

# EE - 451 Motion Detection



May 2024

Vincent Roduit  
Filippo Quadri



OUTLINE

Idea and specifications

Design Process

Final Result

Discussion

# Idea and specifications

- Surveillance Camera

- **Specifications:**

- At least 1 or 2 frames per second
- Should be low power consumption
  - Limited hardware acceleration -> only to meet the constraints!
  - Software optimization (memory access, ...)



O U T L I N E

Idea and Specifications

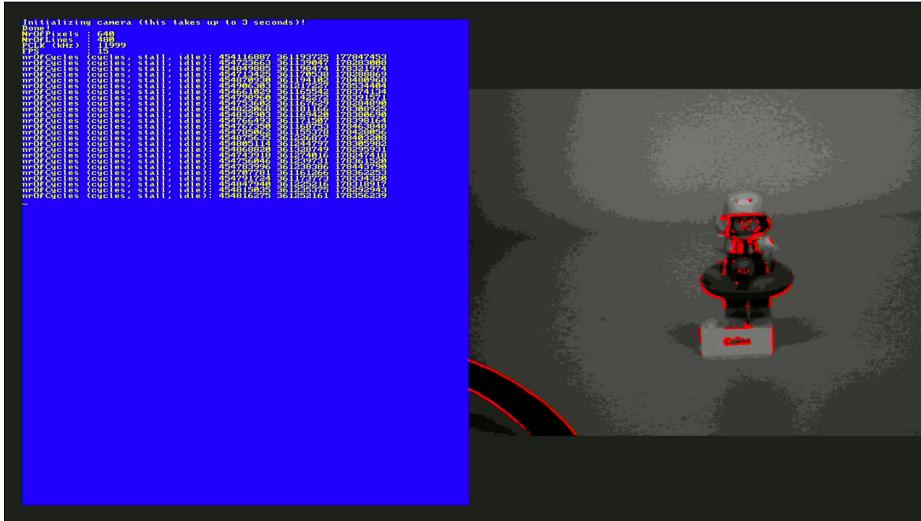
Design Process

Final Result

Discussion

# Design process

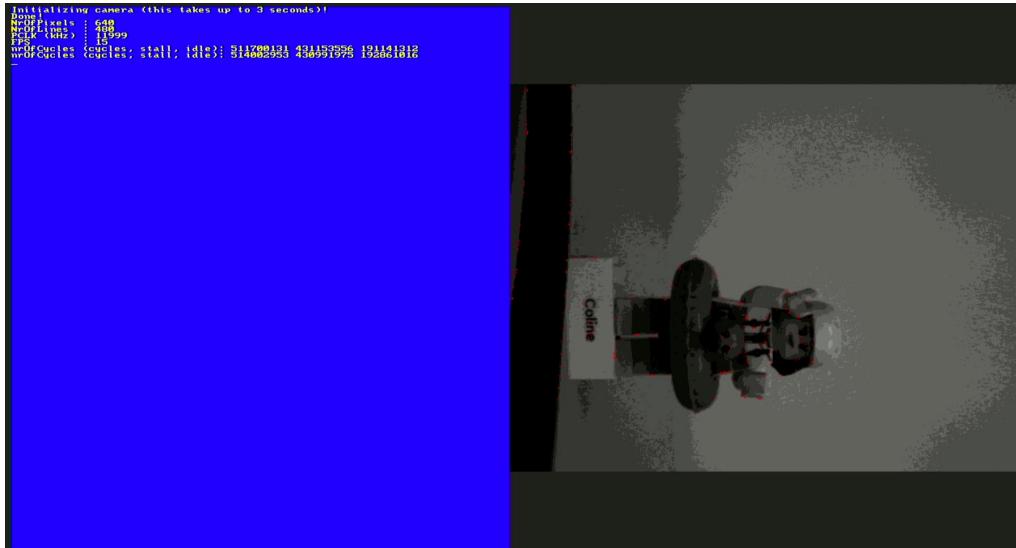
## Step 1: Software implementation



- Bottlenecks:
  - Sobel Computation
  - Bus occupation

# Design process

## Step 2: Custom Instruction for Sobel computation

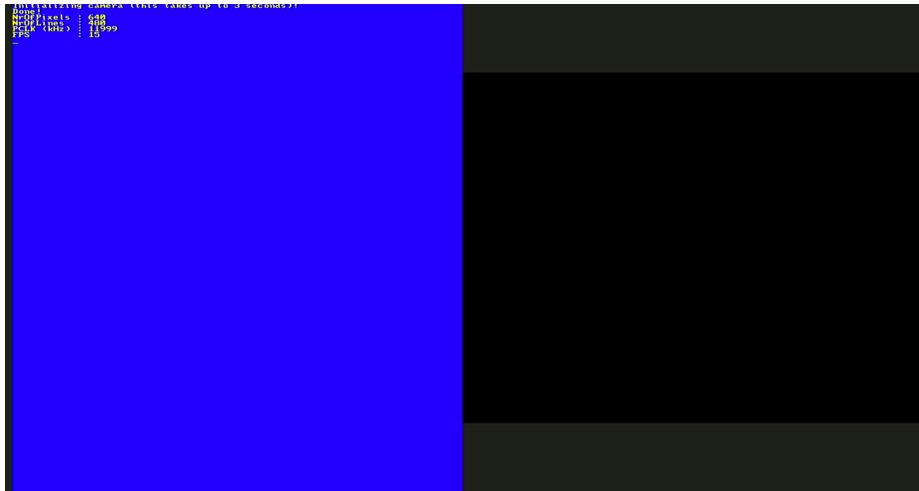


### ■ Bottlenecks

- Bus occupation

# Design process

## Step 3: DMA implementation



- ✓ 3 lines transferred at a time
- ✓ Slightly better performances
- **Bottlenecks:**
  - Same line reads 3 times
  - Same block matrix also reads two times
  - Comparison between two edges done on 8 bits (useless)
  - 1 convolution at a time

# Design process

## Step 4: 4 convolutions at a time

px 0	px 1	px 2	px 3	px 4	px 5	px 6	px 7
px 8	px 9	px 10	px 11	px 12	px 13	px 14	px 15
px 16	px 17	px 18	px 19	px 20	px 21	px 22	px 23

(a) Convolution 1

px 0	px 1	px 2	px 3	px 4	px 5	px 6	px 7
px 8	px 9	px 10	px 11	px 12	px 13	px 14	px 15
px 16	px 17	px 18	px 19	px 20	px 21	px 22	px 23

(b) Convolution 2

px 0	px 1	px 2	px 3	px 4	px 5	px 6	px 7
px 8	px 9	px 10	px 11	px 12	px 13	px 14	px 15
px 16	px 17	px 18	px 19	px 20	px 21	px 22	px 23

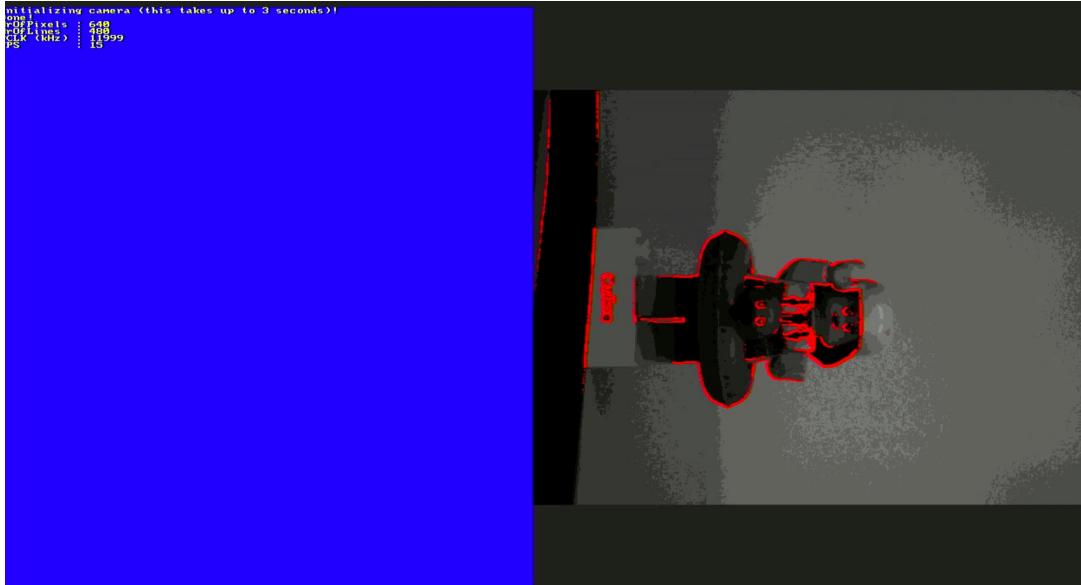
(c) Convolution 3

px 0	px 1	px 2	px 3	px 4	px 5	px 6	px 7
px 8	px 9	px 10	px 11	px 12	px 13	px 14	px 15
px 16	px 17	px 18	px 19	px 20	px 21	px 22	px 23

(d) Convolution 4

# Design process

## Step 4: 4 convolutions at a time



- ✓ 4 convolutions computed at the same time
- ✓ Whole line edges transferred
- ✓ Better performance in terms of speed
- Bottlenecks:
  - Same line reads 3 times
  - Same block matrix also reads two times
  - Comparison between two edges done on 8 bits (useless)

# Design process

## Step 5: Reverse-Forward Reads and Circular buffer

px 4	px 5	px 6	px 7	px 0	px 1	px 2	px 3
px 12	px 13	px 14	px 15	px 8	px 9	px 10	px 11
px 20	px 21	px 22	px 23	px 16	px 17	px 18	px 19

(a) Convolution 1

px 4	px 5	px 6	px 7	px 0	px 1	px 2	px 3
px 12	px 13	px 14	px 15	px 8	px 9	px 10	px 11
px 20	px 21	px 22	px 23	px 16	px 17	px 18	px 19

(b) Convolution 2

px 4	px 5	px 6	px 7	px 0	px 1	px 2	px 3
px 12	px 13	px 14	px 15	px 8	px 9	px 10	px 11
px 20	px 21	px 22	px 23	px 16	px 17	px 18	px 19

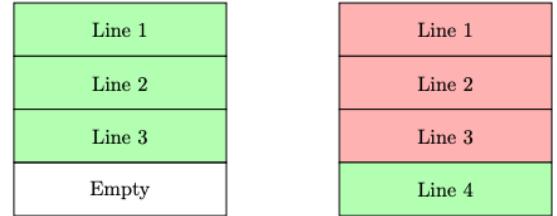
(c) Convolution 3

px 4	px 5	px 6	px 7	px 0	px 1	px 2	px 3
px 12	px 13	px 14	px 15	px 8	px 9	px 10	px 11
px 20	px 21	px 22	px 23	px 16	px 17	px 18	px 19

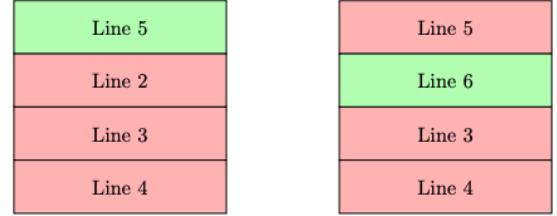
(d) Convolution 4

# Design process

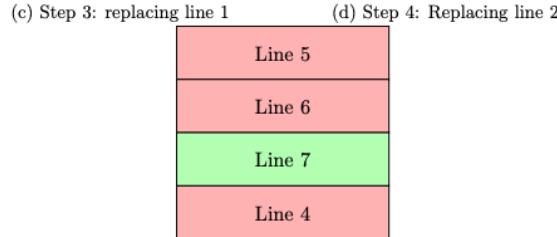
## Step 5: Reverse-Forward Reads and Circular buffer



(a) Step 1: loading the first 3 lines



(b) Step 2: loading Line 4



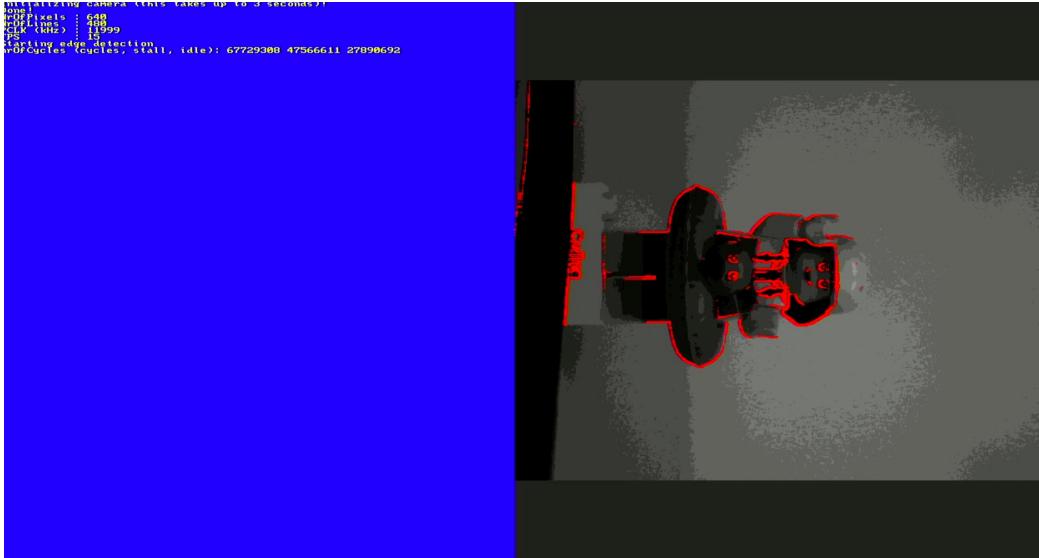
(c) Step 3: replacing line 1

(d) Step 4: Replacing line 2

(e) Step 5: Replacing line 3

# Design process

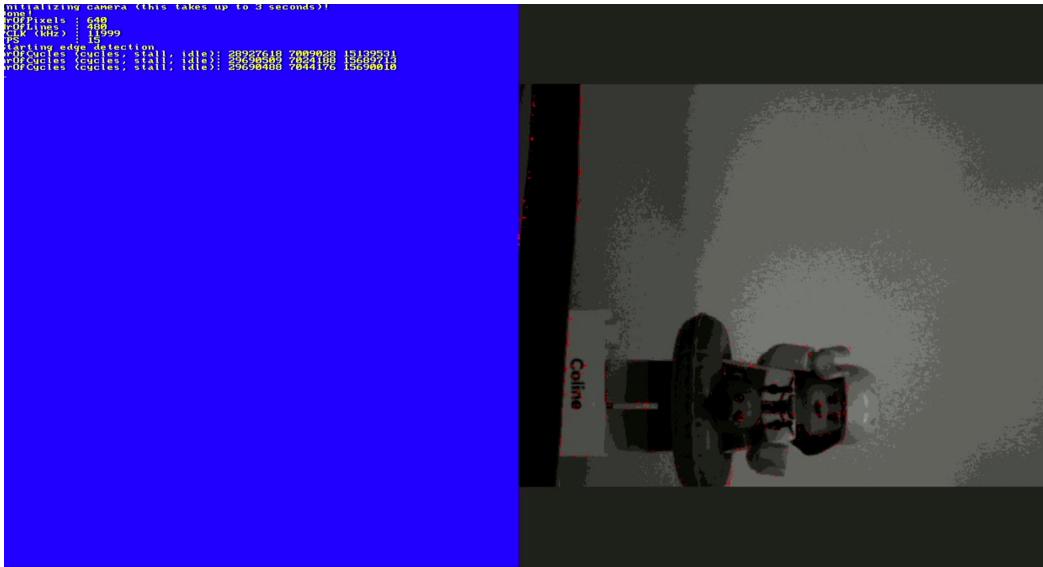
## Step 5: Reverse-Forward Reads and Circular buffer



- ✓ 4 convolutions computed in reverse or forward
- ✓ Efficient reading by avoiding redundant information
- **Bottlenecks:**
  - Comparison on 8-bit
  - Comparison reading not efficient

# Design process

## Step 6: DMA for comparison



✓ Efficient reading for comparison

▪ **Bottlenecks:**

- Comparison on 1-bit but on an inefficient manner (byte enable)

# Design process

## Step 7: Comparison on 1 bit



- ✓ Size reduced
  - ✓ Optimized comparison



O U T L I N E

**Idea and Specifications**

**Design Process**

**Final Result**

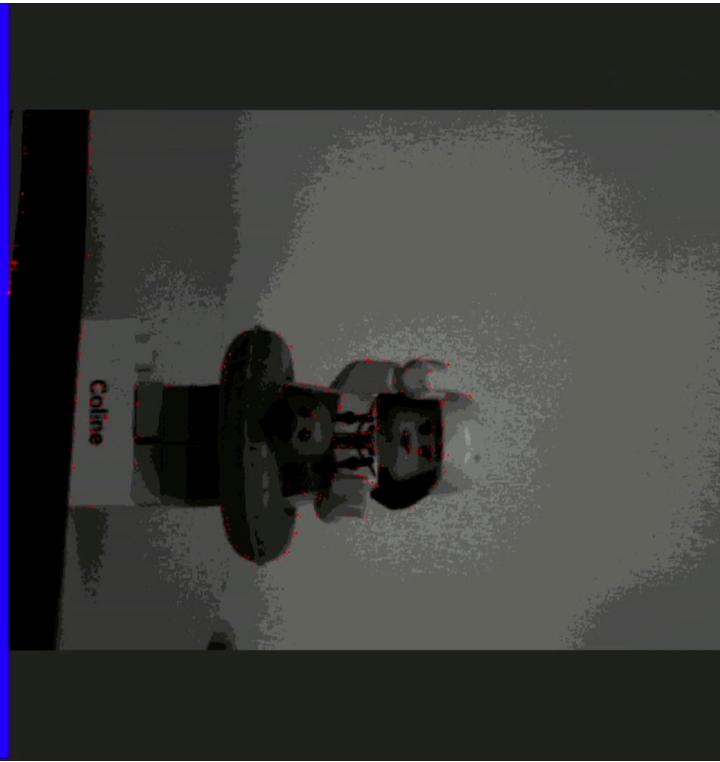
**Discussion**

# Final Result

```
initializing camera (this takes up to 3 seconds)!
```

Rows	640
cols	640
colorlines	11999
clk (kHz)	11999
ps	1
objel_size	9600

```
nrOfcycles (cycles, stall, idle): 23170951 1854368 13711998
```



# Final Result

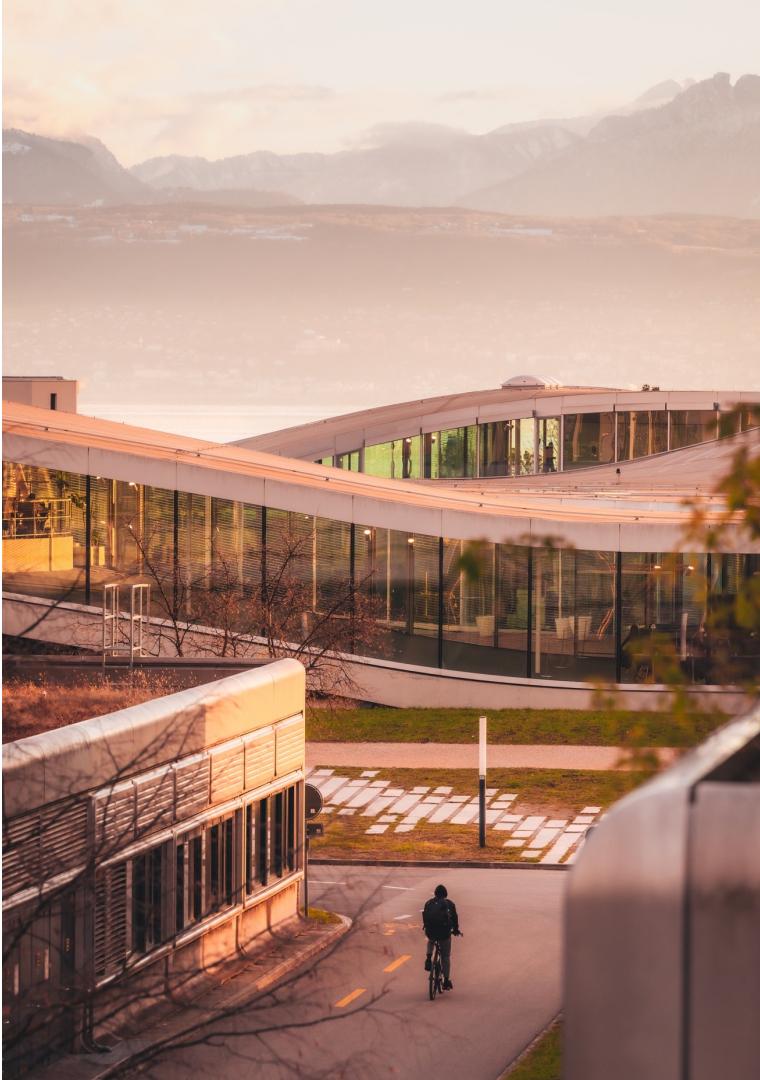
## Methods comparison

Method	Hardware size	Frame per second
Software	46%	~ 0.16 FPS
Sobel as CI	46%	~ 0.14 FPS
DMA Alone	41%	~ 0.2 FPS
4 Conv	41%	~ 1.02 FPS
Reverse-Forward Reads + Circular Buffer	39%	~ 1.02 FPS
DMA for Comp	43%	~ 2.51 FPS
1-bit Comp	40%	~ 3 FPS

# Final Result

## Cycles comparison

Method	CPU Cycles	Stall Cycles	Bus idle	SPF	FPS
Software	$4.54 \times 10^8$	$3.6 \times 10^8$	$1.78 \times 10^8$	$6.11s$	$0.16fps$
Sobel CI	$5.14 \times 10^8$	$4.30 \times 10^8$	$1.93 \times 10^8$	$6.92s$	$0.14fps$
DMA alone	$2.32 \times 10^8$	$1.92 \times 10^8$	$8.80 \times 10^7$	$3.12s$	$0.32fps$
4 Conv	$6.88 \times 10^7$	$4.77 \times 10^7$	$2.83 \times 10^7$	$0.93s$	$1.08fps$
F/R Reads & CB	$6.88 \times 10^7$	$4.74 \times 10^7$	$2.86 \times 10^7$	$0.93s$	$1.08fps$
DMA for Comparison	$2.96 \times 10^7$	$7.04 \times 10^6$	$1.55 \times 10^7$	$0.40s$	$2.51fps$
1-bit Comparison	$2.47 \times 10^7$	$1.90 \times 10^6$	$1.48 \times 10^7$	$0.3s$	$3fps$



OUTLINE

Idea and Specifications

Design Process

Final Result

Discussion

- Project fulfills the objective
  - Time constraint
  - Hardware size
- **Further works :**
  - Parallel convolutions
  - Image hashing
  - LoG



Vincent Roduit  
Filippo Quadri