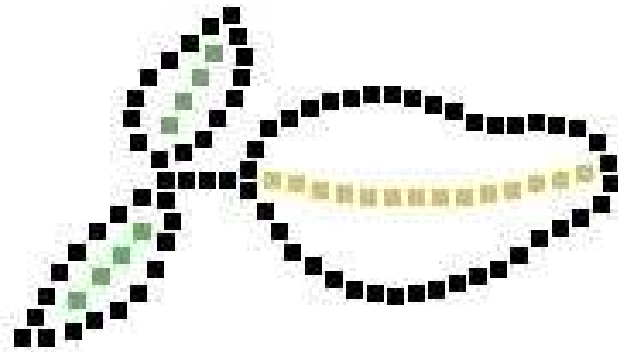


# Mango User Guide

APPLIED MATHEMATICS, ANU  
Holger AVERDUNK 2004,2005 v 1.1



**m a n g o**  
medial axis and  
network generation

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	The Software . . . . .	6
1.2	Data Formats . . . . .	6
1.3	Log Files . . . . .	7
<b>2</b>	<b>The GUI</b>	<b>7</b>
<b>3</b>	<b>Data Flow Chart</b>	<b>8</b>
<b>4</b>	<b>Top Section - Menus</b>	<b>10</b>
4.1	MPI . . . . .	10
<b>5</b>	<b>Top Section - controls</b>	<b>10</b>
5.1	Input Data . . . . .	12
5.1.1	NetCDF . . . . .	12
5.1.2	ASCII . . . . .	12
5.1.3	rawPGM . . . . .	12
5.1.4	binary int . . . . .	12
5.1.5	binary char . . . . .	12
5.1.6	binary uchar . . . . .	12
5.1.7	binary short . . . . .	12
5.1.8	binary ushort . . . . .	12
<b>6</b>	<b>Tomographic Data</b>	<b>12</b>
6.1	Input Data File . . . . .	12
6.2	Generate Geometric Fiducial . . . . .	13
6.3	Analysis . . . . .	13
6.3.1	Intensity Histogram . . . . .	13
6.3.2	Intensity Cut Line . . . . .	13
6.3.3	Intensity Cut Plane - 2D Image generation . . . . .	14
6.3.4	Grid of Images - 2D Image generation . . . . .	14
6.3.5	Intensity Gradient Plot . . . . .	14
6.3.6	Cylindrical Intensity Profile . . . . .	15
6.3.7	Threshold Quality . . . . .	15
6.3.8	Compare Tom Data . . . . .	15
6.4	Preprocessing - Filters and Convolvers . . . . .	15
6.4.1	Mean . . . . .	15
6.4.2	Median . . . . .	16
6.4.3	Conservative Median . . . . .	16
6.4.4	Conservative Smoothing . . . . .	16

6.4.5	Gaussian . . . . .	16
6.4.6	Selective Gaussian . . . . .	17
6.4.7	Convolution FFT . . . . .	17
6.4.8	Sobel - Gradient Edge Detection . . . . .	17
6.4.9	Kuwahara - Edge Enhancing . . . . .	18
6.4.10	Anisotropic Diffusion . . . . .	18
6.4.11	Unsharp Mask . . . . .	18
6.5	Preprocessing - Data . . . . .	19
6.5.1	Rescale Data . . . . .	19
6.5.2	Beam Hardening Correction - BHC . . . . .	19
6.5.3	Subsetting . . . . .	19
6.5.4	Subset OOC - Out of core . . . . .	20
6.5.5	Format Conversion . . . . .	20
6.5.6	Format Conversion OOC . . . . .	20
6.5.7	Diff Tom Data Files . . . . .	20
6.5.8	Edge Detection . . . . .	20
6.6	Segmentation . . . . .	21
6.6.1	The Problem . . . . .	21
6.6.2	Simple Segmentation . . . . .	22
6.6.3	Level Set Segmentation . . . . .	22
6.6.4	Metropolis Segmentation . . . . .	23
6.6.5	Watershed Segmentation . . . . .	23
6.6.6	Converging Active Contours Segmentation . . . . .	24
6.6.7	3 Phase Converging Active Contours Segmentation . . . . .	24
<b>7</b>	<b>Segmented Data</b>	<b>27</b>
7.1	Analysis . . . . .	27
7.1.1	Intensity Cut Plane - 2D Picture generation . . . . .	27
7.1.2	Grid of Images . . . . .	27
7.1.3	Calculate Porosity . . . . .	27
7.1.4	MC Isosurface . . . . .	27
7.1.5	Minkowski Analysis . . . . .	28
7.1.6	Resultant Distribution . . . . .	28
7.1.7	Compare Seg Data Files . . . . .	28
7.1.8	Cluster Tracking . . . . .	28
7.1.9	Analyse Crevice Pore Space . . . . .	28
7.2	Processing . . . . .	29
7.2.1	Compare Seg Data Files . . . . .	29
7.2.2	Subset . . . . .	29
7.2.3	Format Conversion . . . . .	29
7.2.4	Add Flowtube . . . . .	29

7.2.5	Add Boundary Layer . . . . .	30
7.2.6	Spanning Clusters . . . . .	30
7.2.7	Label Clusters . . . . .	30
7.2.8	Isolated Clusters . . . . .	30
7.2.9	Erosion Dilation . . . . .	31
7.2.10	Fiducial . . . . .	31
7.2.11	Write Crevice Pore Space . . . . .	32
7.2.12	Euclidean Distance . . . . .	32
<b>8</b>	<b>Label Data</b>	<b>32</b>
8.1	Compare Label Data Files . . . . .	32
8.2	Subset . . . . .	33
8.3	Format Conversion . . . . .	33
8.4	Intensity Cut Plane . . . . .	33
8.5	Grid of Images . . . . .	33
8.6	Sort Cluster Lables . . . . .	33
8.7	Cluster Size Distribution . . . . .	33
8.8	Smooth Label Data . . . . .	33
8.9	Triple Label Contact . . . . .	33
8.10	Network from Labels . . . . .	34
<b>9</b>	<b>Distance Map Data</b>	<b>34</b>
9.0.1	Subset . . . . .	34
9.0.2	Format Conversion . . . . .	34
9.0.3	Medial Axis . . . . .	34
9.1	Analysis . . . . .	35
9.1.1	Euclidean Voronoi . . . . .	35
9.1.2	Compare Distance Map Data Files . . . . .	35
9.1.3	Intensity Cut Plane - 2D Picture generation . . . . .	35
9.1.4	Grid of Images . . . . .	35
9.2	Processing . . . . .	36
9.2.1	Distance Map Watershed . . . . .	36
9.2.2	Maximal Covering Spheres . . . . .	36
9.2.3	Maximal Spheres . . . . .	36
<b>10</b>	<b>Fiducial Data</b>	<b>37</b>
10.1	Shrink Fiducial . . . . .	37
<b>11</b>	<b>Medial Axis Data</b>	<b>37</b>
11.1	Analysis . . . . .	37
11.1.1	Compare Medial Axis Data Files . . . . .	37

11.1.2	Measure Medial Axis Quality . . . . .	37
11.1.3	Intensity Cut Plane - 2D Picture generation . . . . .	38
11.1.4	Grid of Images . . . . .	38
11.1.5	Medial Axis Percolation . . . . .	38
11.1.6	Medial Axis Neighbour - Anisotropy Measurement . . . . .	39
11.2	Processing . . . . .	40
11.2.1	Format Conversion . . . . .	40
11.2.2	Subset . . . . .	40
11.2.3	Network Construction . . . . .	40
<b>12</b>	<b>Pore Throat Network Data</b>	<b>40</b>
12.1	Processing . . . . .	40
12.1.1	Convert File Format . . . . .	40
12.1.2	Compare Networks . . . . .	40
12.1.3	Connection Matrix . . . . .	40
<b>13</b>	<b>MPE Timings</b>	<b>41</b>
<b>14</b>	<b>Algorithms and Optimisations</b>	<b>41</b>
<b>15</b>	<b>From Tomogram to Network - A short Tutorial</b>	<b>41</b>
<b>16</b>	<b>Conclusion</b>	<b>41</b>
<b>17</b>	<b>Appendix</b>	<b>41</b>
17.1	Compilation . . . . .	41

# 1 Introduction

Mango is a software tool for parallel Segmentation and Network Generation and the pre and post-processing and analysis of associated data. The initial input for Mango is Tomographic Data of porous, or multi-phase objects. Mango has been build to process tomographic data from x-ray computerised tomography, but any voxelated data can be used as input. Processing of the Tomographic Data leads to Segmented, Distance Map, Fiducial, Medial Axis and Network Data. This User Guide adheres strongly to the logical processing layout found in the Mango GUI (General User Interface). Quotation marks are used to indicate parameters and sections which correspond to methods (functions) of the software. Other descriptors for low phase include void, pore and 0, and for high phase, solid, grain and 1 are used interchangeably.

## 1.1 The Software

The Mango Software is implemented in C++ and is controlled by a GUI written in Python. Mango can be compiled on Unix and Linux systems. Most importantly, it can be run in parallel, allowing for the processing of up to 2048 cube voxel data sets. Parallel processing is implemented using the MPI (Message Passing Interface see <http://www-unix.mcs.anl.gov/mpi/> and <http://www.lam-mpi.org/>) Standard. Furthermore, use of the NetCDF library is made to deal with large data sets and a FFT library is used (also see <http://www.fftw.org/>). Boost and jpeg libraries are also used. Compilation therefore requires an appropriate environment. Currently, the Mango core source code amounts to more than 100 000 lines of code.

Developers and Contributors:

Dr. Adrian Sheppard

Dr. Rob Sok

Holger Averdunk

Boris Breidenbach

Dr. Gerd Schroeder

## 1.2 Data Formats

Tomographic data must be in integer values ranging from 0 to 65536 where higher values indicate higher density and vice versa. Segmented data consists of 0's and 1's.

Distance Map data: floating point numbers;  
Medial axis data: floating point numbers (for distance) and row index for voxel location;  
Network data:

Input Data formats are described in the following sections.

Metadata: Non-ascii data files contain metadata which contains information about the dataset size, the original data collection and reconstruction and the history of processing that has been carried out on the data set (eg. filtering, subsetting, etc.). NetCDF files can be converted to ascii with the 'ncdump' command. Ascii files can be converted to netCDF files with 'ncgen'.

### 1.3 Log Files

For each processor that is used during a run, a log file, with the number of the corresponding processor appended to 'log\_P', is written to the working directory. Processor 0 is the master processor and the log\_P0 file contains collated information from all processors. The 'verbosity\_level' (set in the GUI top section) affects how much information is written to those log files.

## 2 The GUI

The GUI is divided into two main areas: The function selection area is on the left (starting with 'Top Section'). To the right, the control parameters of the selected function are displayed. Subsection menus can be minimised and maximised by double-clicking the left mouse button. Options are selected by right-clicking once and then the selected option is displayed in red. To ensure the corresponding parameter window is displayed and highlighted on the right, single click the option again with the left mouse button. The required parameters can then be entered and toggles can be switched. As the GUI just acts as a way to write an input file for Mango (named 'parms.in'), each time parameters have changed and a new processing run is to be started, ensure that the 'Save' button has been pressed. The bottom message window should then display 'Saved parms.in'. The 'Run' button can currently be ignored. The 'File' field at the top (above the 'Save' button) has three menus: The 'Save' menu has the same effect as pressing the 'Save' button. 'Change Working Directory...' opens a dialog box to change the working directory. 'Exit' terminates the GUI. The 'help' buttons may provide helpful information.



Figure 1: The Graphical User Interface.

### 3 Data Flow Chart

Converting tomographic data into a realistic network requires many steps. Unfortunately, complete automation of this process yielding a good end result cannot be achieved. Furthermore, although some functions can be chained (ie. the output of one function directly serves as input to another), results have to be examined and further runs with different parameters are often necessary to achieve good results. The following flow chart gives an overview of what needs to be done. The tutorial at the end of this user guide explains the issues in more detail.



DATA PROCESSING

ANALYSIS

TOMOGRAPHIC DATA

-> does the data need subsetting or a fiducial ?  
Intensity Cut Plane, Intensity Histogram,  
Intensity Cut Line, Cylindrical Intensity Profile,  
Intensity Gradient Plot; Visualise whole or part  
(subset/strided subset) of the data set;  
(get a feel and look of the data set -  
quality, size, shape, phases, etc.)

|  
|  
|  
|  
|  
|  
|  
V

filter: Anisotropic Diffusion -> Intensity Cut Plane, Intensity Histogram;  
Unsharp Mask -> check effect of filter;

|  
|  
V

segment: choose segmentation method based on data

|  
|  
V

SEGMENTED DATA

-> Intensity Cut Plane, Subset:  
Is the segmentation acceptable ?  
does the set need Isolated Cluster Removal ?

create  
distance data:

|  
|  
|  
|  
|  
V

DISTANCE MAP DATA

create  
medial axis:

|  
|  
|  
|  
|  
V

MEDIAL AXIS DATA

-> visualise - is the data OK ?

create  
network:

|  
|  
V

NETWORK

Filtering the data first with Anisotropic Diffusion (usually 20 iterations) followed by Unsharp Mask is advised.

## 4 Top Section - Menus

Input Data File: This menu needs to be used only if the input data is in ascii format. Enter the exact x, y and z dimensions of the input data set.

Output Data File:

'compression': choose method of compression

'num\_mbytes\_per\_file': This parameter determines the maximum file size in Megabytes (larger data sets are written as multiple files).

### 4.1 MPI

If Mango is to run in multi-processor mode, the MPI parameters need to be set: How many processors are needed depends on what is to be done and the particular hardware configuration that the job is running on. Note that the total number of processors used is determined by the multiplication of the numbers entered for num\_processors x, y and z. In order to minimise parallel processing overhead, the sub-domains created by  $X * Y * Z$  (eg. 8 sub-domains if  $x,y,z = 2$ ) should be as close to a cube as possible such that the communication surfaces (sub-domain boundaries) are as small as possible. For example,  $x=2, y=2, z=2$  (8 processors and sub-domains, communication area = 12) is better than  $x=4, y=2, z=1$  (8 processors and sub-domains, communication area = 16).

'num\_bytes\_in\_chunk': The file reading utility reads a part of the data file, then distributes it to the processors, then reads another part, distributes it, and so on. The chunk size must be small enough to fit in the memory of a single processor but otherwise needs to be reasonably large for good performance. Any number greater than 1MB should be sufficient.

## 5 Top Section - controls

'verbosity\_level': Allows for the control of how much processing information is written to stdout (the log files) during the execution of the program.

'phase to check': Sets the phase that is analysed by the 'Euclidean Distance' and 'Medial Axis' functions. In addition, the checked phase is 26-connected while the background phase is 6-connected in Isolated Cluster analysis.

'phase to check' affects:

CrevicePoreSpace

ErosionDilation

EuclideanDistance

IsolatedClusters

LabelClusters

MajorityFilter

MeasureMedAxQuality

MedialAxis

MinkowskiAnalysis

SpanningClusters

'use\_fiducial' and 'read\_fiducial\_from\_file':

In Mango, a fiducial means a surface which completely encloses a volume. If processing is to occur only in the volume enclosed by a fiducial, then the appropriate toggles must be set in the Top Section. There are two types of fiducial data: If a dataset is processed with 'Beam Hardening Correction', data outside the specified radius is set to a flag value indicating outside fiducial. The second type of fiducial data is a file which is in fact segmented data which serves as a mask for a corresponding dataset. Fiducial files can be generated geometrically ('Generate Geometric Fiducial', in the tomographic menu) or a form-fitting fiducial can be generated from 'Fiducial' in the segmented data menu. Therefore, if a file has been processed with 'Beam Hardening Correction', 'use\_fiducial' should be set to true. If a file fiducial is to be used, both 'use\_fiducial' and 'read\_fiducial\_from\_file' must be set to true.

'data\_directory': Press the 'Browse' button to open a choose-directory dialog box which allows setting the data directory (the directory from which the data is read). By default, the data directory is the same as the working directory.

'file\_name\_suffix': Sets the string which together with the type of expected data input is used to construct the file name that is searched for in the data directory. For example, suppose a dataset has been processed with 'Beam Hardening Correction' and one wishes to load the segmented file thereof: the file name suffix can be set to BHC and the program will expect a file called segmentedBHC.nc.

## 5.1 Input Data

### 5.1.1 NetCDF

The 'NetCDF (network Common Data Form) is an interface for array-oriented data access and a library that provides an implementation of the interface.' See the NetCDF website at: <http://my.unidata.ucar.edu/content/software/netcdf/index.html> or google NetCDF. Files in this format have '.nc' as suffix.  
- parameters required in the header -

### 5.1.2 ASCII

If data is read into Mango in ASCII format, the dimensions of the dataset must be entered into the GUI 'Input Data File' section right below 'Top Section'. The file must be terminated by a newline after each row of dimension x and every number must be separated by a whitespace. X \* Y \* Z must equal the number of numbers in the input file. Files in this format have '.ai' as suffix.

### 5.1.3 rawPGM

rawPGM

### 5.1.4 binary int

### 5.1.5 binary char

### 5.1.6 binary uchar

Files in unsigned character format have '.ubc' as suffix.

### 5.1.7 binary short

### 5.1.8 binary ushort

Files in unsigned short format have '.bs' as suffix.

## 6 Tomographic Data

### 6.1 Input Data File

This menu allows setting the expected format of the input file and the base name of the file ('file\_name\_base'). The expected file name is the concatena-

tion of 'file\_name\_base' + 'file\_name\_suffix' + '.' + format type suffix (eg. nc for netCDF).

## 6.2 Generate Geometric Fiducial

IN: tomographic data

OUT: a fiducial file (of type segmented data)

This function generates a fiducial file which encloses some volume of the tomographic dataset. The fiducial is either a cylinder running along any of the three axis with any radius that fits within the dataset (also taking account of the mid point chosen) or a rectangle where offset and size can be set.

## 6.3 Analysis

### 6.3.1 Intensity Histogram

IN: tomographic data

OUT: intensity histogram data written to log\_P0

The Intensity Histogram of a tomographic data set is the crucial preanalysis tool for segmentation. In order to benefit most from this tool, it is helpful to only analyse the volumes of interest (this can be achieved through appropriate subsetting or the use of a fiducial). Given a normal value range, a default bin size of 32 results in a smooth histogram, but other bin sizes can be chosen. If 'per\_process' is true, each processor writes the intensity histogram of its allocated data to its log file. If 'normalise' is set to true, the data is normalised such that the sum of the histogram is 1.

### 6.3.2 Intensity Cut Line

IN: tomographic data

OUT: intensity cut line data written to log\_P0

This function is useful for determining sub-set sizes and getting a general idea of the dataset, like heterogeneity for example. If the 'averaging radius' is 1.0, the analysis axis is one voxel in diameter. An 'averaging out' of local heterogeneity can be achieved by increasing this radius. Set the 'direction' to the axis of interest and fill in the desired coordinates from which the cut line originates. The 'coordinates' toggle is only meaningful for a previously

subsampled data set. In this case, if 'coordinates' is set to 'global', the reference frame is that of the original file.

### 6.3.3 Intensity Cut Plane - 2D Image generation

IN: tomographic data

OUT: jpeg file written to the working directory (Name: 'slice' + axis (X, Y or Z) + '\_' + 'file\_name\_base' + '.jpg')

Here, a two-dimensional slice along one of three axis of the dataset is converted into a gray level jpeg file which is written to the working directory. Choose the desired axis using the toggle switches and enter a valid 'position' in the dataset. If 'auto scale' is set to false, enter appropriate 'manual minimum value' and 'manual maximum value' parameters from which the scale intensity is calculated.

### 6.3.4 Grid of Images - 2D Image generation

IN: tomographic data

OUT: jpeg files written to the working directory (Name: 'slice' + axis (X, Y or Z) + '\_' + 'file\_name\_base' + '.jpg')

standard for all gridders !

If one is interested in more than one plane, 'Grid of Images' can be used: Planes can be created on the x, y, z or all three axis (direction toggle). Enter the desired 'number of planes' and unless 'auto choose planes' is true enter a valid 'middle plane' and 'plane spacing'. The files are written to the working directory.

### 6.3.5 Intensity Gradient Plot

IN: tomographic data

OUT: jpeg file in working directory

The Intensity Gradient Plot is a scatter plot in which each voxel contributes a point where its intensity provides the x value and its gradient the y value. The brighter a certain area is, the more voxels map to it. A file prepended with 'IntensityGradientPlot' to the suffix and with a '.jpg' extension is written to the working directory. This plot can help to determine the 'gradient threshold' for 3 phase CAC segmentation. See section '3 Phase Converging Active Contours Segmentation' below for an example of the output and how to estimate the 'gradient threshold' from the plot.

### 6.3.6 Cylindrical Intensity Profile

IN: tomographic data

OUT: data in log\_P0 file

Given a cylindrical data set, like a drill-core plug for example, it may be useful to determine an intensity histogram of increasing radii from a center to the outside. Should this histogram indicate unexpected density increases from the inside to the outside, the input xct-data may suffer from Beam Hardening Artifacts. If this is the case, 'Beam Hardening Correction' may minimise these artifacts.

### 6.3.7 Threshold Quality

IN: tomographic data

OUT: data in log\_P0 file

Study in progress -

### 6.3.8 Compare Tom Data

IN: tomographic data - 2 data files of the same domain.

OUT: data in log\_P0 file

'Compare Tom Data' takes 2 tomographic data files (which have to be of the same domain) as input and reports all voxels that differ. This function can be used, for example, to compare pre and post-filtered data.

## 6.4 Preprocessing - Filters and Convolvers

With the exception of Sobel, all filters use the parameter 'filter range' which defines the surrounding spherical volume of a center voxel. For example, for range = 1, 6 neighbours, range = 1.5, 18 neighbours, range = 1.8, 26 neighbours, range = 2, 32 neighbours. The tomographic output files from Arithmetic Filters have the letters 'AF' added to the file name suffix of the input files.

### 6.4.1 Mean

IN: tomographic data

OUT: filtered tomographic data

The Mean filter replaces the value of each voxel with the mean of the surrounding voxels where 'filter range' determines the surrounding spherical

volume. The parameters 'filter sigma' and 'max delta' do not apply.

### 6.4.2 Median

IN: tomographic data

OUT: filtered tomographic data

The Median filter replaces the value of each voxel with the median of the surrounding voxels where 'filter range' determines the surrounding spherical volume. The parameters 'filter sigma' and 'max delta' do not apply.

### 6.4.3 Conservative Median

IN: tomographic data

OUT: filtered tomographic data

The Conservative Median filter replaces the value of each voxel with the median of the surrounding voxels (where 'filter range' determines the surrounding spherical volume) if its value is outside the minimum and maximum values of the surrounding. The parameters 'filter sigma' and 'max delta' do not apply.

### 6.4.4 Conservative Smoothing

IN: tomographic data

OUT: filtered tomographic data

The Conservative Smoothing filter replaces the value of each voxel with the maximum (minimum) of the surrounding voxels if it is larger (smaller) than the maximum (minimum) where 'filter range' determines the surrounding spherical volume. The parameters 'filter sigma' and 'max delta' do not apply.

### 6.4.5 Gaussian

IN: tomographic data

OUT: filtered tomographic data

gaussianPassFilter.h says it is not working - but gaussian does something (ie. different histogram) ???distance map data (optional: fiducial); uses 'filter sigma' as well - aps or rms to complete -



### 6.4.6 Selective Gaussian

IN: tomographic data

OUT: filtered tomographic data

does ? uses 'filter sigma' and max delta'; - aps or rms to complete -

### 6.4.7 Convolution FFT

IN: tomographic data

OUT: filtered tomographic data

does ? - aps or rms to complete -

### 6.4.8 Sobel - Gradient Edge Detection

IN: tomographic data

OUT: filtered tomographic data

The Sobel gradient edge detector (not really a filter) convolutes the dataset with the following kernel over the 26 neighbourhood of the central voxel:

x kernel:

[-1 -3 -1]	[0 0 0]	[1 3 1]
[-3 -6 -3]	[0 0 0]	[3 6 3]
[-1 -3 -1]	[0 0 0]	[1 3 1]
x-1	x	x+1

y kernel:

[ 1 3 1]	[ 3 6 3]	[ 1 3 1]
[ 0 0 0]	[ 0 0 0]	[ 0 0 0]
[-1 -3 -1]	[-3 -6 -3]	[-1 -3 -1]
y-1	y	y+1

z kernel:

[-1 0 1]	[-3 0 3]	[-1 0 1]
[-3 0 3]	[-6 0 6]	[-3 0 3]
[-1 0 1]	[-3 0 3]	[-1 0 1]
z-1	z	z+1

Visually, the resulting output corresponds to the outlines of objects. Usually, the Sobel output would not be used as input for a segmentation. It is used in conjunction with the original tomographic data in CAC (perfect) and Watershed Segmentation. No parameters taken.

### 6.4.9 Kuwahara - Edge Enhancing

IN: tomographic data

OUT: filtered tomographic data

This filter is an edge-preserving/enhancing filter. Consider the following 2D labelling of a 5 \* 5 matrix: ( a a ab b b ) ( a a ab b b ) (ac ac abcd bd bd) ( c c cd d d ) ( c c cd d d ) In each of the four regions (a, b, c, d), the mean brightness and the variance are calculated. The output value of the center pixel (overlapping all four regions - abcd) in the window is the mean value of the region that has the smallest variance. The parameter `filter_range` defines the size of the region used to determine the new value of a voxel. No other parameters are taken. The performance of the filter for `filter_range` greater than 3 degrades quickly and is not advised. In most cases, Anisotropic Diffusion will yield much better results than Kuwahara. For 3D, the region is divided into 8 blocks.

### 6.4.10 Anisotropic Diffusion

IN: tomographic data

OUT: filtered tomographic data

Anisotropic Diffusion is an iterative edge enhancing and smoothing filter. It is an implementation taken mostly from: Frangakis and Hegerl, "Noise reduction in electron tomography", J Struct Bio 135, 239-250 (2001). In practice it has been found to perform very well, especially if followed by 'Unsharp Mask'. Its only drawback is that it is computationally expensive: Usually 20 iterations are sufficient, but for noisy data sets 40 iterations can lead to better improvements. Set the parameter 'number\_of\_iterations' accordingly. 'time\_step': - ?

'contrast': - ?

The tomographic output files from 'Anisotropic Diffusion' have the letters 'AD' added to the file name suffix of the input files.

### 6.4.11 Unsharp Mask

IN: tomographic data

OUT: filtered tomographic data

Although there seems to be no theoretical foundation for the 'Unsharp Mask', its results are visually very impressive. The filter works in three steps:

- 1) Blur the image (with a certain radius (Gaussian))
- 2) subtract the blurred image from the original which results in the mask

3) add the mask to the original (controlled by the 'strength' parameter)  
This filter requires three parameters and the defaults of radius = 2.5, cut-off = 8.0 and strength = 2.0 are usually suitable. Too much sharpening can cause a 'halo effect' where darker objects are surrounded by spuriously bright border regions. The tomographic output files from 'Unsharp Mask' have the letters 'UM' added to the file name suffix of the input files.

## 6.5 Preprocessing - Data

### 6.5.1 Rescale Data

IN: tomographic data

OUT: rescaled tomographic data

Tomographic data can be compressed towards its mean if 'scale\_factor' is less than 1 and can be expanded if 'scale\_factor' is greater than 1. The centrum can also be set. Expanded data may lead to a better segmentation. If the data is very compressed and a small change in the segmentation threshold leads to large changes in porosity, expanding the data may prove helpful.

### 6.5.2 Beam Hardening Correction - BHC

IN: tomographic data

OUT: corrected tomographic data

BHC applies a gaussian smoothing kernel on specified regions of the data. A 'Cylindrical Intensity Profile' (see section above) should be plotted to determine if BHC is required. If so, a 'Radial Beam Hardening Correction' can be carried out on the appropriate axis and mid-point. Enter a 'maximum radius' to determine how far along the radius the correction is to be carried out. Data past this radius is set to (Fiducial-outside,right?). If the correction is not to start immediately from the center, set the 'correction start radius' accordingly. The tomographic output files from 'Beam Hardening Correction' have the letters 'BHC' added to the file name suffix of the input file.

### 6.5.3 Subsetting

IN: tomographic data

OUT: subsetted tomographic data

Cutting down of the original dataset to one which contains only the regions of interest has several advantages. A smaller dataset is processed faster and

outside regions do not clutter the intensity histogram. Small subsets can also be used to determine what filtering and segmentation parameters are best for a given dataset. Downscaling of a subset can be achieved by entering stride numbers greater than 1, which, as a consequence, reduces resolution. A new file with the letters 'SS' appended to the suffix is written to the data directory.

#### **6.5.4 Subset OOC - Out of core**

IN: tomographic data

OUT: subsetted tomographic data

OOO- 'Out of Core' Create a subset running on one processor only, without reading the complete dataset into memory.

#### **6.5.5 Format Conversion**

IN: tomographic data

OUT: tomographic data in format selected

This function can convert tomographic input into netcdf (.nc), binary unsigned short (.bs), binary unsigned character (.ubc) and ascii (.ai).

#### **6.5.6 Format Conversion OOO**

IN: tomographic data

OUT: tomographic data in format selected

OOO - do format conversion running on one processor only, without reading the complete dataset into memory.

#### **6.5.7 Diff Tom Data Files**

#### **6.5.8 Edge Detection**

IN: tomographic data

OUT: distance map data

ED

## 6.6 Segmentation

Segmentation output files have the word 'segmented' followed by the suffix of the input file as file name and the files are written to the data directory.

### 6.6.1 The Problem

There is no known general theory of segmentation. Put simply, segmentation can be defined as the process of changing a multi-phase object into one of low phase and high phase, 0 and 1. Today, there are a multitude of different segmentation methods which seek to eliminate the shortcomings of a simple thresholding approach (ie.  $x < \text{threshold}$ :  $x = 0$ ,  $x > \text{threshold}$ :  $x = 1$ ). So what are the difficulties in this seemingly trivial process? What is a good segmentation and what is a bad one? What is the best segmentation method for a given data set and purpose? The issues here are the original data that serves as input and the desired output and what is to be done with it. The quick, cliched answers are: The devil is in the details and different horses for different courses.

**Noise** Data collected from x-ray computerised tomography suffers from a number of sources of noise. These include beam hardening, ring artifacts, cosmic rays, and artifacts caused by 3D reconstruction methods and data collection instruments themselves. Furthermore, resolution and contrast are hardly ever ideal. Specific algorithms and methods are able to ameliorate these artifacts (eg. 'Beam Hardening Correction'). Probably the most important step after data collection and prior to segmentation is filtering. A combination of 'Anisotropic Diffusion' and 'Unsharp Mask' usually leads to good results. Visual inspection and validation of original and filtered data, possibly followed by improving filtering by trying different parameters is necessary. Post segmentation, isolated cluster removal (see section 'Isolated Clusters') can be used in some cases to remove objects judged spurious or unwanted.

**Resolution - the half voxel problem** Border areas between the low and high phase will have voxels with intermediate values. These voxels may not be correctly assigned to one phase or another. In essence, this is a resolution problem: if the resolution were to be increased in these border areas, phases could be assigned more accurately.

**Post Segmentation Analysis** Segmentation should be objective, yet what is to be done with the segmented data should be kept in mind during the segmentation. A small change in the segmentation threshold can lead to enormous effects and differences in post segmentation analysis (eg. surface area, connectedness of fine structures, etc.). Any prior knowledge of the actual object (composition, density, void volume, structure, etc.) can and should be used to confirm the segmentation.

**How to judge a segmentation** Looking at a number of 'Intensity Cut Planes' of tomographic and segmented data is a good first step to check the segmentation. It should be kept in mind that the human visual system is not able to distinguish between various gray-levels all that well. Next, a visualisation tool (eg. mayavi) which can display tomographic data as contour lines and segmented data as isosurfaces simultaneously, should be used to verify the results. Isosurfaces should lie along dense contour lines. Visualising the data in 3D is important as this is the true nature of the data.

SegQuality - study in progress

## 6.6.2 Simple Segmentation

IN: tomographic data (optional: fiducial);

OUT: segmented data;

pros: fast and simple, (given perfect data, this is the method of choice)

cons: does not deal well with noise;

Simple (threshold) segmentation requires 'lower threshold' and 'upper threshold' parameters. Every voxel with a value below and equal to the lower threshold and everything equal and above the upper threshold is set to 0 (Void) and the rest is set to 1 (Grain). Note, therefore, that unless a certain phase is to be 'segmented out' of the data (ie. an intermediate phase is the phase of interest), the upper threshold should be 65536, the highest input value possible. A careful choice of the lower threshold is crucial for a good simple segmentation. Filtering of the original data is advised.

Shortcomings of 'Simple' segmentation and how to improve the segmentation:

## 6.6.3 Level Set Segmentation

IN: tomographic data (optional: fiducial data file)

OUT: segmented data;

method may be discarded - 'Level Set' segmentation is an implementation of the level set formulation for image processing, from Sethian "Level Set Methods and Fast Marching Methods", p68ff and p219ff.

It is an iterative method...

#### **6.6.4 Metropolis Segmentation**

IN: tomographic data (optional: fiducial data file)

OUT: segmented data;

not working yet

#### **6.6.5 Watershed Segmentation**

IN: tomographic data (optional: fiducial data file)

OUT: segmented data;

Watershed segmentation is a region growing algorithm which developed from mathematical morphology. There are a number of ways to implement the basic principle, some of which lay claim to work efficiently in parallel. One way to imagine the region growing in 2 dimensions is as follows: Water starts rising from the lowest level (ie. the lowest value of the dataset). Whenever a new level (next higher value) is reached, a new region starts growing, unless this level has a neighbour from a previously labelled/flooded region. This process continues until all levels have been filled. The borders of the regions represent the watershed lines, hence the name. The various regions are then evaluated in some way (average value for example), and some are merged according to a criteria (eg. threshold) such that a segmentation is achieved. Performing this process on the original tomographic data does in fact not lead to a satisfying segmentation. Instead, the initial region growing is carried out on edge data (ie. Sobel) and then these regions are analysed and merged on the original data. For this reason, two input files are required for this method of segmentation. The file entered in the Top section must be the edge (Sobel processed) data and the second file entered in the Watershed Input data file parameter corresponds to the original data. A threshold value that gives desired results for a simple segmentation is a good value to choose for the 'region average threshold'. Watershed segmentation has been found to perform well on cortical bone (mostly solid with fine, branching structure). Currently, this method is not yet implemented in parallel.

### 6.6.6 Converging Active Contours Segmentation

IN: tomographic data (optional: fiducial data file)

OUT: segmented data;

The paper 'Techniques for image enhancement and segmentation of tomographic images of porous materials, Sheppard, Sok, Averdunk; Physica A, Volume 339, p 145- 151' describes this segmentation method in some detail. The method requires two threshold values ('lower threshold' and 'upper threshold' ) which bound the uncertain intensity region. Voxels with an intensity less than 'lower threshold' are immediately assigned void and voxel intensities above 'upper threshold' are set to high phase. The set phases represent seed regions which grow into the undecided phase until all voxels are assigned 0 or 1. The speed with which these regions grow, is calculated from gradient and intensity information of the immediate voxel neighbourhood. Where the growing regions meet, usually represents the void/solid boundaries of the object.

### 6.6.7 3 Phase Converging Active Contours Segmentation

IN: tomographic data (optional: fiducial data file)

OUT: segmented data (low and intermediate phase or low, intermediate and high phase);

Extracting an intermediate phase from a dataset by simple segmentation with two thresholds can be problematic: If high phase voxels directly border low phase areas, simple thresholding is likely to assign some of these border voxels to the intermediate phase (half voxel problem). This can result in segmentations which include unwanted high/low phase border voxels in the intermediate phase. The 3 Phase Converging Active Contours Segmentation method can achieve a much better result by firstly identifying voxels with high gradient values. Then, regions of pore, intermediate and solid phase, grow, depending on the gradient (the same method as for Converging Active Contours). High/low border voxels are more likely to be assigned either as low or high phase, as one of these phases grows into the border area, and not assigned falsely as intermediate. The method requires four intensity threshold values and one gradient threshold value. These parameters divide the input data initially into six groups: pore, pore or intermediate - to be determined, intermediate, intermediate or solid - to be determined, solid, and edge data (high gradient) - to be determined. Pore, intermediate and solid regions then grow until all voxels have been determined. It may be helpful to carry out a simple segmentation first to determine reasonable threshold intervals.



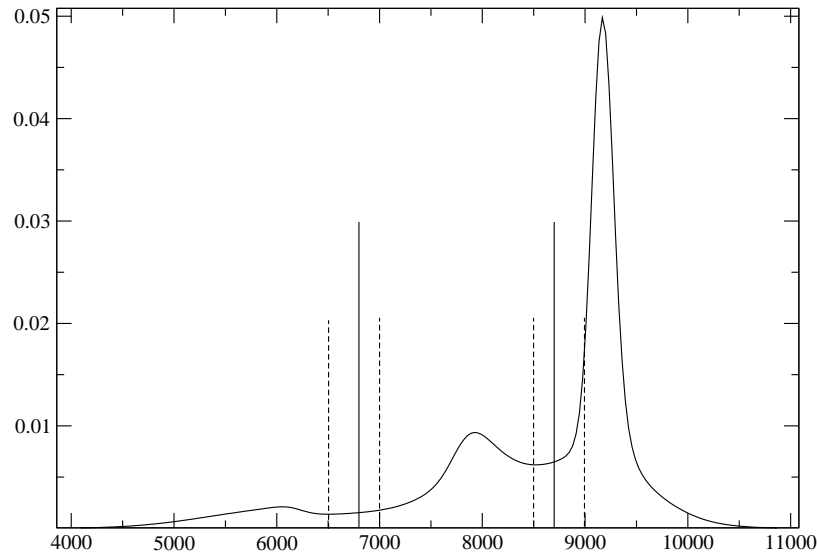


Figure 2: Three phase intensity histogram with thresholding values.

Figure x shows a fairly well separated three phase intensity histogram. The two solid vertical lines indicate good simple thresholding values and the dashed lines indicate appropriate intervals for 3 phase CAC segmentation. To determine the 'gradient threshold', an 'Intensity Gradient Plot' (see section x, above) can be used. The Intensity Gradient Plot is a scatter plot in which each voxel contributes a point where its intensity provides the x value and its gradient the y value. Figure x shows an Intensity Gradient Plot (black and white have been inverted) which has the thresholding parameters indicated as lines on it. 'Segmentation 3 Phase' currently only uses gradient information for calculating growth-speed. The 'use\_intensity' toggle should be left to false - work in progress.

The four vertical lines indicate the intensity thresholding (also compare to the previous intensity histogram). The solid horizontal line indicates the 'gradient threshold'. Notice the arches (at about the height of the dashed lines) connecting the three main/bulk voxel populations. These arches represent high gradient voxels. In general, appropriate 'gradient thresholds' can be found in the area just above the main/bulk voxels.

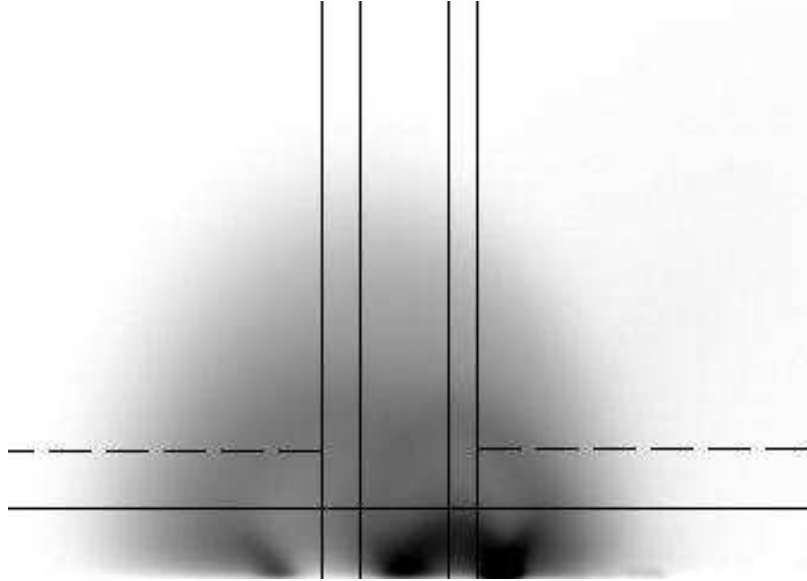


Figure 3: Three phase intensity histogram with thresholding values.

To find the 'gradient threshold' number, load the plot into an application which shows pixel row and column numbers (eg. Gimp). Let  $x$  be the  $y$ -coordinate representing the right height for the 'gradient threshold'. Then (given that the top left pixel is 0,0), 'gradient threshold' =  $(2048 - x) * 32$ . (The 65536 (x-axis - intensity) x 65536 (y-axis - max. gradient) pixel picture is scaled by a factor of 32).

Note that the rectangles bordered by the solid horizontal line and the bottom of the Figure, from left to right, correspond to pore, pore or intermediate - to be determined, intermediate, intermediate or solid - to be determined, and solid. Everything above that line is edge data which will be determined as the three regions grow. The dashed lines indicate the effect of setting the 'optional 2nd gradient threshold'. This parameter has to be above the 'gradient threshold' and is used as a refinement. Voxels with high gradient but low or high intensities can be directly assigned to pore or grain/solid. Since this affects the seeding of the growing regions, segmentation results differ if this parameter is set.

## 7 Segmented Data

### 7.1 Analysis

#### 7.1.1 Intensity Cut Plane - 2D Picture generation

IN: segmented data;

OUT: jpeg file in working directory

Here, a two-dimensional slice along one of three axis of the dataset is converted into a black and white jpeg file which is written to the working directory. Choose the desired axis using the toggle switches and enter a valid 'position' in the dataset.

#### 7.1.2 Grid of Images

IN: segmented data;

OUT: jpeg files in working directory

If one is interested in more than one plane, 'Grid of Images' can be used: Planes can be created on the x, y, z or all three axes (direction toggle). Enter the desired 'number of planes' and unless 'auto choose planes' is true enter a valid 'middle plane' and 'plane spacing'. The files are written to the working directory.

#### 7.1.3 Calculate Porosity

IN: segmented data;

OUT: Porosity data in log\_P0 file

This function calculates the number of void (0) and grain (1) voxels and their percentages. Note that unless an appropriate Fiducial (or subset) is used, regions outside the object proper will contribute to the fractions.

#### 7.1.4 MC Isosurface

IN: segmented data;

OUT: isosurface

Create an isosurface from a segmentation using the Marching Cubes algorithm.

### 7.1.5 Minkowski Analysis

IN: segmented data;

OUT: Minkowski functionals;

Calculate the Minkowski functionals of the segmented data.

### 7.1.6 Resultant Distribution

IN: segmented data and corresponding tomographic data;

OUT: histogram data

Resultant Distribution produces output (in the log\_P0 file) of two sets of numbers which can be plotted to show the distribution of original tomographic values and their corresponding segmented values. It is useful for segmentations other than simple segmentation (as the resulting distribution is trivial in the simple case). The 'histogram bin size' can be set as desired.

### 7.1.7 Compare Seg Data Files

IN: segmented data - 2 data sets of the same domain;

OUT: difference information

Compare Seg Data Files is a tool to compare two different segmentations. It takes two segmented input files (which have to have the same domains) and lists (in the log\_P0 file) all voxels and their coordinates which differ between the two segmentations. The second input file name base can be entered in the 'file name base' parameter.

output spread between log files ??

how about a difference seg file ??

### 7.1.8 Cluster Tracking

not working

### 7.1.9 Analyse Crevice Pore Space

not working

## 7.2 Processing

### 7.2.1 Compare Seg Data Files

IN: 2 segmented data sets of the same domain;

OUT: differences between input datasets;

### 7.2.2 Subset

IN: segmented data;

OUT: subsetted segmented data;

Strided or unstrided subsets of segmented data can be prepared by entering the appropriate offsets and sizes. Note that a strided subset does not necessarily accurately represent the original due to downscaling. A new file with the letters 'SS' appended to the suffix is written to the data directory.

### 7.2.3 Format Conversion

IN: segmented data;

OUT: segmented data in set format;

This function performs the conversion of segmented data to netcdf, ascii and binary bit pack (unsigned char) format with the respective file suffixes of '.nc', '.ai'(not yet !) and 'seg'.

### 7.2.4 Add Flowtube

IN: segmented data;

OUT: segmented data with flow tube;

This function adds a halo around a segmented dataset by extending the border in the x and y direction by one voxel and setting these areas to solid (1). These areas are also extended in the z direction (ie. the top and bottom ends) by an amount scaled on the original z size. This is a preparatory step of the segmented data in order to create medial axis and networks aimed at studying flow in the z axis. The letters 'FT' are added to the suffix of the output file which will be created if 'write data' is true.

### 7.2.5 Add Boundary Layer

IN: segmented data;

OUT: segmented data with extra added layer;

### 7.2.6 Spanning Clusters

IN: segmented data;

OUT: cluster information in log file

A spanning cluster is defined as an object of one phase which extends from the first to the last slice of the dataset in any axis. If 'do statistics' is true, the log file will list the number of clusters in the phase under investigation and in what size range they are. Setting 'convert clusters' to true removes all non-spanning clusters from the dataset which will be written to a file (if 'write file' is true) which has 'SC' added to its file name suffix. Spanning clusters can be checked for each axis individually or for all three axes at once by the 'direction' toggle.

### 7.2.7 Label Clusters

IN: segmented data;

OUT: label data (integer)

Label Clusters assigns a unique integer number to each isolated cluster and writes out a file named 'labels' concatenated with the input file suffix extension if 'write data' is true where each voxel is numbered according to which cluster it belongs to. If 'do statistics' is true, the log file will list the number of clusters in the phase under investigation and in what size range they are. Note that the same data set, when run with a different number of processors, will result in different labelling of the clusters.

### 7.2.8 Isolated Clusters

IN: segmented data;

OUT: segmented data;

The 'Isolated Clusters' function allows for the analysis and removal of isolated clusters of the void, solid or both phases. If 'do statistics' is set to true, the log file will list the number of clusters in both phases and in what size range they are ('convert\_clusters' must be set to true for both pore and grain). If 'write data' is true, a file with the suffix extended by 'IC' is written

to the data directory. Parameters need to be set for both 'Pore' and 'Grain' to control which clusters are removed. Either the single largest cluster is kept - 'keep one cluster only' set to true, or a minimum and maximum cluster size can be set and clusters are kept accordingly. Note that an isolated cluster removal is advised if the dataset is to be processed for a medial axis or fiducial.

### 7.2.9 Erosion Dilation

IN: segmented data;

OUT: segmented data;

This function can erode/dilate or dilate/erode objects in a dataset by a radius specified in the 'radius' parameter where the order is set with the 'order' parameter toggle. If 'write data' is true, a file with the letters 'Ed' appended to the original file name is created with the eroded/dilated data. If 'do statistics' is true, the number of voxels and their percentage that has changed is given in the log files. If the object is close to the dataset borders, 'halo size' should be set to a number larger than the 'radius' to avoid incorrect processing at the border.

### 7.2.10 Fiducial

IN: segmented data;

OUT: fiducial (segmented) data;

The Fiducial function under Segmented data attempts to make a surface around the segmented object(s). It does this through repeated dilation/erosion steps with increasing radii. If 'do statistics' is true, the log files will contain cluster statistics and the number of voxels that changed. If 'write data' is true, and a fiducial is created successfully (one object cluster and one background cluster remain after dilation/erosion steps) a file beginning with 'fiducial' and the input file suffix is written to the data directory.

do we really need 'break on failure' ?

Usually the 'max concavity radius' should be set to a greater number than 'max convexity radius'. The action of the 'max concavity radius' can be imagined as rolling a sphere around the outside of the object and converting to '1' any voxel that the sphere cannot reach. The 'max convexity radius' is equivalent to rolling a sphere around the inside of the object and converting to '0' any voxel that the sphere cannot reach. If the object is close to the dataset borders, 'halo size' should be set to a number larger than the largest radius to avoid incorrect processing at the border. The 'radius increment

factor' determines by how much the radii are increased after each unsuccessful step. The fiducial should be checked to see if it has the desired shape.

### **7.2.11 Write Crevice Pore Space**

IN: segmented data;  
not working;

### **7.2.12 Euclidean Distance**

IN: segmented data (optional: fiducial);  
OUT: distance map;

A 'Euclidean Distance' data file is needed as input for the creation of medial axis data. This function calculates for each voxel the euclidean distance to the nearest voxel of the other phase. Note that if the distance map data is to be used for the creation of medial axis data, it is important that cluster removal is carried out on the segmented data. If 'do statistics' is true, the distance distribution of the data is written to the log\_P0 file. Currently only the 'original' method should be used. Make sure 'write data' is set to true to obtain an output file. The toggle 'make maximal voxel file' should be left to false. The output file will have the name 'distance\_map' prepended to the suffix. The phase whose distance is to be calculated must be set accordingly in the 'Top Section' menu parameter 'phase\_to\_check'.

## **8 Label Data**

### **8.1 Compare Label Data Files**

IN: label data;  
OUT: label difference information;

This function takes two data sets of the same domain. Set the file name base of the second file in the file name base window. All corresponding voxels that have a different label in the two input files, and their values, are listed in the log\_P0 file. Note that the same data set, when run with a different number of processors with 'Label Clusters', will result in different labelling of the clusters.



## **8.2 Subset**

## **8.3 Format Conversion**

## **8.4 Intensity Cut Plane**

## **8.5 Grid of Images**

## **8.6 Sort Cluster Labels**

IN: label data;

OUT: sorted label data;

Functions that produce label data will create different labels depending on the processor layout (number of cpus and x,y,z layout). This function sorts the label numbers of the various clusters, thus allowing for a better comparison between labelled datasets.

## **8.7 Cluster Size Distribution**

IN: label data;

OUT: statistics of clusters

## **8.8 Smooth Label Data**

IN: label data;

OUT: smoothed label data;

Performs a euclidean voronoi tessellation on the label data followed by majority filtering.

## **8.9 Triple Label Contact**

IN: label data;

OUT: cluster contact statistics and label data containing interface surfaces;

Examines the contact areas of differently labelled clusters at the voxel level.

The three neighbours of a voxel are counted in the x,y and z planes.

## 8.10 Network from Labels

IN: label data;

OUT: pore throat network

# 9 Distance Map Data

## 9.0.1 Subset

## 9.0.2 Format Conversion

## 9.0.3 Medial Axis

IN: distance map data (optional: fiducial);

OUT: medial axis data; describe data format -

Top Section: phase to check: - must correspond with distance data

The 'Medial Axis' function requires 'Euclidean Distance' data as input. The axis is created through homotopic thinning of the distance map data where strict voxel ordering is provided by the distance map, and the bit-reversed value of the voxel's global coordinates. If 'do statistics' is true, the coordination number distribution and their spatial distributions are written to the log\_P0 file. If 'write data' is true, a file prepended with 'medial\_axis' to the suffix is written to the data directory. There are a number of ways in which the medial axis can be created at the boundary volumes. In particular, the parameters 'connect boundary', 'remove flow tube' and 'distance averaging radius' affect medial axis construction at the boundary. How those parameters are chosen depends on the purpose of the medial axis construction. If fluid flow studies are to be carried out along the z-axis of the object, a flow tube should be added prior to making the distance map data. If 'connect boundary' is false, medial axis along the inside-boundary may be created, but they will never connect (or extend/point through) the boundary. This option should only be used by the informed user.

The 'distance increment' parameter affects the ordering and the speed with which the medial axis is created. The smaller the number the more strict the ordering and the longer it will take and vice versa. If 'prune dead ends' is true, all dead ends of the medial axis are removed. This results in a less cluttered medial axis. Currently, a medial surface can not be made. The 'min axis cluster size' parameter sets the size below which medial axis clusters are removed. The 'remove flow tube' toggle should only be set to true if the input file contains a flow tube and the output file should be without it.

The 'distance averaging radius' affects how axis connect to the boundary. It is the radius of the region that is searched when deciding if a potential pinning point is a local maximum. A pinning point is a point that connects the medial axis to the boundary and it must be a local maximum in the distance map. The parameter is only effective if 'connect boundary' is true.

## **9.1 Analysis**

### **9.1.1 Euclidean Voronoi**

### **9.1.2 Compare Distance Map Data Files**

IN: distance map data (2 data sets)

OUT: difference information;

This function takes two data sets of the same domain. Set the file name base of the second file in the file name base window. All corresponding voxels that have a different distance data in the two input files, and their values, are listed in the log\_P0 file.

### **9.1.3 Intensity Cut Plane - 2D Picture generation**

IN: distance map data

OUT: jpeg file;

A gray-level jpeg picture of a plane (x, y, or z) of Distance Map Data can be generated by this function. Set the desired position of the cut plane in the 'position' window. Gray-levels are well spread if 'auto scale' is set to true. If 'auto\_scale' is false, the 'manual minimum value' and 'manual maximal value' are used to determine the gray-levels. The jpeg file is written to the working directory.

### **9.1.4 Grid of Images**

IN: distance map data

OUT: jpeg files;

This function does the same as 'Intensity Cut Plane - 2D Picture generation', but allows multiple planes to be produced in one run. The input windows for 'direction', the 'number of planes' and the 'middle plane' and 'plane spacing' control this additional functionality.

## 9.2 Processing

### 9.2.1 Distance Map Watershed

IN: distance map data (optional: fiducial);

OUT: label data; - labelsWS

This method carries out a Watershed algorithm, similar to the one described in 'Watershed Segmentation' above. In this case distance map data serves as the input, leading to label data output where each label number corresponds to a region found by the Watershed algorithm. This function can be used for finding the volume of separate but connected air bubbles in foam or the volume of individual beads in a bead pack for example. The 'do statistics' toggle currently has no effect. The 'seed checking radius' defines the radius of the volume that is searched to find the seed voxel with the highest distance value. It should be set to the radius of the smallest object that is to be labelled as a single region (yet it must be within the parameters of 1.0 to 50.0). If the 'seed checking radius' is too small, more smaller undesirable regions may be formed (in a sense, over-segmentation occurs). The 'threshold increment' value affects how quickly regions grow into areas with lower distance. Usually lower 'threshold increment' values will result in better results but at the cost of computation time.

### 9.2.2 Maximal Covering Spheres

IN: distance map data (optional: fiducial);

OUT: max\_included\_sphere data (float)

The function finds the largest inscribed (euclidean) distance sphere that covers each voxel in the object. This gives an indication of the wetting and non-wetting phases (capillary pressure/hydraulic radius). If 'do statistics' is set to true, a distance distribution of the radii is written to the log\_P0 file. The fast method is more efficient (as it avoids unnecessary calculations), yet gives the same results as 'brute\_force' and should therefore be used preferably. The output file has 'max\_included\_sphere' prepended to the original suffix.

### 9.2.3 Maximal Spheres

IN: distance map data (optional: fiducial);

OUT: medial axis data

Given the euclidean distance transform of an object, this function extracts

the voxels whose euclidean distance spheres are not contained within those of their neighbours. The output file has 'medial\_axis' prepended to its suffix. The number of maximal spheres is reported in the log file. 'do statistics' currently has no effect.

## 10 Fiducial Data

### 10.1 Shrink Fiducial

IN: fiducial (segmented) data

OUT: fiducial (segmented) data

Shrink Fiducial allows for the volume reduction of a fiducial (shrink-wrap). The 'shrink\_by' parameter determines the amount by which the original fiducial gets eroded. If 'write\_data' is true, a file with 'fiducialSH' prepended to the suffix will be written to the data directory. The number of voxels changed are reported in the log\_P0 file. 'do statistics' has currently no effect.

## 11 Medial Axis Data

### 11.1 Analysis

#### 11.1.1 Compare Medial Axis Data Files

IN: medial axis data (2 different files with the same domain)

OUT: location and value of all voxels that differ between the files, listed in the log\_P0 file;

To compare two medial axis data files of the same domain, enter the base name of the second file in the 'file\_name\_base' field and examine the comparison results in the log\_P0 file.

#### 11.1.2 Measure Medial Axis Quality

IN: medial axis data and corresponding segmented data

OUT: statistics in the log\_P0 file;

A measure of the quality of the medial axis is calculated by creating an object consisting of the union of spheres with the radius of each sphere given by the distance transform of the corresponding medial axis voxel at its center. This object is then compared to the segmented data from which the medial axis

was created. The number and percentage of correct, missing and spurious voxels listed, gives an indication of the quality of the medial axis.

### 11.1.3 Intensity Cut Plane - 2D Picture generation

IN: medial axis data

OUT: jpeg of a plane

A gray-level jpeg picture of a plane (x, y, or z) of medial axis data can be generated by this function. Set the desired position of the cut plane in the 'position' window. Gray-levels are well spread if 'auto scale' is set to true. If 'auto\_scale' is false, the 'manual minimum value' and 'manual maximal value' are used to determine the gray-levels. The jpeg file is written to the working directory.

### 11.1.4 Grid of Images

IN: medial axis data

OUT: jpegs of planes

This function does the same as 'Intensity Cut Plane - 2D Picture generation', but allows multiple planes to be produced in one run. The input windows for 'direction', the 'number of planes' and the 'middle plane' and 'plane spacing' control this additional functionality.

### 11.1.5 Medial Axis Percolation

IN: medial axis data

OUT: percolation threshold for X, Y, Z or all three written to log\_P0

This function calculates the percolation threshold using a bisection (binary search) method. It is the threshold at which the medial axis (or rather the geometry it represents) disconnects one face from another. Toggle the 'direction' switch to find the threshold for X, Y, Z or all three axes. The 'tolerance' parameter determines to what accuracy the threshold is calculated - the lower the value, the more accurate the threshold (at the cost of longer time taken to compute). Note that if the threshold is less than 1 (or the tolerance value is within the thresholds), then the axis is not spanning in that direction. Therefore, in usual circumstances, the tolerance value should be below 1. If a flow tube has been added to the input file, 'remove flow tube' should be set to true in the 'Medial Axis' creation (otherwise it will not span). During medial axis creation, the 'connect boundary' switch obviously also has a

strong bearing on percolation.

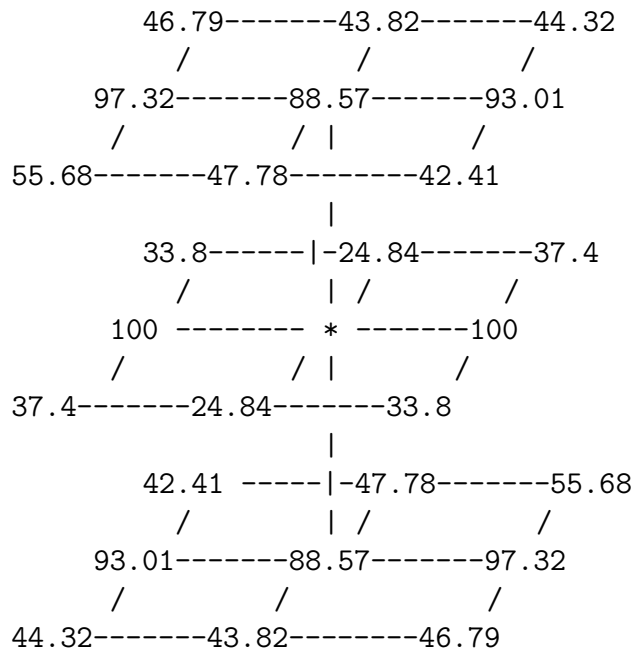
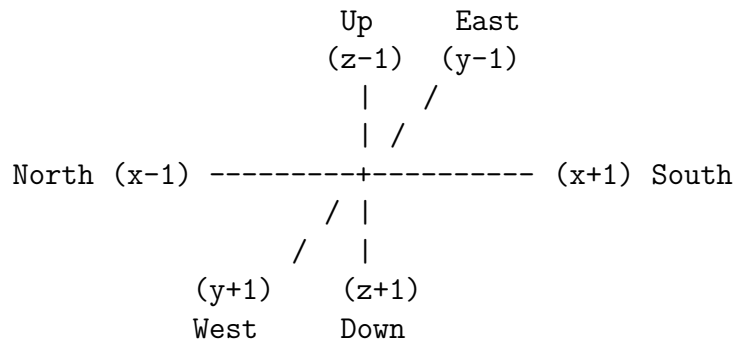
### 11.1.6 Medial Axis Neighbour - Anisotropy Measurement

IN: medial axis data

OUT: 26 neighbour percentage written to log\_P0 file.

This analysis tool adds any of the 26 neighbours of each voxel in the medial axis and then normalises the values such that the maximum value is 100%.

An indication of axis direction and an example output follows:



Here, the major direction of the medial axis is north/south (100%). That means, on average, more voxels have neighbours north and south than in the other directions. Therefore this represents a very simple and basic measure of anisotropy of the medial axis.

## **11.2 Processing**

### **11.2.1 Format Conversion**

IN: medial axis data in netCDF format

OUT: medial axis data in ascii format

With this function medial axis data in netCDF format can be converted to ascii where each line has the x, y and z coordinates of a voxel written in brackets followed by the distance value associated with that voxel. The output file has the extension 'ai'.

### **11.2.2 Subset**

### **11.2.3 Network Construction**

IN: medial axis data

OUT: pore throat network

Generate a pore throat network from medial axis data.

## **12 Pore Throat Network Data**

### **12.1 Processing**

#### **12.1.1 Convert File Format**

#### **12.1.2 Compare Networks**

#### **12.1.3 Connection Matrix**

IN: pore throat network;

OUT: network statistics;



## **13 MPE Timings**

Each log file ends with a MPE log entry (This applies only to Mango versions compiled with MPI). The master log (log\_P0) also has combined totals (ie. summary of all processors). Total time taken, maximum and average time taken and number of visits are shown. These statistics can not be controlled by the user, but are useful for finding an optimal number of processors for a particular job. For example, parallel processing overhead differences between two jobs on the same data with different numbers of processors can be calculated with these statistics.

## **14 Algorithms and Optimisations**

Anisotropic Diffusion, Unsharp Mask, segmentation methods, watershed methods, medial axis construction (homotopic thinning), time warp (medial axis, watershed), erosion/dilation, voronoi tessalation, fast marching algorithms, burn algorithms, marching cubes, etc.

## **15 From Tomogram to Network - A short Tutorial**

## **16 Conclusion**

## **17 Appendix**

### **17.1 Compilation**

Libraries needed:

Compiler directives: fast

mpi/non-mpi: