

*Texture* — User Guide

Local Image Features **Extraction**  
— **LIFEx** —

F. Orlhac, C. Nioche, I. Buvat

LIFEx version 4.nn,  
Last update of document: 2018/04/20



**Part I Protocol description** page 5

Prerequisite	7
Current knowledge	
Step by step	
GUI of Texture protocol	
Intensity quantization and current limits	
Spatial resampling and current limits	
Calculation and Results	
Scripting procedure	

**Part II Texture Features** page 19

<b>GLCM</b>	GLCM	Homogeneity	Energy	Contrast	Correlation	Entropy	Entropy_log2	Dissimilarity	Save results	21
<b>NGLDM</b>	NGLDM	Coarseness	Contrast	Busyness						25
<b>GLRLM</b>	GLRLM	SRE, LRE	LGRE, HGRE	SRLGE, SRHGE	LRLGE, LRHGE	GLNUr, RLNU	RP			27
<b>GLZLM</b>	GLZLM	SZE, LZE	LGZE, HGZE	SZLGE, SZHGE	LZLGE, LZHGE	GLNUz, ZLNU	ZP			31

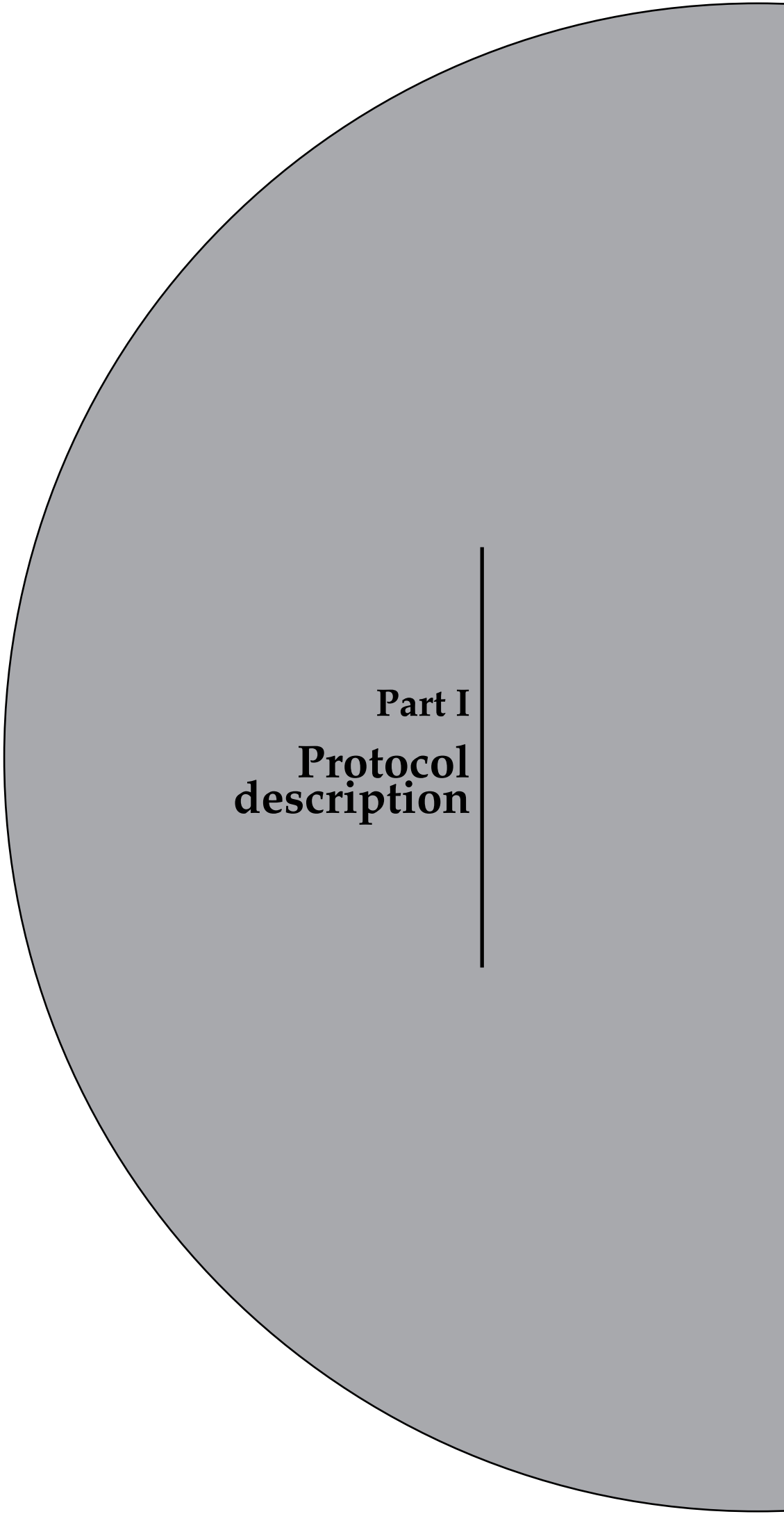
**Part III First Order Features** page 35

<b>Indices from Shape</b>	Sphericity	Compacity	Volume							37
<b>Indices from Histogram</b>	Histogram calculation	Skewness	Kurtosis	Entropy	Entropy_log2	Energy				39
<b>Conventional Indices</b>	SUVmin, SUVmean, SUVmax	SUVpeak	TLG							43

**Part IV References** page 45

<b>Useful references</b>	Bibliography	47
--------------------------	--------------	----





**Part I**  
**Protocol  
description**

---



**Chapter 1**

**How to perform textural analysis  
and estimate conventional  
parameters?  
– step by step –**

### 1.1 Prerequisite

Textural analysis and measurement of conventional parameters from images are based on a number of rules related to the image series, ROI and pixel values, as described below:

#### Image series

- Textural analysis is currently supported for PET, MR, CT, US and NM. Calculations for Computed radiography (CR), Digital Radiography (DX), Radio Fluoroscopy (RF), Slide Microscopy (SM) and Optical Images are available yet only if their are in TIFF or JPG formats;
- Texture indices are calculated in a ROI for one or several layers (1 layer = 1 image series). To properly interpret the index values, it is important to know that the index are calculated for the selected image series (active layer) and for the voxel sampling of the selected image series.

#### ROI:

- The ROI should include at least 64 voxels, and can be a 2D or 3D ROI;
- The ROI should be a single piece (contiguous voxels). It's possible to calculate no contiguous voxels when you use script function, then only the largest cluster is considered for processing);
- The ROI definition is an important step, as edges can largely affect textural index values as well as conventional index values ([Orlhac et al 2017](#)).
- Using scripts may produce numerous ROIs. To check that the resulting ROI are compatible with texture index calculation and before running such calculations that can be time-consuming, you can use the CheckTex button after the ROI creation and before proceeding with textural index calculation. The corresponding action will test that the ROI includes a single cluster and contain a number of voxels greater than that required for consistent textural feature calculation. If one of these two conditions is not met, a warning message will be displayed.

#### Input parameters:

- Number of grey levels: this is the number of grey levels (must be >0) used to resample the ROI content. Can typically be 16, 32, 64, 128. The initial pixel values will be resampled into that number of grey levels;
- Size of bin: the width of bin;
- Resampling: can be absolute (between fixed min and max values regardless of the ROI content) or relative (between the min and max values in the ROI); See [Orlhac et al Plos One 2015](#) for the impact of resampling approach onto tissue classification;
- Min, max bound: to be set for absolute resampling only: min and max bounds defining the range of resampled values.



## 1.2 Current knowledge

**Validated knowledge with TEP images.** Texture matrices are computed after a resampling step with 64 discrete values, between 0 and 20 SUV units, without spatial resampling [Orlhac 2015]. Six texture indices robust to segmentation chosen in a group of highly correlated indices are conventionally computed:

- GLCM\_Homogeneity (from GLCM),
- GLCM\_Entropy (from GLCM),
- GLRLM\_Short-Run Emphasis (GLRLM\_SRE, from GLRLM),
- GLRLM\_Long-Run Emphasis (GLRLM\_LRE, from GLRLM),
- GLZLM\_Low Gray-level Zone Emphasis (GLZLM\_LGZE, from GLZLM),
- GLZLM\_High Gray-level Zone Emphasis (GLZLM\_HGZE, from GLZLM).

**Always in research.** A set of 32 texture indices, 4 histogram indices, 3 shape indices and 4 conventional PET features are computed. For the resampling step, the user can choose the number of discrete values (default: 64) and the range (default: 0-20 SUV units) and the spatial resampling (default: dimension of voxel of images).

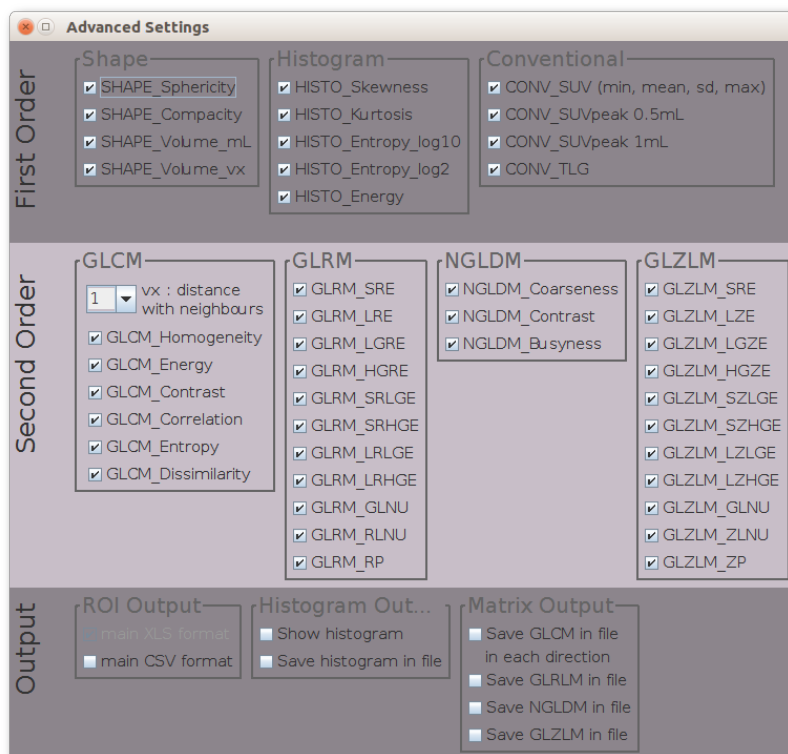


Figure 1.1: Texture feature extract (Advanced settings frame)

## 1.3 Step by step

### 1.3 Step by step

This section describes the steps needed to get textural and conventional index estimates.

1. Reading of the DICOM images for which index should be calculated ([link](#));
2. Drawing or loading the ROI ([link](#));
3. Definition of the input parameters: default values can be changed, and parameters without default values should be set;
4. Definition of the output parameters of interest: index to be calculated should be selected;
5. Running the calculation: click RUN in the textural index setting window;
6. The folder containing the Excel file listing the results is finally opened.

### 1.4 Graphic User Interface of Texture protocol

This section describes how to get textural and conventional index estimates.

The user interface (see figure 1.2 p.11) includes 3 sub-parts:

1. The "Spatial Resampling" and "Intensity Resampling" and "Intensity Discretization"
2. A panel to set advanced parameters (see figure 1.1 p.9)
3. The "Run" button to start the calculations

### 1.5 Intensity quantization and current limits

In PET, several studies already investigated the impact of resampling settings. Default values for number of grey levels, bin width, resampling approach, and bounds, are set based on these studies.

In MR, CT, US, NM, CR, DX, RF, there is a lack of studies regarding the resampling settings. Studies are on-going to define robust parameter values. Meanwhile, users should interpret the results with caution.

Three types of intensity resampling are available:

**Absolute resampling bounds:**  $\min \longleftrightarrow \max$  ROI values are rescaled between the *min* and *max* values as set in the parameter box. These values should be set in the appropriate units depending on the image type (Hounsfield units, SUV, #). All voxels are accounted for in the texture index calculation.

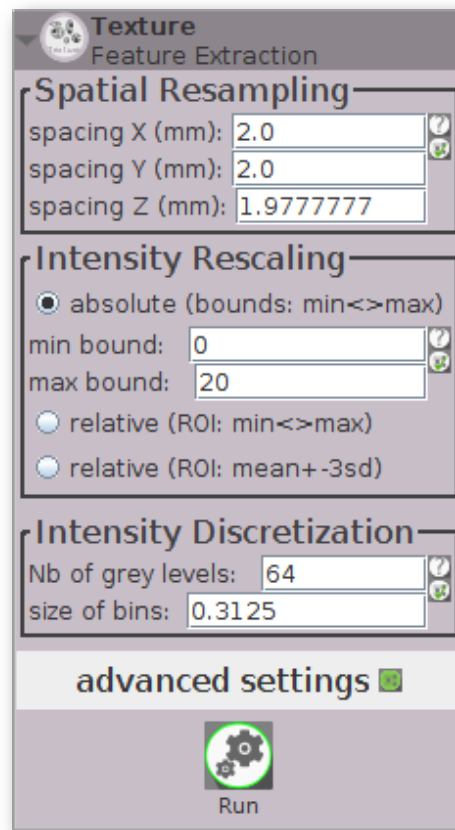


Figure 1.2: Display of textural GUI

**Relative resampling of ROI values:  $\min \iff \max$**  ROI values are automatically rescaled between the *min* and *max* values of the ROI content. All voxels are accounted for in the texture index calculation. Voxels with an initial value less than *min* are set to *min*. Voxels with an initial value greater than *max* are set to *max*.

**Relative resampling of ROI values:  $\text{mean} - 3 * \text{Sd} \iff \text{mean} + 3 * \text{Sd}$**  ROI values are automatically rescaled between  $\text{mean} - 3 * \text{Sd} \iff \text{mean} + 3 * \text{Sd}$  of the ROI content, where mean and Sd are the mean and standard deviation of the voxels included in the ROI. All voxels are accounted for in the texture index calculation. Voxels with an initial value less than  $\text{mean} - 3 * \text{Sd}$  are set to  $\text{mean} - 3 * \text{Sd}$ . Voxels with an initial value greater than  $\text{mean} + 3 * \text{Sd}$  are set to  $\text{mean} + 3 * \text{Sd}$ .

**Recommendation** In PET, we do not recommend to use the relative resampling. As shown in [Orlhac 2015], relative resampling results in a high correlation between many texture indices (TI) and metabolic volume, that is mostly removed when using absolute resampling. Also, changes in TI can be counter-intuitive when using relative resampling instead of absolute resampling [Orlhac 2017]. In PET, resampling using 64 bins between 0 and 20 SUV units (e.g., size bin equal to 0.3) most often yields robust results (0-20 being a frequent range of SUV in oncology with FDG), although the maximum value may need to be changed in case of uptake actually higher than SUV=20.

## 1.6 Spatial resampling and current limits

Textural feature values can be significantly affected by the voxel size (Orlhac et al J Nucl Med 2016, Shafiq-Ul-Hassan et al Med Phys 2017). The voxel size to be used to calculate the textural feature values can now be set in the "spacing X, Y, Z" fields in the textural feature dialog box. Unit is millimeter.

To avoid any memory (RAM) saturation, the values should be set in the range defined by  $0.2 * \text{Min}(\text{Xspacing}, \text{Yspacing}, \text{Zspacing})$  (lower limit) and  $5 * \text{Max}(\text{Xspacing}, \text{Yspacing}, \text{Spacing})$  (upper limit). If this is not appropriate for you, please do not hesitate to contact us as [contact@lifexsoft.org](mailto:contact@lifexsoft.org). In case of RAM saturation, an error message will be displayed: the voxel size in which the value should be performed should then be increased, or the RAM should be increased and the LIFEx application restarted.

A polygon of degree 5 (Lagrangian polygon) is used for interpolating the 3D voxel value within the ROI and performing the textural feature calculation at the requested voxel size.

## 1.7 Calculation and Results

**Calculation:** Calculation starts when the RUN button is clicked. All ROI selected on the reference image series are used for calculation.

**Error messages:** Error messages will be displayed if the prerequisites for calculation are not met. Once the prerequisites are met, calculations should run smoothly. If not, please contact [lifexsoft.org](mailto:lifexsoft.org).

**3D calculation computational speed:** To get the results, about 13 sec are needed for a 3D ROI including 1 million of voxels on standard computers (DELL Precision T5810 with Intel Xeon E5-1660 v3 3GHz, 32 Go RAM, Windows 10 64 bits.). Smaller ROI, leading to faster calculations, are usually used.

**2D calculation computational speed:** Calculations in 2D ROI yield instant results.

**Results ("patient" file):** Results of textural and conventional index are stored in an Excel file named by default `patientName_patientID_Texture_time.xls`; the *time* variable is an approximate estimate of date and time and is different for each created file (atomic variable).

The Result file is always created in the folder `workdir\patientName\RoiValue`.

**Results (session file):** A session file makes it possible to add columns in a result file for multi-patients study. It also makes it possible to close the application without losing any result. For each new calculation, the application will update the session file if you check that option on advanced parameters frame.

If the session file does not exist yet, it is automatically created in the `workdir` folder (see [LIFEx\\_UserGuide](#)). The folder including the session file corresponds to the folders in which all patient folders are created. The directory `workdir` can be changed in the LIFEx preferences window (see Bottom toolbar p.27 in LIFEx UserGuide).

You can create several session files, by modifying the beginning of the file name, keeping `TextureSession.xls` at the end to make it recognizable by the application. If several session files are found by the application, you will be asked to select which one has to be updated.

**Cautious:** updating a previous excel file with additional index values is relevant only if the previously calculated index values are unchanged. For instance, file updating can be relevant for multi-patient studies, or multi-ROI studies, in which always the same indices are calculated.

## 1.8 Scripting procedure

Scripting procedure for texture calculation without user interaction

**Rationale** When you have a large number of ROI and/or a large number of patients in which you want to calculate usual indices (SUV, Volume, ...) and/or histogram-based or textural features, it can be convenient to run a script that will do all the calculations for you without any user interaction. ROI have to be prepared beforehand. You can then run the script many times if you want to perform the calculations using different sets of parameters.

For instance: You have 10 ROI per patient and a set of 100 patients to be processed. You are interested in studying the impact of the *nbGreyLevels* parameter on the 10\*100 ROI, by setting this parameter to 64, 128 and 256. You can create three script files that will all be identical except for the line that sets the *nbGreyLevels* parameter. Running the script will calculate all indices for you automatically and store them in excel files. You will be able to run all three scripts using a single command line, see the « Can I run several script files at once? » p.16.

Another example: You have 10 ROI per patient and a set of 100 patients to be processed. You are interested in studying the impact of the *spatial resampling* parameter on the 10\*100 ROI, by setting this parameter to 2x2x2 mm and to 4x4x4 mm. You can write and run a first script by setting the *spatial resampling* parameter to 2x2x2 mm. Then duplicate the script and change 2x2x2 mm by 4x4x4 mm to produce a second script to be run. Running the two scripts will calculate all indices for you automatically and store them in excel files. Setting the spatial resampling parameter to 2x2x2 mm write on script file:

```
# SpatialResampling
LIFEx.texture.SpatialResamplingZ=2
LIFEx.texture.SpatialResamplingY=2
LIFEx.texture.SpatialResamplingX=2
```

Using scripts may produce numerous ROIs. To check that the resulting ROI are compatible with texture index calculation and before running such calculations that can be time-consuming, you can use the CheckTex button after the ROI creation and before proceeding with textural index calculation. The corresponding action will test that the ROI includes a single cluster and contain a number of voxels greater than that required for consistent textural feature calculation. If one of these two conditions is not met, a warning message will be displayed.

**What is a script file?** A script file makes it possible to run all operations and calculations without any user interaction. You can prepare it in advance. You can have as many script files as you want, with names that you choose and you can save them and modify them. They are simple text files.

**How to write a script file?** The script file describes a list of functions to be executed within LIFEx.

- a) Reading the patient file;
- b) Reading one or several ROI files for that patient;

- c) Setting the parameters involved in the index calculation;
- d) Calculation of the different indices;
- e) Writing results in a new on existing Excel file;
- f) Closing all files;
- g) Back to step a) if needed.

Here is an example of script file based on a example provided with LIFEx. You can save it as `TextureProps1.txt`. The lines starting with `#` are non interpreted by LIFEx, they only include comments.

```
# start of file

# Common
LIFEx.texture.BinSize=3.125
LIFEx.texture.NbGrey=128.0
LIFEx.texture.sessionXls=D:\TextureSessionProps1.xls

# Absolute
LIFEx.texture.ButtonAbsolute=true
LIFEx.texture.MinBound=0.0
LIFEx.texture.MaxBound=400.0

# RelativeMeanSd
LIFEx.texture.ButtonRelativeMeanSd=false

# RelativeMinMax
LIFEx.texture.ButtonRelativeMinMax=false

# SpatialResampling
LIFEx.texture.SpatialResamplingZ=0
LIFEx.texture.SpatialResamplingY=0
LIFEx.texture.SpatialResamplingX=0

# DistanceWithNeighbours
LIFEx.texture.GLCM.DistanceWithNeighbours=1

# Patient0
LIFEx.texture.Img0=D:\nii\IM-0001-0001-0001.nii
LIFEx.texture.Img0Roi0=D:\roi\C1.nii.gz
LIFEx.texture.Img0Roi1=D:\roi\C2.nii.gz

# Patient1
LIFEx.texture.Img1=D:\nii\IM-0001-0001-0001.nii
LIFEx.texture.Img1Roi0=D:\roi\C1.nii.gz
LIFEx.texture.Img1Roi1=D:\roi\C2.nii.gz
LIFEx.texture.Img1Roi2=D:\roi\C3.nii.gz

# end of file
```

## 1.8 Scripting procedure

Here are the functions that are executed when running this text file:

- Reading the images (Img0) of patient0
- Reading ROI0 of patient0 (Img0Roi0)
- Reading ROI1 of patient0 (Img0Roi1)
- Reading the setting of the Common, Absolute, RelativeMeanSd, RelativeMinMax, SpatialResamplingZYX parameters
- Calculating all indices.
- Writing the results in an existing Excel session file in a file named `session.xls`
- Closing all ROI and patient0 data

then:

- Reading the images (Img1) of patient1
- Reading ROI0 of patient1 (Img1Roi0)
- Reading ROI1 of patient1 (Img1Roi1)
- Reading ROI2 of patient1 (Img1Roi2)
- Reading the setting of the Common, Absolute, RelativeMeanSd, RelativeMinMax, SpatialResamplingZYX parameters
- Calculating all indices for ROI0 and ROI1.
- Writing the results in an existing Excel session file in a file named `session.xls`
- Closing all ROI and patient1 data

**How to run a script file.** Check before running a script:

1. Please check that you have no session file in the destination directory that you have specified in your script file;
2. The result filenames should NOT be `*TextureSession.xls`

Running the script :

- Please drag the script file in the panel used to load patient images.
- Alternatively, you can use the « browser + LocalDisk » interface to find and load your script file from your local disk.

**Can I run several script files at once?** You can indeed run several script files one after another automatically. To do so, please select all script files and drag them in the panel used to put the patient image files. The script files will be executed one after another.

Beware: you can not load your script files using the « browser » interface, which support only one script file at a time.



**Advanced execution scripting of grey-level resampling.** When you set the quantization (or grey-level resampling) parameters, you can leave one of the two set to 0. Its value will then be automatically calculated based on the other parameter setting.

```
LIFEx.texture.BinSize=3.125
LIFEx.texture.NbGrey=128.0
```

For instance:

If BinSize = 0 then  $\text{BinSize} = (\text{boundMax} - \text{boundMin}) / \text{nbGreyLevels}$

If nbGreyLevels = 0 then  $\text{nbGreyLevels} = (\text{boundMax} - \text{boundMin}) / \text{binSize}$

If BinSize = 0 and nbGreyLevels = 0 then nbGreyLevels = 64 by default

boundMin and boundMax are the min and max value in the processed ROI.

**Advanced execution scripting of spatial resampling.** When you set the spatial resampling parameters, you can leave the three parameters *SpatialResamplingZ*, *SpatialResamplingY*, *SpatialResamplingX* set to 0. This is equivalent to performing the calculation using the original voxel size of the images.

If you want to change the voxel size of the image, set the parameters to a value expressed in millimeters. The three values (*SpatialResamplingZ*, *SpatialResamplingY*, *SpatialResamplingX*) can be different.

#### Main remarks relevant to the writing of a script in LIFEx:

- What are the image formats that can be managed using LIFEx scripts?
  - image files can be in nifti-1 format. In this case, the complete pathway to the patient image file should be given in the script;
  - to load DICOM images, you must define the root directory of these images (without the final filename); All files included in this directory will be loaded;
- What are the ROI formats that can be managed using LIFEx scripts?
  - the nifti-1 and RTStruct are supported in LIFEx scripts. In both cases, the file name should include the full path to the file.
- Syntax of all pathways of files in LIFEx scripts:
  - the syntax of pathway not necessarily must conform to your system rules:
    - \* Windows: Unit:/Directory/File.extension or Unit:/Your Directory/ for series of DICOM images
    - \* Linux: /Your Directory/Your File.extension
    - \* MacOS: /Your Directory/Your File.extension

If this does not work, feel to try to reverse the slash.





**Part II**  
**Texture Features**



**Chapter 1**  
**GLCM definition**

## 1.1 GLCM

The grey level co-occurrence matrix (GLCM) [Haralick] takes into account the arrangements of pairs of voxels to calculate textural indices. The GLCM is calculated from 13 different directions in 3D with a  $\delta$ -voxel distance ( $\|\vec{d}\|$ ) relationship between neighbored voxels. The index value is the average of the index over the 13 directions in space  $(X, Y, Z)$ . Seven textural indices are computed from this matrix. An entry  $(i, j)$  of GLCM for one direction is equal to:

$$GLCM_{\Delta x, \Delta y}(i, j) = \frac{1}{Pairs_{ROI}} \sum_{p=1}^{N-\Delta x} \sum_{q=1}^{M-\Delta y} \begin{cases} 1 & \text{if } (I(p, q) = i, I(p + \Delta x, q + \Delta y) = j) \\ & \text{and } I(p, q), I(p + \Delta x, q + \Delta y) \in ROI \\ 0 & \text{otherwise} \end{cases} \quad (1.1)$$

where  $I(p, q)$  corresponds to voxel  $(p, q)$  in an image  $(I)$  of size  $N * M$ . The vector  $\vec{d} = (\Delta x, \Delta y)$  covers the 4 directions (D1, D2, D3, D4, Figure 1.1) in 2D space or 13 directions (D1, D2, ..., D13, Figure 1.2) in 3D space and  $Pairs_{ROI}$  corresponds to the number of all voxel pairs belonging to the region of interest (ROI).

**Cost-effective creation of a transposed matrix.** To reduce runtime calculation of GLCM matrix involving 26 directions in 3D (8 directions in 2D) the transposed matrix accounting for 13 directions calculated in 3D (and 4 in 2D) is added to the original matrix.

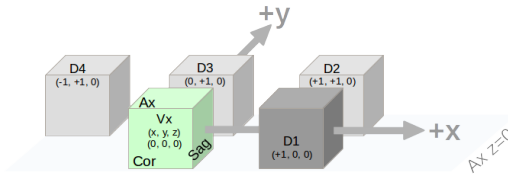


Figure 1.1: 2D Voxel (green) and associated 4 directions

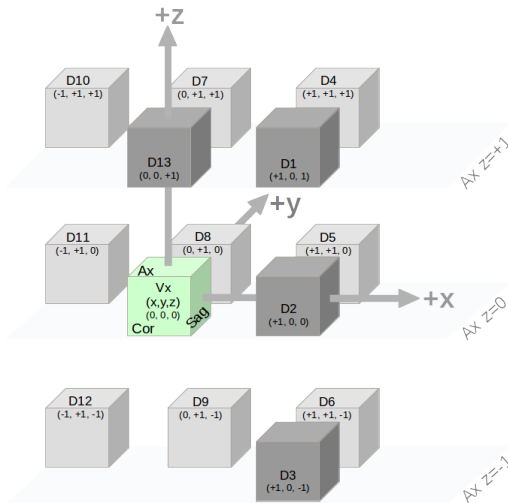


Figure 1.2: 3D Voxel (green) and associated 13 directions

The GLCM describes the distribution of co-occurring pixel values at a given offset. This offset ( $\delta$ ) set to 1 by default, meaning that only neighbour voxels are used to calculate GLCM. It is possible to change this default value from 1 to 10 in the "Distance with neighbours" box, which is in the "Advanced Settings" dialog box, GLCM subpart.

If you specify a distance of 5 voxels the matrix will be built not with the nearest neighbor (dist = 1) but with the neighbor with 5 voxels of distance and without taking into account the intermediate voxels of distance 2, 3, 4.

Be careful: if the distance is too large for small ROIs, some entries of the GLCM matrix will not have any meaning. This is detected and the corresponding results in the output Excel file are noted NaN! or Num!

GLCM

## 1.2 GLCM\_Homogeneity

is the homogeneity of grey-level voxel pairs.

$$GLCM\_Homogeneity = \text{Average over 13 (or 4) directions} \left( \sum_i \sum_j \frac{GLCM(i, j)}{1 + |i - j|} \right) \quad (1.2)$$

## 1.3 GLCM\_Energy

also called Uniformity or Second Angular Moment, is the uniformity of grey-level voxel pairs.

$$GLCM\_Energy = \text{Average over 13 (or 4) directions} \left( \sum_i \sum_j GLCM(i, j)^2 \right) \quad (1.3)$$

## 1.4 GLCM\_Contrast

also called Variance or Inertia, is the local variations in the GLCM.

$$GLCM\_Contrast = \text{Average over 13 (or 4) directions} \left( \sum_i \sum_j (i - j)^2 \cdot GLCM(i, j) \right) \quad (1.4)$$

## 1.5 GLCM\_Correlation

is the linear dependency of grey-levels in GLCM.

$$GLCM\_Correlation = \text{Average over 13 (or 4) directions} \left( \sum_i \sum_j \frac{(i - \mu_i) \cdot (j - \mu_j) \cdot GLCM(i, j)}{\sigma_i \cdot \sigma_j} \right) \quad (1.5)$$

where  $\mu_i$  or  $\mu_j$  corresponds to the average on row  $i$  or column  $j$  and  $\sigma_i$  and  $\sigma_j$  correspond to the variance on row  $i$  or column  $j$ .

## 1.6 Entropy

### 1.6 GLCM\_Entropy\_log10

is the randomness of grey-level voxel pairs.

$$GLCM\_Entropy_{\log_{10}} = \text{Average over 13 (or 4) directions} \left( - \sum_i \sum_j GLCM(i, j) \cdot \log_{10}(GLCM(i, j) + \epsilon) \right) \quad (1.6)$$

where  $\epsilon = 2e-16$ .

Be aware of the logarithm used in the formula. We use the logarithm with base 10 and base 2 in LIFEx ; see `_log2` formula.

### 1.7 GLCM\_Entropy\_log2

is the randomness of grey-level voxel pairs.

$$GLCM\_Entropy_{\log_2} = \text{Average over 13 (or 4) directions} \left( - \sum_i \sum_j GLCM(i, j) \cdot \log_2(GLCM(i, j) + \epsilon) \right) \quad (1.7)$$

where  $\epsilon = 2e-16$ .

### 1.8 GLCM\_Dissimilarity

is the variation of grey-level voxel pairs.

$$GLCM\_Dissimilarity = \text{Average over 13 (or 4) directions} \left( \sum_i \sum_j |i - j| \cdot GLCM(i, j) \right) \quad (1.8)$$

### 1.9 Save results for each direction

Textural indices corresponding to each direction can be saved independently (13 directions in 3D and 4 directions in 2D). To do that, please tick the "SaveGLCM in file" option in the "Output" sub-dialog box.



**Chapter 2**  
**NGLDM definition**

## 2.1 NGLDM

The neighborhood grey-level different matrix (NGLDM) [Amadasum1989] corresponds to the difference of grey-levels between one voxel and its 26 neighbours in 3 dimensions (8 in 2D). Three texture indices can be computed from this matrix. An element  $(i, 1)$  of NGLDM corresponds to the probability of occurrence of level  $i$  and an element  $(i, 2)$  is equal to:

$$NGLDM(i,2) = \sum_p \sum_q \begin{cases} |\bar{M}(p,q) - i| & \text{if } I(p,q) = i \\ 0 & \text{else} \end{cases} \quad (2.1)$$

where  $\bar{M}(p,q)$  is the average of intensities over the 26 neighbour voxels of voxel  $(p,q)$ .

## 2.2 NGLDM\_Coarseness

is the level of spatial rate of change in intensity.

$$NGLDM\_Coarseness = \frac{1}{\sum_i NGLDM(i,1) \cdot NGLDM(i,2)} \quad (2.2)$$

## 2.3 NGLDM\_Contrast

is the intensity difference between neighbouring regions.

**vs. pyRadiomic.** We must highlight that comparisons of results with other software supporting texture analysis should be performed with great care. The calculation of the texture indices resulting from the matrix NGLDM can differ between software. For instance, after the calculation of the matrix NGLDM, in LIFEx, the Contrast formula includes the number of voxels  $E$  in the region in the denominator as in Amadasum & King ([link](#)) formula (see below), while this number of voxels is squared in pyRadiomics, explaining the differences in resulting values.

$$NGLDM\_Contrast = \left[ \sum_i \sum_j NGLDM(i,1) \cdot NGLDM(j,1) \cdot (i-j)^2 \right] \cdot \frac{\sum_i NGLDM(i,2)}{E \cdot G \cdot (G-1)} \quad (2.3)$$

where  $E$  corresponds to the number of voxels in the Volume of Interest and  $G$  the number of grey-levels.

## 2.4 NGLDM\_Busyness

is the spatial frequency of changes in intensity.

$$NGLDM\_Busyness = \frac{\sum_i NGLDM(i,1) \cdot NGLDM(i,2)}{\sum_i \sum_j |(i \cdot NGLDM(i,1) - j \cdot NGLDM(j,1))|} \quad (2.4)$$

*with  $NGLDM(i,1) \neq 0, NGLDM(j,1) \neq 0$*

**Chapter 3**  
**GLRLM definition**

### 3.1 GLRLM

The grey-level run length matrix (GLRLM) [Xu] gives the size of homogeneous runs for each grey level. This matrix is computed for the 13 different directions in 3D (4 in 2D) and for each of the 11 texture indices derived from this matrix, the 3D value is the average over the 13 directions in 3D (4 in 2D). The element  $(i, j)$  of GLRLM corresponds to the number of homogeneous runs of  $j$  voxels with intensity  $i$  in an image and is called  $GLRLM(i, j)$  thereafter.

**vs. pyRadiomic.** We must highlight that comparisons of results with other software supporting texture analysis should be performed with great care. The calculation of the texture indices resulting from the matrix GLRLM can differ between software. For instance, in pyRadiomics (v1.1.1), after the calculation of the matrix GLRLM and before the extraction of the textural indices, the matrix is cropped (grey-level axis of GLRLMs cropped between minimum and maximum grey-levels and run-length axis of GLRLMs cropped to maximum run-length). This moves indexes  $(i, j)$  of the matrix and thus the values of the resulting textural indices.

In LIFEx, we do not shift the index so that it corresponds to grey level  $i$ , and  $j$  corresponds to the number of run  $j$  and we use the formulations defined below.

### 3.2 GLRLM\_SRE, GLRLM\_LRE

**Short-Run Emphasis** or **Long-Run Emphasis** is the distribution of the short or the long homogeneous runs in an image.

$$GLRLM\_SRE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j \frac{GLRLM(i, j)}{j^2} \right) \quad (3.1)$$

$$GLRLM\_LRE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j GLRLM(i, j) \cdot j^2 \right) \quad (3.2)$$

where  $H$  corresponds to the number of homogeneous runs in the Volume of Interest.

### 3.3 GLRLM\_LGRE, GLRLM\_HGRE

**Low Gray-level Run Emphasis** or **High Gray-level Run Emphasis** is the distribution of the low or high grey-level runs.

$$GLRLM\_LGRE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j \frac{GLRLM(i, j)}{i^2} \right) \quad (3.3)$$

$$GLRLM\_HGRE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j GLRLM(i, j) \cdot i^2 \right) \quad (3.4)$$

### 3.4 GLRLM\_SRLGE, GLRLM\_SRHGE

**Short-Run Low Gray-level Emphasis** or **Short-Run High Gray-level Emphasis** is the distribution of the short homogeneous runs with low or high grey-levels.

$$GLRLM\_SRLGE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j \frac{GLRLM(i,j)}{i^2 \cdot j^2} \right) \quad (3.5)$$

$$GLRLM\_SRHGE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j \frac{GLRLM(i,j) \cdot i^2}{j^2} \right) \quad (3.6)$$

### 3.5 GLRLM\_LRLGE, GLRLM\_LRHGE

**Long-Run Low Gray-level Emphasis** or **Long-Run High Gray-level Emphasis** is the distribution of the long homogeneous runs with low or high grey-levels.

$$GLRLM\_LRLGE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j \frac{GLRLM(i,j) \cdot j^2}{i^2} \right) \quad (3.7)$$

$$GLRLM\_LRHGE = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \sum_j GLRLM(i,j) \cdot i^2 \cdot j^2 \right) \quad (3.8)$$

GLRLM

### 3.6 GLRLM\_GLNUR, GLRLM\_RLNU

**Gray-Level Non-Uniformity for run** or **Run Length Non-Uniformity** is the non-uniformity of the grey-levels or the length of the homogeneous runs.

$$GLRLM\_GLNUR = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_i \left( \sum_j GLRLM(i,j) \right)^2 \right) \quad (3.9)$$

$$GLRLM\_RLNU = \text{Average over 13 (or 4) directions} \left( \frac{1}{H} \sum_j \left( \sum_i GLRLM(i,j) \right)^2 \right) \quad (3.10)$$

### 3.7 GLRLM\_RP

**Run Percentage** measures the homogeneity of the homogeneous runs.

$$GLRLM\_RP = \text{Average over 13 (or 4) directions} \left( \frac{H}{\sum_i \sum_j (j \cdot GLRLM(i,j))} \right) \quad (3.11)$$



**Chapter 4**  
**GLZLM definition**

## 4.1 GLZLM

The grey-level zone length matrix (GLZLM) [Thibault] provides information on the size of homogeneous zones for each grey-level in 3 dimensions (or 2D). From this matrix, 11 texture indices are computed. Element  $(i, j)$  of GLZLM corresponds to the number of homogeneous zones of  $j$  voxels with the intensity  $i$  in an image and is called  $GLZLM(i, j)$  thereafter.

## 4.2 GLZLM\_SZE, GLZLM\_LZE

**Short-Zone Emphasis** or **Long-Zone Emphasis** is the distribution of the short or the long homogeneous zones in an image.

$$GLZLM\_SZE = \frac{1}{H} \sum_i \sum_j \frac{GLZLM(i, j)}{j^2} \quad (4.1)$$

$$GLZLM\_LZE = \frac{1}{H} \sum_i \sum_j GLZLM(i, j) \cdot j^2 \quad (4.2)$$

where  $H$  corresponds to the number of homogeneous zones in the Volume of Interest.

## 4.3 GLZLM\_LGZE, GLZLM\_HGZE

**Low Gray-level Zone Emphasis** or **High Gray-level Zone Emphasis** is the distribution of the low or high grey-level zones.

$$GLZLM\_LGZE = \frac{1}{H} \sum_i \sum_j \frac{GLZLM(i, j)}{i^2} \quad (4.3)$$

$$GLZLM\_HGZE = \frac{1}{H} \sum_i \sum_j GLZLM(i, j) \cdot i^2 \quad (4.4)$$

## 4.4 GLZLM\_SZLGE, GLZLM\_SZHGE

**Short-Zone Low Gray-level Emphasis** or **Short-Zone High Gray-level Emphasis** is the distribution of the short homogeneous zones with low or high grey-levels.

$$GLZLM\_SZLGE = \frac{1}{H} \sum_i \sum_j \frac{GLZLM(i, j)}{i^2 \cdot j^2} \quad (4.5)$$

$$GLZLM\_SZHGE = \frac{1}{H} \sum_i \sum_j \frac{GLZLM(i, j) \cdot i^2}{j^2} \quad (4.6)$$



#### 4.5 GLZLM\_LZLGE, GLZLM\_LZHGE

**Long-Zone Low Gray-level Emphasis** or **Long-Zone High Gray-level Emphasis** is the distribution of the long homogeneous zones with low or high grey-levels.

$$GLZLM\_LZLGE = \frac{1}{H} \sum_i \sum_j \frac{GLZLM(i,j) \cdot j^2}{i^2} \quad (4.7)$$

$$GLZLM\_LZHGE = \frac{1}{H} \sum_i \sum_j GLZLM(i,j) \cdot i^2 \cdot j^2 \quad (4.8)$$

#### 4.6 GLZLM\_GLNUz, GLZLM\_ZLNU

**Gray-Level Non-Uniformity for zone** or **Zone Length Non-Uniformity** is the non-uniformity of the grey-levels or the length of the homogeneous zones.

$$GLZLM\_GLNUz = \frac{1}{H} \sum_i \left( \sum_j GLZLM(i,j) \right)^2 \quad (4.9)$$

$$GLZLM\_ZLNU = \frac{1}{H} \sum_j \left( \sum_i GLZLM(i,j) \right)^2 \quad (4.10)$$

#### 4.7 GLZLM\_ZP

**Zone Percentage** measures the homogeneity of the homogeneous zones.

$$GLZLM\_ZP = \frac{H}{\sum_i \sum_j (j \cdot GLZLM(i,j))} \quad (4.11)$$





**Part III**  
**First Order Features**

---



**Chapter 1**  
**Indices from Shape**

## 1.1 Sphericity

### 1.1 SHAPE\_Sphericity

is how spherical a Volume of Interest is. Sphericity is equal to 1 for a perfect sphere.

$$SHAPE\_Sphericity = \frac{\pi^{1/3} \cdot (6V)^{2/3}}{A} \quad (1.1)$$

where  $V$  and  $A$  correspond to the volume and the surface of the Volume Of Interest based on the Delaunay triangulation.

### 1.2 SHAPE\_Compacity

reflects how compact the Volume of Interest is.

$$SHAPE\_Compacity = \frac{A^{3/2}}{V} \quad (1.2)$$

where  $V$  and  $A$  correspond to the volume and the surface of the Volume Of Interest based on the Delaunay triangulation.

### 1.3 SHAPE\_Volume (mL and voxels)

is the Volume of Interest in mL and in voxels.

$$SHAPE\_Volume = \sum_i V_i \quad (1.3)$$

where  $V_i$  correspond to the volume of voxel  $i$  of the Volume Of Interest.

## Chapter 2

# Indices from Histogram

### 2.1 Histogram calculation

To build a histogram *HISTO*, it is necessary to determine a bin width ("bin" parameter). The indices derived from the histogram will depend on this bin width parameter.

This dependence, similar to that found in texture index calculations, is often overlooked in publications.

In LIFEx, with the absolute model the histogram is built a number of bins equal to that entered by the user in the "number of grey level" and "size of bin" fields of the resampling menu.

## 2.2 Skewness

In LIFEx, with the relative model the histogram is built only with "number of grey level" fields of the resampling menu that entered by the user and min and max are extracted values of each ROI.

### 2.2 HISTO\_Skewness

is the asymmetry of the grey-level distribution in the histogram.

$$HISTO\_Skewness = \frac{\frac{1}{E} \sum_i (HISTO(i) - \overline{HISTO})^3}{\left( \sqrt{\frac{1}{E} \sum_i (HISTO(i) - \overline{HISTO})^2} \right)^3} \quad (2.1)$$

where  $HISTO(i)$  corresponds to the number of voxels with intensity  $i$ ,  $E$  the total number of voxels in the Volume of Interest and  $\overline{HISTO}$  the average of grey-levels in the histogram.

### 2.3 HISTO\_Kurtosis

reflects the shape of the grey-level distribution (peaked or flat) relative to a normal distribution.

$$HISTO\_Kurtosis = \frac{\frac{1}{E} \sum_i (HISTO(i) - \overline{HISTO})^4}{\left( \frac{1}{E} \sum_i (HISTO(i) - \overline{HISTO})^2 \right)^2} \quad (2.2)$$

where  $HISTO(i)$  corresponds to the number of voxels with intensity  $i$ ,  $E$  the total number of voxels in the Volume of Interest and  $\overline{HISTO}$  the average of grey-levels in the histogram.

### 2.4 HISTO\_Entropy\_log10

reflects the randomness of the distribution.

$$HISTO\_Entropy_{\log_{10}} = - \sum_i p(i) \cdot \log_{10}(p(i) + \varepsilon) \quad (2.3)$$

where  $p(i)$  is the probability of occurrence of voxels with intensity  $i$  and  $\varepsilon = 2e-16$ .

Be aware of the logarithm used in the formula. We use the logarithm with base 10 in LIFEx but the logarithm base 2 is sometimes used in other software ; see `_log2` formula.

### 2.5 HISTO\_Entropy\_log2

reflects the randomness of the distribution.

$$HISTO\_Entropy_{\log_2} = - \sum_i p(i) \cdot \log_2(p(i) + \varepsilon) \quad (2.4)$$

where  $p(i)$  is the probability of occurrence of voxels with intensity  $i$  and  $\varepsilon = 2e-16$ .



## 2.6 HISTO\_Energy

reflects the uniformity of the distribution.

$$HISTO\_Energy = \sum_i p(i)^2 \quad (2.5)$$



## Chapter 3

# Conventional Indices

### 3.1 CONV\_SUVmin, CONV\_SUVmean, CONV\_SUVmax

**CONV\_SUVmin:** reflects the minimum Standardized Uptake Value (SUV) in the Volume of Interest.

$$CONV_{min} = \min_i SUV_i \quad (3.1)$$

**CONV\_SUVmean:** reflects the average SUV in the Volume of Interest.

$$CONV_{mean} = \frac{1}{N} \sum_i SUV_i \quad (3.2)$$

**CONV\_SUVmax:** reflects the maximum SUV in the Volume of Interest.

$$CONV_{max} = \max_i SUV_i \quad (3.3)$$

### 3.2 SUV<sub>peak</sub>

#### 3.2 CONV\_SUV<sub>peak</sub>

SUV<sub>peak</sub> reflects the mean SUV in a sphere with a volume of ~ 1 mL and located so that the average value in the VOI is maximum [WAHL 2009] or [VANDERHOEK 2012].

#### 3.3 CONV\_TLG (mL)

is the Total Lesion Glycolysis defined as the product of *SUV<sub>mean</sub>* by *Volume* in mL. [LARSON 1999]

$$CONV_{mean} = V \cdot \frac{1}{N} \sum_i SUV_i \quad (3.4)$$



**Part IV**  
**References**

---



# Chapter 1

## Useful references

### 1.1 Bibliography

Journal papers involving the use of LIFEx are listed in <http://www.lifexsoft.org/index.php/resources/publications-journal-papers>

#### 1.1.1 2018

F. Orlhac, S. Boughdad, C. Philippe, H. Stalla-Bourdillon, C. Nioche, L. Champion, M. Soussan, F. Frouin, V. Frouin, I. Buvat. A post-reconstruction harmonization method for multicenter radiomic studies in PET. J Nucl Med. 2018 doi: 10.2967/jnumed.117.199935. ([link](#))

## 1.1 Bibliography

### 1.1.2 2017

F. Orlhac, C. Nioche, M. Soussan, I. Buvat. Understanding changes in tumor textural indices in PET: a comparison between visual assessment and index values in simulated and patient data. *J Nucl Med*. 2017 Mar;58(3):387-392. ([link](#))

### 1.1.3 2016

F. Orlhac, B. Thézé, M. Soussan, R. Boisgard, I. Buvat. Multi-scale texture analysis: from 18F-FDG PET images to pathological slides. *J Nucl Med* 57: 1823-1828, 2016. ([link](#))

### 1.1.4 2015

F. Orlhac, M. Soussan, K. Chouahnia, E. Martinod, I. Buvat. 18F-FDG PET-derived textural indices reflect tissue-specific uptake pattern in non small cell lung cancer. *Plos One* 10(12):e0145063, 2015. ([link](#))

I. Buvat, F. Orlhac, M. Soussan. Tumor texture analysis in PET: where do we stand? *J Nucl Med* 56: 1642-1644, 2015. ([link](#))

### 1.1.5 2014

M. Soussan, F. Orlhac, M. Boubaya, L. Zelek, M. Ziol, V. Eder, I. Buvat. Relationship between tumor heterogeneity measured on FDG-PET/CT and pathological prognostic factors in invasive breast cancer. *Plos One* 9: e94017, 2014. ([link](#))

F. Orlhac, M. Soussan, JA. Maisonobe, CA. Garcia, B. Vanderlinden, I. Buvat. Tumor texture analysis in 18F-FDG-PET: relationships between texture parameters, histogram indices, SUVs, metabolic volumes and total lesion glycolysis. *J Nucl Med* 55: 414-422, 2014. ([link](#))

### 1.1.6 2012

Vanderhoek M, Perlman S, Jeraj R. Impact of the definition of peak standardized uptake value on quantification of treatment response. *J Nucl Med*. 2012;53:4-11. ([link](#))

### 1.1.7 2009

Thibault G, Fertil B, Navarro C, et al. Texture indexes and gray level size zone matrix. application to cell nuclei classification. *Pattern Recognition Inf Process*. 2009;140:145 ([link](#))

Wahl R, Jacene H, Kasamon Y and Lodge MA. From RECIST to PERCIST:evolving considerations for PET response criteria in solid tumors. *J Nucl Med*. 2009;50:122S-50S. ([link](#))



### 1.1.8 2004

Xu D-H, Kurani AS, Furst JD, Raicu DS. Run-length encoding for volumetric texture. Paper presented at: The 4th IASTED International Conference on Visualization, Imaging and Image Processing: VIP; September 6-8, 2004; Marbella, Spain ([link](#))

### 1.1.9 1999

Larson S, Erdi Y, Akhurst T et al. Tumor treatment response based on visual and quantitative changes in global tumor glycolysis using PET-FDG Imaging. The visual response score and the change in total lesion glycolysis. Clin Positron Imaging. 1999;42:159-171. ([link](#))

### 1.1.10 1989

Amadasun M, King R. Textural features corresponding to textural properties. IEEE Trans Syst Man Cybern.1989;19:1264-1274 ([link](#))

### 1.1.11 1973

Haralick RM, Shanmugam K, Dinstein I. Textural features for image classification. IEEE Trans Syst Man Cybern.1973;3:610-621. ([link](#))