応用演習(1)
画像処理, 画像認識とその応用

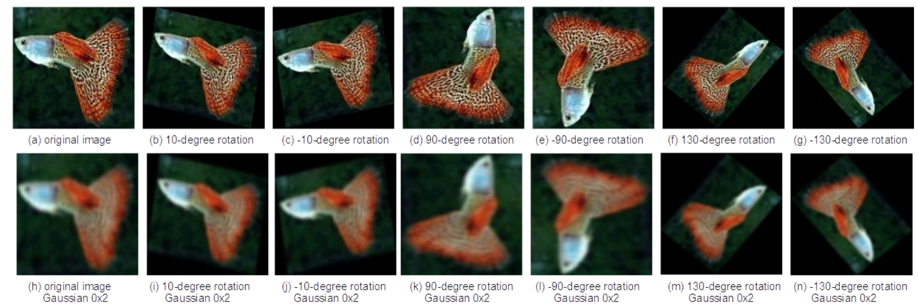
孟　林

2019/11/25

RITSUMEIKAN

画像処理と画像認識

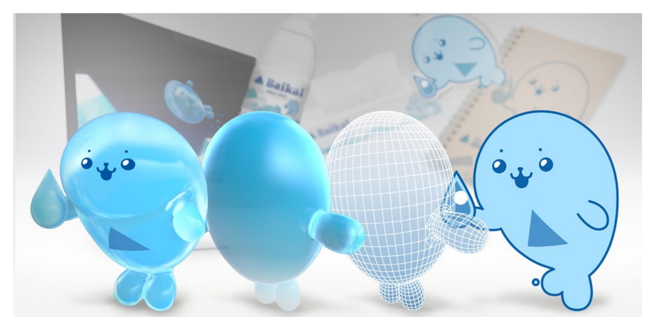
- 画像処理（Image Processing）
  - ➢画像を画像へ変換（処理）する。画質の改善、画像の変換



- 画像認識（Image Recognition）
  - ➢画像の特徴を抽出し、識別を行う

グッピー

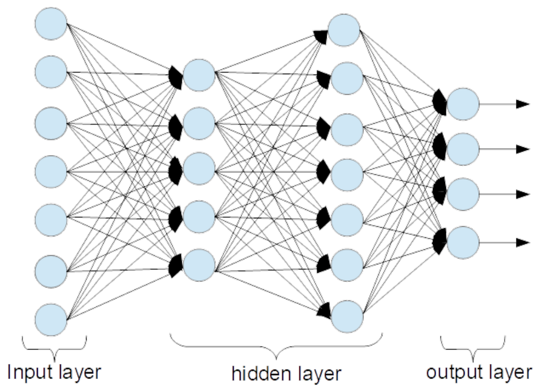
- コンピュータグラフィックス（Computer Graphics）
  - ➢画像を生成する

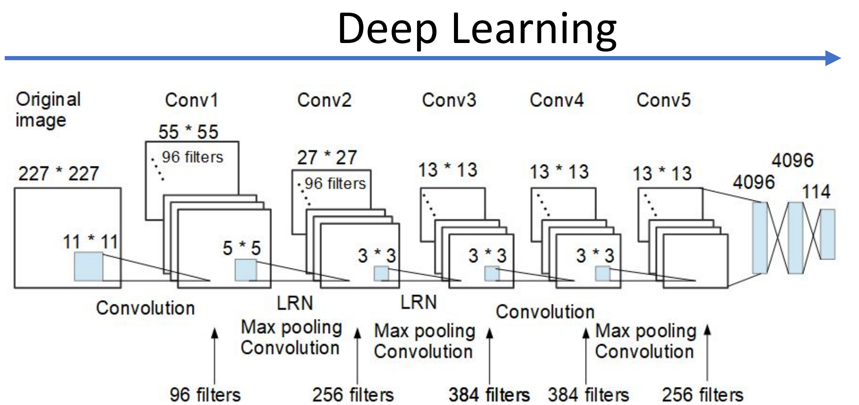


http://www.cg-makers.jp/service/character.html

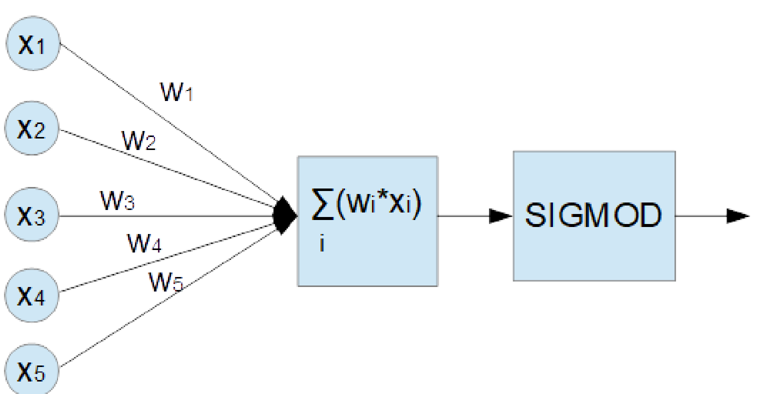
深層学習(Deep Learning)

多くの層を持ったニューラルネットワークモデルを用いた機械学習

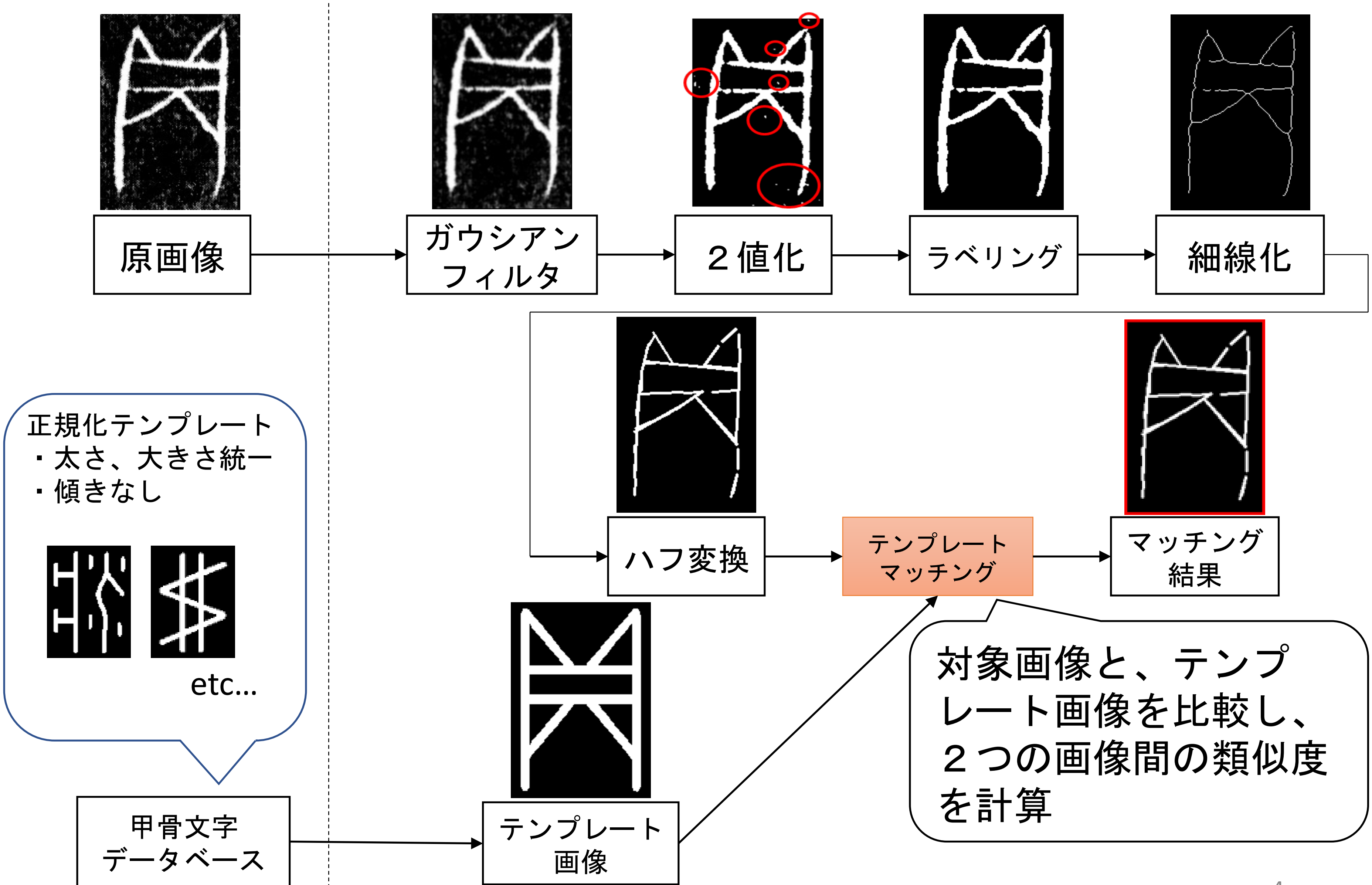




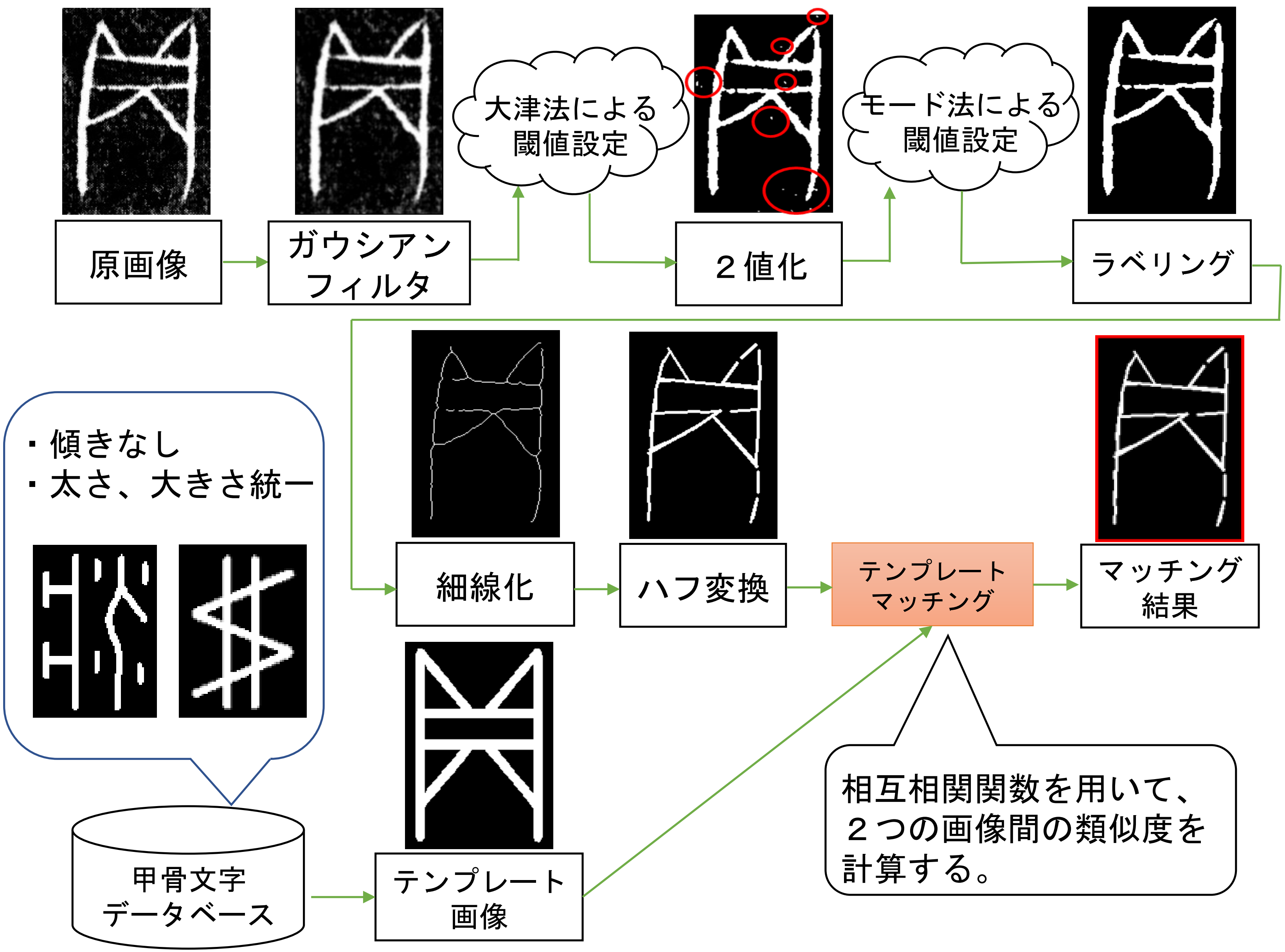
- 機械学習(Machine Learning)
  - ➢具体的な事例などから計算機に帰納的に学習させることを目的とする。



画像認識処理全体像（甲骨文字）



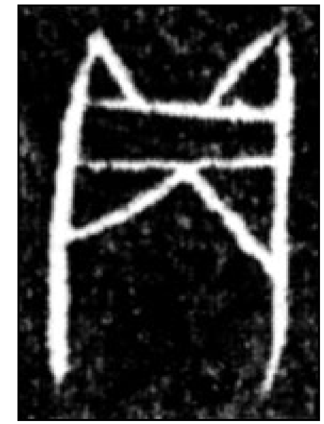
画像認識処理全体像(甲骨文字)



ガウシアンフィルタと２値化

- ガウシアンフィルタ
  - ガウス関数を用いて、画像の平滑化を行う

$$\text{Gaussian\_Filter}(x,y) = \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$$

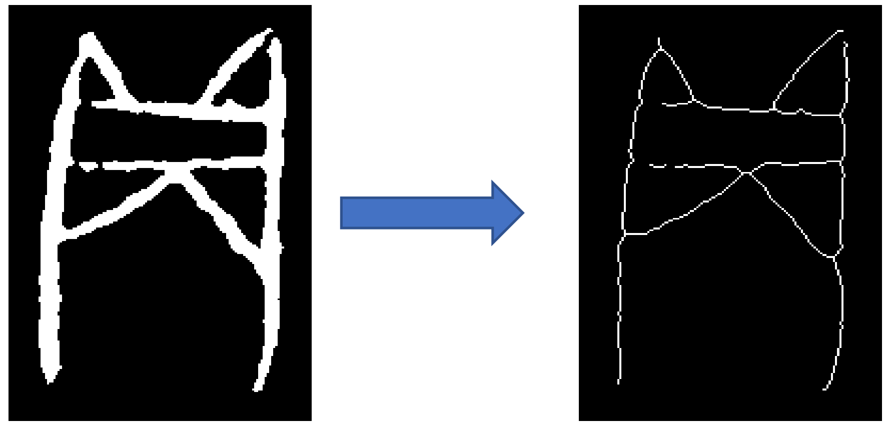


- ２値化
  - 画像を白黒にする

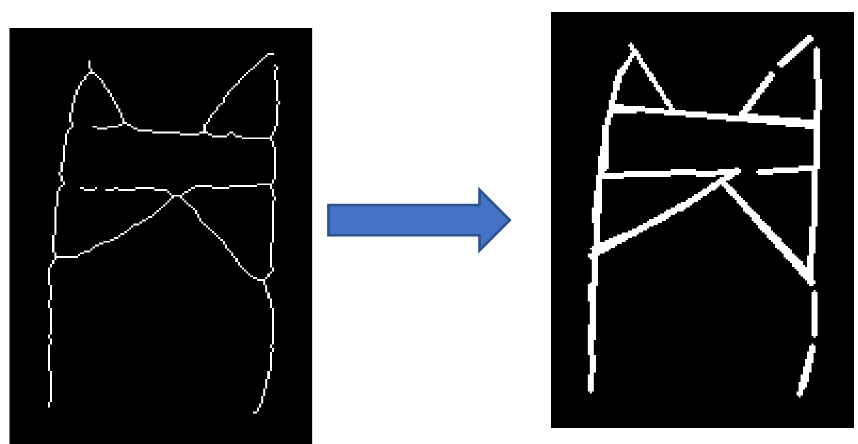
$$\text{Binarization}(x,y) = \begin{cases} 255 (Pixel(x,y) > Threshold) \\ 0 \quad (Pixel(x,y) < Threshold) \end{cases}$$

細線化とハフ変換

- 細線化
  - パターンに基づいて、２値化画像を線幅が１画素の画像に変換
    （田村法）

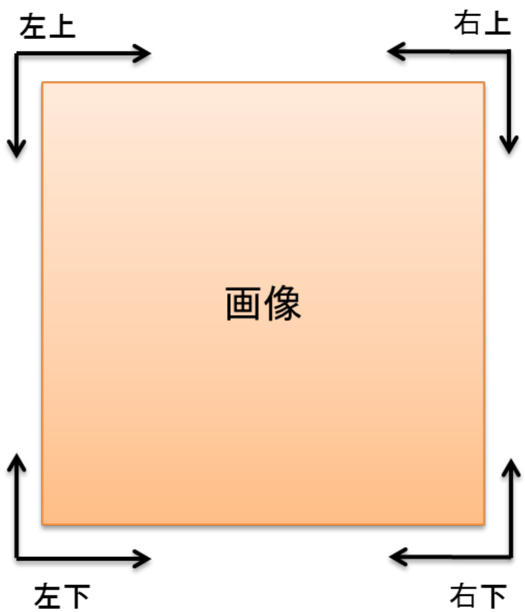


- ハフ変換
  - 直線を表す代数方程式を用いて、x-y空間座標から$\rho - \theta$極座標に変換し、直線を抽出する。

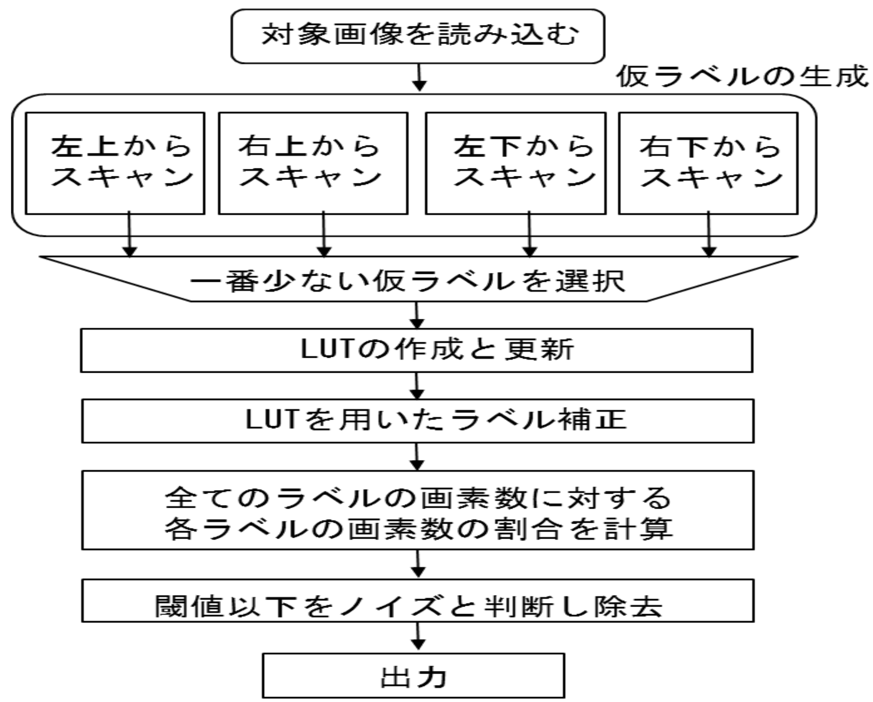


４方向ラベリング

- LUT（ルックアップテーブル）を用いたラベリングの問題
  - 仮ラベル数が多い場合、LUTの更新とラベル補正にかかる時間が長い
- 提案手法
  - 画像の４隅からスキャンを行い、仮ラベルが最少の方向を選択し、ラベル補正を行う

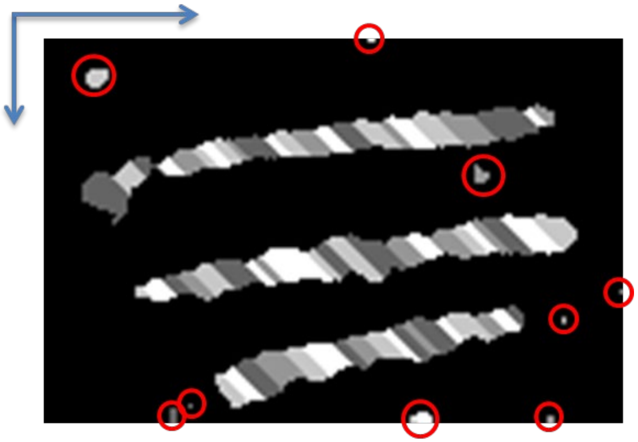


(a) ４方向スキャン



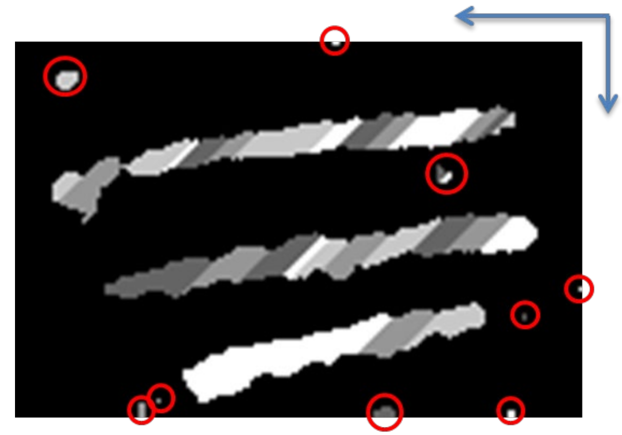
(b) 処理フロー

ラベリングの実験結果



仮ラベル数	75

1方向ラベリングでの仮ラベル生成



仮ラベル数	37

4方向ラベリングでの最少の仮ラベル生成

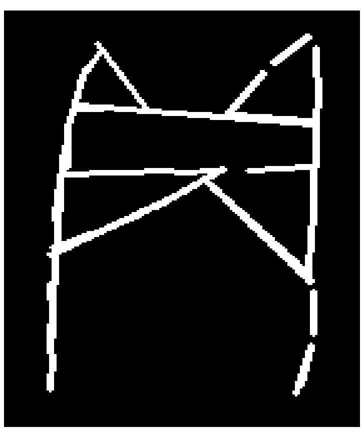
	1方向	4方向
除去したラベル数	9	
補正後ラベル数	3	
処理時間(ms)	1.110	0.613

ノイズ除去と処理時間比較

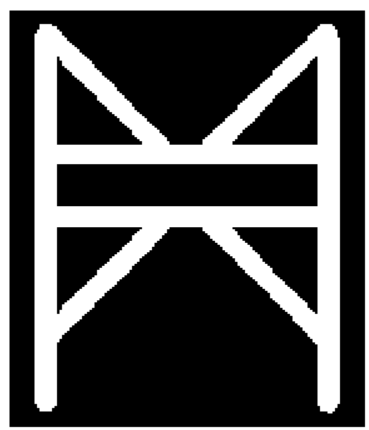
テンプレートマッチング

- 予め用意した既知の甲骨文字のテンプレートと、検出対象の画像を比較し、２つの画像の類似度を計算する
  - 正規化相関関数を使用し、２つの画像をベクトル表現し、ベクトル間の内角でマッチングするかどうかを判断する
  - 閾値（$cos\,\theta$）は0.7に設定する

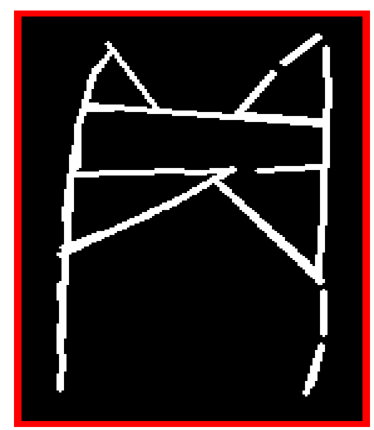
$$R = cos\,\theta = \frac{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(i,j)T(i,j)}{\sqrt{\sum_{j=0}^{N-1}\sum_{i=0}^{M-1} I(i,j)^2 * \sum_{j=0}^{N-1}\sum_{i=0}^{M-1} T(i,j)^2}}$$



ハフ変換後画像



テンプレート画像



マッチング結果

Two-Stage Recognition for Oracle Bone Inscription

Lin Meng
College of Science and Engineering,
Ritsumeikan University
Nojihigashi 1-1-1, Kusatsu, Shiga, Japan

RITSUMEIKAN

2017/09/15 @ ICIAP2017
Catania, Italy,

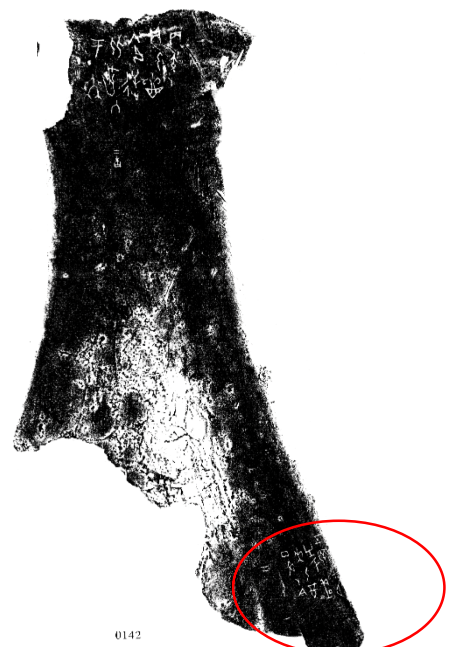
Outline

- Background and objective
- Research flow
- Experimental results
- Conclusion

Oracle Bone Inscriptions (OBIs)

- OBIs are one of the oldest characters in the world, which are inscribed on bone of cattle, turtle shells etc. about 3000 years ago
  - Aging causes these inscriptions are to be less legible

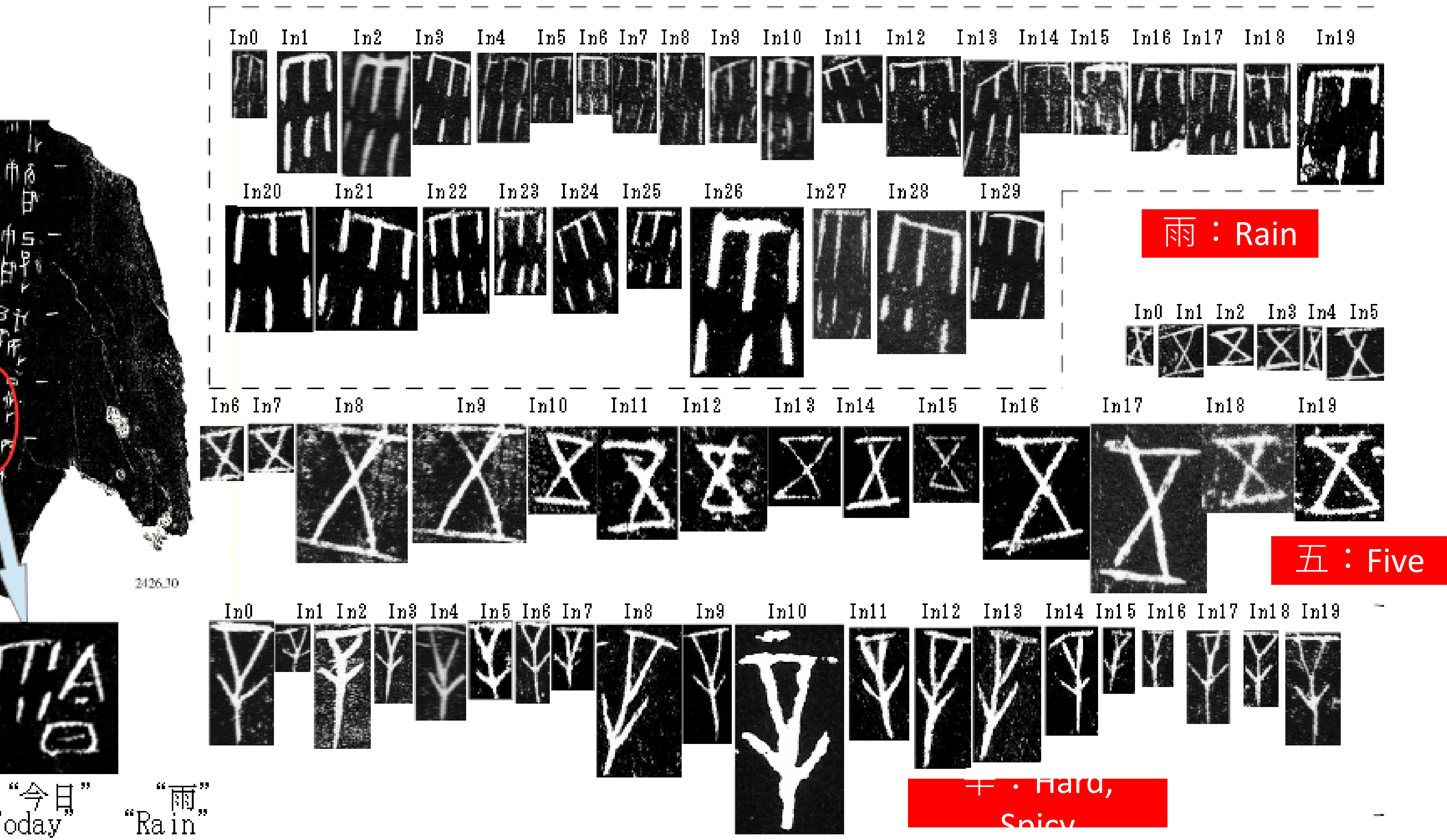
A photo



A rubbing

An expanded edition

Oracle Bone Inscriptions (OBIs)



sizes are uneven; noises exist, broken, non-uniform inclinations

Why recognize? What is difficult?

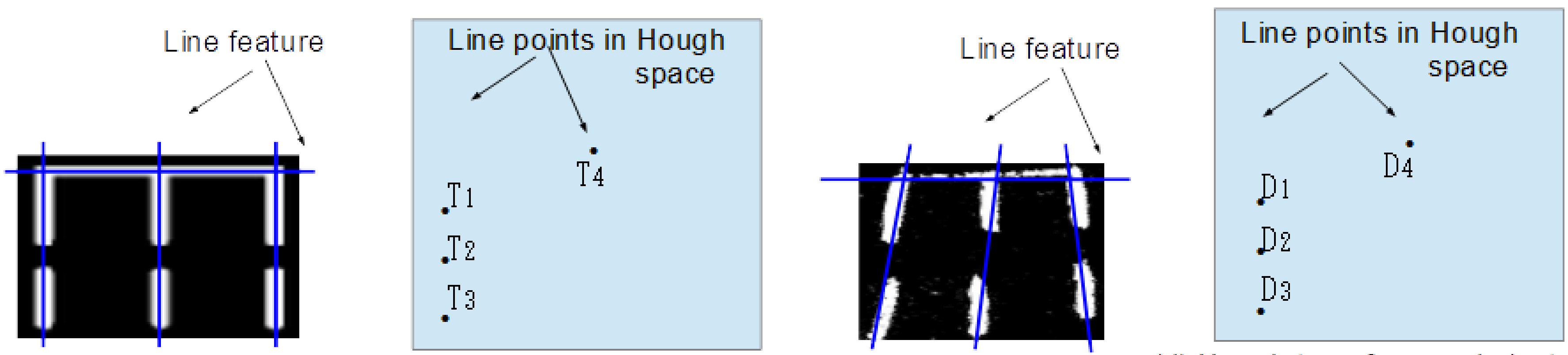
- OBIs were discovered in 1899
  - few papers describe them,
  - the aging process has made the inscriptions less legible.
  - Hence, understanding the inscriptions is important for researching world history, character evaluations, etc.
- What is the difficult
  - the character sizes are uneven
  - some smaller and bigger noises exist
  - some characters are broken
  - the inclinations of the characters are non-uniform
  - Hence, OBIs recognition is very difficult, and common OCR (optical character recognition) for OBI recognition is not easy.

Backgrounds and objectives

- Backgrounds
  - Reading the OBIs is very important to help understand the pre Christ era.
  - With the increasing of calculating ability, the computer are widely used in the archaeology by image processing
- Objectives
  - Do the inscription recognition by using image processing and the science of statistics.
  - Do the inscription recognition for the history research
  - Store and protect the inscriptions by the digital.

Image of proposal

- OBIs are described by sharp objectives and consist of lines
  - Extracting the line of OBIs
  - Calculating the distance of line points in Hough space.



(a) Template image

(b) Hough transform and clustering result for template image

(c) Original image

(d) Hough transform and clustering result for original image

$$Dis = \overline{T_1 D_1} + \overline{T_2 D_2} + \overline{T_3 D_3} + \overline{T_4 D_4}$$

When the characters are same
Dis = 0

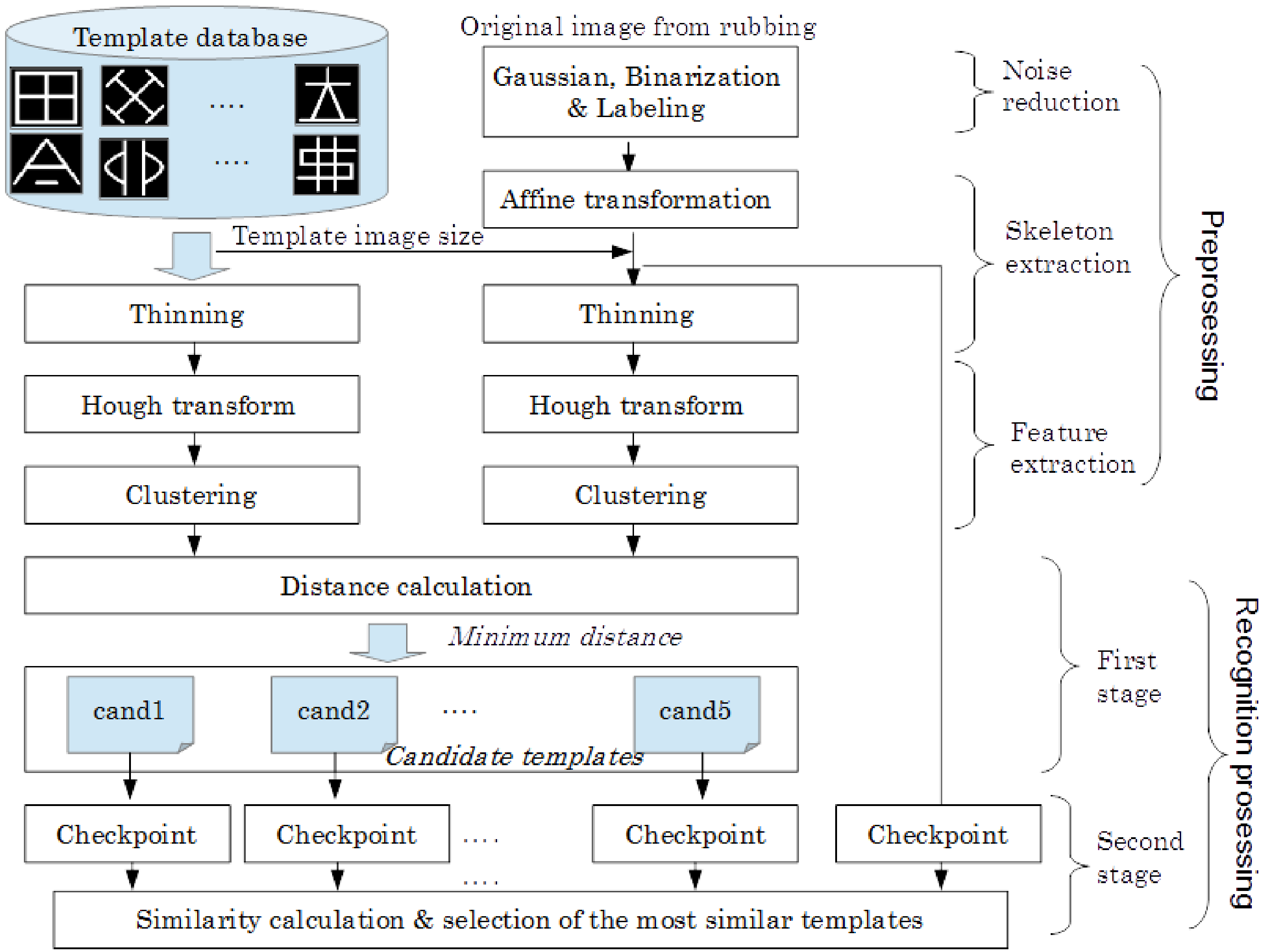
Recognition flow

Pre-Processing

- Noise reduction
- Skeleton extraction
- Line feature extraction

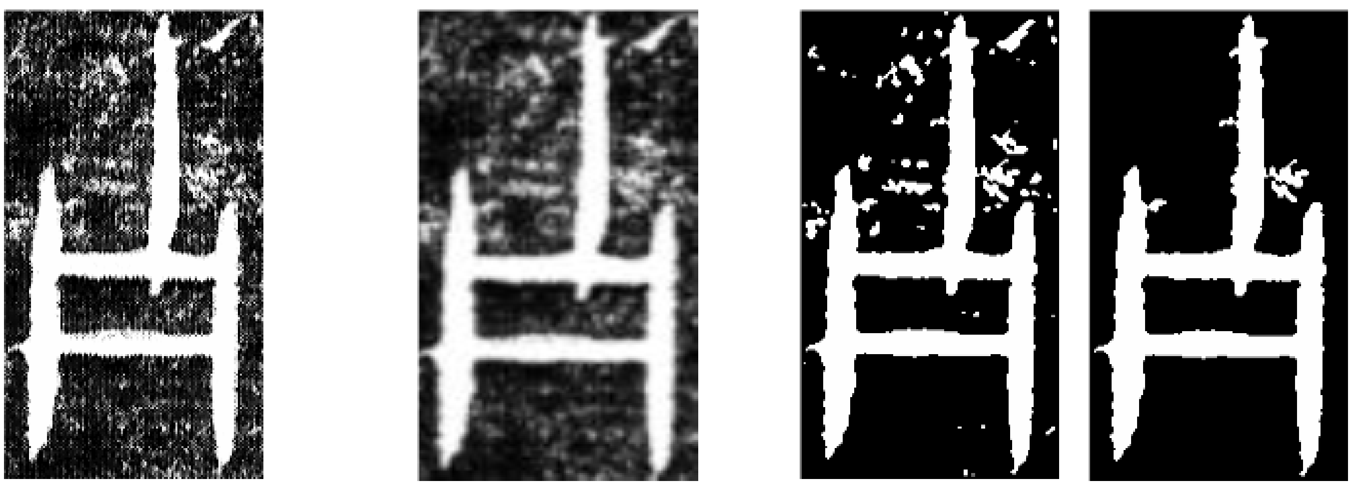
Recognition

- Distance calculation
- Checkpoints recognition



Noise reduction

- Gaussian filtering and binarization for reducing small noise
  - Otsu method for threshold decision
- Labeling
  - A histogram method to detect big changes in the histogram of objects for detecting the threshold.

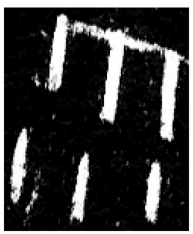


(a) Original image of “zi” (b) Gaussian filtering (c) Binarization (d) Labeling

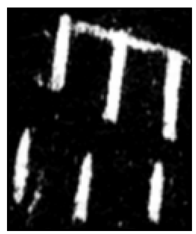
Noise reduction

Noise reduction

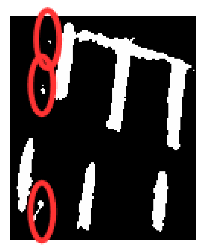
- Smaller noises reduction
  - Gaussian filtering & binariation; Threshold decision by Otsu Method
- Larger noise reduction
  - Labeling , threshold decision by Difference Histogram Method



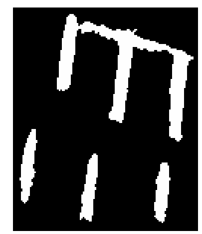
(a) Original image



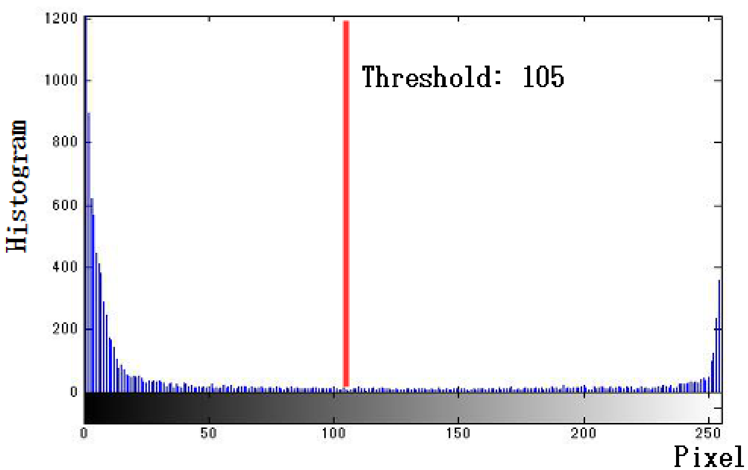
(b) Gaussian filtering



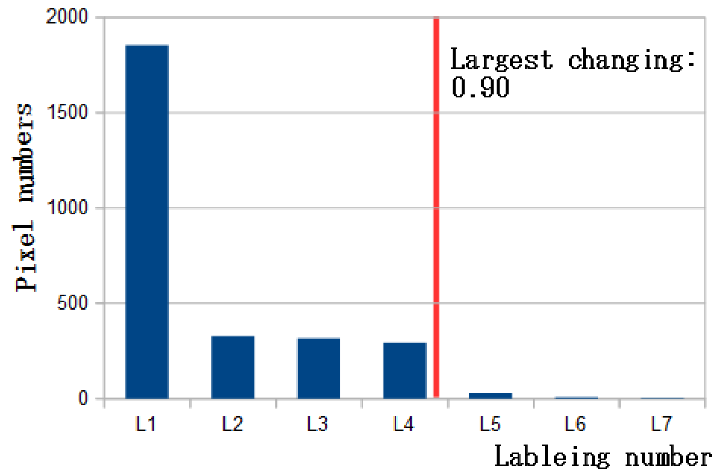
(c) Binarization



(d) Labeling



(e) Histogram of Gaussian filtering result



(f) Lableing histogram after sorting pixle

Skeleton extraction

- Normalization:
  - Affine transformation

- Thinning :
  - Tamura method

(a) Original image (b) Labeling (c) Template (d) Normalization (d) Thinning

Skeleton extraction

- Affine transformation
  - Transforming points and vectors in the original image space into points and vectors in the database image space

$$\begin{pmatrix} x_i^* \\ y_i^* \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} M \times x_j \\ M \times y_j \end{pmatrix} + \begin{pmatrix} x_c \\ y_c \end{pmatrix}$$

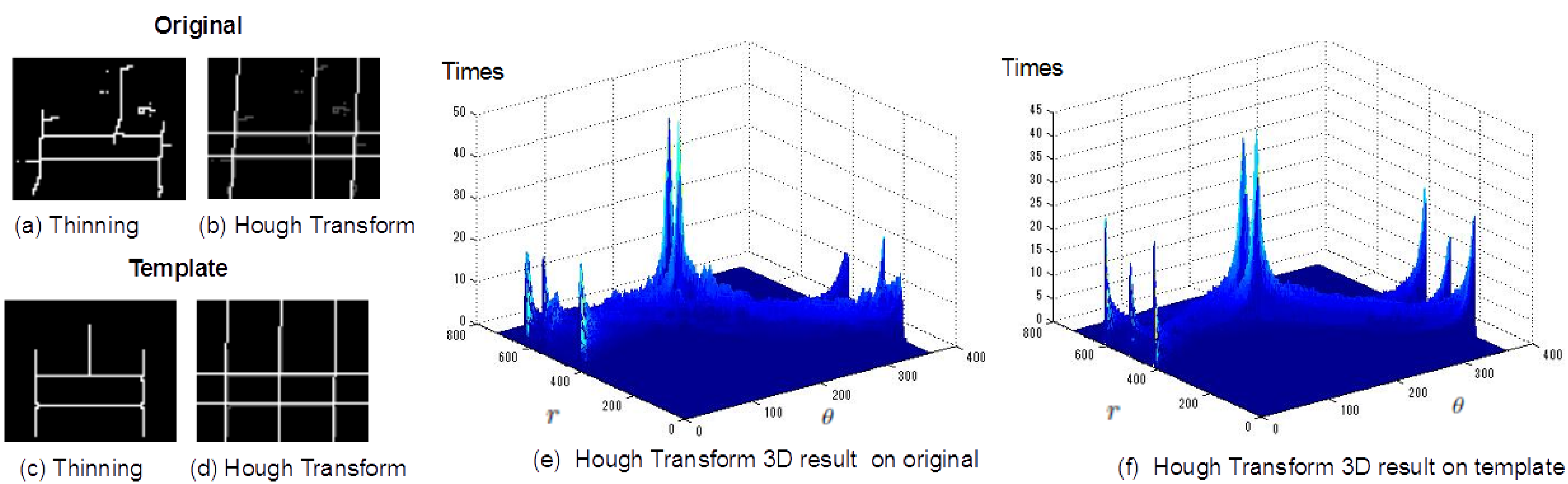
- Thinning
  - Considering each of the eight neigh-borhoods (p2,p3...p9) of the target pixel as one pixel (p1) and decides whether to peel it off or keep it a skeleton.

(d) Labeling (e) template image (f) Affine transformation (g)Thinning

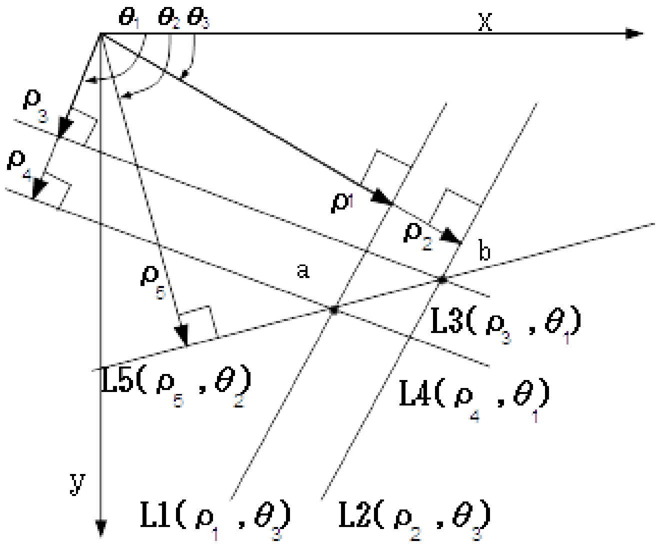
Skeleton extraction

Line feature extraction

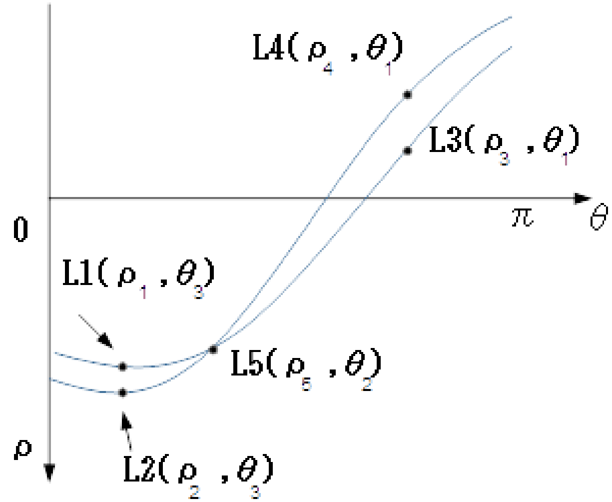
- Hough transform
  - A method for extracting the line by transforming the (x, y) space to the (ρ,θ)
- Clustering



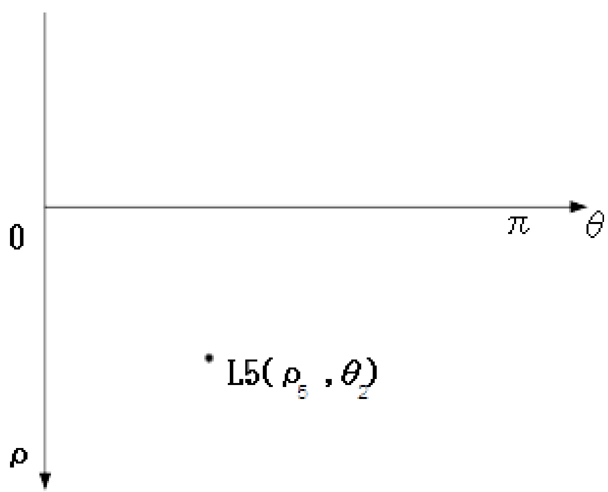
Hough transform



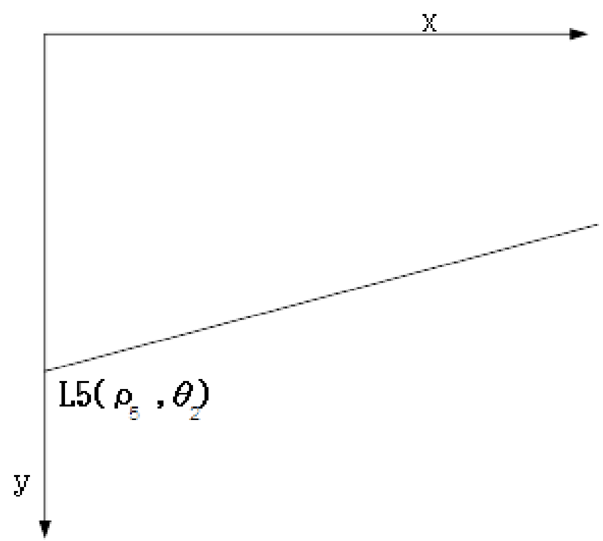
(a) (x,y) space image



(b) hough transform image (ρ, θ)



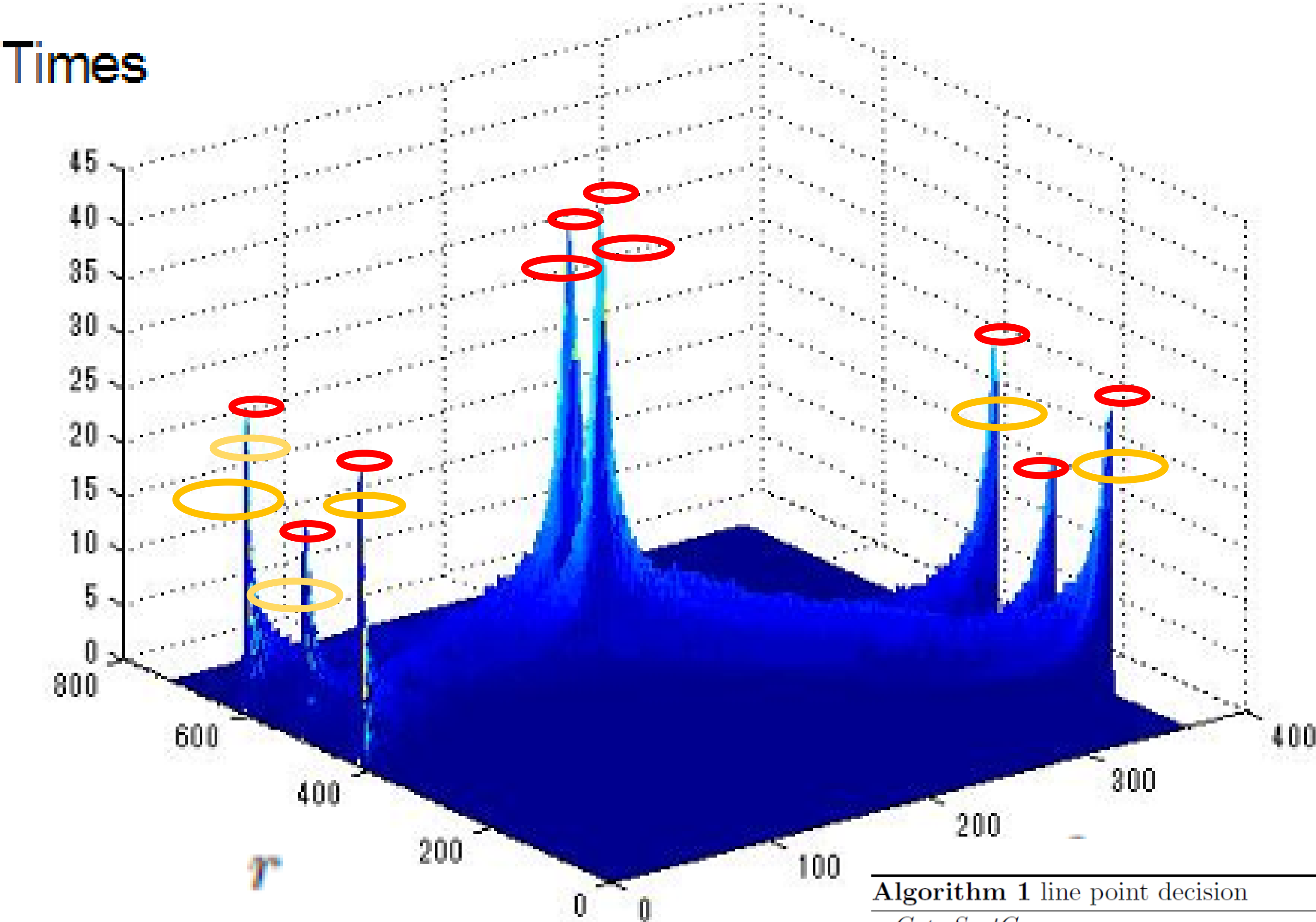
(c) line points is decided by larger point in(ρ, θ)



(d) inveser hough tranform results(x,y)

$$\rho = x\cos(\theta) + y\sin(\theta)$$

Clustering

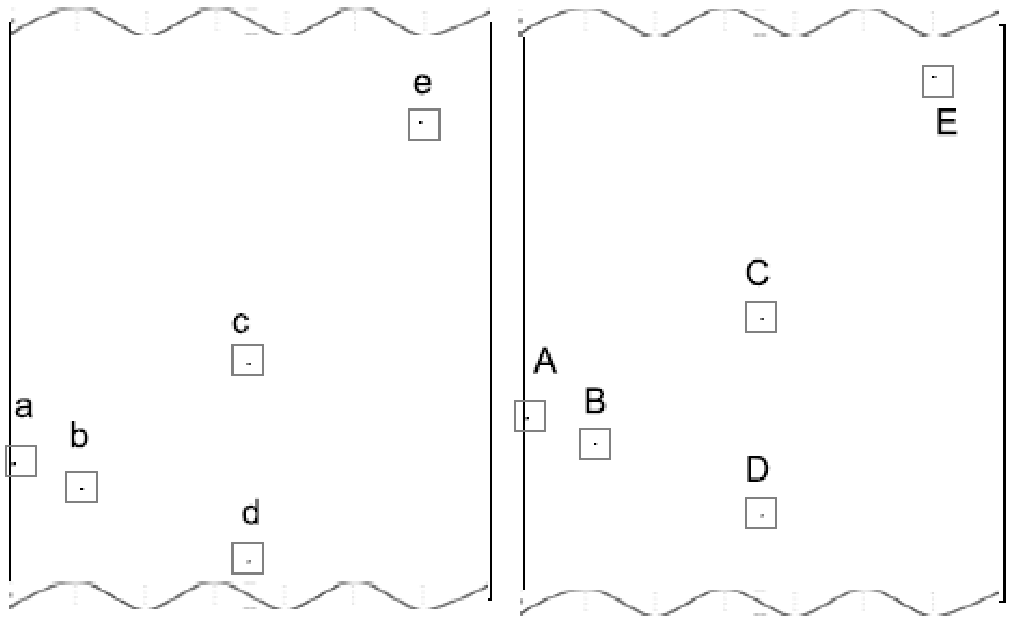


Center of cluster

Algorithm 1 line point decision

$C \Leftarrow SortC$
while $C \neq NULL$ **do**
  Search LC ; Generate $SDis$ by using LC and LPs, Search $MinDis$
  if $MinDis$ is lower than the radius of SLP **then**
    do nothing
  else $\{MinDis$ is lower than (the radius of SLP+30)$\}$
    record (the radius of $SLP \Leftarrow MinDis$) into LPs
  else
    input the LC into LP and keep the axis
  end if
end while

Two-stage recognition
---Distance calculation



(a) Line point clustering results of template

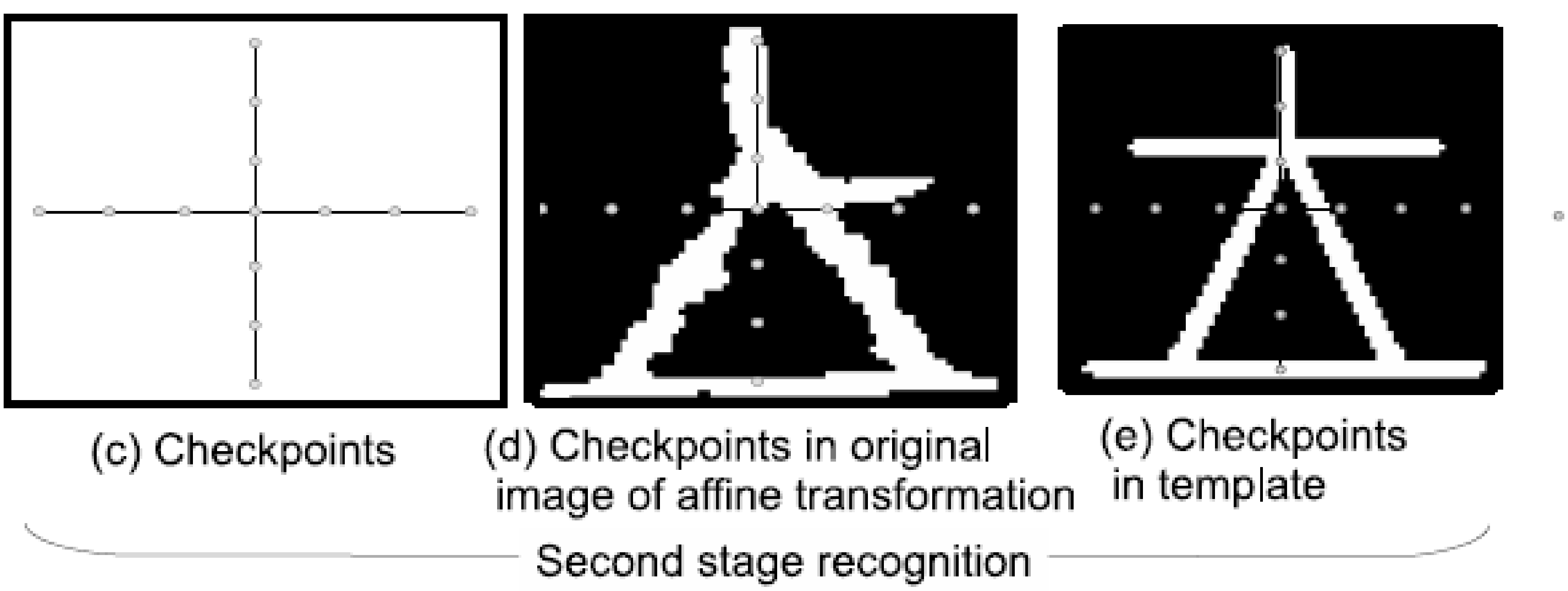
(b) Line point clustering results of original

$$Dis = \overline{aA} + \overline{bB} + \overline{cC} + \overline{dD} + \overline{eE}$$

$$NorDis = \frac{\frac{SumDis}{DisLines} \times MatchLines}{DisLines} = \frac{SumDis \times MatchLines}{DisLines^2}$$

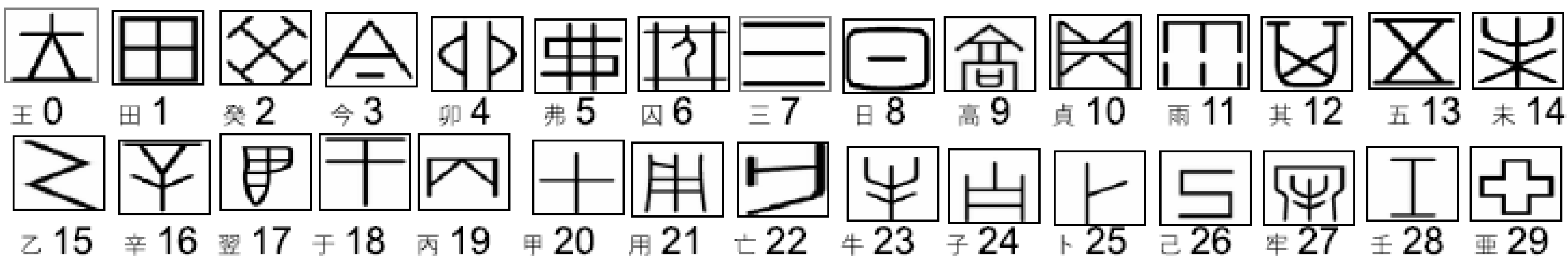
Two-stage recognition ---Checkpoints

we use the checkpoint hit rate for renewing the similarity rank and improving the recognition accuracy again.



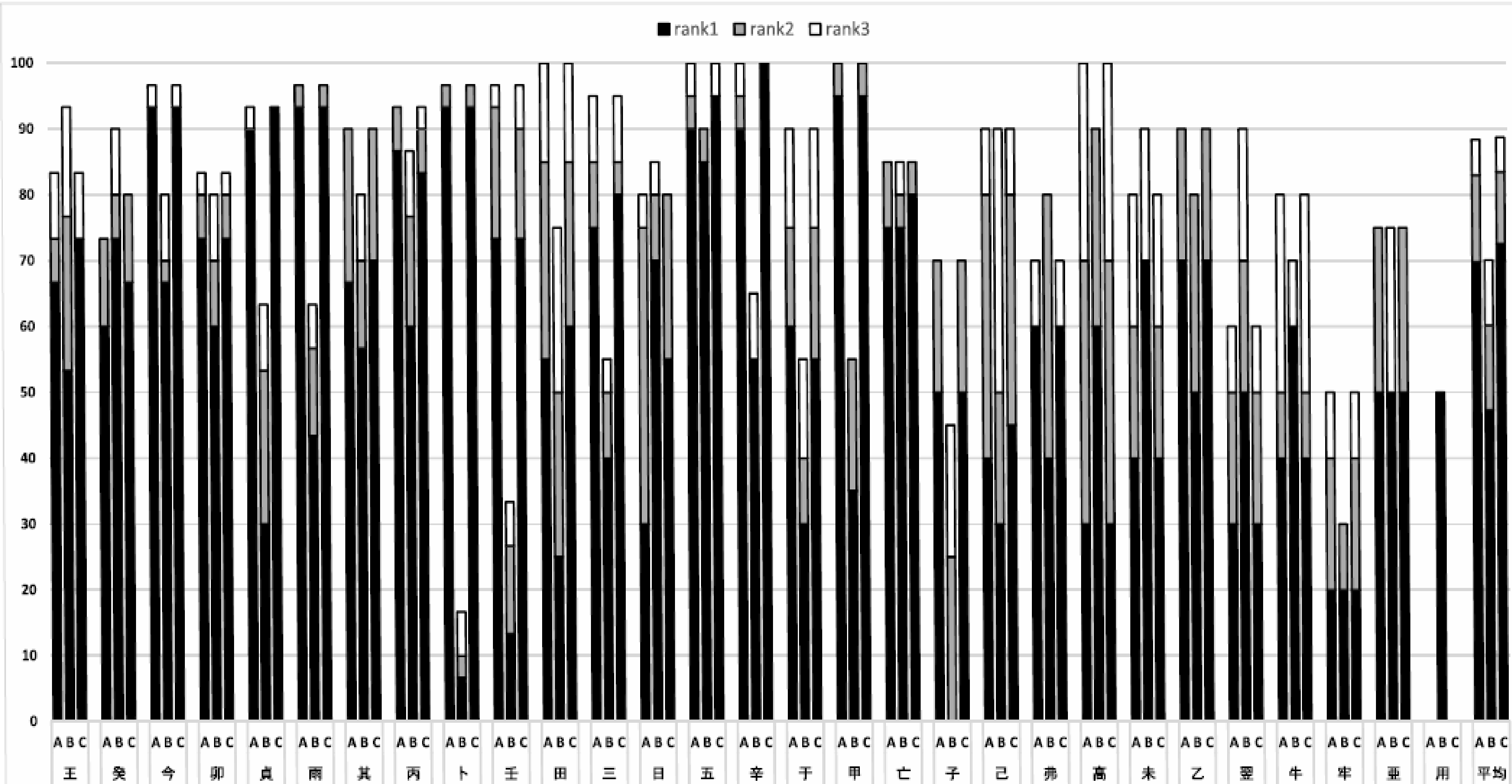
Experimental conditions

- We used 30 templates as the dictionary



- 29 kinds of original OBIs about 576 characters to measure the performance of our proposed method.

Experimental results



A: First stage recognition Rank 1 69.79%; Within rank 2 82.98%; Within rank 3 88.37%

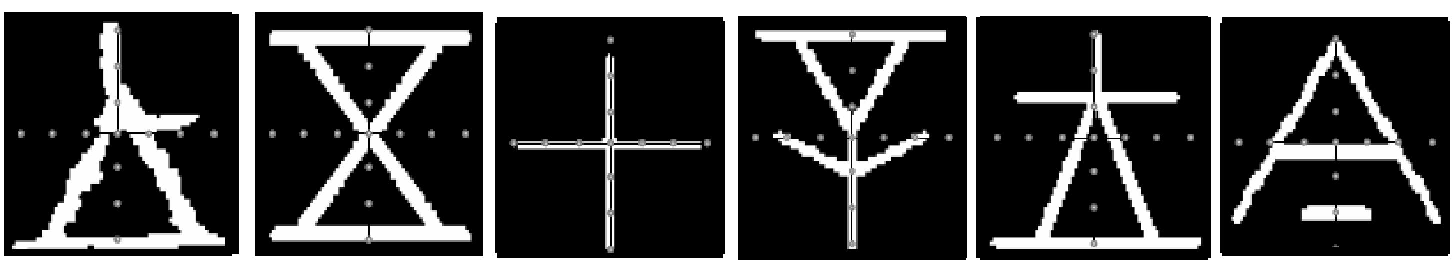
B: Template Matching Rank 1 47.30%; Within rank 2 60.17%; Within rank 3 70.06%

C: Two-stage recognition Rank 1 72.57%; Within rank 2 83.51%; Within rank 3 89.60%

Effective of two-stage recognition

Table 1. Recognition results of " 王 "

	First	Second	Third	Fourth	Fifth
In0	t0 (31.76)	t3 (40.36)	t4 (49.03)	t6 (54.60)	t23 (56.35)
In1	t0 (34.34)	t16 (46.33)	t3 (50.60)	t20 (51.15)	t13 (54.76)
In2	t0 (21.22)	t20 (35.64)	t3 (41.43)	t16 (41.68)	t24 (50.91)
In3	t0 (28.30)	t3 (50.71)	t9 (55.83)	t29 (57.69)	t16 (59.00)
In4	t0 (9.18)	t13 (52.27)	t13 (54.91)	t21 (56.26)	t16 (57.02)
In5	t0 (24.71)	t13 (43.06)	t3 (55.46)	t16 (58.91)	t29 (63.61)
In6	t0 (16.57)	t3 (41.63)	t13 (50.83)	t16 (52.04)	t29 (57.90)
In7	t13 (40.67)	t20 (41.56)	t16 (50.79)	t0(50.97)	t3 (51.86)
In8	t13 (38.15)	t0 (42.42)	t16 (45.07)	t28 (54.40)	t3 (60.63)
In9	t0 (11.86)	t3 (40.97)	t28 (45.81)	t13 (51.62)	t22 (56.05)



(a) Target OB
(b) Template in Rank 1 of first stage
(c) Template in Rank 2 of first stage
(d) Template in Rank 3 of first stage
(e) Template in Rank 4 of first stage
(f) Template in Rank 5 of first stage

Fig. 10. Recognition analysis of "王" in second stage.

Recognition analysis

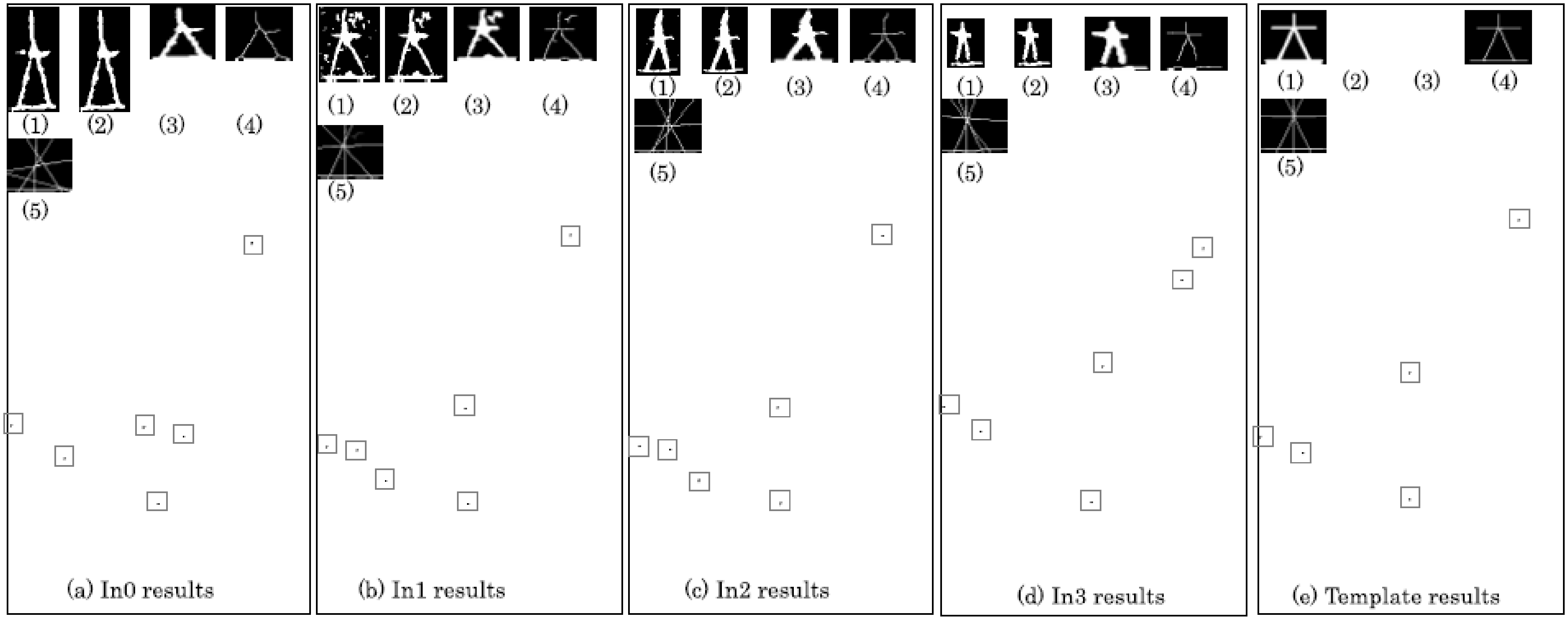


Fig. 9. Recognition analysis of ”王” in first stage.

- Figure 9 (a) shows that parts of the OBIs are broken, and in (b), there is noise still exist.
- However, the cluster extracts the line feature correctly and the inscriptions are recognized.

Conclusion

- Conclusion
  - We propose a two-stage recognition system that consists of line points and check-point recognition
  - In total almost 90% of inscriptions were recognized on the third-most similar templates.
  - The proposed method can recognize inscriptions well, even if noise exists, the original inscription images were tilted and broken, and the whisker exists in the thinning results.
- Future work
  - Analyzing the reason for failure and increasing the recognition rate are major future work.
  - Increasing the dataset of templates and original OBIs will be an important future works too.

- 課題（C言語を用いた画像処理）
  - (1) 与えられた画像の読み込みと書き込み
  - (2) 与えられた画像複数のフィルタをかけ、結果を考察する
    - ガウシアンフィルタ、Sobelフィルタ、
  - (3) 画像のサイズ変更
    - 2倍に拡張する、1/2倍に圧縮する
  - (4)発展問題
    - ラベリングを行い、画像の中の要素数を数える