Visualization to support Identification, Exploitation and Fusion of Data and Information delivered from Heterogeneous Sources in ISR

Kavyashree Jamboti, Florian Van de Camp, Achim Kuwertz,
Daniel Haferkorn, Ralf Eck, Gunther Grasemann
Fraunhofer IOSB, Fraunhofer Straße 1, 76131 Karlsruhe, Germany

ABSTRACT

In ISR (Intelligence, Surveillance and Reconnaissance), heterogeneous sources deliver data and information having spatial and temporal attributes. Real time and non-real time data created for various purposes, present in different formats has to be exploited and fused. The Coalition Shared Data (CSD) concept makes the interoperable sharing of ISR data and information possible. The concept itself and a technical approach for it were developed within the multinational projects CAESAR, MAJIIC and MAJIIC 2 and tested in coalition exercises. The interfaces of software systems providing access to CSD data must allow the user to intuitively use the system and form a substantial part with regard to user acceptance. We describe different systems and approaches we designed and developed to access CSD data that can locate and present the data to the user based on his specific demands. Visualization forms an important part of these systems which share the common challenge of representing data and information with spatial and temporal attributes. The visualization of data and information has to be designed in a manner that supports efficient access, discovery and optionally additional processing (such as filtering and sorting). Given the large amount of data and information that may be available, visualization taking into account their quality and inherent uncertainty is an additional challenge. This publication provides an overview of the systems and approaches we developed to present CSD data and identifies challenges common to these systems. To tackle these challenges, we present new research results regarding visualization of data and information with temporal and spatial attributes.

Keywords: ISR, visualization, spatio-temporal data, CSD, clustering, filtering, heat maps

1. INTRODUCTION

The CSD (Coalition Shared Data) [1] server was developed to enable interoperability in Joint ISR by storing and disseminating data in standard formats defined by STANAGs (NATO Standardization Agreements). Interoperability is an important aspect for any ISR task which involves analysis of data received from various sensor sources and agencies. Products stored in the CSD server and information extracted from products of the CSD server aid in knowledge discovery and information creation which are necessary for decision making processes in a military reconnaissance cycle.

CSD servers can provide access to large amounts of data. It is a challenge for users such as analysts, planners and decision makers to narrow down the data set to find those that are of interest to solve their specific problem. Additionally, just having access to this data does not provide an overview of the quality and attributes of data in order to make useful inferences about trends of the data. At present, for the client applications developed at Fraunhofer IOSB that use data from the CSD servers, complex queries can be made by entering values as text and the resulting data set is displayed as a table with associated icons for the objects in the data set on a map. Visualization of this data has the advantages of providing an overview of the data objects for easy assimilation, helping in decision making processes by enabling discovery of new relationships and patterns in the data. In [2], the following benefits of using visualization to analyze large sets of data are inferred:

- Providing a view of subsets of data as required by filtering out unwanted data.
- Providing an overview of the complete data set to see trends and patterns which are otherwise not visible.
- Providing immediate visual feedback to users interacting with the visual tool.

Hence, we believe that better visualization of CSD data is essential for efficient analysis and decision making tasks. ISR applications and systems such as ISAAC.web PLUS, ISR-OOWM and DigLT being developed by Fraunhofer IOSB and described in the following section present data in one of the two ways: by directly presenting data retrieved from a CSD server or provided by other STANAG conforming sources to the user; or firstly extracting information from the data and then presenting the extracted information to the user. In either case, these systems deal with presenting objects that possess spatial and temporal attributes. All these systems use maps for purposes of query, display of results and interaction with the result set. We would like to enhance the efficiency of tasks that users are required to perform using these systems by providing visualization techniques for efficient querying, interactive filtering and sorting, and using heat maps to display spatio-temporal distributions.

The rest of the paper is organized as follows: In Section 2, some example real-world systems developed at Fraunhofer IOSB which deal with displaying objects on a map are presented and common challenges for visualizing large data sets are identified. Section 3 discusses fundamental concepts which form a basis in the fields of cartography and visualization and describes how they can be used to develop solutions to problems that involve the presentation of objects on a map. Section 4 describes the main contributions of this paper, where additional visualization techniques that can be used for better presentation and interactive modification of data sets displayed on a map are presented. Finally, the conclusion and future work are summarized in Section 5.

2. REAL WORLD SYSTEMS AND COMMON VISUALIZATION CHALLENGES

Below, we present three client systems developed at Fraunhofer IOSB that access CSD servers, obtain data sets with spatio-temporal attributes and display them on a map. For all of these systems, common challenges pertaining to querying and subsequent sorting and filtering of data sets need to be addressed.

2.1 ISAAC.web PLUS

The CSD client ISAAC.web PLUS provides a GUI that allows users to ingest data to the CSD server and to query the CSD server. Queries can be based on all possible attributes of products stored in the CSD, including spatial and temporal attributes. Additional visualization added to ISAAC.web PLUS mainly aims at helping the user to choose the most appropriate product(s) based on his/her needs from the CSD server. Figure 1 shows a snapshot of the application.

The query process: The query UI contains a map where spatial selections can be made in the form of rectangles, circles or polygons. Additionally, a time period can be chosen by selecting two dates from a calendar. The type of product can also be narrowed via a drop down menu. Advanced query options allow further refining query parameters for different products.

Displaying the query results: The products that satisfy the query parameters are shown on the map as thumbnails representing their type and areas representing their extent. The products are also listed in a result list. Clicking on a product in the list shows the metadata of the product in a frame below the result list.

As an aid to the user, results that are very close to each other on the map at the current zoom level are clustered together and displayed as a circle with a number indicating the amount of clustered results. By clicking on this circle, one can zoom in and the clustering in the displayed area is redone. This recursive functionality continues until all currently shown products can be displayed without clustering or until the map display can’t be zoomed further. In the latter case, the products available for that cluster circle are displayed using a star topology upon clicking on the circle.

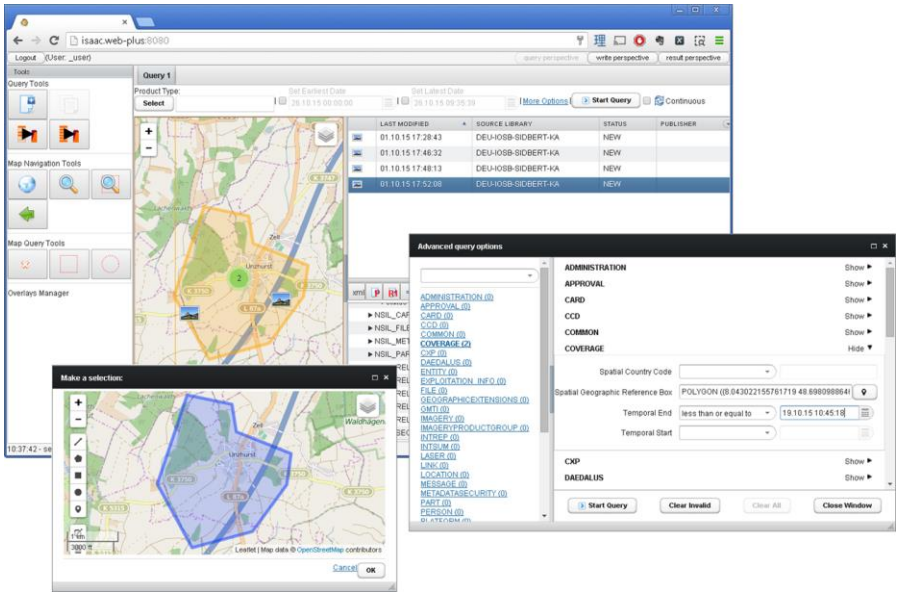

Figure 1. Snapshot of the ISAAC.web PLUS client application.

2.2 ISR-OOWM

The ISR-OOWM (Object-oriented World Model for the Intelligence, Surveillance and Reconnaissance domain) is based on the OOWM system for world modeling [5, 6, 7] which was developed at Fraunhofer IOSB. As a system for world modeling, the OOWM integrates and stores information about real-world objects in a considered domain received from external sources. Received information is consolidated into a representation of the current state of the observed domain and provided to higher-level modules for further processing. The OOWM allows fusing new information about a real-world object with information about this object that is already stored in the OOWM.

In Figure 2, an overview of the ISR-OOWM is shown, including its map-based GUI for information presentation and querying as well as connections to a map server and a database for persistence. Input is received via an information extractor component connected to a CSD server, which breaks down retrieved CSD products into observations and feeds them to the ISR-OOWM. The map server is connected to a GIS (Geographic Information System) via a Web Map Service (WMS).

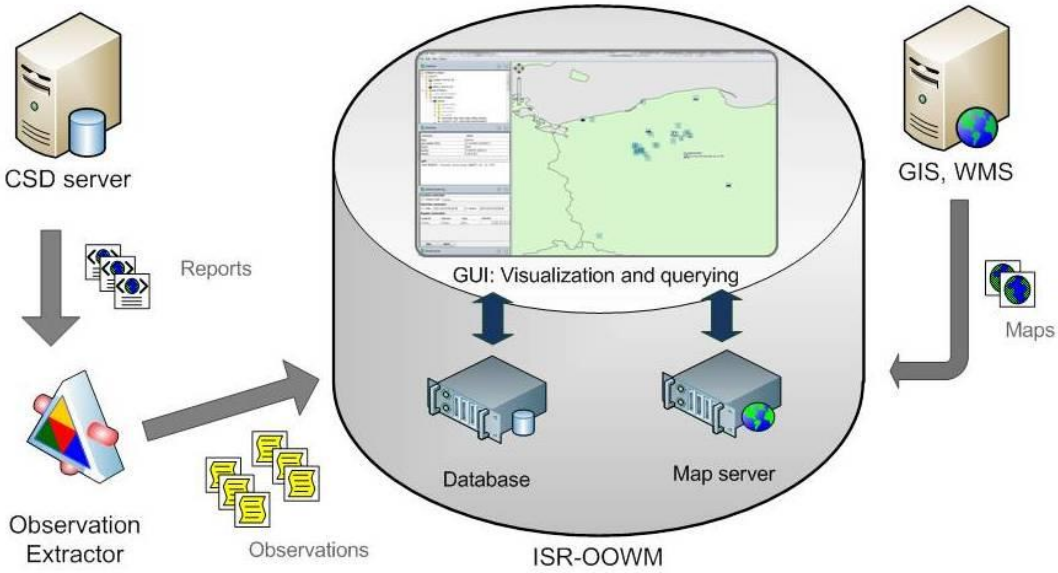

Figure 2. Overview of the ISR-OOWM and its integration into the CSD architecture.

2.3 DigLT

The Digital Map Table (DigLT) is a multi-display workspace for computer supported collaborative work on geodata. Supported is the interdisciplinary solving of tasks by a team, having the need for a variety of views on the same geographical area. These tasks typically arise in disaster management, situation analysis or planning the operation of staff, material or sensors.

The software produces an interactive common operational picture (COP) on mobile devices as smartphones, standard desktop working space, table tops or video walls. The Digital Map Table is an interactive table top showing the geographical overview of terrain or town of concern. This is the basic workspace for the team. Each team member may have an additional tablet-PC or smartphone that can be moved freely over the scene on the table. A patent-registered procedure makes the device always showing the underlying area in higher detail or with different technical details, by this becoming a Fovea-Tablett.

This technology allows specialized analyses of areas of interest without changing the overview. The software presents the users of all devices rich functionality to work with the connected view. The client server architecture allows the participating team members not to work on the table only, but offside too. By the mobile devices the common operational picture can be taken for mobile action.

As an additional information source the Digital Map Table has a background monitor to present context information as Internet pages, live-streamed videos and images, and object information out of data bases such as the CSD server. All team members are permanently kept on the same information level and by this can contribute faster and more precisely their newly gathered information and knowledge to the COP. Forces on location stay up to date. The technology is particularly suitable for disaster management, rescue services, major events, as well as for the management of industrial complexes.

2.4 Common Challenges regarding the Presentation of large Data Sets

For all of the systems described above, we can see that large amounts of data need to be presented to the user and displayed on a map, where we need to address challenges such as overcoming the crowding of elements on a map with large numbers of objects as well as making the query process for data efficient and intuitive, providing techniques to efficiently sort and filter retrieved data sets, and also enable the inference of trends in data to provide an overview and

identify gaps in the data. In Section 4, we describe in detail the common challenges and visualization solutions that can be applied in order to overcome these problems.

3. RELATED WORK AND FUNDAMENTAL CONCEPTS

In order to effectively represent data on a map and to be able to make useful inferences from it, it is important to understand the relationships between data and visualization concepts such as marks, visual variables and their characteristics.

In the subsections below, we present tables summarizing the various relationships between data and visualization concepts and how they can aid us in developing effective visualization techniques.

3.1 Relationship between visual Variables, Scales of Measurement and Marks

In [3], the concepts of marks and visual variables are presented. Marks are entities that can be represented on a map. They can be either points, lines or areas. Visual variables are properties of marks which can be used to indicate values for attributes of an object represented by a mark. Attributes can be given in different scales of measurement. The measurement scale of data [4] in general gives us an indicator about the nature of data. The scales of measurement nominal, ordinal, interval and ratio tell us what operations can be performed on the data and how they can be used.

We have inferred basic relationships between visual variables, their associated scales of measurement and marks that can be used for their representation. These relationships, summarized in Table 1, can serve as a basis during identification of appropriate visualization techniques to represent objects with attributes on a map.

When we want to represent a real-world object, we have to decide whether it is best represented by a point, line or area. Then, we have to think about which attributes of the object we would like to display on the map. Depending upon the measurement scale of the attribute, we can decide which visual variable can be used to represent it.

For example, let us consider CSD products which are represented as points. A CSD product has a type such as image, report or video. The type of the CSD product is an attribute which has values (as image, report or video) and is nominal in its nature. Table 1 tells us which visual variable can be applied to an attribute (depending on its measurement scale) and which mark can be used to represent the real-world object. According to the table, we can use either shape or color hue to represent the nominal attribute ‘type’ of an object being represented by a point mark. However, we can use our discretion to choose shape as the visual variable and provide different icons for different types of CSD products. The visual variable color hue can then be used to represent another attribute of the object that has nominal values.

Table 1. Relationship between visual variables, scales of measurement and marks.

Visual Variable	Measurement Scales	Marks
Size	Ordinal, Ratio and Interval	Point, Line
Shape	Nominal	Point
Color hue	Nominal	Point, Line, Area
Color value	Ordinal, Ratio and Interval	Point, Line, Area
Color saturation	Ordinal, Ratio and Interval	Point, Line, Area
Orientation	Nominal	Area
Texture	Nominal	Line, Area

3.2 Relationship between visual Variables, their Characteristics and Marks

Just using a visual variable to represent an attribute of a real-world object is not enough. Characteristics of visual variables help us to determine whether a visual variable is well suited for a task at hand. Five characteristics of visual variables have been defined in [3] as follows:

- Selective: A visual variable is said to be selective if a mark changed in this variable alone makes it easier to select that changed mark from all other marks.
- Associative: A visual variable is said to be associative if marks that are alike in other ways can be grouped according to a change in this visual variable.
- Quantitative: A visual variable is said to be quantitative if the relationship between two marks differing in this visual variable can be seen as numerical.
- Order: A visual variable is said to be ordered if changes in this visual variable support ordered readings.
- Length: The length of a visual variable is the number of changes that can be used and still retain the task supporting characteristics that are usually associated with this visual variable.

It is recommended that readers refer to [3] for better understanding of the characteristics. Table 2 which we constructed after studying the relationships between visual variables, their characteristics and marks in [3], tells us if we can choose a visual variable for a given task and a given mark. For example, if a task involves ordering real-world objects represented by points, then 'position', 'size' or 'color value' can be used as visual variables. However, here we can note that the visual variable 'position' differs from other visual variables. 'Size' and 'color value' can be used to represent attributes of a real-world object, but 'position' itself is a property of the object when it is displayed on a map. Also, it can be noted that the characteristic 'length' is not a part of the table because it is a characteristic that cannot be used to represent an attribute of an object on a map, it simply tells us how feasible it is to use a visual variable for effective representation of an attribute.

Table 2. Characteristics of visual variables and associated marks.

Visual Variable (Marks) ► / **Characteristic ▼**	**Position (P, L, A)**	**Size (P, L)**	**Shape (P)**	**Color Value (P, L, A)**	**Color Hue (P, L, A)**	**Orientation (A)**	**Texture (L,A)**
Selective	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Associative	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Quantitative	Yes	Yes	No	No	No	No	No
Order	Yes	Yes	No	Yes	No	No	No

Legend:

Yes – the visual variable for the mentioned marks can be used for a task associated with the respective characteristic.

No – the visual variable for the mentioned marks cannot be used for a task associated with the respective characteristic.

P, L, A stand for marks associated with the visual variables: Points, Lines and Areas, respectively.

4. GENERIC VISUALISATION TECHNIQUES

In this section we present generic visualization techniques that can be applied to solve the problems common to the real-world systems presented in the previous section.

4.1 Clustering

While different visualizations can solve many different problems, we have focused on one specific problem that often occurs within map applications. The problem is that of crowding of map elements. Because digital maps can typically be viewed at different zoom levels, the screen estate available for a number of items can drastically change when the user

zooms out. An example for this can be seen in Figure 3. This dense display of map elements is not useful as it does not allow the user to get an overview nor does it allow for interaction, e.g. selection, of specific elements.

Figure 3. Crowded map elements as a result of zooming out. (Image source: Google Maps).

Even if there is room around a number of elements, in the context of a map application the location of an element is actually meaningful and cannot be changed for display purposes as it would change the information that is represented by that map element. A common way to deal with items that require more room than what is available is a clustering of multiple elements into one. This gives the application a clean look but does not solve the initial problem of a lack of overview and interaction with specific elements on the map.

We looked into the tools, libraries and visualization techniques in the state of art and extended the approaches to find solutions that best fit our problem at hand. We have identified three approaches to solving this problem. The key idea of the first approach is to find a visual representation that uses additional space around a cluster of elements while still maintaining the intrinsic geographic information each element provides. Examples can be seen in Figure 4 and Figure 5.

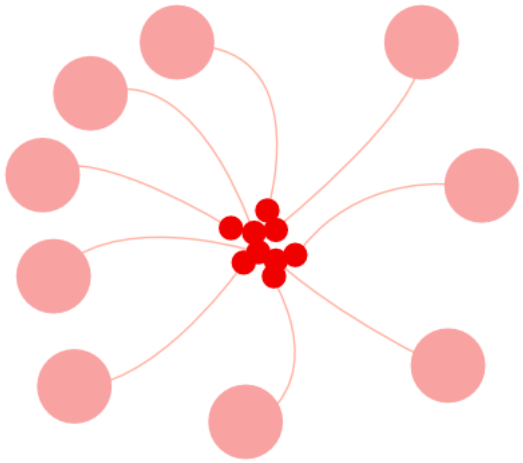

Figure 4. Larger elements with room for displaying additional information using the original item as an anchor to maintain the geographic reference.

Figure 4 shows a proxy element for each actual element that is larger and at a less crowded location. Lines show the connection between proxy element and corresponding original element. Another example is shown in Figure 5. Here, the metaphor of a magnifying glass is used to make sure that the display of map elements at a different position than their geographic position does not confuse the user. This approach requires a rather large empty area close to the cluster of map elements, alternatively it can be displayed in an opaque way and hide other items which might be acceptable for a moment depending on the application.

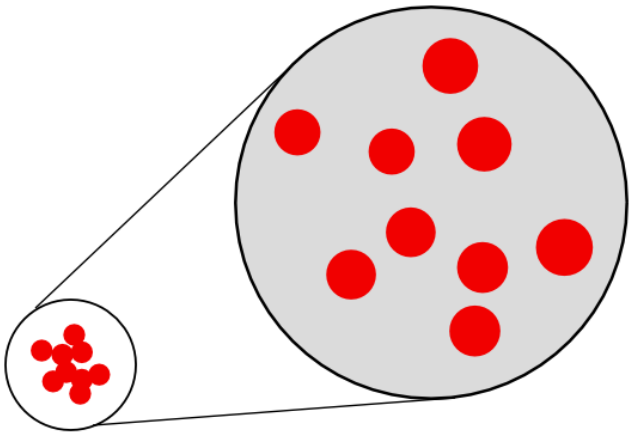

Figure 5. Stylized magnifying glass allows the display of elements at a different position without causing confusion about the geographic information of the items.

The second approach to dealing with too many map items is the use of a meta representation for multiple items. A simple example can be seen in Figure 6. Here, a large number of map items is represented with an icon and a number indicating the number of elements represented by the icon. The actual map items are not shown. This immediately resolves the space issues on the map but at the same time the abstraction does not convey as much information.

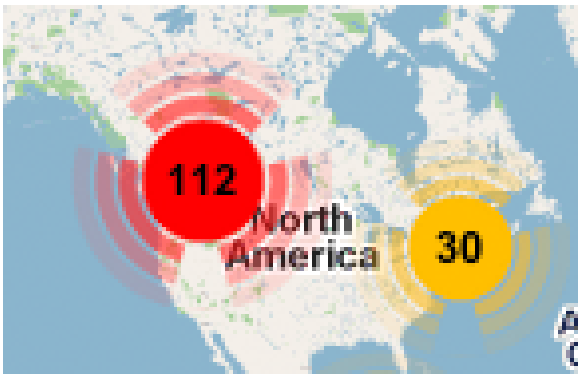

Figure 6. Static meta representation of multiple map elements (Image Source: Google Maps).

A variant of the meta representation concept is shown in Figure 7. The circle on the left shows a distribution of distinct values of a common property of the map elements. While this does not reveal the same amount of information as the actual display of elements would, it can provide the user with enough information to narrow down the search if the displayed parameter is selected well in the specific application context. The same figure shows another extension of the concept. The ring on the right is shown when the ring on the left is hovered over (in case of a mouse based interface). While the additional information (the distribution of additional parameters) needs additional room, it is only shown temporary and only when the user interacts with the ring on the left, making it a reasonable assumption that this area is currently of interest.

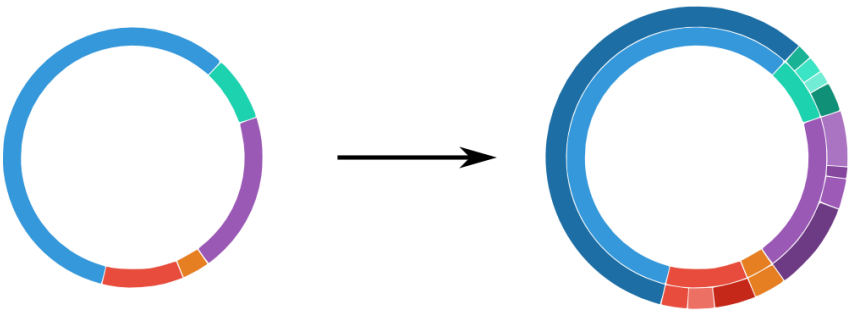

Figure 7. Interactive meta display that gives an insight about the nature of the map elements that are represented.

The third approach to dealing with too many elements on a map aims at reducing the number of elements on the map. The previous approaches only work under certain circumstances (if there is additional room nearby) or hide much of the information by displaying a meta-representation instead of the actual elements. By reducing the number of elements on the map both disadvantages can be overcome. To reduce the number of map elements in a meaningful way, however, the user needs tools to intuitively select which subgroup of elements is important, as this selection is not only application specific but also task and situation specific. There are generally two cases where a user can make this kind of decision. It can be done either during the initial database query or on the displayed results after the initial query. Since this approach has the already mentioned advantages over the previously described approaches, these two options will be described in detail using specific designs within our application context in the next subsection.

4.2 Querying techniques

In this subsection, we present a new, intuitive graphical interface that allows the definition of parameters for querying by applying various state of the art data selection techniques such as sliders, check boxes and geographic area selections. To show how the number of map elements can be reduced during the initial query, screenshots from the DigLT will be used in the following. This has some influence on the size of interface elements as the interaction is touch based, but the relevant functionality could be used with the other systems as well. Figure 8 shows the initial screen, where a section of a map is shown and at its bottom a menu with two different buttons for initiating a query. As the database will most likely

not only contain elements located within the current map view, the button on the left will trigger a query considering all elements within the current map view (Figure 9). The button next to it allows to already refine the query even more and therefore reduce the number of results by allowing to draw a polygon to limit the area in which map elements will be considered during the query (Figure 10).

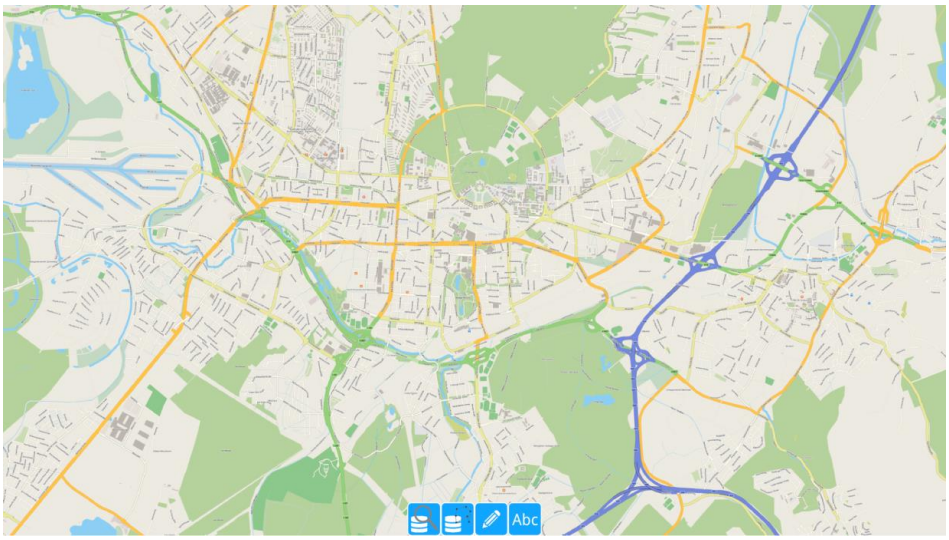

Figure 8. Interactive meta display that gives an insight about the nature of the map elements that are represented.

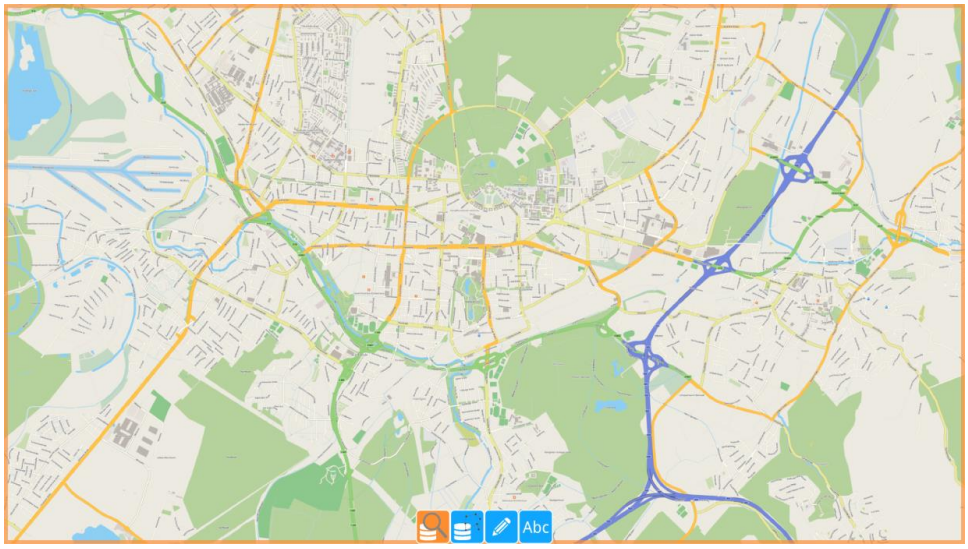

Figure 9. Visualization of the query area (whole screen).

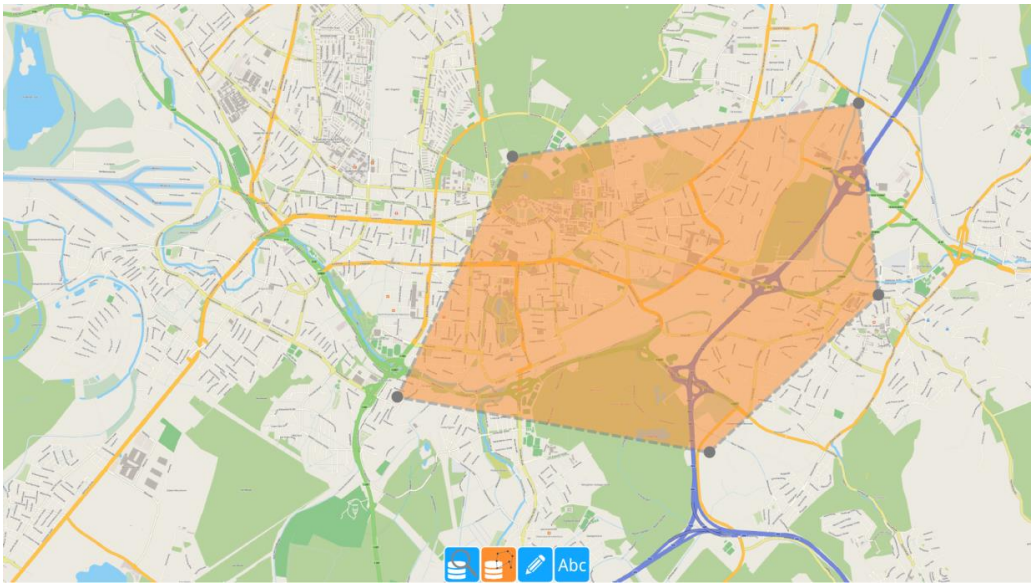

Figure 10. Visualization of the query area (highlighted polygon).

After the selection of the area to consider for the query, the rest of the interaction is the same for both approaches and starts with the display query windows, shown as an overlay on top of the map (Figure 11). Here sections for different properties associated with the map elements are shown. At the very bottom, the total number of elements that will be returned if the query is executed is shown.

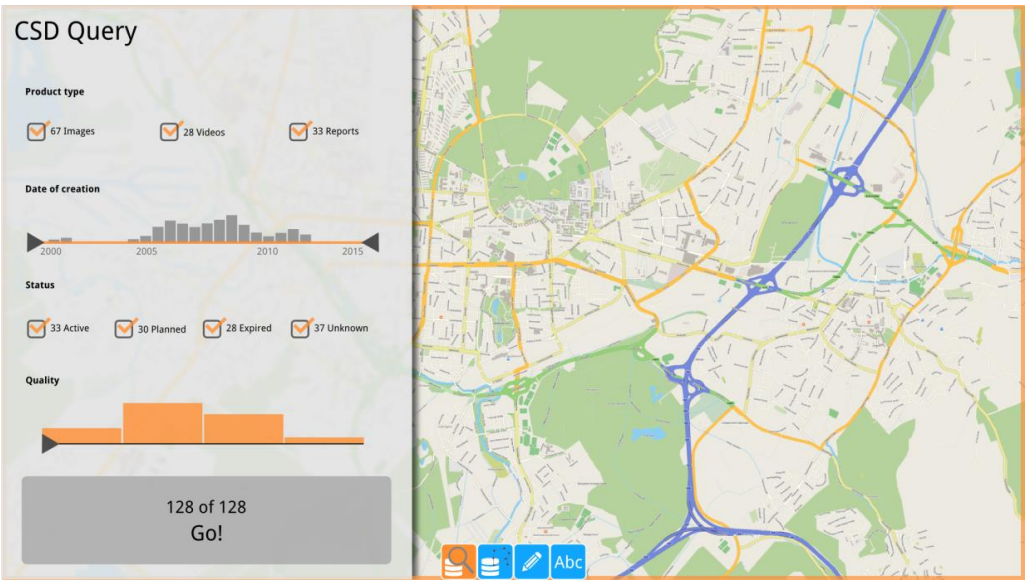

Figure 11. Initial query overlay on the left showing the total number of elements in the query area.

The first section represents the property “Product type” (Figure 12). The database in this example contains images, videos and textual reports. These three possible values for the property are shown as checkboxes. This way, the user can select the types of elements he or she is interested in. For each possible value the number of elements of that type is displayed. This helps the user to balance the decision between getting the data needed and reducing the number of results.

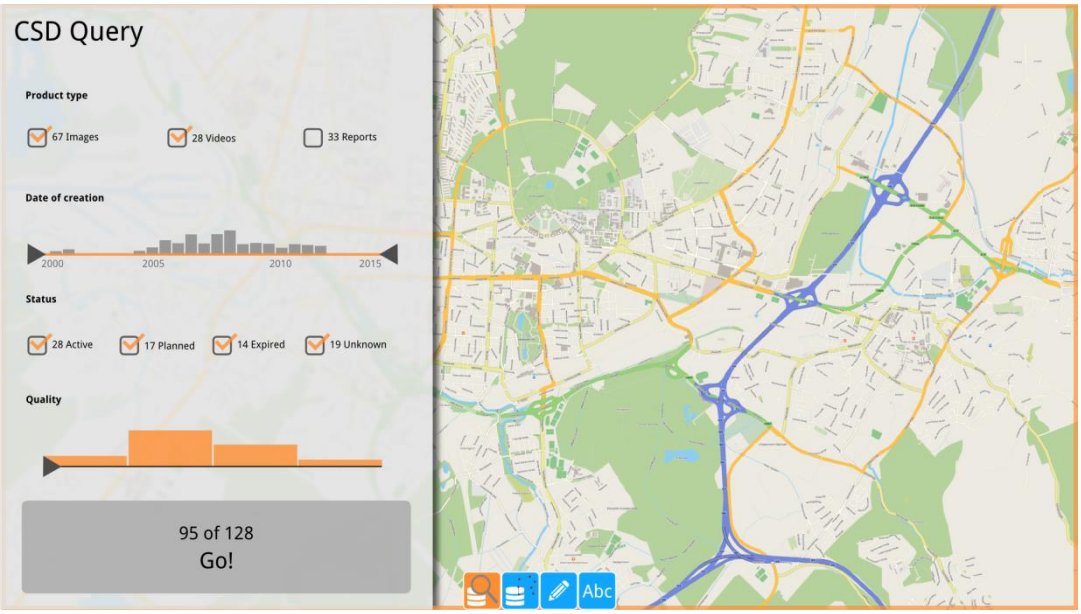

Figure 12. A selection within the first element property "Product type" reduces the number of results.

The next section is the “Date of creation” of map elements. Obviously checkboxes cannot be used to represent the large number of possible dates. Instead, a slider is used to select a range of relevant dates. Above the slider a histogram is displayed (Figure 13) to give the user and indication about the number and distribution of map elements within a range of dates. As before, a selection in this section reduces the overall number of elements that would be returned if the query was executed. The current number of map elements returned is displayed on the “Go!” button.

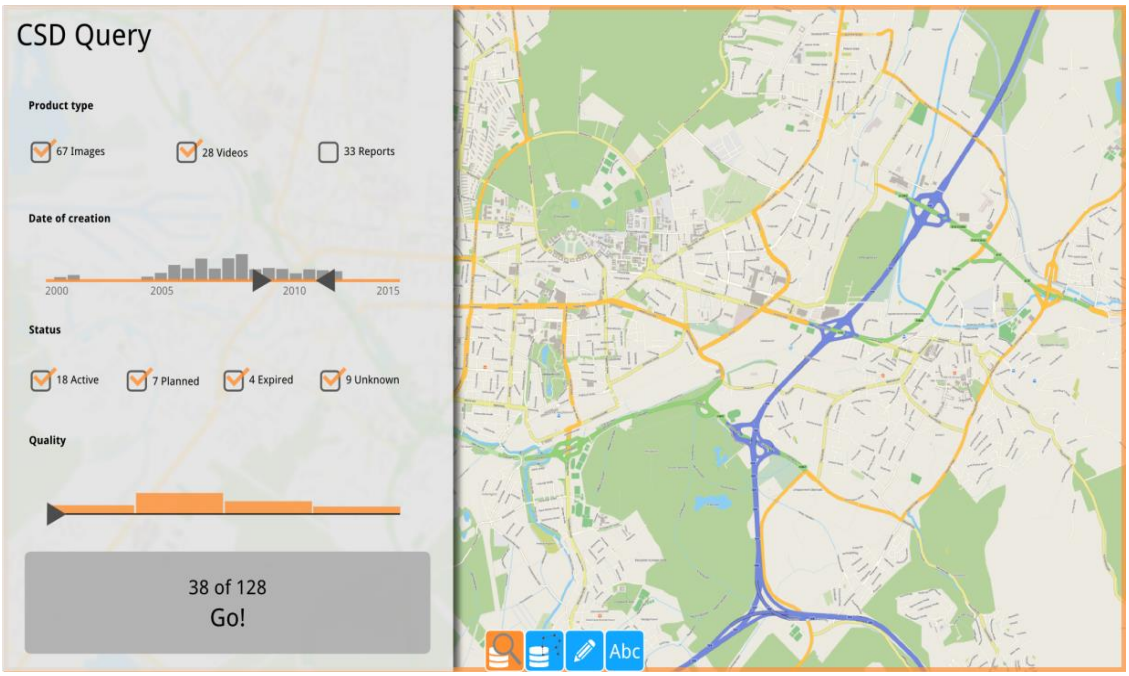

Figure 13. The slider for the property "Date of creation" gives direct feedback about the number of elements in a certain range.

Additional properties, “Status” and “Quality” are displayed as checkboxes and a slider respectively (Figure 14 and Figure 15).

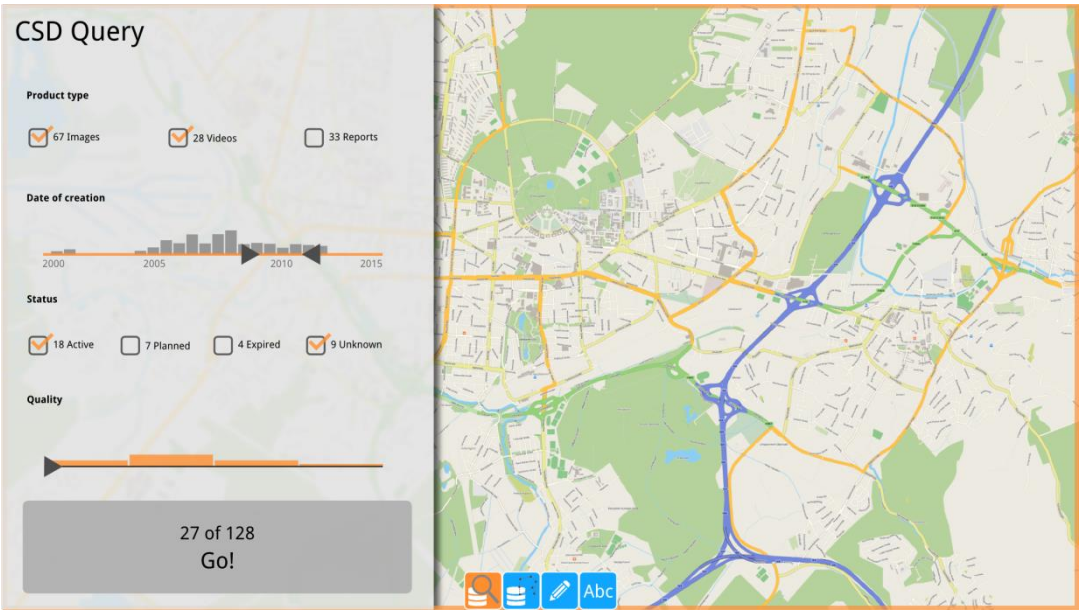

Figure 14. A selection within the element property "Status" reduces the number of results.

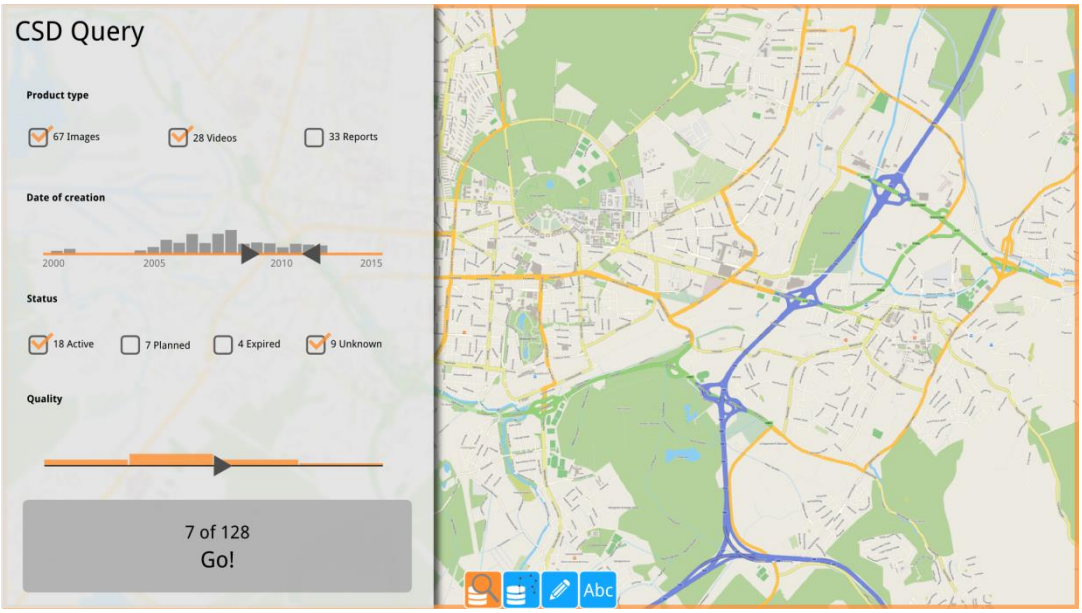

Figure 15. The slider gives intuitive feedback about how the selection affects the number of results.

After values for all properties have been selected and the “Go!” button has been pressed, the query is executed and returns a much smaller number of results than the initial query would have (Figure 16).

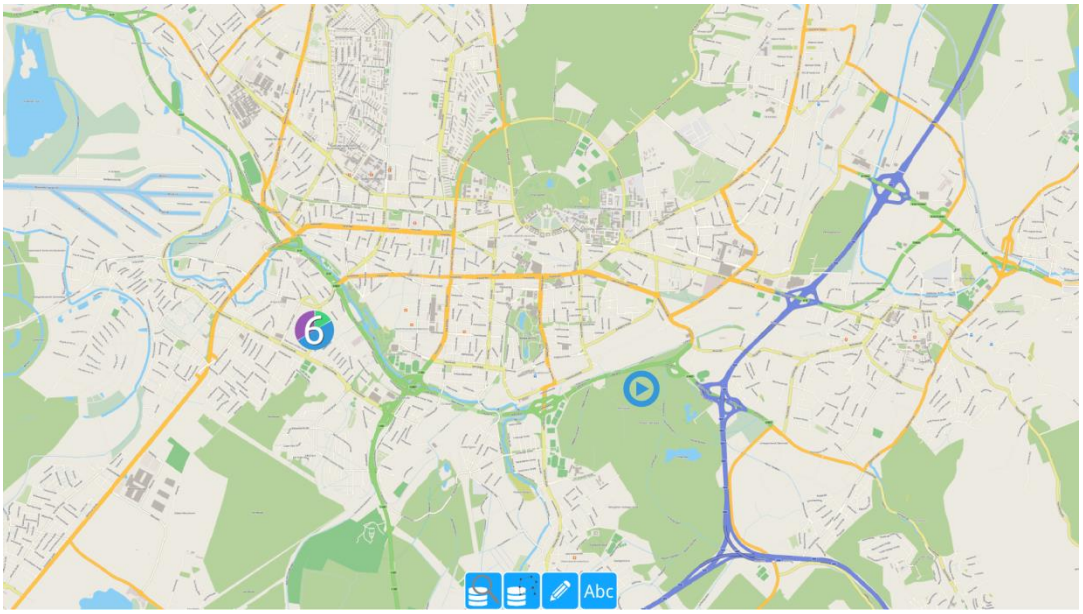

Figure 16. Final display of map elements.

While the number of results can be greatly reduced using this interactive feedback query, it is still possible that too many items are located at the same location and further refinement of a cluster of map elements is required to be able to display the individual map elements. This kind of filtering will be described in the next section.

4.3 Filtering and sorting techniques

In the initial query described above, it is acceptable to use large portions of the screen to display interface elements as well as information. As soon as this initial query is executed, all information is displayed on the map and the user will see everything in its geographic context. This prohibits the use of large menus as above since this might cover potentially relevant information. A typical interaction after the initial query is a further filtering of the returned results. While the interactive query already limits the overall number of results, it is possible that there will still be too many elements at roughly the same geographic location to be displayed at the same time. Filtering is an option to narrow down the results of a search even further in a geographic area with too many items to display. The filtering does, however, need a user interface to allow the user to influence the filtering process which poses some challenges. Ideally, the filtering user interface should not cover any other information or map areas that might be relevant and it should be clear which cluster of elements the filter is being applied to. An intuitive choice would be a circular menu as shown in sketches in Figure 17 - Figure 20

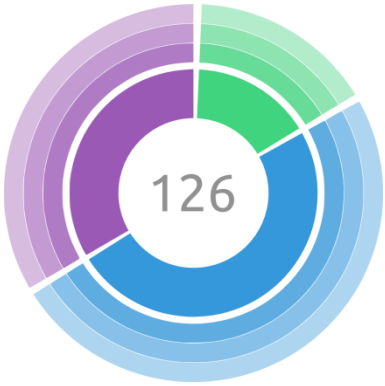

Figure 17. Circular menu showing three object categories.

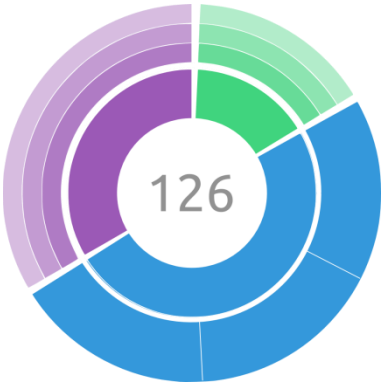

Figure 18. Subdivision of a category in the circular menu.

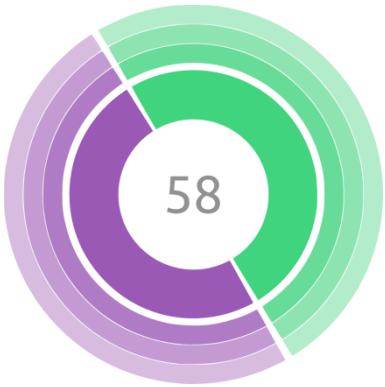

Figure 19. Exemplary resulting circular menu after a first selection.

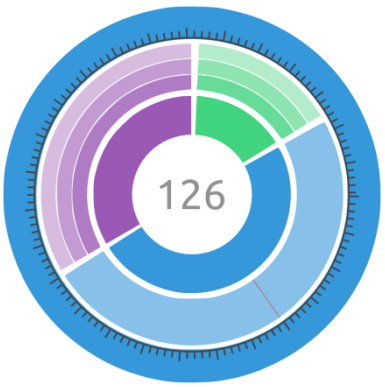

Figure 20. Concept for a slider element in a circular menu.

Unfortunately there are three problems with a circular menu in this case. Firstly, it always covers the whole area around the item cluster, even if not all space is needed. Secondly it requires new concepts for standard input components such as a slider (shown in Figure 20) that the users are most likely not familiar with. And thirdly a circular layout is not well suited to display text as it is hard to read and offers little room for it.

The proposed solution to this is the use of a collapsing, and therefore space saving, rectangular layout with an anchor point to clearly show the correlation between menu and item cluster. Such an anchor can initially simply display the number of items in a cluster as in Figure 21.

Figure 21. Anchor to filter a cluster of items.

The anchor stays at the position of the cluster that is being filtered and reflects all selections made in the filtering menu. Figure 22 shows an example of information reflected by the anchor after a few selections have been made including the current number of elements left if the filter is applied. The filter can be applied at any time by clicking/touching the anchor.

Figure 22. Anchor reflecting filtering choices made.

The actual filtering menu is displayed close to the anchor but can be moved to avoid overlap with required information. In Figure 23 the menu displays a list of the most common object types along with the number of items in the cluster for the respective type. A plus sign indicates that there are more (less common) options that are initially hidden to save space.

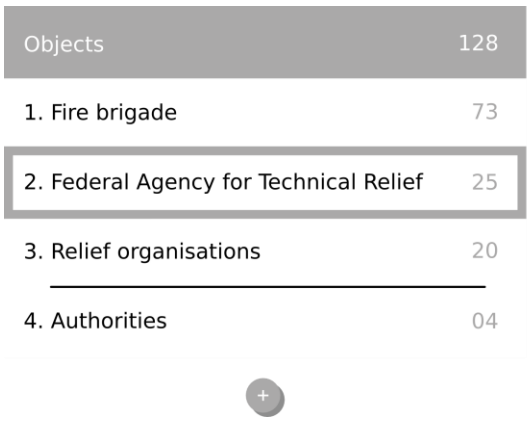

Figure 23. Initial list of most common object types.

A click on the plus sign at the bottom reveals the full list of possible object types as shown in Figure 24.

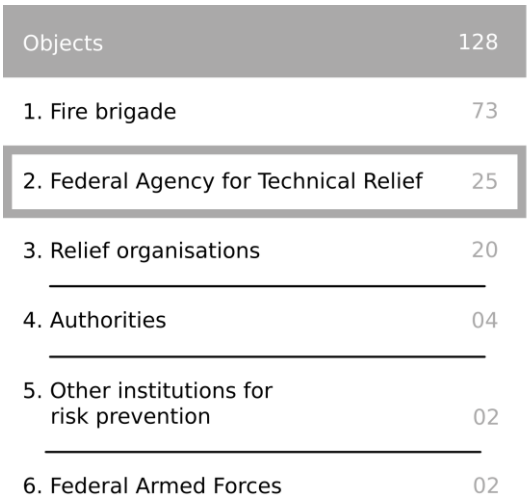

Figure 24. Full list of object types.

Additional filters are placed on top of previously defined filters as shown in Figure 25. By overlapping the filter sections only partially, some screen estate can be saved and the history of defined filters is still visible after several steps (compare Figure 26).

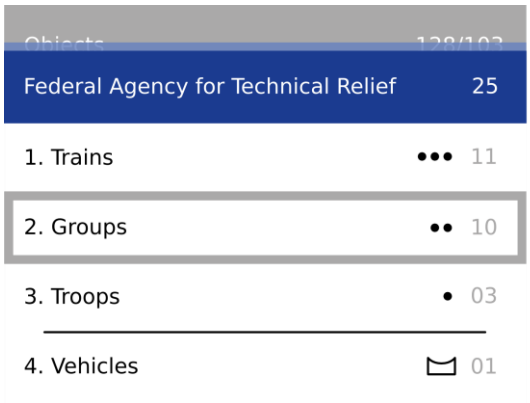

Figure 25. Filters are placed on top of each other.

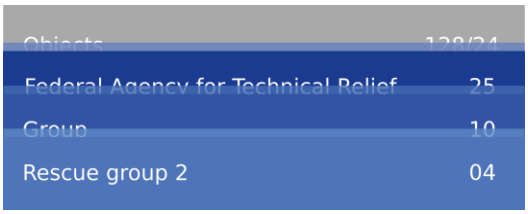

Figure 26. History of filtering steps.

If the user wants to modify previously defined filters (for example, because too many items are filtered out in a certain step) a click/touch on the overlapping filter steps reveals an overview of all filtering steps in which each step can be selected for modification. This overview is shown in Figure 27.

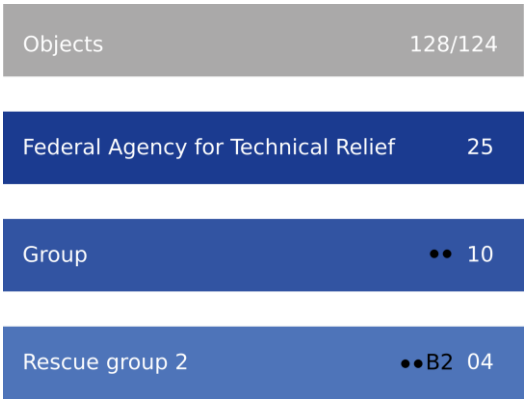

Figure 27. Overview of filtering steps to select a previous step for modification.

In addition to filtering steps that require a selection from a list, arbitrary input elements such as sliders (shown in Figure 28), dials and date pickers can be used in the rectangular menu layout without modification and at the same time keep an appearance well known to most users.

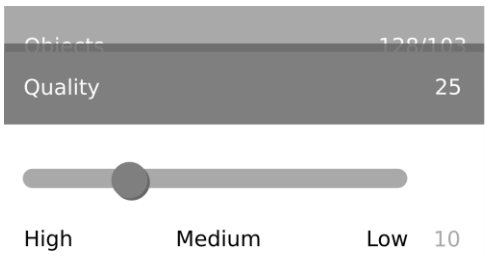

Figure 28. A slider as part of the filtering menu.

4.4 Heat maps

Heat maps are another alternative visualization technique that overcome the problem of cluttering that can occur on a map. Heat maps provide a sophisticated approach to aggregate and display spatial data in intuitive and more easy to understand ways. A heatmap-based map view of a data set for example allows to easily detect spatial hotspots or areas not covered by any product. In combination with filtering mechanisms for the type or creation time of a CSD product, this approach can support a user in quickly refining a data set returned as a result of a CSD query to a manageable size of products, which can be analyzed to solve the current task.

We developed a concept demonstrator for heatmap-based visualization as an extension to the ISAAC.web PLUS client application. The demonstrator allows switching between a marker-based and a heatmap-based visualization. The heat map is updated on the fly, e.g., when the underlying map is moved or when the result data set is changed due to filtering operations. In order to allow for sophisticated time-based filtering, the GUI provides a time slider for selecting the desired time interval of products in the data set, visualized using a histogram of products based on adjustable sizes of histogram bins (e.g., days, weeks, etc.). Due to on-the-fly updates of the heat map view, it is also possible to show an animated evolution of products over time in the heat map, moving a selected time interval forward step-by-step in a sliding window fashion. The aggregation of products in a selected time interval can furthermore be influenced using a weighting function. A weighting defines the influence each product has on the visualization in the heat map based on its position in the selected time interval. Based on such a weighting, it is for example possible to emphasize newer products in a time interval over older ones. In principle, arbitrary weighting functions can be employed. The figure below shows a snapshot of the demonstrator. The filtering and visualization tools implemented in the demonstrator additionally allow supporting a user in analyzing a given dataset.

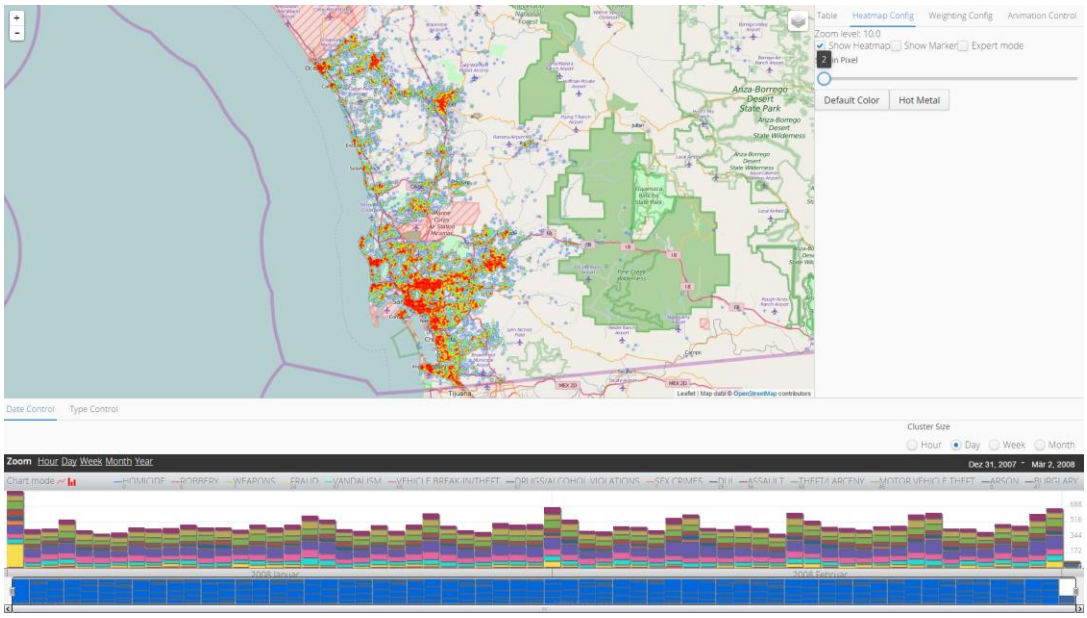

Figure 29. Snapshot of the heat map demonstrator.

5. CONCLUSION AND FUTURE WORK

ISR applications and systems such as ISAAC.web PLUS, ISR-OOWM and DigLT exploit information at various levels and present them to the user. The visualization techniques presented in this paper can be applied to any system that aims to display objects with spatial and temporal properties on a map. These techniques aid the user to narrow down search results to find the product(s) that best answer the query and also give an overview to aid the identification of patterns and trends in the data. This information can equip the user to make better decisions by recognizing gaps in the data with respect to various data attributes and take appropriate corrective measures. For example, the user may be able to identify that a certain location on a map does not have adequate documentation in the form of videos or images, or that not enough images with a good resolution are available for a certain location and arrange for more images and/or videos to be taken for that location.

In the future, we intend to identify more visualization techniques and adapt them to solve problems related to effective visualization of data with temporal and spatial properties. For example, heat maps can be used to represent gaps/holes with respect to a certain CSD product or a certain attribute of a CSD product. We would also like to explore the possibility of using more than one visual variable to effectively display more information on the map. As the architecture of the CSD concept is expanded to accommodate additional processing in the form of data fusion and extraction, we will need to identify new ways of visualizing the generated information and integrate them with existing clients or create a new client to solve a specific problem.

REFERENCES

[1] Essendorfer, B., Kerth, C. and Zaschke, C., “Evolution of the Coalition Shared Data concept in Joint ISR,” IST-SET-126 (2015).

[2] Gonzalez, V., and Kobsa, A., “Benefits of information visualization systems for administrative data analysts,” Proceedings. Seventh International Conference on Information Visualization, 331-336 (2003).

[3] Carpendale, M.S.T., “Considering Visual Variables as a basis for Information Visualization,” Department of

Computer Science, University of Calgary, Calgary, Canada (2003).

[4] Stevens, S. S., “On the Theory of Scales of Measurement,” Science, Bd. 103, 677-680 (1946).

[5] Y. Fischer and A. Bauer, ”Object-Oriented Sensor Data Fusion for Wide Maritime Surveillance,” International Waterside Security Conference (2010).

[6] Bauer, A., Emter, T., Vagts, H. and Beyerer, J., “Object-oriented world model for surveillance systems,” 4th Security Research Conference "Future Security" (2009).

[7] Belkin, A., Kuwertz, A., Fischer Y. and J. Beyerer, “World Modeling for Autonomous Systems,” Innovative Information Systems Modelling Techniques, 137-158 (2012).