

# EN250 - Medical Imaging Analysis

## Lab # 1: CT Simulation & Reconstruction

Due: Friday, Feb 18, 2011

### CT Simulation

1. Create some simple test shapes , *e.g.*, rectangles, triangles, circles *etc.* Save each shape as a binary image.
2. Write a function “rotateGrid” to generate a rotated 2D grid, (*i.e.*, a set of points at grid line crossings,  $(i, j) : i = 0 \dots width, j = 0 \dots height$ ), by a specified rotation angle  $\phi$ . The rotation matrix transforms  $(i, j)$  to  $(i', j')$  is

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix}$$

Make sure that the new coordinates  $(i', j')$  fall within the same range as the old coordinates  $(i, j)$ , see Figure 1, by rotating a larger grid and using only the points that lie within the new range to generate all possible points in rotated coordinates.

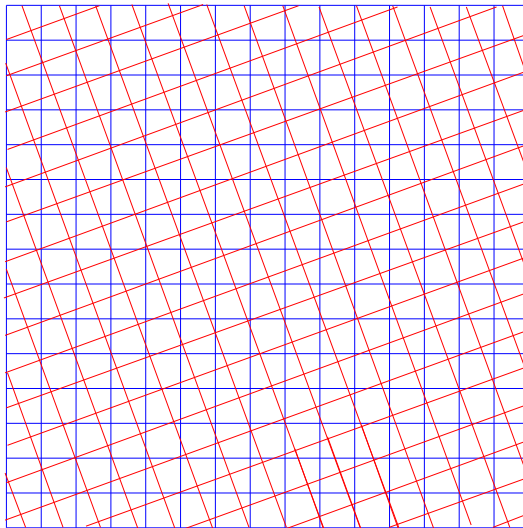


Figure 1: Original grid (blue) and rotated grid (red).

3. **Radon Transform:** Consider the synthetic phantom image defined by the function  $f(i, j), 0 \leq i \leq width, 0 \leq j \leq height$

$$f(i, j) = \begin{cases} \mu_1 & \text{if } \sqrt{(i - x_1)^2 + (j - y_1)^2} \leq R_1 \\ \mu_2 & \text{if } \sqrt{(i - x_2)^2 + (j - y_2)^2} \leq R_2 \\ \mu_3 & \text{if } \sqrt{(i - x_3)^2 + (j - y_3)^2} \leq R_3 \\ 0 & \text{otherwise;} \end{cases}$$

Use the function `rotateGrid` to rotate the grid for different values of  $\phi$ . For each  $\phi$  compute the sum of intensities along the  $j'$  gridline  $P(\xi, \phi)$  as discussed in class and plot these profiles of the sums as a function of the  $i'$  gridline (This is the  $\xi$  variable). Submit plots for  $\phi = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi$ . Choose  $height = width = 100$ ,  $\mu_1 = 10, \mu_2 = 50, \mu_3 = 100$ ,  $(x_1, y_1) = (25, 25), (x_2, y_2) = (35, 75), (x_3, y_3) = (80, 65)$ ,  $R_1 = R_3 = 15, R_2 = 10$ . Compare the results with those using Matlab's Radon transform function.

## CT Reconstruction

**4. Back-projection:** In Problem 3 you computed the sum of intensities along  $j'$  grid lines for different values of  $\phi$ ,  $P_\phi(i')$ . We will now reconstruct the function  $f(x, y)$  from  $P_\phi(i')$  using back projection. Project the data  $P_\phi(i')$  for all values of  $i'$  and  $\phi$  back on to the image grid. At each grid location, the contribution of all the rays passing through that point are added up. You will need to interpolate to find the exact value at the grid location  $(i, j)$  based on the values of the rays passing through the pixel area for each  $\phi$ , Figure 2. Do the back-projection for the synthetic image function created in the Problem 1 and compare the results, reconstructed images and the original images. Also do the back-projection for the CT image, and compare the results for the reconstructed and original images.

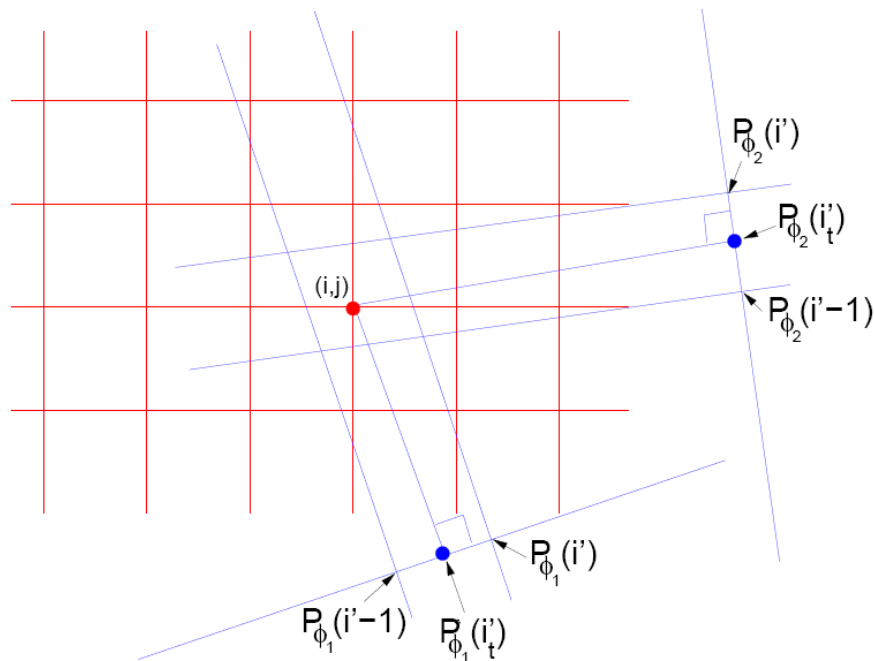


Figure 2: Illustration of back-projection. The value of the function at at grid location  $f(i, j)$  is  $\sum_{\phi_i} P_{\phi_i}(i'_t)$ . The point  $i'_t$  is the projection of the grid point  $(i, j)$  on to the  $i'$  gridline.  $P_{\phi_i}(i'_t)$  is then computed as the linear interpolation of  $P_{\phi_i}(i')$  and  $P_{\phi_i}(i' - 1)$ . In the above figure computation of  $P_{\phi_i}(i'_t)$  for two values of  $\phi_i$ ,  $i = 1, 2$  is illustrated.

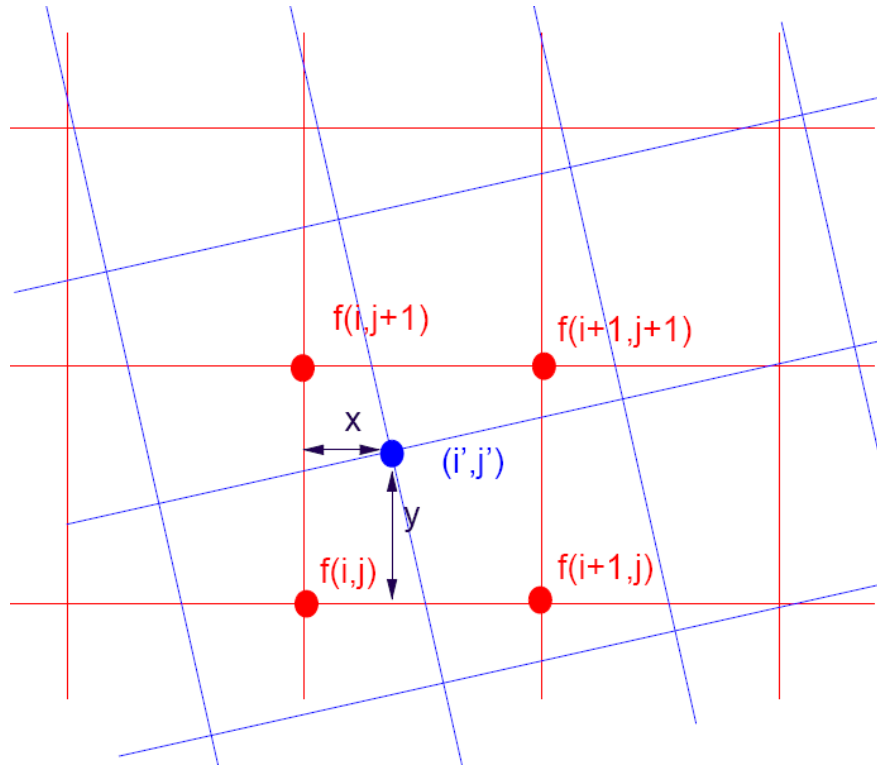


Figure 3: Bilinear interpolation. Original grid (red) and rotated grid (blue). The function value at rotated grid location  $(i', j')$  is given by

$$f(i', j') = (1 - x)(1 - y)f(i, j) + (1 - x)yf(i, j + 1) + x(1 - y)f(i + 1, j) + xyf(i + 1, j + 1).$$