Fidelity Slider: a User-Defined Method to Trade off Accuracy for Performance in Canny Edge Detector

Valery Kritchallo, Koen Bertels, Zaid Al-Ars

Delft University of Technology

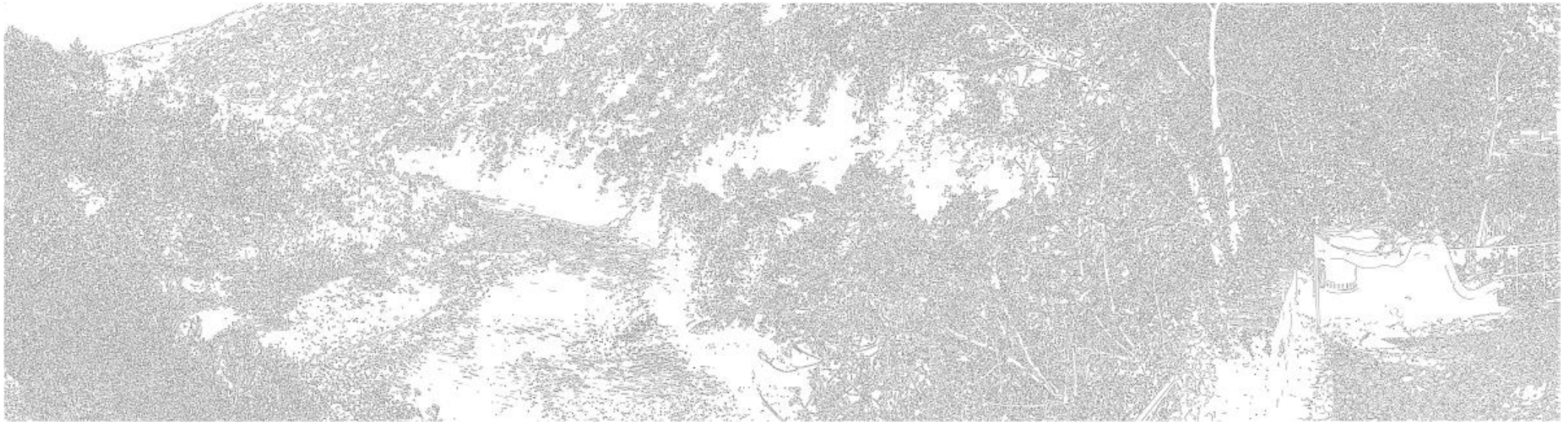
Erik Vermij

IBM Research, the Netherlands

WAPCO 2016, Prague
January 20, 2016

Canny edge effect example (stAnna.pgm, 8896x2397 pixels, 20.3 MB)

original



edge effect applied

Canny edge detector algorithm stages and runtime breakdown

Gaussian smoothing	76.7%
Gaussian derivative	4.4%
magnitude of the gradient	3.8%
non-maximum suppression	8.2%
hysteresis edge thresholding	4.4%
edge tracing	2.3%

Problem formulation

- traditional approaches typically use proprietary accelerator hardware (GPU, Tile64 etc.) -- we want a new, platform-independent, highly scalable multi-core solution;
- the solution should be able to process efficiently images of arbitrary size and dimensions (including non-square);
- can we avoid parallelizing the application using the old loop-by-loop approach?
- can we win extra performance by sacrificing (in a controlled way) some accuracy in the rendered output?

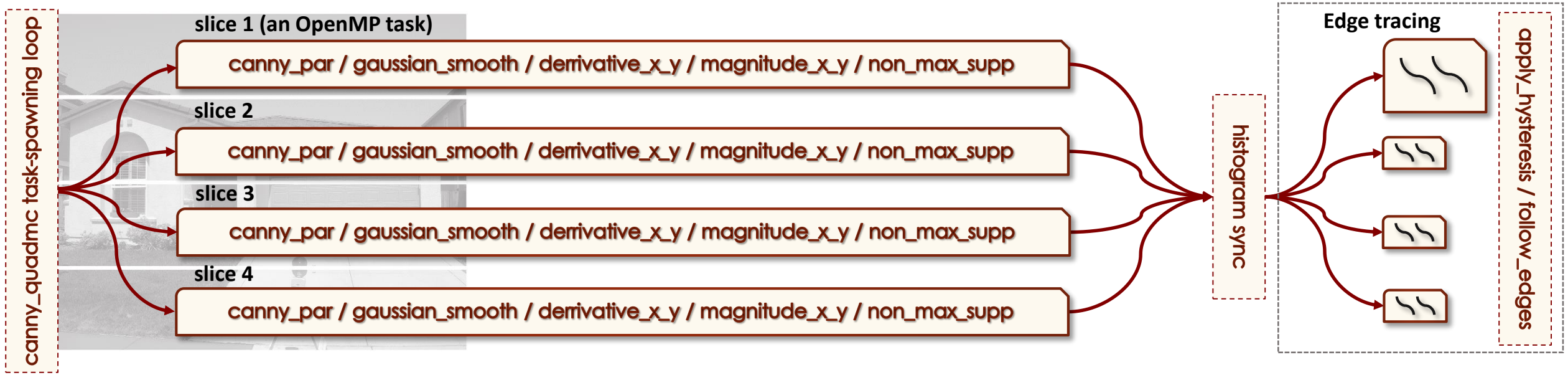
Characteristics of our CED implementation

- coarse-grained data parallelization through domain decomposition;
- introduction of a single top-level data-chunking (image-slicing) loop;
- use of approximate computing principles;
- user-defined control of the tradeoff between performance and quality of the traced output via the fidelity slider.

The source code fragment implementing the main image-slicing loop

```
/*
 * iterate over image slices
 */
for (row_ix=0; row_ix<rows; row_ix+=rows_slice)
{
      #pragma omp task shared(edge_file) if (do_async_tasking)
      {
            /*
             * call the main filter function to process
             * the image slice as a concurrent task
             */
            canny_par(row_ix, rows_slice, cols, image, ...);
      }
}
```

OpenMP-based parallelization scheme of the CED



Observed issues in the rendered output

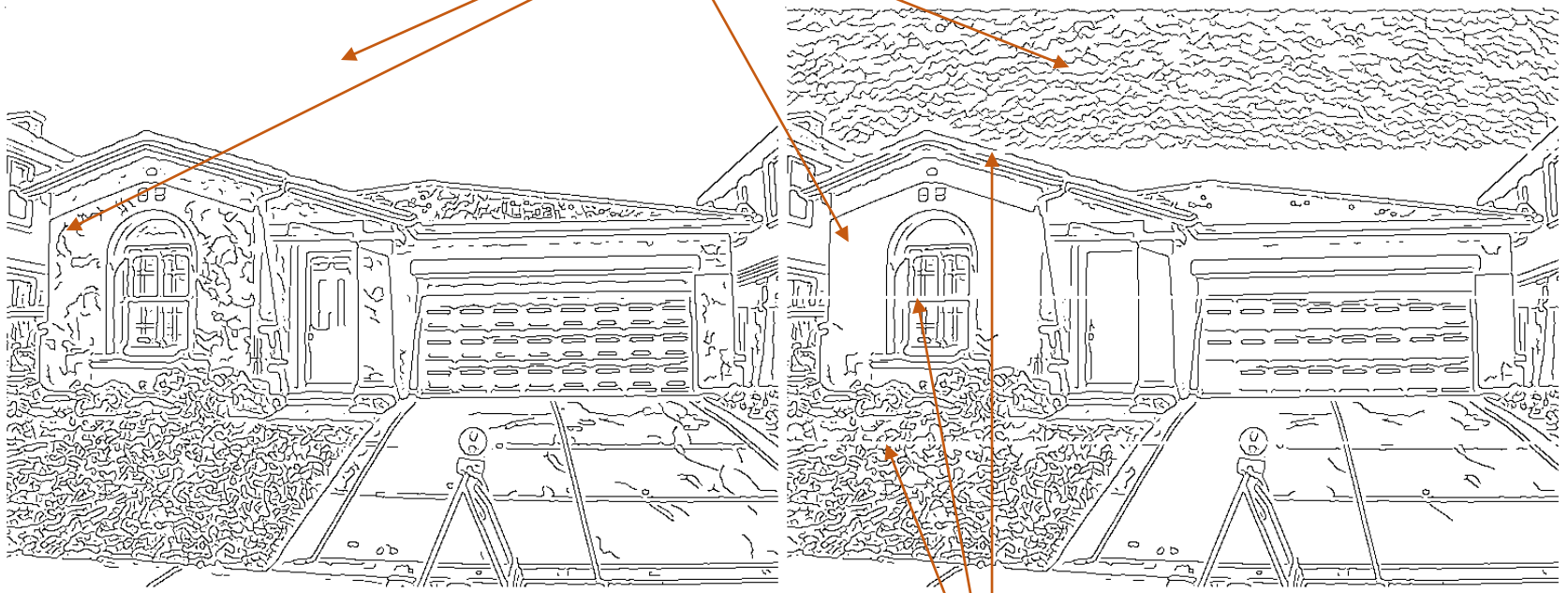
	divergence component / fidelity slider factor
1) horizontal visual breaks resulting from the broken continuity in the Gaussian-smoothing stage;	V1 / F1
2) areas in the image rendered differently from the reference output due to the image histogram array computed piece-wise within each slice;	V2 / F2
3) differently traced edges as a result of violating the logic of the recursive edge-tracing procedure.	V3 / F3

The House image: a case of significant parallel / sequential rendering difference (slicing issues)

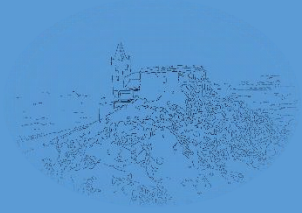
original

sequential

parallel at fidelity 0% (accuracy 95.1%)



Average pixel difference between two images metric



$$RE(p,s) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{|LPij - LSij|}{255.0}}{N * M}$$

$$AC = 100 * (1.0 - RE)$$

Expected accuracy and aggregate divergence equations

$$AC_{exp}(sv, ns, \tau_\sigma) = 100 * (1.0 - V_{ag}(sv, ns, \tau_\sigma))$$

$$0 \leq V_{ag}(sv, ns, \tau_\sigma) \leq 1.0$$

$$V_{ag}(sv, ns, \tau_\sigma) = V_1(sv, ns, \tau_\sigma) + V_2(sv, ns) + V_3(sv, ns)$$

$$\tau_\sigma = 2 * \lceil 2.5 * \sigma \rceil$$

Individual divergence component V1 and slider factor F1

$$V_1(sv, ns, \tau_\sigma) = \sum_{i=2}^{ns} vb_i(F_1(sv, \tau_\sigma))$$

$$F_1(sv, \tau_\sigma) = \left\lfloor \frac{sv - 1.0}{\tau_\sigma} + 1.0 \right\rfloor$$

Individual divergence component V2 and slider factor F2

$$V_2(sv, ns) = \sum_{i=F_2(sv+1,ns)}^{ns} vh_i$$

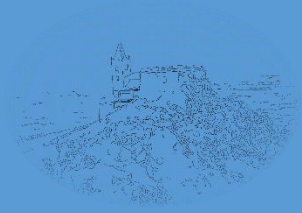
$$F_2(sv, ns) = \left\lceil \frac{sv * ns}{10.0} \right\rceil$$

Individual divergence component V3 and slider factor F3

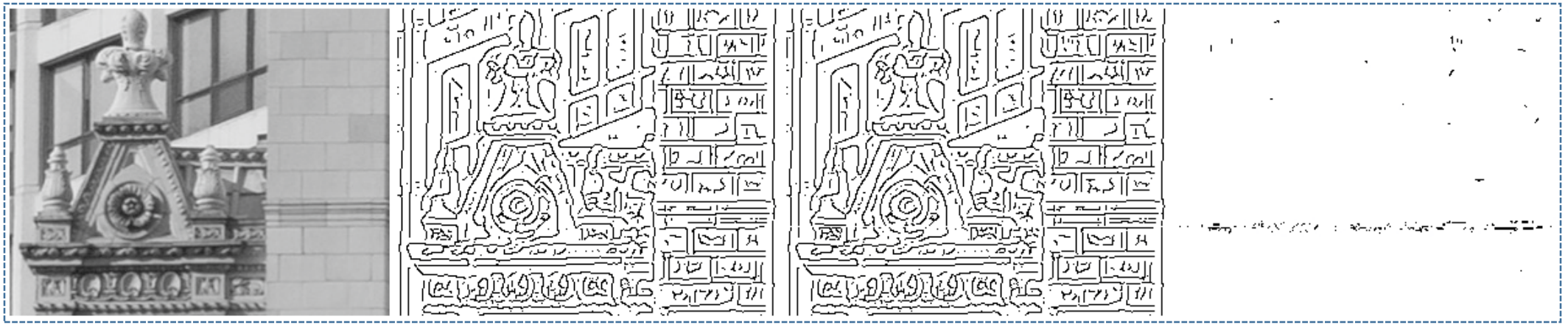
$$V_3(sv, ns) = \sum_{i=F_3(sv,ns)+1}^{ns} ve_i$$

$$F_3(sv, ns) = \left\lceil \frac{sv * ns}{100.0} \right\rceil$$

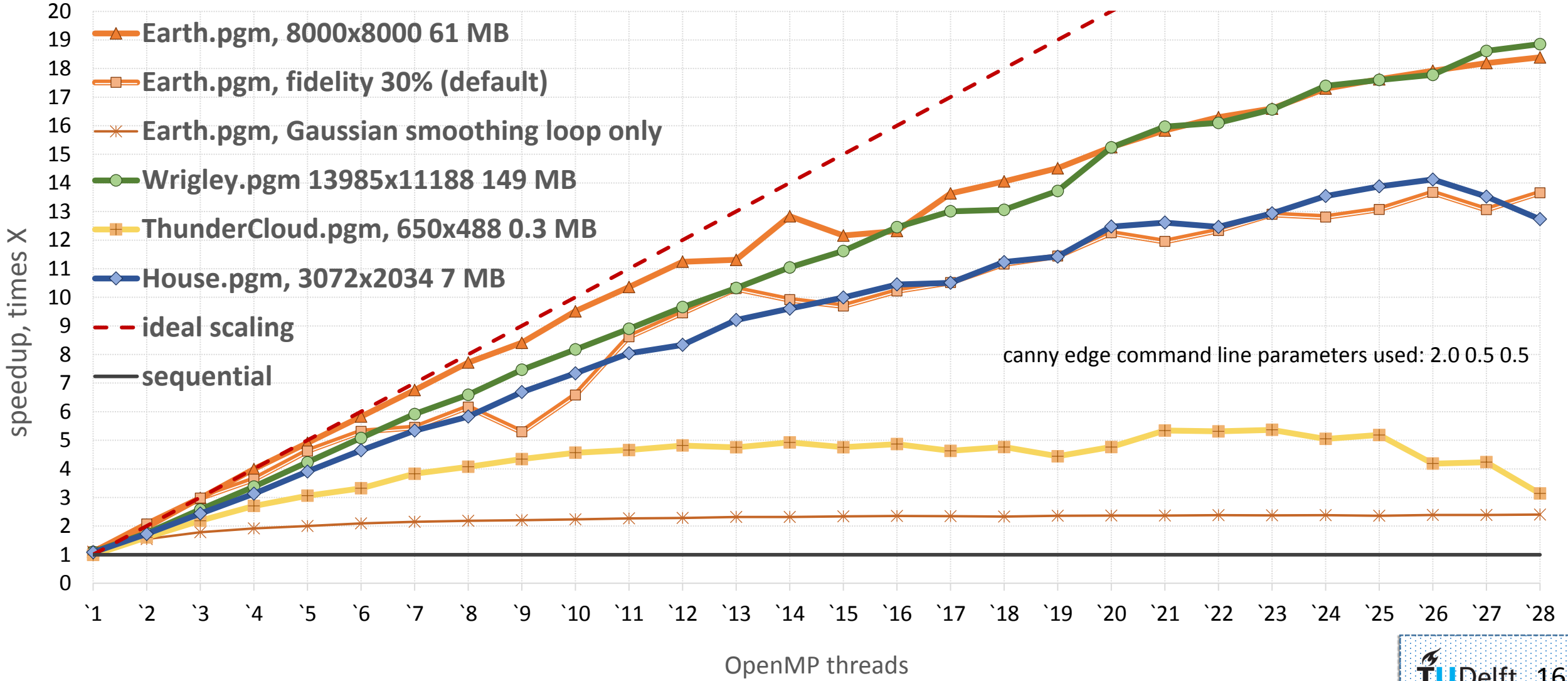
Edge rendering rigorousness at fidelity 1% (Wrigley image)



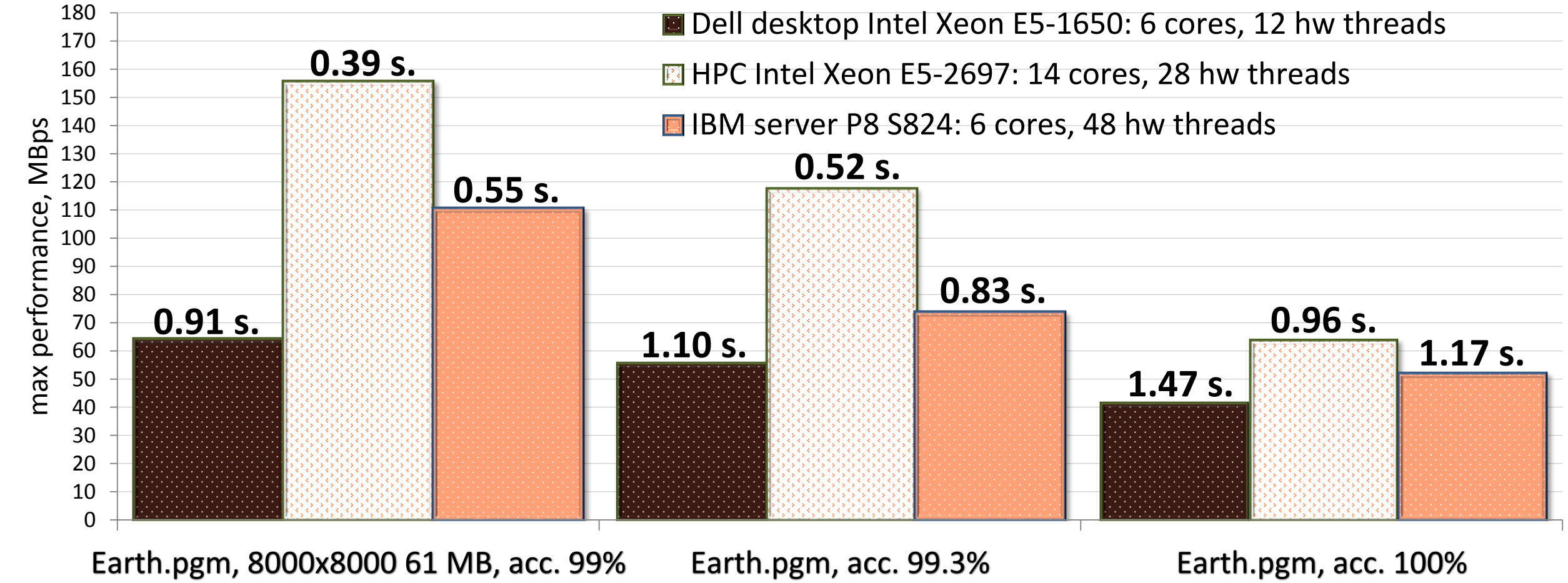
original	sequential	parallel at fidelity 1% (accuracy 98.01%)	difference



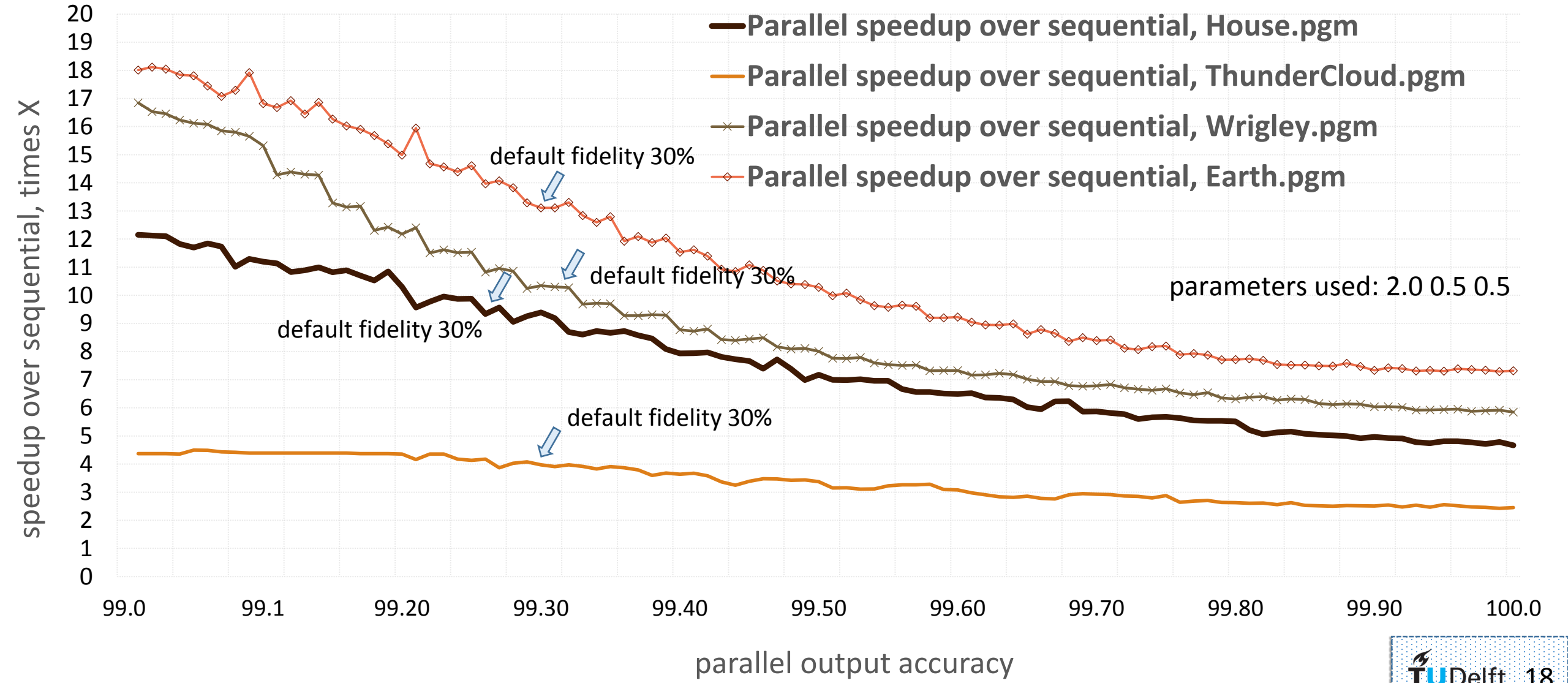
Benchmark results for the parallelized CED, fidelity slider value 1%



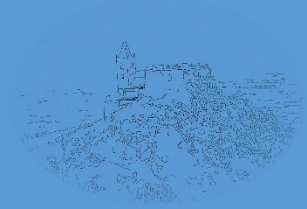
Maximal parallel performance in MBs per second

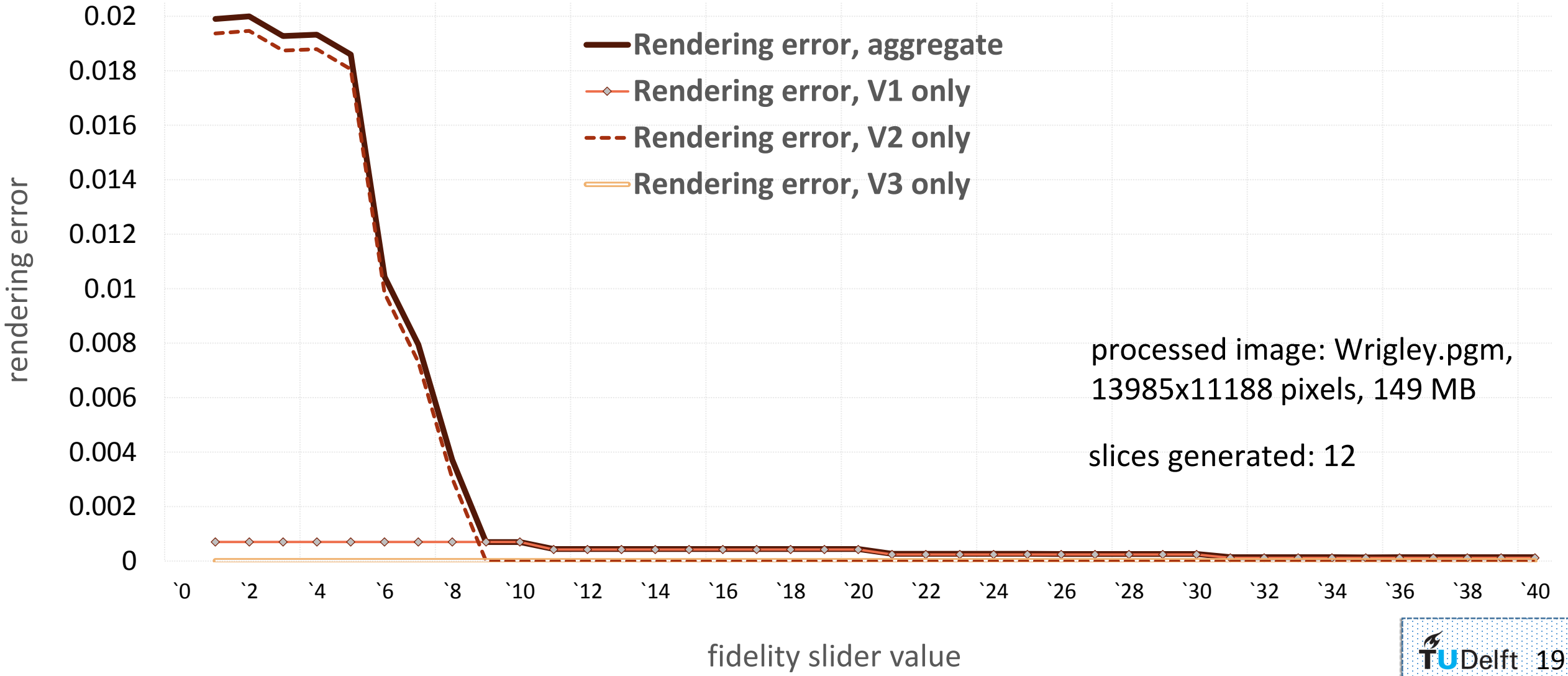


Impact of the output accuracy on the parallel speedup

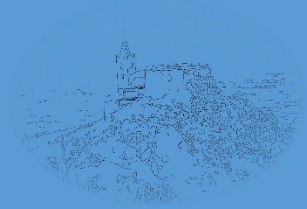


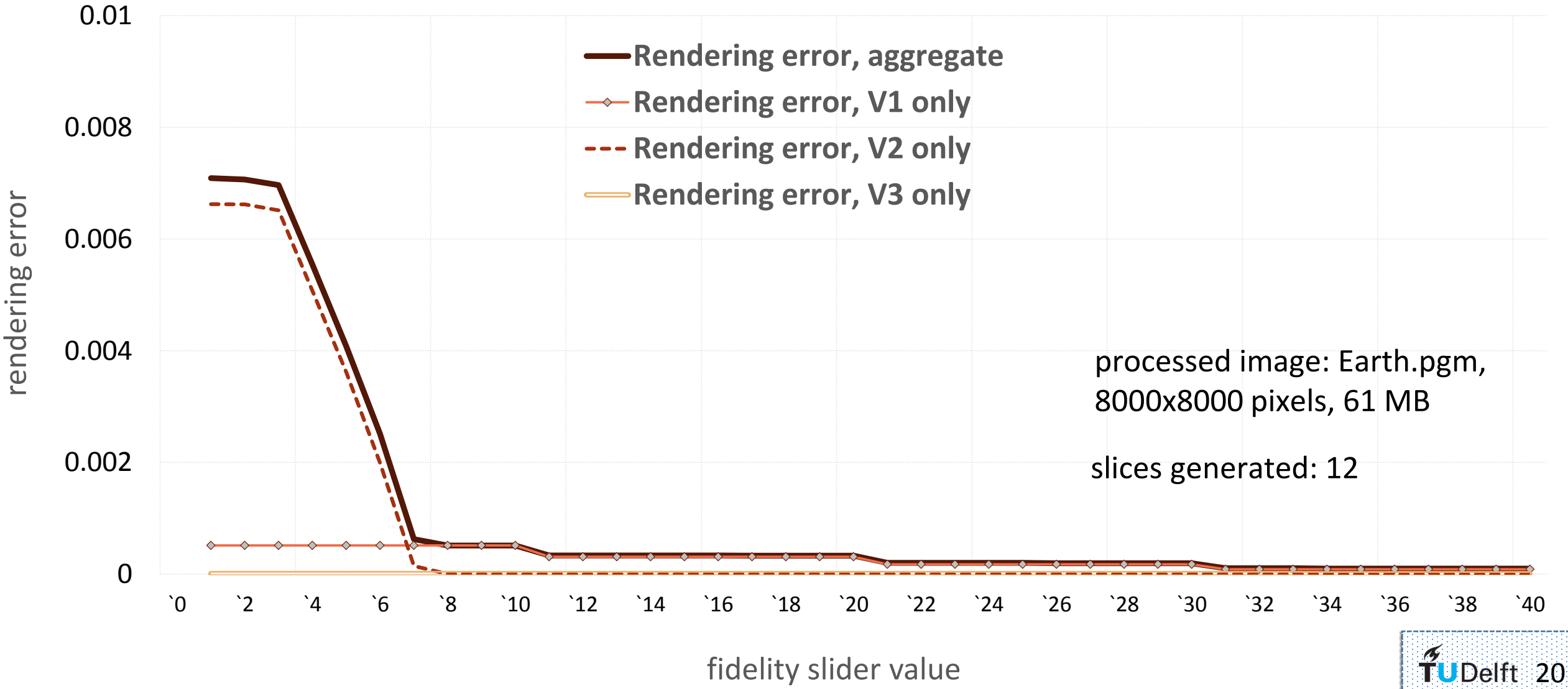
Breakdown of the aggregate rendering error into its V1, V2 and V3 components (Wrigley image)



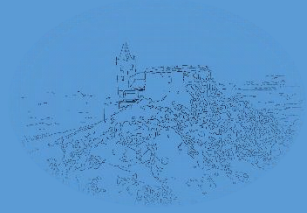


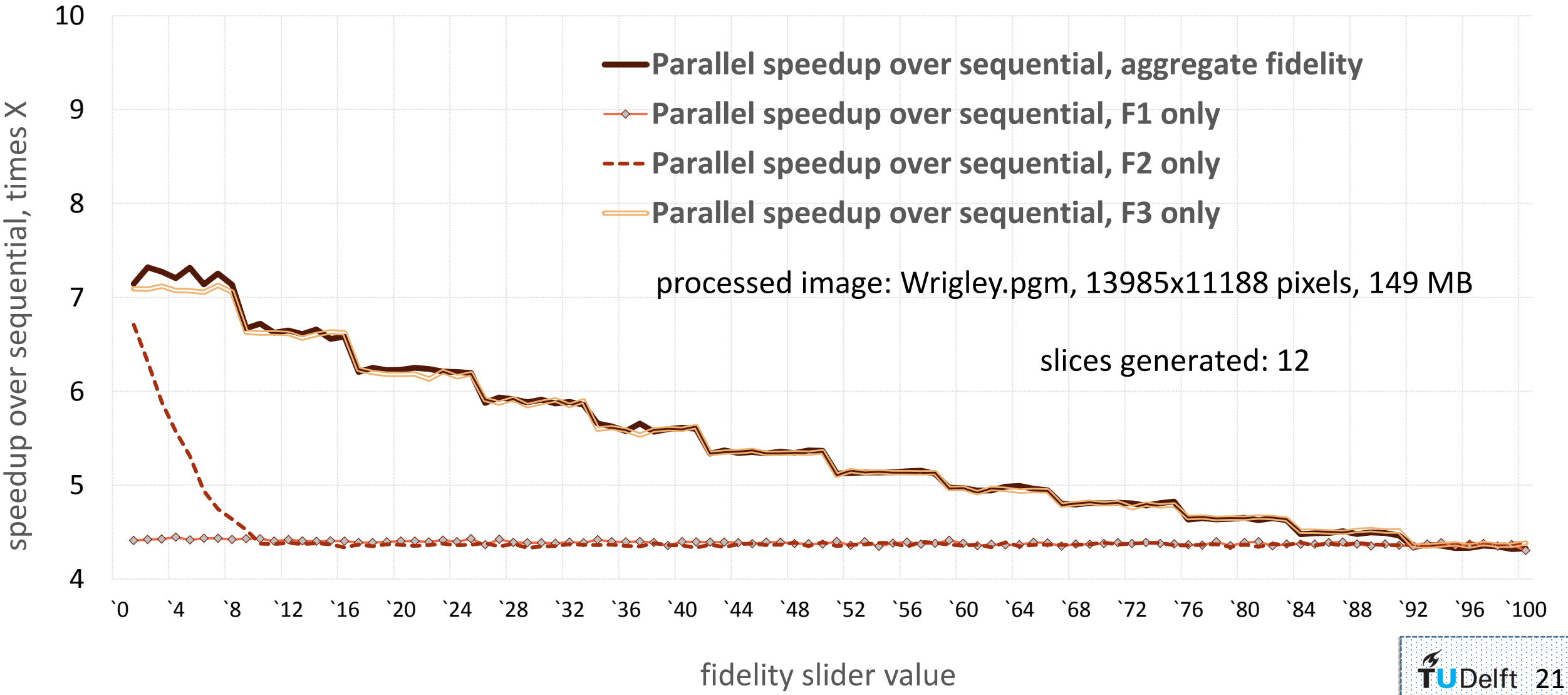
Breakdown of the aggregate rendering error into its V1, V2 and V3 components (Earth image)



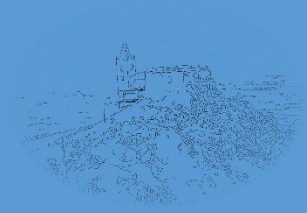


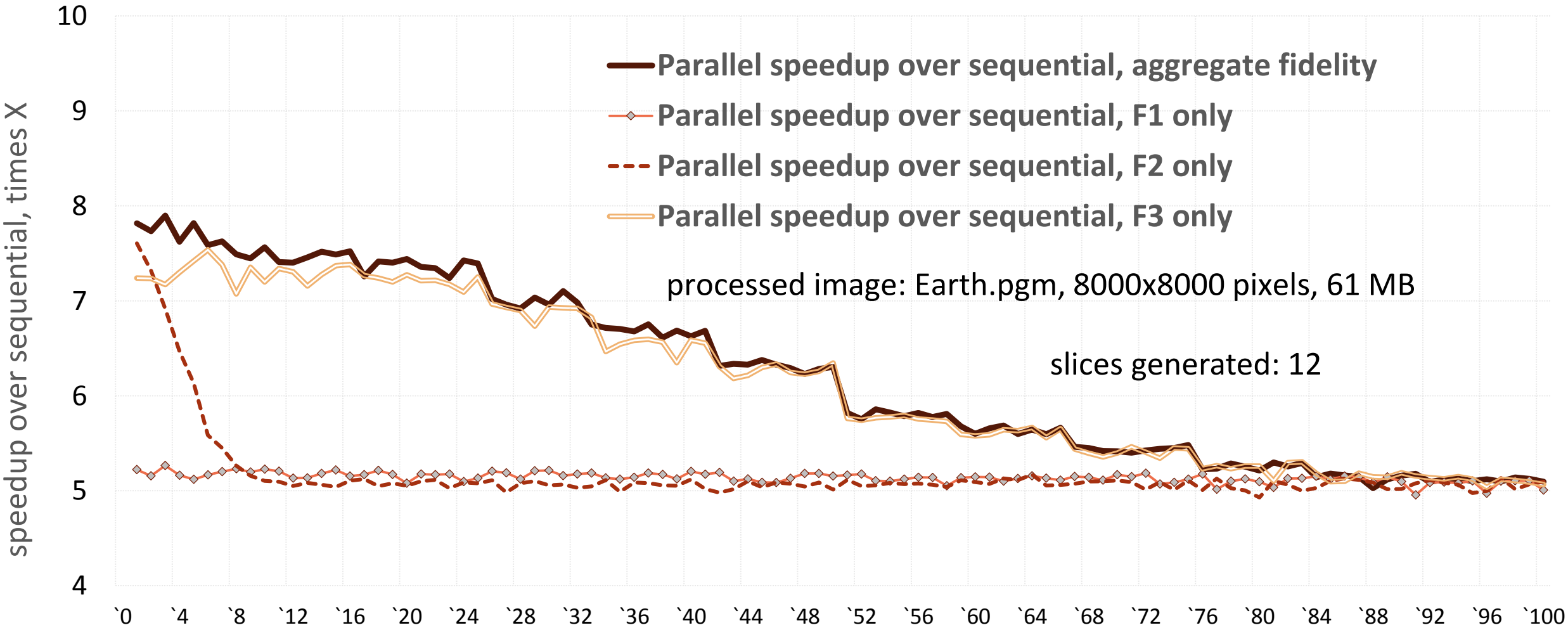
Individual impact of the F1, F2 and F3 factors on the parallel speedup (Wrigley image)



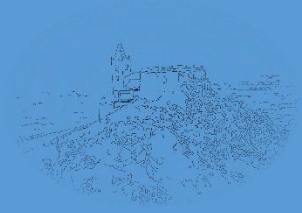


Individual impact of the F1, F2 and F3 factors on the parallel speedup (Earth image)





Summary of the results



- highly-scalable cross-platform solution;
- fidelity slider allows the user to control the tradeoff between performance and accuracy (relative to the sequential output) of the rendered edges;
- speedups from 7x at 100% accuracy to 19x at 99% accuracy on an Intel Xeon with 14 cores and 28 HW threads; still a plenty of room for further acceleration.

Future work

- More sophisticated histogram synchronization scheme with less overhead.

Wrapping up:
where to find the data used for this paper

The images, code and supplemental material used in this paper are available on the OSF site:

https://osf.io/i725h/