

# CS133- LAB 1 REPORT

## SEQUENTIAL RUN BASELINES

Run	512 - (sec.)
1	4.07
2	4.02
3	3.97
4	4.10
5	3.74
<b>Average</b>	3.98

## METHOD 1

Using `#pragma omp parallel for` on the outermost loop (in `dwt3d_main.c`, parallelizing the image slices).

Threads	512 (sec.)	Speedup
2	2.43	
4	1.39	
8	0.82	
16	0.77	~5x
32	0.93	

This parallelization was the most straight forward to do, since the 2D slices of images are completely unrelated to each other and thus there is no communication to worry about in parallelizing this portion. The concern, however, is that parallelizing here uses up threads and makes other improvements in `cdf97` useless. Thus, my later tests removed this line for speedup comparison.

## METHOD 2

In an effort to localize data access, I combined the first two steps into one for loop. I also added more parallel fors. This results in an average of ~0.6 for 16 threads, with lowest at about 0.5s. I attempted to group the second two steps together, but couldn't manage to get the variable renaming correct.

## METHOD 3

Attempting to "pipeline" dependencies, by only waiting as necessary for x to finish before initiating tempbank transforming, using omp sections. This failed miserably (code hangs) ☹.