Iconic Text Entry Using a Numeric Keypad

John Jannotti <jj@lcs.mit.edu>
Massachusetts Institute of Technology

Abstract

An *iconic* text entry system using the widely entrenched numeric keypad is presented. In an iconic system, text entry is designed to remind the user of existing notions of letter shape. For example, the letter "L" is entered by pressing buttons down the center of the keypad, and then to the right, calling to mind the shape of a capital L. The letter "O" is entered by pressing the digit 0, which resembles "O".

An iconic system has a number of advantages over systems designed solely for rapid input by expert users. First, an iconic system lends itself well to deployment on existing equipment, such as TV remote controls, without specialized button layouts or labeling. Second, a user need not become an expert in order to be comfortable entering simple text. Third, it is more easily used when visual feedback is difficult.

1 Introduction

In years past, consumer electornics presented a simple interface to users. A few tens of stations could easily be selected by number, as could the track to play on a CD. Today, increasing choice in programming and increasing functionality in the devices themselves, are producing such a wide range of choices that text input seems to be a necessary addition.

Any application that allows the user to select from hundreds of choices is likely to benefit from the mnemoic value of text. For example, with the explosive growth of cable television channel availability, it is probably easier to "type" ESPN than remember its latest channel assignment.

Similarly, today's DVD players have begun to offer MP3 playback from CDs. Such a CD can easily contain 200 or more tracks. Today's interfaces for selecting from these tracks are notoriously painful. Most offer only the ability to advance or backtrack one song at a time. Some offer the ability to enter tracks by number, but that is almost worthless when playing a 300 song disc. Emerging standards for writable DVDs promise to make this problem much worse, with the possibility of discs containing thousands of tracks. Users would be far better served by the ability to select songs textually by title, artist, or album.

Yet the prospect of adding an entire keyboard to remote controls is of dubious value. The defining characteristic of a good remote control is its ability to be used one handed, without looking at it. Users expect to point the "clicker" at the device and make things happen. Furthermore, such a device would cost more. As computer scientists, we are conditioned to expect that the price of better technology will drop drastically over short periods of time, but the costs of handheld hunks of plastic are not affected by Moore's Law. A more complicated remote control, with 30 extra little moving parts will simply cost more. Yet DVD players are currently available for under $80.

Iconic text entry addresses this problem by allowing text entry using a standard numeric keypad (see Figure 1, available on nearly all remotes today. As such, the "competition", for iconic text entry is other keyboardless input techniques, such as "Multi-tap" used on mobile

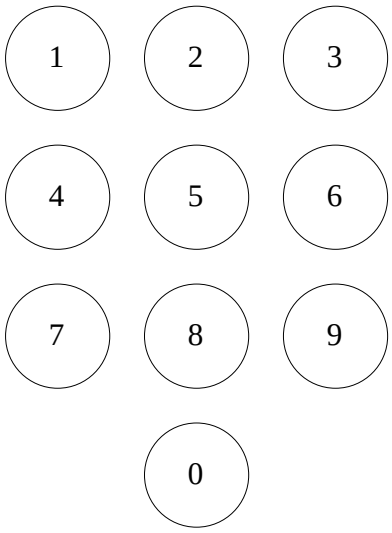

Figure 1: A typical numeric keypad.

phones, and on screen menus of letter choices, such as TiVo's [9], program selection system. Iconic text entry is intended to be simpler to use than multi-tap and less tedious than on screen letter selection. Its simplicity derives from its use of existing notions of letter shape. Further, a user should not need to consult tiny labels on his remote to enter text.

Section 2 discusses the design goals of an iconic text entry system, its intended benefits, and expected limitations.

Section 3 discusses the use of iconic text entry for the simplest of tasks, entering only the letters A-Z, with no regard for case.

Section 4 discusses the use of iconic text entry in more complicated scenarios in which the ability to differentiate case and enter punctuation and symbols is required.

Section 5 compares iconic text entry to other text entry systems. These comparisons show that iconic text entry is only slightly slower than other, less intuitive interfaces.

Section 7 details future work.

2 Design Goals

Iconic text entry was designed to meet the needs of the applications described in the previous section. It should be extremely easy for the owner of a consumer oriented device to learn how to enter text. It should require no new hardware. It should be usable in typical home environments, including dim lighting. In the hands of an experienced user, it should be faster than other systems that meet these goals, even if slower than solutions that sacrifice them. Nonetheless, if should be competitive with specialized solutions.

One button input should be maximized, particularly for common letters. Key sequences, where necessary, should be short. When mistakes are made, it should be easy to continue.

Iconic text entry is inspired greatly by Palm's Graffiti [1]. Studies have indicated that users of grafitti have become adept in a remarkably short time[7]. Five minutes of practice time have produced accuracy ratings over 95%. Those ratings were maintained after a two week lapse in training. Iconic text entry was designed to take advantage of the same tenets that make graffiti os easy to learn. Letters should be entered in a way that seems natural. Entering an "I" should consist of single movement downward. The difference is as small as possible – buttons must be pushed along the way.

3 Basic Text Entry

Iconic text entry is best suited for applications requiring the richness and mnemonic value of text, but not the precise details added by punctuation, case differentiation, and symbols. This section focuses on this simpler problem – the entry of letters only.

3.1 Gestures

The basic idea behind iconic text entry is that each letter should be represented by a set of

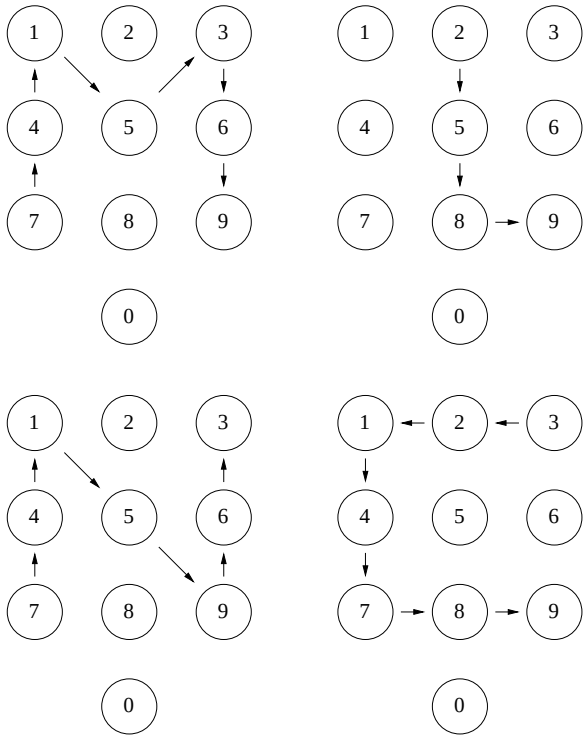

Figure 2: Gesture based iconic representations for M, L, N, and C. A user follows the arrows to enter the intended letter. For example, M would be entered with the sequence: 7415369.

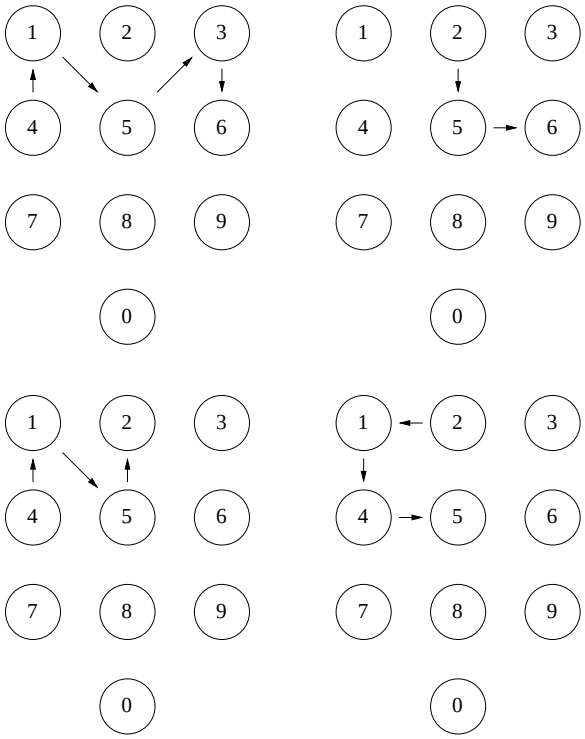

Figure 3: Compact alternative gesture representations for M, L, N, and C. The number of button presses required for each letter has been reduced, and the starting position for the various letters have become more closely clustered.

button presses that somehow evoke the shape of the letter itself. The simplest form of this is the use of "gestures", sequences of button presses that seem to indicate lines or curves. A combination of these gestures produces the letter that the combination resembles. For example, possible gestured representations for the letters M, L, N, and C are demonstrated in Figure 2.

Using gestures presents a number of challenges. First, the gestures in Figure 2 are large – tapping an "N" requires 7 button presses. Second, the number of button presses for various letters is different. This presents a challenge because it may not always be possible to tell when one letter is complete and the next begins.

Addressing these two issues simultaneously is particularly challenging, as the solutions work against one another. In addressing long gesture sequences, smaller sequences will be proposed. Yet smaller sequences have a greater likelihood of matching the beginnings of another letter's sequences. For example, consider Figure 3, in which more compact forms of the previous code sequences are presented.

The gestures for "O" might be the same as the gesture for "C", except slightly longer, to complete the circle. There are two common solutions to this difficulty. Either the user is forced to pause slightly at the completion of each letter, or an "accept" button is used to end an ambiguous letter. Both of these approaches can be expected to slow down and complicate use, so a third approach, inspired by Huffman coding [5] is used. No letter's encoding will be the prefix of any other letter's encoding.

Adopting a prefix-free encoding greatly simplifies interpretation, and, it is hoped, will decrease user frustration associated with timeouts that seemingly randomly change the interpretation of button presses. Using a prefix-free encoding means that a sequence of button presses

Digit	Letter
0	O*
1	I (or l)
2	Z
3	E*
4	A
5	S*
6	G* (or b)
7	T*
8	B*
9	q (or g)

Figure 4: Possible mappings of digits to letters based on the similarity of their shapes. Starred letters indicate actual mappings in the proposed *iconic* system.

will always product the same result, regardless of timing. In this way novices will not be disturbed by changing meanings if they press slowly, while experts will not be annoyed by the need to pause for a long timeout, calibrated for novices, between letters.

Yet, there are complications. It is hard to imagine, using gestures alone, an encoding that would not make either "U" or "C" a prefix of "O". Certainly there are possibilities, such as forcing "O" to be tapped out clockwise instead of the natural counterclockwise. However, iconic text entry stresses a different approach, exploiting the similarity of certain characters to specific digits, outlined in the next section. (Various clockwise "O" encodings are accepted as well, such as 2684 and 12541.)

3.2 Similarities

Instead of using the keypad as a crude drawing area on which to write letters, the direct resemblance between certain letters and digits can be used to assign a letter to a single digit. Taken to an extreme, one can imagine an appropriate letter for nearly any digit, as in Figure 4.

The main advantage of these mappings is obvious. A single button press is certainly going to be faster than a sequence of buttons. This advantage is particularly important for common

Letter	Code	Letter	Code
A	426, 2426	N	4152
B	8, 14758	O	0, 2684
C	2145	P	14725
D	25847	Q	14528
E	3, 147852	R	475
F	14712	S	5, 2154
G	6, 214785	T	7, 12358
H	1425	U	2541, 2563
I	2580	V	153
J	2587	W	14263
K	248	X	1524, 159
L	1478, 2589	Y	158, 1357
M	41536	Z	1245

Figure 5: A complete table showing the button sequences for entering alphabetic characters. Some letters have multiple representations for convenience or to decrease the likelihood of errors, some are shown.

letters such as "E", "S", and "T". There is a more subtle advantage as well. In the previous section it became clear that a number of letters share similar gestures, particularly as smaller representations are used. For example, "U" and "O" might both be represented by the sequence `1452`. By assigning "O" to the digit 0, this problem is eliminated.

At the same time, letter-digit similarity can not be used for all digits. Keeping in mind that no two letters' codes may share a common prefix, it is clear that some digits must be kept free for use at the start of other letter codes. In particular, digits in the upper left corner (124) are particularly valuable, as most letters begin there. 9 is saved for later use as the space character. Its utility for the letter Q is low considering Q's infrequent occurrence.

3.3 Complete Entry Table

Taking into account the use of gestures and letter/digit similarities, Table 5 represents a complete set of code sequences that can be used for iconic text entry. No letter's code sequence is the prefix of any other letter's sequence.

Figure 5 is complete only in the sense that it shows encodings for all letters. It does not show

all encodings for all letters. Many of the more "complicate" letters, such as "K" and "Q" have multiple encodings. Different users will likely have different ways that seem most natural to them for encoding letters. Except where barred by prefix considerations, there is no reason not to accommodate such variation. In particular, extra care was taken to provide an gesture based alternative for every single digit encoding. At times, such as for "E", this produced a fairly impenetrable encoding. Yet some users may find such encodings easier to grasp than the separate "similarity" concept.

4 Advanced Text Entry

The focus of iconic text entry is simplicity. As such, it is best suited for applications that have little need for punctuation and do not distinguish between upper and lower case. However, it is not difficult to imagine small extensions that would allow the user to enter symbols or distinguish case when necessary.

First, the space character is likely to be of use in many applications, regardless of their need for more complicated punctuation. In fact, because space is the most common character in English text, a special concession should be made to ease its entry. It would also be difficult to imagine an iconic representation for a character that has no shape. For ease of entry, space is assigned to the digit 9. There is some mnemonic value to the location of the 9 on the keypad, because it is "out of the way", and therefore might connote a "break" in the normal button presses constituting a word. However, that mnemonic value is probably small. The real value is that space may be entered with a single button press.

Other punctuation, including editing commands like backspace, can be entered using an "escape" system. Nearly all numeric keypads contain two additional buttons, located to either side of the 0. One of these button can be used to indicate that the next gesture is punctuation. A detailed mapping of gestures to punctuation has not been worked out, and it is likely to be less satisfying than the entry of letters because of the tendency of punctuation to so consist of shorter strokes. For example, the differences between a brace, bracket, parenthesis, and a less-than sign are fairly subtle at the resolution of a numeric keypad. Nonetheless, simple punctuation, such as periods, hyphens, and slashes should handled quite easily. Simple editing commnds may actually be more suitable for other available keys on the input device. For example, DVD remotes have arrow keys that would provide an intuitive backspace.

Distinguishing case can be handled in a similar manner. Just as punctuation used one of the extra buttons for an escape, the other button can be used to indicate that the following letter should be capitalized.

5 Evaluation

One way to characterize text entry systems is KSPC [6]. KSPC measures the average number of key presses required to produce a single character. A standard QWERTY keyboard's KSPC is 1. Unsurprisingly, the "two-key" mobile phone text entry system has a KSPC of 2. To enter a character the user must press the button for the desired character and then a second button to select which character from that button is actually desired.

For text input systems that require a variable number of button presses depending on the letter, a language model is required to estimate KSPC for normal text. The language model assigns probablities to each character of the alphabet, based on the frequencies encountered in a large corpus of text. With these probablities, computing KSPC is a simple matter of summing the number of presses required for each character, weighted by the probability of that letter appearing.

Figure 6 compares KSPC for iconic text entry with other inputs systems. Data for LRS,

Scheme	KSPC
LRS	6.4–10.7
LRUDS	3.13
Iconic	2.43
Multi-tap	2.03
Two-key	2
QWERTY	1

Figure 6: KSPC values for various text entry systems. LRS is a selection mehanism using Left, Right, and Select keys. The range shown refers to various details of the implementation (centrally located space character, for example). LRUDS is a two dimensional selection system, adding Up and Down keys. Multi-tap is the mobile phone text input system that requires one tap if the user want the first letter on a key, two for the second, etc.

LRUDS, and Multi-tap were obtained from [6]. A high KSPC might me considered to be highly correlated to user "tedium". By this metric, LRS and LRUDS, two systems that work by allowing the user to navigate a menu of letter choices and then selecting their choice seem significantly more tedious than iconic entry. On the other hand, iconic entry is only about 20% more tedious than other techniques that use the numeric keypad.

LRS and LRUDS, in addition to being tedious, require a high degree of visual feedback to be effective. Iconic text entry hopes to avoid that need. In that light, a more detailed analysis technique using Fitts' Law was pursued to compare input speed using iconic entry to the other input techniques that avoid visual feedback - Multi-tap and Two-key.

Fitts' Law [3] can be used to evaluate the expected text entry speed of an expert using any text entry syste,. The technique used was first described by Silfverberg, Mackenzie, and Korhonen [8]. Fitts' Law is used to predict the time required for a human to move a finger from its current location to a new target as a function of the distance to the target and the target's size.

$$a + b \log(A/W + 1)$$

Scheme	WPM (finger)	WPM (thumb)
Iconic	19.8	17.6
Two-key	25.0	22.2
Multi-tap	27.2	24.5

Figure 7: A comparison of expect input speeds for various text entry schemes. Results are based on the use of Fitts' Law to predict the behaviour of expert users on English text. Words are assumed to consist of five characters.

A expresses the distance to the target, W its size. a and b are empirically determined constants that relate to human physiology. All parameters for this evaluation match those of previous experiments [8] including empirical observations of a and b for index fingers and thumbs.

Fitts' Law predicts the time between any two button presses. To obtain a prediction for an entire character, which may consist of multiple button presses, Fitts' Law is applied repeatedly to obtain the time for each press. These times are then summed.

When applying Fitts' Law in this way, it is necessary to specify a starting point, so that the initial distance to the first key press can be determined. This implies knowing the location of the final key press of the previous character before calculating a time for the current character. Using a table of digram frequencies in English langauge text, it is possible to determine the average time to enter any particular letter by summing over the time taken to enter the letter following any other letter, weighted by the likelihood of the previous letter appearing before the letter in question. Similarly, summing over the average entry time for all letters, weighted by their probability of appearing, provides an average entry time per character.

Using the standard assumption that an average word is 5 characters, a predicted words per minute (WPM) estimate can be derived. Figure 7 shows that iconic text entry is slower than other schemes, but not unreasonable so.

6 Related Work

Iconic text contains many of the same design elements as Palm's Graffiti [1], a pen based input system. Rather than tackle the processor intensive task of full handwriting recognition, Palm PDAs defines a set of gestures that resemble the letters they denote, but are more easily differentiated than free form handwriting. Just as Graffiti occasionally defines simplified forms for some letters (an inverted "V" serves as the letter "A"), iconic text entry uses simplified representations (426 serves as an "A"). In fact, the particular simplified shapes of some letters (A, F) are directly inspired by the simplified shapes of Grafitti. Graffiti's acceptance in a widely used consumer device leads one to believe that everyday users will adopt simple conventions for text input as long as they have some mnemonic value.

There is a vast body of work on alternative text entry systems for devies with a limited palette of buttons. The most closely related are those systems that are capable of deployment without the introduction of new hardware, such as specialized keyboards or touch screens. These systems fall broadly into two classes - *pickers* and *coders*.

Pickers are those interfaces that provide visual feedback allowing the user to select from a list of letters by navigating the list. Tivo [9] is rapidly becoming the best known example of this interface, though arcade video games have used this technique for many years to allow the entry of "high score" initials. The advantage of a picker is incredible simplicity. The disadvantages include the necessity of a large visual feedback device, and the tedious entry process. Interestingly, a Tivo remote has a numeric keypad in addition to the navigation keys used to pick letters. A software upgrade could allow Tivo to accept iconic text entry while maintaining its existing interface.

Coders are interfaces that use a sequence of buttons to signal letters. The use of mobile phones for text entry is the most widely deployed example of this technique. The advantage of a coder is its increase in speed over a picker interface, and a reduced need for visual feedback. However, a picker does require button labeling. Some would argue that expert users would eventually avoid the need to consult labels, but the vast number of computer users who never become touch typists seems to argue against this assumption. In that light, a labeling requirement is a disadvantage for two reasons. First, referring to the labels on buttons can be difficult for any number of reasons – a dark environment, a visual impairment, or simply large fingers obstructing the view. Second, many devices, such as TV remote controls, lack those labels. Iconic text entry allows increased functionality on existing hardware.

T9 [4], a text input system for mobile phones uses a dictionary of known words to differentiate the intended meaning of a sequence of button presses. Using T9, a user need not press a button repeatedly to select the intended letter. Instead, when the user concludes a word (by hitting space), the system selects the most common word that can be spelled using the letters available from each key in the sequence. In case of multiple possibilities, the user can hit a "next" key to request the next word choice. T9's largest drawback in the contexts for which iconic entry is intended is the requirement that a feedback mechanism be constantly monitored to notice mistaken words.

Word prediction techniques, such as WordComplete[2] by CIC, offer the prospect of even greater gains over traditional techniques than T9's disambiguation [6], achieving KSPC scores less than one. These techniques are completely compatible with iconic text entry, as they are completely agnostic with respect to how individual letters are entered. They simply supply possible word completions as text is entered. Naturally, these techniques require a useful visual feedabck device, so they may not be useful in the applications for which iconic text entry is best suited. However, if feedback is available, word prediction can be

used as easily with iconic input as with any other.

7 Future Work

The current implementation of iconic text entry is extremely simple. As buttons are pressed, the software follows a simple state machine. When the state machine enters an accept state, a letter is produced and the state machine resets. Similarly, if a sequence of buttons does not correspond to any possible encoding, the state machine is reset and an error can be reported to notify the user.

While simple to conceptualize, this simple system has some drawbacks that further research might address. First, it may be more intuitive to allow encodings for letters that are prefixes of other letters. This would allow, for example, an alternative encoding of "O" that resembles the encoding for "C", but completes the circle. In general, the larger the number of alternative encodings, the more likely it is that users could begin entering text without consulting instructions.

A future goal is to build recognition software that is powerful enough to determine the intended boundaries between characters without requiring prefix-free encodings. That software might take advantage of natural pauses between letters, or by pruning "impossible" interpretations from the various possibilities derivable from ambiguous input sequences.

For example, the input sequence 14523 can be interpretted as [C error] or [O E]. In such cases, [O E] is the preferrable interpretation. Carrying the idea one step further, in many of iconic text entry's envisioned applications, the user is selecting from a large dictionary of possible choices. In those case, it would be possible to prune otherwise valid input because it does not correspond to a possible choice.

8 Acknowledgments

The combination of two technologies provided the inspiration for this work. First, Jamie Zawinsky's web-based MP3 player, *gronk*, provided a excellent, easily modifiable interface to a large library of cataloged MP3 music files. Second, Evation.com's's *IRman*, provided a mechanism to accept remote control codes from existing infrared remote controls. The desire to control a fundamentally textual application using existing remote controls led to the desire for a simple, easy to remember method of entering short strings of text using a numeric keypad.

Scott Mackenzie was very helpful in providing the details of his experiments, allowing a fair comparison of iconic text entry to multipress and two-key entry.

9 Conclusions

In some applications, the simplicity of an iconic text entry system is more important that the potential for greater expert performance. In home electronics, users expect a simple remote control. They should be able to enter text without consulting labels, and without learning new concepts. Iconic text entry allows new users to begin working right away. Its peak performance is not as high as techniques designed to optimize entry speeds, but it is adequate for simple purposes which stress the mnemonic value of text, not the need to compose detailed prose.

References

[1] C. H. Blickenstorfer. Graffiti: Wow! pages 30—31, January 1995.

[2] Communication Intelligence Corporation. *WordComplete for Palm OS*, March 2002. `http://www.cic.com/support/Faq/Pos/WordUsersGuide.html`.

[3] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movements. *Journal of Experimental Psycholgy*, 47:381—391, 1954.

[4] D. L. Grover, M. T. King, and C. A. Kuschler. Patent No. US5818437, reduced keyboard disambiguating computer, 1998.

[5] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proc. of the IEEE*, 40(9):1098—1101, September 1952.

[6] I. Scott Mackenzie. KSPC (Keystrokes Per Character) as a characteristic of text entry techniques. Submitted for publication.

[7] I. Scott MacKenzie and Shawn Zhang. The immediate usability of Graffiti. In *Graphics Interface*, pages 129–137, May 1997.

[8] Miika Silfverberg, I. Scott Mackenzie, and Panu Korhonen. Predicting text entry speed on mobile phones. In *Proc. SIGCHI: ACM Conference on Human Factors in Computing Systems*, pages 9—16, April 2000.

[9] TiVo. TiVo digital video recorder. `http://www.tivo.com/`.