Mason Pollack and Vladimir Hugec

Assistance:

TA Ashton: Mapping logic
TA Sam: Helped with valgrind errors and leaks

Incorrectly implemented: N/a

Architecture:

UArray2b:

UArray2b contains a UArray2. The UArray2 contains UArrays at all of its indices. The UArrays represent blocks while the indices inside the UArrays hold the data.

ppmtrans:

Either a UArray2 or UArray2b are used to represent the image.
An A2 methods suite is used so these two structs can have functions called them without causing error. Data is read into one of the structs using Pnm_read and then the array is manipulated based on the command line arguments. The amount
Of time taken for the manipulation is clocked and can be put in an output file.

Part E:

Test	Image Size (bytes)	Total CPU time (ns)	Time per pixel (n)	# Instructions per pixel	Computer Info
90 Rotation Row-major	92211	49300271	91.559608	915.59608	Intel Core i5-4590 CPU @ 3.30 GHz 3382.983 MHz
180 Rotation Row-major	92211	43907361	81.543989	815.43989	
270 Rotation Row-major	92211	51264636	95.207793	952.07793	
90 Rotation col-major	92211	55584396	103.230376	1032.30376	
180 Rotation col-major	92211	63613931	118.142689	1181.42689	

270 Rotation col-major	92211	56343714	104.640568	1046.40568	
90 Rotation Block-major	92211	180367221	334.974874	3349.74874	
180 Rotation Block-major	92211	181785489	337.608857	3376.08857	
270 Rotation Block-major	92211	180193112	334.651522	3346.51522	

The pattern that is most prevalent is the increase in computer time from row-major -> column major -> block major. Block major is about 3 times slower than column major which is a couple of nano seconds slower than row major. Block major is the slowest because all of the blocks couldn’t fit on the cache. This means that the cache can only hold part of the data, and because the blocks are not contiguous in memory, it takes time to load new blocks. Column major mapping is slower than row major because the data is not stored contiguously in memory like rows are. However, column is not much worse than row and a lot better than block because when mapping through columns, the memory can use a constant stride to iterate through the data in memory. Row major is the most efficient because it just has to loop through the contiguous data in memory and has to change the values on the cache less frequently than column major because it does not need to use a stride.

For row major, the 180 degree mapping function is much faster than the 90 or 270 degree. This is because in a 180 degree rotation rows map to rows, which allows for contiguous memory to be brought into the cache, and all of this memory is moved to their new location before other memory is brought into the cache.

For column major, 90 and 270 degree rotation are better. This is because columns map to columns in this rotations. The mapping function deals with columns first and these values are put in their final position in the same order that they are mapped to.

For block major, all of the rotations were very close in the time it took to complete. This is because each block was stored contiguously in memory and so access, regardless of the type , was done in the same time.

Time spent: 30 hours