Complex Logarithmic Views for Small Details in Large Contexts

Joachim Böttger, Michael Balzer, and Oliver Deussen

Abstract—Commonly known detail in context techniques for the two-dimensional Euclidean space enlarge details and shrink their context using mapping functions that introduce geometrical compression. This makes it difficult or even impossible to recognize shapes for large differences in magnification factors. In this paper we propose to use the complex logarithm and the complex root functions to show very small details even in very large contexts. These mappings are conformal, which means they only locally rotate and scale, thus keeping shapes intact and recognizable. They allow showing details that are orders of magnitude smaller than their surroundings in combination with their context in one seamless visualization. We address the utilization of this universal technique for the interaction with complex two-dimensional data considering the exploration of large graphs and other examples.

Index Terms—Detail in context, complex logarithm, conformal mappings, analytic functions, interaction.

1 Introduction

Navigating through complex two-dimensional data is a demanding task: We would like to view small details, but at the same time the connections between these details and the general overview are important to us. While traditional explorative paradigms, such as panning and zooming, show these two aspects of information consecutively, the detail in context techniques show both simultaneously in the same image.

Geometric detail in context techniques map the information space to the image space in a non-linear way. Established techniques [3, 4, 5, 6, 8, 9], however, use mappings that distort the pieces of information in a way that makes it hard to recognize them: Parts of the resulting image are scaled differently in different directions, which means that geometrical compression is introduced. This problem worsens if the differences between scales for the details and their context grow larger. The complexity of the data that we deal with today is steadily growing, which means that it often contains tiny details in very large contexts. Quite frequently, the details are orders of magnitude smaller than the whole information space they are contained in. If we still want to visualize details and context in the same image, we have to enlarge them strongly in comparison to their context.

Compression, the scaling of two-dimensional information with different scales depending on the direction, is an obstacle for this attempt because it distorts what we see as the shape of the pieces of information. Accordingly, we need to use mapping functions that can scale parts of the information very differently, but at the same time scale them equally in all directions and keep angles locally constant. The complex logarithm, which is a conformal mapping [7], holds all three properties. In this paper, we propose to use it for the interactive exploration of complex two-dimensional data.

We further propose an interactive technique for showing the transition between complex logarithmic views and the identical mapping utilizing the complex root functions, which are another class of conformal mappings.

The paper is organized as follows: In Section 2 we clarify our understanding of distortion for the following discussion of common detail in context approaches and discuss the properties of widely used mapping functions. We then describe the properties of the analytic mappings that we use in Section 3. Section 4 addresses their use for the interaction with complex data, and Section 5 deals with issues concerning the implementation of such mappings for given triangulated geometry on modern graphics hardware. The paper concludes with the presentation of different example applications and a discussion of the method and future work.

2 Background

The need for detail in context techniques follows from the fact that details have to be shown in a certain size in order to be visually recognizable. The whole information space in the same scale does usually not fit in the available space of the screen, thus it is necessary to use a smaller scale to show the whole context.

Zooming and panning leaves shapes intact, but the context is lost when details are enlarged, and the details are too small to be recognized when the entire context is displayed. One widely used approach to show both, detail and context, simultaneously is to use separate windows for the detailed view as well as for the overview, thus breaking the connections between a detail and its context. However, the need to switch back and forth between these different views puts a mental strain on the user.

The distortion oriented detail in context techniques are used in an effort to show both, detail and context, in one seamless visualization, by mapping them non-linearly from an original space to a display space. This kind of distortion changes several properties of the depicted information. Thus, before we discuss commonly used mapping functions and our approach using conformal mappings, we will clarify what properties such mapping functions can have.

2.1 Distortions

All the mapping functions we are concerned with, map points in a plane to other points in a plane. The depicted shapes have several properties that can be distorted by a mapping: If we look at the infinitesimal small elements from which that information is pieced together, the properties that are of interest to us are size, orientation and the connections to their neighbors preceding the mappings.

We follow [7], which describes mappings as locally linear, meaning that small circles are mapped to small ellipses. The transformation then can be locally described as a concatenation of a stretch in an arbitrary direction $d1$, another stretch in the direction $d2$ perpendicular to $d1$, and a translation and rotation (Figure 1).

The resizing of the different parts of the information is the most critical property for their recognizability: If the two stretching operations are not equally strong, then a so-called compression is introduced in a mapping. This compression leads to shapes being crushed, which, for large differences in scaling factors, leads to them being mapped to almost linear structures that are not recognizable anymore.

Conformal mappings, on the other hand, are those mappings that stretch the pieces of information equally in all directions. This means that small circles are mapped to other small circles without causing distortions to local angles or introducing compression.

- *Joachim Böttger, Michael Balzer, and Oliver Deussen are with the Department of Computer and Information Science, University of Konstanz, Germany, E-mail: {boettger,balzer,deussen}@inf.uni-konstanz.de.*

Manuscript received 31 March 2006; accepted 1 August 2006; posted online 6 November 2006.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

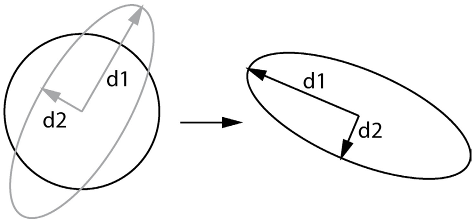

Fig. 1. A locally linear mapping can be described as concatenation of a stretch in an arbitrary direction $d1$, a stretch in the direction $d2$ perpendicular to $d1$, and a translation and rotation.

The rotation in the aforementioned transformation changes the orientation of the pieces of information. Although this makes it more difficult for a user to recognize shapes, a rotation itself leaves intact what we perceive as the shape of objects.

Another property of mapping functions is how they deal with the connections between different parts of the mapped space: If a mapping maps two infinitesimally close points to different positions, a cut between these points is introduced. If this cut separates the details from their context, a user has to mentally reconnect them, which increases the mental strain. This work shows that cuts that still leave a connection between the currently interesting detail and its context are tolerable. Given the case that strong magnifications are required, they are preferable to common approaches, in which parts of the information are compressed to such an extent that they are no longer recognizable.

2.2 Common Mapping Functions

All the common distortion oriented detail in context techniques for Euclidean spaces use mapping functions introducing compression:

Spence [9] was the first to show details and context on a computer in one seamless visualization. In his bifocal views he divided the display in three different areas, one uncompressed for the detail view, and two areas on the left and on the right side that seamlessly connected with the detail view. For the areas on the side of the display, he used the purest form of compression allowing the context to fit on the screen by shrinking the underlying information in the horizontal, but using the original scale in the vertical direction.

The Perspective Wall [6] uses a similar approach that alleviates the problem with compression. It uses a perspective mapping, thus showing information placed at a distance to be smaller, in the same way as is well known from nature and is intuitively graspable for humans. Nevertheless, perspective mapping still introduces compression with growing distances: Information that is very far away is mapped to a line, the horizon,as can be seen in Figure 2(b).

The Table Lens [8] is another approach that uses perspective mapping functions: It uses four areas connected around an undistorted detail view showing the context. In this way, it uses the display space more efficiently than the perspective wall, and allows for showing the context above and below the detail area. However, the problems with the introduction of compression in the process remain.

A different class of mapping functions are the so-called fisheye lenses. Keahey [3] used Fisheye mappings, thus keeping all the connections around a detail intact. These mappings derive from a distance function that maps every point around a detail to a new point in the same relative direction. This leads to symmetrical distortions around the detail. However, the context has to stretch all the way around the detail after the mapping. Consequently, parts of the information cannot be scaled with the same factor in all directions, and will be compressed (Figure 2(c)).

Keahey [4] also developed a technique to find a mapping function for arbitrary magnification fields by non-linear optimization. If a magnification factor for every point in a plane is given, the algorithm computes the fitting mapping by minimizing deviations from this magnification. The used measure for magnification, however, measures the area magnification of the mapping, and thus does not prevent compression of parts of the underlying data.

The Hyperbolic Tree View [5] is a conformal mapping used for the detail in context problem that does not introduce compression. However, it maps the hyperbolic plane to a disk. It is not directly applicable for Euclidean layouts of information.

In a nutshell, all the aforementioned detail in context techniques for the Euclidean plane compress parts of the information we want to show. Strong compression leads to shapes being unrecognizable after the transformation, and is inevitable with the described techniques if there are large differences in magnification factors between details and their context. The goal of this work is to map two-dimensional Euclidean data in a way that avoids compression while enlarging parts of the information in relation to other parts.

The perfect mapping that keeps all other properties intact while scaling parts of a plane differently cannot exist. Thus, choosing a mapping function for a certain problem always has to be a tradeoff between the different kinds of distortion it introduces. The point made in this paper is, that for the exploration of complex data over several orders of magnitude, keeping shapes uncompressed is more important than keeping rotations constant and even justifies giving up some of the connections between points in the plane in the process. Accordingly, we propose to cut the plane open along a line from the center of interest outward, which allows enlarging the center by extreme factors without introducing local compression in the mapping.

3 Complex Logarithmic Views

The logarithm is a function well known for application to a single axis for plots that contain numbers with different orders of magnitude. We propose to use the complex logarithm as a two-dimensional mapping function for the purpose of showing details that are orders of magnitude smaller than their surroundings in their context. The mapping behaves very similar to a stretched version of a panoramic perspective mapping, and thus allows showing all the information around a center of interest simultaneously. If we would wrap the resulting images on a cylinder, the halves of the objects intersected by the cut would be seamlessly connected again. We further propose to use the complex root functions for an intuitive transition from the identical mapping to the complex logarithmic view. It magnifies information in a similar way to a fisheye that is cut open and relaxed. In this section, our goal is to encourage an intuitive understanding for these mappings using the aforementioned connections to the well known mapping functions. We also formally introduce the two mapping functions taken from the field of complex analysis.

3.1 Fisheye Views

Common fisheye techniques map points in the plane to other points using the following mapping:

$$P_t = F + \frac{(P-F)}{|P-F|} a_t(|P-F|) \tag{1}$$

The distance function a_t changes the distance of P to the focus point F, but the relative direction α is kept the same, as can be seen in Figure 3(a). The class of root functions is widely used as distance functions.

Such fisheye mappings have to introduce compression, due to being constrained since the context around the enlarged detail has to remain closed. The magnification factors along these circles are fixed because the circumference of the circles around the center is fixed by their radius. Figuratively speaking, the information lying on these circles has to stretch in order to fill the whole circle.

3.2 Cutting a fisheye open

If we give up the constraint to keep everything connected in only one point on every circle, we can move everything along these circles, until the magnification factors in both directions are equal. Then the resulting mapping is conformal, and does not compress the information or distort angles locally.

For a given fisheye with the n-th root function as distance mapping, as shown in Figure 3(a), we can see how a regular grid of squares is

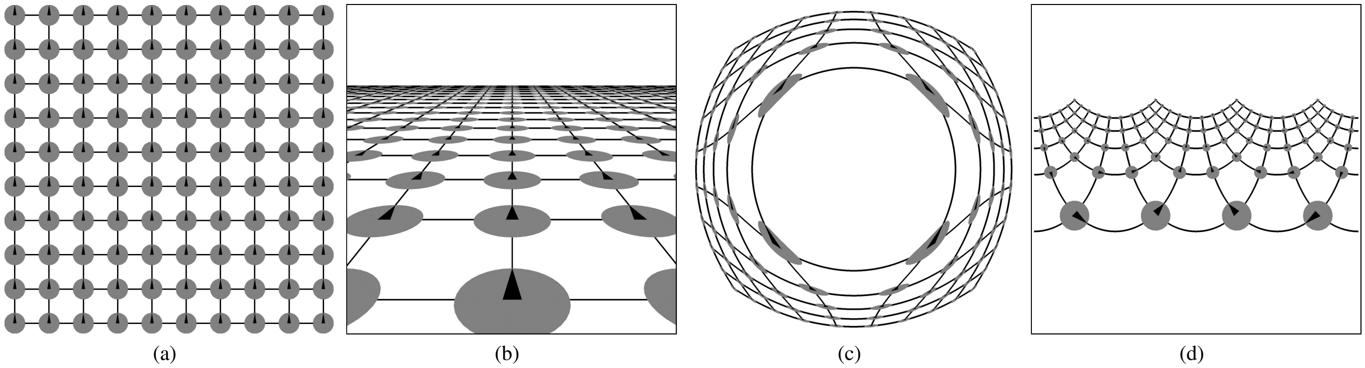

Fig. 2. Identical mapping (a) of a grid of small squares. Perspective (b) and fisheye (c) mapping both enlarge parts of the grid, but introduce compression, visible because circles are mapped to ellipses. The complex logarithmic mapping (d) enlarges parts of the grid without introducing compression.

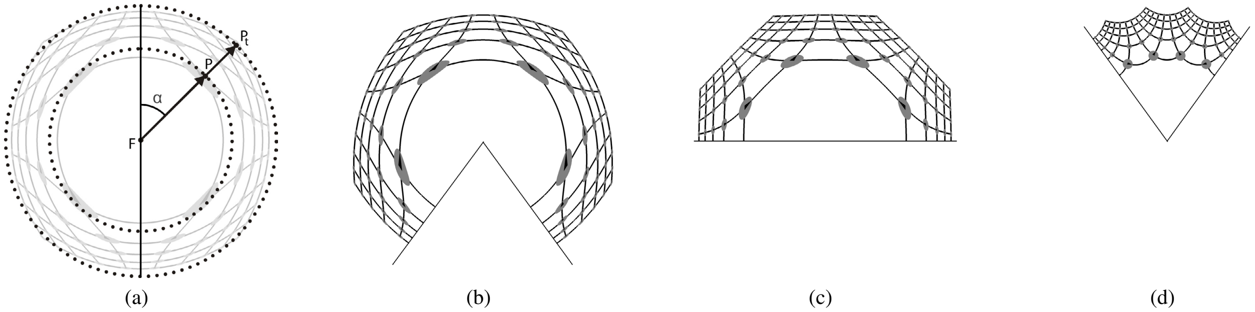

Fig. 3. Cutting a fisheye with the 5-th root function as distance function open by gradually dividing the angle by larger numbers. Once the angles are divided by 5, the resulting mapping is conformal, small circles are mapped to circles. The resulting mapping is a complex root function.

almost distorted to circles. If we move every point P_t along a circle by leaving it at the same distance from the center F, but dividing the angle α by a constant, we can observe how the stretched squares start looking like regular squares again. Once we have moved every point to another point on the same circle, but with a n-th of the angle α, we have reached conformality, and right angles remain the same after the mapping. The resulting mapping is the n-th complex root function, a function treated in the field of complex analysis.

3.3 Complex root functions

The subject matter of complex analysis are complex-valued functions: These can be interpreted as two-dimensional mappings, since complex numbers are mapped to other complex numbers. An advantageous property of the complex-valued functions we use is, that they are conformal, implying that angles are not distorted, and no compression is introduced [7].

The complex root functions map a complex number z to another complex number in the following way:

$$\sqrt[n]{z} = \sqrt[n]{|z|} \cdot e^{i\frac{\arg(z)}{n}} \quad \text{with} \quad -\pi < arg(z) < \pi \tag{2}$$

Like in the fisheye above, the n-th real root function is applied to the magnitude of a two-dimensional position. In addition, the angle is divided by the root factor n. The middle of the grid is enlarged stronger with growing n. The division of the angle leads to a cut in the grid, and to shapes that look like a slice of pie. The square root cuts the angles in half, and thus the grid only occupies half of the space. The 4th root leads to the grid occupying a quarter of the space, and so on. For larger n, the grid is mapped to a narrow wedge, and the central square takes up most of the space because it is magnified that strongly. However, the part of the mapping that shows its structure in Figure 3(d) looks very similar to the complex logarithm in Figure 2(d). In the next part of the paper we will see that we can use shifted and scaled root functions for a smooth transition between the identical and the logarithmic mapping.

3.4 Transition

To yield the transition from the identical mapping to the complex logarithm using the root functions, we use the series expansion of the exponential function:

$$e^z = \sum_{k=0}^{+\infty} \frac{z^k}{k!} \tag{3}$$

Transforming the base of the n-th root function

$$\sqrt[n]{z} = z^{\frac{1}{n}} = e^{\frac{1}{n}\log(z)} \tag{4}$$

and applying the series expansion in Equation 3, we can express the root as

$$\sqrt[n]{z} = 1 + \frac{1}{n}\log(z) + \frac{1}{n^2}\frac{(\log(z))^2}{2!} + \frac{1}{n^3}\frac{(\log(z))^3}{3!} + \dots \tag{5}$$

If we subtract 1 from both sides, and multiply by n, this yields

$$n \cdot (\sqrt[n]{z} - 1) = \log(z) + \frac{1}{n}\frac{(\log(z))^2}{2!} + \frac{1}{n^2}\frac{(\log(z))^3}{3!} + \dots \tag{6}$$

For $n \to \infty$ everything except the first summand converges to 0, and thus we yield this connection between the roots and the logarithm:

$$\lim_{n\to\infty} n \cdot (\sqrt[n]{z} - 1) = \log(z) \tag{7}$$

This shows, that appropriately scaled and shifted root mappings converge to the complex logarithmic mapping. The resulting transition is depicted in Figure 4.

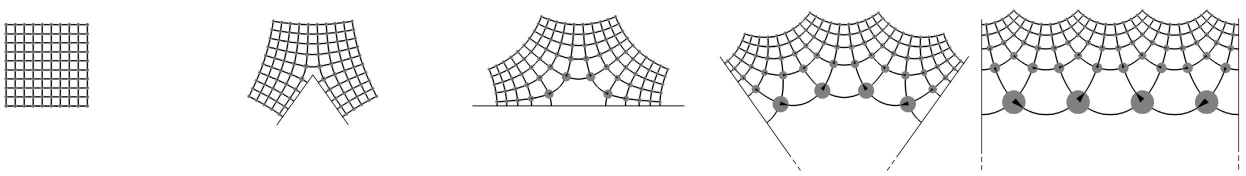

Fig. 4. Transition from the identical mapping (left) to the logarithmic mapping (right) using scaled and shifted complex root functions.

3.5 Complex Logarithm

As stated above, we want to use the complex logarithm as a two-dimensional mapping function. It maps a complex number to another complex number in the following way:

$$\log z = \ln|z| + i\arg(z) \quad \text{with} \quad -\pi < \arg(z) < \pi \qquad (8)$$

It maps the logarithm of the magnitude of a complex number to the real value and the numbers angle to the imaginary value. Thus, points on circles around the origin (with equal distance from the origin) of the complex plane are mapped to parallel, vertical lines. Rays from the origin on outwards (with equal angle) are mapped to parallel, horizontal lines (see Figure 5). The origin itself is a singularity, mapped infinitely far away in the negative real direction, and magnified infinitely.

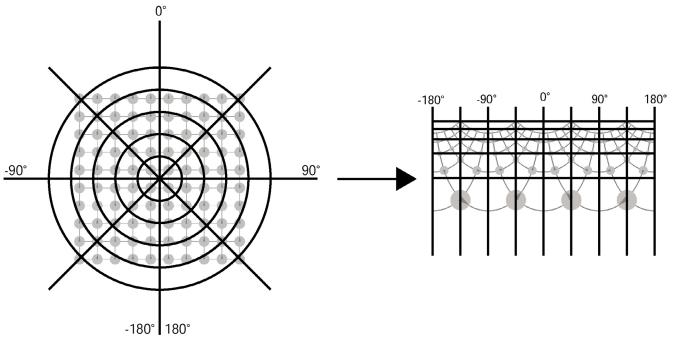

Fig. 5. This complex logarithmic mapping maps rays from the center on outward to vertical lines, and concentric circles to horizontal lines. Note the extreme difference in distances between the horizontal lines on the right and the evenly spaced circles on the left.

The magnification is indirectly proportional to the distance from the origin before the mapping. This means that objects close to the center of the plane are extremely enlarged. Like the one-dimensional logarithm, the complex logarithm maps distances that differ by a certain factor to equal distances: Thus, points that have distances from the center with the same order of magnitude are mapped into equally wide horizontal stripes. Consequently, enlarging the context by orders of magnitude only means enlarging its image by constant stripes.

The large differences between magnification factors we can show in a single image are apparent in Figure 6. While we need several ordinary images to visualize details in a complex graph, the complex logarithmic view simultaneously shows objects with sizes ranging over three orders of magnitude.

4 Interaction

Using the described techniques, it is possible to conformally map two-dimensional information in a way that enlarges one point in it extremely strong, while shrinking the others in relation. The interaction with such mappings has to answer two questions: Firstly, we show how navigation works by changing the point in the center of the magnification interactively. Secondly, we describe how we use the root functions as a transition from the identical to the complex logarithmic mapping, in order to make these mappings easier to understand and to interact with.

For the navigation through the data we allow the user to drag a point of the transformed plane with the mouse to an arbitrary other point. By inverting the actual transformation, we calculate where the starting point and the end point of the dragging operation are located in the original, untransformed space. The resulting points yield a translation vector, which moves the point under the mouse pointer to the intended new position when applied to the original data.

This mode of interaction is very intuitive, since it feels like grabbing and moving objects on the display. Pulling them towards the origin of the transformation enlarges them, and pulling them away from the origin makes them smaller. After getting used to this mode of interaction, it is possible to bring very small details in focus with one dragging operation of the mouse without releasing the mouse button (see Figure 8).

In order to aid the understanding of the complex logarithmic mapping, and enable changing between logarithmic and identical mappings fluently, we employ the root functions as a transition between the two mappings: We let the user interactively manipulate a parameter $0 \leq a \leq 2\pi$, which describes the opening angle of the wedge on which the information is mapped. An angle of 2π means, that there is no cut in the mapping, and we use the identical mapping. Between 0 and 2π, we use the complex root functions. The parameter n in Equation 2 is calculated with

$$n = \frac{2\pi}{a} \qquad (9)$$

to yield the desired opening angles. For small values of a, the wedge would grow very narrow, and only use the upper part of the screen, which, however, is against our intentions. Thus, we scale and shift the wedge in the way described in Section 3.4. The result is the transition shown in Figure 9. Once we are sufficiently close to $a = 0$, we switch to the complex logarithmic view, which is shown in Figure 11.

5 Implementation

Our implementation of the described technique takes geometry consisting of points, lines and triangles as input data. Arbitrary polygons can be rendered by subdividing them into triangles.

We can simply re-map point primitives by calculating their new positions. There are, however, two reasons for dealing with lines and triangles differently: Firstly, straight lines in the original data are mapped to curves in the transformed space. This requires to subdivide lines and triangles in order to avoid artifacts. The details, which we view very closely, have to be divided into enough sub-primitives so that after the transformation the discretization of the curves is not too distracting. For our implementation, we subdivided the geometry empirically, until the artifacts disappeared. This leads to the use of many unnecessary primitives. However, it is possible to divide the primitives adaptively by taking their magnification factors into account, and rendering more triangles where the magnification factor is higher.

Secondly, the mappings we use introduce a cut in the mapped information. It is thus necessary to separate the vertices of line segments and subdivided triangle primitives for the rendering if they are intersected by the cut: The vertices of such primitives are mapped to op-

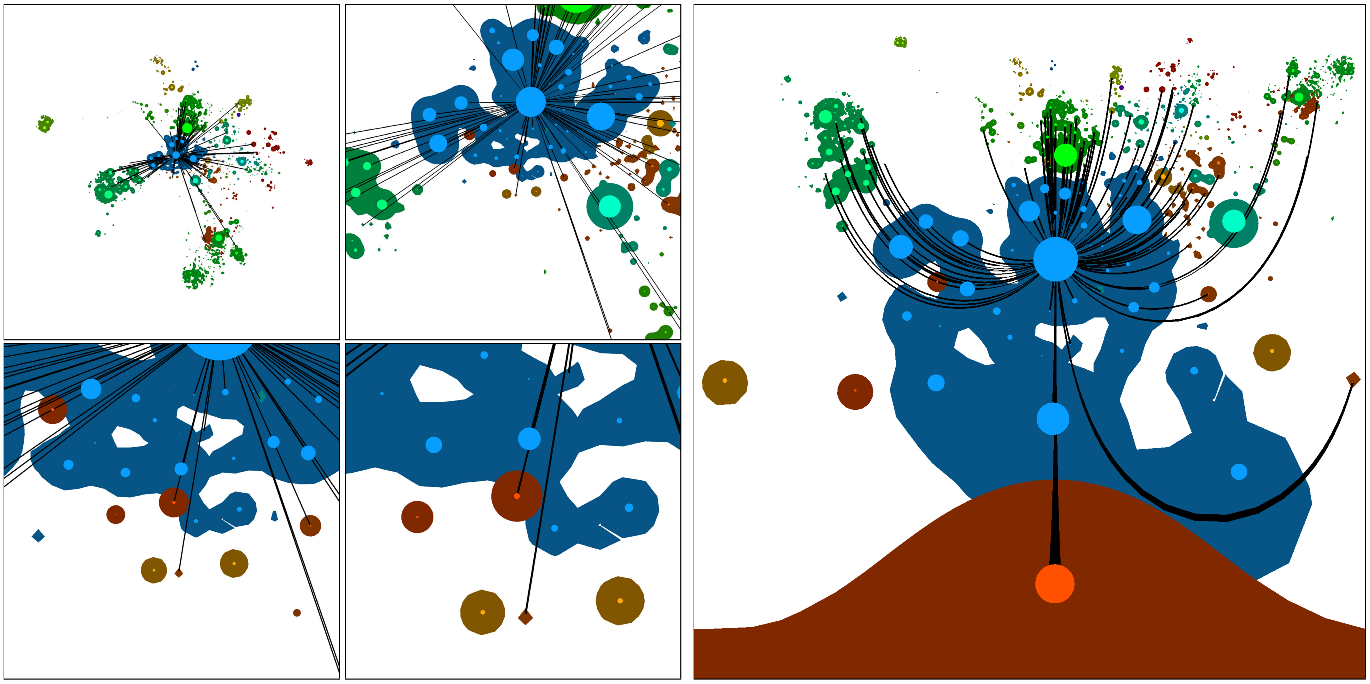

Fig. 6. On the left side: Four views with different zoom levels centered around the same point in a two-dimensional layout of a graph. On the right side: Detail and context of the same layout within one image, using a complex logarithmic view.

posite sides of the cut, and should no longer be connected in order to prevent artifacts. For line segments, the subdivision is straightforward. We calculate the intersection point with the cut, and replace the line segment with two line segments on each side of the cut.

For triangles, there are two different cases in which triangles can be intersected by the kind of cut resulting from the used mappings: In the first case, the start of the cut is located inside the triangle, making it necessary to cut open one of the sides of the triangle, and to replace the triangle with five smaller triangles. In the second case, the cut intersects two sides, which requires to disconnect the two halves of the triangle and to replace one half with two smaller triangles. The two different cases are shown in Figure 7.

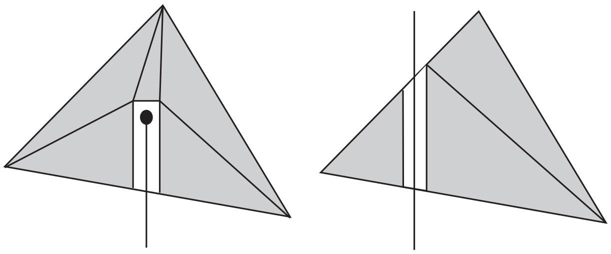

Fig. 7. The two cases of intersection between a cut and a triangle.

The intersection tests can be performed very fast with hierarchical boundings for the primitives, since the cut only intersects a few of them at any given time. After the primitive division, we employ vertex shaders to transform the geometry. We reach interactive frame rates on common graphics hardware for data sets with hundreds of thousands of primitives.

6 Applications

The first application of our proposed method is the exploration of complex graphs that visualize complex, hierarchically organized software packages [1]. The circles in the visualization depict the single software classes. They are surrounded by colored areas of recognizable shapes that visualize the containment within individual software packages. Edges between the circles represent inheritance relations between the corresponding classes. The visualization contains many details that are over a thousand times smaller than the entire context.

The original intention behind our method is to be able to explore single nodes closely, while still keeping the whole context in sight and reachable at all times. Enlarging the single nodes to a considerable size of the screen allows showing additional details for these nodes, such as source code or information about variables.

The application evidences the advantages of our approach over existing methods: The characteristic shapes that show the clustering in our graphs are well recognizable, even if they are far away from the current center of interest, because they are not compressed. This makes it possible to find known places again very quickly, even though they might be very small on the screen. Pulling these places into focus is possible with one simple drag of the mouse.

In addition, the complex logarithmic mapping yields a very intuitive display for connections from a node focused in the center of the mapping to other nodes: All rays emanating from this node are mapped to vertical parallels, and thus separated from each other. This reduces the clutter between the edges starting at a focused node that is eminent in an identical mapping of the whole information space or a common fisheye (see Figure 11).

To show that our approach is versatilely usable, we present our second application. Enlarging Voronoi Treemaps [2] shows several properties of the complex logarithmic views. Since the shapes of the single cells and the borders of the hierarchies in a Voronoi Treemap are close to circles, they are mapped to horizontal curves by the complex logarithmic view. This yields the layered structures shown in Figure 10.

A third domain for our method is geographical information. Figure 12 shows a complex logarithmic view of map data of North America centered at the Capitol in Washington. Well known features of very different sizes like the national mall, the Potomac, Florida, and the Great Lakes are easily recognizable, since they keep their shape after the mapping. In addition, the mapping does not change angles at street intersections, and thus makes it easy to follow known roads. It is also very easy to answer which places in the view are closer to the center of interest or farther away.

Our fourth example application is geographical statistical informa-

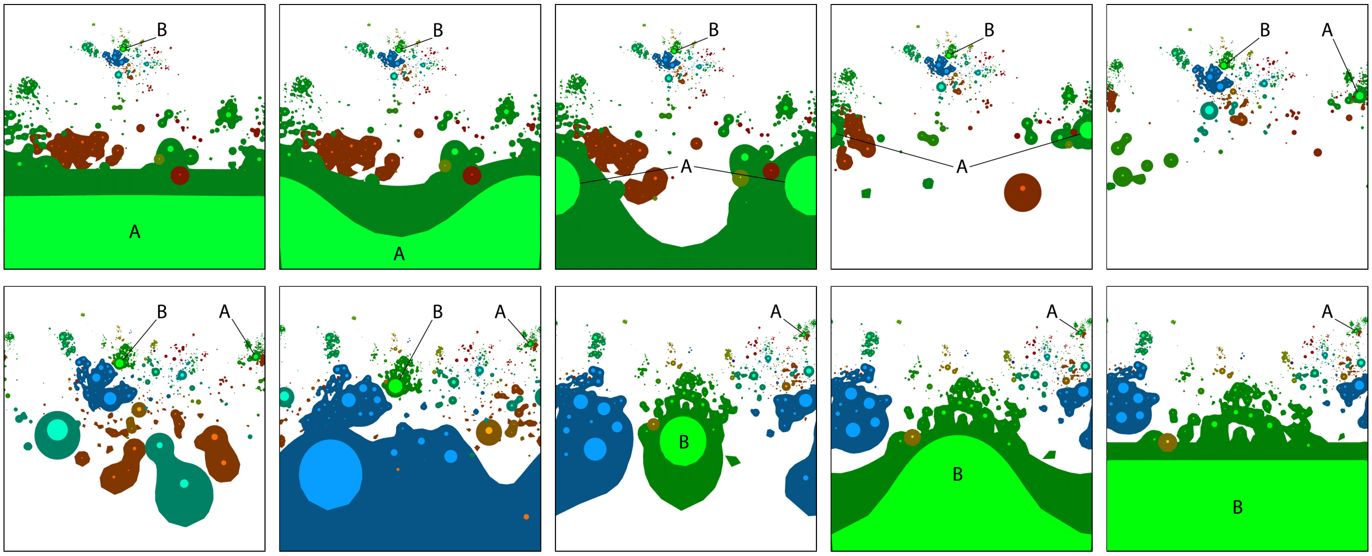

Fig. 8. Traveling from point A to another point B: The user drags B downwards into the center of interest. At first, node A is intersected by the cut in the mapping, and divided in two parts. After the first three images, the center of the magnification has left the green area. Then, after moving through the blue area between A and B, the center of the magnification enters the area around B in the image next to the last.

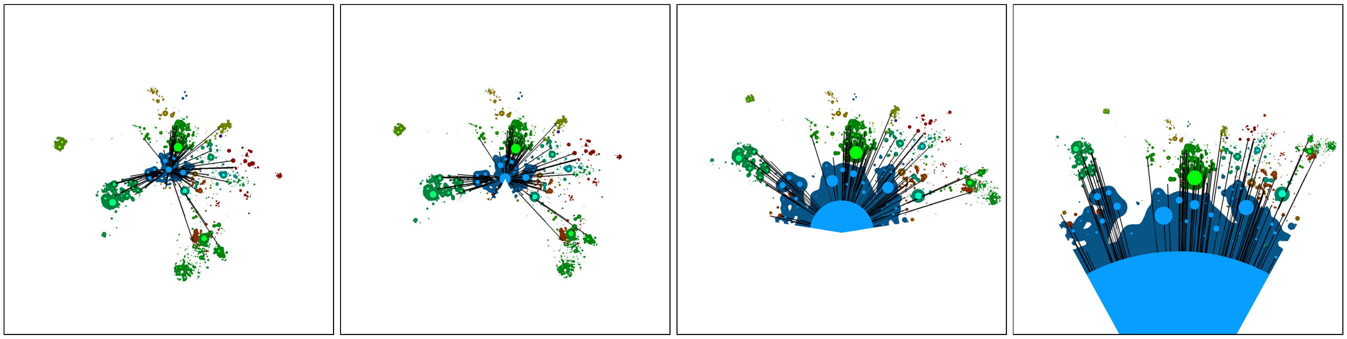

Fig. 9. Unfolding of a complex graph layout with edges. Note how the blue node in the middle is enlarged and cut open.

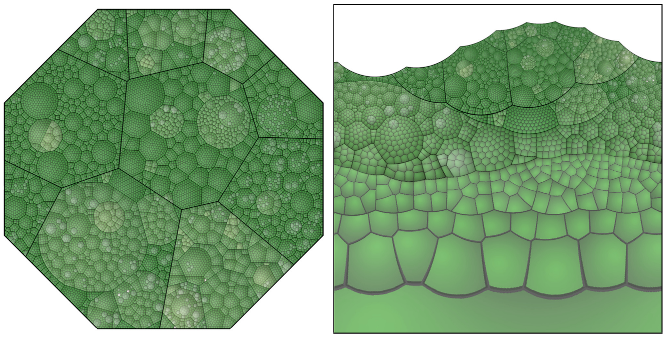

Fig. 10. Voronoi Treemap (left) and complex logarithmic view with one cell in the Voronoi Treemap enlarged (right). The borders of the different hierarchies are transformed to almost horizontal curves.

tion, like the U.S. census data. This data typically contains a lot of information in populated centers, which leads to overplotting, if we want to visualize it in the larger geographical context. Figure 13 shows a complex logarithmic view centered at a point in the central park in Manhattan. It demonstrates that it is possible to show single street blocks in the context of the whole state of New York, while not changing the shape of, for example, single counties in the countryside or long island.

7 Conclusion and Future Work

We presented a new method for the detail in context problem using the complex logarithm as a mapping function. It allows to show tiny details in very large contexts in one seamless visualization.

For an informal evaluation, we required several persons to use our interface and browse through the visualizations. Beforehand, we explained our goals, showed the visualization in the undistorted view, and explained the controls for the common zooming and panning, and the transition to the logarithmic view. After a few minutes, most users were able to move through the logarithmic views and quickly find small details they had previously seen. The strength of our method seems to lie in this task: A change from one place to another typically takes only one mouse dragging operation and is accomplished in about one second, while the same movement from one detail to another with two zooming operations, first outwards, then inwards again, takes considerably longer, since it is necessary to re-center the view between the magnification operations. The difference in speed becomes more significant with more complex data. Some users reported that it helped to imagine themselves to be at the center of the visualization, looking in all directions simultaneously, just like in a panoramic image. Then, the fact that objects become larger once you pull them towards you was reported as intuitively understandable, as well as the fact that things disappearing from the view on one side reappear on the other.

The cut in the mapping and the fact that the orientation of the mapped information is changed drastically, poses the most important obstacle for the interaction with complex logarithmic views. However,

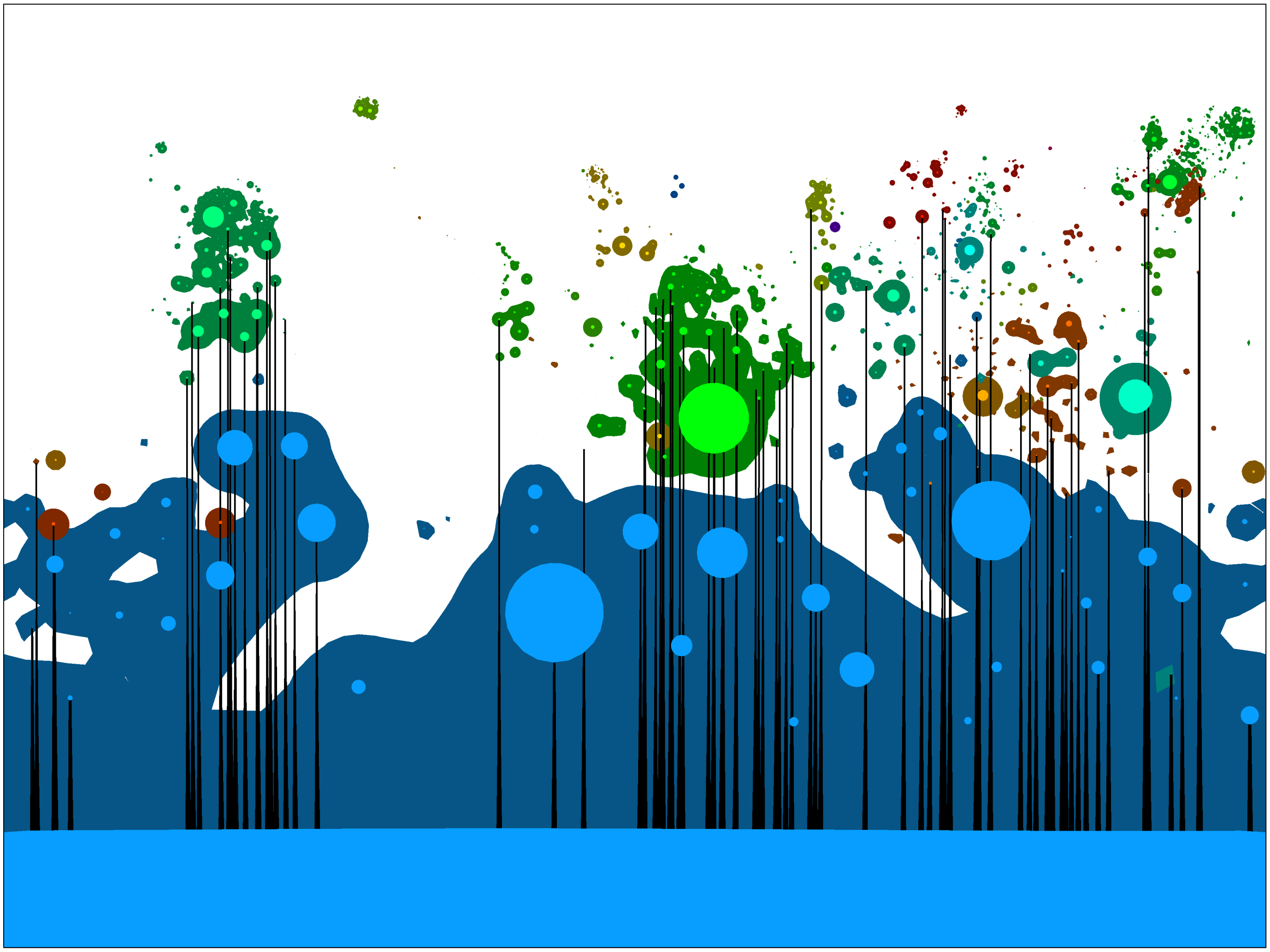

Fig. 11. The origin of the complex logarithmic mapping is inside the node. It is cut open and mapped to the bottom of the image. Edges emanating from the node to other nodes are mapped to parallel lines.

the degrees of freedom for the cut properties are currently not fully explored. The placement of the cut is variable, if we not only translate, but also rotate the information.

The method is suited for arbitrary two-dimensional information. It is our experience, that its advantages are especially evident if the information is containing recognizable, interconnected features throughout all the different orders of magnitude. The user has to be familiar with the shapes in the visualization, since the changes in scale and orientation make it difficult to recognize features that look similar solely by their relative positions. It is thus important to use the method with layouts that produce structures that are typical for the represented data items.

Questions of how labels and other cues might help the exploration, as well as the issue of how to incorporate knowledge of the hierarchical structure of the underlying data for performance and semantic zooming, will be addressed in future work.

Acknowledgements

This work was supported by the DFG Research Training Group GK-1042 “Explorative Analysis and Visualization of large Information Spaces” at the University of Konstanz, Germany. We thank Raouf Hamzaoui for helpful discussions and Anna Dowden-Williams for proof-reading.

References

[1] M. Balzer and O. Deussen. Exploring relations within software systems using treemap enhanced hierarchical graphs. In *Proceedings of the IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pages 89–94. IEEE, 2005.

[2] M. Balzer and O. Deussen. Voronoi treemaps. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 49–56. IEEE, 2005.

[3] T. Keahey and E. Robertson. Techniques for non-linear magnification transformations. In *Proceedings of IEEE Symposium on Information Visualization*, pages 38–45. IEEE, 1996.

[4] T. Keahey and E. Robertson. Nonlinear magnification fields. In *Proceedings of IEEE Symposium on Information Visualization*, pages 51–58. IEEE, 1997.

[5] J. Lamping and R. Rao. Laying out and visualizing large trees using a hyperbolic space. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 13–14. ACM, 1994.

[6] J. Mackinlay, G. Robertson, and S. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 173–176. ACM, 1991.

[7] T. Needham. *Visual Complex Analysis*. Clarendon Press, 1997.

[8] G. Robertson and J. Mackinlay. The document lens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 101–108. ACM, 1993.

[9] R. Spence and M. Apperley. Data base navigation: An office environment for the professional. In *Readings in information visualization: Using vision to think*, pages 333–340. Morgan Kaufmann, 1999.

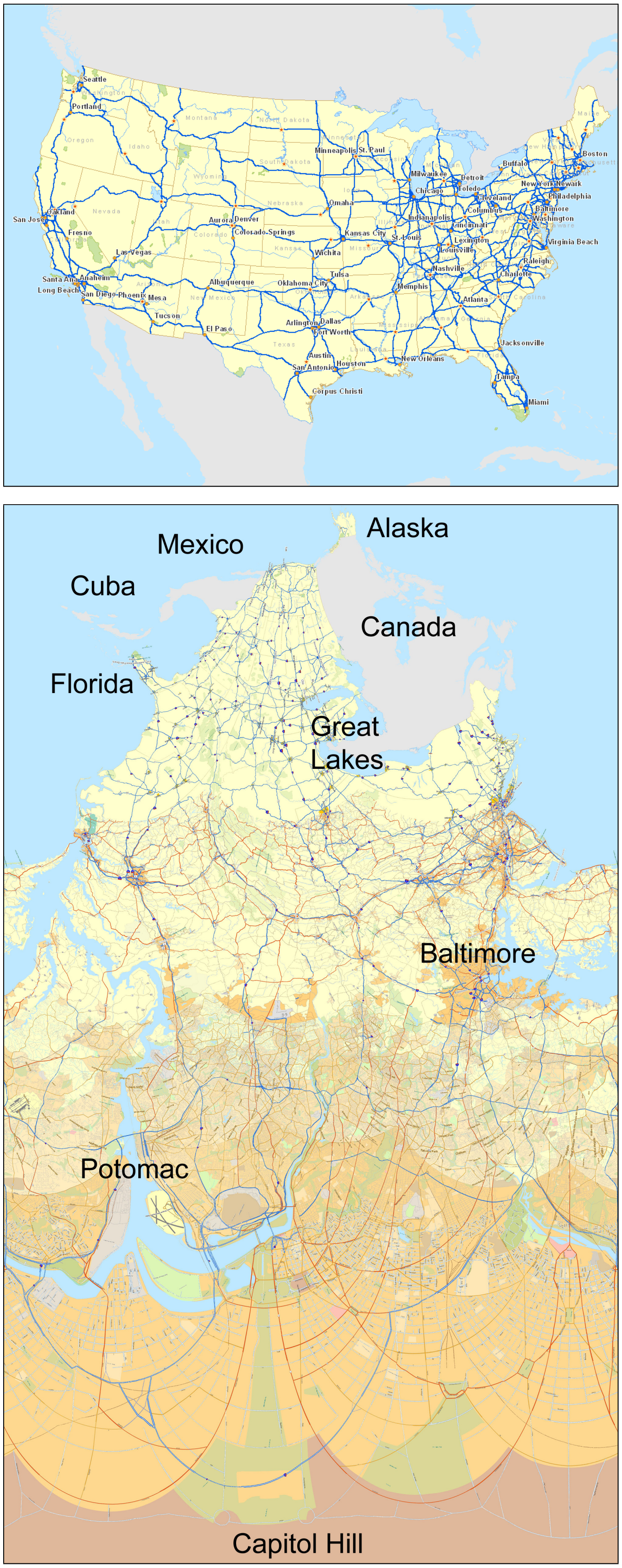

Fig. 12. Complex Logarithmic view (bottom) of map data (top). The view is centered at the Capitol in Washington. Points north-west of the capitol are mapped to a vertical line in the middle of the image. Points south-east are mapped to the very left and the very right.

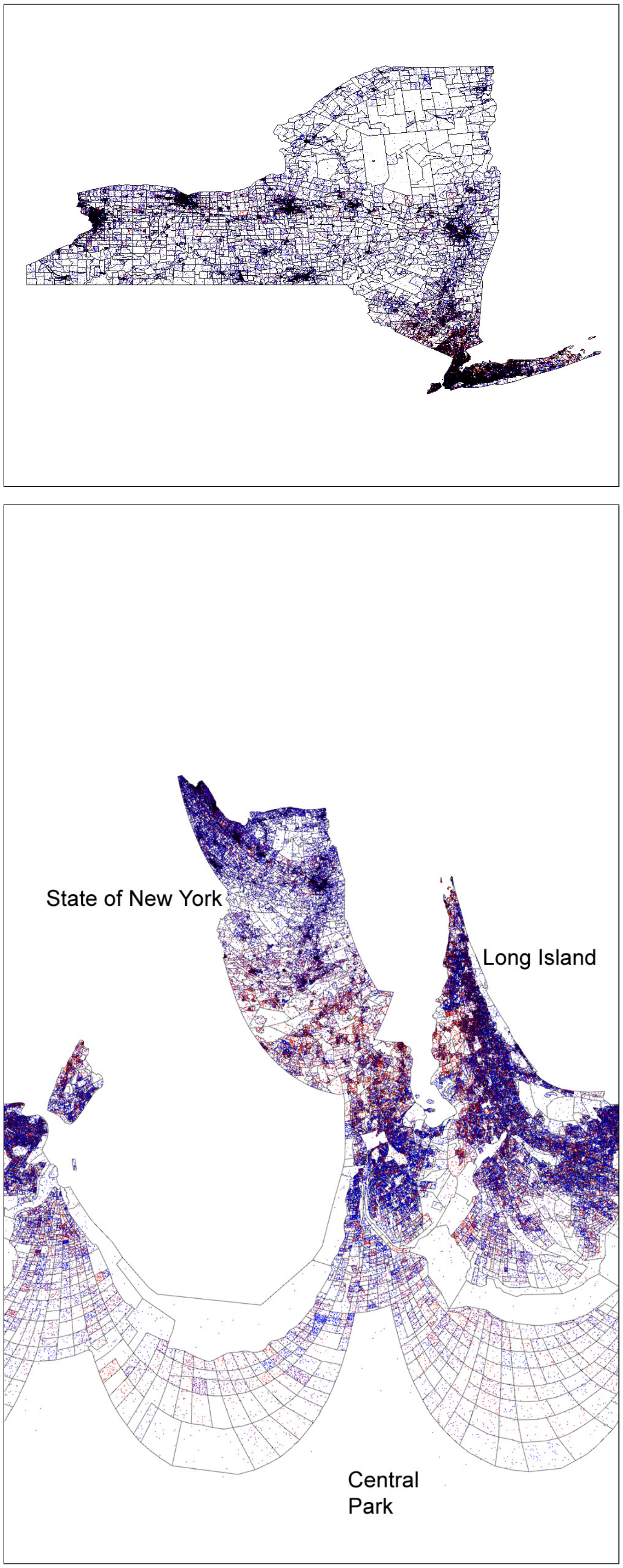

Fig. 13. Logarithmic mapping (bottom) of census data of the state of New York (top). The view is centered at the Central Park in Manhattan. The corners of the park are still right angles, since angles are locally undistorted.