MetroBuzz: Interactive 3D Visualization of Spatiotemporal Data

Wei Zeng[1] , Chen Zhong[2], Afian Anwar [3], Stefan Müller Arisona[1,2], Ian Vince McLoughlin[1]
[1]School of Computer Engineering / Nanyang Technological University, Singapore
[2]Future Cities Laboratory / Singapore-ETH Centre, ETH Zurich, Switzerland
[3]Computer Science and Artificial Intelligence Lab / Massachusetts Institute of Technology, USA

Abstract—**The rapidly increasing use of mobile devices for sensing applications, and the growing amount of simulated data generated from simulation tools such as agent-based transport simulations results in massive spatiotemporal datasets. Therefore, the demand for applications to manage and to visualize this data effectively in an interactive and easily understandable way increases quickly. Spatiotemporal visualization has long been recognized as being able to provide a suitable solution for creating such applications. However it remains a challenging task to combine such a visualization with interactive operation.**

In this paper, we describe MetroBuzz, a prototype system designed for visualizing large amounts of spatiotemporal data, and for interacting with the dataset in a way that is both interesting from a scientific viewpoint as well as for a broader non-expert audience. At its core, MetroBuzz generalizes activities in urban networks, specified as origin-destination trip information in terms of series of line segments in 3D space. These 3D elements are stored in a spatial index that allows quick retrieval of relevant data. Based on this, we implemented interactive tools to define queries on the spatial index in an intuitive manner. We show how such tools can be applied in the case of large transport simulation datasets.

I. INTRODUCTION

Spatiotemporal data was defined as "spatial data changes in a period of time" in [1]. It consists of both space and time information. Typical examples are transportation data with only a single position at a time and wireless communication networks which changes for a period within a geographical region. Visualization of spatiotemporal data has long been an important research area. It has been widely used for analysis and decision-making in diverse applications, such as migration, shopping and traveling simulation in social science [2].

A traditional way to visualize spatiotemporal data is to plot it onto a 2D map as nodes and links with a timestamp variable. However, this method will become visually cluttered and confusing as the data size increases [3]. Thus, 3D representation of spatiotemporal data which concerns changes of information in both space and time has a natural advantage of revealing overall tendencies and movement patterns [4]. It not only helps normal users understand the concept and meaning of spatiotemporal data, but also allows the researcher "to examine the complex interaction between space and time and their joint effect on the structure of human activity patterns in particular localities" [5].

Meanwhile, effective human-computer interaction (HCI), whose goal is to improve the interactions between users and computer systems and applications to make the systems or applications more usable to users' needs is as important as visualization. HCI differs from human factors, like a computer scientist may always concern how the system or application is developed while a normal end user is concerned with how to use it more efficiently. Hence, an interactive system may be considered successful if the users accomplishes what the system was designed for.

MetroBuzz is a framework for interactive 3D visualization of large quantities of spatiotemporal data in a way that is both scientifically interesting as well as understandable for a broader audience. The framework is designed to import any discrete spatiotemporal data, such as transport and pedestrian flows, food or other material flow data. The novel aspect of the work is that we not only display the large amount of data in a clear way but also provide interactive analysis tools, for example to calculate statistics in real-time to reveal relevant information. In addition, we make use of emerging mobile interaction techniques using tablets to let users interact with the framework.

We implemented an application to test the framework on simulated one day transportation data in Zurich. The dataset was generated using the MATSim (Multi-Agent Transport Simulation) software [6].[1] It is a typical spatiotemporal dataset that contains both the trips and activities like shopping and leisure of citizens.

This paper is organized as follows: we give a detailed description of MetroBuzz in Section 2. Results are presented and discussed in Section 3. Section 4 concludes and discusses possible future work.

II. METROBUZZ OVERVIEW

A. Data

The rapid increase in mobile device usage in both urban and rural areas has brought unprecedented opportunities for researchers to collect, analyze and visualize data. According to ITU World Telecommunication / ICT Indicators database, 90% of the world population and 80% of population living in rural areas have access to mobile networks at the end of 2010 and people are moving from 2G to 3G platforms rapidly [7]. Countries such as Sweden, Norway and the United States have started moving towards 4G as the data capacity of 3G is quickly being exhausted [8].

[1] http://www.matsim.org

978-1-4673-1938-6/12/$31.00 ©2012 IEEE

With the rich sensors and GPS, mobile devices used as volunteered geographical information (VGI) tools have high potential to gather, visualize, and share information on a scale never seen before. This will allow millions of contributors to create, record, and save digital information [9]. For example, Beijing is carrying out a project named "information platform of real-time citizen movement" for better traffic management by tracking millions of cell phone users using global positioning technology; ETH Zurich has a project "CitySensing" aimed at creating a human-agent simulation platform using mobile crowd-sourcing to support citizens energy-aware life [10].

Another important source for large spatiotemporal datasets are simulation and analysis toolkits such as MATSim. MATSim provides a toolbox to implement large-scale agent-based transport simulations using modules for demand-modeling, agent-based mobility-simulation and iterative simulation.

B. System Design

MetroBuzz is a open source java project[2] which makes use of open source Java Bindings for OpenGL (JOGL) library. Points and lines are used to build the base map and linked lines are used to represent trajectory. A control application is developed on Android-based tablets to let users interact with the application. A simple UDP/IP based protocol is used to send control messages from the tablet to the host computer.

C. Data Structure

The data structure used in MetroBuzz is a generalization of network-based activity data. As a basis, we used MATSim generated data to identify the relevant elements in the data structure, although real captured data could equally well be used.

Figure 1 provides an overview of the data structure.

Network:

1) node: each node is identified by a unique id and has x, y attributes which are converted from geographical location.

2) link: each link has a unique id and is defined by a start and end node. Further important link attributes are length, capacity (vehicles per hour), free-flow speed and number of lanes.

Plan:

3) agent: in MATSim, each agent refers to a citizen, identified by a unique id and additional attributes, such as sex, age, etc. In contrast, in MetroBuzz, we use the abstract concept of "agent", which does not have to be a human, but can also denote vehicles or parcels.

4) itinerary: each itinerary contains the trips and activities done by the agent throughout a given time period. Within that period, each agent only has one itinerary. Each agent should and only have one itinerary for one day.

5) trip: each trip contains multiple nodes and has distance, start and end time attributes. Hence the exact location of an agent at certain time can be retrieved.

6) activity: each activity is what the agent is doing at a given time. It contains the activity type, such as shopping and education, location, start and end time attributes. Activity can also be considered as a trip with a event type and no change in the location.

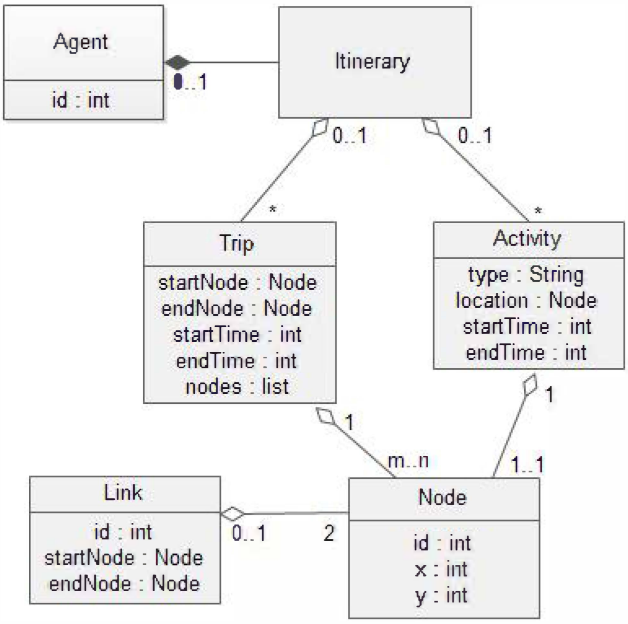

Fig. 1. MetroBuzz data structure overview

From MATSim, the network and plan files are imported into MetroBuzz. The network file stores all the nodes and links and is used to build the base map while the plan file records all the citizens' trips and activities and is used to build the space time cube.

D. Algorithms and Data Index

For the files we imported, there are about four hundred thousand agents, seven hundred thousand trips and forty million nodes. It's a challenging task to render all the data in real time. Hence efficient algorithms and data structures that help retrieve data items quickly are required. Among several spatial index mechanisms we chose to implement R-tree.

The R-tree structure was introduced by Bentley in 1975 [11]. In [12] he argued that the R-tree is a generalization of the binary search tree used for sorting and searching. The R-tree is a binary tree in which each node represents both a sub-collection of the points in the space and a partitioning of that sub-collection. Guttman [13] concludes that R-tree structures are useful for indexing spatial data objects that have non-zero size and it would be easy to add to any relational database system that supported conventional access methods. Moreover, it would work especially well in conjunction with abstract data types and abstract indices to streamline the handling of spatial data.

For users' interest, most operations like calculating how many citizens are shopping in some area during some time or

[2] http://code.google.com/p/metrobuzz/

how many citizens are coming from one area to another area are done on the trips and activities level hence for each trip we build a axis-aligned minimum bounding box for it and add the bounding box in the R-tree structure. After initially adding all the trips and activities, most operations are searching and so the pseudocode below represents the search in R-tree:

```
Algorithm II.1: SEARCH($T, S$)

 comment: search a R-tree T for a search key S
 if T is not a leaf
   then { for each $E \in \mathcal{T}$
            do if E overlaps S
            do search(E, S)
   else T is a leaf
   then { for each $E \in \mathcal{T}$
            do if E overlaps S
            then E is a suitable entry
```

A cylinder is used to represent areas of interest. A preliminary bounding box test does not guarantee the intersection of the trip and cylinder, however reduces the number of trips required for the more complex line-cylinder intersection: Fig. 2 illustrates the intersection test between a trip starting at (x0, y0, t0) ending at (x2, y2, t2) and cylinder with center (xx, yy, ts), radius d and height te-ts. The idea is to project the trip and cylinder to the x-y plane first and check if they intersect there. This is fast since it's only a line and circle intersection test. After that, the intersection point's x-y position will be projected to the trip to find the exact point in 3D space and then time variable is checked if it's between t0 and t2.

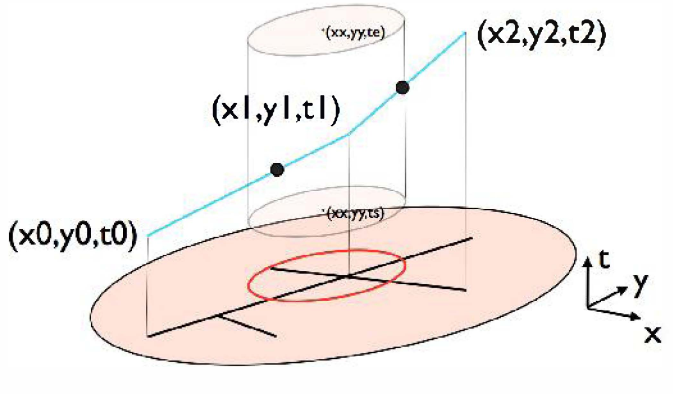

Fig. 2. Intersection test between a trip from (x0, y0, t0) to (x2, y2, t2) and cylinder with center (xx, yy, ts), radius d and height te-ts

E. Data Visualization

Fig. 3 shows the main display of MetroBuzz. Points are used to represent nodes and lines connecting nodes represents links and trips. The base map is the larger Zurich area road network, generated with information in the network file. On the vertical axis, the lines indicate the citizens' one day itineraries starting from 00:00 and ending at 24:00. The two cylinders are the user's areas of interest and the trips inside them are highlighted with brighter color. Users can move them around using the keyboard shortcuts or the specific application we developed for Android tablets (see below). The red and yellow lines are the trips starting from one selected area and ending in the other area and vice versa so users can get a direct impression of the traffic situation between two areas.

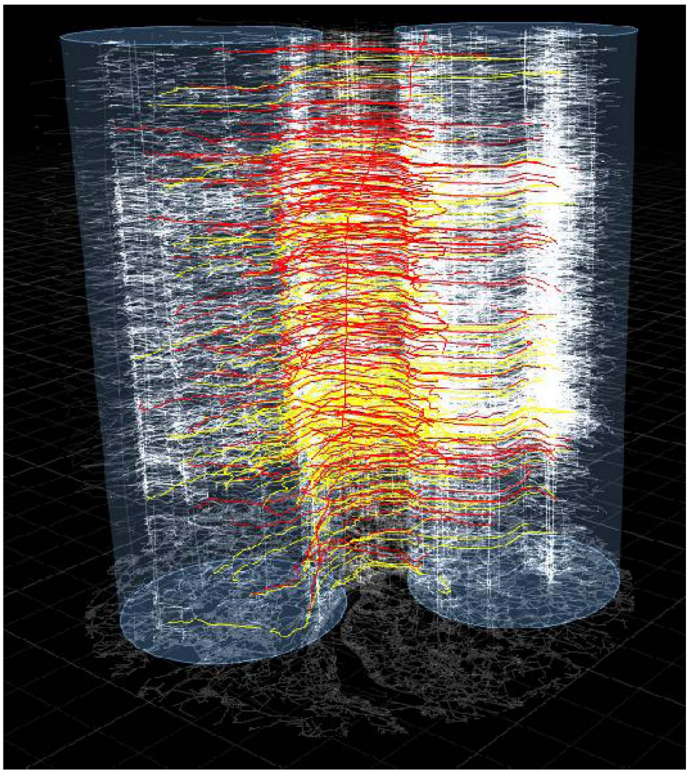

Fig. 3. MetroBuzz display overview: a base map with points and lines, white trajectory indicating all trips, two cylinders representing the selected area and yellow and red trajectory representing connection of the two areas

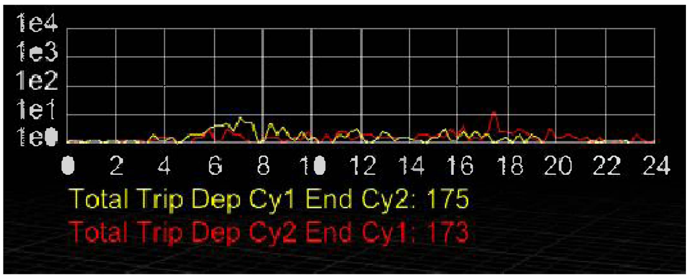

Fig. 4. MetroBuzz statistic showing the exact number of citizen traveling between two areas at a time and total number in one day.

Fig. 4 shows the statistic of the trips connecting the two areas. This figure shows the exact number of citizens traveling between the two areas at certain times, and hence helps citizens to plan their travel. Initially the statistic is between 00:00 and 24:00 but it's implemented to let users determine the time period either using keyboard or the application on the tablet.

F. Tablet Interaction

We developed an application on an Android 3.1 tablet to let users control the MetroBuzz. The source code is also available at Google Code so other developers can benefit and contribute.[3]

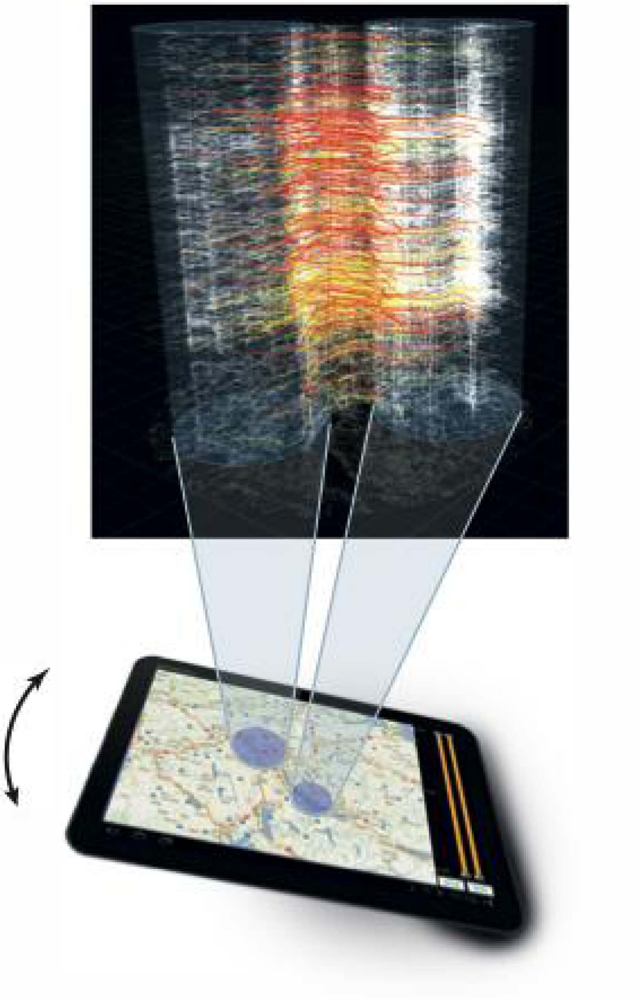

Fig. 5. MetroBuzz interface on an Android tablet with a Google map of Zurich, two circles representing the two cylinders in the host application and two sliders to control the time periods

As Fig. 5 shows, two circles on the base maps are used to control the two cylinders in MetroBuzz correspondingly. The base map is a pre-cached Google map base layer of Zurich. Users can zoom in/out both the circles as well as the map as desired. On the right side there are two sliders which are used to control the time period corresponding to the height of the two cylinder. Finally, we added two more buttons - setup and reset. The setup button is used to set the network address of the host computer while the reset button is used to reset the whole program. Meanwhile, the tilt and rotate functions are implemented using the orientation sensor in the device to control the view angle of the display.

G. Font Renderer

During the implementation of MetroBuzz, we found that the original font renderer of JOGL was not sufficient for our needs in terms of rendering quality, and there were no suitable public text rendering libraries for JOGL available. Therefore we decided to implement our own font renderer.

As a starting point, we ported Nicolas Rougier's freetype-gl project from C to Java.[4] Instead of getting glyph information from the Freetype library directly, we draw each glyph on an image using Graphics package in Java and retrieve the information by measuring the generated image pixel by pixel.

The code is organized as follows:

1) FontAtlas: Used for packing small regions into a bigger texture which is based on the skyline bottom left algorithm which appears to be well suited for storing glyphs [14].

2) FreshFont: Used for creating bitmap glyphs and uploading them to the atlas.

3) FontMarkup: A simple structure for describing text properties (font family, font size, colors ...).

4) Glyph: The structure that describes a texture glyph (advance, kerning...).

5) FontBuffer: A generic vertex buffer used by opengl to draw the text.

We will also make our JOGL font renderer as a public project on Google code so other developers can use and improve it.

III. Discussion

We have built a user friendly interface showing agent-based traffic flows with statistics and allowing users to interact with the data. The source code, including the custom font renderer are available on Google code with Open Source license. A demo of MetroBuzz was shown at the first Future Cities Laborotory Conference "Making Future Cities" held from 12 September to 14 September 2011 at Nanyang Technological University. Another demo was shown at the Swissnex Singapore Christmas Party on 12 December 2011.

We received positive responses from both experts and non-experts alike, and many good suggestions for further improvements were made. Some users think it's really cool to have the tablet control and they can play with it. The 3D visualization details are clear and meaningful. Transportation specialists typically pointed out, that the cylinder shapes should be replaced by arbitrary polygons, so dedicated city areas can be selected, and the tool would become much more useful.

IV. Conclusion and Future Work

MetroBuzz is a project that not only allows expert to explore transportation data but also helps normal users understand the data. Users can interact with the program by keyboard or the application we developed on Android tablets.

However, MetroBuzz is still in the early stage of developments and we will continue to improve it. First, we plan to address the suggestions made by the transportation people, i.e. to make the selection of areas of interest more flexible.

[3] http://code.google.com/p/androidmetrobuzz/

[4] http://code.google.com/p/freetype-gl/

Second, we plan to use different data sets, such as pedestrian or material flows, or real-world data obtained from mobile sensing applications. Finally, we also plan to enhance the interaction by using emerging technologies, such as Microsoft's Kinect for gestural interaction.

Acknowledgments

Special thanks are due to Kay W. Axhausen and Alexander Erath of the Mobility and Transportation research module at the Future Cities Laboratory for providing us the data and guiding us how to use the data. Great thanks to Nicolas Rougier for his simple but useful Freetype-gl code which allowed us to develop our own code fast.

References

[1] D. J. Peuquet, *Representations of space and time*. The Guilford Press, 2002.

[2] M. P. Kwan and J. Lee, "Geovisualization of human activity patterns using 3D GIS: a time-geographic approach," *Spatially integrated social science*, vol. 27, 2004.

[3] K. C. Cox, S. G. Eick, and T. He, "3D Geographic Network Displays," *ACM Sigmod Record*, vol. 25, pp. 50–54, 1996.

[4] C. Zhong, T. Wang, W. Zeng, and S. Müller Arisona, "Spatiotemporal visualization: A survey and outlook," in *Digital Urban Modeling and Simulation*, S. Müller Arisona, P. Wonka, G. Aschwanden, and J. Halatsch, Eds. Springer Berlin Heidelberg, 2012, vol. 242, in press.

[5] I. G. Cullen and V. Godson, *Urban networks: the structure of activity patterns*. Pergamon Press, 1975.

[6] K. Meister, M. Rieser, F. Ciari, A. Horni, M. Balmer, and K. W. Axhausen, "Anwendung eines agentenbasierten Modells der Verkehrsnachfrage auf die Schweiz," *Strassenverkehrstechnik*, vol. 53, pp. 269–280, 2009.

[7] ITU and ICT Data and Statistics. (2010) The world in 2010: ICT facts and figures. [Online]. Available: http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2010.pdf

[8] Motorola. (2008) Beyond mobile broadband. [Online]. Available: http://business.motorola.com/experiencelte/lte-experience.html

[9] S. Elwood, "Volunteered geographic information: future research directions motivated by critical, participatory, and feminist GIS," *GeoJournal*, vol. 72, no. 3, pp. 173–183, 2008.

[10] D. Shin, S. Müller Arisona, and G. Schmitt, "A crowdsourcing urban simulation platform using mobile devices and social sensing," in *CAAD Futures 2011*, July 2011.

[11] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[12] J. L. Bentley and J. H. Friedman, "Data structures for range searching," *ACM Computing Surveys (CSUR)*, vol. 11, no. 4, pp. 397–409, 1979.

[13] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *International Conference on Management of Data*. ACM, 1984, pp. 47–57.

[14] J. Jylänki. (2010) A thousand ways to pack the bin - a practical approach to two-dimensional rectangle bin packing. [Online]. Available: http://clb.demon.fi/files/RectangleBinPack.pdf