WHAT SPECIAL CASES HAVE BEEN IMPLEMENTED?

It is also important to treat the special cases efficiently. For example, suppose you know in advance that the *src* and *dest* blocks are bit-aligned within 32-bit words. A vertical block shift is such a special case. In such a situation, a large fraction of the shifting and masking operations are not required, and a special low-level function for aligned blocks is used. Besides efficiency, however, it is useful to have this function, *rasteropVAlignedLow(),* because the general rasterop function *rasteropGeneralLow()*can be derived as a straightforward generalization of the special one. This is a common situation, where to write the general function it is easiest first to write and debug a simpler, special case, and then to generalize that. An even more specialized rasterop is one where both the *src* and *dest* rectangles have their left edges on a 32-bit word boundary. This is a common situation, such as when the rectangle comprises each entire image, and a specialized function *rasteropWordAlignedLow()* handles it.

Another set of special cases are unary rasterops that operate on a general rectangular region of a single image. There are only 3 non-trivial operations:

PIX_CLR, PIX_SET, and PIX_NOT(PIX_DST).

These operations are implemented as special cases of word-aligned binary rasterops. When called with the high-level, 9 parameter *pixRasterop()* function, the last 3 arguments (src Pix and UL corner coordinates) are ignored. There is even a special case of unary rasterop where the left edge of the rectangle is aligned on a 32-bit word boundary.

Yet another special case of unary rasterops has been implemented to *move* pixels *in-place* within special rectangular regions. These two functions are

- *rasteropVipLow()*. This does an in-place full height vertical block transfer, moving a set of pixel columns up or down by a given amount, and clearing the region that was not "blitted" into. For example, when a pixel column is moved down by n pixels, the lowest rows are moved first, and then the first n rows at the top of the column must be cleared.

- *rasteropHipLow()*. This does an in-place full width horizontal block transfer, moving a set of pixel rows left or right by a given amount, and clearing the region that was unchanged.

For example, for a vertical block transfer, the columns are moved by copying words, properly masked, that have been shifted up or down in the image. If the columns are moved upward, the words are taken row by row, moving sequentially down the image, and written up a number of rows given by the shift amount; and v.v. for shifting a column down. These unary (in-place) block shift functions are particularly useful for performing in-place shear and rotation of an image.

These horizontal and vertical in-place operations have been combined. By taking the block to be moved equal to the entire image, *pixRasteropIP()* is the high-level function that performs an arbitrary in-place shift of an image.

That's the big picture. We now look down at a few of the internal details.

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