ECE 590 Project Work for Jan Fure and Michael Larson

Part of remaining work is to implement a median filter. This operation has been implemented in same style as the other operations supported by the image processor, as a combinatorial gate implemented behaviorally.

The optimal version of the median filter was described as a combinatorial sorter of nine inputs, this could have been implemented using 3x8 cascaded stages, each containing four 2 way min/max comparators. This has not implemented yet, as it is not clear howto make an interface that translates the array input to a bit vector without consuming any clock cycles.

The implementation of the median filter relies on applying three 3 way median filters in parallel, and feed the output into a 3 way median filter, thus extracting the median of the median of each row in a 3 by 3 array. This is repeated for each channel (red, green and blue). A 3 by 3 kernel allows this treatment without too much complexity, as 3 elements represent 6 permutations for comparing each value, each requiring 2 comparators. A single 3 input median filter requires 12 comparators, for a 3x3 image processor kernel with 3 colors; this adds up to 144 comparators.

The median of median approach was inspired by the following Wikipedia article:

https://en.wikipedia.org/wiki/Median\_of\_medians

To test the effect on an image, a matlab function to add image noise was applied “J = imnoise(I,'salt & pepper',0.02);”, and the ability of the filter to remove the noise was tested. The results were quite good, however it appears the noise was single pixel sized, and clusters might represent a bigger challenge. The important observation is that the filter seems equally capable of removing image noise as a true median filter.



Starting picture



After median filter