ABSTRACT

When using your smartphone (or similar handheld electronic device) with one hand, you have to type with the thumb, which isn't the finger you would normally use. The more agile fingers are tied up with holding the device. This is due to the interface concept, not to any functional deficit of the human hand. You could operate these devices with the use of all fingers, and not just for typing, but that calls for a different interface paradigm, one that mirrors the ergonomic capabilities of the human hand. You engage this kind of interface where your fingers spontaneously come into contact with the device when you grasp it, not at predetermined locations that are not necessarily easy to reach. This interface dynamically maps the controls of the device to the contact patches of your fingers and generates a visual, audible or haptic tag to tell you what function a finger controls. To activate a control you engage it with the corresponding finger.

KEYWORDS

Topical† tactile interfaces, atopical* tactile interfaces, operative interaction with back and sides of device, accessibility issues with virtual controls, atypical handheld devices (disks, cylinders, torons etc.)

†characterised by operative interactions confined to defined locations
*characterised by operative interactions *not* confined to defined locations

1 Background

Many applications for handheld electronic devices depend on the user being able to enter text. To this end, the tactile user interfaces of these devices are configured with a miniature keyboard. The idea behind this is presumably to replicate the functionality of a standard keyboard. This may be aiming a little too high, though. Scaling a conventional keyboard to where its rows and columns fit a handheld device gives up much of its usability. There is no way to use the scaled down keyboard the same way as the original. In particular, you can't type with all fingers, not even all fingers of one hand. And, as a consequence of the restrictive conditions of access conventional interfaces afford, typing speed and accuracy are below the level possible on a standard keyboard. This state of affairs has produced a profusion of remedial implements and workarounds designed to overcome the shortcomings of the downsized keyboard. They range from gadgets like the finger stylus (a kind of thimble with a pointy tip) to sophisticated swipe based input apps. In parallel there is a trend towards sidestepping the tactile interface with the use of speech recognition technology for text entry. All of this indicates that the miniature keyboard is not satisfactory. Some of these remedial efforts, like keyboard swiping, amount to entirely new ways of typing that the user has to learn from scratch without benefitting from any experience gained with a standard keyboard. Most are specific to miniaturised keyboards and of no relevance or advantage for a standard keyboard. None of them is widely accepted. Even though the unsatisfactory nature of the available solution is palpable from the attempts at making it better, there has been no serious examination of where the problem these contrivances address comes from. In fact, there is an implicit denial of the existence of a problem. In spite of the many drawbacks of the miniature keyboard, the validity of the approach is taken for granted and the assumptions underlying it are rarely, if ever, questioned. This paper examines the root causes of the problem that emulating the standard keyboard on handheld electronic devices poses and concludes with a proposal for a new approach to user interaction with handheld devices, not just for text entry.

The miniature keyboards of handheld devices only emulate the non-functional features of the standard keyboard, in particular its configuration, its orientation and its layout. The standard keyboard has four rows of 10 or more keys and permits the concurrent use of both hands side by side. The rows are staggered, a vestige of the mechanical typewriter; consequently, the keys are arrayed in slanted columns. Three rows are devoted to the letters. Each finger is dedicated to a column of three keys except the index fingers which take in two columns. This is to make up for the thumbs, which on account of their limited dexterity are relegated to the spacebar, a particularly large key located below the rows. No finger controls more than six keys; most control three and the two thumbs together only one. On a conventional handheld device, the thumb controls all 26 plus keys by itself, notwithstanding its problematic dexterity. This turns the normal way of using a keyboard on its head and undercuts the case for taking it as a model for text entry.

When using a handheld device, the user's grip on it pins the base of the thumb to the right or left of the device (depending on the hand). To reach a key at the opposite side, the thumb has to stretch across the width of the keyboard. The further it has to stretch, the shallower its angle of attack, the lower its accuracy. To reach a key on the near side, the thumb has to double back on itself, which is not easy with only two phalanges. The shorter the reach, the more difficult and strained it gets. On a standard keyboard the thumbs remain poised at the spacebar.

Instead of replicating the widthwise orientation of the standard keyboard it would make more sense to present the keyboard vertically along the edge next to the thumb and to centre it on the pivot point of the thumb. This would bring the rows within a comfortable range of the thumb and would shorten the distance it has to travel to get to the keys at the edges of the keyboard. Replicating the orientation of the standard keyboard is clearly not advantageous. Inexplicably, the elegant solution of the Microsoft Word Flow Keyboard has not taken hold and has been discontinued. It displayed the rows of the keyboard in three concentric arcs centred on the pivot point of the thumb, putting the keys of a row at a constant distance from the centre of the arc.

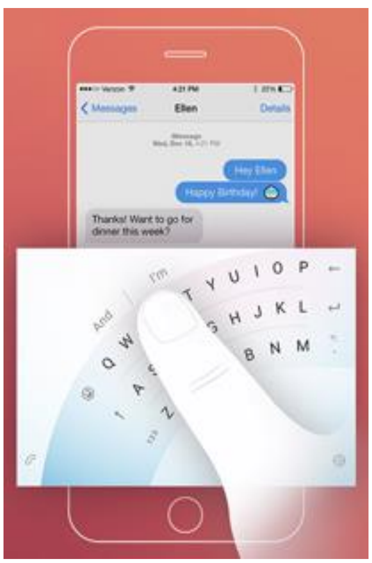

Figure 1: Microsoft Word Flow Keyboard

The layout of the keyboard is another vestige of the mechanical typewriter with no operational advantage. The letters are allocated to the keys in an arrangement designed to avoid type-head clashes. For the English language this yielded the idiosyncratic QWERTY layout. Even though type-head clashes are no longer a concern, handheld devices faithfully replicate it. There is nothing to be gained from this. A typist who has mastered touch-typing on a QWERTY keyboard can not apply this skill, because the miniature keyboard only affords access with one or at most two fingers concurrently. This leaves the user no other choice but the one or two fingered hunt and peck technique, for which a keyboard laid out in alphabetic order, which is at least as familiar as the QWERTY layout, would work just as well.

Arranged in the form of a matrix filling the display like the icons of the typical home screen, the keys could be larger and consequently easier to hit. This could be a step in the right direction, but the user would still have no option but to hunt and peck at them. The efforts, alluded to above, at shoring up the usability of the miniature keyboard on handheld devices fare no better. None of them has made it more workable as an interface for text entry or has otherwise yielded any significant improvement. All are within the scope of the conventional tactile interface for handheld devices. They merely perpetuate an interface paradigm that is not appropriate for this category of device.

Conventional tactile interfaces confine operational interactions to discrete predetermined locations on the interface surface (like the keys of a keyboard). They are topical in that they are only responsive to contacts with specific, predefined locations. Each of these locations is configured with a control, which can be physical or virtual, governing a particular function of the device. To trigger a function the user must seek out the spot the control governing it occupies and manipulate it with a finger. Aside from a stray contact with the wrong location, a contact outside the predetermined locations has no effect. Under the conditions of access handheld devices typically afford, engaging the targeted control may call for a fair amount of dexterity often with the least dextrous finger.

2 Specific Issues

Conceptually, the conventional user interfaces of handheld devices are control panels that have been scaled down to fit handheld devices. In the process much has been lost. Control panels bring the controls and gauges required to operate an apparatus together in one convenient place. When the apparatus is large or dispersed, this makes perfect sense. As a rule, their layout reflects the logical organisation of the apparatus, not its physical configuration. Control panels of ordinary size afford the user ample space and unrestricted access to their controls, if required with both hands. When the entire apparatus including its control panel is small enough to hold in one hand, the utility of the control panel concept reaches its limits. The circumstances and conditions of access for a control panel to be useful are no longer there. Under these conditions the specificities of the device and the ergonomic capabilities of the human hand play a predominant role in determining how its interface needs to be configured to facilitate the interaction between the user's hand and the interface. Indeed, they have a decisive effect on the user's ability to operate this category of device in a natural manner. The implications of this go beyond text entry - though it provides a particularly instructive example - to affect user interaction with this category of device in general.

First among these specificities is that these devices are intended to be operable in-hand, that is while holding and operating them with the same hand. A full-size control panel, on the other hand, a standard keyboard in particular, stands by itself, giving both hands full and unfettered access to its controls (keys). Holding a device has a cost in terms of the fingers it ties up. Relegating the most dextrous fingers of the hand to holding the device together with its control panel/interface, pre-empts using them to engage the interface, in particular for typing. Besides, holding the device anchors the hand to it, handicapping the movement of the fingers that remain available to interact with it, usually only the thumb.

The natural way to hold a handheld device when operating it in-hand is by its sides in a handshake-like grip, as shown in Fig. 2. The index, middle, ring and little fingers hold the device against the base of the thumb

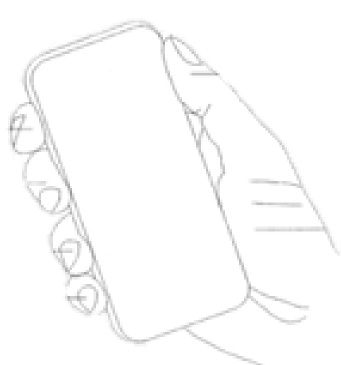

Figure 2: Natural way to hold a device

on the opposite side, leaving the thumb as the only finger with leeway to move. This grip is quite similar to the mechanic's grip, the way you hold a deck of cards when dealing, except that the index finger is usually along the side as shown in Fig. 2 (and not at the top of the deck). This grip is perfectly appropriate for dealing cards, when all the thumb has to do is shove a card off the top of the deck. Manipulating the interface of a handheld electronic device is more demanding, particularly when it is configured as a keyboard. It calls for a degree of precision and control that is difficult to achieve with the thumb. Any of the other fingers would be better at this. Allocating the index, middle, ring and little fingers to holding the device and the thumb to engaging its interface is at odds with the relative dexterity of the individual fingers of the human hand. But with the interface on the user facing side of the device, there isn't much scope for holding it any other way, at least not when using the device with one hand. In view of this, users resort to a variety of stratagems to compensate for the drawbacks of this setup.

For instance, to extend the reach of the thumb users sometimes sacrifice a firm grip on the device, supporting it loosely from underneath with the other fingers, relying on the weight of the device to keep it in place.

Or, to avoid having to use the thumb, users have recourse to the other hand as well. A user may hold the device in one hand or hold it down on a support surface, while operating it with the other. In principle this affords the user the choice of any fingers to access to the interface, because the operating hand is not anchored to the device by dint of holding it. The small size and crowded layout of the keys, however, limits the number of fingers the user can bring to bear concurrently - generally to one, under these conditions usually the index finger. Still, it improves the angle of attack and consequently the user's accuracy. It is worth noting that given the freedom of choice, users chose the index finger and not the thumb, which they only use when there is no alternative[1].

For typing some users resort to an unusual workaround that paradoxically embraces the use of the thumb. Here the user cradles the device in both hands while typing with the two thumbs. In essence this method comes down to a peculiar form of the hunt and peck technique. Many users are quite good at it, a remarkable feat, as is often noted. On account of their limited dexterity and stubby tip, the thumbs are not particularly well-suited to resolve small targets in a crowded layout. However, approaching the keyboard from above improves their reach as well as heir angle of attack and only having to cover half a keyboard with each thumb may also be advantageous. This method too relegates the more dextrous fingers of the hand, which do the lion's share of typing on the standard keyboard - covering all letter keys but not the spacebar - to holding the device, while it relies on the thumbs, which the standard keyboard relegates to the spacebar, to do the hunting and pecking. This doesn't make ergonomic sense. Under normal circumstances no one would use the thumb even to press a doorbell or an elevator button, although their size and the proximity of other buttons is not an issue in these instances.

3 Design Choices

Beyond the problems inherent in any interface that confines operational interactions to predetermined locations, certain design choices also have an unfavourable effect on the usability of the conventional interface. First among them the placement of the user interface, the keyboard in particular, on the user facing side of the device. At first blush this would appear to be the obvious and only place for it. On closer examination, though, it becomes apparent that in view of the limited space available on handheld devices, concentrating all user interactions on the front is neither advantageous nor necessary. As a first consequence of this location, when engaging the interface, the fingers clutter the user's view of the display, sometimes masking important landmarks. This is particularly prejudicial to entering text because the degree of precision

1 Surprisingly Hoober [1] reports that under these conditions 72% of users continue to use the thumb to engage the interface.

a miniature keyboard calls for is not easy to attain, especially with the thumb. Furthermore, when operating the device in-hand, the front is not easy to reach, because the hand comes at it from the sides and back. This tends to suggest extending the interface to these areas to take advantage of this, a possibility that, with few exceptions, has not received much consideration[2].

It is, of course, self-evident that visual output has to be displayed on the user facing side of a device for the user to be able to see it. There is no reason, though, why the interface can not make use of the unseen surfaces of the device for input. When the user is holding the device in a natural manner, the fingers spontaneously come into contact with its sides and back, making these surfaces easier to engage than the front. In turn, this makes them particularly suited for the tactile interface. As long as the user can track the effect of an interaction between a finger and these surfaces, there is no need to have the place where the finger engages them in sight. Touch-typing illustrates this, although the miniature keyboards of handheld devices do not support it[3]. The use of a touchpad to position a cursor is another example. And it is easy to see how this might apply to the surfaces of a handheld device the user does not have in sight.

Consider a generic handheld device, in the shape of a rectangular tablet, that the user holds by its edges. (See Fig. 3.) Aside from a display on its front, it has a rear surface configured as a touchpad, which the user

Figure 3: Front and back view of a generic handheld device in the hand of a user

engages with the index finger. (See Fig. 3 left.) This lets the user place a cursor anywhere in the display without the finger getting in its way. In this way, the user could select one of the icons of a typical home screen and launch it from the back of the device. (See Fig. 3 right.)

When holding a device by its edges, the thumb naturally lines up with the edge it touches in position to make up and down movements along the edge. By the same token, the tip of the index finger naturally touches the opposite edge more or less at right angles, in position to make to and fro movements. If the edges of this device are part of its interface, it could be configured so that the user could, by means of these gestures,

2 But see [2], [3] and also [8].

3 Touch-typing is a key advantage of the full-sized keyboard that does not scale. This is fairly obvious but rarely noted (but see [4]).

perform various functions, e.g. operate a date picker or a volume control or zoom in and out of the display or scroll through a document[4] without the fingers getting in the way of the display[5].

Another consequential design choice is the practice of treating holding a device as an overhead function that can not be combined with anything else, in particular with operating the device. Conventional interfaces are not designed to use a finger holding the device to perform operative functions and, vice versa, a finger employed to operate the device to hold it. The two functions are kept separate. Conventional user interfaces follow this practice unquestioningly, even though it underutilises the human hand and restricts the number of fingers available to engage the interface. This is again a matter of choice, not a necessity. There is nothing in the nature of handheld devices or of the human hand that prevents a finger occupied with holding a device from concurrently engaging its interface. Think of the way one plays a recorder. Only the right thumb does nothing but hold the device, while the little finger of the left hand does nothing at all. All other fingers concurrently serve to cover the holes (making the notes) and to hold the instrument. (See Figure 4.)

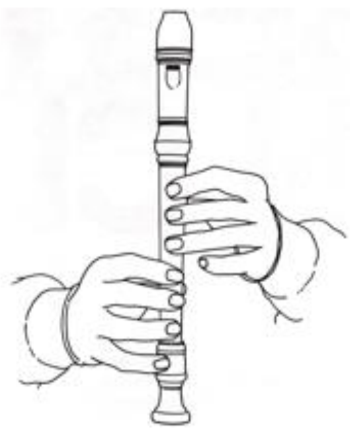

Figure 4: Fingers holding and operating instrument at the same time

Or witness the examples described above. Engaging a touchpad at the back or making back and forth movements at the edges does not jeopardise the user’s grip on the device even when operating it with one hand.

A priori, having the use of additional fingers to engage the interface would be an advantage, particularly for text entry. However, the comparatively small size of the tactile user interface of conventional handheld devices (even of the larger ones) combined with the fact that the hand is partially tied up with holding the device limits the number of fingers the user can employ concurrently. This precludes the use of more than one or two fingers at a time. Enabling the use of all five fingers to operate a handheld device in-hand calls for an altogether different approach.

The choice of the keyboard as the standard interface for text entry is also highly consequential. It is a real choice, not a given; there are alternatives to it, such as the ITU E.161 keypad or chorded keyboards. While

4 Early model Blackberrys had a physical scroll wheel to the right of the display within easy reach of the thumb. While a physical control is tied to its location, the edge at which a user makes a gesture is immaterial.

5 Incidentally, when using a device with one hand, users can not make the familiar pinch and un-pinch gesture for zooming nor any other two finger gesture.

these have well known drawbacks[6], so does the miniaturised standard keyboard, though its drawbacks are rarely acknowledged. It is therefore important to examine them to see whether they justify this choice. This is a central issue of this paper: Is the standard keyboard in miniature a good vehicle for text entry on handheld devices or is there something better?

To fit a standard keyboard to a handheld device without altering its configuration, its keys have to be scaled down to a fraction of their original size and they have to be tightly packed into the limited space available. This is a necessity, not an advantage. The fact that it results in target areas that do not satisfy the best practice guidelines for the size of controls, has not escaped notice (See [e]). It has not led to any questioning of the rationale for this approach, though. The full-size standard keyboard is dimensioned for the average human hand. The keys of the miniature keyboards of handheld devices, on the other hand, are dimensioned for the keyboard to fit the interface, not to accommodate the fingers of the human hand. Here ergonomic considerations have been sacrificed in favour of an a priori conception of what the text entry interface should look like.

When a key is smaller than the pad of the finger the user touches it with, the area in contact between the finger and the interface spills over, particularly in the case of keys without a physical profile to set them off from their surroundings. This runs the risk of inadvertently coming into contact with adjacent keys, which is what makes hitting the keys of miniature keyboards without incurring stray contacts with neighbouring keys a challenge. Furthermore, the closer a finger comes to a key, the more it eclipses the key as well as keys nearby. This means that the closer a finger comes to making contact with its target the more the user loses sight of the target area. If the trajectory of the finger is good, this may be more of a psychological issue, but is not a recommended practice for interface design. The controls of handheld devices can be dimensioned to meet minimum standards, even for text entry, but not when the interface is structured as a keyboard.

Another factor making accurate hits on small targets difficult is the discrepancy between the perceived tip of a finger and its effective point of contact, commonly referred to as the fat finger problem. The perceived point of contact of a finger is at the tip, in line with the visual extension of the finger, whereas the effective point of contact is the centre of the finger pad, that is below and behind the tip. While the user intuitively aims the tip at the key, it is the centre of the finger pad that triggers it. With small targets, like the keys of the miniaturised keyboard of handheld devices, this can make the difference between being on target or not.

The issues highlighted above are consequences of the topical interface concept. They come from the fact that the interface requires the user to bring a finger into contact with a predetermined location at large (under the generally unfavourable conditions handheld devices afford). Opting for the topical interface paradigm is again a matter of choice. There are forms of tactile input that do not involve predetermined targets. Gestures, for instance. Tactile interfaces can recognise gestures no matter where the user makes them. While this makes the point that there are alternatives to the topical user interface, it is unlikely to point towards a practical solution. But there is one.

[6] The number of keystrokes per character the ITU E.161 keypad requires varies (between one and four). Furthermore, the flow of keystrokes is syncopated: When typing the same character twice in a row, the user has to interrupt the already irregular rhythm but only for specific character sequences. For two spaces in a row, to give an example, the user just hits the space key twice without pausing. Not so for two Ts. Typing two Ts in a row without pausing produces a U. This is because U is on the same key as T. U requires two keystrokes and T one. Pressing the key twice without pausing (for the T to register) produces a U. These heteroclitic conditions are an impediment to fluid typing and add complexity to an already complicated system.

As to chorded keyboards, the multiplicity of arbitrary finger combinations they require has made them unappealing from the outset and they have not withstood the test time.

4 Atopical Interfaces

An interface that enables a finger to trigger a control without moving from where it is is quite a different thing. Such an interface may be said to be atopical, in the sense that operative contacts can occur anywhere a finger touches the device, not only at specific predetermined locations. The location of the operative points of contact of the fingers is determined by the hand gripping the device. There is no path a finger has to negotiate to get to its target. It is not possible for a finger to encroach on the contact patch of another, hence there are no issues with stray contacts. The operative points of contact can not be closer to each other than the fingers can. It is impossible for a finger to accidentally come into contact with a spot occupied by another finger, because the spot is occupied. The operative contacts are the size, shape and location of the area in contact; there is no possibility of an off-target hit. There can be no discrepancy between the size of a fingertip and that of its target. This approach eliminates the uncertainties of engaging controls of fixed dimensions at predetermined locations. It does not require the fingers to reach for controls, much less for controls at large that are hard to reach. To make this possible an atopical interface is configured to dynamically map controls to the contact patches of the fingers holding the device. The user controls a function as and where the corresponding finger spontaneously touches the interface. In many cases the controls will be in the form of virtual buttons, which the user engages by pressing, without changing the position of the finger. The controls can also emulate other physical controls, for example sliders or scroll wheels. Of necessity, all controls are at the user's fingertips, as and where they naturally come into contact with the device. While a finger can erroneously trigger the control mapped to it, it can not trigger a control mapped to another finger.

To get a feel for how this works, consider a simple example, a music player in the form of a rectangular tablet, which the user holds in a handshake-like grip. (See figure 5.) Its tactile user interface includes the

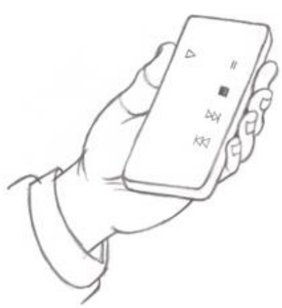

Figure 5: Music player with virtual buttons on the edges

sides of the device. It is configured to sense the contact patches the user's fingers make while holding the device and to map its controls to them. In this case these consist of five virtual buttons: the play, pause, stop, fast forward and rewind buttons. The interface maps each of them to one of the fingers.
To let the user know what function a finger controls, the display shows the conventional symbol for it next to where the finger touches the edge of the device. The user can select any one of them at the press of a finger. For instance, to start playing music the user presses with the thumb and with the index finger to pause it.

The ability to engage tactile user interfaces where the fingers spontaneously come into contact with them, represents an opportunity for making them readily accessible for visually impaired users (indeed any user not able to see the interface, e.g. when touch-typing). Bringing a finger into contact with a predetermined location demarcated only visually on a homogeneous surface without physical profile is problematic for users with a visual impairment. On the other hand, triggering a function mapped to the actual contact patch of a finger is not a problem, provided the user knows what it is. The need to know what function a finger controls is not specific to visually impaired users. That is why atopical interfaces identify the function a contact patch controls by means of a tag, which can be visual, audible or haptic. In the example above the tags were visual. For visually impaired users they have to be audible or haptic. In accordance with established practice, an audible tag is configured so that, pressing a control preselects the function it commands and lets the user hear what it is. If it is the desired function, the user double taps to confirm the selection. If not, the user proceeds to another finger to determine what function it controls. Other than that, visually impaired users engage an atopical interface in the same way as sighted users.

The interface maps the control for a specific function to the contact patch of a particular finger regardless of where the finger touches the interface. Say the interface maps a function X to the index finger. It doesn't matter whether the user's hand grips the device a little higher or lower or whether it is big or small, or the finger is long or short. In particular, it does not matter whether the contact patch of the index finger is on the left side of the device or the right. Atopical interfaces do not distinguish between a left and a right hand. They identify a finger on the basis of the position of its contact patch relative to the position of those of the other fingers and a left index finger is in exactly the same relative position with respect to the other fingers of the left hand as a right hand index finger is with respect to the other fingers of the right hand[7]. The interface identifies a finger based on the context of the other fingers as they grasp the device from opposite sides. It can not identify a finger on the basis of an isolated contact with the interface. For an atopical interface to function it must be able to detect the contact patches of all fingers as the user holds a device in a handshake like grip[8].

The number of controls an atopical interface can make accessible concurrently is limited to five. In view of this it will normally map a control for a different function to each of the fingers. Most often they will be in the form of virtual buttons, which the user activates by pressing the finger (or not). (With an interface capable of distinguishing different levels of pressure, the interaction can be scalar rather than binary.) The interface may also recognise gestures certain fingers perform, in particular those illustrated above.

In view of the paucity of concurrent controls possible, the control allocated to a finger changes from time to time in response to changing requirements; the allocation of controls is context sensitive not static. It can change under program control or in response to a user choice. It changes every time a user takes the device in hand[9]. It changes when a user moves a finger or grasps the device another way. It changes when a user switches to another application. And, every time the mapping changes, the interface refreshes the labelling to show the current attribution of functions to the fingers so the user can always tell what function a finger controls.

For many applications five controls at a time is sufficient, but some call for more. Text entry is one. Standard practice is to present the 26 letters of the alphabet along with additional characters together in the form of a keyboard, i.e. 26 or more keys (buttons) in a conventional arrangement. This is not possible with five virtual buttons. To accommodate more than five choices it is necessary to structure them. For instance, by structuring the choices in the form of a 6 by 6 matrix it is possible to select any one of 36 choices in two steps. In the first step the user choses the row containing the desired item and in the second the item itself. For text entry,

[7] Helke [6] describes the mechanism to identify the fingers in detail.

[8] Based on this, it is possible to configure the device to unlock when it detects a handshake-like grip and, conversely, to lock when it has not detected one for an appropriate interval. All other contacts are without effect eliminating the possibility of triggering the device unintentionally, e.g. pocket dialling, while carrying or otherwise handling it.

[9] Insofar as atopical interfaces are concerned, which way up the user holds a device is immaterial. A suitably configured device can show the contents of the display in a vertical orientation corresponding to the way the user is holding it irrespective of its physical orientation.

take a 6 by 6 matrix populated with the letters of the alphabet, the space character, the principal punctuation marks and a few special characters (see Fig. 6 left). In the first step the user choses the row with the desired letter, by pressing the finger to which it is mapped[10] or, for the sixth row, all fingers together, because each of the fingers individually already has a function mapped to it[11]. When the user has chosen a row, the interface transposes it, allocating its six elements to a new 6 by 1 matrix, one element to a row, from which the user selects the row with the desired character (as its only element), in the same way as before. This adds the character to the input string and starts another iteration.

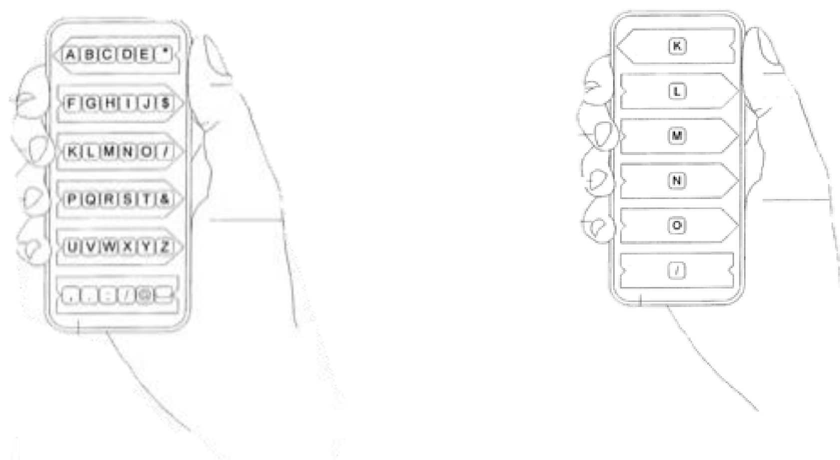

Figure 6: Text entry using atopical interface

For a specific example, refer again to the 6 by 6 matrix in Fig. 6 left. To type the letter N, the user first presses the middle finger to select the third row with six characters including the letter N. This transposes the row into a column, one character to a row, and refreshes the display to show Fig. 6 right. Now the user presses the ring finger to select the fourth row consisting of the letter N by itself. This adds it to the input string[12] and reverts to the first screen to start another iteration.
The letters of the matrix used in the illustration in Fig. 6 are laid out in alphabetic order, to make them easy to find. (Furthermore, the special characters - and the letter Z - are relegated to positions triggered with all fingers pressed together, to minimise the need to use functions triggered with multiple fingers.) Other considerations may lead to other arrangements, not excluding one based on the QWERTY keyboard, if this were deemed to be useful. The right choice is an empirical question, that can not be decided in the abstract.

No matter how the letters are arranged, the text entry matrix comprises 36 elements indexed by row and column. This is on a par with the standard QWERTY keyboard, which, for the letters and special characters, comprises some 26 plus entries arrayed in a matrix with 10 or more columns and three rows (leaving the spacebar aside). The mental effort to memorise the two matrices is equivalent. It is also known to be possible.

[10] Indicated by the directions of the ‘road sign’ surrounding the row.

[11] Mapping a function to a single finger is more intuitive and therefore preferable. It is best to minimise the use of arbitrary combinations of fingers, which is reminiscent of chorded keyboards, because of the complexity it entails. This is almost certainly the reason chorded keyboards have not caught on and there is no cogent reason to resuscitate them.

[12] For visually impaired users a screen reader, an accessibility tool all major operating systems include (e.g. Talkback or VoiceOver), can be set to read the last character added or the input line.

Accordingly, it is fair to expect that, with practice[13], the text entry method described here will permit users to type without the aid of prompts, the same as on a standard keyboard. This constitutes a key advantage of this method over the miniature keyboards, which do not support touch-typing.

5 Discussion

Even though the conventional user interface is fraught with significant usability issues, in the absence of an available alternative, users have come to accept it as the only way to gain mobile access to the many desirable functionalities handheld devices offer. But in fact, there is no need to trade off usability for this. Atopical user interfaces allow users to access these functionalities without the issues besetting conventional topical interfaces. (A table summarising these issues and showing how atopical interfaces avoid them is included as an appendix.) All of the issues are inherent in the topical interface concept, which requires the user to bring a finger into contact with a control in a predetermined position in order to engage it. Atopical user interfaces keep clear of this by actively putting the controls where the fingers are already in contact with the device. This relieves the users of the onus of actively achieving an effective contact with the controls. All the user has to do is engage the control or not.

Technically the difference between topical and atopical interfaces is not all that great. While topical interfaces map functions to predetermined locations in the interface layout in advance of any contact with the user's fingers, atopical interfaces map them to the contact patches the fingers make when gripping the device. Aside from that, the user engages the interface in exactly the same way, by activating its controls with a finger. The difference lies in the fact that atopical user interfaces make use of all five fingers and the fingers do not have to negotiate any distance to make contact with the control. There are no new skills for the user to acquire. The novelty as far as operating the device is concerned is limited and facilitates the user's task. By eliminating the need to reach for a control, it makes engaging the interface simpler. To operate a control all the user has to do is engage it with the corresponding finger at the location where it is as a consequence of holding the device. In most instances this means nothing more than pressing down on the spot, as if pushing a button, or moving the finger in the manner of operating a slider or a scroll wheel[14]. The finger movements this requires are familiar.

The only noticeable difference is that for text entry the atopical interface takes two keystrokes per character[15], not one like the standard keyboard as well as the miniature version on handheld devices. It is important to see this issue for what it is. Text entry is not the primary purpose of handheld electronic devices. Their comparative advantage comes from rendering functions like making phone calls, sending messages, browsing the web, taking pictures, playing music, determining one's position and the like available on a small portable

13 Committing the text entry matrix to memory may be facilitated by a judicious arrangement of its entries as opposed an arbitrary arrangement like the QWERTY keyboard (which is learnable all the same). One possibility would be to reserve the entries on the diagonal for the vowels and the space character. For these the user presses the same finger twice, which makes them stand out. The following matrix shows a way to do this while keeping the letters in alphabetic order.

A	B	C	-	-	-
D	**E**	F	-	-	-
G	H	**I**	J	K	-
L	M	N	**O**	P	-
Q	R	S	T	**U**	-
V	W	X	Y	Z	**_**

14 These controls are virtual, of course. Users do not receive physical feedback from engaging them, because the surfaces of handheld electronic devices typically consist of a rigid material such as plastic, metal or glass. There are methods to reproduce the sensation of engaging a physical control, though. For instance, the Sony TouchEngine, which is a haptic actuator that can vibrate a touch panel with piezoelectric elements. Likewise, Tactus Technology has developed a touchscreen that can dynamically raise physical but nonfunctional keys when the display shows a keyboard. There do not appear to be any current products featuring these technologies, though. An interface with a compressible surface would by itself afford users a degree of tactile feedback.

15 The interface could also be configured to respond to the press of a button by scanning the entries of the row selected until the user releases the pressure, thereby selecting the entry the cursor the cursor is currently on. This may may look like a single keystroke, but the logic of the procedure is based on making two consecutive choices.

device that the user can operate in-hand. Text entry is a service that many applications running on handheld devices make use of, web browsers, for example, or messaging services and social networking. For serious text entry a large display, a full-size keyboard and a computer running a word processor are a better choice. A proper comparison between the two different interface concepts can not be reduced the number of keystrokes per character they require. There is much more at issue. Currently available conventional tactile user interfaces are all somewhat clunky, particularly when it comes to text entry. Atopical user interfaces have many advantages over topical interfaces, including with respect to text entry. As far as the latter is concerned, they enable the use of all fingers, not just one or two (and not the most suitable fingers for the purpose). Furthermore, they enable touch-typing. The fact that the number of keystrokes per character is constant rather than variable (as it is with the ITU E.161 keypad) allows the user to type at a steady rhythm. This favours speed and accuracy. The fingers do not dart around a tiny keyboard in search of the intended tiny control. There are no issues with aiming a finger at a much smaller target away from a resting position; the controls are as big as the finger the user engages them with and at the same location. The fingers engage the interface where they already are in contact with it, including when it comes to typing. Hitting a control never involves reaching for it. The user engages it, without moving the finger from where it is.

All else being equal, one keystroke per character may well by preferable to two, but that is not the case here. All things considered, the significance of the number of keystones per character recedes. Overall, the more favourable conditions under which users interact with atopical interfaces outweigh the additional keystroke typing a character with them requires.

6 Conclusion

An atopical interface is not a variant of the conventional topical interface paradigm that corrects the usability issues it entails, but a completely different interface concept, one that has substantial ergonomic advantages over the conventional interface. It not only supports the functionalities of the conventional topical interface - without the usability issues - but other functionalities beyond the scope of the topical interface paradigm. For instance, it allows the user to engage the interface without being able to see it, which, as mentioned, has advantages for the visually impaired in particular. Furthermore, it can accommodate a device of any shape a user can hold in a handshake like grip including a cylinder or a torus [16]. This opens up entirely new possibilities such as controlling a car radio without letting go of the steering wheel. In spite of the new and distinct approach atopical tactile user interfaces take, they do not call for any new ways of engaging an interface nor for any new or unusual finger movements or unfamiliar ways of holding a device. It will take little time for a new user to become familiar with them. What is more, they obviate the need for the problematic finger movements conventional interfaces require. In particular, they eliminate the need to reach for controls at predetermined locations where they can be difficult to engage, especially when the user is operating the device in-hand and the thumb, which is not the finger of choice for this, is the only finger available. Instead the atopical interface paradigm enables the user to engage an interface with the use of all fingers, not just one or two. To this end, the atopical interface makes use of the computing power of the device to dynamically put controls at the user's fingertips as and where they are in contact with the device as a result of the user's grip on it. This contrasts with the one-size-fits-all approach of conventional topical interfaces where all users interact with one and the same interface layout, possibly tweaked for left or righthand use, regardless of the size and structure of their hand.

Finally, it is worth noting the surfaces the two interface concepts employ for tactile input are complementary. While topical user interfaces concentrate input and output on the user facing side of the device, atopical interfaces use the front only for output and the sides and back of the device for input. Consequently, a device can be configured with both types of interface concurrently. Such a hybrid device enables the user to continue

[16] For more see Helke [7].

to operate legacy apps while transitioning to a new device configured with an atopical interface and new applications designed specifically for it.

REFERENCES

[1] Steven Hoober. 2013. How Do Users Really Hold Mobile Devices? https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php

[2] Shigeo Hiraoka, Isshin Miyamoto and Kiyoshi Tomimatsu. 2003. Behind Touch: Backside Keys with Tactile Sense Interface for Mobile Phones. In IPSJ Transactions Vol. 44 No. 11 2520-2527

[3] Wigdor, D, Forlines, C, Baudisch, P, Barnwell, J, and Shen, C 2007 LucidTouch: A See-through Mobile Device. UIST, 269–278

[4] Ian Hanson. 2018. Can you touch type on a touchscreen phone without looking at it? https://www.quora.com/Can-you-touch-type-on-a-touchscreen-phone-without-looking-at-it

[5] Christian Holst. 2013. A Guide To Designing Touch Keyboards (With Cheat Sheet) https://www.smashingmagazine.com/2013/08/guide-to-designing-touch-keyboards-with-cheat-sheet/#comments-guide-to-designing-touch-keyboards-with-cheat-sheet

[6] Michael Helke. 2016. Accommodative User Interface for Handheld Electronic Devices https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2017221141

[7] Michael Helke. 2018. Unbuttoning the Interface https://design-research.ch/synoptic-presentation-1

[8] Andrew C. Kamin-Lyndgaard. 2013. Detachable sensory-interface device for a wireless personal communication device and method https://patents.google.com/patent/US9285840

A APPENDICES

Summary of Issues with Topical Interfaces

	Issues grouped by subject		Disposition with atopical interface
1	underutilisation of hand	fewer than 5 fingers, usually only 1 or 2, engage interface due to lack of space	all 5 fingers engage the interface operatively
2		Allocating fingers exclusively to holding or operating device reduces number of fingers available to operate it	All fingers engaged in holding and operating the device concurrently
3		When using device with one hand, its interface is only reachable with thumb	all 5 fingers engage the interface operatively where they naturally come into contact with device
4	misallocation of fingers	dextrous fingers allocated to holding device and thumb, with limited dexterity, to engaging interface	all fingers serve to hold and operate the device, each as needed and within the limits of its ergonomic capabilities
6	physical interface	tactile interface coincides with display, fingers interacting with it eclipse display	tactile interface is at sides and back of device, using it does not eclipse display
8		interface at front is difficult to reach when operating device in-hand	interface covers sides and back, where hand naturally is in contact with device
9		limited interface size may call for controls that are very small and difficult to resolve	All operative contacts are the size of the contact patch of a finger, no small buttons
10	ambiguity	effective point of contact may touch outside of target area (fat finger problem)	No difference between contact patch of finger and operative contact area
11		finger may touch two keys	Each finger generates its own operative contact area, no overlap
12		stray contacts with nearby controls	No stray contacts, contact patch of a finger can not encroach on that of another
13	predetermined contact areas	controls can be difficult to reach	operative contacts with interface occur at natural point of contact of fingers

14		may require unorthodox finger movements	natural finger movements
15		user guides finger to location of control	interface places control at fingertip
16	handedness	Separate interface versions for left and right hand	interface recognises identity of fingers based on their position with respect to eachother irrespective of where they touch
17	targets	purely visual targets not usable with visual impairment	operative contacts do not require visual guidance, identification by audible tags.
18		keys too small for touch-typing	keys are full-size, usable without being seen; touch-typing possible but not tested
19	motility	holding device limits motility of finger engaging interface	no movement required; location of operative contact a side effect of holding
20	accessibility	virtual controls not usable	when provided with audible tags, visually impaired users can use controls the same way as sighted users
21	touch-typing	not possible	possible
22	two-finger gestures	not possible	not needed