
Peer reviewed version

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

Link to publication record in Explore Bristol Research

PDF-document
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></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></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

**Cycle Accurate Model**

**Motion-estimation search configuration**

- **Program memory:** D:/projects/hex/estino.output/program.bin
- **Points memory:** D:/projects/hex/estino.output/patterns.bin
- **Full-pel execution units:** 2
- **Sub-pel execution units:** 1
- **Smallest partition:** 16x16
- **Enable Motion Vector cost optimization**
- **Enable Motion Vector candidates**

**Video data**

- **Video file:** D:/test_sequences/1080p/pedestrian_area.yuv
- **Resolution:** 1920 x 1080
- **Maximum number of frames:** 50
- **QP (0 is lossless):** 26
- **Reference frames:** 1

**Results**

**Processor configuration:**

- **Label:** Cfg 24
- **Full-pel units:** 2
- **Sub-pel units:** 1
- **Smallest partition:** 16x16
- **MV cost optimization:** enabled
- **MV candidates:** enabled
- **Logic cells:** 11287

**Video data:**

- **Frames processed:** 50
- **QP:** 26
- **Reference frames:** 1

**Results:**

- **Bit rate (kbit/s):** 6793.27
- **PSNR (dB):** 41.3
- **FPS:** 20.7457
- **Cycles / macroblock:** 1181.44
- **Energy / macroblock (nJ):** 20.9706
- **Full- and sub-pel in parallel:**
  - **FPS:** 31.8295
  - **Cycles / macroblock:** 770.033
  - **Energy / macroblock (nJ):** 13.6681

[Run] [Stop] [New Plot] [Table]
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)
Area: Logic cells

Legend:
- Pedestrian
- Station
- Blue sky
- Sunflower
- Rush hour
- Tractor
- Pedestrian 2

Department of Electrical and Electronic Engineering
The IDE
Prototype implementation

Diagram showing the connections and components of a prototype implementation:

- FPGA
- PCI Core
- AHB/APB Bridge
- Memory Controller
- UART Interface
- Timer Unit
- Interrupt Controller
- AHB Master
- AHB Slave
- APB Slave
- On-board memory

Diagram includes a visual representation of the connections between these components.
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/