

Quilting

Dominic Steiner

Etienne Mettaz

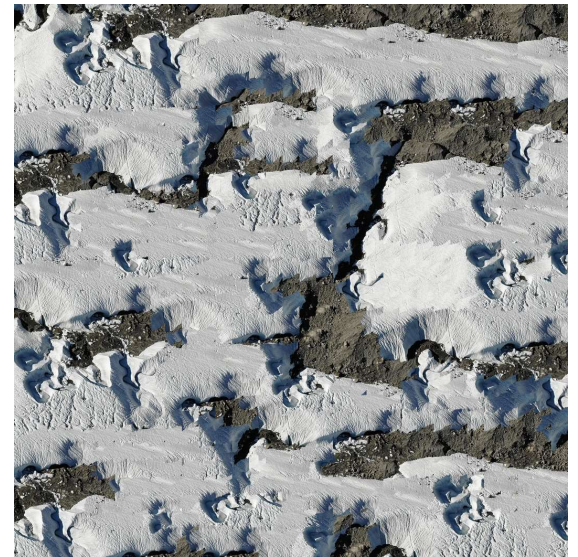
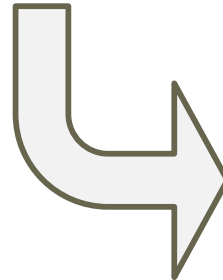
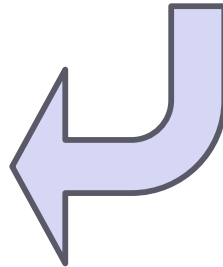
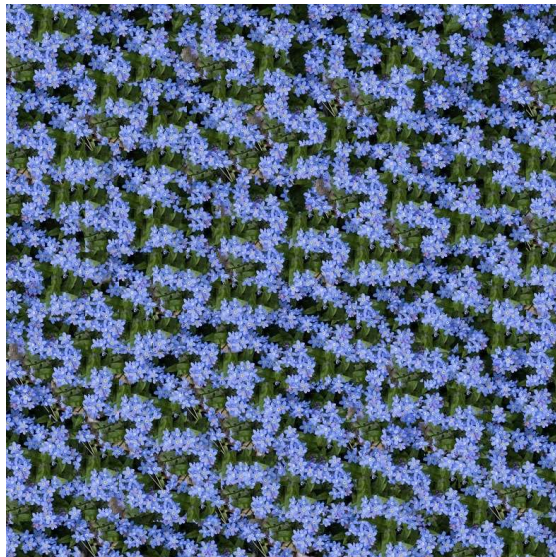
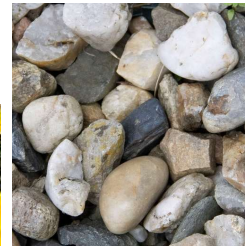
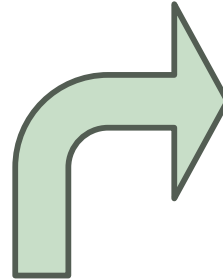
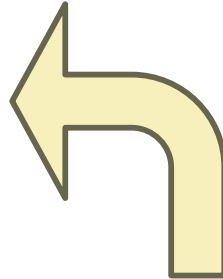
Oleh Kuzyk

Ondrej Cernin



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Quilting Algorithm



Quilting Algorithm

Input Image



Output Image

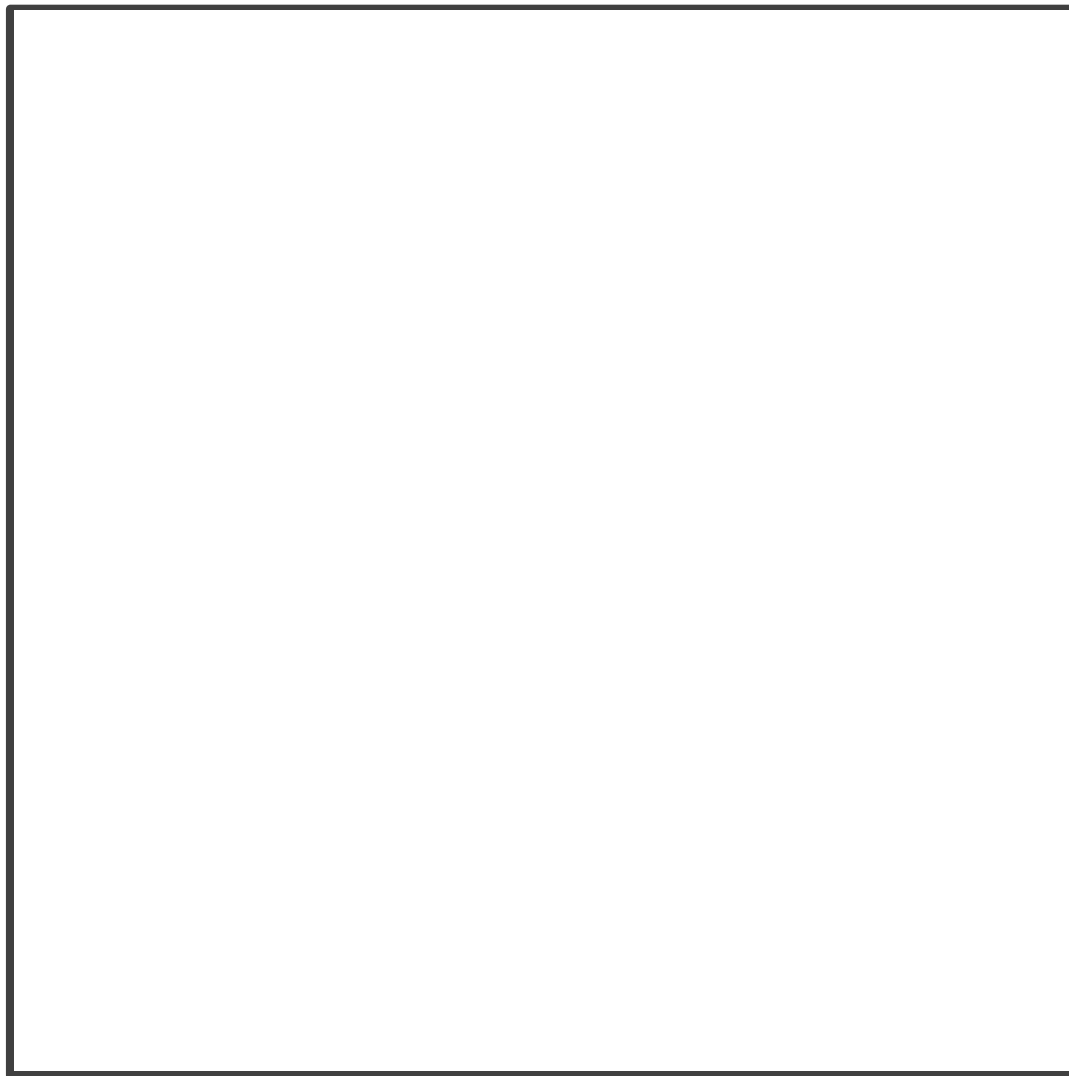


Quilting Algorithm

Input Image

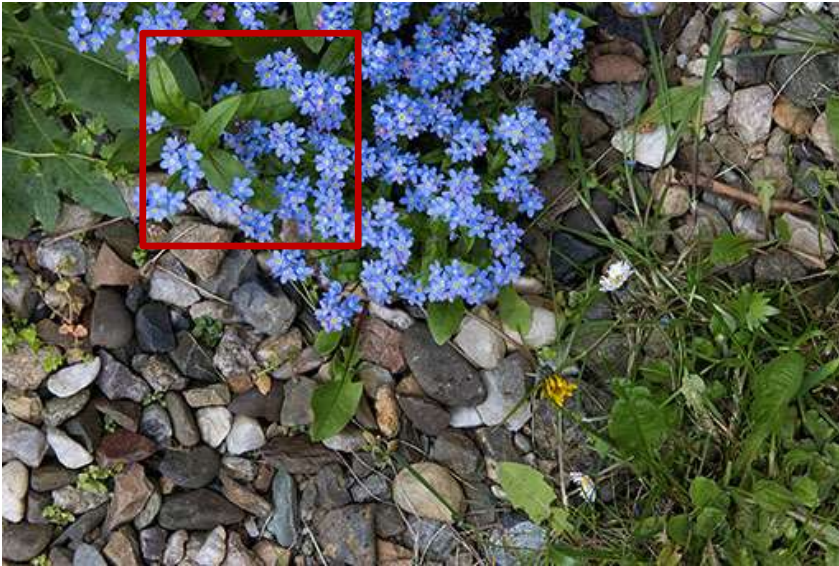


Output Image

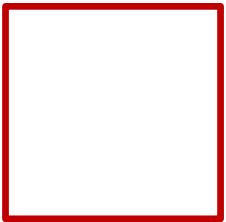
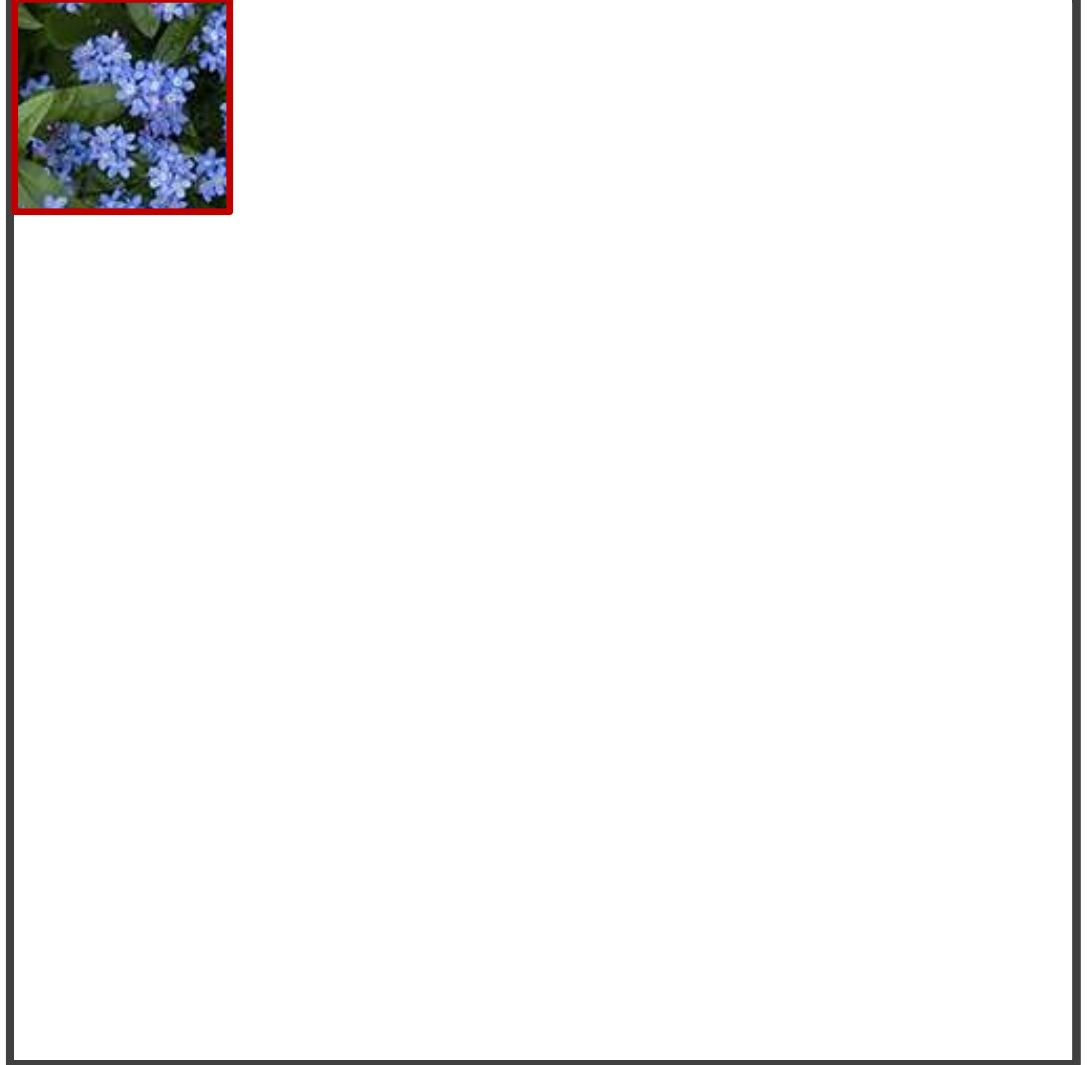


Quilting Algorithm

Input Image



Output Image



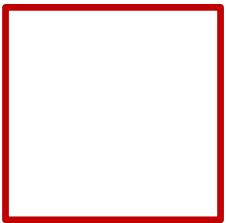
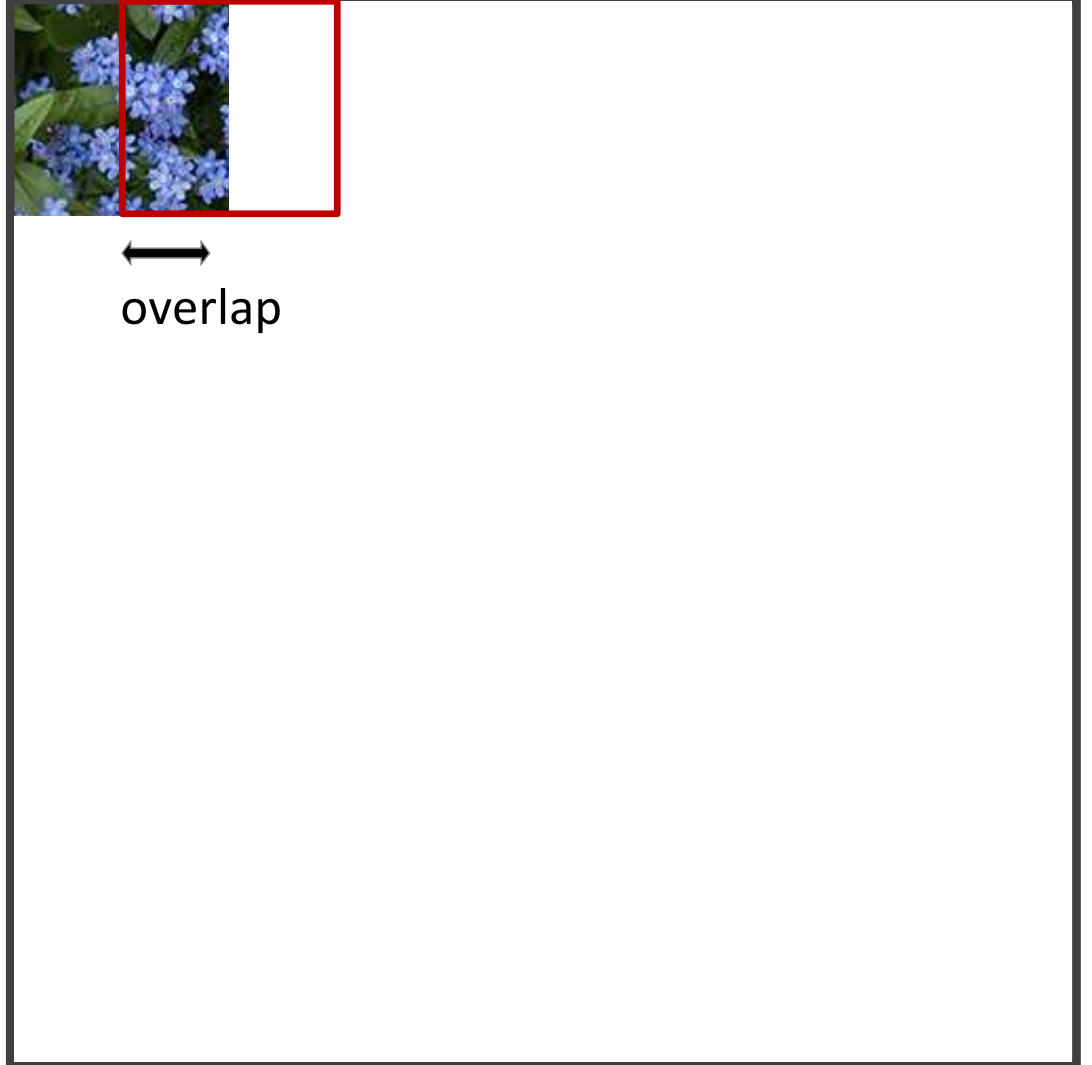
block size

Quilting Algorithm

Input Image



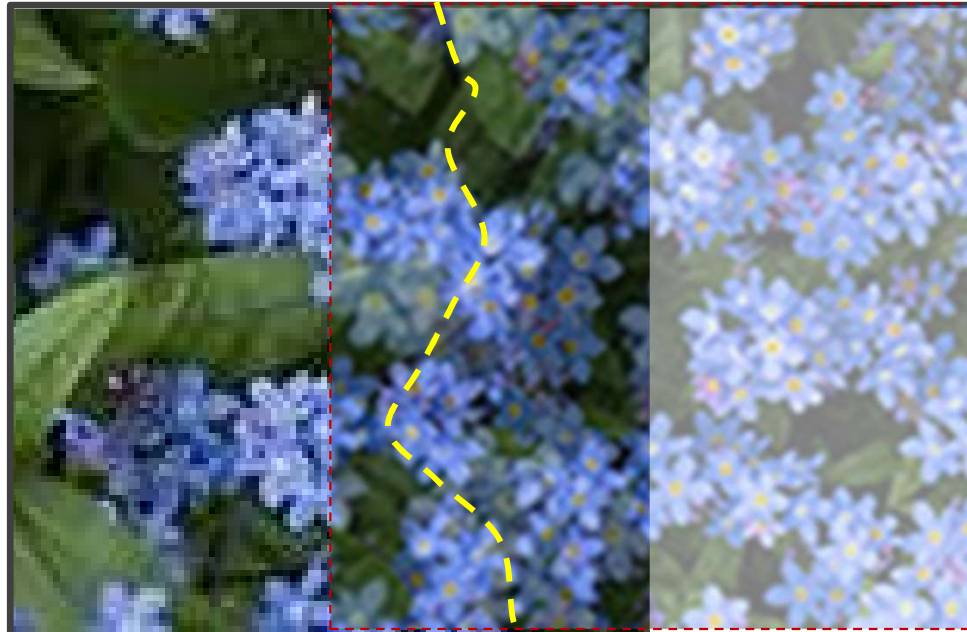
Output Image



block size

Quilting Algorithm

Output Image



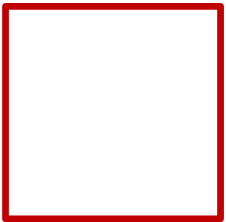
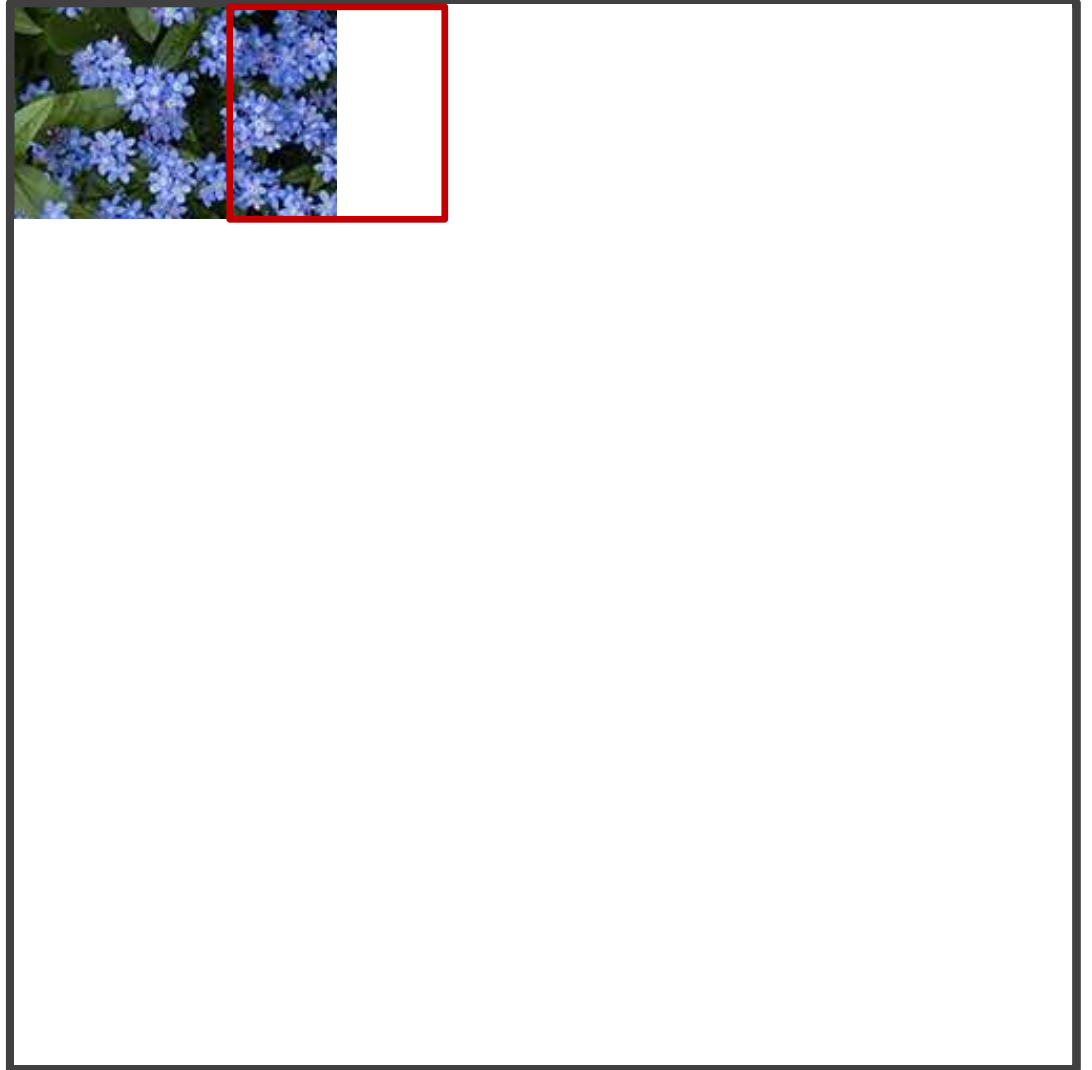
Selected Block

Quilting Algorithm

Input Image



Output Image



block size

Quilting Algorithm - Complexity

- **Error calculation:**

- $time\ complexity = O(b^2 \cdot w \cdot h \cdot n^2)$

- **Min-cut:**

- $time\ complexity = O(b^2 \cdot n^2)$

- **Where:**

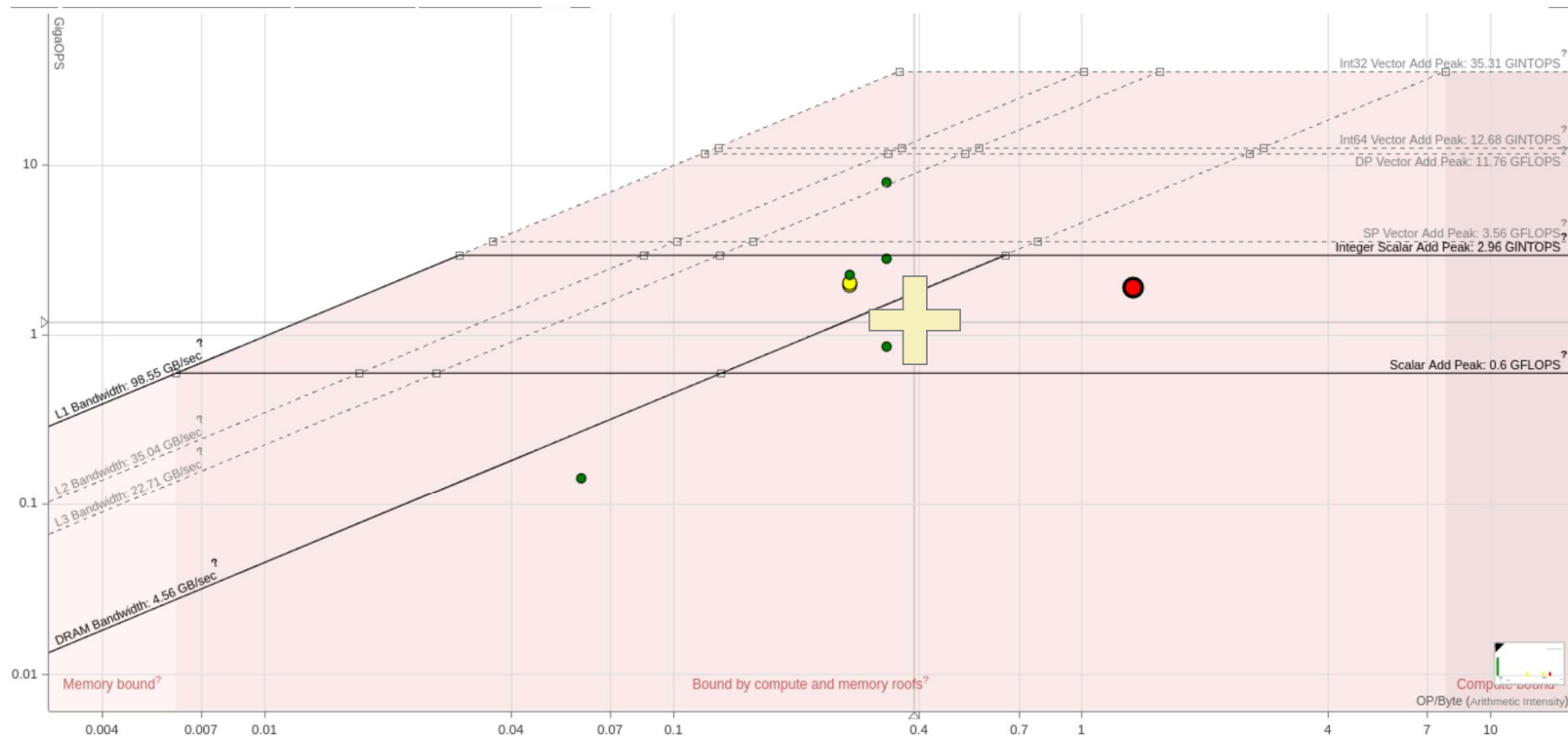
- b = block-size
 - w = input-width
 - h = input-height
 - n^2 = # output blocks

- **Cost function:**

- $C(b, w, h, n) = N_{add} + N_{mul} + N_{div} + N_{cmp}$

Quilting Algorithm - Baseline

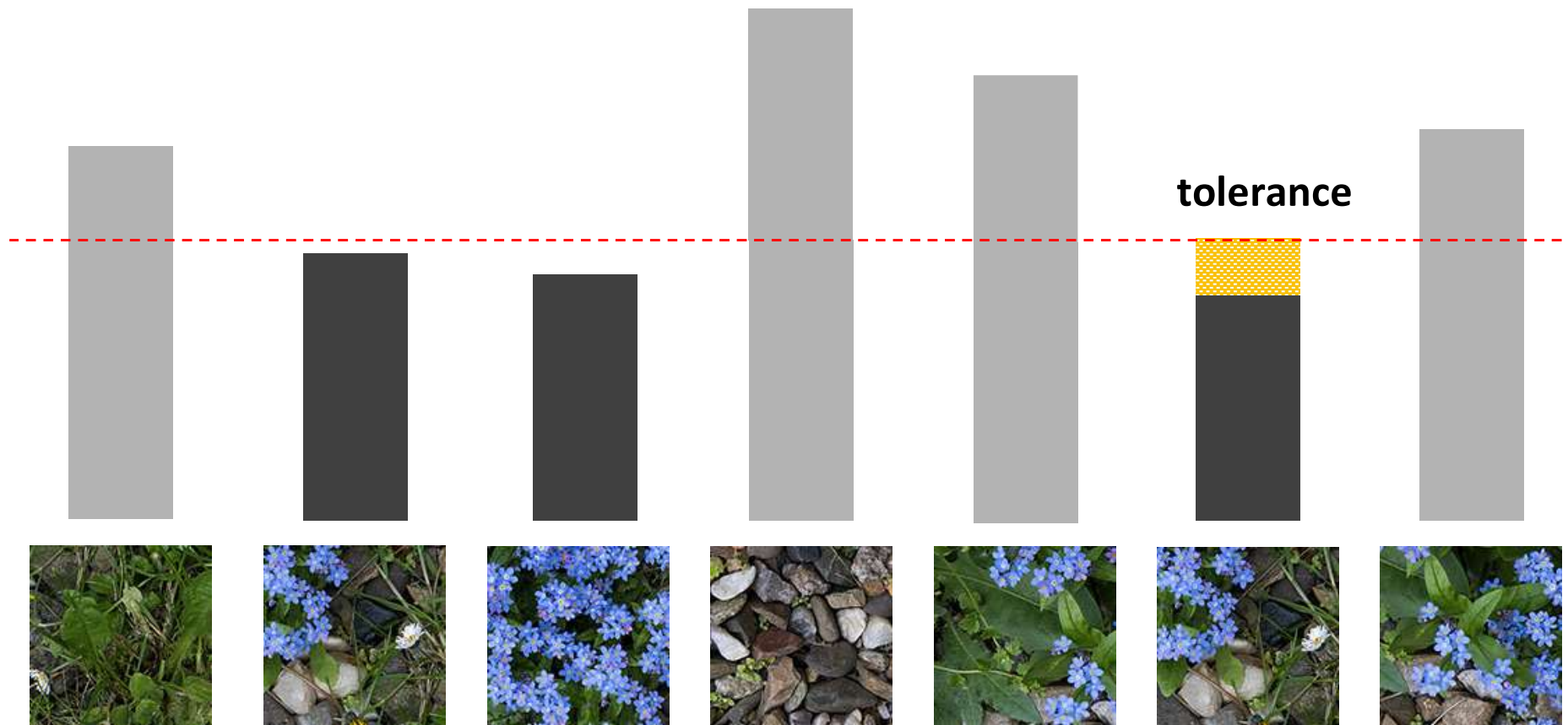
- Operational Intensity: 0.39 OP/B
- Performance: 1.18 GigaOPS
 - 40% integer scalar peak,
 - 3.3% integer vector peak



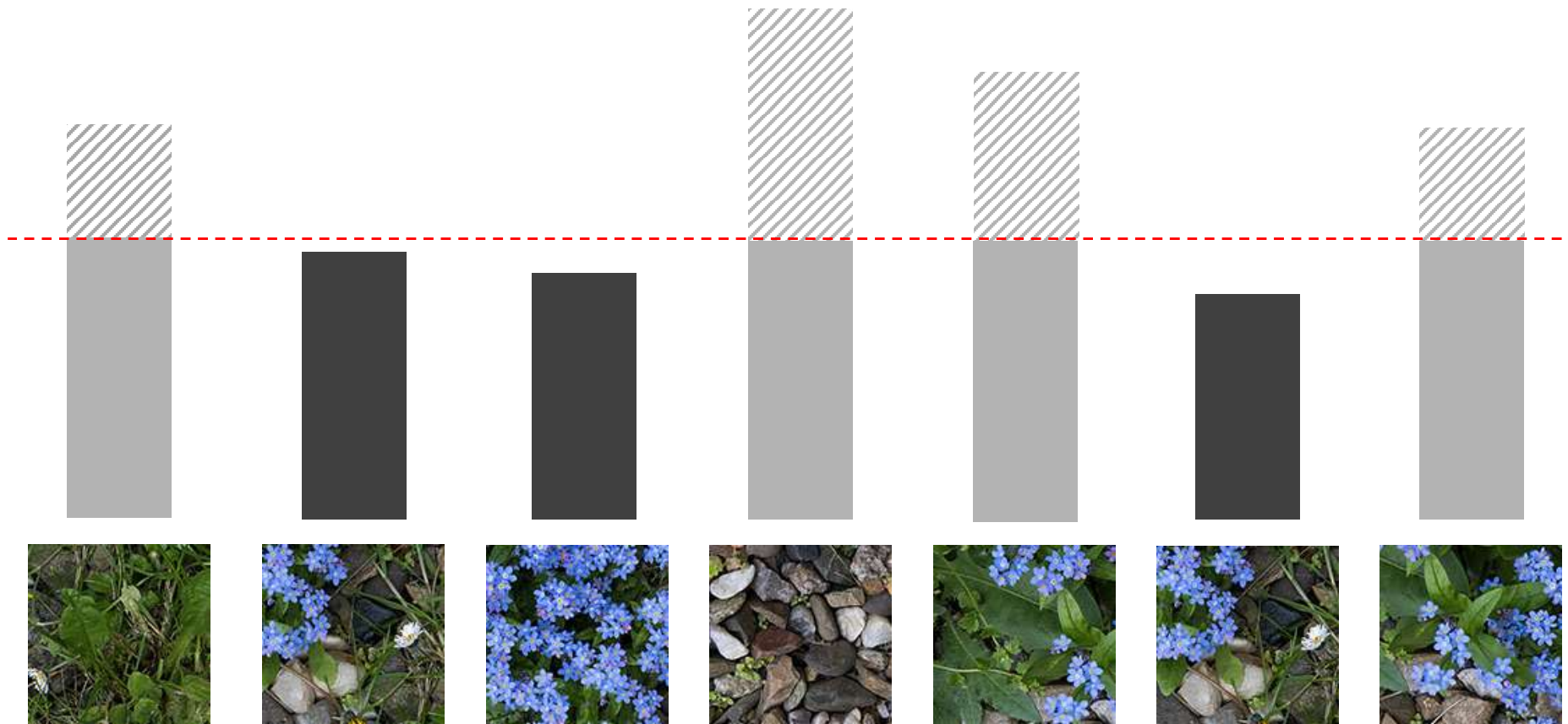
Optimizations

- Opt 1: Loop unrolling + smarter indexing using less computations
- Opt 2: attempt at blocking optimization for cache
- Opt 3: optimized min-cut function, removed transpose
- **Opt 4: removed min. error from index search**
- **Opt 5: change to separate arrays for each RGB value and cut mask from int to char**
- Opt 6: vectorization of min-cut overlap error and marge cut mask
- **Opt 6a: vectorization of calc-error (i16) and improved find block**
- **Opt 6b: initial early stopping**
- **Opt 6c: more early stopping with colors separated**
- **Opt 6d: optimized loop order for input image access**
- Opt 7: Loop unrolling + smarter indexing using less computations for separate RGB array version
- Opt 8: vectorization of calc-error function (i32)
- **Opt 9: finalized early stopping with vectorization**
- **Opt10: automatic color permutation**
- Opt11: minimum error prediction

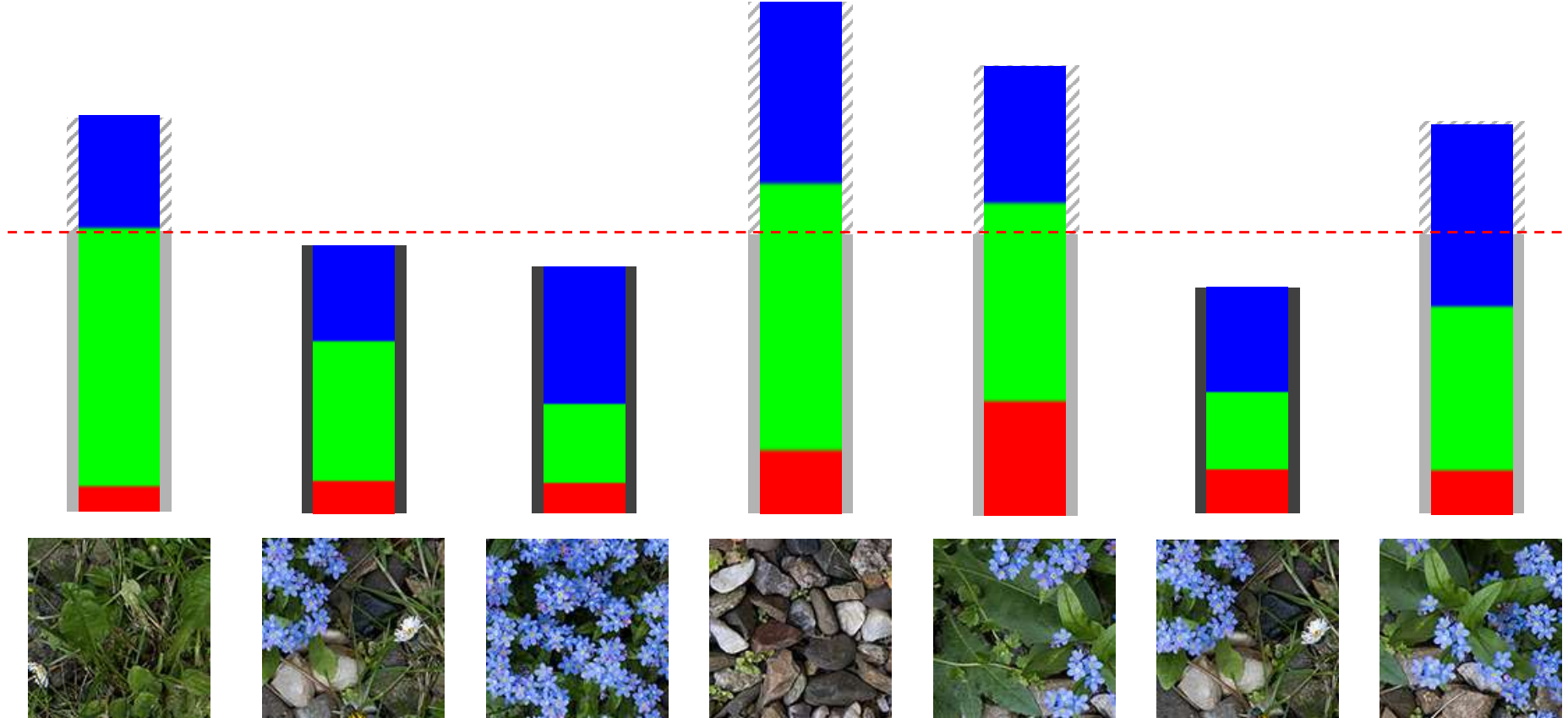
Error Calculation Early Stopping



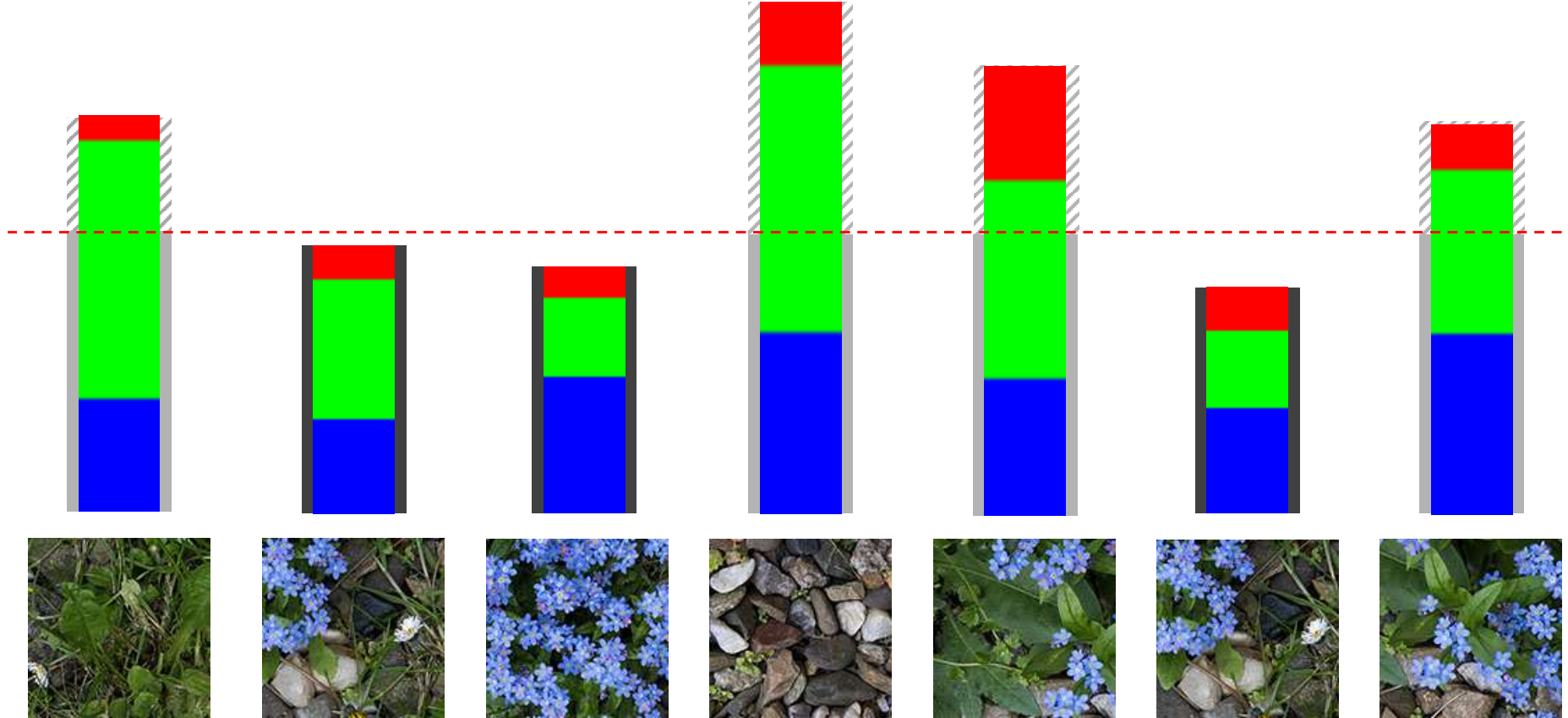
Error Calculation Early Stopping



Error Calculation with RGB Order



Error Calculation with BGR Order



Experimental Setup

- **AMD Ryzen 9 3900x @ 3.6 GHz**

- `gcc -Wall -Wextra -Wpedantic -O3 -ffast-math -march=native -mavx`

- **Intel i7-8550U @ 1.8 GHz**

- `gcc -Wall -Wextra -Wpedantic -O3 -ffast-math -march=native -mavx`

- **Intel i5-1135G7 @ 2.4 GHz**

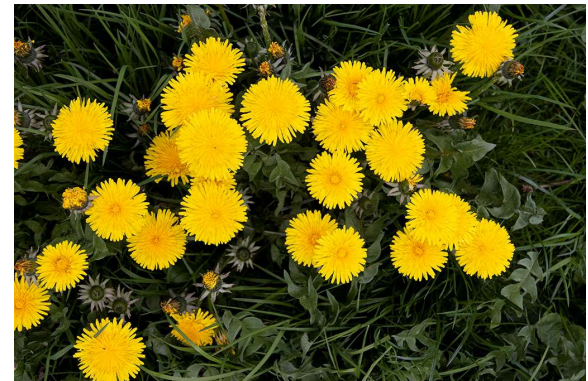
- `icx -Wall -Wextra -Wpedantic -O3 -ffast-math -march=native -mavx -qopt-zmm-usage=high`
- supports AVX-512

Experimental Setup

■ Input Images

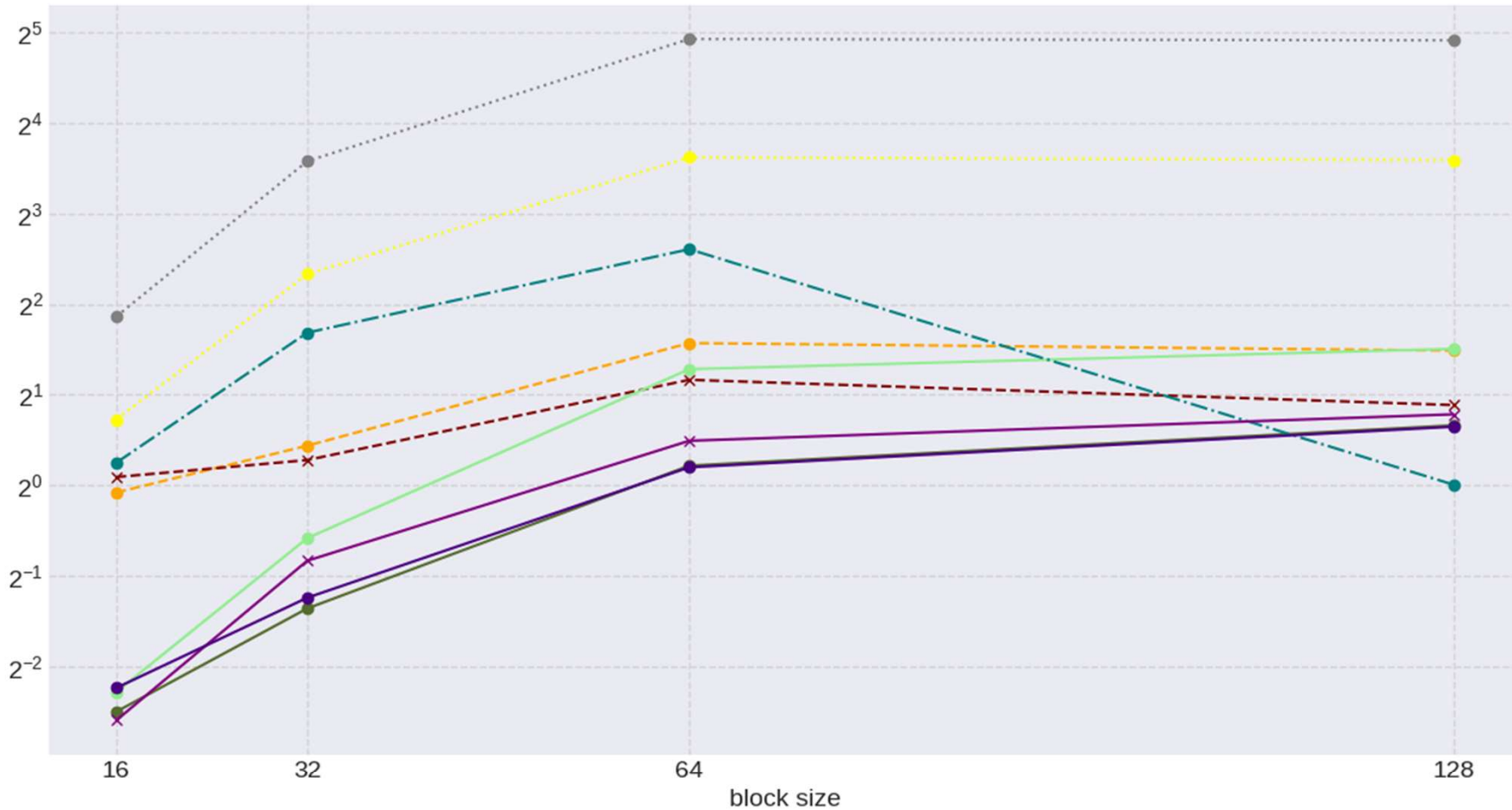
- red-radishes (192x192)
- blue-flowers (192x192)
- large-dandelion (1024x683)

■ Output Image size: 12x12 blocks



Quilting Runtime [Ryzen 9 3900x @ 3.6 Ghz, overlap = 0.5 * block_size]

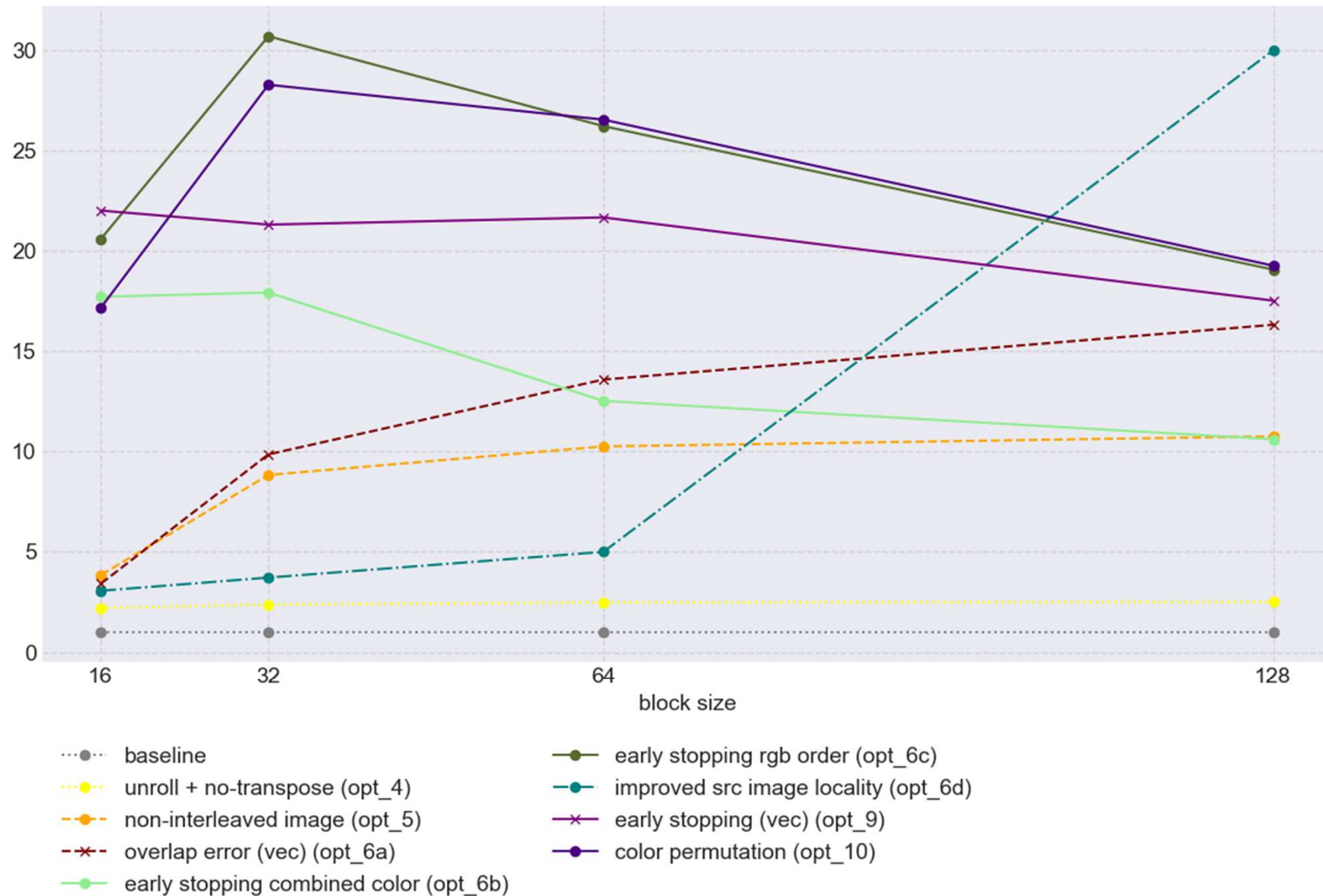
[seconds]



■ red-radishes (192x192), output size: 12x12

Quilting Speedup [Ryzen 9 3900x @ 3.6 Ghz, overlap = 0.5 * block_size]

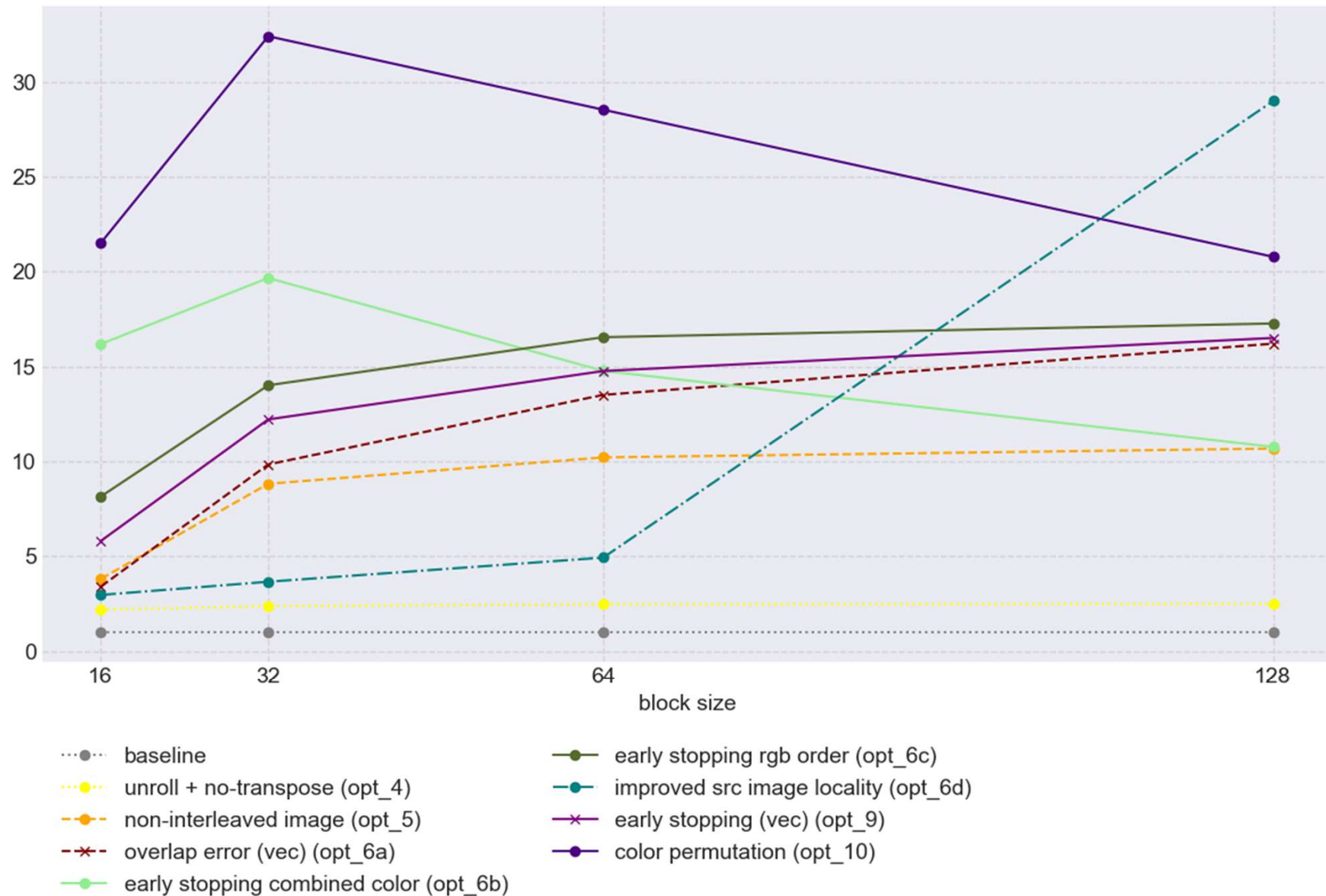
[speedup]



■ red-radishes (192x192), output size: 12x12

Quilting Speedup [Ryzen 9 3900x @ 3.6 Ghz, overlap = 0.5 * block_size]

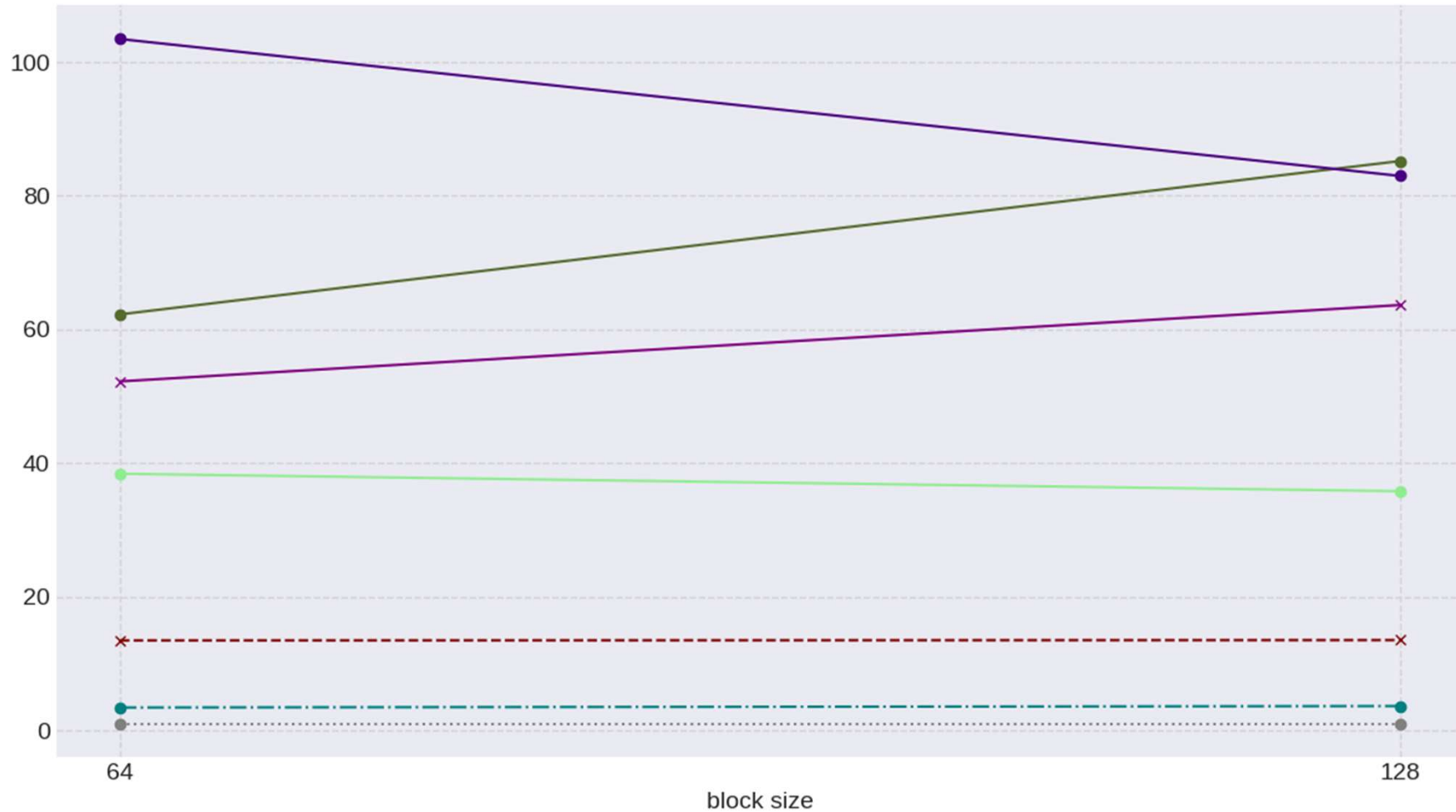
[speedup]



■ blue-flowers (192x192), output size: 12x12

Quilting Speedup [Ryzen 9 3900x @ 3.6 Ghz, overlap = 0.5 * block_size]

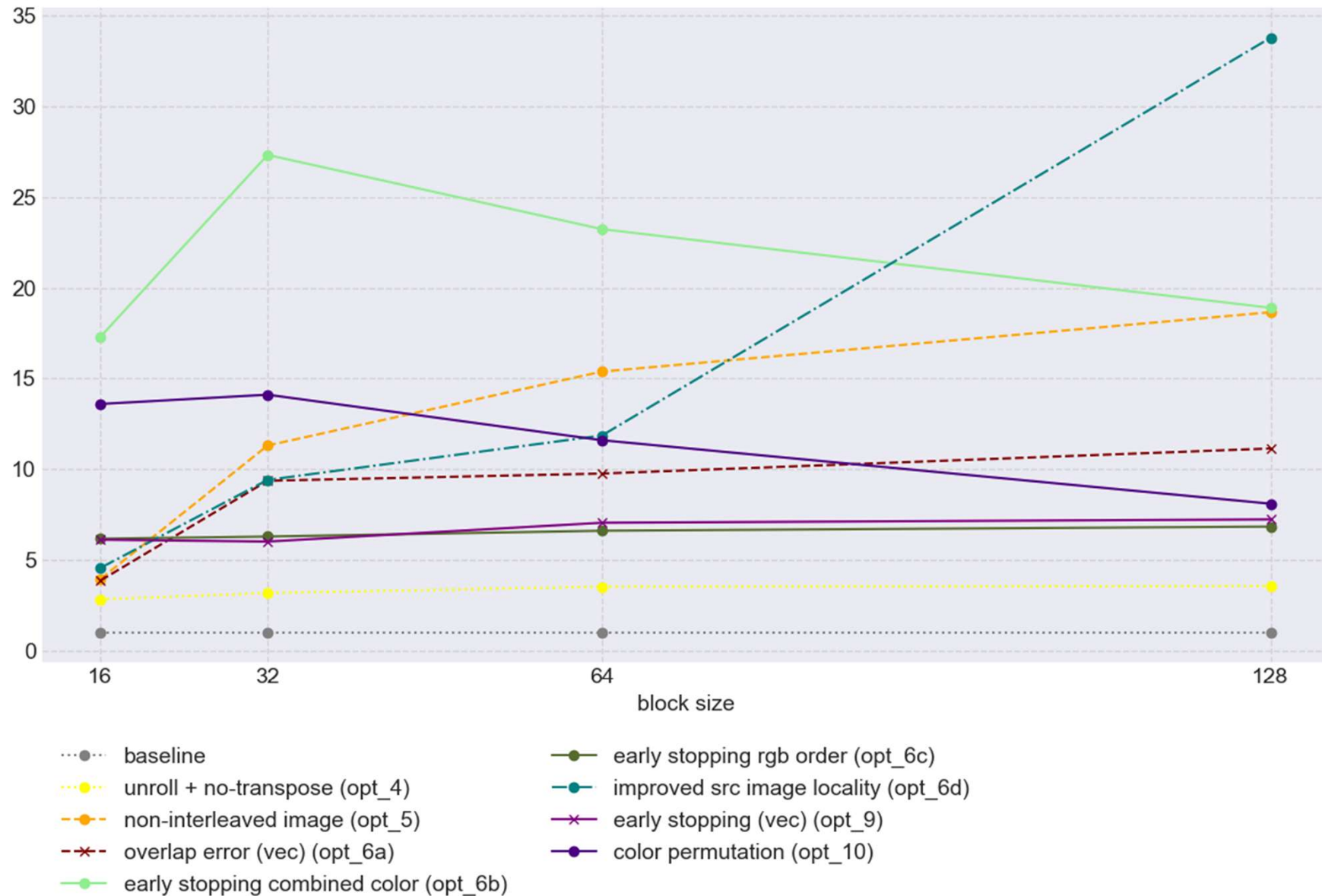
[speedup]



■ large-dandelion (1024x683), output size: 12x12

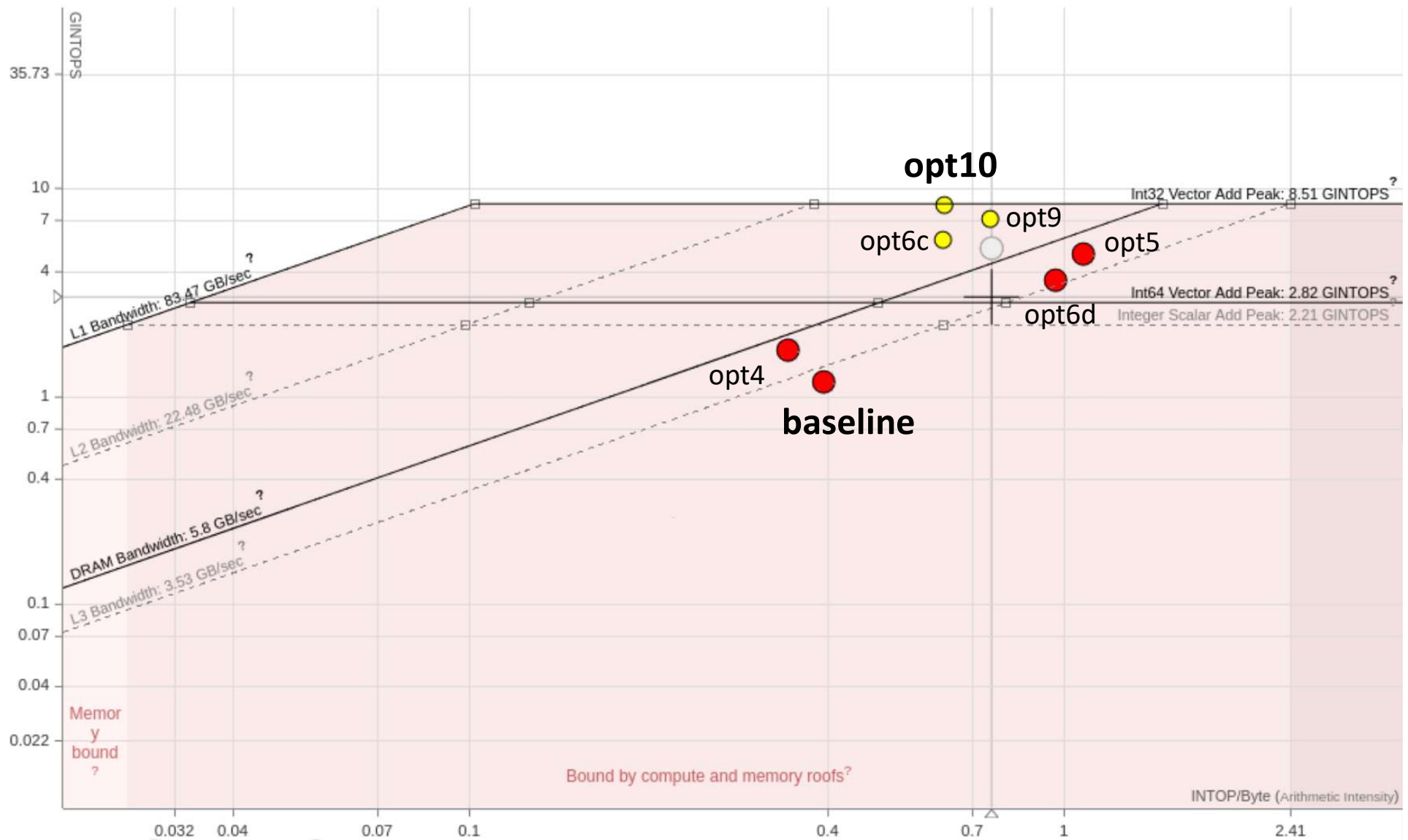
Quilting Speedup [i5-1135G7 @ 2.4 Ghz, overlap = 0.5 * block_size]

[speedup]



■ blue-flowers (192x192), output size: 12x12

Roofline Analysis



■ blue-flowers (192x192), block size: 64, overlap: 16, output size: 12x12

Questions?