Theon

Version 0.9

UNIVERSITY OF LIVERPOOL

User Manual

Andrew Dean, Brodie Payter, Jacob Curtis,
William Christian, Eann Patterson

Table of Contents

1 Introduction

Orthogonal decomposition is a technique for reducing the dimensionality of 2D data fields. These data fields may consist of any type of data, such as image data, strain fields or displacement fields. The data is decomposed into a set of real valued coefficients using a set of orthogonal kernel functions. The coefficients are then collated into a feature vector. A reconstruction of the original data can be obtained by multiplying each of the kernels by their associated coefficient in the feature vector. By using more coefficients, a better reconstruction can be obtained. A detailed description of the mathematics behind orthogonal decomposition can be found in [1]. For smooth data fields (such as most strain and displacement fields measured using experimental mechanics techniques), good reconstructions can typically be obtained using substantially fewer coefficients than there were pixels in the original dataset. The orthogonal decomposition technique has been used as the basis for a validation methodology for finite element models [2, 3] and also as part of a strain-based assessment technique [4].

This software package can be used to decompose and compare 2D spatial data regardless of the geometry of the data field and is capable of importing data from a range of different filetypes. The package can be used to perform simple image operations, such as rotating and cropping data, prior to the decomposition process. Decomposition can be performed using any set of orthogonal kernel functions, but in this package the Chebyshev polynomials have been used.

2 Starting the Program

Once installed, the program can be started by clicking on its icon in the Start Menu or on the desktop. Once loaded, the program will display the “Align Data” window as shown in Figure 1. On the left-hand side is the display, this will show the data that has been imported. On the right-hand side are the program controls, shown in Figure 2. The controls include, the “First Dataset” and “Second Dataset” controls, which load the data into the program. Below these buttons is a control panel used for magnifying, translating and rotating the datasets. Finally, there is the “Decompose data” button which is used to move to the next window. When the program starts, only the “Second Dataset” button will be enabled, this is because at least one dataset has to be loaded into the program before decomposition can occur. Importing data is covered in more detail in the next section.

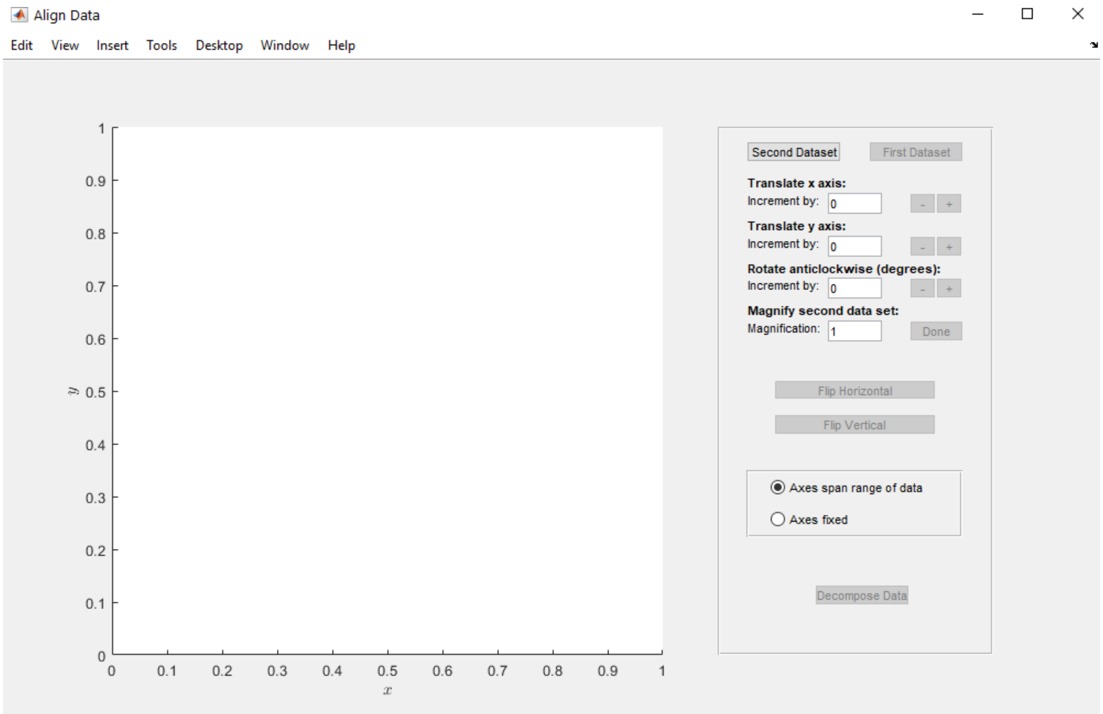

Figure 1: The "Align Data" window as it looks when the program starts.

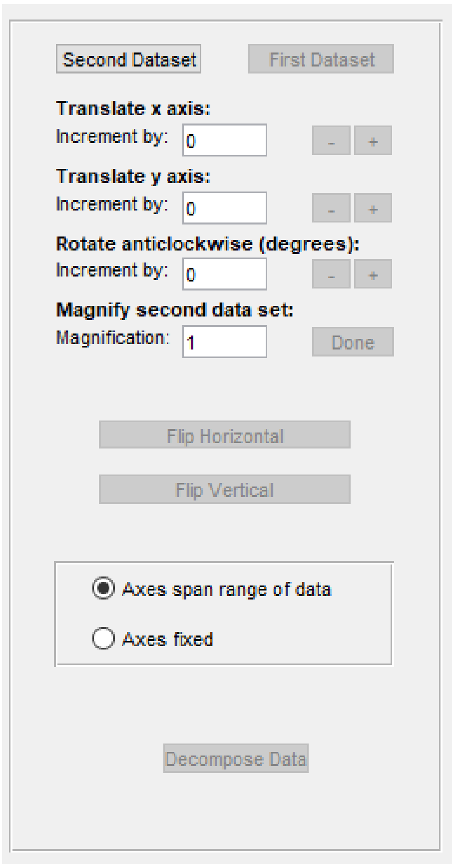

Figure 2: The control panel used to import, align, and progress to the "Data Decomposition" window.

3 Importing Data

3.1 Matlab and Text Files

Data is imported by clicking the “Second Dataset” button, shown in Figure 3. This button is enabled from the start as the second dataset, as defined in [5], is often simulation data and thus tends to be densest. By loading it first it sits below the first dataset and thus makes it easier to align the data. After being clicked, a standard file picker window will appear. This file picker can be used to select files which are of one of three types: Matlab data files (.mat), image files or text files. Note, .mat files and text files must be in the correct format in order for the program to run. The file must contain a ‘D’ variable. If D is a matrix then any holes must be marked with NaN (not a number), if D is a vector then there must also be an ‘x’ and ‘y’ vector of the same size, again with holes marked using NaN. Double clicking on the desired data file will load this dataset into the program. If the file is a Matlab or text file (.mat, .csv, .txt), the dataset will be displayed on the left-hand side, shown in Figure 4. The “Decompose Data” button will be enabled and this dataset can be decomposed. The “First Dataset” button will also be enabled, clicking this will load the next dataset. This will open up a file picker window, this time opening in the same folder as the data that has been selected for the second dataset. Selecting a the file will load the first dataset on top of the second, shown in Figure 5. The data must then be aligned before performing decomposition, this is explained further in “Aligning Data”. When decomposing data the first dataset is processed less than the second. Therefore, when using Theon for validation, the simulation data should be the second dataset, and the experimental data should be the first dataset. This is described further in in [5].

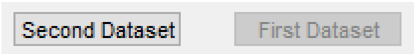

Figure 3: The “First Dataset” button used to import data.

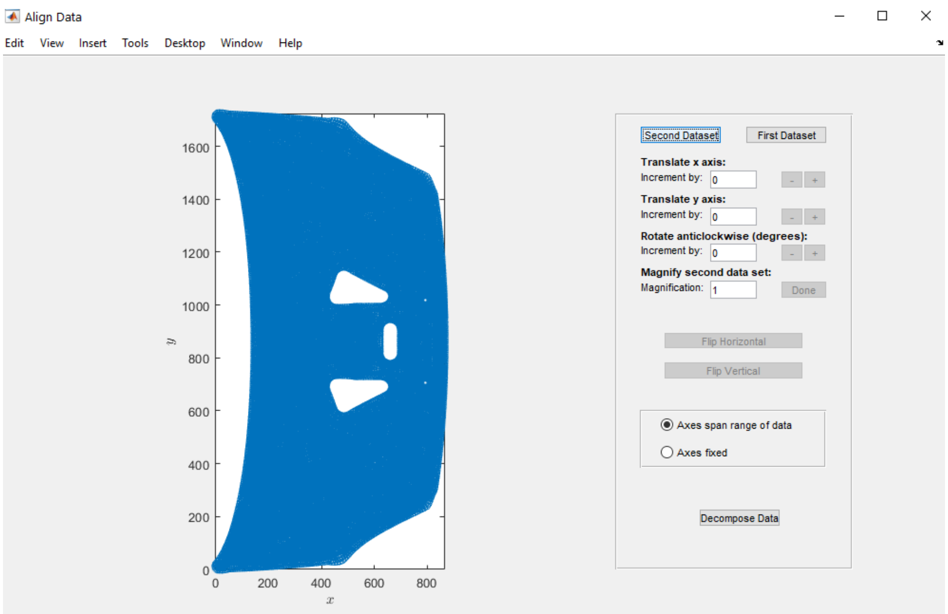

Figure 4: How the program looks after a dataset is imported.

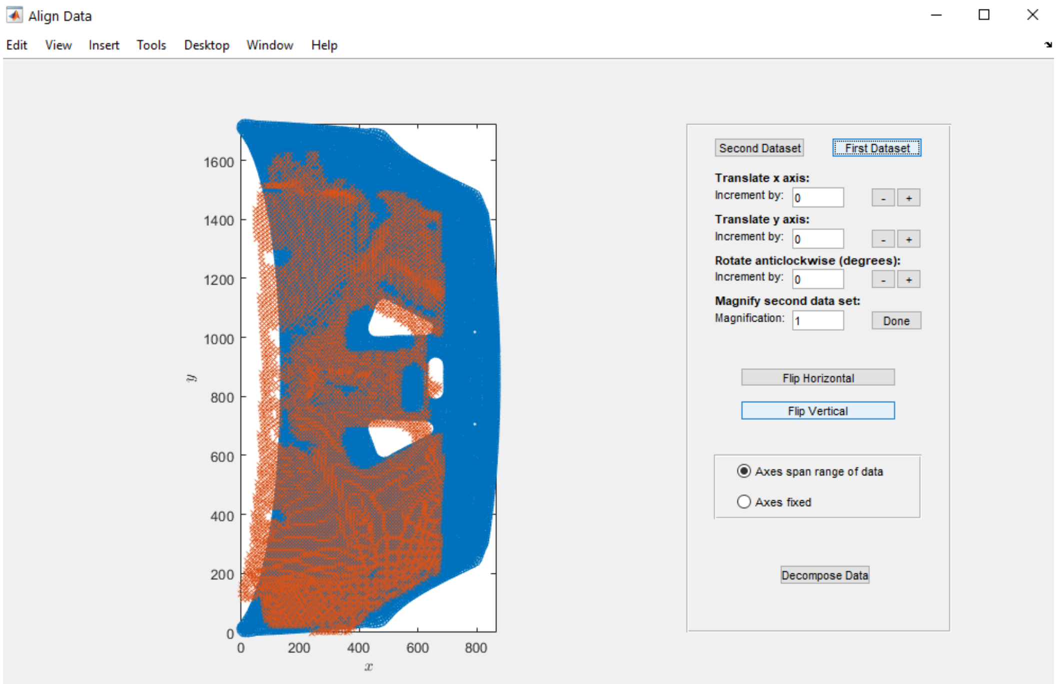

Figure 5: How the program looks when both datasets are imported.

3.2 Image Files

When importing data from an image file, the data will not be immediately displayed on the left-hand side, instead a new window will open, shown in Figure 6. This window displays the imported data on the left-hand side and the controls on the right. Image files store information as a matrix of pixels, each pixel is coloured to represent a value. To import data from an image file the relationship between the colour of each pixel, and the value that colour represents must be determined. This relationship is established in the "Image Picker" window.

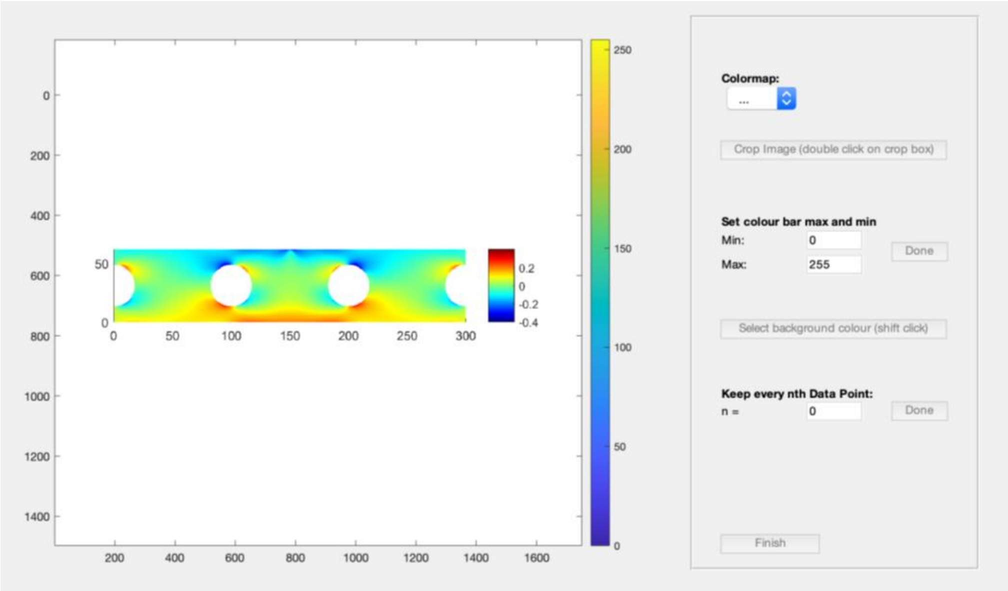

Figure 6: How the program looks after choosing to import an image file.

Firstly, using the drop-down box, shown in Figure 7, the appropriate colour-map must be selected. The options for colour-maps are, "gray", "jet", "parula" and "hsv". The colour-bar will change based on the selected colour-map to indicate the colours that the program expects to find in the image.

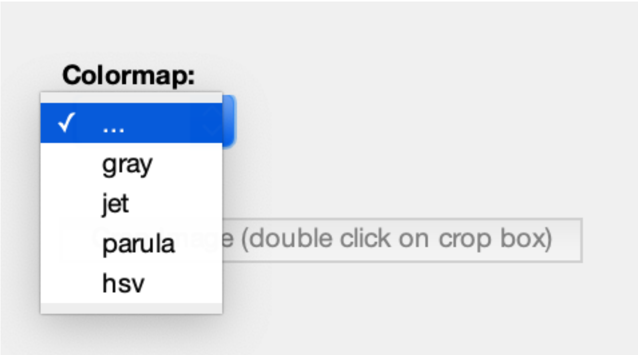

Figure 7: Drop-down menu used to select the colour-map.

After choosing the correct colour-map, the "Crop-Image" button will be enabled, shown in Figure 8. After clicking this button the cursor will change shape to indicate that cropping is in progress. Click and drag the cursor on the image to draw a rectangle around the appropriate data. Resize this rectangle as necessary by clicking on the blue line and dragging it. Once the area has been selected, double-clicking on top of the rectangle will crop the image. Any data cropped out will be discarded. If no cropping is required then drag the edges of the rectangle to the edges of the image.

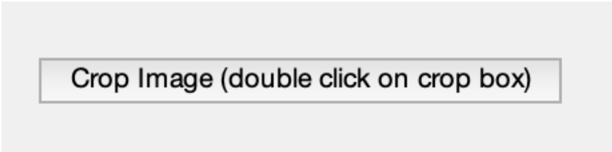

Figure 8: The "Crop Image" button used to change the cursor to be able to crop the image.

The user can now define the extrema of the colour-map. The maximum and minimum values should be written in the two boxes, shown in Figure 9, this will then set the maximum and minimum values of the dataset and the colour-map.

Figure 9: The controls used to set the extrema of the colour-map.

After cropping the image, the "select background colour" button becomes available, shown in Figure 10. In some parts of the image there will be areas that contain no relevant data, to define the colour that represents these areas, first press the "Select background colour" button. Then, whilst holding shift and clicking on the colour, this colour will be defined as indicating no data. An example of an area of no data would be the background colour an image generated by a finite element package.

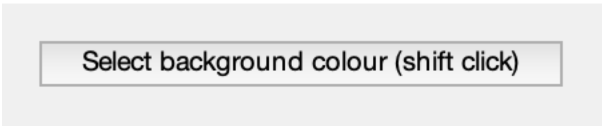

Figure 10: The "Select background colour button".

The final step when importing images is choosing the data points to keep. Using the box shown in Figure 11, every n^{th} data point can be kept. After clicking "Done", the image will alter to reflect the chosen number of data points. This should be performed when high resolution images are being imported. By eliminating data points, accuracy is not necessarily lost while the program will be able to process the image in much less time.

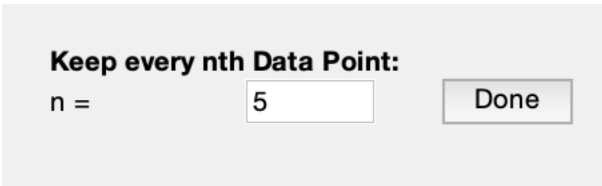

Figure 11: The controls used to define which data points to keep from an image file.

After performing all the steps in the image import wizard, the image is now ready to be imported by clicking the "Finish" button. This will return the user to the "Align Data" window and display the image data on the left-hand side.

3.3 Aligning Data

When working with two datasets it is necessary to move them such that they use the same coordinate system prior to decomposition. The align data window will show the spatial locations of the two datasets with the second dataset represented by blue circles and the first dataset represented by red crosses. Multiple techniques can be used to align the data, as described in the following subsections.

3.3.1 Translate Data on the X and Y Axis

When the two datasets are imported, it is necessary to align them correctly. The simplest method is by dragging the data to the correct position. This can be performed by clicking and holding the first dataset and then dragging the data to the desired position. Alternatively, the user can translate the data along the x and y axis by a defined amount. Using the controls shown in Figure 12, the user can define what increments the data should be translated and then using the '+' and '-' buttons, can translate the data in either the positive or negative direction. Figure 13 demonstrates data before and after being aligned.

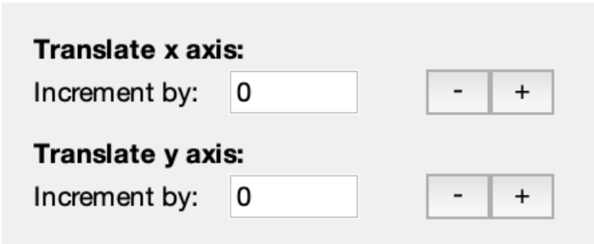

Figure 12: Translation control panel used in the "Align Data" window.

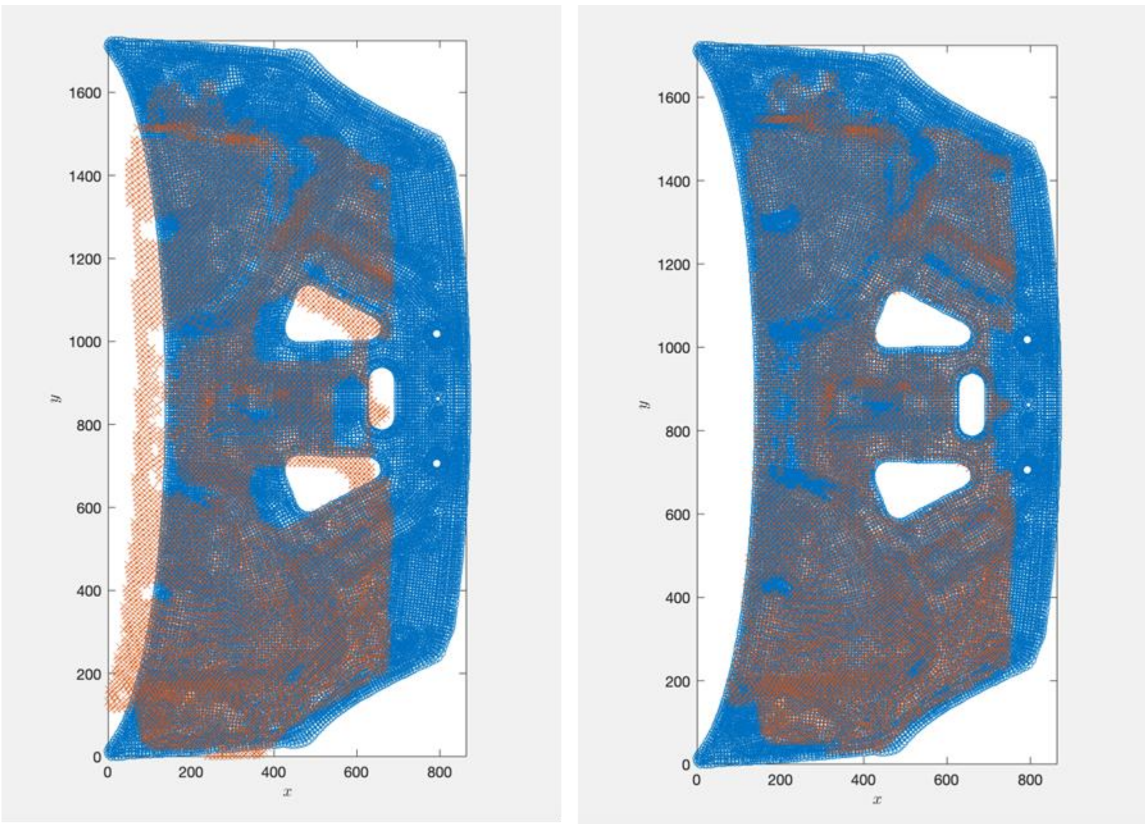

Figure 13: How two datasets look before (left) and after alignment (right).

3.3.2 Rotate, Flip and Magnify Data

When importing data from different sources it is common to find that the data has been rotated or flipped. Alternatively, both datasets may have different orientations. Before aligning both datasets, it is necessary to ensure the correct orientation. The buttons shown in Figure 14 rotate or magnify the dataset by increments defined by the user. Using the "Flip Horizontal" and "Flip Vertical" buttons will flip the data over the y and x axis, respectively.

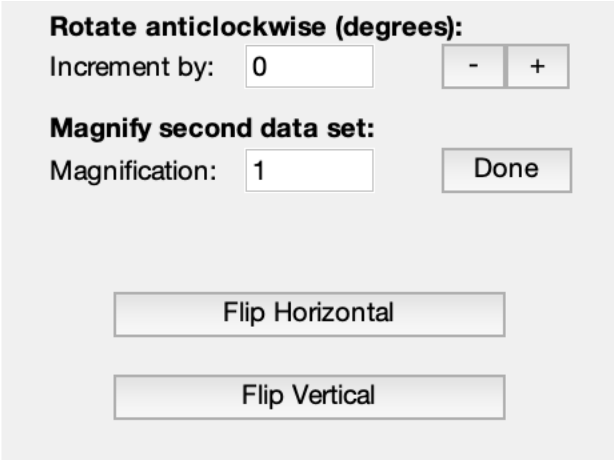

Figure 14: Control panel used to rotate, flip and magnify data.

3.3.3 Axes

The penultimate option in the "Align Data" window is the choice to have fixed axes or for the axes to span a range of data. This only affects the way the data is displayed, and has no effect on the relative positions of the datasets. Figure 15 shows these two options, choosing the first will cause the axes to automatically fit to the data as it is translated. The second option will cause the axes to remain fixed around the second dataset.

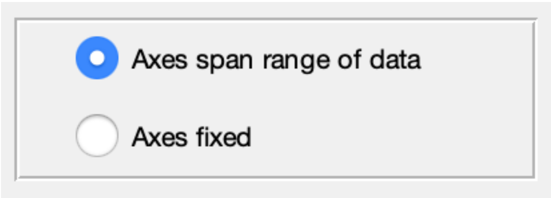

Figure 15: Controls to determine whether axes are fixed or span range of data.

3.4 After Alignment

Once the data is aligned click on the "Decompose Data" button. Pushing this will open the "Data Decomposition" window, shown in Figure 16. Upon pushing this button the program works out the locations at which data is present in one but not the other and eliminates this, allowing for an appropriate comparison, the two data sets are then plotted as the first figure. This window and the eight figures will be explained further in Section 4.

4 Decomposition

4.1 Decomposing Data

The "Data Decomposition" window shows four figures for each of the datasets imported on the left-hand side. While the right-hand side of the window has various controls for processing and interpreting the data, shown in Figure 17. The first two options on the control panel require user input, firstly the "Order", which controls the maximum order of polynomials that should be used to represent the data. The number of terms that will be in the feature vector when using all of the Chebyshev polynomials up to the selected maximum order, can be calculated as:

$$(No.Coeffs) = \frac{1}{2} * (Order + 1)(Order + 2)$$

Higher values in the "order" box will result in a more accurate reconstruction, but will also result in larger feature vectors. The order number can be any integer greater than zero. Next, the user can input the measurement uncertainty associated with the imported data, this will be explained further in Section 4.3 of this document. After defining both parameters, or just the order number, the "Decompose" button should be pressed. If no measurement uncertainty is defined, then the program will use the value of the representation error instead.

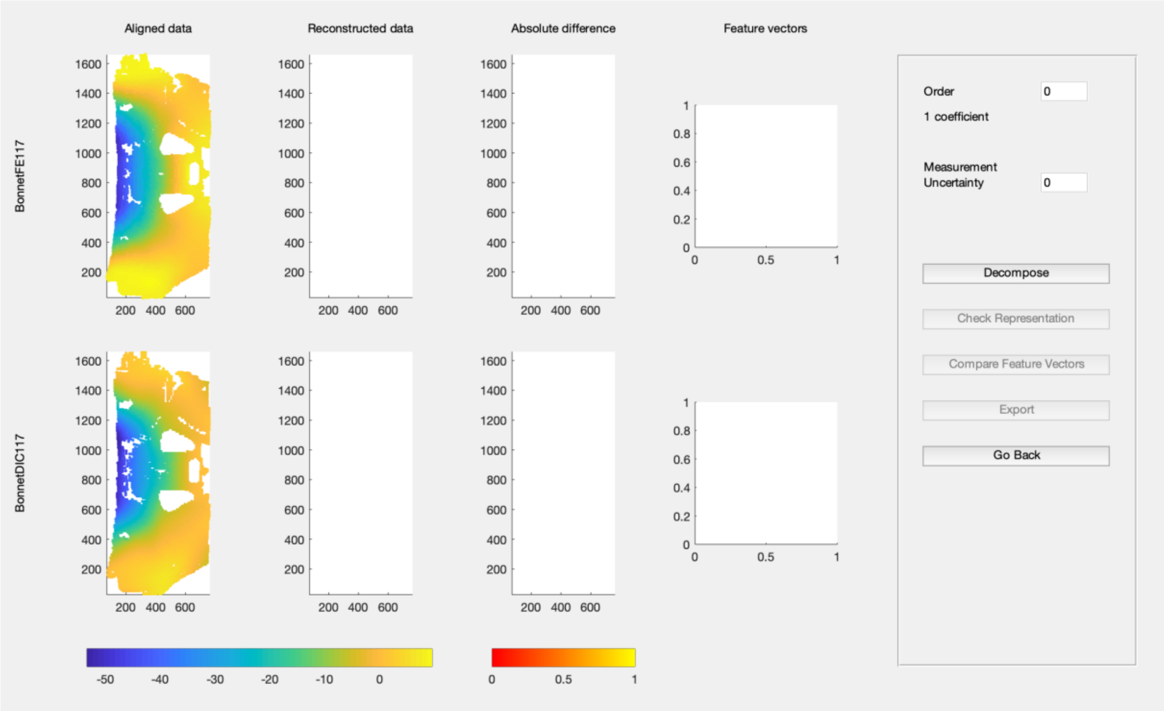

Figure 16: The "Data Decomposition" window after pressing "Decompose".

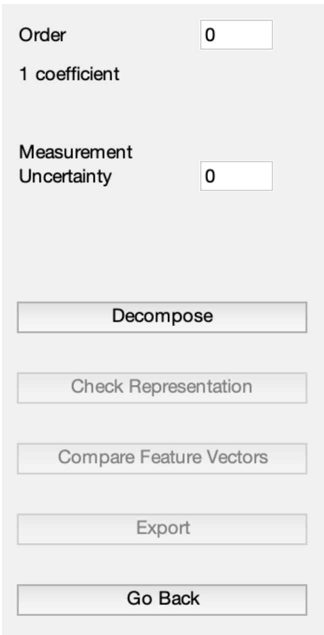

Figure 17: The control panel on the "Data Decomposition" window.

The data is decomposed using a new form of orthogonal decomposition based on the mathematical technique of QR-decomposition. QR Decomposition uses an algorithm called Householder Reflections to modify a set of kernel functions that is common to both data sets such that a set of orthogonal kernel function is obtained for each dataset. This is performed in such a way that the user isn't expected to make any complicated decisions about how the kernels should be orthogonalized. A flow chart summarising the processing of the data during the decomposition process is shown in Figure 18. For more details on how the decomposition is performed refer to [5].

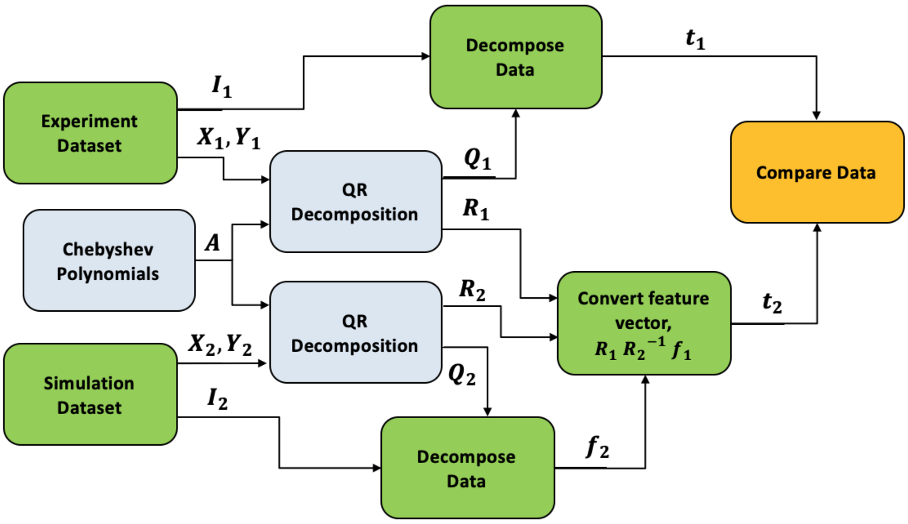

Figure 18: Flow chart to show how two datasets are compared in the program.

4.2 Viewing Data

After decomposing the data, all four graphs will now display data, shown in Figure 19. All buttons in the control panel will also be enabled, each button will be explained further within this document.

After inputting an order number and pressing decompose, the remaining three figures will display data. The figure titled "Reconstructed Data" displays the decomposed data. The colour-bar shown in the bottom left-hand corner shows the range of values in the dataset, and which colour corresponds to each of these values. The units of the colour-bar are the same as the units of the imported data.

The third figure, "Absolute Difference", plots the absolute difference between the original and reconstructed data. The absolute difference for each point has a colour representing it, shown by the colour bar below the graphs. The higher the absolute difference, the more yellow the point is. The quality of the reconstruction is calculated by comparing the reconstructed image with the original. The root mean squared residuals of the difference between the original and reconstructed images are displayed for each dataset above the "Absolute Difference" graphs. The representation error and overall quality of the representation will be discussed further in Section 4.3 of the document.

The final column of figures to be displayed are the feature vectors shown as a scatter plot. These will be further explored when looking at the "Compare Feature Vectors" button.

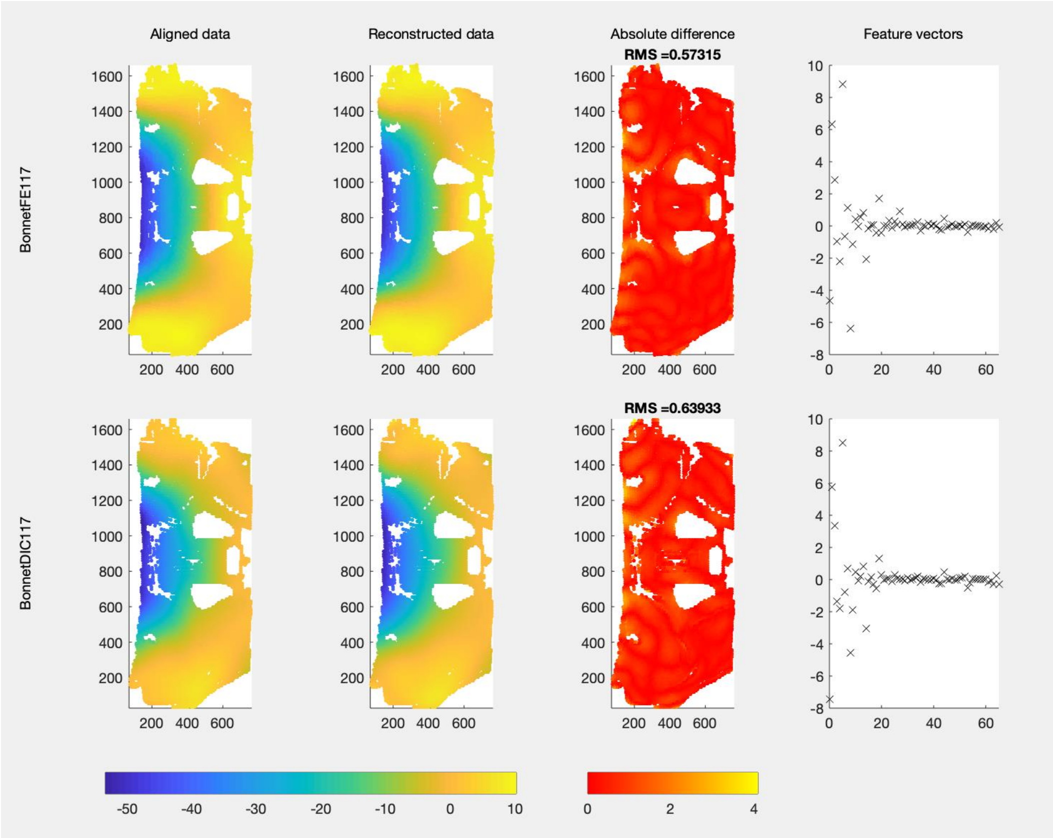

Figure 19: The "Data Decomposition" window after specifying a maximum order and decomposing.

4.3 Checking Representation

In the control panel, there is a "Check Representation" button. After inputting an order number, measurement uncertainty and then decomposing the data, the quality of the reconstruction can be checked by pressing this button thus loading a new window, "Checking Reconstruction", shown in Figure 20. The representation error, previously mentioned, is displayed alongside the Pearson correlation between the original and reconstructed data. The representation error can be defined as the root mean squared residual of the difference between the original and the reconstructed data, and can be calculated by:

$$Representation\ error = \sqrt{\frac{1}{n}\sum^{n}\left(\hat{\boldsymbol{I}} - \boldsymbol{I}\right)^2}$$

where n is the number of datapoints in the dataset, $\hat{\boldsymbol{I}}$ is the reconstructed dataset and $\boldsymbol{I}$ is the original dataset. If the quality of the reconstruction is not sufficient, then the user should close the window and return to the "Decompose Data" window, specify a higher order and decompose the data again. If the representation error is smaller than the measurement uncertainty then it is possible the order is too high and could be reduced. This means the number of terms in the feature vector would be reduced, resulting in faster computation and greater reductions in data size. The representation error is a good statistic for determining if the number of terms should be changed. For measurement data, the representation error should ideally be just below the uncertainty of the measurement system [3].

Whilst the representation error and Pearson correlation statistics offer an indication of the quality of the reconstruction of the entire image, they do not give an indication if there are local regions of the image where reconstruction is poor. To determine these locations, the difference between the original and the reconstructed images is calculated and datapoints with an absolute difference greater than three times the measurement uncertainty (previously entered by the user in “Data Decomposition” window) are identified as poorly reconstructed datapoints. If the value of the measurement uncertainty is not known, the application will identify poorly reconstructed datapoints as those with an absolute value greater than three times the representation error. However, it is more desirable to use measurement uncertainty as the program cluster size will be more stable [5]. The application then finds clusters, where a cluster is three or more adjacent datapoints, these are plotted and shown in red or yellow, depending on the size of the cluster, shown in Figure 20. The process of finding clusters is demonstrated in Figure 21. The number of data points in the largest cluster is finally divided by the number of data points in the image to obtain the largest cluster statistic. It is recommended that the largest cluster be no larger than 0.3% [3].

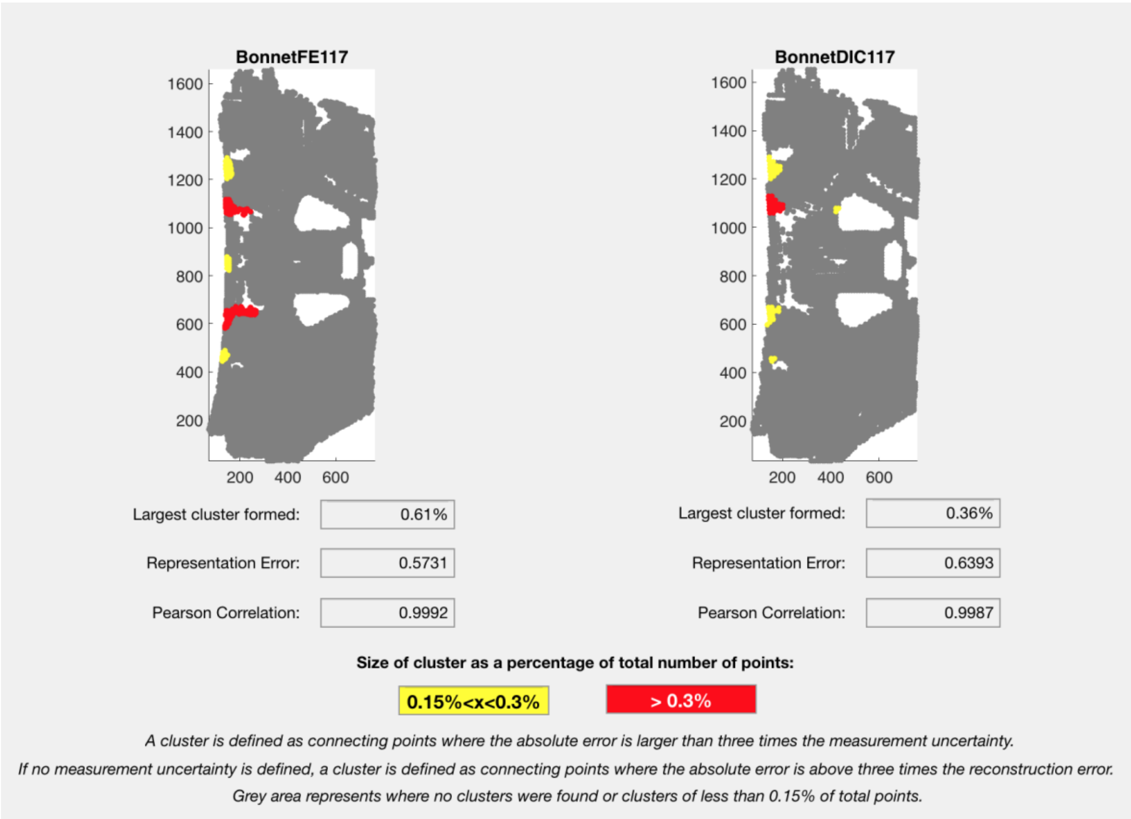

Figure 20: The “Checking Representation” window.

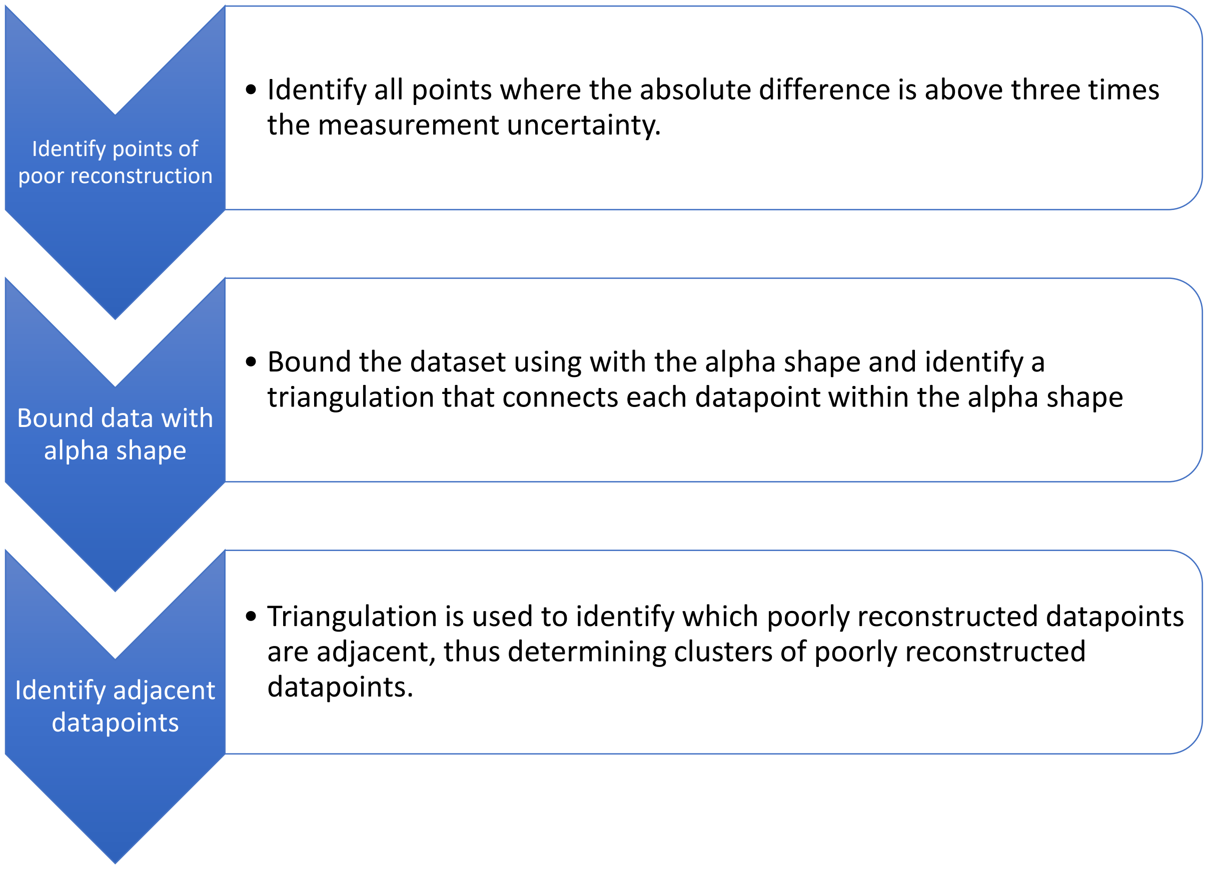

Figure 21: Process of finding clusters of poorly reconstructed data.

4.4 Compare Feature Vectors

After decomposing the data, the "Compare Feature Vectors" button will be enabled. Pushing this will open the "Compare Feature Vectors" window, shown in Figure 22. On the left-hand side there is figure showing the feature vectors represented by each dataset plotted against each other, with the first dataset on the y axis and the second on the x axis. Also plotted is a line given by, $s_{2nd} = s_{1st}$, indicating the expected position of the points if the two datasets are identical. If a measurement uncertainty was inputted in the "Decompose Data" window then two more lines will be plotted indicating the tolerances for the points. The equation of these lines is given by [3]:

$$s_{1st} = s_{2nd} \pm 2u(s_{2nd})$$

Where s_{1st} and s_{2nd} are the feature vectors representing the first and second dataset and $u(s_{2nd})$ is the measurement uncertainty of the second dataset.

At the bottom-right of the window is a display which indicates if all points lie within the tolerance lines. A computational model can be considered a good representation if all the plotted data points lie within the tolerance lines [3]. If any of the points are outside the tolerance lines then the data should be reviewed critically before considering refinement of the computational model [3].

The kernel associated with each point can be viewed interactively. By clicking on the desired point, a figure will display on the right-hand side of the window showing the kernel, shown in Figure 22. Beneath this figure is the kernel number (same as the coefficient number) and the order of polynomial used to create the kernel. The shape of the individual kernel can be used to identify features in the one dataset that are not predicted or present in the other dataset. This can be used to identify how data has changed or how a simulation differs from an experiment.

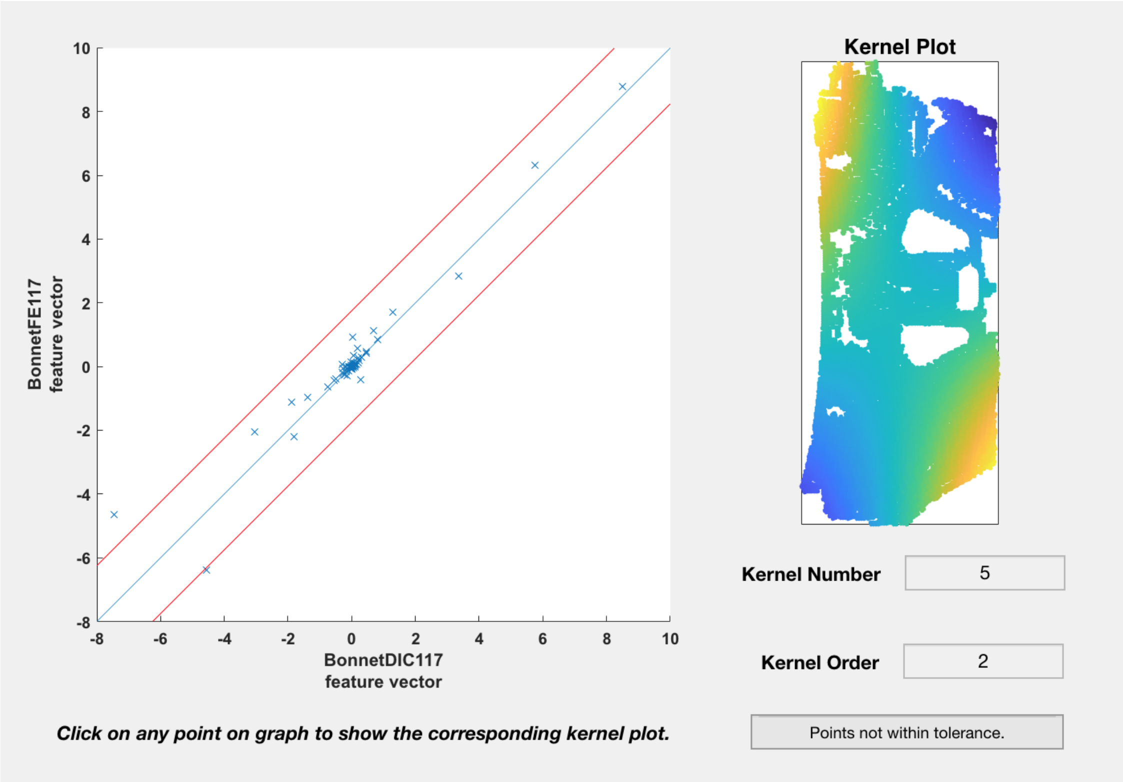

Figure 22: The "Comparing Feature Vectors" window.

4.5 Exporting Data

To export data, press the "Export" button shown in Figure 23. This will then display a file picker window, where the user can specify the save destination. There is an option to save as either a ".mat" file, text file or Excel file. The ".mat" file will contain a cell array containing all outputs from the decomposition process, Table 1 explains what each output is.

Within the file picker window there is also an option to save files as a "Text" file or an "Excel" file. Choosing either of these will save the feature vectors as a ".csv" file or ".xls" file, respectively.

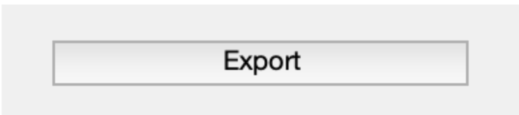

Figure 23: The "Export" button.

Table 1: All outputs and their descriptions that are saved to .mat files.

Output Name	**Description**
`maxOrder`	Order number inputted by user.
`dataName1`	The name of the first dataset.
`dataName2`	The name of the second dataset.
`inputX1`	x coordinates of the first dataset.
`inputY1`	y coordinates of the first dataset.
`inputX2`	x coordinates of the second dataset.
`inputY2`	y coordinates of the second dataset.
`inputData1`	Values of first dataset.
`inputData2`	Values of second dataset.
`overlapX1`	Overlapping x coordinates from first dataset.
`overlapY1`	Overlapping y coordinates from first dataset.
`overlapX2`	Overlapping x coordinates from second dataset.
`overlapY2`	Overlapping y coordinates from second dataset.
`overlapData1`	Overlapping values from the first dataset.
`overlapData2`	Overlapping values from the second dataset.
`recon1`	Reconstructed values from first dataset.
`recon2`	Reconstructed values from second dataset.
`featureVector1`	Feature vector of first dataset.
`featureVector2`	Feature vector of second dataset.
`representationError1`	Representation error (RMS) of reconstructed data from first dataset.
`representationError2`	Representation error (RMS) of reconstructed data from second dataset.
`measurementUncertainty`	User inputted measurement uncertainty.
`pearsonCorrCoeff1`	Pearson correlation between first dataset and its reconstructed data.
`pearsonCorrCoeff2`	Pearson correlation between second dataset and its reconstructed data.
`maximumCluster1`	Size of maximum cluster of poorly reconstructed data in first dataset as a percentage of the total pixels.
`maximumCluster2`	Size of maximum cluster of poorly reconstructed data in second dataset as a percentage of the total pixels.
`xCoords1Clusters`	Cell array containing the x coordinates of the poorly reconstructed clusters of data from the first dataset.
`yCoords1Clusters`	Cell array containing the y coordinates of the poorly reconstructed clusters of data from the first dataset.
`xCoords2Clusters`	Cell array containing the x coordinates of the poorly reconstructed clusters of data from the second dataset.
`yCoords2Clusters`	Cell array containing the y coordinates of the poorly reconstructed clusters of data from the second dataset.

4.6 Go Back

The final button on right of the “Decompose Data” window is “Go Back”, shown in Figure 24. Pushing this will take the user back to the “Align Data” window, it will display the two datasets as they were aligned before progressing to the “Decompose Data” window. The controls can be used as described previously, allowing the user to either realign the data or load in new datasets without having to restart the program.

Figure 24: The “Go Back” button.

5. References

1. Mukundan R, Ong SH and Lee PA, (2001) Image analysis by Tchebichef moments, IEEE Transactions on Image Processing, 10(9), 1357-1364

2. Sebastian C, Hack E and Patterson EA, An approach to the validation of computational solid mechanics models for strain analysis, Journal of Strain Analysis for Engineering Design, 48(1), 36-47, 2013

3. European Committee for Standardization, (2014) CWA 16799:2014, Validation of computational solid mechanics models

4. Christian WJR, DiazDelaO FA, Patterson EA, (2018) Strain-based damage assessment for accurate residual strength prediction of impacted composite laminates", Composite Structures, 184, 1215-1223

5. Christian WJR, Dean AD, Dvurecenska K, Middleton CA, Patterson EA, (2021) Comparing full-field data from structural components with complicated geometries, Submitted