GPU Accelerated Chart Visualization In GIS Using Point Splatting

Extended Abstract Poster

University of Zurich[UZH]
Department of Informatics

Introduction

Interactive visualizations are a key aspect for modern geographic information systems. Domain users as well as all other groups of users have the need to interactively explore and analyse large datasets. An important aspect of geographic visualizations is to display statistical representations in a geographically referenced context such as election data or aspects of population geography. The basic tools to visualize statistical information are chart diagrams such as pie or bar charts. In this extended abstract we present a showcase for integrating different types of charts within a perspective 3D environment in real-time by using modern GPU acceleration techniques.

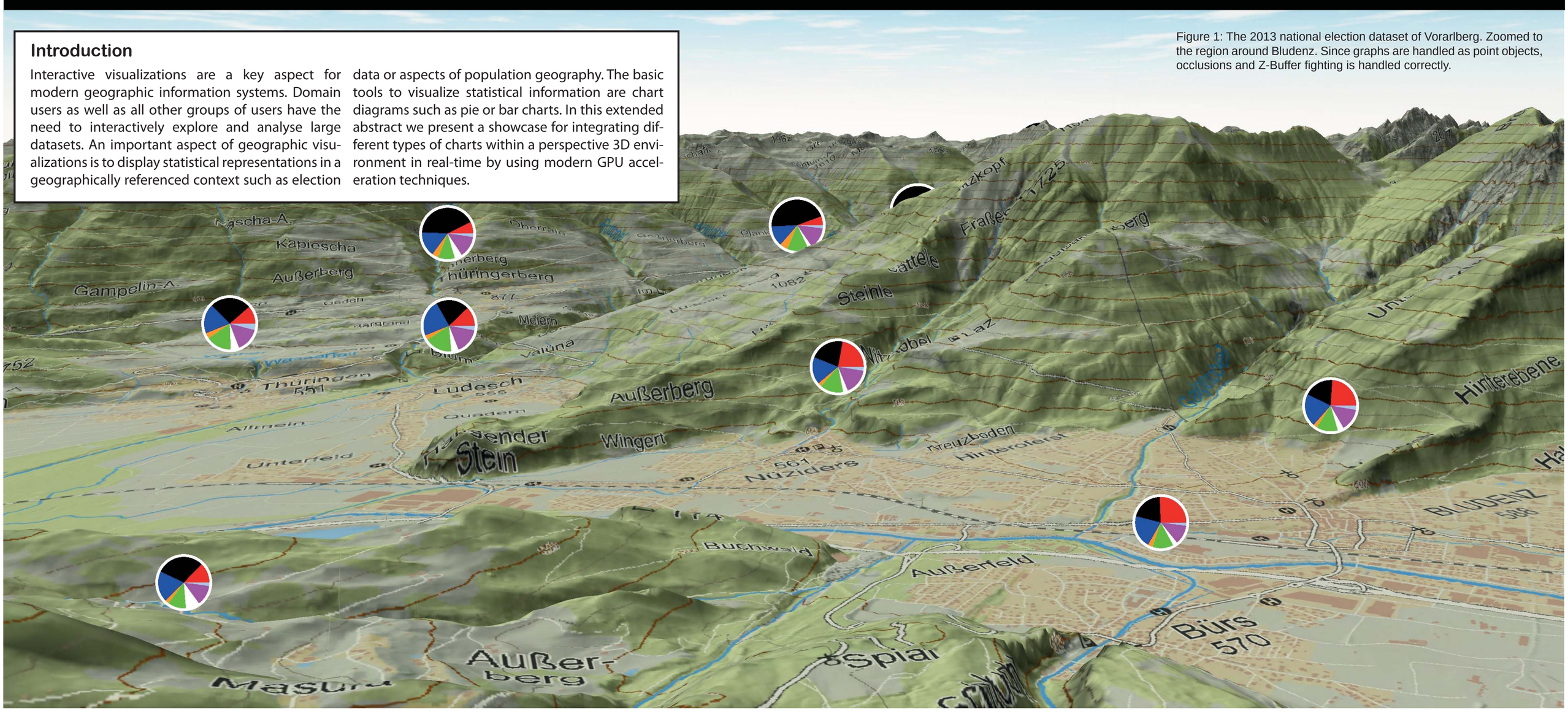


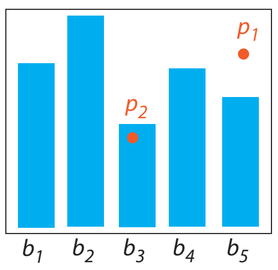
Figure 1: The 2013 national election dataset of Vorarlberg. Zoomed to the region around Bludenz. Since graphs are handled as point objects, occlusions and Z-Buffer fighting is handled correctly.

Related work

Point splatting is a well-known technique to visualize point based data sets. Point splatting can be used very efficient for point information as shown in Gross et al.(2007), Kobbelt et al. (2004) or Sainz et al.(2004). The key idea is to display points by deforming each point shape in a way that no holes are visible in the resulting image. The technique indicates, that there is no complex mesh structure generated and therefore the main advantage of point splatting is the reduced geometric complexity. This visualization technique fits perfectly the needs for large-scale point dataset, e.g. models from laser scanning.

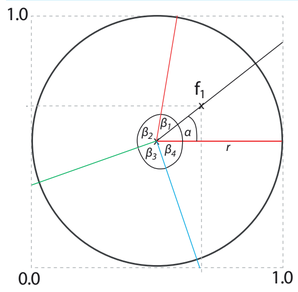
Implementation

Bar Chart



Bar charts are computed with three steps. First, all pixel need to identify their corresponding bar by a modulo operation. The next step is the comparison of height value of this pixel with the bar's height value. The pixel is discarded if it is not inside the corresponding bar (e.g. p_1 lies outside bar b_5). For all remaining pixel the color is choosen according to the bar's color (e.g. p_2 lies inside bar b_3).

Pie Chart



Pie charts are computed by identification of the angle of the current pixel (e.g pixel f_1). This angles is used to decide to which the pie chart piece this pixel belongs to. The angles of the table data values are proportionally divided to the sum of the values. The example shows that the angle of pixel f_1 to the center lies inside β_1 so the pixel will get the color of β_1

Rose Chart

Rose diagrams are a combination of pie and bar charts. In case of a rose diagrams, pie sections have the same angle size, but each slice differs in extent. This means, that the modulo-n-check is done based on the angle, so that the section or group can be identified for every fragment f_1. In addition the radius of each section is adjusted to relative values. The example above shows a red line representing the 50% border.

Result and conclusion

As shown in the resulting figures, our system can displays several thousands of pie charts at the same time even with other scene elements. Our system allows interactive changing of diagram shapes as well as selection of individual highlights within the charts. The approach of using point splats has two major advantages. First, rendering is faster, because the geometric objects are simple point locations containing attribute data and not rectangles containing textures. Second, the memory consumption on the graphic card is reduced. Figure 3 shows the US Presidential election of 2012. The dataset contains 4588 charts, one for every county in the US. This data set can be explored interactively in 3D with several hundred frames per second no noticeable overhead.

Mobile devices or other portable graphic devices can be targeted by this speed up. Most of the time, these devices are limited in graphics memory but with increasing parallelism of the graphics chips, easy parallelisable methods like point splatting will have increased importance. In the future we want to apply algorithms to automatically adjust the position of the elements, so that there is no overlap with other scene elements as well as an automatic rearranging and adjustment of the data elements based on occlusion parameters.

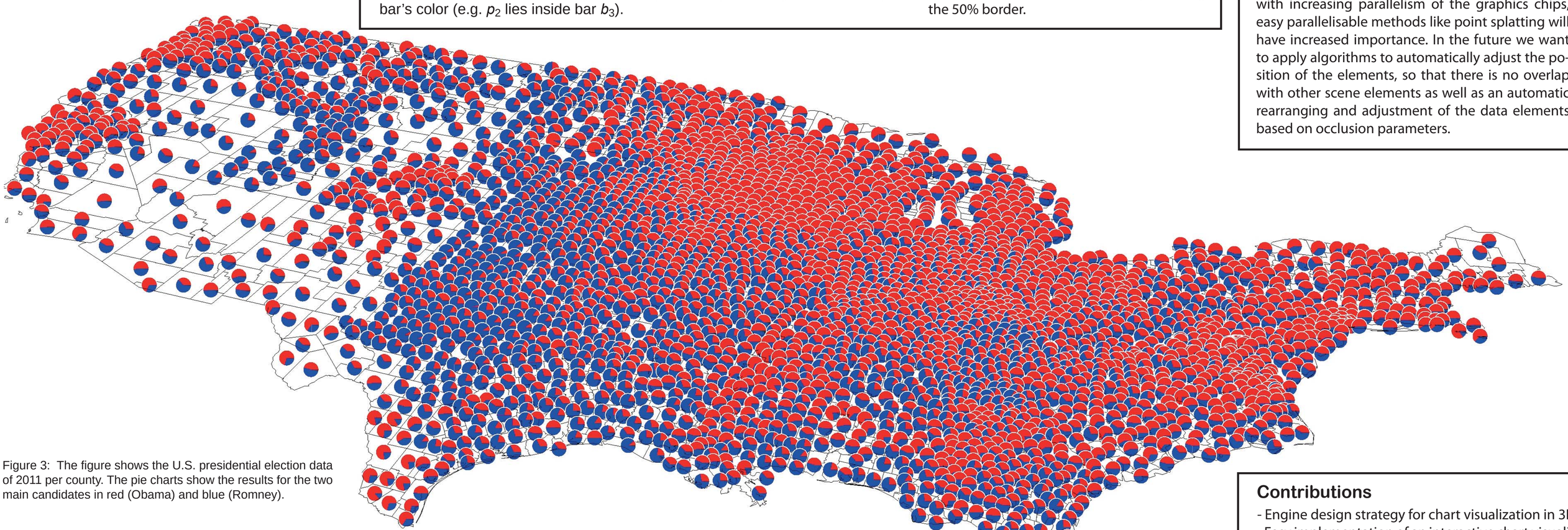


Figure 3: The figure shows the U.S. presidential election data of 2011 per county. The pie charts show the results for the two main candidates in red (Obama) and blue (Romney).

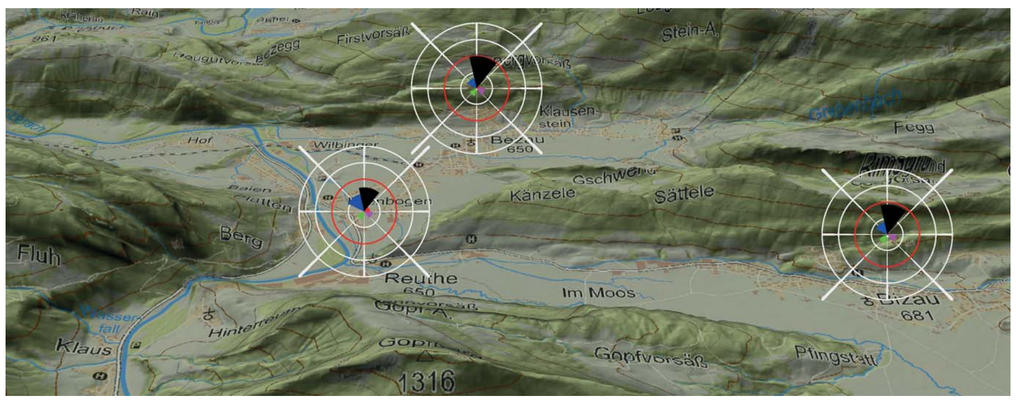


Figure 2: Screenshot of the 2013 national election dataset of Vorarlberg, showing the region of Bregenzerwald. In this example one can see that transparent diagram types like rose diagrams are also possible with point splatting.

Limitation

- Diagram type must be expressed in mathematical form.
- Number of categories (e.g. votable parties) are limited to the graphics hardware (on current devices up to 32 or 64 are possible).

Contributions

- Engine design strategy for chart visualization in 3D
- Easy implementation of an interactive chart visualization system

Acknowledgements

We would like to thank the Landesvermessungsamt Feldkirch for providing the Vorarlberg data sets as well as the Austrian National Election data of 2013 for the region Vorarlberg.

References

Gross M., Pfister H. Editors, 2007, *Point-Based Graphics, Series in Computer Graphics*, Morgan Kaufmann Publishers.
Kobbelt L., Botsch M., 2004, *A survey of point-based techniques in computer graphics*. Computers & Graphics 28(6), 801-814.
Sainz M., Pajarola R., Lario R, 2004, *Points reloaded: Point-based rendering revisited*. In Proceedings Eurographics/IEEE VGTC Symposium, 121-128.

VISUALIZATION AND MULTIMEDIA LAB

Authors: M. Thöny mthoeny@ifi.uzh.ch
R. Pajarola pajarola@acm.org
vmml.ifi.uzh.ch