
Peer reviewed version

Link to published version (if available):
10.1109/FPL.2009.5272247

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
http://www.bristol.ac.uk/pure/about/ebr-terms
A Toolset for the Analysis and Optimization of Motion Estimation Algorithms and Processors
Overview

• Motion estimation takes time, full search expensive for HD.
• Flexible reconfigurable processor.
• IDE to design and test algorithms.
• Toolset to configure the processor.
Saving time

• Motion estimation takes processor time.
• The design space to explore is large.
• Configuring the ME processor takes developer times.
The reconfigurable processor

• Advanced features such as rate distortion optimization using Lagrangian techniques.
• Multiple motion vector candidates allowed.
• Multiple sub-partition sizes allowed.
• Multiple reference frames allowed.
• Can do fractional pel motion estimation that can be used for the H.264 standard.
Simple and complex configurations

Simple (1 integer pel unit)

Complex (4 int. pel units, 1 frac. pel unit, Lagrangian optimizer)
## Processor performance and complexity evaluation

<table>
<thead>
<tr>
<th>Processor Configuration</th>
<th>Speed (cycles/MB, frames/second)</th>
<th>FPGA size (LUTs, slices)</th>
<th>Memory (BRAMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base configuration</td>
<td>625 cycles/MB, 39 fps</td>
<td>2289 LUTs, 1300 slices</td>
<td>21 BRAMS</td>
</tr>
<tr>
<td>(1 integer-pel execution unit)</td>
<td></td>
<td></td>
<td>(2 ref. areas, 112×128 pixels)</td>
</tr>
<tr>
<td>Complex configuration</td>
<td>234 cycles/MB, 104 fps</td>
<td>7074 LUTs, 3703 slices</td>
<td>72 BRAMS</td>
</tr>
<tr>
<td>(4 integer-pel execution units)</td>
<td></td>
<td></td>
<td>(2 ref. areas, 112×128 pixels)</td>
</tr>
</tbody>
</table>

**Video sequence:** 1080p *crowdrun* from SVT HD multi format test set

**FPGA part:** Virtex-4 SX35, 200 MHz clock frequency

**Algorithm:** 6-point hexagonal search (up to 8 steps), then 8-point square
Designing block-matching algorithms

- Estimo C: high-level language for search algorithms.
- Compiler targets the reconfigurable processor.
- No need to know how hardware works.
- Compiled program works across all configurations.

```c
s = 8; // initial step size
cHECK(0, 0);
cHECK(0, s);
cHECK(0, -s);
cHECK(s, 0);
cHECK(-s, 0);
update;
do {
    s = s / 2;
    for (i = 1 to 5 step 1) {
        cHECK(0, s);
        cHECK(0, -s);
        cHECK(s, 0);
        cHECK(-s, 0);
        update;
        if (WINID == 0)
            #break;
    }
} while (s > 1);
for (x = -0.5 to 0.5 step 0.25)
    for (y = -0.5 to 0.5 step 0.25)
        cHECK(x, y);
update;
```
Cycle-accurate simulator

- Analysing processor configurations on hardware takes time.
- Using simulator, no need for synthesizing hardware and configuring board.
- No hardware required for evaluation of processor.
The IDE
The IDE
The IDE

Fps against bit rate for different sequences

Plot title: Fps against bit rate for different sequences
X-axis: Bit rate
Y-axis: Frames / second (parallel)
Point labels: Configuration
Area: Logic cells
The IDE

---

**Cycle Accurate Model**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>PSNR (dB)</th>
<th>Bit rate (kbit/s)</th>
<th>Frames / second</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedestrian</td>
<td>41.3</td>
<td>6793.27</td>
<td>20.7457</td>
</tr>
<tr>
<td>rush hour</td>
<td>41.874</td>
<td>4811.23</td>
<td>16.0346</td>
</tr>
<tr>
<td>station</td>
<td>40.705</td>
<td>2713.17</td>
<td>16.8015</td>
</tr>
<tr>
<td>sunflower</td>
<td>42.326</td>
<td>3585.56</td>
<td>16.7534</td>
</tr>
<tr>
<td>tractor</td>
<td>39.171</td>
<td>12398.4</td>
<td>15.7436</td>
</tr>
<tr>
<td>pedest 2</td>
<td>41.3</td>
<td>6793.27</td>
<td>20.7457</td>
</tr>
</tbody>
</table>

- **Rename**  
- **Export configuration**  
- **Details**  
- **Delete**  
- **Clear**

---

**Processor configuration:**

- **Label:** tractor
- **Program memory:** D:/projects/hex/estimo.output/program.bin
- **Point memory:** D:/projects/hex/estimo.output/patterns.bin
- **Full-pel units:** 1
- **Sub-pel units:** 1
- **Smallest partition:** 16x16
- **MV cost optimization:** enabled
- **MV candidates:** enabled
- **Logic cells:** 9732

**Results:**

- **Bit rate (kbit/s):** 12398.4
- **PSNR (dB):** 39.171
- **FPS:** 15.7436
- **Cycles / macroblock:** 1556.81
- **Energy / macroblock (nJ):** 22.3792

- **Full- and sub-pel in parallel:**
  - **FPS:** 26.1083
  - **Cycles / macroblock:** 938.773
  - **Energy / macroblock (nJ):** 13.4949

**Video data:**

- **Video file:** D:/test_sequences/1080p/tractor.yuv
- **Resolution:** 1920x1080
- **Frames processed:** 50
- **QP (0 is lossless):** 26
- **Reference frames:** 1
Prototype implementation
Summary

- Programmable and configurable processor supports HD motion estimation (supports H.264, MPEG-4, MPEG-2, VC-1, AVS).
- Motion Estimation Processor: http://www.opencores.org/
- Estimo C compiler for easy development of proprietary block-matching algorithms.
- FPGA-based PCI demonstration board available.
- Cycle-accurate simulator for quick evaluation and design space exploration.
- SharpEye IDE: http://sharpeye.borelspace.com/