**Notes/Results with BM3D prototyping**

* Most effective parameters for block sizes and matches appear to be
  + Block size: 32 (or 64 works reasonably well and is faster)
  + Max number of matches: 64
  + Min number of matches: 32
* For the choice of wavelets, db or sym orders 3 and 4 for the x-y dimensions, but haar on the z dimension is definitely most effective. Three or four levels seems to work fine
* A notable improvement is achieved by rotating the image about half the block size and re-evaluating the algorithm, then averaging the result. This way the reference blocks are brand new the second time. One could repeat this shifting many times, but twice appears suitable, with diminishing returns for more.
* Likewise to the note above, cycle spinning with the wavelets reduces the error further. It seems just one extra shift by one unit in each direction provides enough benefit.
* Performing a second Wiener filter step improves the result by another epsilon. The original BM3D code shows more significant improvement however, but I’m not sure why. In my code it doesn’t seem worthwhile. If you were to do it, it’s probably simplest to just avoid rematching blocks, and perform the Wiener filtering in wavelet domain as the hard thresholding is being evaluated. The Wiener filter seems to only really be useful if the image is REALLY noisy.
* In almost all cases, our error is just slightly worse than BM3D. I think we may have to live with this. In the case that Elad and his coauthors are right, denoising images with white Gaussian noise is a solved problem and BM3D is essentially the limit of that problem. So we can only hope to match it.