’);
        elseif maxValue==abs(meanValue2)
            figure(2)
            hold on
            [c,r]=find(mask2);
            plot(r(1),c(1),’*’);
        end
        delete(h)
        answer=questdlg(’Another ROI?’);
        pause(0.1);
    end
end
```

Appendix D

Manual of PET phantom

NEMA IEC Body Phantom Set™



NEMA IEC Body Phantom™

Specifications:
Interior length of phantom: 180 mm
Fillable spheres (6) inner diameter: 10 mm, 13 mm, 17 mm, 22 mm, 28 mm, and 37 mm.
Distance from sphere plane to inside wall: 70 mm
Volume of empty cylinder: 9.7 liters
Cylindrical insert dimension:
Outside diameter: 51 mm
Length: 180 mm

NEMA IEC Body Phantom Set™

Model PET/IEC-BODY/P

Main Features:

- The NEMA IEC Body Phantom Set™ consists of a body phantom, a lung insert and an insert with six spheres with various sizes
- It is designed in accordance with the recommendations by the International Electrotechnical Commission (IEC) and modified by the National Electrical manufacturers Association (NEMA)
- It is recommended for use in the evaluation of reconstructed image quality in whole body PET imaging

Main Applications:

- Simulation of whole-body imaging especially using PET and camera-based coincidence imaging techniques
- Evaluation of reconstructed image quality in whole-body PET and camera-based coincidence imaging
- Determination of the coincidence count rate characteristics in brain and cardiac imaging
- Evaluation of the relationship between true coincidence count rate and radioactivity
- Determination of the address errors caused by address pile up
- Evaluation of the count loss correction scheme
- Research

* *International Standard: Radionuclide imaging devices – Characteristics and test conditions – Part 1: Positron emission tomographs*, International Electrotechnical Commission (IEC), 61675-1, Geneva, Switzerland, 1998.

* *Performance Measurements of Scintillation Cameras*, NEMA Standards Publication No. NU2, National Electrical Manufacturers Association (NEMA), Washington, D.C., 2001.

Appendix E

Table of correlation coefficient

	r_0										
N	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
3	100	94	87	81	74	67	59	51	41	29	0
4	100	90	80	70	60	50	40	30	20	10	0
5	100	87	75	62	50	39	28	19	10	3.7	0
6	100	85	70	56	43	31	21	12	5.6	1.4	0
7	100	83	67	51	37	25	15	8	3.1	0.6	0
8	100	81	63	47	33	21	12	5.3	1.7	0.2	0
9	100	80	61	43	29	17	8.8	3.6	1		0
10	100	78	58	40	25	14	6.7	2.4	0.5		0
11	100	77	56	37	22	12	5.1	1.6	0.3		0
12	100	76	53	34	20	9.8	3.9	1.1	0.2		0
13	100	75	51	32	18	8.2	3	0.8	0.1		0
14	100	73	49	30	16	6.9	2.3	0.5	0.1		0
15	100	72	47	28	14	5.8	1.8	0.4			0
16	100	71	46	26	12	4.9	1.4	0.3			0
17	100	70	44	24	11	4.1	1.1	0.2			0
18	100	69	43	23	10	3.5	0.8	0.1			0
19	100	68	41	21	9	2.9	0.7	0.1			0
20	100	67	40	20	8.1	2.5	0.5	0.1			0
25	100	63	34	15	4.8	1.1	0.2				0
30	100	60	29	11	2.9	0.5					0
35	100	57	25	8	1.7	0.2					0
40	100	54	22	6	1.1	0.1					0
45	100	51	19	4.5	0.6						0
	0	**0.05**	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**	**0.25**	**0.40**	**0.45**	**1**
50	100	73	49	30	16	8	3.4	1.3	0.4	0.1	
60	100	70	45	25	13	5.4	2	0.6	0.2		
70	100	68	41	22	9.7	3.7	1.2	0.3	0.1		
80	100	66	38	18	7.5	2.5	0.7	0.1			
90	100	64	35	16	5.9	1.7	0.4	0.1			
100	100	62	32	14	4.6	1.2	0.2				

Table E.1: Percentage probability $P_N(|r| \geq r_0)$ that N measures of two variables not linearly correlated give a correlation coefficient $|r| \geq r_0$. The empty boxes indicate a probability less than 0.05%.

References

[1] B. Foster, U. Bagci et al., *A Review on Segmentation of Positron Emission Tomography Images*, National Institute of Health, 2015.

[2] A.S. Dewalle-Vignion, N. Betrouni et al., *A New Method for Volume Segmentation of PET Images, Based on Possibility Theory*, IEEE Transactions on Medical Imaging, Volume 30 – $No.2$, 2011.

[3] Florent L. Besson, Théophraste Henry et al., *Rapid Contour-based Segmentation for 18F-FDG PET Imaging of Lung Tumors by Using ITK-SNAP: Comparison to Expert-based Segmentation*, Radiology, 2018.

[4] Klaus P. Schäfers, *The promise of nuclear medicine technology: Status and future perspective of high-resolution whole-body PET*, Elsevier, 2008.

[5] M. Soret, S.L. Bacharach, and I. Buvat, *Partial-Volume Effect in PET Tumor Imaging*, Journal of Nuclear Medicine, 2007.

[6] Jeffrey A. Fessler, *Penalized Weighted Least-Squares Image Reconstruction for Positron Emission Tomography*, IEEE Transactions on MedicaL Imaging, Volume 13 – $No.2$, 1994

[7] T. Mou, J. Huang, and F. O´ Sullivan, *The Gamma Characteristic of Reconstructed PET Images: Implications for ROI Analysis*, IEEE Transactions on MedicaL Imaging, Volume 37 – $No.5$, 2018.

[8] W. Parker, H. Patrocinio, *Clinical Treatment Planning in External Photon Beam Radiotherapy*, IAEA Publications, 2012.

[9] Zhen Ma, João Manuel R. S. Tavares, R. M. Natal Jorge, *A review on the current segmentation algorithms for medical images*, International Conference on Imaging Theory and Applications (IMAGAPP), 2009.

[10] Mathieu Hatt, Baptiste Laurent , Anouar Ouahabi et al., *The first MICCAI challenge on PET tumor segmentation*, Medical Image Analysis, 2017.

[11] Habib Zaidi, Mehrsima Abdoli et al., *Comparative methods for PET image segmentation in pharyngolaryngeal squamous cell carcinoma*, European Journal of Nuclear Medicine and Molecular Imaging, 2012.

[12] R.R. Beichel et al., *Semiautomated segmentation of head and neck cancers in 18F-FDG PET scans: A just-enough-interaction approach*, American Association of Physicists in Medicine, 2016.

[13] E. Di Martino, B. Nowak, *Diagnosis and Staging of Head and Neck Cancer*, Archives of Otolaryngology - Head & Neck Surgery, 2000.

[14] D. A. X. Schinagl, W. V. Vogel, *Comparison of five segmentation tools for 18F - Fluoro - Deoxyglucose - Positron Emission Tomography - Based target volume definition in head and neck cancer*, International Journal of Radiation Oncology, Biology, Physics, 2007.

[15] X. Geets, J.A. Lee et al., *A gradient-based method for segmenting FDG-PET images: methodology and validation*, European Journal of Nuclear Medicine and Molecular Imaging, 2007

[16] JF. Daisne, M. Sibomana et al., *Tri-dimensional automatic segmentation of PET volumes based on measured source-to-background ratios: influence of reconstruction algorithms*, Radiotherapy & Oncology, 2003.

[17] J. C. Dunn, *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters*, Journal of Cybernetics, 1973.

[18] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algoritms*, Plenum Press, New York, 1981.

[19] T. Velmurugan and T. Santhanam, *Implementation Of Fuzzy C-Means Clustering Algorithm For Arbitrary Data Points*, International Conference on Systemics, Cybernetics and Informatics, 2011.

[20] Shilpa Kamdi, R.K.Krishna, *Image Segmentation and Region Growing Algorithm*, International Journal of Computer Technology and Electronics Engineering (IJCTEE), Volume 2, Issue 1, 2012.

[21] W. Cui, Z. Guan, Z. Zhang, *An Improved Region Growing Algorithm for Image Segmentation*, International Conference on Computer Science and Software Engineering, 2008.

[22] G. Deng and L. W. Cahill, *An Adaptive Gaussian Filter For Noise Reduction and Edge Detection*, Conference Record Nuclear Science Symposium and Medical Imaging Conference, 1993.

[23] Jack Xin and J. Ernie Esser, *Filtering and Convolution*, University of California, Department of Mathematics.

[24] Peter Kovesi, *Fast Almost-Gaussian Filtering*,Digital Image Computing, Techniques and Applications, 2010.

[25] Ravi S., A. M. Khan, *Morphological Operations for Image Processing : Understanding and its Applications*, National Conference on VLSI, Signal processing & Communications, 2013.

[26] Prabhdeep Singh, Dr. A K Garg, *Non Uniform Background Removal using Morphology based Structuring Element for Particle Analysis*, International Journal of Computer Applications, Volume $33 - No.6$, 2011.

[27] K.H. Zou, S.K. Warfield et al., *Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index: scientific reports*, Academic Radiology, Elsevier, 2004.

[28] Abdel Aziz Taha and Allan Hanbury, *An Efficient Algorithm for Calculating the Exact Hausdorff Distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume $37 - No.11$, 2015.

[29] C.R. Hansen, R.L. Christiansen et al., *Contouring and dose calculation in head and neck cancer radiotherapy after reduction of metal artifacts in CT images*, 4th Symposium of Nordic Association for Clinical Physics, 2017.

[30] J. Yang, B.M. Beadle et al., *A multimodality segmentation framework for automatic target delineation in head and neck radiotherapy*, Medical Physics, 2015.

[31] L.J. Stapleford, J.D. Lawson et al.,*Evaluation of automatic atlas - based lymph node segmentation for head and neck cancer*, International Journal of Radiation Oncology, Biology, Physics, 2010.

[32] J.R.Taylor, *An introduction to Error Analysis:The Study of Uncertainties in Physical Measurements*, University Science Books, 1997.