OpenMathMap: Accessing Math via Interactive Maps

Jan Wilken Dörrie, Michael Kohlhase, Lars Linsen

Computer Science, Jacobs University Bremen
<first initial>.<last name>@jacobs-university.de

Abstract. World Math literature is growing at an alarming rate (3.3M journal articles today increasing by 120k a year). While much of that can be retrieved online, we lack technologies to navigate and understand the space of math literature. The OpenMathMap project develops and deploys novel interfaces that empower interested parties to find their way. We conjecture that such maps can act as cognitively adequate access mechanisms to many large-coverage MKM systems.
The first concrete interface is an interactive map generated from publication data. We have developed a prototype map generation service based on MSC classifications and deployed the maps resulting from ZBMath Data in OpenStreetMap. Interaction mechanisms allow for providing details on demand.

1 Introduction

In the information age fueled by the Internet, the problem of information and knowledge foraging changed from one of retrieving documents to finding out about them. In particular, navigating the space of available documents efficiently becomes an important subtask.

Even in science, the times where single individuals could have an overview over all of science are long past. Even in the Renaissance polymaths like Leonardo de Vinci were considered a rare exception. The scientific community has developed various tools to work around this problem: encyclopedias, survey articles, classification systems, and review services. But with the proliferation of scientific publication – 50 million articles in 2010 [Jin10] with a doubling time of 8-15 years [LI10] these tools start collapsing under the sheer mass of information. Internet-age tools like search engines, bibsonomies, and citation databases solve (part of) the information retrieval and navigation problems by providing word-based search and browsing along citations. Note that these tools are "myopic" in the sense that they only give very local view of the immediate surroundings of a word or document.

Classification systems like the Math Subject Classification (MSC, see [Msc]), the Physics and Astronomy Classification Scheme (PACS, see [PACS10]), or the ACM Computing Classification System (CCS [ACM-CCS98]) take a more global stance, but they lack user interfaces that give information foragers an intuitive sense of direction and locality that is so helpful to humans in navigation tasks.

In the MathSearch Project [MathSearch] we are currently rethinking access to mathematical knowledge and resources. As a first experiment, we are building a global, map-based navigation service for mathematics. The main idea is that humans are very skilled in spatial navigation and in particular have learned to use map representation to navigate spaces and locate targets. Concretely, we want to create a map of mathematics akin the one in Figure 1 used to visualize usage patterns of online communities. We want to base the map on ideas from Rusin's Math Atlas [MathAtlas] (created 1998, last updated 2001, see also Figure 2), which uses topics from the Math Subject Classification for map regions and calculates the positioning and relative sizes from topic interconnections and the numbers of publications. Based on the visual encoding through the geographical map metaphor, we built an interactive tool for data exploration following the concept 'overview first, detail on demand'. We give examples on how our system can be used to explore the given data.

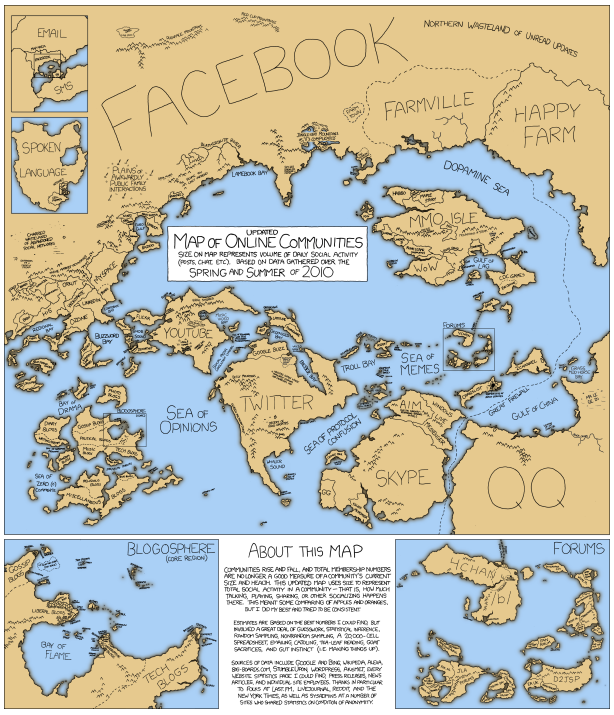


Fig. 1: Map of Online Communities, XKCD 2010 `http://xkcd.com/802/`

Section 2 discusses the publication data used as a foundation and Section 3 the process by which map images can be computed from it. In Section 4 we discuss how such images can be deployed in a map server (here OpenStreetMap). Section 5 discusses topical maps which can be realized with OpenStreetMap. Related work is discussed in Section 6. We show in Section 7 how to integrate information services. Section 8 concludes the paper and discusses future work. Additional details in technical matters can be found in [Dör13].

Acknowledgements Work on the concepts presented here has been partially supported by the Leibniz association under grant SAW-2012-FIZ_KA-2. The authors are indebted to Wolfram Sperber for the Zentralblatt Math publication data and Patrick Ion for initial discussions.

2 ZBMath/MSC Data

In the creation of the map we made use of the 2010 Mathematics Subject Classification [Msc] jointly developed by the American Mathematical Society and Zentralblatt Math. The result are 63 top-level classes, 528 second-level classes and 5607 third-level classes summing up to 6198 classes in total. Each class has an identifier of the form `DDLDD` (`D` for Digit, `L` for Letter) and a title. For example "68R05 Combinatorics". 68R05 is the third-level, "68R Discrete mathematics in relation to computer science" is the second-level and "68 Computer Science" is the top-level.

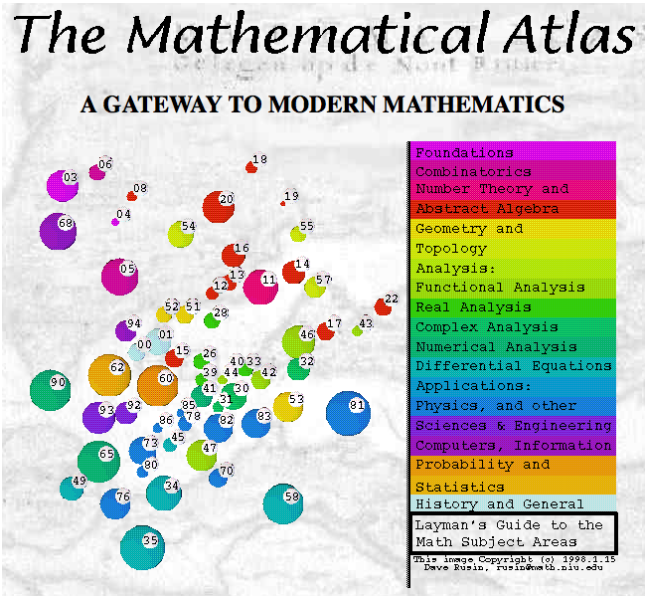


Fig. 2: Rusin's Math Atlas [MathAtlas]

Zentralblatt MATH provided us with the metadata for more than 3 million articles in mathematics, in particular article number, the authors, the classification codes[1], keywords, and publication year. As we will see below, the classification codes allow us to create the map geometry by visualizing proximity and size of MSCs, the publication year gives us the possibility to create maps which only include publications from a given time span.

3 Map Geometry

The first step in map creation is to compute a geometrical representation of the relations in the ZBMath publication data, which will use as the base of the map. In this representation we want to adequately represent the relative sizes and proximities of the MSC classes. The size of a class MSC_c is given by the number of papers[2] (we denote this by $|\mathrm{MSC}_c|$) and define the similarity of two classes as

[1] On average, each classified article carries 3 codes.

[2] Any scalar value that represents a property of each MSC class can be mapped to the size of the respective area in the map. Currently, we just use the number of papers (due to data availability). In future work, we would like to add further functionality that allows the user to interactively choose which measure should be mapped to size.

$$s(i,j) = \frac{|\mathrm{MSC}_i \cap \mathrm{MSC}_j|}{|\mathrm{MSC}_i \cup \mathrm{MSC}_j|}$$

The size of the intersection/union of two MSCs is defined as the number of papers referencing both/any MSCs. This results in $s(i,j) = s(j,i) \in [0,1]$ and $s(i,i) = 1$ for all MSC classes i and j.

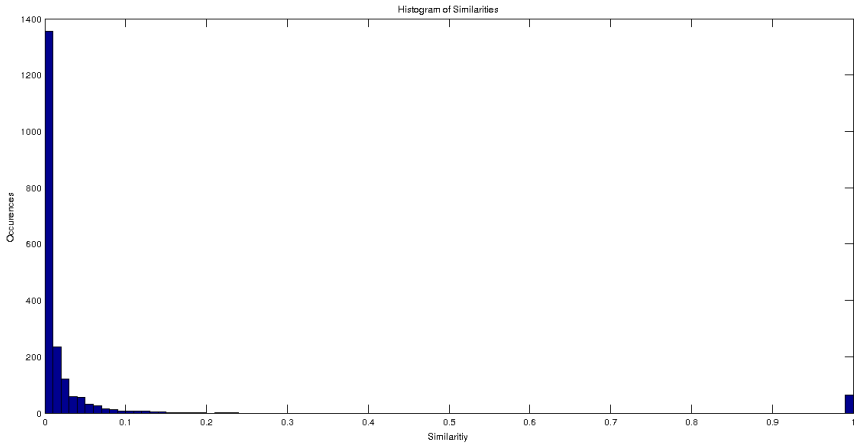


Fig. 3: Histogram of the similarity values

A histogram of all similarity values can be found in Figure 3. Since the similarity matrix is symmetric only the upper triangular part was considered.

For the initial version of the map geometry (see Figure 4), we are calculating the similarity between every pair of top-level MSCs to obtain a similarity matrix of size 63×63. In order to transform the matrix into two-dimensional coordinates for each MSC we applied Multidimensional Scaling (MDS; [KW78]) to it.

3.1 Multidimensional Scaling

Multidimensional Scaling (MDS) is a technique applied to a set of points in n-dimensional space to visualize it in a low dimensional space preserving distances between points as good as possible. Common values for the output dimension are 2 and 3. Formally this can be described as the following. Given a distance function δ, target dimension p and an $n \times n$ dissimilarity matrix

$$\Delta := \begin{pmatrix} \delta_{1,1} & \delta_{1,1} & \cdots & \delta_{1,n} \\ \delta_{2,1} & \delta_{1,1} & \cdots & \delta_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n,1} & \delta_{n,2} & \cdots & \delta_{n,n} \end{pmatrix} \tag{1}$$

one needs to find n vectors $x_1, \ldots, x_n \in \mathbb{R}^p$ such that $\|x_i - x_j\| \approx \delta_{i,j}$ $\forall i, j = 1 \ldots n$.

A solution may be found by numerical optimizations, for example Matlab has a built-in method `mdscale` which takes as arguments the $n \times n$ dissimilarity matrix D and the target dimension p. It returns Y, a configuration of n points in p dimensions, and optionally a stress value, which by default is the stress value given by Kruskal's normalized Stress-1 criterion [Mat]. This criterion is defined in the following way [MDS05]:

$$\sigma_1 = \sqrt{\frac{\sum \left[f(p_{ij}) - d_{ij}(\mathbf{X})\right]^2}{\sum d_{ij}^2(\mathbf{X})}} \tag{2}$$

where p_{ij} is the *proximity* between two given objects, $d_{ij}(\mathbf{X})$ is the Euclidean distance between two points in the current configuration $\mathbf{X}$ and $f\colon p_{ij} \to d_{ij}(\mathbf{X})$ is a *representation function* where the particular choice of f specifies the *MDS model* [MDS05]. For example choosing an appropriate f allows Matlab to work with both similarity and dissimilarity matrices (the exact choice for f is hidden in the implementation of the algorithm).

For the Stress-1 coefficient σ_1 Kruskal suggests the following benchmarks: 0.20 = poor, 0.10 = fair, 0.05 = good, 0.025 = excellent, and 0.00 = perfect [MDS05].

3.2 Radial Basis Functions

The configuration generated by MDS is a collection of points in 2D space respecting the relative distances between given points. However for the visualization one also needs to take the size of each MSC into account.

In order to give a dimension to the points radial basis functions are used. A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin or alternatively on the distance from some other point c, called a *center* [Wik12].

A common choice for RBF is a Gaussian model, but it has the drawback of an infinite support. Therefore, we decided to use a modified cosine function as described in Equation (3).

$$f(d) = \begin{cases} 1 + \cos(d/k) & \text{if } |d| \leq k\pi \\ 0 & \text{else} \end{cases} \tag{3}$$

d denotes the distance of a given point to the origin of a given MSC and k depends on the size of this MSC.

It is zero outside the given range d, which allows us to discard all points with $|d| > k\pi$. In our experiments, we have seen that this finite support of our RBF choice speeds up the calculations by a large factor when compared to a Gaussian RBF.

3.3 Borders and Coast Lines

In order to be able to detect borders in a mechanical way the map was rasterized into a grid of cells. For each cell the influence of the nearby MSCs was calculated using the RBF specified in Section 3.2. Afterwards each cell was associated with the MSC having the highest influence on the cell. If there was no influence at all (i.e. the result of the RBF was zero for every MSC) the cell is considered to be water.

In order to detect borders each non water cell is considered and its 4-neighborhood is checked for changes in the associated MSC. If there are changes, the cell is marked and added to the border set of both MSCs. This also includes coastline detection, since water is considered as the empty MSC.

Finally borders are connected by considering each border set on its own and finding the shortest path going through all contained items. This task was executed using the nearest neighbor heuristic. It was able to produce nearly optimal results due to well defined distances between two border nodes. In a given border the next cell can always be found in the 8-neighborhood of the current cell.

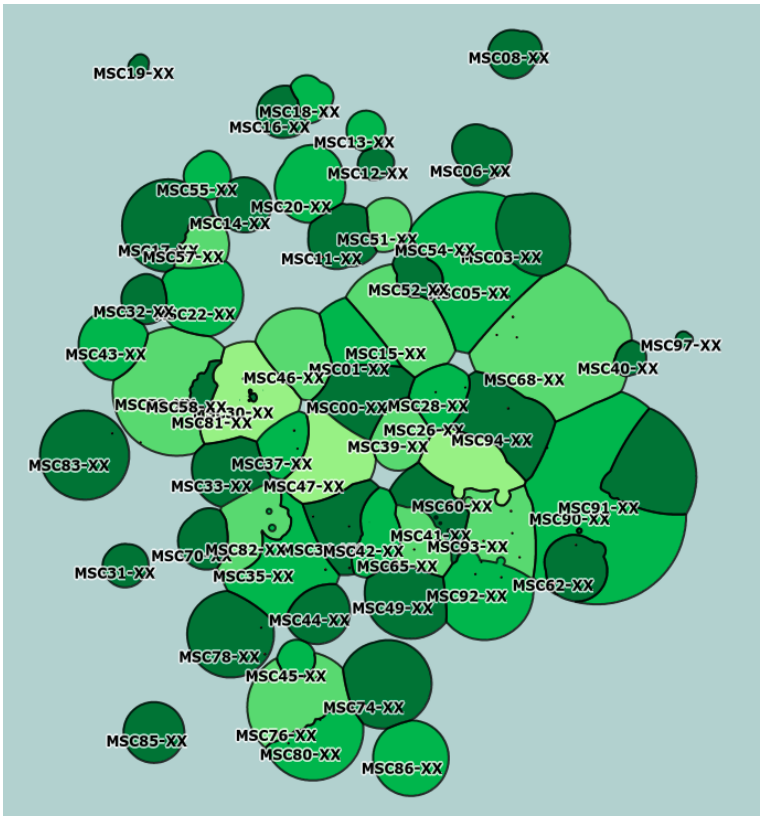


Fig. 4: Geometry of the Math Subject Classifications

3.4 Hierarchical Map Geometry

As the MDS computation becomes intractable for larger similarity matrices[3] we opt for a hierarchical approach to determining finer-grained map geometries (taking second-level and leaf MSC classes into account). In order to incorporate the second-level classes into the map we considered each top-level class independently. We had to give up on information of surrounding MSCs to make calculations tractable. For second-level classes belonging to the same first-level class we created inner-class similarity matrices. This led to 63 different matrices with sizes between 1×1 and 20×20. Then the same operations as before were applied.

Equipped with a color palette resembling natural maps we were able to create a map seen in Figure 4. This map incorporates the Zentralblatt Math data for the first two levels of the MSC. A close-up of second-level classes can be found in Figure 5. In order to distinguish their borders from borders of first level classes they were rendered in light-gray.

3.5 Landmarks and Settlements

Fig. 5: Adding Settlements

Next we populate the map geometry with "cities", "towns", and "villages": we view every classified paper as an "inhabitant", and compute the center of gravity of the points representing the respective MSC codes. As the number MSC combinations is finite, this yields a finite number of positions. The positions are rendered as red dots (see Figure 5 on the left), where the size encodes the number of papers.

As this could lead to up to 3 million different cities, a "population threshold" needs to be defined, to avoid a cluttering of the map with cities. Note that papers with only a single MSC classification will end up in the "capitals" of the respective class. Difficulties arise when it is ambiguous in which MSC a given city lies, or if the computed location is inside water. Those cases need special attention and the cities have to be moved manually.

[3] MDS is super-linear; for the first-level 63×63 matrix computation takes seconds and exceeds 30 hours even for the 528×528 matrix of second-level MSC classes.

4 Mapmaking & Deployment

The next step is to convert the geometry data from the last section into a map that has the features we are used to from familiar maps ranging from street maps to Google Earth. For that we apply a series of transformation to the geometry, these include smoothing the coast line, adding names for the geographic features, and structuring the uniform blue of the ocean. After creating a map with these transformation and extensions of the map geometry, we can deploy the map in a geographical information system; in our case OpenStreetMap [OSM]. For that, the map is transformed into the OSM map representation XML files with points identified by latitude and longitude and ways (point sequences) representing outlines, borders, and contours. Figure 6 shows the result, where we use a "political map" metaphor, coloring the areas by the colors Dave Rusin used in his Math Atlas (see Figure 2 on the right). These colors are distributed "0^{th} level of classification" designed by Rusin to highlight super-domains like Analysis (green), Algebra (red), applications (blue), etc.

To render maps from the data contained in the XML files we make use of the rendering service Maperitive [Map13], a free desktop application that generates small pieces of the maps called "tiles". A tile in OpenStreetMap is a 256×256 pixels sized PNG image making up for a given part of the map, examples are given in Figure 7. Tiles need to be generated only once in order to render a map since the contained information is re-used. Additionally tiles allow for zooming, since they can be pre-generated for different resolutions of the map. This enables us to fill the map with many details which can be shown or hidden depending on the current zoom level.

Note that there is no encoding of the height in Figure 4, this leaves room for visualizing additional information. We are currently experimenting with encoding the "activity level" of an area with this: We can compute the "elevation of an area" by counting the (relative) number of publications in that area e.g. in the last year. This makes research hotspots peaks that can serve as additional landmarks in the map.

5 Topical Maps

Aside from the default map of mathematics described above, we can generate topical maps that visualize additional features of the data. For instance, we can apply the mapping procedure to the data up to a given publication date and obtain a series of maps that visualize the "continental drift" of areas in mathematics; see Figure 8. Note that the maps are more "bubbly" than the two-level map in Figure 4, as they only comprise the top-level MSCs. Nevertheless, we can already see interesting trends: The most obvious one is that the field of Mathematics is growing rapidly. Furthermore, some subdomains of mathematics move around, since their distance to other subdomain change; e.g. the cluster of statistics (MSC62-XX; yellowish/ocre), operations research (MSC 90-XX), and economics (MSC91-XX) moved from the north/west in the 1980ies to the current

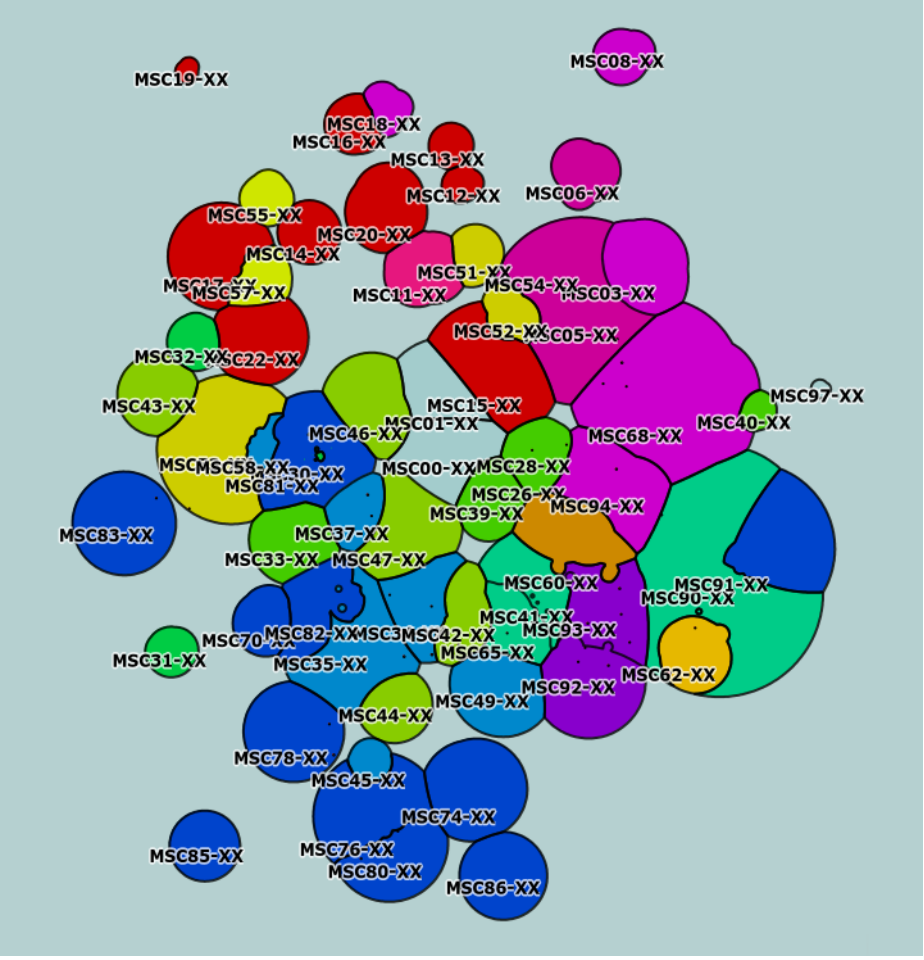


Fig. 6: A "Political Map" of Mathematics using Rusin's Color Coding

position on the south-east. There is a corresponding movement of Computer Science (MSC 68-XX) which has grown in size with the advent of computational methods in mathematics.

As we have data about the journals papers are published in, we can also compute the centers of gravity of journals (based on those of the papers they publish). Based on a radial function technique similar to the one we used to determine MSC areas, we can compute "journal dominions" and generate maps colored by these based on the MSC-induced geometry. Such a map could give mathematicians orientation in the jungle of currently ca. 3500 journals, but might be controversial for publishers.

Fig. 7: OSM tiles showing Northern and Eastern Europe, and Jacobs University Bremen and its surroundings, © OpenStreetMap contributors

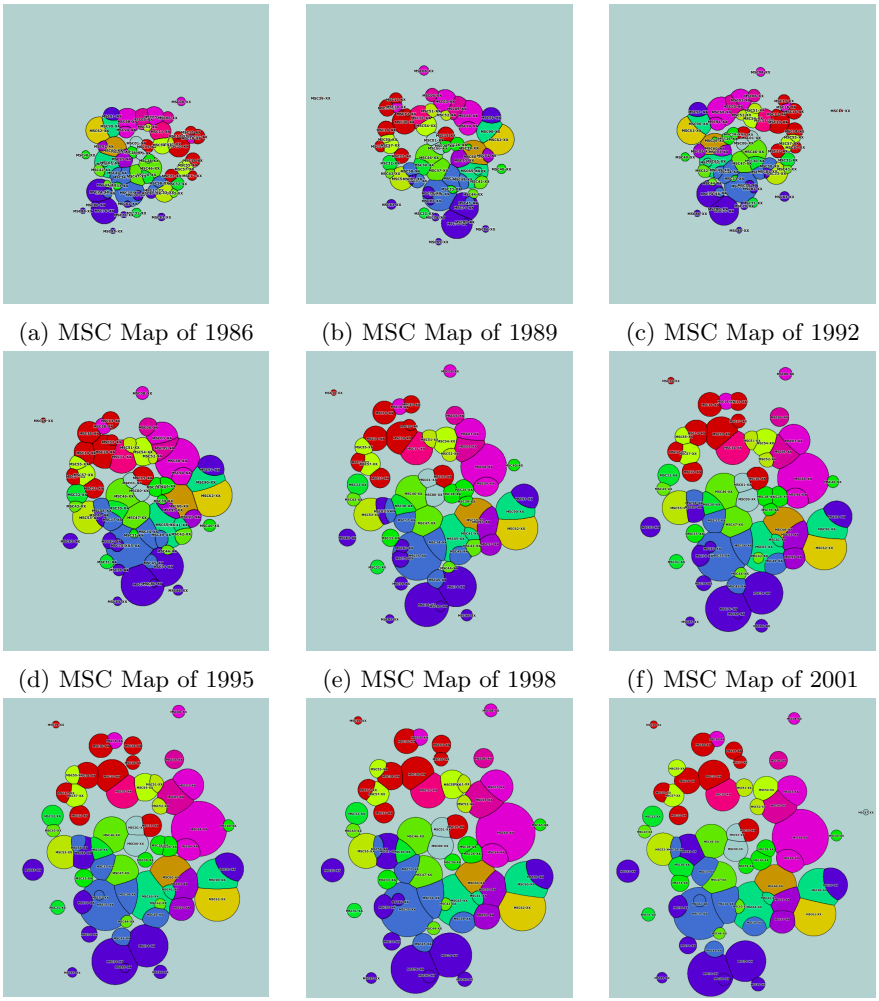
6 Related Work

There are other systems that visualize data in the form of generated maps. The first one is Dave Rusin's "Mathematical Atlas" [MathAtlas] which has been an initial starting point of OpenMathMap. Rusin displays the MSCs as bubbles, where the surface areas of the bubbles are proportional to the number of "recent" (at the time of creation) papers in that area [Rus02] (see Figure 2). Just as in OpenMathMap, the placement of the MSCs is determined by the frequency of cross-listings of papers in two or more areas: the distance of MSCs is inversely correlated to this [Rus02]. Rusin only considers the first level MSCs (61 in MSC 2000) and thus performs a 2-dimensional projection of a 61-dimensional space. This was not perfectly possible and so Rusin did some small modifications manually. He moved outliers inside to reduce wasted space, and spread overlapping bubbles apart into the nearest empty space [Rus01].

OpenMathMap extends the Mathematical Atlas in several ways. It is based on more recent data, integrates second-level MSC, generates non-overlapping territories that make MSCs more recognizable, and visualizes the map in the interactive OpenStreetMap system whose zoom and pan features allow to integrate much more information into the map.

Kuhn et al. [KLN12] describe the process of creating a thematic software map. They defined a similarity measure for code of open source programs based on the used "vocabulary". Just like in OpenMathMap they compute a two-dimensional configuration by Multidimensional Scaling (see Figure 9a). Around every point (representing a class) in this configuration they draw a circle where the area was proportional to the lines of code of a given class (see Figure 9b) and use normal distributions to arrive at a height that is visualized with contour lines and shadings (see Figure 9c).

The main difference lies in the choice of radial functions and the lack of borders – admittedly the "political map" metaphor used in OpenMathMap would



(a) MSC Map of 1986 (b) MSC Map of 1989 (c) MSC Map of 1992

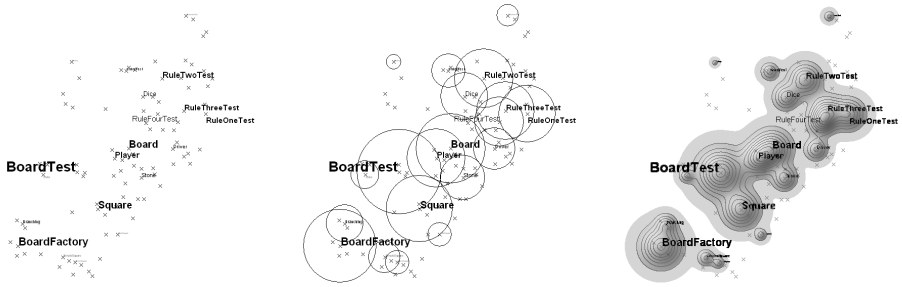
(d) MSC Map of 1995 (e) MSC Map of 1998 (f) MSC Map of 2001

(g) MSC Map of 2004 (h) MSC Map of 2007 (i) MSC Map of 2010

Fig. 8: MSC Maps evolving between 1986 and 2010

be less immediate for software classes than for MSCs. We also observe that the "topographic map" metaphor underlying the Kuhn et al.'s maps uses the third dimension (height) that is still free in OpenMathMap.

In [GJ12] Gronemann and Jünger provide a method how to visualize clustered graphs in a topographic map. After constructing a clustered graph they apply fat polygon partitioning and extract a mesh from the layout that models the terrain features based on the clustering. Finally they apply an edge-routing algorithm to the clustered graph. Their visualization of a collaboration graph filled with data from the Graph Drawing E-print Archive [GDEA13] is available at [Gde]



(a) After MDS (b) Adding Circles (c) Visualizing height

Fig. 9: Steps in creating a thematic software map. Images taken from [KLN12]

Fig. 10: Author Search given Publication Data

and includes interactive search interface (see Figure 10). Although their methods differ from the ones used here the concept is very similar in both approaches. We plan to include similar highlighting of collaborating authors in a future version of OpenMathMap.

7 Interactive Services & Mashups

Having our map deployed on OSM already gives us some base-level interactivity: zooming and panning. Additional location-based interactions can be implemented by adding custom JavaScript to the pages served by OSM subject to availability of date. The JavaScript API Leaflet [Lea12] used to render a contiguous map from the generated tiles allows us among others to resolve mouse-clicks on the map to latitude and longitude coordinates. However, just retrieving the location is not of big interest to us, much rather we would like to know which

MSC was clicked on. In order to map coordinates to MSCs we make use of a RESTful web-service. It takes latitude and longitude as parameters and returns name and description of the MSC in JSONP format allowing for cross-domain communication.

Figure 11 shows an example of this. Mouse-clicks on the map trigger a pop-up providing name and description of the clicked MSC in addition to the respective links to PlanetMath and Zentralblatt MATH. Another immediate example is the generation of custom queries for publication databases like Zentralblatt Math [ZBMath]: a right-click on the map could generate a query to ZBMath for all papers in the "vicinity"; a query for papers in a selected area would be similar. Yet another service might be a query for journals based on the data prepared for the "journal map". Another service might be to localize mathematicians by their publication record[4] and give them a "home address" according to their primary research topic (based on the center of gravity of their publications; possibly with weekend retreats in the woods of an application area). Similarly, research trajectories of mathematicians could be plotted on the map by computing yearly centers of gravity.

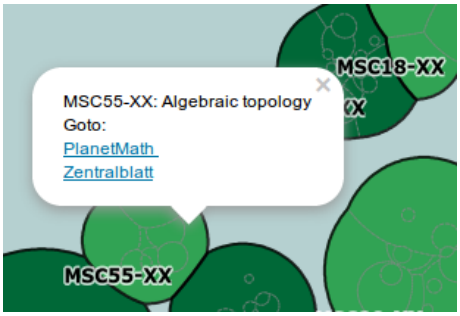


Fig. 11: User Interaction using WebService information

A Web3.0 style interaction could be to allow (groups of) users to name settlements, e.g. in honor of one of the most prolific contributors or landmark papers.

Finally, we could use the math maps as a target for mashups of external services. For instance, the search results of a mathematical search engine e.g. [KMP12] could be shown by localizing them on the OpenMathMap service.

8 Conclusion & Future Work

We have presented a novel access method to mathematical knowledge and resources that makes use of the highly evolved cognitive skills of spatial representations in humans. We have implemented a first prototype (accessible at `http://map.mathweb.org`) that deploys maps computed from mathematical publication data in a standard map server and instruments it with information services.

The geometry computation part of OpenMathMap system is licensed under GNU General Public License [Fre91] and the web front-end part under the GNU Affero General Public License [Fre99]. The code can be obtained from github at [Ope].

[4] We acknowledge that author identification in large corpora is a hard problem that is only partially solved for the mathematics community.

This prototype is just a first step we want to use in experimentation in human-oriented access methods to mathematics. We could imagine that connections between mathematical areas could be implemented as roads, highways or air/sea connections (possibly depending on their salience), important theorems could be entered/visualized as landmarks, and finally, we could imagine to go from interactive map servers to much more immersive environments (from Minecraft to second life).

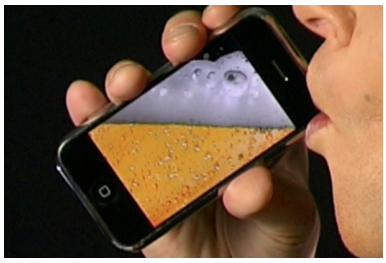
With the current approach we encountered a few problems when two relatively big areas are very close together in the map. In this case the shared area is split between the two MSCs so that the total area is substantially less than the sum of the areas if there were no bordering MSCs. In order to prevent this problem one could implement a force-based model where we allow large MSCs to "push away" surrounding MSCs. Thus even though we want a small overlap between MSCs to have a contiguous land area this overlap would be minimized. This step would have to be executed after MDS but before the application of the RDF since it has a direct influence on the configuration found by MDS.

The current implementation of MDS achieves a Stress-1 coefficient of 0.25 according to Equation (2). While this implies a poor configuration according to Kruskal (see Section 3.1) one has to keep in mind that reducing a 63-dimensional configuration to a 2-dimensional one is certainly not an easy task. An example where a high similarity did not lead to neighboring MSCs are Algebraic Geometry (MSC14-XX) and Complex Variables (MSC32-XX). The map implies that both classes have a higher similarity with Manifolds and cell complexes (MSC57-XX) which is not true.

Unfortunately, in this case the stress can not be decreased without increasing the target dimension (which would mean we have to drop the map metaphor). However, with the introduction of WebGL enabling 3D rendering in browsers and Google Maps already making use of it (`http://maps.google.com/gl`) one can imagine to also develop a 3D version of the MathMap in the future. With the current similarity values an MDS to three dimensions would decrease the stress value to 0.16. But we can also think of non-geometrical ways of "relieving" stress in the map via other map features. For instance, we could shorten the travel distance between two MSCs that are stressed by being too far apart by adding highways or shipping lines.

Finally, we acknowledge that the motivation for the OpenMathMap project was a cognitive question, which we have answered with a technical system.

Even though first feedback from mathematicians ranged from puzzled to enthusiastic (with an emphasis on the latter), we will have to systematically evaluate whether OpenMathMap-like systems and services can help with mathematician's day-to-day navigation problems and access tasks, or if OpenMathMap is essentially the equivalent to the iPhone beer app, a useless, but fun gadget.



References

[ACM-CCS98] *The 1998 ACM Computing Classification System*. 1998. URL: http://www.acm.org/about/class/ccs98 (visited on 11/18/2009).

[Dör13] Jan Wilken Dörrie. "OpenMathMap: Accessing Math via Interactive Maps". B. Sc. Thesis. Jacobs University Bremen, 2013. URL: https://svn.eecs.jacobs-university.de/svn/eecs/archive/bsc-2013/jdoerrie.pdf.

[Gde] URL: http://www.informatik.uni-koeln.de/public/graphmap/gdnet/web/index.php.

[GDEA13] Graph Drawing E-print Archive. *Welcome to GDEA*. 2013. URL: http://gdea.informatik.uni-koeln.de/ (visited on 05/09/2013).

[GJ12] Martin Gronemann and Michael Jünger. *Drawing Clustered Graphs as Topographic Maps*. 2012. URL: http://gdea.informatik.uni-koeln.de/1288/.

[Jin10] Arif Jinha. "Article 50 million: an estimate of the number of scholarly articles in existence". In: *Learned Publishing* 23.3 (2010), pp. 258–263. DOI: 10.1087/20100308.

[KLN12] Adrian Kuhn, Peter Loretan, and Oscar Nierstrasz. "Consistent Layout for Thematic Software Maps". In: *CoRR* abs/1209.5490 (2012).

[KMP12] Michael Kohlhase, Bogdan A. Matican, and Corneliu C. Prodescu. "MathWebSearch 0.5 – Scaling an Open Formula Search Engine". In: *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics (CICM) (Bremen, Germany, July 9–14, 2012). Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 342–357. ISBN: 978-3-642-31373-8. URL: http://kwarc.info/kohlhase/papers/aisc12-mws.pdf.

[KW78] Joseph B. Kruskal and Myron Wish. *Multidimensional Scaling (Quantitative Applications in the Social Sciences)*. Sage Publications, Inc, Jan. 1978. ISBN: 9780803909403.

[Lea12] Leaflet. *An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps by CloudMade*. 2012. URL: http://leaflet.cloudmade.com/ (visited on 12/03/2012).

[LI10] Peder Olesen Larsen and Markus von Ins. "The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index". In: *Scientometrics* 84.3 (2010), pp. 575–603. DOI: 10.1007/s11192-010-0202-z.

[Map13] Maperitive. *Maperitive*. 2013. URL: http://maperitive.net/ (visited on 05/05/2013).

[Mat] MathWorks. *Nonclassical multidimensional scaling*. URL: http://www.mathworks.com/help/stats/mdscale.html (visited on 03/04/2013).

[MathAtlas] Dave Rusin. *The Mathematical Atlas, a Gateway to Modern Mathematics*. URL: http://www.math-atlas.org/ (visited on 11/18/2009).

[MathSearch] *MathSearch*. Project Homepage. URL: http://mathsearch.kwarc.info/ (visited on 02/22/2013).

[MDS05] "MDS Models and Measures of Fit". English. In: *Modern Multidimensional Scaling*. Springer Series in Statistics. Springer New York, 2005, pp. 37–61. ISBN: 978-0-387-25150-9. DOI: 10.1007/0-387-28981-X_3. URL: http://dx.doi.org/10.1007/0-387-28981-X_3.

[Msc] *Mathematics Subject Classification MSC2010*. 2010. URL: http://msc2010.org (visited on 11/16/2011).

[Ope] *KWARC/openmathmap - github*. URL: https://github.com/KWARC/openmathmap.

[OSM] *OpenStreetMap*. URL: http://www.openstreetmap.org/ (visited on 09/01/2012).

[PACS10] *Physics and Astronomy Classification Scheme (PACS)*. 2010. URL: http://aip.org/pacs/ (visited on 12/17/2011).

[Rus01] Dave Rusin. *About the MathMap image*. 2001. URL: http://www.math.niu.edu/~rusin/known-math/collection/mathmap.html (visited on 12/02/2012).

[Rus02] Dave Rusin. *Information about the Mathematical Atlas collection*. 2002. URL: http://www.math.niu.edu/~rusin/known-math/collection/index.html (visited on 12/02/2012).

[Wik12] Wikipedia. *Radial basis function*. 2012. URL: http://en.wikipedia.org/w/index.php?title=Radial_basis_function&oldid=544074242 (visited on 05/08/2013).

[ZBMath] *Zentralblatt MATH*. URL: http://www.zentralblatt-math.org/zbmath/ (visited on 06/12/2012).

[Fre91] Free Software Foundation. *GNU General Public License*. 1991. URL: http://www.gnu.org/copyleft/gpl.html (visited on 05/22/2013).

[Fre99] Free Software Foundation. *GNU Affero General Public License*. 1999. URL: http://www.gnu.org/licenses/agpl.html (visited on 05/22/2013).