Bimanual Interaction on the Microsoft® Office® Keyboard

Hugh McLoone, Ken Hinckley & Edward Cutrell
Microsoft, One Microsoft Way, Redmond, WA 98052, USA
{hughmcl, kenh, cutrell}@microsoft.com

Abstract: The Office Keyboard (on store shelves in major markets since October, 2001) seeks to enhance efficiency through unique application of bimanual interaction principles. The left hand performs navigation tasks (including view scrolling, application switching, and internet forward & back) as well as editing commands (Cut, Copy, and Paste) that are typically part of a compound mouse-keyboard action. The Office Keyboard's Cut, Copy, Paste, and Application toggle dedicated left-side keys are evaluated. Results in three different experimental task contexts show that the Office Keyboard is significantly faster than, or statistically equivalent to, the mouse or keyboard shortcuts (Ctrl-X, Ctrl-C, Ctrl-V and Alt+Tab) for all outcome measures that we collected. Most participants preferred the dedicated left-side keys to the other methods tested.

Keywords: keyboard, mouse, input devices, bimanual interaction, two-handed input, text entry, information navigation

1 Introduction

Keyboards and typewriters have been in use for well over 100 years (Yamada, 1980), yet modern computer keyboards have remained relatively unchanged even as the processing power, memory, display resolution, and the range of applications available for computers have all rapidly advanced. Researchers have explored alternate key layouts (Lewis, Potosnak & Magyar, 1997) and ergonomic designs such as split-angle keyboards (Honan, Serina, Tal & Rempel, 1995; Marklin & Simoneau, 1996). However, these explorations have not altered the fundamental functionality offered by the keyboard.

Advances in technology and the research literature suggest avenues for keyboard innovation. Bimanual interaction has been a significant focus in the literature (Buxton & Myers, 1986; Guiard, 1987; MacKenzie & Guiard, 2001), yet commercial keyboards have been slow to fully develop this approach. Our new keyboard design (Fig. 1) embraces bimanual interaction with a new "left pod" function area at the left of the keyboard. We discuss five design studies related to the left pod, with 64 total participants, and present a quantitative experimental study of the final left pod design, with 12 additional participants.

We discuss properties of keyboards in general that have made them enduring and successful, and critique some design and usability problems of modern computer keyboards. We then discuss the design of the Office Keyboard, including design issues, alternatives considered, and usability evaluations. We present an experimental analysis of performance with the left pod in comparison to the mouse and chorded keyboard shortcuts. We refer to these as "chords" since they require pressing a modifier key in combination with another key, such as Ctrl+X for Cut.

The results show that the dedicated keys of the left pod can have advantages in many task contexts. This is the first performance data we are aware of in the literature for a keyboard intentionally embodying principles of bimanual interaction.

2 Related Work

Review articles on keyboards typically stress factors such as size, shape, activation force, and travel distance of keys; tactile and auditory feedback provided by the keys; and keyboard layout (Lewis et al., 1997). Compared to numerous articles on pointing device design and evaluation (e.g. (Balakrishnan, Baudel, Kurtenbach & Fitzmaurice, 1997; Rutledge & Selker, 1990)), there are few works describing new keyboard designs in the literature (although soft keyboards (Sears, 1993; Zhai, Hunter & Smith, 2000) and keyboard designs for mobile devices (MacKenzie, Kober, Smith, Jones & Skepner, 2001) are recent research areas). The Xerox Star keyboard (Fig. 2) included keys such as Undo, Move, Copy, Open, Prop's (show properties), and Same (copy properties) (MacKenzie & Guiard, 2001; Smith, Irby, Kimball, Verplank & Harslem, 1982). Several of these keys were on the left, suggesting one of the first applications of bimanual control in keyboard design. Since the development of the Office Keyboard, Logitech has marketed a keyboard with a scrolling wheel and a Back button on the left side.

Fig. 1. Final design of the left pod of the Microsoft Office Keyboard (with image of entire keyboard for context).

People perform most tasks using both hands in complementary roles, where the non-preferred hand sets a frame of reference for the detailed actions of the preferred hand (Guiard, 1987). Several researchers have argued that the background task of navigating a document should be assigned to the non-preferred hand, while the preferred hand operates the mouse (Buxton & Myers, 1986; Guiard, Baudouin-Lafon & Mottet, 1999; MacKenzie & Guiard, 2001). Buxton and Myers report that for a compound navigation/selection task, two-handed operation was significantly faster than one-handed operation for both novice and expert users.

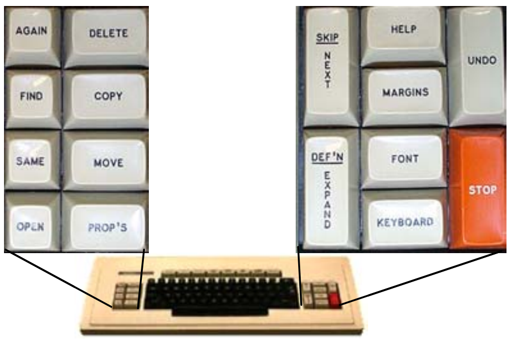

Fig. 2. The Xerox Star keyboard with detail of the function key groups. Note that Copy is on the left.

3 General Design Properties of the Keyboard

Keyboards remain the mechanism of choice for text entry. The resiliency of the keyboard is the result of how keyboards complement human skills, and may make them difficult to supplant with new input devices or technologies.

3.1 Skill Acquisition and Skill Transfer

Procedural memory is a specific type of memory that encodes repetitive motor acts. Once an activity is encoded in procedural memory, it requires little conscious effort to perform (Anderson, 1980). For keyboards, this results in the skill of touch-typing. The process of encoding an activity in procedural memory can be formalized as the *power law of practice*: $\mathbf{T} = a\mathbf{P}^b$, where **T** is the time to perform the task, **P** is the amount of practice, and a and b are constants that fit the curve to observed data. This suggests that changing the keyboard can have a high re-learning cost. However, a change to the keyboard can succeed if it does not interfere with existing skills, or allows a significant transfer of skill; for example, some ergonomic keyboards have succeeded by preserving the basic key layout, but altering the typing pose to help maintain neutral postures (Honan et al., 1995; Marklin & Simoneau, 1996).

3.2 Eyes-Free Operation

With practice, users can memorize the location of commonly used keys relative to the home position of the two hands. Experimental work suggests that this type of reference frame requires little or no visual attention (Balakrishnan & Hinckley, 1999). Chords such as Ctrl+C for Copy leverage these mechanisms to enable command entry that is both fast and minimally demanding of attention for skilled users. By contrast, most graphical widgets require visual diversion (Kabbash, Buxton & Sellen, 1994) to guide the mouse (although marking menus (Kurtenbach, Fitzmaurice, Owen & Baudel, 1999; Kurtenbach, Sellen & Buxton, 1993) offer a notable exception).

3.3 Tactile Feedback

On a mechanical keyboard users can feel the edges and gaps between the keys, and the keys have an activation force profile that provides feedback of the key strike. In the absence of such feedback, as on touchscreen keyboards (Sears, 1993), performance may suffer and users may not be able to achieve eyes-free performance (Lewis et al., 1997).

4 Problems with Chords as Shortcut Keys

Information workers repeatedly use a core set of commands in software applications. For example, nearly all users have cut, copied, and pasted content. These actions still do not have quick and obvious dedicated controls on the keyboard. Most keyboards do provide a few power keys (MacKenzie & Guiard, 2001), such as Delete and Page Down, but many common operations must be accessed by holding down a modifier key in combination with a text character, such as Ctrl+C for Copy. This approach has numerous drawbacks:

Learning time: Users must notice the chord in the menu and remember it. Novice users do not know chords; some users never learn them. In the absence of chords, users may employ inefficient mouse-based (menu or tool bar) methods. Chords may require frequent repetition over a long period of time to be memorized (Kurtenbach et al., 1999).

Biomechanical Issues: Some chords may require hyper-extending, hyper-flexing, or splaying the fingers, and may necessitate uncomfortable static muscular loads (Putz-Anderson, 1988) to hold down modifier keys such as Ctrl and Alt.

Activation Errors: It is easy to hit the wrong key when performing some common chords due to the lack of distinct tactile landmarks. For example, users sometimes hit Ctrl+V when they intend to hit Ctrl+C.

5 Bimanual Design Principles

Assigning functions to the left side of the keyboard should be carefully considered; Guiard suggests three principles for the division of labor between the hands (Guiard, 1987). For right-handers, the principles are *right-to-left reference* (right hand activity occurs within the frame-of-reference defined by the left), *scale asymmetry* (movements of the right hand occur at higher spatial and temporal frequencies than the left), and *left-hand precedence* (action starts with the left hand). We propose the following two general classes of computer tasks that are suitable for left-hand placement on a keyboard:

Computer navigation tasks (i.e. the task of getting specific content to be visible on the screen) seem to fit these criteria: movement of mouse cursor is within the boundaries of the screen, cursor movement and selection is typically a precise action, and viewing the desired content must precede any action upon it. Several researchers have considered scrolling in this context (Buxton & Myers, 1986; Guiard et al., 1999; MacKenzie & Guiard, 2001). But other navigation tasks, such as switching between documents or navigating a history list on the internet, have not been discussed, yet may also benefit from a bimanual task flow.

Compound selection-action command sequences offer another class of mouse/keyboard activities that may benefit from bimanual interaction, by assigning the action (e.g. Copy key) to the left hand and the selection (sweeping out a region with the mouse) to the right hand. This follows Guiard's scale asymmetry principle, but seems to violate left-hand precedence, since the selection comes first. Nonetheless the action seems quick and natural.

The vision for this new keyboard is the unique application of bimanual principles to the frequently performed actions completed by computer users during 1) *consumption* activities such as browsing the web using back / forward navigation and view scrolling and 2) *creation* or *composition* activities such as editing documents using cut, copy, paste; view scrolling; and switching between application windows.

6 Left Pod Design and Usability Testing

The design team was familiar with the literature advocating bimanual interaction and decided to experiment with this paradigm. We designed the keyboard for users who use the mouse in the right hand, as such users represent the largest market segment (at least 95% of users based on survey of mouse location for over 900 computer users)[1]. We pursued a series of user studies with functional prototypes and representative tasks to systematically explore how to implement scrolling and application switching, as well as how to arrange these elements with Cut-Copy-Paste and internet Back /Forward on the left pod.

6.1 Application Switching

Nine participants employed four application switching prototypes: a rocker switch, two horizontal buttons, one button, and a dial (knob) to copy and paste items between several open applications. Participants also performed the task using their "current method," which for most users was using the mouse to select an application from the task bar.

Overall, users preferred the rocker switch because it was quick and had a low vertical profile. Users also felt that its look and feel communicated the function of switching applications. The two buttons were second most preferred with equal mix of positive and negative reactions. Users disliked the single button because if they depressed the button one time too many, they had to cycle through all applications again to find the desired one. The dial was poorly received because it requires multiple fingers to turn, and the finger action required is dissimilar from other keyboard actions.

6.2 View Scrolling

We built four functional scrolling prototypes: a thin wheel, a wide wheel (similar to that seen in the final design of Fig. 1), two vertical keys, and a touchpad, all at the left side of the keyboard. We implemented the touchpad as suggested by the literature (Buxton, 1994; Buxton & Myers, 1986). Twelve participants participated in a study; one participant was left handed but used the mouse in the right hand.

Users employed the devices to scroll to various positions in a long document. Unfortunately, the wide wheel suffered technical problems, so we had participants feel it as a mechanical prototype only. The average rank preference of the other devices was touchpad, 1.5; thin wheel, 2.6; two keys, 3.3. Although they did not rank-order the wide wheel, participants did comment that the wide wheel was easier to acquire than the thin wheel; they also liked being able to use single or multiple fingers to roll it. After using the prototypes, 83% of participants found placement of the scrolling feature on the left of the keyboard acceptable; 17% wanted to move it. Although the touchpad tested well, it did not meet the cost or marketing constraints of the product, so the team proceeded with a wheel.

6.3 Arrangement of Functions on the Left Pod

Two separate studies were conducted to determine optimal arrangement of features on the left-side pod. One study with 10 participants utilized three mechanically but not electronically functioning models. A second study with 9 participants used three functional prototypes constructed with various arrangements for cut, copy, paste, application toggle, and the wheel.

Participants performed tasks using each left-pod design. Tasks included 1) scrolling and then copying and pasting with the left pod (right hand on the mouse), 2) scrolling and switching applications with the pod (right hand on the mouse), 3) typing and scrolling (using the left pod), and 4) typing and then copying and pasting with the left pod.

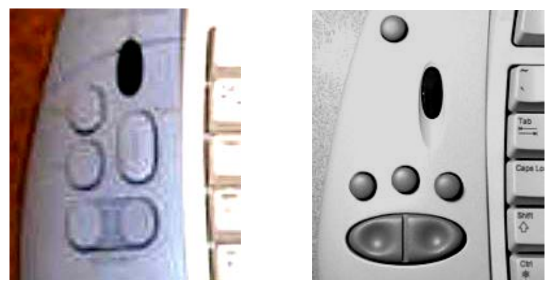

Fig. 3. Left-Pod Models. Both models have Application Toggle on the bottom. *Left*: Triangular arrangement of Cut, Copy and Paste keys. *Right*: Single-row arrangement of Cut, Copy, and Paste keys.

The first study revealed several advantages and disadvantages of the prototypes shown in Fig. 3. For the prototype on the left, the fingers do not naturally fall on the Cut-Copy-Paste buttons because they are in a triangle rather than a row; users have to "learn" where they are. In the prototype pictured at the right, the horizontal layout of the Cut-Copy-Paste keys falls naturally under the ring, middle, and index fingers; the frequency of use for Cut-Copy-Paste[2] matches the dexterity and strength of these respective fingers (Armstrong, Foulke, Martin, Gerson & Rempel, 1994). The horizontal arrangement also has a fluid transition from the home row of the keyboard. It is easy to move between functions by shifting the hand to touch a different

[1] Dennerlein, J., personal communication of unpublished field survey of mouse location among over 900 computer users, Harvard University.

[2] Based on data collected from an instrumented version of Microsoft Office at more than three hundred of test locations, users paste more often than copy, with cut being the least frequently used.

row of the pod.

A second study tested prototypes that followed alternative arrangements of the row-by-row design philosophy of the *Fig. 3,* (*right*) model. Participants preferred having the application switch below Cut-Copy-Paste. Placing the scroll function near the top allows the base of the palm to rest on the keyboard palm rest, supporting the curling motion of the finger needed to roll the scroll wheel.

Overall, participant feedback from the studies showed that implementing a wide scroll wheel was preferred; cut, copy, and paste should be horizontally arranged; and the left-pod functions should be designed for access patterns resulting from either browsing (Scrolling and Back/Forward navigation) or editing activities (scrolling, Cut-Copy-Paste, and switching applications).

7 Experimental Study of the Left Pod

Our goal for the left pod was to provide efficient, easy-to-learn commands that would be competitive with existing chords (i.e. Ctrl+X, C, V for Cut-Copy-Paste, and Alt+Tab for application switching). Many computer users employ these chords in favor of the mouse, presumably because they are fast and keep the hands close to the home row.

For example, many users articulate Ctrl+C by moving the little finger from the letter “A” to Ctrl (5.0 cm) and the index finger from the letter “F” to the letter “C” (2.2 cm). But for our dedicated keys, moving one’s middle finger from the letter “D” to the Copy key involves a lateral motion of 13.3 cm. By Fitts’ Law, this longer movement should be slower. Does it actually take longer to acquire our dedicated keys on the left pod? What is the performance trade-off if multiple functions are required (e.g. Copy, then Paste)? To investigate these issues, we devised three experimental tasks that examine performance for Cut, Copy, Paste, and application switching.

7.1 Conditions

Participants performed our three experimental tasks in three conditions. Order of the conditions was randomly assigned.

- *Chord Keys*: Uses Ctrl + X, C, or V to activate the Cut, Copy, and Paste functions. In the Form Fill task, Alt+Tab is used for application switching.
- *Dedicated Keys*: Uses the Cut, Copy, and Paste keys on the left pod of the Office Keyboard. In the Form Fill task, participants also use the Application toggle switch.
- *Mouse*: Uses the mouse to click on icons in the toolbar (see Fig. 4) to activate Cut, Copy, or Paste. The icons measured 6.7 x 7.6 mm. In the Form Fill task, participants switch applications using the task bar at the bottom of the screen, and the Edit menu is used to Copy since there is no toolbar in Internet Explorer.

All conditions used the Office Keyboard, as currently sold, for required keyboard interactions.

7.2 Participants

Twelve persons (7 male, 5 female), aged 32-65 years, participated in one or more of the experimental tasks. All participants used the mouse in the right hand and used straight (non-ergonomic) keyboards. One user was left-handed but used the mouse in the right hand. None were current users of the Office Keyboard. Six “expert” participants normally used the Ctrl-X, Ctrl-C, Ctrl-V, and Alt-Tab chords, while the remaining 6 users did not know these chords. In total 10 participants performed the Homing Task, 12 participants performed the Multi-Paste task, and 9 participants performed the Form Fill task. All participants received a software gratuity.

7.3 Experiment 1: Homing Task

We began with a compound task designed to analyze hand movement times for a single Cut, Copy, or Paste action interleaved with selection (highlighting a word) and typing. We call this the *Homing Task*. The Dedicated Keys are further from the home row than the traditional Chord Keys, so moving to the left pod just to activate a single function, and then returning to the home row, is perhaps the worst case in which to analyze the left pod.

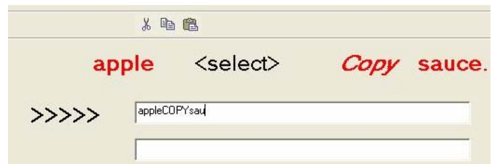

Fig. 4. Homing task. The user is prompted with the exact steps to perform for each trial. As feedback, the word COPY or CUT appeared when the user activated the respective function (the clipboard contents inserted for Paste).

Homing Task Syntax

Our homing task presents the user with a series of fields (type-in boxes) to complete. The homing task consists of typing in a word, activating the Cut, Copy, or Paste function, and then typing in a second word (Fig. 4). The exact task syntax is:

1. Click the mouse in the first empty type-in box to place the insertion point (carat).
2. Type an initial word. This word was always either apple or peach. Following (Myers, 2000), these words were chosen as simple 5-letter words with both the first and last letters on the left (*apple*) or right (*peach*) hands. This allowed us to examine these two different cases, and it also prevented task performance from becoming too repetitive and over-learned.
3. In the same or next type-in box, when the command to activate is Cut or Copy, the user selects the word they just typed by clicking and dragging the mouse over it. We required this step because selection is often a prerequisite to activating Cut or Copy in real task contexts. When the command to activate was Paste, no selection was required.
4. Type in an ending word in the final, ninth type-in box trial. This word was always either sauce (following apple) or lemon (following peach), again with first and last letters on the left (*sauce*) or right (*lemon*) hand. Users typed a period (.) at the end of the word, which ended the trial.

The stimulus for the next trial would then appear and the process would start over from step 1. Participants performed 9 practice trials (discarded from final analysis) with each device, followed by 3 blocks of 9 trials each.

Dependent Variables

Several aspects of movement (switching) time and error metrics are of interest for this task:

Acquisition Time is the time to move from the home row of the keyboard to activate the correct function (Cut, Copy, or Paste) for the trial. It is calculated as the time between typing the last character of a word and the subsequent activation of the correct function. Note that this time includes selection (highlighting the word to Cut or Copy).

Homing Time is the subsequent time to resume typing the first character of the next word (i.e., the time to move one's hand back to the home row position).

Acquisition-After-Select Time is a subset of the Acquisition Time, calculated as the time between completing selection (highlighting) of text with the mouse and the subsequent activation of a Cut or Copy command. (Selection is not required prior to a Paste command; this metric is not defined in that case).

Word1Err & ***Word2Err*** are the frequency of typing errors in the two typed words of a trial. No matter how badly a word was mistyped, just a single error was recorded.

FnErr records any erroneous activation of a function (e.g. Cut instead of Copy).

Users were instructed to repair all errors. We analyzed only the hand switching times, not the time spent typing words. We found that the different word pairs (apple/sauce vs. peach/lemon) had very little impact on our outcome measures, so our analyses collapse this factor.

Homing Task	**Dedicated Keys**	**Chord Keys**	**Mouse**
Acquisition Time	2.67	2.85	3.12
Homing Time	1.08	1.06	1.08
Acq-After-Select	1.42	1.52	1.56

Fig. 5. Timing results (sec) for the Homing Task.

Homing Task Results

Acquisition Time for the Dedicated Keys was 15% faster than the Mouse and 5% faster than the Chord Keys (Fig. 5). The Homing and Acquisition-After-Select times were virtually identical for all three conditions.

For this very simplistic Homing Task, the Dedicated Keys provide performance comparable to, or possibly even slightly faster than existing methods, despite being located further from the home row than the Chord Keys. Previous studies of acquisition times have looked at hand movement times without an integral selection operation, with typical values of about 700-800 msec, including the 300 msec required to strike a key (Card, Moran & Newell, 1983; Myers, 2000). The much larger values for Acquisition Time in our study reflect the additional time required for the integral selection operation. However, our Homing Time metric does not include selection but still is about 300-400 msec longer than comparable studies. This may indicate the presence of a mental preparation step (from the traditional keystroke-level model (Card et al., 1983)), which may be necessary since we varied the stimuli to prevent over-learning of the task sequence.

Homing Task Statistical Analysis

We performed a 3 x 3 x 3 doubly multivariate analysis of variance (repeated measures MANOVA) on the factors of Device (Dedicated Keys, Chord Keys, or Mouse), Function (Cut, Copy, or Paste), and Block (3 blocks). We also performed planned comparisons of the Dedicated Keys to the other two levels of Device. For the Acquisition-After-Selection Time metric, we performed a separate 3 x 2 x 3 (Device x Function x Block) repeated measures ANOVA, since for this metric only two levels of Function (Copy and Cut) exist: recall that no selection was required for Paste. Analyses of all timing data were based on the median times for each cell (3 trials). Using medians helps mitigate the skewing that typically occurs for time measures of human performance. Error metrics were based on the means.

For the factor Device, there was a significant overall effect for Acquisition Time ($F_{(2,16)}=3.74$, $p<.05$). Dedicated Keys and Chord Keys were similar while the Mouse was considerably slower (see Fig. 5).

There were also significant effects for Function, reflecting the different task requirements of the Cut, Copy, and Paste actions. As expected, Acquisition Time for Paste commands was significantly faster than either Copy or Cut ($F_{(2,16)}=74.4$, $p<.001$), since no selection was required for Paste. There was a slightly faster Homing Time ($F_{(2,16)}=4.1$, $p<.05$) after Paste than after Cut (but neither differed significantly from Copy). Finally, there were significant differences in error frequencies for FnErr ($F_{(2,16)}=104.4$, $p<.001$) and Word1Err ($F_{(2,16)}=97.1$, $p<.001$), with the most errors for Cut and the fewest for Paste (Cut > Copy > Paste). These errors did not significantly vary by device.

We observed a learning effect for Acquisition Time; times in Block 3 were significantly faster than in Block 2 ($F_{(2,16)}=18.0$, $p<.001$). The Device X Block interaction for Homing Time was also significant ($F_{(2,16)}=3.0$, $p<.05$). The Dedicated keys improved from slowest in Block 1 to fastest in Block 3, but the Chord Keys and Mouse conditions exhibited nearly constant Homing Time.

7.4 Experiment 2: Multi-Paste Task

The Multi-Paste task extends the Homing task so that the participant performs multiple paste operations following an initial Copy or Cut action. This allows us to examine task performance when the user switches away from the home row keys to perform several successive operations. This is a common pattern that may occur when pasting the same content to several locations in a document or spreadsheet.

The initial steps of the Multi-Paste task are identical to steps 1-4 of the Homing task: (1) place the insertion point; (2) type a first word (apple or peach); (3) select the word; and (4) Cut or Copy the word as prompted. However, the Multi-Paste task differs by now requiring the user to Paste the clipboard contents to one or more locations:

5. A "Paste →" prompt is displayed before the next field. The user clicks on the field and pastes the clipboard contents. This step is randomly repeated 1-5 times.
6. After the last Paste, the user is prompted to complete the task by typing a final word (sauce or lemon, respectively). Typing a final period (.) ends the trial.

Participants performed 4 practice trials with each device, followed by 2 blocks of 10 experimental trials each. The first block used apple/sauce as the word pair to type; the second block used peach/lemon as the word pair.

Results of Multi-Paste Task

We recorded similar outcome measures to Experiment 1, plus two more metrics: *Initial Paste Time,* the time required to perform the first paste function, and *Average Paste Time*, the average time for the participant to perform the subsequent 1-5 paste actions per trial. Of particular note, the average paste time with the Dedicated Keys was 22% faster than the Chord Keys and 36% faster than the Mouse (Fig. 6).

Multi-Paste Task	**Dedicated Keys**	**Chord Keys**	**Mouse**
Acquisition Time	3.15	3.01	3.15
Homing Time	1.96	2.29	2.23
Initial Paste Time	1.53	1.80	1.82
Avg. Paste Time	1.17	1.4	1.82

Fig. 6. Multi-Paste Task means (sec) for timing metrics.

Unlike Experiment 1, Acquisition Time was very similar between the different devices for Multi-Paste Task, as seen in Fig. 6. However, Cut and Copy were used for the initial function (whereas Experiment 1 included Paste, which was significantly faster, as a third possibility). This accounts for the different absolute means obtained in this experiment. It is also possible that mental or physical preparation for the subsequent Paste actions, which did not occur in Experiment 1, caused Acquisition Time to change slightly.

Likewise, Homing Time in Experiment 1 measures resumption of typing after a single Cut, Copy, or Paste operation, whereas Experiment 2 measures the time from the last of possibly multiple paste operations. Thus the frequency of operations, and the reaction time to decide whether a paste operation is the last one, resulted in a different mean for the Homing Time in Experiment 2.

Multi-Paste Task Statistical Analysis

We performed a 3 x 2 x 2 repeated measures MANOVA on the factors of Device (Dedicated, Chord, or Mouse), Function (Cut or Copy), and Block (1 or 2).

Device was significant overall for Average Paste Time, with the Dedicated Keys significantly faster than both other methods ($F_{(2,22)}$=16.0, $p<.001$). There was a trend for Dedicated Keys to be faster for the Initial Paste Time, but this did not reach significance (Dedicated vs. Chord, $p<0.10$, Dedicated vs. Mouse, $p<.06$). These results suggest that once the Dedicated Keys are acquired, subsequent multiple Paste operations can be performed more quickly than with the other methods that were tested.

Function (Cut vs. Copy) was significant for the Initial Paste Time, with Copy faster than Cut ($F_{(1,11)}$=6.9, $p<.025$). Participants were also more likely to hit the wrong function when performing a Copy operation than a Cut ($FnErr_{(1,11)}$, F=80.0, $p<.001$). We also found a significant Function X Device interaction. For the Average Paste Time, there was a greater difference between the Dedicated Keys and Mouse conditions for the Cut function than for the Copy function ($F_{(2,22)}$=9.3, $p<.001$).

For Block, both Initial Paste Time ($F_{(1,11)}$=17.9, $p<.001$), and Average Paste Time ($F_{(1,11)}$=17.9, $p<.001$) showed significant improvement from Block 1 to Block 2. There was also a significant interaction of Device X Block for Homing Time ($F_{(2,22)}$=3.5, $p<.05$), which represents a slight tendency for Homing Time to improve from Block 1 to Block 2 for the Dedicated and Chord Key conditions, but not for the Mouse condition.

7.5 Experiment 3: Form Fill Task

The Form Fill task integrated use of Copy and Paste with application switching. The task was analogous to filling out a form on the web, or looking for information in one document that needs to be pasted into another. No typing was required for this task.

Participants were given cue words (i.e. Stitch), switched to the application with a list of answers ("in time saves nine"), copied the answer, switched back to the form, and pasted the answer in the appropriate type-in box (Fig. 7). The exact steps required for the task were as follows:

1. Place the insertion point in the field for the cue word.
2. Switch to the application with the answers. This was a simple web page hosted in Internet Explorer.
3. Select (highlight) the correct answer using the mouse.
4. Copy the answer.
5. Switch back to the form (our test application).
6. Paste the answer (note the insertion point is already placed in the correct field in step 1).

This completed the task, and the next cue word was shown.

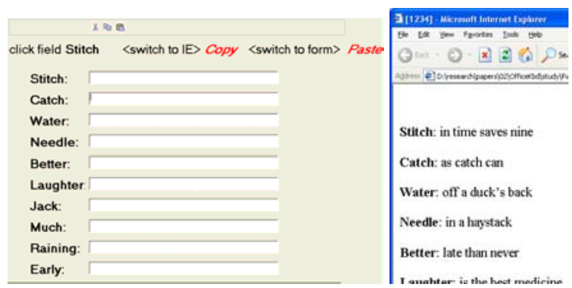

Fig. 7. Form Fill task. Left: Cue words indicate the information to fill in on the form. Right: Web page with the corresponding answers. These were two separate full-screen applications but are shown together here for clarity.

This study was performed with a total of 3 currently running applications (the form, the answer sheet, and one other "distracter" application). The effectiveness of different application switching techniques likely depends on the number of applications running, so future studies should explore this more systematically.

Results of Form Fill Task

We recorded the mean times for switching applications (both Switch Time to Answers, and Switch Time Back to the form), Acquisition-After-Selection Time (for this task, this is the time to acquire the Copy function after highlighting the correct answer with the mouse), and the Paste Time (time between switching to the Form and the activation of Paste). The only error metric for this task was FnErr, which indicates activation of an incorrect function key, e.g. Copy instead of Paste). Participants were able to perform the Form Fill task significantly faster, for most of the above metrics, using the Dedicated Keys. The percent performance advantage for each of the metrics is summarized below (Fig. 8).

Form Fill Task Statistical Analysis

We performed a 3 x 2 repeated measures MANOVA on the factors of Device (Dedicated, Chord, or Mouse) and Block (1 or 2). Device was significant overall for Switch Time to Answers ($F=10.4$, $p<.001$) and Switch Time Back ($F=13.0$, $p<.001$), with Dedicated Keys significantly faster than both Chord Keys and the Mouse condition. Paste Time was also significant ($F=17.5$, $p<.001$), but the Dedicated Keys did not differ significantly from the Chord Keys for this metric (both were significantly faster than the Mouse). Device was not significant overall for the Acquisition-After-Selection Time because of a high standard deviation in the Mouse condition; however, a pairwise comparison revealed that the Dedicated Keys were significantly faster than the Chord Keys ($F=23.3$, $p<.001$) for this metric.

Form Fill Task	**Dedicated Keys vs. Chord Keys**	**Dedicated Keys vs. Mouse**
Switch To Answer Time	+33%	+34 %
Switch Back Time	+25%	+39%
Acq-After-Select Time	+32%	+47%
Paste Time	+4%	+43%

Fig. 8. Percent performance gain for Dedicated Keys.

We observed significant learning from Block 1 to Block 2 for most performance metrics, including Paste Time ($F=27.0$, $p<.001$), Switch Time to Answers ($F=10.9$, $p<.02$) Switch Time Back ($F=5.5$, $p<.05$), and Acquisition Time After Selection ($F=14.2$, $p<.005$). We also found a significant Device X Block interaction for the FnErr metric ($F=4.3$, $p<.05$). Inspection of the means indicated that participants made significantly more errors with the Chord Keys than with the Mouse in Block 1 only.

8 Qualitative Results

Immediately after trying each interaction method, participants scored 7-point Likert scale questions focusing on ease of use and learning, speed and errors, comfort, attention and memory load, and overall enjoyment and interest (see Fig. 9). Participants also shared comments and impressions of the interaction method used for the task.

For the Homing task, the Likert scale questions yielded little difference between the interaction methods, but in all cases, the Dedicated Keys rated equal to or better than the Chord Keys or Mouse conditions. In the Multi-Paste task, participants rated the Dedicated Keys significantly faster and quicker (Q2), less tiresome (Q7), and more enjoyable (Q10) to use than either the Chord Keys or the Mouse (paired t-tests, $p<.05$). Participants liked the Dedicated keys because they were fast and intuitive: as one participant said, "Love it. Natural. Easy for my brain to follow." However, several participants felt that the Dedicated keys were hard to locate by feel and that their feel upon activation was too soft compared to the main keyboard keys.

For the Chord Keys, participants felt that it was more difficult to learn and remember which key performed which function. Participants new to the Chord Keys often reported having to shift their vision up and down to find the Chord Keys, interrupting their flow of typing. Many participants also felt that the Chord keys were awkward and tiresome. In the Mouse condition, participants felt that there was too much hand motion required to complete the task, and felt that it was distracting to go back and forth between mouse and keyboard. However participants felt that using the mouse was simple, familiar, and kept visual focus on the display.

For the Form Fill task, participants rated the Dedicated keys significantly higher than either the Chord Keys or the Mouse on most questions (Fig. 9). Participants liked the Dedicated keys because they would be "very fast for common activities on the computer in real life." Participants also felt that they could keep their attention focused and complete the task with less hand motion. The only negative comment was that it was novel and had to be learned.

Likert Statement	**Ded. Keys**	**Chord Keys**	**Mouse**
Q1. The keyboard actions are simple and easy to learn.	6.8^{A}	4.0^{B}	5.5^{AB}
Q2. The keyboard actions are fast and quick.	6.8^{A}	3.9^{B}	4.5^{B}
Q3. I can easily and quickly move between documents using this keyboard and/or mouse.	6.5^{A}	4.3^{B}	4.6^{B}
Q4. I can easily and quickly cut, copy, and paste using this keyboard and/or mouse.	6.6^{A}	4.6^{B}	4.5^{B}
Q5. I do not make errors while performing these actions.	6.1^{A}	4.6^{B}	5.1^{AB}
Q6. The keyboard actions are comfortable.	6.6^{A}	3.2^{B}	4.1^{B}
Q7. The keyboard actions are not tiresome.	6.6^{A}	3.4^{B}	3.9^{B}
Q8. When using this keyboard, I can keep my attention focused on what I am doing.	6.4^{A}	3.4^{B}	4.8^{B}
Q9. It is easy to remember to use the features of this keyboard.	6.8^{A}	4.2^{B}	5.6^{B}
Q10. The keyboard actions are enjoyable. I like this keyboard.	6.7^{A}	4.0^{B}	4.4^{B}
Q11. The keyboard actions are desirable. I want this keyboard.	6.8^{A}	4.0^{B}	4.4^{B}

Fig. 9. Likert statements (7=strongly agree, 1=strongly disagree), with results for the Form Fill task. Values with a common superscript do not differ significantly (paired t-tests, $\alpha=.05$).

Some participants familiar to the Chord Keys liked them, but most comments were negative. Participants felt that they were awkward, clumsy, unintuitive, unfamiliar, cramping, or required too much reaching with the fingers. Participants also commented that the Chord Keys seemed to require extra steps to complete the action. Participants felt that using the Mouse for this task was too time consuming. Participants noted and disliked the frequent up-down motion required to switch between the Copy function and the task bar.

8.1 Overall Subjective Preferences

After completing all tasks with all interaction methods, participants were instructed to rank order the interaction methods. Eight of twelve (67%) participants selected the Dedicated keys as their preferred means of completing Cut-Copy-Paste actions. Three participants preferred the Chord keys. One person preferred the Mouse. Eight of nine (89%) participants who performed the Form Fill task selected the Dedicated keys as their preferred means of switching applications. One person preferred the Mouse. No one preferred the Alt+Tab method for this task.

9 Summary and Conclusions

We have discussed the interaction properties of keyboards in general and described the design and evaluation of a new keyboard embodying principles of bimanual interaction. Encouraging bimanual interaction may allow keyboards to support more productive work that may also reduce demands on the preferred (mouse) hand. Our keyboard incorporates bimanual navigation (scrolling, application switching, and internet Back/Forward). It also incorporates Cut-Copy-Paste keys which distribute a selection/action sequence across the two hands, allowing the user to perform the sequence in rapid succession.

Our research contributes the first quantitative performance data for a keyboard designed to support a bimanual work flow. For the tasks that we evaluated, our experimental results show that dedicated keys on the left side of the keyboard offer a viable alternative to using either chords or the mouse. Our results also suggest that dedicated functions on the left are well accepted by most users and can provide significantly faster performance for some tasks.

We would like to extend our results with longitudinal studies, by examining user adoption, patterns of use, and collecting further quantitative performance data. We would also like to explore new keyboard designs that might extend the range of activities that can be performed with natural and quick two-handed actions. Bimanual interaction techniques described in the literature, such as panning/zooming interfaces, ToolGlass interactions (Kabbash et al., 1994), and 3D manipulation, suggest promising future directions.

Acknowledgements

We thank colleagues Ken Fry, Steve Bathiche, James Cauthorn, Melissa Jacobson, Carl Ledbetter, Jonathan Hayes, Corey Ladas, Dick Compton, Timothy Muss, William Vong, and James Reger.

References

Anderson, J. R. (1980). Chapter 8: Cognitive Skills. Cognitive Psychology and Its Implications. San Francisco, W. H. Freeman**:** 222-254.

Armstrong, T., J. Foulke, B. Martin, J. Gerson & D. Rempel (1994). “Investigation of applied forces in alphanumeric keyboard work.” J. Armerical Industrial Hygiene Assocation **55**: 30-35.

Balakrishnan, R., T. Baudel, G. Kurtenbach & G. Fitzmaurice (1997). The Rockin'Mouse: Integral 3D Manipulation on a Plane. CHI'97 Conf. on Human Factors in Computing Systems. 311-318.

Balakrishnan, R. & K. Hinckley (1999). The Role of Kinesthetic Reference Frames in Two-Handed Input Performance. Proc. ACM UIST'99 Symp. on User Interface Software and Technology. 171-178.

Buxton, W. (1994). “Two-handed document navigation.” XEROX Disclosure Journal **19**(2): 103-108.

Buxton, W. & B. Myers (1986). A Study in Two-Handed Input. Proceedings of CHI'86: ACM Conference on Human Factors in Computing Systems, Boston, Mass., ACM, New York. 321-326.

Card, S., T. Moran & A. Newell (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ, Lawrence. Erlbaum Associates.

Guiard, Y. (1987). “Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model.” The Journal of Motor Behavior **19**(4): 486-517.

Guiard, Y., M. Baudouin-Lafon & D. Mottet (1999). Navigation as Multiscale Pointing: Extending Fitts' Model to Very High Precision Tasks. CHI'99.450-457.

Honan, M., E. Serina, R. Tal & D. Rempel (1995). Wrist Postures While Typing on a Standard and Split Keyboard. Proc. HFES Human Factors and Ergonomics Society 39th Annual Meeting. 366-368.

Kabbash, P., W. Buxton & A. Sellen (1994). Two-handed input in a compound task. Proceedings of CHI'94: ACM Conference on Human Factors in Computing Systems, Boston, Mass., ACM, New York. 417-423.

Kurtenbach, G., G. Fitzmaurice, R. Owen & T. Baudel (1999). The Hotbox: Efficient Access to a Large Number of Menu-items. CHI'99. 231-237.

Kurtenbach, G., A. Sellen & W. Buxton (1993). “An emprical evaluation of some articulatory and cognitive aspects of 'marking menus'.” Journal of Human Computer Interaction **8**(1).

Lewis, J., K. Potosnak & R. Magyar (1997). Keys and Keyboards. Handbook of Human-Computer Interaction. M. Helander, T. Landauer and P. Prabhu. Amsterdam, North-Holland**:** 1285-1316.

MacKenzie, I. S. & Y. Guiard (2001). The Two-Handed Desktop Interface: Are We There Yet? Proc. ACM CHI 2001 Conf. on Human Factors in Computing Systems: Extended Abstracts. 351-352.

MacKenzie, I. S., H. Kober, D. Smith, T. Jones & E. Skepner (2001). LetterWise: Prefix-based Disambiguation for Mobile Text Input. ACM UIST 2001 Symp. on User Interface Software & Tech. 111-120.

Marklin, R. & G. Simoneau (1996). Upper extremity posture of typists using alternative keyboards. ErgoCon'96. 126-132.

Myers, B., Lie, K., Yang, B. (2000). Two-Handed Input using a PDA and a Mouse. CHI 2000. 41-48.

Putz-Anderson, V. (1988). Cumulative trauma disorders: A manual for musculoskeletal diseases of the upper limbs. Bristol, PA, Taylor & Francis.

Rutledge, J. & T. Selker (1990). Force-to-Motion Functions for Pointing. Proc. of Interact '90: The IFIP Conf. on Human-Computer Interaction. 701-706.

Sears, A. (1993). “Investigating touchscreen typing: the effect of keyboard size on typing speed.” Behaviour & Information Technology **12**(1): 17-22.

Smith, D. C., C. Irby, R. Kimball, W. Verplank & E. Harslem (1982). “Designing the Star User Interface.” Byte **7**(4 (April)): 242-282.

Yamada, H. (1980). “A historical study of typewriters and typing methods: from the position of planning Japanese parallels.” J. Information Processing **24**(4): 175-202.

Zhai, S., M. Hunter & B. A. Smith (2000). “The Metropolis Keyboard- An Exploration of Quantitative Techniques for Virtual Keyboard Design.” CHI Letters **2**(2): 119-128.