2014 IEEE CASE
INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERING
INTERNATIONAL CONFERENCE ON AUTOMATION SCIENCE & ENGINEERING

IEEE
Robotics &
Automation
Society

In-Line Detection of Surface Profile and Defects Using Full-Field and High-Precision Automated Optical Inspection with Cloud-Based Analysis System

Ming Chang, Yu-Cheng Chou, Po Ting Lin

Department of Mechanical Engineering
Chung Yuan Christian University
Chungli, Taoyuan, Taiwan

中原大學

科技部
Ministry of Science and Technology

Optical Measurement
Optomechatronics
Automated Optical Inspection

Parallel Computing
Cloud Computing
Software Design
Intelligent Embedded System

Design Under Uncertainty
Multidisciplinary Design Optimization
Multiobjective Optimization

Ming Chang, Yu-Cheng Chou, Po Ting Lin

Department of Mechanical Engineering
Chung Yuan Christian University
Chungli, Taoyuan, Taiwan

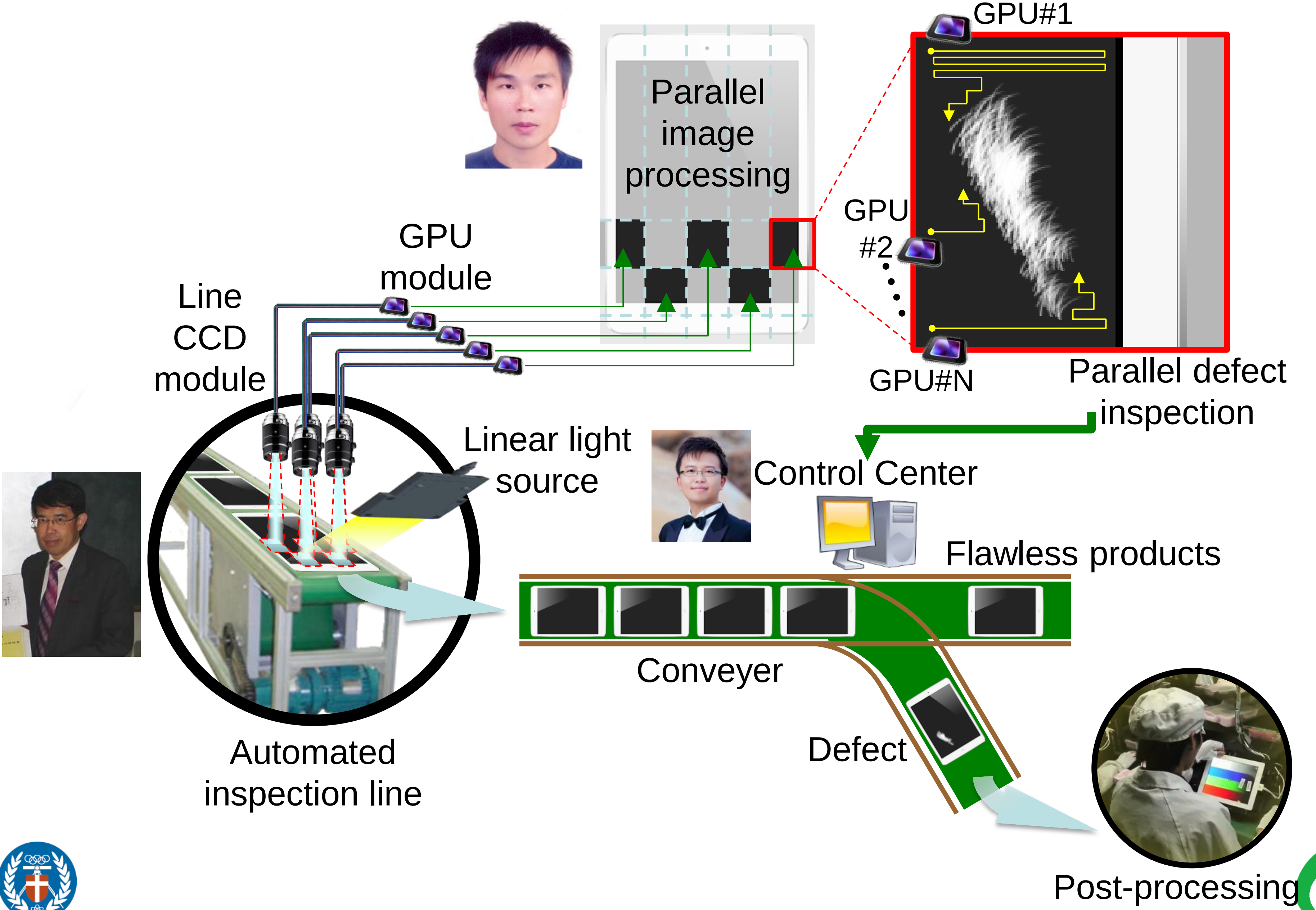
中原大學

科技部
Ministry of Science and Technology

Inspection of Handheld Devices

All images are from google.com.

Full-field In-Line Inspection System



Parallel image processing

GPU#1

GPU #2

GPU#N

Parallel defect inspection

GPU module

Line CCD module

Linear light source

Control Center

Flawless products

Conveyer

Defect

Automated inspection line

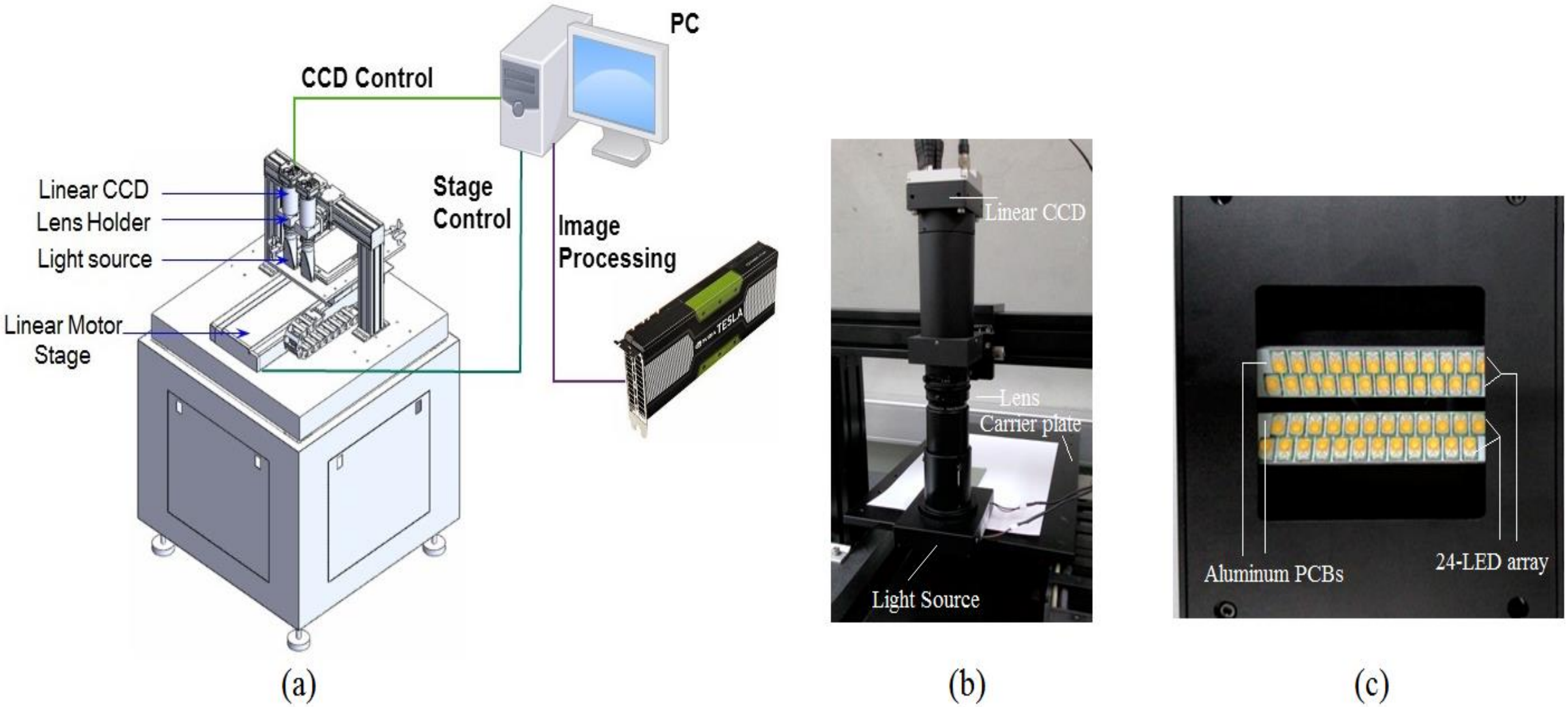
Post-processing

2D Inspection of Defect in Back-coated Glass

Motivation and Objectives

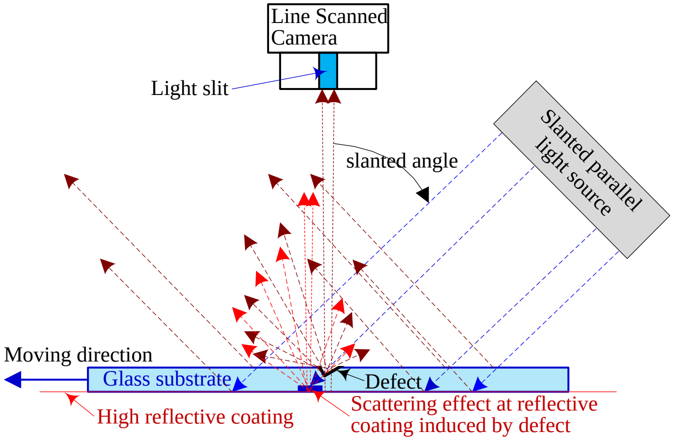
- Numerous methods have been proposed for different applications of surface defect detection, and those methods can generate satisfactory performance.
- However, those methods have not addressed a situation where both the speed and precision requirements need to be satisfied simultaneously
- In other words, an image to be processed by a machine during each time window has hundreds of mega pixels, whereas the time window is within one second.
- This presentation shows a highly expandable distributed image sensor computing system, DISCS, to achieve in-line surface defect detection with high performance on both the speed and precision.

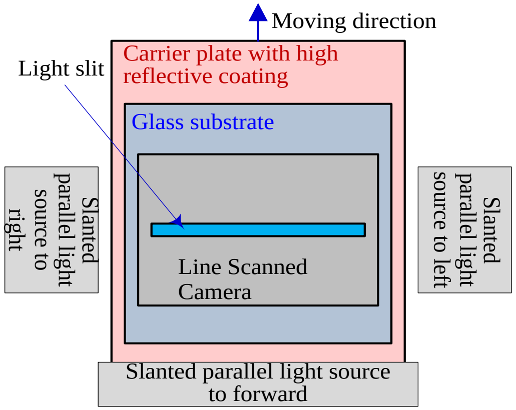
Apparatus

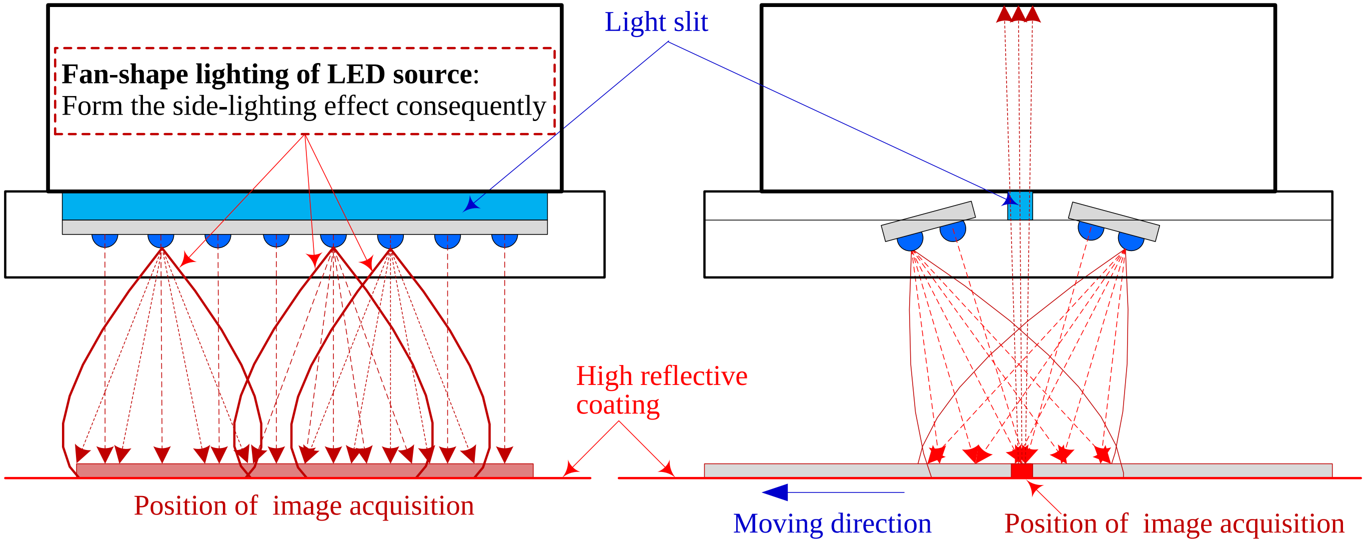


(a) Hardware components of the optical inspection platform.
(b) Actual photo of one opto-mechanical module.
(c) Construction of illumination device with two 24-LED arrays

Inspection Principle



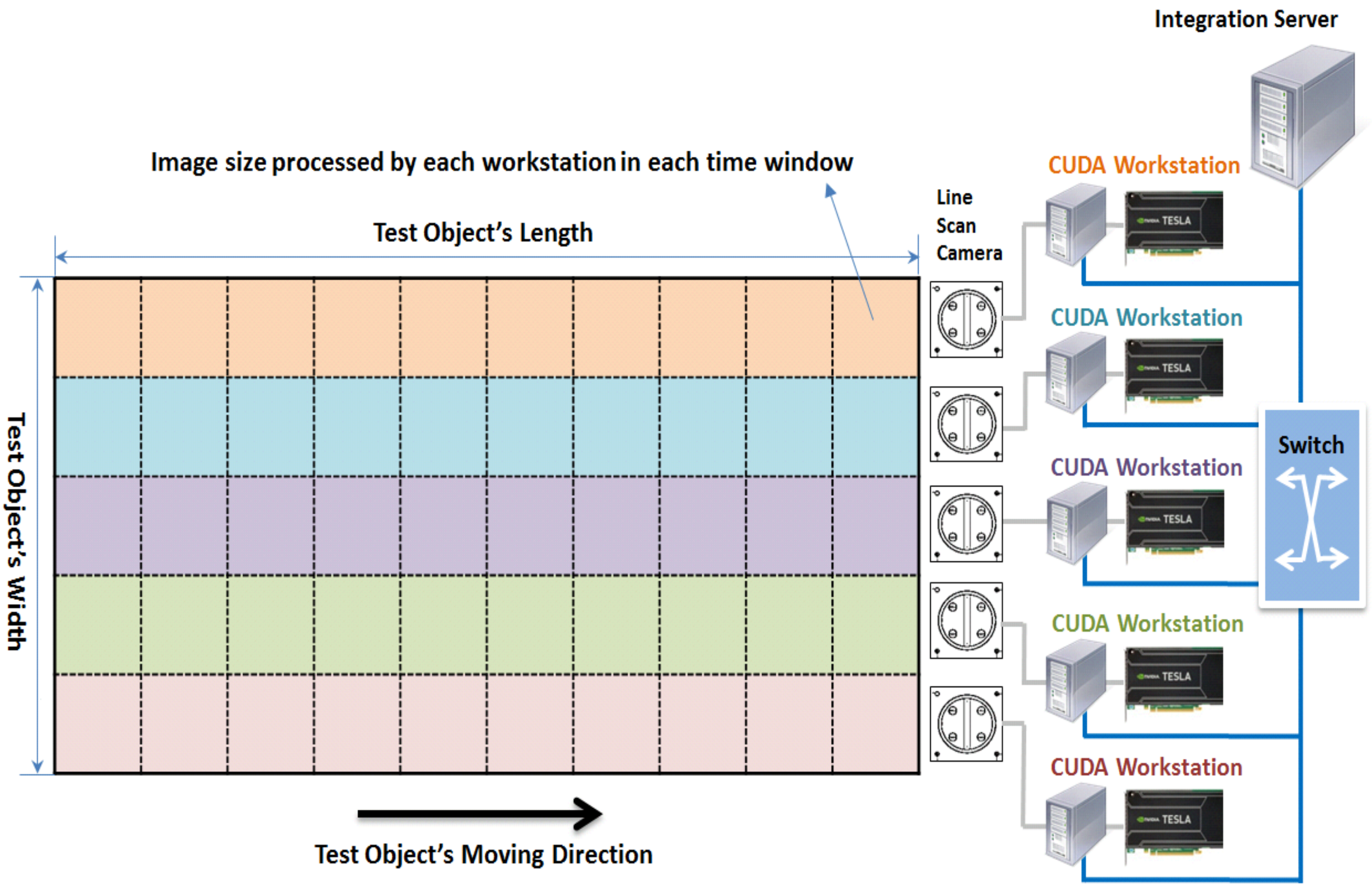




Distributed Image Sensor Computing System (DISCS)

- Characteristics
  - Heterogeneous parallel computing system
  - Consists of multiple CPUs and GPUs
  - Adopts Message Passing Interface (MPI) and Compute Unified Device Architecture (CUDA) programming models
  - High speed and high precision in-line detection of surface defects
- Hardware Development
  - Consists of independent machines that form a master-slave parallel computing model.
  - Each CUDA workstation is a slave machine, which performs the same computations and sends the result to the master machine and has at least one CPU and one GPU.
  - All the machines are connected through a high-speed network.

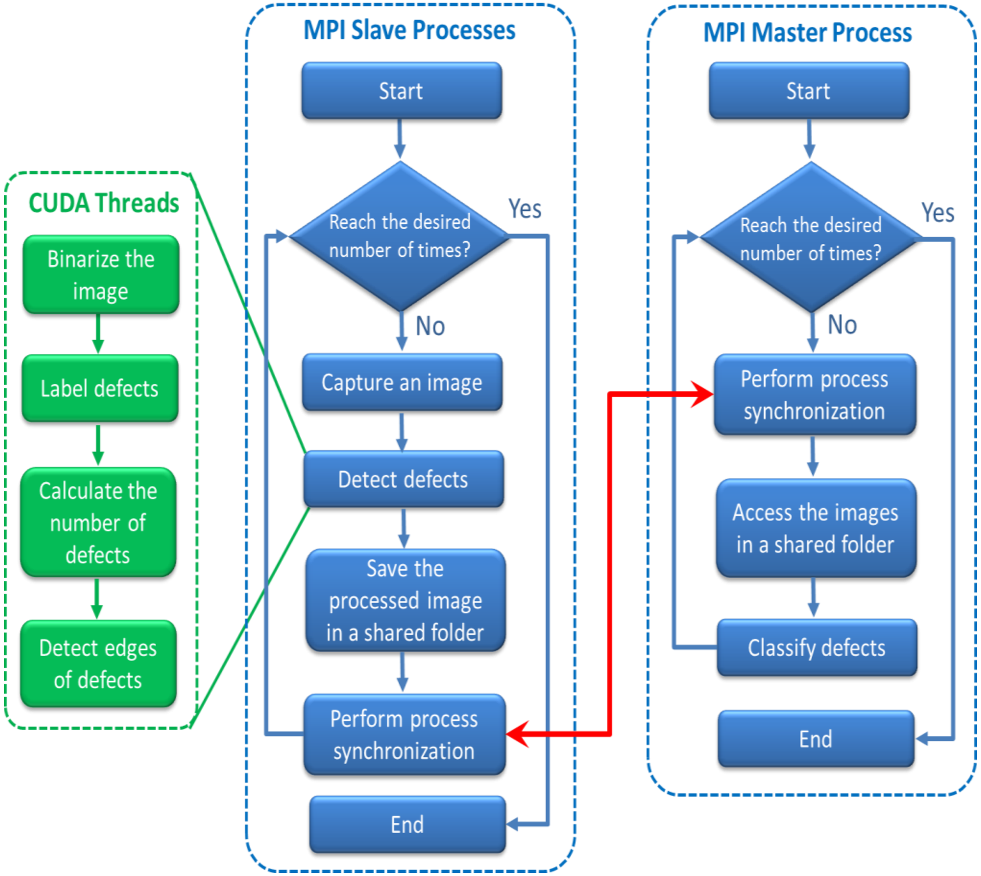
System Architecture - Hardware



Hardware architecture of the DISCS

System Architecture - Software

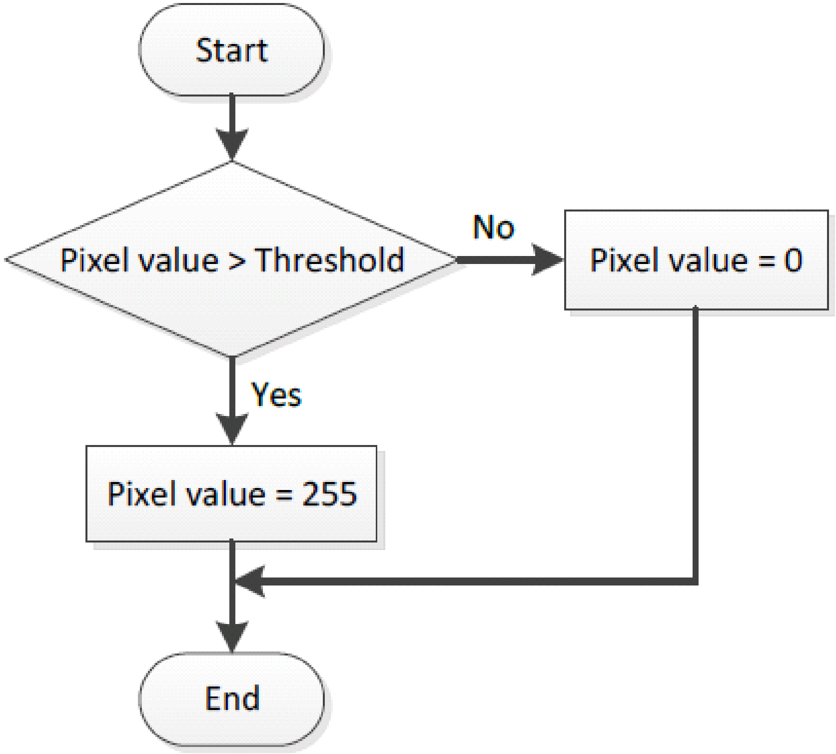
- Consists of MPI processes and CUDA threads.
- MPI processes run in CPUs and CUDA threads run in GPUs.
- The MPI master process runs on the integration server, whereas the MPI slave processes run on the CUDA workstations.
- The idea of the DISCS is to let the MPI slave processes handle the defect detection, and let the MPI master process deal with the defect classification.



Software architecture of the DISCS.

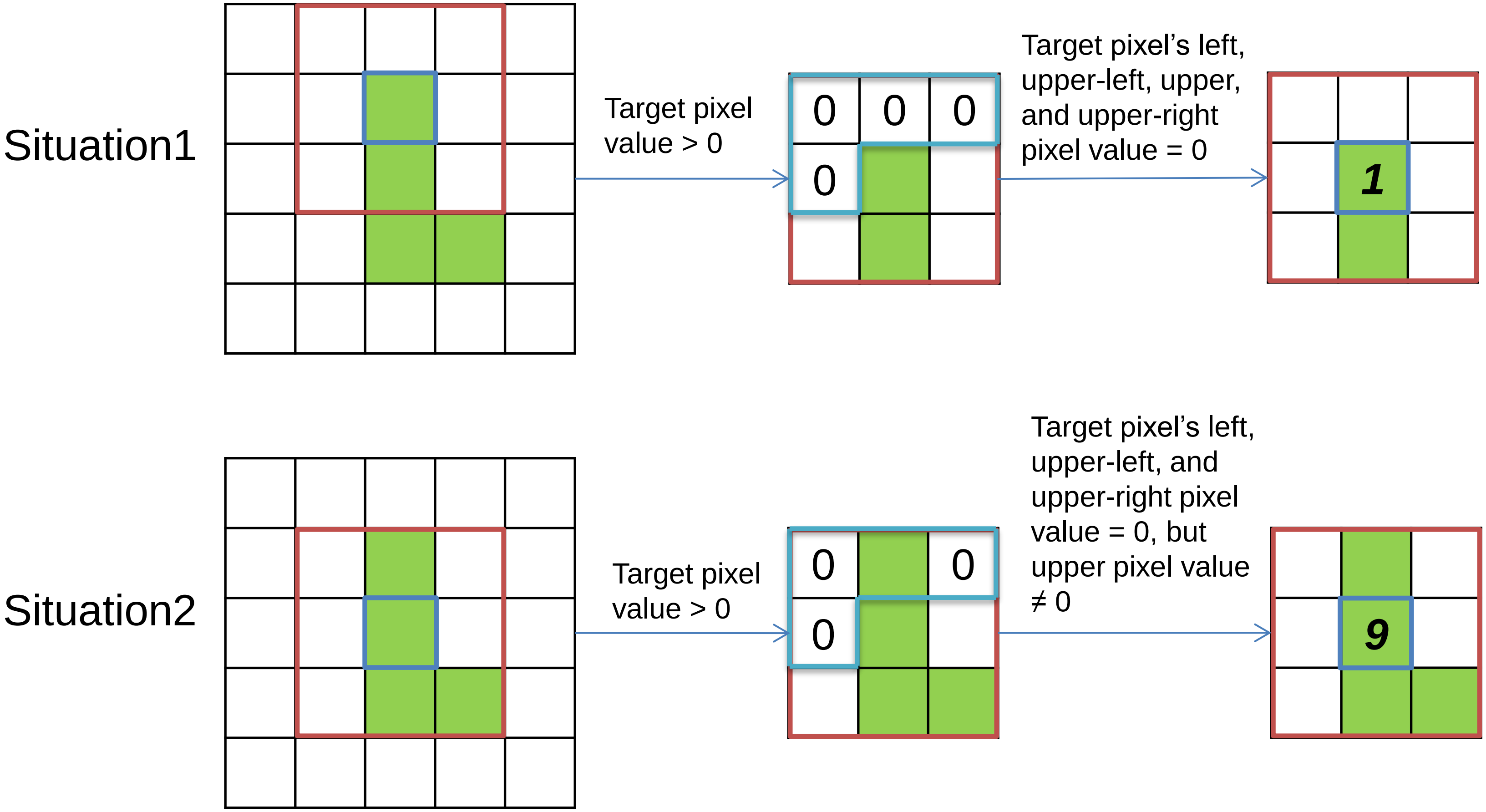
CUDA-Based Defect Detection Algorithms - Binarization

- Straightforward and based on a predefined threshold.
- If a pixel value is larger than the threshold, the pixel value is set to 255.
- Otherwise, the pixel value is set to zero.



The binarization algorithm.

CUDA-Based Defect Detection Algorithms - Labeling



Situation1

Target pixel value > 0

Target pixel's left, upper-left, upper, and upper-right pixel value = 0

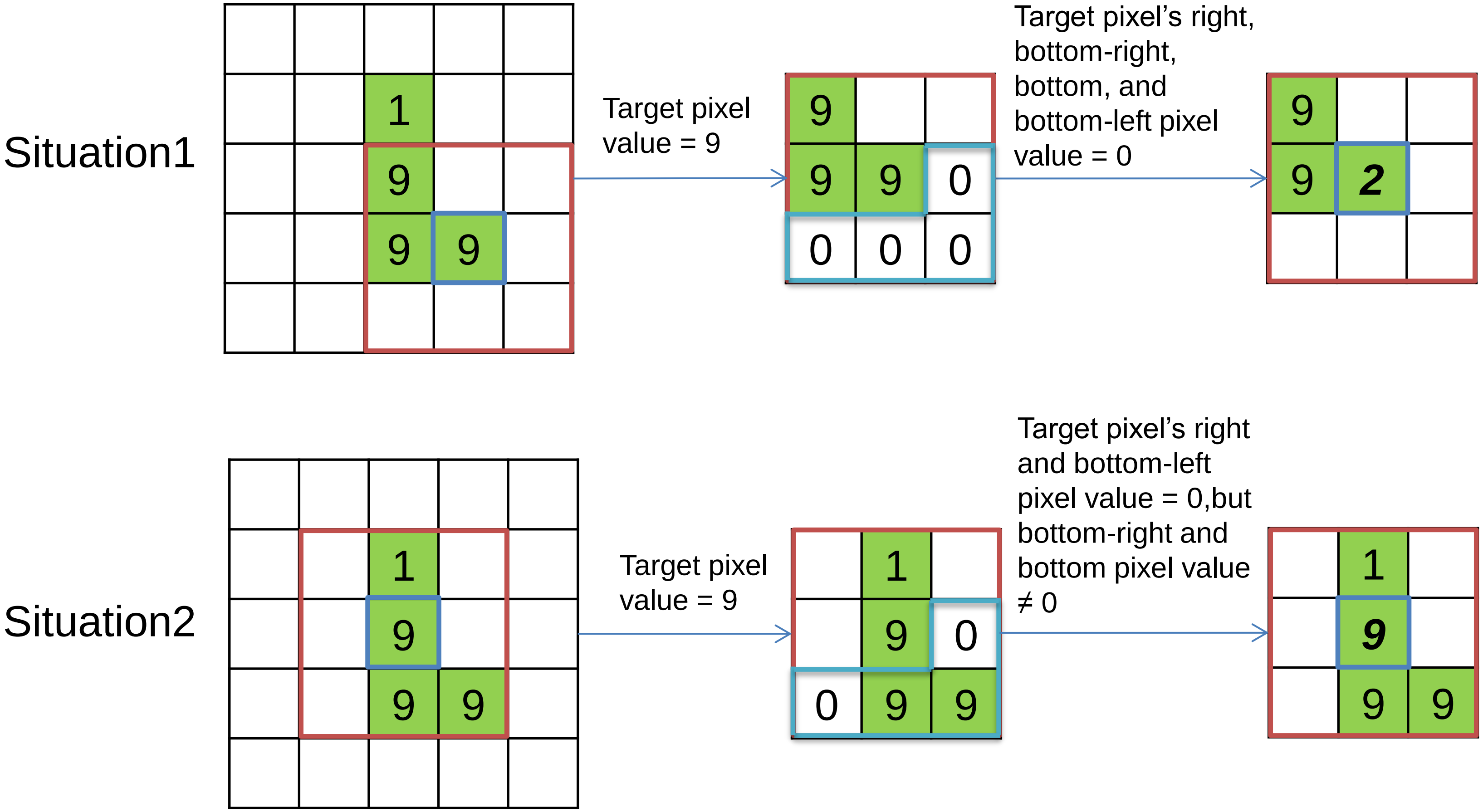
Situation2

Target pixel value > 0

Target pixel's left, upper-left, and upper-right pixel value = 0, but upper pixel value ≠ 0

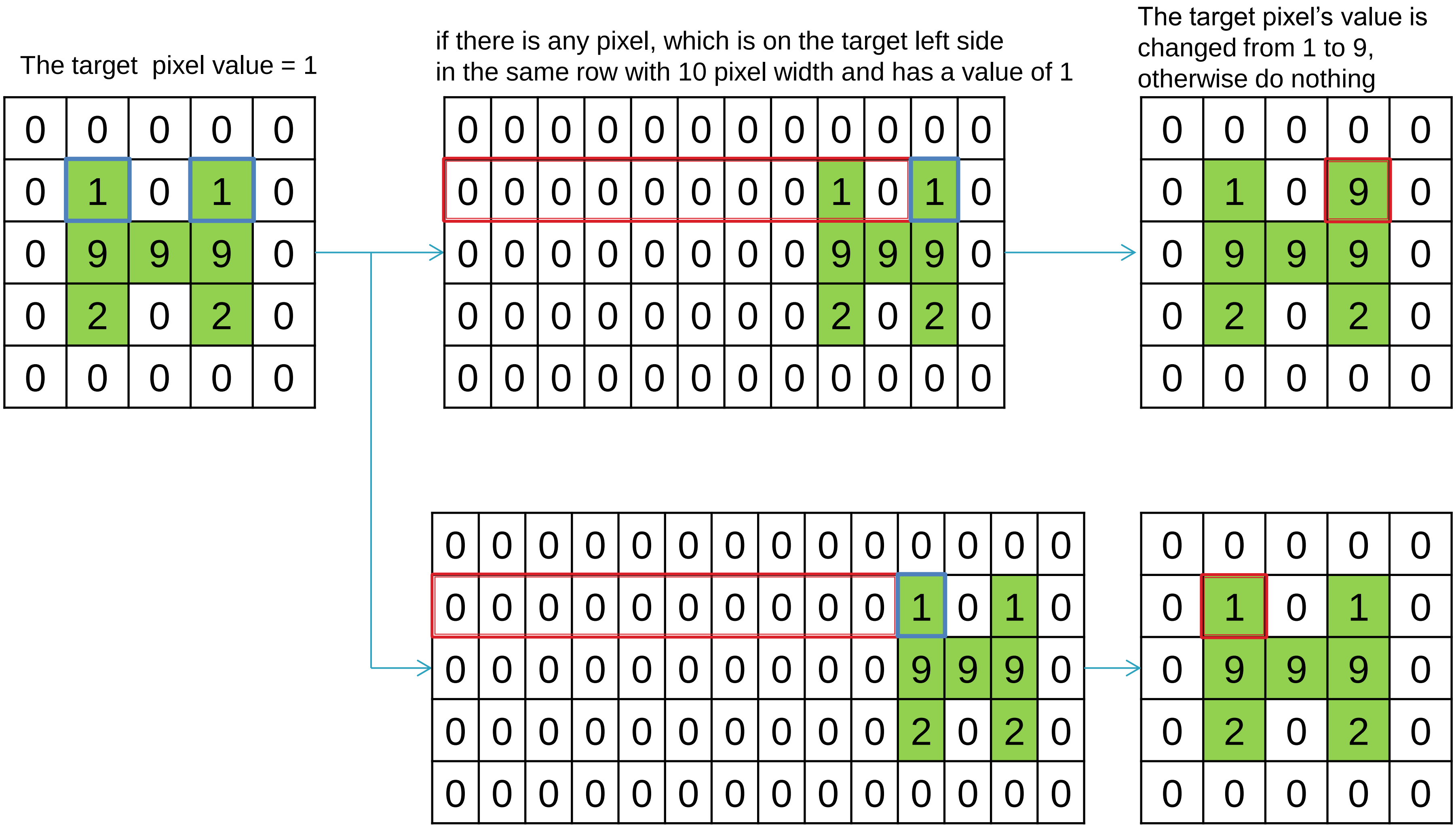
Determination of Starting Point

CUDA-Based Defect Detection Algorithms - Labeling



<u>Determination of End Point</u>

CUDA-Based Defect Detection Algorithms - Labeling



The target pixel value = 1

0	0	0	0	0
0	1	0	1	0
0	9	9	9	0
0	2	0	2	0
0	0	0	0	0

if there is any pixel, which is on the target left side
in the same row with 10 pixel width and has a value of 1

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	9	9	9	0
0	0	0	0	0	0	0	0	2	0	2	0
0	0	0	0	0	0	0	0	0	0	0	0

The target pixel's value is
changed from 1 to 9,
otherwise do nothing

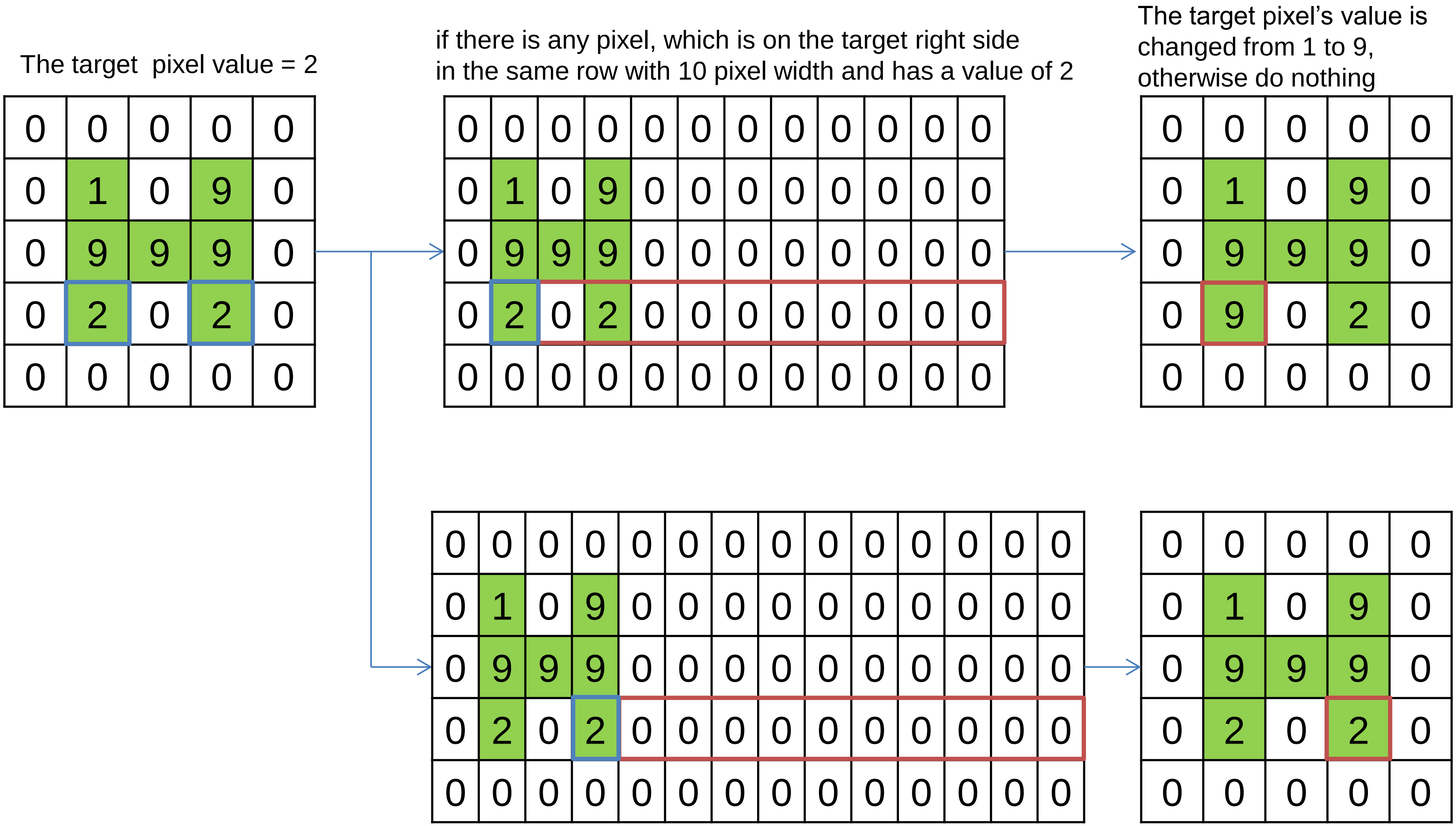
0	0	0	0	0
0	1	0	9	0
0	9	9	9	0
0	2	0	2	0
0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	9	9	9	0
0	0	0	0	0	0	0	0	0	0	2	0	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0
0	1	0	1	0
0	9	9	9	0
0	2	0	2	0
0	0	0	0	0

Elimination of Redundant Starting Point

CUDA-Based Defect Detection Algorithms - Labeling



The target pixel value = 2

0	0	0	0	0
0	1	0	9	0
0	9	9	9	0
0	2	0	2	0
0	0	0	0	0

if there is any pixel, which is on the target right side
in the same row with 10 pixel width and has a value of 2

0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	9	0	0	0	0	0	0	0	0
0	9	9	9	0	0	0	0	0	0	0	0
0	2	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

The target pixel's value is
changed from 1 to 9,
otherwise do nothing

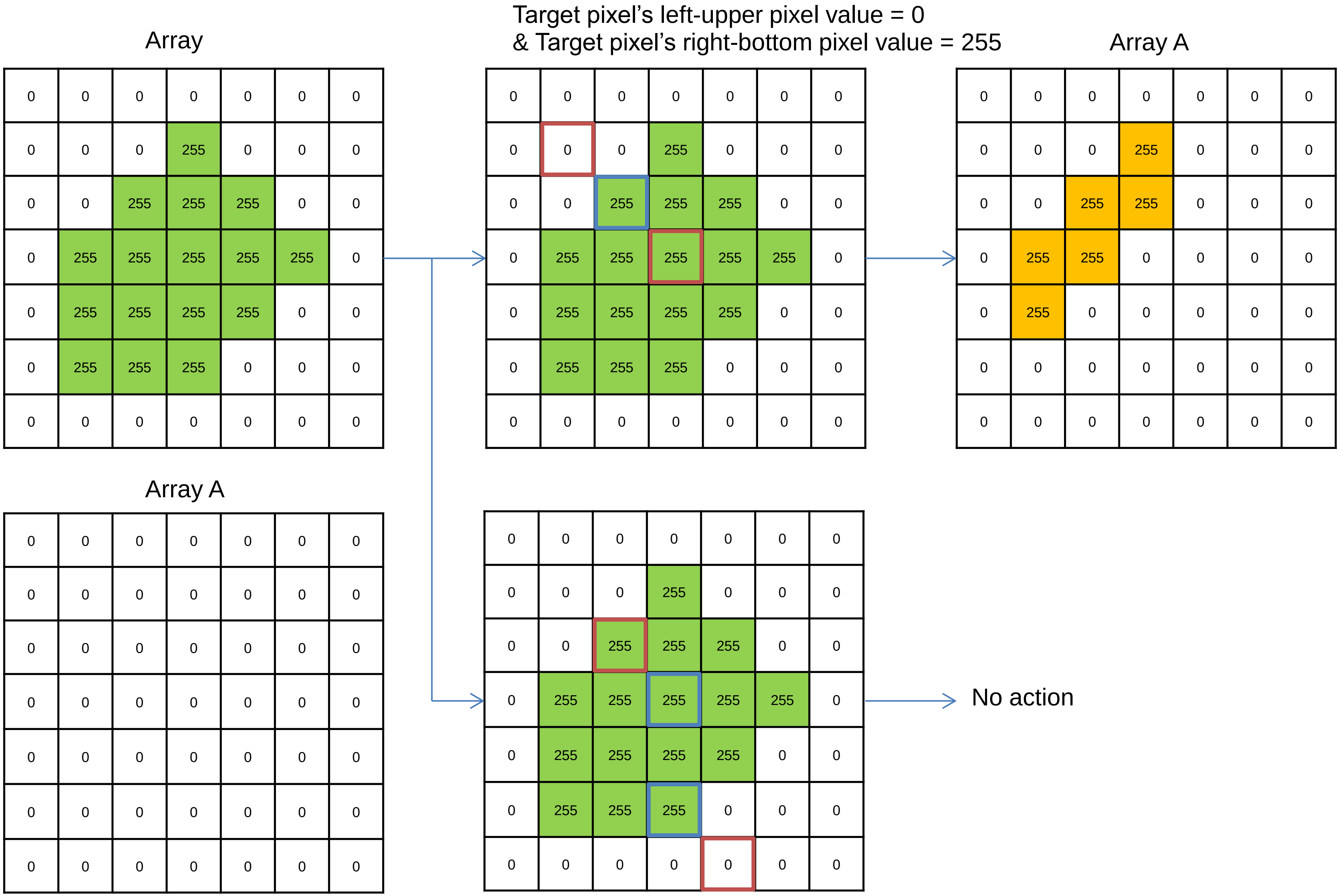
0	0	0	0	0
0	1	0	9	0
0	9	9	9	0
0	9	0	2	0
0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	9	0	0	0	0	0	0	0	0	0	0
0	9	9	9	0	0	0	0	0	0	0	0	0	0
0	2	0	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0
0	1	0	9	0
0	9	9	9	0
0	2	0	2	0
0	0	0	0	0

Elimination of Redundant End Point

CUDA-Based Defect Detection Algorithms – Edge Detection



Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Target pixel's left-upper pixel value = 0
& Target pixel's right-bottom pixel value = 255

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

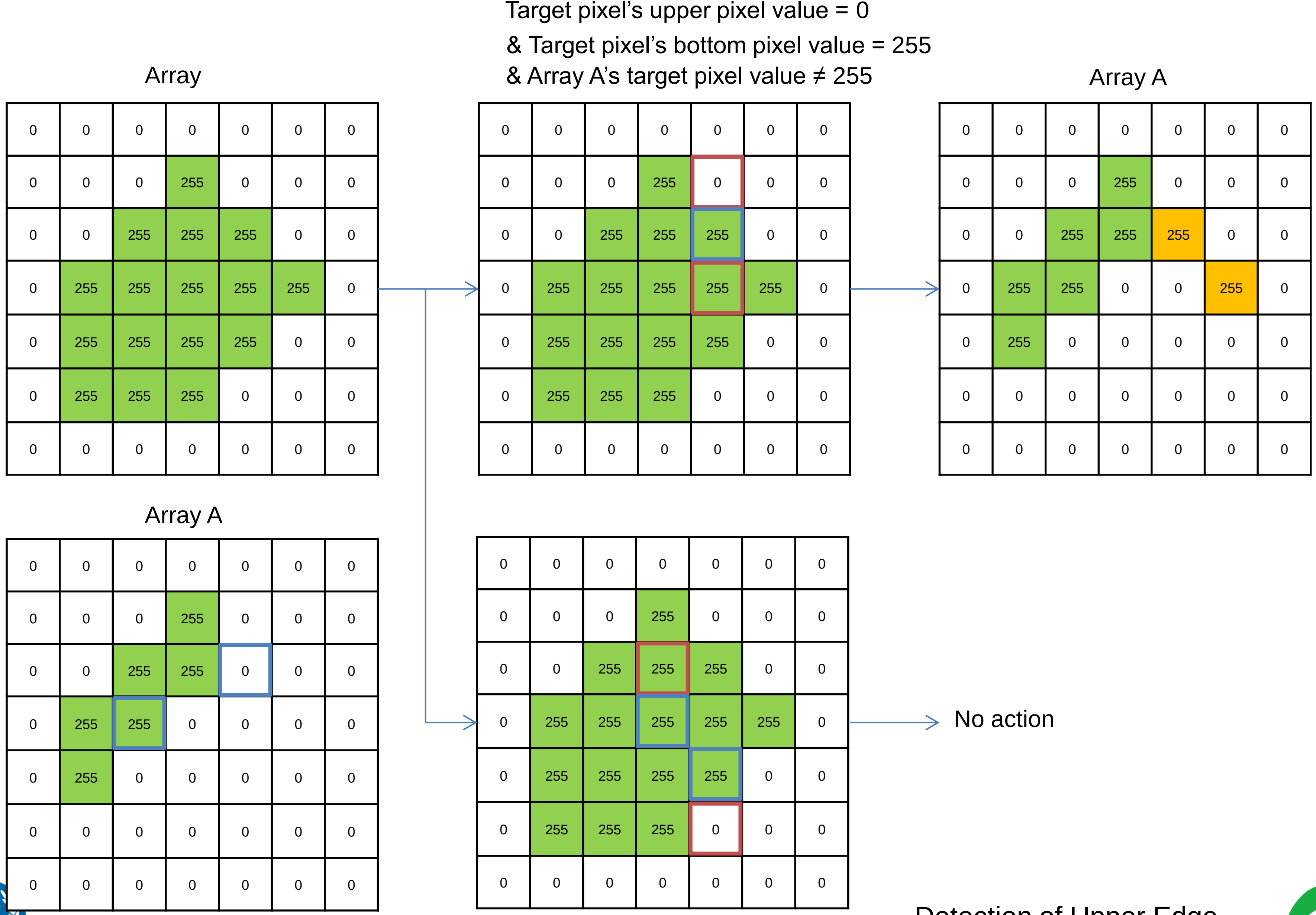
0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	0	0	0
0	255	255	0	0	0	0
0	255	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

No action

Detection of Upper-Left Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Target pixel’s upper pixel value = 0
& Target pixel’s bottom pixel value = 255
& Array A’s target pixel value ≠ 255

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	0	255	0
0	255	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Array A

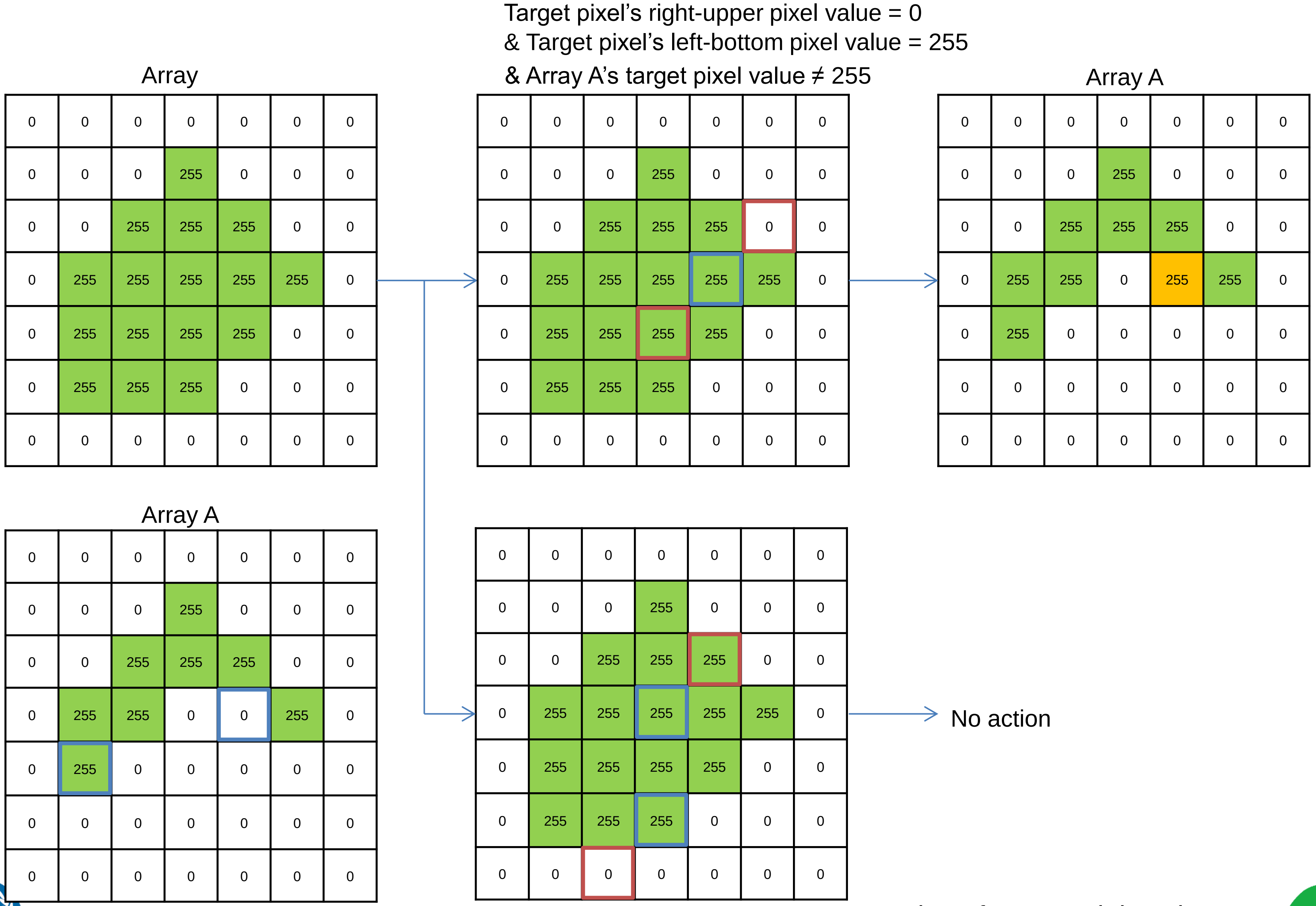
0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	0	0	0
0	255	255	0	0	0	0
0	255	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

No action

Detection of Upper Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Target pixel’s right-upper pixel value = 0
& Target pixel’s left-bottom pixel value = 255
& Array A’s target pixel value ≠ 255

Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Array A

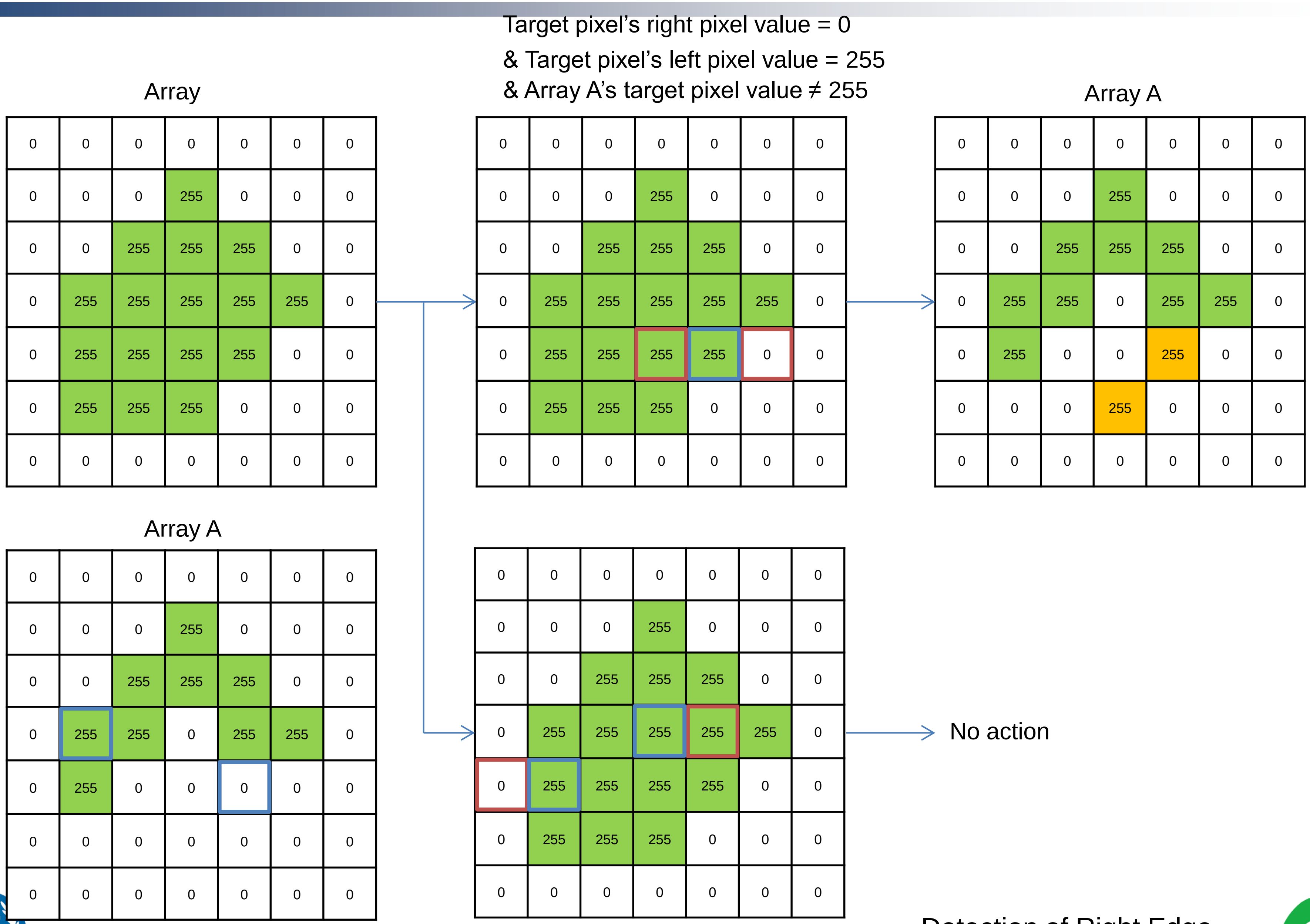
0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	0	255	0
0	255	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

No action

Detection of Upper-Right Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Target pixel's right pixel value = 0
& Target pixel's left pixel value = 255
& Array A's target pixel value ≠ 255

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	0	255	0	0
0	0	0	255	0	0	0
0	0	0	0	0	0	0

Array A

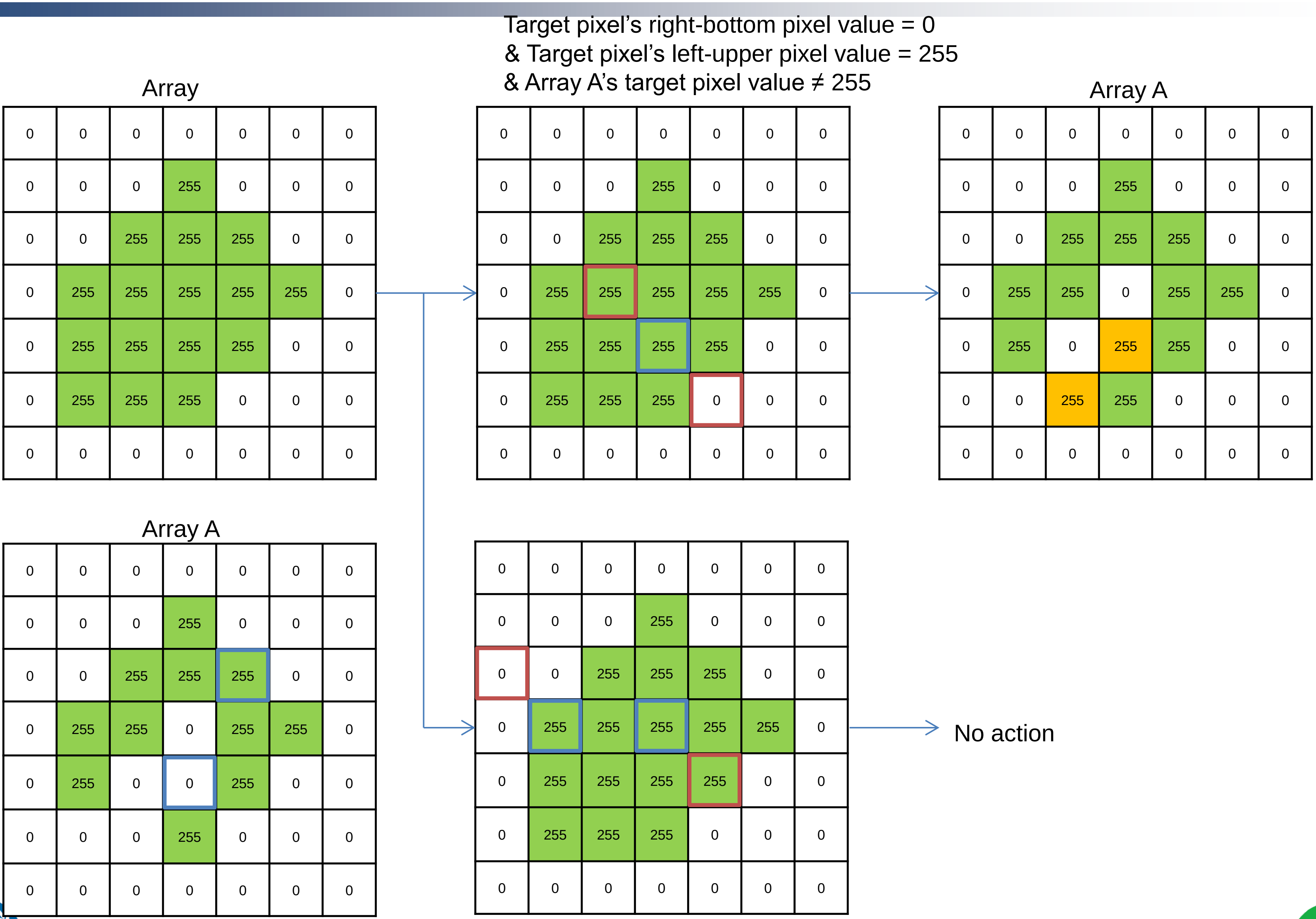
0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

No action

Detection of Right Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Target pixel’s right-bottom pixel value = 0
& Target pixel’s left-upper pixel value = 255
& Array A’s target pixel value ≠ 255

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	255	255	0	0
0	0	255	255	0	0	0
0	0	0	0	0	0	0

Array A

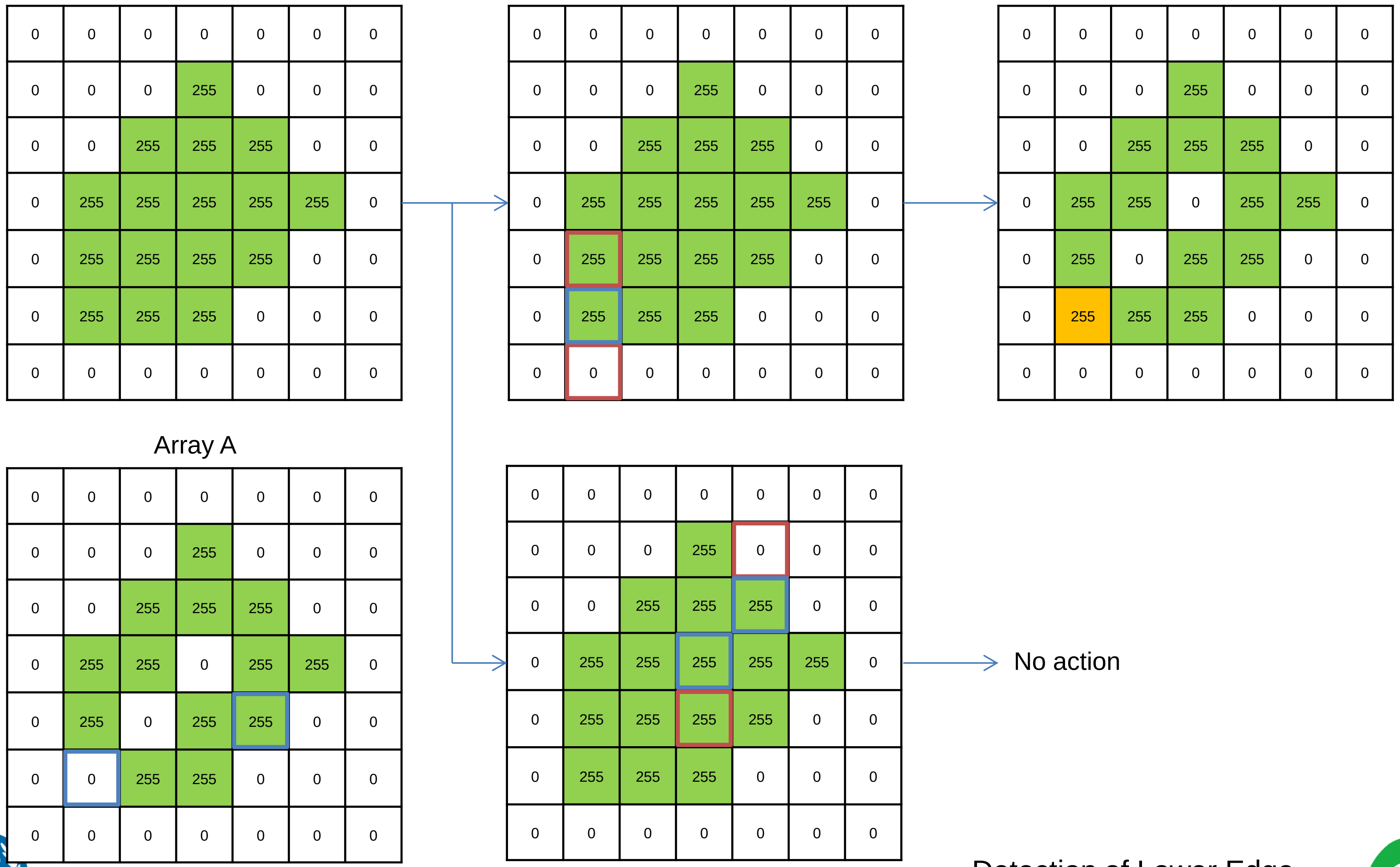
0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	0	255	0	0
0	0	0	255	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

No action

Detection of Lower-Right Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Target pixel’s bottom pixel value = 0
& Target pixel’s upper pixel value = 255
& Array A’s target pixel value ≠ 255

Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Array A

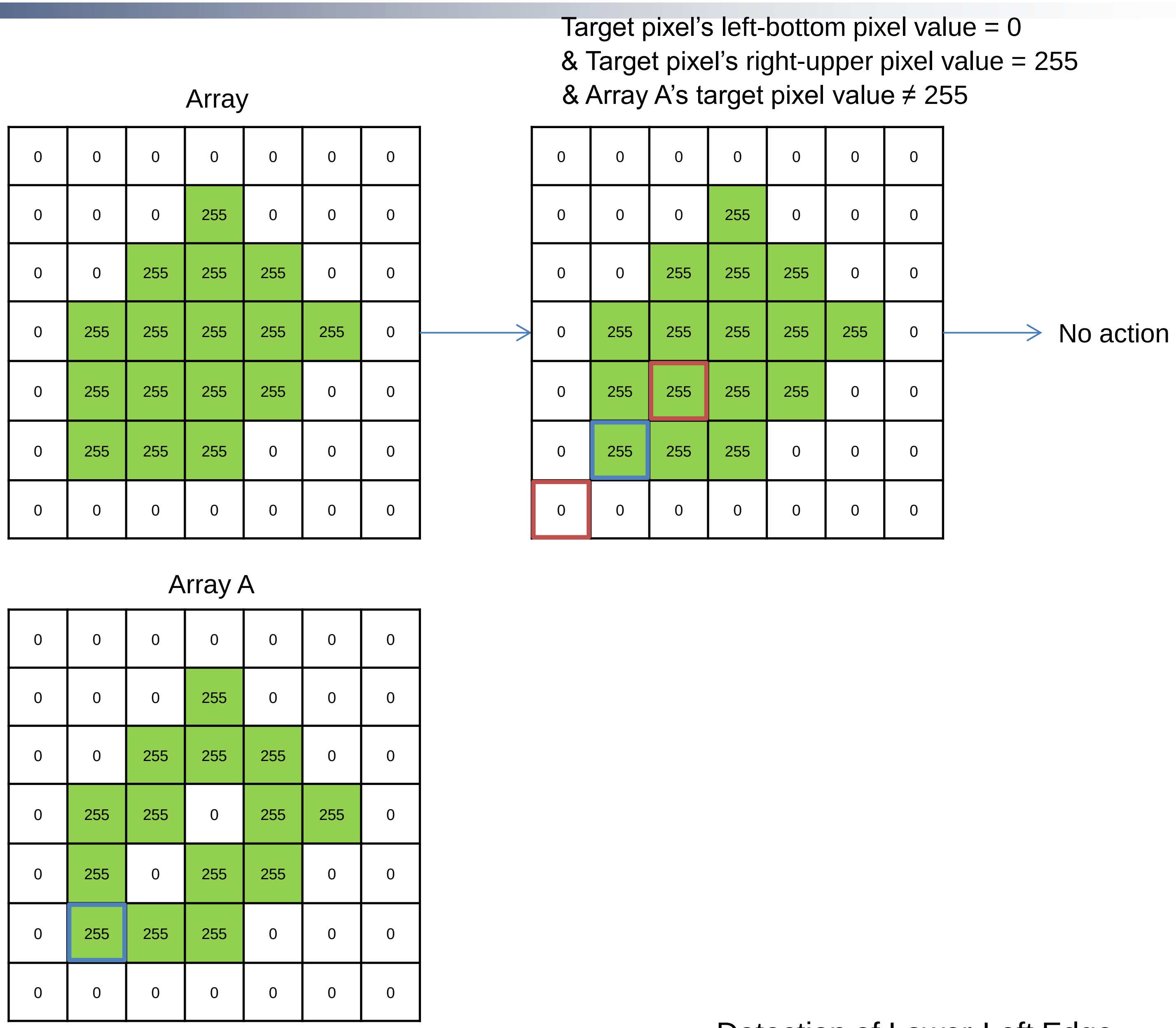
0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	255	255	0	0
0	0	255	255	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

No action

Detection of Lower Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Target pixel’s left-bottom pixel value = 0
& Target pixel’s right-upper pixel value = 255
& Array A’s target pixel value ≠ 255

Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

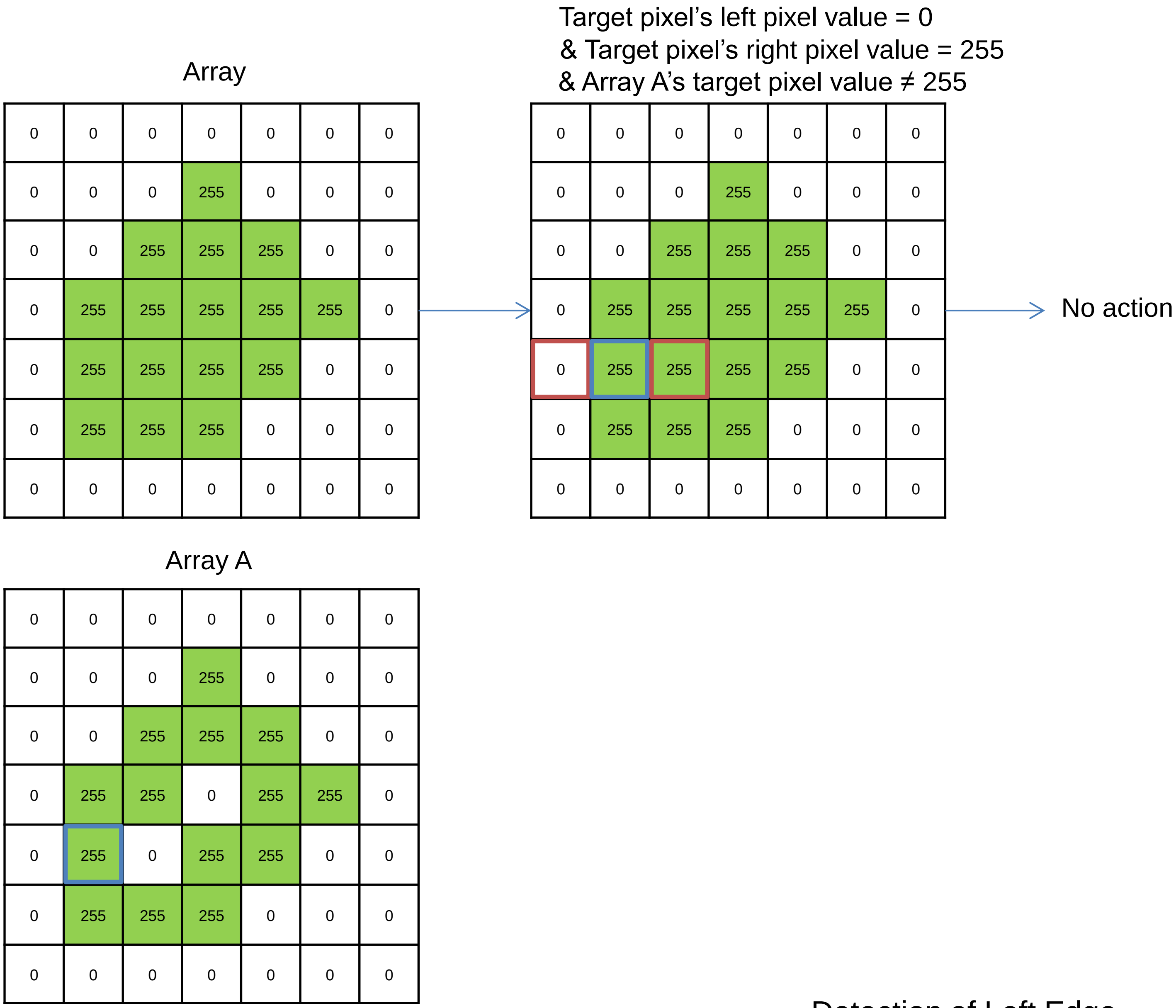
No action

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Detection of Lower-Left Edge

CUDA-Based Defect Detection Algorithms – Edge Detection



Array

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Target pixel’s left pixel value = 0
& Target pixel’s right pixel value = 255
& Array A’s target pixel value ≠ 255

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	255	255	255	0
0	255	255	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

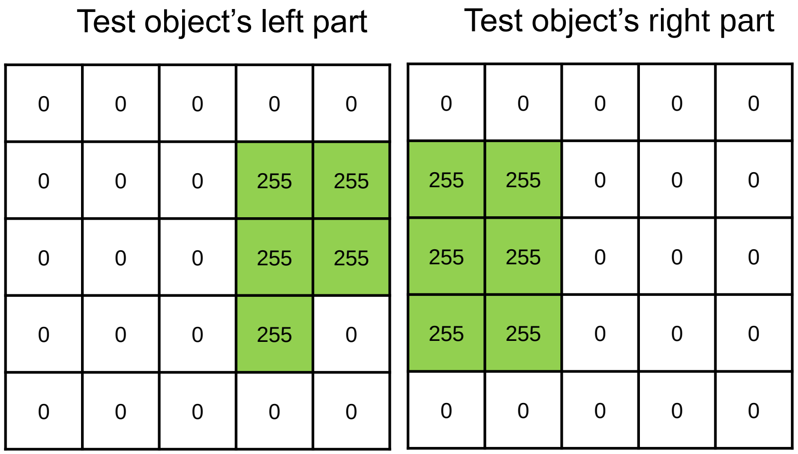
No action

Array A

0	0	0	0	0	0	0
0	0	0	255	0	0	0
0	0	255	255	255	0	0
0	255	255	0	255	255	0
0	255	0	255	255	0	0
0	255	255	255	0	0	0
0	0	0	0	0	0	0

Detection of Left Edge

CUDA-Based Defect Detection Algorithms - Redundancy Detection

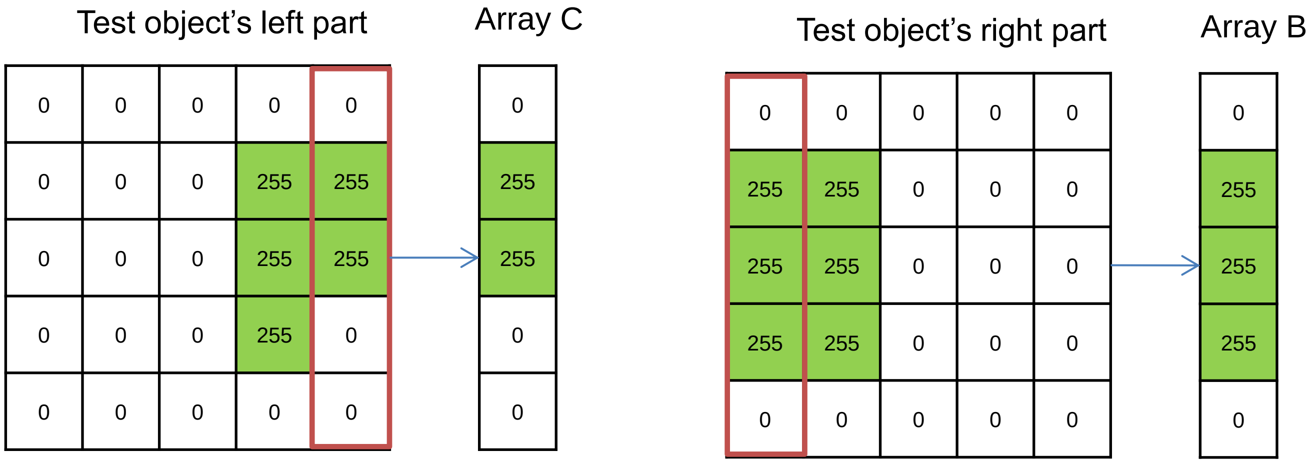


Test object's left part

0	0	0	0	0
0	0	0	255	255
0	0	0	255	255
0	0	0	255	0
0	0	0	0	0

Test object's right part

0	0	0	0	0
255	255	0	0	0
255	255	0	0	0
255	255	0	0	0
0	0	0	0	0



Test object's left part

0	0	0	0	0
0	0	0	255	255
0	0	0	255	255
0	0	0	255	0
0	0	0	0	0

Array C

0
255
255
0
0

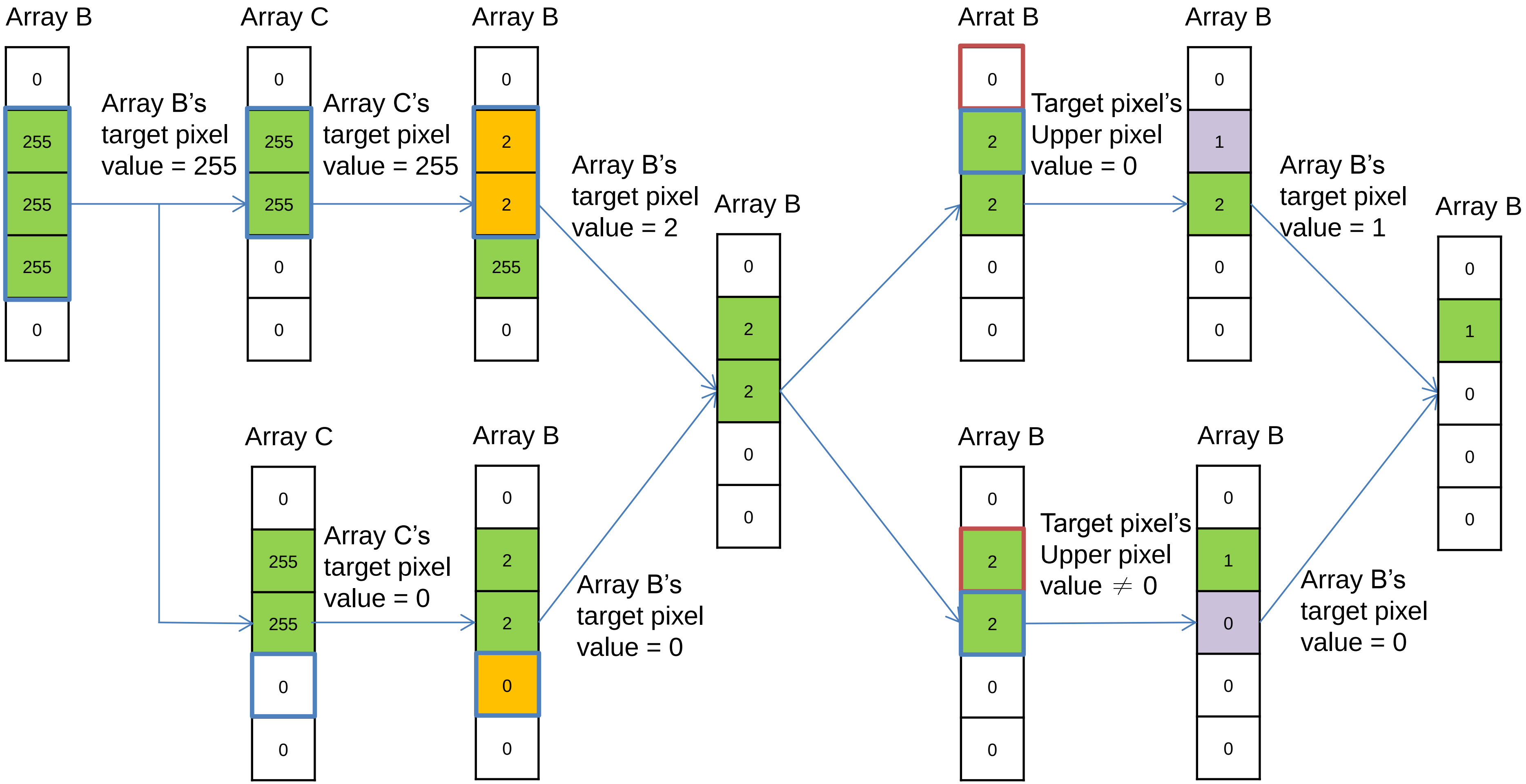
Test object's right part

0	0	0	0	0
255	255	0	0	0
255	255	0	0	0
255	255	0	0	0
0	0	0	0	0

Array B

0
255
255
255
0

CUDA-Based Defect Detection Algorithms - Redundancy Detection



Array B’s target pixel value = 255

Array C’s target pixel value = 255

Array B’s target pixel value = 2

Target pixel’s Upper pixel value = 0

Array B’s target pixel value = 1

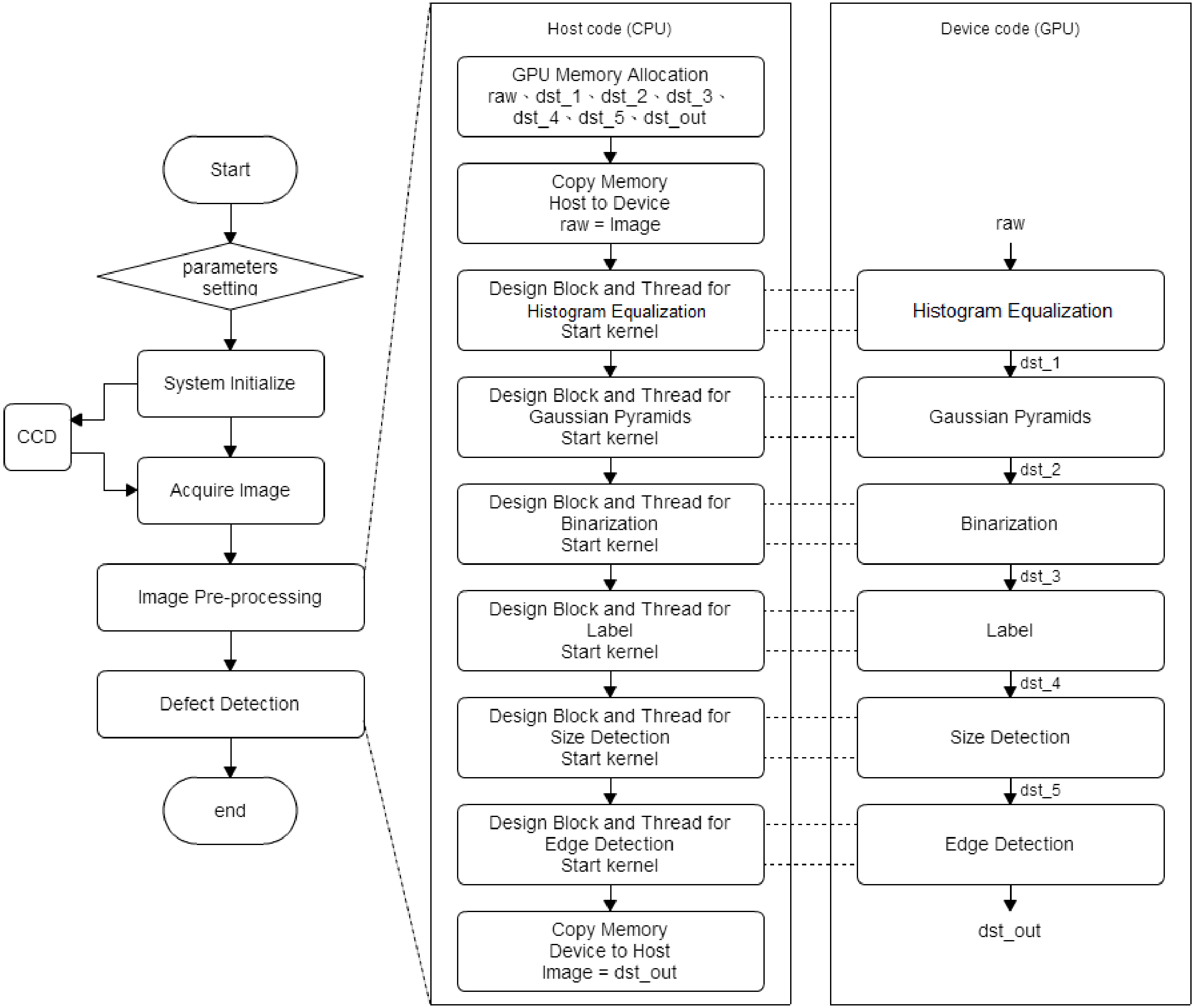
Array C’s target pixel value = 0

Array B’s target pixel value = 0

Target pixel’s Upper pixel value $\neq$ 0

Array B’s target pixel value = 0

Complete Process Flow of 2D Defect Inspection



Start

parameters setting

System Initialize

CCD

Acquire Image

Image Pre-processing

Defect Detection

end

Host code (CPU)

GPU Memory Allocation
raw、dst_1、dst_2、dst_3、
dst_4、dst_5、dst_out

Copy Memory
Host to Device
raw = Image

Design Block and Thread for
Histogram Equalization
Start kernel

Design Block and Thread for
Gaussian Pyramids
Start kernel

Design Block and Thread for
Binarization
Start kernel

Design Block and Thread for
Label
Start kernel

Design Block and Thread for
Size Detection
Start kernel

Design Block and Thread for
Edge Detection
Start kernel

Copy Memory
Device to Host
Image = dst_out

Device code (GPU)

raw

Histogram Equalization

dst_1

Gaussian Pyramids

dst_2

Binarization

dst_3

Label

dst_4

Size Detection

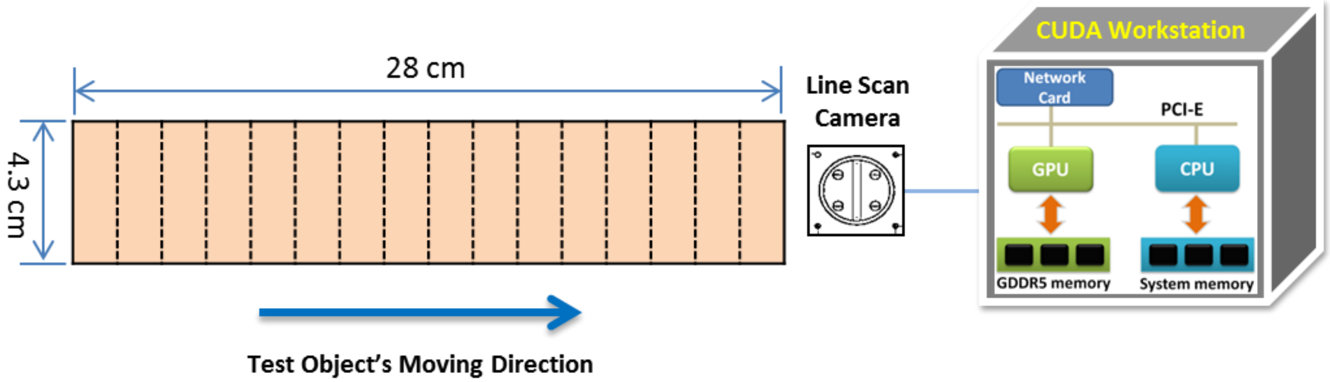
dst_5

Edge Detection

dst_out

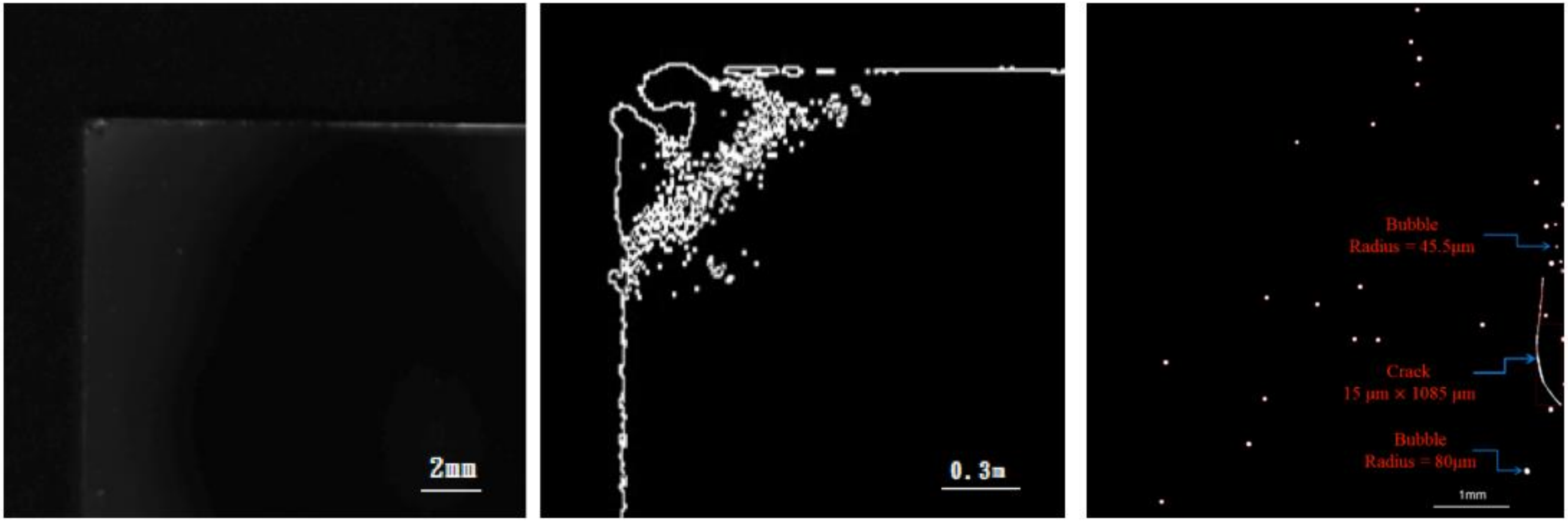
Experimental Results of DISCS

- The hardware configuration of the experiment includes a test object and two CUDA workstations, which controls a line scan camera.
- In the experiment, CUDA C program and MPI functions were used on the CUDA workstation.
- The test object’s width and length are 8.6 cm and 28 cm, respectively.
- The precision requirement specifies that each image pixel represents a 3.5 μm x 3.5 μm area.
- Each time window needs to be calculated within 250 ms.
- In each time window, the CUDA workstation needs to finish the defect detection in an image of 12288 x 5000 pixels.
- Totally 16 image strips are to be inspected within 4 s (70 mm/s).



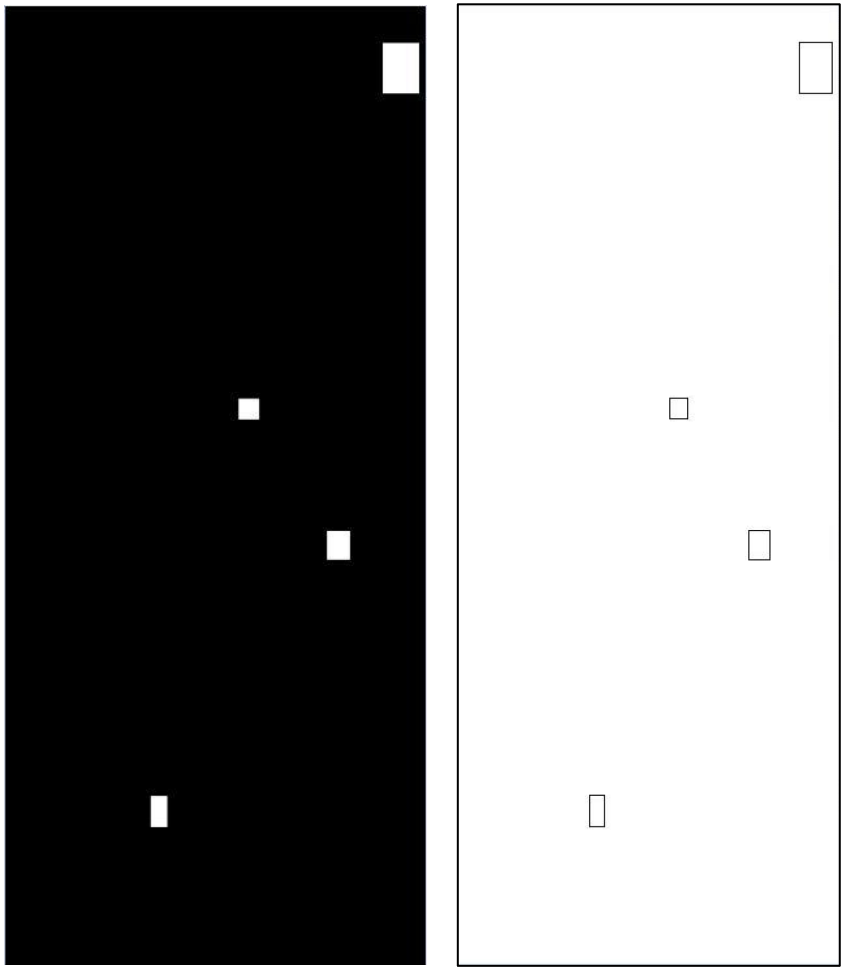
Experimental Results of DISCS

- Dark-field image of a part of one back-coated mirror piece



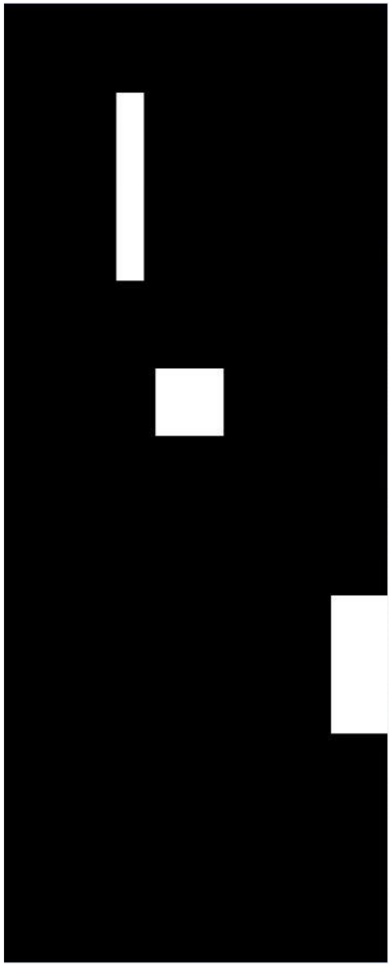
	Features	Average size	Detection Rate
Bubbles	Point-like or circular shape	50 to 160 μm	96 %
Cracks	Thin, elongated, located far from the perimeter of the test object	Width: 10 to 20 μm Length: 1 to 10 mm	94 %
Edge defects	Jagged lines, voids		99 %

Experimental Results of Simulated Defect Patterns (1)



Size Time	12288 × 5000 pixels
CPU to GPU(ms)	59.296
CUDA kernel function(ms)	47.3
GPU to CPU(ms)	57.39
Total time(ms)	163.986
Defect number	4

Experimental Results of Simulated Defect Patterns (2)

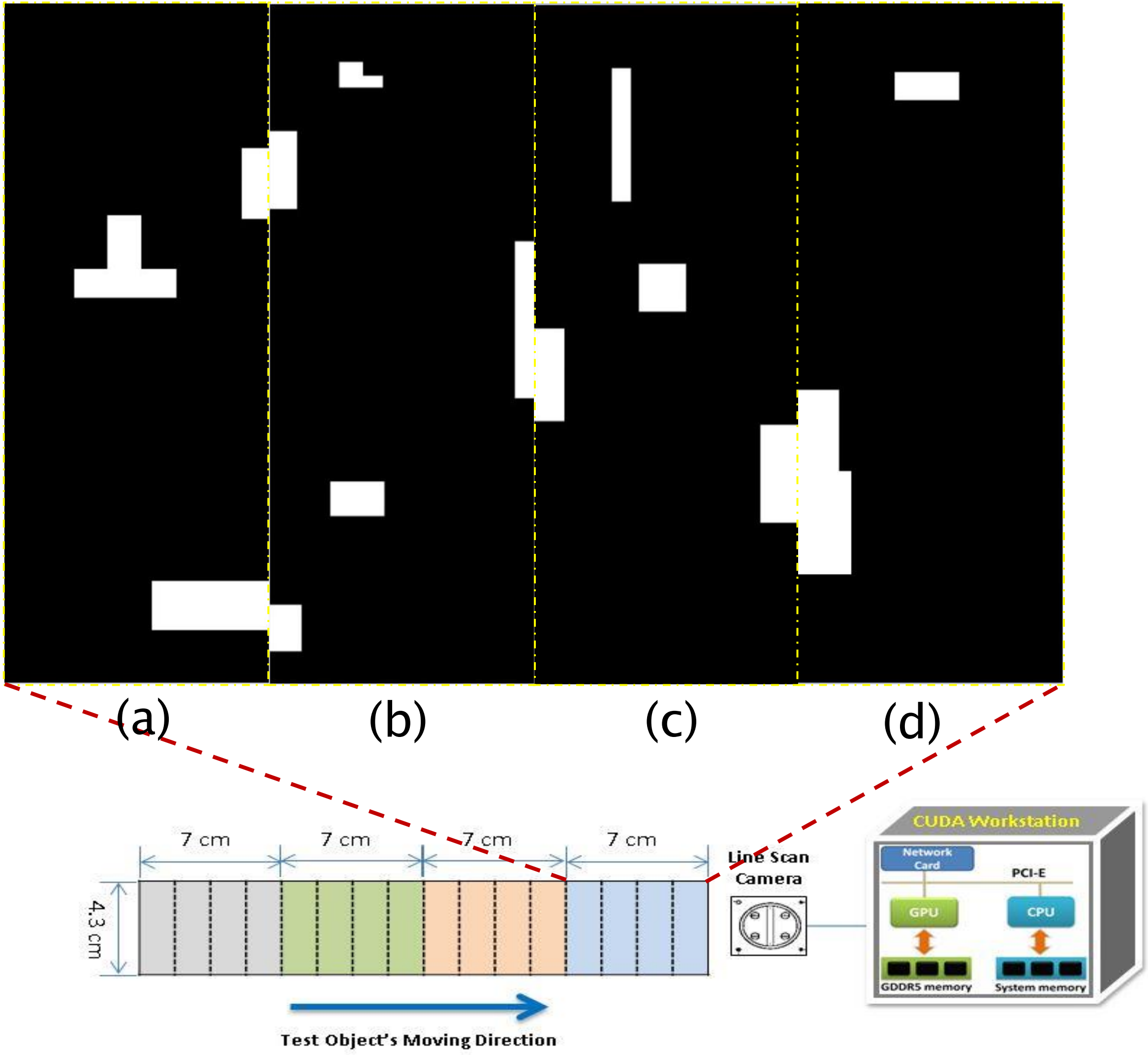


(Left)

(Right)

Size Time	12288 × 5000 pixels (left)	12288 × 5000 pixels (right)
CPU to GPU(ms)	61.0682	62.6171
CUDA kernel function(ms)	51.1604	51.2616
GPU to CPU(ms)	59.8082	59.6373
Redundant defect in CPU(ms)	0.1662	0
Total time(ms)	172.203	173.516
Defect number	3	2
New defect number	4	

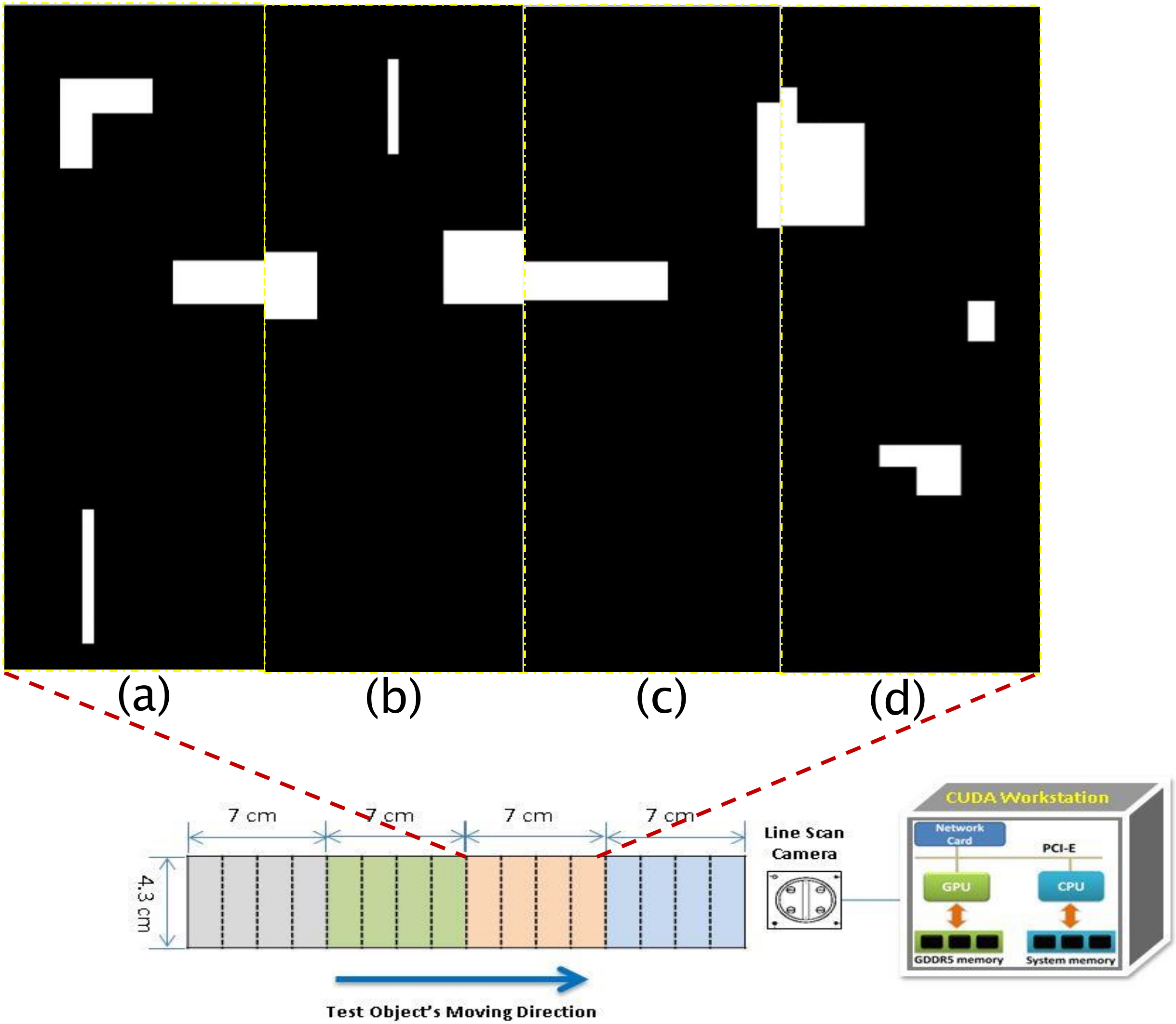
Experimental Results of Simulated Defect Patterns (3)



Experimental Results of Simulated Defect Patterns (3)

Time	(a)	(b)	(c)	(d)
CPU to GPU(ms)	60.9912	58.3117	62.0705	61.5419
CUDA kernel function(ms)	51.4951	51.9195	52.8371	51.2708
GPU to CPU(ms)	60.5924	59.701	59.1734	60.3451
Redundant defect in CPU(ms)	0.1703	0.1847	0.1791	0
Total time(ms)	173.249	170.117	174.26	173.158
Defect number	2	5	4	2
New defect number	9			

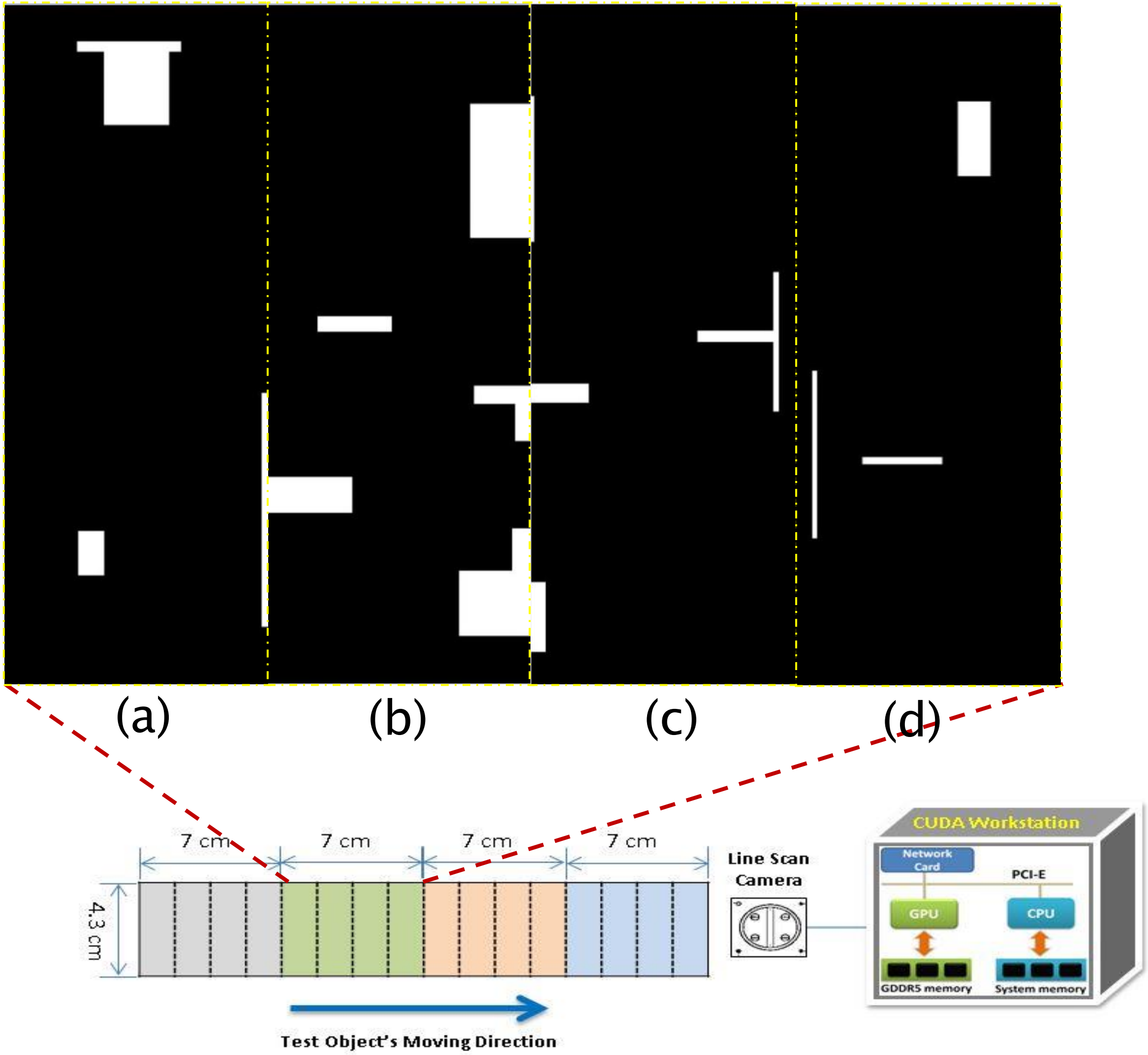
Experimental Results of Simulated Defect Patterns (4)



Experimental Results of Simulated Defect Patterns (4)

Time	(a)	(b)	(c)	(d)
CPU to GPU(ms)	61.0697	61.4254	61.9181	60.9409
CUDA kernel function(ms)	51.8225	50.9516	50.1037	52.3008
GPU to CPU(ms)	59.6589	58.9214	59.3474	59.5655
Redundant defect in CPU(ms)	0.03382	0.3402	0.1719	0
Total time(ms)	172.5849	171.6186	171.5411	172.8072
Defect number	3	3	1	3
New defect number	7			

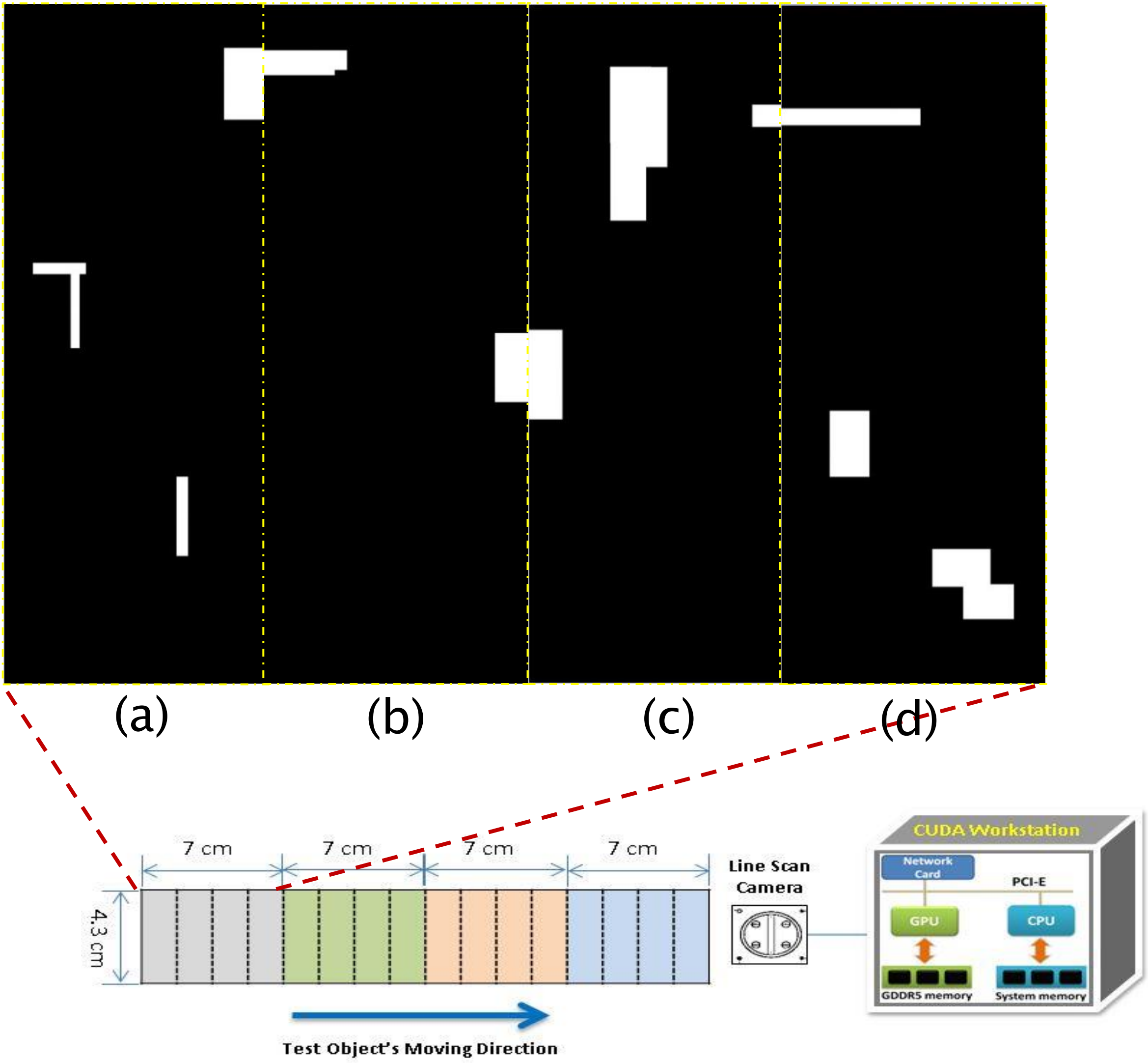
Experimental Results of Simulated Defect Patterns (5)



Experimental Results of Simulated Defect Patterns (5)

Time	(a)	(b)	(c)	(d)
CPU to GPU(ms)	57.8416	61.0743	61.8852	61.4629
CUDA kernel function(ms)	52.1556	55.1779	50.9634	49.6541
GPU to CPU(ms)	56.3132	59.8729	60.3045	61.6009
Redundant defect in CPU(ms)	0.178	0.1914	0.1601	0
Total time(ms)	166.4884	176.3165	173.3132	172.7179
Defect number	3	5	4	3
New defect number	11			

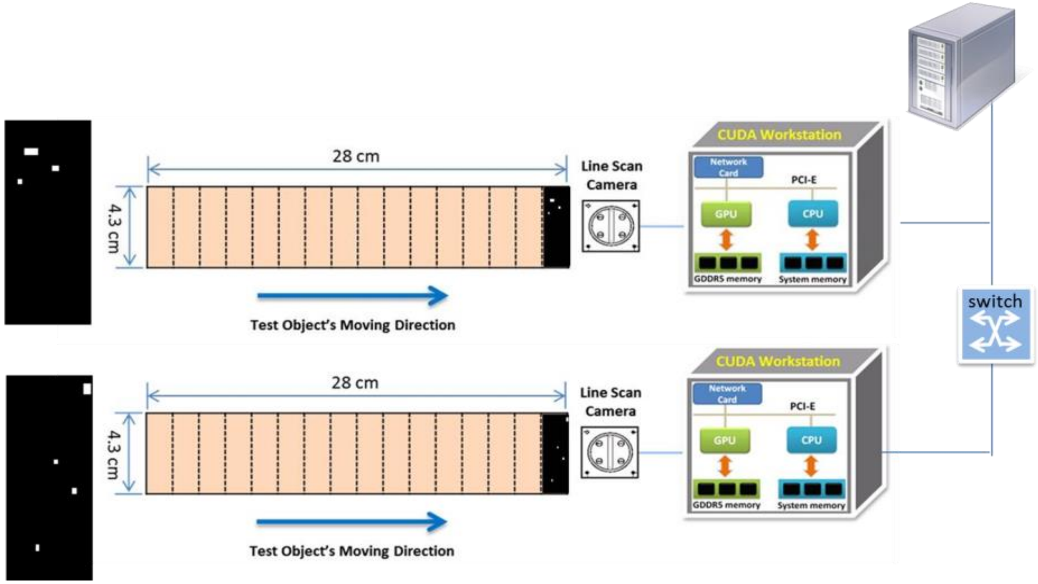
Experimental Results of Simulated Defect Patterns (6)

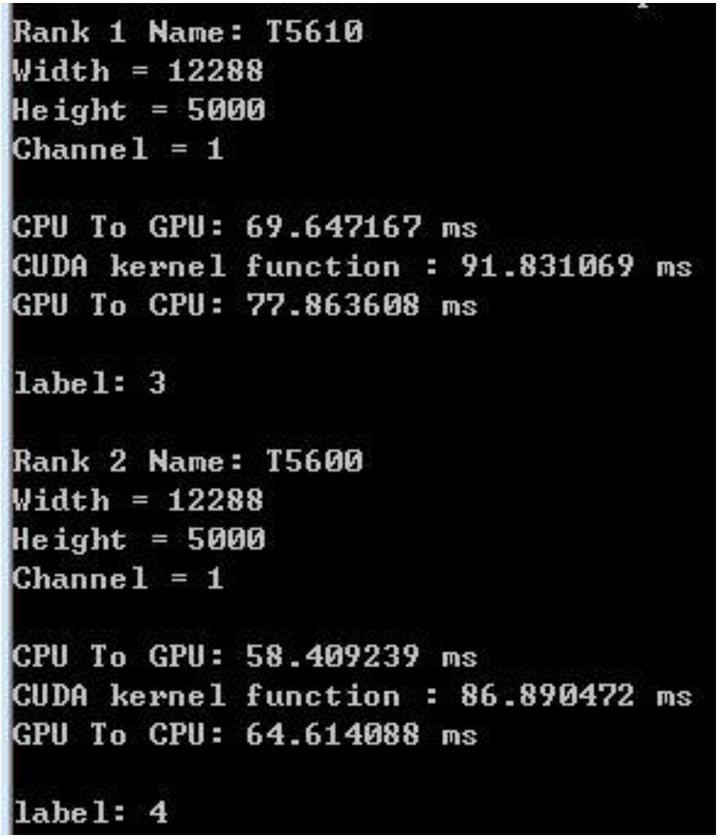


Experimental Results of Simulated Defect Patterns (6)

Time	(a)	(b)	(c)	(d)
CPU to GPU(ms)	62.457	62.1429	62.2435	62.3949
CUDA kernel function(ms)	49.9108	48.172	51.9457	50.0745
GPU to CPU(ms)	97.956	95.8898	85.225	79.3337
Redundant defect in CPU(ms)	0.1765	0.1652	0.1693	0
Total time(ms)	210.5	206.3699	199.5835	191.8031
Defect number	3	2	3	3
New defect number	8			

Experimental Results of Simulated Defect Patterns (7)





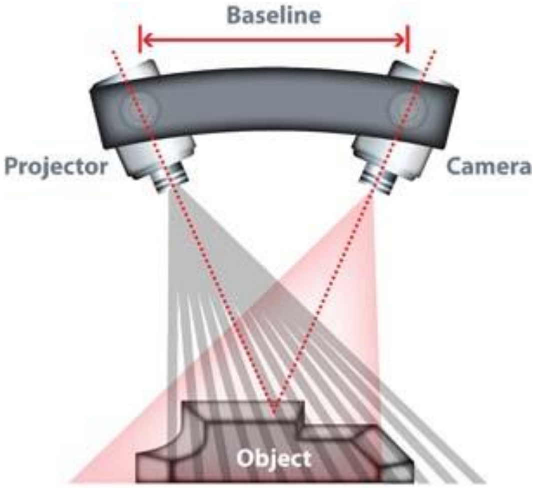
```
Rank 1 Name: T5610
Width = 12288
Height = 5000
Channel = 1

CPU To GPU: 69.647167 ms
CUDA kernel function : 91.831069 ms
GPU To CPU: 77.863608 ms

label: 3

Rank 2 Name: T5600
Width = 12288
Height = 5000
Channel = 1

CPU To GPU: 58.409239 ms
CUDA kernel function : 86.890472 ms
GPU To CPU: 64.614088 ms

label: 4
```

- Total amount of time is calculated as 239.3 milliseconds and 209.91 milliseconds, which is within the time window, 250 milliseconds.

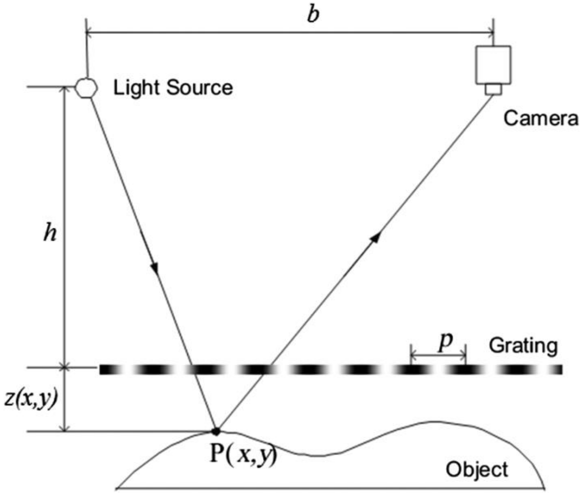
3D Inspection of Defect in Transparent Surface

Determination of 3D Profiles

- Projection fringe technique applies a straight-line grating onto an object to study the surface topography by recording the grating deformation due to topography variation.
- Shadow moiré topography positions the grating close to the object and the contour lines of the shadows at the object surface under the grating are observed.



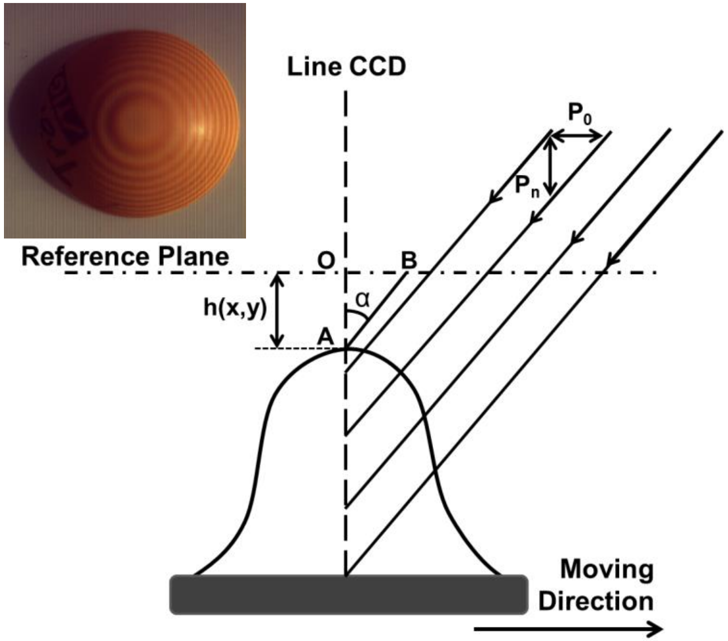
Projection fringes



Shadow moiré topography

Scanning Moiré Topography

- Scanning moiré technique, which is adapted from the traditional projection fringe technique, records contour images at the object surface using a linear CCD camera and a motorized transition stage.
- Similar to the conventional shadow moiré, the surface height distribution of the object is mathematically described as follows:



$$h(x,y)=\left[\frac{\phi(x,y)}{2\pi}\right]P_0\cot\alpha=N(x,y)\,P_n$$

h(x, y) : The object height at an arbitrary point (x, y) relative to the virtual reference plane

ϕ(x, y) : The phase difference distribution between object surface and reference plane for each point (x, y)

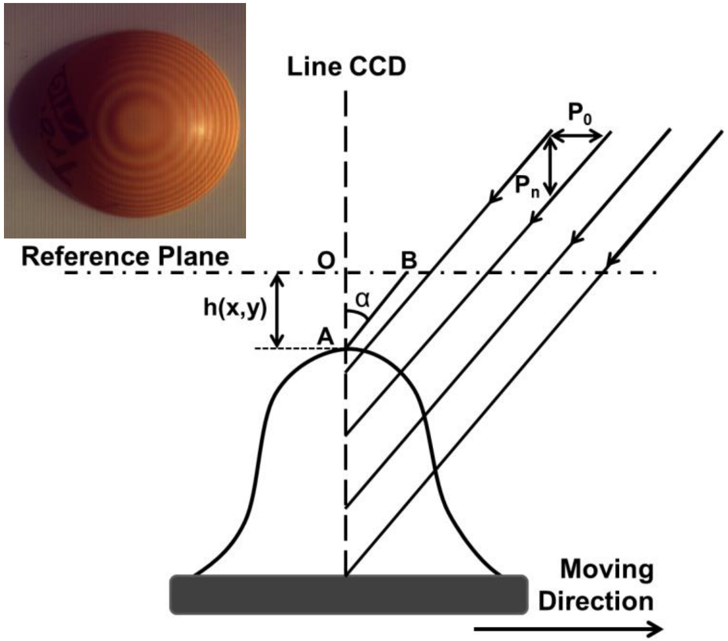
P_o & P_n : Respectively the grating pitch in the direction parallel and perpendicular to the reference plane

α : The grating projection angle inclined to the optical axis of the CCD

N(x, y) : The fringe order of the surface contour at each point (x, y)

Scanning Moiré Topography

- Scanning moiré technique, which is adapted from the traditional projection fringe technique, records contour images at the object surface using a linear CCD camera and a motorized transition stage.
- Similar to the conventional shadow moiré, the surface height distribution of the object is mathematically described as follows:



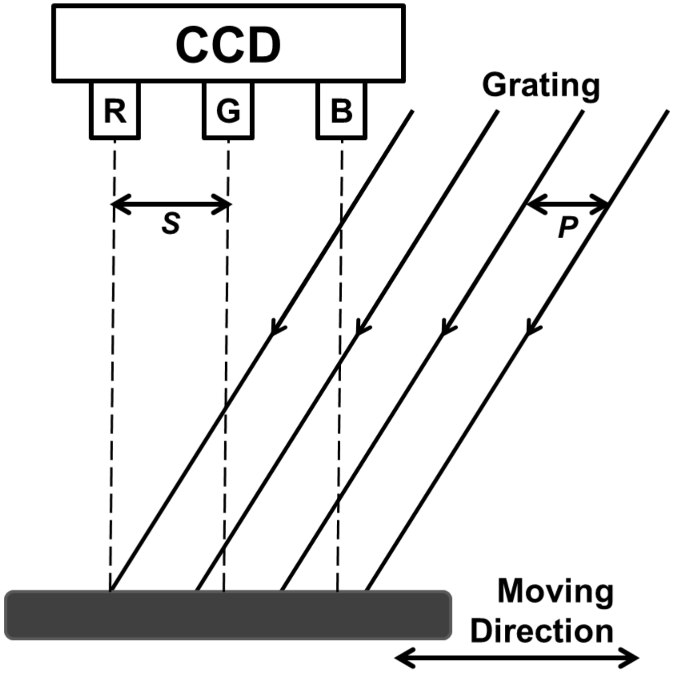
$$h(x,y)=\left[\frac{f(x,y)}{2\pi}\right]P_0\cot\alpha=N(x,y)P_n$$

- **The surface contour is directly related :**

1. The phase distribution of the interferogram
2. The measurement sensitivity of the object height is dependent on the projected grating pitch
3. Grating incidence angle

The fringe order at any point is determined by the phase at that point so that the surface topography of the specimen can be extracted from its phase distribution. With an appropriate phase measuring technique, contour maps can be used to generate surface topography quantitatively

Phase Measuring Technique



A continuous phase shift equivalent to 120^0 exists for the three sets of interferogram. For the red, green, and blue contour fringes, these correspond to 0^0, $+120^0$ and $+240^0$, respectively.

- Since the sets of contour maps will be obtained separately from the RGB channels, the inherent phase shift between each two of the three intergerograms provides sufficient information for Phase Shifting Interferometry (PSI) on the fringe pattern.

$$P = \frac{3S}{1+3n}, \quad n = 0,1,2\ldots$$

P : The grating image on the CCD ***S*** : The pitches of the three RGB lines

- Three frames of intensity data were simultaneously recorded with a 120º phase change between any two adjacent readouts and are presented by

$$I_1(x,y) = I_A + I_B \cos[\phi(x,y)]$$

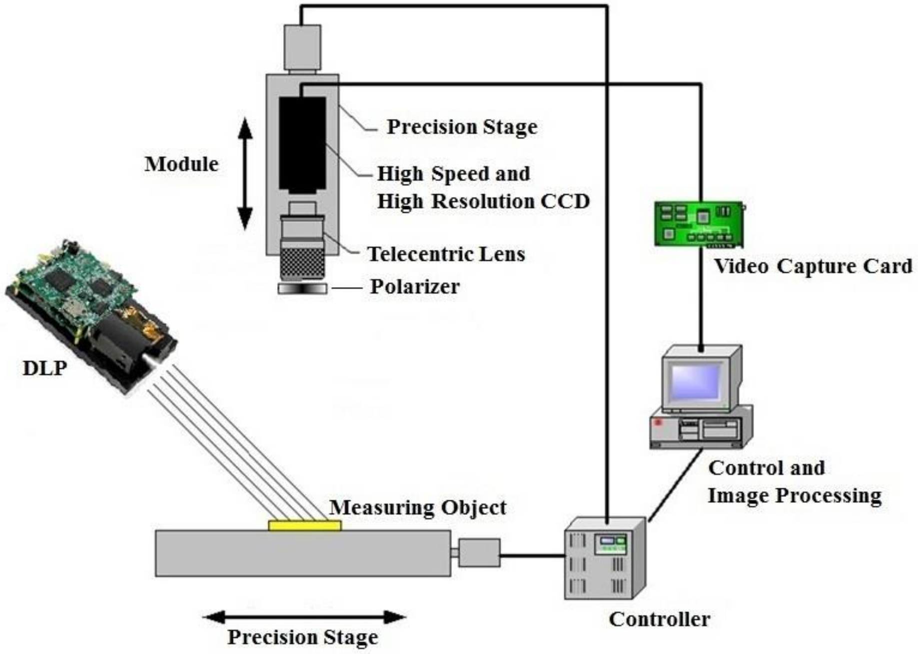
$$I_2(x,y) = I_A + I_B \cos[\phi(x,y) + 2n\pi + 120^0]$$

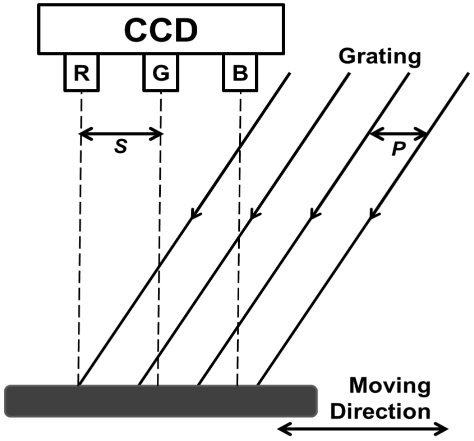
$$I_3(x,y) = I_A + I_B \cos[\phi(x,y) + 4n\pi + 240^0]$$

- The phase distribution of the contour map is obtained:

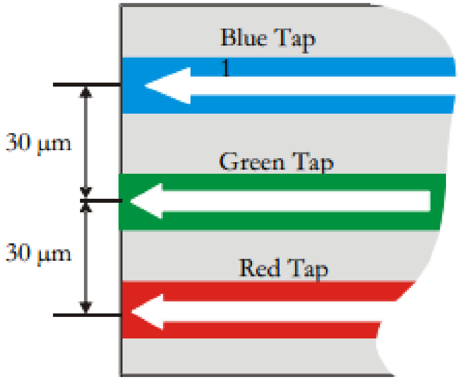
$$f(x,y) = \tan^{-1}\frac{\sqrt{3}(I_1 - I_3)}{\left[2I_2 - (I_1 + I_3)\right]}$$

Scanning Projection Fringe System





$$P = \frac{3S}{1+3n}$$



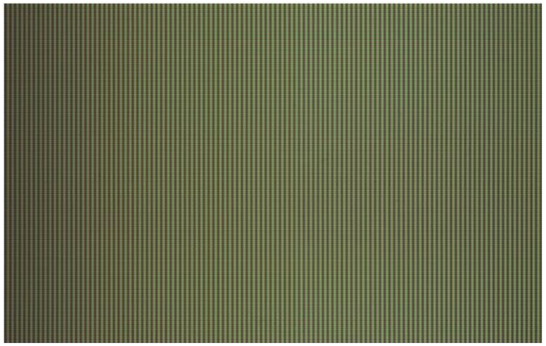
120^0 phase shift between each color channel , the projected pitch of the grating on the CCD should be **22.5 µm for n = 1**

System Specifications	
Resolution	4096 pixels
CCD Pixel Size	10 µm x 10 µm
The current magnification	0.5x

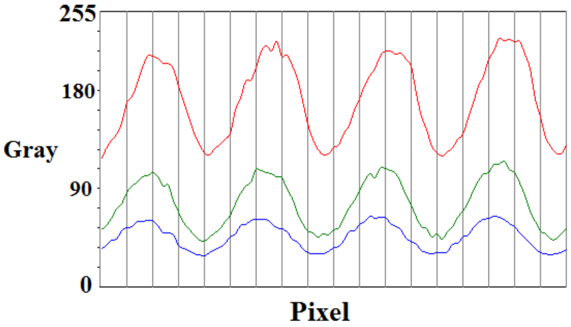
The projected pitch of the grating on the CCD : **45 µm**

RGB Calibration

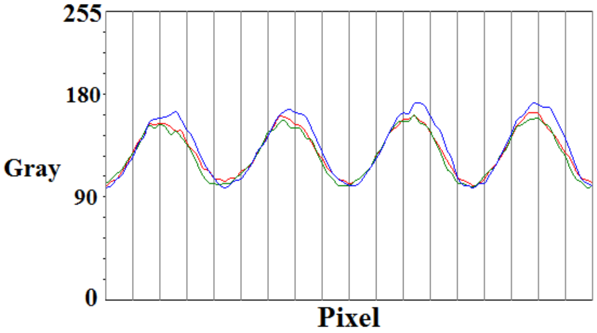
- Each channel has different photosensitivity.
- The RGB inputs to the CCD were calibrated by adjusting the output RGB lines of the DLP.



The projected grating image on a white screen from the DLP illumination

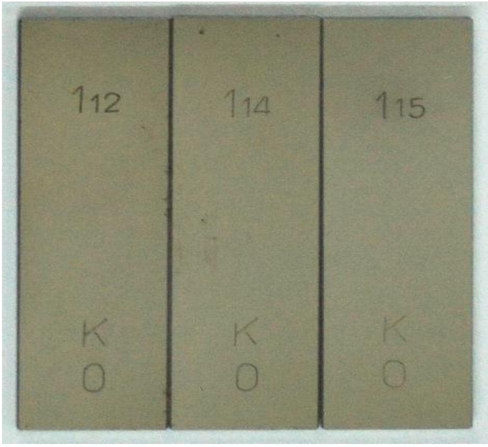


The initial intensity levels of the RGB channels in a color line CCD



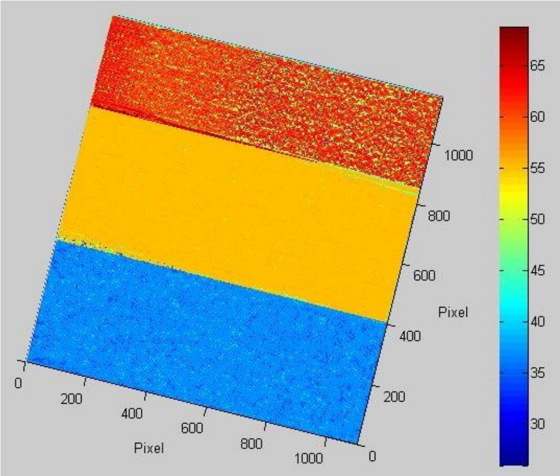
The adjusted intensity levels of the RGB channels

Evaluation of System Performance



Steep surfaces combined with gauge block (heights of 1.12 mm, 1.14 mm, and 1.15 mm)

- Sample Area：10mm x 25mm
- Measurements were repeated 10 times and the measurement repeatability were respectively 0.34 µm and 0.24 µm.

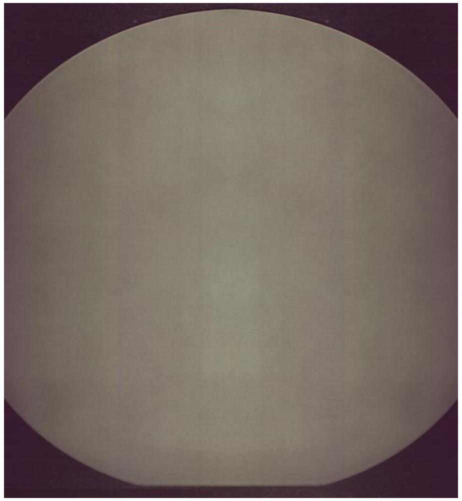


Step Height	10 µm	20 µm
The measured average step heights	10.54 µm	20.78 µm

Sub-micrometer measuring accuracy and high repeatability have been achieved !

* The primary limitation arises from the camera occlusion or shadow caused by steep profile just like the traditional projection fringe method.

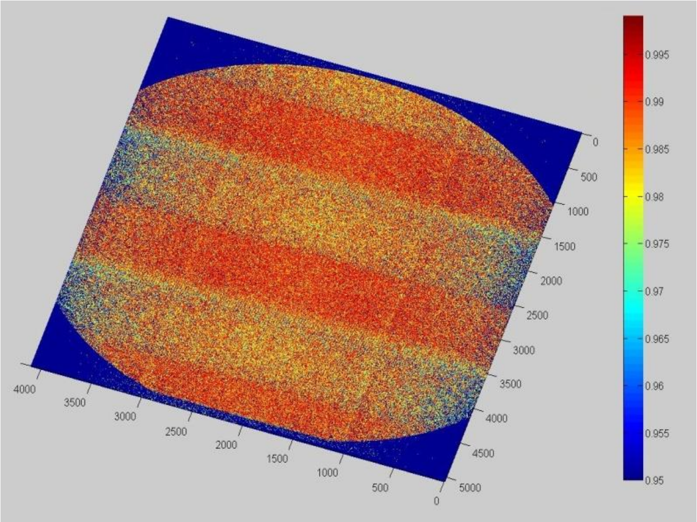
3D Profile: Sapphire Substrate



- 4-inch diameter
- The flatness of the substrate based on the minimum zone evaluation of surfaces is 2.33 μm.
- By comparison with 3.25 μm peak to valley value obtained from a white light interferometer at 1 nm resolution, the measurement uncertainty was found to be roughly 1 μm.

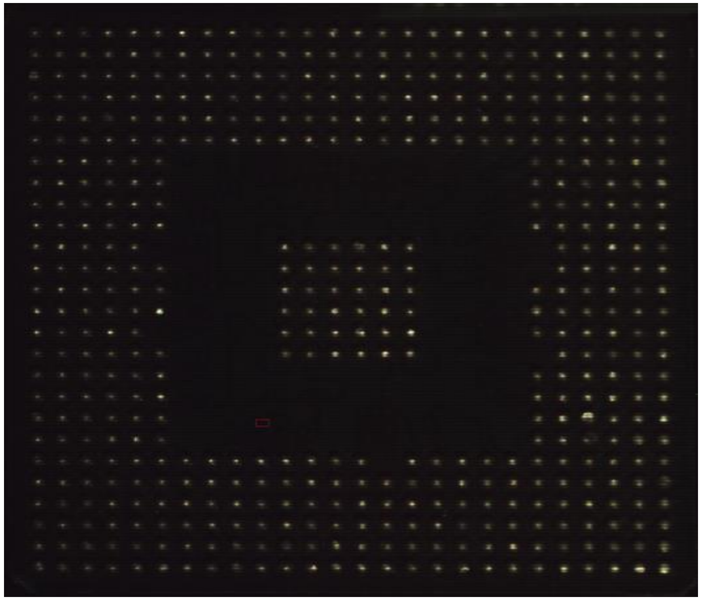
The standard deviation after five trials is **~0.25 μm** which shows possible accuracy in micrometer order and high precision in the **sub-micrometer scale**.

It can be controlled to speeds of up to 100 mm/s thus making it possible to inspect a **4-inch** substrate to within a **test duration of 1 second**.

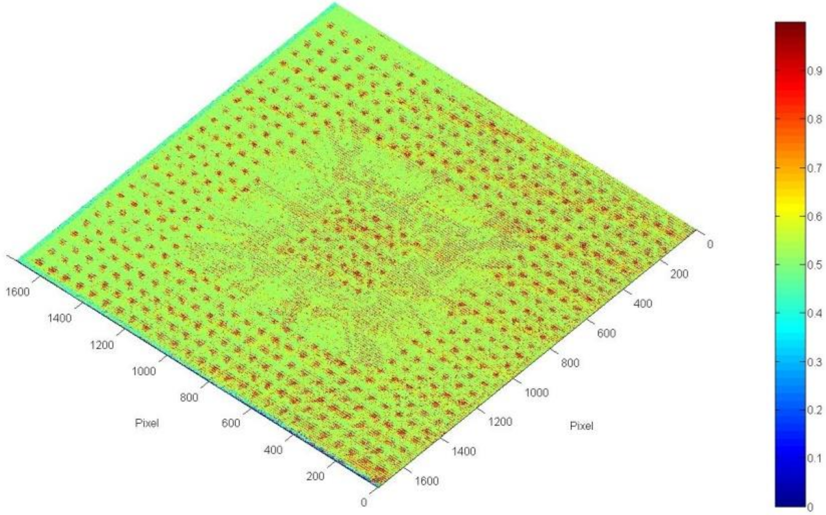


Reconstructed surface profile after PSI algorithm

Co-planarity of a BGA Substrate



Wide-field image of a BGA substrate



3D surface profile of the BGA substrate

- Sample Area：35 mm x 35 mm
- 1750 line images

- The measured results of co-planarity of **3.4 μm** with a measurement repeatability of **0.32 μm** were obtained.

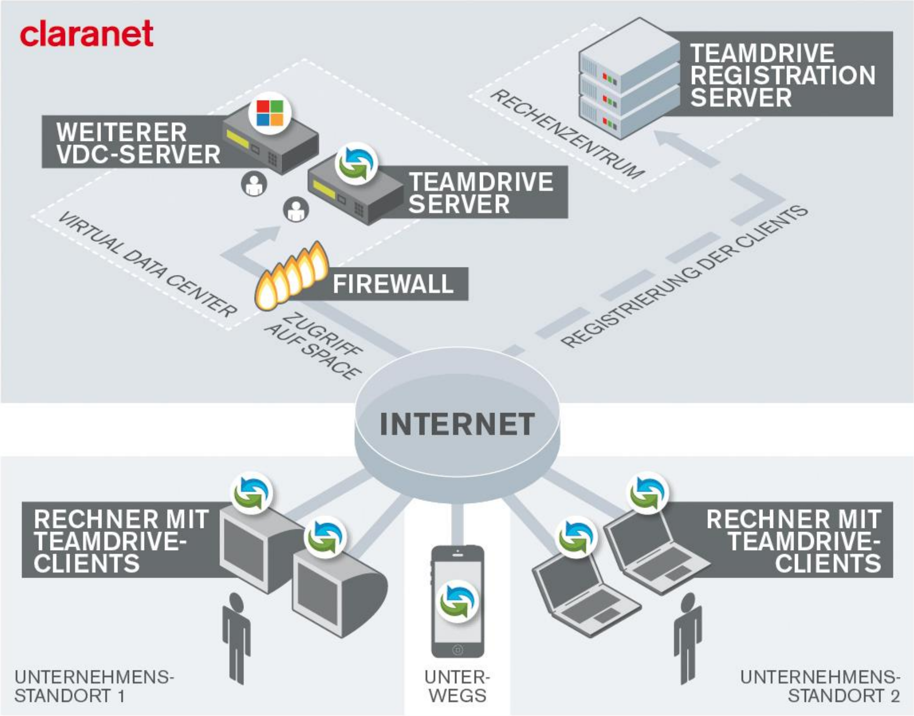
Cloud-Based Analysis System

Why Cloud?

- Collaborations via Cloud

Reference: activeco.com

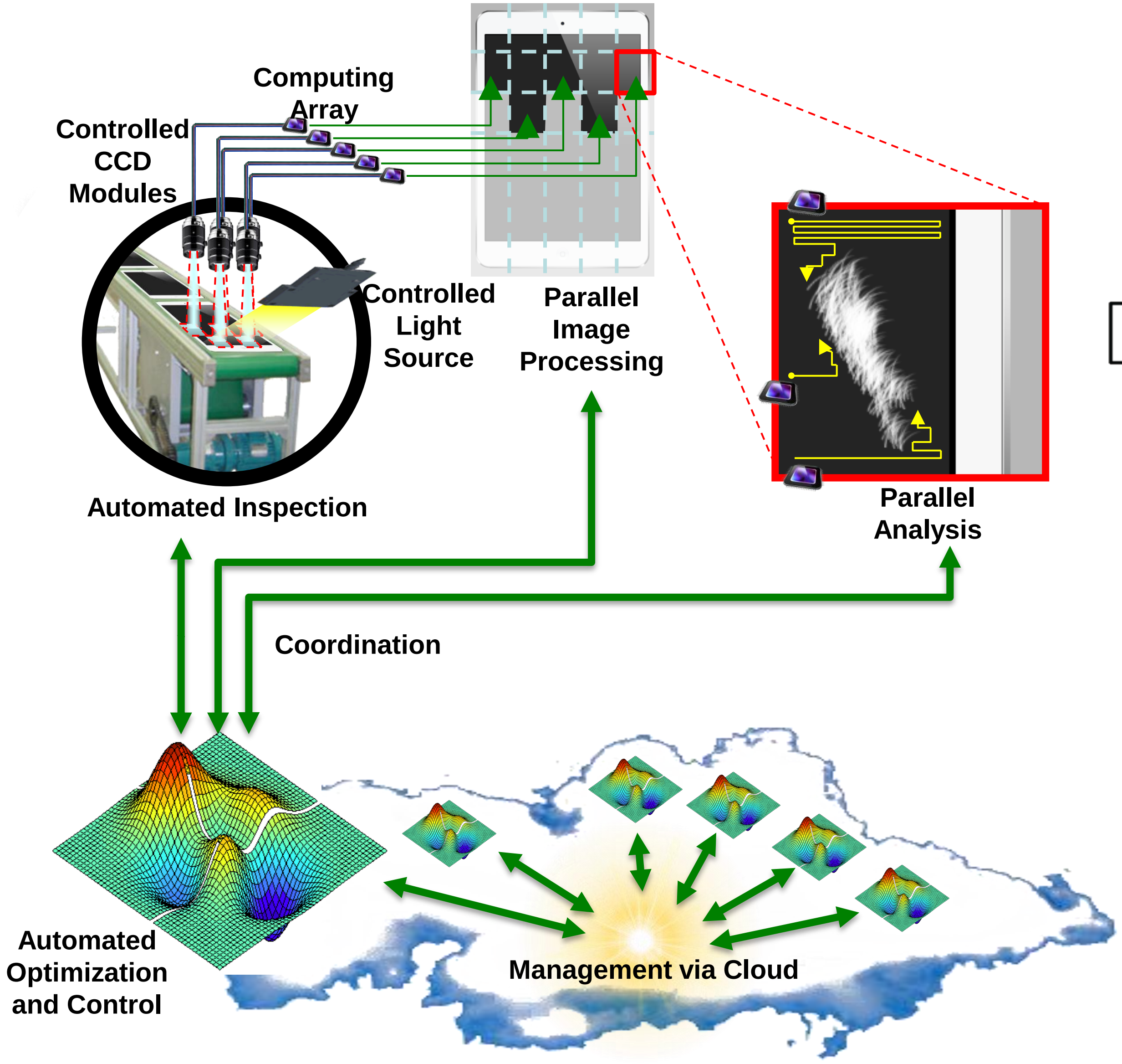
Reference: accellian.com



Reference: claranet.de

- Safely synchronize design activities via Cloud

Next-Generation Collaborations via Cloud



Computing Array

Controlled CCD Modules

Controlled Light Source

Parallel Image Processing

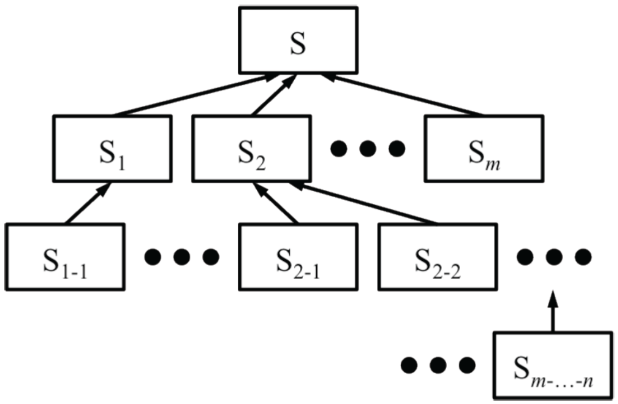
Parallel Analysis

Automated Inspection

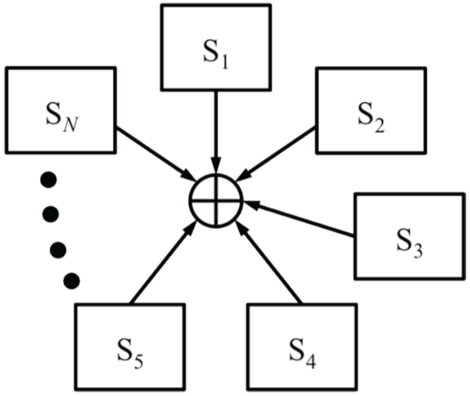
Coordination

Automated Optimization and Control

Management via Cloud



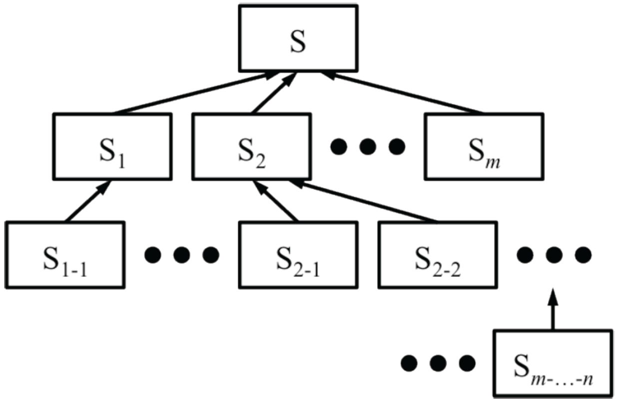
Hierarchical structure



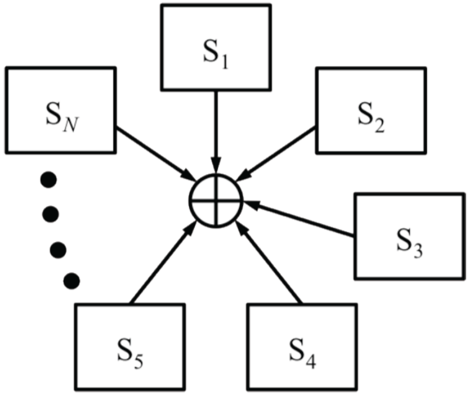
Distributed structure

Comparison of Various Collaboration Models

- A multidisciplinary design optimization problem has been solved by same amount of computing nodes on Cloud but using two different collaboration models.
- Hierarchical model
    - 10 iterations
    - 432 function evaluations
    - 30 units of working time
- Distributed model
    - 14 iterations
    - 168 function evaluations
    - 42 units of working time



Hierarchical structure



Distributed structure

Conclusion and Discussion

- Numerous methods have been developed for surface defect detection; however, little has addressed a situation where both speed and precision requirements are satisfied simultaneously.
- In our research, the requested inspection requirement is measurement area of 28 cm x 23 cm within 4 seconds with the resolution of 3.5 μm x 3.5 μm.
- An expandable Distributed Image Sensor Computing System (DISCS) has been developed to achieve in-line surface defect detection.
    - The hardware architecture consists of independent machines that form a master-slave parallel computing model
    - The software architecture consists of MPI processes that run in CPUs and CUDA threads that run in GPUs.

Conclusion and Discussion (Continued)

- Measurement of 3D profile topography has been developed using moiré techniques.
  - Straight-line grating was projected on the object surface using digital light processing (DLP) illumination.
  - Tri-linear colored CCD grabbed the successive line images with 120° phase difference between each intergerogram.
- The measurement range of the proposed grating projection module and image capture module is flexible from few millimeters to hundreds of millimeters.
- The measurement speed up to 100 mm/s is possible.
- Automated optical inspection of moving substrates on a motorized transition stage has been demonstrated and is suitable for in-line or in-process inspection of conveyed products.
- The proposed method is a very good choice for non-contact profilometry because the inspection process can be handled remotely using simple instruments operating at high speed, yet providing good accuracy, high resolution, and insensitivity to environmental noise.