Fast Object Recognition Algorithm Using CUDA

Seung-ho Kim and Jun-dong Cho

VADA Lab
Department of Electrical and Computer Engineering
Sungkyunkwan Univ.

Contents

Contents

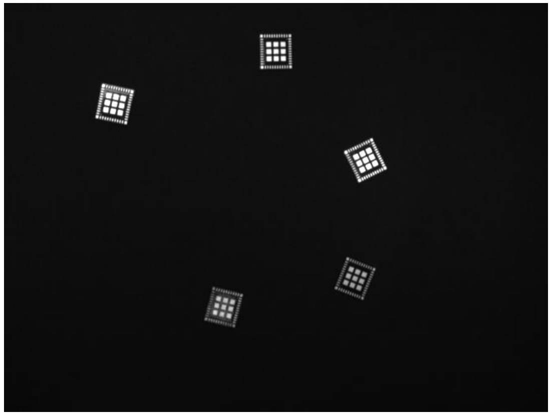
- **Introduction**
- **Related Work**
- **Proposed Method**
- **Experimental Result**
- **Conclusion**

Introduction

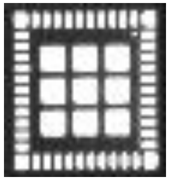
- **In factory automation, robot arms replace human-manual assembly.**
- **To pick the specific object, we should know the location of the object.**
- **To know locations of objects, pattern recognition is essential.**
- **It is used not only for factory automation, but also for the most of automation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

Introduction

- ***Ring Projection Transforms(RPT)*** **is rotation-invariant but very slow on general CPU (ex. Intel Core-i7 series).**
- **For example, Ring Projection Transforms consumes about 6000ms for 45x48 target image on 640x480 scene.**
- **Therefore, processing time is one of major issue in the practical use.**



<Scene image>



<Object image>

Contents

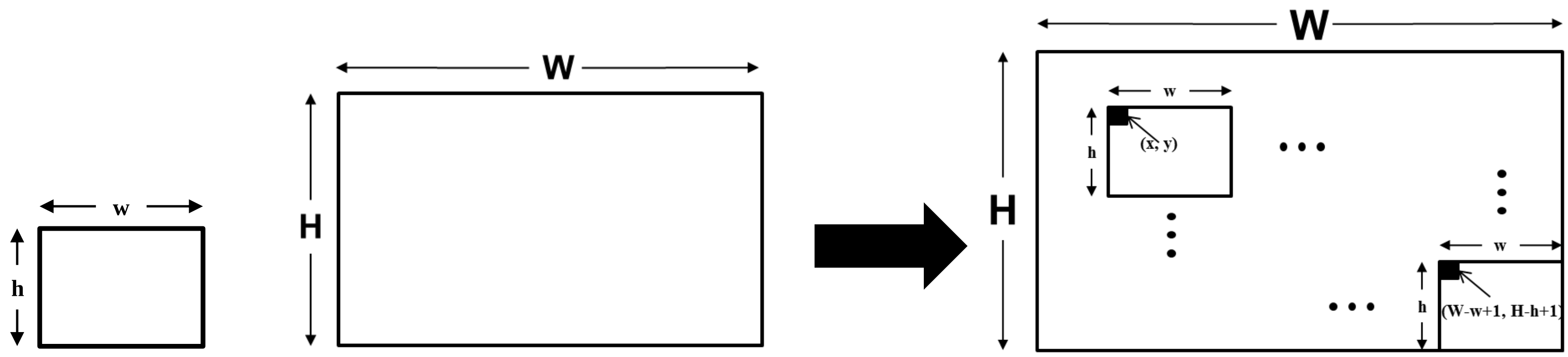
- Introduction
- **Related Work**
- Proposed Method
- Experimental Result
- Conclusion

Related Work

1. Block based matching [2]

- ➢ **To find object on the scene, we slide object over the scene, and calculate the degree of similarity.**

- ➢ **If there is only one object in the scene, the location of highest degree indicates the location of the object.**



<Object> **<Scene>** **<Scheme>**

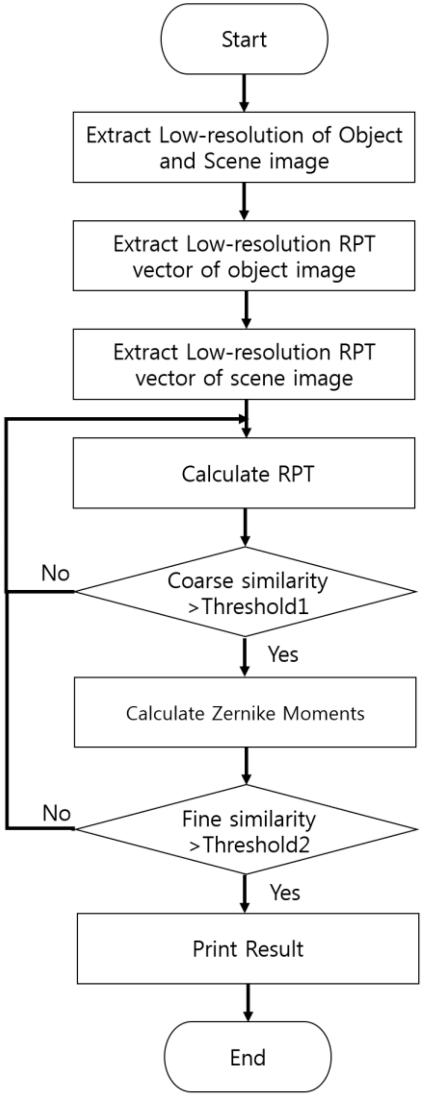
[2] Lee, Wen-Chia, and Chin-Hsing Chen. "A fast template matching method for rotation invariance using two-stage process." Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on. IEEE, 2009.

Related Work

2. Coarse-Fine Strategy [5]

- **To determine the matching candidates, we make the low-resolution of scene and template image.**

- **With the low-resolution template and scene image for the coarse matching, a coarse search method called ring projection transforms (RPT) is adopted.**



[5] Lee, Wen-Chia, and Chin-Hsing Chen. "A fast template matching method for rotation invariance using two-stage process." Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on. IEEE, 2009.

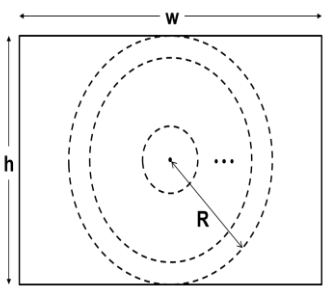
Related Work

3. Rotation invariant called ring projection transforms(RPT) [1]

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

② Extract vector $S_{(x,y)}(r)$ from scene image according to (2).

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



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

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

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

[1] 이상우, and 김회율. "투영법을 이용한 회전불변 템플릿 매칭 (Rotation-Invariant Template Matching Using Projection Method)." 대한전자공학회 학술대회 논문집 제 19 권 1 호 19.1 (1996): 475-478.

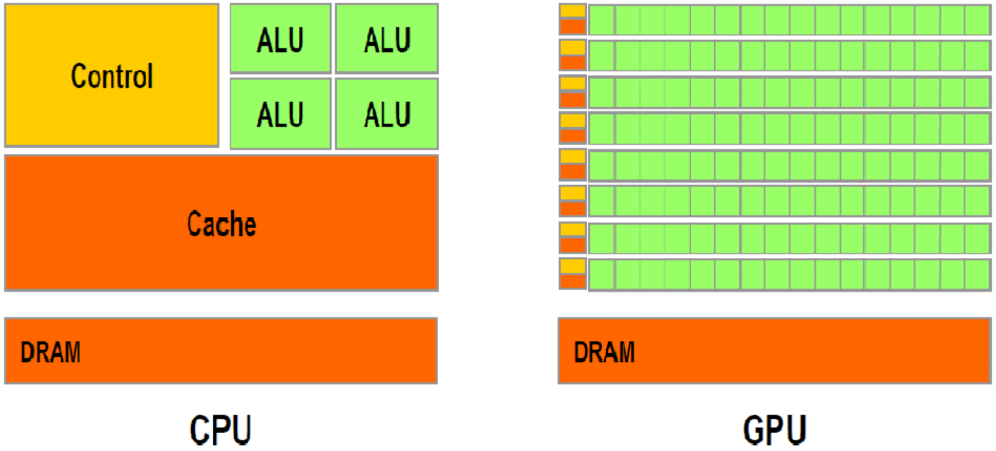
Related Work

4. GPGPU [3]

- **Use GPU at general computing**
  GPGPU is abbreviation of ‘General-Purpose computing on Graphics Processing Units’

- **Unlike CPU, GPU has lots of APU’s, smaller cache, and smaller control units.**

- **GPU is efficient to calculate large data, but to control branches.**



[3] Sanders, Jason, and Edward Kandrot. *CUDA by example: an introduction to general-pupose GPU programming*. Addison-Wesley Professional, 2010.

Related Work

4. GPGPU

- **CUDA is a kind of GPGPU**
- **Abbreviation of "Compute Unified Device Architecture"**
- **A parallel computing platform and application programming interface (API) model created by NVIDIA**

Contents

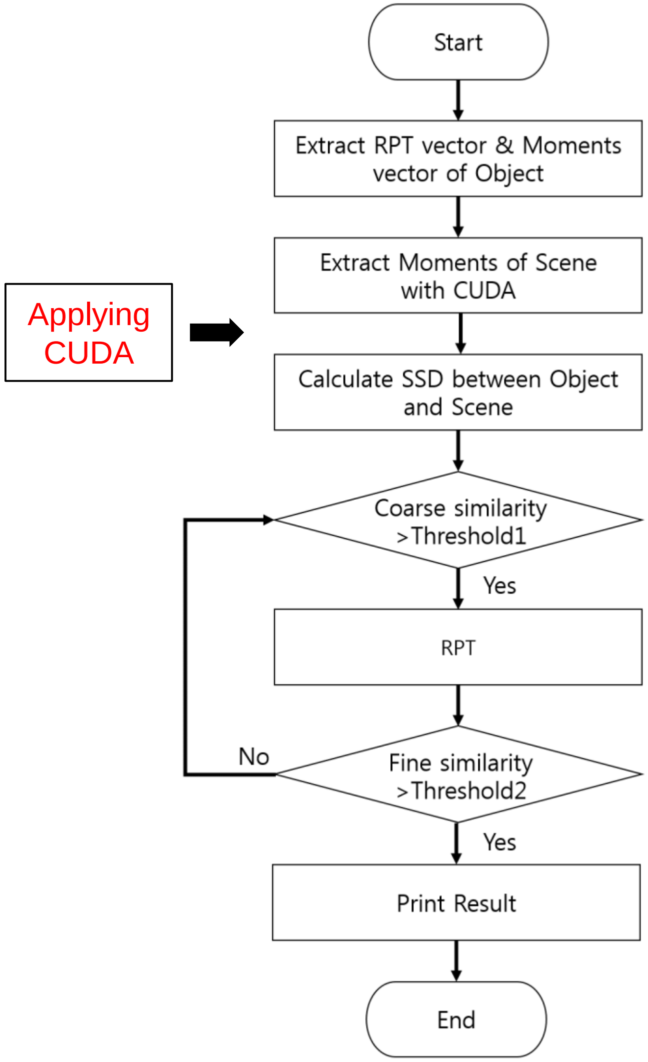
- Introduction
- Related Work
- **Proposed Method**
- Experimental Result
- Conclusion

Proposed Method

1. Coarse-Fine Strategy

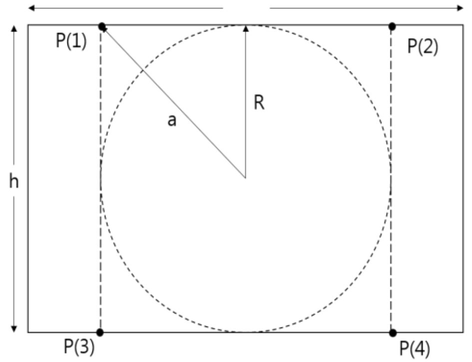
- Main idea is to calculate Coarse-Matching for lower computation.

- We apply for CUDA.



Proposed Method

2. k^{th} central moments



$$\vec{P}(k) = \sqrt[k]{\frac{1}{w*h}\sum_{i=1}^{h}\sum_{j=1}^{w}(I'(j,i) - E[X])^k} \quad (3)$$

$$\vec{A}_{(x,y)}(k) = \sqrt[k]{\frac{1}{w*h}\sum_{i=1}^{h}\sum_{j=1}^{w}(I(x+j,y+i) - E[X])^k} \quad (4)$$

$$O_2(x,y) = \sqrt{\sum_{k=2}^{4}(\vec{P}(k) - \vec{A}_{(x,y)}(k))^2} \quad (5)$$

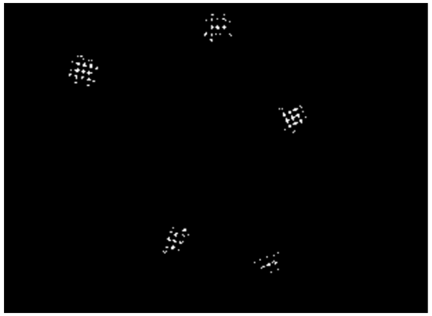
$$C(x,y) = \frac{1}{1+O_2(x,y)} \quad (6)$$

1^{st} Moment	mean of pixel
2^{nd} Moment	standard deviation of pixel
3^{rd} Moment	skewness of pixel
4^{th} Moment	kurtosis of pixel

Proposed Method

2. k^{th} central moments

- Location of candidates using low-resolution RPT



(a) 1/2

(b) 1/3

(c) 1/4

- Location of candidates using k^{th} central moments

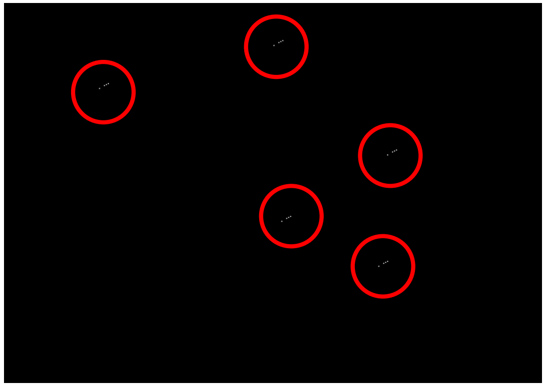


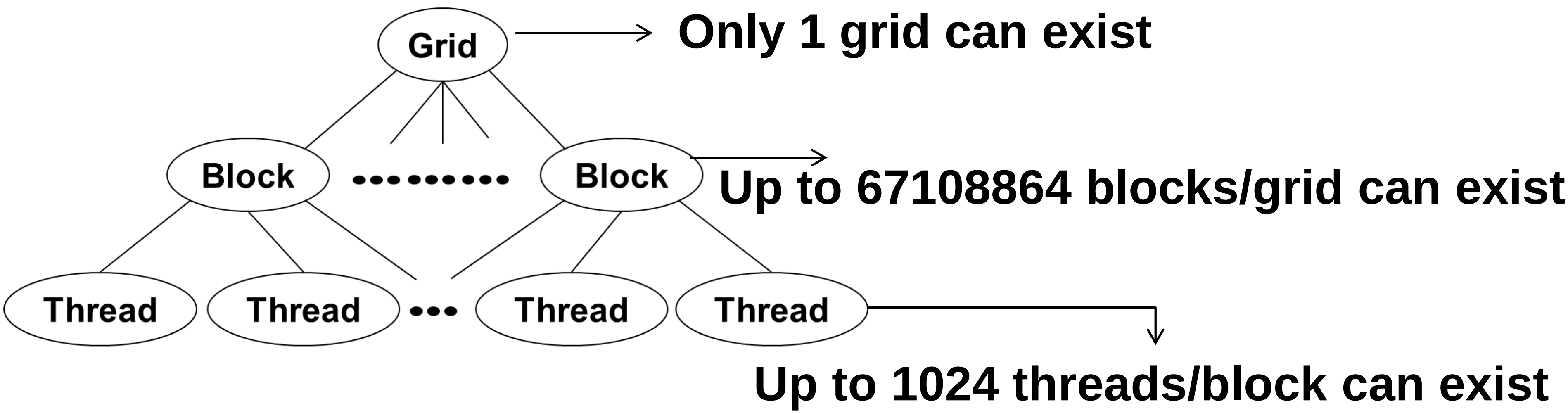
Table 1. PROCESSING TIME FOR COARSE MATCHING

1/2 image	1/3 image	1/4 image	Proposed method with CUDA
1,391.9 ms	594.46 ms	110.08 ms	9.321 ms

Proposed Method

3. Thread allocation

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



Proposed Method

3. 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 like (5).**

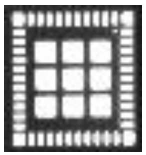
$$\begin{cases} \text{Threads / block} = 1024 \text{ (= Maximum number to allocate)} \\ \text{Blocks / grid} = \dfrac{(W-w+1)(H-h+1) + Threads\ per\ block - 1}{Threads\ per\ block} \end{cases} \quad (5)$$

Contents

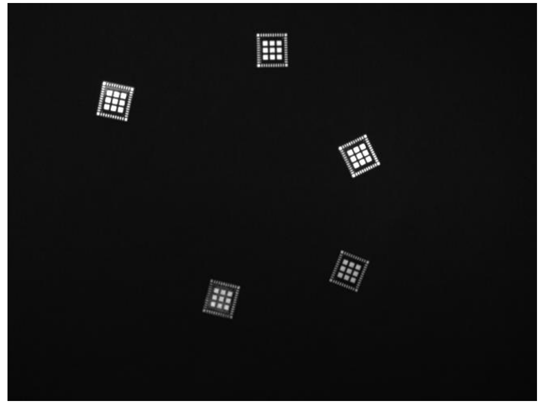
Experimental Result

- The main purpose is to improve speed of Ring Projection Transforms(RPT).
- 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 45x48.

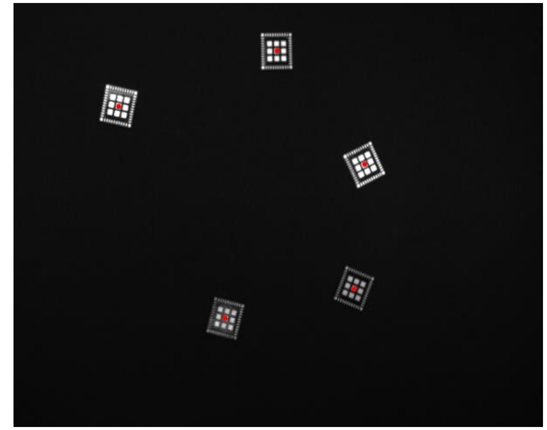
Experimental Result



<Object image>



<Scene image>



<Result image>

Object Size	RPT	Coarse-Fine Without CUDA	Coarse-Fine With CUDA	Degree of Improvement
45x48	1156 ms	113.74 ms	50.7 ms	x22.8
73x71	1486 ms	128.43 ms	19.4 ms	x76.6
87x80	2237 ms	148.82 ms	21.23 ms	x105.37
99x99	3624 ms	137.33 ms	15.12 ms	x239.68
149x113	6727 ms	117.1 ms	13.86 ms	x485.35

Contents

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

Conclusion

- **Our proposed method provides high computing power.**

- **Therefore, our proposed method is effectively used in e.g., robot automation system.**

- **We can apply our method not only to robot object recognition, but also to other recognition problems.**

Reference

[1] 이상우, and 김회율. "투영법을 이용한 회전불변 템플릿 매칭 (Rotation-Invariant Template Matching Using Projection Method)." 대한전자공학회 학술대회 논문집 제 19 권 1 호 19.1 (1996): 475-478.

[2] Lee, Wen-Chia, and Chin-Hsing Chen. "A fast template matching method for rotation invariance using two-stage process." Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on. IEEE, 2009.

[3] Sanders, Jason, and Edward Kandrot. *CUDA by example: an introduction to general-pupose GPU programming*. Addison-Wesley Professional, 2010.

[4] Lee, Wen-Chia, and Chin-Hsing Chen. "A fast template matching method for rotation invariance using two-stage process." Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on. IEEE, 2009.

[5] Kim, Hae Yong, and Sidnei Alves De Araújo. "Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast." Advances in Image and Video Technology. Springer Berlin Heidelberg, 2007. 100-113.

Thank You