Low-level Vision Processing Architectures

Speaker: Sato, Dang

Supporter: Moriai, Yatsu and Miyamoto

Adaptive Systems Lab

The University of Aizu

COSCO V

Computer System Colloquium

Overview

- Introduction
  - What is Vision Processing?
- Low level Algorithms
- Classifications of Computing
  - SISD, SIMD, MISD, MIMD
- Architectures for Low Level Algorithm
- Conclusion

Overview

- Introduction
  - What is Vision Processing?
- Low level Algorithms
- Classifications of Computing
  - SISD, SIMD, MISD, MIMD
- Architectures for Low Level Algorithm
- Conclusion

What is Vision Processing?

- Make computers understand images and video:
  - Images to Models

What kind of scene?

Where are the cars?

How far is the building?

…

"Vision is the act of knowing what is where by looking." – Aristotle

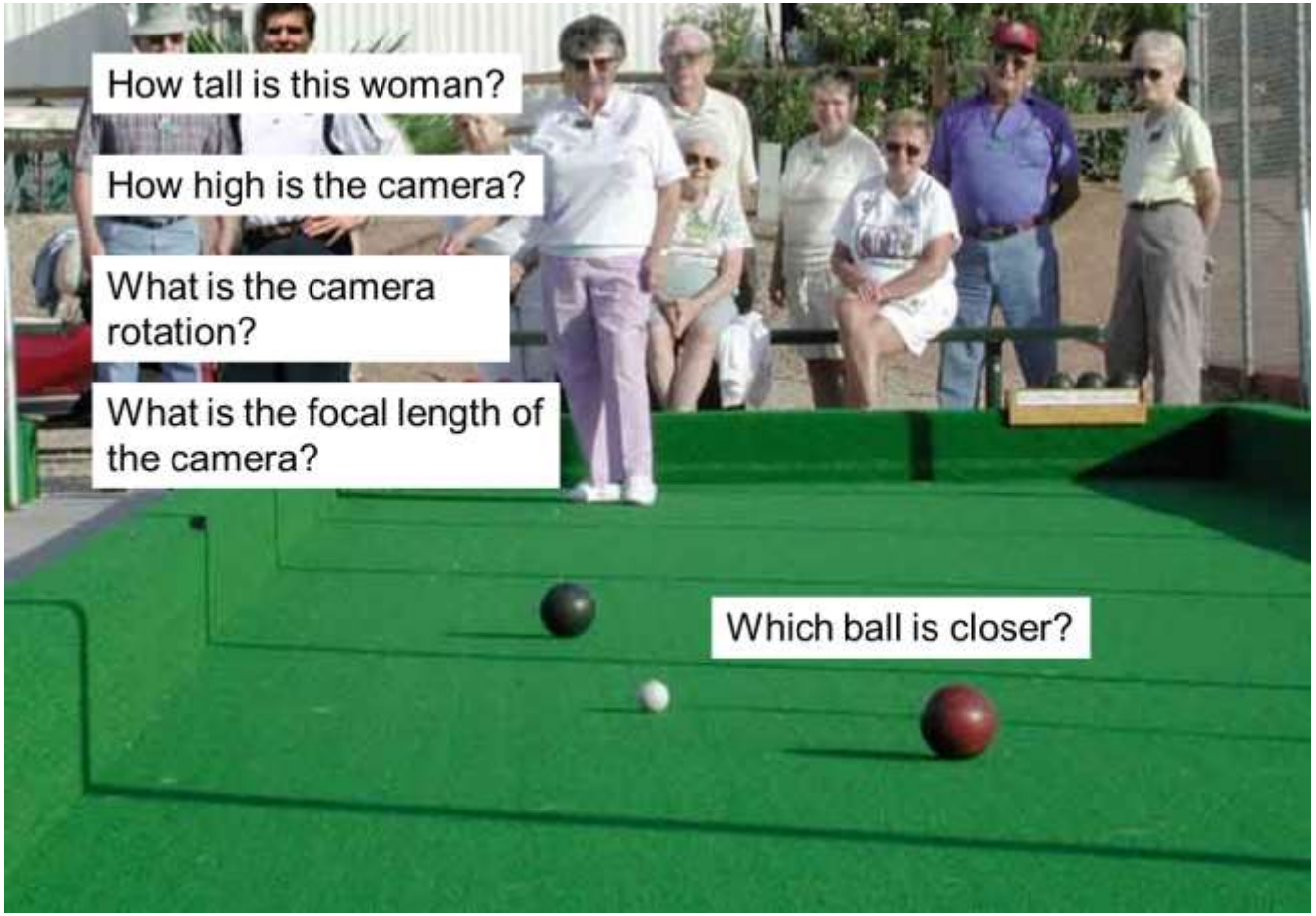
Slide credit James Hays

Application of Vision Processing

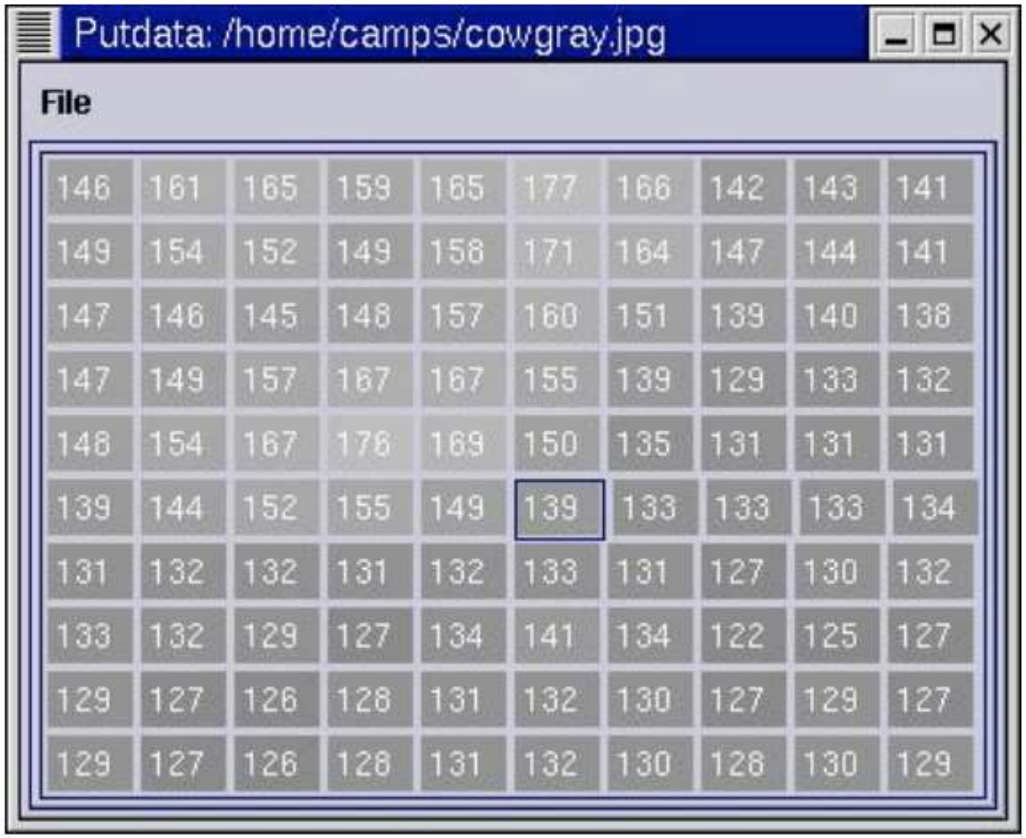
- Optical Character Recognition
- Face detection
- Object recognition
- Shape/motion capture
- Medical Imaging

Why we need?

- Human limitations
  - measurement accuracy, darkness, etc…



Input of Vision Processing



Shoulder of a cow…

HOW????

Credit to Robert Collins

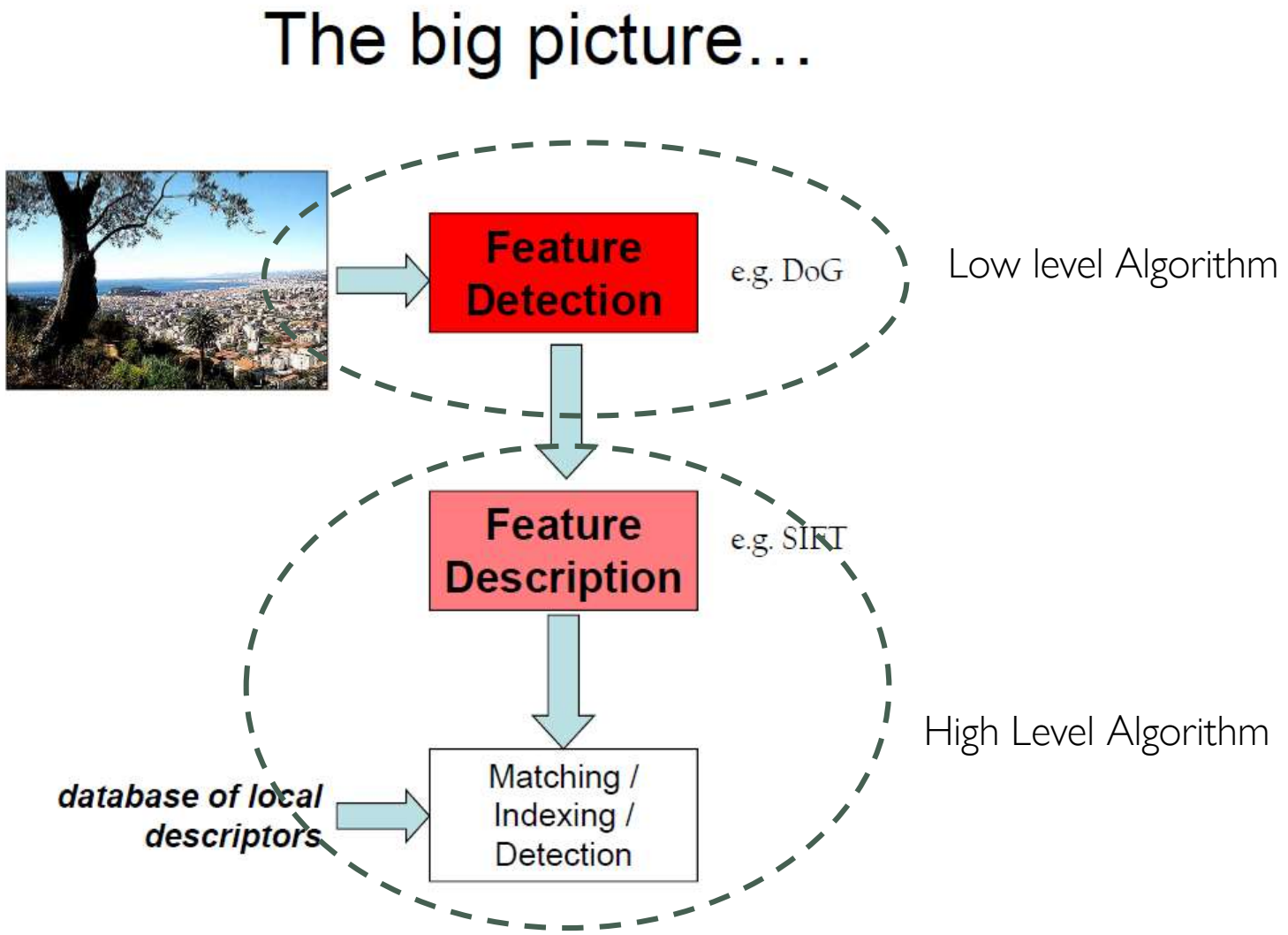
What we need

	Vision-based	IMU-based stabilization	Local contrast enhancement	Multispectral image fusion	Image sensor pipeline*	Target/people tracking	Mosaicking / stitching	Feature / landmark matching	Stereo processing
Commercial UAVs	●	●	●		●	●	●	●	●
Surveillance systems	●	●	●	●	●	●	●		●
Smartphone co-processing / acceleration	●	●	●		●	●	●	●	●
Unmanned aerial reconnaissance systems	●	●	●	●	●	●	●	●	●
Wearable day / night vision systems			●	●	●			●	●
Handheld imaging devices	●	●	●	●	●	●		●	
Vehicle day / night vision systems	●	●	●	●	●	●		●	
Smart cameras	●	●	●	●	●	●	●		●
Medical imaging	●		●	●	●		●		●

*Image warping, histogramming, noise reduction, non-uniformity correction, dead pixel replacement, Bayer filtering, and color correction

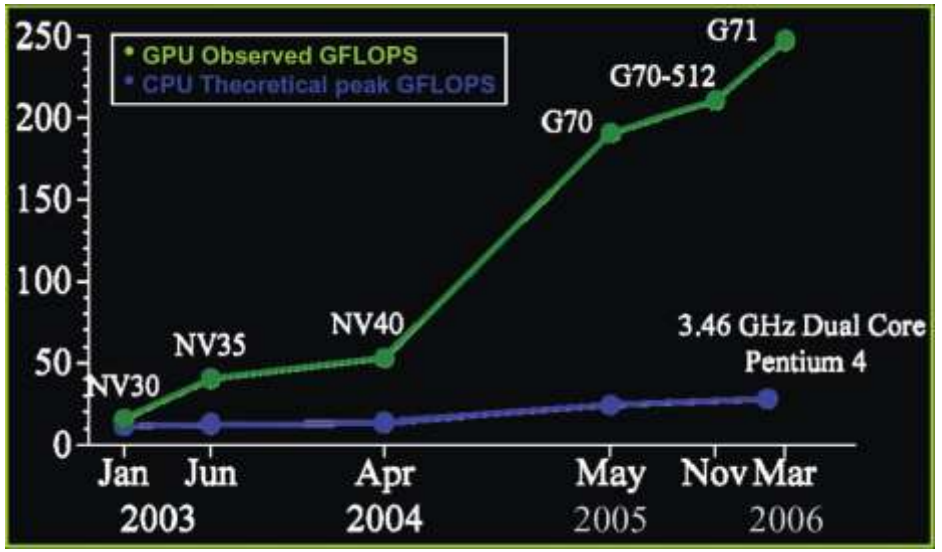
Credit www.sri.com

How we do?


The big picture…
Feature Detection
e.g. DoG
Low level Algorithm
Feature Description
e.g. SIFT
High Level Algorithm
database of local descriptors
Matching / Indexing / Detection

Problems

- Vision Processing requires huge amount of data processing capacity:
  - Better bandwidth
  - Better processing speed
- In order to accelerate the computing speed:
  - Optimize the algorithm.
  - Increase the memory bandwidth, ALU speed.
  - Dedicate hardware accelerator:
    - Ex: GPU



In this presentation

- Re-introduces several low-level algorithms.
- Introduces Classifications of Computing:
  - SISD
  - SIMD
  - MISD
  - MIMD
- Presents several architectures of low-level vision processing:
  - Application Specific Architectures
  - Re-configurable Vision Processing Architecture.

Overview

- Introduction
  - What is Vision Processing?
- **Low level Algorithms**
- Classifications of Computing
  - SISD, SIMD, MISD, MIMD
- Architectures for Low Level Algorithm
- Conclusion

Low Level Vision Processing

- Example of low level algorithms:
  - Filtering.
  - Down-sampling.
  - Border Detection
  - Edge Detection
- This processing can help the system gets basic outputs for further processing.

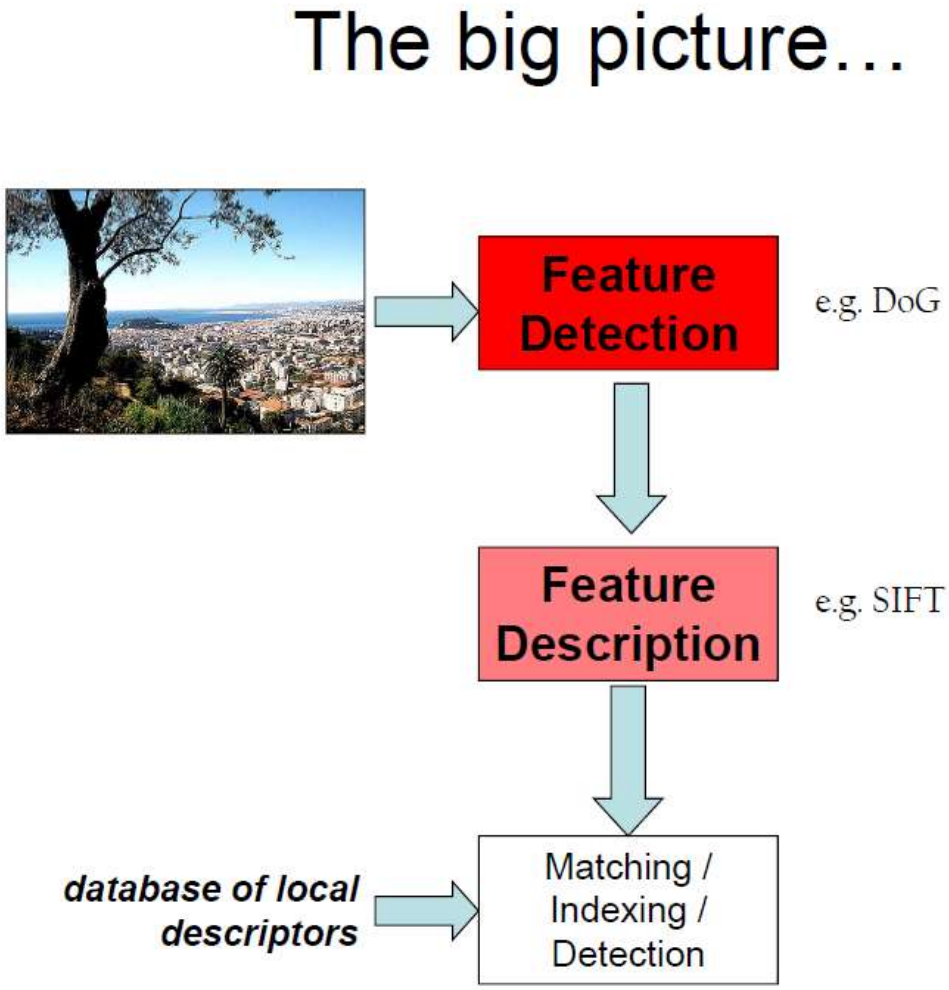


Image Filtering

- Compute function of local neighborhood at each position.
- Why:
  - Enhance images:
    - Denoise, resize, increase contrast, so on.
  - Extract information from images
    - Texture, edges, distinctive points, so on.
  - Detect patterns:
    - Template matching

Image Filtering

$g[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

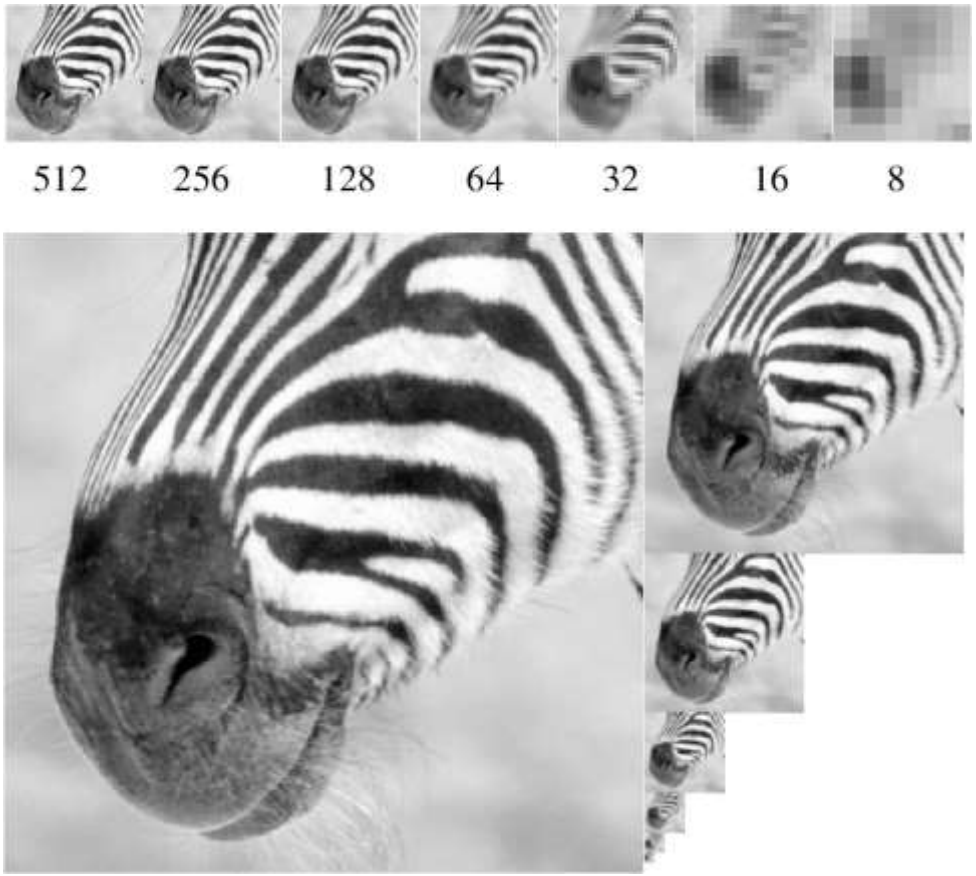
$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

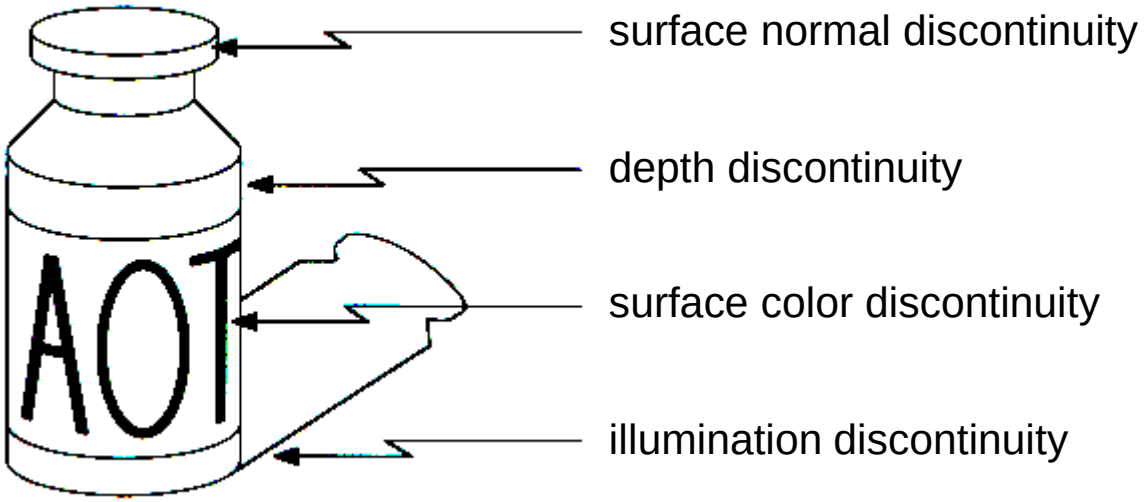
Down-sampling



Source: Forsyth

Edge/Border Detection

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
- **Ideal:** artist's line drawing (but artist is also using object-level knowledge)

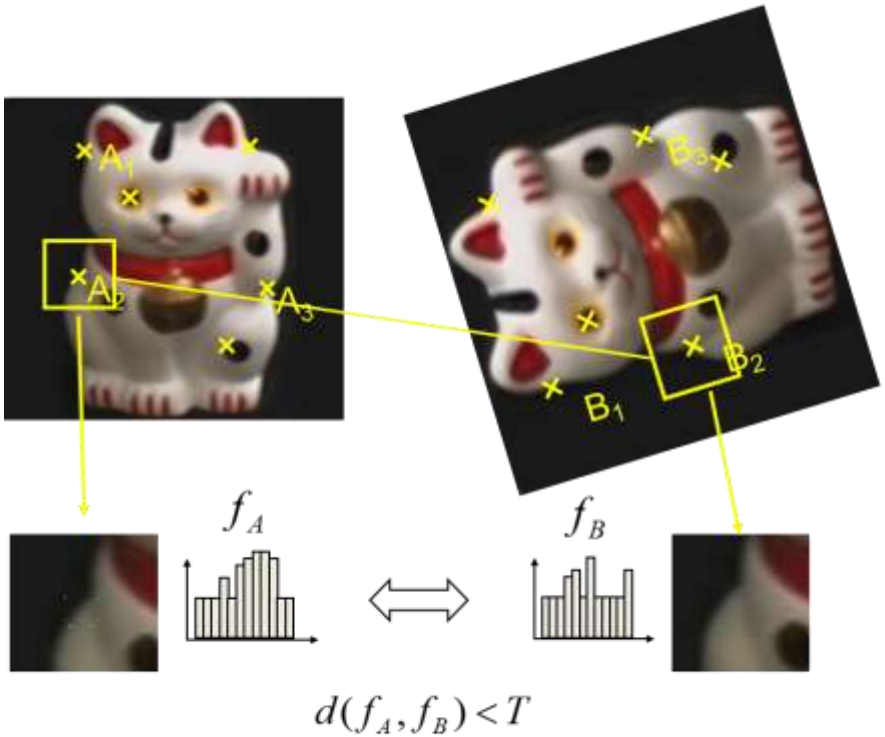


- Edges are caused by a variety of factors

Source: Steve Seitz

Conner Detection

- For matching, interesting points are most concerned.
- Most of interesting points are conner:

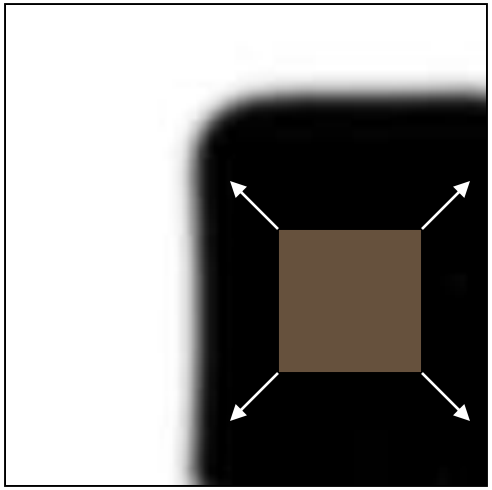


1. **Find a set of distinctive key-points**
2. **Define a region around each keypoint**
3. **Extract and normalize the region content**
4. **Compute a local descriptor from the normalized region**
5. **Match local descriptors**

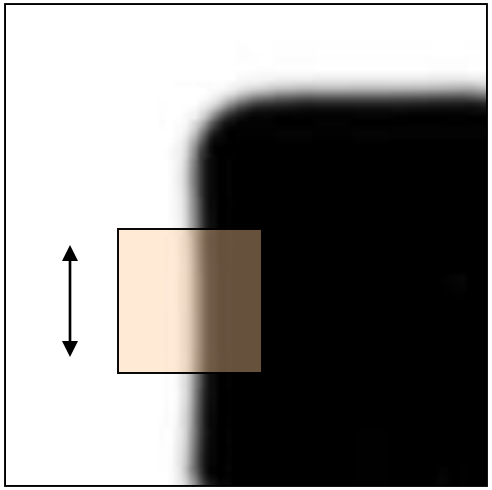
K. Grauman, B. Leibe

Corner Detection: Basic Idea

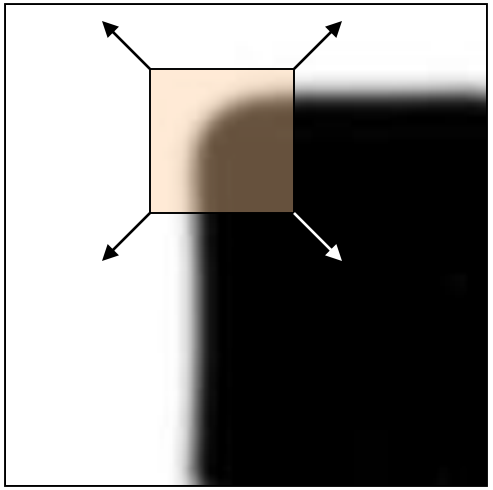
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



"flat" region:
no change in all directions



"edge":
no change along the edge direction



"corner":
significant change in all directions

Source: A. Efros

Overview

- Introduction
  - What is Vision Processing?
- Low level Algorithms
- Classifications of Computing
  - SISD, SIMD, MISD, MIMD
- Architectures for Low Level Algorithm
- Conclusion

Classes of computing

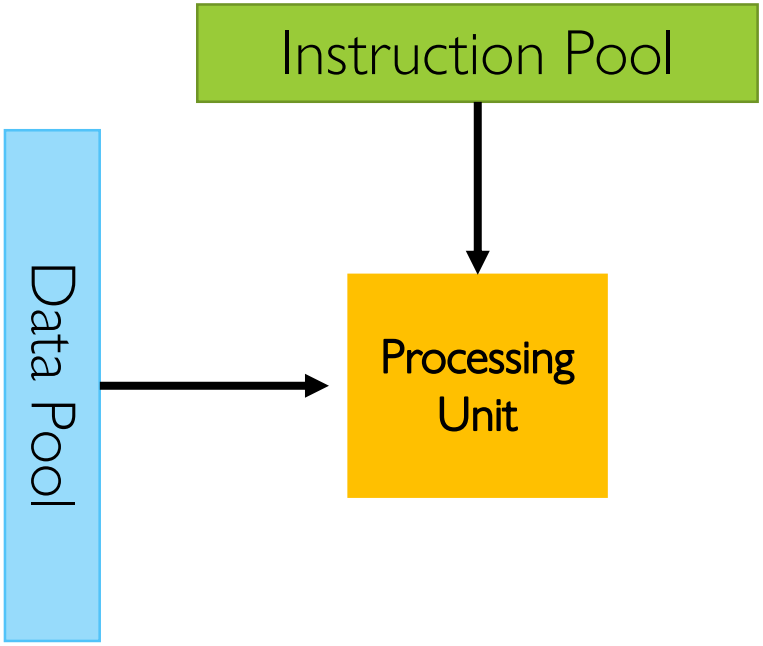
Computation Consists of :

- Sequential Instructions (operation)
- Sequential dataset

We can then abstractly classify into following classes of computing system base on their characteristic instructions and dataset:

- SISD Single Instruction, Single data
- SIMD Single Instruction, Multiple data
- MISD Multiple Instructions, Single data
- MIMD Multiple Instructions, Multiple data

SISD



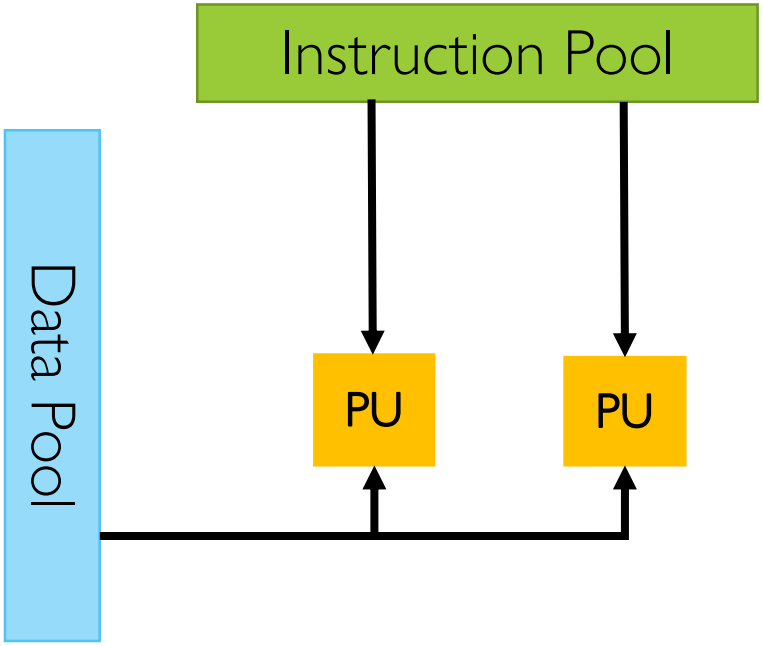
- Single Inistructin Single Data:

At each instant there is only one piece of data being acted upon by one instruction.

MISD

- Multiple Instructions Single Data
  - Multiple streams of instruction
  - Single stream of data
    - Multiple functionally unit operate on single data
    - Possible list of instructions or a complex instruction per operand (CISC)
  - Receive less attention compare to the other

MISD



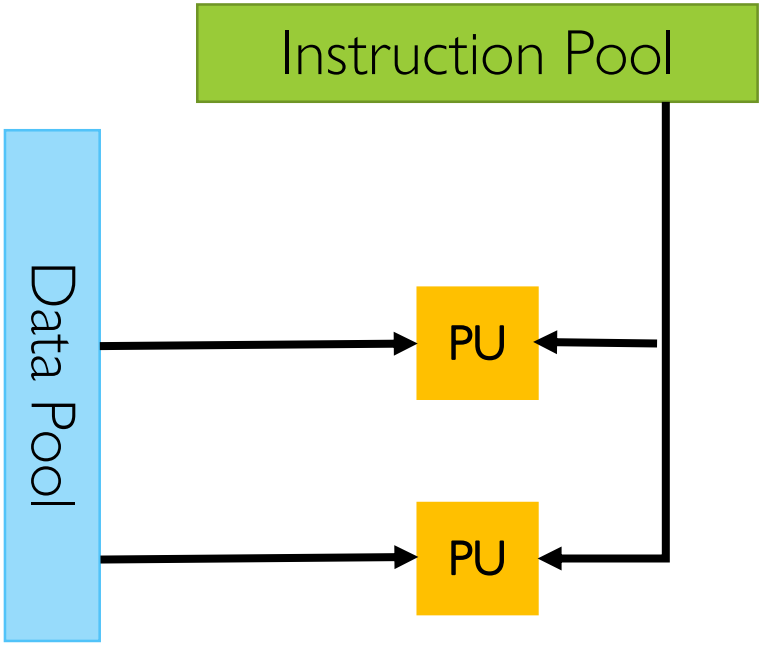
Multiple Instruction Single Data

For each data item more than one instruction is executed at once.

SIMD

- Single Instruction, Multiple Data
  - Single Instruction stream
  - Multiple data streams
- Each instruction operate on multiple data in parallel
- Fine grained Level of Parallelism
- A wide variety of applications can be solved by parallel algorithms with SIMD
  - only problems that can be divided into sub problems, all of those can be solved simultaneously by the same set of instructions
  - This algorithms are typical easy to implement

SIMD



- Single Instruction Multiple Data

One instruction is issued at a time, and each instruction acts on a number of different data items. GPUs work like this.

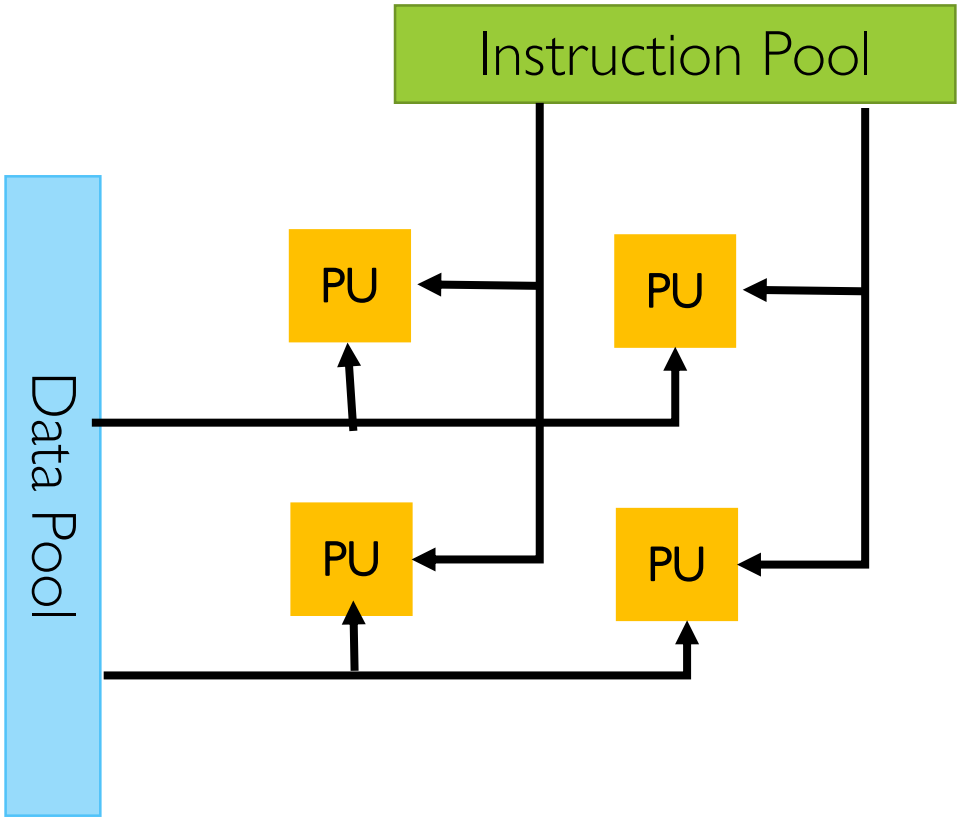
MIMD

- Multiple Instruction Multiple Data
  - Multiple streams of instructions
  - Multiple streams of data
- Middle grained Parallelism level
- Used to solve problem in parallel are those problems that lack the regular structure required by the SIMD model.
- Each execution unit operate asynchronously on their own set of instructions and data, those could be a sub-problems of a single problem.

MIMD

- Requires
  - Synchronization
  - Inter-process communications
  - Parallel algorithms
  - Those algorithms are difficult to design, analyze and implement

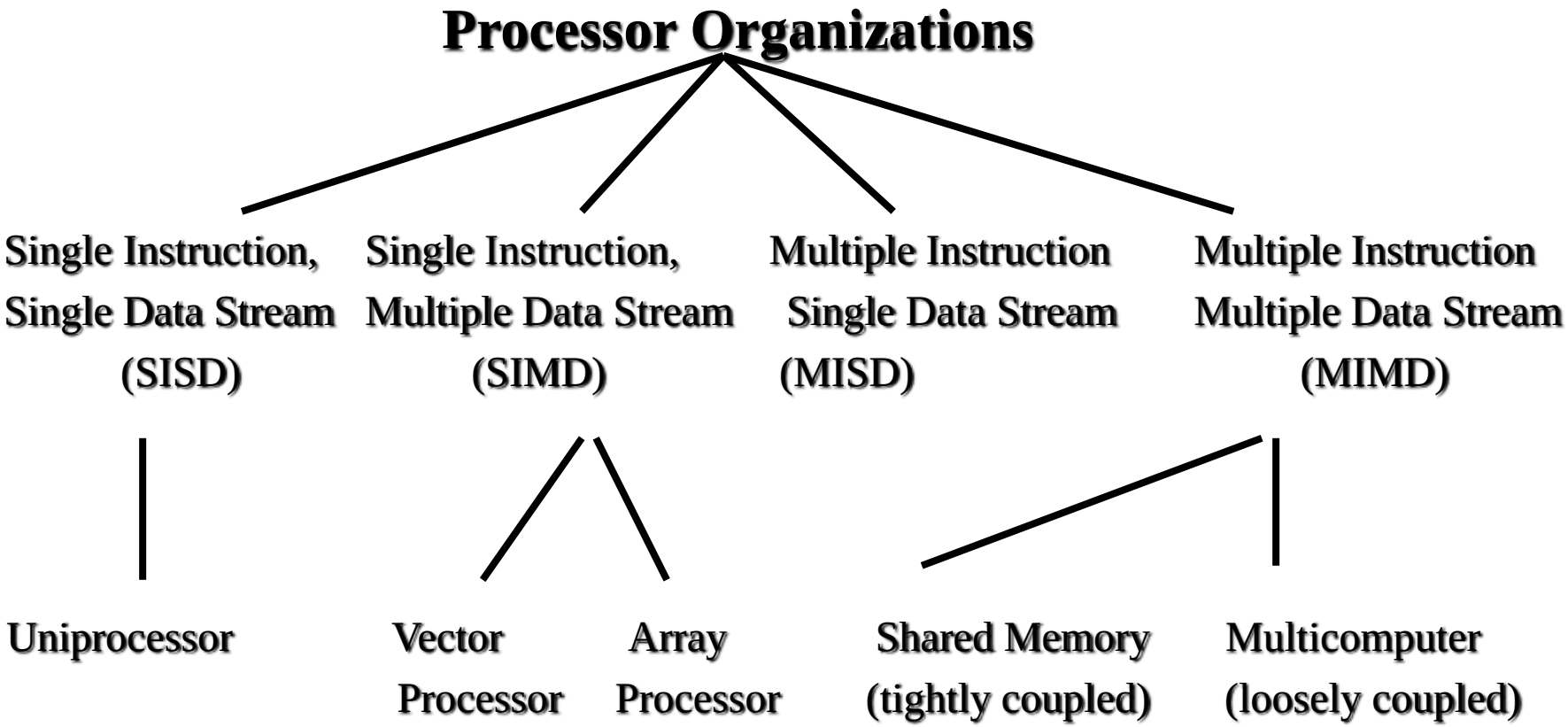
MIMD



Multiple Instruction Multiple Data

Each instruction acts on multiple data items, and many instructions are issued at once.

Computer Architecture Classifications



Overview

- Introduction
  - What is Vision Processing?
- Low level Algorithms
- Classifications of Computing
  - SISD, SIMD, MISD, MIMD
- **Architectures for Low Level Algorithm**
- Conclusion

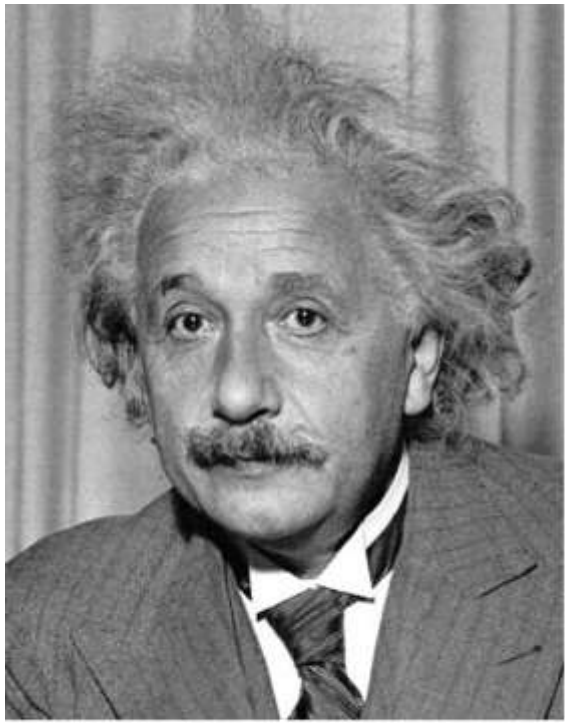
Architectures for Low Level Alg.

- For low-level vision processing, there are four types of architecture:
  - Individual function based architecture.
    - Design specifically for a function, i.e. filter, …
    - Simple instruction set.
  - Re-configurable function based architecture.
    - Design for simple functions.
    - Can be reconfigured to new function.
  - Sensor integrated architecture
    - Instead of loading data from memory, the pixels will be fetched direct to processor.
    - Provide faster speed of processing.
  - Programmable architecture.
    - Consists of multiple Processing Elements (PEs).
    - Each PE has ability to compute a set of instruction (processor based-design).

INDIVIDUAL FUNCTION BASED ARCHITECTURE

SIMD Arch using Sobel Operation

- Sobel Operation:
  - Based on gradient of each pixel to detect the direction of intensity changing.



1	0	-1
2	0	-2
1	0	-1

Vertical Edge
(absolute value)

Rosas, Roberto López, Adriano De Luca, and Francisco Barbosa Santillan. "SIMD architecture for image segmentation using sobel operators implemented in FPGA technology." *Electrical and Electronics Engineering, 2005 2nd International Conference on*. IEEE, 2005.

Operations

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figure 2. Sobel masks.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_5
Z_7	Z_8	Z_9

Figure 3. Image section.

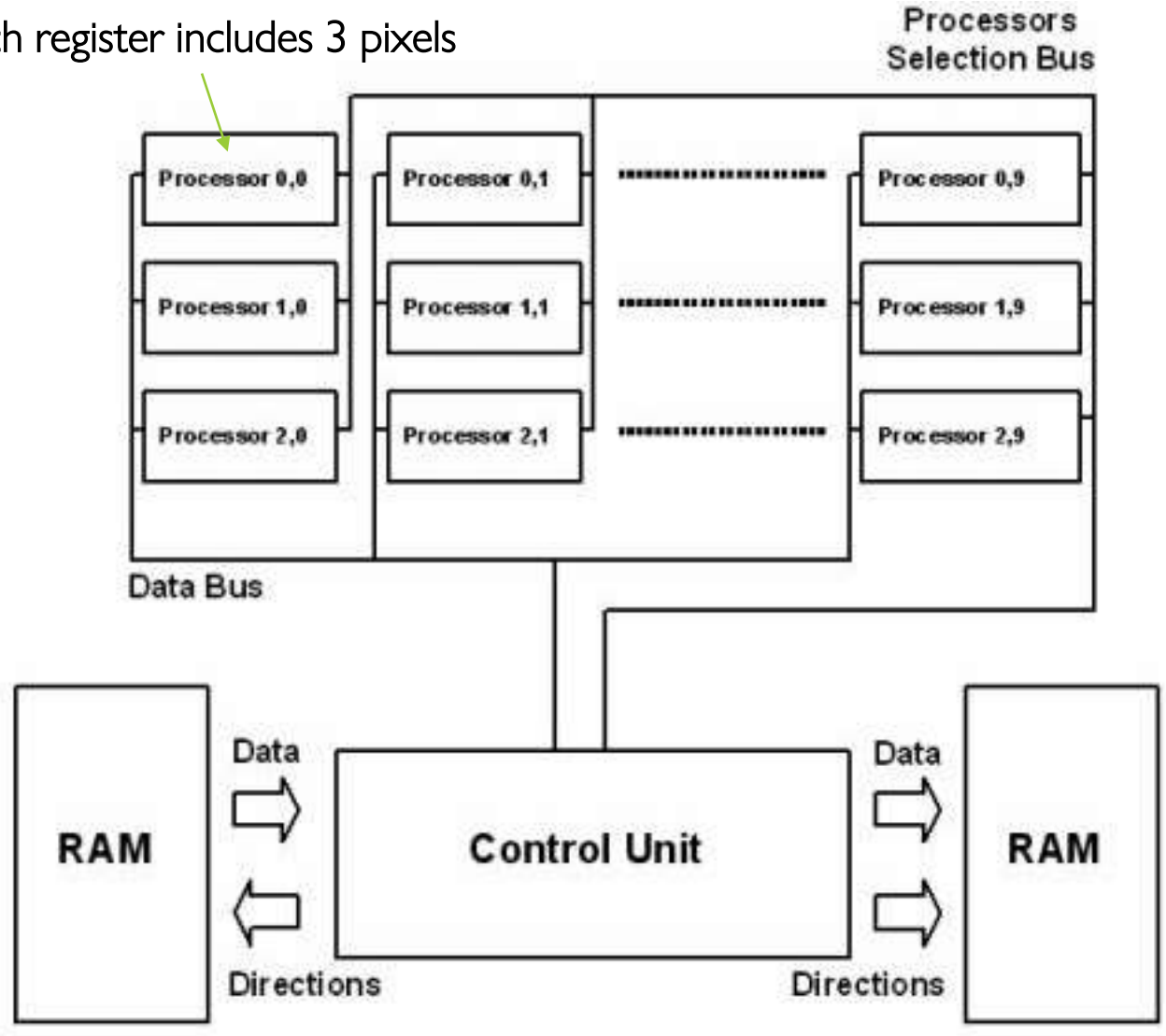
From the Sobel mask Fig. 2 is obtained (2).

$$G_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$$
$$G_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \qquad (2)$$

Where Z_i is the pixels intensity in the i position as it appears in Fig 3. And G_x, G_y are the x axe component and the y axe component of the vector gradient respectively.

Architecture: in FPGA

Each register includes 3 pixels



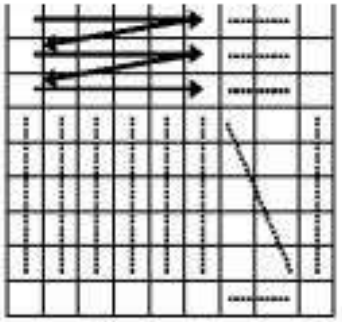
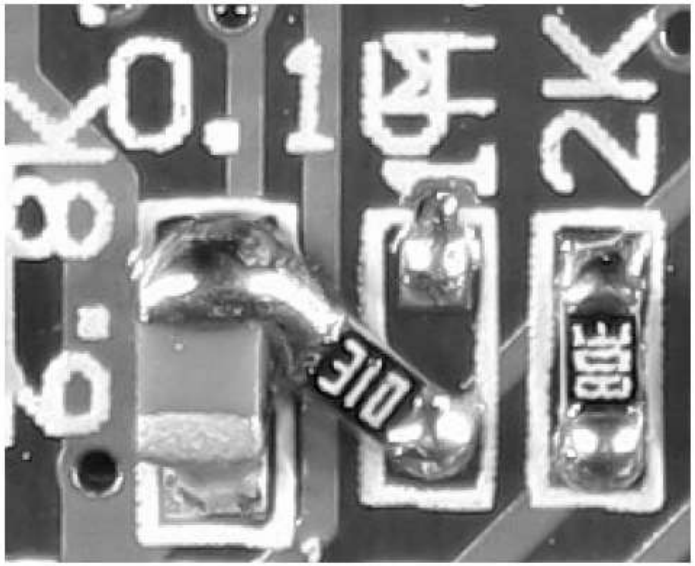
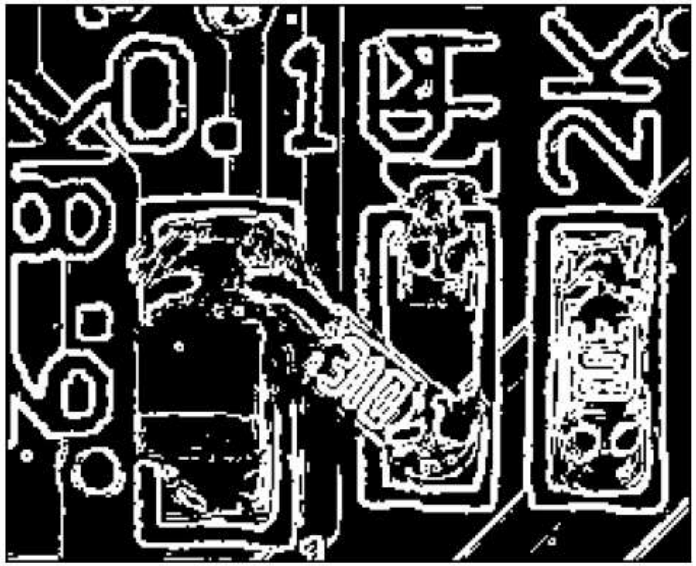


Figure 5. Pixels read sequence.

An example of Result





Result: For a 640x480 pixels image with a 40 MHz clock frequency the processing takes approximately 23.04 ms

When one schemes SPARC-20(Model 40) with von Newman architecture the same processing takes approximately 3.6 s

CONFIGURABLE FUNCTION BASED ARCHITECTURE

FPGA-based Image Processing App.

- The architecture introduces a schema based on the use of local storage buffers to reduce the number of access to data memories and router elements to handle data movement among different structures inside the same architecture
- The architecture can be applied for several applications:
  - 2D Filtering.
  - Motion Computation
  - ….

Saldaña-González, Griselda, and Miguel Arias-Estrada. *FPGA based acceleration for image processing applications*. INTECH Open Access Publisher, 2009.

Architecture Diagram

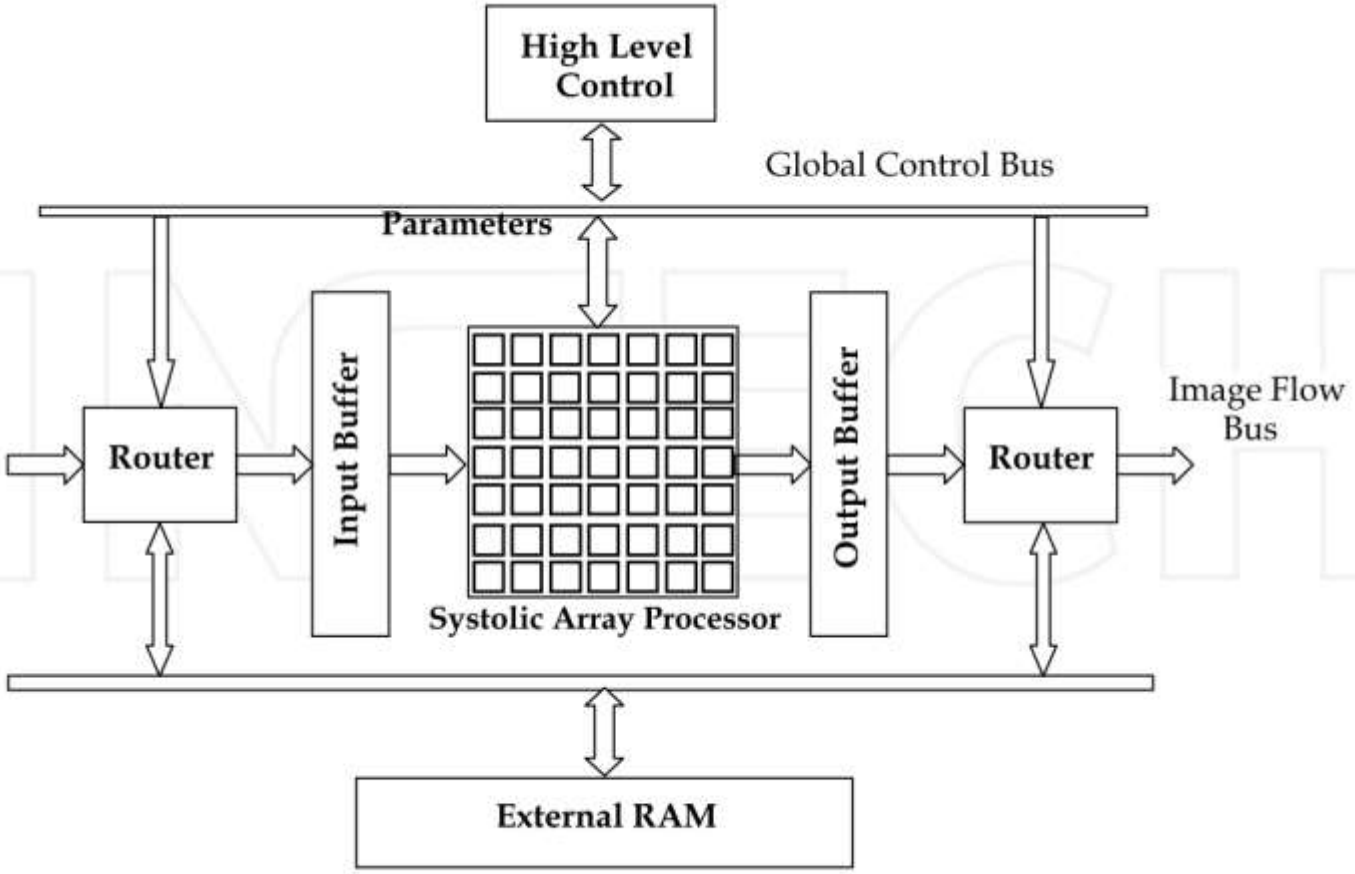


Fig. 2. Block diagram of the architecture

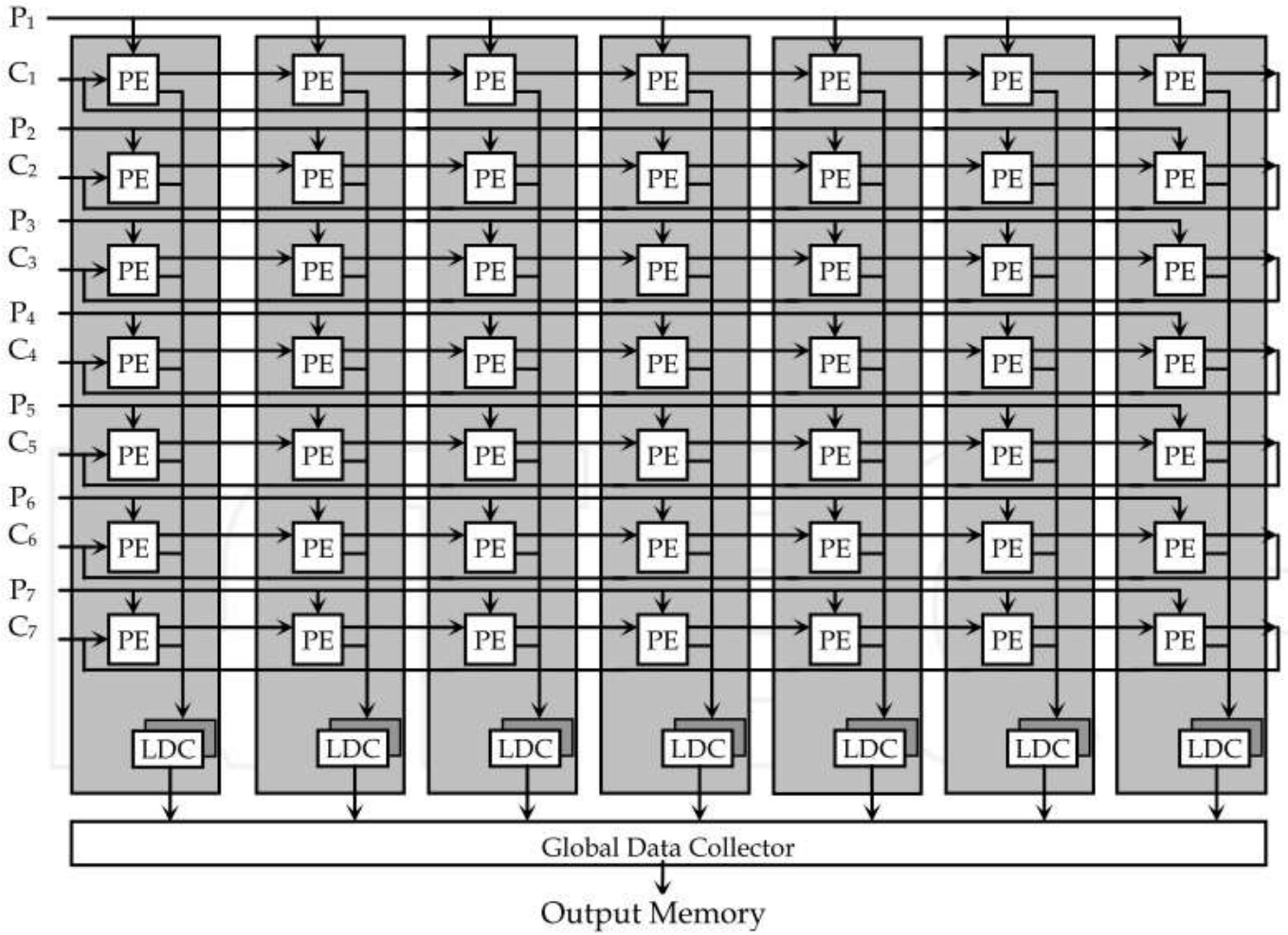


Fig. 3. 2D systolic array implementation

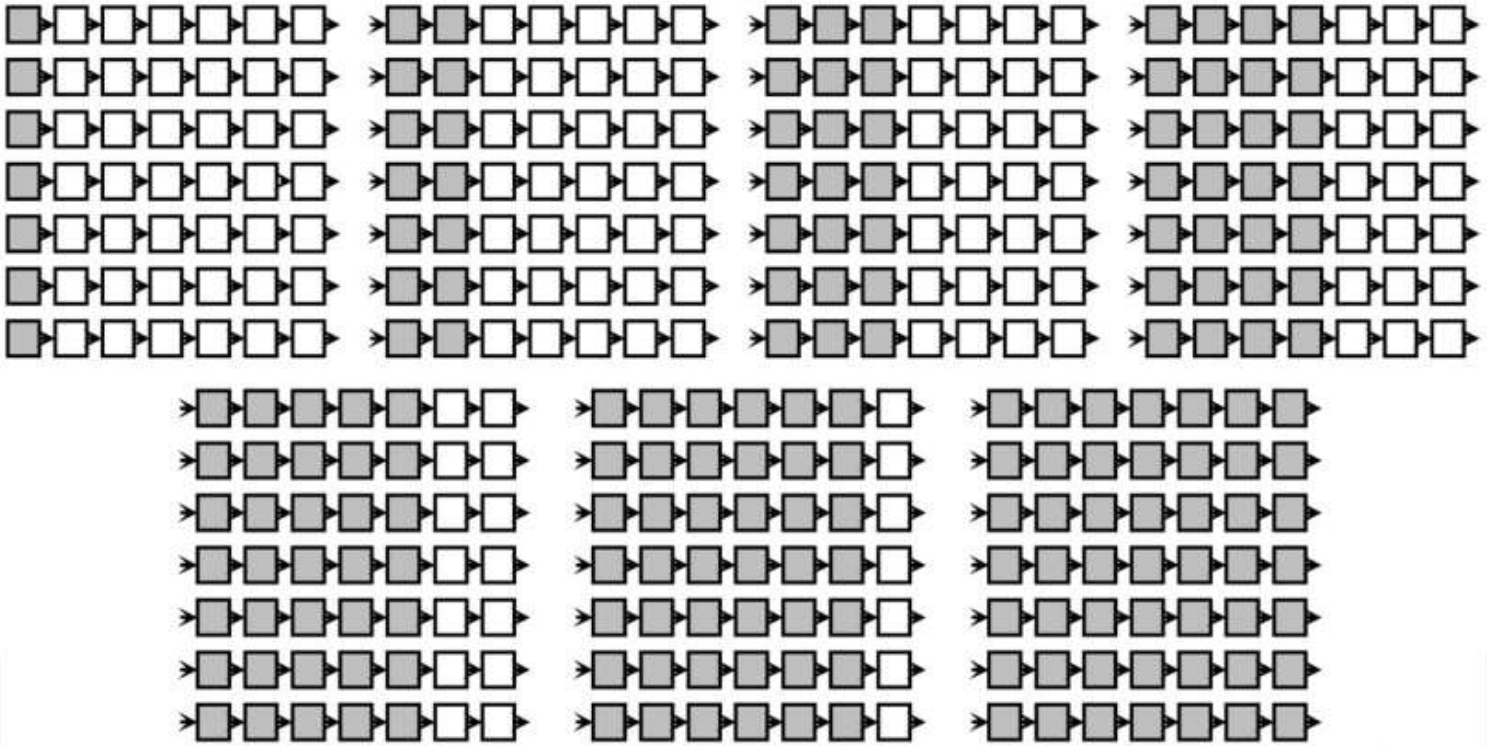


Fig. 4. PEs activation schema

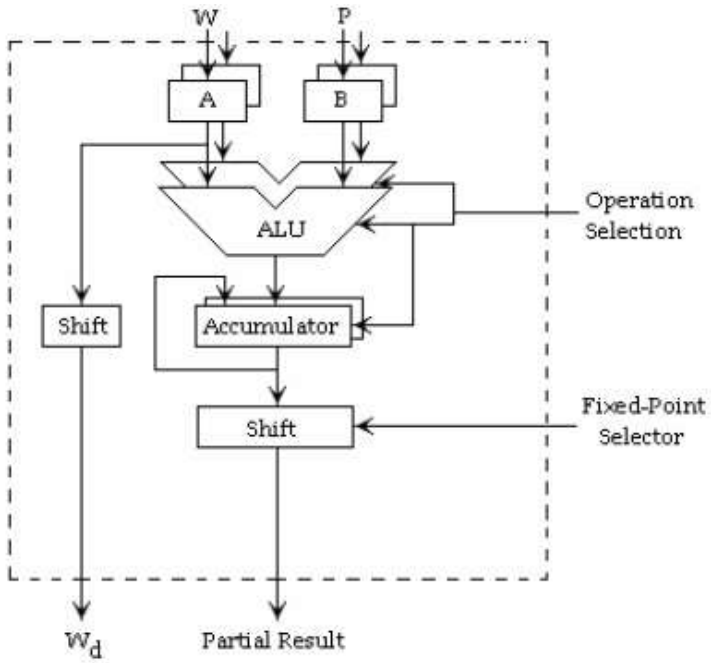


Fig. 10. PE modified structure to support ME algorithm

The processing element can be modified for another application.
For example: Motion Estimation

Summary of Architecture Performance

Application	Number of Slices	Clock Frequency	Power Consumption
Convolution	11,969 out of 19200	66 MHz	2.017 W
Filtering	11,969 out of 19200	66 MHz	2.017 W
Matrix multiplication	11,969 out of 19200	66 MHz	2.017 W
Gaussian pyramid	11,969 out of 19200	66 MHz	2.017 W
Erosion	12,114 out of 19200	66 MHz	2.4 W
Dilation	12,074 out of 19200	66 MHz	2.017 W

Table 1. Summary of the architecture performance

WHAT IF WE INTEGRATE THE SENSOR?

A high speed digital vision chip

- A digital vision chip an image sensor, each pixel of which has a processing element.
- Using a vision chip, high-speed image processing over 1000 frames/s can be achieved.

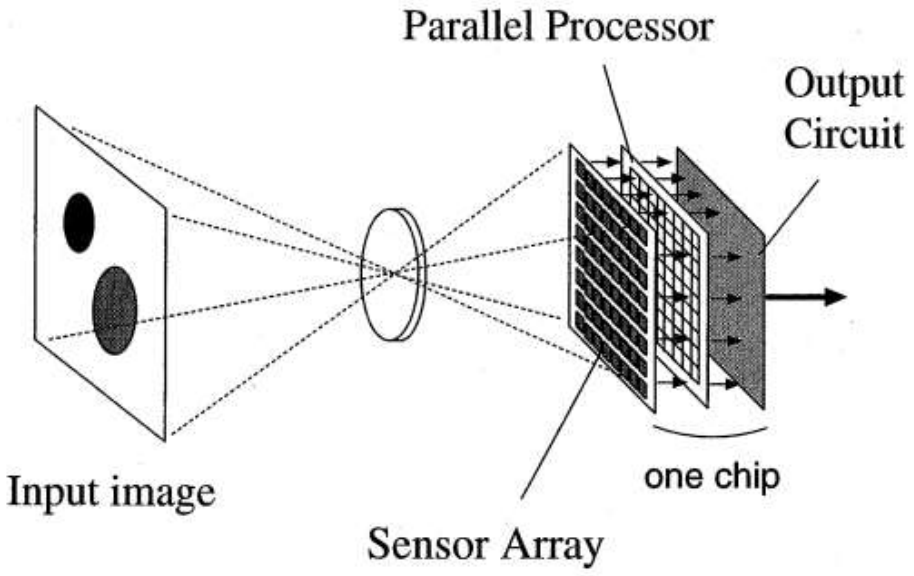
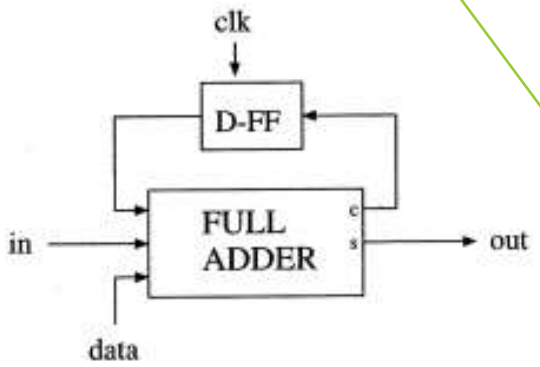


Fig. 1. Model of the digital vision chip.

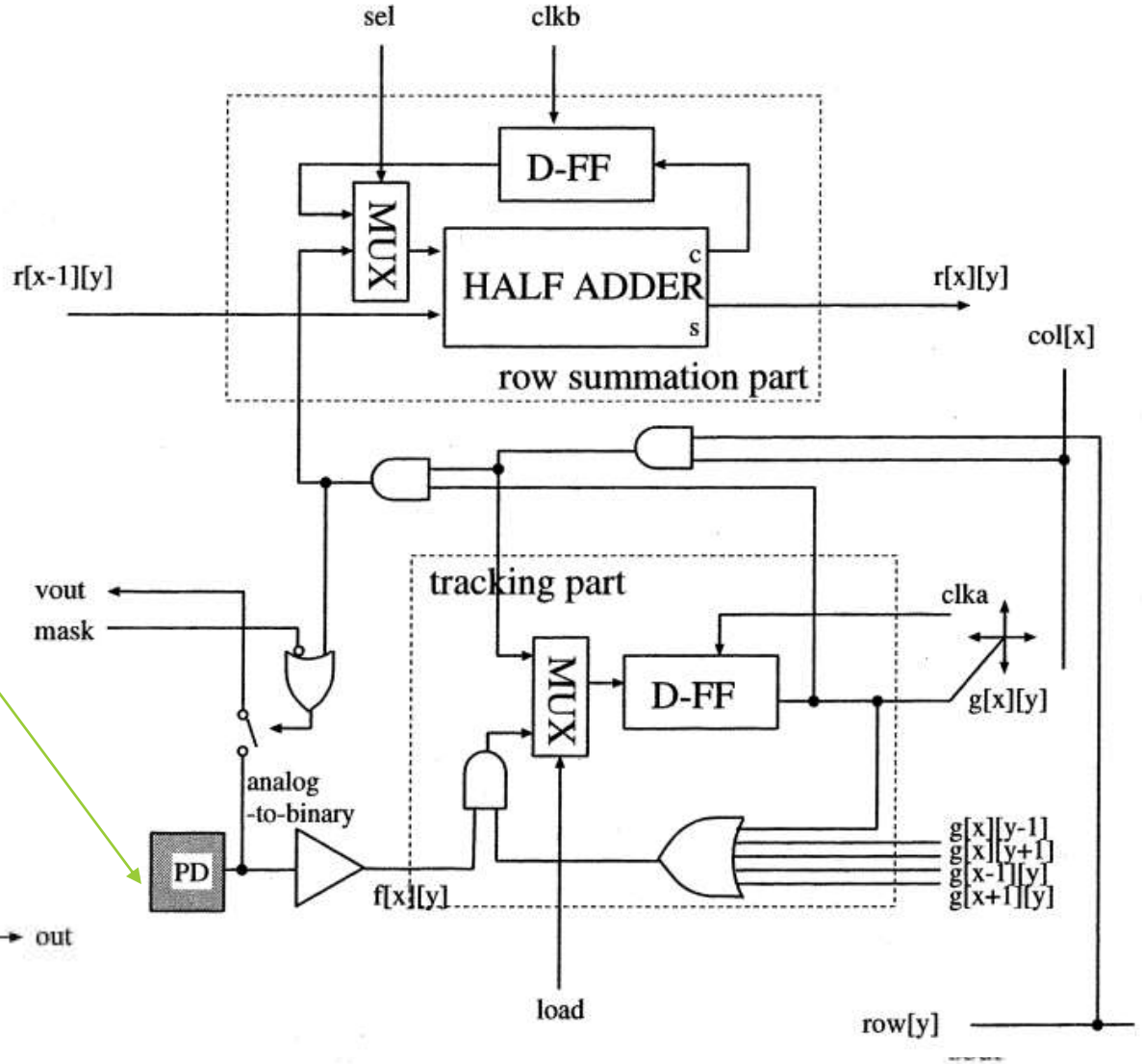
Komuro, Takashi, et al. "A digital vision chip specialized for high-speed target tracking." *Electron Devices, IEEE Transactions on* 50.1 (2003): 191-199.

Architecture of Vision Chip

Photo detector



BCSA



Processing elements architecture

Older version of Vision Chip

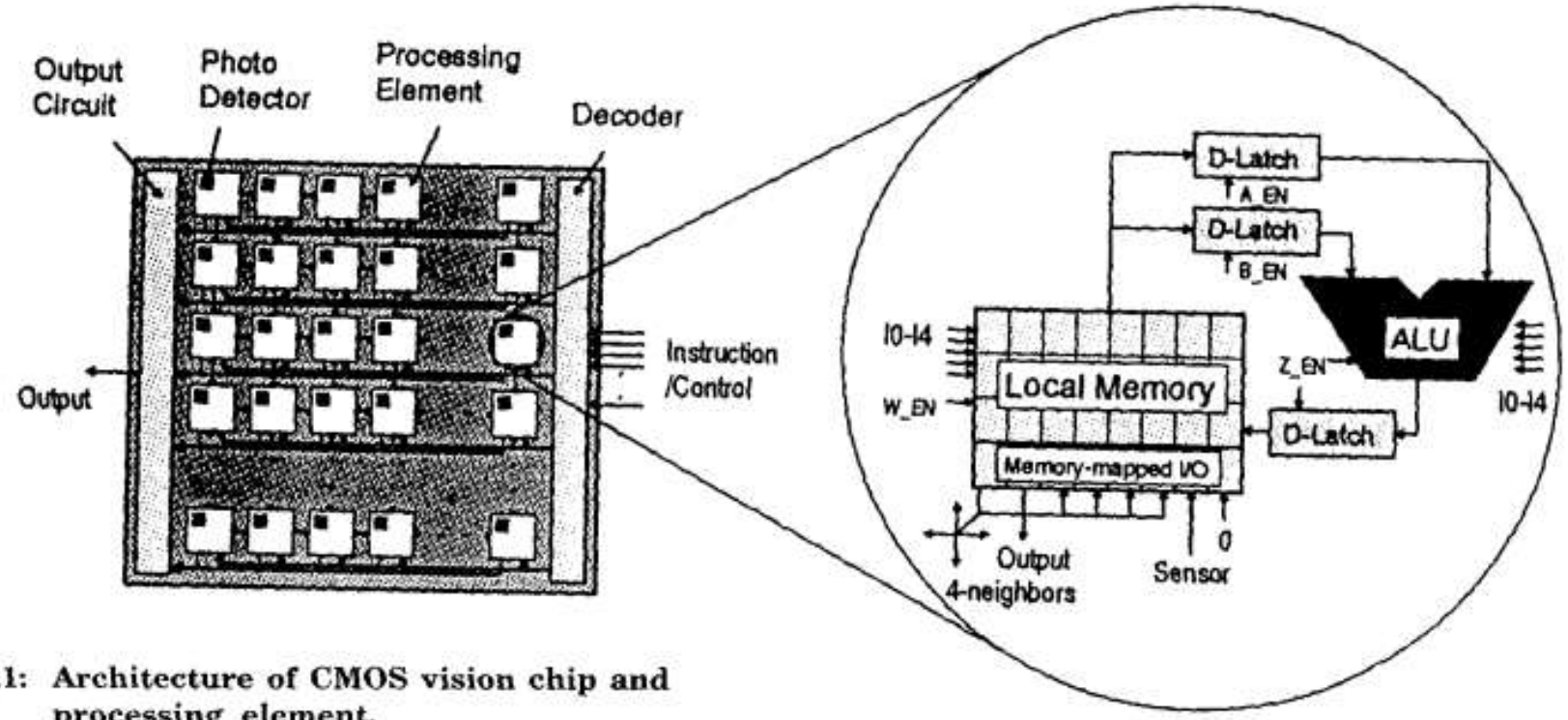
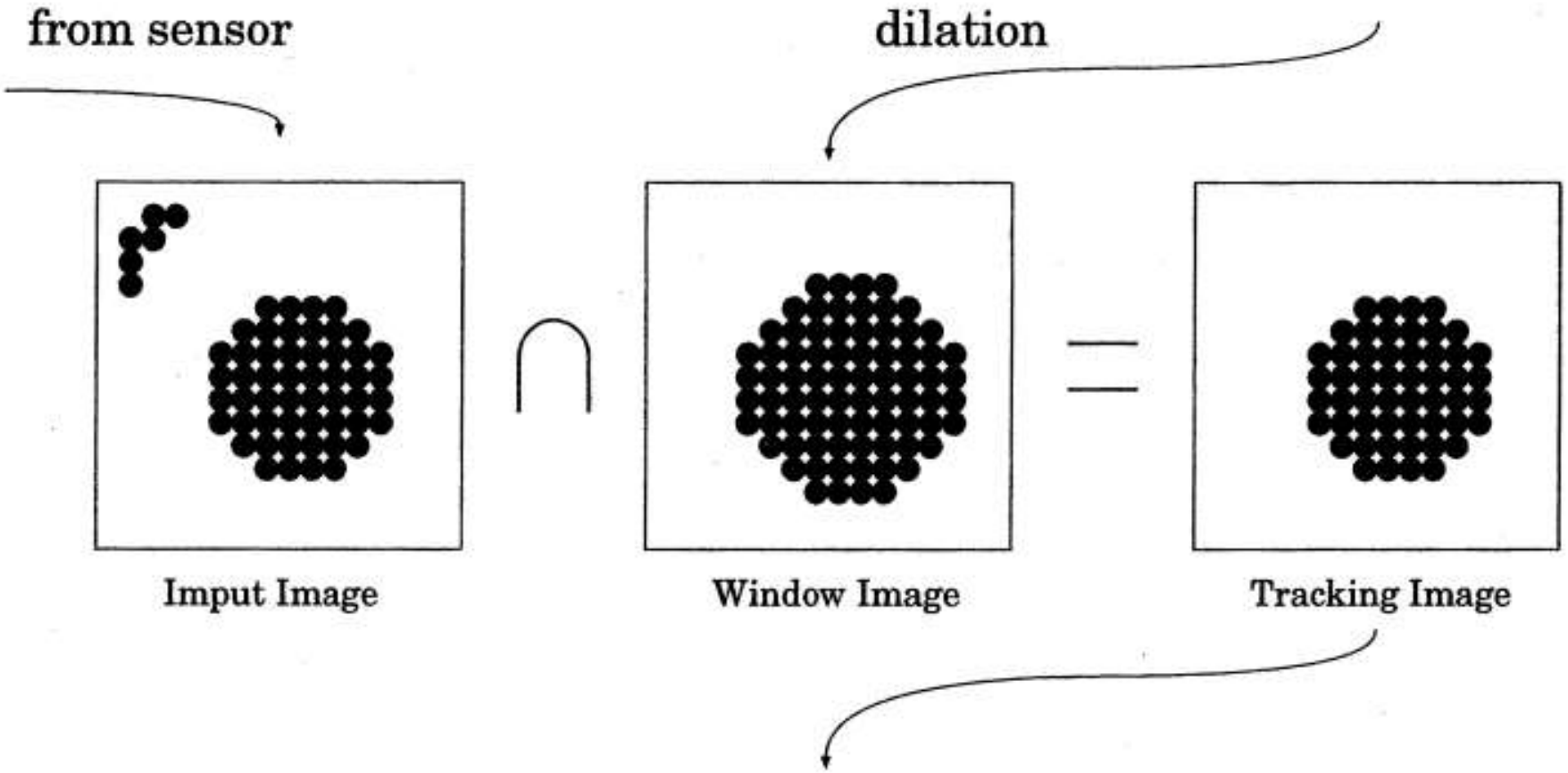


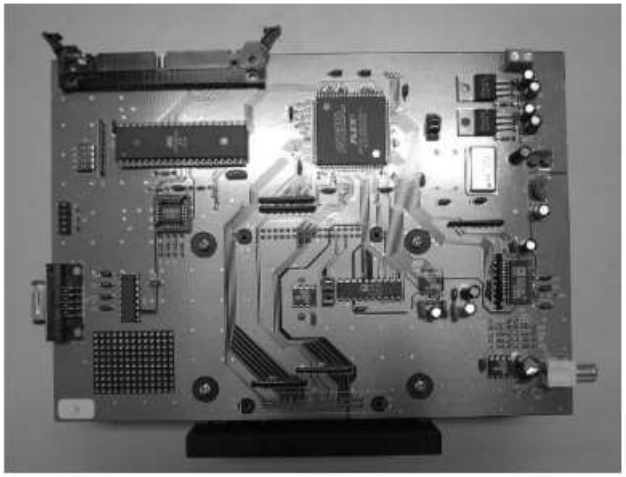
Figure 12.2.1: Architecture of CMOS vision chip and processing element.

Tracking Algorithm

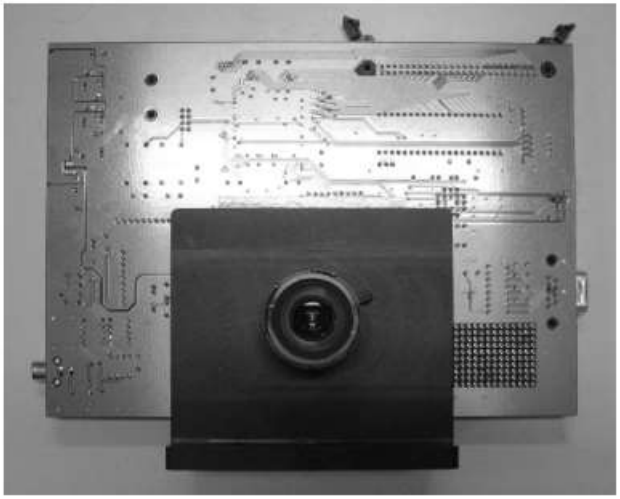


Repeat 2) and 3) with incrementing k

Demonstration



(a)



(b)

(a)

(b)

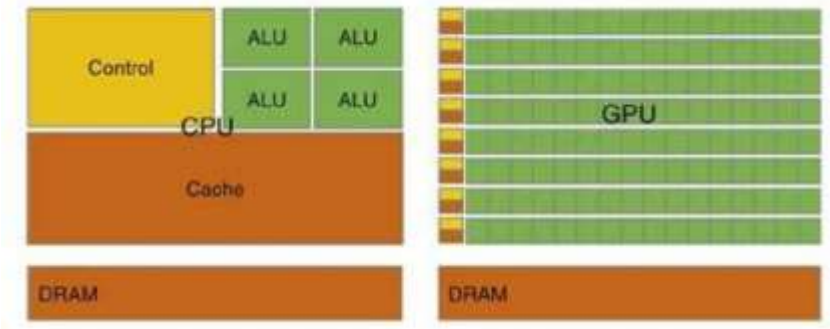
(c)

Fig. 10. Sample output. (a) Hand. (b) Face. (c) Face analog.

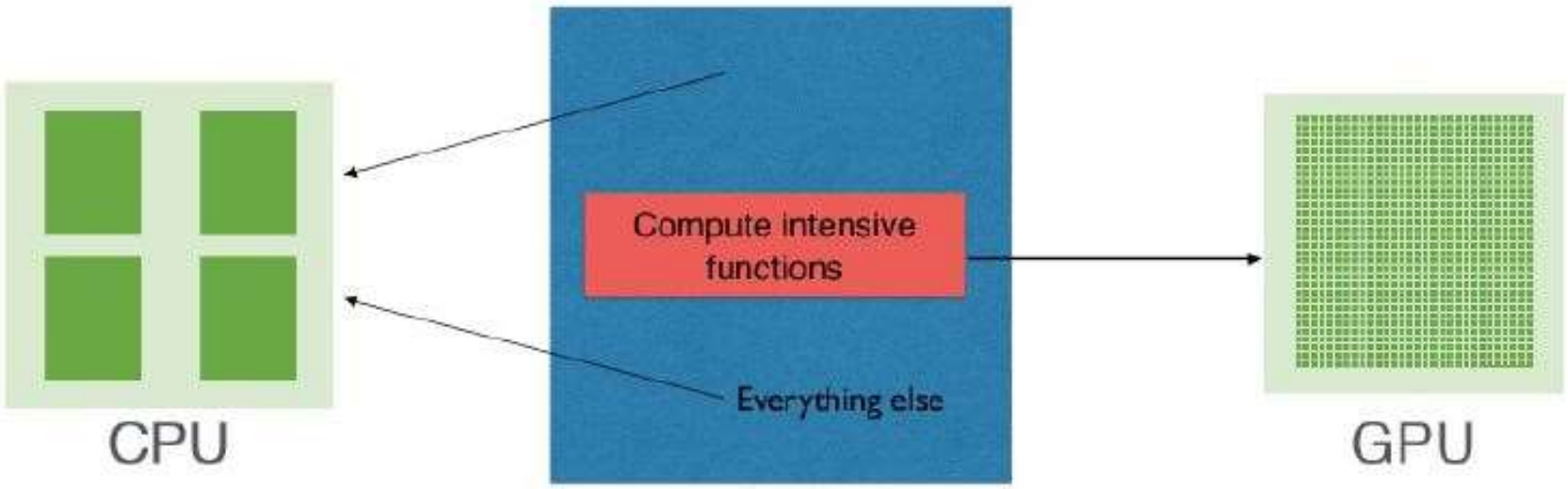
PROGRAMMABLE ARCHITECTURE

Graphic Processing Unit

- Processing is highly data-parallel
  - GPUs are highly multithreaded
  - Use thread switching to hide memory latency
    - Less reliance on multi-level caches
  - Graphics memory is wide and high-bandwidth
- Trend toward general purpose GPUs
  - Heterogeneous CPU/GPU systems
  - CPU for sequential code, GPU for parallel code
- Programming languages/APIs
  - Compute Unified Device Architecture (CUDA)



Both CPU and GPU are required



General Purpose GPU Computing (GPGPU)
Heterogeneous Computing

Getting Started: Software

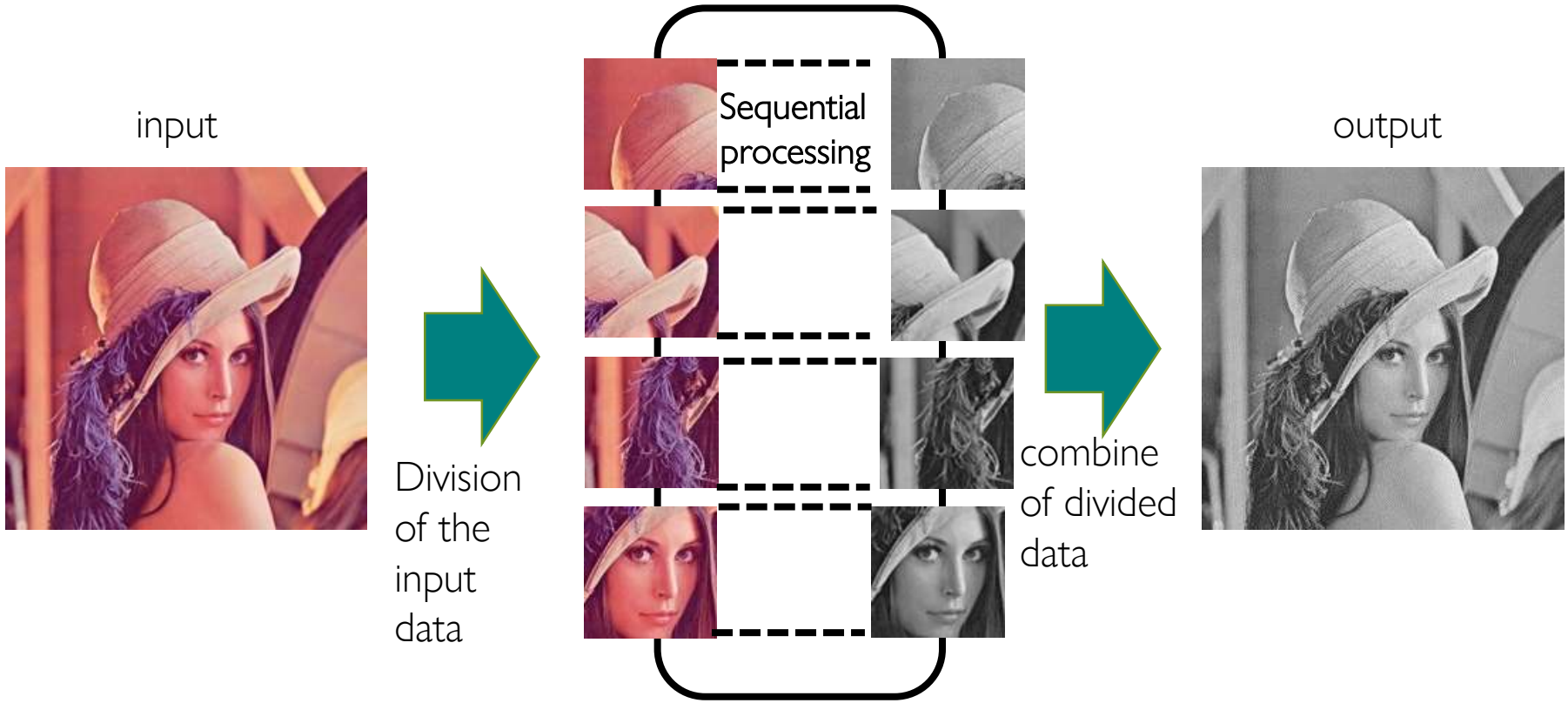
Programming CPU

- Sequential
- Write code top to bottom
- Can do complex tasks
- Independent

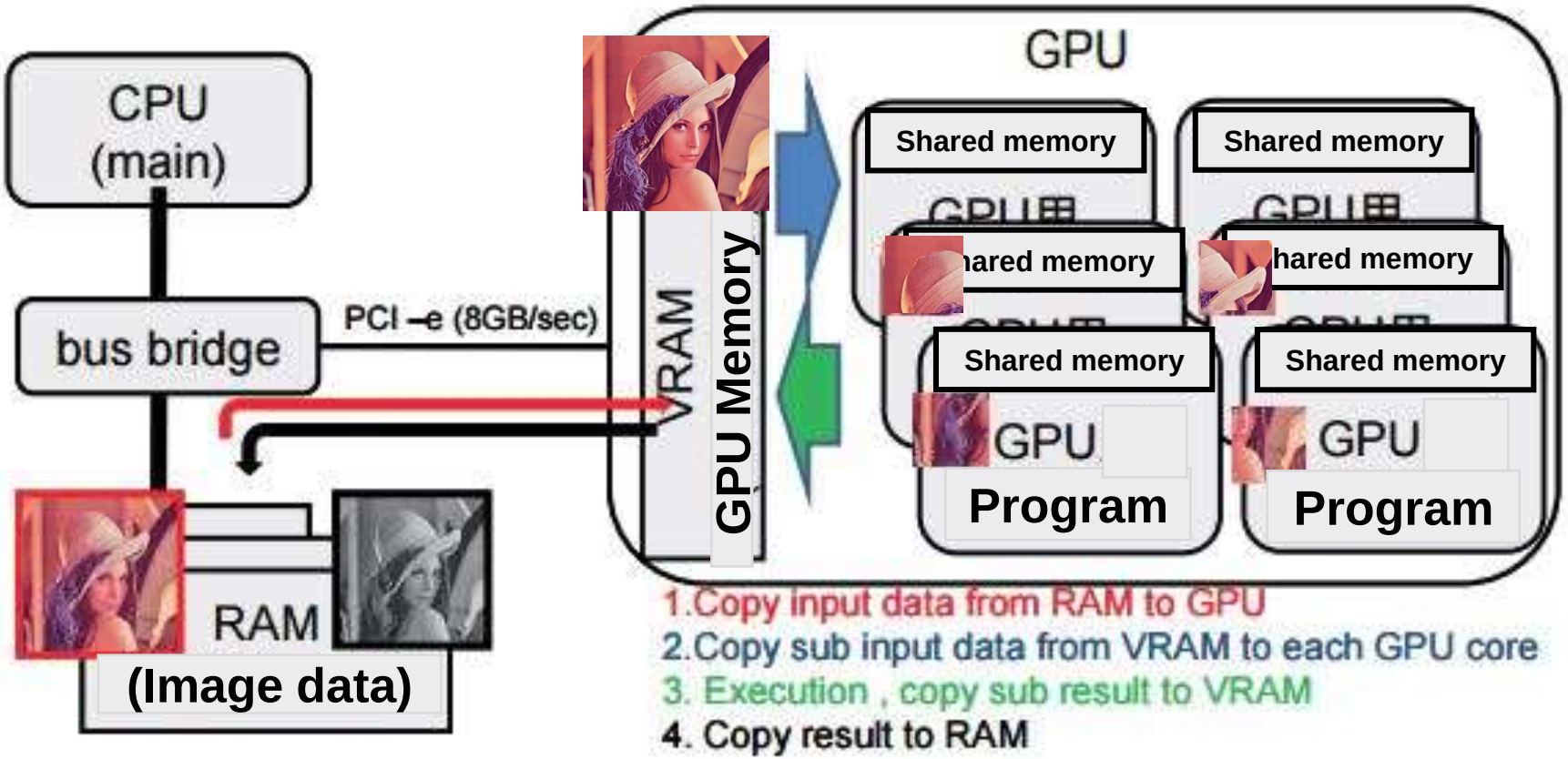
Programming GPU

- Parallel
- Multi-threaded - race conditions
- Low level tasks
- Dependent on CPU

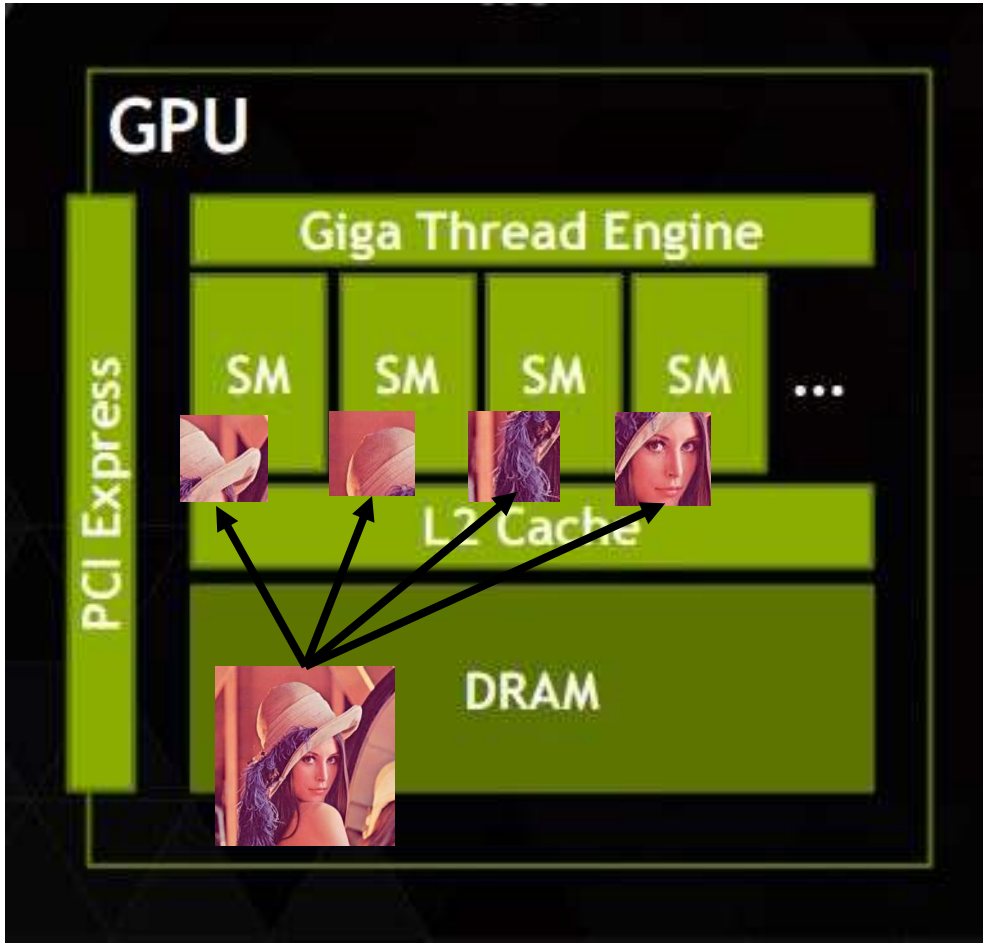
Overview of parallelization



Flow of image processing by the GPU

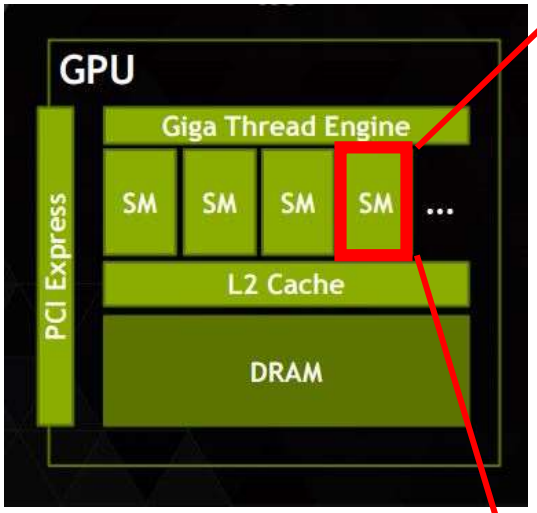


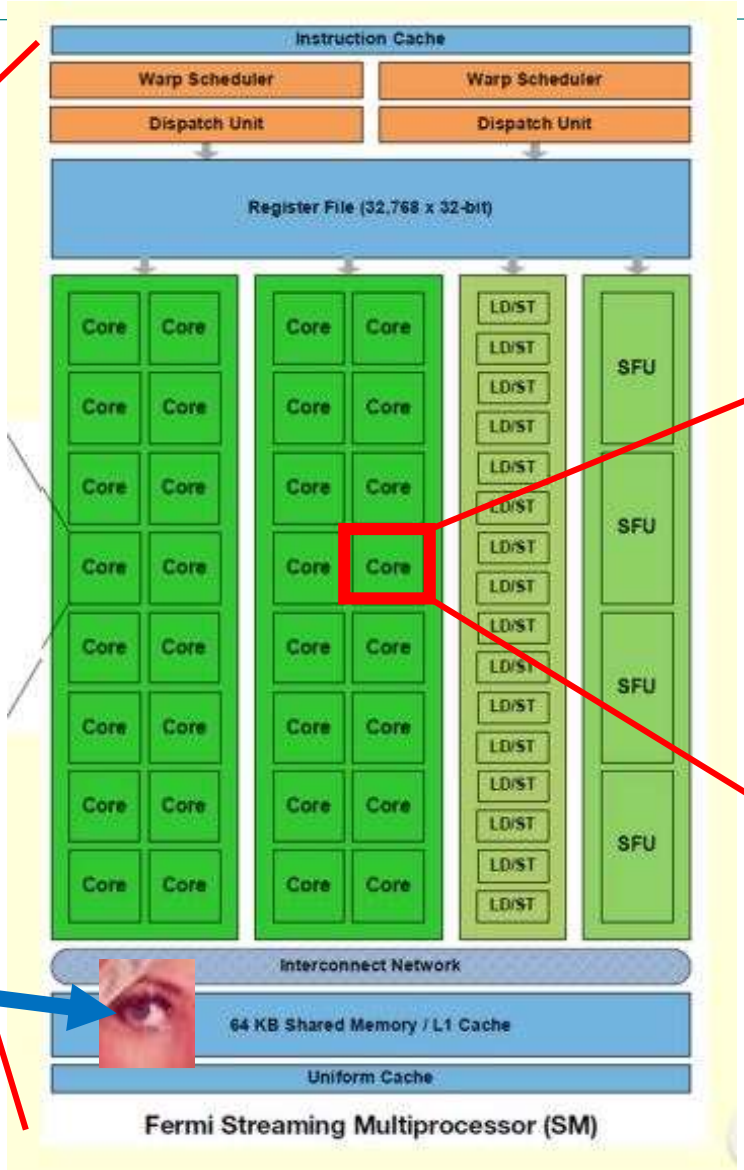
GPU Architecture

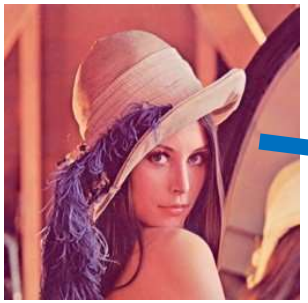


- Giga Thread Engine
  - The Thread Block Scheduler on the left
- SM(Streaming Multiprocessor)
  - Multithreaded SIMD Processors
- L2 Cache
  - Write-back with write allocate
- DRAM
- PCI Express
  - Graphic interface

SM = Streaming Multiprocessor







Conclusion

- "Vision is the act of knowing what is where by looking." – Aristotle
- To do it, the image is firstly pre-processed to extract the feature points.
- In order to accelerate the computing speed, additional hardware is the promising method.
- There are several approaches for hardware accelerators with both advantages and drawbacks.
- Depends on the requirement, the vision processing design can be optimized by a special approach or in combination of them.

Overview

- Introduction
  - What is Vision Processing?
- Low level Algorithms
- Classifications of Computing
  - SISD, SIMD, MISD, MIMD
- Architectures for Low Level Algorithm
- Conclusion

Conclusion

- "Vision is the act of knowing what is where by looking." – Aristotle
- To do it, the image is firstly pre-processed to extract the feature points.
- In order to accelerate the computing speed, additional hardware is the promising method.
- There are several approaches for hardware accelerators with both advantages and drawbacks.
- Depends on the requirement, the vision processing design can be optimized by a special approach or in combination of them.

References

- CS 143 Introduction to Computer Vision - James Hays ; http://cs.brown.edu/courses/cs143/ http://www.cse.psu.edu/~rtc12/CSE486/
- Computer Vision: Algorithms and Applications - Richard Szeliski http://szeliski.org/Book
- Previous COSCO slides: http://aslweb.u-aizu.ac.jp/wiki/index.php?COSCO
- Komuro, Takashi, et al. "A digital vision chip specialized for high-speed target tracking." *Electron Devices, IEEE Transactions on* 50.1 (2003): 191-199.
- Komuro, Takashi, Shingo Kagami, and Masatoshi Ishikawa. "A dynamically reconfigurable SIMD processor for a vision chip." *Solid-State Circuits, IEEE Journal of* 39.1 (2004): 265-268.
- Ishikawa, Masatoshi, et al. "A CMOS vision chip with SIMD processing element array for 1 ms image processing." *Solid-State Circuits Conference, 1999. Digest of Technical Papers. ISSCC. 1999 IEEE International*. IEEE, 1999.
- Rosas, Roberto López, Adriano De Luca, and Francisco Barbosa Santillan. "SIMD architecture for image segmentation using sobel operators implemented in FPGA technology." *Electrical and Electronics Engineering, 2005 2nd International Conference on*. IEEE, 2005.
- Saldaña-González, Griselda, and Miguel Arias-Estrada. *FPGA based acceleration for image processing applications*. INTECH Open Access Publisher, 2009.

Thank you for your attention !

Let's do Question and Answer !!!