Chapter 6

Future Directions

6.1 Sources of Error in the Experiments

6.1.1 The Chording Glove

In the Chording Glove experiment, the finger sensors were changed after it became clear that they could not handle long term use. This calls into question some of the data gathered in the experiments which used the old finger sensors. Two subjects used the old sensors for all their sessions, while three subjects used them for either 1, 3, or 4 sessions, making 28% of the experiment performed using the old sensors. It is worthwhile to understand the effect this could have had on the results. The change in sensors primarily could have effected the chording speed, error rate, or tutorial length. We will address each of these issues in turn to determine the potential effects.

Figure 6.1: Comparison of the average chording speeds including and excluding the old finger sensors

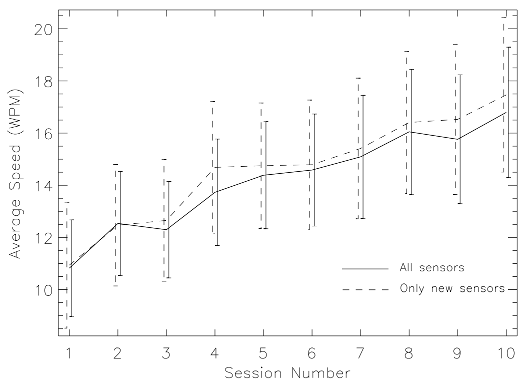

Figure 6.2: Comparison of the average percent error including and excluding the old finger sensors

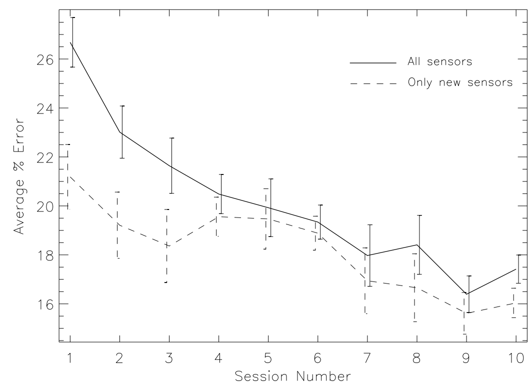

Figure 6.1 shows the average chording speed per session. The solid line is the speed for all the subjects. The dashed line excludes those subjects who used the old finger sensors. The plot shows that speed without the old sensors is consistently higher, although the difference between corresponding points is not statistically significant at the 5% level. The final input speed without the subjects who used the old sensors is 17.5 $\pm$ 3.0wpm. This is only slightly higher than the average calculated with all the subjects, 16.8wpm $\pm$ 2.5wpm.

Figure 6.2 shows the average percent error per session. Like speed, the error is consistently lower when the subjects using the old sensors are eliminated from the calculation. The difference between the cases is statistically significant at the 5% level for all except the middle 3 sessions. An interesting thing to note about these values is the rise in the error for these sessions. The rise comes in sessions 4 and 5, the same time at which two subjects switched over from the old sensors to the new sensors. If these subjects are removed from all the sessions, the rise disappears. This implies that switching between the two sensors requires a period of adjustment before the error rate drops to the lower value. The final error rate for all the subjects was 17.4 $\pm$ 0.6%. Excluding those using the old sensors gives a final error rate of 16.0 $\pm$ 0.6%.

The average time it took to perform the tutorial with the old finger sensors was 86 $\pm$ 32 minutes. The average time with the new sensors was 74 $\pm$ 14 minutes. Again, the new sensors performed better, but the difference is not statistically significant at the 5% level.

These results imply that if only the new sensors were used the Chording Glove's performance would have been higher, but not by very much. The error rate was affected most by the use of the old sensors. The difference in the final error rates is 1.4%. This is not unexpected, because the poor sensitivity of the old sensors served to increase the likelihood of mistakes. The plot of the overall performance of these subjects was the same shape as the other subjects, just worsened by the old sensors. Including these subjects does not significantly degrade most of the results, while using them gives more information on the effect of performance over time and other aspects of the Chording Glove not effected by the finger sensors, such as the learning rate.

The results of the portability tests tended to show no difference between the speed while standing or sitting, but these results have their limits. First, the lack of a mobile computer in the original experiments prevented any situations more mobile than standing up to use the glove. A continuous mobile environment is expected to provide much more demanding situations than this. The second limitation to the experiment was the lack of a control. Would the same experiment done with a QWERTY keyboard show no difference in performance as well? This needs to be looked into. The last problem is the low number of subjects used in the experiment. While there were enough to show that there was no significant difference between the speeds, a larger sample size is needed for the more complicated experiments necessary to prove the extent of the portability. The experiment served to show that the device is portable in one of the simplest portability tasks one might encounter in a mobile environment. If the Chording Glove failed this test it would imply that since it could not handle the easiest portability task, it would be unlikely to suffice for a more intensive task, and no further experimentation need be done to show this. Given the results of the experiment, we know that further testing is necessary and that this experiment needs to be redone with a control, more subjects, and in more demanding situations which require use of a mobile computer.

6.1.2 The Biofeedback Pointer

The first question that comes to mind regarding the Biofeedback Pointer tests is the low number of subjects. These tests were designed as a proof of concept to show that the Biofeedback Pointer works, and will work for more than one person. This only requires two subjects. Three were used to get a better idea of the variability of results. It is true that there were too few subjects to obtain statistically significant results, but the tests were only intended to determine the relative performance, not its feasibility as an input device. It is hoped that by using the results of this experiment, a new prototype can be built which performs closer to the desired level of a graphic input device. Later in this chapter we will discuss potential directions for such research.

Another potential problem regarding the subjects is that the creator of the device was used in the tests. The rest of the subjects used were near-beginners, having only used the Biofeedback Pointer once before. This would give a good idea of the starting performance range of the device. Since no one was more experienced or familiar with the device, the creator was used to give an idea of the expert level performance. This is justified by the fact that knowing the internal workings of the performance test experiment could not have effected the results in any way.

In this experiment, the width of the target (W) was calculated by taking the span of the button along the approach vector (see Section 5.4.2). The potential problem with this method is that the subject will not always move straight toward the target, but may approach via a curved path, which could place their true approach vector perpendicular to their estimated one. In the worst case, the estimated approach vector is perpendicular to the longest side, giving the smallest W of W_0 and the real approach vector passes through one of the corners, giving the largest W of $W_1 = W_0\sqrt{r^2+1}$. Where r is the aspect ratio. This makes the "real" index of difficulty:

$$ID_1 = \log_2 \frac{2A}{W_0\sqrt{r^2+1}} \tag{6.1}$$

where the calculated index of difficulty is $ID_0 = \log_2 \frac{2A}{W_0}$. Simplifying and calculating the potential change in ID gives:

$$ID_1 = ID_0 - {}^1\!/_2 \log_2(r^2+1) \tag{6.2}$$

$$ID_1 - ID_0 = \Delta ID = -{}^1\!/_2 \log_2(r^2+1) \tag{6.3}$$

Except for the "Done" button, the aspect ratio of the buttons varies from 1.0 to 1.39, which means that ΔID is at most -0.78, and therefore the ID used in the experiment will be 0.78 bits less, or, in other words, slightly less difficult. Odds are that the errors will be random and cancel out, but we will look at the worst case situations to see the maximum effects. If the all the lowest IDs are off by -0.78 and the highest are all correct, this would alter the range of ID from a width of about 6 bits to 6.8 bits. The opposite effect gives a width of around 5.2 bits, which is a 13% difference in either direction. Since MT is the same, this means that IP could be off by as much as 13% in either direction. The mean value of the normalised IP would then range from 0.125 to 0.163, which is only slightly larger than the standard deviation. In the worst case, this error would only increase the standard deviation by 60%, which is small enough for the results to remain valid.

Another potential problem with the experiment was that number of trials was not the same for all the subjects. Each subject performed at least four trials on the Biofeedback Pointer before they were given the option to leave. It was believed that four trials would be enough to generate a respectable number of points. Any number of trials beyond this should only improve the accuracy of the calculation and should not do any harm. Each of the subjects' data were analysed separately, meaning that the number of trials of one should have no effect on the results of the others.

A related issue is that there were fewer mouse trials than Biofeedback Pointer trials. The reason for this is that the IP of the Biofeedback Pointer was more important to calculate, so more time was spent performing trials on the Biofeedback Pointer. The smaller number of mouse trials might account for a slightly larger standard deviation in the mouse's IP. The average ratio of σ_{IP} to IP for the mouse is 0.108, while for the biofeedback pointer it is 0.100. However, this difference is unlikely to have any effect, since the experiment was only to give an idea of that values of IP could be expected.

The largest potential source of error in the Biofeedback Pointer experiment was the occasional trouble with the window manager during the experiment. The computer used was a Pentium 133MHz. The computationally intensive task of analysing the data was near the limit of the processor's abilities. The performance test application was run in the Windows 95 operating system. Data reading was done as a thread run by the OS, once every millisecond. Every 64 milliseconds, the data analysis was performed, repositioning the pointer. This thread had the highest priority. Lower on the priority level were window messages, such as button clicks. As a consequence, sometime the OS would get too overwhelmed by the calculations and other necessary maintenance that a button click would get delayed. The effect of this would be that the user would click on a button, and nothing would happen. Often times they would think they missed and try clicking again. These events would happen anywhere from zero to six times per trial, but usually occurred only once or twice. It was noted when these events occurred and the bad data were deleted from the calculations. It may be possible that a few short delays were unnoticed and the corresponding data were not deleted. This would have the effect of an unnecessarily high MT, which would very likely lower the IP. Beyond this problem, the most noticeable effect would be the confusion and annoyance of the subject, which may have effected their subsequent performance in that trial.

6.2 Future Work

6.2.1 The Chording Glove

Future Experiments

There are a few questions left unanswered from the Chording Glove experiment than can only be answered by further experimentation. The first is the time to learn the alphabet part of the keymap. The time to learn the entire set of 97 chords was found, but most chord keyboards tend to specify learning time in terms of just the alphabet, not all possible characters. Further experimentation must be done to determine this learning time to allow a fair comparison between the Chording Glove and other chord keyboards.

Another question the experiment left unsatisfactorily answered was the keymap retention. Only one subject was used to determine this. The exact change in her performance is in doubt due to the change of sensors between her last session and when she returned, but there is no doubt that one person's performance did not suffer after a three month absence. In order to sufficiently determine the long term effects of absence, this experiment must be repeated with a larger number of subjects.

The experiment did not use the full functionality of the Chording Glove in order to limit the scope of the experiment. This left several aspects of the Chording Glove untested. The only one of the function keys which was tested was <`Help`>. The rest of the function keys, and the use of <`Control`> were are untried. Are <`Help`>, <`Pause`>, <`Escape`>, <`AutoCaps Toggle`> and the arrow keys the only ones truly necessary? Would more be useful? How many function keys can conveniently fit on the hand? What about the possibility of using a “soft” function keypad made from a small touchscreen? The full range of possibilities has yet to be explored and needs to be researched.

Improving the Chording Glove

The area with the most vital need for future research on the Chording Glove is the finger sensors. It is clear that the current sensors are insufficient for commercial use. The experiment shows that the sensors need to be smaller than the current metal plate ones in order to allow more flexibility in hand-orientation while chording. The activation pressure needs to be less to reduce fatigue, and the accuracy must improve to allow faster chording with fewer errors. A soft sensor, like the foam ones described in Appendix C, would be ideal, since they could be made light and flexible, and possibly even sewn into the lining of the glove, causing minimal disturbances to real-world actions. Unfortunately, all the foam sensors tried so far were insufficient for this task. The most promising alternative for the sensors would be miniaturised switches or sensors based on capacitive technology like that used in touchpads.

The use of the function keys was briefly mentioned in Section 3.2.1, but not in any great detail. The use of <Help>, <Escape>, and the arrow keys translates easily from their use with normal keyboards and requires no further explanation. <AutoCaps Toggle> and <Pause> both act in somewhat different ways than one would expect from a normal keyboard and need further explanation.

The AutoCaps feature has been mentioned before as a method for automatically capitalising the next word when a period, question mark, or exclamation point are pressed. This is to allow the user to continuously type sentences without stopping to press the shift at the start of each one. The usefulness of such a feature is untested. It may improve performance, but may get in the way as well. The AutoCaps idea may be extended to the possibility of using a predictive keyboard style input. In this method, the most likely ending for a word is displayed on the monitor. This is updated with each letter entered. If the predicted word is incorrect, the user keeps entering more characters. If the word is correct, the user presses a trigger key, (for example, double-pressing <Space>) and the word is completed for them.

There needs to be a fast and easy way to turn off or ignore the Chording Glove's input to allow interaction with the real world. As it stands, pressing the <Pause> function key will cause the glove to ignore all the finger sensors until <Pause> is pressed again. This requires use of the other hand, which may be inconvenient. Another alternative is that the <Control>-S sequence could be used to stop input until either <Pause> is pressed or the <Control>-Q sequence is made. The <Control>-S, <Control>-Q pair is a good choice for two reasons. The first reason is that this sequence is currently in common use as the pause-unpause commands for many existing systems. The second is that <Control>-S can be performed very quickly. The <Control> sticky shift is first pressed by the thumb. Then the sensors on the thumb and middle finger are pressed together to make S. This can be quickly done, even without a surface to chord against. <Control>-Q is a good choice because Q is one of the harder to make chords and thus less likely to be made by accident.

One possible alternative to the suggested system is to place an entire handheld-style computer on the back of the hand, in the current location of the function keys. This would be used for graphic input and display while the Chording Glove would provide text input. The entire system would be contained on the glove, instead of distributed about the person like a wearable computer. The first question which arises is the performance of the non-preferred hand for controlling the graphic input. Non-preferred hands work as well as dominant hands when performing gross tasks such as scrolling, but leaves something to be desired for tasks which require detailed pointing (Kabbash et al., 1993). Another possible problem would be placing the display on the back of the chording hand. The hand's orientation while chording may partly occlude the display, or the movement involved in chording might make it hard to focus on the display.

6.2.2 The Biofeedback Pointer

Future Experiments

Selection Control In the experiment, selection was performed by using the buttons of a mouse with the left hand. This method was only used to test the use of the pointer and is inappropriate for general usage. The buttons need to be operated by the same hand as the pointer. It is intended that the shift buttons on the Chording Glove are to be used as selection buttons for the Biofeedback Pointer. These buttons are mounted on the index finger and pressed by the thumb. This leads to the important question: Will the action of pressing a button with the thumb be interpreted as pointer motion? This would create jitter when pressing a button, similar to the jitter problems of a touchpad. The muscles used to move the thumb tend to be either deep or located far from the electrodes sites. The flexor pollicis longus is the only muscle which might interfere (see Table 3.8), but it is likely to have no effect because it is occluded by the superficial muscles in the upper forearm in the area where the electrodes are positioned. Preliminary tests show that only strong actions of the thumb cause visible jitter when using the Biofeedback Pointer. Experiments must be done to show the likely levels of thumb use and to see if interference is a potential problem.

Alternative Muscle Groups It is possible to use other muscle groups to control the pointer without any changes in the hardware or software. If other factors weigh more than the limitations imposed by convenience as mentioned in Section 3.3.2, one can theoretically place the electrodes anywhere they can acquire a reliable signal.

The fingers, eyes, and head are particularly interesting sites to choose for pointer control. If the fingers are used, the electrodes can be placed in more or less the same location as for the wrist, since the muscles which move the fingers can be sensed from there. The only problem would be abduction and adduction of the fingers. These are difficult to measure since the muscles which control them are located in the hand. Using smaller electrodes might avoid this problem.

The Biofeedback Pointer could be used for eye tracking. According to Gips & Oliveri (1996), the voltage per degree of arc is around 20μV. The Biofeedback Pointer can measures voltages in multiples of 2.5μV, thus being able to discriminate down to 7.5 minutes of arc. The eye angle is a function of the difference between two of the electrode sensors. This is a linear combination of the electrodes which can be generated by the neural network. There is no reason why the Biofeedback Pointer would not be able to be used as an eye tracker.

The movements of the head and neck also can be easily measured. Like the wrist, the rest location in is the middle of the motion. The size and shape of the electrodes currently used may make them difficult to position over the necessary muscles, but smaller ones may be sufficient.

Any other body part can be used as long as the EMG is conveniently measurable and the user can consistently control it. It is theoretically possible to set up a system in which left and right motions are controlled by flexion of the left and right elbows respectively, and up and down motions are controlled by flexion of the left and right knees. This setup is easily measurable, but the user may find it hard to visualise, causing difficulties in training and using. However, there are no hardware or software problems preventing such a setup. Given sufficient time and need, the Biofeedback Pointer should be operable with just about any set of muscles.

Improving the Biofeedback Pointer

Larger motions generated by the Biofeedback Pointer are scaled up by a factor of 3. This is an arbitrary scaling which was chosen for its ease of calculation and success in empirical testing. However, this is not the only option. The piecewise function used (Equation 3.9) is first order continuous, but not second. The scaling factor jumps from 1 to 3 for $r \geq cs$. It may be better to use a second order continuous piecewise function (splines) or even a non-piecewise function like r^2 or a higher degree function. This must be further investigated in order to improve the efficiency of the Biofeedback Pointer.

Another important factor in the performance of the Biofeedback Pointer is the training. Currently training is done by moving the wrist through each of the basic directions in the theory that if the neural network can handle those, it can handle any input given it. One problem is the reaction time in following the cursor. This is compensated for by a lag component in the calculations, but the lag may not be consistent throughout the training period. Also the user's motions may not be as precise as necessary. For example, the program expects the motion to be a diagonal of 45°, but the user might move at only 15°. A solution for these problems must be found. One solution is to perform more than one set of motions. This might compensate for the inconsistencies, but may require a fairly long time to perform. Another solution is to put a motion sensor on the hand to get the exact orientation of the wrist. This is similar to the approach tried by Hiraiwa et al. (1993). It would be preferable to avoid this method because it requires extra software and expensive hardware. A simpler method might involve using audio cues to help synchronise the screen motion with the subject. What we need from an improved set of training data is a relatively short training process and to require no external hardware beyond the current system. In addition the training process needs to use a set of motions which are easier to follow with less lag, and less possibility of deviation.

Improving the recognition of the neural network is essential in improving the accuracy of the Biofeedback Pointer. What we would like is to find a way for the neural network to be improved after the initial training session. At the moment the network can only be retrained from scratch. The tests implied that averaging different sets of weights would not yield a better set. This was confirmed directly in an informal test where the average of multiple sets of weights was found to be completely unusable.

It is possible that another kind of linear combination, or even a non-linear one may work in combining the weights. Another solution is to find a less computationally expensive back-propagation neural network. The BPN mentioned in Section 3.3.4 took up most of the computing power of the computer, required too much training, and gave insufficient accuracy. It may be possible to use another type of BPN, or even a genetic algorithm-based system, which would be fast and accurate enough to use.

Another option is to change the electrode setup. Since the ECU was found to be used least of all the muscles, it may be possible to find a better position for the electrode, which will weigh more in the neural network. Alternatively it may be useful to remove the fourth amplifier completely and just use three in the calculations, reducing the amount of work for the computer. It may also be possible to move the electrode to another site (e.g. the thumb) and use it to recognise gestures for selection. This would provide a system using biofeedback for pointer control and selection.

One advantage of using the current set of muscles is that they are located in a circle around the arm, just distal to the elbow. In this arrangement, the connectors can be placed on an arm band, which can be easily affixed to the proper location, facilitating measurements. In addition this provides a much tidier package, since there would be only one bundled wire containing the four signals coming from the arm band, as opposed to four separate wires from various locations on the arm. Ultimately the hardware from the main box will be merged with the Analogue/Digital converter and put on a board or PCMCIA card. Moving the FFT from the software to specialised hardware would free up a great deal of memory and computational power. This would yield a very small device consisting of an arm band which plugs into the computer. This could be worn underneath clothing, allowing a very discrete method for interacting with the computer.

6.2.3 General Input Device Research

Combining the Biofeedback Pointer and Chording Glove

The Chording Glove and Biofeedback Pointer are intended to make a unified graphic and text input system for a wearable computer. There is one major unanswered question we need to solve before this can be done: How can we switch between graphic and text input? The finger motions used in chording will doubtlessly be interpreted by the Biofeedback Pointer as motion, at least through the extensor digitorum, if not other muscles. This is not so much of a problem, since the nature of the devices precludes their simultaneous use. The Biofeedback Pointer requires the wrist to be free to move, while the Chording Glove needs to orient the wrist next to a solid surface. The problem is how to switch between the two quickly and easily. There are a few options for this. The first option is to use a function key on the back of the hand. This is undesirable because it would sometimes require constant use of the other hand, eliminating the one-handed benefits. The second option is to use a special chord key like `<Control>-A` or double-`<Space>`. The former has the benefit of being able to be made without a typing surface, but it may not be comfortable or fast enough to perform as often as might be needed. Double-`<Space>` is likely fast and easy enough, but it may be desired for use by another action like predictive text entry. It also has the disadvantage of needing a surface to make the chord.

Another option would be to use a gesture. It is possible to have a quick and easy to recognise gesture be interpreted to mean “switch input devices”. One such gesture could be to quickly flex and extend the fingers (or just one finger). Since this uses the extensor digitorum and flexor carpi ulnaris, it is easy to detect with the existing setup. However, gesture recognition software must be added. Experimentation must be done to determine if this is a viable option.

A final alternative is to use the Chording Glove on one hand and the Biofeedback Pointer on the other. The Chording Glove should be used with the dominant hand to enable maximum performance for text input. The non-dominant hand should be used to control the Biofeedback Pointer. Kabbash et al. (1993) determined that the non-dominant hand is just as effective as the dominant one for low-precision tasks. Since the Biofeedback Pointer is ideal for those sort of tasks, it should be just as usable in the non-dominant hand. The Biofeedback Pointer also has the advantage that the hardware is hand-independent, unlike the Chording Glove. This means that if use by one hand is ineffectual, it can be easily switched to the other one.

In situations where both hands are expected to be free to use the PC, there may be increased performance from being able to enter text and control a pointer at the same time. As mentioned in Section 3.2.4, evidence exists showing the benefits of performing tasks in parallel. Not all tasks can be made more efficient by performing simultaneously. The most efficient tasks are those in the non-dominant had supports or alternates tasks with the dominant hand. This does not work well in tasks where the hands perform the exact same tasks, or completely different tasks (Buxton, 1995). While it is possible on a desktop computer to use a mouse and keyboard simultaneously, this is done very rarely in practice. This could be interpreted in two ways. First that there is little practical need to perform text and graphic tasks simultaneously. The second interpretation is that people tend not to have the cognitive ability to perform text and graphic tasks at the same time. In either case, it is unlikely there would be much benefit from devoting one hand to each task. Then again, if the other hand would be idle the entire time, nothing would be lost from separating the devices. In the best case, performance could be improved. In the worst case, the problem of switching between text and graphic input would be solved.

The experiments used to measure the performances of the Biofeedback Pointer and Chording Glove made the implicit assumption that computer input would be the primary task and that real-world interaction would be secondary or non-existent. It is likely that the performance of the devices would be different if they were used for secondary tasks, with the primary task involving a real-world interaction. Siegal & Bauer (1997) describe an experiment in which a wearable computer was used to assist with the primary task of aircraft maintenance. The input device was a board-mounted dial, which was worn on a belt and operated with one hand. It was found that a one-handed system was more efficient, since certain tasks required the users to lean over objects, bracing themselves with one hand while operating the computer with the other. The body-mounted nature of the device did have disadvantages since leaning against the plane could accidentally activate the device. This demonstrates the advantage of an arm-based system such as the ones described here, which cannot be accidently activated in such a manner. In addition this also illustrates the need to easily deactivate and reactivate the input devices, possibly using the method described above.

Experimentation needs to be done to determine the performance of a wearable computer using the Biofeedback Pointer and Chording Glove. This not only needs to measure the performance of the input devices (input speed, error rate, IP, etc.), but the performance of the primary real-world task as well. This experiment should be able to determine if there is any degradation in the performance of either task, and to what extent it occurs. Furthermore, the experiment should also provide some insight into how the devices can be changed to be more efficient.

Terminal Typing Speed

For some tasks, text entry speed is the most important performance factor for the text input device. If a text input device is used solely for entering previously composed text or data, with little or no interaction, the time to complete the job is dependent only on how fast the user types. This is what the typewriter was designed for. This used to be the most common use for a text input device. Most modern text input tasks require interaction, such as reading email or searching for a file. A user only enters a minimal amount of text, followed by a great majority of time spent viewing the output. Speed is no longer the most important factor in text entry.

Take for example, the one of the most text-entry intensive task on a computer: composing a document on a word processor. In this, the author spends time considering what to write. This generally means text entry in short bursts, with significant pauses in between. This gives an effective speed limit, which could be far below the maximum speed for the device. This speed limit is the point where the ability to enter text faster will not be any more productive.

Assume a decent sized paragraph has around 100 words. If the entire paragraph is conceived of at once, a 60wpm typist could enter it in 100 seconds, while a 30wpm typist would take 200 seconds. Is that 100 seconds significant? How long will it take to conceive the next paragraph? What about error correction or reconsidering the words while typing? If the fast typist takes six minutes to write the paragraph, will the slower one take six minutes as well, or seven and a half? These are all important considerations. A speed of 60wpm is more difficult to maintain that 30wpm. Is there anything to be gained with such speeds in casual use? Ultimately, is the “terminal velocity” low enough to be significant (20–40wpm) or is it excessively high (50wpm+). This is a potentially important factor determining the appropriate application for a text input device.

Other Uses for the Biofeedback Pointer

Currently the Biofeedback Pointer has only been used for control of a 2D pointer in cartesian coordinates. It is possible to adapt the software to use another coordinate system. For example, one can use supination and pronation of the wrist instead of adduction and abduction. This lends itself quite well to polar coordinates. This would not be an ideal input for a windows-type system, partly because it singles out the origin as a special point, and partly because the wrist cannot move into the full $360°$ circle it would need to reach all the points on the screen. This may however be a useful interface for a navigation system which required turning left or right, and acceleration or deceleration. Because the muscles are in such close proximity, the electrodes can be left in the same position as they are for adduction and abduction. This would allow the user to be able to switch between the two interaction methods in software. There is precedent for this. Some existing applications use the mouse in non-standard ways, such as for navigation in 3D environments, so this is a reasonable option.

In Section 6.2.2 we suggested that the Biofeedback Pointer can be used on the neck muscles to control a pointer. It may be possible to further adapt the system to yield the orientation of the head to provide a head tracking system for virtual reality. Most head tracking systems work on broadcast technology, limiting the range to a specific room, or even a small part of a room. Using the EMG to determine the head’s orientation removes the range constraints.

Another potential adaptation of the Biofeedback Pointer would be to use it for gesture recognition. A simple neural network, similar to the one currently used, could be trained to recognise a small number of gestures to perform various actions. In Section 6.2.2 we suggested the possibility of using gestures for selection instead of a hardware button. One possible gesture would be a quick flick of the thumb. This could provide the equivalent of a single or double click. In Section 6.2.2 we also suggested a gesture-based method for switching between text and graphic input.

As mentioned above, the coordinate system can be switched in software. There is no reason that the Biofeedback Pointer cannot be switched as needed from graphic to gesture input as well. In certain situations a set of arrow and function keys is faster than pointer input (Greenstein & Arnaut, 1988). In environments where interaction is constrained to a few responses, it may simplify matters to use a few easily recognised gestures. For example, in filling out a checklist, one might just need gestures for “yes”, “no”, “forward”, and “back”. Alternatively, gestures can be recognised as application specific macros or function keys.

Another use of the Biofeedback Pointer could be to replace a DataGlove-style virtual reality interface. Placing a 6-degree of freedom motion tracker on the arm would yield its position and orientation. The Biofeedback Pointer can then be trained to yield the orientation of the fingers and wrist, in a manner similar to Hiraiwa et al. (1993). This would give a hand-based VR interaction tool, without needing to wear a DataGlove or hold a 3D mouse.

Most of these options can be done by changes in software, not changes in electrode positioning or hardware. So the same device can be used as a 2d pointer, gesture recognition system, and (with the addition of a motion tracker) a VR interface.

6.3 Summary

There were a several potential problems with the implementation of both experiments. An analysis of the errors in the Chording Glove experiment showed that the experimentally determined performance is likely to be worse than the true performance. Thus the results are, in fact, a lower bound for the Chording Glove’s performance. The performance test carried out on the Biofeedback Pointer was intended to give a range for the expected performance of the device. The possible problems with the tests were insufficient to cause a significant difference in the results. Consequently, like the Chording Glove, the errors could only serve to lower the performance, thus the Biofeedback Pointer’s results also provide a lower bound for the true performance. Future experimentation and research on these devices needs to be done. The Chording Glove needs further research on keymap learning rates and retention, and the use of function keys. Research on the Biofeedback Pointer needs to be done on the selection buttons and the use of alternative muscle groups.

Several improvements for both devices have been suggested in this chapter. The Chording Glove would benefit from improved finger sensors, specialised software interaction, and possible “soft” function keys. The performance of the Biofeedback Pointer could be improved by exploring possible alternatives to the current motion scalings, training methods, and neural networks. Specialised hardware could be used to decrease the size, improve the speed, and lessen the drain on computer resources.

In this chapter we have suggested areas for input device research for the Chording Glove, Biofeedback Pointer and a wearable system using the two. From this we have made a few recommendations. The Chording Glove and the Biofeedback Pointer are designed to work together, but there are potential problems with switching between text and graphic input, and possible jitter from selection that need to be looked into. Research should be done to find out the true requirements of input speed for casual text entry to determine what upper limits we need from text input devices. The Biofeedback Pointer hardware and software is very flexible and it is worthwhile investigating its potential use for 3D interaction and gesture recognition.