



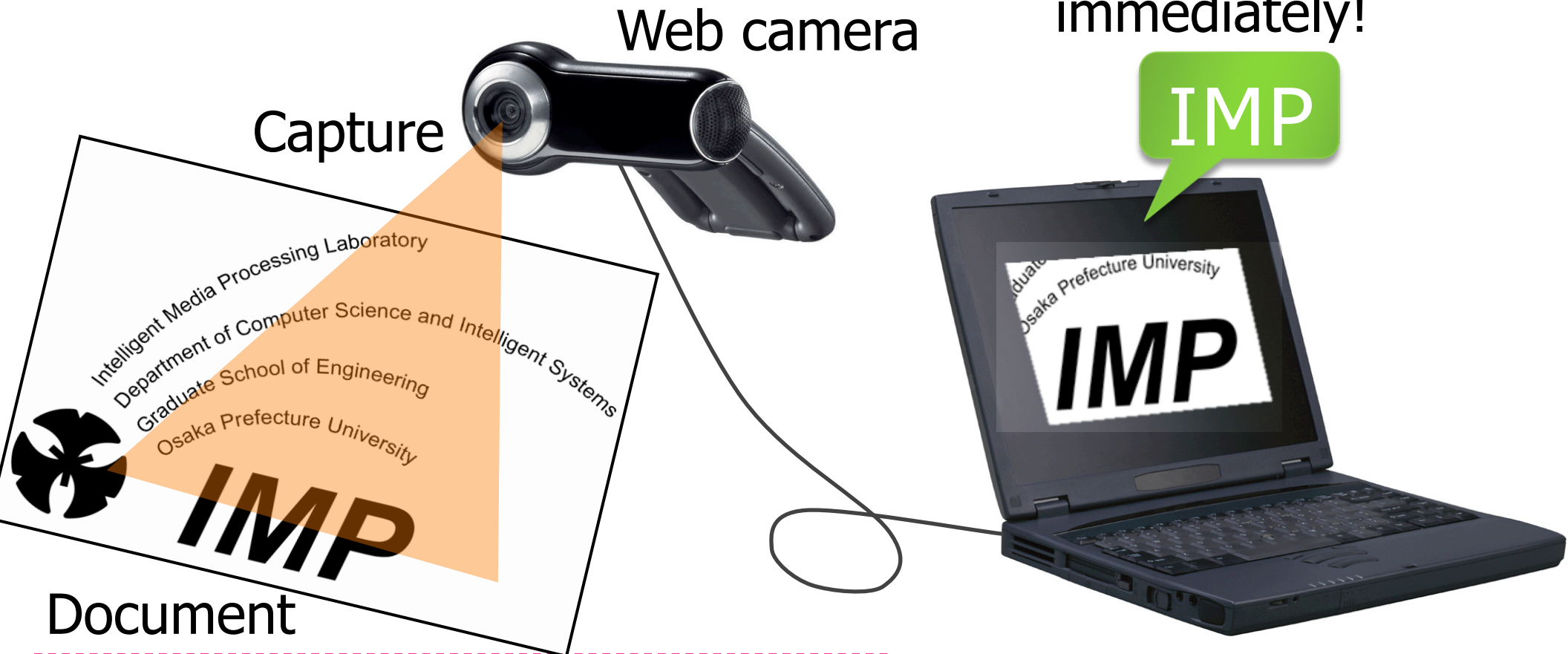
# Real-Time Camera-Based Character Recognition Free from Layout Constraints

M. Iwamura, T. Tsuji, A. Horimatsu, and K. Kise

# Real-Time Camera-Based Character Recognition System

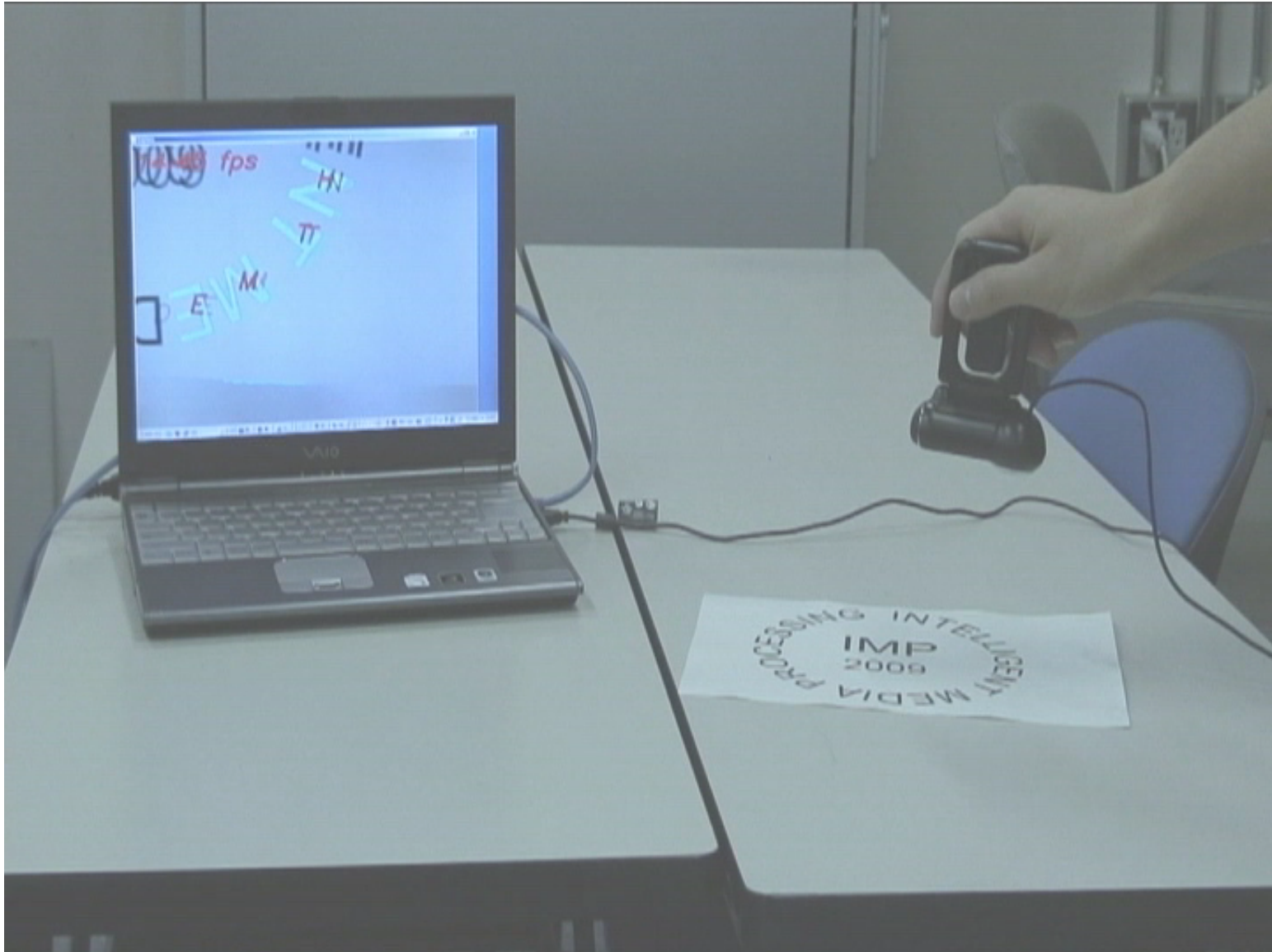
Recognizes ~200 characters/sec

Recognizes character immediately!



# DEMO

---



# Applications

Recognizes all characters in a scene and provide useful information only

Voice navigation for visually disabled people

"Push button" is on your right side



Translation service for foreign travelers

Car-free mall



歩行者天国





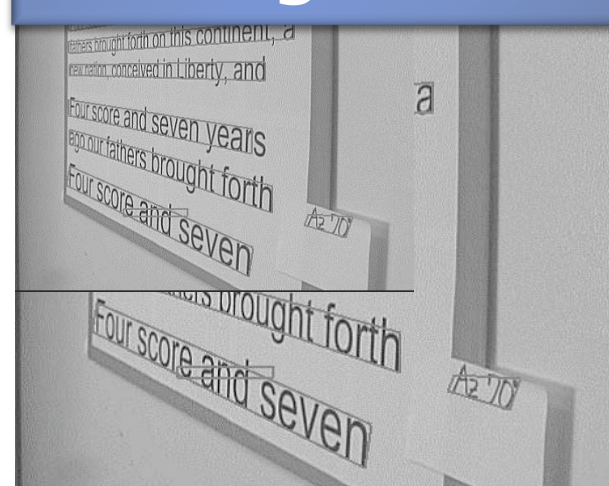
# Existing Methods and Problems

1. Real-time recognition capable **only for characters in a straight text line**



Not recognizable

Recognizable



2. Can recognize each character in a complex layout **with much computational time**



# Existing Methods vs Proposed Method

1: Real-time

2: Perspective distortion

3: Layout free

Myers 2004	○	○	×
Kusachi 2004	×	○	○
Li 2008	×	○	○
Proposed method	○	○	○

Recognition of Individual Characters



Real-time Processing



# Contents

---

1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. Proposed Method
    1. Real-Time Processing
    2. Recognition of Separated Characters
    3. Pose Estimation
  5. Experiment
  6. Conclusion
- 





# Overview of the Proposed Method 1

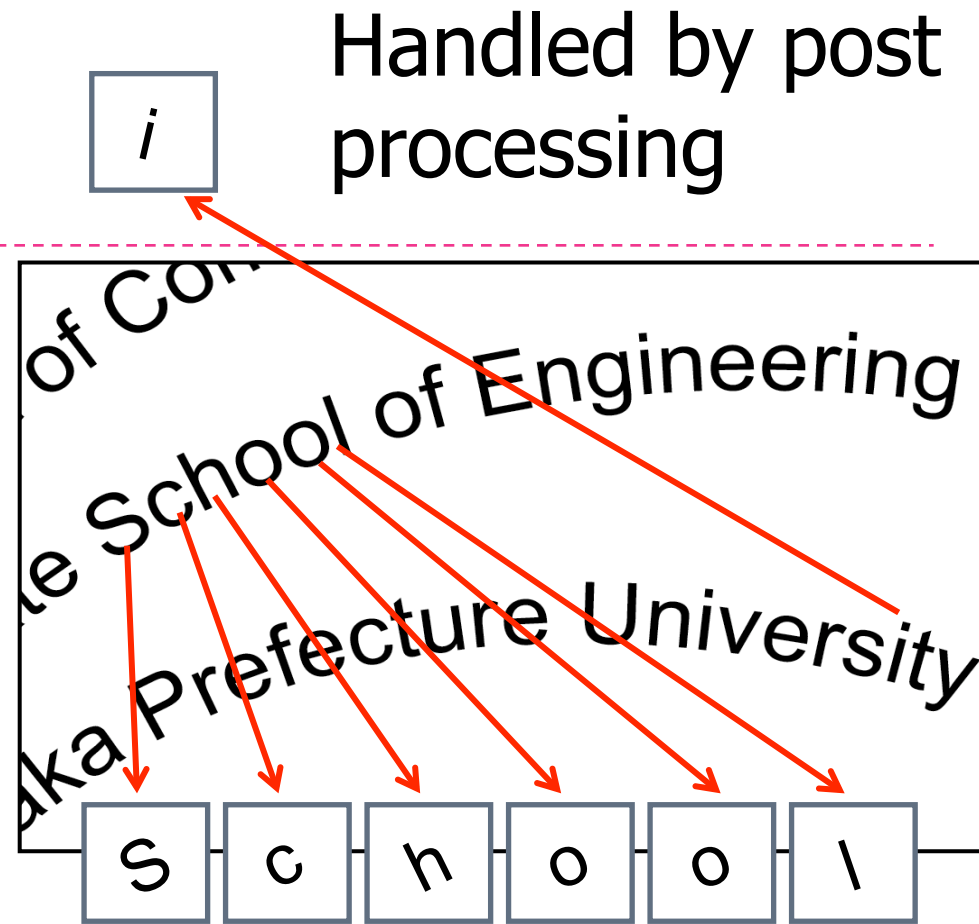
Recognizes individual connected components

Realizes 3: Layout free

Assumptions

- ▶ Black characters are written on a flat white paper
- ▶ All connected components are easily segmented

How to quickly match segmented connected components

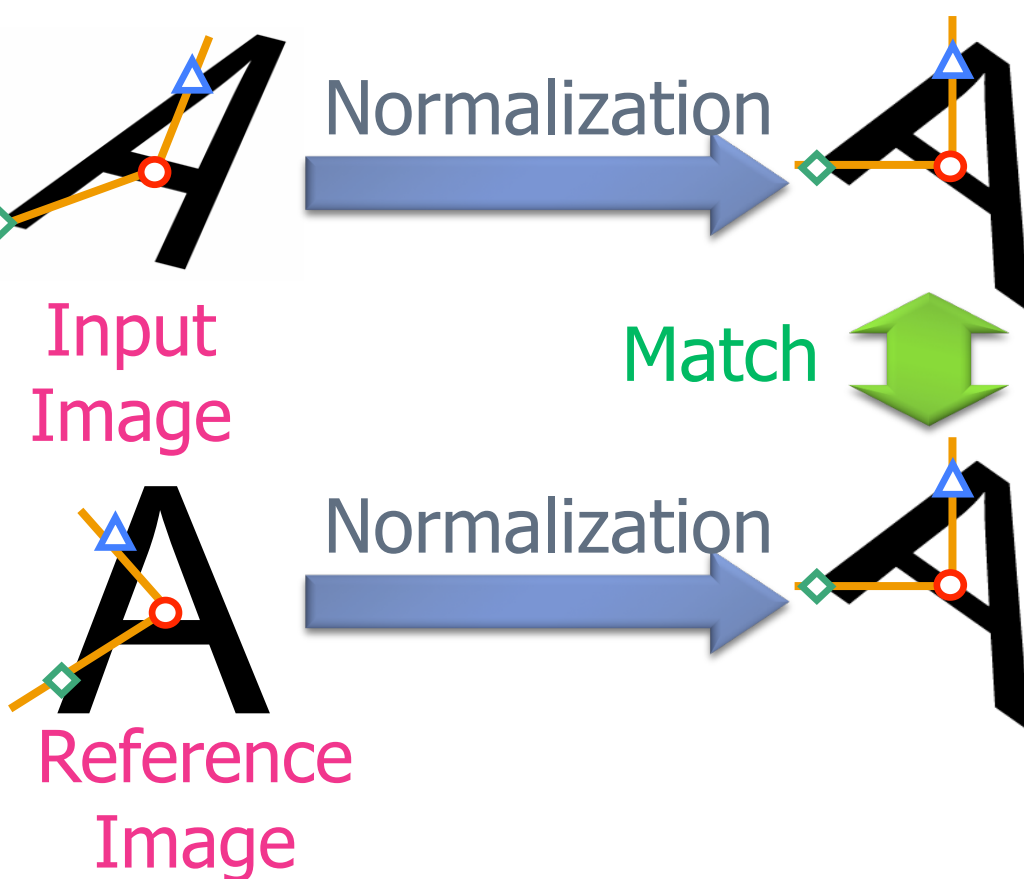


# Overview of the Proposed Method 2

- ▶ Affine invariant recognition
  - ▶ Three corresponding points help matching

Realizes robust recognition  
to

2: Perspective  
distortion



# Overview of the Proposed Method 2: Contour Version of Geometric Hashing

Existing method :  
Geometric Hashing (GH)

Contour Version of GH

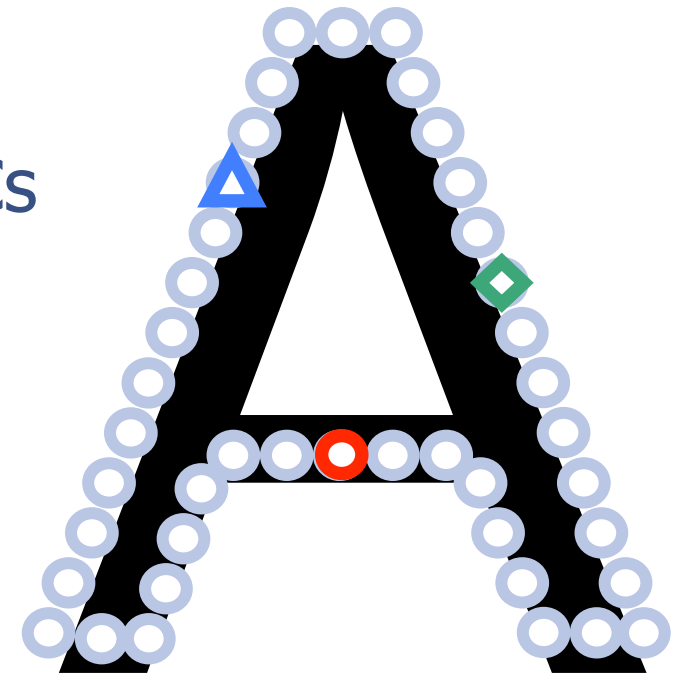
Start point of the  
proposed method



Applied GH to  
recognition of CCs



No. of Points :  $P$

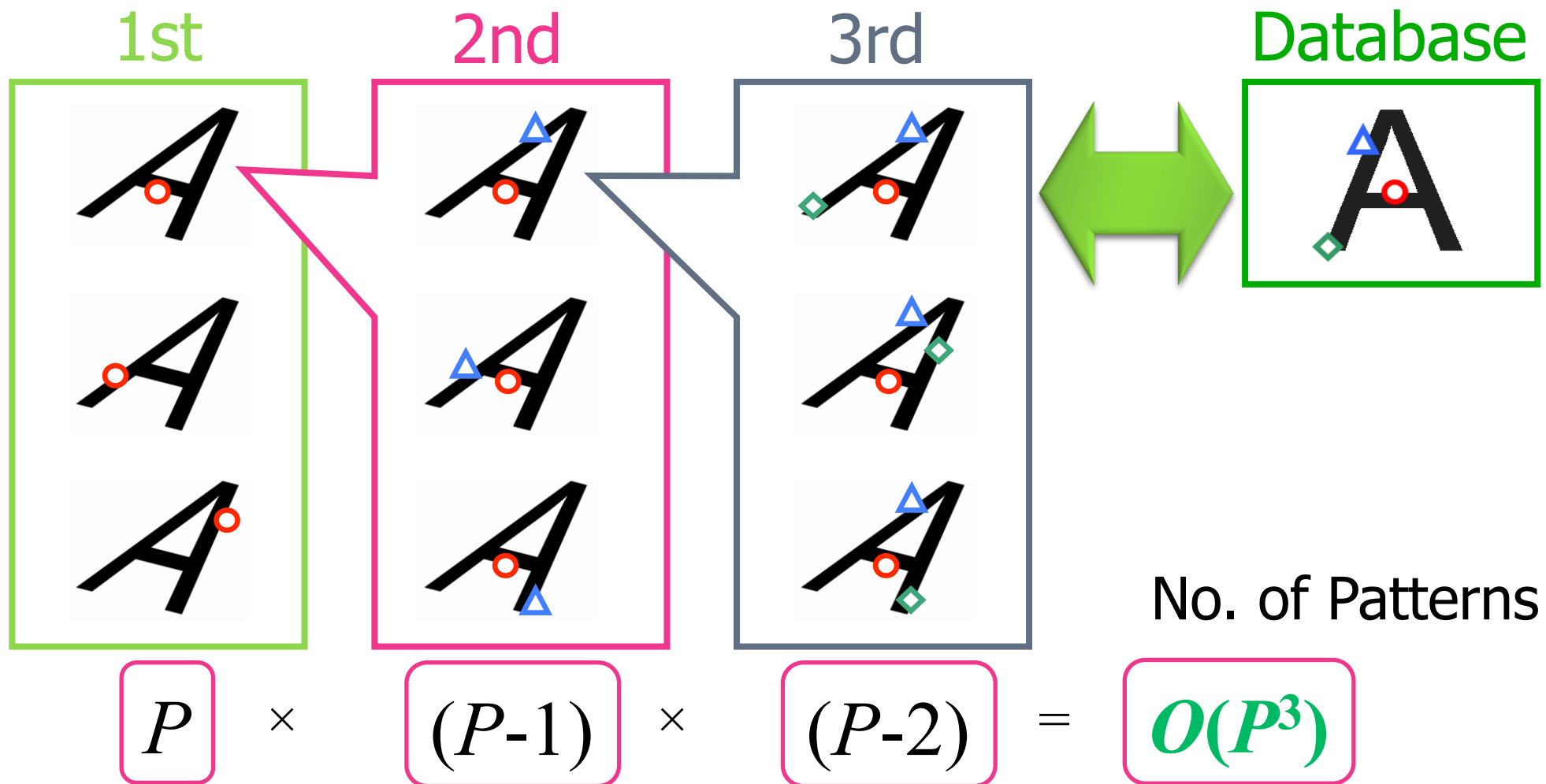


Matching of point arrangement

Matching of Shape

# Overview of the Proposed Method 3: Three-Point Arrangements of CVGH

- ▶ CVGH examines all three points out of  $P$  points





# Overview of the Proposed Method 3: Three-Point Arrangements of Prop. Method

- ▶ Proposed method snips useless three-point arrangements

In case of  $P=100$

CVGH

970,200

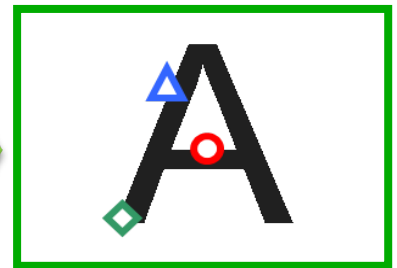
Proposed Method

100

Realizes

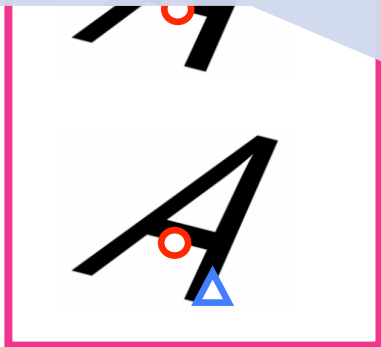
1: Real-time

Database



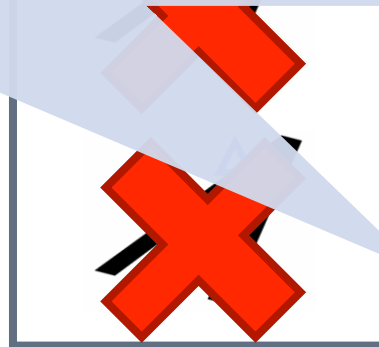
1

×



$P$

×



1

=

$O(P)$

No. of Patterns

$O(P^3)$



# Contents

---

1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. Proposed Method
    1. Real-Time Processing
    2. Recognition of Separated Characters
    3. Pose Estimation
  5. Experiment
  6. Conclusion
- 

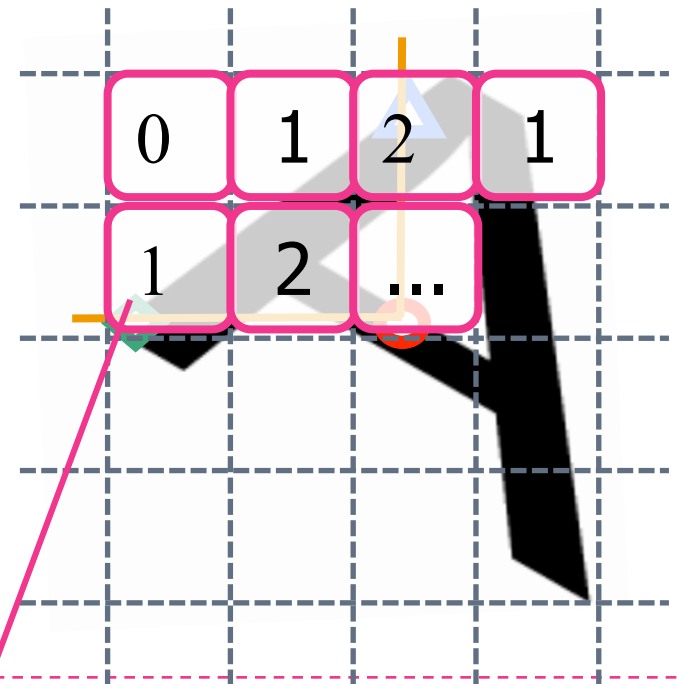
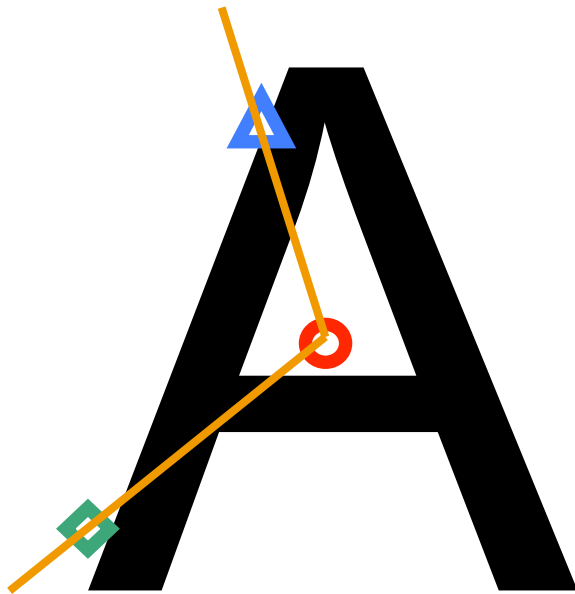


# Contour Version of GH: Matching by Feature Vectors

## ► Calculation of feature vector

1. Normalize
2. Divide into subregions
3. Create a histogram of black pixel
4. Quantize

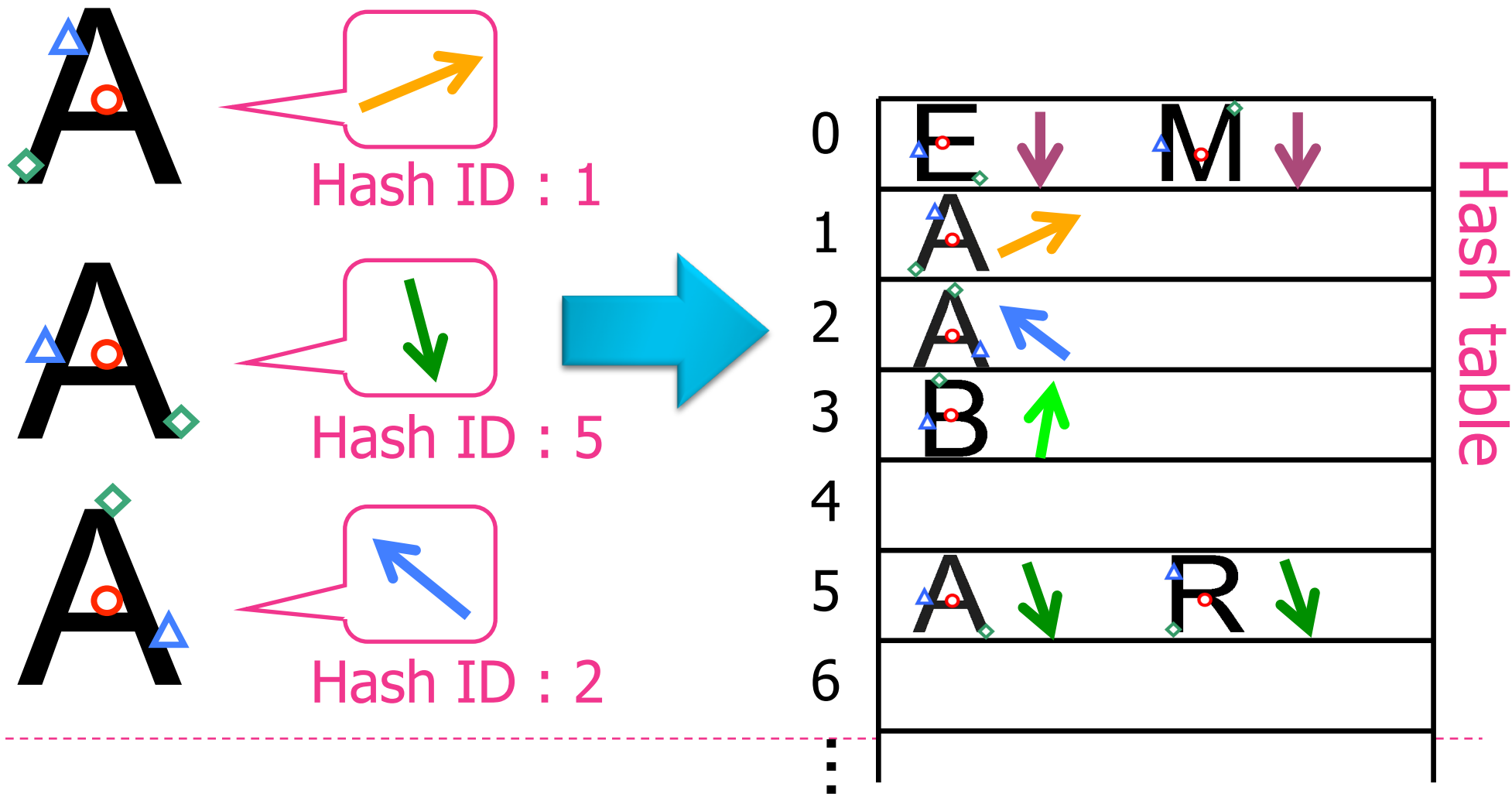
4x4 Mesh Feature



Feature Vector

# Contour Version of GH: Storage

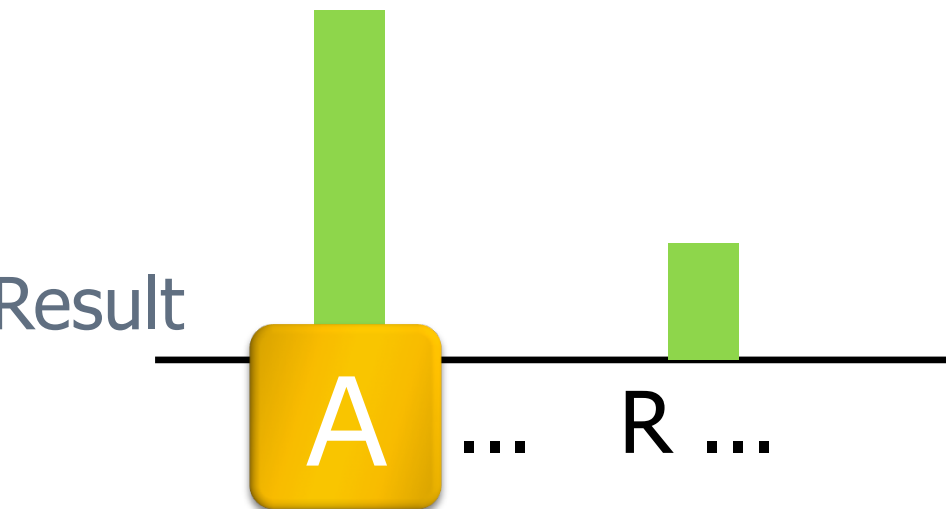
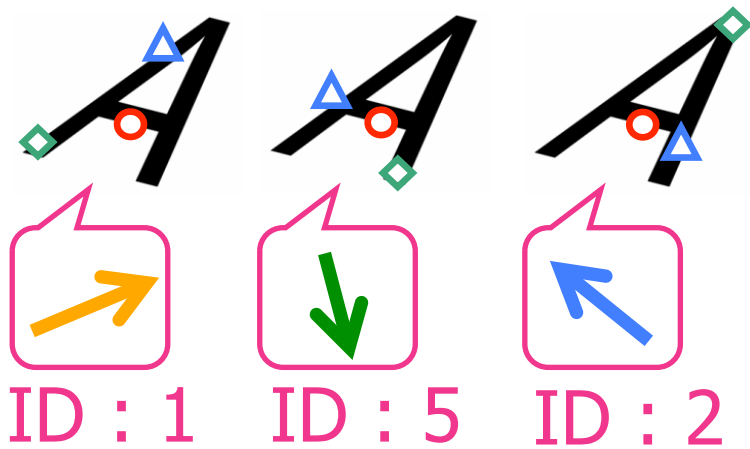
- Feature vectors are stored in the hash table





# Contour Version of GH: Recognition

1. Calculate feature vectors
2. Cast votes



0	E	↓	M	↓
1	A	→		
2	A	←		
3	B	↑		
4				
5	A	↓	R	↓
6				
⋮				

Hash table

# Contents

---

1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. Proposed Method
    1. Real-Time Processing
    2. Recognition of Separated Characters
    3. Pose Estimation
  5. Experiment
  6. Conclusion
- 

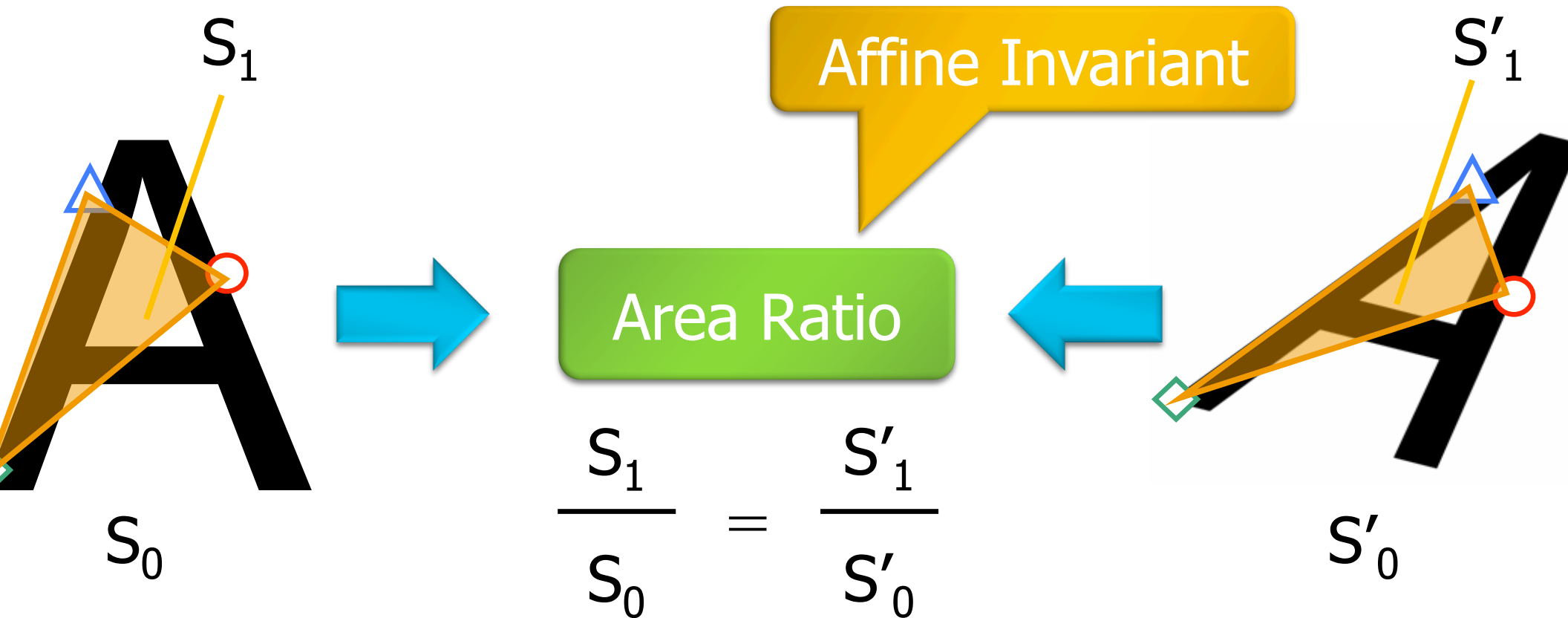


# Proposed Method 1: Real-Time Processing by Affine Invariant

## ► Area ratio

Usual usage

► Three-point arrangement → Area ratio

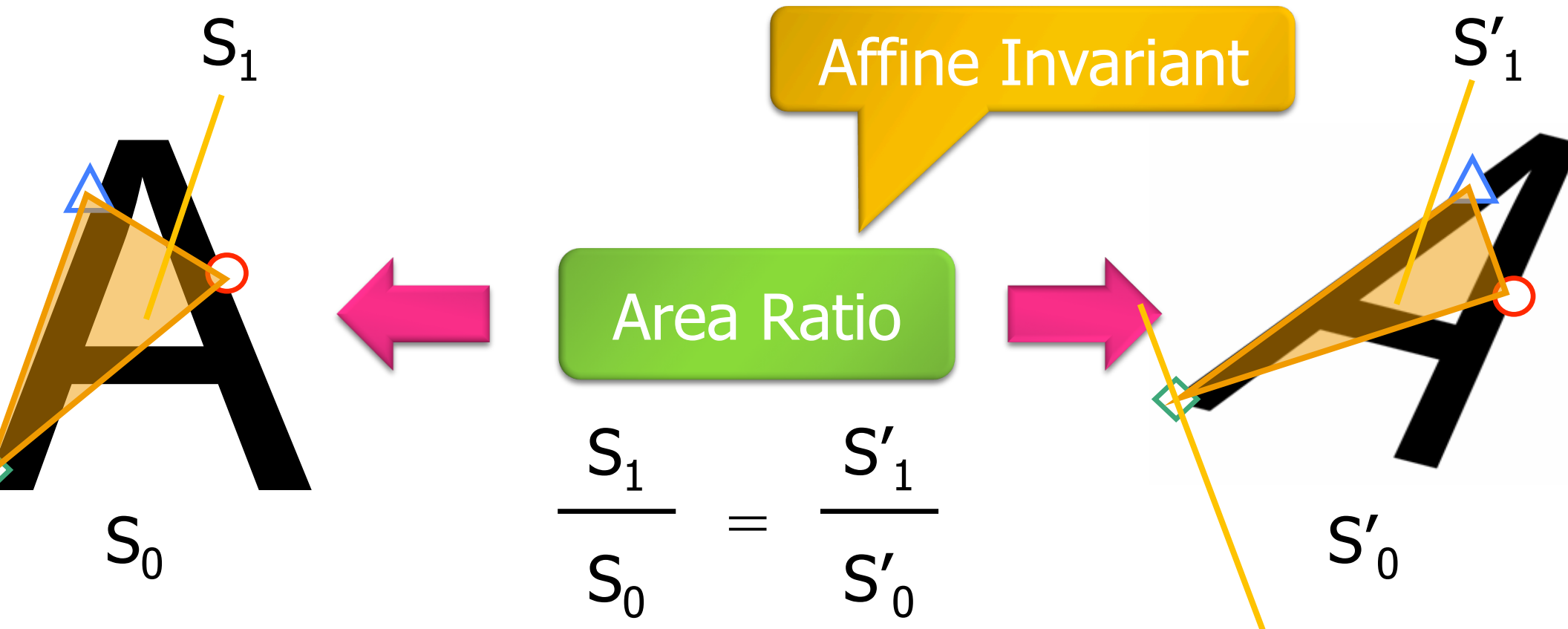


# Proposed Method 1: Real-Time Processing by Affine Invariant

- ▶ Area ratio

Unusual usage

- ▶ Two-point arrangement + Area ratio → Third point



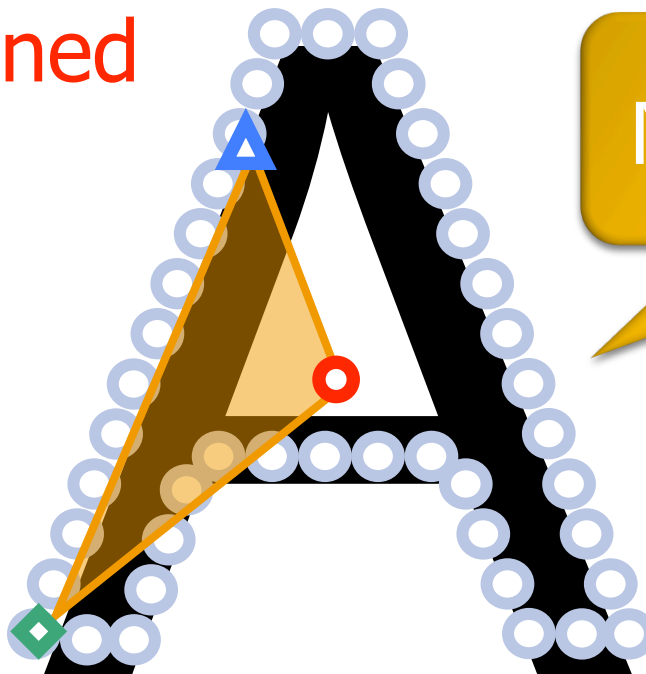


# Proposed Method 1: How to Select Three Points

---

- ▶ ★ 1<sup>st</sup> point: Centroid (Affine Invariant)
- ▶ 2<sup>nd</sup> point: Arbitrary point out of  $P$  points
- ▶ ★ 3<sup>rd</sup> point: Determined by the area ratio

★ Uniquely Determined



No. of Points :  $P$



# Contents

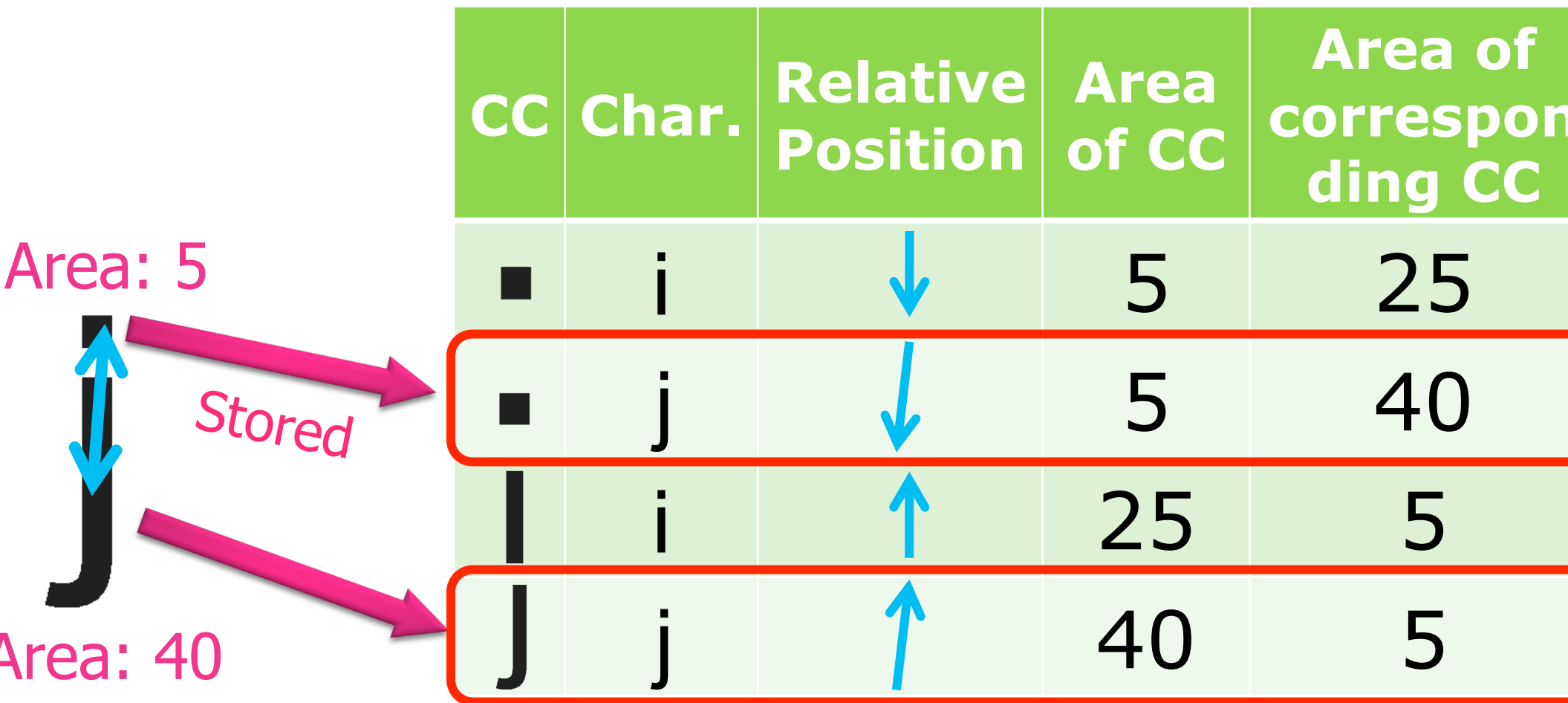
---

1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. Proposed Method
    1. Real-Time Processing
    2. Recognition of Separated Characters
    3. Pose Estimation
  5. Experiment
  6. Conclusion
- 



# Proposed Method 2: Recognition of Separated Characters

- Create a separated character table for post processing



The diagram illustrates the recognition of separated characters 'J' and 'j'. On the left, a large black 'J' is shown with a blue double-headed vertical arrow indicating its height. A pink arrow labeled 'Area: 5' points from the top of the 'J' to the first row of the table. Another pink arrow labeled 'Area: 40' points from the bottom of the 'J' to the fourth row of the table. A pink arrow labeled 'Stored' points from the middle of the 'J' to the second row of the table. The table has five columns: 'CC', 'Char.', 'Relative Position', 'Area of CC', and 'Area of corresponding CC'. The first row contains a small black square, 'i', a downward arrow, '5', and '25'. The second row contains a small black square, 'j', a downward arrow, '5', and '40'. The third row contains a small black vertical bar, 'i', an upward arrow, '25', and '5'. The fourth row contains a large black 'J', 'j', an upward arrow, '40', and '5'. The second and fourth rows are highlighted with a red border.

CC	Char.	Relative Position	Area of CC	Area of corresponding CC
■	i	↓	5	25
■	j	↓	5	40
┃	i	↑	25	5
J	j	↑	40	5

# Contents

---

1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. **Proposed Method**
    1. Real-Time Processing of CVGH
    2. Recognition of Separated Characters
    3. **Pose Estimation**
  5. Experiment
  6. Conclusion
- 



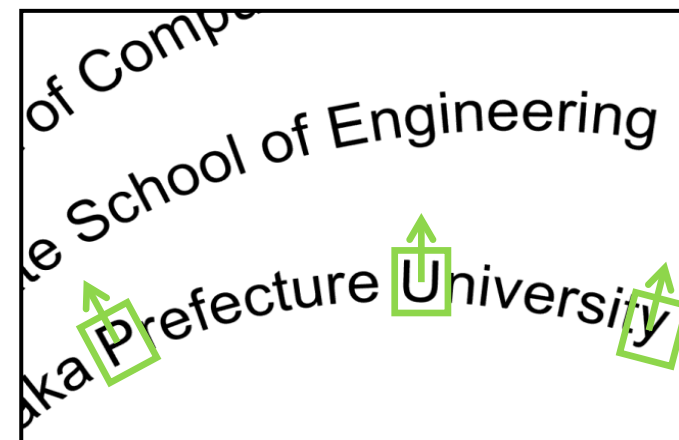
# Proposed Method 3: Pose Estimation

- Estimation of the

Pose of Paper



Pose of Characters



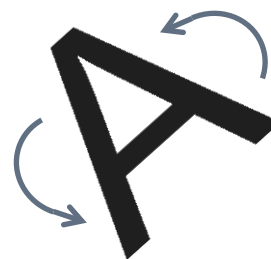
Parameters



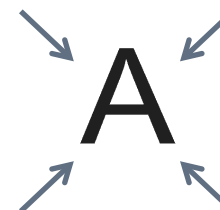
Independent  
Scaling



Shear



Rotation



Scaling

# Contents

---

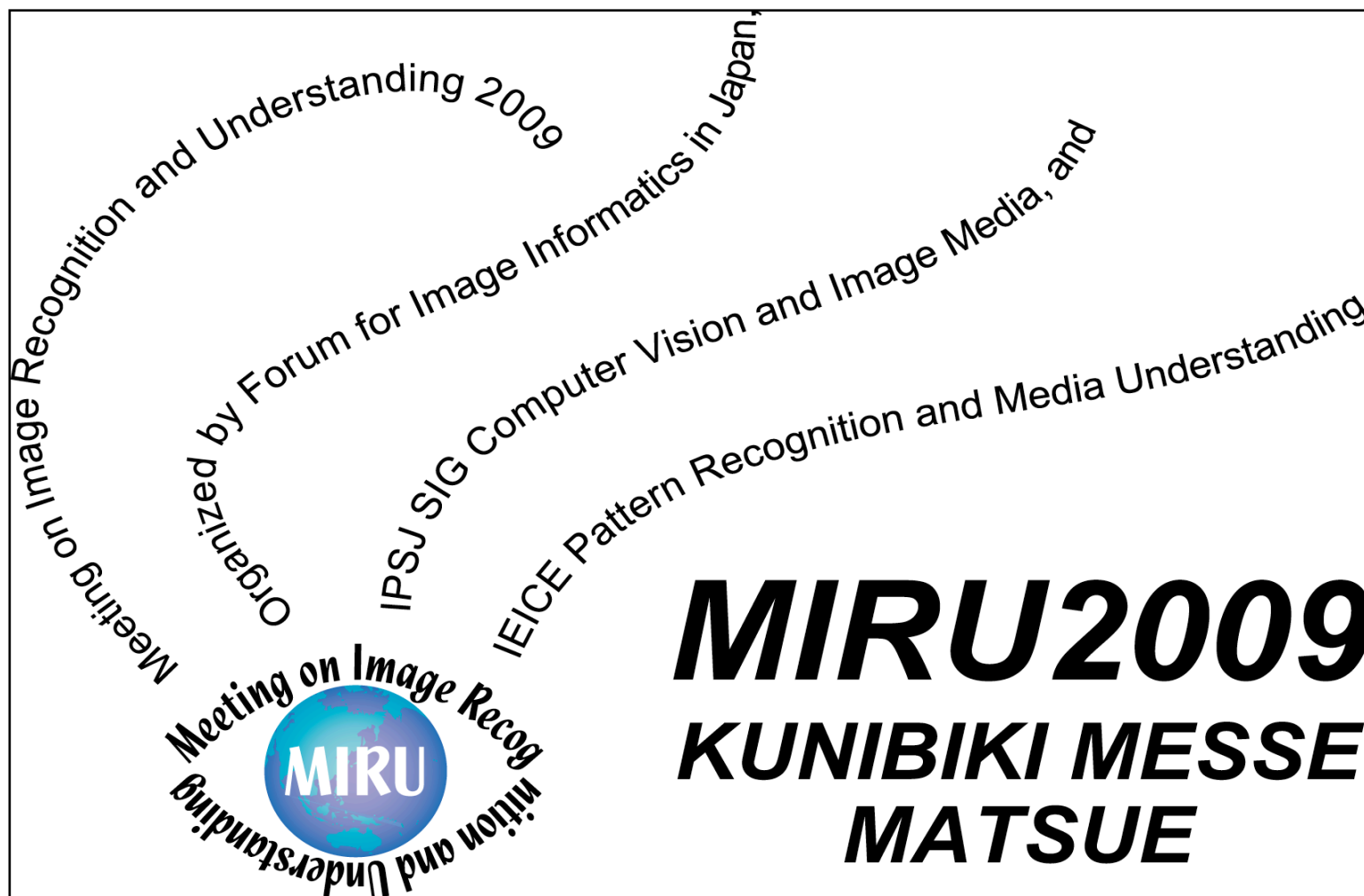
1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. Proposed Method
    1. Real-Time Processing
    2. Recognition of Separated Characters
    3. Pose Estimation
  5. **Experiment**
  6. Conclusion
- 



# Experiment: Recognition Target

3 Fonts

236 Chars





# Experiment: Recognition Target

---

- ▶ Captured from three different angles
- ▶ A server was used
  - ▶ CPU: AMD Opteron 2.6GHz



Angle: 0 deg.



Angle: 30 deg.



Angle: 45 deg.

---

# Experiment: Conditions

---

- ▶ Some characters are difficult to distinguish under affine distortions
  - ➔ Characters in a cell were treated as the same class

0 O o	W w
6 9	X x
C c	N Z z
I l	p d
S s	q b
u n	7 L V v



# Experiment: Recognition Result

- ▶ Achieved high recognition rates and high speed by changing a control parameter

180-210 characters/sec

Settings	High recognition rates			High speed		
Angle (deg.)	0	30	45	0	30	45
Time (ms)	7990	7990	7020	1300	1260	1140
Recog. Rate (%)	94.9	90.7	86.4	86.9	81.8	76.3
Reject. Rate (%)	0.4	3.0	6.4	6.4	9.3	16.5
Error Rate (%)	4.7	6.4	7.2	6.8	8.9	7.2

# Contents

---

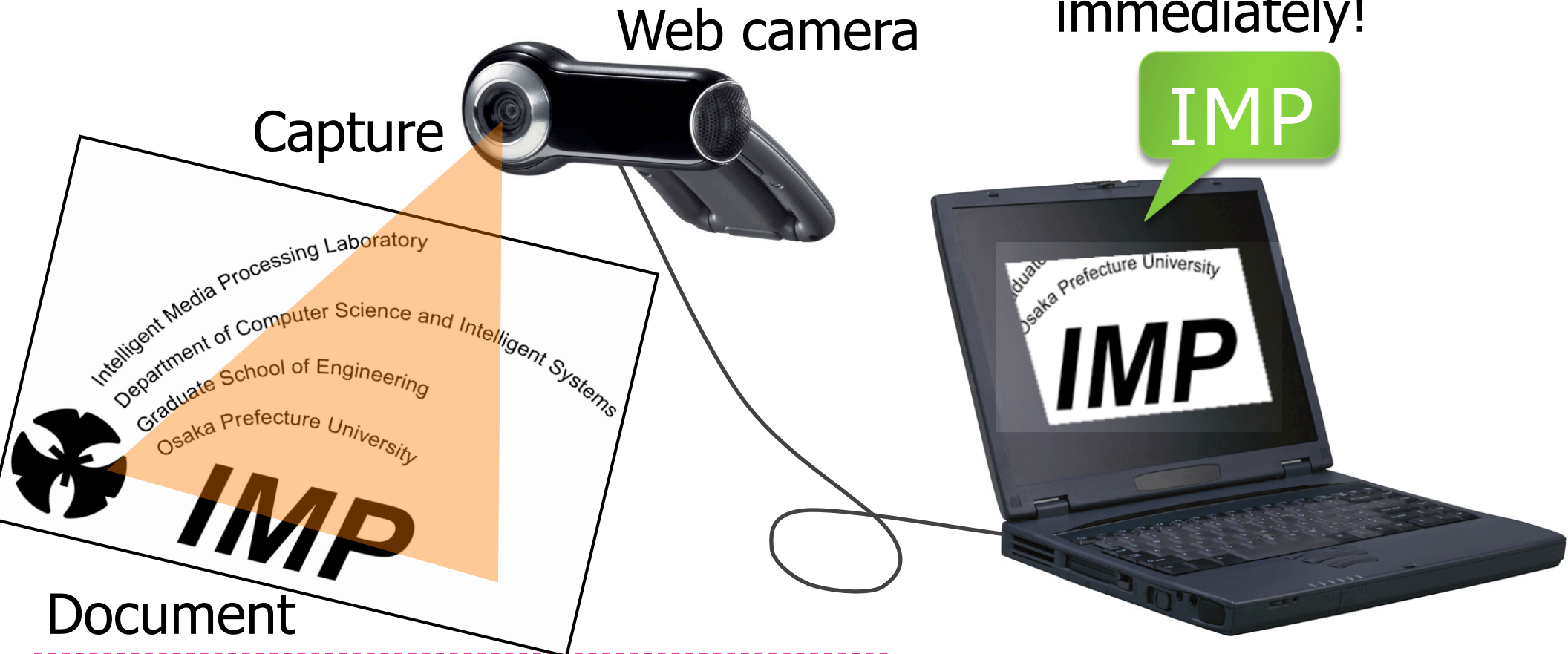
1. Background
  2. Overview of the Proposed Method
  3. Contour Version of Geometric Hashing
  4. Proposed Method
    1. Real-Time Processing
    2. Recognition of Separated Characters
    3. Pose Estimation
  5. Experiment
  6. Conclusion
- 



# Real-Time Camera-Based Character Recognition System

Recognizes ~200 characters/sec

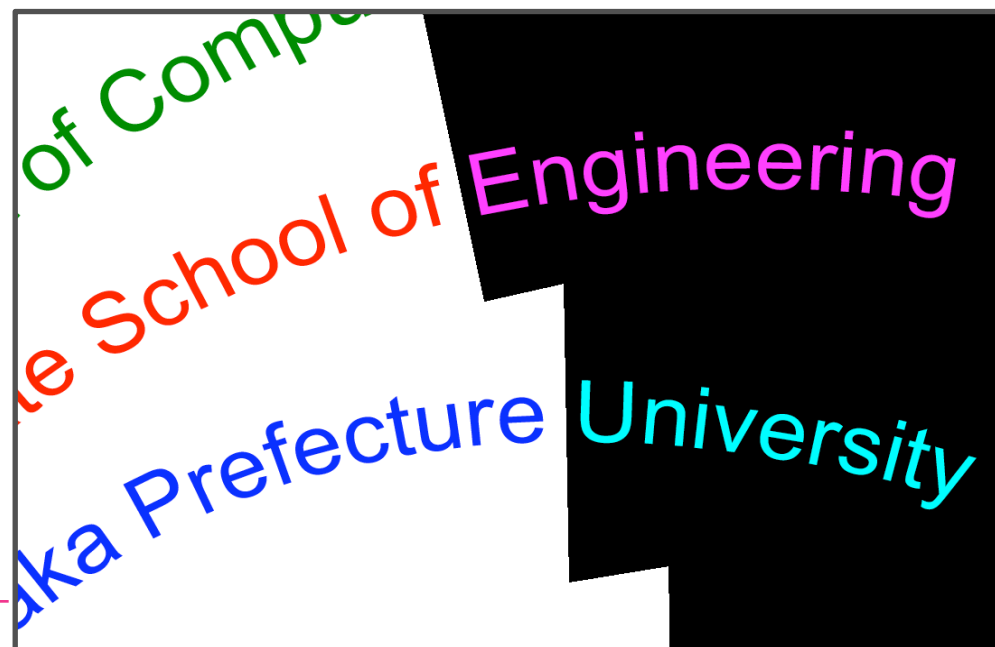
Recognizes character immediately!



# Future Work

---

- ▶ Recognition of Chinese characters
- ▶ Improvement of segmentation for
  - ▶ Broken connected components
  - ▶ Colored characters





# Real-Time Camera-Based Recognition of Characters and Pictograms

M. Iwamura, T. Tsuji, A. Horimatsu, and K. Kise