

Beyond-the-Desktop Interactive Visualizations

Till Ballendat

Abstract— Desktop computers are capable of visualizing and interacting with large information spaces. Since mobile devices have been increased in both popularity and processing power, they have the potential of executing these desktop tasks. Still there are limitations such as low screen size and reduced input capabilities which have to be considered. This paper summarizes a number of techniques for mobile devices like PDAs, handhelds or smart phones which provide visualization and interaction with large 2D graphical data. First it presents general approaches and then specializes on geographical data, image data and web site techniques. The techniques are reviewed and evaluated using four criteria : The design space of the input device, The size of the required display, The type of the required data and the achievements of the implementation. The result is a matrix providing a comparable overview of the discussed techniques. This can be useful for further researches or application development.

Index Terms—Geographical data, map, web, visualization, zoom and pan, small screen, mobile devices

1 INTRODUCTION

As consequence of rising popularity and increasing hardware power of mobile devices, information spaces that have been traditionally available on desktop computers and notebooks, are now introduced to smaller devices. These information spaces can be large pictures, web pages, maps or many other kinds of documents. This brings new challenges to application developers. One major problem is the small screen size (see figure 1). Even if the display resolution and size of modern mobile devices are still growing, they are physically limited to the dimensions of the device. Some recent devices have already reached these limitations. So displaying a large information space in its entirety will lack in detail information, whereas a detail cutout makes it impossible for an user to perform spacial tasks like discovering the nearest ATM on a city-map. Another limitation is the devices processing power. Even if this factor could be lapsed with upcoming increase of processing power, considering todays devices it is still important to think of efficient ways to handle the large data. One more problem which developers for mobile devices have to face, is the limitation of input peripherals. Most desktop computers use a mouse with a qwerty keyboard. This enables fast text input and a high number of key commands. Furthermore, most people are trained to accurately select items or navigate through content using a mouse with a scroll wheel. Both can't be provided on a mobile device. Hence, it is necessary to elaborate effective and easy to use navigation techniques. Besides the attention towards the application may be highly reduced for mobile users because they might be affected by environmental conditions. For example the interaction with a mobile navigation system should be suspendable at any time, for example that the user can focus on other road users when it comes to the point.

This is a very complex research topic and there are several existing approaches. Chapter 2 collects general ideas for handling large information spaces. The following three chapters will present specialized techniques for managing geographical data, image data and web sites. Chapter 6 reviews all the techniques and compares and evaluates them.

2 GENERAL APPROACHES WITH

Visualization and interaction of large 2D graphical data on mobile devices raises several problems. Some of them are associated to the visualized content, but there is also a big subset of general problems. This section will present a number of general approaches towards interacting with large 2D graphical data on small mobile devices. Most of them were derived from existing desktop applications and adopted



Fig. 1. Desktop map superimposed by a mobile device [16]

to the characteristics of the mobile devices. The main limitations you have to deal with are the low processing power and the small display size, but also the lack of input devices is highly influencing the application design.

2.1 Double Scrollbar Panning

The classic approach of panning in large 2D graphical data is to show two scrollbars. One for horizontal and one for vertical scrolling [7]. The screen shot (see figure 2) shows a PDA device displaying a small section of a map. The user can now drag the scrollbars with a pen input device. The same technique is also used for smart phones and mobile devices without touch input by using arrow keys. An additional feature is the adaptation of the scroll bar's size to the amount of possible scrolling, to allow the user an estimation of the document's dimensions. The larger the scrollbar the smaller the remaining off screen space. Mark Harrower and Benjamin Sheesley call them smart scroll bars [14].

The big disadvantage of this approach is that only one dimension can be modified at a time. So if the user wants to see an off screen part of the displayed data which is situated in the bottom right, s/he has to switch between dragging the horizontal and vertical bar multiple times, until the screen displays the desired part. But there is also an advantage of this technique: As there is no direct pointing at the actually displayed data, these pointing actions can be used for other goals like displaying context information on a map, modifying image data or interacting with a web interface.

- Till Ballendat is studying Media Informatics at the University of Munich, Germany, E-mail: till.ballendat@campus.lmu.de
- This research paper was written for the Media Informatics Advanced Seminar on Information Visualization, 2008/2009

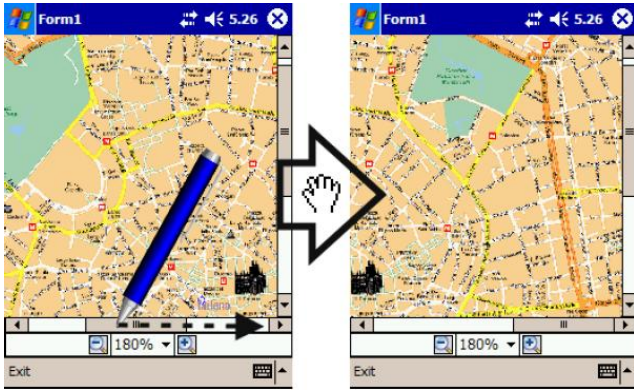


Fig. 2. Standard Double Scrollbar [7]

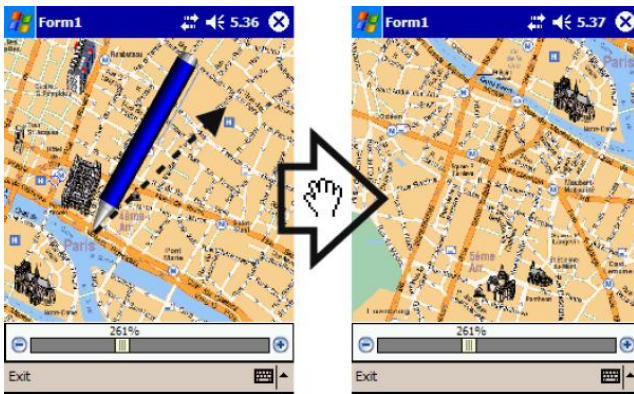


Fig. 3. A map interface using The Grab and Drag technique [7]

2.2 Grab and Drag

Like the Double scroll bar approach, Grab and Drag is already widely used in desktop applications. For example Adobe Acrobat Reader or Google Maps provide a "Hand-Tool" for the user, which can be used for moving the displayed information. This idea is very intuitive, because it's the same thing people do in real life situations. The pointing device simulates a hand or a finger which can move the content by dragging it. Therefore the technique can be applied to every mobile device, which has a touch input based or any other kind of pointing device. The screen shows the grabbing action executed on a map (see figure 3). The dashed arrow indicates the movement of a pen having contact to the touch surface [7].

2.3 Overview and Detail

This approach provides both an overview of the complete data and a detail view. The two views are contextual coordinated so that the user always knows the position of his detail view in relation to the whole data. The context view displays a rectangle representing the current section displayed in the detail view [26]. This technique can make ideal use of hardware with two screens. But as most of the existing mobile hardware provides only one screen, many applications use a relatively small context overview and a possibly large detail view, as this is what the user is actually interested in.

Karstens and Rosenbaum [17] developed this approach for mobile handhelds with pen input and called it Large Focus-Display (see figure 4). There are three possible user interactions. First users can pan the detail view by using Grab and Drag. As both views are contextual coordinated, the rectangle in the detail view instantly updates its position and helps the user to keep the context. Another possible action is to drag the rectangle inside the overview in order to change the detail view. For faster navigation it is also possible to jump to the desired sector by pointing on it in the context view.



Fig. 4. Large Focus-Display [17]

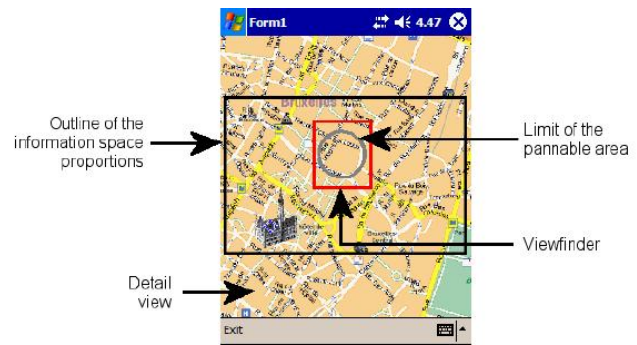


Fig. 5. Zoom-Enhanced Navigator [7]

A more sophisticated approach based on the classic overview and detail method is presented in an article by Stefano Burigat [7]. He called the new technique Zoom-Enhanced Navigator or ZEN. The information can be displayed full-screen. There are three shapes on top of the detail view (see figure 5). The biggest one represents the whole data and gives the user an idea of the total measures. The small rectangle shows the current detail view and the inside located circle indicates the drag able area allowing the user to move the view. Unlike all previous mentioned techniques, ZEN provides integrated zooming functionality. Zooming is achieved by dragging the area between the circle and the small rectangle. Of course, while zooming or panning, the dimensions and locations of the shapes will instantly be adapted to provide context awareness. Burigat conducted user studies comparing double scroll bar panning and Grab and Drag with ZEN. He identified significant advantage of ZEN concerning orientation cues. This goes with the fact, that the whole display space can be used for visualizing the desired data, because there are no additional screen objects for panning or zooming actions like scrollbars, zoom level selectors or sliders. Still some users found it hard to select the small areas which provide the zooming and panning abilities.

2.4 Focus and Context

Focus and Context techniques try to provide context information and detail information in the same view. Therefore only a small part of the data is shown in full detail. The remaining space is used to show the complete context information by distorting it. Of course this is only suitable for users, who are interested in a specific region and their interest decreases with the distance from the focus view. For all spacial tasks like measuring or estimating distances or dimensions, Focus and Context is rather ineligible, because the view is mostly strongly distorted.

A very popular class of Focus and Context techniques are fisheye views. There are many different implementations varying in complexity and visual representation.

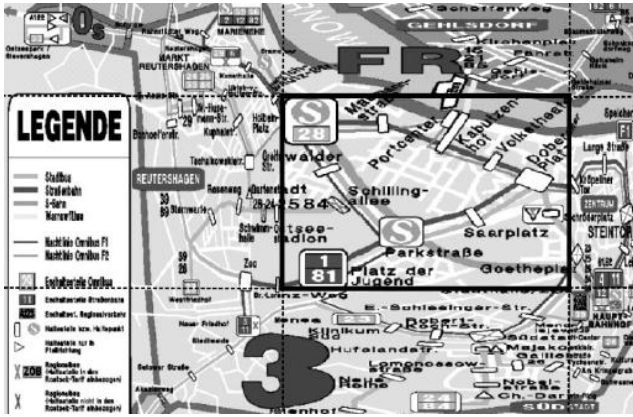


Fig. 6. Rectangular FishEye-View [21]

With respect to the low computational power of mobile devices, Robertson proposed a technique called Document Lens [23]. A rectangular shape is used for displaying the focus view. The distortion of the context is realized by simply showing it in four linear perspective views. Especially at the edges the context is hardly observable.

However there exists a more sophisticated approach called Rectangular Fisheye-View (see figure 6). As the name indicates, again a rectangular region is used for presenting the detail information. This technique improves the context visualization of Document-Lens by providing three different distorting implementations [21]:

- **Uniform context scaling:** This computational inexpensive method divides the whole context belt into eight rectangular grids. Each grid is independently scaled in uniform manner. As a result there are visual discontinuities between the grids borders and the detail view rectangle.
- **Belt-based context scaling:** This method splits the context into several belts. A scaling factor is assigned to every belt, so that the degree of detail decreases with increasing distance to the focus view. This improves the visual continuity, while still computing relatively fast.
- **Non-uniform context scaling:** Smooth transitions for the whole visualization can be accomplished by non-uniform context scaling. The scaling factor is continuously decreased with increasing distance from the detail view. The computational expense is rather high in this approach, but provides the best visual result.

A slightly different approach is to provide a circular focus view. Harrie developed an implementation called Variable-Scale Maps [13]. The technique achieves very smooth transitions and a natural visualization, but demands the highest process power of all the previously presented techniques.

Most Focus and Context implementations do not only give a static view, but allow the user to move the focus area or change the focus size or the scaling factor in order to obtain the desired information. By taking this into account, an implementation has to be chosen with respect to the devices processing power. So that a smooth interaction is assured at any time.

2.5 Tilt-based Zooming and Panning

Tilt based interaction opens new possibilities for zooming and panning in 2D design spaces such as maps, images or image collections. Jun Rekimoto [22] proposes a novel interaction technique which uses rotation sensors for all three spatial dimensions. He implements several prototypes like a tilt menu and a 3D object inspector, but also a browser for navigation in large 2D spaces. In addition to tilting input, the device requires at least one button. The basic screen provides a simple bird's eye view of a detail selection of the 2D data. By holding down the button, the user activates the tilting navigation. When

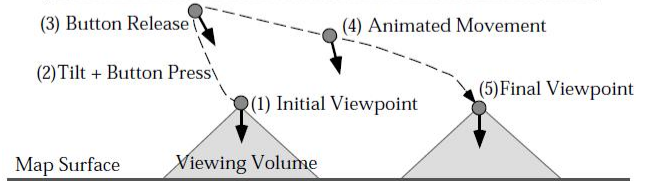
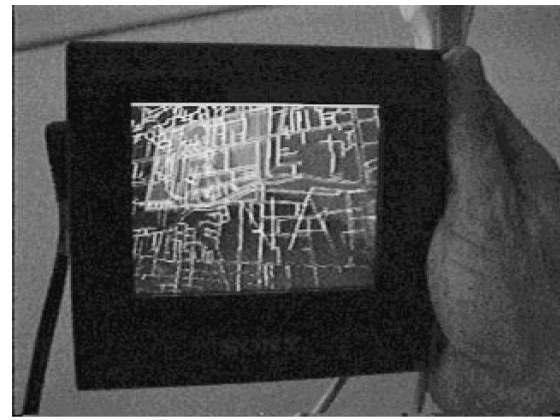


Fig. 7. Navigating a map by tilting the display [22]

the device is tilted to the right, the viewpoint changes to an upper-left position and shows a perspective overview of the right part of the 2D space. A cursor is shown in the center of the display. By tilting and rotating the device, the user can explore the map and point to the desired position. When releasing the button, the program shows a smoothly animated movement to the selected destination (see figure 7).

A similar technique called Speed Dependent Automatic Zooming or SDAZ handles the zooming functionality automatically by observing the users scroll speed. The faster the user navigates through the data the more of the context is shown, because the user could no longer handle the rapidly scrolling information. Existing interfaces require explicit control of both zooming and panning in the information space. Switching between the two actions is time consuming and often requires the use of both hands. SDAZ provides a constant scrolling speed by automatically adapting the zoom level. So the user can navigate very fast without losing the orientation. An implementation by Eslambolchiar and Murray-Smith [12] uses the tilt and the acceleration of the device for controlling the scroll speed.

Both techniques can be used one handed which makes them especially useful for field workers. On the other hand these techniques can't be used accurately when the user him or herself is moving or the device has to be attached to the users arm or an external object. For example when riding a bike and attaching the device to the handlebar or using the device while running. The user has to hold still when interacting with tilt.

2.6 Information Hiding

Reducing the amount of presented information in an appropriate way is called information hiding. The goal is to provide an overview of the displayed information and present the desired information in detail. That for the approach requires appropriate interaction techniques giving users the ability to explore the hidden information. The hiding of information can't be easily generalized, but depends on its type. For example images can be scaled down or represented by an icon, text paragraphs can be reduced to the first words or maps can be reduced to a main street overview. So actual implementations can be found in chapter 3 and 5.

3 VISUALIZATION/INTERACTION WITH GEOGRAPHICAL DATA

Browsing geographical data on mobile devices is a highly relevant research topic in information visualization. Not only because Geographical data is used in a variety of applications, but also because

the developed techniques can be applied to a broad range of visualizations like wiring diagrams, network plans for subways or computer networks. Beside the general problems like zooming and panning like which were presented in chapter 2, there are also a number of specific problems which come along with map interaction and visualization of.

3.1 Visualization of Off-Screen Objects

When navigating in large geographical data the user always has certain points of interest. As the view changes into higher detail it is likely that these objects disappear from the screen. To keep the user from having to scroll and zoom in and out all the time, techniques were developed to indicate the distance and direction of those off-screen locations. This is for example relevant when users want to find the nearest gas station. Maybe they also take other factors into account, like their travel direction or the condition of the road. So users want to have an overview of all nearby stations without having to zoom out and not being able to see the complete street information.

One approach which visualizes off screen objects is called City Lights [27] (see figure 8). A map related implementation of this approach displays boxes at the screens border. Their position is defined by the intersection point of the display border and the connecting line between the screens center and the point of interest. The further the boxes reach into the display, the nearer the representing object is located towards the center of the view. This technique still has two major disadvantages. First the off screen objects distance can not be correctly visualized if the corresponding box would appear in the display corner. Second no absolute distance can be estimated by looking at the boxes. Only a relative distance comparison between objects is possible.

Another possibility is to simply draw arrows at the screens border. Again the arrows are positioned on the connecting line pointing at the off screen object. The distance information can be encoded in the arrow length. Longer arrows indicate far away objects. Another possibility is to encode the distance in the arrows color shade. Like City Lights, also the arrow representation can't provide an absolute distance encoding. A solution for this problem would be to attach to every box or arrow a label containing the textual representation of the distance. But this could interfere with the maps information and distract the user from his task.

A different technique called Halo (see figure 8) was developed by Patrick Baudisch [2]. The approach draws a circle around every off screen object. The radius size exceeds the distance from the display to the object. So that the user can see an arch at the display border. This allows the user to not only estimate the direction, but also the distance to the point of interest. While having only one visual representation per object.

All these techniques remain the possibility to encode further scenario specific information. Using color encoding, labels or symbols. For example when looking for a restaurant it could be usefully to show symbols like a fish, burger or a pizza to indicate the type of food the place cooks.

To compare the performance of the approaches both Baudisch [2] and Burigat [6] conducted an user study. Participants had to accomplish several map tasks using all three techniques embedded into the same zoom able user interface. Only Baudisch study resulted in a significantly faster task completion with his Halo interface. Both error rate and users preference did not show any significant advantage. This of course does not abnegate the demand of visualizing off screen locations, but requires a scenario specific choice of the ,most appropriate implementation by the developer.

3.2 Scenario specific Adaptation and Information Hiding

Reducing information to a minimum while still providing sufficient data for the user to complete his task, is not only a very effective way of enhancing visualization on small screens but also hard to realize. This approach can't be easily generalized and automated, because it highly depends on contextual information. Gokker and Myrhang classified the categories personal, environmental, task, spatio-temporal and social context to be relevant for adoption of geographical data. This context information can be gathered by sensors like GPS or camera.

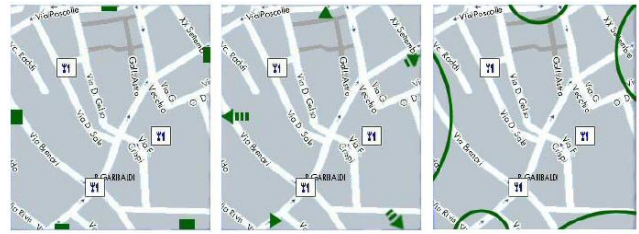


Fig. 8. City Lights (left), Arrows (middle), Halo (right) [6]



Fig. 9. Maps of the same area for different seasons and age groups [19]

Often the information is simply provided by the user or narrowed down by the purpose of the program.

Nivala and Sarkoski [19] developed a map generalization technique which takes personal and temporal context information into account. Depending on the users age group different symbols are chosen for representing points of interest. As traditional map symbols may not be familiar to children, bigger and more intuitive symbols were designed. Furthermore the points of interest are filtered by age and temporal criteria, with respect to both season and day. For example at night the map shows bars, restaurants and cinemas and filters shopping places or museums. In the winter the map displays skiing resorts and mountains are displayed in white (see figure 9).

Agrawala and Stole [1] proposed a technique called LineDrive which considers task and spatial context information. Its goal is to provide an automatic generalization of cartographic data for navigation purposes on mobile devices. Based on cognitive psychology they discovered, that displaying turning points in detail for a route is of much higher importance than showing the exact shape and length of the road. By knowing the starting and ending point of the track, the use of screen space can be optimized by rotating and skewing the map to align the route to a vertical axis. Considering both facts the implementation was able to display a much longer route on the available screen space than using the original map data. But still without losing essential information for the desired task (see figure 10). The screens also show that LineDrive is very similar to a hand drawn map, which indicates a natural abstraction technique.

4 VISUALIZATION/INTERACTION WITH IMAGE DATA

Modern mobile devices do not only provide high processing power, but also include a high number of additional features. Many of them have high speed connections to other devices or computer networks which offer image libraries. Also, built in photo cameras are a very common feature in current mobile devices. This leads to the demand for image data exploration techniques including both the exploration of image libraries and single large image data.

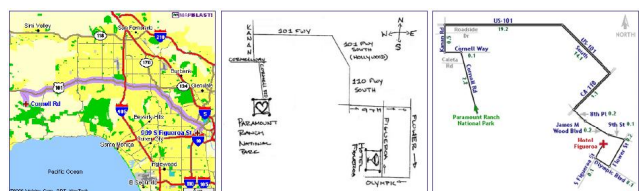


Fig. 10. Original map (left), hand drawn map (middle), LineDraw (right) [1]

4.1 Grid-based Image Browsing

The Grid-based image Browsing technique was introduced by Rosenbaum and Schumann [24] for exploring large images which are too big to fit the small screen of a mobile device in their full resolution. The approach is based on the idea of Overview and Detail like described in section 2.3. As an overview the implementation displays a transparent grid overlay. Every grid cell has the size of one detail view which would exactly fit the whole display space. The currently displayed section is emphasized in the grid. With this the user can easily estimate the dimension of the whole image in relation to the actually displayed detail view. The transparent grid can also be hidden to prevent visual distractions. The implementation provides interactions for jumping, panning and zooming. Therefor pen-based input and a 2D-rocker or simple arrow keys have to be provided by the device. Jumping to a desired place in the image is done by tipping on the corresponding cell of the transparent grid overlay. This initiates a "fly-over"-animation which shows the intermediate content. for panning purposes the user operates the 2D-rocker. In this way panning is not assigned to the division of cells, but can be performed in arbitrary steps. The emphasized rectangles position, which represents the current screen, is always dynamically updated. For zooming out of the image the user can drag the pen over several cells which will then be merged into a new cell. The grid will dynamically reduce the number of cells. Each cell represents the size of the merged cell. For magnification the user holds down the pen on a cell for a predefined delay time. The cell is split up recursively depending on the scale factor until the user releases the pen. As this approach makes use of the JPEG2000 image format implementation, the required processing power can be reduced.

4.2 AutoZoom and GestureZoom

Having a large number of images on a mobile device does not only require the exploration of one image but also the exploration of a whole image gallery.

A basic approach is to provide a list of thumbnail views ordered in rows and columns. The list can be explored by using double or single scroll bars like described in section 2.1. By selecting a thumbnail the user sees a full screen representation of the corresponding image. of course a high number of images leads to a lot of scrolling.

An implementation called Pocket PhotoMesa has been presented by Khella and Banderson [18]. The user navigates through the images by selecting groups of images with a pen. That way the thumbnails are magnified to automatically fill the whole screen. the technique does not require any scrolling, but displaying large image galleries on one screen highly reduces the explanatory power or the thumbnail views. Again this leads to a high number of zooming actions.

The Funkyzoom project developed two more sophisticated approaches [20]. Both techniques order the thumbnail image representation in a single image wide vertical list in chronological order. A mobile device with pen input is required. The first implementation is called AutoZoom. To scroll the list the user can drag and hold the pen into the desired direction. The technique uses speed dependent automatic zooming like described in chapter 2.5. The speed level is determined by the distance of the drag action. The automatic zooming is done by smoothly reducing the thumbnails size. This enables the user to keep track on the thumbnails while providing a much faster scrolling speed (see figure 11). In the second implementation called GestureZoom, the zoom level is not automatically adapted. The user can manually control the zoom level of the thumbnails by the amount of dragging he carries out in a horizontal direction. The higher the dragging distance, the smaller the images appear on the screen.

In both implementations the normal image size is restored using smooth animations after releasing the drag action. An user study compared the performance of AutoZoom and GestureZoom with a basic approach using discreet zooming. It showed that a more accurate and faster identification of images was possible with the developed approaches.



Fig. 11. AutoZoom with low(left) and high(right) scrolling speed [20]

5 VISUALIZATION/INTERACTION WITH WEBSITES

Mobile devices provide a rising number of connection possibilities like GPRS, HSDPA or wireless LAN. With the increasing availability of these networks, people can access the internet hardly everywhere at any time. The high bandwidth enables users to access content which was originally designed for desktop users. Also only a very small subset of web content is explicitly provided for mobile devices, like using the wireless application protocol (WAP). This again results in the problem that a large information space has to be displayed on a small screen.

Web browsing provides the additional possibility to manipulate the data on a proxy server. Some approaches make use of this possibility, because proxy servers have fewer limitations in processing power. Also this practice can assure that only the required data has to be transferred to the device. This is especially useful if the connection has only low bandwidth or when data transmission is expensive for the user.

5.1 WEST: A Web Browser for Small Terminals

WEST has been developed by Bjrk in 1999 [4]. That's why it requires very little resources from the mobile device. The main work is done by a proxy server. The server splits the original website into smaller chunks. Additionally the text is reduced by extracting keywords and links from every chunk. The processing is done in real time, so that the technique can be applied to every html-page. The client application downloads the prepared data and shows the chunks using a FlipZoom view. FlipZoom is a Focus and Context approach originally designed for desktop applications [15]. It shows one big box in the center of the screen containing a detail view of the first chunk. All the other chunks are arranged around the main chunk in smaller down scaled thumbnail boxes. The user can select a thumbnail to exchange it with the detail view. Because of the devices low screen size and resolution the small chunks could hardly be identified. Thus WEST shows the extracted key words or links instead of a down scaled image. So this approach uses a Focus and Context technique 2.4 combined with information hiding 2.6. For navigation it is possible to use a pen based input or also number keys like provided by smartphones. Every thumbnail would be associated to a number.

5.2 Powerbrowser with Text Summation

Similar to WEST, an approach by Buyukkokten relies on intelligent summation of text paragraphs [9]. Again the page is split into semantic textual units. The text is summarized by a sentence divider and a sentence ranking module. In addition keywords are extracted for a summary headline. This processing is implemented on a proxy server. The clients browser has been developed on top of the Powerbrowser implementation [8]. It displays the keywords for each unit using a tree based page overview. The user can then access the summation or display the whole unit full screen. The technique also takes html forms into account and adapts them into a mobile representation.



Fig. 12. Expand(1) and Collapse(2) action in the Collapse-to-Zoom browser [3]

5.3 Web Page Analysis

Another approach which adapts web pages for small screen divides was developed by Chen [11] in 2005. While WEST did not consider the original visualization of multimedia content like images or videos using current technologies like javascript or cascading style sheets, this approach has a higher potential for displaying any kind of web content. The main part of the concept is the page analysis. It considers the DOM tree to identify explicit separation and also considers visual gaps for identifying implicit separators. That way the page is divided into content blocks. This can either be done by a proxy server or on the clients device itself. After the user requests a web page and page analysis is done, the whole page is shown as an annotated image where the identified content blocks are marked with different colors. By clicking on the blocks, the user can access a detail view of the corresponding sub-page. The page analysis performance was tested with the 50 most popular web sites. More than 90 percent of the pages could be split in an appropriate way.

5.4 Collapse-to-Zoom

Patrick Baudisch faces a major problem of Chens browser implementation in his more sophisticated approach called Collapse-to-Zoom [3]. When entering the web site, the browser provides an overview of the whole page. Again the implementation tries to identify the page structure and divides it into blocks, which could be the header, the navigation, advertisements or any content. The problem which also accrues in Chens approach is that users might not be able to identify the desired content area, when the whole page is reduced into the available screen space. He would have to zoom in and out several times to find the content. So Collapse-to-Zoom offers the possibility to carry out a drag action from the top-right to the bottom-left of an obviously irrelevant block (see figure 12). It leads to the minimization of this area so that the rest of the page magnifies into the newly available screen space. After all the irrelevant data is collapsed the user can easily identify the designated block and expand it to full detail by dragging on it from the bottom-left to the top-right (see figure 12). Still the user is able to select links at any time by just tapping on them. Collapsing irrelevant content provides an additional advantage. Next time the user visits the same page, it shows up pre collapsed.

6 DISCUSSION

The previous chapters presented an overview of existing approaches for visualizing and interacting with large 2D data using mobile devices. They range from general to very specific techniques. Some address different goals and require different hardware and input devices. Others address the same goal, but differ in their capabilities. To be able to compare and evaluate them for further studies or application development, it can be helpful to provide a structured overview. Because of the variety of techniques the evaluation criteria have to be very general but still easily comparable. Hence, four global criteria

were motivated by the interaction process which every technique has to obey (see figure 13).

- **Input:** First Criteria is the design space of the input peripherals. Card [10] analyzed the design space for the most common input devices. He considered many factors like the sensed property (position, motion or pressure), the sensing type (touch or mechanical) and the number of dimensions. As mobile devices have limited input possibilities the distinguishing feature will be the number of dimensions which the input device provides. Dimensions can be either linear or rotary movements along the x,y or z axis sensed in an absolute or relative way. A computer mouse for example has can be linearly moved in the x and y axis, which results in two dimensions. Many mice provide a scrolling wheel which can be rotated in several steps around the x axis and some buttons which can be pressed. This classifies the the pc mouse as 2.5 dimensional. The presented techniques can either use touch, key or tilt input or a combination. Key input can be described as a mechanical, pressure based linear input. Touch panels are classified as a two dimensional linear input device because movements can be made along the x and the y axis. Tilt based input devices can be rotated around all three axis. So they use three dimensions.
- **Device:** The Device itself has three main distinguishing factors. The size and resolution of the display, the processing power and the kinds of additional features like sensors or connectivity. It is obviously hard to specify the processing power of a technique without holding an actual implementation and carrying out tests to comparative tests. The additional features do only affect a small subset of the presented approaches. so the screen size and the corresponding resolution is used as a criteria. To provide an easy to use overview, the resolution is split into three categories. "L" stands for a technique which requires a lot of screen space like being provided by modern PDAs and handhelds. "M" means that the technique can be used on screens with medium resolution or higher. Devices in blackberry format or modern smartphones can provide sufficient screen space. "S" represents techniques which are not particularly screen consuming, they can be used on hardly any mobile device providing a pixel based screen. beginning with modern mobile phones or standard smart phone sized devices.
- **Data:** The proposed techniques can be classified by the type of data the can manage. Some require very specific data like vector based Maps or simple html-pages. Others can handle hardly every two dimensional graphical data. These techniques are simply marked as "any".
- **Output:** The output represents the achievement of the visualization technique. This is the main performance indicator and therefor has to be checked in detail. Shneiderman defined seven tasks in his "Task by Data Type Taxonomy" by which a visualization can be evaluated [25]. These Are:
 - **Overview:** Does the technique give an overview of the entire data. one part of giving an overview of the data is to provide *Panning*
 - **Zoom:** *Can the user zoom in on designated items or parts of the data*
 - **Filter:** *Is it possible to hide irrelevant information*
 - **Details-on-Demand:** *Does the approach give the possibility to view detail information when needed*
 - **Relate:** *Can the user view relationships among items*
 - **History:** *Does the implementation keep a history of actions to allow undo and replay*
 - **Extract:** *Is it possible to save data which users have obtained by using the application and utilize them for further processing*

Table 1. Matrix overview

S = small, M = medium, L = large

| Technique | Input | Device | Data | Output |
|---|-------------------|--------|-------------------|--|
| Double Scroll Bar | Keys or Touch(2D) | S | Any | Panning |
| Grab and Drag | Touch(2D) | S | Any | Panning |
| Overview and Detail - Large Focus-Display | Touch(2D) | L | Any | Overview, Panning |
| Overview and Detail - ZEN | Touch(2D) | L | Any | Overview, Panning, Zooming |
| Focus and Context - Fisheye | Touch(2D) | L | Any | Overview, Panning, Zooming |
| Tilt-based | Tilt+Key(3D) | M-L | Any | Overview, Panning, Zooming |
| Halo, Arrows, CityLights + ZUI | (Touch(2D)) | M | Annotated maps | Overview, (Panning, Zooming) |
| Map Generalization | not specified | M | Vector based maps | Filter |
| Line Drive | not specified | M | Vector based maps | Filter |
| Grid-based Image Browsing | Touch(2D) | M | Images | Overview, Panning, Zooming |
| Pocket PhotoMesa | Touch(2D) | L | Images | Overview, Panning, Zooming |
| AutoZoom and GestureZoom | Touch(2D) | M | Images | Overview, Panning, Zooming |
| WEST | Keys or Touch(2D) | L | Simple html sites | Overview, Filtering, Details-on-Demand |
| Powerbrowser | Keys or Touch(2D) | M | Simple html sites | Overview, Filtering, Details-on-Demand |
| Web Page Analysis | Touch(2D) | L | Web sites | Overview, Panning, Zooming, Details-on-Demand, History |
| Collapse-to-Zoom | Touch(2D) | L | Web sites | Overview, Panning, Zooming, Filtering, Details-on-Demand, History, (Extract) |

These four criteria have been precisely analyzed for every technique which was presented in chapters 2 to 5 resulting in a matrix overview (see table 1). The data column reflects the structure of this paper. The more general approaches can handle different kinds of data. For example it is also possible to explore a large spreadsheet by using Grab and Drag. The more data-specific approaches are often based on, build a subclass of or use one of the general approaches. For example Halo 3.1 does only address the goal of helping the user to estimate the location of off-screen locations. In the implementation of a prototype, the map navigation can be done by Grab and Drag combined with zooming functionality. By looking at the Device and the Output column, one can also see that mostly the approaches are ordered by their complexity starting with the simple techniques. They accomplish less tasks than more complex techniques, but can often be used on smaller screens.

Techniques which deal with more complex data structures to often provide additional achievements which exceed the ones from the global approaches. For example Line Drive can do a filtering using vector based maps. The vectors provide the possibility to analyze the routes and transform them into an easy to use map which contains only the relevant information. Having only an image map, this task would hardly be feasible. Also the high number of tasks web browsing approaches can offer, highly depends on the complex structure of the web sites data. In addition to Overview, Panning and Zooming the Collapse-to-Zoom technique (see section 5.4) provides Filtering, Details-on-Demand and History tasks. Filtering is done by collapsing irrelevant blocks. Users can navigate to the previous and the next page by using arrow buttons, which results in a History achievement. The nature of web site structure is to provide further information by clicking on a link. So hardly every web browser implements a Detail-on-Demand functionality. Also parts of the Extract task can be identified in Collapse-to-Zoom. The application stores information about the collapsed blocks and provides them to the user when he revisits the site.

7 CONCLUSION

The Matrix (see table 1) shows an overview of techniques which were discussed in scientific papers and researches. Most of them provided an actual implementation which has been evaluated by a small group of people. Especially some basic approaches are already frequently used in every days life. Still many approaches do only exist as prototype implementations and are not used in commercial products. This may be because of the high demand in processing power or the novel interaction techniques like tilt which are provided on most of the

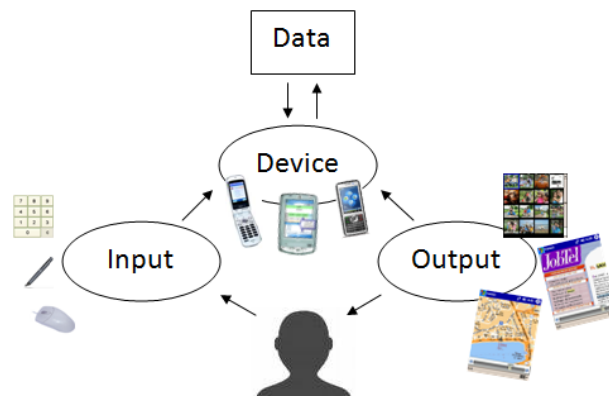


Fig. 13. interaction graph motivating the four criteria

mobile devices. But with ongoing development and the high popularity of mobile hardware, some of the ideas presented in this paper will soon support the mobile user. Other techniques still seem very futuristic. For example LAMP3D [5] is a technique for location-aware presentation of 3D content. Using a GPS-sensor the user can see a 3D visualization of the environment, which he is exploring in the real world on, his mobile device. by tapping on items additional information is requested and pops up on the display. Also points of interest are highlighted and a very intuitive navigation can be provided. Even if many people believe that this is only a scientific project which cant be established, with the automated digitalization of real world data, these scenarios can soon become everyday life.

REFERENCES

- [1] M. Agrawala and C. Stolte. Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 241–249. ACM New York, NY, USA, 2001.
- [2] P. Baudisch and R. Rosenholtz. Halo: a Technique for Visualizing Off-Screen Locations. In *CHI-CONFERENCE-*, pages 481–488. ASSOCIATION FOR COMPUTING MACHINERY INC, 2003.
- [3] P. Baudisch, X. Xie, C. Wang, and W. Ma. Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. In *Proceedings of the 17th annual ACM symposium on User*

- interface software and technology, pages 91–94. ACM New York, NY, USA, 2004.
- [4] S. Björk, L. Holmquist, J. Redström, I. Bretan, R. Danielsson, J. Karlgren, and K. Franzén. WEST: a Web browser for small terminals. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 187–196. ACM Press New York, NY, USA, 1999.
- [5] S. Burigat and L. Chittaro. Location-aware visualization of VRML models in GPS-based mobile guides. In *Proceedings of the tenth international conference on 3D Web technology*, pages 57–64. ACM New York, NY, USA, 2005.
- [6] S. Burigat, L. Chittaro, and S. Gabrielli. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 239–246. ACM New York, NY, USA, 2006.
- [7] S. Burigat, L. Chittaro, and S. Gabrielli. Navigation techniques for small-screen devices: An evaluation on maps and web pages. *International Journal of Human-Computer Studies*, 66(2):78–97, 2008.
- [8] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and T. Winograd. Power browser: efficient web browsing for pdas. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 430–437, New York, NY, USA, 2000. ACM.
- [9] O. Buyukkokten, O. Kaljuvee, H. Garcia-Molina, A. Paepcke, and T. Winograd. Efficient Web Browsing on Handheld Devices Using Page and Form Summarization. *ACM Transactions on Information Systems*, 20(1):82–115, 2002.
- [10] S. Card, J. Mackinlay, and G. Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems (TOIS)*, 9(2):99–122, 1991.
- [11] Y. Chen, X. Xie, W. Ma, and H. Zhang. Adapting Web Pages for Small-Screen Devices. *IEEE INTERNET COMPUTING*, pages 50–56, 2005.
- [12] P. Eslambolchilar and R. Murray-Smith. Tilt-Based Automatic? Zooming and Scaling in Mobile Devices-A State-Space Implementation. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 120–131, 2004.
- [13] L. Harrie, L. Sarjakoski, and L. Lehto. A Mapping Function for Variable-Scale Maps in Small-Display Cartography. *JOURNAL OF GEOSPATIAL ENGINEERING*, 4(2):111–124, 2002.
- [14] M. Harrower and B. Sheesley. Designing Better Map Interfaces: A Framework for Panning and Zooming. *Transactions in GIS*, 9(2):77–89, 2005.
- [15] L. Holmquist. Focus+ context visualization with flip zooming and the zoom browser. In *Conference on Human Factors in Computing Systems*, pages 263–264. ACM New York, NY, USA, 1997.
- [16] P. Irani, C. Gutwin, and X. Yang. Improving selection of off-screen targets with hopping. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 299–308. ACM New York, NY, USA, 2006.
- [17] B. Karstens, R. Rosenbaum, and H. Schumann. Presenting Large and Complex Information Sets on Mobile Handhelds. *E-Commerce and M-Commerce Technologies*, pages 32–56.
- [18] A. Khella and B. Bederson. Pocket PhotoMesa: a Zooming Image Browser for the Pocket PC. Technical report, Technical Report, University of Maryland HCI Lab, 9 pp, 2004.
- [19] A. Nivala and L. Sarjakoski. Adapting Map Symbols for Mobile Users. In *Proc. of the International Cartographic Conference*, pages 9–16, 2005.
- [20] D. Patel, G. Marsden, S. Jones, and M. Jones. An Evaluation of Techniques for Browsing Photograph Collections on Small Displays. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 132–143, 2004.
- [21] U. Rauschenbach, S. Jeschke, and H. Schumann. General rectangular fisheye views for 2D graphics. *Computers & Graphics*, 25(4):609–617, 2001.
- [22] J. Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 167–168. ACM Press New York, NY, USA, 1996.
- [23] G. Robertson and J. Mackinlay. The document lens. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 101–108. ACM New York, NY, USA, 1993.
- [24] R. Rosenbaum and H. Schumann. Grid-based interaction for effective image browsing on mobile devices. In *Proc. SPIE*, volume 5684, pages 170–180, 2005.
- [25] B. Shneiderman. The eyes have it: a task by data type taxonomy for informationvisualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343, 1996.
- [26] B. Shneiderman, S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
- [27] P. Zellweger, J. Mackinlay, L. Good, and P. Baudisch. City lights: contextual views in minimal space. In *Conference on Human Factors in Computing Systems*, pages 838–839. ACM New York, NY, USA, 2003.