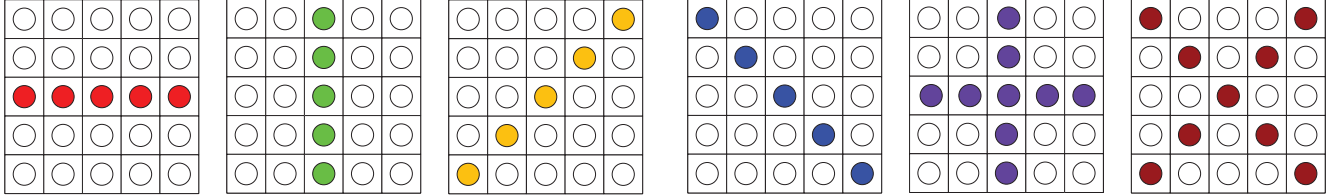


Scalable FPGA Median Filtering via a Directional Median Cascade

Oscar Rahnama, Stuart Golodetz, Tommaso Cavallari and Philip H. S. Torr



I. INTRODUCTION

The 2-D median filter, one of the oldest and most well-established image-filtering techniques, still sees widespread use throughout computer vision. Despite its relative algorithmic simplicity, accelerating the 2-D median filter via a hardware implementation becomes increasingly challenging as the window size increases, since the resources required grow quartically with the window size. Previous works, in a non-FPGA context, have shown that separately applying several directional median filters to an image, and then taking the median of their results, yields performance that is competitive with, and in some cases even better than, that of a classic 2-D window median. Inspired by these approaches, we propose a novel way of substituting a 2-D median filter on an FPGA with a sequence of directional median filters, in our case arranged as a pipeline, in the pursuit of an FPGA implementation that achieves better scalability and hardware efficiency without sacrificing accuracy. We empirically show that the combination of three particular directional filters, in any order, achieves this, whilst requiring quadratically fewer resources on the FPGA and allowing for much higher throughput.

II. METHOD

We adapt classic non-FPGA works on multi-pass filtering to suit the FPGA context, with the key novelty being the use of a sequential pipeline in place of the original ‘median-of-medians’ architecture. To demonstrate the effectiveness of our approach, we consider all possible sequences of the six directional medians shown in the figure, and compare their performance to that of a classic 2-D median. We show experimentally that any permutation of a +-patterned median, a column median and a row median outperforms a 2-D median, whilst achieving the aforementioned resource usage and throughput gains.

This work was done whilst OR was with the University of Oxford and Five, and whilst SG was with Five. SG is now with the University of Oxford. TC is with Five. PT is with the University of Oxford and Five.

E-mail: oscar.rahnama@linacre.ox.ac.uk, tommaso.cavallari@five.ai, stuart.golodetz@cs.ox.ac.uk, philip.torr@eng.ox.ac.uk.

III. EXPERIMENTS

We compare numerous different sequences of directional medians with a standard 2-D median filter on an augmented version of the black-and-white Kodak Lossless True Color Image Suite [1]. Each possible permutation of the six basic filters, up to a sequence length of four filters, was tested for window sizes ranging from 3×3 to 15×15 (we only show up to 11×11 below for space reasons). We made 77 targets from each 3072×2048 image in the dataset by adding different amounts of zero-mean Gaussian noise to a fraction of its pixels in “salt and pepper” fashion. We evaluated each median filter variant (filter sequence/window size) on each target and used the results to compute a score for each variant. Let V_n be the set of all $n \times n$ median filter variants we consider, T be the set of all targets, and $\text{MSE}(v, t)$ denote the mean-squared error between the result of filtering $t \in T$ with variant $v \in V_n$ and the base image from which t was originally derived. We define the normalised MSE of v on t as $\text{NMSE}_n(v, t) = \text{MSE}(v, t) / \min_{v' \in V_n} \text{MSE}(v', t)$. The overall performance $S_n(v)$ of v on the dataset can then be defined as the average NMSE over all targets, i.e. $S_n(v) = (\sum_{t \in T} \text{NMSE}_n(v, t)) / |T|$. The smaller $S_n(v)$ is, the better the overall performance of v for its window size (clearly the smallest $S_n(v)$ can be is 1).

IV. RESULTS

	3×3	5×5	7×7	9×9	11×11
std	1.605	1.707	1.985	2.201	2.326
+pat→row	1.427	1.203	1.263	1.348	1.417
+pat→col	1.481	1.323	1.448	1.558	1.638
+pat→col→row	1.290	1.446	1.710	1.905	2.046
+pat→row→col	1.290	1.447	1.710	1.906	2.046
row→col→+pat	1.373	1.543	1.829	2.044	2.172
row→+pat→col	1.367	1.480	1.753	1.966	2.118
col→+pat→row	1.381	1.507	1.782	1.987	2.114
col→row→+pat	1.379	1.553	1.839	2.050	2.177

REFERENCES

- [1] Kodak, “Kodak Lossless True Color Image Suite,” <http://r0k.us/graphics/kodak/>, available online as of 01-07-2019.