ScanPyramid Midterm Project.

Background.

In this project, we are modeling the project of muon imaging of the great pyramid. In muon imaging, detectors are placed around the structure to be investigated. Muons arrive at random angles as cosmic rays. However, when the muons pass through dense substances (such as rock), some of them are absorbed or deflected.

Experimenters can record the number of muons that arrive on a given flight line and use these counts to gradually build up an integral of the density of the structure along the flight line. Given enough of these flight lines, the internal structure of the object can be recomputed.

The data.

Since this is an exercise, we're going to use a two-dimensional model of a pyramid instead of a 3d model. The density of the pyramid is a function $\mu(x,y)$ where $x \in [-1500,1499]$ and $y \in [-1273,1273]$. This rectangle is divided into an array of 3000 x 2547 square pixels, where $\mu(x,y) = 1$ if that square is occupied by solid material and $\mu(x,y) = 0$ if the corresponding square is empty.

We will parametrize the space of possible flight lines in the plane by two coordinates θ and d, where the line (θ,d) has equation $d = x\cos\theta + y\sin\theta$. The lines will have θ values between $-\pi/2$ and $\pi/2$, and d values between plus or minus

```
Sqrt[3000^2 + 2547^2] / 2
```

$$\frac{3\sqrt{1720801}}{2}$$

```
N[3 Sqrt[1720801] / 2]
```

```
1967.69
```

which is the diagonal of the image (in pixels). The coordinate system is centered at the center of the image. In these coordinates, the pyramid defines a density function $\mu(x,y)$ which is 1 for pixels within the solid portion of the pyramid and 0 for empty pixels.

For a selection of lines with uniformly random θ and d coordinates, we've computed the integral

$$R(\theta,d) = \int_{-\infty}^{\infty} \mu(-t\sin\theta + d\cos\theta,\ t\cos\theta + d\sin\theta)\, dt$$

that is, the total density along the line (θ,d).

The data has been normalized so that the largest value of the integral is 1 and the smallest is 0. The file containing the data is located on the course webpage, and called Pyramid-Scan-Data.csv.

It contains a gracious plenty of data (half a million samples), so you probably want to test your methods on a smaller sample of the data before running on the entire dataset. As is so often the case with data, the columns in the file are unlabeled. You'll have to look at the values in each column to decided which is which.

Mission.

Your mission is to recover the density function μ(x,y) as well as you can from the information in Pyramid-Scan-Data using one of the methods listed below.

Full Credit Methods.

There are a number of known methods for recovering the density map μ(x,y) from the R(θ,d) function. You must implement one of these using numerical integration and differentiation code *of your own devising*, based on the methods we've learned in class.

The Direct Formula.

$$\mu(x,y) = \frac{-1}{2\pi^2}\int_{-\pi/2}^{\pi/2}\int_{-\infty}^{\infty}\frac{\partial R(\theta,d)}{\partial d}\,\frac{1}{d - x\cos\theta - y\sin\theta}\,dd\,d\theta$$

Notice that the function R(θ,d) is 0 (and hence has partial derivative 0) for d values larger and smaller than half the diagonal of the original image, so the limits of integration on the inner integral amount to "integrate over all the nonzero values".

The Fourier Method.

Compute the Fourier transform

$$\hat{R}(\theta, d) = \int_{-\infty}^{\infty} R(\theta, x)\, e^{2\pi i x d}\, dx$$

Suggested first step. First, isolate all the data corresponding to a single θ value (for instance, $\theta = 0$, which refers to vertical lines). You should have enough different x values present to make a plot of the function R(0,x). What should this function look like? You should have enough data to construct a pretty good interpolation of R(0,x), and compute the Fourier transform of that function. Depending on how you interpolate, you may be able to get an explicit formula for the Fourier transform function $\hat{R}$. But you should definitely be able to get values at enough points to make a plot of $\hat{R}$(0,d).

Suggested second step. If you're confident in your computation for $\hat{R}(0, d)$, automate the work above and construct a table of data for $\hat{R}(\theta, d)$ for a table of θ values. This will give you a new dataset for the final integration. (It's up to you and your computer exactly how big this dataset is. But I'd eventually make it pretty large.)

Define a new function on the u-v plane in polar coordinates, where u = d cos θ and v = d sin θ by

$F(u,v) = F(d\cos\theta, d\sin\theta) = \hat{R}(\theta,d)$

Then it turns out to be the case that

$\mu(x,y) = \iint F(u,v)\, e^{2\pi i (x u + y v)}\, du\, dv$

where the integral is taken over everywhere that F(u,v) is nonzero.

The cutoff function method.

Amazingly, it's ***also*** true that

$\mu(x,y) = \lim_{z\to 0} \int_{-\pi/2}^{\pi/2}\int_{-\infty}^{\infty} R(\theta, t - x\cos\theta - y\sin\theta)\, \Gamma(z,t)\, dt\, d\theta$

where

$\Gamma(z,t) = \frac{1}{\pi z^2}$, if $-z \le t \le z$

and

$\Gamma(z,t) = \frac{1}{\pi z^2}\left(1 - \frac{1}{\sqrt{1 - z^2/t^2}}\right)$, for $|t| > z$.

Partial Credit Methods.

For partial credit, you can use Mathematica's built-in integrators instead of writing your own. In partial, you can use Mathematica's built-in Fourier transform methods to compute the Fourier transforms in method 2.

Implementation notes.

One of the largest issues in this kind of integration is the question of how to deal with missing data values. Of course, 1-d polynomial interpolation is your friend here, but you're encouraged to read the Mathematica tutorial "Approximate Functions and Interpolation" and consider your other options as well.

You're going to have to synthesize data on a rectangular grid in θ, d coordinates in order to use any of our methods going forward to approximate the numerical derivatives and integrals. (Why?)

Start by recovering ONE pixel value of $\mu(x,y)$ that you can guess before leaping into computing millions of them.

The Piecewise function in Mathematica can be very helpful when you're working with functions like $\Gamma(z,t)$ which are defined, well, piecewise.

If your method is slow, it may be useful to know that the machines in the classrooms are pretty good and have Mathematica installed. Don’t burn them out; if they’re running, you need to be there.

Rules.

The rules are those of a term paper. This is not an exam. Your group may consult any inanimate source of information (that is, books, Mathematica documentation, Google searches, but not crowdsourcing sites such as StackExchange). No people, except for me.

Data will be posted by Midnight on Tuesday October 24. Completed Mathematica notebooks are due by email by Midnight on Tuesday November 7.

One Mathematica notebook per group.

Your notebook should discuss the numerical methods you tried and developed, document your code, cover the material in any reading or research you did on the problems involved, and finally present a single solution as a bitmapped image which is 3000 x 2547 pixels. You may also describe any areas inside the pyramid which you believe to contain voids even if you can’t reconstruct an entire image. You are allowed to use the Mathematica image processing functions to improve your results.

You may ask me questions about your term paper project after class and during office hours.

Acknowledgements.

Thanks to Blake Pinto for pointing out an error in the original dataset, to Project DANGEROUS WRENCH for pointing out that the formula for the Fourier transform was wrong. DANGEROUS WRENCH continued to help contribute edits and waypoints to the Fourier method.