Thumbcode: A Device-Independent Digital Sign Language

Vaughan R. Pratt
Department of Computer Science
Stanford University
Stanford, CA 94305-9045
Email: pratt@cs.stanford.edu
Phone: 650-723-2943

Abstract

This paper describes Thumbcode, a device-independent digital sign language. We list and discuss requirements, then describe Thumbcode and explain how it meets these requirements. We propose several approaches to the machine recognition of Thumbcode, and conclude with a brief discussion of potential ergonomic health issues.

1 Background

The rumors of the demise of text input have been much exaggerated, and it continues to be in demand by many though far from all users of computers large and small. The most effective text input device has proved to be the full-sized keyboard, having a lateral key pitch of 19 mm and vertical key travel of 3-4 mm. But while these dimensions are well suited to desktop computers and $8\frac{1}{2} \times 11$ laptops, they are not a comfortable fit to the dimensions of personal digital assistants (PDA's) and wearable computers.

The ideal wearable computer approaches the utility and usability of one's desktop computer while interfering as little as possible with one's routine activities. In practice one settles for a computer that is 3 to 5 years behind the times in capacity and performance, which technology has in the meantime shrunk to wearable proportions.

The most shrink-resistant components are the monitor, keyboard, and power source. But even with the monitor and power source there is considerable opportunity for further shrinkage [1]. A head-mounted display that writes on the retina can be tiny while producing an apparently huge display, while every reduction in computer power dissipation allows the size of the battery to be reduced, for a given battery life.

The keyboard is particularly hard to shrink. To achieve input speeds of better than 60 words per minute (wpm)[1] appears to require two-handed typing and therefore a large keyboard.

As keyboards shrink below the size at which regular touchtyping is feasible, speed quickly falls off. For example a good touchtypist using a PDA-scale keyboard such as those on the HP-200, Psion, Sharp Zaurus, or IBM PC-110 can expect to achieve around 30 wpm with the additional practice necessitated by the small size. Virtual or popup keyboards such as on the PalmPilot and Newton, operated by touching a pen to a touch-sensitive screen superimposed on an image of a miniature keyboard, are somewhat slower. But to even approach PDA keyboard speeds the user must pay close attention to the screen while typing, unlike the situation with touchtyping where the feel of the keys provides adequate feedback allowing the user's eyes to be elsewhere, e.g. reading a document being transcribed.

Handwriting recognition tends to be slower yet. The 30 wpm that a writer can typically reach drops to below 20 wpm when writing with the care demanded by today's handwriting recognition software. Furthermore users who are sufficiently attuned to the software to be able to write without losing a further factor of 2-4 to errors made by the recognition software seem to be in a distinct minority; handwriting recognition though much improved since the early days of the Apple Newton still remains an iffy proposition.

Graffiti is a pen-based text input notation used on the PalmPilot. It is like handwriting but more styl-

[1] We define a word as six characters, regardless of whether the characters are letters, numbers, punctuation, or space. All speeds quoted are for straight text input without the assistance of macros. Macros are a wild card in such statistics in that they can improve performance 30-70% or more with sufficient training. Typical users will begin with straight text and can look forward to further improvements when they start making serious use of macros.

ized, which makes recognition considerably more reliable and allows the 20 wpm potential of handwriting to be achieved in practice with only a few hours of experience. Graffiti could in principle be used head-up, but in practice one tends to drift off the input region so the user ends up with head down paying almost as much attention to the screen as with a popup keyboard.

All the above subsize solutions are tied to the computer, either because the keyboard is integral or the input is done via a touch-sensitive screen. In contrast to these, a chording keyboard is a detached device with typically 5 to 7 keys that are depressed simultaneously then released to form a character. One may think of it as consisting only of control keys. It was introduced by Doug Engelbart in the late 1960's simultaneously with the mouse, the idea being that one hand would be dedicated to each with the chord set being for text input (mainly two-letter commands) and the mouse for locating and picking. Both may be used head-up.

The Handkey Twiddler is a one-handed detached device that is a cross between a keyboard and a chord set, having a 3×4 array of keys under the finger tips and a ring of control keys under the thumb. Like the chord set it permits headup use and by some accounts permits very good performance (Thad Starner, 43 wpm, email of 2/16/97 to wearables@media.mit.edu). At a volume of about 12 cu.in. vs. the PalmPilot's 8 cu.in it is smaller than any regular keyboard but still has bulk it could usefully shed.

Thomas et al [3] compare a chord set, a virtual QWERTY keyboard, and a forearm QWERTY keyboard (for use by one hand while mounted on the other forearm). They measured speeds of users whose training consisted of six one-hour sessions. For text of between 50 and 65 characters, subjects achieved 12 wpm with the forearm keyboard, 5 wpm with the virtual keyboard, and 4 wpm with the chord set. While these rates are significantly lower than those given above we would expect them to improve noticeably with 100 or more hours of practice; 6 one-hour sessions is not enough to build up a mental model of the device that would permit the user to plan several keystrokes ahead while typing relatively fast.

2 Requirements

Our approach to text input was worked out in the context of the foregoing background, with the following more specific requirements in mind.

2.1 Device Independence

Device independence is less an end in itself than a means to scalability of keyboards for wearable computers, which is our real goal. In order to continue to benefit from shrinking component size and dissipation, all components must shrink together, the keyboard included.

The fundamental difficulty with a shrinking keyboard is that every factor of two in shrinkage calls for a radical revision in how the user approaches touchtyping and headup use. In particular imagine that the keyboard has been shrunk to invisibility. How could anyone possibly type on so small a device?

Scalability is the exact opposite of this: far from requiring radical changes in mode of use as the device shrinks, usage should be as independent of device size as possible.

We achieve this goal via device independence. which in turn is achieved by using the hand itself as the keyboard rather than any particular device. One types not just *with* the hand but *on* the hand.

If we were to permit two-handed use we could type with one hand on the other, or even both on each other.[2] However the benefits of one-handed use appear to us to outweigh the speed gain we would expect to be possible with two hands used in a device-independent way, which seems to us unlikely to be competitive with touch-typing on a full-size keyboard. We therefore assume one-handed typing, right or left as the user prefers, leaving the other hand free.

A hand "typing" on itself is a form of signing. Starner et al [2] have explored American Sign Language (ASL) as a word-oriented signing language. Their preliminary experiments were confined to a 40-word subset of ASL's 6000-word vocabulary, for which they achieved a per-word recognition accuracy of 97%.

In contrast Thumbcode is intended as a keyboard replacement, and as such is purely character-oriented. Thumbcode caters for all 128 7-bit ASCII characters as well as some of the other commonly used scan codes and features of standard PC keyboards such as cursor keys, function keys, the ALT key, simultaneous depression of SHIFT, ALT, and CTRL, etc. ASL is less well suited for this purpose: it does not have standard signs for all of ASCII, and those signs it does have for letters are quite ad hoc compared to the approach we propose here.

[2] Pressing the palms together, rotate the hands oppositely about an axis normal to the palms to make the fingers of each hand point in opposite directions. Now separate the hands slightly and let each hand "type" on the palm of the other.

What distinguishes Thumbcode from other sign languages is that it works very much like typing on a keyboard. One "types" with the thumb on the twelve phalanges of the fingers as though they formed a 3×4 keypad. (In this respect the fingers can be thought of as a Twiddler keypad built into the hand.) Simultaneously three bits of "control-key" information are provided by holding some fingers together and some apart.

Scalability is achieved by making no commitment to the method of recognizing Thumbcode as a sign language. No matter how small technology is able to shrink the recognition device, Thumbcode continues to be thumbed in exactly the same way as when recognizing it with bulkier devices.

2.2 Digital

"Digital" here has two meanings for us. First, a key advantage of digital computation over analog is freedom from error. By factoring a large state system as the product of a modest number of two- and three-state systems, we are able to maintain good separation of states in each component, which results in a much lower chance of error than with an analog representation of the large set of states. Applying this principle to text input helps keep down the rate of misidentified characters.

The second meaning of digital for us is computational, by which we mean specifically the ASCII character set, as opposed say to an alphabet one might use in setting a newspaper or typing a novel.

2.3 Usability

Just as one wants to minimize obtrusiveness of wearable computers, so is it desirable to minimize the obtrusiveness of methods of communicating with it, such as sign language. To this end a good sign language should be usable head-up as with touchtyping.

It would also be very useful if one could type even when very much engaged in some activity, e.g. while jogging, or lying on one's back under an automobile.

And it goes without saying that the faster one can type with any given system the more useful that system.

2.4 Learnability

Although learnability is not at the top of our list of criteria, we still attach significance to it. The more learnable a sign language is, the more casual users will be able to benefit from it.

Learnability divides up into two main concerns: learning the sign language itself, and learning the idiosyncrasies of particular devices for recognizing the language.

2.5 Implementability

Device-independence does not mean "no device," but rather that how the user signs should be the same no matter what device is used for recognition.

The more devices a system is compatible with, the more useful the system. The kinds of devices one would consider for accepting text input include keys and switches, gloves with embedded position sensors, tones injected into one part of the environment and searched for in other parts, and vision systems depending on either visible light or infrared light (useful for security at night).

2.6 Low stress

Typing is a notorious health hazard. Any system for text input that is likely to be used for extended periods must address all relevant ergonomic concerns.

3 Thumbcode: Description

Thumbcode is our proposal for meeting the requirements of the previous section. It is a one-hand device-independent chording notation usable equally well with the left or right hand. (In principle a sufficiently coordinated thumber could carry on two independent conversations one with each hand.)

Although usable for communication between humans, Thumbcode is primarily intended for human-computer interaction (HCI) with wearable computers. Unlike traditional hand sign languages such as American Sign Language [2], Thumbcode provides for the full ASCII character set and other standard keyboard codes. In addition its discrete and regular structure should facilitate machine recognition by a wide range of devices including switches, position and motion sensors, and cameras.

The human hand has four fingers, index, middle, ring, and pinky. Each finger consists of three *phalanges*, approximately one-inch rigid segments, which we shall refer to as tip, medial, and base. A Thumbcode character is signed or *thumbed* by pressing the tip of the thumb against one of the phalanges. This defines the twelve *thumb states* of Thumbcode.

At the instant the thumb presses a phalanx, adjacent fingers may be open or closed (pressed together).

Open | | | |

1	t	e	a
2	s	i	n
3	4	5	6

Shift Open || | |

!	T	E	A
@	S	I	N
#	$	%	ˆ

Pair | | ||

7	o	h	↩
8	d	r	␣
9	0	-	=

Shift Pair || ||

&	O	H	⌈
*	D	R	←
(	)	_	+

Trio ||| |

b	c	f	g
j	k	l	m
[	]	;	′

Shift Trio | || |

B	C	F	G
J	K	L	M
{	}	:	″

Closed ||||

p	q	u	v
w	x	y	z
,	.	/	\

Shift Closed | |||

P	Q	U	V
W	X	Y	Z
<	>	?	|

Table 1. Thumbcode Assignments
View of right-hand palm
Across: Pinky, ring, middle, index
Down: Tip, medial, base
↩ Return, ␣ Space, ⌈ Control, ← Backspace

The three pairwise closures are index to middle, middle to ring, and ring to pinky, defining three bits of information and hence eight *closure states*. In combination with the twelve thumb states this gives a total of 96 basic thumbcodes.

The closures are divided into unshifted and shifted. The unshifted closures are called Open, Pair, Trio, and Closed. In Open and Closed the fingers are held apart and together respectively. Pair holds just the index and middle fingers together, while Trio holds middle, ring, and pinky together. Each shifted closure is obtained from its unshifted counterpart by complementing (i.e. changing) whether or not the pinky is separated from the ring finger.

Thumbcode associates ASCII characters to these basic thumbcodes according to Table 1. (Tilde and backquote are accommodated with CTRL, see below.)

Each of the eight 3×4 arrays in Table 1 should be visualized as being superimposed on the fingers of the right hand. (Left handed thumbers should first reflect the tables laterally and then superimpose each table on their left hand.)

Except for the Control key, denoted ⌈in Table 1, the remaining 95 characters are ordinary ASCII characters, including Return (↩), Space (␣), and Backspace (←).

The Control or CTRL key is used to create compound thumbcodes. Its effect is to modify the meaning of the next thumbcode. The characters a-z (or A-Z) along with

```
@ ^ _ [ ] \
```

are modified in the same way as when CTRL is pressed on a standard keyboard (with CTRL-[being ESC). CTRL-digit (1 through 9 except 5) behaves like SHIFT on a numeric keypad: for the even digits, 2 denotes down-arrow, 4 left, 6 right, 8 up, and for the odd, 1 denotes END, 3 PG-DN, 7 HOME, and 9 PG-DN. CTRL-0 denotes Insert and CTRL-. denotes Delete. Backquote is obtained as CTRL-′ and tilde as CTRL-″.

The effect of CTRL on respectively ←, CTRL, and + (all on the same closure as CTRL for convenience) is to create tertiary thumbcodes, requiring a total of three thumb presses: CTRL itself, the character it modifies, namely BS, CTRL, or +, and one more character.

The effect of CTRL-BS on the third character is as though the ALT key of a standard keyboard were held down instead of CTRL. The effect of CTRL-CTRL is as though both CTRL and ALT were held down. And the effect of CTRL-+ is as though the third character were a function key, with 1-9 being F1-F9 and a-c being F10-F12.

Other combinations of control and shift keys can be obtained via CTRL-&, CTRL-, CTRL-(, CTRL-), etc. We leave these unspecified for the time being. One use for them would be for a numeric keypad: we have put the numeric symbols in a slightly awkward position, and for extended entry of numeric data it would be preferable to put the numeric symbols at more comfortable locations.

4 Rationale for Thumbcode

Thumbcode meets our requirements as follows.

Device independence is achieved by designing Thumbcode around the hand rather than around any particular device, while ensuring that it can be recognized by a variety of devices.

Thumbcode is *scalable* as a corollary of device independence: as technology permits more compact recognizers, those recognizers can be shrunk without limit. The hand itself of course does not shrink, but by definition no part of our body or normal clothing is an encumbrance, only what must be added to the hand for recognition of Thumbcode.

Thumbcode is *digital* in two senses. First it provides for the whole of the ASCII character set. Second it consists of a regularly organized system of discrete states. The regularity results from being factored as a product of three bits times four fingers times three phalanges per finger, the five *primitive states* of Thumbcode, which greatly simplifies recognition. Discreteness is the wide separation of those states, resulting from wide separation within each of the five primitive states, greatly reducing the likelihood of error.

American Sign Language is not digital in either of these senses. It does not provide for the full ASCII character set. And its ad hoc structure means that any recognition strategy will in general need to distinguish states at the level of characters rather than bits. It is much more work for both the user and the recognition method to ensure that every pair of ASL characters is adequately separated than the two, three, or four states in each of the five primitive factors of a Thumbcode state.

4.1 Usability

We have addressed the usability of Thumbcode first by ensuring that all states can be comfortably achieved, second by making all distinctions clear and easy to communicate, and third by classifying its states according to difficulty. For this last, the base phalanges are hard for the thumb to reach, as is the pinky when it is held apart from the ring finger (the Open and Pair closures in the unshifted case). This gives 28 "easy" unshifted thumbcodes, to which we assign the 26 letters, space, and carriage return.

Identifying which finger is being thumbed is progressively harder and hence less reliable for the Open, Pair, Trio, and Closed closures. The shifted closures are harder than the unshifted for the user as they require individual control of the middle-ring-pinky or *trio* group; it is relatively easy to open or close the whole trio. Unshifted ASCII characters are much commoner than shifted, so we assign these to the unshifted closures. We assign the commonest six letters to the easiest closure, namely Open, the next commonest four together with Space and Return to Pair, and the remaining 16 letters in alphabetical order to Trio and Closed.

The remaining ASCII characters are assigned to the remaining thumbcodes essentially according to how they are laid out on a standard PC keyboard, in aid of learnability.

4.2 Learnability

Learnability would ideally be addressed by using one or another well-known arrangement of the letters, e.g. alphabetical as on the Handykey Twiddler or QWERTY as on standard keyboards. Unfortunately the well-known arrangements are far from optimal with regard to usability.

However by careful design it is possible to combine a near-optimal arrangement with a reasonable degree of learnability. For the latter we have made use of four techniques: a mnemonic system for the ten commonest letters, alphabetical order combined with regularity of subsetting for the remaining letters, a close match to customary PC keyboard layout of numerics and punctuation, and assignment of pairs, namely parentheses, square brackets, curly brackets, and slashes, to pairs of base phalanges.

The rationale for this assignment is as follows. The ten most commonly used letters, along with Space and Enter, are assigned to the six easy phalanges of the easiest two closures, namely Open and Pair. Within those ten letters, the order is chosen for its mnemonic value, forming the words "tea," "sin," "oh," and "dr" (doctor). (Actually d is slightly less common than l and c, but moving d up ahead of l and c results in an attractively regular structure for the remaining 16 letters that should make it easy to learn.)

For the Trio and Closed closures, the thumb can reach the pinky more comfortably, and all four fingers are used in assigning the remaining 16 letters to the two easy phalanges. The layout of these letters in the alphabet, namely *bcg**fg**jklm**pq***uvwxyz, leads to a natural and therefore easily memorized grouping: bcfg, jklm, pquv, wxyz.

The numerics and - and = are laid out on the hard phalanges in the easy closures, in the order found on the keyboard. The remaining lower-case glyphs continue the keyboard order on the hard phalanges on the hard closures, followed by \ at the end (which is happy to be next to /).

4.3 Implementability

Machine recognition of Thumbcode may be implemented by any of several approaches described below. Some approaches are more technically challenging than others with respect to their design and initial implementation. All but the video approach however should permit economical mass-production, and the widespread availability of NTSC/PAL black-and-white cameras and framegrabbers makes even that approach practical.

4.4 Low stress

The more commonly used thumbcodes can be comfortably signed, by design. There is potential for stress with the less commonly used thumbcodes, particularly those in which the thumb has to reach across to the pinky or down to the base phalanges. Also potentially stressful are the shifted closurs, especially Shift Trio which holds just the middle and ring fingers together.

Stress is greatly reduced by judicious exercise of those degrees of freedom of the hand not constrained by the Thumbcode specification. It is natural for a beginner to picture the closure positions as being realized with a flat hand, with the fingers held straight and lying in a plane. However Thumbcode does not require this, and it is much more comfortable to thumb with the hand held in a relaxed cupped position, allowing the thumb to reach the tip phalanges without needing to stretch. (In this position the author finds Pair the most natural closure.)

Nor is it necessary to separate the fingers sideways. The pinky can be moved away from the ring finger by bringing it closer to the palm as well as sideways; likewise the index finger can be separated from the middle finger by straightening it. (Moving it towards the palm obstructs the thumb's access to the middle finger.)

In general the user should experiment with ways of holding the hand that realize the Thumbcode specifications with the minimum of discomfort. Hand anatomy varies widely, and different people can be expected to come up with different positions that maximize comfort.

5 Recognition Devices

In this paper we consider three approaches to the machine recognition of Thumbcode, namely switches, electrical tones, and video. The detailed design of these approaches is beyond the scope of this paper, and we shall confine our treatment of each approach to its general principles.

5.1 Switches

The switch approach places twelve contacts somewhere on the palm or thumb side of the phalanges and one the tip of the thumb. These permit detection of a thumbcode and identification of the thumbed phalanx. In addition three switches are placed one between each adjacent pair of fingers, permitting identification of the closure.

The twelve finger contacts call for twelve wires leading from the hand to the encoding device, and the interdigit switches an additional four assuming a common ground. That ground can also connect to the thumb, for a total of fifteen signal wires and one common ground.

5.2 Tones

Four easily distinguished tones in the vicinity of a kilohertz are coupled to the tip phalanx, one tone per finger. At the tip of the thumb is a probe which the user presses on the skin of the target phalanx. The probe is monitored to determine which frequency is loudest, indicating the finger, and by how much, indicating which phalanx. This assumes that the thumbed finger's tone gets softer and the other fingers' tones get louder nearer the base of the thumbed finger. This approach has the advantage of making phalanx identification relatively insensitive to the resistance between the probe and the thumbed phalanx. This is necessary because the resistance can vary widely with pressure of the thumb and dampness of the skin.

Obvious choices of tones are the low group frequencies (697, 770, 852, and 941 Hz) or the high group frequencies (1209, 1336, 1477, and 1633 Hz) of the Dual Tone MultiFrequency (DTMF or TouchTone) standard used in telephony. These two groups have the advantage of universally available circuits for their generation and detection.

This scheme requires only five wires between the hand and the encoder: four for the tones to the fingers, and one for the probe on the thumb. No separate ground wire is needed.

5.3 Video

In the video approach a camera monitors the hand positions and attempts to infer from the position of the thumb which phalanx is being thumbed. It also estimates the degree of separation of each adjacent pair of fingers.

This is by far the most technically challenging approach. From the user's point of view however it is also by far the most convenient to use, as the hand remains unencumbered other than by the requirement that it be positioned and oriented to suit the camera. The software should be smart enough not to demand too many concessions by the user, which otherwise can become more of an encumbrance than the switches and contacts of the other approaches.

The ready availability today of miniature black-and-white NTSC or PAL cameras and framegrabbers means that the hardware side of this approach, while certainly sophisticated, is a largely solved problem. In contrast the software side involves determining the position and orientation of the hand as a whole and the phalanxes as its parts, and deciding when a character has been thumbed.

This approach is complicated by the need to take into account the variety of finger positions different users may adopt for comfort. It is further complicated by widely varying lighting and background conditions.

The optimal camera position has its line of sight to the fingers normal to the plane defined by the tip phalanges of the middle-ring-pinky trio when held comfortably in the Open closure. Furthermore the camera should be able to see the tip of the thumb at all times, achievable by positioning the hand and/or camera to view the outer side of the thumb rather than the back.

The determination of when a character has been thumbed is particularly difficult, requiring constant monitoring of a steady and high-speed video stream. This problem is greatly simplified by a hybrid approach involving a switch at the thumb tip which is closed when a character is thumbed. With this approach a video stream is no longer needed and it suffices to simply photograph the hand once at the moment of thumbing the character.

6 Health Hazards

We conclude with a brief and inconclusive examination of the ergonomic issues likely to be encountered with Thumbcode.

Among the health hazards real and imagined presented by computers, the keyboard features as a prominent offender. The author is as aware of these concerns as anyone, having had acute tendonitis in both elbows on different occasions in the distant past, apparently resulting from typing.

Thumbcode presents similar risks to regular keyboards, involving repetitive small motions.

Assigning the most comfortable thumb positions to the most frequently typed characters should not only improve typing efficiency but minimize stress on tendons, muscles, and joints. As noted earlier, the user can further reduce stress on tendons, joints, and muscles by finding the most comfortable positions consistent with reliable identification of thumbcodes.

One common recommendation with keyboards is to take breaks and give your hands a rest, or exercise them in some way different from typing. This is equally good advice for thumbing.

In conclusion, Thumbcode is a digital sign language carefully designed to accommodate the ongoing shrinking of wearable computers. Currently we have no experience with Thumbcode to justify our confidence in our design. However we hope in the near future to experiment with some of the devices suggested above for recognition of Thumbcode signing.

References

[1] M.B. Spitzer, N.M. Rensing, R. McClelland, and P. Aquilino. Eyeglass-mounted displays for wearable computing. In *Int. Symp. on Wearable Computers*, Boston, October 1997.

[2] T. Starner, J. Weaver, and A. Pentland. A wearable computer based American sign language recognizer. In *Int. Symp. on Wearable Computers*, Boston, October 1997.

[3] Bruce Thomas, Susan Tyerman, and Karen Grimmer. Evaluation of three input mechanisms for wearable computers. In *Int. Symp. on Wearable Computers*, Boston, October 1997.