Automatic Annotation Placement for Interactive Maps

Daniel Kaneider, Thomas Seifried, Michael Haller
Media Interaction Lab, University of Applied Sciences Upper Austria
Hagenberg, Austria
{daniel.kaneider | thomas.seifried | haller}@fh-hagenberg.at

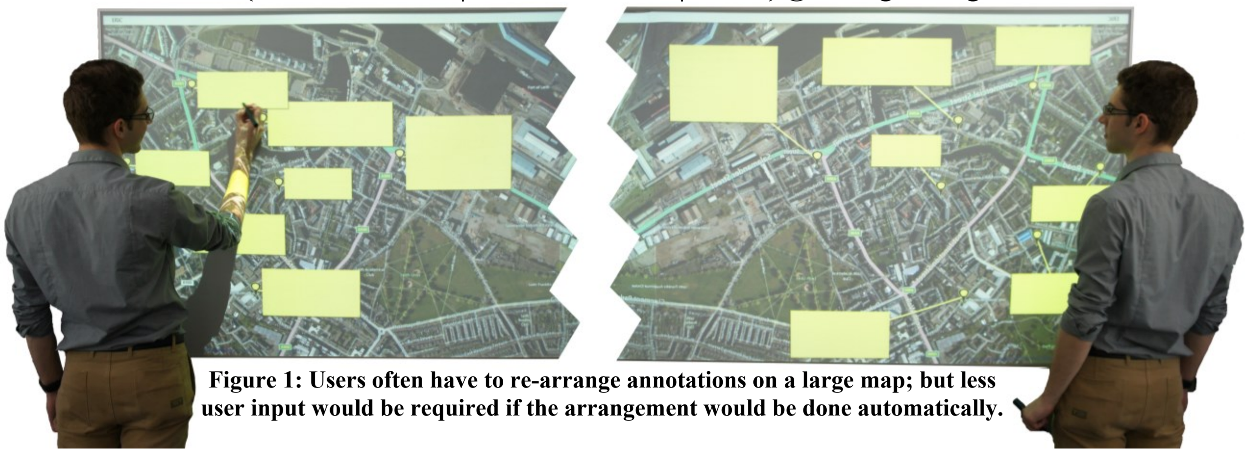
Figure 1: Users often have to re-arrange annotations on a large map; but less user input would be required if the arrangement would be done automatically.

ABSTRACT
Situation maps play an important role in planning and decision making in emergency response centers. Important information such as operating units or hazards is usually shown on these maps. Arranging the huge amount of information can be challenging for operators, as this information should always be visible while not occluding important regions of the underlying geographic map. As large interactive whiteboards are increasingly replacing traditional analog maps, new ways to assist with arranging information can be provided. In this paper, we present a new approach for placing annotations automatically on top of a map. For finding the optimal placement, our metrics are based on geographic features, on the users' sketched input, and on the annotations geometry itself. Moreover, we also added additional features to minimize occlusions for multiple annotations. First results were highly promising and showed that our approach improves input performance while keeping an optimal view of the map.

Author Keywords
GIS; annotation placement; large display; situation map

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

ITS '13, October 06 - 09 2013, St Andrews, United Kingdom
Copyright 2013 ACM 978-1-4503-2271-3/13/10…$15.00.
http://dx.doi.org/10.1145/2512349.2512809

INTRODUCTION
Operators at an emergency response center work in a stressful environment. They often need to coordinate rescue squads from different organizational units on varying levels of detail. Hence they facilitate large maps to depict the current situation. These so-called *situation maps* play an important role for planning and decision making. Such maps usually consist of the geographic map, superimposed by additional data (*annotations*), such as mission symbols, drawings, handwritten sketches, or geotagged photos coming from external devices. Even though emergency response centers often use large displays to visualize the current state of an operation, maps are mostly used in the traditional, analog way, because simple projected screens offer limited ways to interact with them. With the development of large interactive whiteboards (see Figure 1) this situation may change. Since direct interaction with whiteboards is possible, they have a high potential for supporting collaborative data analysis and decision-making [21, 23]. However, arranging mission symbols and annotations is still necessary, and requires a considerable amount of time. This becomes particularly evident as the number of required annotations increases.

In order to better understand how annotations are used and consequently in which way the placement could be automated, we conducted a background study. In three meetings at the police department of Linz (Austria) we held informal interviews with five officers experienced with working on situation maps. The results were collected in the form of notes by the interviewers. Additionally we observed their current usage of analog maps during a training session in which 10 participants (1 female) attended. The entire session was recorded on video and the observers took notes. The observation was concluded with informal interviews. We identified the following different annotation types:

- *Point annotations* which provide additional information about target objects at certain points.
- *Geotagged photographs from external sources* that support the operators to get a deeper understanding about certain locations.
- *Sketched information on the map* which is mainly used by operators to highlight areas on the map by drawing strokes. Such areas could be used to designate zones with different access levels during a police operation.

The overlaid information helps operators get a better understanding about the current mission and is an important source to improve their situation awareness. Digital situation maps are capable of seamlessly integrating the rising amount of existing digital data. Since operators in emergency response centers are under extremely stressful circumstances, the design of a user-friendly digital situation map that minimizes visual clutter is highly important. This is especially critical with a high density of annotations on the map, and also with important information (e.g. an important intersection, an important building etc.). For example, Figure 2 depicts a typical scenario of a situation map. Not surprisingly, the huge amount of data results in a cluttered map that most operators have difficulties using to get an understanding of the big picture at a glance. Therefore, all data should be clearly visualized and annotations should not occlude information that might be highly relevant for operators.

In this paper, we mainly focus on how the operators are supported by using an automated annotation placement approach. It is highly important to provide a solution that offers a fluent and efficient interaction with annotations on the situation map. This is because the operators have to focus on the mission itself. Manually placing the labels is a time consuming task. Looking back to cartographic history, the label placement was very cumbersome for human cartographers and an intensive task that can take up to 50% of the total map production time [25]. This is a problem in a time-critical environment. Therefore, we explore a novel approach for optimal annotation placement based on analyzing the background data of the map. By identifying important and less-important regions, we find the optimal space, where the annotation labels should be placed first. The potential of this design has been explored by doing a first test with police officers using our setup extensively. To explain the context for this work we first present the related work. Next, we present both the interaction techniques as well as the implementation of our approach. Finally, we present the overall results and discuss them by providing an outlook for future work.

RELATED WORK

The automatic placement of labels is a well-known topic in cartographic research [9, 10, 3]. Our work focuses on automatic annotation placement on maps.

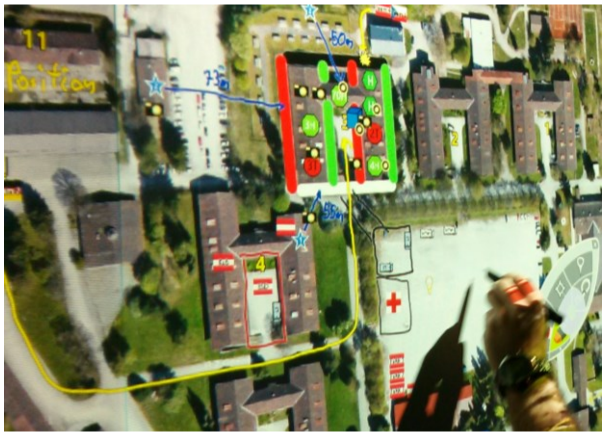

Figure 2: So-called situations maps often result in a cluttered view, where annotations are occluding others, but also areas that are highly relevant.

As depicted in Figure 1 , annotations consist of a placemark icon, the anchor point, and a label containing the textual or graphical content. Annotations often have a solid background which ensures they are readable regardless of the underlying background. Usually the content and placemark icon are connected using a so-called *leader line*.

Annotation placement

Previous research on automatic placement of annotation labels primarily focused on 2D diagrams and technical drawings as well as 3D content [18, 20]. As with static labels (see section below), the goal of these algorithms is to place labels in a way that results in the fewest number of disturbances as possible. In [13] the background of diagrams is considered for label placement. Two approaches were suggested to consider the shape of a diagram: an image- and a vector-based approach. Another approach – presented in [14] – considers depth information of 3D scenes in order to improve the quality of annotation placement. However, in all these works, the annotated content either had clear boundaries (there was "empty" background that could be used for annotation), or content occlusion was not considered at all.

In the past, the automatic placement of annotations on maps has not been studied deeply. A first approach in this area was proposed by Wu et al. [24]. In this work a computer vision algorithm was facilitated to place image annotations in the empty areas on a metro map. However, in comparison to topographic and satellite map data, metro maps show a very limited amount of data. Hence, annotations can be placed without any occlusion. Moreover, the vision-based approach only detects whether areas are occupied by map-data but it does not take into account the importance of specific map features.

Our placement is done by using information from the openly available vector-based map database OpenStreetMap (OSM)[1]. We get the background information which contains meta-information about a huge collection of map features. Similar to Jones' [8] work on static map labeling, this meta-information can be used to classify the importance of certain areas on the map and therefore can influence annotation placement. It has been shown in recent research that OSM data is complete enough to be used for a variety of implementations: Kohtake et al. published a system for indoor and outdoor positioning relying on OSM data [12]. Traffic simulation worked well in a small-scale scenario by also using OSM data [26].

In our scenario, annotations are placed dynamically. Therefore, every annotation will occlude certain content on the cartographed map. Moreover, the visual representation and aesthetics of *static labels* in a map differs from *annotations* on top of a map. Static labels are placed close to the corresponding feature on the map. The association between feature and text is usually done via proximity [2] rather than use leader lines [9].

Map labeling

The concept of our idea has some similarities to the classical map labeling. Imhof [7] and Yoeli [25] summarized rules that are often used by human cartographers. Their goal was to automate the process of manual cartographing. According to Imhof [7], the cartographic conventions done in 1960 specify three general kinds of labels to be placed:

- *Point features* describe point-like objects,
- *Line features* are linear objects on a map. They are mostly used to depict different kind of roads, and
- *Area features* have a bigger extent and can be described using polygons.

Traditionally, map labeling has to deal with all of these three problems. Point Feature Label Placement (PFLP) is a subset of this [4]. Our problem shares certain similarities with PFLP, because we have annotations around point-like objects.

Automatic label placement on static maps

The first rule-based algorithms used the conventions of the human cartographers. The placement for PFLP was mainly at fixed positions around the anchor point [25].

The Greedy algorithm is based on heuristics to find an optimal solution. The algorithm is neither able to escape from local minima nor does it perform well using complicated data sets [4]. An alternative to the heuristic approaches are genetic algorithms. Many genetic algorithms claim to give good results [6, 1]. However, they need to be highly adapted for each problem domain.

[1] www.openstreetmap.org

The approach we use in our case is based on simulated annealing [11]. It is a heuristic algorithm and very often used for label placement [3]. It uses the notation of temperature of solid materials from physical models [10]. Simulated annealing uses a declining temperature to escape from local minima [17]. The algorithm is inherently sequential and therefore slow for large problems [17]. But, in the case of map annotations, the label set is small compared to the number of labels in traditional cartography.

Summary

In this paper, we do not present a new map labeling algorithm for static maps. Instead, our approach places annotations on existing cartographed maps. This placement is done by extending existing quality measures based on data from OSM. Further, we introduce a new metric for map features. They are based on Van Dijks' formalized quality criteria [22] to measure the qualitative performance of label placement algorithms. So, we are able to consider important features of the underlying map such as roads or buildings. The implementation uses the commonly known simulated annealing approach.

INTERACTION TECHNIQUES

Simple interaction

In our application, operators can interact using an on-screen menu (cf. Figure 3, *top left*). A placemark can be pinned on the map (Figure 3, *top right*). Text can then be written in the associated annotation field. When the operators have finished sketching (Figure 3, *bottom*), the annotation is automatically placed at an "optimal" position. Similarly, once the user edits an existing annotation, we again search for the optimal placement.

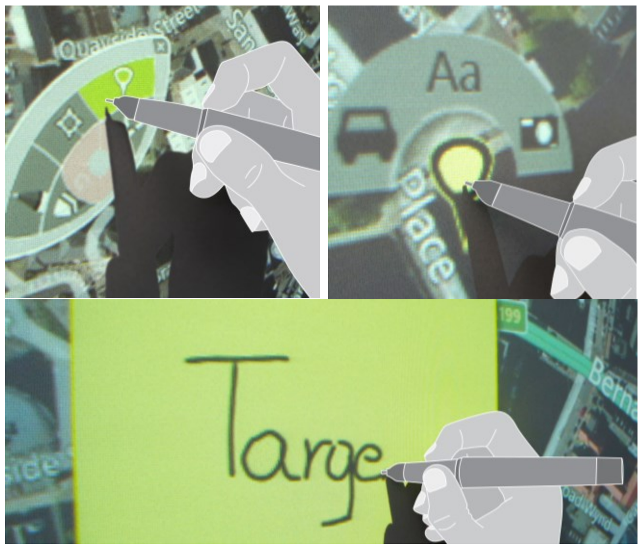

Figure 3: The tool palette (*top left*) for placing textual annotations (*top right*) or drawing sketched content on the map (*bottom*).

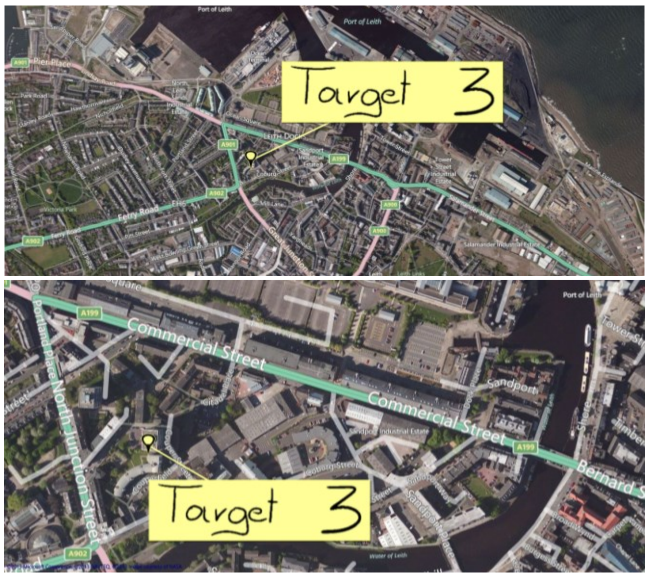

Figure 4: Once the operator is zooming in (*top*), we re-calculate the optimal position of the annotation according to the additional data of the background to avoid overlap with important areas (*bottom*).

The operator can also draw directly onto the map. This, as opposed to the point-based annotations, is not meant for writing text, but to highlight buildings or areas on the map as shown in Figure 2. Finally, icons (e.g. cars) can be placed on the map. They denote the current or future location of oper ational units, hazards or any other important item.

Zooming in/out

An automatic placement also happens during panning and zooming of the map. Zooming into a certain region on the situation map can become complicated since additional *(important)* information is getting further occluded as depicted in Figure 4. At a lower zoom level, more annotations are visible on the map, thus making it more difficult to find the optimal placement. The system then re-arranges the annotations in order to keep an optimal visibility. Additionally, the user can still trigger the arrangement manually.

Automatic Annotation Placement

As mentioned before, all annotations that are super-imposed on top of the satellite view are arranged automatically. So, operators never have to think about optimal spaces, but can fully focus on the task. Even if they zoom-out, the arrangement will be done again to provide an optimal view.

IMPLEMENTATION

Figure 6 depicts the main components of our annotation placement approach. We use different input data for finding the best place. This set of data, according to Christensen et al. [4], has been called *search space*.

In our algorithm, the search space is generated from three data sources: low-level map features, user-sketched content on the map, and the annotation geometry. For the searching process, we pick randomly chosen positions for each geometry which we call "candidates". Next, the candidate's position and arrangement are rated. Therefore, we calculate different *metrics,* such as map features rating, distance rating, anchor-annotation overlapping rating, annotation overlapping rating, and leader line crossing rating. These individual values are aggregated (different rating values and multiple annotations' values into one value), weighted (multiplied with a constant), and combined, resulting in a float value. This is done with the so-called *quality function*. If a candidate's rating is better than all previous ones, the candidate is temporarily stored. For the next step in the loop, we calculate the next new candidate position and repeat the process for a fixed amount of iterations (currently we do so 1000×). At the end the candidate with the best rating is used to place the annotation on the map.

In our approach, we consider all visible annotations – thus, we neither try to close them nor resize them. Currently, such annotation changes are devolved to the user. This lets us focus on the placement algorithm.

Map Features

In our case, we combined the satellite/hybrid map with a layer that comes from OpenStreetMap (OSM).

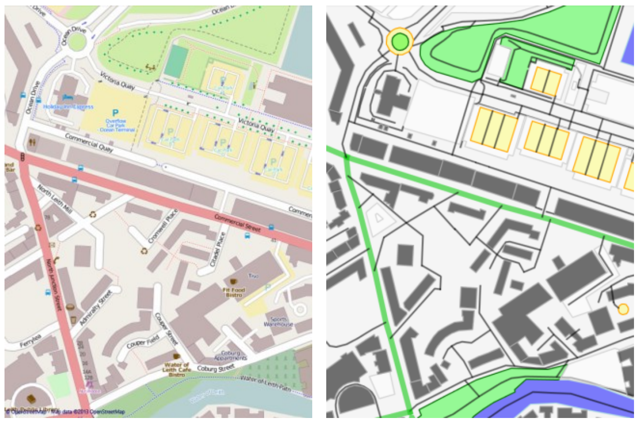

Figure 5: An OpenStreetMap map excerpt of Edinburgh comparing a rending from OSM *(left)* and one using our own rendering, which is based on line and area features *(right)*.

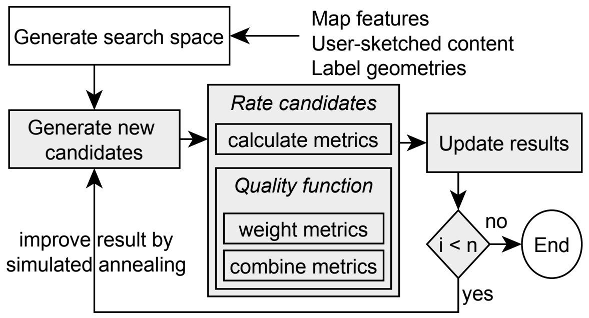

Figure 6: The diagram shows the overview of the implementation, from the input to the final result.

Having the possibility to access the OSM data provides a huge advantage over the image processing approach as used in [19]. With the OSM data, we reliably know where map features, such as roads or buildings, are located. Furthermore, the OSM data includes detailed information on the maps content and metadata such as the type of a road or the usage of a certain house.

Figure 5 *(left)* shows a map excerpt rendered from OSM. Starting from the OSM data, we then implemented a simplified renderer that only highlights all line and area features, as depicted in Figure 5 *(right).* Once we know where specific streets, buildings or areas are located, we weight them. We used the metadata in terms of a key-value string stored along with the line and area features. For our *map features*, we used these key-value pairs that are provided by the OSM community to categorize data[2], e.g. building-hotel, building-hospital, highway-motorway. The primary map feature includes different kinds of roads. They vary from busy highways to small footpaths with many subdivisions in between. In our current implementation, we distinguish 11 different map features ranging from buildings, roads, to military areas.

Weighting map features

The weighting is done according to the importance of an object. For example, main streets, and especially intersections are given more weight than to smaller roads. Thus, they should not be occluded by annotations.

The weights we used for the line and area features are based on interviews taken during the background study and user feedback sessions. They are partly influenced by *taginfo*[3], which shows the overall usage of all keys and values in OSM. Additionally the weights have been tweaked by feedback acquired from emergency room operators. We found that natural physical land features (e.g. forests, nature reserves, parks, lakes, or rivers) should have a very low weight (meaning that these objects could be occluded by annotations). Moreover buildings, consuming 44.71% of all area/line elements, should not be occluded so often. Very important objects (e.g. *military/hospital*) get a very high importance on the map, although they are not often used based on *taginfo* (with less than 0.1% of all area features they are rarely used). However, this weighting can differ between different units of the police and depends also on the case. Usually larger roads are more important than smaller ones, but special units operating on a building-scale, are more interested in accessing roads. So, waterways are not important, unless divers are on duty. Furthermore, police officers suggested that the size of a map feature covered is not much relevant for its weight, but the number of objects occluded should be minimized. However, given the fact that bigger objects in OSM are often decomposed into smaller ones, the object size is indirectly considered. Moreover, a simple check for geometry intersection is faster than computing the exact area of intersection and can be easily applied to both area and line features.

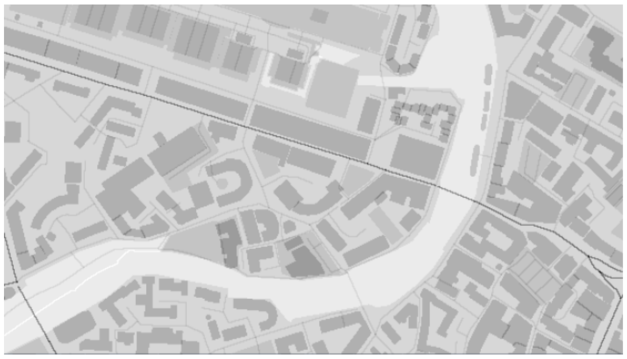

Figure 7: The line and area features are weighted accordingly and resulting in a gray-shaded rendering where more-important features are darker than less-important features.

Figure 7 depicts a gray-shaded version of both the line and area features based on the weight. Dark-gray areas highlight regions that are considered more important than others. The river, for example, is lighter gray-shaded than the rest of the map, resulting that the river might be a good candidate to be occluded by annotations.

User-sketched content

As mentioned before and derived from the requirements, users can sketch directly on the satellite view and highlight very important regions (see Figure 8). Generally, these regions are highly relevant and therefore they should never be occluded by any annotations. Hence, they can have a strong influence on the placement algorithm.

Annotation geometry

The final input for defining the search space is the annotation geometry itself, which consists of the anchor shape, the annotation's content, and a line between the content and the anchor (see Figure 10), the leader line. The geometry has to be considered due to the fact that we primarily want to arrange the annotations, and moreover we have to avoid occlusions

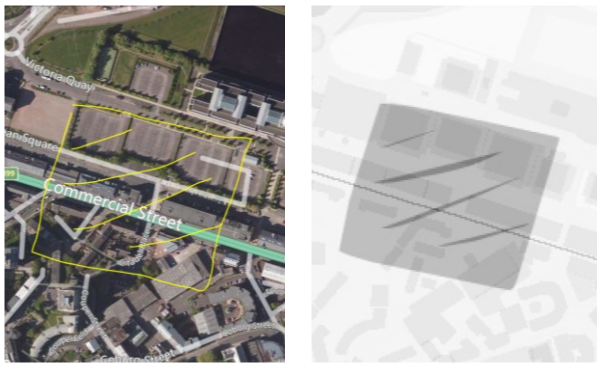

Figure 8: A user-sketched drawing highlighting an important area should not be occluded by any annotation (*left*). Therefore, the weight for such areas is rated high (*right*).

[2] http://wiki.openstreetmap.org/wiki/Map_Features

[3] http://taginfo.openstreetmap.org

of multiple annotations, as well as keep a short distance between the anchor and the annotation's content.

Generating the search space

According to Christensen et al. [4], an element of the *search space* is a function from line/area features to label positions, which they call labeling. The set of potential label positions for each of these features characterizes the search space [4]. We use different attributes of the objects on the map and information about the current screen for defining the optimal search space. Figure 9 depicts a simple example with different buildings and areas. The yellow box in the middle depicts the potential place of an annotation. The question now is, whether the location of the annotation is already good or if there is a better place for it. For defining the quality of the current location, we used different attributes (as they have been described previously).

Annotation candidate positions

The current approach is based on a simulated annealing as presented in [11]. It can be seen as a mid-range performing algorithm with a general and easy to implement approach [3]. A new candidate position is generated randomly, close to the current best annotation position and within the current screen. As with the simulated annealing algorithm, the new positions get closer to the current best position with the increasing number of iterations. Our approach tries to cover the whole search space since we do not want to privilege certain metrics. This allows us to examine them individually.

Metrics

Metrics are an important way to define the overall quality function. Basically, in each metric we look at a specific aspect of an arrangement and return a value in the interval [0, 1], 0 (bad) to 1 (good). The concept is similar to the one presented by Van Dijk [22]. However, we had to define our own in order to consider textual point annotations and map features from OSM. In our implementation, we used the following independent metrics to compare the quality of an annotation's position.

Map features rating

In the example of Figure 9, the annotation overlaps three objects: a building, a dock area, and a lake. Therefore, we determine the weights of the overlapped area/line features,

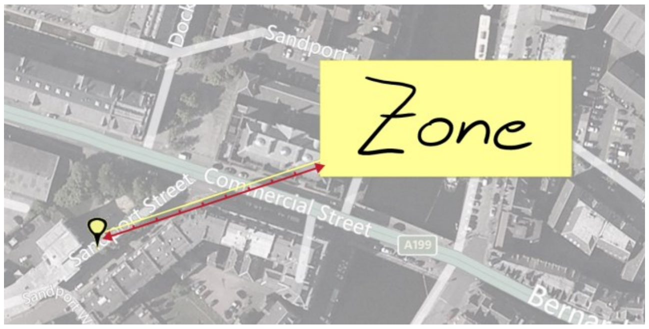

Figure 10: The annotation geometry consists of the anchor, the annotation itself, and the line between the anchor and the annotation.

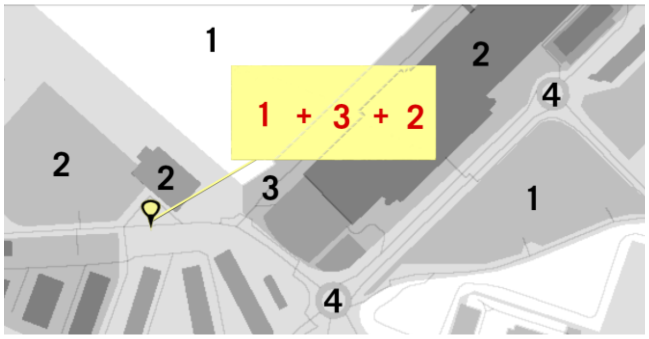

Figure 9: Three map features (building, lake and a dock) with associated weights are overlapped by one annotation.

which will then be summed up afterwards. In our example, the sum would be 6 (= 1+3+2). The normalized value of this metric is finally calculated using the maximum and minimum encountered value of the current viewport.

User-sketched content rating

User-sketched content is treated as other map features, just with higher weights (since they are highly relevant and therefore they should not be occluded by an annotation). Another difference is that we always take the convex hull of the strokes, since users tend to draw open polygons.

Annotation geometry rating

As mentioned before, the annotation itself also has a strong influence of where to place the annotation. On the one hand, the annotation should be placed close to the anchor (the users' input), on the other hand, it should not occlude parts of the anchor. Therefore, we introduced two additional ratings, the distance rating and the anchor-annotation overlapping rating.

Distance rating: An important metric is the distance of the annotation's content box to the anchor point (see Figure 10). Obviously, this distance should not be too long. Again, this value is normalized. A value of 1 (good) means the box touches the anchor point, 0 (bad) means the Euclidean distance is greater than 75% of the viewport height, and a fraction is used for values in between. The fraction was used to ensure resolution independency.

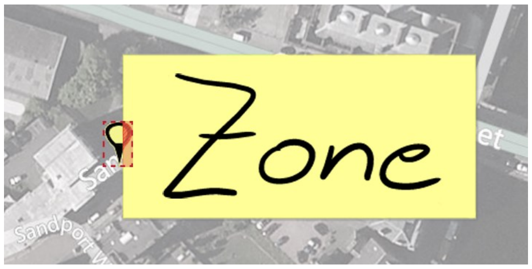

Figure 11: The anchor-annotation overlapping rating is 0.75 in this example, since ¼ of the anchor area is covered.

Anchor-annotation overlapping rating: While it is good to keep the annotation close to its anchor point, it should not overlap it. Figure 11 shows an example, where one quarter of the anchor area is covered by the annotation. Therefore, we introduced an additional metric, the anchor-annotation overlapping rating, where we look at the area of intersection between the anchor and the annotation, calculated as follows:

$$1 - \frac{\text{area of anchor-annotation intersection}}{\text{area of anchor}}.$$

Metrics for multiple annotations

The previous metrics are used to describe certain attributes, which apply when only *one* annotation is placed on the map. However, more problems arise once *multiple* annotations have to be placed simultaneously (see Figure 12). We therefore introduce additional metrics to solve these problems.

Annotation overlapping rating

The biggest challenge with multiple annotations is the overlap between each of them. The overlap not only happens between different annotations, but also if one of the annotations occludes one of the anchors of another one. In a first version we implemented a metric based on the ratio between intersecting area and the annotation size, resulting in too many overlaps that occurred. In the current version, the size of the intersecting area is not considered anymore; every overlap is rated with the lowest value of 0. Finally, the mean for an annotation overlapping, or not overlapping others, is taken. In the example depicted in Figure 12 (*left*) the rating would be 0, since every annotation overlaps another.

Leader line crossing rating

Leader lines should not cross each other as depicted in Figure 12 (*right*). If a line intersects another line, it gets the rating 0; 1 otherwise. If more than two annotations are on the map, we simply take the mean for all other lines.

Aggregation

Each of the metrics, as presented before, should return one single float value for a current arrangement of annotations on the map. We know, for example, how the *distance rating* is calculated for one annotation. But we might have more than one annotation on the map, for example 10. Therefore, we need to aggregate 10 distance ratings to get one single value. In this case, we simply take the mean of all values.

Some metrics like the *distance rating* have, however, a high importance. Assume a scenario where there are ten annotations in total, where eight annotations have a *distance rating* of 1 (meaning that they touch their anchors) and two are far away from the anchor (*distance rating* of 0). An arithmetic mean would result in 0.8. As we want to give outliers a higher weight, we are using the power mean. For distance, annotation overlapping and leader line crossing ratings the power mean is used, for the other features, we use the arithmetic mean.

Quality function

The *objective function* as described in [4] is also referred to as *quality* [22, 7]. It puts all values for each metric together. The overall quality function can be defined as:

$$quality = \sum_{i=1}^{n} m_i \cdot w_i, \text{ where}$$

all n individual metric values m_i are taken and multiplied by additional weights w_i, which we measured empirically (during our observation study with the police officers). These weights define the influence of each of the metrics for the final quality.

Iterations

The goal, to find a better arrangement, is done by maximizing the overall quality value. The main loop of the search algorithm consists of generating a new candidate at a random position, rating the new annotation arrangement and combining it with the previous state. After each iteration we only accept a new candidate if the quality value is higher than the previous ones. The search loop is repeated for a fixed number of times, making a tradeoff between the quality of a found solution and its runtime. The goal is to find an adequate solution within a short amount of time.

Temporal cohesion

The placement algorithm cannot just be triggered by the creation of new annotations, but also by changes of the map. In the case where the user zooms in or out the map, we perform an arrangement of all annotations when the user action is finished. To ensure temporal cohesion, the current position of an annotation is used as the first candidate and is also given a higher weight. Moreover, all annotations are animated towards their new position. On the contrary, annotations will

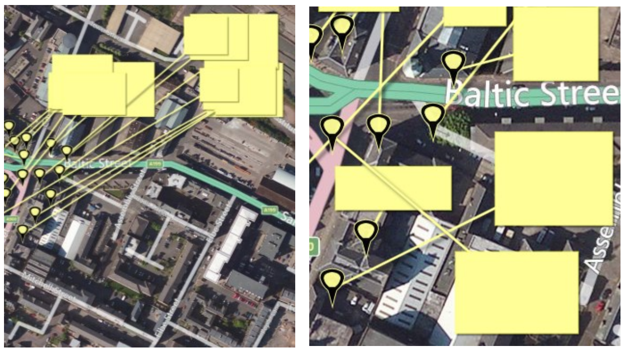

Figure 12: Annotation placement without considering overlapping (*left*) and line crossing (*right*).

remain untouched when the map is panned. In that case they will occlude exactly the same area on the underlying map as before. Hence a re-arrangement is unnecessary.

RESULTS

All the tests in this section have been done using the following weights w_i: 0.2 (*map features rating*), 0.1 (*distance rating*), 0.1 (*anchor-annotation overlapping rating*), 0.1 (*user-sketched content rating*), 0.5 (*annotation overlapping rating*), 0.4 (*leader line crossing rating*). They have been adapted according to user feedback. Distance and user-sketched content are more important than map features. But since the distribution of values for map features is mainly in the center, the weight value is higher. Overlappings and leader line crossings are perceived as very bad by the users, giving them higher weights.

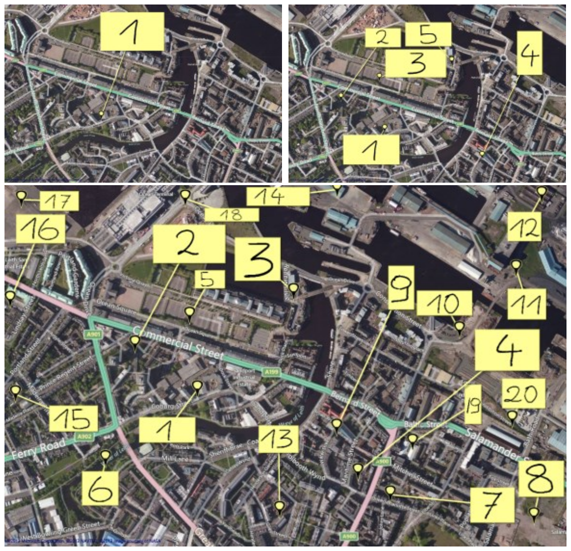

Figure 13: Arranged annotations using 1 (*top left*), 5 (*top right*), and 20 annotations (*bottom*) respectively.

The annotations in Figure 13 are neither placed on top of street crosses, nor on the main road. Instead, they are placed close to the waterway thanks to the quality function. Figure 15 (*left*) depicts an extreme case, where 17 annotations are positioned in a very small area. However, they do not occlude the main road.

Our approach is based on a composition of different weights, which obviously influences the results. Annotations are preferably placed on natural map features (e.g. waterways). Once a user-sketched drawing is made, the annotation gets placed along the outside, cf. Figure 14.

On the other hand, the quality of annotation placement gets exceedingly worse, if important weights are ignored. Figure 12 depicts two different scenarios. On the left side, the overlapping has been allowed, resulting in too many annotations located on the same place. Figure 12 (*right*), shows a scenario, where the crossing of the leader lines is allowed. Again, for the operator, the result is difficult to interpret.

As discussed before, zooming can lead to annotations covering important map features. Figure 16 shows how annotations are arranged differently before (*left*) and after (*right*) zooming out of the map. Again, the main roads are not occluded. Zooming-in results in a more detailed map. Therefore, we can easily re-calculate optimal positions for the annotations.

DISCUSSION

During our first tests with the police officers, we noticed that the constant subtle re-arrangement of annotations while interacting with the map (e.g. while zooming) does not influence them negatively. Although we have not done a formal user study, we noticed that in general, the operators appreciated our approach. The comparison was drawn between a setup where the database, thus the placement algorithm, was not available. It led to occlusion of important map features

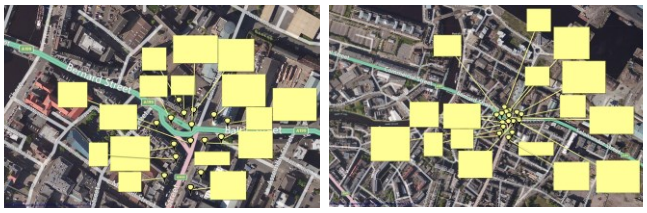
Figure 15: Annotations get arranged differently before (*left*) and after (*right*) zooming.

and other important content after creating just few annotations. Moreover, the operator liked not having to wait longer than one second for the optimal placement to be found. Annotation placement in urban areas is tricky, because all the map features are somehow important. As shown in Figure 17 features like roads and the highway are not occluded.

A considerable amount of research has been done recently to analyze the quality of OSM data compared to commercial databases. It was found that for example OSM in Ireland does not perform worse than Google or Bing Maps [5]. A comprehensive analysis of the data in Germany states that OSM provides 27% more data than TomTom's commercial dataset with respect to road data [16]. Since our approach makes intensive use of geographic features, a complete data set is more than welcome. We also observed less information provided for European rural areas. However, this just partially affects the outcomes of the weight function. On one hand, we do not just use map data as input, but we also consider overlapping between annotations and sketched user content. On the other hand, line features representing roads are often gathered, which are of great importance for weighting. Thus, despite partly incomplete datasets, an acceptable result can still be achieved. We also observed that the text labels from the underlying map are still mostly readable. Current map providers like Google/Bing usually place the names (e.g. road names) directly on top of map features. Therefore, if we tend not to occlude roads, we also do not occlude the corresponding names.

Performance

One of challenges was the huge amount of OSM data we had to deal with. We use the object-relational database management system PostgreSQL for handling the OSM data, since

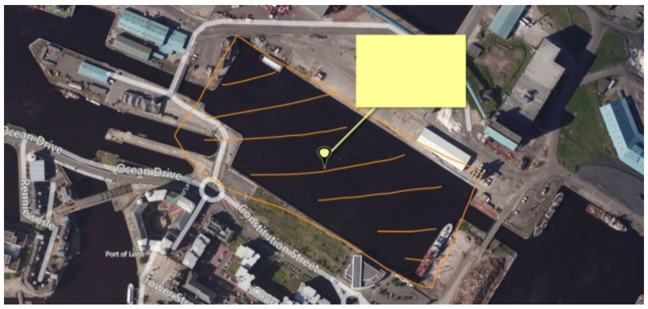

Figure 14: Sketched areas signal a high importance, thus annotations get placed outside.

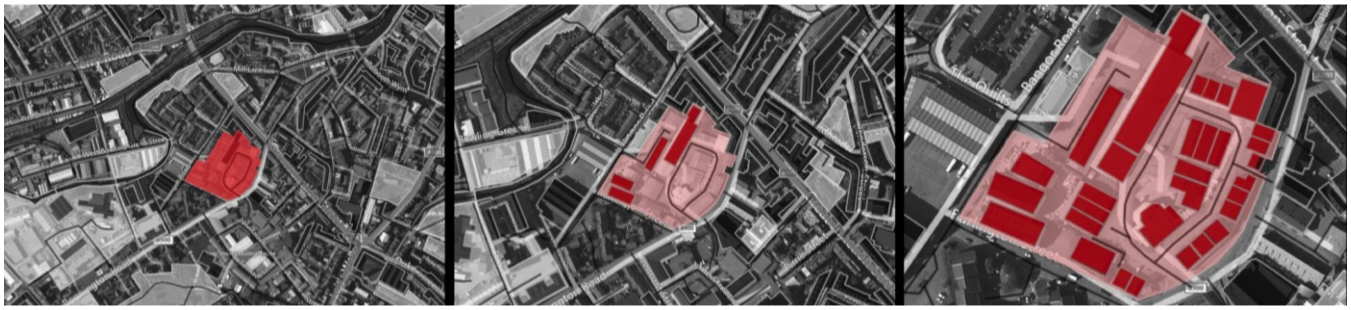

Figure 16: Zooming into the map results in a higher resolution of the area features. While the figure on the left only shows one polygon, the figure in the middle already depicts more buildings of an area. The figure on the right shows a detail map with all sub-buildings.

geographical queries are supported and backed by spatial indexes. The table structure uses the *pgsnapshot* scheme for storage and is optimized for analysis. It stores all metadata (e.g. type of area (building, public place), elevation, name, purpose, …) and corresponding bounding boxes as additional columns, which improves the performance while querying map data. Nevertheless, using an excerpt of a single country (e.g. UK) keeps the database size around 30GB. This means that retrieving geographical features from the database takes around 100 to 200ms on a PC with a HDD and 12GB RAM. If the zoom level was quite low (country or region wide), queries could take up to a few seconds. Thus, users might perceive the system as being not that reactive. We increased the performance by querying the database before actually arranging the annotations. When the operator, for example, sketches a new annotation, the database query is already triggered after the placemark has been pinned.

In addition, we implemented a kind of *Level Of Detail* (LOD) approach to improve the performance. Figure 16 depicts a scenario, where the operator is zooming in, resulting in a more detailed resolution of the building. While on the first zoom level (Figure 16, *left*) only the whole block of the building has been taken as a map feature, more buildings and streets within the block of the building are taken if the operator is zooming in (Figure 16, *middle, right*). We noticed that this approach still provides good results. In contrast, small details which cannot be seen anyway do not negatively influence the quality function. In some rare cases, however, roads in OSM are split into segments. Thus, when small segments are excluded from the search space they can result in roads with gaps.

CONCLUSION & FUTURE WORK

In this paper, we presented a new approach for placing annotations automatically on a geographic map when using an interactive whiteboard in an emergency response scenario. For finding the optimal placement, our metrics were based on geographic features, on the users' sketched input, and on the annotation's geometry itself. Moreover, we also added additional features to avoid occlusions while using multiple annotations. First results were highly promising and even on non-rural areas the automatic placement performs well.

In this work we focused mainly on the placement algorithm. We did not alter existing annotations by scaling or closing them (since the police officers requested to visualize all annotations all the time), although this might be an interesting extension for the future. Extracting content of handwritten text using OCR could reduce their size further and give better results since less area gets covered on the map. Similarly, it's the user's task to close the annotations if too many of them are in the current screen. Otherwise overlapping and occlusion of important map features are more likely to occur.

The weights of the quality function are sensitive components in our current system. A formal investigation on the influence of individual ratings on the overall result would be useful. A good trade-off between map features and the distance rating is currently unknown, although we favor the first. Moreover user hints could be considered. Do Nascimento gave examples for how this could look like [15]: users can customize the positions of some annotations after the automatic placement; thus, that could influence the weights of the quality function.

Additionally, the weighting of the features could vary, depending on the user's input. Currently, our approach avoids occluding roads and favors natural areas instead. It has some significance if the operator places an annotation directly on some grassland. In this case, the algorithm could change the weights accordingly and give natural areas a higher priority. At the moment, this is solved using sketched drawings on the map. However it is possible that the operator could place an annotation with a lower degree of precision than is expected;

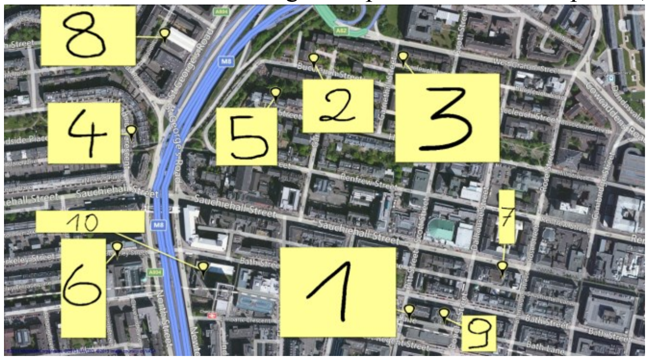

Figure 17: 10 annotations placed in an urban environment show that the main roads are not occluded.

hitting a road at a lower zoom level could be tricky. The system should therefore account for such inaccuracy to ensure high quality annotation placement.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the Austrian Research Promotion Agency (FFG) within the research program KIRAS under grant agreement n° 29335581 (ERiC).

REFERENCES

1. Bae, W. D., Alkobaisi, S., Narayanappa, S., Vojtechovský, P. and Bae, K. Y. Optimizing map labeling of point features based on an onion peeling approach. Journal of Spatial Information Science, 2 (2011), 3–28.
2. Bekos, M. A., Kaufmann, M., Symvonis, A. and Wolff, A. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry*, 36 (2007), 215–236.
3. Christensen, J., Marks, J. and Shieber, S. Placing text labels on maps and diagrams. in Heckbert, P. S. ed. *Graphics gems IV*, Academic Press Professional, Inc., (1994), 497–504.
4. Christensen, J., Marks, J., and Shieber, S. An empirical study of algorithms for point-feature label placement. *ACM Trans. on Graphics*, 14 (1995), 203–232.
5. Ciepłuch, B., Jacob, R., Mooney, P., and Winstanley, A. Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. *Proc. of the 9th Int. Sym. on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, (2010).
6. Hong, F., Kaijun, L., and Zuxun, Z. An efficient and robust genetic algorithm approach for automatic map labeling. *Int. Cartographic Conference*, (2005).
7. Imhof, E. Die Anordnung der Namen in der Karte. *Annuaire Int. de Cartographie II*, Orell Füssli Verlag, (1962), 93–129
8. Jones, C. B. Cartographic Name Placement with Prolog *IEEE Comp. Graphics and Appl.*, 9 (1989), 36–47.
9. Kakoulis K., Tolli I. G. Labeling algorithms. in Tamassia, R. ed. *Handbook of graph drawing and visualization*. Chapman & Hall/CRC, 2007.
10. Kern, J. P., and Brewer, C. A. Automation and the Map Label Placement Problem: A Comparison of Two GIS Implementations of Label Placement. *Cartographic Perspectives*, 60 (2008).
11. Kirkpatrick, S., Gelatt, C. D, and Vecchi, M. P. Optimization by Simulated Annealing. *Science*, 220 (1983), 671–680.
12. Kohtake, N., Morimoto, S., Kogure, S., and Manandhar, D. Indoor and Outdoor Seamless Positioning using Indoor Messaging System and GPS. *2011 Int. Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Guimarães, Portugal, (2011).
13. Luboschik, M., Schumann, H. and Cords, H. Particle-based labeling: Fast point-feature labeling without obscuring other visual features. *IEEE Trans. on Visualization and Computer Graphics*, Educational Activities Department, 14 (2008), 1237–1244.
14. Maass, S., Jobst, M., and Doellner, J. Depth Cue of Occlusion Information as Criterion for the Quality of Annotation Placement in Perspective Views. *The European Information Society*, Springer Berlin Heidelberg, (2007), 473–486.
15. Nascimento, H. A. D., and Eades, P. User Hints for map labeling. *Journal of Visual Languages & Computing*, 19 (2008), 39–74.
16. Neis, P., Zielstra, D., and Zipf, A. The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011. *Future Internet*, 4 (2011), 1-21.
17. Ram D. J., Sreenivas T.H., and Subramaniam K.. Parallel Simulated Annealing Algorithms. *Journal of Parallel and Distributed Computing*, 37 (2), 1996, 207–212.
18. Shen, J., Wen, Y., Wang, Y., Chen, M., and Su H. On Point Feature Automatic Annotation Placement in 3D Environment. *Proc. 21th ISPRS Congress,* 37 part B2 (2008), 981–984.
19. Stadler, G., Steiner, T., and Beiglbock, J. A Practical Map Labeling Algorithm Utilizing Morphological Image Processing and Force-Directed Methods. *Cartography and Geo. Information Soc.*, 33 (2006), 207–215.
20. Stein, T., and Décoret, X. Dynamic label placement for improved interactive exploration. *Proc. of the 6th NPAR*, (2008), 15–21.
21. Tobiasz, M., Isenberg, P., and Carpendale, S. Lark: Coordinating co-located collaboration with information visualization. *IEEE Trans. on Visualization and Computer Graphics 15*, 6 (2009), 1065–1072.
22. Van Dijk, S, Van Kreveld, M., Strijk, T., and Wolff, A. Towards an evaluation of quality for names placement methods. *Int. Journal of Geographical Information Science*, Taylor & Francis, 16 (2002), 641–661.
23. Wigdor, D., Jiang, H., Forlines, C., Borkin, M., and Shen, C. WeSpace: the design development and deployment of a walk-up and share multi-surface visual collaboration system. *Proc. of CHI'09*, (2009), 1237–1246.
24. Wu, H.-Y., Takahashi, S., Lin, C.-C., and Yen, H.-C. A zone-based approach for placing annotation labels on metro maps. *Proc. of the 11th Int. conference on Smart graphics*, Springer-Verlag (2011), 91–102.
25. Yoeli, P. The Logic of Automated Map Lettering. *The Cartographic Journal*, 9 (1972), 99–108.
26. Zilske, M., Neumann, A., and Nagel, K. OpenStreetMap For Traffic Simulation. *Proc. of the 1st European State of the Map Conf.*, Vienna, (2011), 126–134