BE280A, Principles of Biomedical Imaging 10/29/08
Fall Quarter 2008

BE280A Midterm Project Assignment

Due Date: Completed project (hard copy) is due at 12 pm (noon) on Monday, November 10, 2008 – please turn in at Room 1001 of the fMRI Center between 11 am and noon. In addition to the hard copy, please submit a PDF version of the report via e-mail by 2 p.m. on that day. For full credit, the subject line of your e-mail should read **BE280A08 Midterm Project**. The report filename should follow the following format ct_{initials of partner 1}_{initials of partner 2}.pdf – e.g. ct_lk_pb.pdf. The oral exam component of the project(10 minutes per student) is tentatively scheduled to occur on Friday November 14, 2008 from about 9:30 to 2 p.m.

Guidelines:

1) Select a partner to work with (there are 18 registered students, so there will be 9 groups).
2) Discussion of **general ideas** is encouraged between groups, however, each report submitted should reflect each group's own understanding of the material. Significant discussions with other groups should be given appropriate credit.
3) An electronic copy of the MATLAB code should be submitted with the PDF of the report. The code file should be named in a similar fashion to the *.pdf file, except with a *.m extension.
4) Use a word-processing program to write the report, including all equations (no handwritten reports! Use an equation editor.). Neatness and clarity of exposition will play a **significant** role in the grading of the report. Other grading criteria include technical correctness and originality.
5) You may use external references (print or electronic). If you do so, please cite them at the end of your report.
6) Title and label the axes on all plots and images.
7) In addition to answering the questions below, please provide additional details and original insights as appropriate. If you noticed something interesting or learned something new in doing this project, please comment on that.

Description of Problem
Consider a square object of width 20mm centered at (-10mm, -10mm). The attenuation coefficient of the object is 1 mm^{-1}. The object is imaged with a 1st generation CT scanner with a beamwidth of 1mm. The desired FOV is 80 mm.

1. (5 pts.) Label your e-mail, PDF file, and MATLAB code as indicated above.

2. (10 pts.) Make sure that your MATLAB code executes without errors. In addition, all figures should be generated simply by typing the name of the main MATLAB code provided.

3. (5 pts.) Determine the appropriate detector size Δr and the number of radial samples needed to span the FOV. Assume that the middle two samples are acquired at coordinates of $-\Delta r/2$ and $\Delta r/2$.

4. (5 pts.) Determine the number of angular samples required. For the simulations, round this up to the nearest multiple of 4.

5. (10 pts.) **Ignoring** the finite beamwidth, derive an analytical expression for the projections as a function of angle. Using your expression, use MATLAB to generate a sinogram of the object. Note that the sinogram should cover the angles from 0 to $(N-1)\pi/N$ where N is the number of angular samples. Comment on the features of the sinogram – i.e. why does it look the way it does?

6. (30 pts.) Now consider the effect of the finite beamwidth on your projections. Provide analytical expressions for the projections at 0, 30, 45, and 60 degrees. For all other angles, you may derive analytical expressions or use MATLAB to compute the projections. Your answers should match your analytical expressions at the angles indicated above. If you use MATLAB, describe in detail how you are doing the computation. Use your projections to generate a sinogram of the object. How does this sinogram compare to the one from Part 5? **HINT:** Start with the projections derived in Part 5 and think about how the finite beamwidth affects the projections.

7. (10 pts). Use MATLAB to generate the backprojection of the object using the projections from Part 6. An easy way to do this is to backproject at a projection angle of 0 degrees and then use the MATLAB function **imrotate** to rotate each backprojection. Use the bilinear and crop options in imrotate. Comment on the features of the backprojected image.

8. (20 pts.) Filter the projections from part 6 using both (i) a Ram-Lak filter and (ii) a Hanning windowed filter. Plot and compare the filters and the filtered projections at a projection angle of 0 degrees. Repeat this for a 45 degree angle. Describe the design and the practical implementation of the filters. You will want to use the MATLAB functions conv.m or filter.m.

9. (10 pts.) Now backproject the filtered projections. Comment on features of the filtered backprojection images and on the effect of the choice of filter. Compare to the backprojected images. How does your reconstructed object compare to the original object? Discuss the normalization of the images that is required to get correct estimates of the attenuation coefficients.

10. (10 pts) Using the code you have developed, repeat parts 7 through 9 with the projections from Part 5. How do the reconstructed images compare with those obtained in part 9?

11. (20 pts.) Experiment with acquiring fewer samples in the radial and angular dimensions (e.g. take every 2nd sample; take every 4th sample). Comment on what you see. Does changing the bandwidth of the filters help at all? Explain your answer.

12. (30 pts) Oral exam component (10 minutes). Be prepared to explain in detail all aspects of the project. Exams will be performed on an individual basis, with partners scheduled to take their exams in contiguous time slots.

HINTS ON USING MATLAB

When using MATLAB, it's important to keep in mind that MATLAB really likes vector and matrices. So as much as possible, you want to use vectorized operations (see below). In general, it's a good idea to minimize the use of looping structures (e.g. for loops, while loops). You will need to use some loops for the project, but try to avoid them where possible.

Here are a few examples of how to best use MATLAB.

Defining a RECT function

BAD:
```
i = 1;
for t = -10:0.1:10;
   if t < -1 | t >1
     rect(i)=0;
   else
    rect(i) = 1;
  end
i = i+1;
end
```

Good:
```
t = -10:0.1:10;
rect = ((t <= 1) & (t >=-1);
```

Defining a two-dimensional Gaussian function;

Bad:
```
i = 1 ;j = 1;
for x = -10:.1:10;
   for y = -10:.1:10;
      g(i,j) = exp(-x^2 –y^2);
      j = j + 1;
   end
 i = i + 1;
end
```

Good:
```
[x,y] = meshgrid(-10:.1:10,-10:.1:10);
g = exp(-x.^2 –y.^2);
```

You can find more information on the web, such as:
http://web.cecs.pdx.edu/~gerry/MATLAB/programming/performance.html