Framy – Visualizing Spatial Query Results on Mobile Interfaces

Luca Paolino, Monica Sebillo, Genoveffa Tortora, and Giuliana Vitiello

Dipartimento di Matematica e Informatica, Università di Salerno
Via Ponte don Melillo,
84084 Fisciano(SA), Italy
{lpaolino,msebillo,tortora,gvitiello}@unisa.it

Abstract. The widespread use of mobile devices in basic map navigation tasks has recently attracted researchers on usability problems arising from the reduced visualization area and the limited interaction modes allowed by small keypads. In this paper we analyse the most common approaches suggested in the literature and propose a new visualization technique, named *FRAMY*, which is based on one of them and exploits a novel interaction metaphor for picture frames to provide hints about off-screen objects. It was conceived to cover a wider range of spatial data visualization tasks, which may simultameously involve different geographic layers.

Keywords: Mobile devices, geographical information systems, spatial visualization methods, spatial functions.

1 Introduction

People use maps in a number of tasks, including finding the nearest relevant location, such as a gas station, or for hand-optimizing a route. Using a map, users can easily compare alternative locations, such as the selection of restaurants. Users can see how far away a restaurant is from his/her current location, and whether it lies close to other locations the user considers visiting. Advantages deriving from maps are uncountable, they provide people with the ability to show and analyze the world as they perceive it, namely by verifying topological, directional and metrical relationships. However, a complete and exhaustive evaluation of a region is possible just in case users are able to:

1) have a global or wide enough view of the map portion they are taking into account and simultaneously
2) have a vision of the map where features can be well distinguished from each other.

In other cases, comparisons as well as evaluations of spatial relationships may result very difficult because users have either no sufficient information or confused information.

J.M. Ware and G.E. Taylor (Eds.): W2GIS 2007, LNCS 4857, pp. 175–186, 2007.

This is the case for maps visualized on mobile devices, where viewing a wide portion of the map conflicts with showing features in a well defined and separate fashion, due to the small dimensions of screens.

As an example, if we perform a search for the closest hotel from our current location, the result may fall either inside or outside the screen. In the first case, we do not have problems to reach the hotel because we can distinguish it on the map. Vice versa, in the second case, we do not know which direction we have to follow to arrive at the target and we have to gropingly advance.

Visualizing the entire map is not the right solution. As a matter of fact, objects resulting from a query task might appear improperly overlapped, if a reduced map scale is adopted. As an example, if we are visualizing a layer showing the locations of farmhouses located in a given region and we want to see all of them together, we might want to represent the entire region on the screen, hence using a reduced scale for the map. In that case, there is a high probability that, below a certain proximity threshold, farmhouses will be overlapped and will then be hard to distinguish.

In the literature, several techniques have been proposed to display large information spaces on desktop screens. An interesting paper by Chittaro classifies the different approaches adopted so far to visualize maps and images on small screens [2]. The author argues that the techniques traditionally adopted to combine panning and zooming with either compression (the so-called *overview-detail* approach) or distortion operations (known as *focus-detail* approach) may still present severe limitations when applied to mobile devices, due to the very small screen dimensions. A promising approach, specifically conceived for visualizing distance and direction of off-screen locations, consists in providing visual references to off-screen areas, by augmenting the detail view with interactive visual references to the context. Such an approach has been successfully adopted in systems like Halo to enable users to complete map-based route planning tasks [1]. This suggested us that a similar idea could be adopted and extended for performing more complex GIS activities on a mobile device, which also involve spatial queries on different map layers.

Thus, in order to provide an appropriate tradeoff between the zoom level needed to visualize the required features on a map and the amount of information which can be provided through a mobile application, we propose a new visualization technique, named *FRAMY*, which exploits an interaction metaphor for picture frames, to provide hints about off-screen objects, resulting from a given query. The idea is to visualize a frame along the border of the device screen. The frame is divided into several semi-transparent colored portions, corresponding to different sectors of the off-screen space. For a given query, the color intensity of each frame portion is proportional to the number of resulting objects located in the corresponding map sector. Thus, the frame may indicate both the distance and direction of specific Point Of Interests (POIs), as in Halo, but it may also represent the amount of POIs located towards a specific direction. In general, the frame portion color intensity represents a summary of data located besides the screen border. Moreover, the number of frame portions can be interactively increased so as to refine the query results, indicating, e.g., the exact direction where certain objects can be found. If the query involves several layers, nested frames are visualized along the borders, each one corresponding to a different layer, with a different color.

FRAMY is currently being experimented on a prototype of a mobile GIS application, called *MapGIS*, designed to perform typical GIS operations and queries, and combines typical mobile interaction controls with the proposed visualization technique.

The paper is organized as follows. In Section 2 we give a brief overview of the common approaches followed to visualize spatial data on mobile devices. In Section 3 we describe the MapGIS application and explain how the proposed visualization technique supports the corresponding functionalities. In Section 4 we illustrate three typical scenarios where *MapGIS* can be employed, which show the effectiveness of the *Framy* technique. Some final remarks conclude the paper.

2 State of the Art

One of the simplest ways to analyze the information depicted in maps shown on mobile devices is the classic *Pan & Zoom* method.

Maps may be seen at different levels of detail by applying the *Zoom* operation while the visualized region of interest may be changed applying the *Pan* operation. Typical styles of interaction concerning with the former are the selection of the detail level by means of specific buttons (one for each level), by sliders, by menus or by managing the wheel of the mouse. The *Pan* operation is usually implemented by dragging the mouse over the map or by scrolling panels.

An approach combining Pan and Zoom together was presented in [5]. At the start of an action, two concentric circles are placed so that the location of the action on the display is at their centre. As the user drags the pointing device, a direction line is drawn between the starting position and its current location, indicating the direction of work. If the pointer remains within the smaller circle, no scaling of the information space occurs, only scrolling. As the pointer moves further away from the starting position, the scroll rate increases. When the pointer moves beyond the inner circle, both scaling and scrolling operations take place.

Several papers have been also presented concerning with the technique named "Jump & Refine"[4][9][10]. This technique segments the map into several sub-segments, each of which is mapped to a key on the number keypad of the smartphone. By pressing one of the buttons, the region represented by that button is automatically puts in evidence on the screen. At this point the user may either pan the visualized region in order to reach the required zone or recursively apply the segmentation for augmenting the detail level.

Another interesting approach has been presented in [1]. The software implement-ting this methodology focuses the attention on discovering points of interest located outside the visualized region. As a matter of fact, each point of interest is the center of a circle whose ray raises or decreases as we get close or far from that point. In such a way, we may verify anytime the proximity of the point. A variation of this approach is CityLights [7]. Here, compact graphical representations such as points, lines or arcs which are placed along the borders of a window to provide awareness about off-screen objects located in their direction.

In the past, other interesting techniques for visualizing data on small devices have been also presented. Generally, they may be divided in two main categories, Overview&Detail and Focus&Context [6].

In order to have the awareness of a particular region and the context in which it stay, Overview&Detail techniques present simultaneously two maps where one refers to a limited/detailed region and cover the entire screen while the second one is the whole scaled map located just in a corner. Optionally, the detailed region is also highlighted by a rectangle on the whole map. Panning the first map changes the rectangle position on the second one. Even though today this approach is largely used, it was firstly described in [2]. The *perspective overview-display* is another example of a Overview&Detail technique. Here, two completely distinct areas are used to display the overview (top) and detail (bottom) of an image. To save valuable screen space while displaying the overview, the image is prospectively warped.

The Focus&Context technique is based on the reasonable assumption that the attention of users is focused on the center of the screen. FishEye View[8] is an example of this technique. Here, the center of the map is clearly displayed in the center of the screen while distortion augments starting with less distortion near the focus to strong distortion near the borders of the available display size. Based on the FishEye approach, three possible implementations of the distortion function are provided in [6], uniform context scaling, belt-based context scaling, non-uniform context scaling.

Finally, an approach merging together Overview&Detail and Focus&Context is presented in [6]. Two different images are used. These images are created by a transformation function. To display the context, the whole image in the logical coordinate system is scaled to the available display size. The focus is created as during panning, but using an equal or smaller size than the available display size. Afterwards the images are combined by the transformation function by applying a fixed or time-variant α-blending.

3 The *Framy* Visualization Technique Supporting a Mobile GIS Application

In this section we describe the *MapGIS* application, namely a tool which allows mobile device users to perform typical GIS operations and queries whose results are depicted according to the *Framy* visualization methodology.

MapGIS is based on a typical GIS architecture realized in a J2ME platform for mobile applications. It allows users to gather information about localities on the considered map by querying the data pre-loaded and stored in a structured repository of the device. It manages the browsing of the map by using a specific pointer that is moved all round the screen through the directional keys on the device.

Two kind of functionalities are currently available on *MapGIS*, that is:

- basic spatial operations, and
- advanced selection operations.

Panning and zooming are examples of basic operations. They allow users to easily explore maps everywhere and at different scales. So, in order to accomplish such tasks as easily and accessible as possible and guarantee a good level of usability, rapid controls have been provided through the device keyboard.

As for the advanced selection operations, they can be used to process both geographic and descriptive data in order to answer specific user's requests about the real world data. As an example, in *MapGIS* a search functionality is provided on descriptive data by pointing out specific localities of the map. This functionality represents the generally called operation "Spatial Selection". It allows users to reference every point located on the map by means of the browsing pointer and returns the descriptive data which are strictly connected to the pointed locality.

Besides the usual route planning functionalities, which also rely on some optimization algorithms, *MapGIS* provides an advanced search functionality, which allows to localize points of interest of specific categories (theaters, parks, museums, cinemas, etc) by specifying some descriptive information set up by users.

Once the user performs the spatial requests, results are visualized on the map by applying the *Framy* visualization method which helps users to discover results located off-screen.

The user may choose to divide the map into any number n which is a power of 2. Starting from the center of the screen a fictive circle is drawn and the map is divided into n sectors of equal width ($360°/n$). Figure 1 gives an idea of this subdivision using $n = 8$, that is we divide the screen in 8 sectors. We use the notation *Ui* – invisible- to represent the parts within a sector which lie outside the screen, whereas *Vi* – visible- represents the inside screen parts.

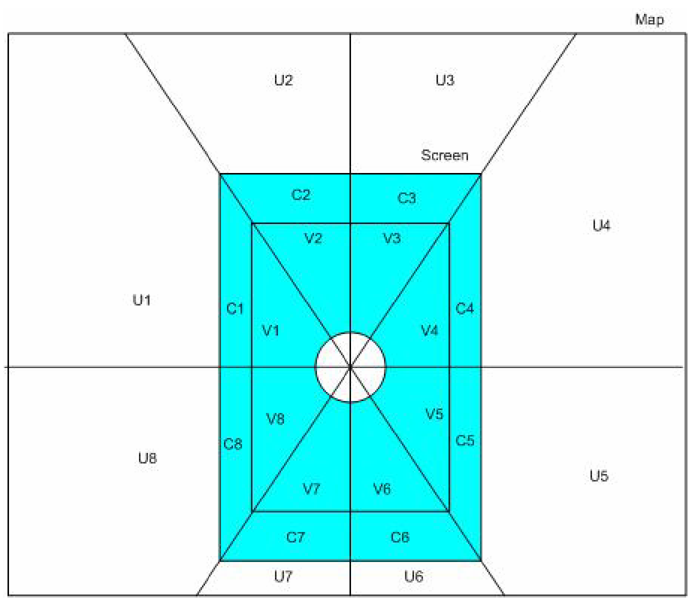

Fig. 1. The (off-) screen subdivision by *Framy*

The idea underlying this approach is to inscribe a semi-transparent area (*Cornice*) inside the screen. This area is partitioned in *C1, …, Cn* portions, each identified by the intersection between the *Cornice* and the *Vi* parts.

Once the user has posed a query, each *Ci* is colored with a different intensity on the basis of an aggregate function *f* applied on the resulting set associated with *Ui*.

Formally, we state that the color intensity is:

$$\text{Color}(Ci) = f(Ui) \text{ for each } i \in \{1, \ldots, n\} \quad (1)$$

where *f* is any aggregate function which calculates a numeric value starting from a set of spatial data.

As an example, let us suppose that *f* is the *count* operation applied on the Hotel layer where features are selected only if they have four stars, namely:

```
SELECT count(geometry)
FROM Hotel
WHERE category = 4;
```

Then, the higher the number of selected features is within *Ui*, the more intense the color *(Ci)* is.

Analogously, if *f* computes the shortest distance from the current position then the *Ci* with the highest color intensity will indicate the *Ui* containing the closest feature.

With respect to other kind of approaches, *Framy* is not only addressed to visualize a results of a specific function (i.e. the proximity of features), but may be customized according to chosen aggregate function. As a matter of fact, the user may even choose the layer onto which it applies and the granularity level of the subdivision, that is how many parts compose the cornice.

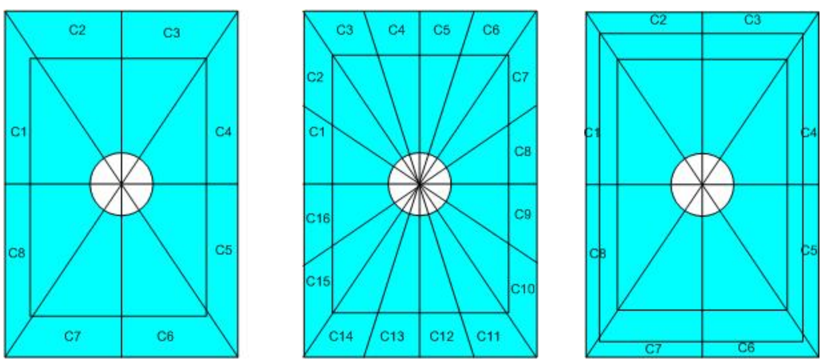

Fig. 2. (a) a Framy subdivision using 8 parts, (b) one using 16 parts, and (c) a concentric subdivision

For this reason, as an example, the user may apply the shortest distance function on both the Hotel layer and the Parking area layer in order to locate the *Ci* with the closest hotel and the *Cj* with the closest parking area, respectively. Moreover, in order to identify the direction to reach a particular target, the user may increase the number of subdivisions so that the *Ci* identifying the correct direction is as smaller and more precise as possible (see Fig. 2 (a) and 2 (b)).

Another important characteristic of *Framy* is the possibility to create more cornices and visually compare them. As a matter of fact, as shown in Figure 2(c) cornices may be nested in order to compare color intensity inside a sector. This is to say, the user may visualize the closest hotels on a cornice and the closest parking areas on a nested cornice in order to discover which hotel/parking pair is best located with respect to the current position. This may support user to choose the best place to reside by taking into account the hotel proximity and the number of points of interest in a map.

Another way to use *Framy* is by checking the color variation as the center of the map changes. As an instance, if the cornice is used to locate the closest hotel for each *Ci*, we can verify whether we are approaching to a particular hotel by controlling the intensity of the color in the chosen direction. This is to say, the hotel is approaching by a panning operation when the color gets more intense.

Defining Color Intensity

Let us present the technique we adopted for assigning colors to cornice. It is based on the Hue, Saturation, Value (HSV) model that we briefly recall in the following.
The HSV model, also known as HSB (Hue, Saturation, Brightness), defines a color space in terms of three components, namely:

- *Hue*, the color type (such as red, blue, or yellow). It ranges from 0 to 360 in most applications. Each value corresponds to one color. As an example: 0 is red, 45 is a shade of orange and 55 is a shade of yellow (see Fig. 3 for the complete color scale).

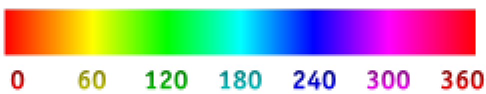

Fig. 3. The hue scale

- *Saturation*, the intensity of the color. It ranges from 0 to 100%. 0 means no color, that is a shade of grey between black and white. 100 means intense color. Sometimes it is called the "purity" by analogy to the colorimetric quantities excitation purity and colorimetric purity
- *Value*, the brightness of the color. It ranges from 0 to 100%. 0 is always black. Depending on the saturation, 100 may be white or a more or less saturated color.

At this point, let us now introduce the formulae we used to assign color intensities to the *Cis* . Let us suppose we are measuring the distribution of the n_i objects within the *Ui* sector and that we want to represent this value by using yellow tonalities. To reach this aim we set H to 60 (corresponding to the yellow color) as Figure 3 and Figure 4 depict, then we calculate the incremental value (step) that each object provides in term of tonality and finally, we multiply this value for the number of objects located in each Ui.

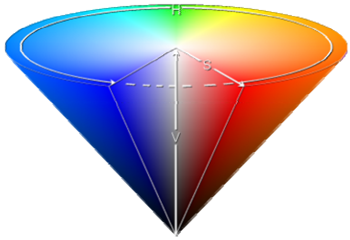

Fig. 4. The conical representation of the HSV model is well-suited to visualize the entire HSV color space in a single object

In order to calculate the step, we have to know the maximum and minimum number of objects that *Ui* could contain. As a matter of fact, the maximum one should correspond to the available most intense yellow, namely S=100, whereas the minimum one should correspond to absence of color, namely S=0. Then, the step will be:

$$Step=(MAX-MIN)/100 \quad (2)$$

and the color for each Ci will be calculated by using the following triple:

$$Ci = (H_{Layer}, Si, V) \quad (3)$$

where:

- *H is assigned for each layer,*
- $Si = step * n_i$
- $V = 100$

4 Exploiting *Framy* in Typical Spatial Query Scenarios

In this section three scenarios will be shown in order to exemplify three different modalities to use *Framy* as a visualization technique in *MapGIS*. In particular, in the first example we use *Framy* to indicate the quantity of objects belonging to the layer of interest. The second example shows how *Framy* may be used to reach a desired destination by using the intensity color variation. Finally, in the third scenario, we show how to use concentric cornices to compare different information.

Example 1: Framy as a visual synthesizing technique

The first scenario we describe shows the usage of *Framy* to support users in getting an idea about the distribution of the whole set of data over a map, disregarding of the actual visible map extension.

Figure 5 depicts the layers of the banks and streets in Sydney. The screen of a mobile phone is overlapped with them and the gray dotted lines indicate the chosen subdivision of the cornice, namely 8 portions. Finally, the center of the map is visualized as a cyan colored target.

It is possible to note that the color intensity of each *Ci* is proportional to the number of banks located in the corresponding off-screen sector. As an example, the C_is corresponding to the bottom-left side of the map are associated with a peripheral zone, then they result either transparent or colored with a very light yellow. On the other hand, the yellow color associated with the C_is on the opposite corner gets more intense as the corresponding sectors capture a greater quantity of banks.

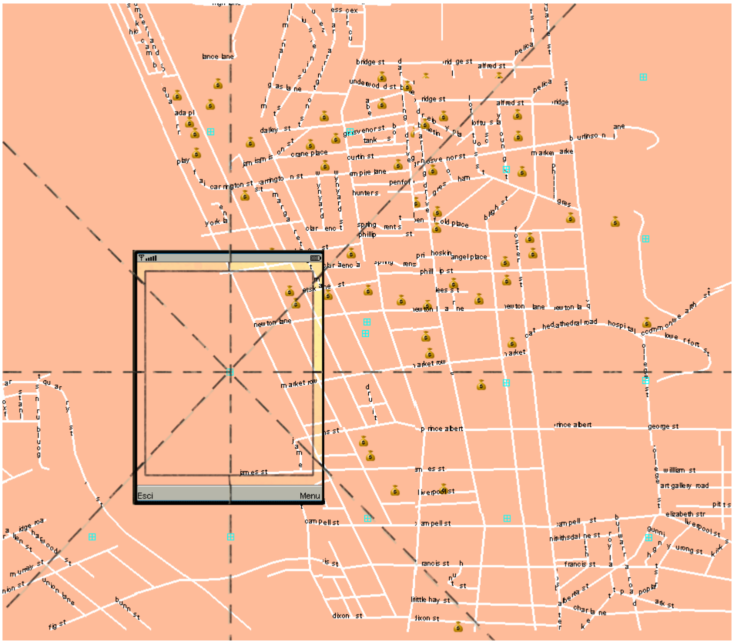

Fig. 5. Overlapping layers of interest with the screen of a mobile phone

Example 2: Locating a position by capturing color variations

By moving the device screen all around the map, the *Ci* colors are re-calculated and depicted at run-time in order to visualize the current situation around the map focus. This property provides users with the opportunity of understanding how the value of the aggregate function they have chosen changes.

This property is particularly interesting in many cases. As an example, let us suppose we are looking for the closest bank with respect to the map focus. In this case, the *Ci* which results more intensely colored indicates the sector containing the closest bank. Moreover, the color gets more intense as we move the map focus towards the request feature, thus indicating the right direction to it.

As previously said, this property may be further improved by dividing the screen in more portions. In such a way, choosing the right direction becomes easier because the most intense part is associated with a more narrow sector. At the limit, the number of divisions is so high that the exact direction may be indicated by a line.

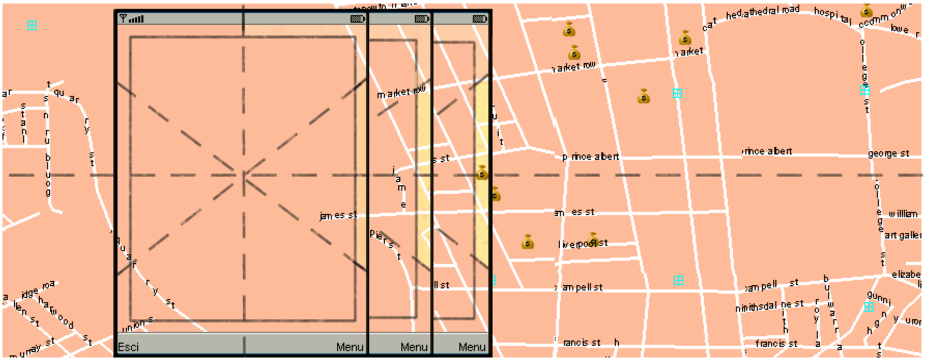

Fig. 6. The device screen taken in three successive instants

The example shown in Figure 6 depicts the device screen captured in three successive instants, where the *Ci* of interest dynamically indicates the closest bank. It is possible to note that as we move the screen towards the right side, the *Ci* color gets more intense.

For the sake of readability, we placed a large portion of the map overlapped with the screens in order to make it easier to understand the context we move in.

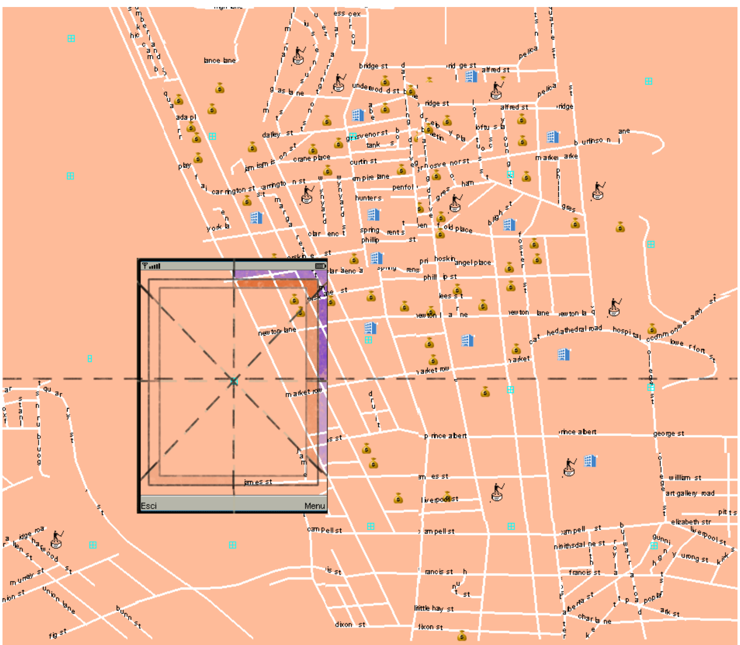

Fig. 7. A visual comparison by using Framy

Example 3: Framy as technique for visually comparing distributed information
In the last scenario, we show how to use *Framy* in order to compare values resulting from spatial aggregations.

In this example two layers showing hotels and monuments have been added to the map. Besides taking a look at the map, the request is to understand in which direction we should go in order to find a zone where it is easier to find simultaneously both a hotel and particular monuments.

In order to derive this information, we configured two concentric cornices – red for monuments and blue for hotels, respectively.

Given the data distribution, few *Ci* have been colored. In particular, colors are more intense on the right and up sides where the map shows numerous POIs. As a result, the top part better satisfies the initial requirement because the red and blue colors are simultaneously more intense with respect to other parts.

5 Conclusion and Future Work

In this paper, we defined a new method to visualize a large amount of geographic data on devices having small screens such as mobile phones, pda, etc.
Our solution adds a cornice inscribed inside the mobile screen which can be partitioned in agreement with data distribution on the map. Coloring such portions according to some spatial aggregate functions provides users with the opportunity to obtain a quantitative synthesis of features located off-screen, thus overcoming the inconvenient of a limited vision.

Some advantages come from this method. With respect to other kinds of approaches, this visual representation may be configured to work in different modalities. As a matter of fact, besides choosing the aggregating function on which the query is performed (e.g., count, distance, etc.), the user may choose the layer(s) on which the function is applied as well as the color representing the off-screen results. Moreover, the granularity level of the subdivision can be suitably adapted to the user's needs. Another important feature of this visual representation is the possibility to create more nested cornices and visually compare them. As a matter of fact, frames may be concentrically drawn in order to compare results of different aggregating functions.

In the future, we plan to accomplish some experiments to single out possible improvements into localizing information on mobile devices, also by comparing our approach with other specific methods. User experiments are also planned to verify the usability of this method in order to find out possible weakness.

An optimization of the color choice should be applied in order to avoid problems with different color intensities which cannot be distinguished.

References

1. Baudisch, P., Rosenholtz, R.H.: A Technique for Visualizing Off-Screen Locations. In: CHI 2003. ACM CHI Conference on Human Factors in Computing Systems, pp. 481–488. ACM Press, New York (2003)
2. Card, S.K., Mackinlay, J.D., Shneiderman, B.: Readings in information visualization - Using vision to think. Morgan Kaufman, San Francisco (1999)

3. Chittaro, L.: Visualizing information on mobile devices. IEEE Computer 39(3), 40–45 (2006)
4. Hachet, M., Pouderooux, J., Tyndiuk, F., Guitton, P.: Jump and Refine for Rapid Pointing. In: Mobile Phones in Proceedings of CHI 2007, pp. 167–170 (2007)
5. Jones, S., Jones, M., Marsden, G., Patel, D., Cockburn, A.: An evaluation of integrated zooming and scrolling on small screens. International Journal of Human-Computer Studies 63(3), 271–303 (2005)
6. Karstens, B., Rosenbaum, R., Schumann, H.: Information Presentation on Handheld Devices. In: Candance Deans, P. (ed.) E-Commerce and M-Commerce Technologies Ch. 2, Idea Group Publishing, USA (2004)
7. Mackinlay, J., Good, L., Zellweger, P., Stefik, M., Baudisch, P.: City Lights: Contextual Views in Minimal Space. In: CHI 2003. Proc. ACM CHI Conference on Human Factors in Computing Systems, pp. 838–839. ACM Press, New York (2003)
8. Rauschenbach, U., Jeschke, S., Schumann, H.: General rectangular fisheye views for 2D graphics. Computers & Graphics 25(4), 609–617 (2001)
9. Robbins, D.C., Cutrell, E., Sarin, R., Horvitz, E.: ZoneZoom: Map navigation for smartphones with recursive view segmentation. In: AVI 2004. Proc. International Conference on Advanced visual interfaces, pp. 231–234. ACM Press, New York (2004)
10. Rosenbaum, R.U., Schumann, H.: Grid-based interaction for effective image browsing on mobile devices. Proc. SPIE Int. Soc. Opt. Eng. 5684, 170–180 (2005)