LOCATION BASED CONTEXT AWARENESS THROUGH TAG-CLOUD VISUALIZATIONS

V. Paelke, T. Dahinden, D. Eggert, J. Mondzech

IKG, Institute for Cartography and Geoinformatics
Leibniz Universität Hannover
Appelstr. 9a, 30167 Hannover, Germany
{paelke, dahinden, eggert, mondzech}@ikg.uni-hannover.de

KEY WORDS: Information Visualization, Context Awareness, Tag-clouds, Location Based Services

ABSTRACT:

People often only see what that they already know or explicitly look for. Thus, while mobile users might be interested in the historic background of their surrounding, its current significance or its relation to specific events, they are likely to miss places of interest if they are not explicitly pointed out. Location based services (LBS) like mobile tourist guides offer a potential technological solution, but the production cost of multimedia content is prohibitive in many cases, limiting the coverage of such services to major tourist areas.

To provide mobile users with information on the spatial context of a location or a route we present an approach that gathers context information from freely available sources like Wikipedia and creates visualizations of this data that provide users with the necessary cues to increase awareness of their spatial context. In our approach we first gather geo-referenced information that is located close to a point or route. In the second step this data is filtered to extract key-words that characterize the environment. These are then rendered as a tag-cloud in the third step. By skimming the tag-clouds a user gets a good impression of the characteristic features of an environment and in essence performs a further filtering step. The user can interactively adjust the level of detail of the visualization or follow up on individual key-words to adjust the presentation to his interests.

By combining web 2.0 technologies and public data sources with filtering and visualization techniques we exploit the browsing capability of humans to provide a service that increases location awareness at arbitrary locations.

The approach makes it easy to author an additional text and it can incorporate the ever increasing amount of available geo-referenced information.

1. MOTIVATION

Advances in mobile computing and wireless communication technology enable the creation of location based services (LBS) on a variety of mobile devices ranging from mobile phones to PDAs and other portable computing devices. Modern Smartphones like Apple's iPhone, the Android based T-Mobile G1 and similar devices combine GPS based positioning with a digital compass, high resolution displays and high-bandwidth data connections. As the technical prerequisites for location based services become widely available the development of practical services and the creation of the content required for them becomes an increasingly important question.

One possible approach is the development of specific authoring tools that support the creation of location based multimedia content and its integration into user interfaces that consider the constraints of mobile devices. In this paper we consider a complementary approach: Information is gathered from freely available sources like Wikipedia and a combination of automatic filtering and processing techniques with information visualization techniques based on tag-clouds are used to provide the user with a display of the available information.

Location based services like mobile tourist guides provide two main services: They point out potential points of interest in the surroundings of the user and can provide users with detailed background information on these on demand. This functionality is useful because people often only recognize what that they explicitly look for. Because the production effort for content of current location based services is significant such information is often only available for touristic areas, where the high cost of content production can be apportioned on a large number of users.

A possible alternative to content that is explicitly authored for use in a LBS system is to exploit information that is already available, e.g. on websites. To make this information useful for the user two steps are required: First, information has to be spatially selected, so that only information pertaining to the surrounding of the user is used. Second, the information has to be refined into a form that is suitable for easy interpretation by the user.

In the approach that we present in this paper we exploit information from websites like Wikipedia and process it into a visualization that is based on the concept of tag-clouds to provide a presentation that can be quickly browsed by the user. After an introduction of the concept in the following section we discuss related work in section 3. In section 4 we discuss the current implementation. Examples are presented and discussed in section 5. Finally we discuss the experience with the approach and future work.

2. CONCEPT

As illustrated in Figure 1 our system starts either by acquiring the spatial position of the user or by planning a route through the environment. Current Smartphones feature a GPS unit that can be used for the localization, otherwise a more imprecise localization can be derived by identifying wireless network cells in the environment.

Using this position it becomes possible to supply the user with information on his spatial surroundings. The second case - involving route planning - is a bit more complex. Here not only the current surroundings of the user are queried but a spatial buffer around the planned route is used. This can be exploited to aid pedestrian navigation, by using the context data to improve route descriptions.

As the examples in section 5 show landmarks of special interest can usually be identified from the tag-cloud, thus supplying a textual representation of landmarks for pedestrian navigation. The user can interactively influence the processing at this stage both by supplying start and destination information for the route planning.
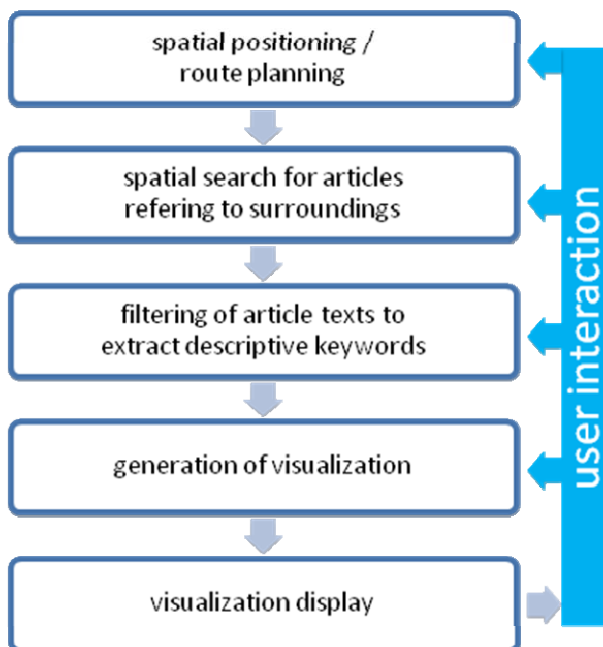

Figure 1: System processing workflow

The second step of the processing pipeline uses a spatial query to search for spatially referenced articles in the surroundings of the user. This is either a simple proximity query around the user's current location or a buffer operation around the planned route.

To influence the process the user can interactively specify a query radius in which information should be retrieved. It thus becomes possible to adjust the amount of data to the preferences of the user.

At this stage it is also possible to introduce additional information into the process. In some cases it could be desirable to provide explicitly authored or preselected information for specific areas. These are simply added to a spatial database and retrieved in the same fashion as the articles. If desired they can be flagged as additional" and thus prioritized in the following processing steps.

In the third step the text from the retrieved articles is filtered to remove irrelevant words. Usually verbs and articles are less descriptive than nouns and therefore they are removed. Very common words (e.g. the name of the city in which the user is currently located) also add little to the location awareness and should therefore also be removed. The filtering can be adjusted by the user and specific filter lists can also be supplied to suppress unwanted words from the tag-cloud.

In the fourth step a tag-cloud visualization is created from the filtered text. The user can influence this step by adjusting font type and size, layout criteria and the number of tags that should be displayed. A smaller number of tags is easier to browse and more descriptive and therefore the default setting. If a user is especially interested in his surroundings and wants to investigate more deeply or if the information provided in the display is not sufficient the user can easily choose to display more tags in the cloud.

The final step is the display of the tag-cloud. To improve readability the user can zoom in and out and scroll the display if required. This is especially useful for route visualizations. In addition the user can also "drill down" into the tag-cloud to retrieve more detailed information on tags of potential interest. If the user selects a specific tag the system can provide him with a list of the original occurrences in the source articles. Alternatively a web-search for the tag-cloud be initiated. While the tag-cloud provides an overview visualization of information of potential interest, exploiting the user's browsing capability to further filter the information, the possibility to link to the original articles can implement the second function of a LBS, namely to provide detailed information on points of interest in the environment.

A description of the hardware platform and the software used to implement the different processing stages in our prototypical implementation is given in section 4. The implementation concerns an evolutionary prototype where a sophisticated route planning and the interaction options of the tag-cloud are not yet implemented.

3. RELATED WORK

Our work was motivated by previous work on the development and use of conventional LBS using authored content. LBS like tourist guides (Zipf, 2002; Baus et al., 2005; Schilling et al., 2005) have become popular to use in areas where adequate content is available. Unfortunately, the production effort of such content is still very high and commercially successful implementations are usually restricted to top tourist destinations.

Central to our approach is the exploitation of the user's browsing capability using the tag-cloud visualizations. Tag-clouds have become popular through their use on Web 2.0 sites, namely Flickr (Flickr, 2009). The first widely published use of the format is usually attributed to Douglas Coupland who used a similar form of visualization in his book Microserfs (Coupland, 1995). As Viegas and Wattenberg (2008) point out there have been predecessors to this. Of special interest for our approach is the study of Milgram (Milgram, 1976) in which he aggregated the mental maps that people have of a space (in the case of this study Paris) into an aggregate visualization that has a lot of similarity to the tag-clouds produced by our system.

From a cartographic perspective it is also interesting to note the relations between tag-clouds and more structured lists of keywords. Lamantia (Lamantia, 2009) has pointed out the similar relation between structured lists and tag-clouds on one side and maps and cartograms on the other. In recent years the study of tag-cloud visualizations for different purposes has become a research topic in information visualization (Halvey and Keane, 2007; Rivadeneira et al., 2007). A large number of layout algorithms and libraries that implement them have been proposed and studied (for an overview see e.g. Kaser and Lemire, 2007).

Our approach relies on the availability of spatially referenced information in publicly accessible sources like Wikipedia (Wikipedia, 2009). In recent years there have been several attempts to exploit this information for purposes in a Geographic Information Science context, e.g. the GeoSR System (Hecht and Raubal, 2008) used spatially references Wikipedia data to explore semantic relations in spatial data.
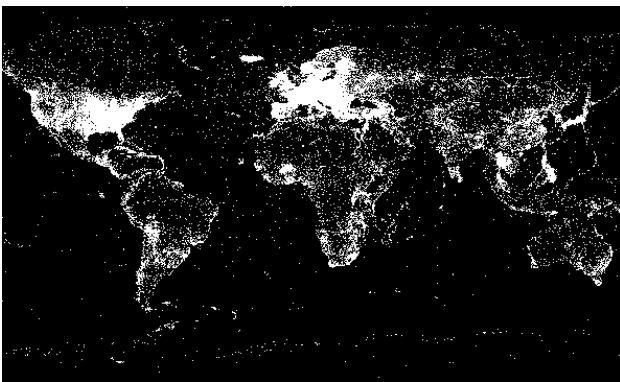

Figure 2: Distribution of Wikipoints of the German Wikipedia

4. IMPLEMENTATION

The implementation of our system demands a client-server approach – a client visualizing the tag-cloud and providing user interaction, as well as a server harvesting the desired data and creating the corresponding tag-cloud information. As mentioned before, this information can be obtained from various sources like Wikipedia, Flickr or any other knowledge repository with geo-referenced articles. Our prototypical implementation utilizes Wikipedia only.

4.1 Client

The tag-cloud client component is implemented on an Android Smartphone using the Android SDK v1.5. The Java-based Android SDK provides the Java 5 compliance level, which facilitates the use of nearly all Java libraries written for standard PC environments. In order to utilize the Java ObjectStream class to send and receive the tag-cloud data, the server component is also written in Java. Furthermore a shared tag-cloud object specification is needed, which is covered by the OpenCloud Java library (http://opencloud.sourceforge.net) used on the client as well as on the server side.

For the purpose of visualization and interaction the MapView and MapOverlay classes, provided by the Android Google API Add-On, are extended. As the name reveals the MapView class provides a map view based on GoogleMaps. The tag-cloud is visualized by extending the MapOverlay class. For resource saving purposes a received tag-cloud is precompiled into a picture object. Various output filters (e.g. alphabetic or score-based ordering) can be applied to the tag-cloud to manipulate the shown tags in the displayed picture. The user interaction is provided by extending the MapOverlay to a MarkRegionOverlay class. The MarkRegionOverlay enables the Smartphone's touch capability to let the user mark a region the tag-cloud is created on.

4.2 Server

As mentioned before the tag-cloud server component is implemented in Java as well. The server component consist of three modules with the corresponding tasks: *network communication*, *harvesting data* as well as *data filtering and tag-cloud creation.*

The network communication module utilizes the Java ObjectStream class to exchange data with the client. The following simple communication protocol is used between the client and server nodes, as shown in Figure 3. After the TCP/IP connection is established, both nodes exchange version information to guarantee compatibly. Following this, the client node transmits the coordinates and region buffer values for the desired tag-cloud *(1)*. After successfully creating the tag-cloud *(5)*, the server sends it back to the requesting client *(6)*. Finally the connection is closed and the server is listing for incoming client connections again.

The second server module is responsible for harvesting the data for the desired location/region. At first the module queries the Wikipedia World gazetteer with the given region properties: latitude, longitude and region buffer *(2)*. In result all article names for this region are returned. Finally the entire content of those articles are harvested using the Wikipedia API (http://de.wikipedia.org/w/api.php) *(3) + (4).*

The third module, utilizing the OpenCloud library, creates a tag-cloud from the previously harvested Wikipedia data. Various input filter mechanisms are supported by the library. So far the following filters are used in our prototype:

- *blacklist filter*: discarding all words matching a predefined blacklist
- *length filter*: discarding words shorter than 4 letters
- *score filter*: discarding all words with minor occurring-score

The blacklist contains Wikipedia specific patterns, like "edit", "hide" or "contents". They represent common patterns of the Wikipedia user-interface and appear in nearly all articles. This high occurrence results in a high tag score, while the tags are of minor relevance. For this reason the blacklist filter discards these words.

Following the blacklist filter, a length filter is applied. As threshold we choose a length of 4 letters, meaning all tags shorter than 4 letters are removed from the tag-cloud. It's the simplest way to remove most pronouns, verbs and articles from the tag-cloud. As the blacklist patterns, these tags get a high score, while they are of minor relevance. This admittedly restrictive filter is easy to implement, but implies in the drawback of discarding short names and abbreviations, which might be of relevance, as well. Therefore the filtering of pronouns, verbs and so on demands further research. For instance a suitable verb filter, discarding all verbs, could be applied.

The last applied filter is a score filter discarding all words with a minor occurring-score, which means in our case they are of minor relevance. This shrinks the tag-cloud to the highest

scored tags and therefore saves valuable bandwidth during the transmission back to the clients Smartphone.
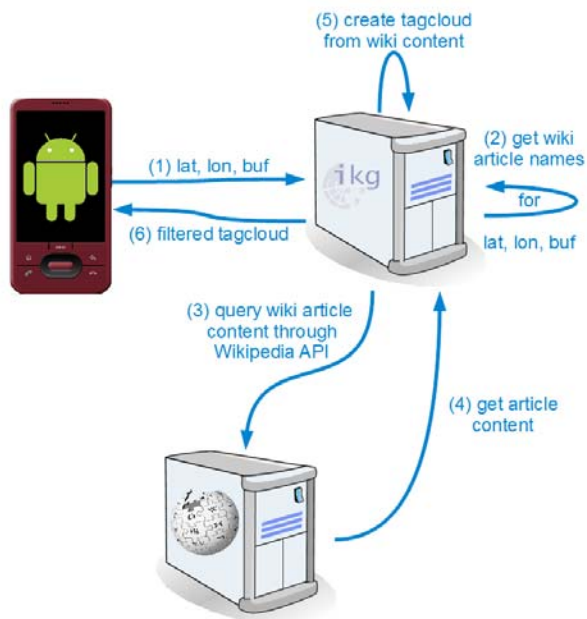

Figure 3: Client - server configuration and dataflow

5. EXAMPLES

As stated before we implemented a prototype on an Android based mobile phone. Figures 4 – 7 are showing some screenshots of the implemented application, demonstrating the user interaction, server communication und tag-cloud visualization parts.
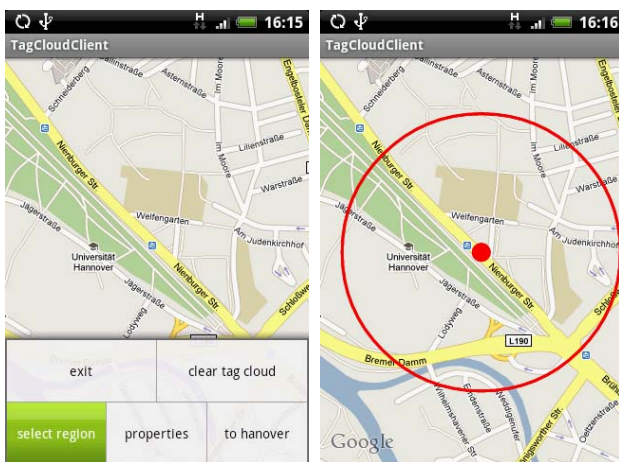

Figure 4: Query user interface (left); Select area dialog (right)

Figure 4 (left) shows an example of a possible user interaction. The user activates the *select region* option menu item. This enables the touch capability, so the user can select a region to build the corresponding tag-cloud.
After the user has marked a region, as shown in Figure 4 (right) the associated coordinates and the region radius are presented, as shown in Figure 5. After the user has confirmed the region, the region values are sent to the server. A progress dialog is

shown to indicate the server harvesting the corresponding Wikipedia content which could take some time, depending on the size of the selected region.
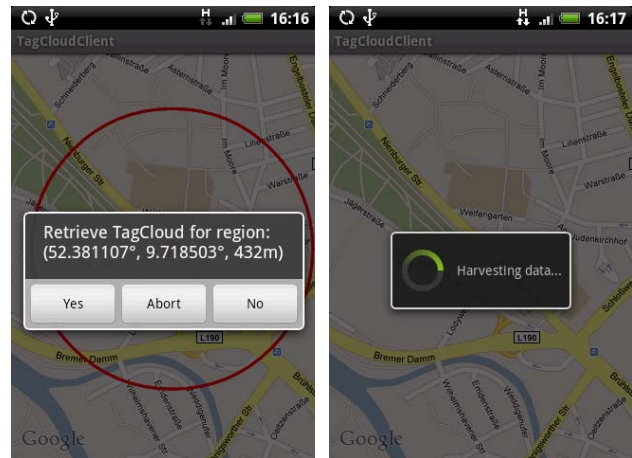

Figure 5: Harvest tag-cloud data dialog

After the server has finished harvesting the data and building the filtered tag-cloud, the tag-cloud is sent to the client device. The client compiles a picture from the tag-cloud and adds it to the map overlay, as shown in Figure 6.

Figure 6: Visualization of generated tag-cloud

As described before the approach is not limited to individual points. Tag-clouds can be calculated along a route (either for decision points along the route, which is desirable if the tag-clouds are to be used to support pedestrian navigation or in a spatial buffer along the route if information on the surrounding area is of primary interest). Figure 7 shows a route visualization with tag-clouds for two decision points.

Figure 7: Visualization of route

6. RESULTS AND OUTLOOK

In order to validate the results from the given example, shown in Figure 6, we marked the locations of the presented tags in Figure 8. The tag-cloud from Figure 6 presents, among other, the tags: *Gottfried, Wilhelm, Leibniz, Universität, Uni-Hannover, Hochschule, Friedhof, Königsworther, Platz, Leine* and *Ihme*. Figure 8 shows their corresponding locations in the investigated area.

In contrast to the common Wikipoints overlay known from Google Maps or Google Earth, the generated tag-clouds present additional information, e.g. related events or related objects without an own geo-referenced article. For instance, the tag "Nikolai-Friedhof" (Nikolai-Cemetery) in the upper tag-cloud of Figure 7 is a Wikipoint, since it's the name of the Wikipedia article. This would be shown as a POI in the Wikipoints Google Maps overlay. Furthermore, the visualized tag-cloud presents the tag "Weltkrieg" (World War) as well. This tag would not be shown by the Wikipoints overlay, since it's not a geo-referenced Wikipedia article name for the investigated area. The tag gives the user the hint, that the marked area is related to one of the World Wars or there is a World War related object in that area.

We implemented and refined the tag-cloud visualization system as a client-server system that runs in part on mobile android devices and in part on a PC based server. The development experience using the Android platform was very constructive, especially compared to previous developments conducted for the Windows mobile platform: for instance, using Java on client as well as on server side made the prototype development quick and coherent. Furthermore, the use of an existing tag-cloud specification library (the OpenCloud library) simplified the

communication between the client and server nodes significantly, reducing the communication to simple exchanges of serialized Java objects.

Currently no open library for rendering tag-clouds exists for the Android SDK, we therefore implemented a simple renderer for test purposes that lacks many of the advanced visualization facilities provided by current PC based tag-cloud renderers. While this limited the variety of the tag-cloud visualizations used in our prototype it was sufficient to validate the concept itself.

In order to improve the visualization further research on possible tag-cloud visualization schemes aligned to mobile phone displays would be worthwhile.

In addition sophisticated filter techniques are another field of interest. First, filtering words is highly language depended, which means that each language needs its own filter setup. For instance verbs will nearly always need to be discarded, so a language specific verb filter would be a good one to setup. Second a filter needs to be aligned to the source of the used content (e.g. Wikipedia). Each content source needs its own blacklist filter to discard typical source dependent words like *hide* or *edit* in the case of Wikipedia.

Figure 8: Found tags in the area under investigation

Initial experiences with the system are encouraging. In areas well covered by spatially annotated Wikipedia articles like the central areas of Hannover between 40 and 80 percent of the tags are meaningful landmarks. Since tag-clouds support fast browsing of information this is usually sufficient to provide users with pointers to potential objects of interest. Initial test users got quickly used to the system and were positive about the results delivered. Even test users who knew an area well were often surprised by the detail of information available.

The intention of the first prototype described in this paper has been to validate the viability of idea. In the future we aim to address several areas for refinement:

First, we aim to experiment with more advanced filtering algorithms to increase the signal to noise ratio in the tag-cloud and to extend the system to other languages, notably English.

Second, we intend to extend the data used for harvesting tags to other sources of geo-referenced data beyond Wikipedia, e.g. by incorporating Flickr tags.

We are also working on more refined rendering and adaptive layout strategies for the tag-clouds. Other open questions include the best size, the optimal tag count and the best visualization style to combine the cloud display with maps. Thus, while the use of public data sources seems to be promising to fill the gaps in spatial coverage in current LBS there remain many open questions to be addressed by future work.

7. REFERENCES

Flickr, 2009: http://www.flickr.com/photos/tags/ (accessed 28 Oct. 2009)

Lamantia, J. 2009: http://www.joelamantia.com/ideas/ cartograms-tag-clouds-and-visualization/ (accessed 28 Oct. 2009)

Wikipedia, 2009: http://en.wikipedia.org/wiki/Wikipedia: WikiProject_Geographical_coordinates (accessed 28 Oct. 2009)

Coupland, D., 1995. *Microserfs.* Harper Collins.

Zipf, A., 2002. Location aware mobility support for tourists. Trends & Controversies. *IEEE Intelligent Systems*. Nov./Dec. 2002. pp. 57-59.

Baus, J., Cheverst, K. and Kray, C., 2005. A Survey of Map-based Mobile Guides. *Map-based Mobile Services Theories, Methods and Implementations*, Springer, pp.197-213.

Schilling, A., Coors, V. and Laakso, K., 2005. Dynamic 3D Maps for Mobile Tourism Applications. *Map-based Mobile Services Theories, Methods and Implementations*, Springer, pp. 233-244.

Viégas, F.B., Wattenberg, M., 2008. Tag Clouds and the Case for Vernacular Visualization. *ACM Interactions,* Vol. 15, No. 4, pp. 49–52.

Hecht, B., Raubal, M., 2008. GeoSR: Geographically Explore Semantic Relations in World Knowledge. *Proc. AGILE Conf. 2008*, pp. 95-113.

Kaser O., Lemire, D., 2007. Tag-Cloud Drawing: Algorithms for Cloud Visualization, CoRR, .

Milgram, S. *Environmental Psychology: People and Their Physical Settings*, 2nd ed. Holt, Rinehart and Winston, New York, USA, 1976, pp. 104–124.

Halvey, M. J. and Keane, M. T., 2007. An assessment of tag presentation techniques. *Proc. ACM WWW '07*, Banff, Alberta, Canada, May 2007, pp. 1313-1314.

Rivadeneira, A. W., Gruen, D. M., Muller, M. J., and Millen, D. R., 2007. Getting our head in the clouds: toward evaluation studies of tagclouds. Proc. ACM CHI '07, San Jose, California, USA, April 2007, pp. 995-998.