Eurographics Conference on Visualization (EuroVis) 2016
E. Bertini, N. Elmqvist, and T. Wischgoll
(Guest Editors)

Short Paper

SpaceCuts: Making Room for Visualizations on Maps

Juri Buchmüller, Dominik Jäckle, Florian Stoffel, Daniel A. Keim

University of Konstanz, Germany

Figure 1: *Illustration of the principle of SpaceCuts: By cutting a map along a selected feature, for example Broadway in New York (left), the resulting map parts can be pulled apart to create additional space for visualizations. The yellow shaded area illustrates the gained space (right) and an example how it could be used for an arbitrary visualization in the upper half.*

Abstract

Visual map features like streets, rail tracks, or rivers do not provide enough space to visualize multiple attributes on them. Related approaches to solve space issues distort the map with lenses, apply distortion techniques to the map geometry, or employ three dimensional visualizations. All these techniques come at the cost of distortion or overlapping of relevant map features or they even produce overlap of visualized data. In this paper, we present SpaceCuts, a technique to generate additional space for data visualization on maps that does not distort the map and introduces only minimal overlap by cutting the map along a geographic structure and pulling the resulting areas apart. Besides introducing the basic technique, we discuss possible interactions, further extensions, application scenarios, and outline potential future research.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Computer Graphics]: Information Interfaces and Presentations—User Interfaces – Graphical user interfaces (GUI)

1. Introduction and Motivation

Visualization of data on maps is a common task. Yet, a main challenge of spatial visualization is that visual map features like streets, rail tracks, or rivers do not provide enough space to visualize multiple attributes on them. While there is usually enough room to display a single dimension, adding further dimensions quickly fills more space than the map feature can provide. Compare Figure 4, where time series of radiation measurements clearly take up more space on the map than the highway along which they were taken does.

Occluding parts of the map should be avoided, because contextual information is getting lost. But the display of that very same geographical context is a strength and a major reason to utilize geographical visualizations based on maps in the first place. Overplotting and occlusion of map features can have a negative impact on decisions users make based on the displayed information, as the user is not able to see all relevant information.

This is a general problem when putting data with exact position on map displays: for objects that have the same position, overplotting of the map and even of some of the additional data is unavoidable. Pixel placement algorithms [JHM*13, PSKN06] have been proposed

DOI: 10.2312/eurovisshort.20161163

to solve this problem by introducing a specific amount of positional error in favor of benefiting from additional space to display the information near its original location without occlusion of the data. With the growing number of data points at the exact same position, these methods show scalability issues, because the positional error is growing. This can be solved by introducing aggregation of different level-of-detail-based variants of the placement algorithm, yet leading to hidden information which we want to prevent at all cost.

Other than placement algorithms, state of the art distortion-oriented techniques follow the approach of Apperley et al. [ATS82] and allow to interactively focus specific areas of interest on a map using image-based distortion. A comprehensive review of distortion-oriented techniques has been carried out by Cockburn et al. [CKB09]. Despite recent improvements (e.g. [Gut02, PPCP12, SLQW16]), distortion impairs the ability to perform precise judgments about distance and direction [CKB09, BR03].

Another family of visualization techniques that manipulate the image space based on properties of the data to visualize are cartograms [Rai62, PSKN06, vKS07, Dor11]. These algorithms resize or distort geographical primitives so that properties of the data can be mapped on properties of the primitive. Application examples include resizing of polygons representing a state in order to reflect the number of inhabitants. This can be used to normalize visualizations of election results. The output generated by cartogram algorithms is optimizing the visual appearance with respect to a number of metrics, but the required distortions are coming with perceptual costs. In addition, resulting visualizations can look unfamiliar, although users are familiar with the displayed terrain. With SpaceCuts, we want to prevent this effect. Aforementioned techniques operate in two dimensional space. In contrast, MacEachren [Mac95] and Tominski et al. [TSAA12] use the third dimension to stack temporal information along selected routes. Yet, issues with respect to perspective and occlusion remain due to the representation in three-dimensional space.

We propose *SpaceCuts*, a novel technique to create and expand space for information visualization on maps without adding any further overplotting or occlusion. To do so, SpaceCuts expands the information space of a map along geographic entities such as streets, railways, or rivers, or in general any structure where the location is more important than its graphical appearance. The image space is cut and pulled apart along these structures which gives additional space on the map without the need of distortion or zoom. As result, our technique allows adding more information to the image space with none to minimal occlusion of the underlying map.

The remainder of this paper is organized as follows: first, we introduce the SpaceCuts algorithm in Section 2 and compare it using established quality metrics with similar approaches. Then, we introduce interaction concepts leveraging unique properties of SpaceCuts in Section 3. Afterwards, we discuss limitations and possible future research challenges in Section 4.

2. Technique

The goal of SpaceCuts is to expand the space available for data visualization on maps by cutting along map entities as illustrated in Figure 2 (left). In this section, we identify the fundamental kinds

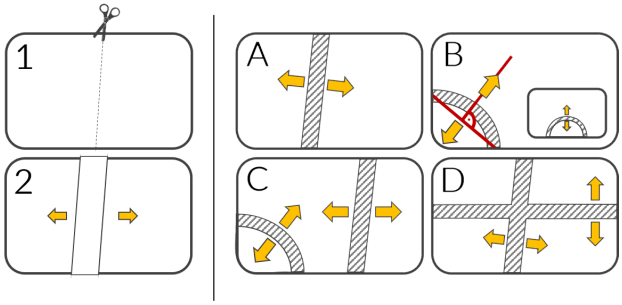

Figure 2: ***Left:*** *Illustration of the SpaceCuts principle. First, (1) the user cuts the image-space from view border to border. Then, (2) the image-space is pulled apart along the cut.* ***Right:*** *Fundamental cut scenarios: (A) Cut between opposing view borders. (B) Cut between either adjacent view borders or along the same view border. (C) Combination of cuts without intersecting. (D) Combination of intersecting cuts. The orthogonal between the end points of the cut in (B) illustrates how the pull angle is derived.*

of cuts that can be treated, how we derive the pull angle and how SpaceCuts can be created algorithmically based on the cut types.

Figure 2 on the right illustrates possible cut scenarios that are application-relevant. In case (A), a horizontal cut between two parallel view borders is done before the two resulting parts of the map are pulled apart. In the second case (B), a curved or even winded shape of the cut on one border or between adjacent borders. Here, determining the pull axis is not as straightforward as in the first case, because an optimal pulling angle that maximizes the created space has to be determined. A suitable method is to take the orthogonal of the connection between the start and end point of the cut (compare the orthogonal in red in Figure 2 (B)). We apply this strategy for all cut scenarios. The third case (C) features multiple, non-crossing cuts on the view bounds. For every individual cut, the pull angle is determined as described for case (B). Afterwards, all pulls are executed sequentially. It is important to note that the act of pulling always moves the whole image space, including any previously created cut spaces. The same applies to the fourth case (D), where the cuts to be executed cross each other. Here, we first treat one axis and then the other, resulting in two independent pulls without the need for a special crossing treatment. In general, cuts can be executed in any order, even if they cross each other.

Input: Pull Factor p, Map m
Output: Map with cuts m

```
C := findCuts(m);
foreach c in C do
    a0, a1 := cutAlong(m, c);
    α := determinePullAngle(c);
    m := pull(m, a0, a1, p, α);
end
```

Algorithm 1: Naïve SpaceCuts Algorithm.

Taking into account the findings from the case analysis, we can derive an algorithm for creation of SpaceCuts. The pseudo code in Algorithm 1 illustrates the fundamental operations of SpaceCuts.

The algorithm works in the image space of the given map and requires a pull factor that is used to determine the absolute dimension of a pull. The pull factor depends on the actual visualization needs. For example, for multiple time-series, the pull factor could be determined by the number of time-series multiplied by the desired width or height of a single time-series.

As described in Algorithm 1, the different cuts needs to be determined in the beginning. In an application, this can be done by a contour detection in the image space of the map as demonstrated by Gendron and Ioup [GI00]. If vector-based map information is available, cuts can also be determined by prominent structures like highways or railroad tracks. It is also possible that the users define the cuts interactively, which will be discussed in Section 3 in detail. After storing all desired cuts in the set C they are processed sequentially. The cut produces two areas a_0 and a_1 that still lie directly next to each other. After determining the areas a_0 and a_1 that need to be pulled apart and the pull angle α, the last step of SpaceCuts is to pull the two areas apart in image space. All pull operations need to be processed sequentially, since consecutive pull operations are required to move any previously modified area of the map as well.

Not mentioned in Algorithm 1 are application-dependent pre- and postprocessing steps. A suitable postprocessing step could be a correction of the viewport of the map after SpaceCuts has been executed to make sure the navigational context of the user is kept and all cut areas visible.

2.1. Comparison with Cartograms

To set SpaceCuts in context of similar methods, we apply the cartogram cost metrics from Heilmann et al. [HKPS04], because they have been developed for an algorithm with rectangular (read horizontal and vertical) modification of the image space, which is similar to most of the basic cases of SpaceCuts, illustrated in Figure 2. They introduce five metrics to judge the quality of their algorithm with respect to properties of a polygon before and after the cartogram algorithm has been applied. For SpaceCuts, we apply their metrics on geographic primitives as they are modified by SpaceCuts. The metrics are: 1) area error $\mathcal{A}$: area deviation of the primitive; 2) shape error $\mathcal{S}$: shape deviation; 3) topology error $\mathcal{T}$: the divergence of the neighborhood based on an adjacency graph; and 4) polygon position error $\mathcal{P}$: the positional drift of a primitive. We leave out the empty space error $\mathcal{E}$ since SpaceCuts is intentionally not restricted to vertical or horizontal operations.

When applying these metrics to a map transformed by SpaceCuts, it gets clear that they have to be applied locally along the cut areas only. Other areas are not touched by SpaceCuts, and thus, no errors or deviations are introduced. Starting with the area error $\mathcal{A}$, we see that there is no error introduced, independent of the cut. The same argumentation holds for shape errors $\mathcal{S}$ and the topology error $\mathcal{T}$.

Along each cut, we intentionally introduce positional errors $\mathcal{P}$, because this is part of the basic methodology of SpaceCuts. For each geographic primitive, the absolute positional error can be quantified as the sum of all previous cuts affecting its position. Despite the possibly large absolute positional error, the relative positional error is zero in areas that have not been cut.

SpaceCuts introduces only positional errors along cuts, which occur by design of the algorithm. This is not a problem for narrow cuts but perceptual issues arise when the cut space is getting very wide because of a large pull factor p or a large number of data visualizations to put into the cut area. Mitigation of this issue is left to concrete application cases, where also the type of visualization and used colors in the newly created, empty area on the map contribute to perceptional issues of wide cuts.

3. Interaction Concepts

Although SpaceCuts overcomes deficiencies of related approaches and can be applied to multiple trajectories simultaneously, the perceptual complexity increases with the number of cuts displayed and/or the size of the generated splits. Thus, we envision the application of SpaceCuts in combination with powerful interaction concepts to select the cuts and to expand a selected feature on demand.

As depicted in Figure 3, to select a spatial feature like a street, the user can just point at it and the system performs the SpaceCuts on the nearest available entity. The resulting space can then be used to display further data dimensions, additional time series of the same attribute or any other kind of visualization related to the entity.

Yet, in complex environments like dense street networks, it can be difficult to extract a full view-crossing cut from the available spatial entities. For example, if a street in the user's interest splits, an automatic approach can hardly decide which way to follow. To deal with this problem, a user can not only point, but in fact sketch the desired cut as depicted in the middle part of Figure 3. Automatic snapping to the closest or most important spatial entity can assist a user to make the desired cut. Another conceivable interaction especially for

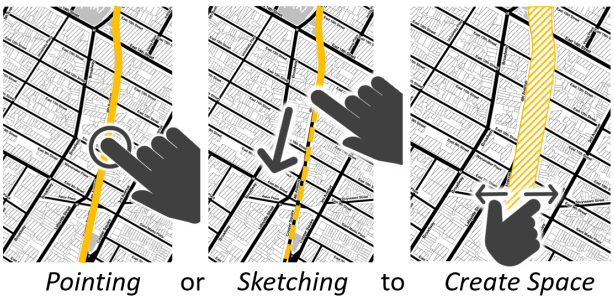

Figure 3: *Interactive creation of SpaceCuts for details-on-demand. Cuts can be invoked by pointing at a prominent map feature or in more complex situations by sketching along the map feature. The cut area can be expanded by pinching.*

mobile devices is to allow the user to define the pull factor himself by pinching the marked gap apart (Figure 3 on the right). This serves as an intuitive filter, as the user can quickly decide, how much screen space he can spare without loosing context. Context-sensitive visualizations would then adapt to the newly added space. We expect that a user-based parameterization of the cut will outperform an automatically determined cut size. Especially in mobile application cases there are factors that are hard to externalize in the use case at hand, so an algorithmic solution will fall short in those aspects.

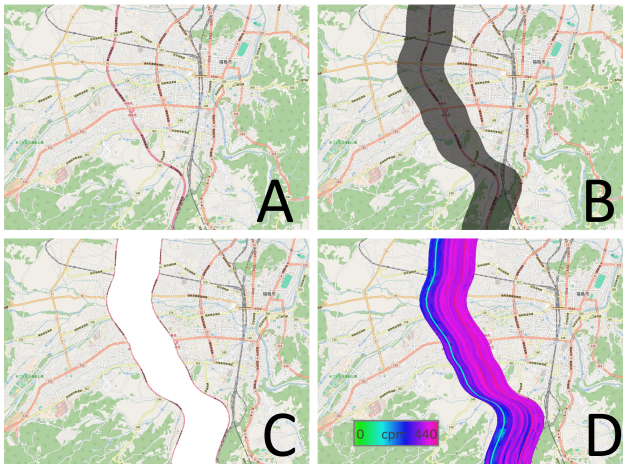

Figure 4: *SpaceCuts example showing 18 time series of radiation measurements along a highway in Fukushima, Japan. (A) depicts the original map. (B) shows the part of the map and its features that would be occluded by the visualization without SpaceCuts. (C) is the map after the cut, and (D) shows the time series visualization inserted in the additional space. The time series are ordered by time from left (May 2011) to right (June 2015), measurements are taken in counts per minute (cpm) with a linear color scale.*

4. Discussion and Future Work

In practice, we envision SpaceCuts to be applied in scenarios related to transportation networks like road or rail traffic. In this context, SpaceCuts also proves to be valuable for applications in environments with high information density where it is no option to occlude information present directly next to the cut or where large amounts of information need to be represented with minimal positional error. If there is not enough space to visualize data in place, the positional error introduced by placement techniques or the perceptual load caused by multi-scale interfaces [FB95] impairs the user to relate a value to its exact position. As the visual space SpaceCuts generates has no geographical extent, no positional error is introduced.

An example: Radiation is a local phenomenon, and values can greatly differ even over short distances, so it is important that radiation measurements are represented with as little as possible positional error. Figure 4 shows the construction of a SpaceCuts around the city of Fukushima, Japan. The data to be displayed are time series of radiation measurements on a highway taken between 2011 and 2015 after the nuclear desaster in 2011. (A) shows the original map of the city, while (B) shows the area which would be occluded by the visualization of the time series. (C) displays the map after SpaceCuts has been executed, to make room for data visualization along a street. In (D), the cut area is filled with a visualization of radiation data over time.

The example shows that simply plotting over the map occludes large parts of the city map, also creating a false impression of the radiation values' position. By applying SpaceCuts to visualize the data in the cut area, the map context is preserved while still showing the data. Compared to related techniques, SpaceCuts does not distort the map like Cartograms do, introduces no positional error as pixel placement techniques do and does not employ the perceptionally challenging third dimension as Space-Time-Cubes do. Yet, applying SpaceCuts is not only limited to the geographical context, although we use it to motivate the technique. Conceivable is also the application to on topological maps, e.g. for supply networks, or generally on node-link diagrams or even visualizations with clear structures in general, such as histograms or bar charts.

As discussed, SpaceCuts features advantages over related techniques, but limitations are evident in specific application scenarios. First, the size of the created visualization space per distance unit of a cut is dependent on the curvature of the cut relative to the pull angle. With increasing parallelity of a cut to the pull angle, the created space decreases in size, so SpaceCuts does not guarantee the same amount of space for every position on a curved cut. Second, unusual and complex cut structures, e.g. L-shaped forms, can potentially overlap or dissect the created space. Relatedly, there is no strategy for completely contained or circular structures so far. For these reasons, we propose the application of SpaceCuts on structures with limited geometric complexity. As a rule of thumb, structures with a clearly visible main axis are well suited for SpaceCuts.

Exceeding the scope of this work, there are further open research questions. First, we want to explore the perceptional impacts of SpaceCuts and compare user performance and accuracy to related techniques, deriving meaningful quality metrics and design guidelines in the process. Second, we want to find an appropriate way to handle cuts that end or are even completely contained in the viewport, although in current state workarounds like zooming or panning can be used to resolve this issue. In this context, we also want to investigate whether SpaceCuts could be combined with related techniques like cartograms or lenses with further benefits for users. Third, comparing more sophisticated technqiues to calculate the optimal pull angle is a goal to find different functions that maximize the space gain throughout the cut or that could help minimize possible perceptional issues. Fifth and last, we want to explore techniques to solve the evident overlap problems of two or more crossing cuts in the crossing area, where crossing visualizations like by Scheepens et al. [SWvdWvW12] may serve as inspiration.

5. Conclusion

We introduced a novel technique to create additional space along geometric, line-based structures on maps, called SpaceCuts, that can be used to display arbitrary visualizations without map distortion and none to minimal occlusion of other map features. We discussed the essentials for constructing SpaceCuts and illustrated the advantages of our approach in context of state of the art cartogram generation methods. As well, we gave suggestions for the interactive application of SpaceCuts and discussed possible extensions and limitations to open new application perspectives for SpaceCuts.

6. Acknowledgement

This work was supported by the German Research Foundation (DFG) within the project "Visual Spatiotemporal Pattern Analysis of Movement and Event Data" (ViaMod) and partially by the EU project "Visual Analytics for Sense-making in Criminal Intelligence Analysis" (VALCRI), grant number FP7-SEC-2013-608142.

References

[ATS82] APPERLEY M. D., TZAVARAS I., SPENCE R.: A bifocal display technique for data presentation. In *Proceedings of Eurographics* (1982), vol. 82, pp. 27–43. 2

[BR03] BAUDISCH P., ROSENHOLTZ R.: Halo: A technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2003), CHI '03, ACM, pp. 481–488. URL: http://doi.acm.org/10.1145/642611.642695, doi:10.1145/642611.642695. 2

[CKB09] COCKBURN A., KARLSON A., BEDERSON B. B.: A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv. 41*, 1 (Jan. 2009), 2:1–2:31. URL: http://doi.acm.org/10.1145/1456650.1456652, doi:10.1145/1456650.1456652. 2

[Dor11] DORLING D.: *Area Cartograms: Their Use and Creation*. John Wiley & Sons, Ltd, 2011, pp. 252–260. URL: http://dx.doi.org/10.1002/9780470979587.ch33, doi:10.1002/9780470979587.ch33. 2

[FB95] FURNAS G. W., BEDERSON B. B.: Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1995), CHI '95, ACM Press/Addison-Wesley Publishing Co., pp. 234–241. URL: http://dx.doi.org/10.1145/223904.223934, doi:10.1145/223904.223934. 4

[GI00] GENDRON M. L., IOUP J. W.: Wavelet multi-scale edge detection for extraction of geographic features to improve vector map databases. *Journal of Navigation 53* (1 2000), 79–92. URL: http://journals.cambridge.org/article_S0373463399008607, doi:null. 3

[Gut02] GUTWIN C.: Improving focus targeting in interactive fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), CHI '02, ACM, pp. 267–274. URL: http://doi.acm.org/10.1145/503376.503424, doi:10.1145/503376.503424. 2

[HKPS04] HEILMANN R., KEIM D. A., PANSE C., SIPS M.: Recmap: Rectangular map approximations. In *10th IEEE Symposium on Information Visualization (InfoVis 2004), 10-12 October 2004, Austin, TX, USA* (2004), pp. 33–40. URL: http://dx.doi.org/10.1109/INFVIS.2004.57, doi:10.1109/INFVIS.2004.57. 3

[JHM*13] JANETZKO H., HAO M. C., MITTELSTÄDT S., DAYAL U., KEIM D. A.: Enhancing scatter plots using ellipsoid pixel placement and shading. In *46th Hawaii International Conference on System Sciences, HICSS 2013, Wailea, HI, USA, January 7-10, 2013* (2013), pp. 1522–1531. URL: http://dx.doi.org/10.1109/HICSS.2013.197, doi:10.1109/HICSS.2013.197. 1

[Mac95] MACEACHREN A. M.: *How Maps Work - Representation, Visualization, and Design*. Guilford Press, 1995. URL: http://www.guilford.com/cgi-bin/cartscript.cgi?page=pr/maceachren.htm. 2

[PPCP12] PINDAT C., PIETRIGA E., CHAPUIS O., PUECH C.: JellyLens: Content-Aware Adaptive Lenses. In *UIST - 25th Symposium on User Interface Software and Technology - 2012* (Cambridge, MA, United States, Oct. 2012), Proceedings of the 25th Symposium on User Interface Software and Technology, ACM, pp. 261–270. doi:10.1145/2380116.2380150. 2

[PSKN06] PANSE C., SIPS M., KEIM D. A., NORTH S. C.: Visualization of geo-spatial point sets via global shape transformation and local pixel placement. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 749–756. URL: http://doi.ieeecomputersociety.org/10.1109/TVCG.2006.198, doi:10.1109/TVCG.2006.198. 1, 2

[Rai62] RAISZ E.: *Principles of Cartography*, vol. 315. McGraw-Hill, New York, 1962. 2

[SLQW16] SUN G., LIANG R., QU H., WU Y.: Embedding spatio-temporal information into maps by route-zooming. *IEEE Transactions on Visualization and Computer Graphics PP*, 99 (2016), 1–1. doi:10.1109/TVCG.2016.2535234. 2

[SWvdWvW12] SCHEEPENS R., WILLEMS N., VAN DE WETERING H., VAN WIJK J.: Interactive density maps for moving objects. *Computer Graphics and Applications, IEEE 32*, 1 (Jan 2012), 56–66. doi:10.1109/MCG.2011.88. 4

[TSAA12] TOMINSKI C., SCHUMANN H., ANDRIENKO G., ANDRIENKO N.: Stacking-based visualization of trajectory attribute data. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (2012), 2565–2574. 2

[vKS07] VAN KREVELD M. J., SPECKMANN B.: On rectangular cartograms. *Comput. Geom. 37*, 3 (2007), 175–187. URL: http://dx.doi.org/10.1016/j.comgeo.2006.06.002, doi:10.1016/j.comgeo.2006.06.002. 2