ISCAS 2021
Daegu, KOREA, MAY 22-28
IEEE International Symposium on Circuits and Systems

IEEE
Advancing Technology for Humanity

CAS

FPGA acceleration of the Horn and Schunck Hierarchical Algorithm

Ilias Bournias, Roselyne Chotin and Lionel Lacassagne

Sorbonne Université, CNRS, LIP6

2021 IEEE International Symposium on Circuits and Systems
May 22-28, 2021 Virtual & Hybrid Conference

SCIENCES SORBONNE UNIVERSITÉ

cnrs

LIP6

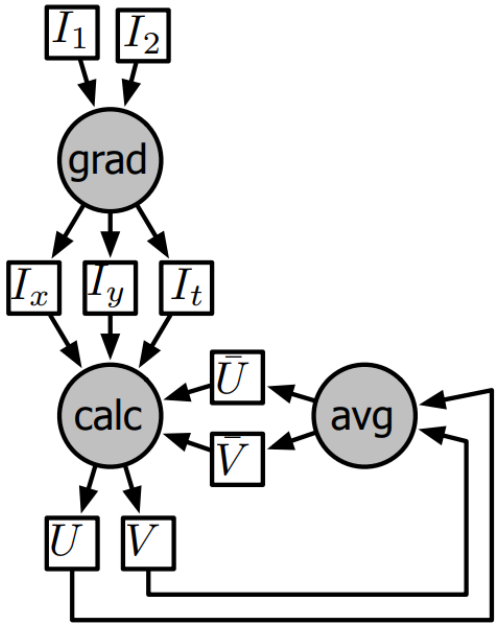
Outline

- **Introduction**
- Proposed Architecture
- Design Space Exploration
- Results of the Implementation
- Conclusion and Future Work

Introduction

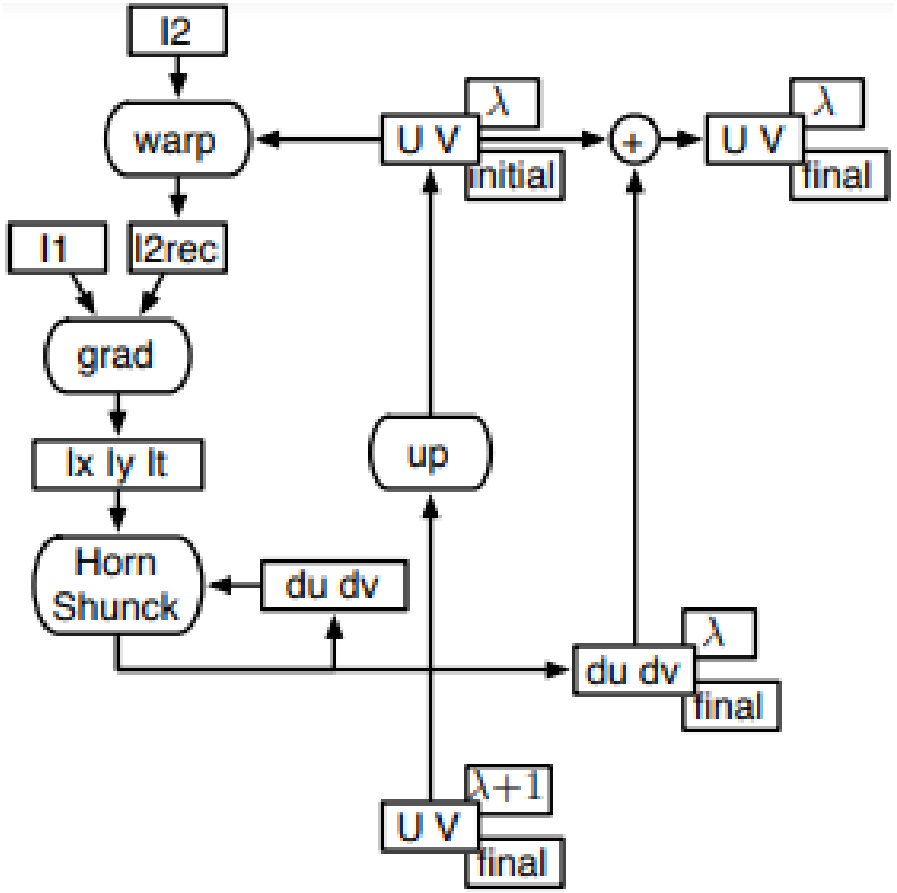
- **Optical Flow**
  - Object detection
  - Motions compensation
  - Autonomus driving
- **Common optical Flow algorithms used**
  - Mono-scale or multi-scale Lukas-Kanade in CPUs ,GPUs,FPGA
  - Mono-scale Horn and Schunck for CPUs ,GPUs , FPGAs
  - Multi-scale Horn and Schunck only for CPUs, GPUs
- **This work**
  - Multi-scale Horn and Schunck algorithm the first for FPGAs

Introduction



- **Mono-scale Horn and Schunck**
  - $u = \bar{u} + \mathrm{I}_x \frac{\mathrm{I}_x u + \mathrm{I}_y v + \mathrm{T}_t}{\alpha^2 + \mathrm{I}_x^2 + \mathrm{I}_y^2}$
  - $v = \bar{v} + \mathrm{I}_y \frac{\mathrm{I}_x v + \mathrm{I}_y u + \mathrm{T}_t}{\alpha^2 + \mathrm{I}_x^2 + \mathrm{I}_y^2}$
  - I_x, I_y, T_t estimated with a $2x2x2$ kernel
  - $(u, v) \leq \frac{pixel}{frame}$

Introduction



- **Multi-scale Horn and Schunck extension**
- From $(u,v)^{\lambda+1}_{final}$ of level $\lambda+1$
- $(u,v)^{\lambda}_{final} = 2\ x\ upscale(u,v)^{\lambda+1}_{final}$ for level λ
- H&S iteration provide $(\delta u, \delta v)$.
- $(u,v)^{\lambda}_{final} = (u,v)^{\lambda}_{final} + (u,v)^{\lambda}$

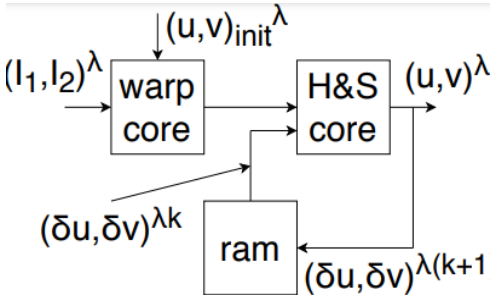
Introduction

- **Contributions**
  - 3-level pyramid with a x2-factor (20,10,5)
    - motion estimation in the range of $(u, v) < |7|$.
    - image size 1024x1024 pixels
  - Parallelism degree
    - number of the H&S cores,
    - floating point format
    - warp interpolation
      - bi-cubic
      - bi-linear
- **Goal**
  - Not address accuracy
  - Computation speed
  - Size of the design

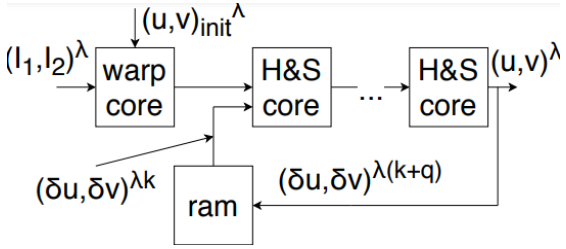
Outline

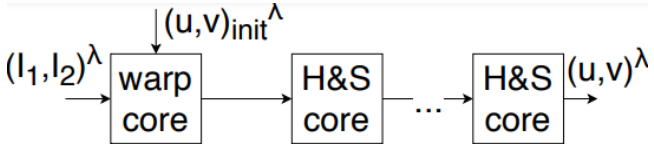
- Introduction
- **Proposed Architecture**
- Design Space Exploration
- Results of the Implementation
- Conclusion and Future Work

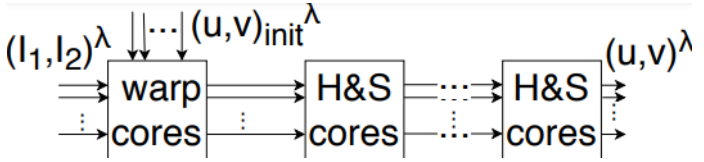
Proposed architecture



- **Horn and Schunck core**
  - Standard iterative (I)
  - Partial iterative (P)
  - Fully Pipeline (F)
  - Fully Pipeline with π parallel cores (F_{π})

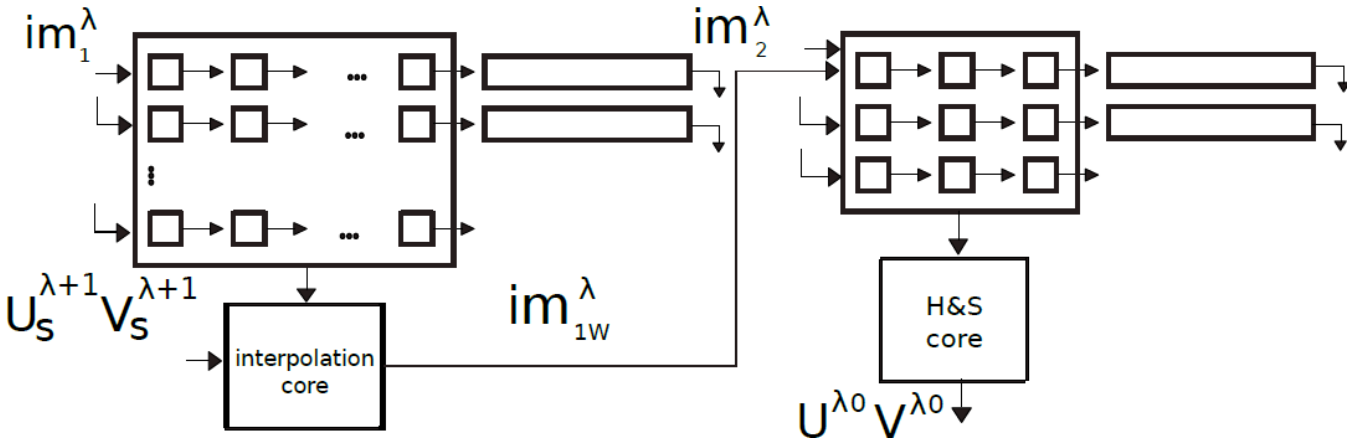






Proposed architecture

- **Warping core**
  - Bi-linear 2x2 neighborhood
  - Bi-cubic 4x4 neighborhood
  - Interpolate one or more pixels per clock
    - streaming window
- **Sum, Up-scaling core**
  - Sum the velocities of previous level with those of the current level.
  - Up-scale for next level



Proposed architecture

- **Pyramid Creation**
  - 5x5 Gaussian kernel (down-sampling)
- **Pipeline and Parallelism**
  - Up-scaling, warping , H&S and sum can be pipelined

Outline

- Introduction
- Proposed Architecture
- **Design Space Exploration**
- Results of the Implementation
- Conclusion and Future Work

Design space exploration

- **Depending on the number of iterations, different Horn and Schunck modes**
- **With respectively 20 , 10 , 5 iterations for $level_2$, $level_1$, $level_0$ and 10 available cores**
  - $level_2$ partial pipeline mode
  - $level_1$ fully pipeline mode
  - $level_0$ fully pipeline parallel
- **Computation Time**
  - $T = \frac{Height * weight * (i_0 + \frac{i_1}{4} + \frac{i_2}{16})}{f * \Pi} + lat * \frac{1}{f}$
- **Logic Block**
  - $N = \Pi * N_{HS} + \left\lceil \frac{\Pi}{i_0} \right\rceil * (N_{warp}) + N_1$
  - $N_1 = \left\lceil \frac{\Pi}{i_0} \right\rceil * N_{sum} + \left\lceil \frac{\Pi}{i_0} \right\rceil (N_{upscaling}) + N_{downsampling}$

Design space exploration

- **On chip memory**
  - $M = \Pi * M_{HS} + M_{warp} + M_1 + M_2$
  - $M_1 = M_{sum} + M_{upscaling} + M_{downsampling}$
  - $M_2 = width * 2 * \frac{Height * weight}{16}$
- **External memory bandwidth**
  - Maximum bandwidth defined by $level_0$ and $level_1$

Outline

- Introduction
- Proposed Architecture
- Design Space Exploration
- **Results of the Implementation**
- Conclusion and Future Work

Results of implementation

- **Implementation on FPGA Altera Stratix V with FloPoCo library**
- **Mono-scale & Mono-core results for F_{32} and F_{16}**

Core F_{32}	L. U.	Registers	Memory
Bi-linear Interp.	7.6k	18.2k	605.6k
Bi-cubic Interp.	44.9k	105.8k	624.2k
H&S	9.6k	19.7k	270.4k
Sum	4.3k	11.7k	16.7M
Up-scal.	2.8k	9.7k	135.6k
Down-sampl.	24.3k	68.6k	190.4k

Core F_{16}	L. U.	Registers	Memory
Bi-linear Interp.	4.5k	13.2k	292.2k
Bi-cubic Interp.	17.9k	51.3k	326.8k
H&S	4.6k	13.7k	150.3k
Sum	1.8k	5.7k	9.4M
Up-Scal.	1.7k	2k	67.1k
Down-Sampl.	13.3k	23.1k	94.3k

Results of implementation

- **Design Space exploration results**

Implem. F_{16}	Mono v_1	Multi v_2	Multi v_3	Multi v_4	Multi v_5	Multi v_6	Multi v_7
Iterations	20	20,10,5	20,10,5	20,10,5	20,10,5	20,10,5	20,10,5
Cores (Π)	20	1	5	10	20	10	20
Mode	F	I^{20}, I^{10}, I^{5}	P^{4}, P^{2}, F	P^{2}, F, F_2	F, F_2, F_4	P^{2}, F, F_2	F, F_2, F_4
Interp.	-	Bi-cub	Bi-cub	Bi-cub	Bi-cub	Bi-lin	Bi-lin
L. U.	93.5k	53.5k	67k	109.2k	192.2k	82.4k	148.5k
Registers	250k	150.7K	189.7k	293.6k	529.4	208k	382.2k
Memory	3.1M	10.2M	13.2M	11.6M	13.5M	11.6M	13.3M
frequency	339	276	293	281	270	286	274
fps	310	29.4	149	275	477	277	484

Results of implementation

- **Comparison with other works**

Implem.	Algo.	size	Π	Frame R.	Thr.	Freq.
v_2	Multi H&S	1024x1024	1	29.4	30.8 (4.4)	291(40)
v_4	Multi H&S	1024x1024	10	275	288 (40.6)	296(40)
v_7	Multi H&S	1024x1024	20	484	507 (70.4)	278(40)
[Kunz]	Mono H&S	640x512	1	30	9.2	295
[Tomasi]	Multi Ph. B.	640x480	-	31.5	9.6	45
[Barranco]	Multi L&K	640x480	-	32	9.8	83
[Seyid]	HBM + Ref.	640x480	-	39	12	200
[Seong]	Mono L&K	800x600	-	170	81.6	94
[Komorkiewicz]	Mono H&S	1920x1080	128	84	174.2	150
[Komorkiewicz]	Mono H&S	1920x1080	32	96.5	200	200
[Kunz]	Mono H&S	4096x2304	20	30	283.1	-
[Ishii]	Mono L&K	1024*1024	-	1000	1048	90

Results of implementation

- **2 clock frequencies**
- **Fastest except [Ishii]**
- **Fastest than previous mono-scale H&S designs**
- **Achieves at least x3.9 throughput**
- **With lower frequency**

Outline

- Introduction
- Proposed Architecture
- Design Space Exploration
- Results of the Implementation
- **Conclusion and Future Work**

Conclusion and Future Work

- **Parametrable hierarchical implementation of gradient based H&S**
- **Smallest design with 30.8 Mpixel/s and 23% usage of the Logic Blocks**
- **Fastest at 507 Mpixel/sec and 63% of logic utilization**
- **Multi-scale with range of 7 calculated velocities**

- **Future Work**
  - deal with accuracy
  - exploring fixed point format

References

[Kunz] et al. “An FPGA-optimized architecture of horn and schunck optical flow algorithm for real-time applications”

[Tomasi] et al. “High-Performance Optical-Flow Architecture Based on a Multi-Scale, Multi-Orientation Phase-Based Model”

[Barranco] et al. “Parallel Architecture for Hierarchical OpticalFlow Estimation Based on FPGA”

[Seyid] et al.“FPGA-Based Hardware Implementation of Real-Time Optical Flow Calculation”

[Seong] et al. “A Novel Hardware Architecture of the Lucas–Kanade Optical Flow for Reduced Frame Memory Access”

[Komorkiewicz] et al.“Efficient hardware implementation of the Horn-Schunck algorithm for high resolution real-time dense optical flow sensor”

[Ishii] et al.“High-Frame-Rate Optical Flow System”

THANK YOU FOR YOUR ATTENTION

QUESTIONS?