A Map-Like Visualisation Method Based on Liquid Modelling

Robert P. Biuk-Aghai[a,*], Muye Yang[a], Patrick Cheong-Iao Pang[b], Wai Hou Ao[a], Simon Fong[a], Yain-Whar Si[a]

[a]*Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Avenida da Universidade, Taipa, Macau S.A.R., China*
[b]*Department of Computing and Information Systems, The University of Melbourne, Victoria 3010, Australia*

Abstract

Many applications produce large amounts of data, and information visualisation has been successfully applied to help make sense of this data. Recently geographic maps have been used as a metaphor for visualisation, given that most people are familiar with reading maps, and several visualisation methods based on this metaphor have been developed. In this paper we present a new visualisation method that aims to improve on existing map-like visualisations. It is based on the metaphor of liquids poured onto a surface that expand outwards until they touch each other, forming larger areas. We present the design of our visualisation method and evaluations we have carried out to compare it with an existing visualisation. Our new visualisation has better usability, leading to higher accuracy and greater speed of task performance, as well as a lower error rate.

Keywords: Information visualisation, polygon expansion, liquid modelling, map, Wikipedia, category hierarchy

1. Introduction

Today's internet-scale applications produce huge amounts of data. Applications in the domains of social networking, collaborative filtering, online discussion, user-contributed content and others have up to hundreds of millions of users generating large amounts of data on an ongoing basis. There is much value to be gained in understanding this data and perceiving patterns in it. However, given the amount of data, doing so remains a challenge. Information visualisation can be employed to visually represent large amounts of data in a form that makes patterns and associations in the data appear as visual patterns, which aids assimilation of data [1].

*Corresponding author
Email addresses: robertb@umac.mo (Robert P. Biuk-Aghai), muye.yang@gmail.com (Muye Yang), cipang@unimelb.edu.au (Patrick Cheong-Iao Pang), aowaihou@hotmail.com (Wai Hou Ao), ccfong@umac.mo (Simon Fong), fstasp@umac.mo (Yain-Whar Si)

Our research is framed within this context of perceiving patterns in large-scale user-contributed content. More specifically, we focus on visualising large hierarchies, namely the category data of Wikipedia, "the free encyclopedia that anyone can edit" (Wikipedia's own slogan). Since it was created in 2001 Wikipedia has experienced tremendous growth. Today (March 2015) there are over 4.8 million articles, over 24 million registered users, and over 760 million edits in the English Wikipedia alone[1], and there are more than 280 other language editions. Moreover, the English Wikipedia keeps growing at a rate of around 800-900 new articles every day, and is edited about 3 million times per month[2].

Wikipedia articles are classified into categories. In English Wikipedia alone there are about 2 million categories, and it is common for an article to be assigned to multiple categories, in some cases dozens of categories. The classification of an article to its categories is displayed on each article page, but it is difficult to obtain a larger picture of the distribution of articles among categories. Such an overview would be useful to gain an understanding of the distribution of content among topic areas, which reflects their relative importance within the community of their contributors.

Previously we had devised a method for visualising the Wikipedia category hierarchy in the form of a geographical map, using an approach of tiling a plane of hexagons [2]. Experience with using that visualisation led us to identify several issues related to the readability and visual quality of the map, thus we set out to devise a new method that would produce a more readable visualisation of better visual quality. An early design of this new method, which places areas in the map by expanding polygons instead of tiling hexagons, was documented in a previous paper [3]. Our early design, however, also had some shortcomings, specifically it could not guarantee that areas in the map are displayed in the correct size since areas surrounded on all sides by other areas are prevented from expanding to their intended size. Since then we have significantly revised the design of our new visualisation method, solving its earlier shortcomings and overcoming the weaknesses of our previous hexagon-tiling method. In this article we present the design and evaluation of our new method.

The remainder of this article is organized as follows. Section 2 briefly reviews related work on map-like visualisation. Then we introduce the design of our visualisation method in Section 3, and present an evaluation of our visualisation in Section 4. Finally, we make conclusions in Section 5.

2. Related Work

Structure and composition of large hierarchies are difficult to perceive, and various traditional visualisation techniques have been developed to try to visualise such data, including tree-maps [4], voronoi treemaps [5], mosaic plots [6], cone trees [7] and hyperbolic trees [8]. Tree-maps make efficient use of display space, but despite their name they do not actually look like (geographic) maps. Voronoi treemaps are an interesting variation of the traditional treemap, and can assume any shape, not only rectangular

[1] http://en.wikipedia.org/wiki/Special:Statistics
[2] http://stats.wikimedia.org/EN/TablesWikipediaEN.htm

shapes, but have a very artificial visual appearance. Mosaic plots somewhat resemble tree-maps in that they have an overall rectangular shape, and rectangular sub-divisions. Cone trees and hyperbolic trees, on the other hand, are some of the techniques used to represent the tree structure of hierarchical data visually as nodes and edges, and as such make no attempt to resemble a map. In sum, these traditional techniques all visualise hierarchical data using a more or less abstract representation.

A more recent approach [2, 9, 10, 11, 12] is to visualise hierarchical and other relational data in the form of a geographic map. Maps consist of land and sea, of countries, provinces and counties that are separated by borders and that are labelled with their name. Maps are usually also coloured to represent a certain aspect of the terrain, such as elevation in the case of topographic maps, or another attribute such as economic output, unemployment rate, language spoken or others in the case of a thematic map.

The map form can be used to represent non-geographic data as well. Clusters of data can be represented as regions in the map, and attributes of this data can be mapped to visual attributes of map objects. The use of the map form has the significant advantage that most people understand geographic maps easily thanks to early exposure to maps in school. Even among pre-school children essential mapping abilities are well developed [13]. Elements such as mountains, valleys, land, sea, rivers, and cities, as well as the meaning of each, are readily recognized by people even without special training. Therefore visualising information structures in the form of a geographic map enables people to relate to such representations more easily without requiring prior instruction.

Skupin has used the map form to represent the geographic knowledge domain [9]. Taking 2200 conference abstracts, he applied the self-organizing map (SOM) method to produce a map-like representation of the topics contained in this document collection. Topics were clustered hierarchically, and this hierarchy was mapped to areas in the map. The final map output and labelling were produced by a GIS (geographic information system) software.

Hu et al. have devised another method for representing dynamic relational data in map form [10]. The approach, called GMap, transforms an arbitrary graph into a map representation. It represents clusters of nodes as areas in the map, and displays most areas fused together into one large continent, with only a few small separate islands, if any.

Gronemann and Jünger have visualised clustered graphs as topographic maps [11]. The input is a graph with an overlaid tree, expressing both relationships and hierarchies among nodes. This graph data is preprocessed to determine node placement, from which a triangle mesh is generated that assigns an elevation to each node. The result data is fed into a GIS software that outputs a map showing islands of areas whose elevation is above sea level, whereas other areas that have a negative elevation appear as submerged below the sea.

Auber et al. have proposed another method that produces visualisations resembling a geographic map, which they term GosperMap, as it relies on the use of a Gosper curve [12]. The input data has a tree structure, the leaves of which are projected to a 2D space-filling curve. Finally, regions containing nodes are filled to become areas corresponding to sub-trees in the input tree. The resulting maps consist of a single

continent with countries that have highly irregular borders.

Biuk-Aghai and Pang have devised a different method to create map-like visualisations [2]. Their method takes a tree of hierarchical data and constructs a map that represents nodes at different levels as nested areas, reflecting the hierarchy. Areas are constructed by tiling hexagons in a hexagon matrix, starting from the area's centre and moving outwards in random order.

Examples of each of these five map-like visualisations are shown in Figure 1.

Our new visualisation method presented in this article aims to overcome shortcomings of our previous work that used the hexagon-tiling approach [2] and that resulted in areas with rough border lines, as can be seen in Figure 1e. Details of our new method are presented in the following section.

Our method makes use of force-directed layout and overlap removal algorithms. Here we give a brief overview of a few related algorithms in these two areas.

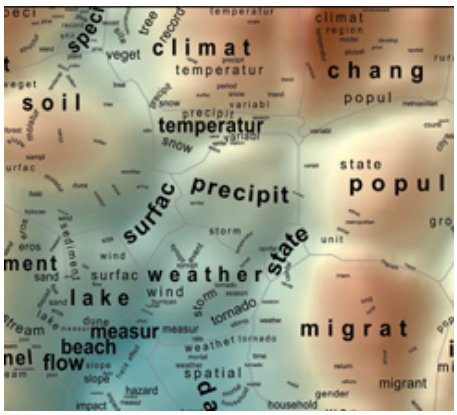
Force-directed layout algorithms position nodes in a graph, in either two or three dimensions, modelling the edges of the graph after physical springs that contract when extended; most algorithms in this class also model nodes of the graph as electrically charges particles that repel each other. Many such algorithms exist, the most well-known of which include those by Kamada and Kawai [14], and Fruchterman and Reingold [15]. Kobourov provides a good review of this class of algorithms [16]. Our visualisation method makes use of the algorithm of Kamada and Kawai which only uses spring forces and no repulsion forces, and is thus simpler. Moreover, many implementations of this algorithm are available, facilitating use in our visualisation prototype.

Overlap removal algorithms take a diagram consisting of overlapping objects and transform it into a diagram that is free of overlaps. There are many overlap removal algorithms, including, among others, the Force-Scan Algorithm (FSA) [17], Force-Transfer Algorithm (FTA) [18], and Mixed Integer Optimization for Layout Arrangement (MIOLA) algorithm [19]. They work by pushing overlapping geometric objects away from each other until no more overlap exists. These algorithms have in common that they preserve the mental map of the user [20], i.e. the arrangement of objects in a visualisation after the overlap removal resembles the arrangement before the overlap removal. In our method we employ the Force-Transfer Algorithm because of its speed and the resulting compactness of the layout.

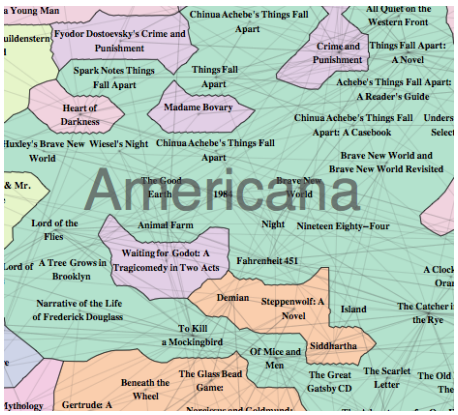
3. Visualisation Method

We have devised a novel map-like visualisation method that differs from the other approaches introduced in Section 2 above. An earlier version of this liquid modelling method was presented in 2013 [3]. We have further significantly evolved our method to overcome some limitations that existed, and to improve the visual quality of the result, which we present here. This method has a smoother appearance, and has better usability than our previous hexagon-tiling visualisation [2].

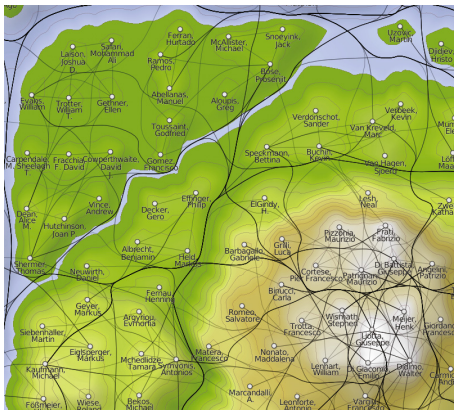
Our visualisation method is loosely based on the metaphor of expanding liquids: each area in the map is represented by an immiscible liquid that is poured onto a point in a plane. As more of the liquid is poured, it expands outwards until it reaches a minimal height, at which it stops expanding. As liquids are immiscible (incapable of being



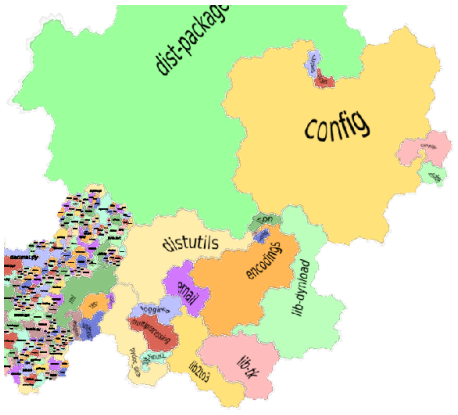
(a) Skupin [9]



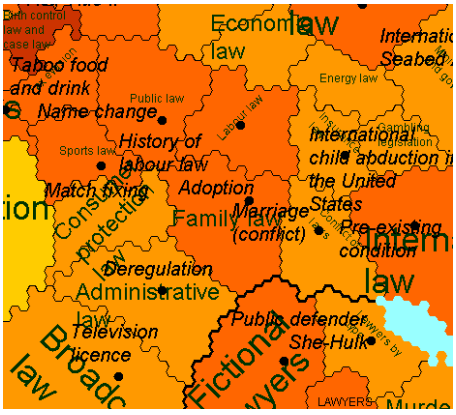
(b) Hu et al. [10]



(c) Gronemann & Jünger [11]



(d) Auber et al. [12]



(e) Biuk-Aghai & Pang [2]

Figure 1: Examples of existing map-like visualisations

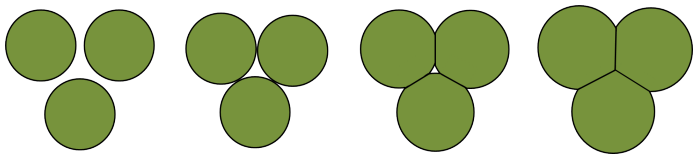


Figure 2: Area expansion

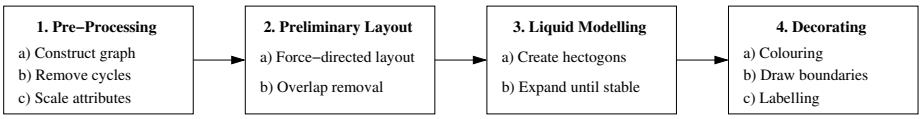


Figure 3: Visualisation process

mixed), when one liquid expands to touch another liquid, it can not expand further in that direction and will have to expand in a different direction where further expansion is possible. This is illustrated in Figure 2. However, if a liquid is surrounded by other liquids on all sides, it will push those other liquids outwards, which in turn may push other neighbouring liquids, until the outermost liquids have been pushed sufficiently outward. The liquids in all areas are assumed to have the same density and height, and only differ in the surface area occupied, so that the size of an area is proportional to its volume and mass. This volume or mass is mapped from a size attribute of the underlying data to be visualised.

Our method begins with all liquids having an equally small surface area but different height. As a result, each liquid will expand its surface area until its height reaches the minimum height and expansion stops. As mentioned above, in this process a liquid may push or be pushed by its neighbours. The entire process is loosely based on the observable behaviour of liquids, without strictly modelling all details of their physical behaviour.

The visualisation process consists of four main steps, as illustrated in Figure 3. In the first step, the input data is converted to a tree representation, which is then used as the input for the next step. In the second step, nodes are laid out using several layout methods, according to the estimated surface size, hierarchical relation, and similarity, producing a preliminary layout. The final position of each visible node is used as the initial position of that node's liquid. In the third step, the liquid modelling is applied until all areas reach a stable state. Finally in the fourth step the map is decorated using colours, borders, and labels. Next we explain these steps in detail.

3.1. Preprocessing

Our visualisation represents a hierarchy of data as nested areas in a map-like appearance – counties nested in provinces nested in countries, for example. To achieve this, our visualisation method requires its input data to have a tree structure, and each node in the tree to have an attribute that can be mapped to the size of its corresponding area in the visualisation. Preprocessing of the source data converts it into the required form, the details of which differ depending on the source data. In our case we visu-

alised the category hierarchy of Wikipedia, and the following explanation illustrates the pre-processing required for this particular data set.

Our source data are the publicly available Wikipedia database dumps made available by the Wikimedia Foundation[3]. From these we retrieve the database tables corresponding to category links and wiki pages. Retrieving the category links, we construct a graph consisting of nodes for Wikipedia categories and edges for parent-child relationships between categories. According to Wikipedia editing guidelines this graph is supposed to be a tree, however because category creation and assignment in Wikipedia is performed by human editors, it is possible that some anomalies exist in the graph, such as cycles. Thus we remove these cycles from the graph by performing a breadth-first search starting from the root node, and maintaining a list of visited nodes. When an edge causing a cycle is encountered it is simply removed. This results in a tree structure. In the process we also eliminate non-content categories which are meaningless for our visualisation.

Next, we select a size attribute to be mapped to the volume of each area. In the case of the Wikipedia data we use the article count, i.e. we aggregate a count of the number of articles that are assigned to each category. We then scale the size attribute using a logarithmic scale so as to avoid extremes of area sizes.

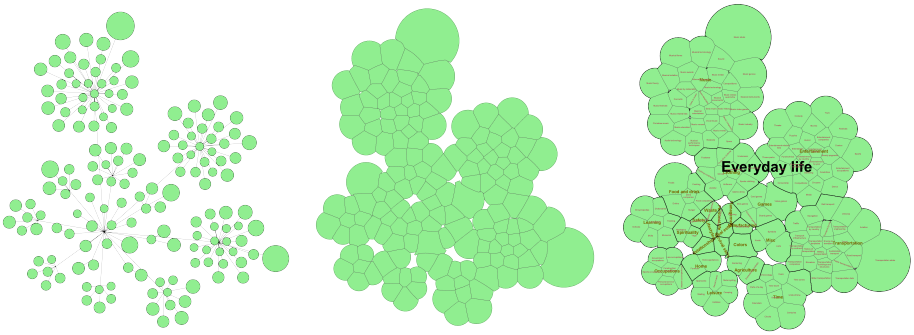
We also calculate similarities between categories using the method detailed in [21], which we use to place nodes in the preliminary layout. These similarity values express the degree to which different content categories share co-assigned articles, implying that their content is similar.

3.2. Preliminary Layout

We produce the layout of our visualisation in two steps: first we create a preliminary layout, and then we apply liquid modelling to create the final layout. The preliminary layout defines approximate positions and estimated area sizes of the final visualisation. To create the preliminary layout we apply a force-directed algorithm [14]. In order to obtain reproducible visualisations, we fix the random seed. Category similarity values are used to provide additional constraints to the force-directed algorithm, acting as additional springs between similar nodes, pulling them closer to each other. Similarity values, however, are optional and our method is also able to produce visually pleasing maps even when no such attribute exists. After the layout reaches a stable state we have initial node positions for all nodes in our tree. Using the estimated size of a node which is based on its size attribute we temporarily place a disc at each node's position and then apply an overlap removal algorithm [18] to produce node positions where the discs are free of overlap. When the discs are removed the result is a preliminary layout with positions of all nodes.

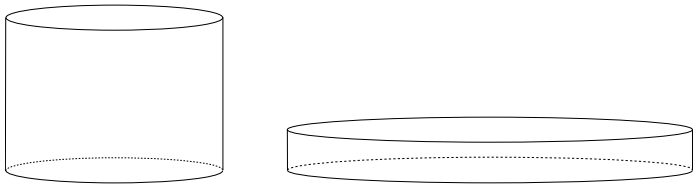
Figure 4a shows an example of the preliminary layout for a small tree of 158 nodes, showing the temporary disc of each node. The size attribute reflected by the disc's area size is the article count.

[3] http://dumps.wikimedia.org/



(a) Preliminary layout (b) Liquid modelling (c) Decorating

Figure 4: Main steps of producing the visualisation



(a) (b)

Figure 5: Flattening of a prism representing a column of liquid: (a) before flattening, (b) after flattening

3.3. Liquid Modelling

Given node positions from the preliminary layout, we next model the pouring of liquids onto these positions. Our method uses hectogons (initially regular 100-sided polygons) instead of circles to represent areas in the map. Hectogons are nearly indistinguishable from circles unless zooming in very closely, but as they are polygons they have the advantage that the position of each vertex can be separately controlled.

The column of liquid on each node forms a hectogonal prism with initial base area of 1 and a height corresponding to its volume, i.e. related to the node's size attribute. The aim of the modelling process is to flatten all prisms so that they all have the same minimal height while preserving their volume. This will result in a collection of prisms, all of whose base areas are proportional to their corresponding size attribute. This is illustrated in Figure 5, where a tall narrow prism is flattened to a short wide one.

To flatten a prism, we iteratively expand or shrink its base polygon. We also periodically detect and remove overlaps. As the volume remains unchanged, the height of a prism can be calculated from the base area.

During flattening of a prism, we first check if its height is greater than the minimum height. If so, either this prism's base area will be expanded or that of another prism will be shrunk, depending on whether this prism has enough space to expand. If the prism is already of the minimum height it will be left unchanged.

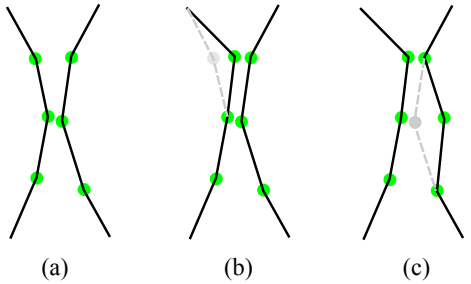


Figure 6: Expanding of a polygon: extract from two polygons with three highlighted vertices each; (a) before expanding; (b) after moving one vertex of the left polygon outward to grow that polygon; (c) after moving one vertex of the right polygon inward to shrink that polygon in order to make room for further growth of the left polygon

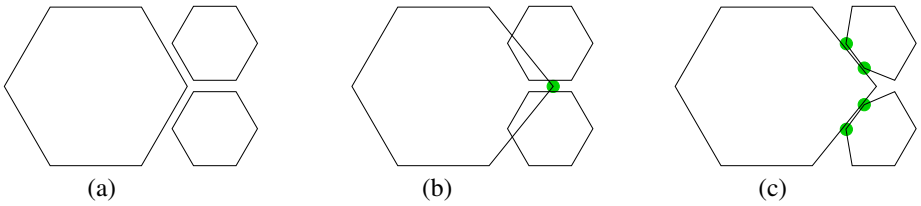


Figure 7: Overlap removal: (a) before the overlap, (b) expanding the left polygon by moving the highlighted vertex outward causes an overlap with the two smaller polygons, (c) the overlap is removed by moving the four highlighted vertices inward in the two polygons

Collision handling. The polygonal base of a prism is expanded by moving one of its vertices outward by a small distance. If there is a potential collision, e.g. if the moved vertex will fall inside the region of another prism, the polygonal base of the opposite prism will be shrunk by moving its nearest vertex inward by a small amount. The height of that prism will thus grow as its base shrinks. This is illustrated in Figure 6.

The above polygon expansion method may lead to collisions with neighbouring polygons. For performance reasons, however, we do not perform collision detection immediately. Instead we allow temporary overlaps after expanding, and periodically detect and remove overlaps after a certain number of iterations (in our case after 100 iterations).

An illustration of a collision situation is shown in Figure 7. Expanding the polygon caused two collisions. When a vertex falls inside another polygon, the overlap can be removed by moving one or more vertices in the overlapped polygon inward. This also causes an increase of the height of the affected prism, and a decrease of its base surface area.

Iteration continues until either all polygons have been fully expanded, i.e. their corresponding prism's height has reached the minimum height, or until a pre-defined maximum number of iterations has been reached (in our case we use a maximum of 100,000 iterations, but usually complete the liquid modelling in much fewer iterations, depending on the size of the source data). During one iteration, all polygons are processed, and for each polygon, all its vertices are processed. At first, almost all vertices

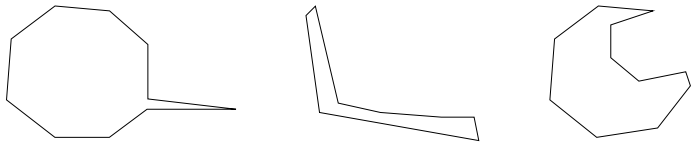


Figure 8: Examples of extremely irregular shapes to be avoided during vertex selection for polygon expansion by considering the included angle: (a) spike protruding from the polygon, (b) strip-shaped polygon, (c) polygon with a deep cavity

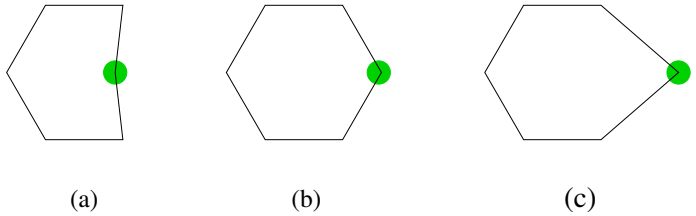


Figure 9: Examples of included angles: (a) -167 degrees, (b) 120 degrees, (c) 82 degrees; the estimation function would score angle (a) highest after getting its absolute value, i.e. 167 degrees

are moved during an iteration, but toward the end most polygons have stopped expanding and only few vertices are moved during one iteration.

Polygon expansion will choose one of the hectogon's 100 vertices to be moved outward. The choice of vertex is made by applying an estimation function to every possible move and selecting only the best choice for execution. The estimation function takes three factors into account: (1) included angle at the vertex, (2) pressure of this polygon, and (3) pressure of the opposite polygon. This is explained in more detail below.

Included angle. To avoid extremely irregular shapes, such as spikes, strips and deep cavities as illustrated in Figure 8, our estimation function considers the included angle at the vertex. The included angle is the angle between a given vertex and its two immediate neighbours on either side. The optimal included angle is near 180 degrees. An angle greater than 180 degrees leads to a concave polygon and so that vertex should be pushed outward. An angle significantly less than 180 degrees leads to long spikes or strips and thus the involved vertex should be pulled back. We achieved best results by giving vertices with larger absolute value of included angle a higher priority, as illustrated in Figure 9.

Pressure. Another factor to be considered is pressure. When there is not enough space for a polygon to expand, such as when it is surrounded on all sides by other polygons, it may push one of its neighbours. This puts pressure on the vertices that touch the neighbour, which pass the pressure along to the neighbour. Likewise, the neighbour also returns pressure. The pressure difference at a vertex is calculated by comparing the height of one prism to its neighbour. If there is no neighbour, that pressure is considered

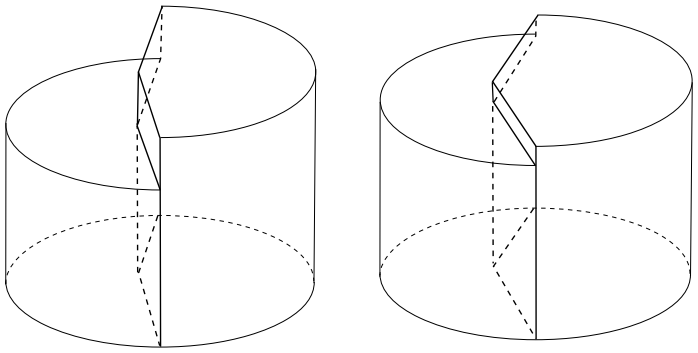
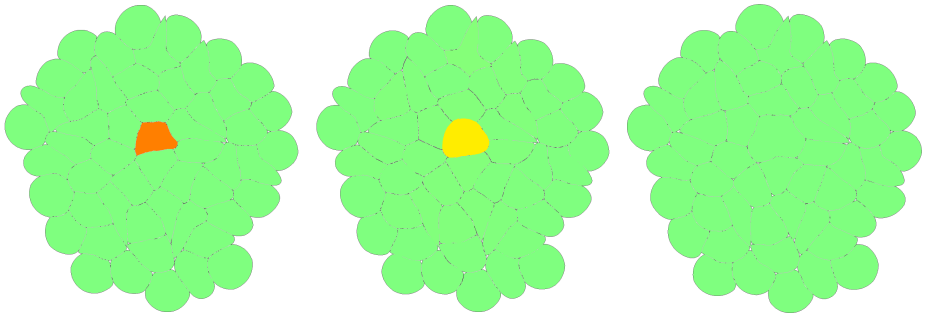


Figure 10: Two neighbouring prisms pushing against each other: (a) the right prism is significantly higher than the left prism, (b) the right prism has pushed into the area of the left prism, thereby lowering its height and increasing the height of the left prism, resulting in a decrease of the height difference between the two prisms



(a) Initial state (b) Intermediate state (c) Final state

Figure 11: Expansion of an area surrounded by several layers of other areas. After several iterations the area has expanded to reduce the height of its column of liquid to be equal to that of all other areas in the map.

to be zero. The estimation function assigns highest priority to the vertex that has the largest positive pressure difference. As a result, after several iterations the heights of neighbouring prisms will tend to become the same. Figure 10 illustrates this pushing of one prism into the area occupied by another prism. Even polygons surrounded by many layers of other polygons are eventually, after a sufficient number of iterations, able to reduce the height of their column of liquid to the minimum height, and the map thereby reaches a steady state. Figure 11 shows an example of this where one shape in the centre of the map that is surrounded by three to five layers of other shapes initially has a high column of liquid (represented by the darker fill colour), and then gradually expands, pushing its neighbours outwards, who in turn push their neighbours outwards, until this push reaches the edge of the combined shape and the expansion is complete.

By adjusting the weights of the angle and pressure factors of the estimation function, liquid modelling can produce different results. If pressure is given a higher pri-

ority, polygons will expand more aggressively, resulting in more irregular shapes, e.g. exhibiting long spikes. If angle is given a higher priority, the resulting shapes will be more regular, close to squares or circles. Through experimentation we determined the suitable weights of the estimation function are 0.75 for included angle and 0.25 for pressure. An example of the output of liquid modelling is shown in Figure 4b.

3.4. Decorating

The last step of our visualisation is to decorate the map using colours, borders and text labels, as in Figure 4c. We define a colour function to map an attribute of the data to the fill colour of areas in the map. Labels are placed near centroids of polygons. An overlap removal algorithm (FTA) [18] is applied to ensure labels at the same level in the hierarchy do not overlap with each other. As polygons of all areas do not overlap, there may be small gaps (several pixels wide) between neighbour polygons. We remove such gaps by joining neighbour vertices that are near enough to each other.

A complete map of the Simple English edition of Wikipedia is shown in Figure 12. In this map, two attributes of the underlying data are mapped to three visual attributes: (1) the number of articles in each category is mapped to the initial area size before liquid modelling; (2) the number of articles in each category is additionally mapped to the area's fill colour; (3) hierarchy relations (parent-child node) are represented through proximity and borders: all child nodes are placed surrounding a parent node, and border lines of different thickness represent the areas of different hierarchical levels.

4. Evaluation

To assess the effectiveness, visual quality and performance of our liquid-modelling visualisation method we performed four evaluations, focusing on following aspects:

1. *Usability*: We conducted a user evaluation to determine standard usability factors, such as ease of use, ease of learning, speed of task performance, etc.
2. *Error rate*: We evaluated the rate of error in accurately representing the size of each area in the map.
3. *Visual quality*: A second user evaluation determined users' perception of the visual, or aesthetic, quality of our visualisation.
4. *Performance*: We evaluated the running time of our method.

Usability was already evaluated in the earlier version of our liquid-modelling based visualisation method [3]. As the differences between the earlier version and the current version of our method are mainly internal, related to the workings of the algorithm, aspects of usability are unaffected by these differences and we present the results of our earlier usability evaluation here, but augmented with a more detailed statistical analysis.

The evaluations of usability and error rate compared our liquid-modelling visualisation method against our previous hexagon-tiling visualisation method [2]. We chose that visualisation for, of all the visualisations reviewed in Section 2, it looks most similar to our liquid-modelling visualisation.

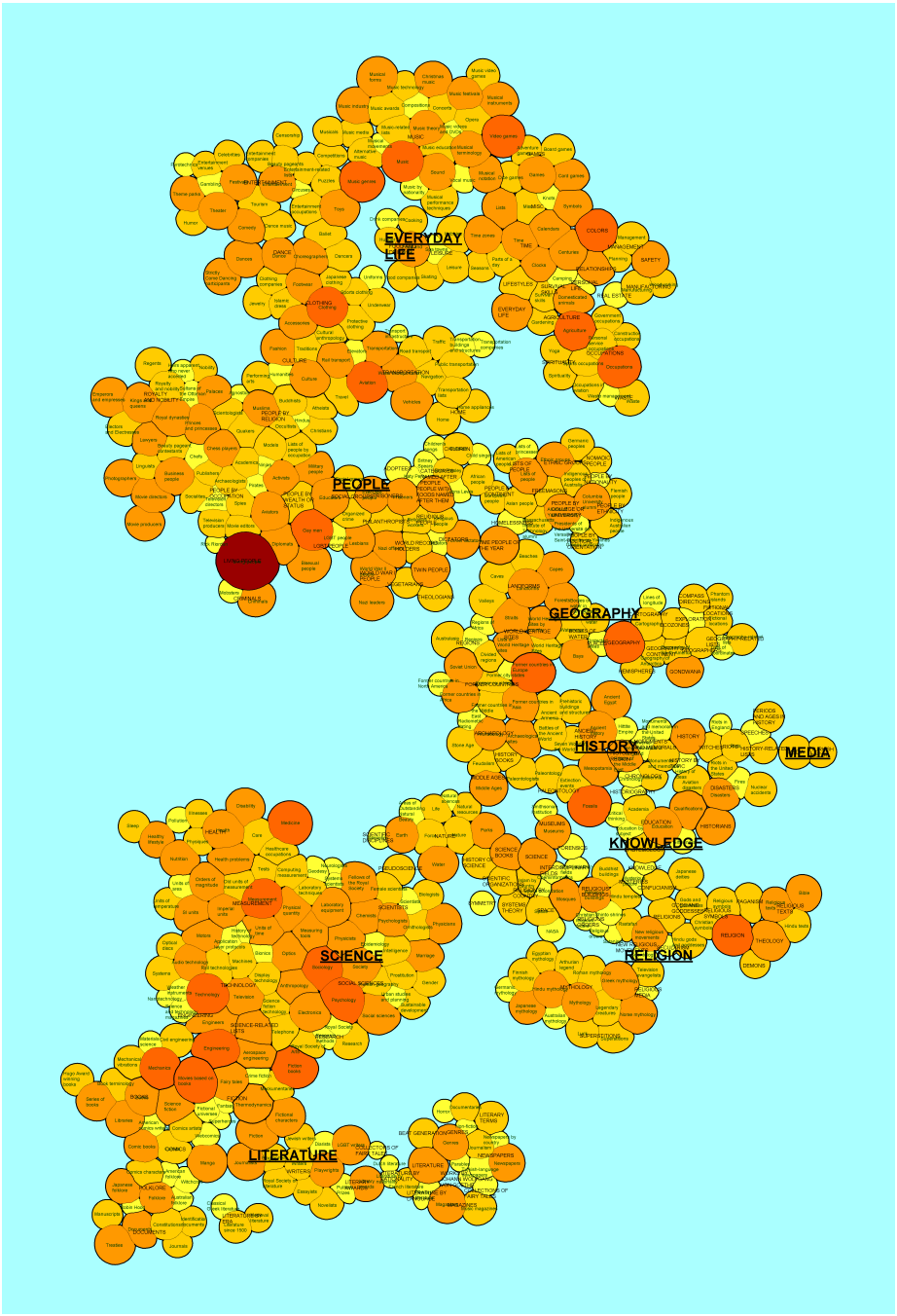
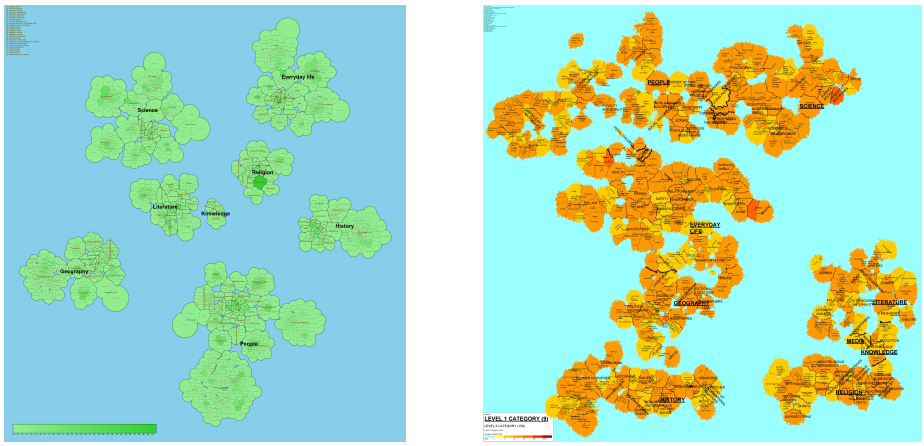


Figure 12: Visualisation of Simple English Wikipedia



(a) Map A (b) Map B

Figure 13: Visualisations used in the evaluation

4.1. Usability

We evaluated the usability of the early version of our liquid modelling-based visualisation with a controlled experiment, which is a widely used methodology to discover and test potential benefits and limitations of an information visualisation [22, 23].

4.1.1. Method

For the setup of this evaluation, we obtained visualisations of the Simple English edition of Wikipedia for both visualisation methods. We refer to our liquid modelling-based visualisation as Map A, and to the hexagon tiling-based visualisation of [2] as Map B. Samples of both maps are shown in Figure 13. We recruited 20 volunteer subjects to evaluate the usability of Map A and Map B. All subjects were undergraduate students of software engineering, aged 21-22, with 16 males and 4 females. All subjects had read articles on Wikipedia before, but none had edited any Wikipedia articles, and none had used any visualisation in the past (other than simple bar charts, pie charts, etc.).

Subjects were divided in two groups of 10 students each, and performed two rounds of evaluation, one each with Map A and Map B. The tasks in both rounds were similar, but not identical, requiring subjects to find the answers in each round. Between groups the order of visualisations presented was reversed: group 1 used Map A in the first round and Map B in the second round, whereas group 2 used Map B in the first round and Map A in the second round. This within-subject design allowed us to control for the learning effect and ordering effect that occur when the same kinds of tasks are performed twice [24].

Subjects were introduced to the visualisation presented to them and then asked to perform the following 9 tasks (this is the set of tasks used in the first round):

1. How many countries are there in the map?

Table 1: Mean accuracy values (%)

Group	**Map A**	**95% Confidence Interval**	**Map B**	**95% Confidence Interval**
Group 1	93.33	[87.78, 98.89]	80.00	[71.79, 88.21]
Group 2	90.00	[83.04, 96.96]	87.78	[79.88, 95.69]
Overall	91.67	[87.58, 95.76]	83.89	[78.43, 89.34]

Table 2: t-test result of mean difference in accuracy (%)

Mean Difference	**95% Confidence Interval**	**t**	**df**	**p-value**
7.778	[2.416, 13.140]	3.036	19	0.007

2. Which country is the biggest one?
3. Compare country Religion and History, which one has more provinces?
4. Point out the position of country Science.
5. Name any 3 provinces in country Science.
6. How many provinces are there in country People?
7. How many counties are there in province Nature?
8. Which province is the biggest one in country Science?
9. What is the level of area Technology (country, province or county)?

After each round of evaluation subjects filled an exit questionnaire giving feedback using a 7-point Likert scale (from "strongly disagree" through "neutral" to "strongly agree") on following six usability statements which are informed by [25] and [26]:

1. I enjoy using the map
2. The map is easy to use
3. The map is easy to learn
4. The information of the map is easy to understand
5. It's easy to perform tasks in the map
6. I am satisfied to perform tasks in the map

4.1.2. Results

We used the answers to assess accuracy, speed of task performance, and usability. The descriptive statistics for accuracy are summarized in Table 1, which indicate that subjects using Map A achieved a higher accuracy of answers. By looking at the 95% confidence interval of the means, we observe larger intervals for Map B subjects than for Map A in both groups, meaning a higher variance of accuracy for Map B subjects. This is not favoured as a more greatly varying accuracy implies obstacles in training and learning for general users. To sum up, we observed higher accuracy and lower variance in the trials with Map A subjects. A two-tailed paired t-test was performed to verify the accuracy differences between Map A and Map B. As shown in Table 2, subjects who used Map A had an average accuracy 7.78% higher, statistically significant at the $p < 0.01$ level. Clearly Map A helped achieve higher accuracy. Effect size (Cohen's d) was medium ($d = 0.44$).

Table 3: Mean speed of task performance (m:ss)

Group	**Map A**	**95% Confidence Interval**	**Map B**	**95% Confidence Interval**
Group 1	4:13	[3:57, 4:31]	4:01	[3:25, 4:37]
Group 2	3:20	[3:02, 3:39]	4:29	[3:52, 5:06]
Overall	3:47	[3:30, 4:05]	4:15	[3:51, 4:39]

Table 4: t-test result of mean difference in speed (seconds)

Mean Difference	**95% Confidence Interval**	**t**	**df**	**p-value**
-27.900	[-58.637, 2.837]	-1.900	19	0.073

The results for speed of task performance are shown in Table 3. The average speed of Map A was 28 seconds faster than Map B, representing an 11% higher speed. Also, the learning effect could be observed, resulting in faster task performance in the second evaluation round (Map B for group 1, Map A for group 2). The speed-up from round 1 to round 2 is particularly great in group 2 which used Map A in the second round. While we cannot conclude this finding is statistically significant (Table 4), the result still reflects that Map A facilitated faster task completion than Map B in our experiment. Effect size (Cohen's d) here was also medium ($d = 0.63$).

For the assessment of usability, we mapped responses from the Likert scale to numerical values, with 1 corresponding to "strongly disagree" and 7 to "strongly agree". We tested for statistical significance using a two-tailed paired t-test. The results for group 1, group 2 and all participants are shown in Table 5, respectively.

4.1.3. Discussion

By looking at the figures between group 1 (first using Map A, then Map B) and group 2 (first using Map B, then Map A), we conclude the impact of learning and ordering effect is eliminated by the counterbalancing design of the experiment [27]. This can be observed from the small variances across two rounds of experiments for both types of visualisations.

For group 1 (Table 5), all usability measures except "easy to perform tasks" and "easy to learn the map" show a statistically significant result at the 0.05 level (or better) while comparing the advantages of Map A over Map B. This shows that group 1 subjects perceived easier usage, higher enjoyment and greater satisfaction in trials of Map A. The mean values for both visualisations of all usability measures in this group are 5.42 and 4.50, respectively. A paired t-test shows that the difference of the mean between Map A and Map B is statistically significant at the 0.01 level, demonstrating that Map A has generally a better usability.

For group 2 (Table 5), all usability measures show a statistically significant result at the 0.05 level (or better). Among all surveyed usability items, "easy to use the map" recorded the highest difference, which means a significant perception of easier use over Map B. The mean values for both visualisations of all usability measures in this group are 5.48 and 4.63, respectively. The differences between Map A and Map B in this group are greater than the previous group, as verified by the paired t-test at the 0.001 level.

Table 5: Usability measures for Group 1, 2 and both groups combined

		Map A		**Map B**		**Diff.**	**Sign.**
Group	**Statement**	mean	median	mean	median	δ	p
Group 1	Enjoy using the map	5.10	4.5	4.30	4	0.80	0.037
	Easy to use the map	5.30	5	4.30	4.5	1.00	0.032
	Easy to learn the map	5.70	5.5	4.80	4.5	0.60	0.051
	Easy to understand the map	5.70	6	4.80	5	0.90	0.010
	Easy to perform tasks	5.20	5	4.50	4.5	0.70	0.132
	Satisfied to perform tasks	5.50	6	4.30	4	1.00	0.008
	Mean	5.42	5.3	4.50	4.4	0.83	0.045
Group 2	Enjoy using the map	5.10	5	4.20	4	0.90	0.010
	Easy to use the map	5.40	5.5	4.10	4	1.30	0.002
	Easy to learn the map	5.60	6	4.90	5	0.70	0.025
	Easy to understand the map	5.70	6	4.90	5	0.80	0.037
	Easy to perform tasks	5.40	5.5	4.70	4.5	0.70	0.045
	Satisfied to perform tasks	5.70	6	5.00	4	0.70	0.006
	Mean	5.48	5.7	4.63	4.4	0.85	0.021
Combined	Enjoy using the map	5.10	5	4.25	4	0.85	0.000
	Easy to use the map	5.35	5	4.20	4	1.15	0.000
	Easy to learn the map	5.50	5	4.85	5	0.65	0.002
	Easy to understand the map	5.70	6	4.85	5	0.85	0.001
	Easy to perform tasks	5.30	5	4.60	4.5	0.70	0.012
	Satisfied to perform tasks	5.50	6	4.68	5	0.85	0.001
	Mean	5.41	5.3	4.57	4.6	0.84	0.003

The overall differences for both groups combined (Table 5) between Map A and Map B of all six usability measures were statistically significant, at the 0.05 level for the measure "ease of performing tasks", and at the 0.005 level for the remaining five measures. This comparison demonstrates that Map A had significantly better usability than Map B. Moreover, all measures of Map A are on average between 5 ("somewhat agree") and 6 ("agree") in the Likert scale, whereas Map B is at the level of 4 ("neutral") to 5 ("somewhat agree"). This represents a stronger perception of better usability of Map A compared with Map B.

Subjects also gave qualitative feedback on both visualisations and several subjects stated that Map A is clearer than Map B. This could be because in Map A countries are kept separate from each other whereas in Map B many countries are fused together. Moreover, text labels in Map A are much less cluttered, and area boundaries are less ragged than in Map B. All of this contributes to a clearer map, better readability and greater usability.

4.2. Visual Quality

Following the above usability evaluation, we tried to improve readability of our visualisation by adjusting aspects of visual quality.

4.2.1. Method

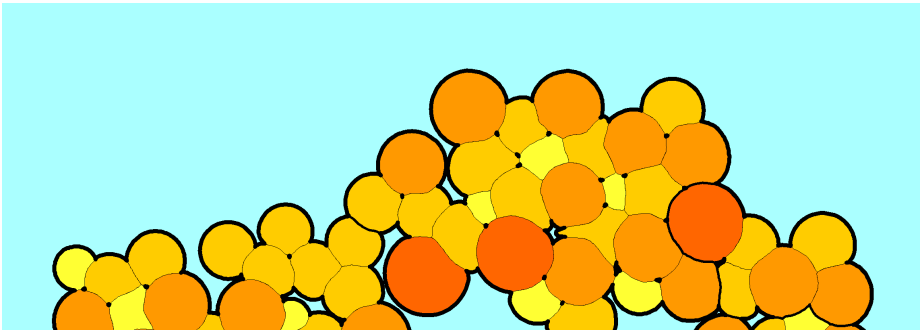
We conducted a user evaluation focusing on aspects of visual quality, and comparing several different variations of layout and decoration to determine which one is more favoured by users. To this end we controlled as many variables as possible, using the same dataset, the same pre-processing output, including the same preliminary layout positions, the same colour scheme, and the same font style and size in all visualisations. The dependent variables were the enlargement factor and border style. We recruited 26 students aged 18 to 25 (mean age 21, median age 20) with 17 females and 9 males. Among them were 21 undergraduate students and 5 master students; 5 students majored in technical subjects (computer science, engineering), whereas the remaining 21 students majored in non-technical subjects (accounting, business, economics, finance, language). Students self-reported their IT skills, with 15 stating basic IT skills, 5 claiming advanced IT skills, and 6 who reported having programming skills. On prior use of Wikipedia, 21 students had read Wikipedia articles before, 3 had edited Wikipedia articles, and 2 had never used Wikipedia at all.

We conducted two rounds of evaluation:

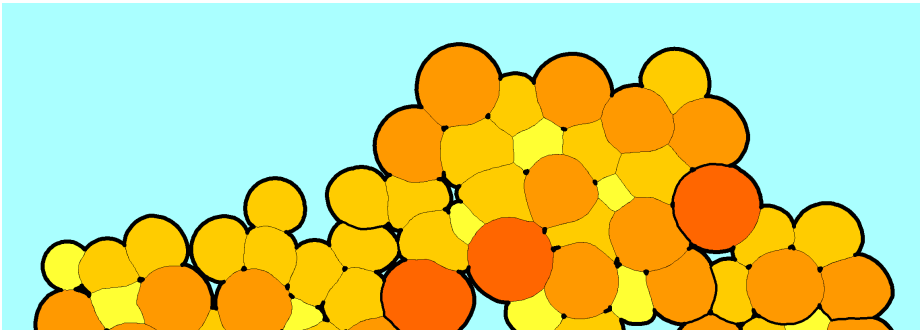
1. Comparison of different scaling factors of our liquid-modelling visualisation
2. Comparison of different border widths of our liquid-modelling visualisation

Scaling factor. Our visualisation method allows the size of areas in the final layout to be controlled by applying a *scaling factor*. This factor is multiplied by the area size to uniformly grow or shrink all areas. Controlling this factor results in a map that has a more sparse or dense appearance. Our evaluation aimed to determine the suitable scaling factor based on user feedback. Figure 14 shows extracts from the visualisations with different scaling factors used in our evaluation.

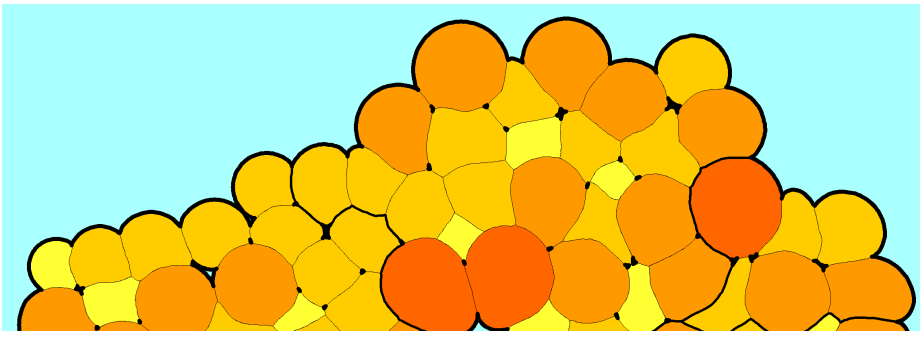
Subjects were presented with these visualisations (maps A1, A2, A3) and with following statements:



(a) Map A1: Scaling factor 0.64



(b) Map A2: Scaling factor 0.81



(c) Map A3: Scaling factor 1.0

Figure 14: Visualisations used in the evaluation of visual quality – scaling factor

1. Related to how pleasing the different versions of the map look, I rank them:
2. Related to how beautiful the different versions of the map look, I rank them:
3. Related to how clear the different versions of the map look, I rank them:
4. Related to how easy it is to perceive boundaries between areas in the different versions of the map, I rank them:
5. Related to how easy it is to understand the hierarchy of areas in the different versions of the map, I rank them:
6. Related to the compactness of the layout in the different versions of the map, I rank them:

For each statement, subjects indicated which of the three maps (A1, A2, A3) they ranked first, second or third.

Border width. We also experimented with different border widths, ranging from fine to thick borders. To determine the suitable border width, the second round of evaluation elicited user feedback. Figure 15 shows extracts from the visualisations with different border widths used in our evaluation.

Subjects were presented with these visualisations (maps A4, A5, A6) and with six statements; the first five statements were the same as for the first round of evaluation of the scaling factor above, and the sixth statement was:

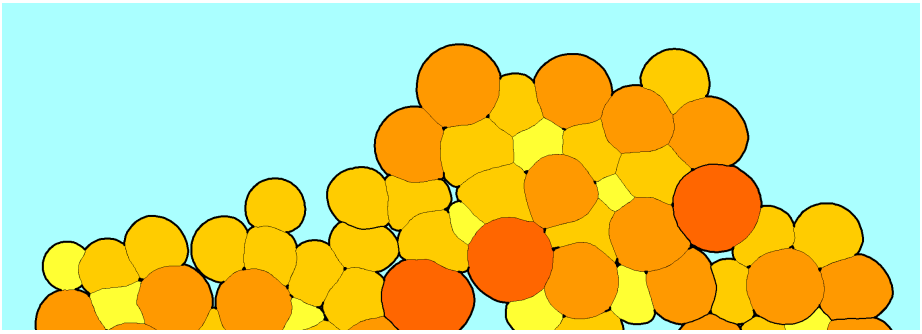
6. Related to the look of the borders between areas in the different versions of the map, I rank them:

Again, subjects indicated which of the three maps (A4, A5, A6) they ranked first, second or third.

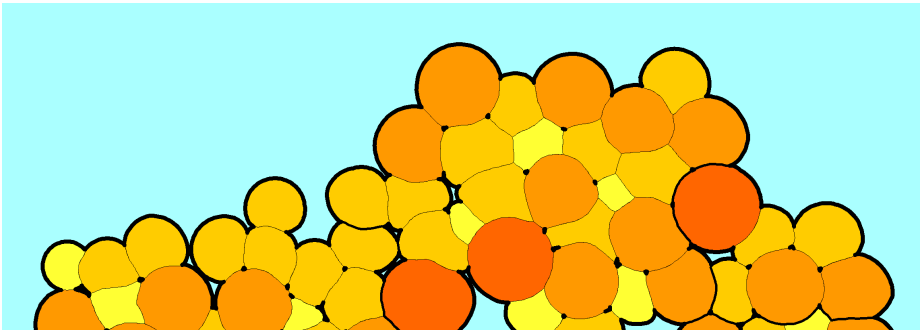
4.2.2. Results and Discussion

We received 26 valid responses for the evaluation of scaling factors. Counts of responses per map and rank are summarised in Table 6. Some responses ranked two or even all three maps at the same rank where subjects felt that they were equally preferred. The counts indicate that map A2 (with scaling factor 0.81) was considered the most pleasing and beautiful (statements 1 and 2) by the majority of subjects, followed by map A1. However, for the remaining four statements we noticed an identical pattern: map A1 had the highest count for first rank, A2 was highest for second rank, and A3 had the highest count for 3rd rank. For these four statements our subjects clearly preferred a map with more gaps between areas over a denser one. Moreover, for all six statements map A3 trailed behind the other two maps, which reflects a clear preference against map A3. This surprised us, as we considered map A3 (which is the densest one) to look most similar to a geographic map, and personally preferred it for that reason.

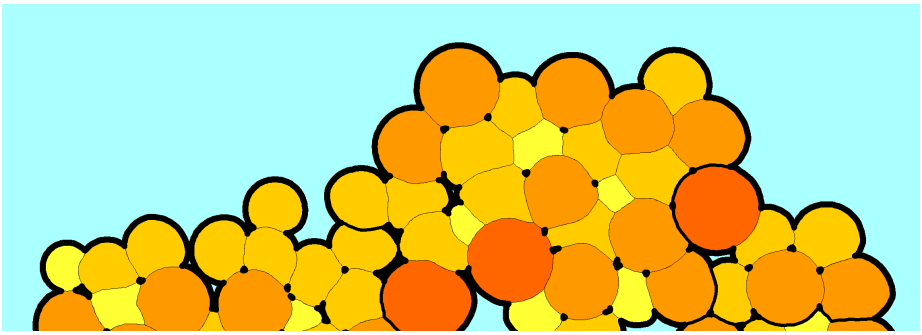
We analysed the responses statistically to test for significance. Firstly we normalised the rank responses to scores (applying min-max normalisation, converting ranks 1, 2 and 3 to scores of 2, 1 and 0, respectively). Means of these scores and their standard deviations are presented in Table 7, along with significance. For significance we ran a two-tailed paired t-test on the scores of the two highest-scoring maps (i.e. maps A1 and A2). For statements 3 (“the map looks clear”) and 4 (“it is easy to perceive boundaries”), the preference for map A1 was statistically significant, at



(a) Map A4: Thin borders



(b) Map A5: Medium borders



(c) Map A6: Thick borders

Figure 15: Visualisations used in the evaluation of visual quality – border width

Table 6: Evaluation results: count of responses for scaling factor per map and rank (highest count per rank highlighted in bold)

Statement	**Rank**	**A1**	**A2**	**A3**
1. Map looks pleasing	1st	8	**13**	5
	2nd	**12**	**12**	3
	3rd	6	1	**18**
2. Map looks beautiful	1st	9	**11**	6
	2nd	9	**15**	3
	3rd	8	0	**17**
3. Map looks clear	1st	**23**	2	1
	2nd	2	**24**	2
	3rd	1	0	**23**
4. Easy to perceive boundaries	1st	**20**	5	3
	2nd	2	**21**	4
	3rd	4	0	**19**
5. Easy to understand hierarchy	1st	**13**	7	6
	2nd	6	**17**	5
	3rd	7	2	**15**
6. Compactness of layout	1st	**13**	7	8
	2nd	5	**19**	2
	3rd	8	0	**16**

Table 7: Evaluation results: statistical analysis for scaling factor (highest mean per statement highlighted in bold)

	Map A1		**Map A2**		**Map A3**		**Sign.**
Statement	mean	stdev	mean	stdev	mean	stdev	p
1. Map looks pleasing	1.08	0.74	**1.46**	0.58	0.50	0.81	0.076
2. Map looks beautiful	1.04	0.82	**1.42**	0.50	0.58	0.86	0.076
3. Map looks clear	**1.85**	0.46	1.08	0.27	0.15	0.46	0.000
4. Easy to perceive boundaries	**1.62**	0.75	1.19	0.40	0.38	0.70	0.046
5. Easy to understand hierarchy	**1.23**	0.86	1.19	0.57	0.65	0.85	0.852
6. Compactness of layout	1.19	0.90	**1.27**	0.45	0.69	0.93	0.713

the $p < 0.001$ level for statement 3, and at the $p < 0.05$ level for statement 4. The results for the other statements, however, while pointing to a preference for map A1 (statements 3, 4, 5) or map A2 (statement 6), were not statistically significant.

For border width, we received 25 valid responses, and once again some responses ranked two or even all three maps at the same rank. Counts of responses per map and rank are summarised in Table 8. The results indicate that for a pleasing map the medium border width was preferred, but for a beautiful map the thinner borders were preferred. For each of the remaining four statements we again saw an identical pattern: map A6 had the highest count for first rank, A5 was highest for second rank, and A4 had the highest count for 3rd rank. For these four statements our subjects clearly preferred a map with thicker borders which helped to increase clarity and improve the perception of boundaries and hierarchies. The different results between these six statements thus indicate that no one border width was favoured for all purposes and that probably a medium border width would be the most acceptable compromise.

Again, we analysed the responses statistically to test for significance. Rank responses were normalised to scores in the same way as described above. Means of these scores and their standard deviations are presented in Table 9, along with significance, calculated by a two-tailed paired t-test on the scores of the two highest-scoring maps (i.e. maps A4 and A5 for statements 1 and 2, and maps A5 and A6 for statements 3–6). For border with, however, none of the preferences expressed by the voted ranks were statistically significant.

4.3. Performance

To evaluate performance of our method we tested it with maps of different size. We randomly generated input data consisting of different numbers of areas of the same size, ranging from 61 to 1258 nodes (i.e. areas) and ran 10,000 iterations on each test run. The hardware used was a PC with Intel i7-4770 CPU (3.40GHz) and 8GB RAM, running Windows 7. Our program was implemented in Java. The time complexity of our algorithm is polynomial, i.e. of the order $O(n^k)$. A plot of the performance figures is shown in Figure 16.

We designed our method for generating static visualisations, so achieving fast performance was not our primary concern. Performance tuning, deployment on a faster

Table 8: Evaluation results: count of responses for border width per map and rank (highest count per rank highlighted in bold)

Statement	**Rank**	**A4**	**A5**	**A6**
1. Map looks pleasing	1st	9	**12**	4
	2nd	4	**12**	10
	3rd	**12**	1	11
2. Map looks beautiful	1st	**11**	9	4
	2nd	6	**14**	5
	3rd	8	2	**16**
3. Map looks clear	1st	5	7	**13**
	2nd	6	**17**	5
	3rd	**14**	1	7
4. Easy to perceive boundaries	1st	2	7	**17**
	2nd	6	**17**	4
	3rd	**17**	1	4
5. Easy to understand hierarchy	1st	7	10	**14**
	2nd	3	**13**	8
	3rd	**15**	2	3
6. Look of the borders	1st	3	**10**	**10**
	2nd	5	**15**	11
	3rd	**17**	0	4

Table 9: Evaluation results: statistical analysis for border width (highest mean per statement highlighted in bold)

	Map A4		**Map A5**		**Map A6**		**Sign.**
Statement	mean	stdev	mean	stdev	mean	stdev	p
1. Map looks pleasing	0.88	0.93	**1.44**	0.58	0.72	0.74	0.055
2. Map looks beautiful	1.12	0.88	**1.28**	0.61	0.52	0.77	0.527
3. Map looks clear	0.64	0.81	**1.24**	0.52	**1.24**	0.88	1.000
4. Easy to perceive boundaries	0.40	0.65	1.24	0.52	**1.52**	0.77	0.244
5. Easy to understand hierarchy	0.68	0.90	1.32	0.63	**1.44**	0.71	0.543
6. Look of the borders	0.44	0.71	**1.40**	0.50	1.24	0.72	0.461

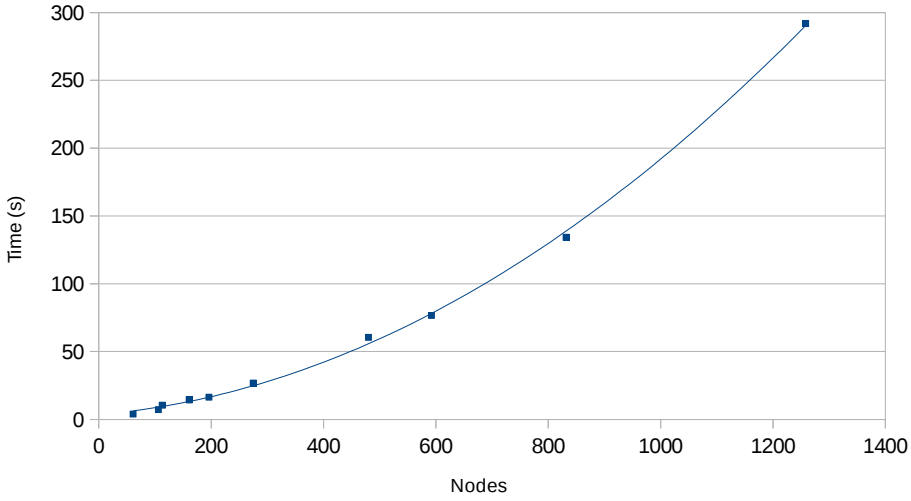


Figure 16: Performance

machine, and parallelisation of our algorithm should all be explored if real-time visualisation is desired. Another possibility for speed-up is to reduce the number of iterations run, at the expense of accuracy. By setting a low non-zero error rate, large performance improvements are possible. For example, when testing a data set consisting of 275 nodes, reducing the number of iterations by a factor of 10 to only 1,000 iterations resulted in a residual error rate of less than 1% for 16 nodes. For many applications, such an error rate is acceptable and would allow big performance gains.

4.4. Error Rate

The earlier version of our visualisation method [3] had a limitation in that it was not able to grow areas fully surrounded by other areas. In terms of liquid modelling this means that the column of liquid in that area was higher than in the surrounding areas. We have significantly revised our visualisation algorithm so that it is now able to push neighbouring areas, which means that after a sufficient number of iterations the layout reaches a steady state in which all areas have their previously calculated target size.

In order to evaluate this, we tracked the error rate, calculated as the average of 1 minus the ratio of actual area size over target size for each area. We observed that from an initially high error rate (close to 100%, meaning that almost all areas are of the wrong size), the error rate drops until after about 14,000 iterations it approaches 0%, as seen in Figure 17.

In contrast, the hexagon-tiling visualisation method of [2] does not guarantee that all areas are shown of the correct size, or even that all areas are shown at all. Using the same data as above, the hexagon-tiling visualisation method failed to display 9 of about 500 areas, and displayed 1 area in only about 36% of its target size. That is, 2% of areas are either not shown, or not shown in the correct size. Our new visualisation

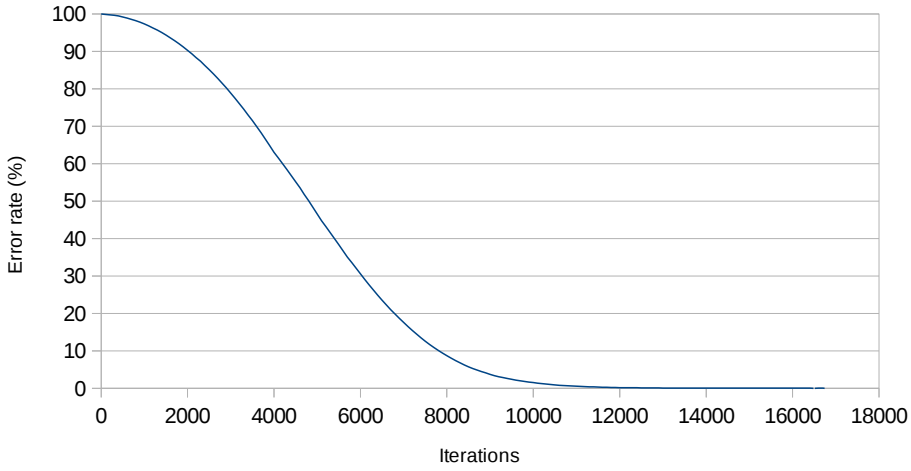


Figure 17: Error rate

method does not have this problem, as we not only show all areas, but show them in their correct target size.

5. Conclusions

Today's internet-scale applications produce "mountains of data", and making sense of this data is a challenge. New visual metaphors can help understand this data, identify clusters and perceive patterns. One of these metaphors is that of a geographic map. Our earlier research produced map-like visualisations using a hexagon-tiling approach [2], but we identified several weaknesses of that approach, leading us to devise a new approach based on expanding polygons that loosely models the expansion of a liquid poured onto a plane. This visualisation represents hierarchical data with different levels shown as areas nested inside other areas, from country to province to county. We have evaluated our visualisation comparing it with our previous visualisation method [2] and found our new visualisation to have improved usability, resulting in higher accuracy and faster speed of task performance. Moreover, it has an error rate close to zero, unlike the method of [2] which fails to display several areas. Finally, we have identified aspects of visual quality that lead to user satisfaction.

The potential of map-like visualisations in presenting data in a more intuitive and easy to understand form shows great promise for further applications, and we expect that our method will be employed beyond the domain of Wikipedia in other domains with large amounts of hierarchical data.

6. Acknowledgements

This work was supported by the University of Macau Research Committee (Grant numbers MYRG061(E1-L1)-FST11-RPB and MYRG2014-00172-FST).

References

[1] C. Ware, Information Visualization: Perception for Design, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[2] R. P. Biuk-Aghai, C.-I. Pang, Y.-W. Si, Visualizing large-scale human collaboration in Wikipedia, Future Generation Computer Systems 31 (2014) 120–133.

[3] R. P. Biuk-Aghai, W. H. Ao, A novel map-based visualization method based on liquid modelling, in: Proceedings of the 6th International Symposium on Visual Information Communication and Interaction, VINCI '13, ACM, New York, NY, USA, 2013, pp. 97–104. doi:10.1145/2493102.2493114.

[4] B. Johnson, B. Shneiderman, Tree-maps: A space-filling approach to the visualization of hierarchical information structures, in: Proceedings of the 2nd conference on Visualization '91, VIS '91, IEEE Computer Society Press, Los Alamitos, CA, USA, 1991, pp. 284–291.

[5] M. Balzer, O. Deussen, Voronoi treemaps, in: Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on, 2005, pp. 49–56. doi:10.1109/INFVIS.2005.1532128.

[6] H. Hofmann, Multivariate categorical data – mosaic plots, in: Graphics of Large Datasets, Statistics and Computing, Springer New York, 2006, pp. 105–124.

[7] G. G. Robertson, J. D. Mackinlay, S. K. Card, Cone trees: Animated 3D visualizations of hierarchical information, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '91, ACM, New York, NY, USA, 1991, pp. 189–194. doi:10.1145/108844.108883.

[8] J. Lamping, R. Rao, P. Pirolli, A focus+context technique based on hyperbolic geometry for visualizing large hierarchies, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995, pp. 401–408. doi:10.1145/223904.223956.

[9] A. Skupin, The world of geography: Visualizing a knowledge domain with cartographic means, in: Proceedings of the National Academy of Sciences, Vol. 101 (Suppl. 1), 2004, pp. 5274–5278.

[10] Y. Hu, S. Kobourov, D. Mashima, Visualizing dynamic data with maps, IEEE Transactions on Visualization and Computer Graphics 18 (9) (2012) 1424–1437. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.288.

[11] M. Gronemann, M. Jünger, Drawing clustered graphs as topographic maps, in: W. Didimo, M. Patrignani (Eds.), Graph Drawing, Vol. 7704 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 426–438.

[12] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, A. Saulnier, Gospermap: Using a gosper curve for laying out hierarchical data, IEEE Trans. Vis. Comput. Graph. 19 (11) (2013) 1820–1832.

[13] M. Blades, J. M. Blaut, Z. Darvizeh, S. Elguea, S. Sowden, D. Soni, C. Spencer, D. Stea, R. Surajpaul, D. Uttal, A cross-cultural study of young children's mapping abilities, Transactions of the Institute of British Geographers 23 (2) (1998) 269–277. doi:10.1111/j.0020-2754.1998.00269.x.

[14] T. Kamada, S. Kawai, An algorithm for drawing general undirected graphs, Information Processing Letters 31 (1) (1989) 7–15.

[15] T. M. J. Fruchterman, E. M. Reingold, Graph drawing by force-directed placement, Software: Practice and Experience 21 (11) (1991) 1129–1164. doi:10.1002/spe.4380211102.
URL http://dx.doi.org/10.1002/spe.4380211102

[16] S. G. Kobourov, Spring embedders and force directed graph drawing algorithms, CoRR abs/1201.3011.
URL http://arxiv.org/abs/1201.3011

[17] P. Eades, W. Lai, Algorithms for disjoint node images, in: Proceedings of the 15th Australasian Computer Science Conference, Hobart, Tasmania, Australia, 1992, pp. 253–265.

[18] X. Huang, W. Lai, Force-transfer: A new approach to removing overlapping nodes in graph layout, in: The Twenty-Fifth Australasian Computer Science Conference (ACSC2003), 2003, pp. 349–358.

[19] E. Gomez-Nieto, W. Casaca, L. Nonato, G. Taubin, Mixed integer optimization for layout arrangement, in: 26th SIBGRAPI - Conference on Graphics, Patterns and Images, 2013, pp. 115–122. doi:10.1109/SIBGRAPI.2013.25.

[20] K. Misue, P. Eades, W. Lai, K. Sugiyama, Layout adjustment and the mental map, Journal of Visual Languages and Computing 6 (2) (1995) 183–210. doi:http://dx.doi.org/10.1006/jvlc.1995.1010.

[21] C.-I. Pang, R. P. Biuk-Aghai, A method for category similarity calculation in wikis, in: Proceedings of the 6th International Symposium on Wikis and Open Collaboration, WikiSym '10, ACM, New York, NY, USA, 2010, pp. 19:1–19:2. doi:10.1145/1832772.1832798.

[22] C. Plaisant, The challenge of information visualization evaluation, in: Proceedings of the Working Conference on Advanced Visual Interfaces, ACM, 2004, pp. 109–116.

[23] C. Chen, M. P. Czerwinski, Empirical evaluation of information visualizations: An introduction, International Journal of Human-Computer Studies 53 (5) (2000) 631–635.

[24] I. S. MacKenzie, Human-Computer Interaction: An Empirical Research Perspective, Morgan Kaufmann, 2013.

[25] J. Preece, Y. Rogers, H. Sharp, Interaction design: Beyond human-computer interaction, John Wiley & Sons, New York, 2002.

[26] A. M. Lund, Measuring usability with the USE questionnaire, STC Usability SIG Newsletter 8 (2) (2001) 3–6.

[27] C. J. Goodwin, Research In Psychology: Methods and Design, John Wiley & Sons, 2009.