

Object Recognition by Selective Algorithm with CUDA Implementation

Sang-hyeob Song, Jun-dong Cho

Sungkyunkwan Univ.

***Department of Mobile Communications and Power
Electronics.***

VADA Lab.

Contents

- **Introduction**
- **Related Work**
- **Proposed Method**
- **Experimental Result**
- **Conclusion**

Contents

- **Introduction**
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Introduction

- **In factory automation, robot arms replace human-manual assembly.**
- **To pick the specific object, we should know the location of the object.**
- **To know locations of objects, pattern recognition is essential.**
- **It is not only for factory automation, but can be applied to the most of automations.**

Introduction

- ***Ring Projection Transforms (RPT)* can be a method to find the location.**
- **RPT is rotation-invariant, but very slow on general CPU (ex. Intel Core-i7 series).**
- **For example, RPT consumes about 2500ms for 99x99 target image on 640x480 scene.**
- **Therefore, processing time is one of major issue to be used in practice.**

Contents

- Introduction
- **Related Work**
- Proposed Method
- Experimental Result
- Conclusion

Related Work

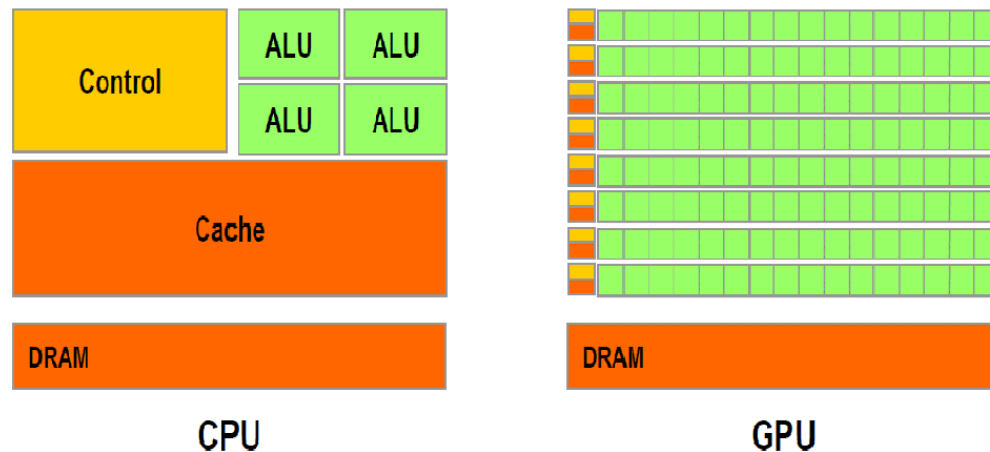
1. What is GPGPU?

1) Using GPU at general computing

- GPGPU is abbreviation of 'General-Purpose computing on Graphics Processing Units'

2) Why?

- Unlike CPU, GPU has lots of ALU, smaller cache, and smaller control units.
- It means that GPU is good to calculate large data, but poor to control branches.



Related Work

2. What is CUDA?

- 1) A kind of GPGPU
- 2) Abbreviation of “Compute Unified Device Architecture”
- 3) Developed by nVIDIA

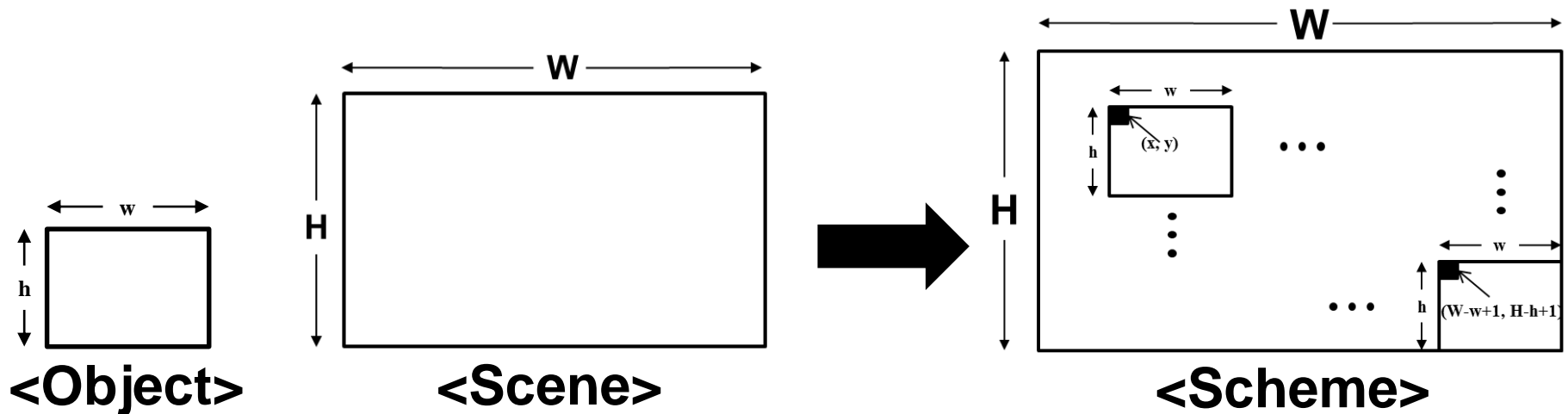


Related Work

3. What is RPT in detail?

1) A kind of block based matching

- To find object on the scene, slide object over the scene. Whenever you slide, calculate the degree of similarity.
- If there is only 1 object in the scene, then the location of highest degree indicate the location of object.



Related Work

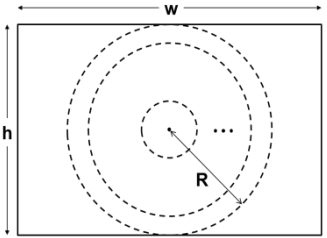
3. What is RPT in detail?

2) Rotation-invariant matching

Step 1: To calculate similarity, extract vector $T(r)$ from object image. $I(x, y)$ is the value of pixel at (x, y) on the object image.

Step 2: extract vector $S_{(x,y)}(r)$ from scene image according to equation (2).

Step 3: calculate 'Normalized Cross-Correlation(=NCC(x, y))' between $T(r)$ and $S_{(x,y)}(r)$.



$$\vec{T}(r) = \frac{1}{2\pi r} \int_0^{2\pi} I\left(\frac{w}{2} + r\cos\theta, \frac{h}{2} + r\sin\theta\right) d\theta, \quad (0 < r \leq R) \quad (1)$$

$$\vec{S}_{(x,y)}(r) = \frac{1}{2\pi r} \int_0^{2\pi} I'\left(x + \frac{w}{2} + r\cos\theta, y + \frac{h}{2} + r\sin\theta\right) d\theta, \quad \begin{cases} 0 \leq x \leq W-w+1 \\ 0 \leq y \leq H-h+1 \\ 0 < r \leq R \end{cases} \quad (2)$$



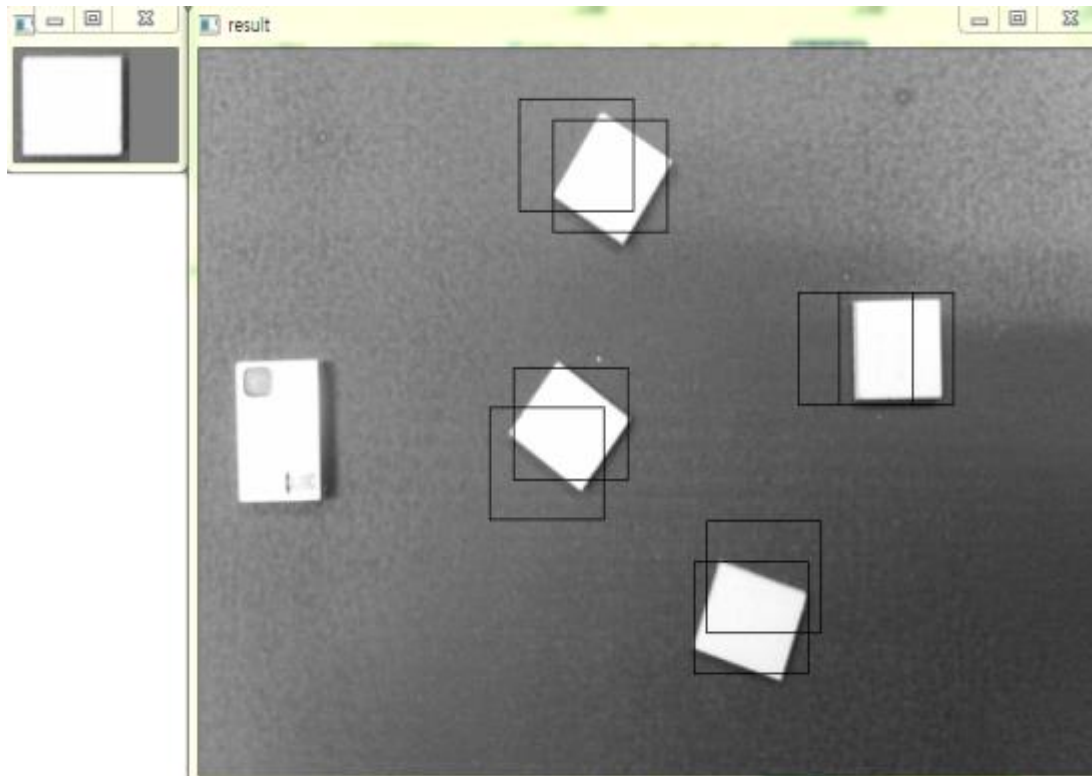
$$O(x, y) = \text{NCC}(\vec{T}, \vec{S}_{(x,y)}) = \frac{\text{Co-variance}(\vec{T}, \vec{S}_{(x,y)})}{(R-1)\sigma(\vec{T})\sigma(\vec{S}_{(x,y)})} \quad (3)$$

Related Work

3. What is RPT in detail?

3) Unavailable for plain pattern object

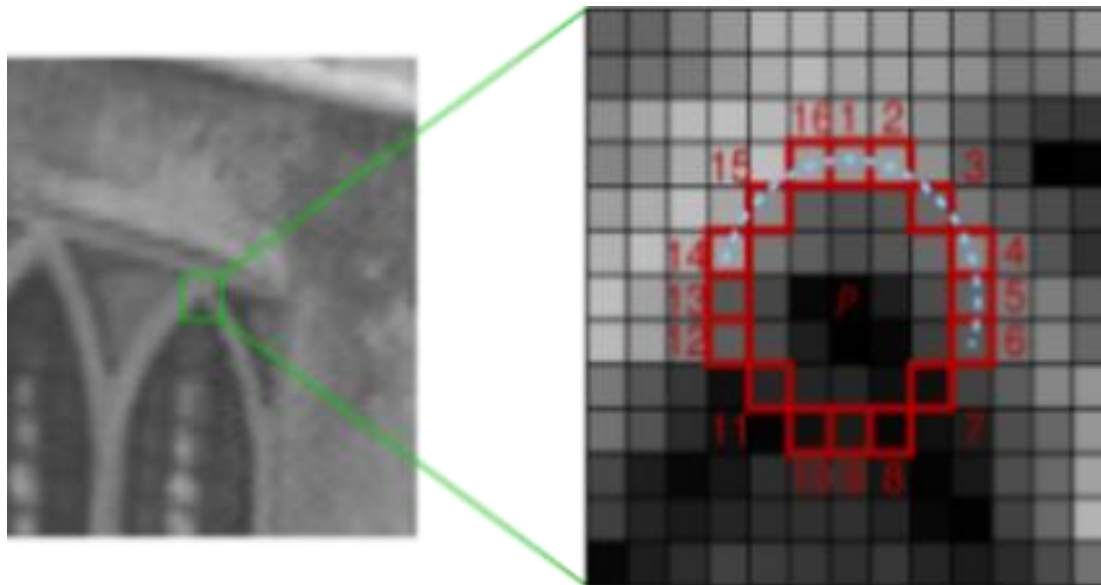
- If object has plain pattern, then RPT will find wrong point which is located in nearby object.



Related Work

4. What is 'FAST' in detail?

- 1) Abbreviation of 'Features from Accelerated Segment Test'
- 2) One of method for detect corner points



Related Work

5. What is 'Contour Matching' in detail?

- 1) Method to find object by only contour set.
- 2) Since it uses only contour, it is suitable for plain pattern object which is not common object.
- 3) It is calculated by moments of image.

$$hu(0) = \eta_{20} + \eta_{02}$$

$$m_{ji} = \sum_{x,y} (I(x,y) \cdot x^j y^i)$$

$$hu(1) = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\mu_{ji} = \sum_{x,y} (I(x,y) \cdot (x - \bar{x})^j (y - \bar{y})^i)$$

$$hu(2) = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$$

$$hu(3) = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\eta_{ji} = \frac{\mu_{ji}}{m_{00}^{(1+(i+j)/2)}}$$

$$hu(4) = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$hu(5) = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$hu(6) = (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

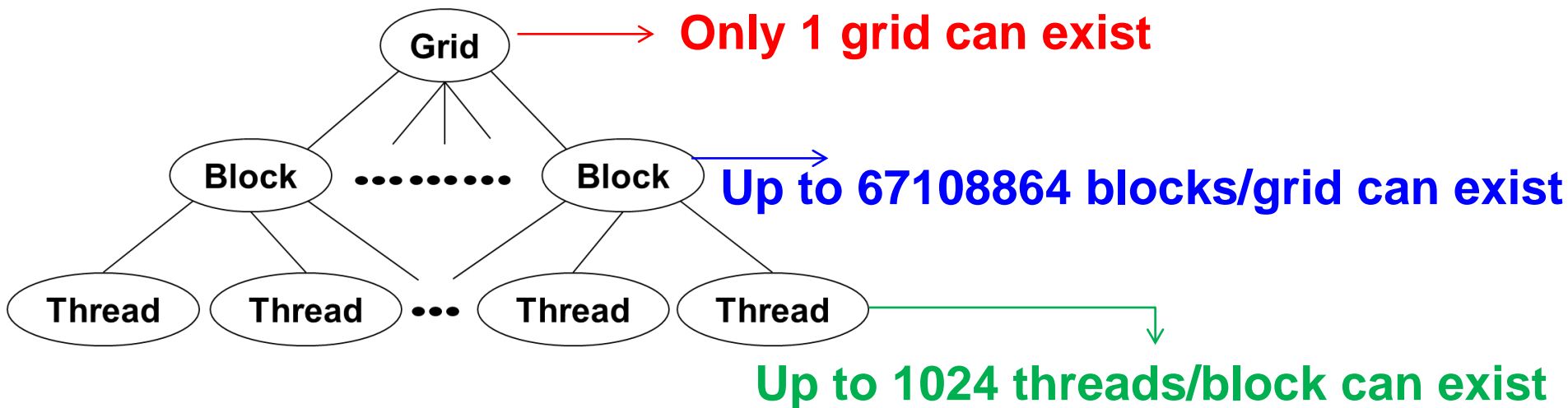
Contents

- Introduction
- Related Work
- **Proposed Method**
- Experimental Result
- Conclusion

Proposed Method

1. Thread allocation

- 1) CUDA provides logically hierarchical structure with grid, block, and thread



Proposed Method

1. Thread allocation

- 2) After calculating RPT, we will get $(W-w+1)(H-h+1)$ values of result
- 3) Therefore, to assign 1 thread/value, we will allocate an in Equation (4).

$$\begin{aligned} & \text{block}(x, y) \\ &= \left(\frac{(W-w+1) + \text{thread}.x - 1}{\text{thread}.x}, \frac{(H-h+1) + \text{thread}.y - 1}{\text{thread}.y} \right) \end{aligned} \quad (4)$$

Proposed Method

2. Strategy of using memory

- 1) **CUDA provides several kinds of memory.**
 - **Global memory : Slowest but largest.**
 - **Shared memory : Fastest but smallest.**
 - **Texture memory : L1-cached memory.**
 - **Constant memory : L1-cached and broad casting.**
- 2) **In case of constant memory, we can achieve 16 times speed for accessing memory.**
- 3) **To maximize this advantage, we will store $T(r)$ in constant memory because whole threads will load $T(r)$ at the same time.**

Contents

- Introduction
- Related Work
- Proposed Method
- **Experimental Result**
- Conclusion

Experimental Result

- **The main purpose is to improve speed of RPT algorithm.**
- **Therefore, we measure time for calculating.**
- **The experiments were conducted on a PC with running Windows 7 64bit, 16 GB RAM, Intel Core i5-4670 3.4GHz processor and GeForce GTX 770. The graphics card has 2GB RAM on board.**
- **The test image set contains a scene image with size 640x480 and template image with size 99x99.**

Experimental Result



<Object image>



<Scene image>



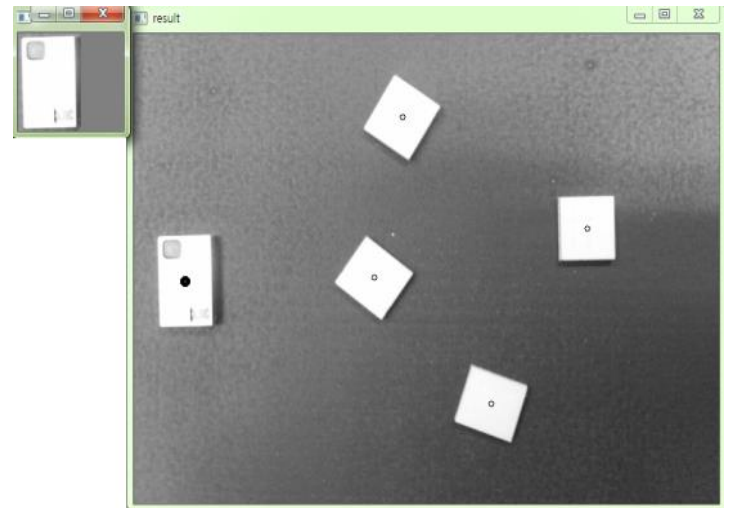
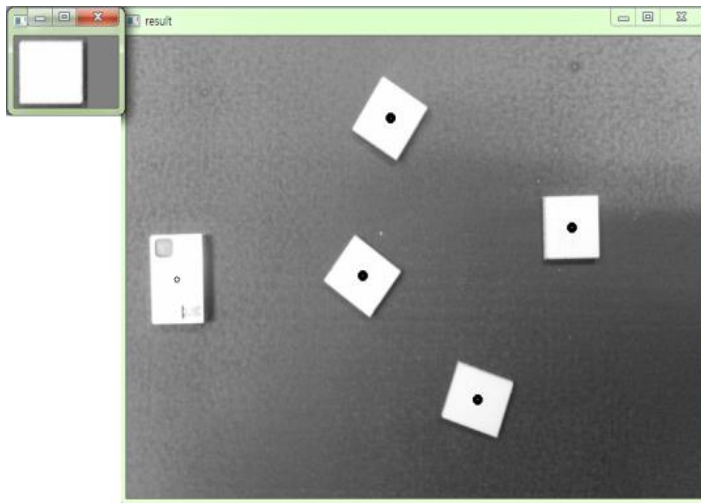
<Result image>

TIME CONSUMPTION FOR PROCESS

Without CUDA	With CUDA	Degree of improvement
2645.00[ms]	75.37[ms]	x35.1

Experimental Result

- Case of Plain Pattern Object.



Contents

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- **Conclusion**

Conclusion

- **Our proposed method provides high computing power and applicability for general use. Thus, we can make this algorithm available option for other automation systems.**
- **Also, our proposed method can be a suitable solution to shorten processing time for pattern recognition.**
- **Furthermore, we also can apply our method not only to pattern recognition but also to other computing problems.**

Thank You