'91
18

Design of A Multi-font Chinese Character Generator for Personal Computers

C.C. Hsieh — Research Fellow, Inst. of Information Science, Academia Sinica, Taipei
C.T. Chang — Professor Dept.EET., National Taiwan Institute of Technology
P.H. Chen — Research Associate, Chinese Character Analysis Group, NTIT

ABSTRACT

In this paper, a new high quality, high performance Chinese character pattern generator is presented. The major improvements over the passed ones includes the eliminition of unnecessary computations, systemize the structure and modularize the structure to minimize the modifications reguired for a font change .

An experimental system has been implemented on NEC 9801 personal computer. The statistics show that a basic system of ten thousand character patterns reguires 232k bytes memory space. For change or adding a new font to the system, only 34 subroutines need to be redefined and programmed. This is less than 24k bytes in BASIC in our experimental system. For farther add-on characters, an average of 16 bytes/character is needed and the rest of the system may remain unchanged.

This paper also indicate that this system is very suitable for persuading VLSI implementation.

Acknowledgement

This research is a part of the work done by the Chinese Character Analysis project supported by the Council of Culture Planning and Development.

1.Introduction

Some pioneer studies of Chinese character pattern generator (cpg) ultilizing a composition mechanism over a set of component patterns can be found in the early 70's [1,2,3,4,5,6,7,8]. Since then, many proposals have been made by varing the composition mechanism and the definition of the component set [9,10,11,12,13]. All these early systems were generating low density, line-stroke patterns, and aimming at saving memory space required for pattern storage. As the micro-electronic LSI technology keeps on progress, the design goods of cpg changed somewhat. Many systems emphisized on the system throughput [14,15] in order to match the requirement of a high resolution and high speed printer, ie 200 to 300 dots per inch and 1 to 2 pages per second. Some others emphisize on the quality of the pattern image [16,17,18,19]. The development of high quality cpg's is very interesting, because they start to generate bold-face, nice-looking characters which makes the system acceptable for office applications with quality.

In the paper, a new high quality, high speed cpg is proposed. The major improvements over the passed studies include the eliminating of unnecessary complex computations, systemize the structure of cpg and modulize the structure which minimize the modifications required for font changes. An experimental system with 800 character patterns has been implemented and the results are quite satisfactory.

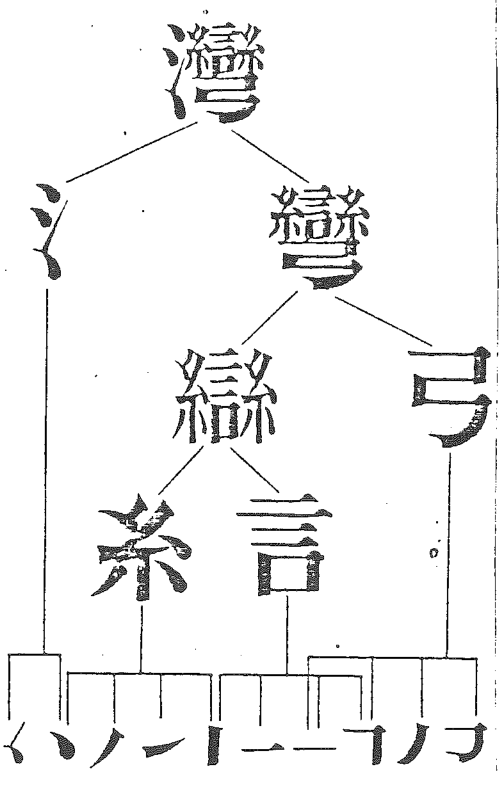


Figure 1 An illustration of character pattern represention.

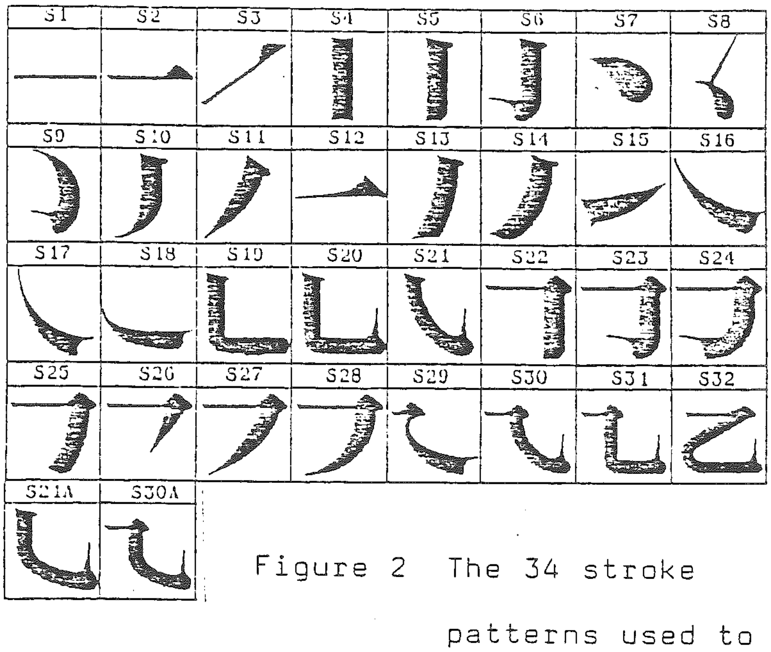


Figure 2 The 34 stroke patterns used to generate Soon-Font

2. Representation of Character Patterns

Let p be a character pattern and P be the set of character patterns to be genersted. Let r be a component pattern and R be the set of component patterns (includes an empty component) that can generate every p in P. Then, each p can be expressed by the components which p is composed of and the location l and size z of each components, respectively. This is shown in Eq-1.

$$p = r_1(l_1,z_1),\ r_2(l_2,z_2)\ldots r_n(l_n,z_n) \quad - - - - - \quad (1)$$

In equation 1, n is a positive integer that indicates the number of components in a pattern p. An illustration of Equation is show in Fig-1.

Let t be the pattern generated by a stroke and T be the set of all stroke patterns that can generate all the r's in R. Then for every r in R, r can be expressed as a series of stroke patterns associated with their locations in r and their arguments that characterize the shapes of the strokes. It is shown in Eq-2. An example of a set T for generating Soon-Font character patterns is shown in Fig-2. In Fig-2, there are 34 strokes. The arguments of strokes vaary from 2 to 4, including a center point which is used for painting or filling the inside of the stroke envelop. This is a special arraugement for programming the character pattern in BASIC language with graphic instruction "PAINT" . For other implementation, the "paint" can be considered as a special function required for generation the patterns.

$$r = t_1(l_1,a_1),\ t_2(l_2,a_2)\ldots t_m(l_m,\ a_m) \quad - - - - - \quad (2)$$

In Eq-2, m is the number of stroke patterns of which r is composed. For every stroke pattern, the envelop is expressed as a series of line-segments and arcs instead of the cubic-splines [16,17,18,19,20]. This simply the computation complexity proposed by previous works. The reason of doing this is to meet the fact that the character patterns are designed. For example, by examine the drawing envelop of basic stroke patterns of the Soon-Font, only Line-segments and ares of a circle are found [20]. An example of the character "永" is shown in Figure-3, and a structure of the basic stroke pattern "丶" is shown in Fig-4.

What described above is proposing a general structure model for character pattern generation. The system is flexible for different implementations as well as for tailormade every stroke or component for a specific character by properly select those parameters shown in Eq-1 and Eq-2. For generating different fonts of characters, only T needs to be changed. Other part of the system may remain unchanged. Some examples of Soon-Font and Bold-Font patterns by different T sets are shown in Fig-5(a) and Fig-5(b), respectively.

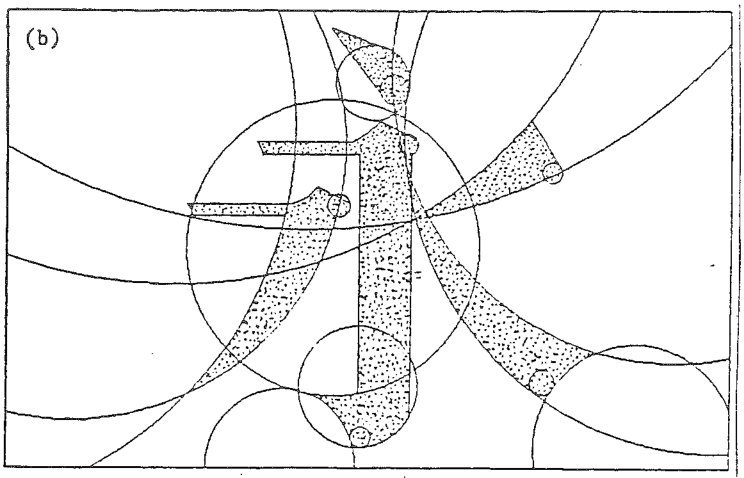
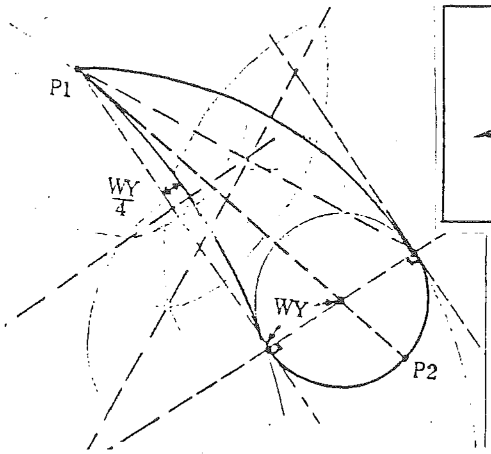


Figure 3 An illustration of stroke envelops consisting of line-segments and arcs of circle only.



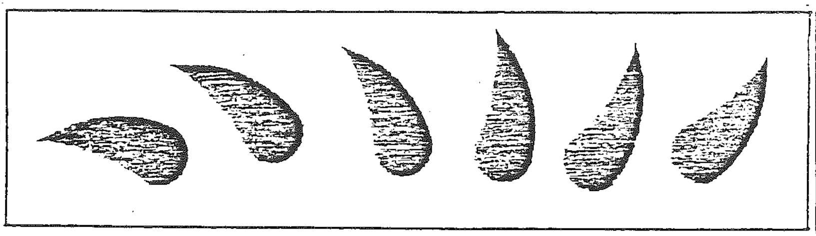
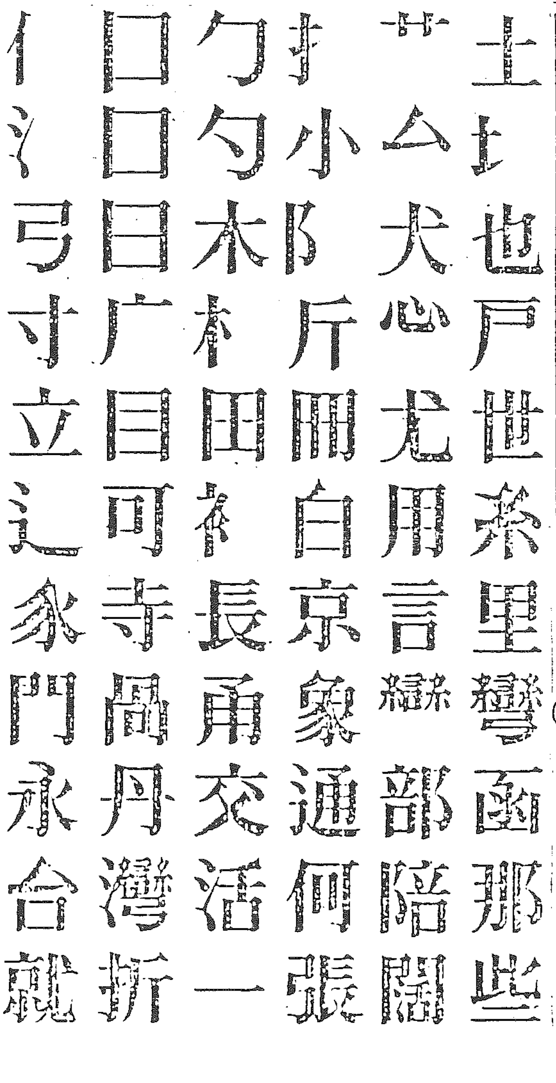
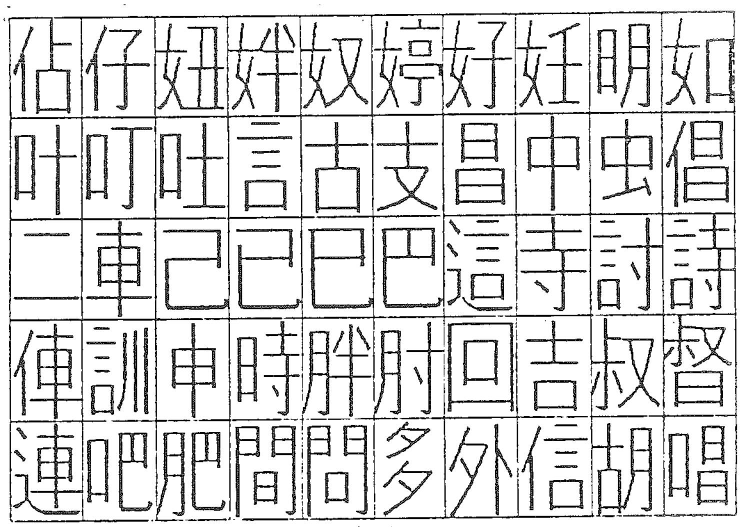


Figure 4 The definition of stroke pattern of "丶" and same patterns samples.



(a)



(b)

Figure 5 (a) character samples of Soon-Font. (b) character samples of Bold-Font by substituite 34 subroutines of stroke definitions in the same generator used in (a).

3. Implementation

An experiment of implementing a cheracter pattern generator following the model described is done on an NEC-9801 personal computer. All programs are done in BASIC.

The coordinate and frame of a character pattern is selected to be 256X256 dot matrix with the left-top cornor as origin. And so are the components and basic strokes. All parameters in Equations (1) and (2) are relative with respect to the frame they belong to.

A collection of Equation 1 for the set P is called the pattern description file (pdf). The pdf ultilizes binary-tree structure in our implementation. Each p is decomposed onto two components. If neither one of those two components is in P or in R, a new non-terminating sub-pattern q has been found. Let the collection of q's be Q. The representation and processing of q's will be exactly the same as those of the p's in P. According to previous studies [5,12,22], for 10 thousands characters, the number of q is around 3 thousand. And the ratio of the number of q's over the number of p's decreases as the number of p's increased. This is because q may be shared by some characters.

The binary-tree structure of pdf provides a uniform address space for locationing the pattern discription for each characters. Therefore, its address can be computed if the coding for characters is a continuous space in the internal representation of a computer system.

Let C be the union of the sets P,Q and R. Define a partial ordering relation $\geq$ as folloows. For elements u and v that belong to C, $u \geq v$ iff v is a sub-pattern of u. By this definition, the relation is reflective, transitive and antisymmetric. Therefore, it is a partial ordering relation. Besides, in set C, there is no two elements with identical pattern, therefore, $\geq$ may be replaced by >. The relation > defines a non-disjoint partition on C. And, for each domain of the partition, a binary tree can be constructed by the relation >.

By the above discussion, the cpg is a forest (a collection of trees) of binary trees. And each p in P can be selected as the entry point to the cpg. A formal presentation of the cpg is formulated as follows.

$$\left.\begin{array}{ll} \text{(Entry point)} \rightarrow p & \text{where } p \in P \\ p \rightarrow b,\ b & \text{where } p > b \\ b \rightarrow p \mid q \mid R & \text{where } q \in Q \\ q \rightarrow b,\ b & \text{where } q > b \\ R \rightarrow [\text{ the set of components }] & \end{array}\right\} \text{------ (3)}$$

In the above equations, p,b, and q are variables, and the only terminator set is R. Therefore, after some iterations under the constrain the p >b and q >b, a character pattern expressed in Eq-1 can be found with the relative location and size associated with each b in Eq-3.

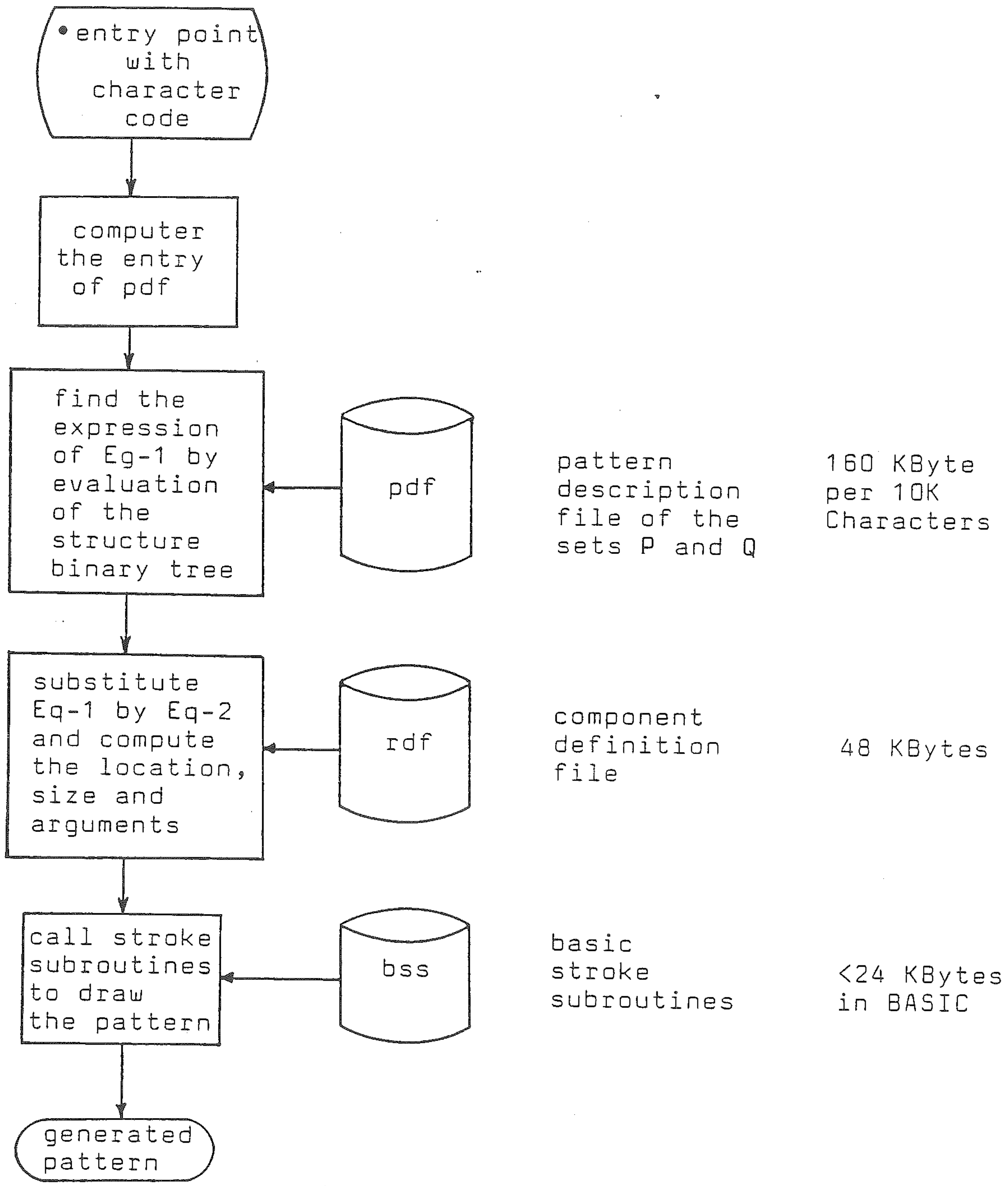


Figure 6 A Block Diagram of the System Structure and Flow.

When Eq-1 has been found for a character pattern, the next step is to substitute each r by its expression in Eq-2. This will give an expression of character pattern in terms of strokes. The effective location and size of each stroke with respect to the frame of a character pattern can be easily computed by a transformation of the coordinate. And so are the arguments for drawing the stroke patterns. Those transformations are tedious, they are omitted here. The final step is to call stroke subroutines to draw the pattern for the character. A system flow digram of the described steps is shown in Figure 6.

4.Discussions

The component set we choised is the one developed at chiao-tung University [7,8,10] with an extension made. This is due to the fact that a component may have slightly difference appearance when used at different positions in characters. For such variations, they are considered as different components in our system in order to generate more elegant patterns. The total number of components is thus approaching 1000, and the total memory occupied is slightly less than 48K Bytes.

For pdf, the average space required is slightly less than 16 Byte/character. This is averaged over 800 characters implemented in our experimental system. This number is very much consistant with the numbers of our early studies [10,11,12]. Therefore it is reasonable to estimate that for each 10K characters, 160K Bytes are required.

As for the programs, since all programs are writen in BASIC, from the readings on the disk file, the total is 24K Bytes. This includes 34 basic stroke drawing subroutines.

From the statistics shown above, the total space required for 10 thousands character patterns are 48+160+24=232KBytes. The size is perfact for 16-bit PC's.

For each font change, at most 24KByte changer are needed for those 34 basic stroke drawing subroutines. And for further add-on new characters, only 16 byte/character is needed.

Some more examples of the print-out are shown in Figure 7, 8, 9 and 10. Please be noticed that the change in size, shape and orientation can be easily achieved. This is another unique fecture of this character generator.

Because the system is processed by a BASIC interpreter with floppy disc drives, the speed of producing the character images is not fast. The NEC 9801 we used is not equiped with 8087 arithmetic processor. Under this circumstances, for 32X32 matices, the speed is about 5 characters/second. The speed depends on the number of strokes per character and the size of the character. But the former one is of major concern.

The size and complexity of the system is very suitable for persuading VLSI implementation. For VLSI implementation, parallelism can be arranged at component or stroke level and the speed improvement to reach 100 to 1000 character patterns per second is possible [14,15].

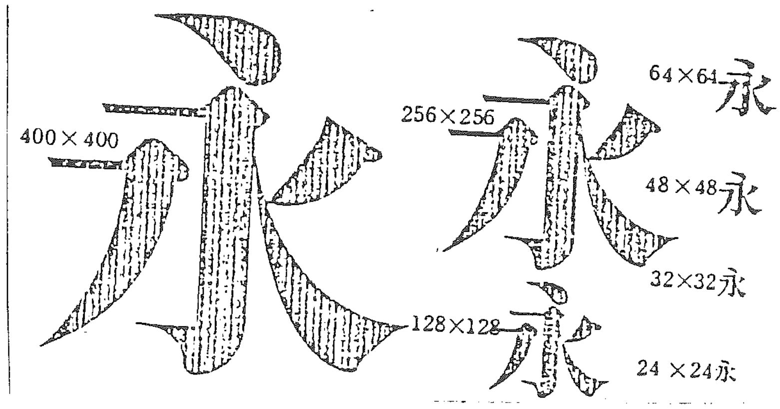


Figure 7 Examples of varistions of the character "永".

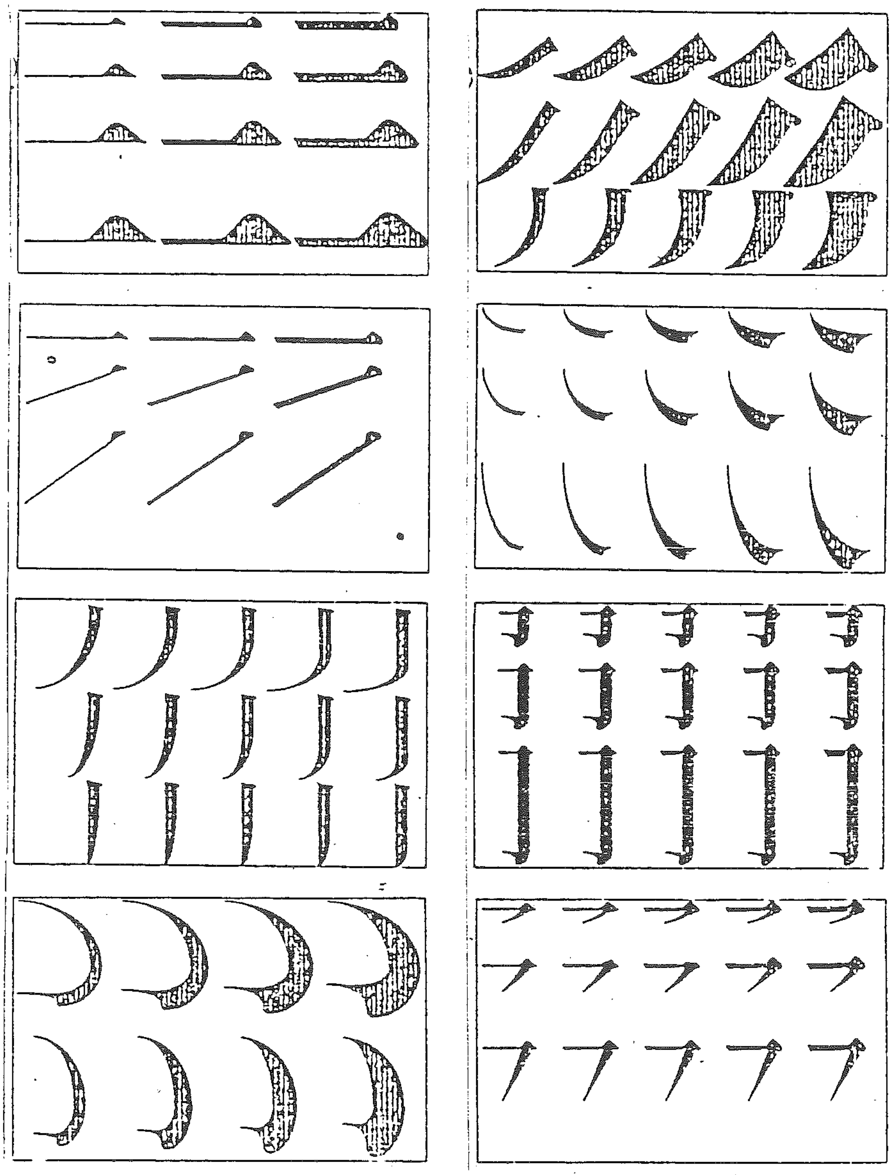


Figure 8 Examples of the variations of stroke potterns by varing their parameters.

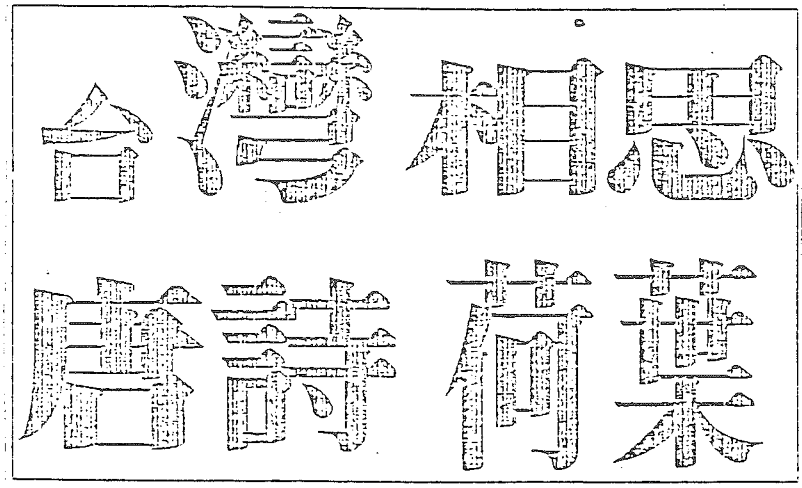


Figure 9
Printing with different sizes.

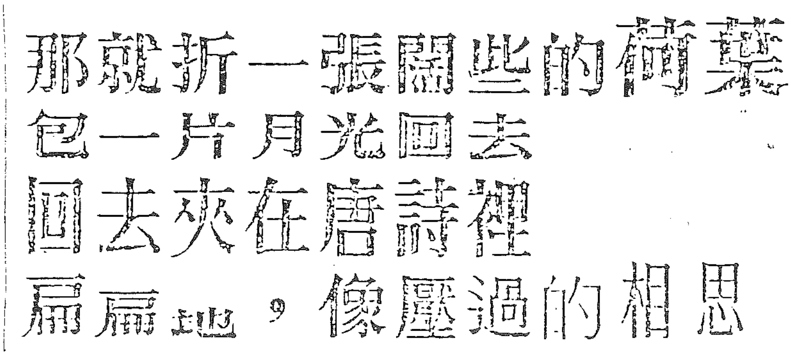


Figure 10
Printing with different sizes and thickness.

References

1. B.K. RANKIN, and L.L.TAN, "Component Combination and Frame-Embedding in Chinese Character Grammar", National Bureau of standard, Technical Note 492, U.S.A. Feb. 1970.
2. T.SAKAI, M. NAGAO, and H. TERAI, "A Description of Chinese Characters Using Sub-patterns", Information Processing in Japan, vol.10, 1970.
3. K.Ni, "A Linquistic Structure Model for Chinese Characters", M.S. Thesis, National Chiao-Tung University, Hsinchu, Taiwan, June 1972.
4. C.C. HSIEH, Y.H. Hwang, and S.LIN, "Analysis of Chinese Radicals", Science Bulletin, National Chiao-Tung University, vol. 6, no. 1, Hsinchu, Taiwan, Feb. 1973.
5. C.C. HSIEH, M.W.DU, and S.H. CHI, "The storage of Radicals and composition of Characters", Science Bulletin, National Chiao-Tung University, vol.6, no. 1, Hsinchu, Taiwan, Feb 1973.
6. S.K. Chang. "An Interactive System for Chinese Character Generation and Retrieval", IEEE Trans on System, Man, and Cybernetics, vol. SMC-3. no.3, May, 1973.

7. C.C. HSIEH, Y.M. HWANG, S. LIN and K.M. HSU, "The Chiao-Tung Radical system part I: The Analysis and Design of the Chiao-Tung Radical System", Proceedings of the First International Symposian on Computers and Chinese Input/Output Systems, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, August 1973.
8. M.W. DU, C.C. HSIEH, S.H. CHI and S.C. CHU, " The Chiao-Tung Radical System Part II: Character Composition and Methods to Represent Radicals", Proceedings of the First International Symposian on Computers and Chinese Input/Output Systems, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, August 1973.
9. E.F.Yhap, "Keyboard Method for Composing Chinese Characters", IBM Journal of Research and Development vol.19, no. 1, Jan. 1975.
10. M.W. Du, J.S. Tsay and C.C.Hsieh, "The Design of CHIPS- a Chinese Information Processing System", Proceedings of International Computer Symposium, vol.1, National Chiao-Tung University, Hsinchu, Taiwan, Aug. 1975.
11. Y.M.Hwang, "A New Component System with Geometrical Grammar and Improved Character Appearance", Proceedings of International Compater Symposium, vol.1, National Chiao-Tung University, Hsinchu, Taiwan, Aug. 1975.
12. T.Y.Kiang, "A Concise Chinese Language Input/ Output System Using Bi-Portion Composition Techuique (BP CLIOS)", Proceedings of the 1979 international Conference of the Chinese language Computer Society, vol.1, San Francisco, California, Feb. 1979.
13. W.T.Huang and W.C. Cheng, "A Pattern Drawing System for Chinese Character", Proceedings of the 1979 International Conference of the Chinese language Computer Society, vol.1, San Francisco, California, Feb. 1979.
14. L.S.Shiaw, "The design of High-performance Chinese Character Generating System", M.S. Thesis, National Chiao - Tung University, Hsinchu, Taiwan, June, 1979.
15. S.H. Lam, "Effect Encoding of High-Density Chinese Character Fonts by stroke Composition", M.S. Thesis M.I.T., Cambridge Mass. June 1980.
16. Y.M. Dong, "An Experimental System for Generating High Quality Chinese Characters", Chinese Journal of Computers, vol.3, no.1, 1980.
17. K.J. Chen, K.Y. Cheng and S.M. Chion," A Syntatic Approach in Dot Matrices", Proceedings of the International Computer Symposium '82, Taipei, Taiwan, 1982.
18. K.J. Chen and K.Y. Cheng, " A Structured Design Methoclology for Chinese Character Fonts", Proceedings of the International Comference on Text Processing with very large character set, Tokyo, Japan, Oct. 1983.
19. J.Hobby and G.GU "A Chinese Metafont", Proceedings of the International Comference on Text Processing with very large character set, Tokyo, Japan, Oct. 1983.
21. D.E. Knuth, " Mathematical Typography", Bulletin of the American Mathematical Society, vol.1, no.2, March 1979.
22. C.C.HSIEH, C.T. CHANG, P.H. CHEN, "A Character generator for personal computers", Technique Report, Chinese character Analysis Group, Taipei, Taiwan, July 1984.