Thumb Based Interaction Techniques for Input on a Steering Wheel

Ivan E. Gonzalez

School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213

This research was made possible by the guidance of
Brad A. Myers, and Jacob O. Wobbrock,

Abstract:

As more systems that require user input are integrated into motor vehicles or carried in by the driver, it is necessary to investigate new interaction techniques that are well suited for in-vehicle interaction. The systems today include navigation/information systems, MP3 players, satellite radios, and cellular phones, PDAs, and more will certainly be available in the future or carried into the vehicle by the user. Current interaction techniques in automobiles mostly use buttons, custom knobs or on-screen keyboards, which require the user's visual attention. Handheld devices also require looking at their screens. A proposed alternative is voice recognition; but it has numerous drawbacks, including low accuracy in general, susceptibility to environmental noise, vulnerability to user differences, and a tendency to convey greater intelligence than warranted. In order for in-vehicle user input to be safe and feasible, it is necessary to investigate interaction techniques that allow the driver to focus on the road by minimizing required visual attention and movement away from the steering wheel.

This investigation presents a hardware prototype of a steering wheel with a touch sensitive area that the user's thumb can interact with while gripping the wheel. Methods for interacting and entering text using a thumb on a touch sensitive area of a steering wheel were developed and evaluated. Furthermore this investigation reveals that some dominant interaction methods such as a two dimensional and one dimensional soft keyboards are slower than some effective long list scrolling techniques such as rate controlled clutching and dialing. The impact of simulated driving while performing selection task has on driving performance and selection time is also presented.

Introduction:

The automotive experience has been augmented by many devices in current years and the list of devices that will be integrated into the driving experience or carried in by the user will increase every year. Currently some of the devices that could be found in an automotive setting include: navigation/information systems, MP3 players, satellite radios, and cellular phones, PDAs. Some of the solutions that have been proposed in the past in order to solve the problem of interaction in an automotive setting include: on-screen keyboards, custom knobs, remote controls, all of which require the users visual attention. Voice Recognition has been posed as an alternative which does not require visual attention but has many fallbacks that have not made it a promising solution to this problem including:

- Speech interfaces are the “holy grail” of user interfaces
- Unreliable in practice since its susceptible to environmental noises and vulnerable to user differences.
- It conveys greater intelligence than warranted
- Requires the user to remember specific language syntax.
- Processing intensive and requires expensive hardware

- Poor for continuous control task such as:
    - Scrolling though a long list
    - Dragging a map to a new location.

The main interaction tasks that are users need in an automotive setting are text entry and selection task. Gestural text entry techniques provide a potential solution that is eyes free. The proposed solution for this problem is EdgeWrite.

EdgeWrite is a new unistroke text entry method that has been implemented for handheld devices, joysticks, pointing devices and wheelchairs. Gesture recognition in EdgeWrite is accomplished trough the sequence of corners that are hit in a square region. EdgeWrite's properties of: high tactility, compactness, physical stability, high accuracy, and technological simplicity, pose an efficient and economic solution to text entry in automobiles.

- High tactility: EdgeWrite can be made with raised edges or even just "Braille bumps" that can be felt by a user's finger. High tactility can alleviate the need for visual attention.
- Compactness: EdgeWrite input areas can be made very small, fitting easily on the open surfaces of a steering wheel for use with the user's thumb or forefinger.
- Physical stability: EdgeWrite is not particularly susceptible to vibration because the finger of the hand that is gripping the steering wheel could be held steady while moving over a small EdgeWrite square.
- High accuracy: EdgeWrite's three previously-mentioned properties result in high accuracy because users can "feel" their writing, they can do it in a small space with littlemotion, and they can overcome vibration.
- Technological simplicity: EdgeWrite can work with as few as four binary sensors. It does not require a full digitizing surface or a stylus to trace smooth high-resolution paths.

Simulated picture of EdgeWrire on a steering wheel based on an un-posed photo

Figure 1

The other main task that users need supported in an automotive setting is selection task. One possible way to complete selection task is thought text entry, but it is also possible through techniques that allow continuous control of a system pointer. Continuous control techniques such as clutching and dialing have been successfully implemented in other settings such as MP3 players. This work explored the different hardware iterations that a device for input on a steering wheel can go through and subsequently the potential interaction techniques that might be supported by such device.

Physical Prototypes:

Four Buttons

The technical requirements for the physical prototype were dictated by the needs to implement EdgeWrite on a steering wheel. In order to have EdgeWrite on a steering wheel it was necessary for the prototype to have four sensing regions for each of the corners that are used to create EdgeWrite characters. The four EdgeWrite sensing regions do not have to be fully digitizing surfaces but instead could be of a binary nature. The simplest possible physical interface to EdgeWrite consists of four binary sensors that represent each of the EdgeWrite corners.

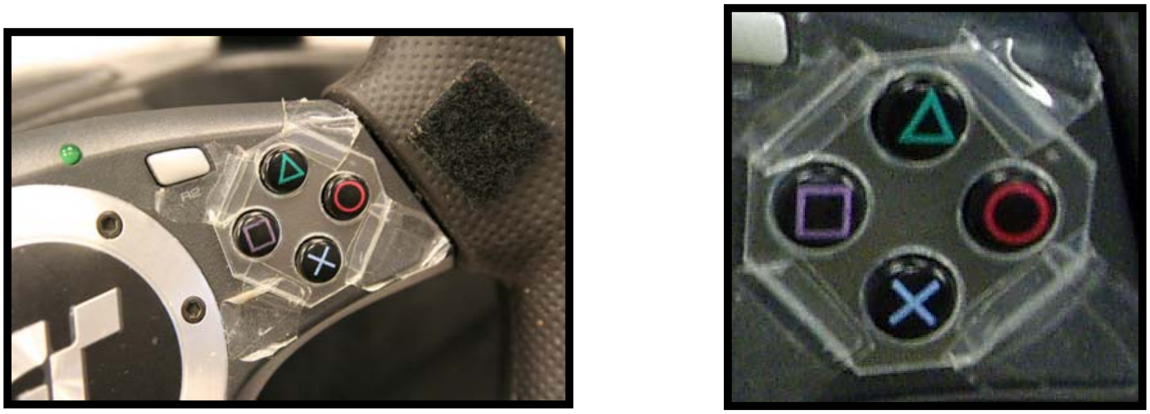


Figure 2

The initial explorations of prototypes intended to leverage the four binary sensor feature of EdgeWrite by implementing a version of EdgeWrite using four built in buttons of a Logitech Driving Force Pro Steering Wheel see Figure 2 above . The buttons on the steering wheel pertruded above the surface of the steering wheel. Since the user would need to move from button to button in order to create an EdgeWrite character it was essential that the transitions be smooth. In order to overcome this physical challenge, a plastic template was placed on top of the surface of the buttons in order to fill the gap that exists between the buttons. The plastic template was created by measuring the buttons with a caliper and cutting a plastic thick enough to cover the height of the buttons using a high precision laser cutter. This allowed the users to more smoothly transition from button to button in order to create the EdgeWrite characters.

Initial uses of this prototype proved to be difficult. The buttons required the user to push down upon them in order to activate them and have them be recognized as a corner. This proved difficult for users since the users would expect the corner to be recognized once they enter it.

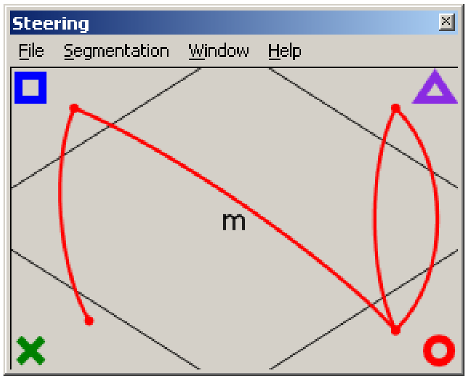


Figure 3

The software implementation for this prototype consisted of a program with an interface with the Direct X Direct Input libraries that facilitate the development applications that use input devices such as joysticks see Figure 2. The custom application was integrated with the EdgeWrite recognizer. In each of the EdgeWrite corners there is a symbol that matches the symbols found in each of the four buttons, this helped users understand the mapping of the physical corners found in the steering wheel to the ones in the recognizer, since the physical corners where at an angle.

Some of the contributions of this version of EdgeWrite that were eventually integrated into other version of EdgeWrite include:

- Time out algorithm that allows the segmentation of letter strokes based on the average time that it takes to make one segment of the stroke.
- Arc drawer that supported the revisiting of corner sequences by incrementing the width of the arcs depending on the number of arcs already present between any two corners.
- Addition of sound cues by assigning unique tones to each corner. Since each EdgeWrite character is a unique sequence of corners, if each corner has a unique tone, this results in each letter having a unique sequence of tones associated with it.

The main lessons learned from this prototype were that users with thick fingers could potentially activate more than one button at a time. Also users with long finger nails were unable to user the prototype since it was difficult to smoothly transition from corner to corner in this prototype.

The D Pad (Directional Pad)

Figure 4

The Logitech Driving Force Pro Steering Wheel also had a D-Pad (Directional Pad) see Figure 4 above . A D-pad is plus signed shaped control used for directional control. The directional pad has four buttons for the four cardinal directions. The D-pad was thought to be a better candidate for EdgeWrite corner entry since there was no gap to be filled between the buttons. Once it was implemented it proved to be difficult to control since the D-pad centers itself after every press of a button, which makes entering diagonal EdgeWrite strokes difficult. In a later design the mechanism for centering was removed in order to see if that would make the D-pad a better suited for EdgeWrite but it remained very difficult to use. The software implementation of this prototype was very similar to the one of the four buttons.

The Touchpad

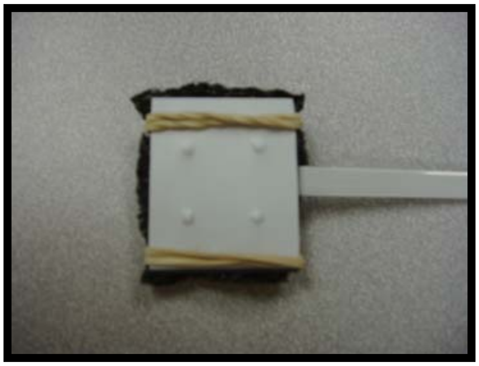


Figure 5

Participants in a user study that compared a touchpad and a joystick as input hardware for EdgeWrite felt that a touchpad was easiest to use, easiest to learn, fastest, most accurate, most enjoyable, most comfortable, and most liked overall [1] An implementation of EdgeWrite on a touchpad mounted on the Logitech steering wheel seemed like a viable option that would be well received by users. In order for a touchpad version of EdgeWrite to be successfully mounted onto the steering wheel, it needed to be small. Synaptics a touchpad manufacturer was developing a small form factor touchpad called a StampPad and donated a prototype to the Carnegie Mellon University Human Computer Interaction Institute (Figure 5). The touchpad required a stable and safe mounting onto the steering wheel in order to allow the user to seamlessly interact with it.

Figure 6

Figure 7

The initial idea to mount the touchpad was to cover its back side with Velcro and have it attach to the steering wheel with Velcro. The initial use of this prototype proved to be difficult since the touchpad was not sturdily mounted and also the touchpad's cable would get in the way of the users steering see Figure 7.

One of the biggest challenges of working with the touchpad was the fact that the small touchpad was a prototype obtained from the manufacturer. The challenge was two fold since everything that required the used of the small touchpad required an additional level of certainty since it would be very hard to replace it. Furthermore since it was a prototype the electronics in the back of the touchpad were exposed, therefore mounting it on the steering wheel required additional caution as not to damage the electronics. The back and front of the touchpad with the exposed electronics can be seen in Figure 8 bellow.

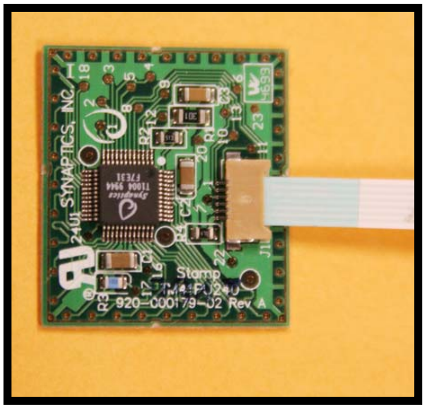


Figure 8

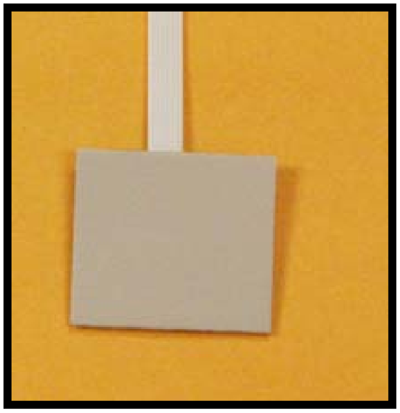


Figure 9

In order to overcome the sturdiness problem that the touchpad presented, it was necessary to build a chasse that would hold the touchpad in place and facilitate its mounting onto the steering wheel see Figure 10. The chase was designed to fit the touchpad and also had two layers of depth in order to protect the electronic components that were exposed on the bottom of the touchpad. The chase of the touchpad was then attached onto the steering wheel using Velcro. This second iteration of the touchpad prototype overcame the sturdiness problem. The cable of the touchpad being in the way of the users steering still remained a problem.

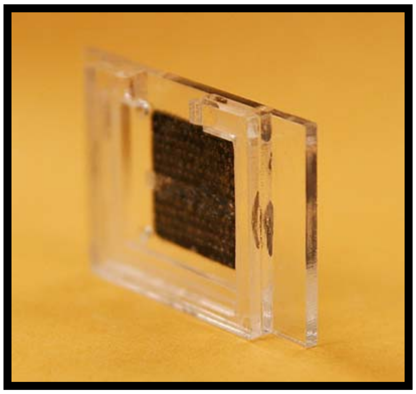


Figure 10

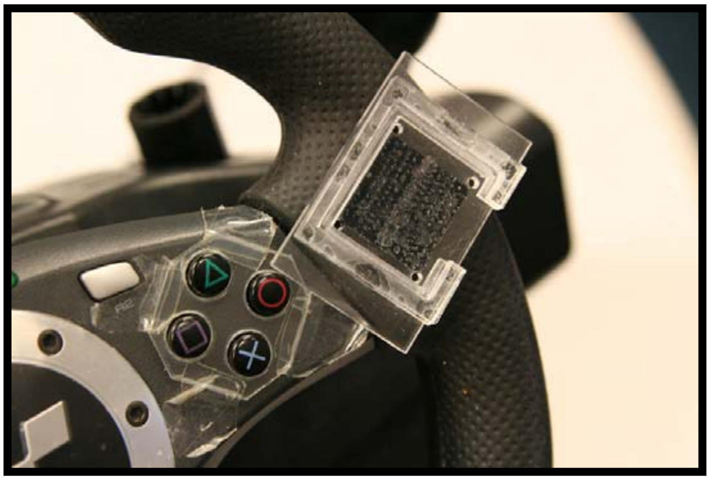


Figure 11

In order to provide corners to the users a sheet of styrene with a bump in every corner, in order to cue the user that they have entered a corner. Styrene was used in order not to interfere with the conductive surface of the touchpad see Figure 12 .

Figure 12

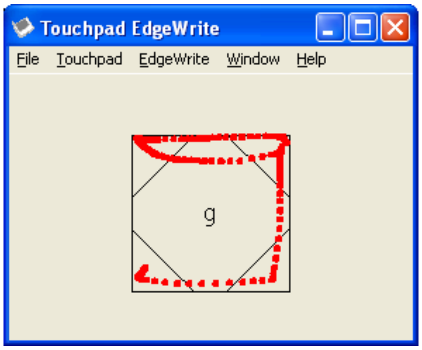


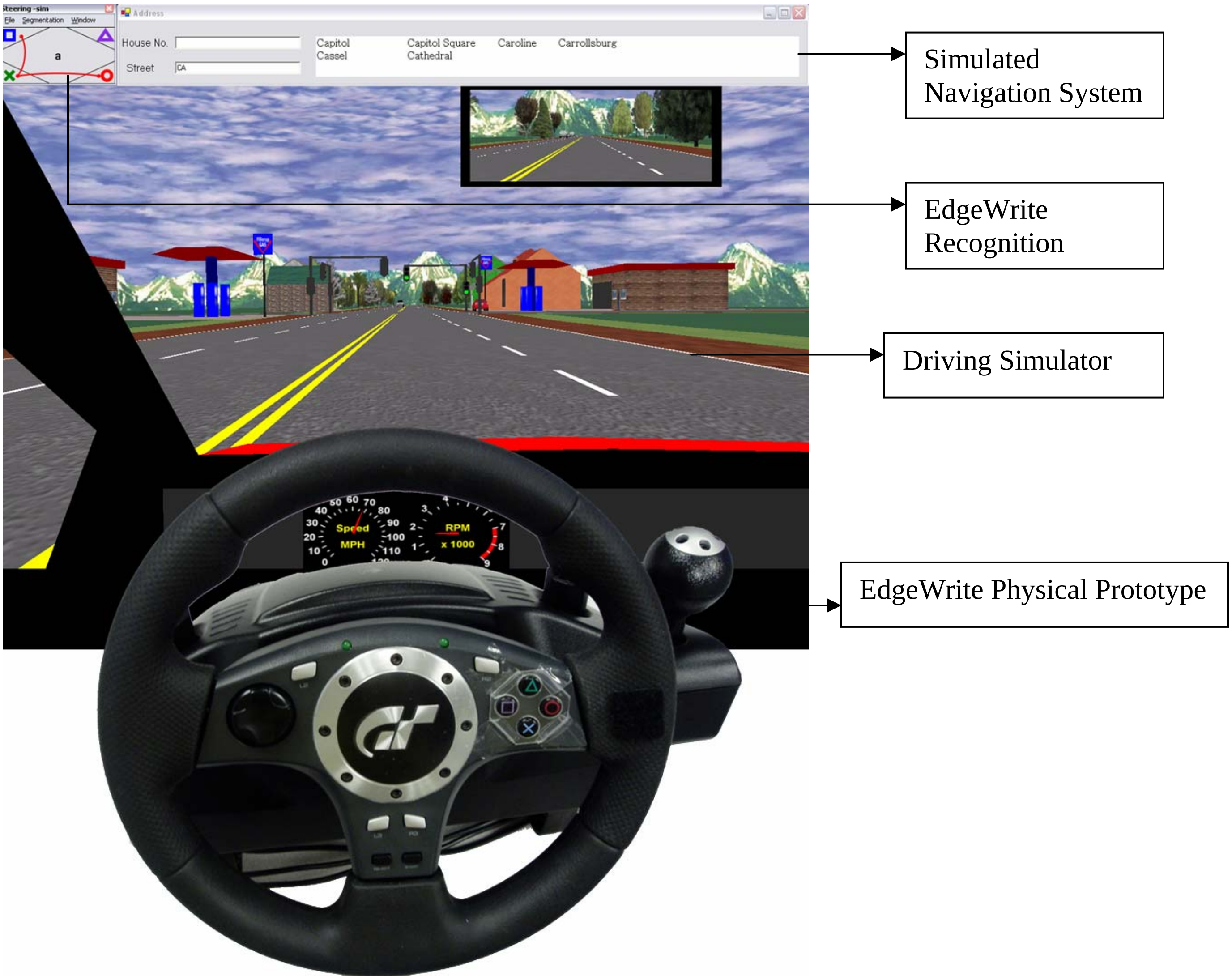
Figure 13

The software implementation for the touchpad was already completed and therefore it was not necessary to implement a new version of the software for this hardware prototype.

Initial uses of this prototype reveled that users preferred the feeling of sliding or traveling from corner to corner without having to explicitly signal through the push of a button that they are in a specific corner.

All the prototypes in the initial exploration were evaluated through pilot usage using a driving simulator and a simulated navigation system with predictive text completion like the one shown bellow Figure 14.

Figure 14



After the initial hardware prototypes were completed it was concluded that it was necessary to implement a prototype that had the best features of the touchpad and the four button prototype. The four button prototype was easy to use because it was flushed with the surface of the steering, with the drawback that transitioning from corner to corner was deemed difficult and not smooth. In term of the touchpad, it was much easier to travel from corner to corner but it was awkwardly mounted onto the steering wheel and had a connector that was in the way of the users steering.

Another general observation that came out of testing the prototype with the driving simulator was that the buttons and the touchpad were placed in such a way that the user would not be able to hold the wheel properly while interacting with either the four button or touchpad version. This raised a major redesigned in the prototype since it was necessary to have the user perform the primary task of driving in the most natural way possible.

In order to overcome the short comings of the initial prototypes it was necessary to leverage the good features of each of the design and put them together in a new design. The new design would need to be flushed onto the surface of the steering wheel and also it should be easy and smooth to transition from corner to corner. Furthermore the prototype should seem like a product that has been fully integrated without dangling cables that might distract the user or illicit a feeling that they are interacting with a device that is not well developed. Another design goal of the new prototype would be to have the sensing area placed in such a way that it does not make user grip the steering wheel in a way that is not natural.

In order to complete this stage of development we inquired a local design firm with the specifications that came out of our initial explorations. Our main criteria were high tactility, compactness and physical stability. The design firm we contacted estimated that completing the hardware prototype would cost an amount outside of the research budget.

In order to prototype a new steering wheel it was necessary to find a new steering wheel that could be easily modified. The steering wheel that was found to be most suitable for modification was the Logitech Driving Force for the PlayStation 2 Figure 15. This wheel was unique compared to the other ones available in the marker because the buttons rested on the grip of the wheel at 10 o’clock and 2 o’clock. This steering wheel seemed like a possible match for modification since it was possible to fully disassemble it. The main challenges in producing a prototype with this steering wheel would be making a wheel that was initially intended for the PlayStation platform work with a PC and modifying the plastic regions at the 10 o’clock and 2 o’clock regions of the wheel in order to integrate the touchpad.

Figure 15

The steering wheel was fully disassembled and the plastic region that had the four buttons was identified as the optimal place to mount the touchpad Figure 15. The four button region was picked because it would allow for the testing of the prototype with right-handed participants, which are a larger percentage of the population. It is also worth noting that a similar device could be implemented for left handed users if the touchpad was mounted on the opposite side of the steering wheel. The other reason why the four button region was selected for integrating the touchpad was because it would allow users to interact with the touchpad seamlessly as they naturally held the steering wheel.

It was recognized that such prototype would allow the user to interact with the touchpad with his or her thumb placed on top of it. In order to provide tactile feedback a square plastic template Figure 16 was placed on top of the touchpad which allow for guidance of the users gestures and also would facilitate the transition from corner to corner while using EdgeWrite. Another iteration of the square region for tactile feedback which is not pictured consisted of the use of PDA screen protectors cut out in as squares and placed on top of the touchpad sensing region. The material of such touchpad screen covers is conductive; therefore it would not interfere with the touchpad. In practice this proved not to be a viable option since it did not give users strong enough of a tactile cue in order to know that they had reached the end of the square region. The screen cover square region resulted in users running their fingers off the surface of the touchpad, which was suboptimal.

Figure 16

The prototype required about two months worth of work since it needed to be carefully assembled and mounted onto the steering wheel. One of the main inhibitors for progress in the development of the prototype was the fact that the touchpad was a one of a kind prototype, therefore all mounting and gluing onto the steering wheel had to be completed with the up most degree of certainty. Another challenge that had to be overcome was the fact that the steering wheel was initially intended for PlayStation use and not PC use. The main advantages of the final prototype are the seamless integration into the surface of the steering wheel and also the placement of the touchpad which allows for natural interaction. Furthermore compared to the previous prototyped all the electronics of the touchpad are integrated inside the steering wheel. A quick progression of the building of the prototype can be found in Figure 16.

Once the final hardware prototype was completed it was necessary to explore new possible interactions that were feasible with the new device. Since the main interaction tasks in automobiles are a combination of text entry and selection task, it would only be natural to explore how this prototype could enable novel selection and text entry task in an automotive setting.

Interaction methods

Interaction methods can be classified according to two criteria: visualization and interaction technique. Visualization refers to how the system provides feedback to the user of their selection. Interaction technique refers to how the user interacts with the system. Some possible selection techniques that are possible with a touchpad are described in the table bellow:

	Visualization		
Interaction Technique	Linear Menus	Linear Keyboard	Selection Keyboard
Clutch	✓	✓	✓
Small Displacement*	✓*	✓*	✓*
Big Displacement	✓	✓	✓
Dialing	✓	✓	✘ (not possible)
EdgeWrite	✓	†	†

Figure 17

* Eventually not developed or tested

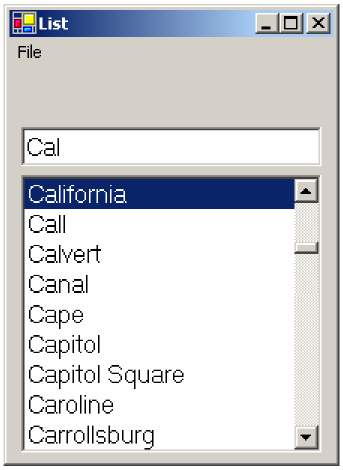
† EdgeWrite is a text entry method and can not be used with other text entry methods.

Visualization Techniques:

Visualization techniques allow the user to see how their interaction with the system has changed the system and also to judge if they have reached their end goal. The visualization techniques that were evaluated are the following:

Linear Menus:

Figure 18



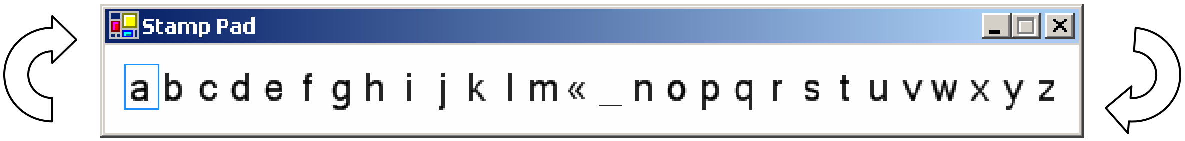
Cal

California
Call
Calvert
Canal
Cape
Capitol
Capitol Square
Caroline
Carrollsburg

The linear menu visualization allows the user to view and select from all the possible selections in the system. One of the drawbacks of the linear menu is the need for taxonomy in order to be a viable option for many items. It is possible for the linear menu to have multiple levels, but the one currently explored in this system consisted of a single level menu. The linear menu that was implemented in this system included a prefix matching feature. The prefix matching of the linear menu would allow the user to type in a prefix and once the match was unique the system would complete the user's selection. In the case where the prefix was not unique the top most item selected in the list would be the first alphabetical match to the prefix without having to further specify their selection. The user was also able to interact with the linear menu directly by scrolling up or down the list until they found their selection.

Linear Keyboard:

Initial implementation



Final implementation

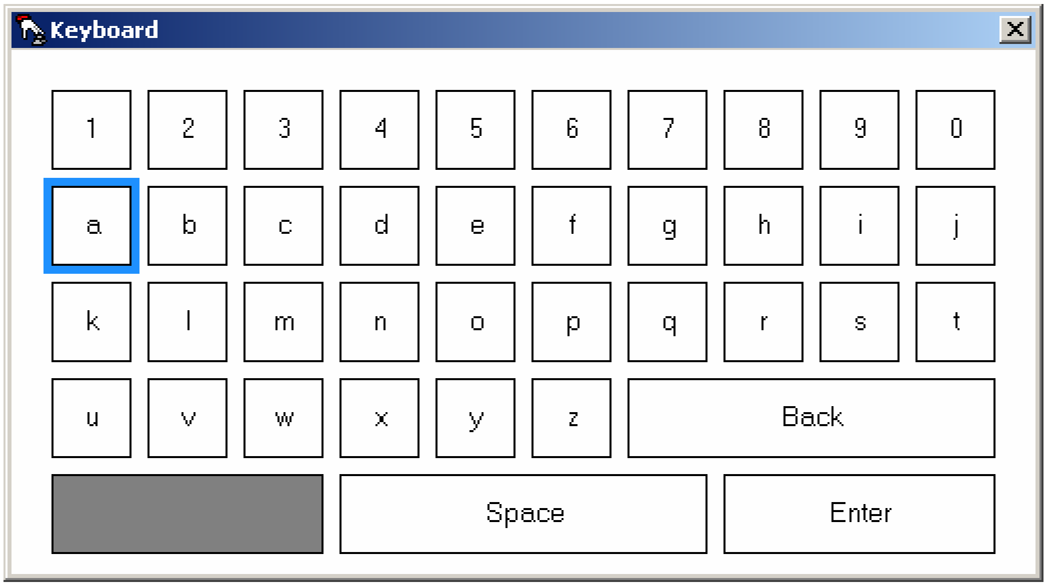
Figure 19

The linear keyboard also know as a Stamp Pad consist of a linear lay out of all the roman characters, numbers and some special keys such as enter, space and backspace. The interaction with the linear keyboard is one dimensional. The user moves in one direction back and forth in order to move the light blue halo that represents the current selection. Once the user commits the character it is sent to the designated input region.

The layout in Figure 19, is the result of a couple of modifications to some of the most common implementations of the linear keyboard. Some linear keyboard implementations such as the one in Figure 19 [top] place the space character in the middle since it account for 18% of written English Characters [2]. In the traditional implementation of the linear keyboard the light blue selection halo snaps back to space after every commit and the user can also wrap around from one edge of the keyboard to the other. These features were dropped in the implementation tested in this system, because pilot user observations noted that most street names did not include spaces and also the snapping back to space was found to be confusing for some users especially when the wrong character had been previously committed. The wrapping around the edges was removed in order to allow users to benefit from the infinite target feature, or allowing the user to overshoot the desired character at either extreme and have the selection always result in the last character in the direction that the user overshot. This is intended to decrease the amount of time required to do target acquisition at the extremes of the linear keyboard. The backspace "<<" was placed at the left extreme since it would allow users to revert previously committed characters. The enter "↵" character was placed at the right end of the linear keyboard in order to reduce the target acquisition time for the commit command. The space "_" character was placed a location to the left of the enter character since considering the domain of selection of street names space does not occur with high frequency, and when used it usually results in a unique selection therefore, the user can easily slide over to the enter character after a space has been committed and has uniquely identified the users selection.

Selection Keyboard :

Figure 20



The selection keyboard mimics the physical keyboard. Selection keyboards require two dimensional movements, in order to be able to move between the rows and columns. The light blue selection halo allows the user to see what their current selection is. The selection keyboard used in this implementation was based out of the keyboard found in the Xbox, except for the enter key. The layout of selection keyboards can be either QWERTY or alphabetic, in this case alphabetic was chosen, since it's the layout found in most in car navigation systems. The initial design allowed the user to wrap around the edges, but after initial explorations it was found that users felt they had better control if the halo did not wrap around the edges.

Interaction Techniques

Clutching

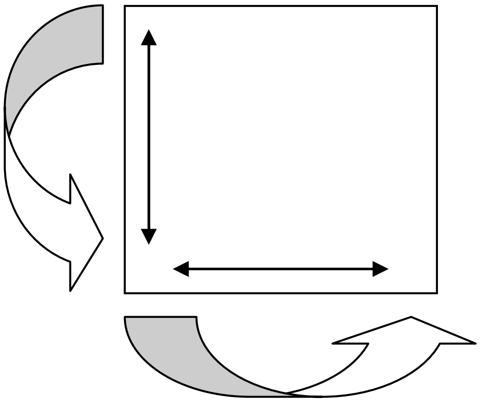
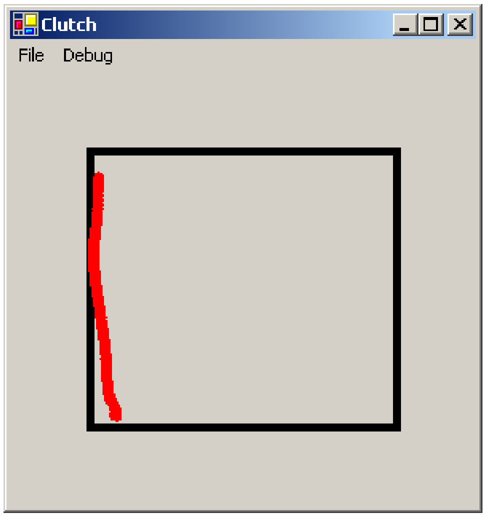


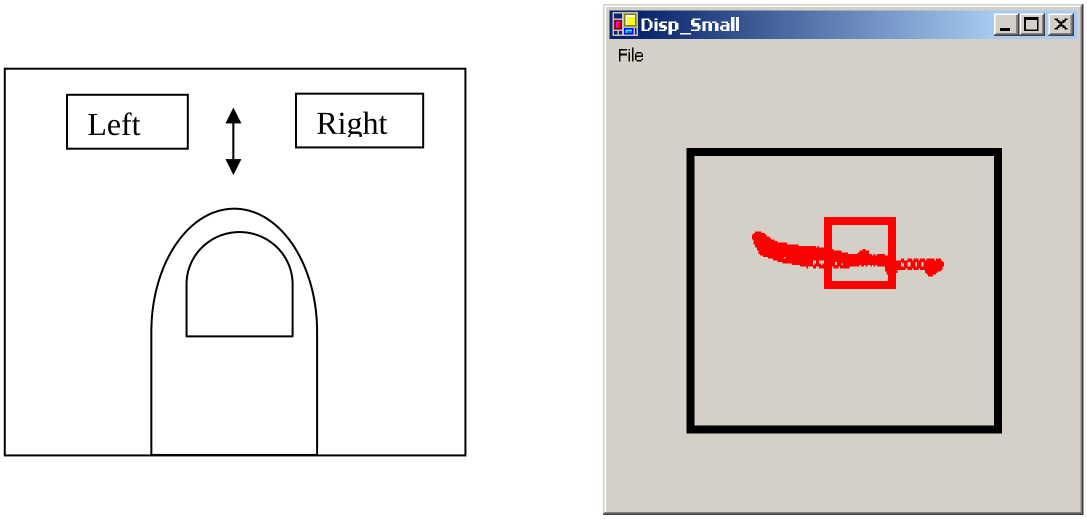
Figure 21



The clutching technique allows for two dimensional movements. The user can move in either the horizontal or the vertical axis. Once the user motion reaches the end of the touchpad input region they are required to lift their finger and place it in the direction they intend to continue their motion. The name of this interaction technique comes from the need to lift ones finger at the edge of the touchpad and reposition it. The implementation of the clutching interaction technique in this system has a transfer function that allows the user to determine the speed of the system pointer based on the speed with which they move through the sensing area of the touchpad. This function was implemented in order to facilitate the browsing through long list.

Small Displacement:

Figure 22



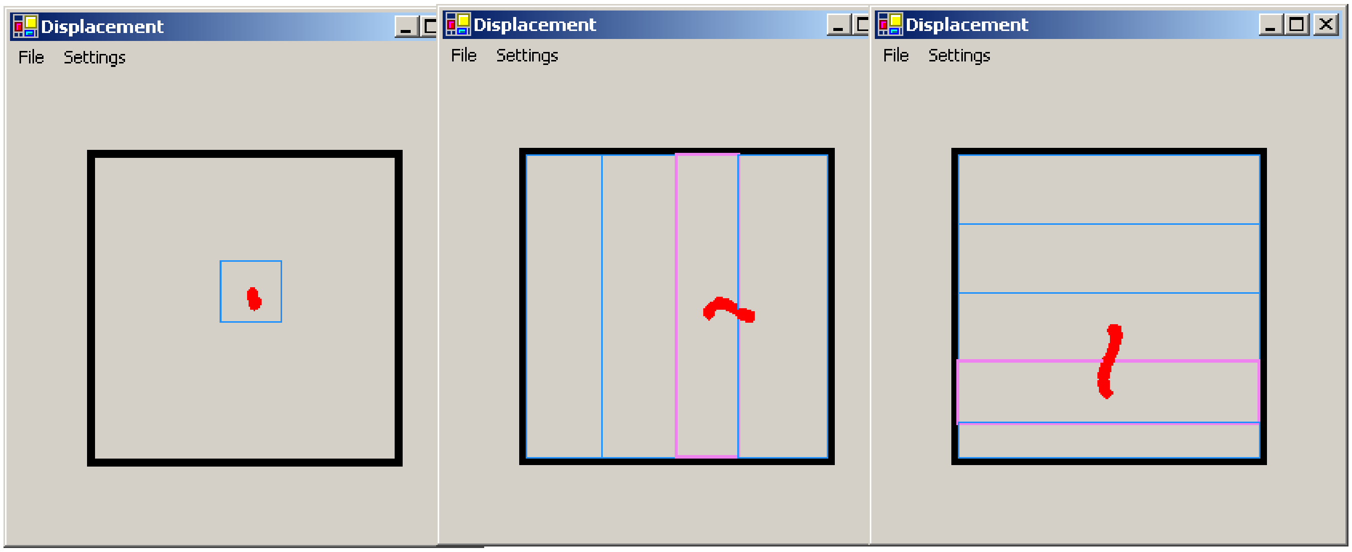
The small displacement interaction technique was intended to require small movements from the center location that the thumb is placed down upon the touchpad. This technique would allow the user to move in two directions by rocking their finger up and down or side to side after placing it on the touchpad. In practice this technique proved very difficult to successfully implement since user have different finger sizes, therefore the size of the neutral region was difficult to determine. The small displacement technique was also difficult to implement since it is virtually impossible for users to place their finger back in the absolute position they had placed their thumb on initially. After a couple of initial attempts to implement this technique it was found to be unreliable for interaction since the users intention was not always properly recognized by the system.

Big Displacement (*Displacement*)

Figure

Figure 23

A B C



In the displacement technique when the user first touches down on the touchpad a bounding region is determined as seen in Figure 23.A, the square regions size is determined as a percentage of the total input area . The direction in which the user exits the bounding region determines which way the system pointer moves. Once the user exist the bounding region the system pointer starts moving in the direction in which the user exited the region, with the possibility of moving in the opposite direction if the user starts moving in the opposite direction. The rate at which the pointer moves in the desired direction is determined by how far the user is from the farthest region in the desired direction. The pink region is initially determined as the region of the bounding box, once the user changes direction the pink region represents the last region that was exited.

The user changes direction by crossing the pink region, which results in the system pointer moving in the opposite direction and also at the slowest speed. If the user desires to change the direction of the pointer to one that is not opposite to the one the system pointer is currently moving in, it is necessary to lift their finger followed by touching down on the touchpad and exiting the initial bounding region in the desired new direction.

The regions in the displacement technique are determined to be of the same width as the initial bounding region if side to side movement is made and of the same height as the initial bounding region if up down movement is made. In the case where there is not enough space for a full sized region, the closest region is stretched in order to cover the space that would have been occupied by the region that did not fit within the square.

One of the drawbacks of the displacement technique is the need to clutch if the user touches down on the touchpad in a region that does not allow a large range of movement in the desired direction. Although the displacement technique is intended to minimize clutching in practice it requires some amount of clutching if the user does not put down their finger in a place that allows them a good range of motion in their desired direction.

Initial Displacement design

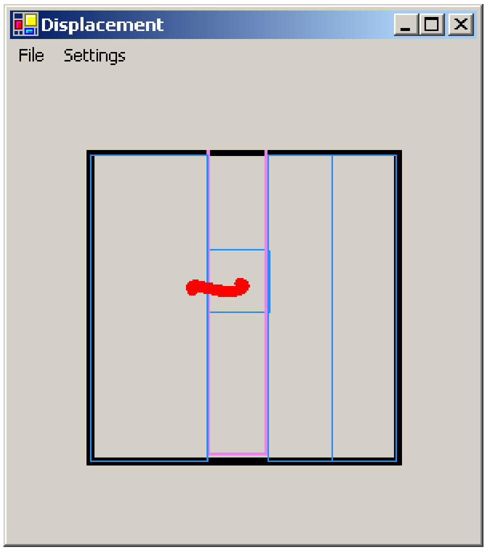


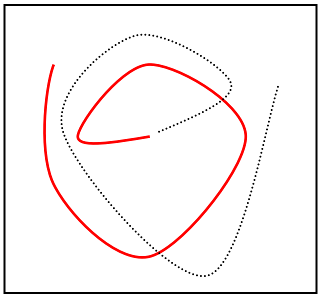
Figure 24

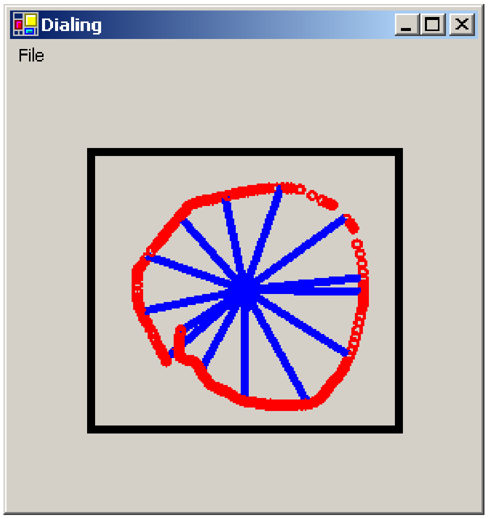
The initial design of the displacement technique had the same bounding region features and rate control features that are available in the final implementation. One of the main differences in the initial implementation of displacement was in the way in which the pink region was defined. In the initial version of displacement the pink region was always static and it represented the original bounding region. When the user entered the pink region, the system pointer would stop. This proved to be inefficient since it required the user to move their finger to a specific region of the touchpad in order to stop the pointer, which they could simply do by just lifting their finger from the touchpad. The removal of this feature brought two main advantages, it allowed the use of another region for rate control, and it also made the user travel less distance in order to reverse the direction in which the system pointer moves, since the user would no longer have to cross the inactive region in order to have the system pointer move in the opposite direction.

The initial implementation of the displacement technique by the nature of having a fix pink region, resulted in a static rate and direction associated with each of the regions. The regions in the initial implementation of the displacement technique also had constant sizes based on percentages of the input region which resulted in regions that were roughly not of equal size. Although the final implementation of displacement can result in unequal regions, if the user initially places their finger in roughly the center of the touchpad, the resulting regions are of equal size.

Dialing

Figure 25





The dialing interaction technique has been recently popularized by the IPod as a successful technique to select from long list. In this technique the user is able to control the speed and movement of the system pointer by the direction and angular speed in which they dial. The system pointer moves to the right when the user dials in a clock-wise direction and to the left when the user dials in a counter-clockwise direction. The angular speed with which the user scrolls in either direction determines the speed with which the system pointer moves in the desired direction as determined by a transfer function. One of the drawbacks of the dialing interaction technique is its one dimensional nature. Dialing can not be used to interact with a selection keyboard since the keyboard requires two dimensional control.

EdgeWrite

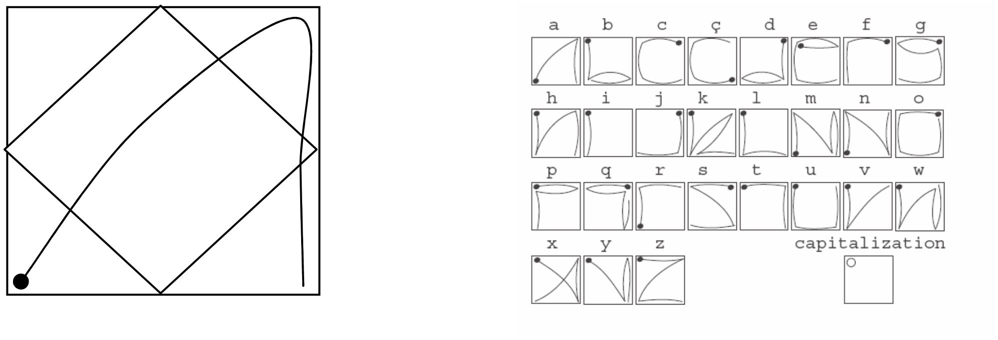


Figure 26

EdgeWrite is a unistroke text entry method that has been previously implemented for a stylus on a touch screen, a finger on a touch pad, a joystick on a game controller and wheelchair, a mouse, trackball and TrackPoint connected to a regular computer. For a touchpad, gesture recognition in EdgeWrite is accomplished by looking at the sequence of corners that are hit on the touchpad. Gestures are committed by lifting the finger and result in a keystroke being sent to the destination input region.

Visualization and Interaction Technique Integration:

The interaction techniques presented previously can be used in companion with the visualization techniques in order to select trough a list by means of text entry. The interaction techniques presented can also be used in order to directly select from a list. It is worth noting that selection can be made by entering text with prefix matching and word completion, or by direct selection through means of scrolling through a list.

Experimental Setup

In order to evaluate the interaction and visualization techniques it was necessary to run a user study that would evaluate how effective each of the interaction techniques was in companion with the possible visualization techniques. Since the initial motivation of this work was to be implemented in automotive settings, the study should also measure performance of some of the input methods while simulated driving.

Participants

There were four participants, all of which had previously been trained on EdgeWrite and were considered expert users. All the participants recruited were right handed, since our initial prototype was intended to be tested by right handed users. The users were all male, aged [20, 23], familiar with the use of a touchpad. Three out of the four participants were either state or internationally licensed drivers. None of the participants had used or own a car navigation system.

Apparatus:

Figure 27

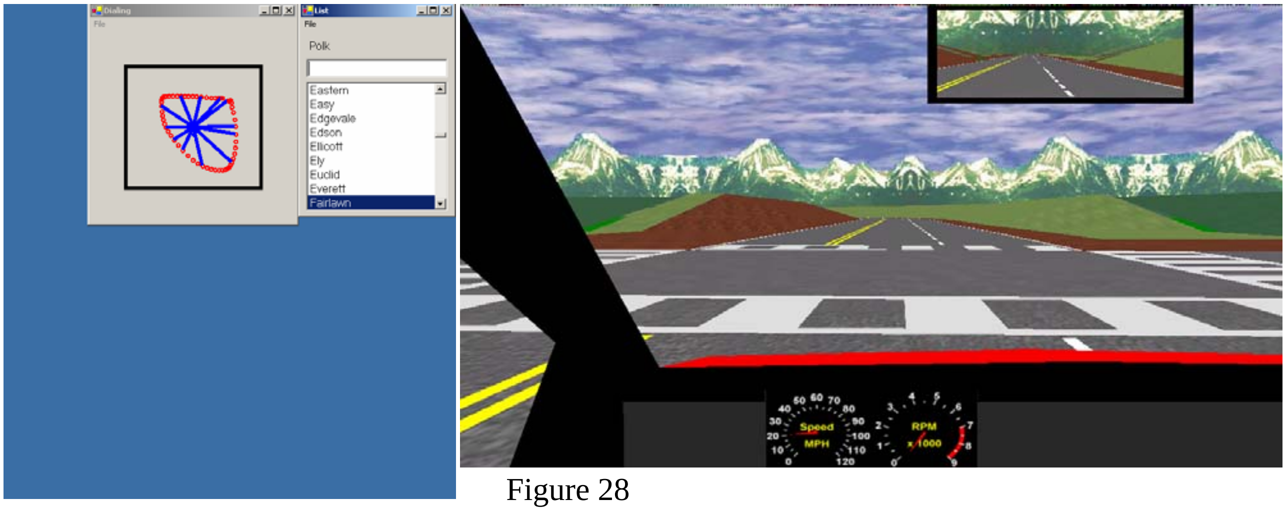
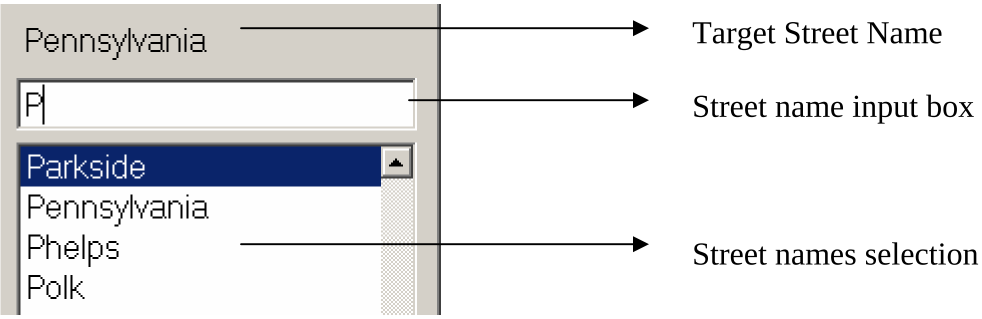


Figure 28

The apparatus for the experiment consisted of hardware and software. The hardware prototype mentioned in Section 1 was used for testing clamped onto a table, in companion with the pedal set that came with the original steering wheel. In order to test the user's selection time it was necessary to present the user with a street name to select from a list of streets. Depending on the technique that the user was being tested for, they would either directly select from the list or enter the street name text into a text box. The target street name was always presented the same way regardless of the condition.



Once the user finished all entering all the task the text where the target street name would usually appear would turn red and display the text "!!Done!!". The timing of the task begins after the target screen name has been displayed and the user has put their thumb down on the touchpad. The timer ends when the user commits the selection.

The driving simulator is the STISIM Drive high end driving simulator that records driving performance data such as:

- Road Edge excursions.
- Off road accident
- Centerline crossing
- Speed exceedance

The user has two screens in front of them, the one on the left is used for displaying selection based information such as interaction technique and visualization. The screen on the right is used in the simulator as it can be seen in the above Figure.

Design

In order to balance for the effects of learning it was decided that the best way to test the combination of interaction techniques and visualization techniques was to use a four by four balanced latin square [3].

In the first part of the study the user will be selecting using a combination of techniques as determined by the latin square. Once the user is done with all the conditions in the latin square, the input method with the least selection time will be tested under driving conditions and compared to EdgeWrite. EdgeWrite is always tested in the second part of the study regardless of the user's performance in EdgeWrite in the first part of the study since it is the only gestural text entry method that is being tested.

Procedure

The list of words the user were given to select from were chosen from a list of 228 streets in the Washington DC area. The street names were chosen based on word length, prefix length, distance from the beginning of the list, number of items with the same first letter, midpoint item, and last item.

The participants were first presented with the interaction technique and visualization technique being used for each condition. This was followed by a demonstration by the experimenter of two successful selections from the training set using the same interaction method that the user was being tested with. The user would then be allowed to train on five selections using the current condition. During the practice session for each condition the experimenter intervened when necessary in order to advise the participant how to use the system or clarify any questions that the participant had. Upon completion of the initial five practice selections, the user was informed that they would be measured based on their speed for the following ten selections and that they would not be given any clarification or could ask for help once they started the completion of the ten measured task.

In the second part of the study, the user would be informed of the speed limit in the track that they would be driving and they would be presented to the functions of the pedals and steering wheel. Users were allowed to first drive a track in the driving simulator without any secondary task. Once they completed the driving control condition they would be allowed to practice five selections with either EdgeWrite or the selection method with which they had the least selection time from the first study. The order of the conditions EdgeWrite or the best from the first part of the study were alternated between users. Once the user completed the driving simulated course for the five practice selections, they would be informed that their selection time and driving accuracy would be measured for the next driving trial for selection time over ten selections and driving performance.

Upon completion of the second part of the study the participants were give a short questionnaire asking question in relation to their experience with the system.

Measurements

In the first part of the study selection time was the only measurement, based on the premise that the time to select is a measure that takes into account the accuracy of the user while completing the task, since inaccurate selections would result in greater correct selection time. In the second part of the study selection time will also be analyzed in combination with driving performance as measured by the total percentage of driving time that the user: crosses the centerline, touches the edge of the road, exceeds speed.

Results

Experiment Part 1:

We first ran an interaction method order test and found that there was no order effects ($F_{8,8}$= 1.01, n.s.) , indicating that there was adequate counterbalancing. The average time to select in seconds and the standard deviation for each of the interaction methods is presented in Figure 29

Interaction Method	Selection Time(ms)	Selection Time StdDev
Dialing List	6830.0185	2064.327
Clutching List	7028.6282	2287.565
Displacement List	10097.946	3345.547
EdgeWrite	12692.51	7982.619
Clutching Linear Keyboard	12864.46	5003.697
Dialing Linear Keyboard	13023.938	5517.539
Clutching Selection Keyboard	14076.147	6758.112
Displacement Linear Keyboard	16098.845	8405.113
Displacement Selection Keyboard	26900.307	18410.63

Figure 29

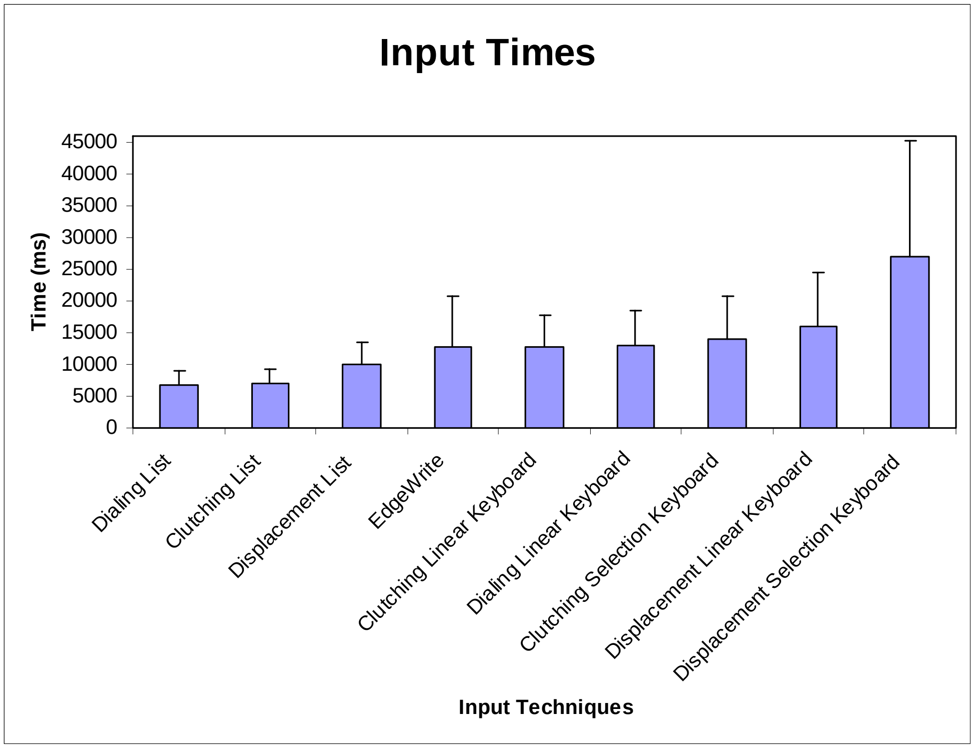


Figure 30

A main effect for selection time is significant for interaction methods ($F_{8,8}$= 8.35,p<.005)

Selection Time:

Interaction Techniques:
Selection time was used as the dependent variable for the first part of the study. The averages and standard deviations of the interaction methods are presented in Figure 30. The fastest technique on average was dialing with a list, and the slowest was displacement with a selections keyboard. Contrast of the interaction techniques show that dialing was significantly faster than displacement ($F_{1,24}$= 16.97,p<0.01) and clutching was also significantly faster than displacement ($F_{1,24}$= 14.27,p<0.01). Although the dialing interaction (9927 ms) technique was in average faster than clutching (11323 ms) there was no detectable difference between clutching and dialing ($F_{1,24}$=0.55, p=0.47).

Visualization techniques:
In terms of visualization techniques, the list based interaction methods dominated. The list based interaction methods (2300 ms) were faster than the linear keyboard based methods (3900 ms) with significance ($F_{1,24}$=12.68,p< 0.01) and were also faster than selection keyboard based methods (4300 ms) with significance ($F_{1,24}$=43.90, p< 0.01). Furthermore the linear keyboard methods were faster than the selection based methods ($F_{1,24}$=11.84,p< 0.01).

EdgeWrite:

Although EdgeWrite was not significantly faster than the other text entry interaction methods, on average it was the fastest technique among them. The trend as it can be seen in Figure 30 was in EdgeWrite's favor. EdgeWrite was faster than the slowest text entry interaction method of selection keyboard with displacement with a significance ($F_{1,24}$=23.62, p<0.01). Furthermore EdgeWrite was not significantly different from the fastest interaction method overall of dialing with a list ($F_{1,24}$=4.02, n.s.).

The dialing with a list was the method on average with the fastest selection time. Dialing with a list was not significantly different than the other list based interaction methods (displacement with a list, and clutching with a list), but it was found to be significantly faster than clutching with a linear keyboard ($F_{1,24}$=4.26, p<0.05) and clutching with a selection keyboard ($F_{1,24}$=6.14, p<0.05).

Within the selection keyboard and linear keyboard interaction methods, the clutching interaction technique was the fastest on average but was not significantly faster than the other interaction techniques except for displacement with a selection keyboard ($F_{1,24}$=19.25, p<0.01). Also the selection keyboards performed on average worse than the linear keyboard when using the same interaction technique.

Experiment Part 2:

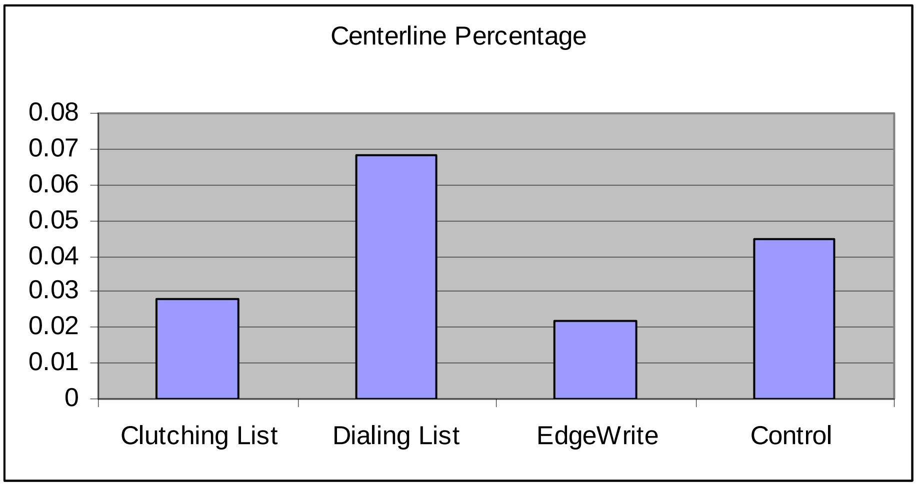
The dependent measures in the second part of the study were:

- Percentage of total drive time spent over the centerline (*centerline percentage*)
- Percentage of total dive time spent along edge of the road (*edge percentage*)
- Percentage of total drive time spent speeding (*speeding percentage*)
- Crashes per minute
- Task completed per minute

For none of the dependent measures does the order of the input method yield a significant main effect, indicating that adequate counter balancing in the second part of the study.

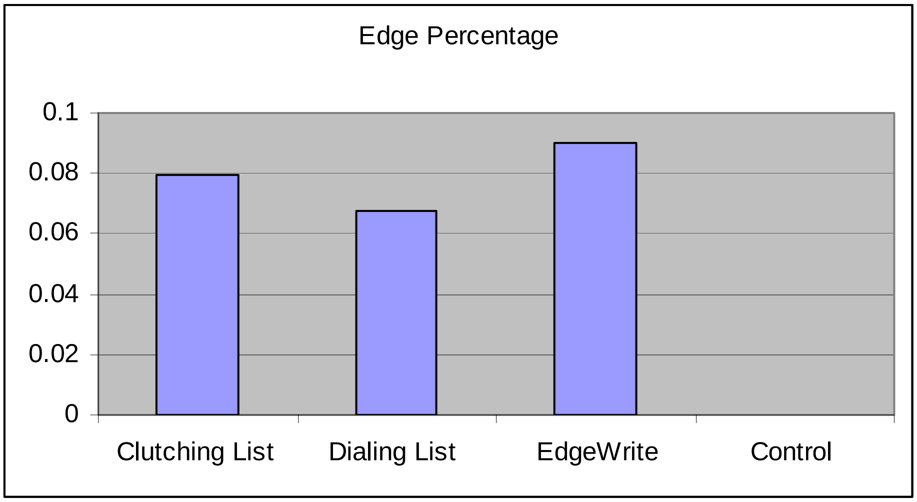
The main effect of input method on centerline percentage is not significant ($F_{3,5.11}$=1.47, n.s.). EdgeWrite has the least centerline average out of all the condition. It is interesting to note that dialing with a list on average is worst in terms of centerline percentage, which is interesting considering how in average it was the fastest in the first part of the study. The centerline percentage averages can be are graphed in Figure 31:

Figure 31



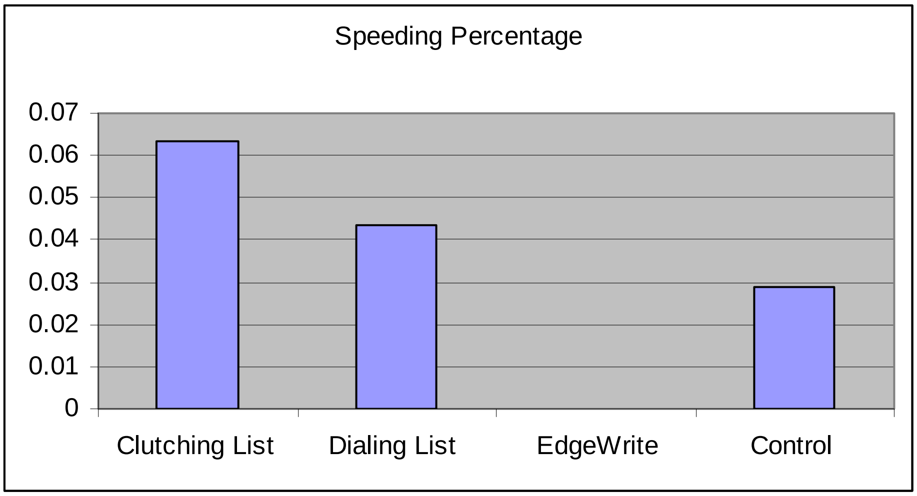
The main effect of input technique on edge percentage is not significant ($F_{3,5.64}$=1.21, n.s.) . None of the subjects touched the edge of the road in the control condition of driving without selecting, in the other conditions the participants touched the edge of the road approximately 8% of the total driving time. The edge percentage averages are graphed in Figure 32.

Figure 32



The main effect of input on speeding percentage is not significant ($F_{3,5.49}$=2.09, n.s.). An interesting observation to make is that at no time did any of the subjects speed while using EdgeWrite. This was not true for any of the other conditions, including the control condition of driving without selecting. The speeding percentages are graphed in Figure 33.

Figure 33



The main effect of input method on crashes per minute is not significant ($F_{3,6.02}$=1.42, n.s.). Although it is worth noting that the average among the three input methods was in favor of EdgeWrite. No crashes occurred in the control condition of driving without selecting. Means are show in Figure 34.

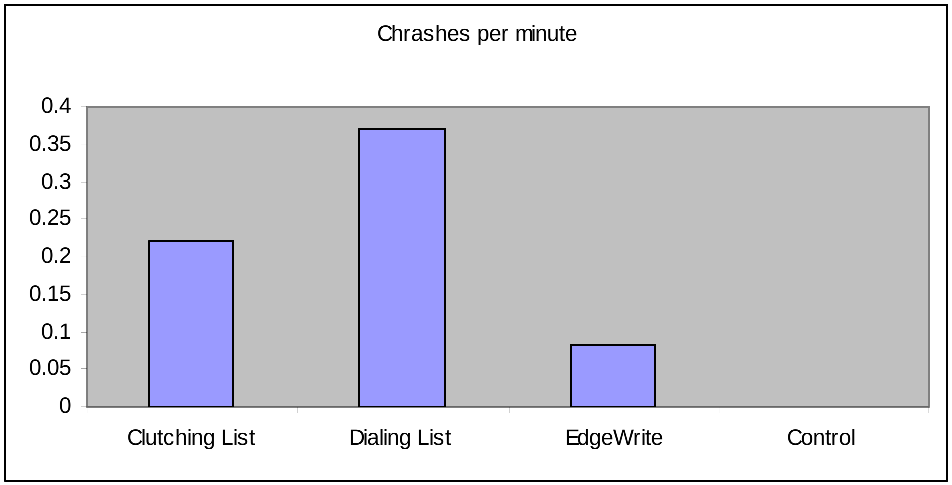


Figure 34

The main effect of input method on task completed per minute is significant ($F_{2,2.10}$=8.04, p=.10.). Looking closer we can see that the clutching list and dialing list which were nearly identical in the non-driving case, performed much differently once the participants were driving. Clutching with a list was found to be faster than dialing with a list with a trend that is nearly significant ($F_{1,2.24}$=13.60, p=.056.). EdgeWrite was found not to be significantly slower than clutching with a list, but significantly faster than dialing with a list ($F_{1,2.09}$=16.93, p<0.05.).

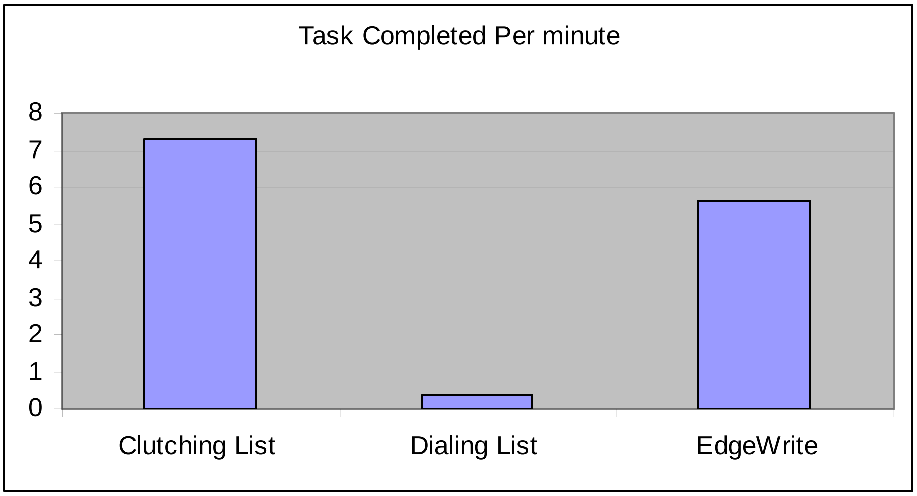


Figure 35

The second part of the study lacked enough power in the driving study in order to obtain statistical significance for most of the measures. A larger set of participants would be needed in order to draw statically significant conclusions.

Conclusions

The results from the first part of the study revealed that the list visualization technique dominated in terms of selection time. This is an interesting finding since most of the navigation systems that are in place in today use some form of keyboard. It is also worth noting that from the date obtained in this study it would be recommended that if a keyboard is to be used for text entry on average a linear keyboard with either clutching or dialing as an interaction technique would be faster for selection speed than a selection keyboard.

It is also worth noting how EdgeWrite could be a feasible solution to doing text entry in an automotive setting whether stationary or while driving. In the stationary condition EdgeWrite proved on average to be faster than the other text entry techniques. EdgeWrite was even more encouraging in the driving condition since it resulted in less centerline percentage crossings, crashes per minute. Furthermore EdgeWrite resulted in no speeding percentage.

The interaction method that resulted in the fastest selection time in the stationary condition dialing with list, performed much poorer than its counterparts in the driving condition. Dialing with list also performed worst than the other driving conditions in terms of crashes per minute, and centerline crossing percentage.

Future Work

The results of the user study and reflection upon the data analysis show a room for improvement in the techniques that on average had faster selection time. Possible future directions for work to iterate over the input techniques include:

EdgeWrite was on average the fastest text entry technique, and from this study it was also found that selection based interaction methods were the fastest in selection overall. An exploration of how EdgeWrite could be used to specify the first couple of characters in a prefix followed by a special mode that would allow the user to switch to clutching or another one of the interaction techniques that is faster than EdgeWrite for selection could potentially yield better results than EdgeWrite or the list based techniques overall.

A couple of the participants mention how it was difficult to keep focus on both screens at the same time. Although there was initial exploration into possible ways of integrating the visualization techniques with the simulator, and the initial prototype with which the initial prototypes were tested used fully integrated visualization. It would be interesting to explore the idea again, regardless of the main constraining factors which is the amount of screen real space while the simulator is running. One possible concepts would be to simulate a heads up display as a semi transparent form that lies ahead of the users line of sight. Another possible exploration in this domain is to analyze the impact that only having a couple of items viewable from the list at a time has on selection time, and if it could be feasible to just display a couple of items as the user is scrolling though the list at the top of the simulator screen.

One of the main challenges that were observed from the user study was the difficulty that users had in paying attention to the word completion. Two observed cases of failure were: the system would complete the desired selection or would extend the prefix and the user would not notice, or the user would continue entering text regardless of an invalid prefix. Possible solutions to this problem include auditory feedback that alerts the user when their selection has resulted in an invalid prefix. Another alternative solution could be smart corrective text completion, which ignores the most recent character that the user entered if it matches the prefix that was extended by the previous character entry. An example of this would be when the user is asked to enter the word “Capitol” and they have entered “Cap” the user then enters the letter “i” which results in an extension of the prefix to “Capito”, many users when not looking at the feedback of the text completion would enter the “t” and would result in “Capitoi”. Simple rules can be implemented to take into account the last prefix extension and ignore characters that are entered after a prefix extension that could not refer to another longer prefix in the list.

References

1. Wobbrock, J.O., Myers, B.A. and Aung, H.H. (2004) Joystick text entry with date stamp, selection keyboard and EdgeWrite. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '04). Vienna, Austria (April 24-29, 2004). New York: ACM Press, p. 1550

2. .R. William Soukoreff and I. Scott MacKenzie Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard (1995, Behaviour & Information Technology, 14, 370-379)

3. “Within-subjects vs. Between-subjects Designs: Which to Use?” I. Scott MacKenzie “http://www.yorku.ca/mack/RN-Counterbalancing.html”