

Interactive Maps of Mathematics

Jan Wilken Dörrie

*Computer Science
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany*

*Type: Guided Research Proposal
Date: December 7, 2012
Supervisor: Prof. Dr. Michael Kohlhase*

Abstract

My Bachelor's Thesis tries to solve the problem of Mathematics evolving rapidly today by a visualization of the world of Math into an interactive map. This idea is motivated by the fact that humans in general have a intuitive understanding of maps and are able to navigate around them easily. This work is inspired by Dave Rusin's "Math Atlas" which models Mathematics as a set of bubbles, but lacks interactivity. The map orientates itself strongly on AMS' Mathematics Subject Classification and is populated with data taken from Zentralblatt Math. The visualization of the map will be done through the help of OpenStreetMap where the relations between different MSCs will be realized with continents, oceans, mountains, roads, high ways and train rails. The implementation will provide options for the user to change the appearance of the math world dynamically.

Contents

1	Introduction	3
2	Related Work	3
2.1	The Mathematical Atlas	4
2.2	OpenStreetMap	4
2.3	Math Subject Classification	6
2.4	Zentralblatt MATH Publication Data	6
2.5	Similarity and size of MSCs	7
2.6	Multidimensional Scaling	7
2.7	Mass-Spring-System	8
3	Work Plan	8
4	Timeline	10
5	Evaluation Criteria	10
6	Conclusions	11

1 Introduction

Mathematics is a very quickly expanding science, even more so since the beginning of the Internet. Even though this choice is highly subjective, John von Neumann is considered one of the last “universal mathematicians” [Mat10], however he passed away more than 50 years ago. Since his death in 1956 there have been more than 2.8 million [MAT12a] indexed publications, with 1.8 million publications in the last 22 years [MAT12b]. One of the biggest reasons for this rapid increase in publications is certainly the existence of the world wide web.

Since then people tried to solve the challenge of making this large amount of data more accessible, where one of the most promising ideas is Dave Rusin’s “Math Atlas”. Rusin used the classification of mathematics into Math Subject Classes (MSCs) for his atlas, where the size of a given MSC represents the number of publications. Moreover he managed to incorporate the similarity between two MSCs by the distance they have in the atlas, short distances mean large similarities and long distances correspond to small similarities. Similarity can be displayed due to the fact that usually publications are not only part of one specific MSCs, but combine topics from several different MSCs. By keeping track of how many publications refer to two distinct MSCs it is possible to define their similarity.

However, Rusin did his work in the late 90s, so that the whole atlas is very static and seems to be old-fashioned compared to modern visualization standards. Here is where my Guided Research Topic “Interactive Maps of Mathematics” begins. I will take Rusin’s idea of modeling MSCs by their size and similarity, however the modeling will not be done through colored bubbles on a white background. It will be achieved through a transformation of the world of mathematics into a world map consisting of continents, oceans, mountains and so on. The motivation for this implementation is the fact that humans in general have a intuitive understanding of planar maps and are skilled to navigate around them easily. Moreover this implementation allows for dynamic changes of the environment through user interaction. For example it will be possible to restrict the time frame the displayed populations are chosen from.

2 Related Work

For the related work one needs to especially look at Dave Rusin’s work to understand how he tackled the problem of visualizing Mathematics and where his work can be improved. Building on that OpenStreetMap needs to be considered to be able to render individual maps. After that one has

to investigate the provided input data to be able to extract the relevant data. Finally one needs to focus on Mathematical techniques to create a good looking two dimensional rendering of the processed input data.

2.1 The Mathematical Atlas

This Guided Research is largely inspired by Dave Rusin’s “Mathematical Atlas” (<http://math-atlas.org>). 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]. The placement of the MSCs is determined by the frequency of cross-listings of papers in two or more areas. Due to this the similarity of two MSCs is inversely correlated to the distance of the bubbles in the map. [Rus02]. Since Rusin considers 61 bubbles he had to perform 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].

2.2 OpenStreetMap

The implementation of the map will be realized with the help of OpenStreetMap which is a free and edible map of the world [Ope12b]. Unlike proprietary datasets OpenStreetMap is open data and licensed under the Open Data Commons Open Database License [Ope12a].

OpenStreetMap XML

The OpenStreetMap XML file format [Wik12a] has a very simple structure and thus allows for easy creation and modification of any map, be it real life data or a custom map as in this case. It consists of a block of nodes, a block of ways and a block of relations. Nodes have a unique id and location given by latitude and longitude alongside with other information about the creator and time of creation. Moreover nodes can have tags in form of key-value pairs. Ways are a collection of nodes referenced by their ids. They have information about the creator and creation date as well and are also able to have key-value tags. Ways represent parts of any streets, for example highways or cycleways. Relations are a collection of nodes and ways and can represent tram-lines, buildings, parks, and so on. Once again they carry information about their creation and can have key-value tags. Due to the verbosity of XML the data is usually represented in binary format, however it is perfectly possible to use the plain-text version as well. As already said it’s very simple structure allows for easy creation of new and modification

of already existing map data. This is a very nice property for the creation of custom maps.

PostGIS Database

With increasing size of the map it becomes unfeasible to store the map data in plain OSM XML files. It is more reasonable to have a database containing map information. For this we make use of PostGIS, which is an extension of PostgreSQL and provides geospatial functions and additional tables [Wik11]. We use Osmosis, a command line Java application for processing OSM data, to populate the database with the data from the XML files. [Wik12b]

Tiles

In order to be able to run OpenStreetMap on a local machine we need to set up an own tile server. A tile in OpenStreetMap is a $256 * 256$ pixels sized PNG image making up for a given part of the world, examples are given in Figure 1. The tiles used by OSM are retrieved via the url [http://\[abc\].tile.openstreetmap.org/z/x/y.png](http://[abc].tile.openstreetmap.org/z/x/y.png), where z specifies the zoom level and x and y are the respective x and y coordinates for a given level. For example the left tile in Figure 1 has the values $z = 3$, $x = 4$, $y = 2$ and the right tile $z = 14$, $x = 8585$, $y = 5324$. z ranges between 0 and 18 and doubles the number of tiles in each dimension when it is increased by 1. Consequently, x and y take values between 0 and $2^z - 1$ [Wik12c].

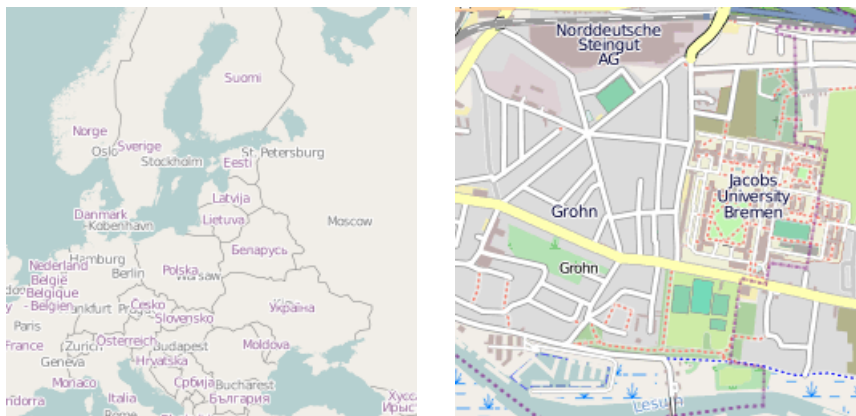


Figure 1: OSM tiles showing Northern and Eastern Europe, and Jacobs University Bremen and its surroundings

To be able to generate the tiles from the provided map data, a map renderer is needed. A good choice is Mapnik, because it is able to apply anti-aliasing

to the rendered images and supports OSM XML files and PostGIS databases as data sources.

Rendering a continuous map

Once we have a method to generate tiles we need to find a way to create a continuous map out of them. This is possible with the JavaScript library Leaflet [Lea12]. It allows for navigating around the map as well as zooming in and out. Relevant JavaScript code to implement it in a local test page is taken from switch2osm [swi12].

2.3 Math Subject Classification

In order to be able to relate different areas of Mathematics we need to understand how the Math Subject Classification works. MSC2010 [Soc09] is a combined effort of Zentralblatt MATH and the Mathematical Review Database to unify the classification of Mathematical publications. The result are 63 top level classes, 519 second level classes and 5557 third level classes summing up to 6139 classes in total. Each class is of the form Digit Digit Letter Digit Digit and a description, 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.

2.4 Zentralblatt MATH Publication Data

Having access to the classifications of mathematical publications done by Zentralblatt MATH I am able to extract relevant data for the visualization of the MSCs. The data provided has information about the article number, the authors, the classification code, keywords and the abstract. Of most interest are the article number and the classification codes. An example entry is the following:

```
AN: 1227.22008
AU: Arnal, Didier; Selmi, Mohamed; Zergane, Amel
CC: 22E27; 37J15; 22E45; 22D30
UT:
AB: Summary: In [Bull. Sci. Math. 135, No. 2, 141--165 [...]]
```

2.5 Similarity and size of MSCs

With the provided Zentralblatt data it is possible to calculate a value for the similarity between two given MSCs. This is done in the following way:

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

The size of the intersection/union of two MSCs are the number of papers referencing both/any MSCs. This results in $s \in [0, 1] \quad \forall \text{MSC}_i, \text{MSC}_j$ and $s(\text{MSC}_i, \text{MSC}_i) = 1$. The size of a MSC is simply the number of publications belonging to this class.

2.6 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 and a dissimilarity matrix

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

one needs to find I vectors $x_1, \dots, x_I \in \mathbb{R}^N$ such that $\|x_i - x_j\| \approx \delta_{i,j} \quad \forall i, j = 1 \dots I$. This can be formulated as an minimization problem, for example [Wik12d]:

$$\min_{x_1, \dots, x_I} \sum_{i < j} (\|x_i - x_j - \delta_{i,j}\|)^2$$

A solution may be found by numerical optimizations, for example Matlab has a built-in method `mdscale` which takes as arguments the n -by- 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 stress1 criterion [Mat12c]. This criterion is defined in the following way [KW78]:

$$\sqrt{\frac{\sum_i \sum_j (f(x_{ij}) - d_{ij})^2}{\sum_i \sum_j d_{i,j}^2}}$$

The stress value is a measurement for how well the returned configuration preserves distances, if the result is perfectly preserving, the stress value is 0.

2.7 Mass-Spring-System

The configuration generated by the previous respects the similarities between given points, however for our visualization we also need to take the size of the MSCs into account. In order to do this we make use of a Mass-Spring-System. This operates on a local range and requires a good initial configuration, however if this is given it will provide decent results reflecting the size and the similarities of MSCs. I am confident that MDS will create a good enough initial configuration such that a Mass-Spring-System will work.

3 Work Plan

The work will be split into the following work packages:

WP1 Retrieve data for similarities or compute myself

This part is already done, a simple formula to compute the similarity between two given MSCs is already developed (see 2.5). However, a more sophisticated algorithm might produce better results in the end. It would be interesting to see how Dave Rusin tackled this problem. Another interesting task it is to find a good algorithm to parse the provided data and calculate the similarity between any two MSCs. The naïve approach runs in $\mathcal{O}(n^2)$ which takes several hours for $n \approx 6000$. Therefore a better algorithm needs to be found which preferably has a time complexity of $\mathcal{O}(n)$. Also the algorithm needs to be able to work with already existing data, so that adding new data will not lead to another computation of all the data.

WP2 Find a suitable metric for the gathered data

The computed similarities in WP1 need to be transformed into a dissimilarity metric so that the algorithm from Matlab can be applied. A first guess at a suitable metric is the following [Wik12e]:

$$d_{ij} = \sqrt{c_{ii} + c_{jj} - 2 \cdot c_{ij}}$$

It remains to be shown if this formula works well in practice or a different approach needs to be done.

WP3 Apply multidimensional scaling

After retrieving the dissimilarity matrix in WP2 we apply multidimensional scaling (MDS) to it. The proposed way of doing so is to make use of Matlab's `mdscale` with an output dimension of 2 (see 2.6). In order to test the quality of the result we make use of the provided

stress factor as well as a visualization. It is very likely that some manual modifications similar to the ones Dave Rusin performed need to be done. This includes moving around overlapping vertices or to make better use of the provided screen space. In case the multidimensional scaling done by Matlab does not lead to sufficient result, another algorithm needs to be applied. I propose the following [Wik12e]:

- (a) Define the matrix $A = (a_{ij})$ with $a_{ij} = -\frac{1}{2}d_{i,j}^2$
- (b) Define the matrix $B = (b_{ij})$ with $b_{ij} = a_{ij} - a_{i\bullet} - a_{\bullet j} + a_{\bullet\bullet}$ where
 - $a_{i\bullet} = \frac{1}{n} \sum_{j=1}^n a_{ij}$ denotes the average of row i ,
 - $a_{\bullet j} = \frac{1}{n} \sum_{i=1}^n a_{ij}$ denotes the average of column j , and
 - $a_{\bullet\bullet} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n a_{ij}$ denotes the average of all a_{ij} .
- (c) Calculate the eigenvalues λ_i and eigenvectors $\gamma_i = (\gamma_{ij})$ of the matrix B such that $\sum_{j=1}^n \gamma_{ij}^2 = \lambda_i$.
- (d) Finally the coordinates of the points in an m -dimensional space can be taken from the m largest eigenvalues: $x_i = \gamma_i \sqrt{\lambda_i}$.

WP4 Apply the Mass-Spring System

After receiving a good initial configuration for the MSCs via MDS we also need to account for the size of the different MSCs. In order to do so we apply several iterations of a mass-spring system. To find an equilibrium we need to have a good initial configuration, but as I already stated I am confident that MDS provides this.

WP5 Decide on additional information for the map

Even though our map is a 2-dimensional object, it is possible to visualize another dimension, height. This is easiest done via color encodings, similarly to how it is done in regular maps, e.g. green for normal land area, (dark) brown for (high) mountains and (dark) blue for (deep) oceans. With this new possibility one needs to decide what information is encoded in height. One option is that MSCs from the same top level will be on the same height, and the gap between them will be filled forming a ridge in the end. Another possible option is to correlate height to the number of publications in the last n years of

that specific MSC. Doing this enables the user to change n manually, resulting into a very dynamic map, where mountains form into oceans and vice versa.

WP6 Modeling in a mapping service

After creating the map it needs to be implemented into an already existing mapping service, possible choices here are Google Earth or OpenStreetMap. Since OpenStreetMap is (as the name suggests) open source, it will be easier to create mapping data for this service. The steps to do so are described in detail in 2.2. The main task here will be to program a tool which takes the modeled map and outputs it in the OpenStreetMap XML format. Here it is also a challenge to provide the user with a choice of given parameters. Taking the example from WP5 changing n should result in a real time update of the map.

WP7 Deployment on the Internet

In the end it is planned to deploy the map on the Internet to serve as an intuitive access point for Mathematics. It will be used as the entry page to <http://mathweb.org> and an access method at <http://planetmath.org>.

4 Timeline

The work packages will be done at the following times:

Task	Done by
WP1: Retrieve data for similarities or compute myself	07/12/12
WP2: Find a suitable metric for the gathered data	05/02/13
WP3: Apply multidimensional scaling	20/02/13
WP4: Apply the Mass-Spring System	05/03/13
WP5: Decide on additional information for the map	10/03/13
WP6: Modeling in a mapping service	20/03/13
WP7: Deployment on the Internet	31/03/13
Writing Bachelor's Thesis	01/05/13
Submission Deadline	12/05/13

5 Evaluation Criteria

Since the result of this Bachelor's Thesis will be an interactive map, it is relatively hard to find objective evaluation criteria. The main criteria for

the user is how intuitive it will be to navigate around the map, however intuitiveness is very individual.

Nevertheless other important objective criteria are:

- Completeness with regard to the provided input data from Zentralblatt MATH
- Response time to user interactions
- Extensibility of the algorithm, i.e. how easy is it to add further data
- Quality of the applied multidimensional scaling (stress factor)

6 Conclusions

My Bachelor's Thesis takes Dave Rusin's idea of modeling Mathematics and extends it in several ways. Not only will it be based on more recent data, but also the visualization will be dynamic and allows for user interactions. In order to do so the map will be implemented in OpenStreetMap, so that it is possible to navigate around freely and to add further information which are incorporated into the map. This can be for example the number of recent papers in a given subject. For the initial map layout modeling techniques like multidimensional scaling and mass-spring-systems will be used. Additionally the algorithms will be extensible, allowing for easy addition of new data. Because of this my work will continue to be useful in the future and serve as an intuitive access point to mathematics and as a solid foundation for additional work on the same or a similar topic.

References

- [KW78] Joseph B. Kruskal and Myron Wish. *Multidimensional Scaling (Quantitative Applications in the Social Sciences)*. Sage Publications, Inc, January 1978.
- [Lea12] Leaflet. An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps by CloudMade. <http://leaflet.cloudmade.com/>, July 2012. [Online; accessed 3-December-2012].
- [Mat10] MathOverflow. Who is the last mathematician that understood all of mathematics. <http://mathoverflow.net/questions/27881/>, June 2010. [Online; accessed 3-December-2012].
- [MAT12a] Zentralblatt MATH. Simple Search. <http://www.zentralblatt-math.org/zblmath/search/?q=py%>

- 3A1956-2012, December 2012. [Online; accessed 2-December-2012].
- [MAT12b] ZentralBlatt MATH. Simple Search. <http://www.zentralblatt-math.org/zblmath/search/?q=py%3A1990-2012>, December 2012. [Online; accessed 2-December-2012].
- [Mat12c] MathWorks. Nonclassical multidimensional scaling. <http://www.mathworks.com/help/stats/mdscale.html>, December 2012. [Online; accessed 3-December-2012].
- [Ope12a] OpenStreetMap. Copyright and License. <http://www.openstreetmap.org/copyright>, December 2012. [Online; accessed 3-December-2012].
- [Ope12b] OpenStreetMap. The Free Wiki World Map. <http://www.openstreetmap.org/>, December 2012. [Online; accessed 3-December-2012].
- [Rus01] Dave Rusin. About the MathMap image. <http://www.math.niu.edu/~rusin/known-math/collection/mathmap.html>, July 2001. [Online; accessed 2-December-2012].
- [Rus02] Dave Rusin. Information about the Mathematical Atlas collection. <http://www.math.niu.edu/~rusin/known-math/collection/index.html>, March 2002. [Online; accessed 2-December-2012].
- [Soc09] American Mathematical Society. MSC2010. <http://www.ams.org/mathscinet/msc/pdfs/classifications2010.pdf>, December 2009. [Online; accessed 7-December-2012].
- [swi12] switch2osm. Getting started with Leaflet. <http://switch2osm.org/using-tiles/getting-started-with-leaflet/>, October 2012. [Online; accessed 3-December-2012].
- [Wik11] OpenStreetMap Wiki. PostGIS. <http://wiki.openstreetmap.org/w/index.php?title=PostGIS&oldid=607893>, March 2011. [Online; accessed 3-December-2012].
- [Wik12a] OpenStreetMap Wiki. OSM XML. http://wiki.openstreetmap.org/w/index.php?title=OSM_XML&oldid=801583, August 2012.
- [Wik12b] OpenStreetMap Wiki. Osmosis. <http://wiki.openstreetmap.org/w/index.php?title=Osmosis&oldid=834077>, November 2012. [Online; accessed 3-December-2012].

- [Wik12c] OpenStreetMap Wiki. Slippy map tilenames. http://wiki.openstreetmap.org/w/index.php?title=Slippy_map_tilenames&oldid=801364, August 2012. [Online; accessed 3-December-2012].
- [Wik12d] Wikipedia. Multidimensional scaling. http://en.wikipedia.org/w/index.php?title=Multidimensional_scaling&oldid=525924960, December 2012. [Online; accessed 3-December-2012].
- [Wik12e] Wikipedia. Multidimensionale Skalierung. http://de.wikipedia.org/w/index.php?title=Multidimensionale_Skalierung&oldid=111175245, December 2012. [Online; accessed 7-December-2012].