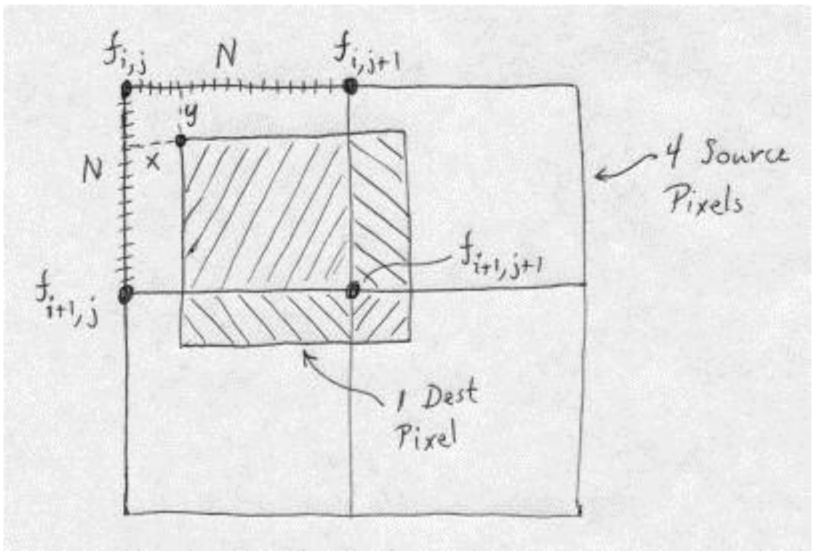
ROTATION & FAST ROTATION BY AREA MAPPING

RAM is limited in application to situations where you have an 8 bpp grayscale or a 24 bpp rbg image, you want the best quality rotated image, and you can sacrifice some speed for quality. On current computers, speed may not be an issue, as RAM can rotate 24 bpp rgb images at a rate of about 1 pixel in 200 machine cycles (about 15 million pixels/sec on a 3 GHz machine).

Area mapping works as follows. For each dest pixel you find the 4 source pixels that it partially covers. You then compute the dest pixel value as the area-weighted average of those 4 source pixels. We make two simplifying approximations:

- The areas are computed as if the dest pixel were translated but not rotated.
- The overlap area is computed on a discrete subpixel grid. Because we are using 8 bpp images with 256 levels, it is convenient to break each pixel into a 16x16 sub-pixel grid, and count the number of overlapped sub-pixels.



In the figure, $f_{i,j}$ is the value of the source pixel whose UL corner is at the location [i,j] indicated. Pixel vertices are labelled in standard [row, column] matrix notation. Coordinate values increase as you move down and to the right, which is the standard for image processing. Four source pixels are shown, each with their pixel value labeled in its UL corner. Each source pixel is subdivided into NxN subpixels; we show the subdivision only on one of the pixels. We want to determine the value of the dest pixel whose UL corner is at location V, which has coordinates x and y, in subpixel units, relative to the UL corner of the upper-left source pixel. Taking the contributions from the four source pixels in the order UL, UR, LL, LR, the normalized average of the destination pixel, using area averaging, is:

Dest pixel value = $(1/N^2)$ $[(N - x)(N - y)f_{i,j} + x(N - y)f_{i,j+1} + y(N - x)f_{i+1,j} + xyf_{i+1,j+1}]$

This is the same digital filter that is obtained for *linear interpolation when arbitrarily scaling grayscale images*!

Fast Rotation By Area Mapping

For RGB color images, area mapping interpolation can be slightly sped up (by about 10 to 20 percent) using 4x subsampling (i.e., 16 subpixels), inlining the code for the area mapping for each of the 16 cases, and using the accurate coefficients for the linear interpolation. The constants sum to a power of 2, allowing fast division. (We incorporate the division into the re-positioning of each of the R, G and B components into the destination pixel, so it takes no extra time.) As usual, all operations are done on 32-bit quantities using multiplication, addition and shift.

For most applications, the loss in accuracy (relative to the 16 x 16 subpixel version) is minimal, and the rotation speed for RGB images on a 3 GHz processor is about 18 million pixels/second.

Source: http://www.leptonica.com/rotation.html