E. F. Yhap

Keyboard Method for Composing Chinese Characters

Abstract: A keyboard method is described that allows a user to form Chinese characters. The user's keystrokes activate internal logic, which performs the necessary scaling and positioning of the character components. An initial design using a 40-key keyboard is described, together with approximations of the character shapes produced by given keying sequences. Rough estimates of speed range from eight to 33 characters per minute.

Introduction

This paper describes an initial design for a method of keying Chinese characters [1]. The keyboard is expected to have about 40 to 50 keys. In contrast, today's commercial Chinese typewriters have up to four plates with 2,000 character slugs on each plate. The work was motivated by the desire to provide a keyboard approach to forming Chinese characters, which could be used without extensive training by anyone who can read and write Chinese characters.

Our method can be characterized as a composition approach, in the sense that we compose a character from component parts. Moreover, we term our approach *generic* composition, as opposed to other composition methods, which we term *nongeneric* [2, 3]. The difference is that we store only one set of normalized representations of Chinese radicals (78 in our initial design). (Radicals are elementary characters used in many Chinese dictionaries as indices to more complex characters that contain one or more of the radicals.) Any scaling or positioning is then done by logic circuits. We can then construct any character from the keyboard. We provide a one-character display of the (partially) constructed character as a visual aid for training and correction purposes. Despite this feedback to the user, the complexity of some Chinese characters raises a natural concern about the difficulty of their construction. Actually in modern Chinese complex characters that occur frequently either have been or probably will be simplified. Thus infrequent occasions are expected when additional time is required to form correctly those complex characters that have not been simplified.

Two other general approaches to forming Chinese characters now exist, and these do not resort to composition:

1. The large keyboard, which may contain 2,000 to 8,000 keys or slugs.
2. Keyboard entry of an index code, which causes access to a stored dictionary of characters [4-7]. In case the code does not select a unique character, then a set of possible characters is displayed for the typist to choose from.

As opposed to storing a large dictionary, our generic composition approach should reduce the amount of storage required by two or three orders of magnitude. (We require storage for less than 100 characters, compared to dictionary approaches that may require storage for 10,000 to 50,000 characters.)

In this paper we first describe how the user applies his knowledge of Chinese to strike keys on the keyboard corresponding to the natural strokes and pattern formations he uses when manually writing Chinese characters. We then consider the internal logic, which translates the user's keystrokes into logical sequences to construct the desired character. Keying sequences for some examples are then given, together with the approximate character shapes they produce. Although the set of symbols chosen for the keys is not optimal, the set does produce reasonable shapes as shown by the examples. Moreover, they are capable of producing the 214 radicals listed in Appendix A from which the Chinese characters are constructed. The keying sequences for these radicals are included. Also given, in Appendix B, are corresponding keying sequences and shapes for some frequently occurring characters. Appendix C gives some gross estimates of speed.

User interface

The keyboard shown in Fig. 1 represents a possible distribution of symbol keys on the keyboard; the proportioning and relative positions of the symbols are represented as accurately as possible. Table 1 lists these assignments in alphanumeric order.

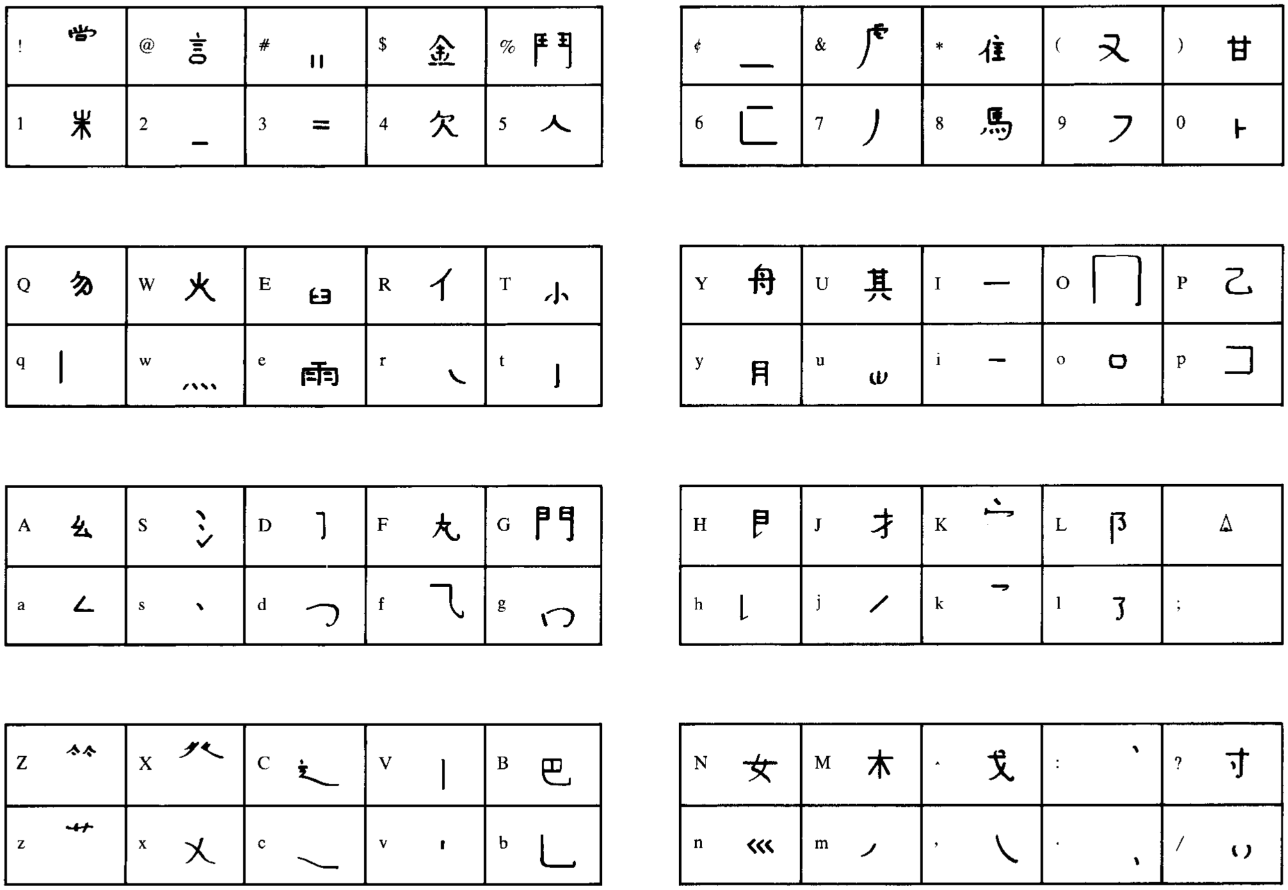


Figure 1 Chinese symbol keys.

In addition to the Chinese symbol keys, scaling and positioning keys are included; Fig. 2 shows their position with respect to the Chinese symbol keys.

To elucidate the use of the symbol and scaling-positioning keys in composing a character, we give two sets of examples. Because the second set involves the use of subblocking, we defer its presentation until we explain the subblocking procedure.

Examples of the first set are shown in Table 2. The keying sequence is given by the English letter, number, or symbol (usually) found in English typewriters in the key position to which the Chinese symbol is assigned. The keystroke ∆ indicates end of character. Depression of the U or R bar shown in Fig. 2 indicates that a symbol is to be moved about 25% upwards or to the right, respectively. In operation, therefore, either the U or the R bar may be depressed by the thumb, while another finger simultaneously strikes a symbol key. If neither the U nor the R bar is depressed, then by default the normal positioning of the symbol is used. In the examples, depression of the U or R bar is indicated by a subscript U or R (e.g., g_U in Example 1). The following trace shows in detail the composition sequence for the first example:

Keystroke	*Partial character*	*Comments*
g_U		The position is raised.
i		
v		
2_U		The position is raised.
∆		The character is fully composed.

Similar traces can be worked out by the reader for the other three examples.

The four examples in the first set involve composing symbols in only one block, in the sense that we do not have to compose the character out of recognizable subcharacters or radicals that occupy subportions of the character space. However, a character space can be broken up into subspaces horizontally or vertically and in various proportions, as shown in Fig. 3. We call any

Table 1 Symbol key position assignments.

a	ㄥ	A	幺	n	巛	N	女	1	米	!	[illegible]
b	乚	B	巴	o	口	O	冂	2	一	@	言
c	㇏	C	辶	p	コ	P	乙	3	二	#	〃
d	㇇	D	亅	q	丨	Q	勿	4	欠	$	金
e	雨	E	臼	r	㇏	R	亻	5	人	%	鬥
f	㇈	F	丸	s	丶	S	氵	6	匚	¢	一
g	勹	G	門	t	亅	T	小	7	丿	&	虍
h	㇙	H	戶	u	ω	U	其	8	馬	*	隹
i	一	I	一	v	丨	V	丨	9	㇇	(	又
j	丿	J	扌	w	灬	W	火	0	卜	)	甘
k	乛	K	亠	x	乂	X	癶	,	㇏	^	戈
l	了	L	阝	y	月	Y	舟	.	丶	:	
m	丿	M	木	z	艹	Z	⺮	/	八	?	寸

Figure 2 Scaling and positioning keys.

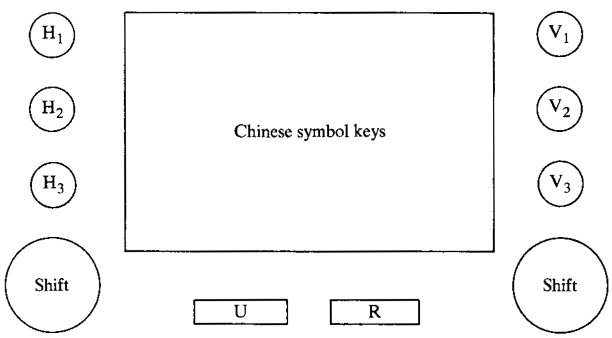
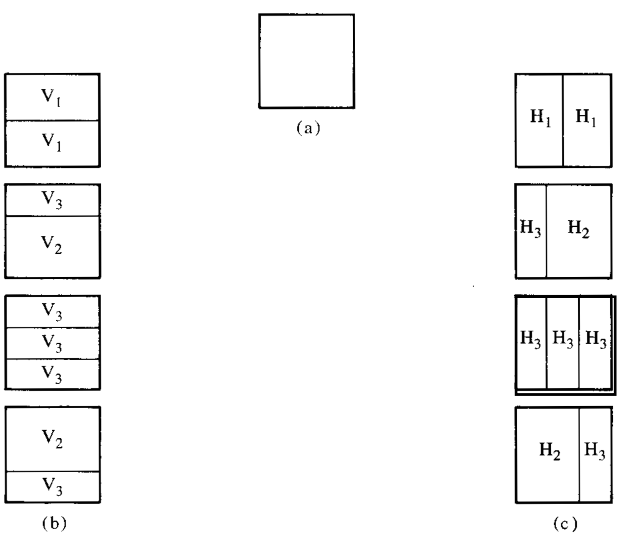


Figure 3 Outside block (a), vertical subblocks (b), and horizontal subblocks (c).



one of these subspaces a subblock. By *outside block* we mean the block that includes the entire character space, as in Fig. 3(a).

Figures 3(b) and 3(c) show how the various subblocks are labeled with a numerically subscripted H or V, which indicates that the subblocks are to occupy $\frac{1}{2}$, $\frac{2}{3}$, or $\frac{1}{3}$ of the horizontal or vertical dimension of the outside block. The outside block is then an *enveloping block* with respect to the depicted subblocks. In turn, any subblock can become an enveloping block by containing subblocks of its own. When we refer to an enveloping block, we mean the immediate enveloping block that has the same length in one of its dimensions as its immediate subblocks.

In the examples to follow, we show how appropriately subscripted H or V symbols and semicolons are used to bracket the symbols that are to occupy the subblocks.

Table 2 Keying sequence for four characters.

Example	*Character*	*Keying sequence*	*Number of key strokes*
1.	[illegible]	$g_U iv2_U \Delta$	5
2.	[illegible]	$\Phi q¢\Delta$	4
3.	[illegible]	$m_U 3V¢\Delta$	5
4.	[illegible]	$03V\Delta$	4

Table 3 Character construction using subblocking.

	Character	*Keying sequence*	*Number of keystrokes*
5.		V_1dm; V_1o; Δ	8
6.		GV_1; V_1o; Δ	7
7.		H_1A2; H_1y¢; Δ	9
8.		V_1e; V_1H_1M; H_1y2;; Δ	13

We refer to a numerically subscripted H or V symbol as a *header* and to a semicolon as a *tail*. Thus, for example,

H_3IV;

means that the symbols given by IV will occupy a horizontal space $\frac{1}{3}$ as large as its enveloping block. This header and tail bracketing is used for all subblocks. The outside block, however, needs no header, so none is used; as already noted, the tail for the outside block has its own special symbol Δ.

The relative scaling information for each subblock is thus given by the header. For the positioning of each subblock, we take advantage of the natural directions inherent in writing Chinese characters: from left to right and/or from top to bottom. Hence, whenever subblocks occur either horizontally or vertically, their ordering implies their position: left to right for horizontal subblocks and top to bottom for vertical subblocks. See Fig. 4. Figure 4(b) is an example of the nesting of subblocks, in that the two horizontal blocks are nested within the second vertical block. What about symbols that straddle subblocks? In Fig. 4(a) let us assume there is a symbol (c) that straddles H_2 and H_3; this symbol can be keyed in any one of the following three ways:

cH_2(   ); H_3(   ); Δ

or

H_2(   ); cH_3(   ); Δ

or

H_2(   ); H_3(   ); cΔ

Note that it is the outside block that envelops the H_2 and the H_3 subblocks. (See example 6 below.) Examples using subblocking are shown in Table 3.

In example 5, as already mentioned, the numerically subscripted V_1 indicates that the symbols between V_1 and; are to be assembled, scaled, and positioned to occupy $\frac{1}{2}$ of the vertical block space. The next set of symbols enclosed within the V_1 and ; then occupy the remaining $\frac{1}{2}$ of the vertical block (V_2 indicates occupancy of $\frac{2}{3}$ of a vertical block; V_3 indicates occupancy of $\frac{1}{3}$ of a vertical block; the same subscripting is used for horizontal block occupancy in H_1, H_2, and H_3).

Example 6 shows a symbol in the outside block (as in examples 1-4) combined with symbols in subblocks. The following sequences would also result in the same construction.

V_1; GV_1o; Δ

V_1; V_1o; GΔ

Note that the outermost block has no bracketing symbol on the left but has its Δ on the right. Example 6 also shows the use of an empty block V_1; which allows the positioning of V_1o; in the desired location.

Example 7 is like example 5 except that we have horizontal instead of vertical subblocking.

Example 8 shows how nesting of subblocks takes place. In the example the outside block is partitioned vertically in half. The lower half, in turn, is constructed by partitioning it into two horizontal subblocks. This example also demonstrates a situation in which the user can omit some semicolons; the two semicolons preceding the end of character, Δ, can be omitted. Thus only 11 keystrokes are really needed:

V_1e; V_1H_1M; H_1y2Δ

Each of the 39 lower case symbols and each of the 39 upper case symbols can be shifted one-fourth space upwards or one-fourth space towards the right. Part of the symbol may thus be lost, and this is intentionally used for two of the symbols

 and .

Figure 4 Implied positioning of subblocks.

H_2( ); H_3( ); V_3( ); V_2H_2( ); H_3( );;

(a)   (b)

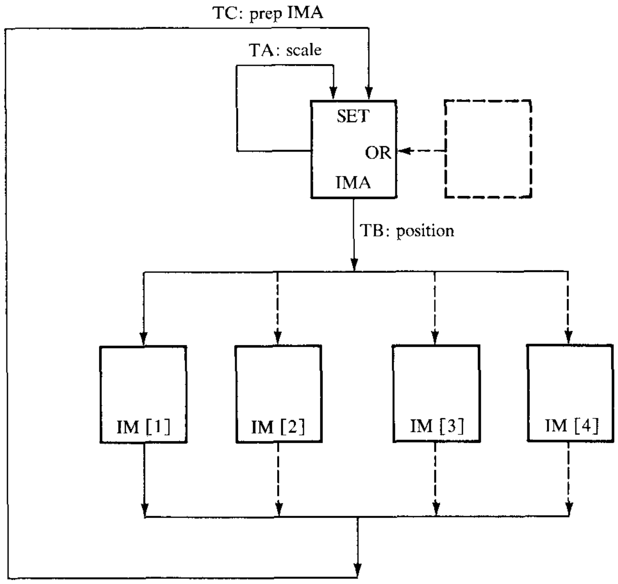


Figure 5 Data flow in construction of characters.

which shifted upward become

; and ⟶ ,

respectively.

In the examples given in this paper, the sequences of strokes that pertain to each subblock are indicated by giving the letter, number, or symbol that corresponds to each required key position on the keyboard. Movements upward or towards the right are indicated by U and R subscripts, which indicate simultaneous depression of the U or R bar, respectively.

In summary, then, the following rules apply to use of the keyboard.

- When no subblocking is required, only the outside block is needed to construct a character.
- The outside block needs no header and is terminated by the key Δ
- Subblocks are created and bracketed by striking one of the keys $\{H_1, H_2, H_3, V_1, V_2, V_3\}$ first, then striking whatever symbol keys are needed to construct the pattern in the defined subblock, and finally closing the subblock by striking the key ; .
- Subblocks can be nested to any number of levels. (Our initial design allows up to four levels of nesting, not counting the outside block).
- Subblocks at the same level of nesting follow an implicit positional order rule associated with how a Chinese character is manually written: from top to bottom and/or left to right. (Empty subblocks can thus be used for positioning subsequent subblocks.)
- Simultaneous depression of the U or R bar at the time that a symbol key is struck results in upwards or rightwards movement, respectively, of the symbol.
- Within each subblock, the order of striking symbol keys is immaterial.
- In cases such as example 8, in which the end of character symbol Δ is preceded by a number of semicolons, the semicolons can be omitted.

Internal logic

The internal logic for implementing the present approach can be readily designed [1]. We summarize here some preliminary design considerations.

Figure 5 shows a data flow for internal logic that is activated by sequences of keystrokes to produce the desired character patterns from the component symbols. Each of the basic symbols assigned to a key (lower case or upper case) can be stored as a dot matrix in bit storage, one bit per matrix element. The logic applies to any size matrix; typical sizes are 24×24 or 36×36 dots. The dotted square with an arrow pointing to the square labeled IMA is intended to represent any one of the 78 dot matrix symbol representations. Location IMA is then a matrix storage area with the same dimensions as any one of the 78 symbol dot matrices. Patterns from among 78 stored symbols are gated into IMA so as to be logically ORed with whatever patterns already exist in IMA. If either the up bar (U) or the right bar (R) is depressed while a symbol is being keyed, the logic performs either at 25% upward or 25% rightward movement of the pattern before ORing the symbol pattern into the contents of IMA. If the contents of IMA need to be scaled, the scaling logic, using the contents of IMA as input, *sets* the scaled output back into IMA.

The scale function compresses the contents of IMA either in the horizontal or in the vertical dimension based on whether the subblock header is H_1, H_2, H_3, V_1, V_2, or V_3. As noted, the scaling information given by the header is relative to the dimensions of its enveloping block. Therefore, the pattern information that is loaded into IMA for compression is always in a sequence such that the correct relative compression takes place.

Header H_1 causes only horizontal compression by $\frac{1}{2}$ via omission of every other horizontal pattern bit. Header H_2 causes an apparent horizontal compression of $\frac{2}{3}$ by the expedient of compressing by $\frac{1}{2}$ and then recentering the results. Header H_3 causes a horizontal compression by taking only every third horizontal point. Headers V_1, V_2, and V_3 cause similar compressions but in the vertical dimension only.

Finally the scaled or compressed pattern is returned to IMA such that the upper left-hand corner of IMA is the origin or registration point for the upper left-hand corner of the compressed pattern.

As with IMA, the IM[I] (I = 1, 2, 3, 4) are matrix storage areas, each with the same dimensions as IMA. They are used to assemble subblocks at various levels of nesting. Thus, subblocks that have the outside block as their enveloping block are assembled in IM[1]; IM[2] is used to assemble the next level of subblocks, and so on. (How this occurs for some of the examples is given later.)

Scaled subblocks are transferred from IMA to the appropriate IM[I], and in the process the subblock is properly positioned for ORing into the IM[I].

Associated with each IM[I] there are two position parameters, one for the horizontal and one for the vertical dimension of IM[I]. Together these two parameters provide the positioning function with the $x - y$ position that serves as a registration point for the upper left-hand corner of the scaled image being transferred from IMA. The positioning parameters are kept updated after each transfer from IMA. The information needed to update the positioning information comes from the subblock header.

Traces for examples 3 and 8 are shown in Tables 4 and 5, respectively. The trace for example 8 shows how the relative scaling and positioning taking place in IMA and IM[I] are recursively coordinated to produce the correct absolute scaling and positioning. Note how this is accomplished by completing the processing of the subblocks from the inside out. Note also the working of the implied ordering for the subblocks: top to bottom and/or left to right. The example also shows how IM[2] is cleared (and its position control parameters reinitialized) ready for reuse. This need for reuse can be readily seen by the reader if he visualizes an example in which, instead of having the enveloping block (assembled in IM[1]) partitioned into two vertical subblocks, he visualizes a partitioning into three vertical subblocks. It will then be clear that IM[2] is ready for reuse to assemble what is to eventually go into the third vertical subblock, all appropriately scaled and positioned.

Table 4 Logic trace for example 3.

Keying sequence	*IMA*	*IM[1]*	*IM[2]*	*IM[3]*	*IM[4]*
m_u	丿				
3	⺈=				
V	牛				
¢	生				
Δ	生				

Table 5 Logic trace for example 8.

Keying sequence	*IMA*	*IM[1]*	*IM[2]*
V_1			
e	雨		
:	雨	雨	
V_1		雨	
H_1		雨	
M	木	雨	
:	木	雨	木
H_1		雨	木
y	冃	雨	木
2	目	雨	木
:	目	雨	木
:	木目	霜	
Δ	霜	霜	

Character shapes, keying sequences

Even though the choice of symbols listed in Table 1 is preliminary, it represents an example of what might be an adequate set. To test its adequacy, the set of 214 Chinese radicals out of which all Chinese characters are composable was "keyed," i.e., formed manually following the rules.

Appendix A gives the keying sequences for these radicals. Approximations of the shapes resulting from the typing sequences are given in [1].

Appendix B gives the keying sequences for some of the more frequently occurring characters [6] and also the approximations of the shapes resulting from these sequences.

Table 6 gives the keying sequences and the approximate resulting shapes for the following sentence relating to acupuncture: "Acupuncture is a curing technique developed from out of the midst of our peoples' struggles and battles with sickness and disease." Table 7 is the adage: "Unless there is some reason, one simply does not approach the temple of the three precious ones." (Loosely translated this means: "Here comes a man with an axe to grind.") Table 8 shows the keying sequence for: "With courage and determination, success is attainable."

Conclusions

An approach to keyboard composition of Chinese characters has been described that does not impose upon the

Table 6 A sentence on acupuncture.

Character	*Keying sequence*	*Number of keystrokes*
針	$H_1\$;\ H_1iV\Delta$	7
灸	$V_1m_Uqr_RV_1Q\Delta$	8
是	$V_1Oi\text{¢};\ V_1\phi_Um_Uc\Delta$	10
我	$H_1j_Uj;\ H_1\wedge\Delta$	7
国	$Oi_Uiv2_U\cdot\text{¢}\Delta$	8
人	$7r\Delta$	3
民	$V_3p;\ hV_2I_U,\ \Delta$	8
在	$I_UH_1R;\ H_1iv\text{¢}\Delta$	9
同	$Oi_Uo\Delta$	4
疾	$s_UI_UqH_3S_U;\ H_2V_3;\ V_2m_U35\Delta$	14
病	$s_UI_UqH_3S_U;\ H_2iq5\Delta$	11
作	$H_3R;\ H_2R3_Ri_R\Delta$	8
斗	$S_U\text{¢}_UV_r\Delta$	4
争	$V_3m_Uu;\ V_3pI;\ t\Delta$	10
中	$g_U2_UV\Delta$	4
发	$a_U7(_R\colon\ \Delta$	5
明	$H_1oi;\ H_1y_U\Delta$	7
的	$H_1m_Uoi;\ H_1m_Udi\Delta$	10
一	$I\Delta$	2
种	$H_1j_UM;\ H_1g_U2_UV\Delta$	9
治	$H_3S;\ H_2V_1ar;\ V_1g_U2_U\Delta$	12
疗	$s_UI_UqH_3S_U;\ H_2I\Delta$	9
方	$s_UI_Um_UD\Delta$	5
法	$H_3S;\ H_2V_1iv\text{¢};\ V_1a_Ur_U\Delta$	13

Average number of keystrokes: $187 \div 24 = 7.8$

Table 7 A Chinese saying.

Character	*Keying sequence*	*Number of keystrokes*
无	$I_UI7b\Delta$	5
事	$V_1I_Ug2V_U;\ V_1p_UI_U;\ t\Delta$	11
不	$9_Utr\Delta$	4
登	$xio/\text{¢}\Delta$	6
三	$I_Ui\text{¢}\Delta$	4
宝	$KV_3;\ V_2i_Uiv\text{¢}.\Delta$	10
殿	$H_1V_3p;\ qV_3\#_U3;\ V_3m_Ur_U;;\ H_1V_1qf;\ V_1(\Delta$	21

Average number of keystrokes: $62 \div 7 = 8.9$

Table 8 Another Chinese saying.

Character	*Keying sequence*	*Number of keystrokes*
有	$I_U7y_R\Delta$	4
志	$V_1IV2;\ V_1qbs_R\colon\ \Delta$	11
竟	$V_1si/\text{¢};\ V_1o_Ui_Umb_R\Delta$	12
成	$H_1qI_Ud;\ H_1\wedge\Delta$	8

Average number of keystrokes: $35 \div 4 = 8.8$

user the need to learn an unnatural system to describe the character he is trying to type. Instead the user has only to strike keys corresponding to symbols with which he is thoroughly familiar, since he uses them constantly in the manual writing of Chinese.

He only needs to learn the position of a reasonable number (39, with lower and upper cases) of key positions, and we believe that the subblocking approach can also be readily learned, since the user subconsciously follows a similar approach in manually writing Chinese.

The user can also construct new or unfamiliar characters. Because there is no internal character dictionary as such, he does not have to worry about choosing among a subset of dictionary characters in the event that a dictionary index does not access a unique character. Also he does not have to worry about what to do if the character is not in the dictionary.

Of course, much work remains to be done. Further work is needed particularly in the areas of standardization and simplification of frequently used characters. This is, of course, a function of the area of usage, e.g., technical, commercial, governmental, with many varying preferences that can be resolved only after adequate study.

Acknowledgment

The author wishes to thank S. C. Hsieh for her valuable comments, discussions, and participation in the work reported here.

References

1. E. F. Yhap, "A Stand-alone Chinese Typewriter," *Research Report RC 4764*, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1974.

Nongeneric composition systems

2. O. Fujimura and R. Kagaya, "A Pattern-structural Code for Kanji," *First USA-Japan Computer Conference Proceedings*, 1972, p. 287. (Composition rules defined between each pair of strokes).
3. C. S. Hsieh, Y. Huang, S. Lin, and K. Hsu, "The Chiao-Tung Radical System," *Proceedings of the First International Symposium on Computers and Chinese I/O Systems*, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, 1973. (496 radicals are used, together with weighted variations for scaling; 11 fixed composition rules.)

Indexing/prompting systems

4. G. King and H. Chang, "Machine Translation of Chinese," *Sci. Am.* **218,** No. 6, 124 (1968). (Prestored dictionary is used, and user keys a radical, which causes display of choice of possible characters.)
5. F. Fang, C. Liu, and D. Tang, "An Intelligent Terminal for Chinese Character Processing," *Proceedings of the First International Symposium on Computers and Chinese I/O Systems.* Institute of Mathematics, Academia Sinica, Taipei, Taiwan, 1973. (Provides facility for more than one way of indexing and narrowing down display of choice of characters.)
6. S. Kuno, H. Hayashi, and S. Duncan, "Graphical Input/Output of Nonstandard Characters," *Commun. ACM* **11,** 613 (1968). (Provides facility for more than one way of indexing and narrowing down display of choice of characters.)
7. E. Pugh and C. C. Tsao, "Chinese Input-Output with Standard IBM Selectric Typewriter Terminal," *Proceedings of the First International Symposium on Computers and Chinese I/O Systems*, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, 1973.

Statistical frequencies of character usage

8. S. Lin, C. Hsieh, and Y. Hwang, "Analysis of Chinese Radicals," *J. Nat. Chiao-Tung Univ.* (1972). (Analysis of 9130 characters.)
9. K. L. Su, "Creation of a Set of Alphabets for Chinese Written Language," *Proceedings of the First International Symposium on Computers and Chinese I/O Systems*, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, 1973. (Analysis of 4,600 characters by which an alphabetic code can be used to identify characters.)
10. Y. S. Liu, "New Approach on Chinese (Tele)Typewriter which can be Used for Telex, Data Terminal and Computer I/O Devices," *Proceedings of the First International Symposium on Computers and Chinese I/O Systems*, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, 1973. (3,000 most used characters in Chinese newspapers.)
11. A. Ishii, Y. Hagiwara, Y. Masui, and Y. Aida, "Photoelectrostatic Kanji Printer," *Proceedings of the First International Symposium on Computers and Chinese I/O Systems*, Institute of Mathematics, Academia Sinica, Taipei, Taiwan, 1973. (5376 often used Kanji characters on a rotating disk.)

Appendix A: Chinese radicals

This appendix provides the keying sequences for the 214 Chinese radicals. These radicals are numbered in their standard numerical order.

1.	I
2.	V
3.	s
4.	7
5.	P
6.	t
7.	3
8.	sI
9.	5
10.	mb_R
11.	m,
12.	mr
13.	0
14.	K_U
15.	S_U
16.	gf
17.	q¢V_R
18.	dm
19.	d7
20.	$m_U d_R$
21.	bj
22.	6
23.	6
24.	IV
25.	Vs_R
26.	d_UV
27.	I_Uq
28.	ar
29.	(
30.	o
31.	0¢
32.	iv¢
33.	Iv2
34.	m_U(
35.	m_U(
36.	m_U9s
37.	I7r
38.	N
39.	lI
40.	K
41.	?
42.	T
43.	I_U7b_R
44.	V_1p; q
45.	V_1q¢V_R; 7
46.	VV_1; V_1q¢V_R;
47.	n
48.	3v
49.	V_1p; b
50.	gV
51.	3_UV
52.	A
53.	$s_U I_U$q
54.	H_1l; c
55.	I$\#_U$
56.	$I_{U'}$:
57.	V_1p; V_1d;
58.	pi
59.	V_3j; V_3j; V_3j;
60.	j_UR
61.	/V
62.	ˆ
63.	$V_1 s_U$p; q
64.	j
65.	$i_U v_U$(
66.	$V_1\phi$; V_1(;
67.	$s_U I_U$x
68.	s s_UIV_R
69.	j_UqR
70.	$s_U I_U$mD_R
71.	I_UI$m_U b_R$.
72.	oi
73.	g_Ui2_U
74.	y
75.	M
76.	4

77. ϕq¢
78. $i_U m_U$9s
79. V_1qf; V_1(;
80. oIv
81. H_1vj; H_1bj;
82. j3b_R
83. $j_U$hx
84. V_3m3_R; V_2f;
85. S
86. W
87. j_Uu
88. $m_U r_U$x
89. v_1x; V_1x;
90. V_1a; $V_1$9; V_R
91. qV_3V; $V_3$3; V_3V;
92. $V_1$6; tm
93. m_U3V
94. I7r:
95. $s_U I_U$A
96. $i_U$iv¢.
97. $j_U$qar$,_R$
98. I_UF2
99.)
100. m_U3V¢
101. 03V
102. $g_U$iv2_U
103. $k_U \phi m_U$C
104. $s_U I_u$q$H_3 S_U$;
105. X
106. m_Ug2_U2
107. kqv_U(
108. g#¢
109. y2
110. $V_3$9; $s_U$ktm
111. $m_U i_U$I7r
112. $I_U m_U$o
113. s_U9Vr
114. gar
115. j_Um
116. Kmr
117. $s_U i_U$/¢
118. Z
119. 1
120. A2
121. $m_U i_U$iv¢r
122. gH_1x; H_1x;
123. $/_U i_U$3V
124. $H_1 DS_U$; $H_1 DS_U$;
125. V_1iV¢; j V_1bj;
126. $I_U m_U$g#
127. 3_UM
128. I_Uq3V_R¢
129. V_1pI; $V_1 I_U$I; V
130. 0$V_1$5; $V_1$5;
131. $V_3$9; qV_3p; V_3V¢;
132. m_Uy2
133. $V_1 I_U$ar; V_1iV¢;
134. E
135. $j_U V_1$IV; V_1o;
136. $H_1 m_U$9s; $H_1 m_U i_U$IV;
137. Y
138. Hr
139. $V_3 j_U$9; V_2B;
140. z
141. &
142. oV¢.
143. m_Ug#¢
144. $H_1 j_U$R; $H_1 3_U$t;
145. $s_U I_U H_3$R; H_2x;
146. I_Ug$\#_U$#¢
147. V_2y; $V_3$7b_R;
148. $V_3 m_U$9; $V_2$03V;
149. @
150. $m_U r_U 5 V_1$; V_1o;
151. $V_1 i_U$o; V_1/¢;
152. I_UQ:r
153. $V_1 j_U$u; V_1Q;
154. V_2y2; $V_3 5_U$;
155. V_3iV¢; $V_2 \#_U$#mr;
156. V_3iV¢; $V_2 \phi m_U$c;
157. V_3o; $V_2 \phi m_U$c;
158. $V_1 j_U$q3¢; Dj
159. $i_U g_U$i2_U¢V
160. $V_1 s_U i_U$/¢; $V_1 I_U$V;
161. i_U3$m_U V_1$; V_1hx_R;
162. C
163. L
164. V_3I#; $V_2$0$\#_U$i¢;
165. j_U1
166. V_1giv2; V_1iv¢;
167. \$
168. $V_2 i_U$q3¢; V_3hx_R;
169. G
170. L
171. V_1piV; V_1T/;
172. *
173. e
174. $V_1$3V¢; V_1y;
175. $H_1 7_R i_U$3; H_1qi_U3;
176. V_3Ij; $V_2$0o¢;
177. zi_Uo¢V
178. V_3pV; V_3o; $V_3 m_U i_U$IV;
179. $H_1 V_R i_U$3; H_1qi_U3; ¢
180. $V_1 s_U i_U$/¢; V_1g2_U2;
181. V_3Ij; $V_3 y_U$; $V_3 5_U$;
182. qfx
183. fs_R.
184. $5_U s_U$Hr

185. V_3/Ij; V_2y2;

186. V_1j_UM; V_1g2_U2;

187. 8

188. V_3gx; V_2K_Uy;

189. $V_3s_Ui_U$o; $V_2$0o;

190. $H_1V_2qi_U$3¢; V_3ar;; H_1V_3j; V_3j; V_3j;;

191. %

192. V_1xq¢V_R; V_1bj;

193. V_3i_Uo; $V_2$0/$_U i_U$v;

194. $V_1m_Ug_U$iv2_U; $V_1m_Ub_Ra_Rr_R$;

195. V_3m_U9; $V_3$0IV¢; V_3w;

196. V_1s_UH; V_1dI;

197. $V_3\phi$; $V_2$0x¢;

198. s_UI_UqV_3; V_3p#$_U$; V_3H_1bj; H_1bj;;

199. $V_1$3V¢; V_3m_U);

200. s_UI_UqH_1M; H_1M;

201. zV_3; $V_3I_Ug_U$iv2_U; V_3mr;

202. V_1j_UM; V_15_UT/;

203. V_1g2_Uv2; $V_1i_Uv_U$Iw;

204. V_3#/¢; V_2/$_U$gT;

205. $V_3\sigma$; V_2g_Ui2_Ub_R;

206. $V_2y_U2_U$q¢V_R; $V_3H_1$9V_R; H_1qi,;;

207. H_1V_3iV2; V_3o; V_3/2;; $H_1i_Uv_U$(;

208. V_1E; V_1hh_R3,$_R$;

209. V_3j_Uy2; V_3g_Uiv2_U; V_3I_U#$_U$#;

210. $V_1s_UI_U$x; V_1#$_U$#;

211. $V_1\phi$q¢; $V_1$5q¢V_R;

212. $I_Um_Up_Rj_R$;

213. V_3m_U9; V_30iV¢; V_3b_R;

214. $V_3$5i; V_3H_3o; H_3o; H_3o;; $V_3$0I#$_U$#;

The symbol

is a suggested symbol for both

 and .

Modern representations are used where it results in very desirable simplication. Some examples are:

黾(黽) 龟(龜) 飞(飛)

Appendix B: Typical sequences

Here we provide the keying sequences for some frequently occurring characters, as determined by Hsieh. The average number of key strokes for these twenty-five characters is 9.0 strokes per character.

Character	*Keying sequence*	*Number of keystrokes*
的	H_1m_Uoi; H_1m_UdisΔ	10
是	V_1oi¢; $V_1\phi m_U$cΔ	10
有	I_U7$y_R\Delta$	4
他	H_3R; $H_2d_UV_U$bΔ	8
這	@$_R$CΔ	3
国	0$i_U$iv2_U¢·Δ	8
們	H_3R; H_2GΔ	6
説	H_3@; H_2V_3/; oV_2mb$_R\Delta$	12
个	5_UVΔ	3
就	$H_1s_UI_U$oV_1; V_1T;; H_1I7b_R; Δ	16
要	V_1I_Ug#$_U$#2; V_1NΔ	10
全	5_Ui_Uiv¢Δ	6
到	H_1I_UaIV_1; V_1iV¢;; H_1v$t_R\Delta$	16
你	H_3R; $H_2V_1m_U$d;TΔ	10
时	H_1oi; H_1?Δ	7
以	H_1ar; $H_1$7rΔ	8
那	H_1D3V; H_1LΔ	8
裡	$H_1s_UI_U$R3_R; $H_1V_1g2_U$2; 2_UV¢Δ	17
和	H_2j_UM; H_3oΔ	7
道	CV_3/; $I_Uj_UV_3$y2Δ	10
得	H_3j_UR; H_2V_1oi¢; V_1?Δ	13
家	KiV_3; V_2QH_2; H_3j_U, Δ	12
么	$m_U$arΔ	4
怎	V_3m_U; qi_UoΔ	6
样	H_3M; H_2V_3/; i_Ui¢$_U$VΔ	12

Appendix C: Gross estimates of keying speed

From the number of keystrokes given in Appendices A and B we can make some very gross estimates of keying speed.

- *Optimistic estimates*

From Appendix B we can use the average figure of 9 keystrokes per character. If we optimistically assume that a user can attain a proficiency level of five strokes/second, then we obtain

$$\left(5\frac{\text{strokes}}{\text{sec}} \times 60\frac{\text{secs}}{\text{min}}\right) \div 9\frac{\text{strokes}}{\text{char}}$$

$$= 33 \text{ characters/minute}$$

- *Pessimistic estimate*

From Appendix A, one can calculate an average of 5 keystrokes per radical. If we assume a worst case of 3 radicals per character, we can use a worst case average figure of 15 keystrokes/character. If we then pessimistically assume that a user can only strike two keys per second, then we obtain

$$\left(2\frac{\text{strokes}}{\text{sec}} \times 60\frac{\text{secs}}{\text{min}}\right) \div 15\frac{\text{strokes}}{\text{char}}$$

$$= 8 \text{ characters/minute.}$$

Received February 22, 1974

The author is located at the Thomas J. Watson Research Center, Yorktown Heights, New York, 10598.