Image Reconstruction Software Guide

A. Mather

February 9, 2006

Contents

Appendix A

Filtered Backprojection GUI

A.1 Introduction

The following document aims to provides a beginners user guide to the filtered back-projection graphical user interface (FBP GUI) developed using the Java programming language from Sun Microsystems [WebSun]. The FBP GUI application provides a simple graphical interface for loading single or multiple slice sinogram input files, executing the FBP algorithm with a choice of related options and then outputting the resultant reconstructed images to a number of different file formats.

FBP GUI is a cross-platform application, any operating system providing an implementation of the Java Runtime Environment should be able to run this software.

A.2 Installation and setup

A.2.1 Java installation requirements

To run the FBP GUI program, firstly a Java Runtime Environment (JRE) or Java Software Development Kit Standard Edition (J2SE SDK) version 1.4.x or later must be installed on the local computer or network. The JRE is sufficient to simply run the software while the J2SE SDK is required to recompile the software from the base source code. JRE and SDK packages and installation instructions for most popular operating systems can be downloaded freely from the Java homepage [WebSun] if it has not been installed as standard.

An additional add-on package, Java Advanced Imaging API (JAI) may also be required if problems are experienced in saving output to imaging formats such as png. In most cases this package will be included with the basic Java installation and no further action is needed. Full JAI download and installation instructions can be found at: http://java.sun.com/products/java-media/jai/

A.2.2 Running FBP GUI

To start the FBP GUI program simply type the command `java runFBP_GUI` at a command prompt or xterm window from the directory containing the `runFBP_GUI.class` file. Alternatively type `java -jar runFBP_GUI.jar` from the directory containing the `runFBP_GUI.jar` file[1]. Some operating systems may also execute the program by simply double clicking the jar file icon.

A.3 General program layout

Figure A.1 shows the general user-interface, the exact appearance may vary slightly from that shown depending on the computer display settings and operating system. The interface is separated into four distinct sections:

1. **Load Sinogram Data:** The input data selection panel.
2. **Run FBP Algorithm:** The FBP algorithm settings and execution panel.
3. **Output Options:** Output data selection panel.
4. **Reconstructed Image:** Display panel for the reconstructed image(s).

[1] A jar file is simply a single file containing a collection of files packaged together and compressed to simplify transferring java programs, especially over the Internet.

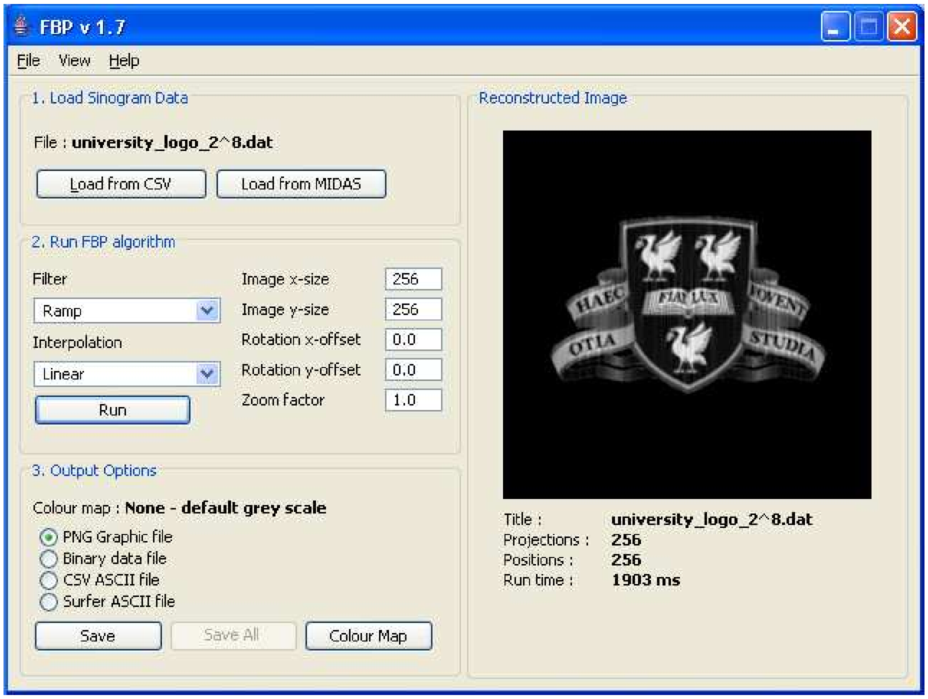

Figure A.1: Screenshot of the FBP GUI applications user-interface

The following sections will explain in more detail the purpose and functionality of each of these panels.

A.4 Loading sinogram data

The input data, the sinogram can be loaded into the application as either an ASCII file or a 2D spectra binary file generated in Eurogram format by the Multi Instance Data Acquisition System [MIDAS] software program.

Projections are expected to be equally spaced and span the region $0^o \leq \theta < 180^o$, the first projection is assigned as the 0^o angle. The positions within each projection should also be equally spaced. It is recommended that the number of these be equal to 2^n where n is an integer value, due to the Fast Fourier Transform (FFT) used by the reconstruction algorithm. The FBP algorithm will automatically zero-pad each projection up to the nearest 2^n value where required.

ASCII files should contain one projection per line and each element within a projection should be separated by either a tab, space or comma character.

MIDAS spectra are assumed to be the same size in both dimension as the sinogram itself i.e. they should not have blank elements. The sinogram should be orientated with the x-axis containing the positions and the y-axis being the projections.

Multiple files can be selected but it should be noted that the FBP options mentioned in the next section are applied to all files. All selected files should therefore contain the same number of data elements. Multiple file selection is ideal for sets of sinograms forming slices along the axial direction of the image space i.e. 2.5D image reconstruction.

A.5 Selecting FBP options

The FBP options selected highly influence the final reconstructed image, this section will explain the role of the different options found in the middle panel of the FBP GUI application and how they relate to the reconstructed image.

A.5.1 Filter selection

Ths is possibly the most important option, the filter refers to the 1D filter applied to each projection in the frequency domain before the filtered sinogram is backprojected. Although an in-depth discussion is beyond the scope of this document mathematically the ramp filter is the ideal filter to deconvolve the 1/r blurring inherent from back projecting sinograms. Real data however will contain varying degrees of high frequency noise and a ramp filter will strongly amplify this noise. Hence filters with different level of suppression for high frequency components can often provide optimal performance for real data. Figure A.2 graphically displays the frequency response of the different inbuilt filters up to the Nyquist frequency.

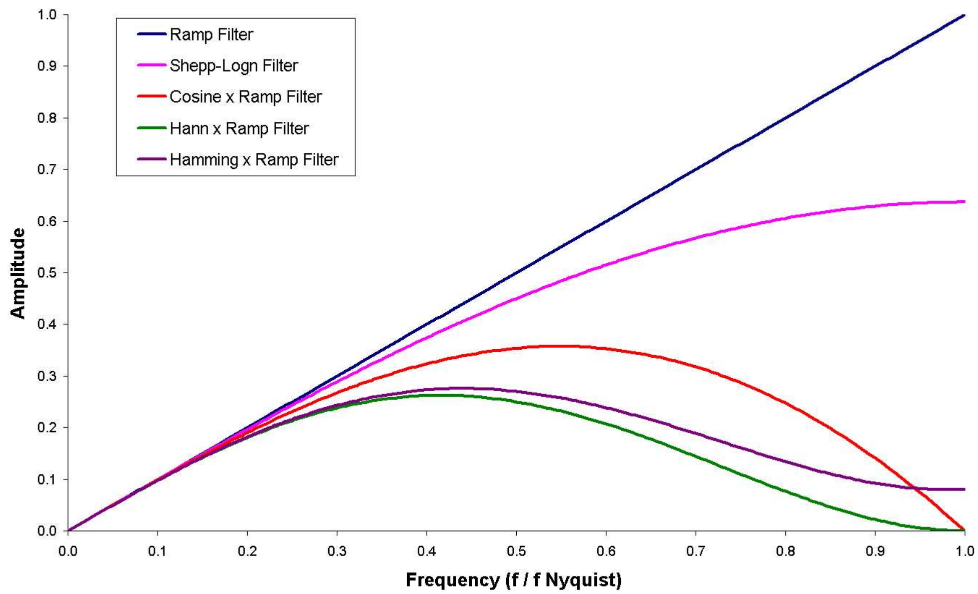

Figure A.2: Filter response as a function of frequency

A.5.2 Interpolation method

During the backprojection stage the sinogram data measured in a polar co-ordinate system is mapped to the Cartesian based pixel grid requiring the use of a 1D interpolation method. Several interpolation methods are provided from simple nearest neighbour to the more complex cubic spline although in most cases linear interpolation is more than adequate.

A.5.3 Image size

The size of the reconstructed image in pixels. The x and y values can be set separately but it is recommended to use the same value for both and therefore produce square dimensioned images.

A.5.4 Rotation offset

The two rotation offset values are used for the centre-of-rotation (COR) correction of the system from the centre of the detector system in the x and y axis. The units for these values are scaled to the distance between detector bin elements in the projection and are independent of zoom value settings.

A.5.5 Zoom

The zoom value is self explanatory from the name but understanding of how the value relates to the image size is important to ensure correct interpretation of image scale. If the input data contains for example 64 equally spaced elements per projection and the size of the reconstructed image is 256 pixels then only the central quarter of the image will represent areas measured by the data. To ensure the data spans the entire image space then a zoom factor of 256 / 64 or 4.0 would be required.

A.6 Running the FBP algorithm

Once the input file(s) have been loaded and all relevant FBP settings selected the algorithm is run by simply clicking the button marked 'Run' in the central panel. The algorithm may take under a second to several minutes depending on the number of lines of response, number of input files, FBP options selected and computer performance. During this time the application may seem unresponsive. Once complete the resultant images will be displayed in the image reconstruction panel on the right. If multiple input files exist then back and forward buttons will be displayed under the image area to move between the different images.

A.7 Output results

The output data panel contains controls for saving reconstructed images to file. Four different file formats are currently supported:

Tag	Replaced With
<index>	One based index value of each image
<index-1>	Zero based index value of each image
<title>	Name of the original input file from which the image was produced.

Table A.1: Available tags for use in multiple image output filenames.

1. **Graphic:** Output as a Portable Network Graphics (PNG) file.
2. **Binary:** Simple binary file format containing just the raw data values as double precision numbers.
3. **CSV ASCII:** Text file with values being separated by comma characters.
4. **Surfer ASCII:** Text file format readable by the analysis application Surfer

If multiple input files have been reconstructed then the second button titled 'Save All' will be enabled. Use of the 'Save' button with multiple images will result in the image currently displayed in the image reconstruction panel being saved to disk only. For multiple file output special tags can be used within the output filename path. Table A.1 lists the available tags and the content they are replaced by before output.

A.7.1 Colour Maps

By default reconstructed images are displayed using a linear greyscale normalised to the range of the data values. Other colour maps can be selected to provide different colour schemes when outputting to graphical format (and displaying within the image reconstruction panel). A number of different colour maps have been provided with the FBP GUI application. The user is encouraged to view these files in a text based editor to understand the simple format used and create new colour maps if they are required.

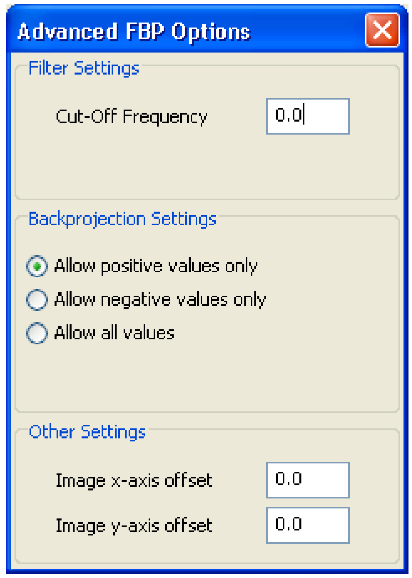

Figure A.3: Screenshot of the advanced options dialog

A.8 Additional features

A.8.1 Advanced options

The advanced options dialog can be activated from the file menu and is shown in figure A.3. The menu provides additional options relating to the FBP algorithm that are not displayed in the main application window.

Filter cutoff is used to limit the maximum frequency produced by the FFT, by default the value of zero is assigned as being the Nyquist frequency.

Backprojection settings provides control of the type of values allowed in the final image after the FBP. In PET data the option should be set to require positive only values while in transmission data sets the negative only option may be more appropriate. The application in general assumes higher values relate to higher concentrations while in transmission data higher values would represent lower concentrations along that line-of-response.

The x and y axis offset values simply act as a translation of the imaging area relative to the centre-of-rotation of the system.

A.8.2 Relative colour scaling

When multiple input files are selected by default the display colouring is scaled individually for each reconstructed image. The view menu contains an option which can toggle whether the set of images are scaled individually or relative to the whole set. This can be useful when comparing the relative intensity of different slices in a multiple slice data set.

Appendix B

Sinogram Generation Software

B.1 Introduction

During the development of the FBP algorithm (Appendix A) input sinogram data was required in order to test the algorithm was responding correctly. A set of secondary software programs were developed to generate simulated sinograms to provide test input. The following document outlines the basic functions of these programs and their usage.

The programs were again developed in the Java programming language and use a simple command-line driven interface. Two different methods where developed, the first generates a sinogram from an input text file containing a list of ellipse properties and relative intensity. The second calculates a sinogram from any grayscale image file (portable network graphic) with black pixels assumed to be zero intensity and white maximum intensity.

B.2 Ellipse based method

`GenerateSinogram1` and its accompanying command-line interface program `runGenerateSinogram1` can calculate the expected sinogram for a given input file of ellipses, which specify the source geometry and relative strength. To run the program from an xterm or command prompt window in the directory containing the `runGenerateSinogram1_CommandLine.class` file enter the following command:

```
java runGenerateSinogram1_CommandLine [Arguments]
```

Where the arguments should be:
`inputPath outputDataPath No_of_Projections No_of_Positions Sigma_of_Noise [outputImagePath outputSourceDistribution]`

Input and output paths are required and are relative to the current location. The input path is the location of the ellipse list data file, the output is where the program output will be saved (ASCII CSV format). The number of projections and number of positions per projection are also required along with a value for the noise level applied to the image. If no noise is required use a sigma value of zero. Noise is generated using a Gaussian spread function. An image of the sinogram in png format is an optional output, if required add a path for this to the arguments list (outputImagePath). An estimated grayscale image of the source distribution obtained from the ellipse data file can also be optionally produced in the same way (outputSourceDistribution). Input ellipse data should be stored in an ASCII file with a single ellipse per line. Each line should list the required properties for the ellipse separated by a space character. The format of this file is as follows:

x-offset y-offset major_axis minor_axis rotation_angle(degrees) relative_strength
.
.

The output sinogram ranges from -1 to +1 in both axes so the ellipse data major and minor axis should not exceed these values. The Shepp-Logan phantom is provided as an example ellipse data file and users are suggested to study this for further clarification of the required structure.

B.2.1 Example usage

`java runGenerateSinogram1_CommandLine mydata.dat result.dat 256 512 0.0 sinogram.png`

B.3 Image based method

Sinogram generation from an image file works in much the same way as outlined in the ellipse method and users should refer to the information provided there. To run the program from an xterm or command prompt window in the directory containing the `runGenerateSinogram2_CommandLine.class` file enter the following command:

```
java runGenerateSinogram2_CommandLine [Arguments]
```

Where the arguments should be:
`inputPath outputDataPath No_of_Projections No_of_Positions Sigma_of_Noise [outputImagePath]`

The greyscale image is used as an intensity map, with white being the highest intensity and black being the lowest or zero. It is recommended to use a png file with a full 16 million colour palette even though only greyscale is allowed in the image. Input images must be a square, i.e. pixel width = height. Due to the implementation method `GenerateSinogram2` does not support more positions than the pixel-size of the input image, no limit exists on number of projections.

B.3.1 Example usage

`java runGenerateSinogram2_CommandLine myimage.png result.dat 256 512 0.0 sinogram.png`

References

[MIDAS] Website, http://nnsa.dl.ac.uk/MIDAS/

[WebSun] Website, http://java.sun.com/

[Tai01] Y.C. Tai *et al.*, Phys. Med. Biol. **46**, (2001), 1845.