Interactive Visual Analysis of Geographic Data on Mobile Devices based on Dynamic Queries

Stefano Burigat and Luca Chittaro

HCI Lab, Dept. of Math and Computer Science, University of Udine,
Via delle Scienze 206, 33100, Udine, Italy

Abstract

The capabilities of current mobile devices, especially PDAs, are making it possible to design and develop mobile applications that employ visual techniques for using geographic data in the field. These applications can be extremely useful in areas as diverse as tourism, business, natural resources management and homeland security. In this paper, we present a system aimed at supporting users in the exploratory analysis of geographic data on PDAs through a highly interactive interface based on visual dynamic queries. We propose alternative visualizations to display query results and present an experimental evaluation aimed at comparing their effectiveness on a PDA in a tourist scenario. Our findings provide an experimental confirmation of the unsuitability of the typical visualization, employed by classic dynamic query systems, which displays only those results that fully satisfy a query, in those cases where only sub-optimal results are obtainable. For such cases, the results of our study highlight the usefulness of visualizations that display all results and their degree of satisfaction of the query.

Key words: Visualization, geographic information systems, mobile devices, dynamic queries.

1 Introduction

A relevant portion of all digital data generated every year by the information society concerns geographic datasets (MacEachren and Kraak, 2001) and gaining useful knowledge from these datasets is fundamental in many domains.

Email addresses: burigat@dimi.uniud.it, chittaro@dimi.uniud.it (Stefano Burigat and Luca Chittaro).

While automatic analysis methods based on statistics and artificial intelligence can be used to extract such knowledge, the analysis process is often too complex for users to understand and control and each method can only lead to the discovery of a limited amount of specific knowledge. Thus, information visualization techniques are often used as an alternative or a complement to automatic methods, presenting abstract data visually and exploiting the abilities of human perception to allow users to explore complex information spaces and obtain valuable knowledge (Keim, 2002). By allowing users to easily and directly interact with data, the visual data exploration process becomes fast and intuitive and increases confidence in the findings.

While visual data exploration techniques have been studied so far only in desktop scenarios, the continuous evolution of mobile computing devices is opening up new possibilities. Advances in mobile communication technology significantly affect the nature and availability of geographic data, since mobile devices are an increasingly important means of accessing information as well as a source for geo-referenced data in the field (Dykes and Mountain, 2003). Extending visual exploration and analysis tools and techniques to the mobile scenario would allow to support user's decisions exactly where and when they are needed. Effective visualizations could allow users to easily access and understand the information that is relevant for their tasks and proper interactive techniques could allow them to explore geographic data and take decisions in the field.

Unfortunately, the mobility context and technical limitations constrain the design and development of mobile visualization and interaction solutions, making it impossible to simply port applications from desktop computers to mobile devices (Chittaro, 2006). In particular, geovisualisation on mobile devices is affected by restrictions such as the small size of displays, limited CPU power and storage capabilities, battery life, slower connectivity and limited input peripherals.

In this paper, we present MAGDA (*M*obile *A*nalysis of *G*eographic *Da*ta), an application aimed at supporting users in the exploration and analysis of datasets of geo-referenced elements on PDAs. Some preliminary ideas for this application were formulated while we were studying the problem of supporting tourists in searching for specific points of interest (POIs) such as hotels and restaurants in a geographic area (Burigat et al., 2005). Then, we extended the scope of our research with the aim of developing a more general system to deal with geo-referenced multi-attribute elements (e.g., cities, mines, pollution sources, archaeological sites, etc.). MAGDA allows users to query datasets through a highly interactive interface and provides different visual representations to display query results and to help gain useful insights into the data. As a practical example, we will illustrate how MAGDA can be used with two case studies respectively concerning: (i) the tourist domain, to support tourists in searching for POIs that best satisfy their preferences, and, (ii) the soil science domain, to allow researchers to retrieve information about soil chemical properties in a geographic area. We will then present the results of an experimental evaluation aimed at comparing the three different visualizations of query results provided by MAGDA, to highlight their strengths and weaknesses.

The paper is organized as follows. Section 2 surveys related work on the visualization and analysis of geographic data on mobile devices. Section 3 introduces MAGDA, describing its features in detail. In Section 4, we illustrate the use of MAGDA in the two above mentioned case studies. Section 5 describes the experimental evaluation of MAGDA visualizations and the results we obtained. Section 6 concludes the paper by presenting future work.

2 Visualizing Geographic Data on Mobile Devices

In the latest years, research effort on employing geographic data on mobile devices has increased. In general, research has mostly focused on how to best represent and interact with mobile maps, and *mobile tourist guides* (Baus et al., 2005) have been a particularly studied application.

Several authors have proposed approaches for adapting maps to mobile devices. In particular, since a map is always strongly related to the context of its use, various contextual elements (e.g, user, location, task, device) have been considered to perform the adaptation so that the user could get what is most suitable for her needs. For example, Chalmers et al. (2001) have studied how to adapt map-based applications to variations in display specifications, network quality, and user's current task, while Reichenbacher (2004) has focused on how to personalize the presentation of spatial information according to the current user, context and device. Zipf (2002) provides an overview of the design steps involved in adaptive map generation and introduces the concept of *focus map*. Focus maps are based on subdividing maps into different regions of interest and displaying each region with a different amount of detail according to its degree of interest for the user (Zipf and Richter, 2002). Similarly, *fisheye views* (Rauschenbach et al., 2001) and *variable-scale maps* (Harrie et al., 2002) display only a specific area of a map in full detail, while surrounding areas are distorted to fit the available screen space. One of the design steps identified by Zipf (2002) and investigated by different authors is shape simplification through generalization. Generalization is a graphic and content-based simplification of the data presented on a map that aims at abstracting irrelevant details to reduce the cognitive effort of the user, and at simplifying the process of creating a lower scale map from a detailed one. For example, Agrawala and Stolte (2001) developed some techniques for the generalization of cartographic data that improve the usability of maps for road navigation on mobile devices: by distorting road lengths and angles and simplifying road shape, they show that it is possible to clearly and concisely present all the turning points along the route in less screen space.

Many researchers have focused on mobile tourist guides (Abowd et al., 1997; Cheverst et al., 2000), which are applications that exploit information such as user position, place, current time and task, to provide tourists with information and services related to a specific geographic area. Most mobile tourist guides display maps with superimposed information, such as paths or POIs, and allow users some (limited) interaction through panning, zooming, search and selection operations on maps. In some of these projects, researchers have experimented with perspective (Gartner and Uhlirz, 2001) or 3D visualizations (Laakso et al., 2003; Burigat and Chittaro, 2005) of geographic data to increase the quantity of data displayed on the same screen, and take advantage of users' natural spatial abilities.

Much less effort has been devoted to the design and development of theories, methods and tools for visual exploration and analysis of geographic data on mobile devices. Major Geographic Information Systems (GIS) companies, such as ESRI (2006) and Trimble (2006), are proposing some GIS functionalities for the mobile context, providing users with stripped-down versions of their mainstream products that can be tailored to support data acquisition in the field, e.g., archaeological surveys (Tripcevich, 2004). However, more powerful solutions that could take into account the mobile context more specifically and employ novel visual techniques in it have yet to appear. A step in that direction has been made by Lodha et al. (2003), who have focused on supporting mobile users in visually accessing and querying GIS datasets. They offer users a variety of queries (e.g., how far, where, closest) for many different types of geometric primitives (e.g., points, lines, polygons) and elements (e.g., buildings, metro stops). Users perform queries by directly interacting with the displayed geographic data. For example, a user can select a building on an aerial map, query for nearby buildings in that area, and receive information such as the building name in a schematic view. The user can draw points, lines, and arbitrary polygons on the display, and use these primitives as input to queries. For example, the user can draw two polygonal regions and find buildings contained within their intersection. User's location obtained from GPS can also be used as input to queries, for example to locate the nearest metro stop and telephone, highlighting buildings close to the user's path.

3 The MAGDA System

MAGDA brings *dynamic queries* (Ahlberg et al., 1992; Shneiderman, 1994) to mobile devices to support users in the analysis of datasets of geo-referenced elements. Dynamic queries are typically used in desktop systems to explore large datasets, providing users with a fast and easy-to-use method to specify queries and visually analyze their results. The basic idea of dynamic queries is to combine input widgets, called "query devices" (Ahlberg and Truvé, 1995), such as rangesliders, comboboxes, check buttons, with graphical representations of results, such as maps, scatterplots (Ahlberg and Shneiderman, 1994) or other visual displays. By directly manipulating query devices, users can specify the desired values for the attributes of elements in a dataset and can easily explore different subsets of the data. Visual results are rapidly updated to enable users to learn interesting properties of the dataset as they play with the query devices. Dynamic queries have been successfully employed in domains such as real estate (Williamson and Shneiderman, 1992) and tourism (Dunlop et al., 2004) and, as shown by some user studies (Ahlberg et al., 1992; Williamson and Shneiderman, 1992), are a very usable and powerful technique.

3.1 Interacting with MAGDA

MAGDA combines dynamic queries with specific visualizations of query results, in an interface that aims at facing the screen estate problem typical of mobile devices.

3.1.1 Interface overview

MAGDA displays elements of the explored datasets as icons superimposed on the map of the considered geographic area, in their correct georeferenced position. Each dataset has an associated icon that is used for all its elements on the map.

In the current version of the system, datasets are stored as XML files and can be loaded through the "File" menu at the bottom of the screen (see Fig. 1). The "Datasets" menu allows users to choose which ones of the loaded datasets should be displayed on the map. After dataset selection, the icons associated to the selected datasets appear in the toolbar in the lower part of the screen and the map is populated with the corresponding elements (see Fig. 1). Users can also change the currently displayed map (provided multiple maps are associated to the same geographic area by the XML files, e.g., a topographic map, a road map, a geologic map, ...) through the "Options" menu.

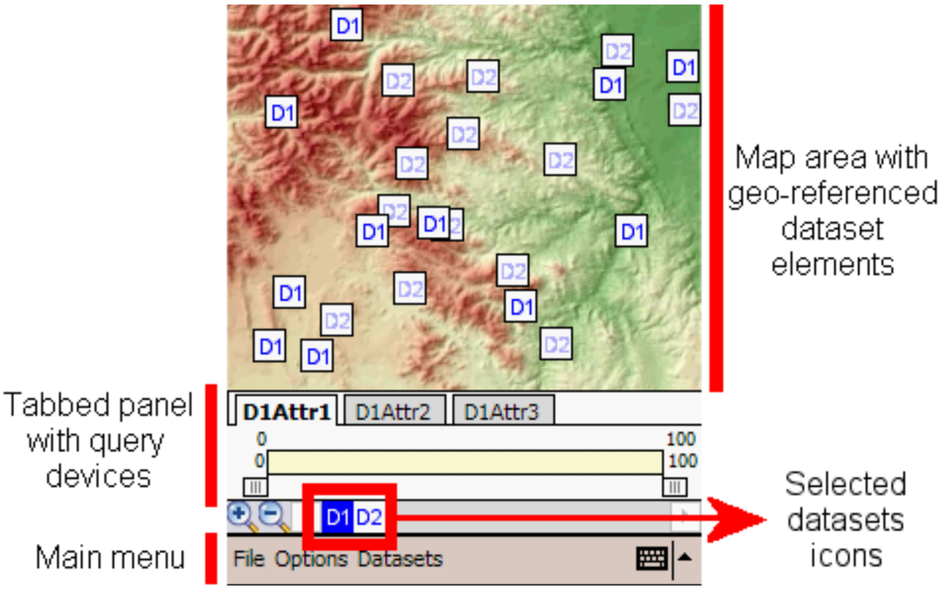


Fig. 1. The map displays all elements of two datasets (generically called D1 and D2 in the figure) selected by the user. A tabbed panel contains all query devices related to the currently explored dataset (D1), which is highlighted in the toolbar at the bottom of the screen.

The interface displays query devices and query results at the same time, allowing users to easily specify conditions on different attributes for each dataset. Query devices are organized in a tabbed panel, placed above the toolbar in the lower part of the screen, and are displayed when the user taps on a dataset icon on the toolbar or on any icon on the map. Each tab allows users to specify values for a single attribute of the considered dataset. While only three attributes are considered in the following examples, the system can manage an arbitrary number of attributes. The number of tabs in the tabbed panel is increased accordingly and the user can move through them by clicking on two arrows that appear in the tab area. Figure 1 shows an example where elements of two generic datasets “D1” and “D2” are shown on the map, and it is possible to specify values for the attributes “D1Attr1”, “D1Attr2” and “D1Attr3” of dataset “D1” by accessing the appropriate tabs. Dataset “D1” could, for example, represent hotels (with attributes such as room price, number of stars and services), while dataset “D2” could represent museums. It must be noted that elements belonging to the currently explored dataset are highlighted (by shading all other elements) to increase their visibility. The interface layout gives users the flexibility to specify conditions in any order while visualizing results at the same time, thus simplifying the exploratory analysis of datasets.

3.1.2 Query devices

Various query devices are commonly used in dynamic query systems to specify queries. In MAGDA, we implemented four types of query devices that can be operated with the stylus on the touch screen and are used to deal with different types of attributes.

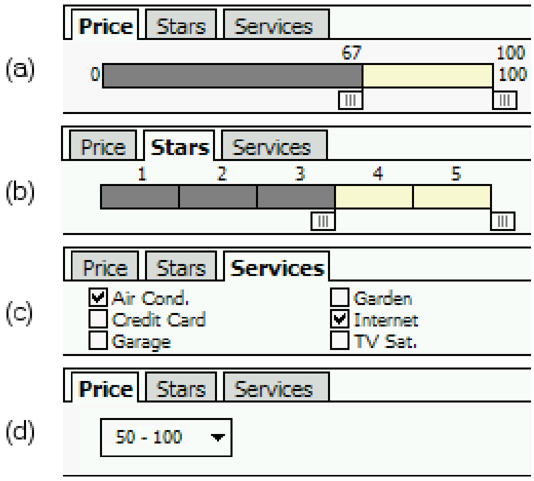


Fig. 2. Query devices in MAGDA: (a) rangeslider, (b) discrete rangeslider, (c) checkboxes, (d) combobox.

The first device is the classic rangeslider, usually associated with continuous attributes (e.g., price, temperature, etc.): the user acts on two independent handles to change the range of values of the related attribute. An example of this query device is shown in Figure 2(a), where a range between 67 and 100 has been specified for the “Price” attribute of a hotel single room. The selected range is highlighted with color and by showing the numeric value of the two bounds.

The second query device, called “discrete rangeslider”, is a modified rangeslider that can be used to deal with ordinal values (e.g., the number of stars of a hotel as illustrated in Fig. 2(b)). This query device can be used as the classic rangeslider but when the user stops dragging one of the two handles, the handle is automatically snapped to the nearest lower limit (for the lower handle) or the nearest upper limit (for the upper handle).

The third query device is based on checkboxes and can be used mainly for multiple-choice queries (e.g., services offered by a hotel). An example is shown in Fig. 2(c) where “Air conditioning” and “Internet connection” have been checked as required for the “Services” attribute. This query device can also be used as an alternative to the discrete rangeslider.

The fourth query device is based on a combobox (Fig. 2(d)) and can be employed as an alternative to other query devices, in particular the rangeslider, when the user can choose among pre-defined ranges of values and does not need to specify a custom range.

MAGDA can automatically select the right query device to use according to the type of attribute by considering if the attribute is continuous, ordinal, or non-numeric. The measurement units (e.g., meters, euros, ...) for each attribute are not shown in the query devices to save screen space, but they can be displayed beside the name of the attribute by activating this possibility in the "Options" menu that can be accessed through the main menu.

3.1.3 Details-on-demand

Users can obtain detailed information about each element by tapping on its corresponding icon on the map, and the icon of the currently selected element on the map is highlighted in blue. Attribute values for the selected element are then visualized inside query devices by using blue horizontal bars inside sliders (Fig. 3(a)(b)) and by displaying blue boxes beside checkboxes and comboboxes (Fig. 3(c)(d)). This functionality aims at making the system easier to use by enabling rapid and contextual access to information which is usually more difficult to obtain in traditional systems and can be used to better compare user's specified values with specific query results. Further information about the selected element (e.g., name, geographic coordinates, telephone number, address, etc.) as well as an overview of all its attribute values can also be retrieved by tapping on a specific tab named "Info".

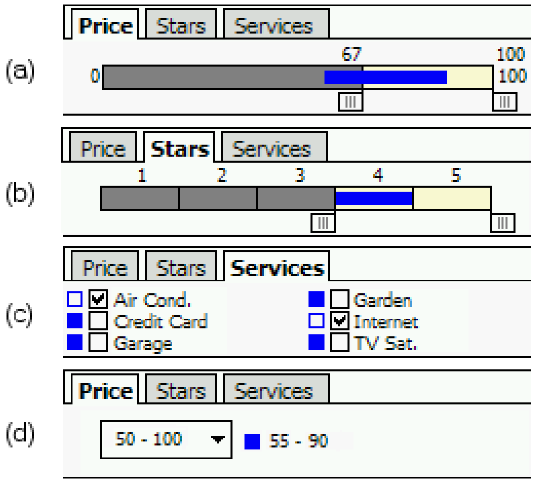


Fig. 3. Details on attribute values for each element are provided by using blue horizontal bars inside sliders (a)-(b) and blue boxes beside checkboxes and comboboxes (c)-(d).

3.1.4 Visualization of query results

MAGDA supports different visualizations of query results that can be selected in the "Options" menu. The first visualization is typical of classic dynamic query systems and is based on displaying all and only the elements of the explored dataset that satisfy the current query (hereinafter, we call it *on-off* *visualization*). Thus, while users manipulate query devices to specify attribute values, icons representing elements that satisfy (or not) the query are turned on (or off), providing perceptual feedback to allow users to understand the effects of changes in the query (see Fig. 4).

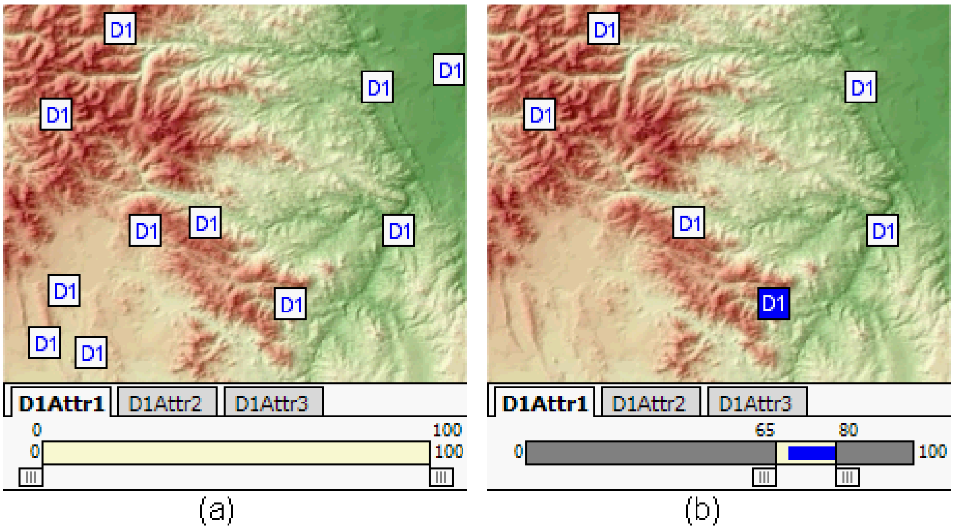


Fig. 4. On-off visualization of query results. On the left: no condition has been specified for attributes and all elements of dataset D1 are visualized. On the right, a specific range of values has been specified for one attribute and only elements satisfying that condition are visualized.

While the on-off visualization is suitable to quickly obtain visual feedback from the system, it also makes it difficult to extract all useful information from geographic data. Indeed, as users sequentially manipulate different query devices, they can only see those elements that satisfy all current conditions, thus losing sight of the whole dataset they are exploring (Spence, 2001). As a consequence, it is impossible to quickly determine how elements are distributed in an area according to how much they satisfy a query. This prevents users, for example, from easily identifying suboptimal results (which may be particularly important when a query produces only a few or no results at all) or finding out interesting patterns, trends or anomalies in the data that may prompt further investigations in the field (such as elements that fully satisfy a query surrounded by elements that do not satisfy it).

For the above reasons, we designed alternative visualizations of query results that allow users to obtain contextual information on the whole dataset they are exploring.

The first alternative visualization, called *bar visualization*, augments all icons on the map with a vertical bar that shows how much each element satisfies the user's query (Fig. 5). This technique is an elaboration of an idea presented by Fishkin and Stone (1995) who introduced the concept of "real-valued queries" by assigning a score in the [0,1] range to each element in a dataset (based on

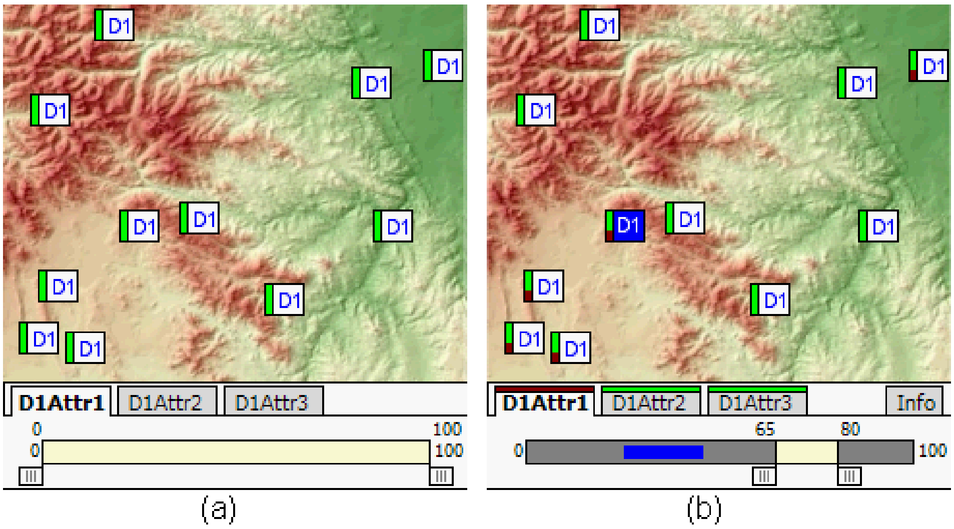


Fig. 5. Bar visualization of query results. Each icon is augmented by a vertical bar showing how much the corresponding element satisfies the user's query. Users can visually perceive the effects of their queries by observing changes in the color-filled areas of bars while manipulating query devices.

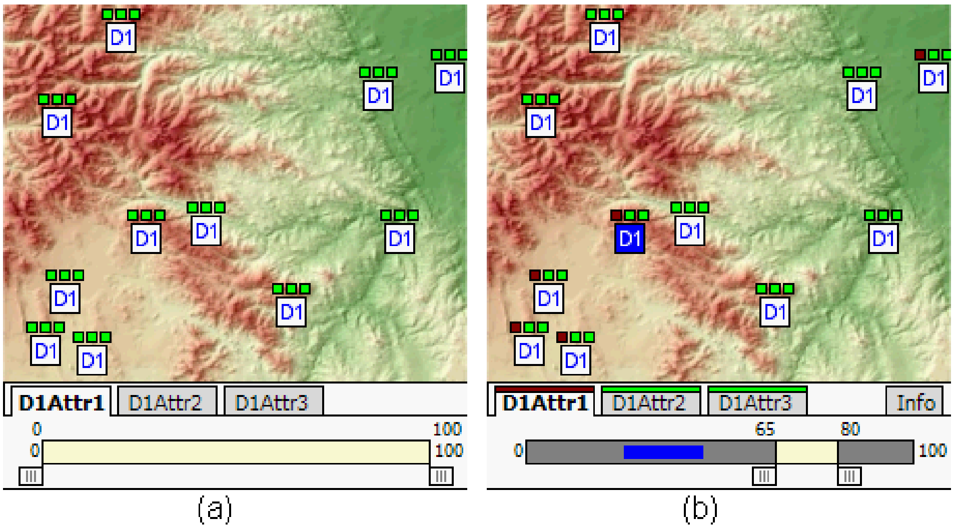


Fig. 6. Led visualization of query results. A set of boxes above each icon provides information about query satisfaction for each attribute.

the values of the element attributes and on a specific scoring function) and then visually presenting it by showing each element as a partially filled-in bar (the higher the score, the more the bar is filled). Instead of visually displaying a score dependent on the value of an attribute, we display a score dependent on the number of conditions satisfied by each specific element. By default, we then divide the bar associated with each element into a green part whose height is proportional to the number of satisfied conditions, and a red part whose height is proportional to the number of unsatisfied conditions. Using this visualization, it is easy to visually track elements that fully satisfy a query (i.e., those with a fully green bar) as well as to visually compare how much different elements satisfy the specified set of conditions, thus possibly identifying interesting features of the dataset.

With this visualization, it is also possible for users to precisely assess which conditions are satisfied by each element. Indeed, when tapping an icon on the map, a color coded line over each attribute name in the query interface (as shown in Fig. 5(b)) points out if the selected element satisfies (green line) or not (red line) the condition on that attribute. Without this feature, a user would have to go through each query device and look at the detailed attribute values of elements.

Figure 6 shows the second alternative visualization, called *led visualization.* This visualization employs a set of small colored boxes (the "leds") above each icon to show which conditions are satisfied by the corresponding element. Each led is directly associated to a specific attribute and tells if the condition specified by the user on that attribute is satisfied (green) or not (red). This visualization makes it easier to identify which specific conditions are satisfied or not for each element (with the bar visualization, users need to tap on icons and look at the query interface to obtain this information). By being able to check all attributes separately, users can also perform interesting visual analyses (for example, identify all clusters of elements having similar properties).

It is also worth noting that the bar as well as the led visualization allow users to identify elements satisfying more complex queries than the simple AND of all conditions on attributes (which is the standard behavior of dynamic query systems). For example, elements with a green part on their bars or a green led satisfy the logic OR of all set conditions.

The two visualizations have also drawbacks. In particular, by providing information on whole datasets at the same time, they may suffer more than the on-off visualization from cluttering and icon overlapping issues when large numbers of elements are visualized on a limited area. This could make it more difficult for users to compare results of queries. Furthermore, an icon on the map in the led visualization requires a portion of screen space which is proportional to the number of attributes, making the technique less and less practical as the number of considered attributes increases.

3.2 Managing off-screen elements

A significant problem of map-based visualizations on mobile devices is that elements often disappear into off-screen space as users perform panning and zooming operations. This can hide relevant information, preventing users from maintaining an overview on all data and making it harder to perform spatial tasks (for example, a user may rely on well-known regions of a map to orient herself in the field).

Several techniques proposed in the literature to mitigate this problem are adaptations of existing approaches to view large documents on desktop systems. For example, *Overview&Detail* visualizations (Plaisant et al., 1995) are based on simultaneously displaying a detailed view of a map area and an overview (usually at a reduced scale) of the whole map, while *Focus&Context* visualizations, such as the Rectangular FishEye-View (Rauschenbach et al., 2001), exploit only a single view of the map with higher detail in the focus area and lower detail elsewhere. The disadvantage of these approaches is that the different scales and the distortions introduced make it more difficult for users to integrate map information into a single mental model and interfere with any task that requires precise geometric judgments (Baudisch et al., 2002).

To provide information on elements that fall outside the currently displayed area on the screen, we employ a technique that does not involve scaling or distortion operations. Halo (Baudisch and Rosenholtz, 2003) is one such technique described in the literature. Halo shows "off-screen" elements locations by surrounding them with circles that are just large enough to reach into the border region of the display window. From the arcs visualized on-screen, users can derive the off-screen location of the object. In Halo, however, the portion of screen border occupied by an arc grows as the location of the corresponding object is further away, so that more distant elements (which are in many situations less significant for the user) are associated to wider arcs. Moreover, it takes only one far enough object to fill one of the four border regions of the screen and scenarios involving a large number of off-screen elements can thus easily get cluttered.

The solution we implemented is instead based on displaying arrows that point towards off-screen elements and are placed on the border of the currently visualized map area (Fig. 7). Arrow orientation can be used to derive the direction of off-screen elements with respect to the currently displayed area. When the on-off visualization is used, only arrows pointing towards fully satisfied elements are displayed. When the bar or led visualization are used, arrow color conveys detailed information on how much each object satisfies the user's query. By default, specific colors are mapped to specific ranges of satisfaction (green is used for fully satisfied elements, yellow, orange and red are then used for decreasing satisfaction levels).

Arrows are interactively updated as users perform zooming and panning operations. Since this is a computationally demanding operation on a mobile device and also to reduce visual complexity, MAGDA allows users to track only off-screen elements within a certain distance from the center of the visu-

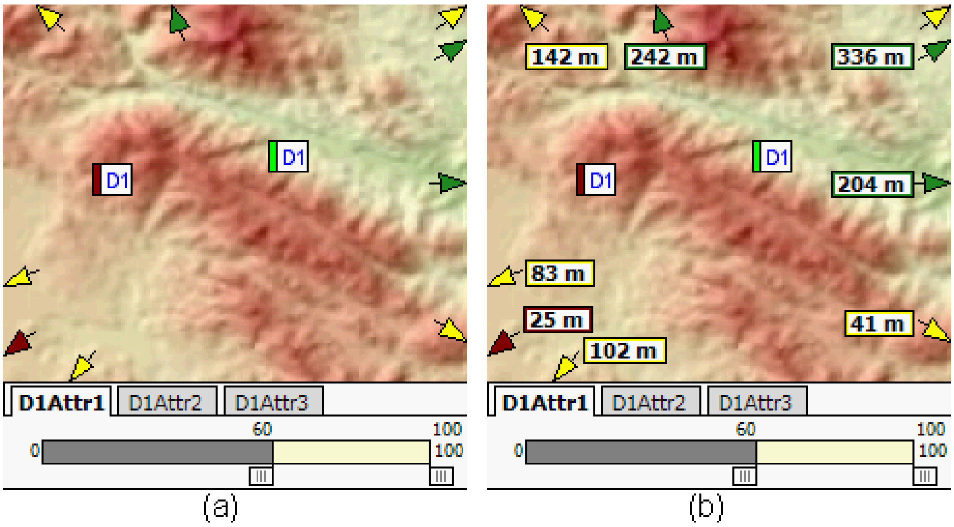


Fig. 7. Small arrows are used to point towards off-screen elements. Arrow color conveys information on how much each object satisfies the user's query. On the right: textual labels provide information about the distance of elements.

alized area or with a satisfaction level higher than a certain threshold.

Currently, we employ textual labels attached to arrows to provide information about the distance of elements. To reinforce the association between labels and their respective arrows, we use arrow color for the border of each area containing the label, as shown in Figure 7(b). Text labels can be switched on or off independently of arrows because they tend to occupy relatively large portions of the display area and thus it is convenient to turn them off when they are not needed.

In a recent work (Burigat et al., 2006), we have specifically evaluated the possibility to map the distance of off-screen elements on geometric features of arrows (e.g., their size or length) to avoid the use of labels.

4 Application Examples

The general approach to visual analysis employed by MAGDA allows one to easily exploit the tool in different domains. Since query devices are automatically generated according to the type and values of attributes, using MAGDA in a specific domain is simply a matter of defining the datasets and corresponding graphical resources (i.e. maps and icons) through XML files. In the following sections, we illustrate how MAGDA can be used to support users having very different needs, on two case studies concerning respectively the tourism and the soil science domain.

4.1 Preference-based search for tourists

In the tourist domain, we have used MAGDA as a Preference-Based Search Tool (PBST), which is a decision support system that helps users in finding products or services that best satisfy their needs and preferences. In particular, we have employed it to filter data about different datasets of POIs (e.g., hotels, restaurants, pubs, museums, parks, etc.) in a geographic area, supporting users in the search of the most suitable solution to their needs (e.g., a POI or combination of POIs satisfying a set of preferences). Current PBSTs for mobile devices are still very limited in terms of flexibility and capabilities and usually allow users to search for very specific POIs in a city (usually hotels or restaurants) by entering their preferences through drop-down lists and checkboxes and looking at results on ranked lists of textual data.

Unlike current mobile solutions, MAGDA can visually manage multiple POI datasets at the same time, independently of the number or type of attributes characterizing each dataset. Users are thus able to specify separate queries for each dataset, observe results on the map and identify the best POIs according to their satisfaction level, their specific position in the examined area and also proximity among POIs of different datasets.

As an example, if a user needs to find a hotel to stay overnight and a suitable restaurant nearby, she can start by specifying her preferred values for hotel attributes (e.g., price, number of stars and services). If different results satisfy her query, she can base her choice on their geographic position or their attribute values as provided by the details-on-demand feature described in Section 3.1.3. On the other hand, if no result fully satisfies her query, she can try to relax some conditions on attributes, or directly check suboptimal results by exploiting the bar or led visualizations. Similarly, the user can define a suitable query for the restaurant dataset, by setting appropriate values for attributes such as price range, quality and opening days through the corresponding set of query devices. By inspecting results, the user is able to identify restaurants satisfying all her conditions, but she can also examine partially satisfied restaurants to find results that provide a good tradeoff between distance from the chosen hotel and restaurant attributes (especially when no result fully satisfies a query). Figure 8(a) illustrates a possible outcome of a search involving hotels and restaurants using the on-off visualization. In this case, two hotels and two restaurants fully satisfy the specified query. Fig. 8(b) illustrates a case where no hotel fully satisfies the query and the bar visualization allows tourists to find suboptimal results.

A preliminary prototype of this PBST application that used only the bar visualization has been informally tested on a small group of users (Burigat et al., 2005). The study pointed out some problems with the interface and

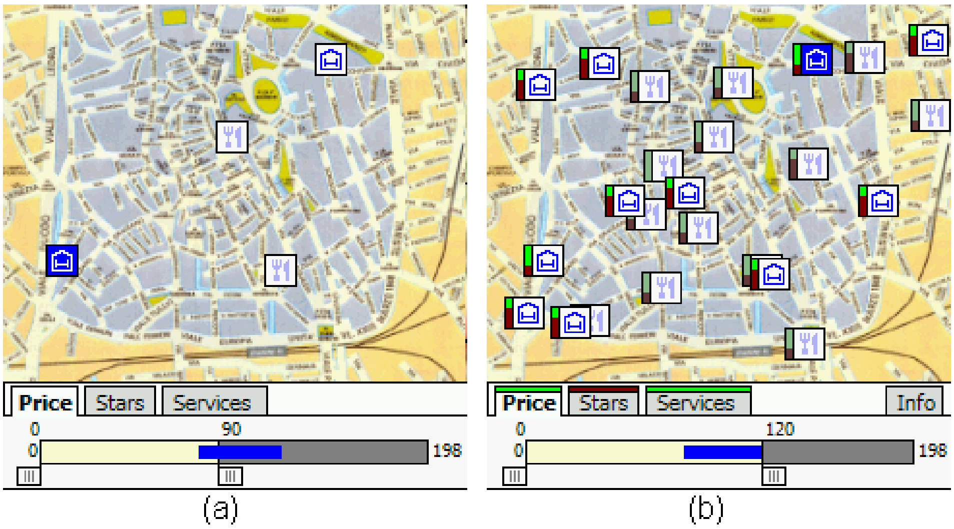


Fig. 8. Examples of the use of MAGDA in the tourist domain: (a) the on-off visualization shows that two hotels and two restaurants satisfy the specified query; (b) in a case where no hotel fully satisfies the query, the bar visualization allows tourist to find suboptimal results.

allowed us to plan improvements that have been subsequently integrated in the current version of the system. However, most of the system features were rated very positively by users.

4.2 Exploratory analysis of soil chemical properties

To explore the application in a completely different domain, we used MAGDA to analyze a dataset concerning soil chemical properties in a geographic area.

Each element in the dataset is a geo-referenced soil probe, and is characterized by three continuous attributes: pH, salinity and cation-exchange capacity (CEC). By manipulating query devices associated with these attributes and looking at the results on the map of the considered geographic area, one is able to visually identify patterns, outliers and clusters in the data. The distribution of values for each attribute can be studied by continuously varying the range of required values through rangesliders. Figure 9 illustrates an example of usage with the on-off visualization where the user progressively reduces (by increasing the lower limit) the required range of values for the pH attribute. As one can easily notice from the effects of the slider manipulation, probes on the left have the lowest pH values, while those in the center have the highest pH values. Moreover, most probes have pH values in the range [6-9], with no probe having a pH value lower than 6, and one probe having a very high pH value (possibly prompting researchers in the field to perform further investigations).

The ability to check different attributes at the same time also allows users to

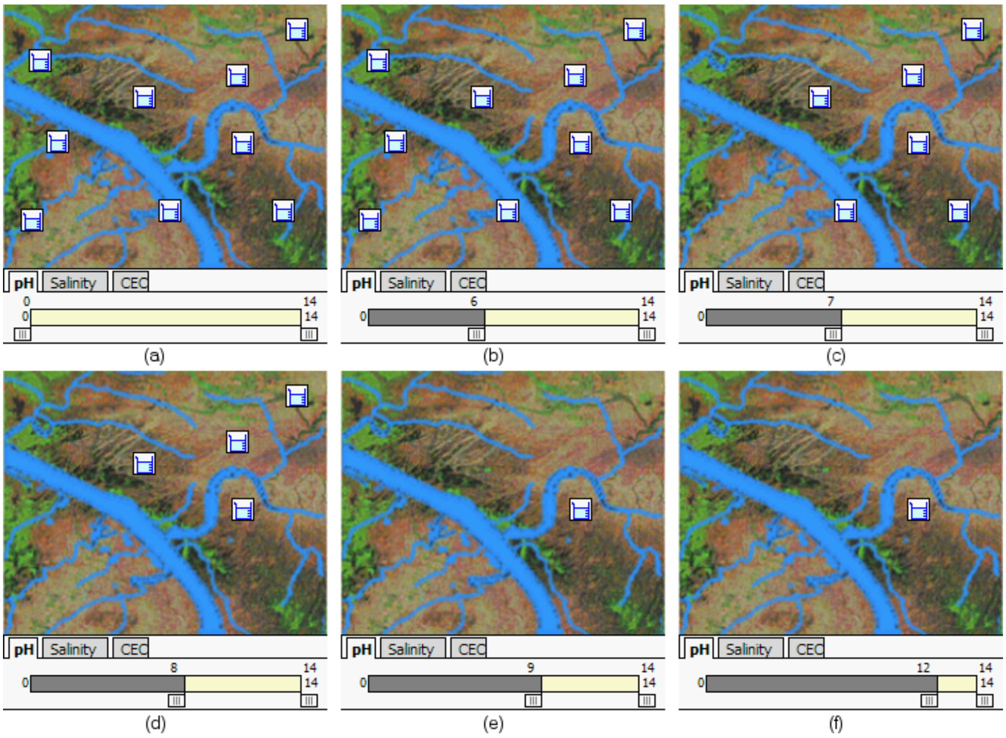


Fig. 9. Using rangesliders to study the distribution of elements according to their pH values.

perform more complex analyses. For example, it is possible to focus only on probes with acidic pH values (specifying a [0-7] range for the pH attribute) and explore what is the distribution of salinity values for these probes, thus identifying interesting or anomalous patterns on this specific subset of the data.

Other analyses can take advantage of the bar and led visualizations. For example, since there are known correlations between soil chemical properties and vegetation, it is possible to identify what probed areas are the most suitable for specific plant species. Figure 10 shows a possible scenario: a 60-200 interval has been specified for the CEC property (these are typical CEC values for an organic soil type), a 5.5-7.5 interval for the pH property (optimal pH values for plant growth) and a 0-4 interval for the Salinity property (values above 4 may restrict the growth of many plants). As shown by the bar visualization, areas on the left are more suitable for most plants compared with areas on the right. However, while in the field, researchers may be interested in exploring areas that are only partially suitable for plants (e.g., to acquire data on plant species that are able to survive in those areas). Furthermore, if interesting phenomena are identified (e.g., a probe with a totally different satisfaction level with respect to all other probes in the same area), they may prompt researchers to perform more accurate investigations.

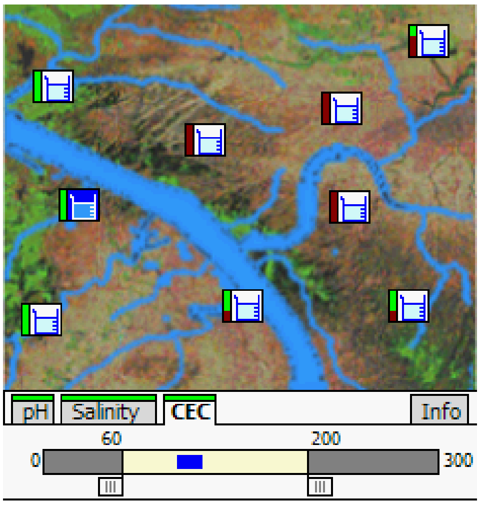


Fig. 10. Analyzing correlations between soil chemical properties and vegetation.

5 Experimental Evaluation of MAGDA Visualizations

To compare the three different visualizations of query results provided by MAGDA, we carried out an experimental evaluation. In particular, we tested if the differences in the way the three visualizations provide information about query satisfaction could affect users' performance in terms of time, errors and number of interface actions required to identify specific elements in a dataset.

5.1 Participants

The evaluation involved a sample of 15 users (9 male and 6 female). They were undergraduate students or graduates from different fields (8 Computer Science, 1 Physics, 1 Engineering, 2 Economics, 2 Humanities). Their age ranged from 21 to 26, averaging at 22.4. Most users (11 out of 15) had never used a PDA before, and had at least some basic experience in using maps (13 out of 15) and web-based tourist search engines for searching POIs (12 out of 15) on desktop PCs.

5.2 Tasks

We designed three types of tasks of increasing complexity. They concerned a scenario where users played the role of tourists who used MAGDA to find specific hotels in a city area. Each task required users to specify appropriate values for one or more of the attributes that characterized hotels in the datasets. The considered attributes were price of a room, number of stars of the hotel and services offered (e.g., air conditioning, Internet connection, etc.). The attributes were respectively associated to a continuous range slider (price), a discrete rangeslider (stars) and a group of checkboxes (services). All three tasks also required users to choose hotels that were located at precise geographic locations on the considered maps. The three tasks were the following:

- "Find the less expensive 4-star hotel in the city area where most restaurants are located". This task requires one to set the number of stars attribute to 4 and to progressively increase the lower price value and identify the first hotel that fails to satisfy the price condition and is located near most of the displayed restaurants.
- "Find all the 3-star hotels with the highest number of available services in the city centre". In this case, one can set first the number of stars attribute to 3 and then check services checkboxes until the hotels with the highest number of available services located in the city centre are identified.
- "Find all the seafront hotels that satisfy the highest number of the following conditions: 3-star quality, room price lower or equal than 120 Euros , credit card payment, garage and air conditioning". This task requires users to set the price to a 0-120 Euros range by lowering the upper handle of the price rangeslider, set the number of stars attribute to 3, check the credit card, garage and air conditioning checkboxes for the services attribute and finally find the target hotels only among those located on the seafront. If no hotel satisfies all the conditions on attributes, one has to relax some of the conditions, e.g. by increasing the price range or unchecking some of the required services.

The second and the third task were repeated twice, once with a dataset containing at least one hotel that satisfied all conditions and once with a dataset containing no hotel satisfying all conditions so that only suboptimal results were obtainable. The total number of tasks users had to carry out was thus 5. In the following, we will identify the first task as Task1; the version of the second task with optimal results as Task2-Opt and the version with with only suboptimal results as Task2-Sub; the version of the third task with optimal results as Task3-Opt and the version with only suboptimal results as Task3-Sub.

5.3 Materials

The experiment was carried out on a 624Mhz PocketPC with a 3.5" display and QVGA (320x240) resolution. Fifteen city maps and 15 POIs configurations were used in the experiment (3 maps and 3 configurations for each task). The three maps used in Task3-Opt and Task3-Sub concerned seaside cities and displayed at least a coastal area. We took care of producing configurations of equal complexity in terms of number of POIs and their distribution. Figure 11(a) shows a sample of a city map and initial POI configuration for Task1 with the on-off visualization, while Fig. 11(b) shows a sample of a seaside city map and initial POI configuration for Task3-Opt with the led visualization.

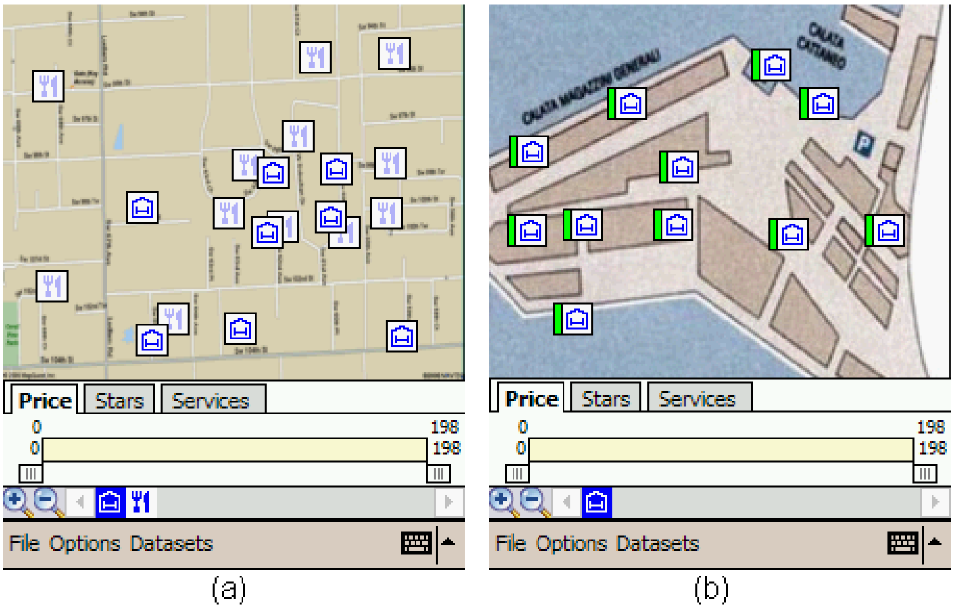


Fig. 11. Two samples of the maps and initial POIs configurations used in the experiment.

5.4 Procedure

The experiment followed a within-subjects design (every participant was presented with every experimental condition). Participants were initially briefed by the experimenter about the nature of the experiment and were then asked to fill a questionnaire asking about their age, gender, occupation, computer experience, PDA experience and web-based tourist search engines experience. Then, they were provided with an introduction and demonstration of MAGDA and the available visualizations of query results. This was followed by the presentation of training tasks for familiarizing with the system. Training tasks were similar to those used for testing (described in the Tasks section) but required users to specify different conditions on attributes. During the training tasks, participants were allowed to ask any question to the experimenter to clarify possible doubts concerning the interface, the visualizations or the tasks. Then, users carried out the experimental tasks. The description of each task was provided through printed task sheets. For each task, participants were asked first to read the task description on the printed sheet, then (when ready) to tap on a "Start Task" button that was initially displayed on the PDA and perform the required task. When participants believed to have found all the targets, they had to tap on them and verbally inform the experimenter. This was needed to allow us to distinguish between explorative taps and target selection taps. The task finished when participants tapped on the last target they identified. At the end of the experiment, participants were briefly interviewed to collect their comments and were asked to tell which visualization they preferred.

All possible care was taken to counterbalance learning effects due to repetitive testing:

- Every user was presented with a different order of the tasks and a different order of the experimental conditions.
- Three different configurations of POIs were produced for each task. The number of initial items in the considered dataset and the number of targets for each task was kept constant.
- There was no fixed association between experimental conditions, map and target configuration. This way, a condition could not benefit by possibly unaccounted factors that might make a target configuration easier to complete than others.

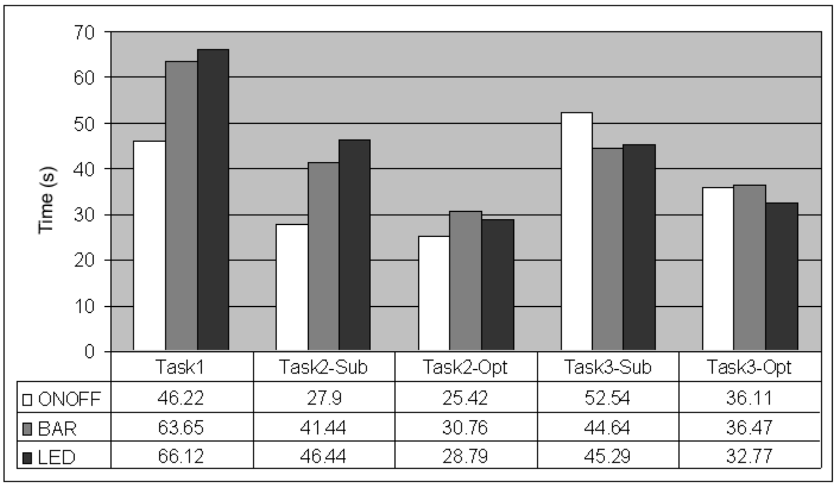
Logging code automatically recorded the time needed by users to complete each task, defined as the time elapsed between the tap on the "Start Task" button and the last tap on a POI, and the number of distinct user interface actions to carry out the task. A "widget use" action was recorded when users interacted with a slider or a checkbox. A "POI selection" action was recorded when users tapped on a POI on the displayed map. A "tab switch" action was recorded when users changed the currently selected tab in the tabbed interface.

5.5 Results

5.5.1 Task completion times

The times needed by users to complete the tasks were subjected to the Kolmogorov-Smirnov test of normality prior to further analysis. The test revealed a moderate degree of non-normality due to the presence of outliers. Since these outliers were legitimate values and were not the result of external factors (e.g., technical faults, uncorrectly understood tasks, etc.), we performed a logarithmic transformation of the data to reduce their impact and make the distribution more symmetric, as recommended in (Cohen, 2000). Figure 12 illustrates the means of the untransformed completion times for all the tasks.

A one-way analysis of variance (ANOVA) was employed on the transformed times. The within-subjects factor was the type of visualization with three



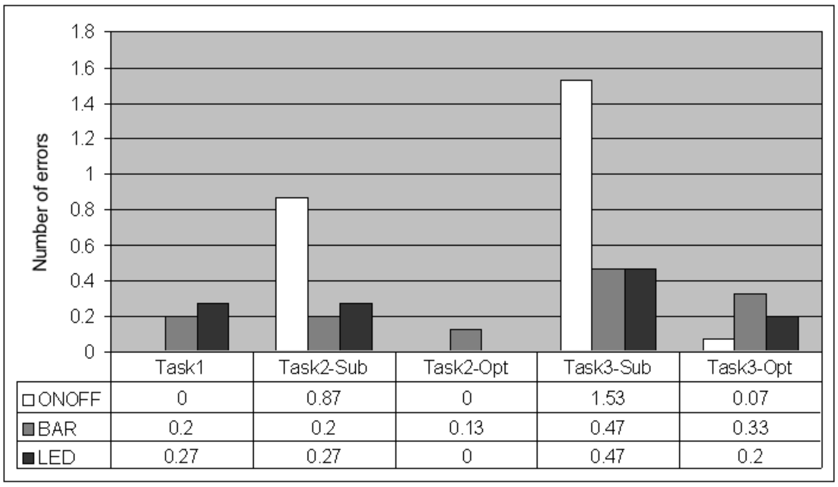
	Task1	Task2-Sub	Task2-Opt	Task3-Sub	Task3-Opt
□ ONOFF	46.22	27.9	25.42	52.54	36.11
▨ BAR	63.65	41.44	30.76	44.64	36.47
■ LED	66.12	46.44	28.79	45.29	32.77

Fig. 12. Mean task completion times.

levels: on-off, bar, led. The ANOVA revealed a significant effect for Task1 ($F(2, 42) = 4, 31, p < 0.05$) and Task2-Sub ($F(2, 42) = 12.52, p < 0.001$). Therefore, we employed the Tukey post-hoc test for comparison among pairs of means for those two tasks. For Task1, post-hoc analysis showed that users spent significantly less time to find the POIs with on-off than they did with led ($q = 3.72, p < 0.05$). For Task2-Sub, users spent significantly less time to find the target POIs with on-off than they did with bar ($q = 5.39, p < 0.01$) as well as led ($q = 6.67, p < 0.001$).

5.5.2 Errors

We employed Friedman's test (Friedman, 1937) to analyze the number of errors made by users during execution of tasks. An error was counted for each POI that was not in the solution set but was considered as a target by participants



	Task1	Task2-Sub	Task2-Opt	Task3-Sub	Task3-Opt
□ ONOFF	0	0.87	0	1.53	0.07
▨ BAR	0.2	0.2	0.13	0.47	0.33
■ LED	0.27	0.27	0	0.47	0.2

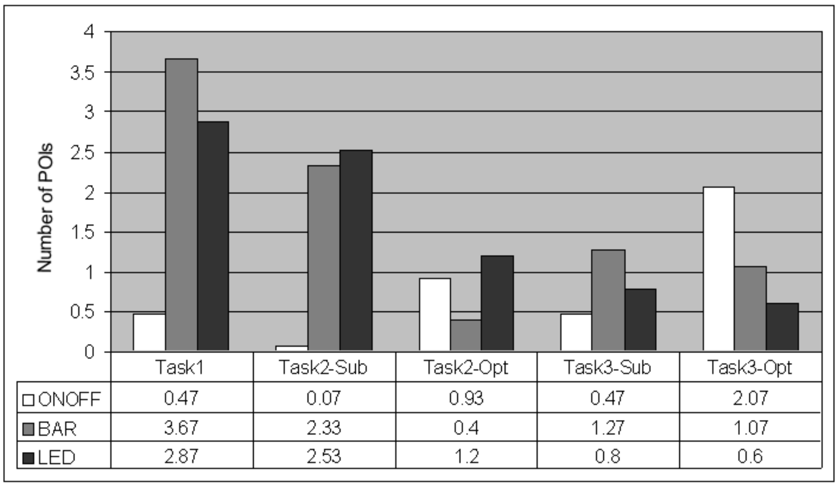
Fig. 13. Mean number of errors.

and for each target POI that was left out by participants. Fig. 13 illustrates the mean number of errors for all the tasks.

The analysis revealed a significant effect in Task2-Sub ($T = 18.20, p < 0.0001$) and Task3-Sub ($T = 14, 73, p < 0.001$). Dunn's Multiple Comparison post-hoc test (Dunn, 1964) was then used to compare pairs of means for those two tasks. A statistically significant difference was found for Task2-Sub between on-off and bar ($p < 0.05$), as well as between on-off and led ($p < 0.05$) with users making more errors with on-off in both cases. Similarly, users made more errors in Task3-Sub with on-off than they did with bar ($p < 0.05$) or led ($p < 0.05$).

5.5.3 Selected POIs

Friedman's test was also used to analyze the number of POIs tapped by users during execution of tasks. Means are shown in Fig. 14. A significant effect was found in Task1 ($T = 13.43, p < 0.005$) and Task2-Sub ($T = 13.50, p < 0.005$). Dunn's Multiple Comparison post-hoc test revealed a statistically significant difference between on-off and bar in Task1 ($p < 0.01$) and in Task2-Sub ($p < 0.05$) as well as between on-off and led in Task1 ($p < 0.05$) and in Task2-Sub ($p < 0.05$), with users selecting less POIs with on-off in all cases.

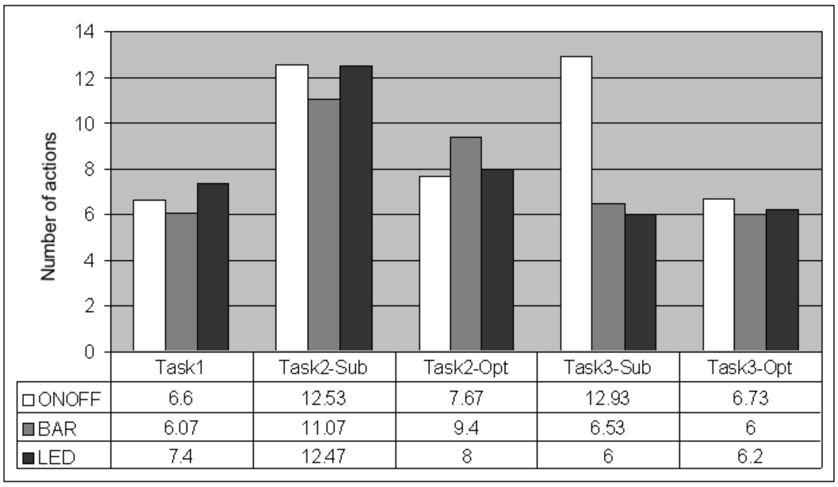


	Task1	Task2-Sub	Task2-Opt	Task3-Sub	Task3-Opt
□ONOFF	0.47	0.07	0.93	0.47	2.07
▨BAR	3.67	2.33	0.4	1.27	1.07
■LED	2.87	2.53	1.2	0.8	0.6

Fig. 14. Mean number of selected POIs.

5.5.4 User interface actions

Friedman's test on the number of interface actions (i.e., the number of interactions with sliders and checkboxes), whose means are illustrated in Fig. 15, indicated a significant effect in Task3-Sub. The post-hoc test revealed a statistically significant difference between on-off and bar ($p < 0.05$) as well as between on-off and led ($p < 0.05$), with users doing more actions with on-off.

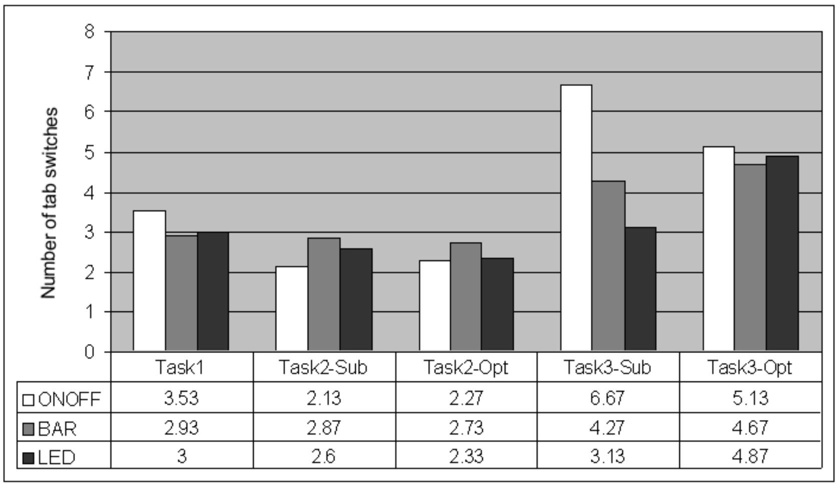


	Task1	Task2-Sub	Task2-Opt	Task3-Sub	Task3-Opt
□ONOFF	6.6	12.53	7.67	12.93	6.73
▨BAR	6.07	11.07	9.4	6.53	6
■LED	7.4	12.47	8	6	6.2

Fig. 15. Mean number of user interface actions.

5.5.5 Tab switches

We analyzed the number of tab switches made by users using Friedman's test. Means are shown in Fig. 16. Similarly to the analysis of the number of user interface actions, we found a significant effect only in Task3-Sub. Dunn's post-hoc test shows that users made a higher number of switches with on-off than they did with bar ($p < 0.05$) and led ($p < 0.001$).



	Task1	Task2-Sub	Task2-Opt	Task3-Sub	Task3-Opt
□ONOFF	3.53	2.13	2.27	6.67	5.13
▨BAR	2.93	2.87	2.73	4.27	4.67
■LED	3	2.6	2.33	3.13	4.87

Fig. 16. Mean number of tab switches.

5.5.6 Subjective preference

When users where asked to choose their preferred visualization, on-off got the highest number of votes (8), followed by led (5) and bar (2). However, the difference among the three is not statistically significant (chi-square test, $\chi^2 = 3.60$, $p > 0.05$).

5.6 Discussion

The results of our analysis show that there are important differences among the visualizations we compared. In particular, we found each visualization to be more or less appropriate according to the task users have to carry out. In Task1 and Task2-Sub, users are faster with on-off than they are with the other two visualizations, while there are no statistically significant differences in completion times among the three visualizations in the remaining three tasks. User's performance with on-off in Task1 is positively affected by the drastic reduction in the number of POIs which are displayed after users specify the desired number of stars for the hotel. In this way, the visual identification of a hotel satisfying the query is simplified. On the contrary, all POIs are always visible in the other two visualizations and, after setting the number of stars, users need to look for the first POIs whose bar or set of leds becomes partially red during the manipulation of the price slider. As a consequence, users have to carefully observe the results and to tap on POI icons to look at their attributes values and identify the correct target. Moreover, it is easier for users to perceive icons that disappear from the screen rather than observe changes in the color of bars or leds. Task2-Sub is similar to Task1 because users should first set the number of stars and then interact with the checkboxes to set the services attribute. The main difference with Task1 is that finding the hotels providing the highest number of services requires users to try different combinations of services. In this case, besides what we have already said for Task1, we also observed that, when users found a target with on-off, most of the times they stopped searching for other possible targets, thus reducing the time for carrying out the task. This behavior is also confirmed by the higher number of errors made by users with on-off in Task2-Sub.

Users made significantly more errors with on-off than they did with bar or led in both Task2-Sub and Task3-Sub. We must note that these were the tasks where no optimal but only suboptimal solutions could be found. In these situations, on-off could not provide any indications that allow users to refine their queries and find the targets. Users were thus forced to blindly relax the conditions on attributes and were not able to identify all the required suboptimal hotels. The result was an increase in the number of missed targets or wrong targets (i.e., results that were erroneously identified as targets).

In Task3-Sub, users made more interface actions with on-off. Again, this is due to the lack of indications on how to relax conditions. Users thus had to perform a high number of actions to identify possible targets. Finally, in Task3-Sub users made also a higher number of tab changes with on-off than they did with bar or led. This is also a result of the lack of indications to refine queries. Since Task3-Sub required users to interact with all 3 attributes of hotels, they often moved from one attribute to the other to modify the values and possibly find targets.

While we did not find any statistically significant preference for a visualization over the others, it is interesting to note that more than half of the users preferred on-off, probably because it allows to reduce the number of POIs to be controlled at once, even if it may result in users failing to identify the best targets. However, it must be noted that we did not tell users when they missed some targets during the tasks. If they were provided with this information, probably they would have had more difficulties in completing the task using on-off and they would have been less willing to choose on-off as preferred visualization.

These results highlight that none of the proposed visualizations outperforms the others in all the context. With on-off, users are faster in carrying out simple tasks but they might have difficulties in finding optimal results, thus leading to errors that can impair the decision process. The overview of query results provided by bar and led visualizations allows instead to overcome that drawback of on-off but may lead to an increase in the time needed to actually identify the correct results when a high number of elements is visualized at once.

6 Conclusions and Future Work

By combining an intuitive interaction method based on dynamic queries with appropriate representations of query results that simplify the visual exploration of data, the PDA-based system presented in this paper can support users in the analysis of datasets of geo-referenced multi-attribute elements in the field. The possibility to view suboptimal results and to compare results belonging to different datasets allows users to find solutions based on tradeoff analyses and proximity criteria and the interactive exploration of a dataset allows users to acquire new knowledge which is likely to prompt users to carry out further analysis and investigations in the field.

The results of the experimental evaluation we carried out to compare MAGDA visualizations show the usefulness of providing users with different ways to display query results, to accomodate diverse user's needs and overcome the limitations of the classical on-off visualization used in dynamic query systems.

We are currently studying different approaches to face the cluttering problem mentioned in Section 3.1.4, especially for the bar and led visualizations, with the aim of improving user's performance while keeping most of the visualizations strengths. For example, a basic approach could be to filter out those query results whose degree of satisfaction is lower than a predefined threshold, thus selectively reducing the number of visualized elements. An alternative approach would be to identify clusters of mutually overlapping icons and replace them with special aggregator icons, whose purpose consists in pointing out the presence of aggregated icons as well as providing users with a means to access information about each of them. This technique could help to declutter the visualization, freeing map space by removing all icon clusters without causing information to be permanently lost.

We will also design visualizations that can provide overviews of the distribution of attribute values for a specific dataset, so that users do not have to separately examine each POI to obtain its details.

Another important direction for future work is the design of a query engine that allows users to perform spatial and temporal queries. In this way, it would be possible for users to determine different types of spatial and temporal relations between the elements of the explored datasets, such as "Find all cities within a 10 km range from the selected pollution site" or "Find all the buildings that have been built after the currently selected one".

Finally, we will refine the system interface, adding features that could improve its usability. For example, we will allow users to change the query device associated to a specific attribute, so that they can adapt the interface to their preference.

7 Acknowledgment

Luca De Marco played an important role in the design and implementation of a preliminary version of the proposed system. Gianluca Forcione helped in carrying out the user study.

This work has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under the PRIN 2005 Project "Adaptive, Context-aware, Multimedia Guides on Mobile Devices".

References

Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R. and Pinkerton, M. (1997), 'Cyberguide: A mobile context-aware tour guide', *Baltzer/ACM Wireless Networks* **3**(5), 421–433.

Agrawala, M. and Stolte, C. (2001), Rendering effective route maps: improving usability through generalization, *in* 'Proc. Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)', ACM Press, pp. 241–249.

Ahlberg, C. and Shneiderman, B. (1994), Visual information seeking: Tight coupling of dynamic query filters with starfield displays, *in* 'Proc. Conference on Human Factors in Computing Systems (CHI 94)', ACM Press, pp. 313–317.

Ahlberg, C. and Truvé, S. (1995), Exploring terra incognita in the design space of query devices, *in* 'Proc. Working Conference on Engineering for Human Computer Interaction (EHCI 95)', Chapman & Hall, pp. 305–321.

Ahlberg, C., Williamson, C. and Shneiderman, B. (1992), Dynamic queries for information exploration: an implementation and evaluation, *in* 'Proc. Conference on Human Factors in Computing Systems (CHI 92)', ACM Press, pp. 619–626.

Baudisch, P., Good, N., Bellotti, V. and Schraedley, P. (2002), Keeping things in context: A comparative evaluation of focus plus context screens, overviews, and zooming, *in* 'Proc. Conference on Human Factors in Computing Systems (CHI 2002)', ACM Press, pp. 259–266.

Baudisch, P. and Rosenholtz, R. (2003), Halo: a technique for visualizing off-screen locations, *in* 'Proc. Conference on Human Factors in Computing Systems (CHI 2003)', ACM Press, pp. 481–488.

Baus, J., Cheverst, K. and Kray, C. (2005), A survey of map-based mobile guides, *in* 'Map-based mobile services - Theories, Methods, and Implementations', Springer-Verlag, pp. 197–216.

Burigat, S. and Chittaro, L. (2005), Location-aware Visualization of VRML Models in GPS-based Mobile Guides, *in* 'Proc. Conference on 3D Web Technology (WEB3D 2005)', ACM Press, pp. 57–64.

Burigat, S., Chittaro, L. and De Marco, L. (2005), Bringing Dynamic Queries to Mobile Devices: a Visual Preference-based Search Tool for Tourist Decision Support, *in* 'Proc. IFIP Conference on Human-Computer Interaction (INTERACT 2005), LNCS 3585', Springer, pp. 213–226.

Burigat, S., Chittaro, L. and Gabrielli, S. (2006), Visualizing Locations of Off-Screen Objects on Mobile Devices: A Comparative Evaluation of Three Approaches, *in* 'Proc. Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile HCI 2006)', ACM Press, pp. 239–246.

Chalmers, D., Sloman, M. and Dulay, N. (2001), Map adaptation for users of mobile systems, *in* 'Proc. World Wide Web Conference (WWW 2001)', ACM Press, pp. 735–744.

Cheverst, K., Davies, N., Mitchell, K. and Friday, A. (2000), Experiences of developing and deploying a context-aware tourist guide: the GUIDE project, *in* 'Proc. MobiCom 2000', ACM Press, pp. 20–31.

Chittaro, L. (2006), 'Visualizing information on mobile devices', *IEEE Computer* **39**(3), 40–45.

Cohen, B. (2000), *Explaining Psychological Statistics*, Wiley.

Dunlop, M., Morrison, A., McCallum, S., Ptaskinski, P., Risbey, C. and Stewart, F. (2004), Focussed palmtop information access combining starfield displays and profile-based recommendations, *in* 'Proc. Mobile HCI 2003 Workshop on Mobile and Ubiquitous Information Access, LNCS 2954', Springer-Verlag, pp. 79–89.

Dunn, O. (1964), 'Multiple contrasts using rank sums', *Technometrics* **5**, 241–252.

Dykes, J. A. and Mountain, D. M. (2003), 'Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications', *Computational Statistics & Data Analysis* **43**(4), 581–603.

ESRI (2006), 'http://www.esri.com'.

Fishkin, K. and Stone, M. (1995), Enhanced dynamic queries via movable filters, *in* 'Proc. Conference on Human Factors in Computing Systems (CHI 95)', ACM Press, pp. 415–420.

Friedman, M. (1937), 'The use of ranks to avoid the assumption of normality implicit in the analysis of variance', *Journal of the American Statistical Association* **32**(200), 675–701.

Gartner, G. and Uhlirz, S. (2001), Cartographic Concepts for Realizing a Location Based UMTS Service: Vienna City Guide Lol@, *in* 'Proc. International Cartographic Conference', pp. 3229–3239.

Harrie, L., Sarjakoski, L. T. and Lehto, L. (2002), 'A mapping function for variable-scale maps in small-display cartography', *Journal of Geospatial Engineering* **2**(3), 111–123.

Keim, D. A. (2002), 'Information Visualization and Visual Data Mining', *IEEE Transactions on Visualization and Computer Graphics* **8**(1), 1–8.

Laakso, K., Gjesdal, O. and Sulebak, J. (2003), Tourist information and navigation support by using 3D maps displayed on mobile devices, *in* 'Proc. Mobile HCI Workshop on HCI in Mobile Guides', pp. 34–39.

Lodha, S., Faaland, N., Wong, G., Charaniya, A., Ramalingam, S. and Keller, A. (2003), Consistent Visualization and Querying of GIS Databases by a Location-Aware Mobile Agent, *in* 'Proc. Computer Graphics International (CGI 2003)', IEEE Press, pp. 248–253.

MacEachren, A. and Kraak, M. (2001), 'Research Challenges in Geovisualization', *Cartography and Geographic Information Science* **28**(1), 1–12.

Plaisant, C., Carr, D. and Shneiderman, B. (1995), 'Image-browser taxonomy and guidelines for designers', *IEEE Software* **12**(2), 21–32.

Rauschenbach, U., Jeschke, S. and Schumann, H. (2001), 'General Rectangular FishEye Views for 2D Graphics', *Computers and Graphics* **25**(4), 609–617.

Reichenbacher, T. (2004), Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices, PhD thesis, Technischen Universitat Munchen.

Shneiderman, B. (1994), 'Dynamic Queries for Visual Information Seeking', *IEEE Software* **11**(3), 70–77.

Spence, R. (2001), *Information Visualization*, Addison-Wesley & ACM Press.

Trimble (2006), 'http://www.trimble.com'.

Tripcevich, N. (2004), 'Flexibility by Design: How Mobile GIS Meets the Needs of Archaeological Survey', *Cartography and Geographic Information Science* **31**(3), 137–151.

Williamson, C. and Shneiderman, B. (1992), The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System, *in* 'Proc. Conference on Research and Development in Information Retrieval (SIGIR 92)', ACM Press, pp. 338–346.

Zipf, A. (2002), User-Adaptive Maps for Location-Based Services (LBS) for Tourism, *in* 'Proc. Conference for Information and Communication Technologies in Travel & Tourism (ENTER 2002)', Springer-Verlag.

Zipf, A. and Richter, K.-F. (2002), 'Using Focus Maps to Ease Map Reading: Developing Smart Applications for Mobile Devices', *Kunstliche Intelligenz* **4**, 35–37.