Implementation of a Motion Estimation Algorithm for Intel FPGAs using OpenCL

Manuel de Castro[1], Roberto R. Osorio[2], Yuri Torres[1],
Arturo González-Escribano[1], Diego R. Llanos[1]
[1]**Universidad de Valladolid, Spain**, [2]**Universidade da Coruña, Spain**
mdcc.prof@gmail.com roberto.osorio@udc.es
{yuri.torres|arturo|diego}@infor.uva.es

CMMSE 2022, July 2022

Universidad de Valladolid

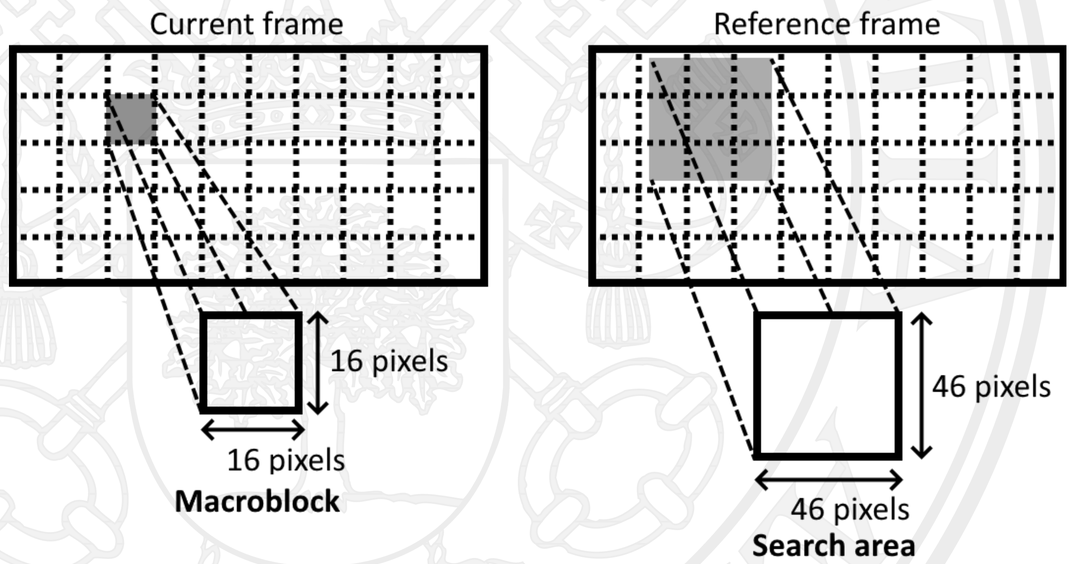
Grupo Trasgo
Universidad de Valladolid

Introduction

- **Block-matching Motion Estimation** is widely used in advanced video codecs to achieve high compression rates
  - √ Blocks of pixels **compared** between **current and reference** frame; allows to encode blocks as **motion vectors** (spatial references)
  - √ Responsible for most of the compression in any video codec
  - × **Computationally intensive task**, usually implemented by means of application specific hardware (ASICs, **FPGAs**, ...)
- **FPGA solutions** are mostly developed using **hardware description languages** (Verilog and VHDL)
  - √ Allow for **low-level optimizations**
  - × **High development complexities**, specially for HPC software developers
  - ⇒ Solutions using **higher-level languages** are desired
- **Proposal: OpenCL** [1] **solution targeting Intel FPGAs**
  - ⇒ **Intel FPGA SDK for OpenCL** with **task kernels**
  - ⇒ Test the **expressiveness** of OpenCL as a design language for video processing applications

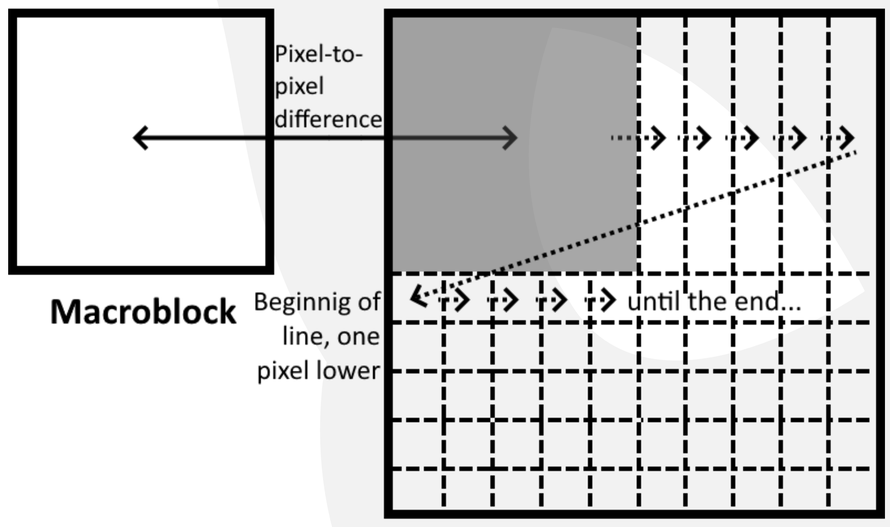
Current & reference frame blocks

- Divide current frame in equally-sized **macroblocks**
- **Search** for closest match in nearby area in **reference frame**
  - ⇒ All possible matches within the search area are **candidate blocks**
    - ⇒ candidate motion vectors



Block-matching Motion Estimation

- **Sum of Absolute Differences** (SAD) as similarity metric between blocks
  - ⇒ Lower SAD = higher similarity
  - ⇒ Less costly than Sum of Squared Errors (SSE).
- Straightforward **full-search** algorithm



Proposal

- Works with **Full HD frames** (1920 × 1080 pixels), **luminance component** only
- Uses **16×16-pixels macroblocks** and **46×46-pixels search areas** (961 motion vectors tested per macroblock)
- **Two alternatives developed** to address macroblocks at the **borders** of the frames
  - **b)** Add logic to **detect frame borders**
  - **a)** Work with **expanded frames** by repeating border pixels
- Computes each candidate motion vector **fully in parallel**
- **Comparison** with references
  - Compiled with **gcc 7.5.0** and **at least -O2** optimizations
  - Intel Xeon Platinum 8256 CPU vs Intel Stratix 10 FPGA

Version	ms/frame	frames/s
Sequential reference	1627.39	0.614
MMX vector registers (8 bytes)	145.31	6.882
SSE vector registers (16 bytes)	126.64	7.896
-O3 optimizations (uses SSE)	89.49	11.174
FPGA border detection logic	90.12	11.096
FPGA expanded frames	88.43	11.309

Evaluation

- Compiled for **Intel Stratix 10 FPGA** [2]
- **Version with additional logic for border detection**
  - **Working frequency:** 308 MHz
  - **Resource usage:**

	ALM	REG	MLAB	RAM	DSP
Entire system	247 311 (27%)	433 503 (12%)	783 (1%)	1 198 (10%)	3 (0%)
Kernel system	52 468.9 (6%)	149 222 (4%)	783 (1%)	767 (7%)	5 (0%)
ME kernel logic (estimated)	24 122 (3%)	92 840 (2%)	1 294 (1%)	678 (6%)	2.5 (0%)
Available	933 120	3 732 480	93 312	11 721	5 760

- **Version that works with expanded frames**
  - **Working frequency:** 316 MHz
  - **Resource usage:**

	ALM	REG	MLAB	RAM	DSP
Entire system	244 913 (26%)	421 377 (11%)	990 (1%)	1 187 (10%)	0 (0%)
Kernel system	50 536 (5%)	137 020 (4%)	990 (1%)	756 (6%)	0 (0%)
ME kernel logic (estimated)	20 922.5 (2%)	72 446 (2%)	1 440 (2%)	663 (6%)	0 (0%)
Available	933 120	3 732 480	93 312	11 721	5 760

Conclusion and future work

- We present a **block-matching motion estimation implementation for Intel FPGAs using OpenCL** that is **fully parallel**.
- **Two versions** have been developed, that deal differently with the borders of the frames
- OpenCL **easens** development of FPGA applications and offers **competitive results**
- OpenCL presents some downsides: **high compilation times** and **lack of estimated design latency**
- **Future work:** performance comparison with **parallel** (OpenMP) and **GPU** implementations.

References

[1] Khronos OpenCL Working Group et al. The OpenCL Specification, version 1.0.29, 8 December 2008

[2] Intel Intel Stratix 10 FPGAs & SoC FPGA, https://www.intel.com/content/www/us/en/products/details/fpga/stratix/10.html

Acknowledgements

This research has been partially funded by the Spanish Ministerio de Economía, Industria y Competitividad with the ERDF program of the European Union, project PCAS (TIN2017-88614-R); Junta de Castilla y León - FEDER Grants, projects PROPHET and PROPHET-2 (VA082P17, VA226P20); the Fundación General de la Universidad de Valladolid by means of the contract 062/204051; the Spanish Ministerio de Ciencia e Innocación (PID2019-104184RB-I00, AEI/FEDER/EU, 10.13039/501100011033); and by the Xunta de Galicia and FEDER Grants (Centro de Investigación de Galicia 2019-2022, ref. ED431G 2019/01, as well as the Programa de Consolidación y Estructuración de Unidades de Investigación Competitivas, ref. ED431C 2017/04).