

# Shape Representation Robust to the Sketching Order Using Distance Map and Direction Histogram

Department of Computer Science  
Yonsei University  
Kiwon Yun

# CONTENTS

## □ Review Topic

## □ Proposed Method

- ❖ System Overview
- ❖ Sketch Normalization
- ❖ Spatial Information using Distance Map
- ❖ Direction Histogram
- ❖ Sketch Recognition

## □ Experiment

- ❖ Database
- ❖ Experimental Result

## □ Demo

## □ Academic Activity

## □ Q &A

# Review Topic

## □ The Goal

- ❖ Find shape representation method to recognize handwriting sketches drawn on Tablet PC

## □ Contribution

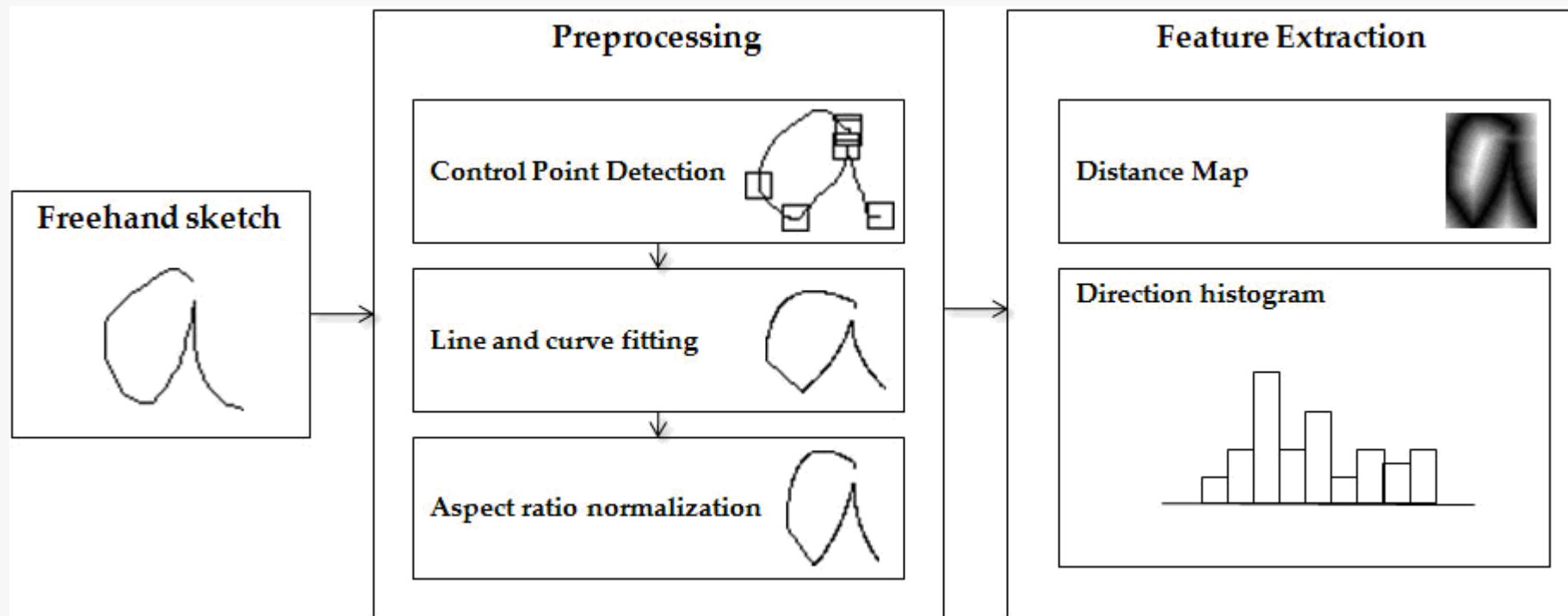
- ❖ Robust to the sketching order

## □ Proposed Method

- ❖ Spatial Information using Distance map
- ❖ Directional Information using Direction Histogram

# Proposed Method

## □ System Overview



# Proposed Method – Sketch Normalization

## □ Sketch Normalization

### ❖ The reason of normalization

- On-line handwritten sketches have large variation
- Need to remove noises

### ❖ Method

- Control Point Detection
- Line and Curve Fitting
- Aspect Ratio Normalization

# Proposed Method – Sketch Normalization

## □ Control Point Detection

### ❖ A Input Stroke

- $S_i$  is the a stroke
- $N_i$  is the total number of input points within a stroke  $S_i$

$$S_i = \{P_1, \dots, P_{N_i}\}, \quad P_i \in \mathfrak{R}^{2 \times 1}$$

### ❖ A Control Point

- $C_i$  is the control point
- It is the vertex at which dramatically change the direction, such as corner and curve

$$C_i \in S_i$$

# Proposed Method – Sketch Normalization

## □ Control Point Detection

### ❖ The dot product

1.  $S_i$  is the a stroke,  $N_i$  is the total number of input points within a stroke  $S_i$
2.  $C \leftarrow P_0$
3. For  $i = 1$  to  $N$

1. Compute the dot product

$$D(i) = \frac{PC_i^T NC_i}{\|PC_i\| \|NC_i\|}$$

$$\text{where } PC_i = P_i - C_P$$

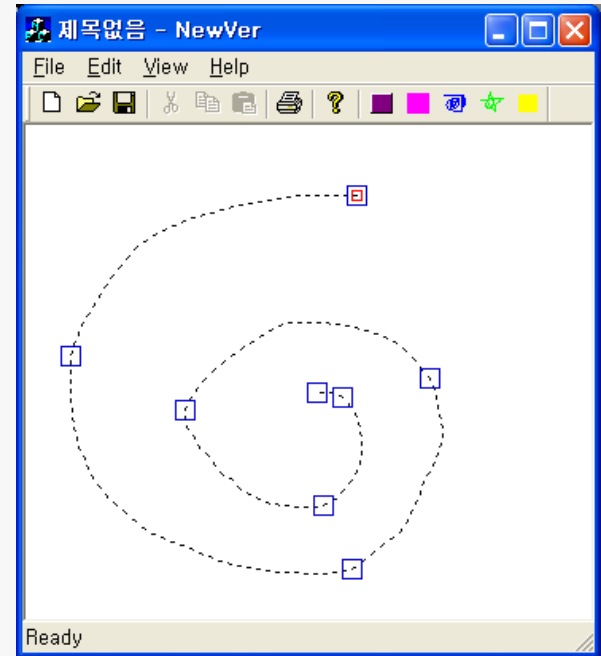
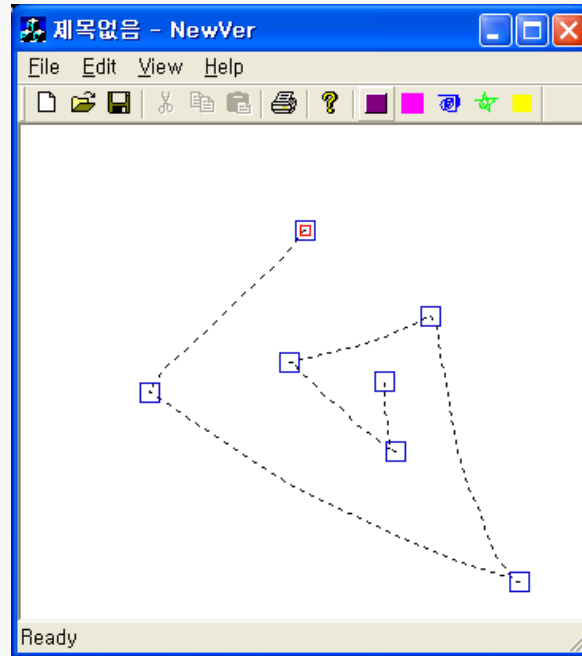
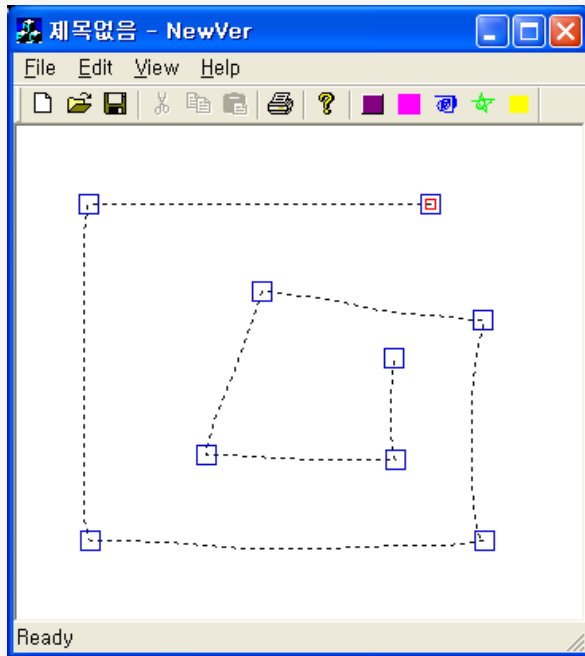
$$NC_i = P_i - P_{i+1}$$

2. If  $D(i) < \text{Threshold}$

$$C \leftarrow C \cup P_i$$

# Proposed Method – Sketch Normalization

## □ Control Point Detection





# Proposed Method – Sketch Normalization

## □ Line and Curve Fitting

### ❖ Method

- Classify each control point pair as a line or a curve

$$L(S_i) = \begin{cases} 1 & \text{if } \frac{1}{N_i} \sum_{i=1}^{N_i} \left[ \sqrt{P_i - \tilde{P}_i} \right] > \phi \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{P}_i = \Phi^T (P_i - P_1)$$

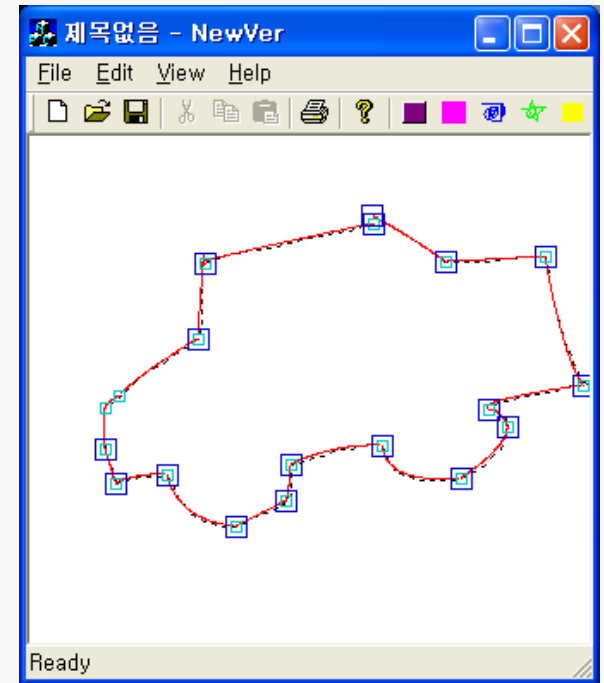
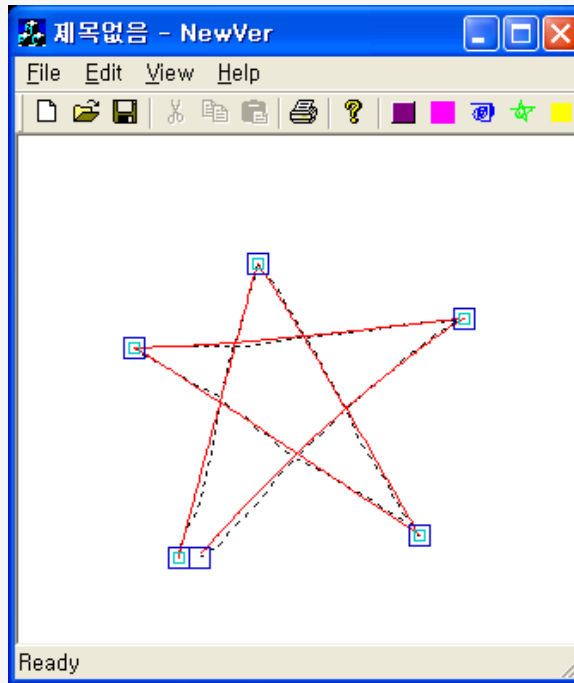
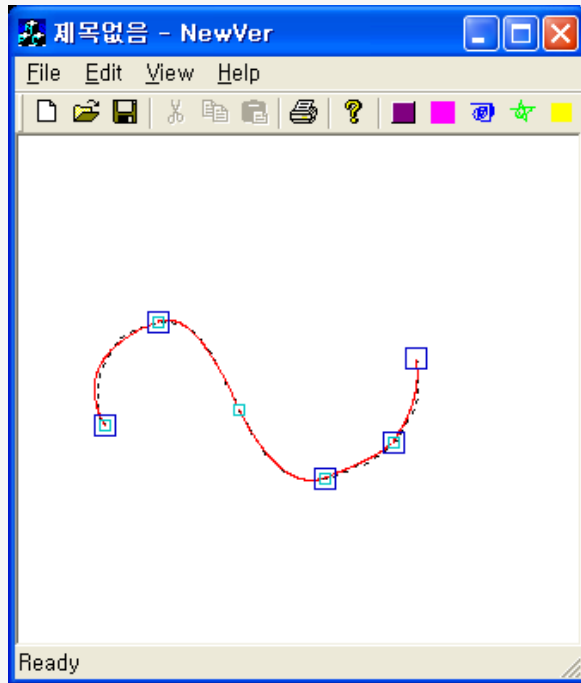
$$\text{where } \Phi = P_{N_i} - P_1$$

- Beautification

- ✓ Line : Draw line using the start and end point of the pair
- ✓ Curve : Draw curve using a quadratic Bezier curve

# Proposed Method – Sketch Normalization

## □ Line and Curve Fitting



# Proposed Method – Sketch Normalization

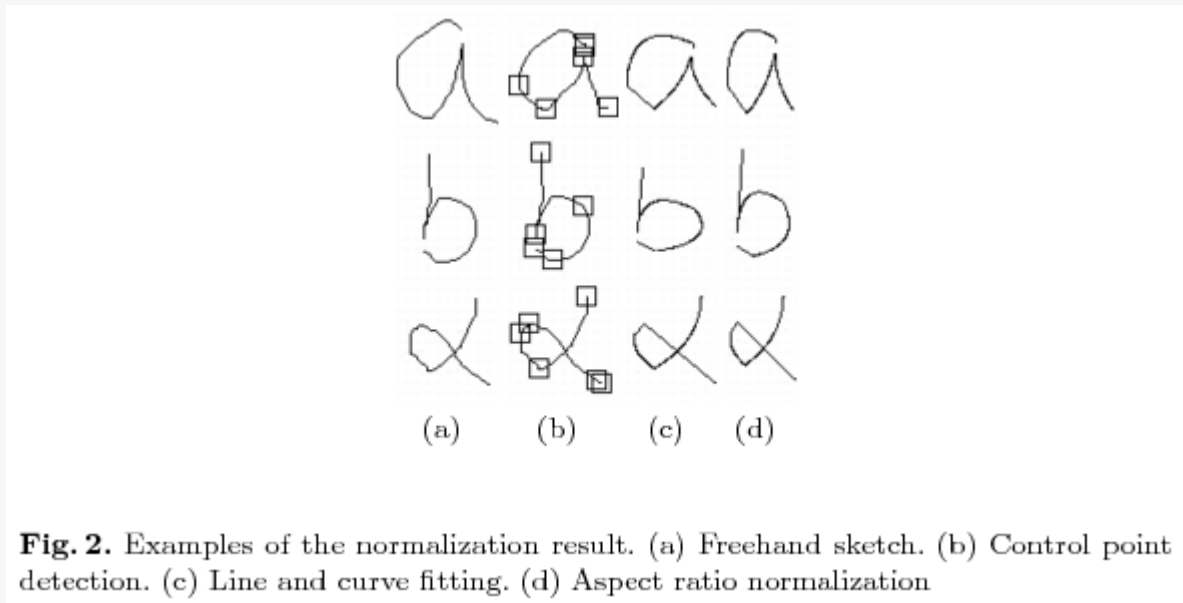
## □ Aspect Ratio Normalization

- ❖ To reduce the effect of different aspect ratio.
- ❖ Interpolate with the scale ratio which adjusts the width and height of sketch into  $m$  and  $n$
- ❖ Interpolated Sketch is defined as

$$\mathbf{I}(x, y) \in \mathfrak{R}^{m \times n}$$

# Proposed Method – Sketch Normalization

## □ The result of Sketch Normalization



# Proposed Method – Feature Extraction

## □ Spatial Information using Distance Map

- ❖ Represents how far each pixel is from edges of object
- ❖ Rich feature about spatial information of shape.
- ❖ Created by Distance Transform



Examples of distance map

# Proposed Method – Feature Extraction

## □ Distance Transform

- ❖ Start with zero-infinity image : set each edge pixel to 0 and each non-edge pixel to infinity.
- ❖ Make 2 passes over the image with a mask:
  - Forward, from left to right and top to bottom
  - Backward, from right to left and from bottom to top

For each position of mask on image,

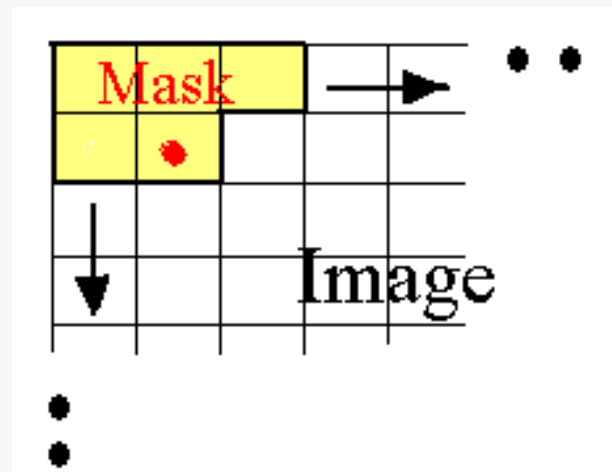
$$V_{i,j} = \text{minimum}(v_{i-1,j-1}+d2, v_{i-1,j}+d1, v_{i-,j+1}+d2, v_{i,j-1}+d1, v_{i,j})$$

d2	d1	d2
d1	0	

Forward Mask

	0	d1
d2	d1	d2

Backward Mask



Example of forward scan



# Proposed Method – Feature Extraction

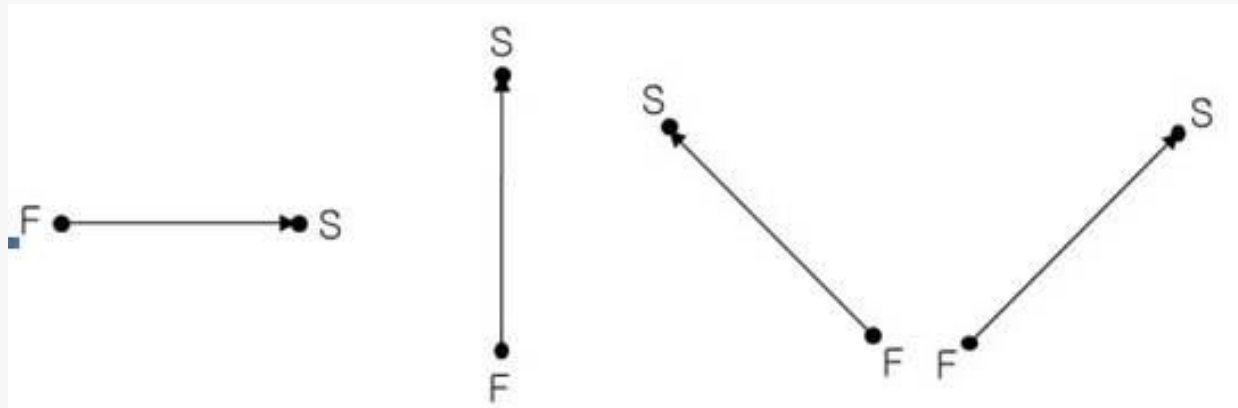
## □ Direction histogram

### ❖ The factor of consideration

- The same shape can be drawn with different order by different person

### ❖ How to solve this problem

- Align between neighbors of sampling points as the regular rule



(a) Horizontal

(b) Vertical

(c) Diagonal



# Proposed Method – Feature Extraction

## □ Direction histogram

- Compute direction

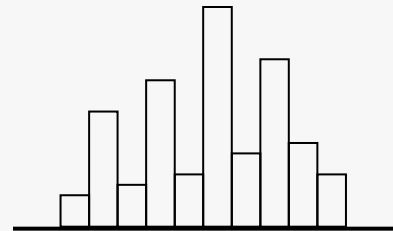
$$D(i) = \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) + \pi, 0 \leq D \leq 2\pi$$

- Angle quantizes as n level

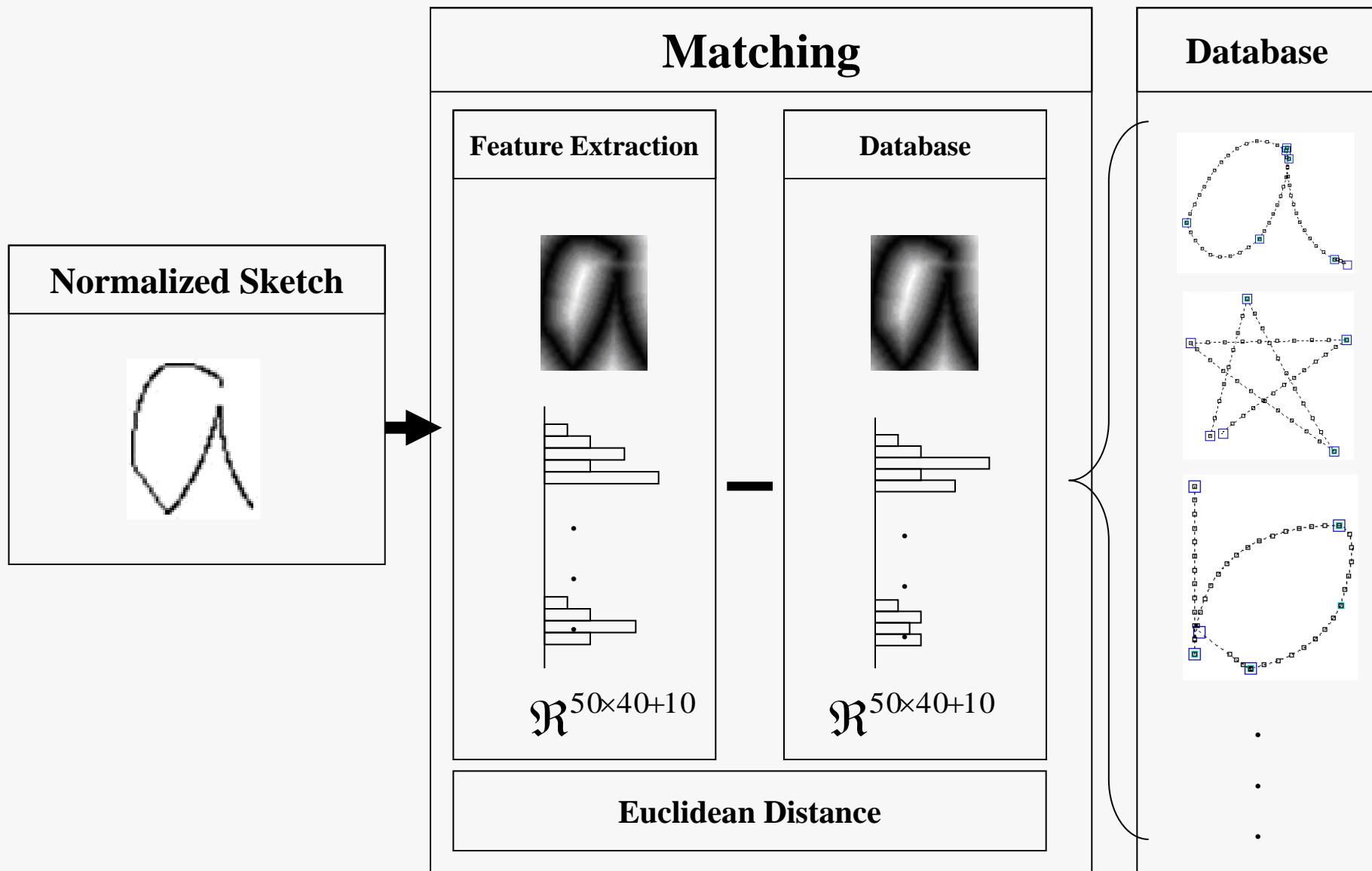
$$D(i) = \left[ \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) + \pi \right] / 18, 0 \leq D(i) \leq 9$$

- Make direction histogram

- ✓  $n \times 1$  dimension vector



# Proposed Method – Sketch Recognition



# Experiment - Database

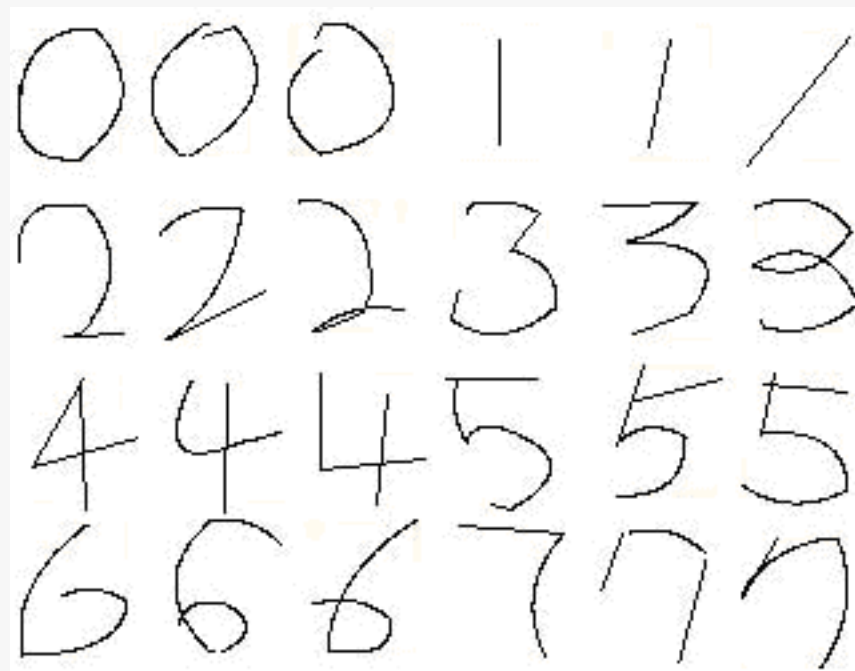
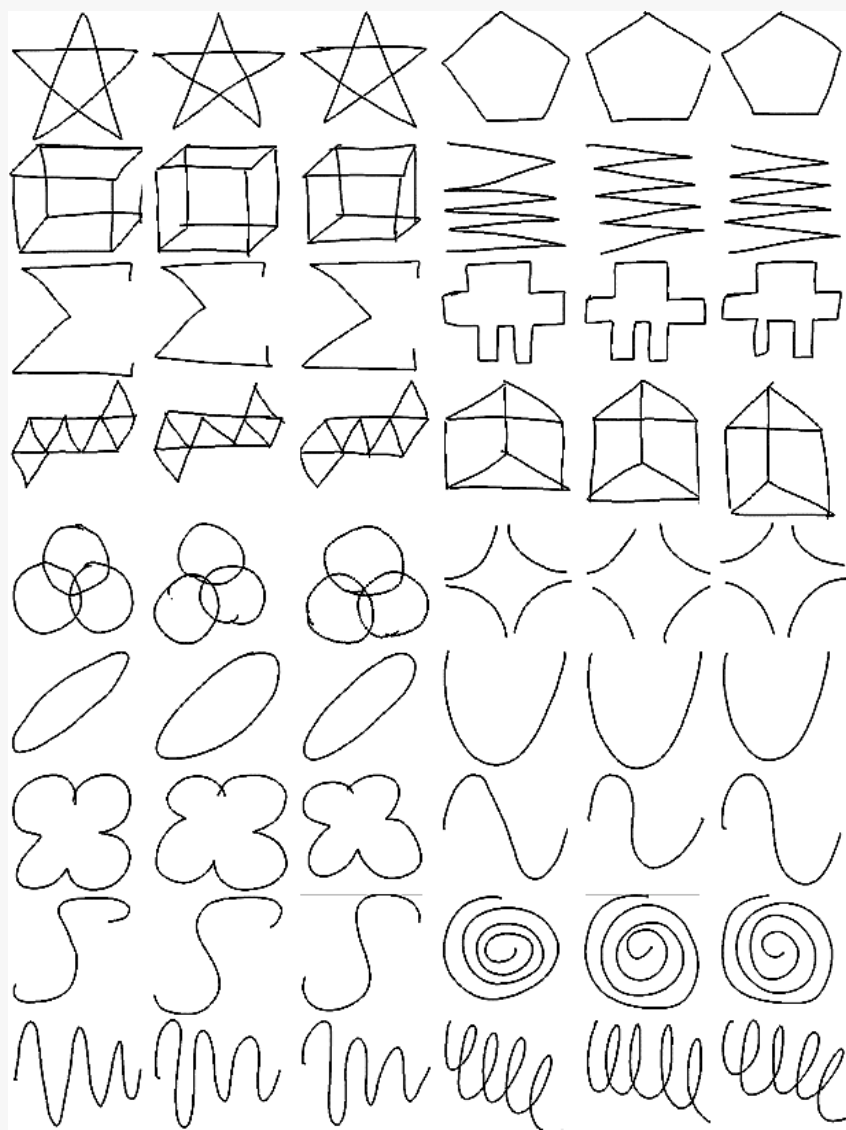
## □ Database 1 : Freehand sketch

- ❖ We give 28 shapes to 12 person
- ❖ let them draw the shapes by using the pen of Tablet PC.
- ❖ Each person repeatedly drew each shape 3 times
- ❖ Finally, we collected the total  $1008(28 \times 12 \times 3)$  shapes.

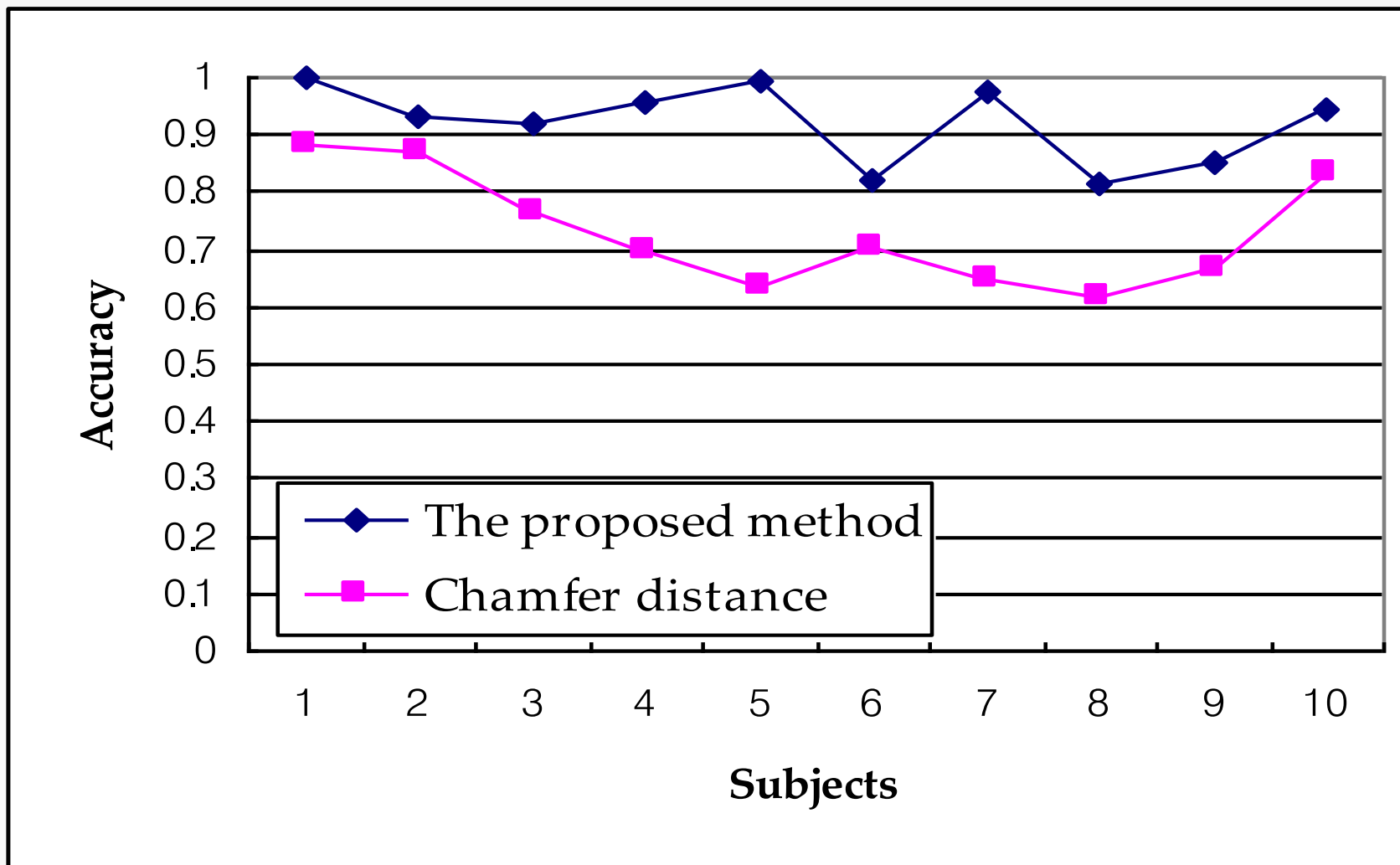
## □ Database 2 : Numeric symbol

- ❖ We give 0~9 number to 40 person
- ❖ let them draw the shapes by using the pen of Tablet PC.
- ❖ Each person repeatedly drew each shape 3 times
- ❖ Finally, we collected the total  $1200(40 \times 10 \times 3)$  shapes.

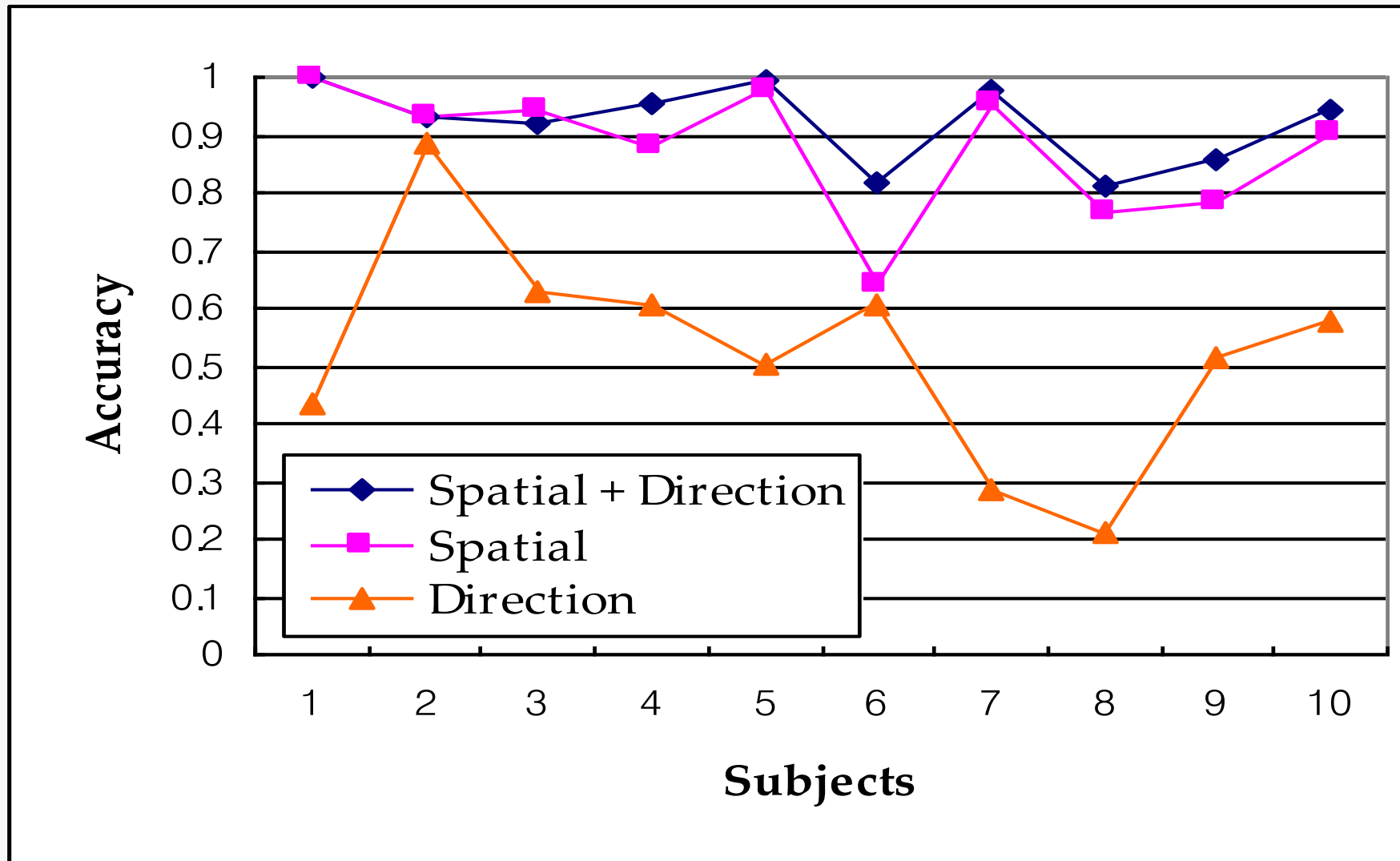
# Experiment - Database



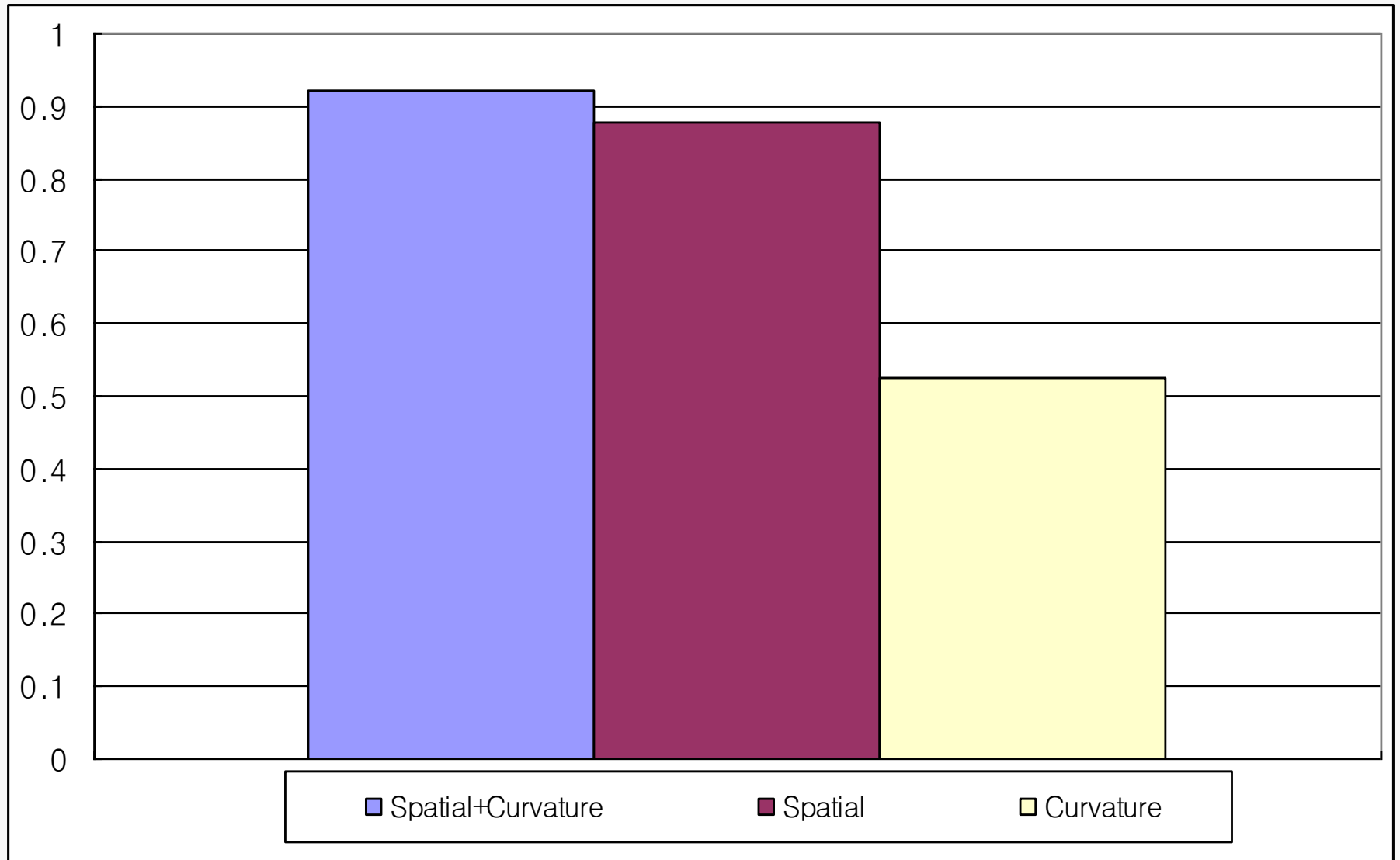
# Experimental Result : Shape recognition



# Experimental Result : Shape recognition



# Experimental Result : Shape recognition



# Experimental Result : Shape recognition

**Average recognition rate in our database : 96.0%**

**Table 2.** The result of experiment about all sketches in our database by different weight to spatial feature and directional feature. In this table,  $w_1$  is the weight to spatial feature, and  $w_2$  is the weight to directional feature. Also, all values are the average of recognition accuracy in each database. Bold text is the highest accuracy rate for different weight.

Weight	Freehand Sketches	Numeric Symbols	Both Sketches
$w_1 : 1.0, w_2 : 0.0$	0.93	0.85	0.89
$w_1 : 0.9, w_2 : 0.1$	0.96	0.85	0.91
$w_1 : 0.8, w_2 : 0.2$	0.97	0.87	0.92
$w_1 : 0.7, w_2 : 0.3$	0.97	0.89	0.93
$w_1 : 0.6, w_2 : 0.4$	0.98	0.90	0.94
$w_1 : 0.5, w_2 : 0.5$	0.98	0.92	0.95
<b><math>w_1 : 0.4, w_2 : 0.6</math></b>	<b>0.98</b>	<b>0.93</b>	<b>0.96</b>
$w_1 : 0.3, w_2 : 0.7$	0.97	0.91	0.94
$w_1 : 0.2, w_2 : 0.8$	0.97	0.88	0.93
$w_1 : 0.1, w_2 : 0.9$	0.92	0.72	0.82
$w_1 : 0.0, w_2 : 1.0$	0.64	0.52	0.58



# Demo

# Academic Activity

## □ **Submit Research Paper**

- ❖ **International Workshop on Structural and Syntactic Pattern Recognition (SSPR 2008)**
- ❖ **Orlando, Florida, USA, Dec 4-6, 2008**

## □ **Internship on Computer Vision & Pattern Recognition**

- ❖ **Participate in the project of sketch recognition with Microsoft**
- ❖ **Always, stay in the lab**
- ❖ **Participate in lab meeting & lab events**

# Q & A

□ Any questions?

**Thanks for your attention!**

**Department of Computer Science  
Yonsei University  
Kiwon Yun**