Using 3D Touch Interaction for a Multimodal Zoomable User Interface

Florian Laquai, Markus Ablassmeier, Tony Poitschke, and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
Theresienstrasse 90, 80333 Munich, Germany
{laquai,ablassmeier,poitschke,rigoll}@tum.de

Abstract. Touchscreens are becoming the preferred input device in a growing number of applications. They are interesting devices which are more and more introduced into the automotive domain. Current implementations impose problems like precise pointing or high visual attention and therefore the capabilities of projected capacitive touchscreens are investigated. Unlike traditional sensing techniques like resistive or optical measurements, projected capacitive sensors register the presence of a hand or finger even before the actual touch. This enables such devices to be used not only for 2D input but for interaction using the position and distance of the users finger relatively to the screen. The additional distance information is then applied to control a Zoomable User Interface. In this study touch and speech interaction is being applied to allow fuzzy input and lower visual attention demand than conventional Touchscreens. A demonstrator was developed using the mentioned input strategies for the automotive domain and finally evaluated in a user study.

1 Introduction

As displays become larger and resolutions are increasing, touch interaction is the input modality of choice in a growing number of devices. While such screens have been around in PDAs for several years and have proven to be capable of everyday use, the real breakthrough was achieved by portable music players, Tablet PCs or industrial applications. The traditional technique for touch recognition is resistive sensing which has the nature of wearing out after extensive use and therefore reducing recognition performance. A new way of achieving touch sensitivity is capacitive touch sensing which can be divided into ”projected” and ”contact” sensing. Both work with the fact, that the human body can store an electrical charge and thus can be used in a measurement process. The technology used here is projected capacitive sensing which means that the human body can be thougt of as one plate of a plate capacitor which means that no actual touch is needed. The sole presence of a finger can therefore be detected by a sensor array and the distance to it can be measured.

Given this technology, the finger distance information can be used to control a Zoomable User Interface, see [1], [2]. Such interfaces use zooming and magnification effects to manage large amounts of displayable data and usually give graphical objects

M.J. Smith and G. Salvendy (Eds.): Human Interface, Part I, HCII 2009, LNCS 5617, pp. 543–552, 2009.

a geographically defined place in the menu structure. This interface class can roughly be split into two subclasses, where zooming can be applied to the overall interface or just to portions of it. In this application the latter version is used. The amount of data displayed by an object is defined by its zoom level which can be interpreted as a selective information reduction or increase. With the finger distance controlling the zoom level and the 2D finger position controlling an invisible cursor such an interface can be navigated very intuitively and efficiently. As a result users can point to an object to increase the zoom level and preview the information behind it.

Speech input is an additional modality which can be used in combination with finger position to support interaction with small graphical objects. This way minimal visual distraction from the main task (e.g. driving a car) can be achieved while maintaining the possibilities of direct manipulation. The employed technology is being described in the first part which covers the touchscreen, the finger detection algorithm and the speech recognition engine. After that the demonstrator for the Zoomable User Interface and multimodal input is presented and finally the results of a user study are being discussed.

2 Technology

This first section gives an overview regarding the technical components. As a key component the touchscreen will be described in the following part.

2.1 Projected Capacitive Touchscreens

Unlike most traditional touch sensing technologies, Projected Capacitive sensors can detect the presence of a finger or a hand even without an actual touch. This is achieved by using conductive sensor pads made of copper or ITO (Indium Tin Oxide). In this case the human body acts as one plate of a plate capacitor and the sensors as the other plate. Depending on the finger distance, the capacity changes following the basic plate capacitor equation

$$C = \varepsilon_0 \varepsilon_r \cdot \frac{A}{d}$$

where C is the Capacity, ε the dielectric coefficient, A the surface area and d the plate distance. Thus the capacity of a sensor pad is related to the finger distance and can be used as an measurement variable. A capacity measurement for a sensor pad can be accomplished using a relaxation oscillator as seen in Figure 1 on the right. The output value of the timer component is directly related to the measured capacity and consequently to the finger distance. The device from Elo Touch Systems used for this application is built of two layers of glass and 16 sensor strips for x- and y-axis located between these layers.

All 32 sensors are connected to a single controller which sends out the raw data on a serial port. The data string is encoded as follows:

[X0][X1]...[X15][Y0][Y1]...[Y15][NULL]

A visualization of such a data set can be seen in Figure 2. From this raw data output the static offset first has to be measured and then subtracted from every following

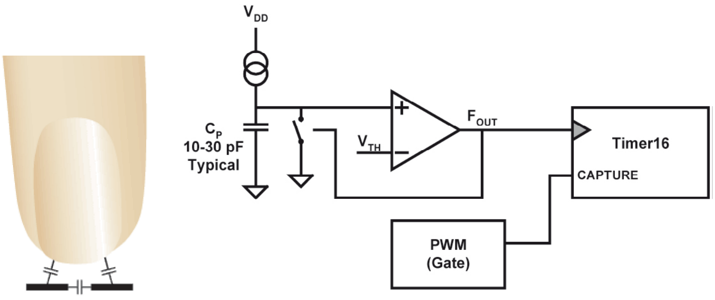

Fig. 1. Projected Capacitive sensing with a relaxation oscillator (Cypress Semiconductor)

sample. In the following step the values for each axis are smoothed using a sliding average filter with a window length of 2 samples in both directions to minimize introduced errors. After that a temporal averaging is performed over the actual and last data set. These measures significantly improve noise robustness while maintaining a quick sensor response.

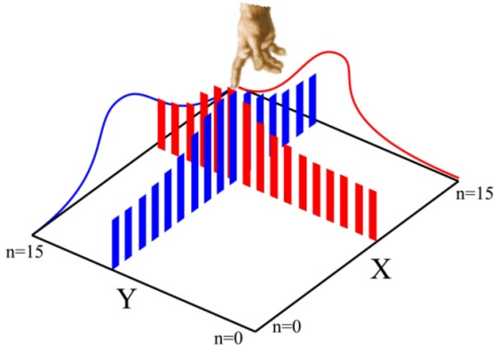

Fig. 2. Visualisation of sensor data output

2.2 Finger Detection Algorithm

After preprocessing the data a detection algorithm is being applied. For the first tests two detection algorithms were implemented:

- Cross Correlation
- Relative Maximum Search

For the Cross Correlation a set of 300 pre-calculated reference samples is used. These samples are created using a Gaussian which was fitted into a typical finger shape. The shape is then moved from one screen edge to the other simulating all possible finger positions. During operation the algorithm computes the correlation coefficient ρ_i for every reference sample using the following equation:

$$\rho_i = \sum_{n=0}^{15} sensor_n \cdot sample_{i,n}$$

Where

- i: index of reference sample
- $sensor$: data vector of one axis
- $sample$: data vector of the reference sample

In the next step the absolute maximum of all coefficients for every axis is determined and a quadratic interpolation around the maximum is performed to exceed the spatial resolution of 16 sensor strips per axis. The resulting interpolated maximum then corresponds to the finger / hand position.

The ”Relative Maximum Search” uses the fact, that the user’s index finger is the first distinct relative maximum when searched from the top left corner. This can be assumed because only one hand coming from a very predictable direction can be used in a automotive setup. Thus the absolute maximum is searched as a reference and then the index finger is assumed at the first relative maximum that exceeds a fixed percentage of the absolute one. The next step is like in the above algorithm a quadratic interpolation using the points around the finger position.

For the finger distance calculation the difference between the absolute extremums averaged for both axis is calculated ($\Delta MaxMin_{x/y}$). The following equation shows this relationship, whereas the constants 60 and 10 have been determined empirically and z is the finger distance.

$$z = 60 - 10 \cdot \sqrt{\frac{1}{2} \cdot (\Delta MaxMin_x + \Delta MaxMin_y)}$$

It was found that the usable working space of the sensor is approximately 0 - 4 cm above the screen distributed over the whole sensor area.

During the first expert evaluations the Relative Maximum Search algorithm proved to be more robust and precise. The Cross Correlation was too prone to false detections for various touching techniques.

Whilst still better than Cross Correlation, the Relative Maximum Search still did not show the necessary accuracy when actually touching the screen surface. Therefore a combination was used with the manufacturer driver handling clicks and dragging while the described algorithms (correlation and maximum search) handle operation when the finger is distant to the screen.

Problems arise when large parts of the palm are closer to the screen than the index finger due to signal ambiguities (see [7]). This can only be solved by applying more sophisticated detection algorithms.

2.3 Speech Recognition

The commercial engine ”ScanSoft VoCon 3200” was used for speech recognition. It provides the possibility of a constantly open microphone, which was favored in this application. The vocabulary to be recognized can be defined using a BNF (Backus-Naur-Form). For every recognition iteration the engine returns a list of possible words which is sorted by the corresponding confidence value. This is an index for the estimated recognition quality and can be used as a criterion for rejecting or accepting a recognition result.

3 Application

Both input modalities ”Touch” and ”Speech” are fed into the demonstrator application depicted in Figure 3. The screenshot shows the main menu with the user’s finger pointing at a ”Contacts” menu and selecting a list entry. The application includes zooming and previewing content as well as multimodal input, window management and text input using zoom techniques.

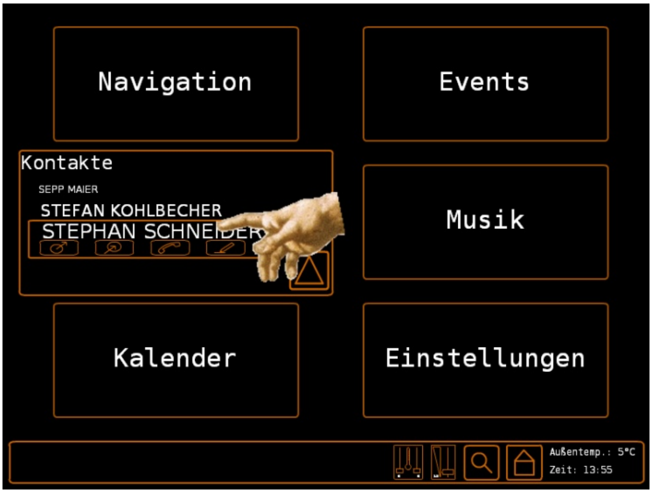

Fig. 3. Screenshot of main menu with one zoomed menu

In the following sections the different features are discussed in detail.

3.1 Zooming

As mentioned in the introduction, partial zooming is being used in this implementation. The desired item to be zoomed is selected by pointing towards it in the sensor’s working space. If the finger hovers over a graphical object, the object can change its size and state depending on the finger distance. The concepts shown here divide the working distance into four zones:

- finger not detectable
- finger is distant
- finger is close
- touch

The zones and the corresponding states of a list item are depicted in Figure 4. This way the user can literally „dive“ into parts of the user interface and preview or even select previewed items. All interactions are designed in a way that pointing to an object and zooming it never causes data to be manipulated. This can only be done with an actual touch that naturally gives haptic feedback.

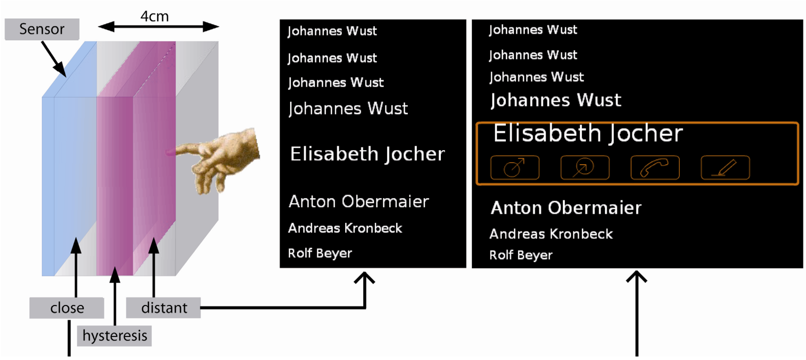

Fig. 4. Proximity Zones and corresponding states of a list item

Figure 3 shows the possibility to select the ”Contacts” menu by pointing to it in the ”distant” sensor area. This causes the menu to be increased in size and a list of contacts is shown.

3.2 Zoomable Lists

The list uses a kind of ”Fisheye Zooming” where the item under the user’s finger is the largest while the others are shifted aside. As a result the item can be touched more easily while maintaining the amount of displayed data.

By moving the finger closer to the screen into the ”close” sensor area a frame is being displayed around the item and buttons are attached that represent the most common functions for this item class. These can be ”Call contact”, ”Use as route destination” or ”Edit contact” for the contact list or ”Play song” and ”Add to playlist” for a song list.

In consequence it is possible to

- preview and search a list entry
- select it and choose an action
- execute the action

with just one finger movement and a final touch of the screen. Scrolling in the list is achieved by hovering the finger over the top or bottom end of the list as can be seen in Figure 5. The more the finger reaches the list end, the faster the content is scrolled.

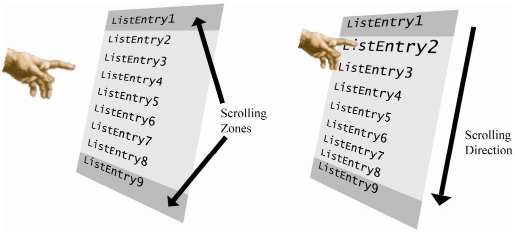

Fig. 5. Fisheye Zooming for lists and scrolling zones

3.3 Text Input

The distance information can also be used to enhance existing or create new text input methods. As a first example an onscreen keyboard was developed. It uses zooming effects combined with distance related key displacement. This means that the zoom level of each key in the area around the user's finger is indirectly proportional to the 3D-finger distance. To avoid overlapping, the adjacent keys are being moved away from the projected finger position, see Figure 6 left. The combination of the two effects makes it possible to minimize space requirements of the virtual keyboard which is of special interest on small screens. Consequently the user still has a good overview of the actual context while typing in opposite to a full screen version as it is often used in portable devices. The maximum zoom factor for the keys is 2.5 which equals 6.25 times the initial key area. This improves both readability and error rate significantly compared to a static keyboard of equal size. The keyboard is activated by touching a text input area and placed with the arrow on the bottom left corner pointing to the selected input line.

The second keyboard implementation (Figure 6 right) arranges the keys in a square raster which is divided into 3x3 sections, see [5]. Seven of these sections contain 3x3 keys and the remaining sections are used as ”Space” and ”Backspace” key. This results in a total of 65 usable keys. Again the required space is to be minimized and therefore the key size is chosen so that a finger tip covers about 4 keys. Here the finger proximity causes underlying section of 9 keys to be zoomed over the whole keyboard area (Figure 6 right shows the zooming status). As described for the menu states, the detectable finger distance is divided into two sections corresponding with the two zoom states and a hysteresis for increased stability in the transition area.

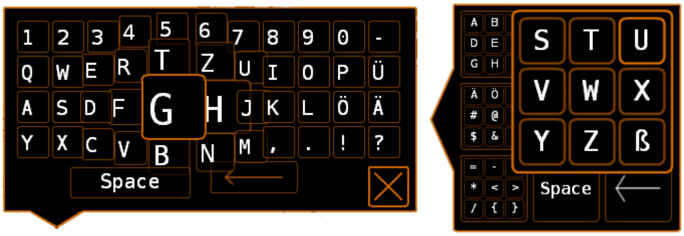

Fig. 6. Fisheye QWERTZ keyboard and Matrix keyboard

3.4 Combining Modalities

In this application two kinds of multimodal interaction have been implemented. Both parallel / independent and sequential interactions (see [3], [4]) are used for different commands. Main menu items like „Music“ or „Navigation“ can be activated at any time and System state. In that special case speech input is even more powerful than touch input. An example for mode-independent and equally available manipulation is shown in Figure 7 for a volume slider element. It can either be pointed at for zooming and dragged or be changed by saying „Volume Up“ or „Volume down“.

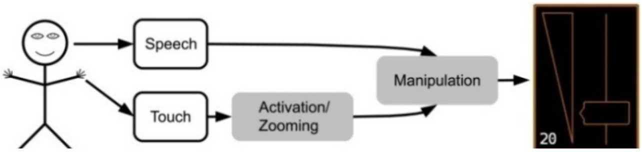

Fig. 7. Independent use of modalities for a volume slider

By combining finger position information with speech input, the demand for visual attention can be reduced while maintaining input possibilities. Figure 7 shows an application where list items are selected by pointing at them and saying the desired action. So the user does not need to touch a certain small area but only points roughly towards it.

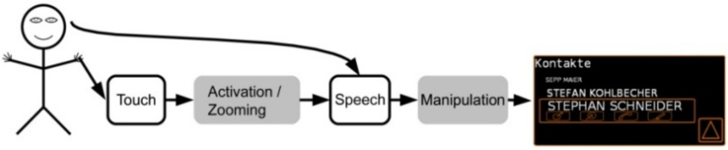

Fig. 8. Selecting list entries by touch and issueing commands by speech

The same actions that are available in the surrounding box in the „Close“ state can then be triggered by speech input.

4 User Study

The system evaluation was conducted in the driving simulator of the Institute for Human-Machine Communication. It consists of a real car cockpit with the Touchscreen installed in the center console and a projection screen in front of the car for the environment visualisation. After a short system introduction several tasks had to be performed by the 20 test persons while driving in a highway scenario without other cars. The time for the completion of each subtask was measured and after that the participants filled out a questionnaire about their subjective impression.

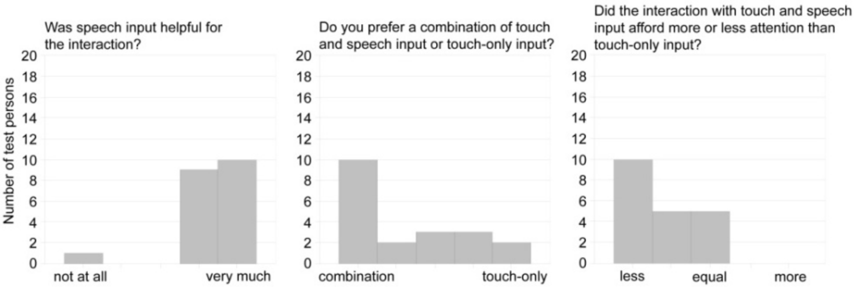

Fig. 9. Comparison of multimodal and touch-only interaction

First the findings regarding multimodal input shall be discussed. Generally multimodal input was considered superior to touch-only input and less stressful, see Figure 9.

Next, the Zoomable User Interface structure had to be rated by the participants. Again the overall opinion was very positive for the intuitiveness and usefulness of such a menu system.

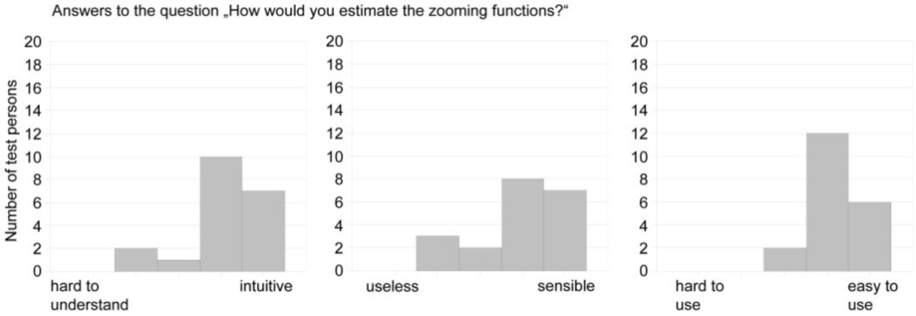

Fig. 10. User estimations for zooming functions

Finally the average times for text input were measured and compared against the BMW iDrive speller. This systems uses a rotary controller for character selection and word estimation. The average time needed for the city name „Schrobenhausen“ with the iDrive system was 0:36 min, for the Fisheye keyboard 0:56 min and the Matrix keyboard 1:06 min. The main reason for the much longer input times of the touch input variants was identified to be the accuracy and jittering of the developed touch driver software.

5 Conclusion

The presented system shows the potential of capacitive Touchscreens with proximity detection. A Zoomable User Interface was developed to provide a proof of concept and test environment for user studies. The main findings are a high acceptance of such Graphical User Interfaces and the combined touch and speech interaction.

The main weakness at this point is the accuracy of the touch driver software which sometimes missed a touch event or jittery cursor position. This could be improved by applying more sophisticated finger detection algorithms like neural networks or classical image recognition algorithms.

References

1. Good, L., Stefik, M., Bederson, B.: A Comparison of Zoomable User Interfaces and Folders For Grouping Visual Objects. HCIL-2004-33, CSTR-4641, UMIACS-TR-2004-83 (November 2004)
2. Bederson, B.B., Meyer, J.: Implementing a Zooming User Interface: Experience Building Pad++. J. Software: Practice and Experience (1998)

3. Oviatt, S.: Handbook of Human-Computer Interaction. In: Jacko, J., Sears, A. (eds.). Lawrence Erlbaum, New Jersey (2002)
4. Seifert, K.: Evaluation multimodaler Computer-Systeme in frühen Entwicklungsphasen. Dissertation, Fakultät V - Verkehrs- und Maschinensysteme, TU Berlin (2002)
5. Ingmarsson, M., Dinka, D., Zhai, S.: TNT – A Numeric Keypad Based Text Input Method. In: CHI 2004, Vienna, Austria, April 24-29 (2004)
6. Hassenzahl, M., Burmester, M., Beu, A.: Engineering Joy. IEEE Software 1(2), 70–76 (2001)
7. Westerman, W.: Hand Tracking, Finger Identification and Chordic Manipulation on a Multi-Touch Surface, Dissertation, University of Delaware (1999)