Purpose: Learn how to perform computed tomography (CT) reconstruction of biomedical projection data. Given a data file containing fan-beam x-ray data, create a program to perform CT reconstruction on the data. Refer to the lesson slides, the handout (from Rosenfeld and Kak), your text, and the course web site for help.

Procedure: For *each* MATLAB m-file you write, be sure to comment the code sufficiently, and include initial comments as a crude help system. That is, if a user types "help" followed by your m-file name, comments will print on the screen which tell the user what the program does, what the proper syntax is to use the program, and any other useful tidbits. No Image Processing Toolbox programs should be used.[1] You may use the Signal Processing Toolbox program `hamming` if you wish.

- ➪ You will be performing basic computed tomography (CT) as part of this project. You will use the CT algorithms with the given data to create a single 1-D plot of x versus amplitude (i.e., gray level) for a single value of y, plus one full 2-D image reconstruction. Note that the maximum range of gray levels for a standard Shepp-Logan head phantom is from 0.0 to 2.0, so a scaled display is implied for the 2-D image to "fill" the typical range of 0 to 255 for pixel values. It's almost certain you will also need to apply some image enhancement techniques (see Chap. 3 in your text) to the 2-D reconstruction to obtain an acceptable image; be sure to fully explain any and all such postprocessing steps in your report. The 1-D plot should *not* be scaled and should not exceed the 0.0 to 2.0 range, and zooming in to a range of 0.95 to 1.05 may help you see the detail you seek.

  To get you started you will have available to you the following files.

  - A file containing fan-beam projection data (in ASCII format) for a Shepp-Logan head phantom (see Fig. 4 in your handout). The file name is `FANAGL.DAT`.
  - An associated "read me" file that describes the data. The file name is `FANAGL_readme.txt`.

- ➪ You can find these files on the course web site, in Assignments subdirectory of the Files area.

- ➪ This project will have only one part:

  Write your own MATLAB program(s) to process the fan-beam data such that you generate at a minimum two figures (that's right, just two figures; this must be easy!). Your two figures should essentially be the fan-beam equivalents of Fig. 17(a) and Fig. 17(b) in your Rosenfeld and Kak handout. As another point of comparison, your full 2-D image reconstruction of the Shepp-Logan head phantom should also be similar to Fig. 5.49 on p. 440 of your Gonzalez and Woods text, reconstructed to a spatial domain resolution of 256×256 pixels or higher. Note that if your program is working correctly, your reconstruction should have a level of detail somewhere between Fig. 5.49(b) and Fig. 5.49(c) in Gonzalez and Woods. Regarding the 1-D plot, your equivalent of Fig. 17(b) in the handout should correspond only to the reconstructed values (*not* the labeled true values), evaluated at $y = -0.609$ rather than at $y = -0.605$, and should have a resolution along the x-axis of at least 256 points. If you wish to show other figures in your Results section you are free to do so, but it is not required. You are free to use any combination of frequency domain techniques and spatial domain techniques to create acceptable figures, but be sure to fully explain and justify your methods. *Hint:* most students have found it easier to get the 1-D reconstruction working first, then modify that slightly to get the 2-D reconstruction.

[1] While you can't use them in your project, it might be instructive to investigate Image Processing Toolbox programs such as `phantom`, `radon`, `fanbeam`, etc.

Questions/Discussion: The write-up for this project report should be concise, using wording and formatting suitable for an IEEE technical journal.[2] Your report should provide a short background of the CT technique (with any appropriate figures and/or equations), explain succinctly how you came up with your program, what important concepts or lessons that you learned in the process, and intelligently *interpret the quality of your results* compared to the results shown in the associated figures from the Rosenfeld and Kak handout and the Gonzalez and Woods text.

Turn in: For this Project, turn in (as one or more e-mail attachments sent to me as part of a single e-mail message):

➪ An electronic project report (as a PDF file regardless of the program used to create the report). Name your PDF file "`Last1_Last2_proj01.pdf`" please, where "Last1" is the last name of team member 1, and "Last2" is the last name of team member 2. **Be sure to follow the format** given on the course web site (see the `Admin` section of the course web site). The most common reason students needlessly lose points is due to completely avoidable formatting errors! There is an example report for you to view on the course web site that shows the proper formatting,[3] as well as a LaTeX style file and skeleton report for you to use if you wish.

➪ Any MATLAB m-files you created for this project. A top-level m-file that shows the syntax and the order in which you called all your other m-files to complete the project should be included. There is no need to send me any data or image files that you generated. I'll run your m-files and generate your images myself as needed. If for some reason I need additional files from a particular student, I'll request them separately.

Be sure to thoroughly read these instructions. The second most common reason students needlessly lose points is a failure to follow the directions, and the penalty for that can be rather harsh.

Be sure to **critically assess** your results (and compare to other teams). The third most common reason students needlessly lose points is to turn in obviously erroneous results and either not mention anything about it, or, even worse, pretend (or actually think) that the results are correct. The penalty for this can also be rather harsh.

Final thoughts. While there are only two figures required in the Results section for this assignment, don't let that fool you. It typically takes each team several iterations of their code before everything works to their satisfaction, so start early! Compare your intermediate and final results with the other teams. If you paid attention in class, you should know how to avoid most of the common pitfalls. Finally, don't wait until the last minute after the code is all finished to start writing the report; that's how sub-standard reports are created. **Remember: you are trying to demonstrate that you can write in a professional manner and communicate a technical topic clearly and unambiguously.** Write as if you would be willing to submit this to a professional journal, whereupon it would undergo peer review.

❄❄❄ Enjoy... ❄❄❄

[2]Your team's project report must follow the format referenced on the course web site. Be sure to use proper grammar and logical, organized sentences. **Any equations in your report should be typeset with an equation editor** (or use the math mode of LaTeX). All figures and tables should have descriptive, numbered captions.

[3]I'm trying to give you practice similar to what's needed to submit a paper to a high-quality journal. They have zero tolerance for formatting mistakes.