Computers, Environment and Urban Systems 45 (2014) 114–124

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/compenvurbsys

Concepts and techniques for integration, analysis and visualization of massive 3D point clouds

Rico Richter *, Jürgen Döllner

Hasso-Plattner-Institut, University of Potsdam, Germany

ARTICLE INFO

Article history:
Available online 13 August 2013

Keywords:
3D point clouds
System architecture
Classification
Out-of-core
Visualization

ABSTRACT

Remote sensing methods, such as LiDAR and image-based photogrammetry, are established approaches for capturing the physical world. Professional and low-cost scanning devices are capable of generating dense 3D point clouds. Typically, these 3D point clouds are preprocessed by GIS and are then used as input data in a variety of applications such as urban planning, environmental monitoring, disaster management, and simulation. The availability of area-wide 3D point clouds will drastically increase in the future due to the availability of novel capturing methods (e.g., driver assistance systems) and low-cost scanning devices. Applications, systems, and workflows will therefore face large collections of redundant, up-to-date 3D point clouds and have to cope with massive amounts of data. Hence, approaches are required that will efficiently integrate, update, manage, analyze, and visualize 3D point clouds. In this paper, we define requirements for a system infrastructure that enables the integration of 3D point clouds from heterogeneous capturing devices and different timestamps. Change detection and update strategies for 3D point clouds are presented that reduce storage requirements and offer new insights for analysis purposes. We also present an approach that attributes 3D point clouds with semantic information (e.g., object class category information), which enables more effective data processing, analysis, and visualization. Out-of-core real-time rendering techniques then allow for an interactive exploration of the entire 3D point cloud and the corresponding analysis results. Web-based visualization services are utilized to make 3D point clouds available to a large community. The proposed concepts and techniques are designed to establish 3D point clouds as base datasets, as well as rendering primitives for analysis and visualization tasks, which allow operations to be performed directly on the point data. Finally, we evaluate the presented system, report on its applications, and discuss further research challenges.

1. Introduction

Remote sensing technology, such as airborne laser scanning, has been used for decades to capture cities and landscapes. Aerial images and photogrammetric approaches allow the derivation of dense 3D point clouds and digital surface models from a *bird's eye view* (Leberl et al., 2010). Similarly, terrestrial and mobile scanning systems are well established to capture the environment from a *pedestrian's view* (e.g., building façades and streets Kang & Lu, 2011; Nebiker, Bleisch, & Christen, 2010). Image-based approaches for digital images (e.g., photos) allow even non-experts to generate 3D point clouds of objects and structures (e.g., via Autodesk 123D or Microsoft Photosynth Snavely, Garg, Seitz, & Szeliski, 2008).

Other potential 3D point cloud sources include driver assistance systems, for example. Radar, laser, GPS, and camera technologies are used for permanent monitoring of a vehicle's environment, which enables the derivation of geo-referenced data such as images and 3D point clouds. The combined and collaborative use of these technologies allows to generation of dense 3D point clouds for large parts of our physical world. The use of existing technologies and low-cost devices allow a cost-efficient, redundant, area-wide, and fully automatic data acquisition for the establishment of *point cloud databases* that can be updated over short time intervals (e.g., weekly, daily, hourly).

The availability, accuracy, density, and massivity of 3D point clouds are assumed to vastly increase within the next several years. However, the traditional 3D point cloud applications used in urban planning, environmental monitoring, disaster management, and simulation in general are not designed for selective and immediate processing and updating of 3D point clouds (e.g., if the input data partially change). Consequently, concepts and techniques are required that can be applied directly to 3D point clouds or subsets of the data. For instance, analysis and simulation results can be provided as per-point attributes to avoid the generation of mesh-based 3D models, which become outdated when the 3D point cloud partially changes (Richter, Kyprianidis, & Döllner, in

* Corresponding author. Tel.: +49 33155093910.
E-mail addresses: rico.richter@hpi.uni-potsdam.de (R. Richter), doellnerg@hpi.uni-potsdam.de (J. Döllner).

0198-9715/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compenvurbsys.2013.07.004

press). Other applications are 3D geovirtual environments (3D GeoVEs) and virtual 3D city models that require time-intensive and costly workflows for model creation and geometry updates. These 3D GeoVEs can alternatively be represented by dense 3D point clouds, which significantly minimizes the effort for preparation and updates (Nebiker et al., 2010). This becomes particularly important when the 3D point cloud database permanently changes and up-to-date models and analysis results are required.

The collection, preparation, and integration of 3D point clouds from different, heterogeneous data sources (e.g., aerial and terrestrial LiDAR scans or image collections) and autonomously operating scanning systems (e.g., vehicle fleets) require a *service-based infrastructure* (i.e., loosely coupled and distributed sensors (Bröring, Stasch, & Echterhoff, 2012), processing (Schut, 2007), data provision (Vretanos, 2010), and visualization components (Hagedorn, 2010) that allow maintenance of a point cloud database). Continuous data acquisition results in redundant 3D point clouds for large parts of the captured area (Kang & Lu, 2011). To avoid the need for redundant storage, *incremental database updates* and *change detection* algorithms are required to determine the differences between collected and already available 3D point clouds. This allows for *selective updating* for parts of the data.

Applications, systems, and workflows often require only a subset of 3D point clouds that belong to a specific *object category*, such as building (Jochem, Höfle, Wichmann, Rutzinger, & Zipf, 2012), vegetation (Höfle, Hollaus, & Hagenauer, 2012), ground (Meng, Currit, & Zhao, 2010), city furniture (Golovinskiy, Kim, & Funkhouser, 2009), and infrastructure networks (Clode, Kootsookos, & Rottensteiner, 2004). However, 3D point clouds generally provide only geometric information and per-point attributes (e.g., color and timestamp). The object category information for each point can be determined with an *object class segmentation* that is based on the structure and topology of the data. In this context, the challenge is to obtain object class segmentations that combine the concepts and techniques designed for 3D point clouds with specific characteristics (e.g., resulting from terrestrial, mobile, or airborne scans).

Visualization and presentation techniques for massive 3D point clouds are essential for the understanding and communication of spatial information and of the analysis and simulation results (Bettio et al., 2009; Kim & Medioni, 2010; Kreylos, Bawden, & Kellogg, 2008). Point-based rendering techniques can cope with massive 3D point clouds and enable an interactive visualization and exploration of the data (Richter & Döllner, 2010; Wimmer & Scheiblauer, 2006). These techniques can be applied to a 3D point cloud database, thereby enabling different rendering styles (e.g., photorealistic, non-photorealistic, or based on point attributes (Botsch, Hornung, Zwicker, & Kobbelt, 2005)).

In general, rendering techniques require direct data access, which complicates the provision of interactive visualization for a large number of users. Service-based visualization strategies can overcome this limitation and enable easy data access even on client systems with limited computing capabilities (Paravati & Sanna, 2011). For example, Web View Services (WVS) can provide 3D geodata (Hagedorn, 2010) similar to the established Web Map Services (e.g., Google Maps and OpenStreetMap). These web services separate the effort required for data management, rendering, and presentation, and provide 3D content to a large number of users (Döllner, Hagedorn, & Klimke, 2012).

In this contribution, we present a system architecture for the integration, update, management, analysis, and visualization of massive 3D point clouds. This system allows the integration of data into a point cloud database by utilizing a GPU-based change detection that enables selective database updates. In addition, we describe an object class segmentation approach that divides 3D point clouds into multiple object classes. The system provides interactive rendering techniques as well as visualization services to allow a customized, task-specific, and interactive exploration on client systems.

This paper is structured as follows: In Section 2, we present the current usage scenarios and applications of 3D point clouds. Section 3 introduces the overall system architecture for 3D point cloud management, analysis, and visualization. Integration, updating, and storage concepts are presented in Section 4. In Section 5, we present ways to identify object classes as well as static and mobile structures in 3D point clouds. Section 6 presents out-of-core visualization techniques and a service-based rendering concept. As a case study, in Section 7, we present change detection, object class segmentation, and visualization results for a real-world dataset with billions of points. Section 8 gives a conclusion and outlook on future challenges.

2. Problem statement

Traditional applications, systems, and workflows use 3D point clouds only as input data for the derivation of 3D models of the captured target (Vosselman, Gorte, Sithole, & Rabbani, 2004). Several approaches have been proposed to reconstruct mesh-based 3D models from (1) aerial scans (e.g., 3D city models, elevation models, building models), (2) terrestrial scans (e.g., façades models, facility models), (3) mobile scans (e.g., infrastructure models for streets and power lines), and (4) images and deep images (e.g., indoor models).

When the number of 3D point increases, 3D model construction becomes time consuming and results in large 3D models that pose challenges for analysis. A typical solution is to use generalized 3D models that have less details than their 3D point cloud representations. Other approaches use multiple Level-of-Detail (LoD) representations to cope with the 3D model size (Hoppe, 1996). These LoD representations are generated by preprocessing and cannot be simply built by reducing data (i.e., removing triangles). In contrast, 3D point clouds allow the generation of any LoD representation by adaptively removing points (Marton & Gobbetti, 2004). Further advantages of point-based 3D models are rapid updates that can be performed for any part of the data without the need to generate a new 3D model and corresponding LoDs. In summary, massive 3D point clouds can be integrated, updated, and accessed with much less effort than is required for 3D models.

3D point clouds are often visualized with points or splats as rendering primitives that result in non-solid surface representations (Nebiker et al., 2010; Wimmer & Scheiblauer, 2006). In contrast to solid geometry (e.g., modeled by triangular irregular networks), these representations are less suited for interaction, navigation and perception. However, several rendering techniques have been proposed to render and represent 3D point clouds by solid surfaces (Duranleau, Beaudoin, & Poulin, 2008) and different stylizations (e.g., photorealistic or non-photorealistic). These rendering techniques require per-point attributes such as surface normals, local point densities, colors, and object class information that are typically computed in a preprocessing step.

We have identified the following requirements that need to be addressed to establish 3D point clouds as basic datasets to provide 3D contents:

- a data infrastructure to collect 3D point clouds from multiple, heterogeneous sources;
- system components to integrate, update and access 3D point clouds in a database;
- applications and algorithms to operate directly on 3D point clouds;
- rendering techniques to enable a task and application-specific visualization of 3D point clouds;