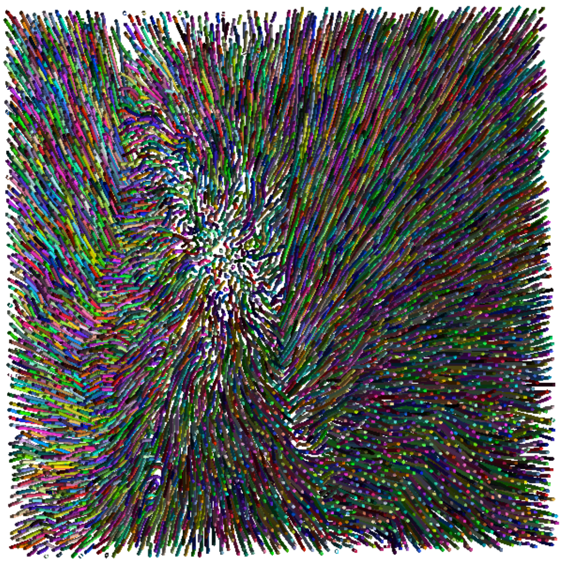
INSEGT FIBRE

A Powerful Segmentation Tool for Quantifying Fibre Architecture in Composites



ABOUT

Insegt Fibre is a user-friendly software tool to measure individual fibre centre lines from 3D/4D X-ray computed-tomography (CT) scans of fibre-rich composites.

The method requires minimal user input and, for its robustness to image quality, it is applicable to:

i) Scans acquired under reduced times at synchrotrons, as well as at lab sources.

ii) Scans acquired at limited spatial resolutions, so as to cover fields of view containing a representative number of fibres in a single scan (e.g. complete bundles).

iii) Time-lapse sequences of fast scans acquired while loading a material in-situ.

Insegt Fibre makes the characterisation of fibre-rich composites via imaging accessible to a broader public and increases the scenarios in which the structure of fibres and its changes can be quantified with X-ray CT.

Monica Jane Emerson

2nd Edition of the Workshop on Insegt Fibre

TABLE OF CONTENTS

1 MATERIAL FOR THE WORKSHOP

In the desktop of your workstation, you will find a folder with the name `InsegtFibre_workshopDTU`, containing the material for the 2nd Edition of the workshop on *Insegt Fibre*. Inside the folder, you will find this workshop manual (`InsegtFibre_workshopManual.pdf`) and two folders called `code` and `data`. You are welcome to take home the folder `InsegtFibre_workshopDTU` and use the algorithm for your future projects. Please read Appendix A for the conditions of use of *Insegt Fibre*.

In the workshop we will analyse 2D, 3D and 4D X-ray CT scans of unidirectional (UD) glass and carbon fibre composites. We will learn how to use *Insegt Fibre* step by step. First, we will learn how to obtain the coordinates of fibre cross-sections from 2D images. Secondly, we will measure 3D fibre centre lines. To end with, we will validate the accuracy of the fibre centre lines in a qualitative manner. On the second day there will be the possibility of analysing two 3D images from a time-lapse scan, to measure fibre by fibre the changes introduced by loading a sample in compression.

1.1 INSEGT FIBRE SOFTWARE

Insegt Fibre is composed of a set of scripts and functions that run in MATLAB. You will be running the scripts section by section while guided by this manual. Please read each section of the manual carefully before running the corresponding section of code. User-friendly windows will pop up when input from you (the user) is required. Therefore, it is not a prerequisite to have experience in MATLAB. However, if you are familiar with MATLAB, you will be able to understand the details of the implementation, and better tailor the scripts to your specific needs. At the moment, *Insegt Fibre* aims to be quite general, so as to cover a variety of scenarios.

The folder `code` contains several MATLAB scripts. In Exercise 1, we will work with `InsegtFibre_2D.m` to get you familiarised with the important parameters of *Insegt* [2]. The software *Insegt* (contained in the folder `texture_gui`), is a tool for interactive segmentation of repetitive image patterns developed at the Department for Applied Mathematics and Computer Science of the Technical University of Denmark (DTU Compute). We will use this tool to obtain a pixel-wise segmentation of fibre centre regions, from which we will compute the coordinates of the fibre centres. In Exercise 2, we will use the script `InsegtFibre_3D.m` to track the fibres and obtain 3D centre lines. In Exercise 3, we will validate the accuracy of the measured fibre tracks. On the second day of the workshop, you will have three options: i) write MATLAB code to characterise the fibre tracks (orientations, curvature, diameters), ii) analyse 4D data to measure how fibres reorient and bend under load or iii) measure fibre tracks from your own CT data.

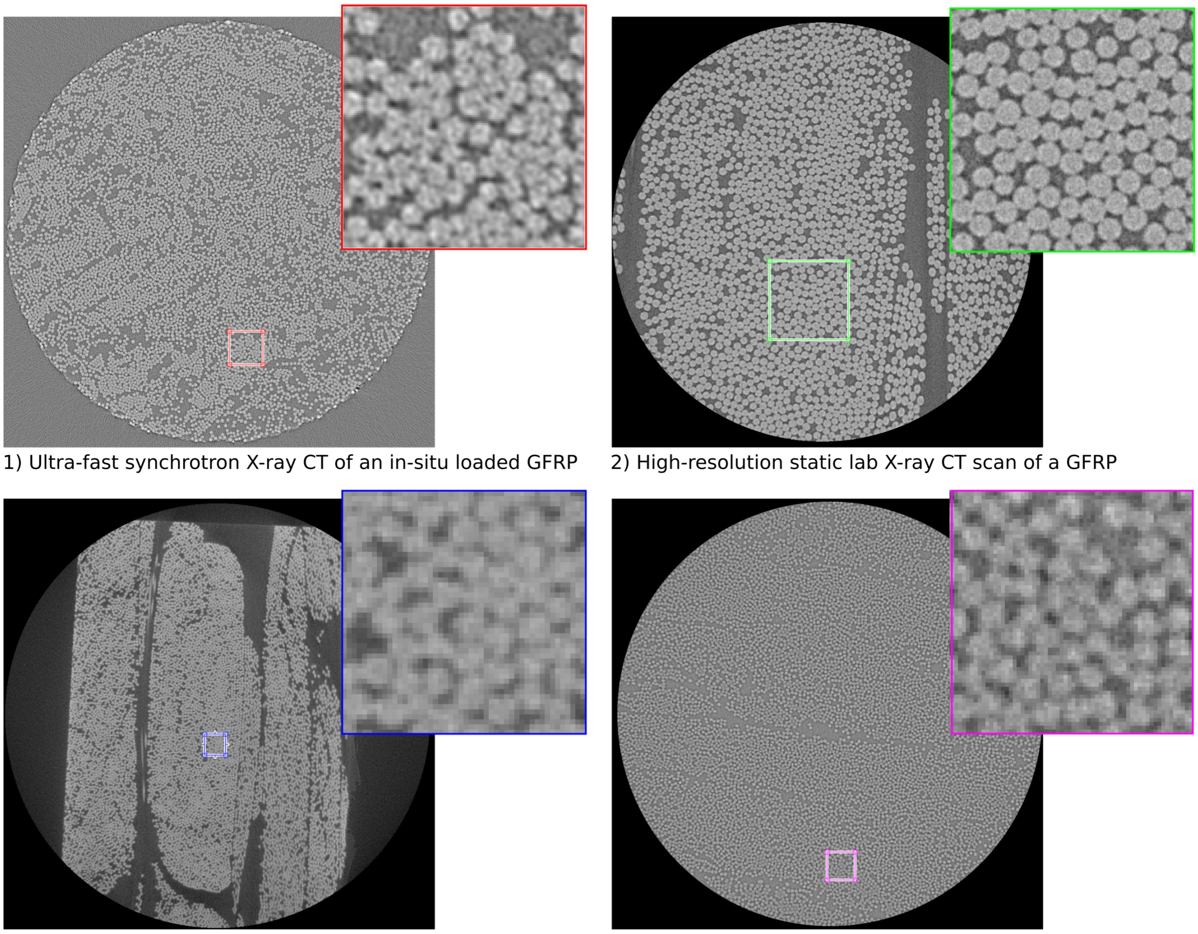
1.2 EXAMPLE CT DATA-SETS

The data-sets used for the workshop are X-ray CT scans of UD glass and carbon fibre-reinforced composites (GFRP and CFRP). The scans were acquired at laboratory scanners and large-scale synchrotron facilities.

1.2.1 2D Data

You will find the 2D data in the folder **data/2Ddata**. The folder contains four 2D images which show fibre cross-sections. Each 2D image is a cross-sectional slice from a CT scan. We have chosen four scans of varying quality, fibre material and spatial resolution, so that you learn how to use *Insegt Fibre* in different scenarios. Figure 1 shows the four 2D images and the relevant information for these data-sets is described in Table 1.

FIGURE 1. 2D X-RAY CT CROSS-SECTIONS OF THE EXAMPLE SCANS USED IN EXERCISE 1.



1) Ultra-fast synchrotron X-ray CT of an in-situ loaded GFRP
2) High-resolution static lab X-ray CT scan of a GFRP
3) Low-resolution static lab X-ray CT scan of a GFRP
4) Static lab X-ray CT scan of a CFRP

TABLE 1. PIXEL SIZES AND FIBRE DIAMETERS FOR THE EXAMPLE SCANS USED IN EXERCISE 1.

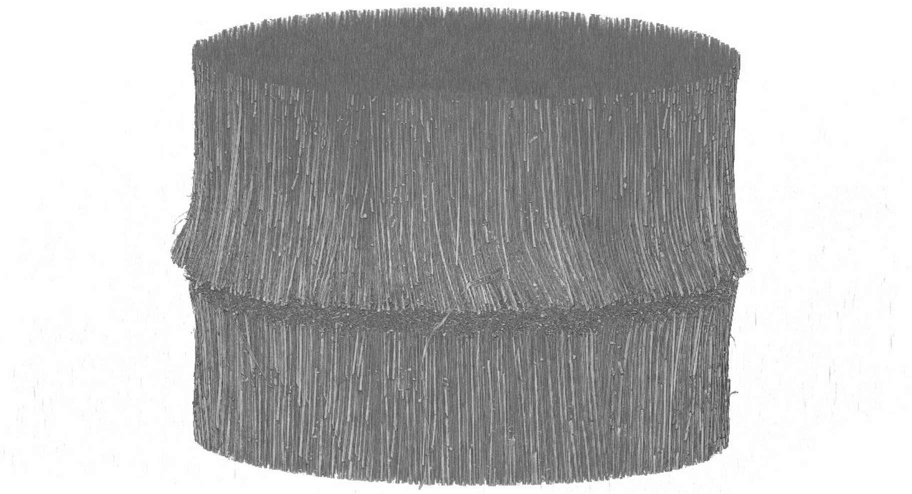
Data-set	Pixel size	Fibre Diameter
1. Ultra-fast synchrotron X-ray CT of an in-situ loaded GFRP	1.1 μm	12 μm
2. High-resolution static lab X-ray CT scan of a GFRP	1.04 μm	17 μm
3. Low-resolution static lab X-ray CT scan of a GFRP	2.81 μm	17 μm
4. Static lab X-ray CT scan of a CFRP	0.96 μm	7 μm

1.2.2 3D DATA

For Exercise 2 we will use a 3D image acquired at the TOMCAT beamline of the Swiss Light Source. The image is part of a time-lapse data-set acquired while compressing a fibre-reinforced composite in the fibre direction. The sample of 1.5 mm in diameter contained UD glass fibres of around 12 μm in diameter. The imaging set-up allowed an acquisition speed of one full data-set per second at a voxel size of (1.1 μm)[3]. Figure 2 shows a 3D volume rendering of the composite, note the notch carved at mid height.

The 3D data-set is hosted at Zenodo https://zenodo.org/record/2597498#.XLAX2BMzZ0s as **`GFRP_Initial.zip`**. We have downloaded it and placed it for you in the folder **`data/3Ddata`**. This 3D image corresponds to the first time/load step of the time-lapse scan, i.e. it was acquired under a load of 0N. Please read Appendix B for the conditions of use of this time-lapse data-set.

FIGURE 2. 3D VOLUME RENDERING OF THE EXAMPLE 3D DATA



1.2.3 4D DATA

On the second day, some of you might want to see how *Insegt Fibre* can be used to track fibre movement across a time-lapse data-set. For this purpose, you will use two 3D images from the aforementioned time-lapse data-set acquired at TOMCAT. These data-sets, located in Zenodo https://zenodo.org/record/2597498#.XLAX2BMzZ0s and named **`GFRP_Continuos_No0.zip`** and **`GFRP_Continuos_No12.zip`**, have already been downloaded and placed in folder **`data/4Ddata`**.

2 Exercises

2.1 Insegt Fibre for 2D Data

The goal of this exercise is to get you familiarised with the *Insegt* graphical user interface (GUI) and the parameters that are required to learn a dictionary that will adequately model your fibre data-set. A good dictionary will be a compact representation of the information contained in the training data, comprised of an intensity image and annotations indicating which pixels are close to fibre centres. To segment the fibre centres in your data-set, we will use the dictionary to perform a pixel-wise labelling of the images in your data-set. For more information, the user is referred to [1,2].

We will now open the script named `InsegtFibre_2D.m` with MATLAB. Make sure your current folder in MATLAB is **`InsegtFibre_workshopDTU/code`**. Run the script step by step by placing your cursor inside each section and pressing 'Ctrl+Enter', or by pressing the 'Run Section' button inside the MATLAB Tab 'Editor'.

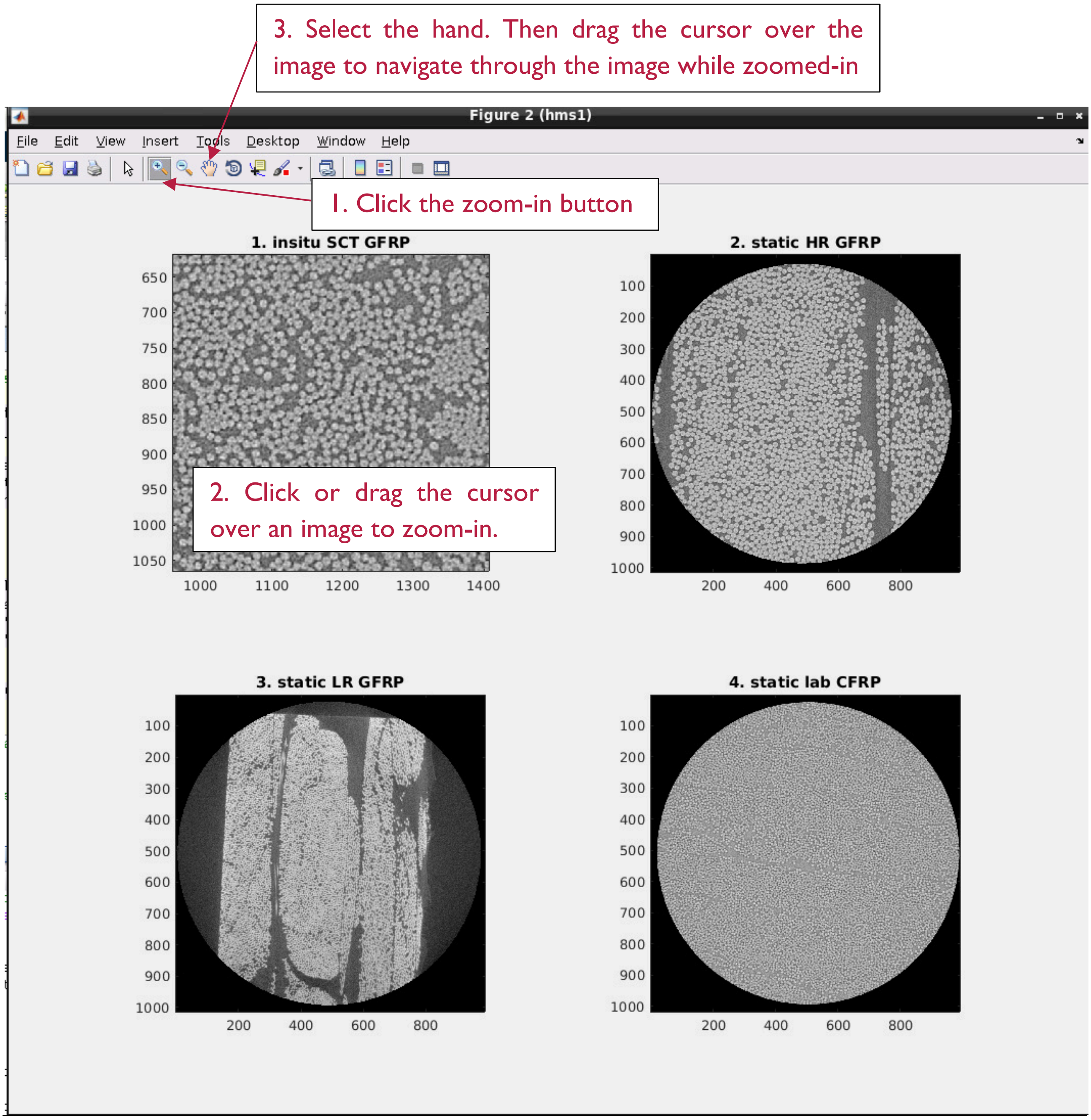
2.1.0 Step 0: Add Paths and Prepare Workspace

```
%% Step 0: Add paths of GUI and other functions and scripts used in Insegt Fibre
close all,
clear all,
addpath(genpath('./texture_gui'))
addpath('./scripts')
addpath('./functions')

disp('Step 0 completed')
```

2.1.1 Step 1: Load and Visualise the Four 2D Images

Inspect the quality of the scans by zooming into the images, appreciate the contrast between material phases, intensity changes inside the fibre cross-sections, pixellation and sharpness of fibre borders.

```
%% Step 1: Load and visualise the four 2D images
%the path of the 2Ddata folder in your computer
str_datafolder = '../data/2Ddata/'; %default value

%Visualise all four datasets
contents_datafolder = dir(str_datafolder);
num_datasets = length(contents_datafolder)-2;

figure
for k= 1:num_datasets
    %read image
    im = imread([str_datafolder,contents_datafolder(k+2).name]);
    %create subtitle for image
    subtitle = [num2str(k),'. ',contents_datafolder(k+2).name(1:end-4)];
    ind = find(subtitle=='_');
    subtitle(ind) = ' ';
    %plot image
    subplot(2,2,k), imagesc(im), axis image, colormap gray, title(subtitle)
end

h = msgbox('Inspect the quality of the four different scans');
waitfor(h)

disp('Step 1 completed')
```



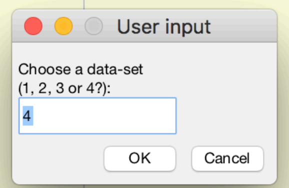
(NOTE THAT IN MATLAB VERSIONS 2018B AND AFTER THE ZOOM AND HAND ICONS APPEAR IN A DIFFERENT LOCATION AND NOT UNTIL THE CURSOR IS LOCATED OVER AN IMAGE)

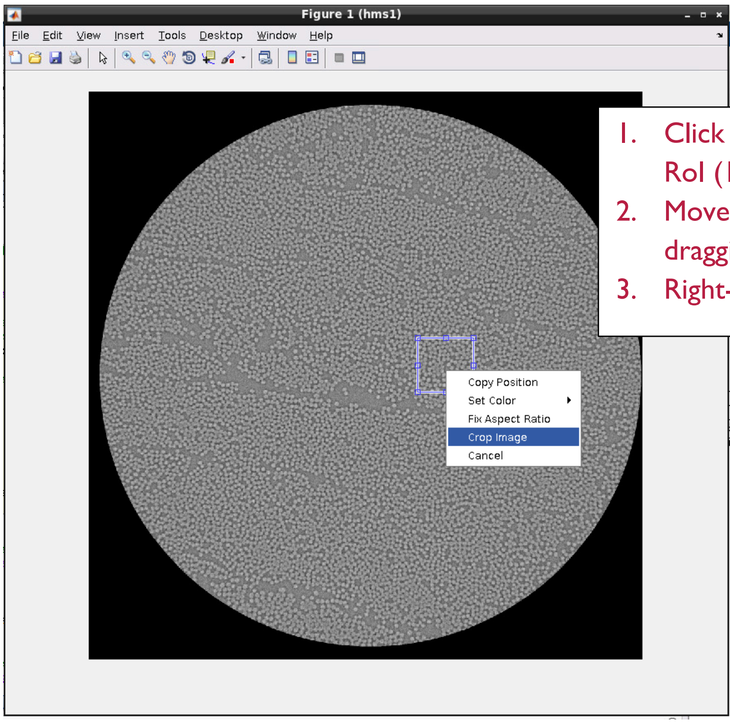
Now, run through sections 2.1.2 to 2.1.5 using the values that appear as default in the windows that ask for user input (i.e. starting with data-set 4). Then, run these sections again for data-sets 1, 2 and 3, introduce appropriate values into the windows as they pop up.

2.1.2 STEP 2: SELECT A DATA-SET AND A REGION OF INTEREST (ROI)

Run sections 3 to 5 with all four data-sets. Start with data-set 4. Select a RoI that contains in between 100 and 1000 fibres (approximately).

```
%% Step 2: Select data-set and region of interest (RoI)
close all
%USER INPUT: select dataset
x = inputdlg('Choose a data-set (1, 2, 3 or 4?): ','User input',[1,20],{num2str(4)});

%read the 2D image chosen by the user
dataset = str2double(x{1});
im = imread([str_datafolder,contents_datafolder(dataset+2).name]);

%print the name of the selected data-set
disp(['Selected data-set: ', contents_datafolder(dataset+2).name(1:end-4)]);

%USER INPUT: select RoI
h = msgbox('Crop a region of interest containing 100-1000 fibres.');
waitfor(h)
im_crop = imcrop(im);

%visualise the selected RoI
if isempty(im_crop)
   h = msgbox('There was an error in the cropping. Run the section again.');
   waitfor(h)
else
    figure, imagesc(im_crop), axis image, colormap gray,
    title('Region of Interest for training the dictionary')
end

disp('Step 2 completed')
```



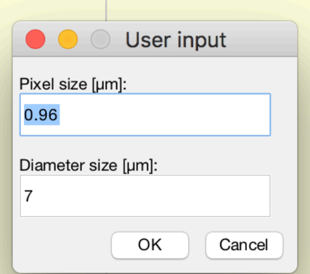


1. Click on the image and drag to select the size of the RoI (100-1000 fibres from a UD bundle).
2. Move the square by placing the cursor inside it and dragging.
3. Right-click inside the square to crop image.

2.1.3 Step 3: Set the Dictionary Parameters

Prior to running the dictionary method, a few input parameters need to be set. The dictionary method is most sensitive to the size of the image patches. We ought to set the patch size to the scale of the fibres in our data. The patch size is computed as the closest odd integer to the ratio between the physical size of the fibres and the pixel size of the scan (see Table 1). The rest of the dictionary parameters are set to default values. If you would like to know more about the patch size and other dictionary parameters, read Appendix D: Considerations on Dictionary Parameters.

```
%% Step 3: Set default parameters and the patch size at the scale of the fibres
close all
%USER INPUT: Parameters that give the scale of the fibres in pixels
prompt_px = ['Pixel size [', char(181),'m]: '];
prompt_diam = ['Diameter size [', char(181),'m]: '];
x = str2double(inputdlg({prompt_px, prompt_diam},'User input',[1,30],{num2str(0.96),num2str(7)}));

%check that the patch size is in the optimal range to ensure precision and
%speed
if ((x(2)/x(1)) < 7)
    factor = 9/(x(2)/x(1));
    patch_size = calcu_psize(factor*x(2)/x(1));
    msg = ['We will upscale your images with a factor ',...
        num2str(factor,'%0.2f'),' to obtain better precision. The patch size is now ',...
        num2str(patch_size),'.'];
    h = msgbox(msg);
    waitfor(h)
elseif ((x(2)/x(1)) > 15)
     factor = 11/(x(2)/x(1));
    patch_size = calcu_psize(factor*x(2)/x(1));
    msg = ['We will downscale your images with a factor ',...
        num2str(factor),' to obtain reduce the computational time. The patch size is now ',...
        num2str(patch_size),'.'];
    h = msgbox(msg);
    waitfor(h)
else
    factor = 1;
    %compute patch size as the closest odd integer to the fibre
    %diameter measured in pixels diam/pixel_size
    patch_size = calcu_psize(x(2)/x(1));
end

%set patch size
dictopt.patch_size = patch_size;
%set the default ditionary parameters
dictopt.method = 'euclidean';
dictopt.branching_factor = 3; %it should at least be 3
dictopt.number_layers = 5; %at least 5. The higher the more accurate, but also slower and more com
num_dict_elts = calc_eltsdict(dictopt.branching_factor,dictopt.number_layers);
dictopt.number_training_patches = 5000; %at least an order of magnitude more than the number of di
dictopt.normalization = false; %set to true if the global intensity of the slices varies along the

disp('Step 3 completed')
```



2.1.4 STEP 4: RUN THE GUI

Use the graphical user interface *Insegt* (see Figure 3) to create the training data-set from which the dictionary model will be learnt.

```
%% Step 4: Run InSegt
%disp the important dictionary parameters
fprintf('\nImportant Dictionary Parameters:\n');
disp(['patch size: ', num2str(dictopt.patch_size)]);
disp(['number dictionary elements: ', num2str(calc_eltsdict(dictopt.branching_factor,dictopt.number_layers))]);

%resize cropped region of interest with factor
im_train = imresize(im_crop, factor);
%open the GUI with 2 labels for pixel annotation (fibre centre region or not)
image_texture_gui(im_train,dictopt,2)

disp('Step 4 completed')
```

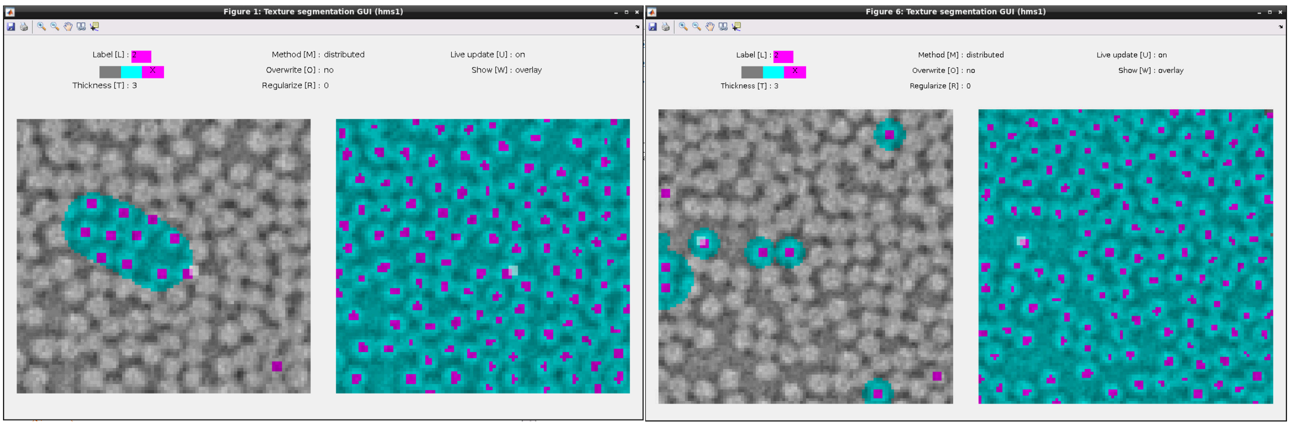
Command Window

```
Important Dictionary Parameters:
patch size: 7
number dictionary elements: 363
```

Annotate the image on the left placing markings as shown in Figure 3, use pink to mark the centre regions of fibres and blue to annotate the rest of the image.

- Change the thickness of the brush by pressing the key ‘t’ or the up and down arrows.
- Change the label colour with ‘l’, or by pressing with your cursor over the selected colour.
- On the right, choose to see the segmentation over the intensity image by pressing ‘w’ until ‘Show [W]: overlay’, or see if the probabilistic segmentation is differentiating centre regions by setting ‘Show [W]: probability map 1’.
- Continue annotating on the left until you obtain a good segmentation result on the right. Before closing the GUI remember to press ‘e’ to export your work, else it will be lost.

FIGURE 3. GRAPHICAL USER INTERFACE (GUI) OF INSEGT, SHOWING TWO DIFFERENT WAYS OF ANNOTATING THE TRAINING IMAGE.

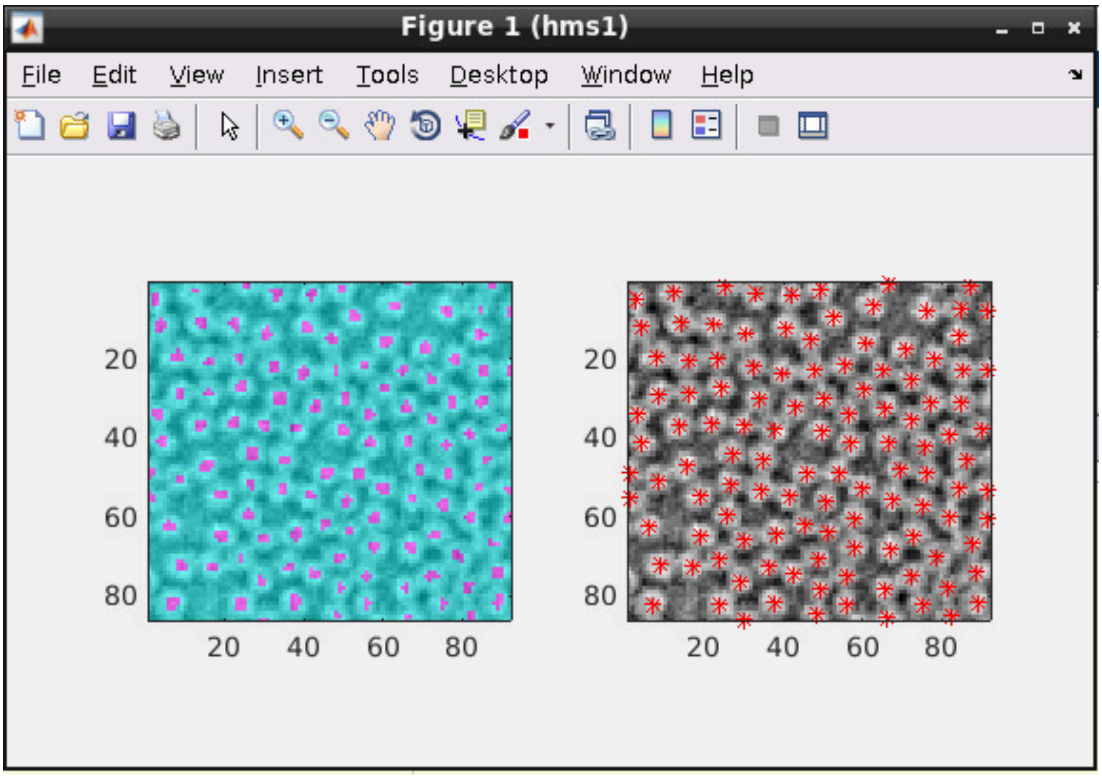


2.1.5 STEP 5: OBTAIN THE CENTRE COORDINATES

Compute the coordinates of the fibre centres as the centroids of the connected components (in pink) in the segmentation of fibre centre regions (gui_S), exported when pressing 'e' while in the GUI. Figure 4 will pop-up when running the following section of code.

```
%% Step 5: Obtain the centre coordinates from the hard segmentation
close all
%USER INPUT: label used for the centre regions
label = 2; %1 (blue) or 2 (pink)

%calculate the centroids of the connected centre regions
c = regionprops(gui_S==label,'centroid');
centrePoints = cat(1,c.Centroid);

%show fibre centre coordinates over slice
figure(1), subplot(1,2,1), imagesc(im_crop), axis image, colormap gray,
hold on, h = imagesc(ind2rgb(imresize(gui_S==label,1/factor), cool(2))); set(h, 'AlphaData',0.5);

figure(1), subplot(1,2,2), imagesc(im_crop), axis image, colormap gray
hold on, plot(centrePoints(:,1)*1/factor,centrePoints(:,2)*1/factor,'r*');

disp('Step 5 completed')
```

FIGURE 4. FIBRE CENTRES DETECTED BY PERFORMING CONNECTED COMPONENT ANALYSIS ON THE SEGMENTATION GIVEN BY THE DICTIONARY OF IMAGE PATCHES. ON THE LEFT, THE SEGMENTATION OF CENTRE REGIONS EXPORTED FROM *INSEGT* (GUI_S). ON THE RIGHT, THE FIBRE CENTRES CALCULATED AS THE CENTROIDS OF THE CONNECTED PINK REGIONS.



Go back and run sections 2.1.2 to 2.1.5 with the other data-sets. Then, run through sections 2.1.6 to 2.1.9 with a data-set of your choice.

2.1.6 Step 6: Save the Dictionary to Process Other Regions of the Scan

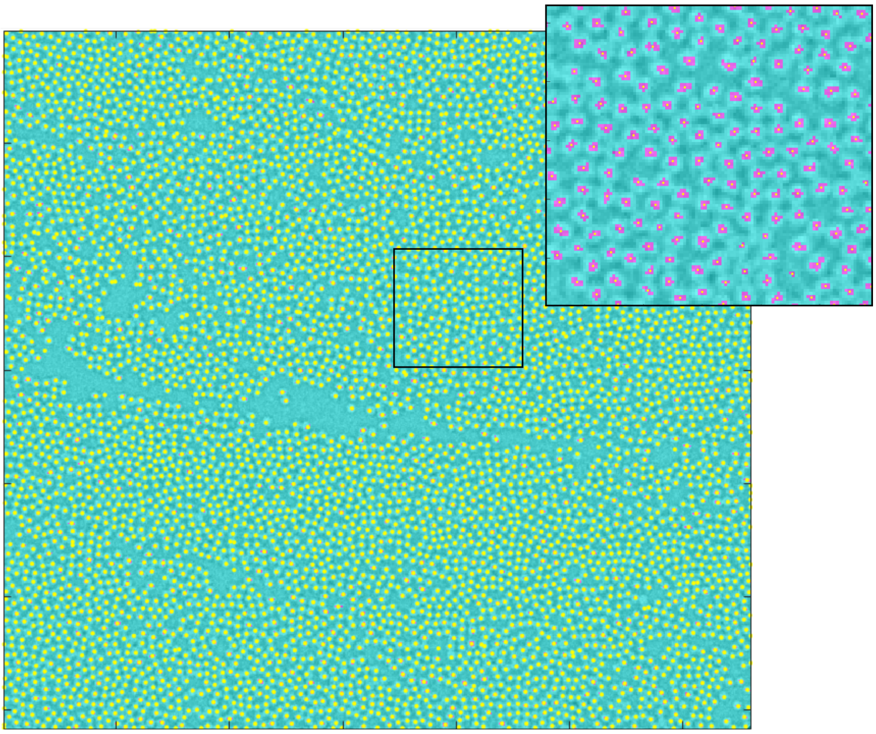
The dictionary created in the GUI by annotating a small training image should be used to process other regions of the scan, or even scans of that sample at other load steps (useful for in-situ loading experiments). To save the dictionary so that we can process other regions of the scan we have to:

i) build the dictionary of intensities, before calling the GUI.
ii) export the dictionary of labels created from the manual annotations, by pressing 'e' before closing the GUI.
iii) update the dictionary of intensities with the exported dictionary of labels.

In this section we learn the dictionary model from the small RoI selected in Section 2.1.2, as we did in Section 2.1.4, but this time we will save the dictionary. We then use this dictionary to process a larger RoI (see Figure 5), which you will crop out from the field of view (FoV). Before running this section, you should have run Sections 2.2.2 and 2.2.3 with the data-set of your choice. Else, it will be run on the last data-set that you analysed.

```
%% Step 6: Save the dictionary to process other regions of the scan
close all
%train the dictionary with the small RoI
dictionary = build_dictionary(im_train,dictopt); %create the dictionary of intensities
image_texture_gui(im_train,dictionary,2) %learn the dictionary of probabilities by annotating in the
dictionary = update_dictionary(dictionary,gui_dictprob); %update dictionary to include the learnt p

%USER INPUT: select a RoI to process with the learnt dictionary
h = msgbox('Crop a region of interest, it can be as big as you like.');
waitfor(h)
im_p = imcrop(im);
if isempty(im_crop)
   h = msgbox('There was an error in the cropping. Run the section again from the cropping.');
   waitfor(h)
else
    figure, imagesc(im_p), axis image, colormap gray,
    title('Region of Interest for finding fibre centres')
end

%apply the dictionary to process the larger RoI
im_process = imresize(im_p,factor);
[S,allP] = process_image(im_process,dictionary);

%calculate the centroids of the connected centre regions
label = 2; %1 (blue) or 2 (pink)
c = regionprops(S==label,'centroid');
centrePoints = cat(1,c.Centroid);

%Centre regions and coordinates over the image to evaluate the accuracy of the segmentation
figure, imagesc(im_process), axis image, colormap gray,
hold on, h = imagesc(ind2rgb(S, cool(2))); set(h, 'AlphaData',0.5);
hold on, plot(centrePoints(:,1),centrePoints(:,2),'y.');
title('Segmented centre regions and fibre detections over the processed image');

if factor ~=1
    figure, imagesc(im_p), axis image, colormap gray,
    hold on, plot(centrePoints(:,1)*1/factor,centrePoints(:,2)*1/factor,'y.');
    title('Centre detections over the original resolution')
end

disp('Step 6 completed')
```

FIGURE 5. EXAMPLE OF FIBRE CENTRE DETECTION WITH DICTIONARY MODEL IN A LARGER REGION OF INTEREST



It might be that after applying the dictionary to another region of the scan, you are not satisfied with the detection of fibre centres. In Sections 2.1.7. and 2.1.8. you will learn how to improve the detection of fibre centres, for when it has not gone as well as expected, i.e. some fibre centres have been missed, or there are false fibre detections in the background.

2.1.7 Step 7: Tune the Threshold Applied to the Probability Map

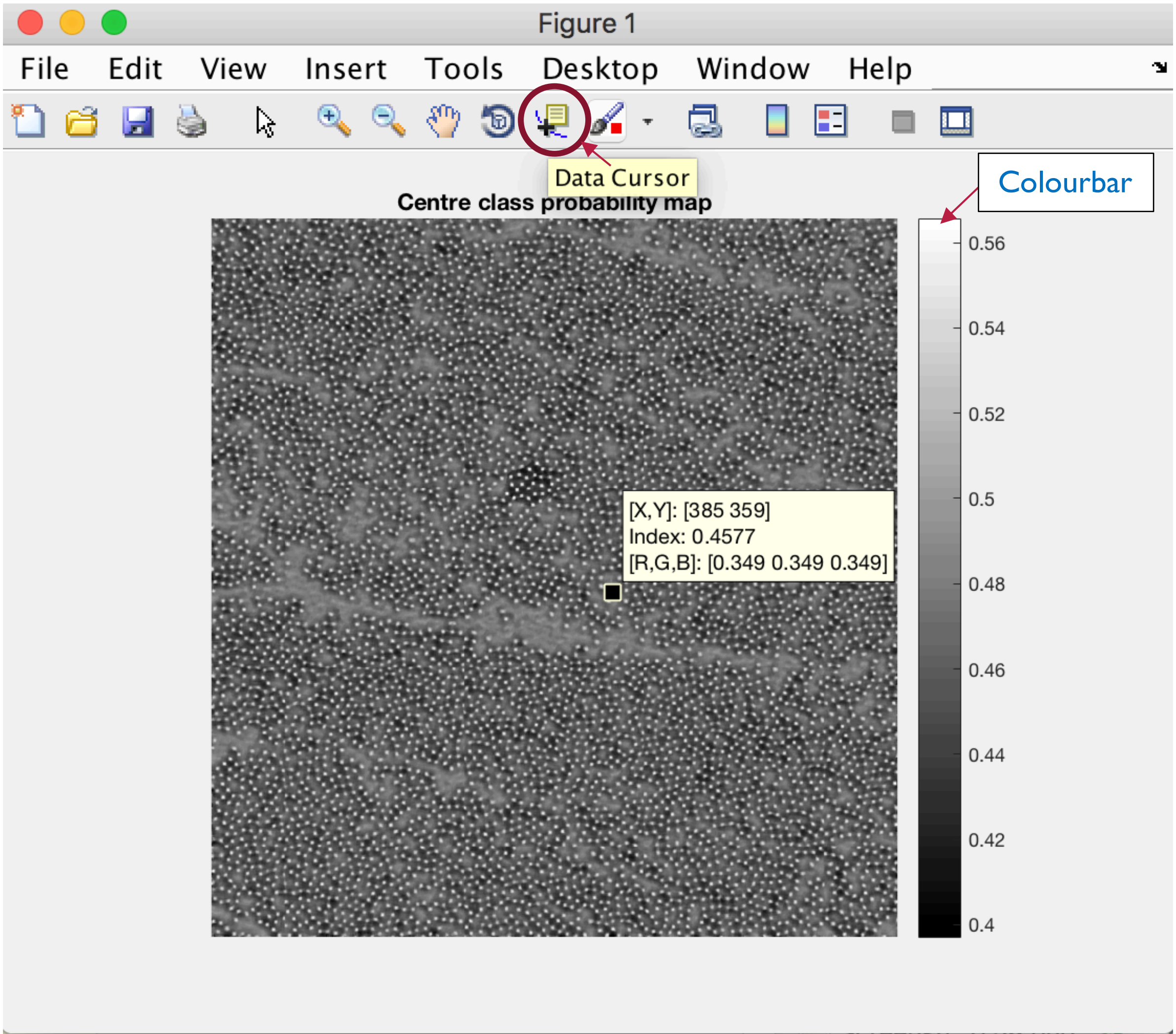
To process an image with the dictionary model learnt from the training data, we call the function *process_image(image,dictionary_options)*. This function provides two outputs, i) the hard segmentation we have been working with until now, which determines whether a pixel belongs to a fibre centre region or not (pink or blue in Figure 4) and ii) a probabilistic segmentation, indicating the likelihood of each pixel belonging to the fibre centre class, see Figure 6. In *process_image*, the hard segmentation is computed from the probabilistic segmentation by setting a default threshold value of 0.5. In this section, we will work with the probabilistic segmentation and set our own threshold value to try and obtain a better hard segmentation of the fibre centre regions.

Run this section to visualise the probabilistic segmentation of centre regions, as shown in Figure 6. Decide a threshold value by looking at the range of probability values (shown in the colourbar) and inspecting the individual pixel values with the data cursor. Apply different threshold values and see how the segmentation of centre regions changes.

```
%% Step 7: Tune the threshold applied to the probability map
close all
%visualise the propabilistic segmentation of fibre centre regions
figure(1), imagesc(allP(:,:,2)), axis image off, colormap gray, title('Centre class probability map'), colorbar

y = 'yes';
while strcmp(y,'yes')
    %USER INPUT: threshold value for the probability map
    x = inputdlg('Choose threshold: ','User input',[1,20],{num2str(0.5)});

    %Centre regions over the image to evaluate the accuracy of the segmentation
    figure, imagesc(im_process), axis image, colormap gray,
    hold on, h = imagesc(ind2rgb(allP(:,:,2)>str2double(x{1}), cool(2))); set(h, 'AlphaData',0.5);

    %USER INPUT: do you want to try another threshold??
    y = questdlg('Do you want to try another thresold?: ','User input','yes','no','yes');
end

disp('Step 7 completed')
```

If you have experience with MATLAB and image analysis, you might want to try using a different method than thresholding and connected component analysis for computing the fibre centre coordinates from the probabilistic segmentation, for example, blob detection.

FIGURE 6. PROBABILISTIC SEGMENTATION OF FIBRE CENTRE REGIONS.



2.1.8 Step 8: Save Your Manual Input to Add Additional Markings Later On

It might be that after determining the threshold for the probability map yourself, you still don't obtain results that are satisfactory enough. In this case, the dictionary model should be improved. In this section we will learn how to save your manual input so that you can improve over it in the future, instead of starting all over again from scratch.

Start by labelling just a few fibres and save the markings with the name `dataset_number.mat` in the folder **`InsegtFibre_workshopDTU/data/saved_data/trainingData`** by pressing 's'. If the dictionary created with this amount of input is not good enough, you can call the GUI with the labelling you saved, as shown in this section.

```
%% Step 8: Save your manual input to add additional markings later on
close all
%annotate a little in the GUI and save your manual input
h = msgbox(['Annotate a little in the GUI and press [S]. Save your annotations '...
    'in the folder "InsegtFibre_workshopDTU/data/saved_data/trainingData" as dataSetNrXX.']);
waitfor(h)
image_texture_gui(im_train,dictopt,2)
%save the training region, over which you placed the markings
save('../data/saved_data/trainingData/im_train.mat','im_train');

%Call the GUI with a saved labelling
load('../data/saved_data/trainingData/im_train.mat');
[filename,pathname] = uigetfile('*_labels_indexed.png', 'Choose label image', '../data/saved_data/trainingData/');
labelling = imread([pathname,filename]); %read labelling
image_texture_gui(im_train,dictopt,2,labelling) %call the GUI with the saved labelling

disp('Step 8 completed')
```

2.2 Insegt Fibre for 3D Data

In this exercise we will combine the learnings from Exercise 2 concerning the detection of fibre cross-sections in 2D images. These are providing relevant manual input and saving it for improving in the future, saving the dictionary model to process other regions of the scan and tuning the threshold applied to the probability map. Once we have detected the centres of the fibre cross-sections, we will perform the tracking step, so as to connect the centre points that belong to the same fibre, and obtain the desired 3D fibre trajectories.

Open the script named `InsegtFibre_3D.m`, make sure your current folder in MATLAB is **`InsegtFibre_workshopDTU/code`** and run the script step by step. In the first run, use the default values that appear in the input windows wherever possible. In a second run, you can try making changes.

2.2.0 Step 0: Prepare Workspace

Tells MATLAB where to find the functions and scripts called by `InsegtFibre_3D.m`.

2.2.1 Step 1: Locate Folder with CT Data

In the window that will pop-up when running this section, select the folder where the 3D data is located. For the workshop, we have placed the data for the 3D example in **`InsegtFibre_workshopDTU/3Ddata/GFRP_Initial/rec_16bit_0`**.

2.2.2 Step 2: Load and Visualise the Data

Select the region of the sample that you would like to analyse. You can do this interactively or by specifying the coordinates which define your volume. In the first run, choose 'I know the coordinates'. Bear in mind that it can take up to 5 minutes to load the full volume, comprised by 1600x1600x1400 voxels.

2.2.3 Step 3: Compute a Dictionary Model from the CT Scan

In the first run, you will create the labelling of the training data from scratch, and select a region containing between 100 and 1000 fibres interactively. In future runs, you can load labellings saved in the past. Do not worry if the hard segmentation does not show all the centres, just make sure that the probability map does by pressing 'w'. Remember that you can always tune the threshold later!

Before the GUI opens you will be asked to save the training image. Then, in the GUI, you should press 's' to save the training data. Call it dataSet4_version1, for example, and save it in the folder **`InsegtFibre_workshopDTU/data/saved_data/trainingData`**. Before closing the GUI remember to press 'e' to export the dictionary trained from your annotations.

2.2.4 Step 4: Obtain the Centre-class Probabilistic Segmentation for a Slice

Apply the trained dictionary to a slice and inspect the values of the probability map to choose a good threshold. Pay attention to the colourbar, look at the histogram of probabilities and use the data cursor, as shown in Exercise 2.

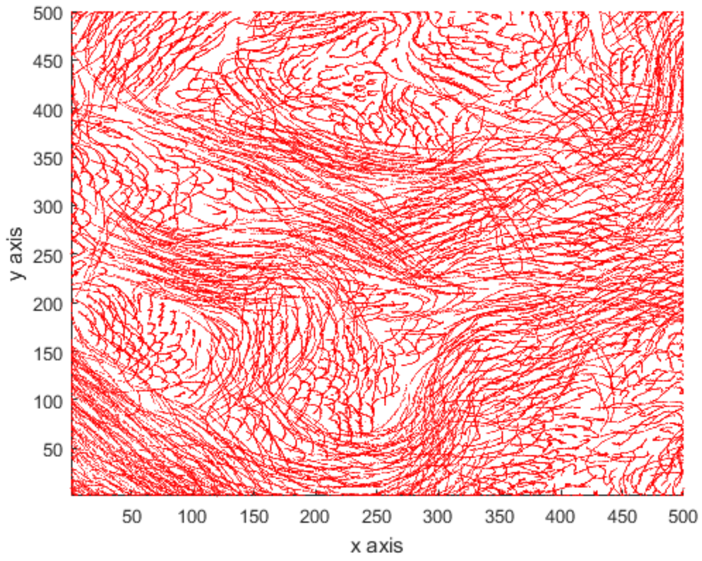
2.2.5 Step 5: Obtain the Fibre Centres by Thresholding

Run the section multiple times to try different threshold values until you are satisfied with the fibre detections. If you cannot find a threshold which gives a good segmentation of the centre regions, run Step 3 again and add more markings to your saved labelling.

2.2.6 Step 6: Detect the Fibre Centres in the Batch of Slices Forming the Volume

Process the full 3D volume with the selected threshold to obtain the 2D coordinates for each fibre cross-section at every slice. It **can take up to one hour to process the full example volume**. However, if you use the default settings you will get a 500x500x142 voxel volume, which you will be able to process in just over one minute. We are using a step = 10, meaning that we are processing one out of every ten slices. A step = 10 will often give enough precision in the depth direction, as the type of fibres we work with are aligned and cannot physically bend with a curvature in the scale of one diameter. However, to obtain more redundancy and possibly better accuracy, the step size can be decreased at the cost of increasing the computational time.

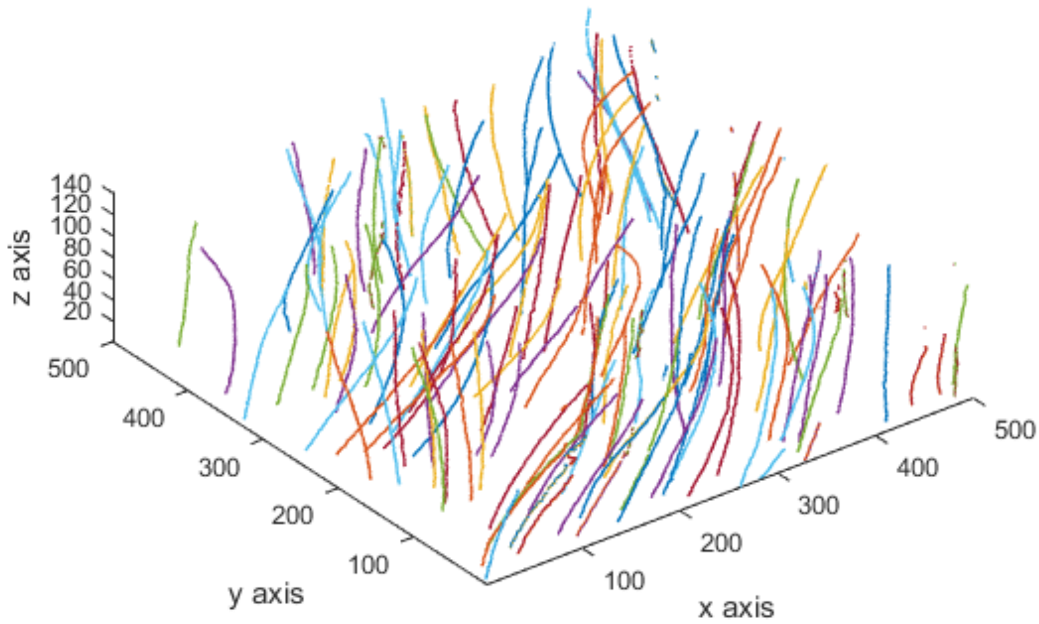
Use the rotating tool to visualise the centre points from different directions, appreciate the fibre misalignment and curvature by looking at the tracks from the top of the sample, as shown in the following illustration.



2.2.7 Step 7: Compute the 3D Fibre Trajectories Via Tracking

Track the 2D fibre centres along the depth of the volume to connect the detections belonging to the same fibre and thereby obtain the three-dimensional fibre trajectories. Look at the figure produced in the previous section, concerning the number of fibres detected per slice. Choose to start the tracking from either the top or the bottom slice, depending on where you have found a larger amount of fibres.

Use the rotating tool to visualise the tracks from different directions. You will be able to select the number of fibres to show, visualise a smaller number of fibre tracks so that you can inspect them better.



2.3 Validate Accuracy of Centre Positions and Fibre Tracking

In this exercise we will perform a qualitative inspection of the precision in the detection of fibre cross-sections and the accuracy of the tracking. Run the script `InsegtFibre_validate3DTracks.m` section by section, use the default parameters to validate the tracks obtained from `InsegtFibre_3D.m` in your first run, when you were using the default parameters.

The graphical user interface for validation will open up when you run the last section. In the GUI, use the arrows of your keyboard to navigate through the volume. 'Up' and 'down' arrows for moving a slice up and down, 'left' and 'right' arrows for moving 10 slices up or down at a time.

To assess the accuracy of the tracking, zoom in until you see numbers over the fibres. Slice through the volume to check whether the same number is assigned to the same fibre at each point in depth.

To assess the accuracy of the centre detections, slice through the volume checking the placement of the detection with respect to the actual fibre centres. When zoomed in you can press the key 'n' to change from fibre numbers to centre detections.

Appendices

A. Conditions of Use for Insegt Fibre

When using our algorithm and scripts for your future projects please remember to credit our work by citing articles [1, 2]. These articles cover the algorithm employed for detecting the individual fibre centres and the method for training the algorithm efficiently and with minimal user input.

If you need support on running Insegt Fibre and analysing your specific CT data, contact Monica Jane Emerson via email at monj@dtu.dk. If the method is not directly applicable to your data, at the Centre for Quantification of Imaging Data from MAX IV (QIM) we are happy to discuss potential collaborations and pursue publications in common. For more information visit qim.compute.dtu.dk.

B. The Applications of Insegt Fibre

In collaboration with material scientists and mechanical engineers, we have used Insegt Fibre to characterise fibre orientations, fibre diameters and fibre deflection under load [1, 3, 4]. Thanks to Insegt Fibre's robustness to image quality, the method is able to analyse a representative size of the material from a single scan (e.g. a full bundle), fibres made of either glass or carbon, scans acquired at reduced scan times and ultra-fast scans acquired while loading a material in situ. In [5] we published the multimodal and multiresolution data-sets used for validating the precision of Insegt Fibre.

To hear more about how Insegt Fibre has aided material scientists in the study unidirectional fibre reinforced composites, contact our collaborators Ying Wang (ying.wang-4@manchester.ac.uk), Lars Pilgaard Mikkelsen (lapm@dtu.dk), Kristine Munk Jespersen (kmun@dtu.dk) or Philip John Withers (p.j.withers@manchester.ac.uk).

C. Conditions of Use for the Time-lapse Data-set

For use of the data at https://zenodo.org/record/2597498#.XLAX2BMzZ0s, please cite the DOI of the Zenodo repository (doi: 10.5281/zenodo.2597498) and the relevant papers [3,6]. These papers explain details on the material, the sample preparation and the in-situ scanning of the unidirectional composite used as example in this workshop.

D. Considerations for the Parameters of the Dictionary

To calculate the **optimal patch size**, divide the fibre diameter by the pixel size and round the resulting number to the closest odd integer. If the optimal patch size is lower than 7, the image should be up-scaled to obtain better precision in the centre estimation. If it is higher than 17 (because the resolution of the data-set is very high compared to the fibre size), the image should be downscaled to avoid the unnecessary use of memory and the longer than needed run time.

The **number of dictionary elements** is determined by the branching factor and number of layers. The number of elements should be large enough to capture the variety in the image; however, one should bear in mind that the dictionary is meant to be a compact representation of the data-set. Due to the regularity of the fibres under study, the accuracy of the method will not increase significantly after a certain point. A good value is around 300, but this number could be increased for data-sets with poor quality. We will stick to 363 in the workshop.

The **branching factor** should at least be 3, and could be increased to 5 for better performance when the quality of the data-set is very low. We stick to 3 during the workshop.

The **number of layers** should be increased for low-quality data-sets so as to have more dictionary elements and better represent the variety in the data. We use 5 in Exercise 1 and 7 in Exercise 2.

The **number of training patches** should be at least an order of magnitude larger than the number of dictionary elements. We will use 5000 in this workshop.

The **normalisation** parameter. If the intensity varies across the cross-sectional slices in the volume, but is constant in the field of view, set the normalisation parameter to 'true'. We will stick to ´false´ for the workshop.

Pre-processing of the images might be beneficial in some cases. In case of beam hardening, where the intensity of the image will vary across the field of view, pre-processing for intensity equalisation is advised.

REFERENCES

[1] M.J. Emerson, K.M. Jespersen, A.B. Dahl, K. Conradsen, L.P. Mikkelsen, Individual fibre segmentation from 3D X-ray computed tomography for characterising the fibre orientation in unidirectional composite materials, *Composites Part A: Applied Science and Manufacturing* 97 (2017) 83-92.

[2] V.A. Dahl, C.H. Trinderup, M.J. Emerson, A.B. Dahl, Content-based propagation of user markings for interactive segmentation of patterned images, *arXiv preprint*, arXiv:1809.02226, 2018

[3] M.J. Emerson, Y. Wang, P.J. Withers, K. Conradsen, A.B. Dahl, V.A. Dahl, Quantifying fibre reorientation during axial compression of a composite through time-lapse X-ray imaging and individual fibre tracking, *Composites Science and Technology* 168 (2018) 47-54.

[4] M.J. Emerson, V.A. Dahl, K. Conradsen, L.P. Mikkelsen, A.B. Dahl, Statistical validation of individual fibre segmentation from tomograms and microscopy, *Composites Science and Technology* 160 (2018) 208-215.

[5] M.J. Emerson,V.A. Dahl, K. Conradsen, L.P. Mikkelsen, A.B. Dahl, A multimodal data-set of a unidirectional glass fibre reinforced polymer composite, *Data in brief* 18 (2018) 1388-1393.

[6] Y. Wang, S. C. Garcea, T. Lowe, E. Maire, C. Soutis, P. J. Withers, Ultra-fast time-lapse synchrotron radiographic imaging of compressive failure in CFRP, in: *17th European Conference on Composite Materials*, Munich, Germany, 2016.

[7] Y. Wang, T. L. Burnett, Y. Chai, C. Soutis, P. J. Hogg, P. J. Withers, X-ray computed tomography study of kink bands in unidirectional composites, *Composite Structures* 160 (2017) 917-924.