Hardware and Software architecture of a bio-inspired vision system for mobile robots

Thomas Lefebvre

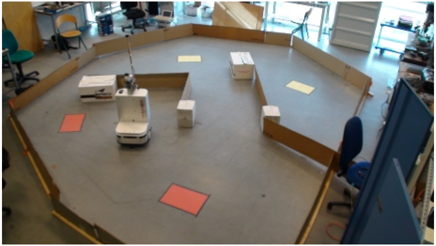
ETIS / ENSEA - Université de Cergy-Pontoise - CNRS UMR 8051
95014 Cergy-Pontoise Cedex, France

5 avril 2012

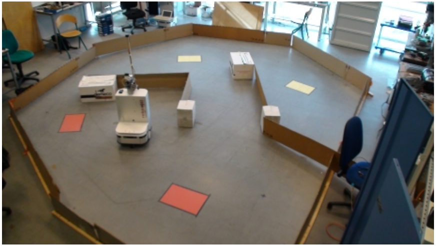
Plan

Summary

Bio-inspired robotics

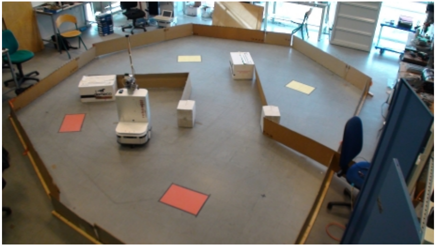


Bio-inspired robotics



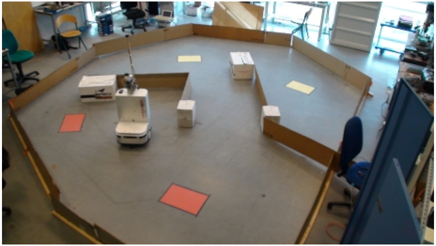
- Autonomous navigation / facial expressions recognition / ...

Bio-inspired robotics



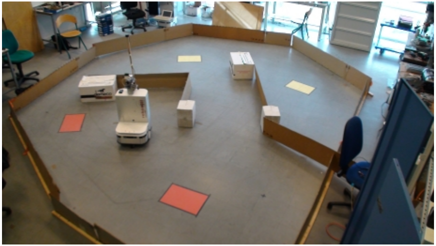
- Autonomous navigation / facial expressions recognition / ...
- Bio-inspired neural-networks

Bio-inspired robotics



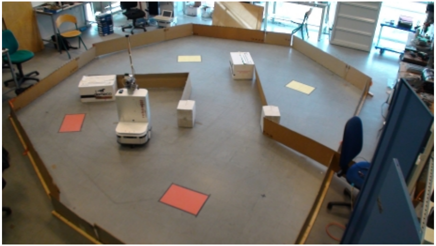
- Autonomous navigation / facial expressions recognition / ...
- Bio-inspired neural-networks
- Camera $\Rightarrow$ Vision system $\Rightarrow$ neural network

Bio-inspired robotics



- Autonomous navigation / facial expressions recognition / ...
- Bio-inspired neural-networks
- Camera ⇒ Vision system ⇒ neural network
- Current vision system : deported on workstations

Bio-inspired robotics



- Autonomous navigation / facial expressions recognition / ...
- Bio-inspired neural-networks
- Camera $\Rightarrow$ Vision system $\Rightarrow$ neural network
- Current vision system : deported on workstations
- Goal : multi-resolution approach - embedded

Goals and constraints

Goals Constraints

Goals and constraints

Goals		Constraints
integration	$\Rightarrow$	Reduced power and volume available

Goals and constraints

Goals		Constraints
integration	$\Rightarrow$	Reduced power and volume available
speed	$\Rightarrow$	25 frames per second

Goals and constraints

Goals		Constraints
integration	$\Rightarrow$	Reduced power and volume available
speed	$\Rightarrow$	25 frames per second
precision	$\Rightarrow$	640×480 pixels camera or more

Goals and constraints

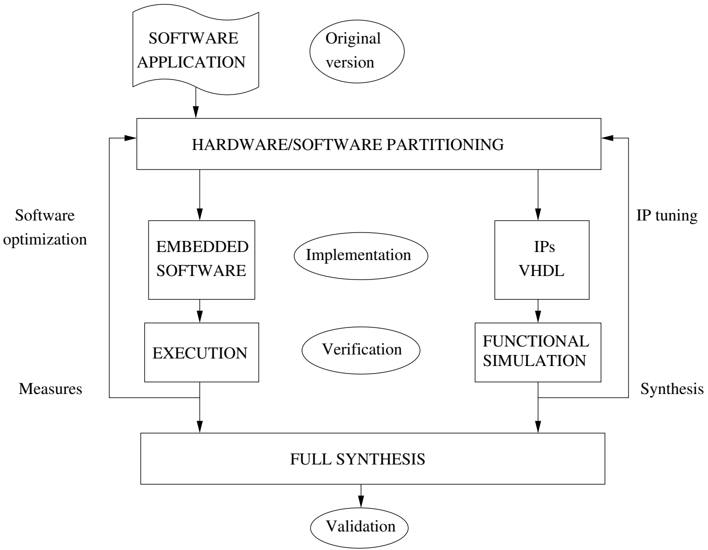
Goals		Constraints
integration	$\Rightarrow$	Reduced power and volume available
speed	$\Rightarrow$	25 frames per second
precision	$\Rightarrow$	640×480 pixels camera or more
flexibility	$\Rightarrow$	Some parameters may vary at runtime

Goals and constraints

Goals		Constraints
integration	$\Rightarrow$	Reduced power and volume available
speed	$\Rightarrow$	25 frames per second
precision	$\Rightarrow$	640×480 pixels camera or more
flexibility	$\Rightarrow$	Some parameters may vary at runtime

Proposed solution : FPGA - embedded processor and hardware IPs

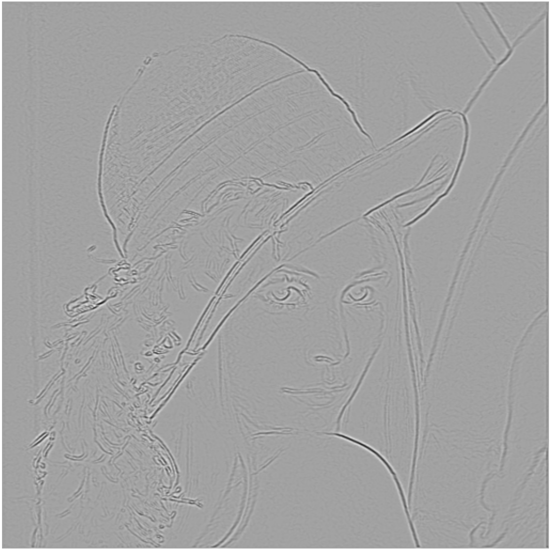
Design flow


SOFTWARE
APPLICATION
Original
version
HARDWARE/SOFTWARE PARTITIONING
Software
optimization
EMBEDDED
SOFTWARE
Implementation
IPs
VHDL
IP tuning
EXECUTION
Verification
FUNCTIONAL
SIMULATION
Measures
Synthesis
FULL SYNTHESIS
Validation

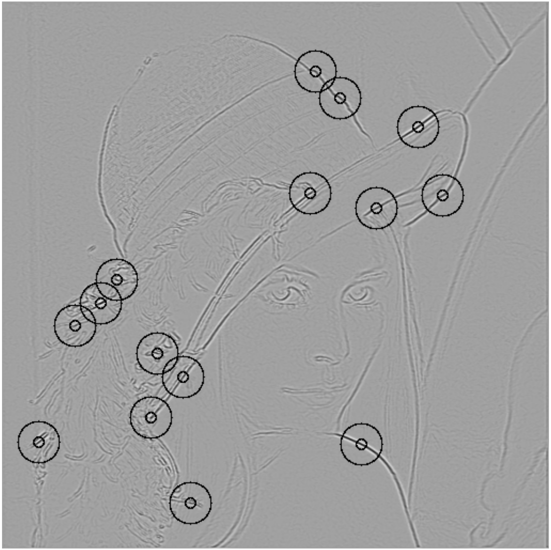
Summary

Gradient, Gaussian filtering, subsampling

Difference of Gaussians (DoG)



Keypoint search



Log-polar transform

Log-polar transform

Log-polar transform

Summary

Temporal behavior

embedded - 1GHz Cortex A8, 256MB RAM
192×144 pixels frames

Temporal behavior

embedded - 1GHz Cortex A8, 256MB RAM
192 $\times$ 144 pixels frames

Function	Total execution time per frame	Percentage of the total
Gradient	11.1 ms	5.4%
Gaussian filtering	145.7 ms	70.7%
Subsampling	1.3 ms	6.3%
DoG	9.1 ms	4.4%
Keypoint search	27.3 ms	13.3%
Neighborhoods	11.5 ms	5.6%
Total	205.9 ms	100%

Temporal behavior

embedded - 1GHz Cortex A8, 256MB RAM
192 × 144 pixels frames

Function	Total execution time per frame	Percentage of the total
Gradient	11.1 ms	5.4%
Gaussian filtering	145.7 ms	70.7%
Subsampling	1.3 ms	6.3%
DoG	9.1 ms	4.4%
Keypoint search	27.3 ms	13.3%
Neighborhoods	11.5 ms	5.6%
Total	205.9 ms	100%

205.9 ms ⇒ 4.85 fps

Temporal behavior

embedded - 1GHz Cortex A8, 256MB RAM
192×144 pixels frames

205.9 ms $\Rightarrow$ 4.85 fps

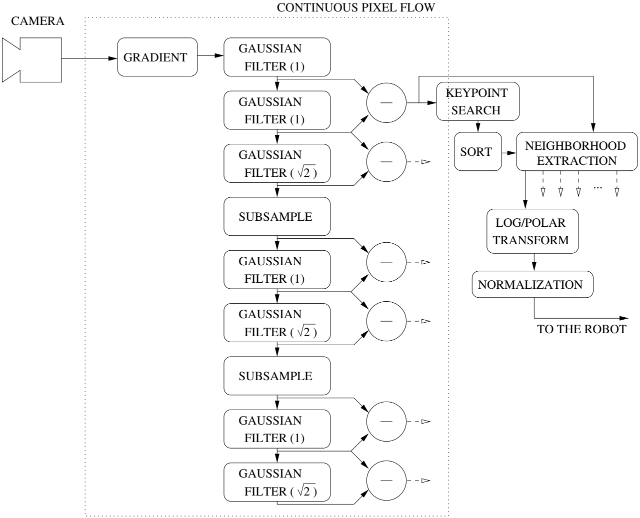
laptop - 1.66GHz Intel Core Duo, 2GB RAM :

39.4 ms $\Rightarrow$ 25.38 fps

What about 640×480 pixels frames ? HD video ?

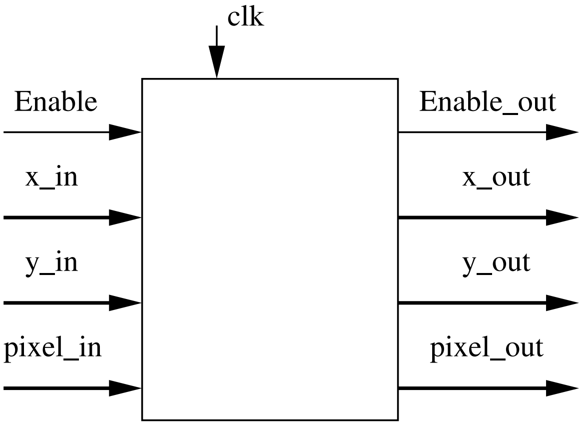
Summary

Pixel flow


CONTINUOUS PIXEL FLOW
CAMERA
GRADIENT
GAUSSIAN FILTER (1)
GAUSSIAN FILTER (1)
GAUSSIAN FILTER (√2)
SUBSAMPLE
GAUSSIAN FILTER (1)
GAUSSIAN FILTER (√2)
SUBSAMPLE
GAUSSIAN FILTER (1)
GAUSSIAN FILTER (√2)
KEYPOINT SEARCH
SORT
NEIGHBORHOOD EXTRACTION
LOG/POLAR TRANSFORM
NORMALIZATION
TO THE ROBOT

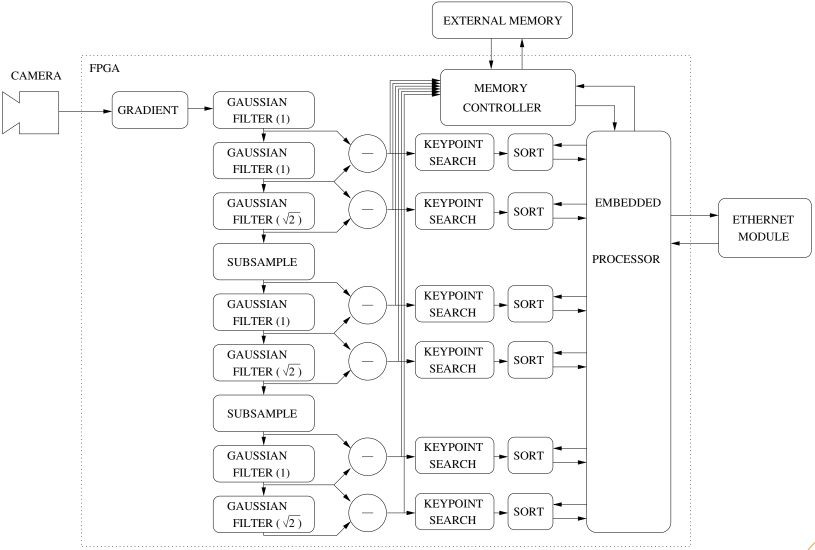
Pixel flow

Modular, generic shell ⇒ greater flexibility



Summary

global system view



hardware acceleration

hardware acceleration

- The IPs produce one grayscale pixel per clock cycle

hardware acceleration

- The IPs produce one grayscale pixel per clock cycle
- 1920×1080 pixels : 2.07 M pixels

hardware acceleration

- The IPs produce one grayscale pixel per clock cycle
- 1920×1080 pixels : 2.07 M pixels
- 25 full-HD fps $\Rightarrow$ 51.84 M pixels per second

hardware acceleration

- The IPs produce one grayscale pixel per clock cycle
- 1920 $\times$ 1080 pixels : 2.07 M pixels
- 25 full-HD fps $\Rightarrow$ 51.84 M pixels per second
- 100 MHz OK for the proposed IPs

hardware acceleration

- The IPs produce one grayscale pixel per clock cycle
- 1920 $\times$ 1080 pixels : 2.07 M pixels
- 25 full-HD fps $\Rightarrow$ 51.84 M pixels per second
- 100 MHz OK for the proposed IPs
- The rest depends on :
  - the embedded processor (Features + Ethernet)
  - the camera interface (RAW ? $\Rightarrow$ grayscale)

Summary

Gaussian filter - 2d convolution

2 solutions for Gaussian filtering :

1. 2d window convolution

Gaussian filter - 2d convolution

2 solutions for Gaussian filtering :

1. 2d window convolution
   large number of multiplications (w_{window}^2)

Gaussian filter - 2d convolution

2 solutions for Gaussian filtering :

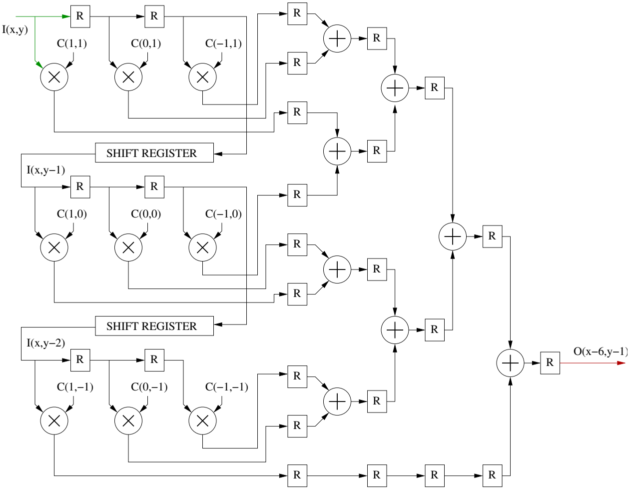
1. 2d window convolution
   large number of multiplications (w_{window}^2)
2. vertical + horizontal 1d windows convolutions

Gaussian filter - 2d convolution

2 solutions for Gaussian filtering :

1. 2d window convolution
   large number of multiplications (w_{window}^2)
2. vertical + horizontal 1d windows convolutions
   fewer multiplications ($2 \times w_{window}$)

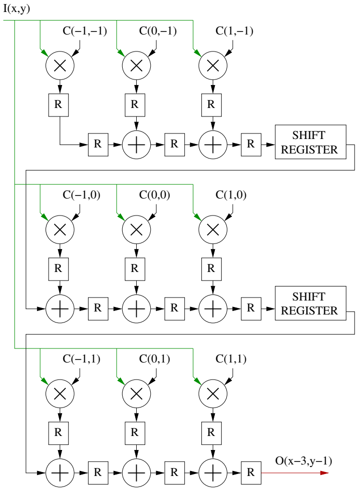
Traditional architecture :



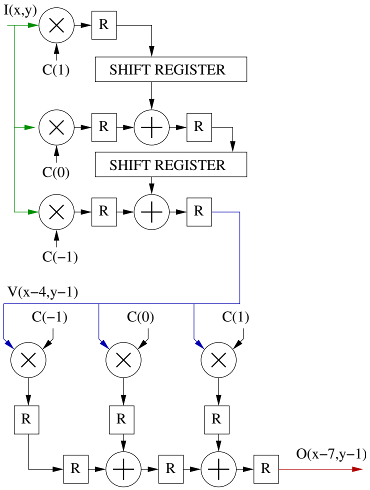
MACC convolution operators :

- Very low latency for the sum of products
- Regular structure, generic code easier to write

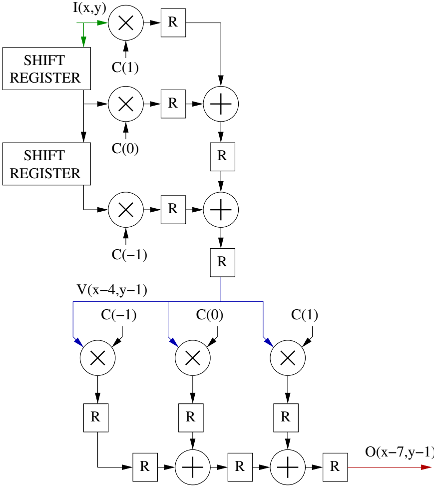
MACC-based architecture :



1D vertical+horizontal :



1D vertical+horizontal :



Summary

Synthesized for Xilinx Virtex 6 family
(equivalent results for Kintex 7 family)

Synthesized for Xilinx Virtex 6 family
(equivalent results for Kintex 7 family)

IP	Registers		LUTs		36k BRAM		DSP48E1	
Gradient	160	<1%	281	<1%	1	<1%	0	
Gaussian filter	5639	8%	5494	7%	21	15%	91	100%
Subsample	288	<1%	259	<1%	2	2%	0	
DoG	192	<1%	624	<1%	6	4%	0	
Keypoint search	58352	81%	63550	77%	106	78%	0	
Sorting	7032	10%	12048	15%	0		0	
Total	71663	100%	82256	100%	136	100%	91	100%

Synthesized for Xilinx Virtex 6 family
(equivalent results for Kintex 7 family)

IP	Registers		LUTs		36k BRAM		DSP48E1	
Gradient	160	<1%	281	<1%	1	<1%	0	
Gaussian filter	5639	8%	5494	7%	21	15%	91	100%
Subsample	288	<1%	259	<1%	2	2%	0	
DoG	192	<1%	624	<1%	6	4%	0	
Keypoint search	58352	81%	63550	77%	106	78%	0	
Sorting	7032	10%	12048	15%	0		0	
Total	71663	100%	82256	100%	136	100%	91	100%

Chip	Registers	LUTs	36k BRAM	DSP48E1
Virtex 6 LX240T FPGA	301440 24%	150720 55%	416 33%	768 12%
Kintex 7 K355T FPGA	445200 16%	222600 37%	715 19%	1440 6%
Zynq Z-7045 (Kintex 7 FPGA)	448000 16%	224000 37%	545 25%	900 10%

Influence of parameters

Gaussian Filter IP - 4 parameters :

- Bus width of pixels and coefficients

Influence of parameters

Gaussian Filter IP - 4 parameters :

- Bus width of pixels and coefficients
- Width of coefficient window

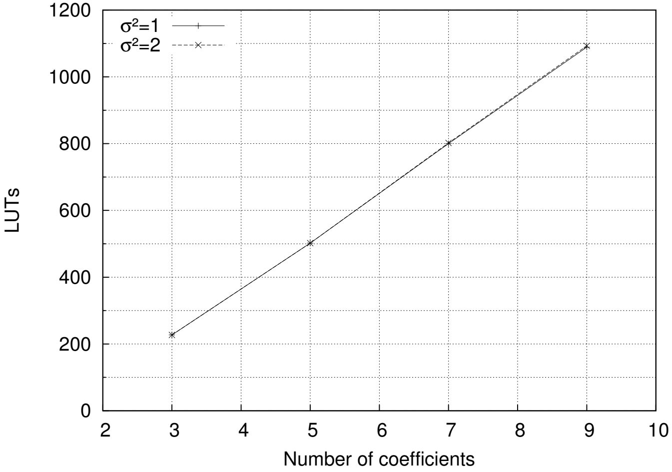
Influence of parameters

Gaussian Filter IP - 4 parameters :

- Bus width of pixels and coefficients
- Width of coefficient window
- σ coefficient

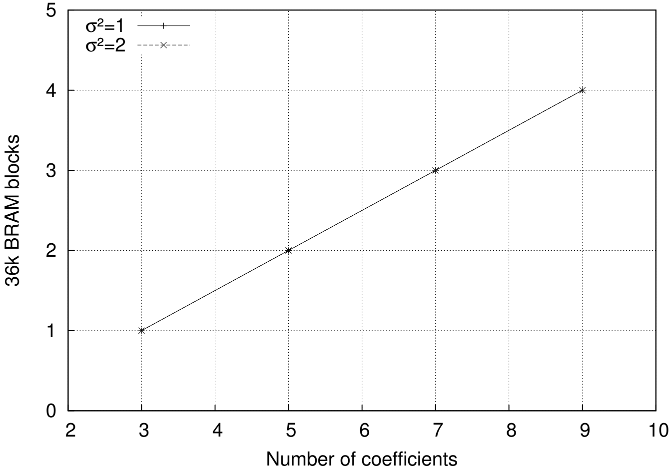
Influence of parameters

number of coefficients per window / σ value



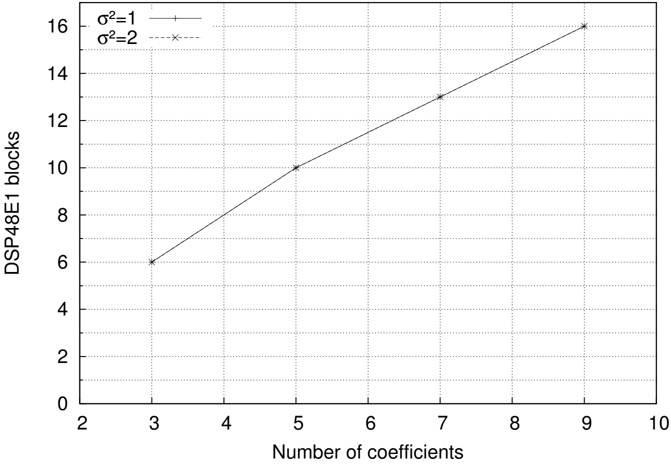
Influence of parameters

number of coefficients per window / σ value



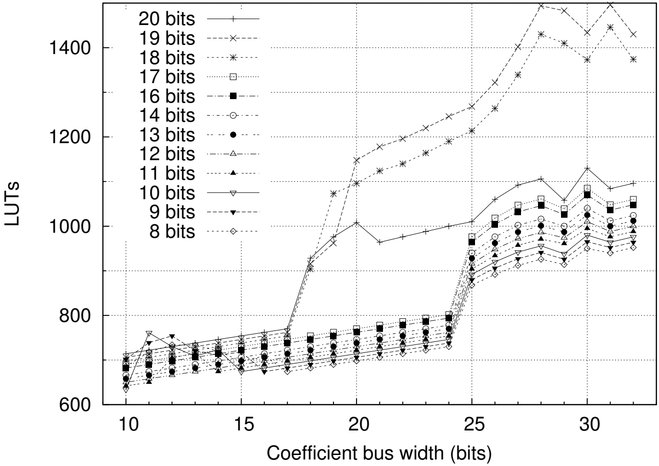
Influence of parameters

number of coefficients per window / σ value

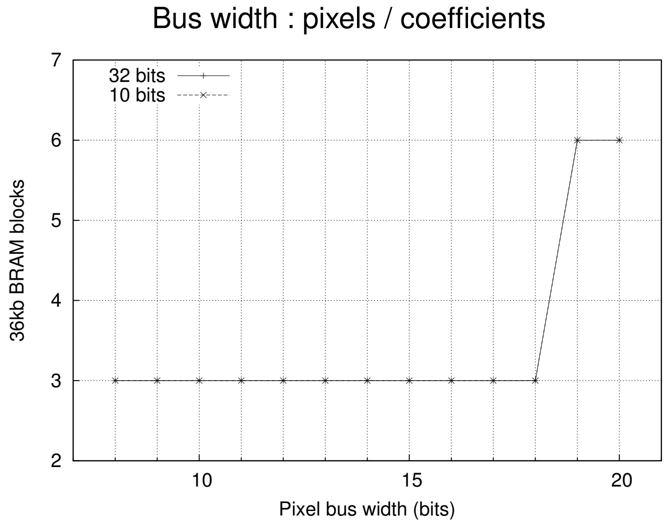


Influence of parameters

Bus width : pixels / coefficients

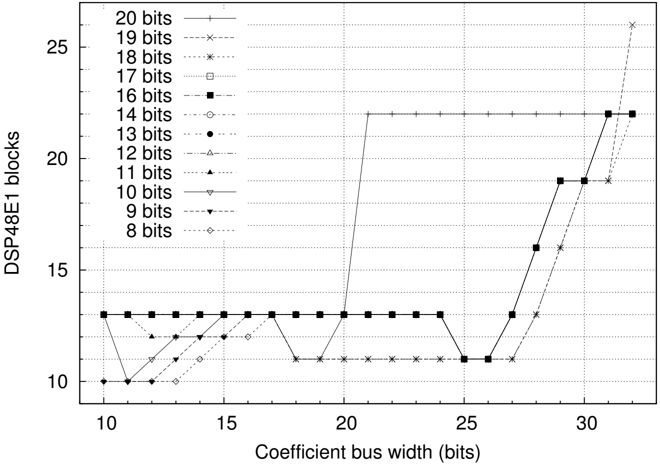


Influence of parameters



Influence of parameters

Bus width : pixels / coefficients



Influence of parameters

Keypoint search - 2 IP parameters :

- Bus width of pixels

Influence of parameters

Keypoint search - 2 IP parameters :

- Bus width of pixels
- Search radius

Influence of parameters

Keypoint search - 2 IP parameters :

- Bus width of pixels
- Search radius

1 pixel within the search radius $\Rightarrow$ 1 comparator

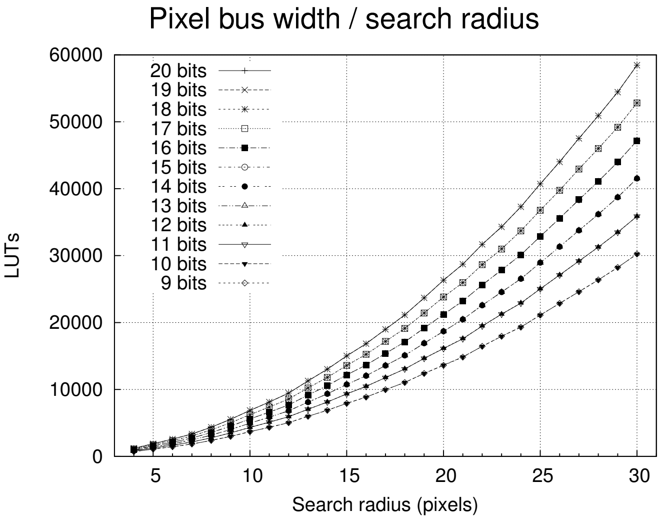
Influence of parameters

Keypoint search - 2 IP parameters :

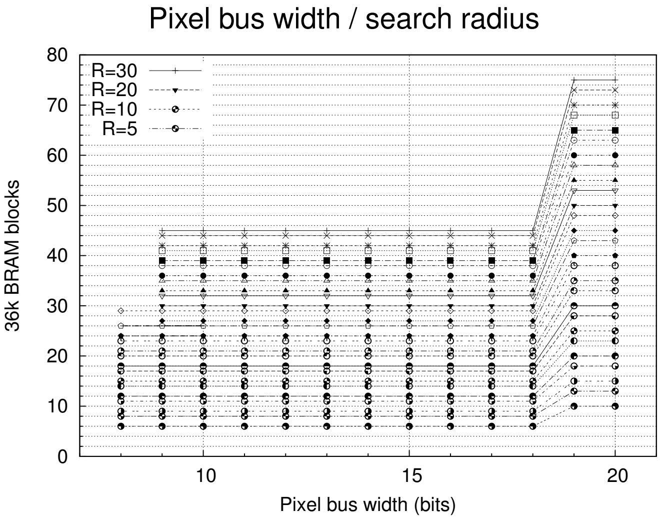
- Bus width of pixels
- Search radius

1 pixel within the search radius $\Rightarrow$ 1 comparator
R=20 $\Rightarrow$ 1256 comparators

Influence of parameters



Influence of parameters



Summary

Embedded platform

Validation of the integration in the robot

Embedded platform

Validation of the integration in the robot
Neural network $\Leftrightarrow$ Vision system (through Ethernet)

Embedded platform

Validation of the integration in the robot

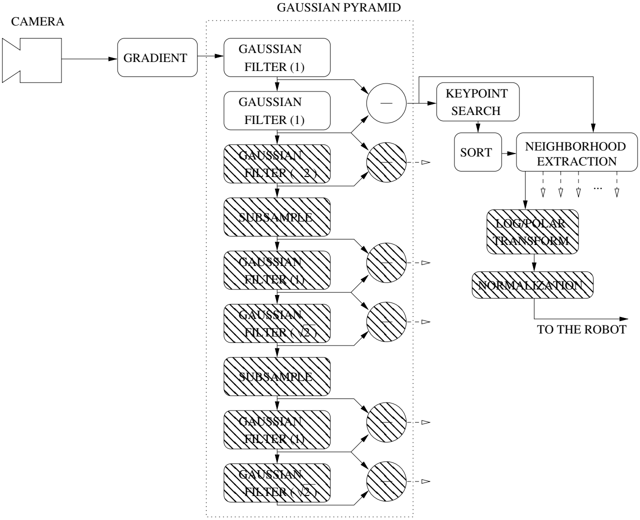
Neural network $\Leftrightarrow$ Vision system (through Ethernet)

Only one spatial frequency band (small FPGA)

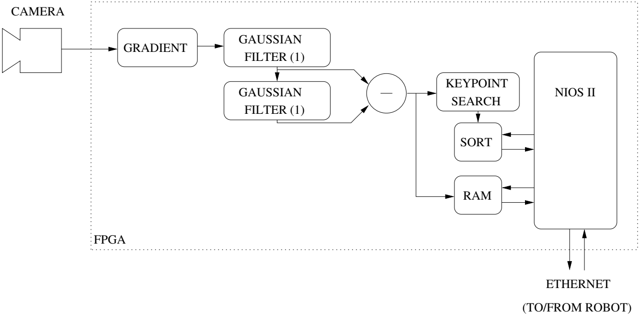
Embedded platform

Validation of the integration in the robot
Neural network $\Leftrightarrow$ Vision system (through Ethernet)
Only one spatial frequency band (small FPGA)
9 fps (320×240 pixels), limited by the NIOS II networking

Embedded platform



Embedded platform



Demonstration platform

Vision system ⇒ screen : VGA through SDRAM (Terasic IPs)

Demonstration platform

Vision system $\Rightarrow$ screen : VGA through SDRAM (Terasic IPs)
Only one spatial frequency band

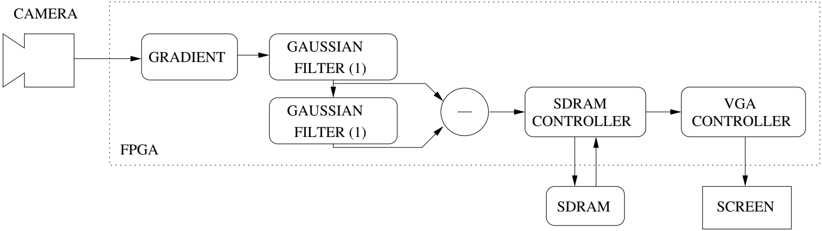
Demonstration platform

Vision system $\Rightarrow$ screen : VGA through SDRAM (Terasic IPs)
Only one spatial frequency band
No keypoint search (no image to display after the DoGs)

Demonstration platform

Vision system $\Rightarrow$ screen : VGA through SDRAM (Terasic IPs)
Only one spatial frequency band
No keypoint search (no image to display after the DoGs)
18 fps (640×480 pixels), limited by the camera module

Demonstration platform



Synthesis results (Embedded PF)

Terasic DE2-115 board - Cyclone IV 4CE115

Images : Grayscale 320 $\times$ 240 pixels
Parameters : pixels on 16 bits, convolution coefficients on 16 bits, R=20, convolution windows : 7 coefficients

Logic Elements : 72k (on 114k - 63%)
FPGA Memory : 1956kb (on 3981kb - 49%)
9-bit multipliers : 4 (on 532 - <1%)

Conclusion

- The proposed architecture matches all the constraints of the system (time, volume, energy)
- Hardware IPs allow for the processing of Full-HD at over 50 fps
- The main IPs have been profiled in regard to their parameters
- IP genericity permits to change these parameters to aim for optimal performance for various FPGAs

Future works

- Implement the full system (multi-resolution, decent frame size) on a camera/FPGA couple
- Validate the behavior of the system and tune the parameters for each use case
- Design and implement log-polar transform as hardware IPs

Thanks for your attention.

Do you have any question ?