TR-86-003

Complex Chinese Character Image Recognition and Walsh Transform

印刷複雜中文字照像識別及瓦氏轉換

黃俊雄　　鐘馬龍
Jan S. Huang and Ma-LungChung
Institute of Information Science
Academia Sinica
Taipei 11529, Taiwan
Republic of China

參考書
不外借

Januany 1986

Abbr. of title: Chinese Character Image Recognition

Mailing address:Jan S. Huang
Institue of Information Science
Academia Sinica
Taipei 11529, Taiwan
Republic of China

中研院資訊所圖書室
3 0330 03 000055 3
0055

摘　要

印刷中文字照像識別很有實用價值而且可以製成產品，因爲識別率可達百分之九十九點九。一般識別法是用四角碼法及四邊碼法來做大分類，共可分成4096類。對常用漢字5401字來大分類，由於有雜訊（由字之大小，不正等因素產生）而使同一字可能落在不同兩大類或三大類。因此大分類結果平均每類有一字到六字。通常爲相似字集在一起。進一步把每一類細分而認出每一字是要花點功夫。簡單字好認複雜字難認。這裏提出用瓦氏轉換來細分至認出每一字。瓦氏轉換與常用傅氏轉換不同，後者有週期性，而前者有序列性很適用文字特性。實驗時，用明體字（常用率百分之八十六），每字大小不同照像共八次，再計算瓦氏轉換，實驗用的複雜中文字有三組：(1)蘇、蔣、藏、蔽、藥，(2)圍、圓、國、圈、圖、園，(3)朦、鵬、臘、曠。針對某一組我們只要由45個瓦氏係數中挑出區分力量最好的三、四個係數就能成功的把該組細分至每一字。

ABSTRACT:

Typed (machine printed) Chinese character recognition is practically feasible and the recognition rate can be as high as 99.9%. The first step of the recognition procedure is to use 4C code and 4P code to partition the commonly used 5401 characters. 4C code is defined by encoding four corner zones of a character, each into two levels, and 4P code is defined by encoding four peripheral rectangular zones,each into 32 levels according to the number of points having some particular runs of black and white. In this way we can obtain approximatly 4096 classes, most containing 1 to 6 characters. Characters with similar peripheries are grouped together. Here we use Walsh transform to seperate these similar characters in each class. The Walsh transform has 'sequency' instead of "periodicity" that Fourier transform has, and Walsh transform is also easy to calculate. In our experiment, we use three groups of complex Chinese characters (Ming font), each containing 4 to 6 characters. Each character is imaged 8 times by changing its size, position and thresholding value. We find that most Walsh coefficients are stable under these changes. Thus we pick up 2 to 5 coefficients that have most seperability power, and we are able to use these coefficients to recognize each character in each group. This shows Walsh transform is a simple, fast and reliable method for seperating complex Chinese characters with similar peripheries.

KEY WORDS: Chinese Document Processing, Similar Characters,
Walsh Transform, Character Recognition.

1. INTRODUCTION

Typed Chinese character recognition is practically feasible and because the recognition rate can be as high as 99.9% its technology now has reached the stage of product development and the product will soon appear in the consumer market. There are many papers discussing typed Chinese character recognition (see [1-3]). Here we try to develop a digital document processing machine [4] with the aim to recognize the Chinese documents, so that editing, compression and basic text understanding can be achieved. Thus in the first stage we segment a sequence of typed Chinese characters in a document and use the conventional 4C code and 4P code [3] to partition the commonly used 5401 Ming font characters. 86% of the Chinese machine printing uses Ming font character in Taiwan. The 4C code is defined by encoding four corner square zones of a character, each into two levels according to the size of black points. The 4P code is defined by encoding four peripheral rectangular zones, each into 32 levels according to the number of white points having some particular runs of black and white. These codes are somewhat stable under the variations due to document segmentation and the size change of character. By using these codes the 5401 characters will be partitioned into 4096 groups, most containing 1 to 6 characters. Characters having similar peripheries are grouped together.

In the second stage we try to find a way to distinguish (or classify) similar characters in each partition. A conventional way is by pattern matching. However, it needs the normalization of character size, which is not a simple task. Another way is by heuristic method (or A.I. approach). Although this method is the most reliable one but it takes too much effort to write thousands of small programs to achieve the recognition goal. The other way is by 4D code [3] but it is noise sensitive. The final way we choose is by two dimensional Walsh transform [5]. Walsh transform involves only 1 or -1 operation and thus is a fast computation. It distinguishes from the conventional Fourier transform in that the former has "sequency" property and the latter has "periodicity" property, whereas the sequency property is more suitable for the analysis of the central portion ($\frac{1}{2} \times \frac{1}{2}$ of original size) of each Chinese character as shown in Figure 1. Please note that the central portion of a Chinese character contains the most valuable information.

In our experiment, we use three groups of complex Chinese characters, each containing 4 to 6 characters, and most characters in each group have the same corner and peripheral characteristics. Each character is imaged 8 times by changing its size, position and thresholding value. We find that most Walsh coefficients are stable under these changes. Thus we pick up 2 to 5 coefficients that have most seperability power, and we are able to use these coefficients to recognize each character in each group (or partition). This shows Walsh transform is a simple, fast and reliable method for seperating complex similar Chinese characters.

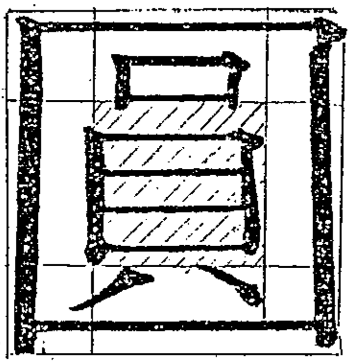


Fig. 1. Peripheral and central portions of a Chinese character

2. WALSH TRANSFORM OF CHINESE CHARACTERS

Walsh functions WAL(n,θ), $1/2 < \theta \le 1/2$, are defined recursively as follows:

$$\mathrm{WAL}(0,\theta)=\begin{cases} 1 \text{ if } -1/2 < \theta \le 1/2, \\ 0 \quad \text{otherwise.} \end{cases}$$

And

$$\mathrm{WAL}(2j+q,\theta)=(-1)^{[\frac{j}{2}]+q}\,[\mathrm{WAL}(j,2(\theta+1/4))$$

$$+(-1)^{j+q}\,\mathrm{WAL}(j,2(\theta-1/4))],\ q=0 \text{ or } 1,\ j=0,1,2\ldots,n.$$

And if θ is not in $(\frac{-1}{2}, \frac{1}{2})$, then WAL$(j,\theta)=0$.

The ordering of Walsh functions is sequency (Walsh) ordering:

WAL$(0,\theta)$, WAL$(1,\theta)$, WAL$(2,\theta)$, WAL$(3,\theta)$, ...

Let the input character image be MxM matrix $[y_{ij}]$, $i=1,\ldots M$; $j=1,\ldots M$, and let the origin of the coordinate system be translated to the center of the image. We extract the central portion of the input image, which contains M/2 x M/2 (truncated to integers) submatrix centered at the origin. The reason for extracting the central portion of the input image for use of Walsh transform is that the rest part, other than the central part, of the image has been used by 4P and 4C code in the first stage of partitioning. Let $N=M/2$ and when N is odd, say $N=2K+1$, we extract N equally spaced points from interval $[\frac{-1}{2}, \frac{1}{2}]$ and they are

$$\frac{-1}{2},\ -\frac{k-1}{2k},\ -\frac{k-2}{2k},\ \ldots,\ \frac{-1}{2k},\ 0,\ \frac{1}{2k},\ \ldots,\ \frac{k-1}{2k},\ \frac{1}{2}.$$

When N is even, say N=2k, then the N equally spaced points are

$$-\frac{2k-1}{4k}, -\frac{2k-3}{4k}, \ldots, \frac{-3}{4k}, \frac{-1}{4k}, \frac{1}{4k}, \frac{3}{4k}, \ldots, \frac{2k-1}{4k}.$$

Substituting these N points into θ of WAL(n, θ) given above, we get sequence of values WAL(n,i), i=0,1,2,...,N-1.

Now rename the extracted central portion as $[X_{ij}]$, i=0,...,N-1; j=0,...,N-1. Then the two dimensional Walsh transform of $[X_{ij}]$ is

$$C_{mn} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \ \mathrm{WAL}(m,i)\mathrm{WAL}(n,j),$$

where C_{mn} is called Walsh coefficient and m=0,...,N-1; n=0,...,N-1. The inverse transform clearly is

$$X_{ij} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} C_{mn} \ \mathrm{WAL}(m,i)\mathrm{WAL}(n,j).$$

Please note that since X_{ij} and Walsh functions have values 1 or -1 the calculation of C_{mn} can be implemented in TTL "and or" circuits so that the computational speed can be very fast. Since we limit our attention to complex (i.e. the number of strokes is high, say more than 10) Chinese characters, the coefficients corresponding to low sequencies will be discarded. Thus we do not compute C_{mn}, m=0,1 ,n=0,1. Also very high sequencies are not realistic in Chinese pattern, so we truncate those C_{mn}'s with $m \geq 7$ and $n \geq 7$. Hence we finally have 7x7-2x2=45 coefficients for consideration of feature selection.

Since we in general have 2 to 6 complex similar characters for classification (or recognition), among 45 coefficients we can select 2 to 5 coefficients that have most seperability power to classify the given characters. For practical usages, the size M of input character may change and in general M is between 30 to 120 depending on scanner resolution (say 400 dots per inch) and the character size. Too small character will not be considered unless the scanner can read it clearly. Theoretically C_{mn} should be invariant under the size change and small variation due to scanning noises and segmentation. Like Fourier transform, Walsh transform has been proved to be a powerful tool in image processing and pattern recognition (see[5] Chapter 8, also [6]). Thus for the classification of small group of Chinese characters, Walsh transform can provide features with high seperability power.

3. EXPERIMENTAL RESULTS

To demonstrate the power of Walsh transform in discriminating the complex similar Chinese characters, we choose three groups of complex Chinese characters derived from 4C and 4P codes. These three groups are

(1) 蘇、蔣、藏、蔽、藥，(2) 圍、圓、國、圈、圖、團，

(3). 朦、鵬、臘、噴。

To conduct the experiment and obtain meaningful results we must consider all possible happenings during the experimentation. From statistical point of view the sample data obtained from the experiment must be as complete as possible. Thus the variations due to character size change, segmentation (poistion change) and binary thresholding are simulated in the experiment. Each character is read eight times with the simulated variations. The mean, standard deviation, maximum and minimum of 16x16 Walsh coefficients for each character are calculated. Figure 2 shows the result of these calculations for character 藏 .

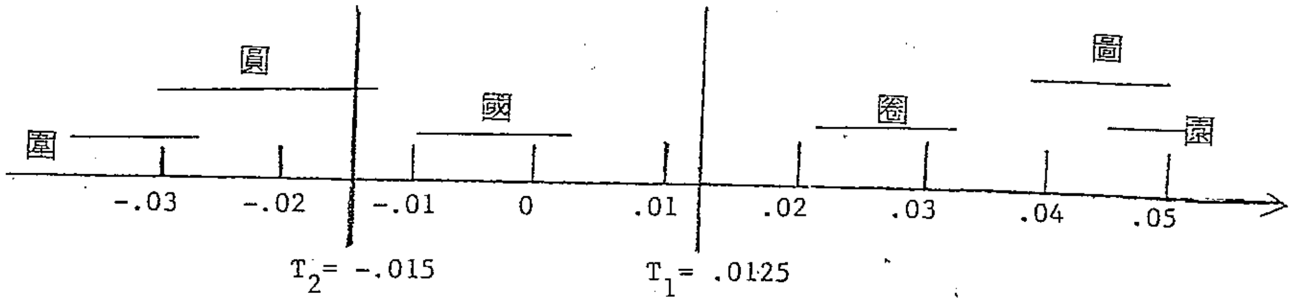
In each group, we preselect two characters and each one has a statistical table of Walsh coefficients we then compare these two tables entry by entry but only on 45 coefficients mentioned in section 2. From these pairwise comparisons we choose one coefficient (or feature) that has most seperability power which depends on the maximum, minimum and standard deviation of each character. From this selected coefficient we choose proper threshold values and partition the group into two or three subgroups; each subgroup may or may not overlap the other. Continuing in this way we are able to use only two to five coefficients to seperate

```
Cmn COEFFICIENTS FOR  CHARACTER 藏      3

THE # OF LOOP(N/2) OF THIS CHAR. IN DIFFERENT PICTURE
      FIG 1= 91  FIG 2= 93  FIG 3= 79  FIG 4= 96  FIG 5= 97  FIG 6=101  FIG 7= 61  FIG 8= 83

                 1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16
              ****** ****** ****** ****** ****** ****** ****** ****** ****** ****** ****** ****** ****** ****** ****** ******

 1) MEAN      0.7405 -.0189 0.0298 -.0053 0.0542 -.0167 -.0771 0.0178 0.0262 -.0106 0.0024 0.0028 -.0030 0.0185 0.0213 -.0482
    STAND.DEV 0.0300 0.0040 0.0141 0.0093 0.0191 0.0056 0.0079 0.0174 0.0282 0.0200 0.0084 0.0156 0.0181 0.0133 0.0077 0.0119
    MAXIMUM   0.7895 -.0133 0.0457 0.0051 0.0597 -.0091 -.0377 0.0412 0.0614 0.0335 0.0134 0.0173 0.0121 0.0312 0.0317 -.0329
    MINIMUM   0.7062 -.0243 -.0016 -.0187 0.0412 -.0245 -.0060 -.0161 -.0242 -.0281 -.0144 -.0245 -.0438 -.0086 0.0071 -.0634

 2) MEAN      0.0322 -.0351 -.0237 0.0039 0.0154 0.0525 0.0253 -.0557 -.0759 -.0385 0.0042 0.0035 0.0380 0.0206 -.0285 0.0227
    STAND.DEV 0.0069 0.0062 0.0109 0.0080 0.0061 0.0067 0.0116 0.0148 0.0167 0.0061 0.0101 0.0048 0.0056 0.0051 0.0153 0.0064
    MAXIMUM   0.0411 -.0274 -.0011 0.0209 0.0233 0.0523 0.0400 -.0317 -.0419 -.0317 0.0157 0.0117 0.0441 0.0272 -.0011 0.0326
    MINIMUM   0.0163 -.0449 -.0365 -.0038 0.0064 0.0328 0.0036 -.0811 -.0997 -.0501 -.0161 -.0029 0.0150 0.0152 -.0429 0.0133

 3) MEAN      -.0798 0.0252 -.0053 0.0079 0.0048 0.0330 -.0355 -.0505 0.0141 0.0113 0.0065 -.0177 -.0051 -.0014 0.0070 -.0162
    STAND.DEV 0.0094 0.0057 0.0030 0.0036 0.0071 0.0054 0.0063 0.0077 0.0047 0.0121 0.0072 0.0057 0.0056 0.0048 0.0023 0.0067
    MAXIMUM   -.0633 0.0345 0.0000 0.0116 0.0136 0.0427 -.0425 -.0270 0.0298 0.0354 0.0162 -.0095 0.0007 0.0100 0.0130 -.0043
    MINIMUM   -.0930 0.0165 -.0086 0.0029 -.0039 0.0215 -.0627 -.0575 0.0070 -.0004 -.0099 -.0280 -.0150 -.0254 0.0060 -.0233

 4) MEAN      -.0536 0.0020 0.0070 -.0042 -.0300 -.0407 0.0151 0.0035 -.0077 0.0082 0.0123 0.0254 -.0091 0.0073 -.0197 -.0117
    STAND.DEV 0.0170 0.0065 0.0021 0.0072 0.0065 0.0103 0.0042 0.0026 0.0084 0.0071 0.0088 0.0055 0.0082 0.0076 0.0054 0.0033
    MAXIMUM   -.0247 0.0115 0.0098 0.0019 -.0172 -.0198 0.0234 0.0063 0.0107 0.0188 0.0318 0.0319 -.0018 0.0174 -.0080 -.0065
    MINIMUM   -.0698 -.0102 0.0032 -.0197 -.0362 -.0504 0.0100 -.0016 -.0149 -.0006 0.0051 0.0185 -.0254 -.0045 -.0242 -.0173

 5) MEAN      0.0341 0.0234 -.0005 0.0137 -.0106 -.0178 -.0252 0.0055 -.0112 -.0218 0.0158 0.0101 -.0153 -.0273 0.0101 0.0035
    STAND.DEV 0.0139 0.0050 0.0046 0.0063 0.0043 0.0037 0.0076 0.0029 0.0035 0.0076 0.0045 0.0033 0.0062 0.0081 0.0078 0.0083
    MAXIMUM   0.0556 0.0317 0.0060 0.0222 -.0043 -.0141 -.0147 0.0108 -.0047 -.0096 0.0174 0.0149 -.0016 -.0086 0.0170 0.0170
    MINIMUM   0.0136 0.0172 -.0097 0.0036 -.0178 -.0236 -.0366 0.0033 -.0145 -.0328 0.0060 0.0060 -.0206 -.0339 -.0027 -.0064

 6) MEAN      0.0117 -.0062 -.0043 -.0225 -.0047 -.0250 -.0023 -.0056 0.0271 0.0396 -.0073 -.0024 -.0121 0.0030 -.0091 -.0045
    STAND.DEV 0.0070 0.0023 0.0071 0.0058 0.0040 0.0064 0.0085 0.0071 0.0075 0.0095 0.0049 0.0049 0.0079 0.0044 0.0036 0.0042
    MAXIMUM   0.0236 -.0041 0.0097 -.0134 -.0027 -.0113 0.0125 0.0058 0.0354 0.0456 -.0011 0.0039 0.0045 0.0074 -.0043 0.0021
    MINIMUM   0.0024 -.0113 -.0126 -.0324 -.0133 -.0310 -.0132 -.0151 0.0129 0.0188 -.0147 -.0103 -.0213 -.0040 -.0127 -.0123

 7) MEAN      0.0307 -.0121 0.0227 0.0053 0.0194 0.0285 -.0222 -.0271 0.0021 -.0088 -.0130 -.0117 0.0138 0.0214 0.0091 -.0174
    STAND.DEV 0.0070 0.0025 0.0049 0.0021 0.0025 0.0051 0.0039 0.0043 0.0047 0.0062 0.0044 0.0042 0.0067 0.0057 0.0030 0.0029
    MAXIMUM   0.0395 -.0090 0.0308 0.0083 0.0227 0.0354 -.0177 -.0206 0.0109 0.0048 -.0038 -.0043 0.0217 0.0280 0.0057 -.0112
    MINIMUM   0.0163 -.0151 0.0188 0.0016 0.0156 0.0184 -.0260 -.0327 -.0051 -.0149 -.0197 -.0173 0.0029 0.0118 -.0045 -.0202

 8) MEAN      -.0379 -.0075 -.0342 0.0016 -.0029 -.0117 -.0174 -.0049 -.0065 0.0008 0.0077 0.0122 -.0027 -.0025 -.0052 0.0078
    STAND.DEV 0.0121 0.0053 0.0058 0.0027 0.0025 0.0058 0.0041 0.0030 0.0046 0.0030 0.0026 0.0046 0.0032 0.0035 0.0024 0.0047
    MAXIMUM   -.0187 0.0005 -.0161 0.0057 0.0010 -.0027 -.0077 -.0004 0.0011 0.0042 0.0123 0.0135 0.0009 0.0048 -.0018 0.0150
    MINIMUM   -.0565 -.0134 -.0458 -.0018 -.0072 -.0198 -.0235 -.0089 -.0111 -.0036 0.0043 0.0048 -.0077 -.0067 -.0086 0.0010

 9) MEAN      -.0522 0.0296 -.0084 0.0020 -.0006 -.0223 -.0130 0.0109 0.0013 0.0012 -.0001 0.0048 -.0060 -.0103 -.0036 0.0062
    STAND.DEV 0.0346 0.0060 0.0141 0.0033 0.0023 0.0052 0.0037 0.0022 0.0047 0.0033 0.0054 0.0036 0.0013 0.0043 0.0080 0.0057
    MAXIMUM   0.0027 0.0377 0.0095 0.0072 0.0037 -.0142 -.0057 0.0132 0.0075 0.0053 0.0124 0.0119 -.0037 -.0032 0.0040 0.0134
    MINIMUM   -.1057 0.0193 -.0312 -.0027 -.0046 -.0296 -.0170 0.0066 -.0070 -.0033 -.0039 0.0008 -.0078 -.0165 -.0210 -.0008

10) MEAN      -.0345 0.0095 0.0013 -.0067 -.0119 -.0198 -.0053 0.0106 0.0108 0.0047 0.0017 0.0044 -.0073 -.0059 0.0048 0.0031
    STAND.DEV 0.0165 0.0113 0.0124 0.0037 0.0019 0.0093 0.0064 0.0055 0.0044 0.0061 0.0031 0.0023 0.0040 0.0034 0.0028 0.0061
    MAXIMUM   -.0193 0.0237 0.0270 -.0005 -.0092 -.0039 0.0064 0.0199 0.0175 0.0077 0.0054 0.0053 -.0021 -.0016 0.0107 0.0090
    MINIMUM   -.0832 -.0113 -.0093 -.0126 -.0151 -.0317 -.0140 0.0018 0.0049 -.0091 -.0045 0.0010 -.0136 -.0112 -.0001 -.0113

11) MEAN      -.0505 -.0022 -.0067 -.0087 0.0100 -.0015 0.0053 0.0041 0.0030 0.0036 -.0060 0.0030 0.0025 0.0052 -.0071 0.0001
    STAND.DEV 0.0110 0.0043 0.0044 0.0022 0.0030 0.0041 0.0038 0.0036 0.0049 0.0055 0.0037 0.0033 0.0027 0.0025 0.0070 0.0018
    MAXIMUM   -.0435 0.0037 -.0013 -.0055 0.0141 0.0002 0.0137 0.0194 0.0077 0.0125 0.0009 0.0067 0.0052 0.0107 0.0094 0.0021
    MINIMUM   -.0714 -.0097 -.0159 -.0118 0.0038 -.0123 0.0039 -.0011 -.0059 -.0018 -.0117 -.0028 -.0016 0.0028 -.0173 -.0039

12) MEAN      -.0593 0.0109 -.0234 0.0041 0.0000 0.0033 -.0296 -.0005 0.0105 -.0011 -.0070 -.0074 -.0023 -.0060 0.0110 0.0066
    STAND.DEV 0.0244 0.0080 0.0074 0.0029 0.0049 0.0047 0.0035 0.0032 0.0060 0.0033 0.0036 0.0039 0.0035 0.0031 0.0040 0.0047
    MAXIMUM   -.0044 0.0187 -.0075 0.0062 0.0077 0.0134 -.0150 0.0037 0.0294 0.0042 -.0047 -.0010 0.0032 -.0008 0.0178 0.0115
    MINIMUM   -.0858 -.0070 -.0313 -.0025 -.0071 -.0013 -.0272 -.0053 0.0022 -.0056 -.0144 -.0116 -.0077 -.0100 0.0051 -.0038
```

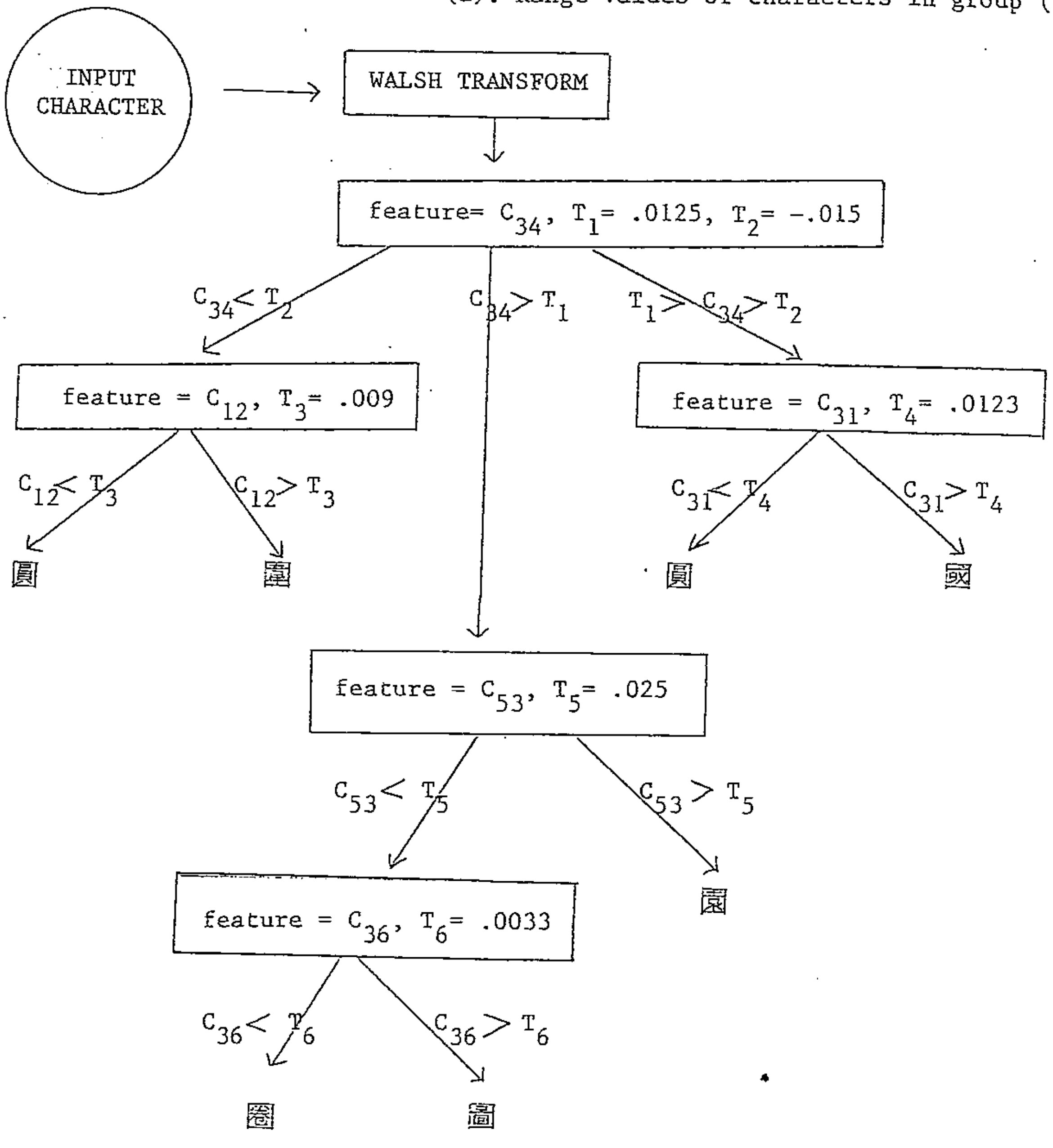
Fig. 2. statistical values of Walsh coefficients for character 藏.

all characters in a given group. A typical example for group (2) is given in Figure 3, where the horizontal bar of a character denotes its statistical range of a coefficient value C_{34}, and T_1, T_2 denote the threshold values, calculated from ranges and standard deviations. The recognition procedure in this example is described as follows. Given an input of unknown character from group (2), calculate its Walsh coefficient C_{34}. If $C_{34} < T_2 = -0.015$ then calculate C_{12} and check if $C_{12} > T_3 = 0.009$ ('yes' it is 圍 and 'no' it is 圓). If $T_1 = 0.0125 > C_{34} > T_2$ then calculate C_{31} and check if $C_{31} > T_4 = 0.0123$ ('yes', it is 國 and 'no' it is 圓). If $C_{34} > T_1$ then calculate C_{53} and check if $C_{53} > T_5 = 0.025$ ['yes' it is 團 ; 'no' then calculate C_{36} and check if $C_{36} > T_6 = 0.0033$ ('yes' it is 圖 ; 'no' it is 圈)].

Having set up the recognition system, we then try to run the system hundred times. During the testing we find that some threshold values should be readjusted and some characters may jump into other classes that are supposed to exclude them. By tuning the system five times we are able to get a recognition rate 99.5% for a total of two hundred runs (i.e. only one failure).



(a). Range values of characters in group (2).



(b). The classification tree.

Fig. 3. (a)Range values of characters in group (2).
(b)The classification tree.

4. CONCLUSION

We have used the Walsh transform for complex similar Chinese character image recognition and found that this method is reliable and simple. This method is more reliable than 4D code since 4D code is noise sensitive whereas Walsh transform is not. This method is also better than other methods in that it is simple, fast and reliable.

REFERENCES

1. Kenichi Mori and Isao Masuda, Advances in recognition of Chinese characters, Proceedings of International Conference on Pattern Recognition, 1980, 692-702.
2. Jun Tsukumo and Ko Asai, Machine printed Chinese character recognition by improved loci features, IECE Trans. on Pattern Recognition and Learning, PRL 85-10, 9-16, 1985 (In Japanese).
3. Shin-ichi Meguro and Michio Umeda, A multi-font printed Chinese characters reader, The Transactions of the Institute of Electronics and Communication Engineers of Japan (Trans. IECE) Vol. J6T-D, 908-915, 1984.
4. H. S. Hou, Digital Document Processing, John Wiley & Sons Inc., New York, 1983.
5. K. G. Beauchamp, Walsh Functions and Their Applications, Academic Press, New York, 1975.
6. A. Rosenfeld and A. C. Kak, Digital Picture Processing, Vol. 1, 2, Academic Press, New York, 1982.