N.V.I.D.I.A.

“Neat Video Interleaved Deblocking Integer Algorithm”

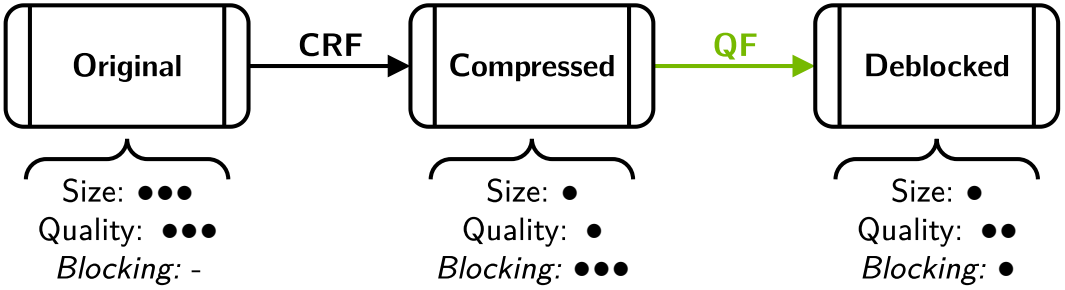
`GPU Programming` — 2022/2023

by *Fulvio Castello* (s301102)

Application attributes

- *Neat* — efficient and lightweight computation
- *Video* — computer vision-oriented project
- *Interleaved* — two-pass processing on individual frame channels
- *Deblocking* — aimed at improving visual quality
- *Integer* — comprising integer arithmetic only
- *Algorithm* — dynamic, conditional execution

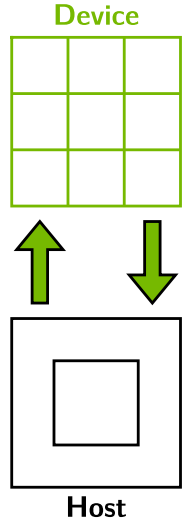
H.264 video compression



$$\text{CRF} = 0 \ldots 23 \ldots 51 \longleftrightarrow \text{QF} = 1 \ldots 127 \ldots 255$$

Iteration timeline

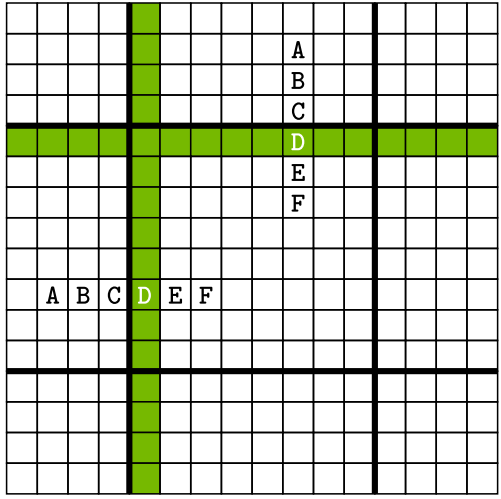
1. Input >> Frame
2. BGR $\rightarrow$ YCrCb
3. JPEG $\rightleftarrows$ Luma/Chroma
4. *HostToDevice*
5. Horizontal pass
6. **Intermediate frame**
7. Vertical pass
8. *DeviceToHost*
9. Luma/Chroma $\Rrightarrow$ JPEG
10. YCrCb $\rightarrow$ BGR
11. Frame >> Output



Proposed algorithm

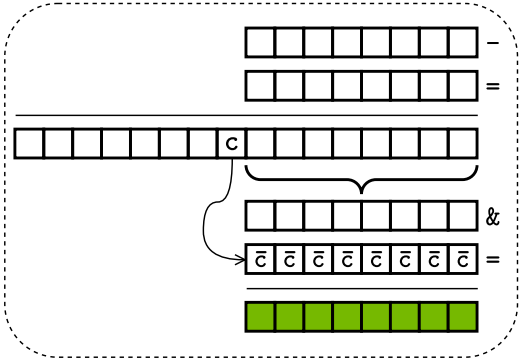
if $|B - C| < 5 \wedge |D - E| < 5$ **then**
  if $|C - D| < (2.0 \cdot \mathrm{QF})$ **then**
    $x \leftarrow D - C$;
    $a \leftarrow A + \frac{x}{8}$; $b \leftarrow B + \frac{x}{4}$;
    $c \leftarrow C + \frac{x}{2}$; $d \leftarrow D - \frac{x}{2}$;
    $e \leftarrow E - \frac{x}{4}$; $f \leftarrow F - \frac{x}{8}$;
  end
else
  if $|C - D| < (0.8 \cdot \mathrm{QF})$ **then**
    $x \leftarrow D - C$;
    $b \leftarrow B + \frac{x}{8}$; $c \leftarrow C + \frac{x}{2}$;
    $d \leftarrow D - \frac{x}{2}$; $e \leftarrow E - \frac{x}{8}$;
  end
end

8×8 JPEG macroblock:

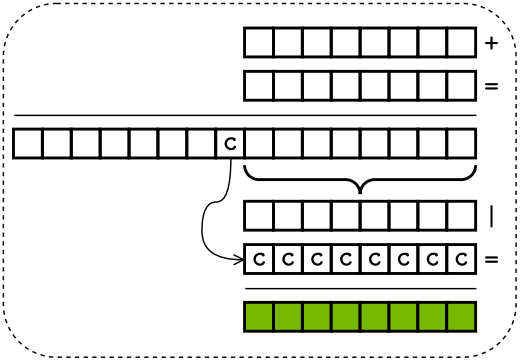


8-bit saturation

Negative overflow detection:



Positive overflow detection:



$A, B, C, D, E, F = 0 \ldots 255$

Execution flow

Synchronous host:

1. Sample current tick
2. Evaluate 1 pixel alone
3. Repeat for $W \cdot H$ times
4. Repeat for 3 channels
5. Accumulate elapsed time
6. Restart for second pass

↻ *Until no frames left*

Asynchronous device:

1. Upload 3 channels
2. Wait for 3 events
3. Accumulate elapsed time
4. Evaluate $W \cdot H$ pixels simultaneously
5. Wait for 3 events
6. Accumulate elapsed time
7. Repeat evaluation for second pass
8. Download 3 channels
9. Wait for 3 events
10. Accumulate elapsed time

↻ *Until no frames left*

Performance improvements

At least **300%+** faster than the CPU, depending on:

- GPU architecture
- number of SMs
- video format
- stream length
- framerate
- resolution
- original CRF
- selected QF

to obtain the same end result $\Longrightarrow$ *always* the superior choice.