ARTICLE IN PRESS

ISPRS Journal of Photogrammetry and Remote Sensing xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks

Dimitri Bulatov *, Gisela Häufel, Jochen Meidow, Melanie Pohl, Peter Solbrig, Peter Wernerus

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Gutleuthausstr. 1, 76275 Ettlingen, Germany

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Building detection
Building reconstruction
Pose estimation
Simulation
Texturing
Urban terrain

ABSTRACT

Highly detailed 3D urban terrain models are the base for quick response tasks with indispensable human participation, e.g., disaster management. Thus, it is important to automate and accelerate the process of urban terrain modeling from sensor data such that the resulting 3D model is semantic, compact, recognizable, and easily usable for training and simulation purposes. To provide essential geometric attributes, buildings and trees must be identified among elevated objects in digital surface models. After building ground-plan estimation and roof details analysis, images from oblique airborne imagery are used to cover building faces with up-to-date texture thus achieving a better recognizability of the model. The three steps of the texturing procedure are sensor pose estimation, assessment of polygons projected into the images, and texture synthesis. Free geographic data, providing additional information about streets, forest areas, and other topographic object types, suppress false alarms and enrich the reconstruction results.

1. Introduction

1.1. Motivation

The importance of accurate and realistic modeling of urban terrain has been demonstrated in various applications such as automatic navigation, urban planning, disaster management, and security-related rapid response tasks. More than the others, the latter two applications are time-critical. A precise plan requires up-to-date knowledge about the terrain including its recent and perhaps sudden changes. Though from a certain moment on, field training becomes indispensable for such time-critical application, the advantages of simulation are numerous and evident: In many cases, it is relatively easy to tailor the simulated environment to a concrete situation, to manipulate weather and time-of-day conditions, and to experiment with new, expensive sensors and equipment, which, contrary to the real world, does not need to be prepared, transported, and maintained. Therefore, simulation is continuously gaining popularity as a training tool for decision-taking within a short time and under enormous pressure. From the point of view of the abstraction of a simulated environment, it can be stated that the smaller the training unit is and the lower in the chain of command it is, the less abstract the simulated training environment usually has to be. This allows to train small unit leaders under realistic conditions like stress and unclear situations. Commercial simulators give a good example: Strategy games often make use of more abstract 2D maps whereas flight simulators show detailed and realistic virtual 3D environments the aircraft crew can interact with. Our system of choice is Virtual Battlespace 2 (VBS2).[1] Although chiefly a training simulator for military applications, VBS2 can also be used to demonstrate the characteristics of automatically instantiated 3D models and provides a convenient access to our results, including for evaluation purposes (see Fig. 1). As reported in Bulatov et al. (2012a), VBS2 offers relatively user-friendly and clearly structured tools, which allow importing not only imagery and free geographic data (shapefiles) but also reconstruction results such as elevation maps and building models. However, time-critical applications and their simulation also put certain requirements on the 3D models and the underlying algorithms.

First of all, the *context information* is essential for a plausible interaction with the environment even though models may become less accurate than 2D manifolds in space without context information. Expert users who want to find their way in an unknown urban terrain will rate the simulation the more convincing the more information of the scene this simulation includes; like

* Corresponding author. Tel.: +49 7243992136.
E-mail address: dimitri.bulatov@iosb.fraunhofer.de (D. Bulatov).

[1] Virtual Battle Space 2 by Bohemia Interactive, http://products.bisimulations.com/products/vbs2, May. 2013.

http://dx.doi.org/10.1016/j.isprsjprs.2014.02.016
0924-2716/

Please cite this article in press as: Bulatov, D., et al. Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. ISPRS J. Photogram. Remote Sensing (2014), http://dx.doi.org/10.1016/j.isprsjprs.2014.02.016

small details of buildings, texture of walls and roofs, and information about vegetation and streets. For example, in order to make vehicles move faster on streets than in other areas of the terrain, or to make some trees lose their leaves in winter while others do not, it is necessary to identify and to model the relevant topographic objects types "streets" and "vegetation". The same argument holds for buildings: To customize collision geometry and damage models, it is important to identify buildings and their parts. Context information is important both within a simulation process and for faster updating the terrain if some changes have happened. For example, if after a certain building measure, a wall has been removed, a semantic model can be updated by deleting a couple of lines in a text document whereas a model without semantics would probably require a mesh-processing software.

Closely related to the previous point is the *level of compression* of a model. Effort must be made to generalize structures and to increase the level of abstraction. For example, in many applications, it is sufficient to describe a building wall by a single polygon. A tree model at the approximate position of the trunk is more efficient since it can be instantiated over and over again. It is also more visually appealing than fitting a shape around the data points describing this tree, which leads to a "tree-mushroom" with many small textured polygons.

Finally, for time-critical applications, urban terrain models must be instantiated and integrated into the simulation environment within a reasonable time. This motivated us to develop a rather modular procedure and avoid, for instance, graph-based methods that are computationally expensive and run danger to end up in a local minimum. Also, evolution of available data has to be considered. At the beginning of an operation, often only availability of satellite/aerial images or airborne laser scans may be assumed, which leads to the assumption that the scene has a rather 2.5D structure and that requirements on sensor data must be kept to a minimum. As the operation advances, more detailed data of higher detail taken from lower altitude is acquired. Thus, as soon as an additional source of information becomes available, the first thing we must be able to do is to enrich the available model by this information – without need to recompute the model – and to make the simulated world more realistic and suitable for training and planning purposes. An inspiring publication about how to increase the level of detail of 3D models by sensor data fusion has been written by Haala (2005). In our paper, oblique UAV images are used to cover building walls with real textures aiming for better recognizability of the scene. The second possibility, not followed in this paper, is to update and to improve the geometry of the reconstruction. For example, it was shown in (Bulatov et al., 2012b) how streets imported from free geographic data can be used to filter out spurious hypotheses for buildings caused by moving objects.

1.2. Previous work

In the recent years, the advantages of 3D reconstruction from optical data, such as aerial images, including and especially images and videos captured by unmanned aerial vehicles, have become evident. Besides relatively low cost and easier availability of optical images compared to data collected by traditional active sensors, there is also rapid progress in the quality of imaging sensors and sensor platforms. In addition, there is a broad spectrum of algorithms that can process optical data, even if it is noisy, contains many outliers (Bulatov and Lavery, 2010), or covers a large urban area at a high resolution (Hirschmüller, 2008). There is a large number of excellent algorithms that do not presuppose a dense point cloud as input, but rely on edge matching in images (Baillard and Zisserman, 2000), color-segmentation (Henricsson et al., 1996), or hierarchical assessment (Fischer et al., 1998). However, these methods are less suitable for building detection in large scenes than for model instantiation of a few single buildings. Most state-of-the-art approaches for context-based modeling of buildings therefore rely on dense point clouds, see (Elberink and Vosselman, 2009; Sohn et al., 2012; Huang et al., 2013; Lafarge et al., 2013; Lafarge and Mallet, 2012; Brenner, 2000; Gross et al., 2005; Rottensteiner, 2006), preferably resampled into an *elevation map*. This is the main input of a typical 2.5D approach: The vertical direction is the dominant one and elevation is interpreted as a function of longitude and latitude. If optical images are given, such point clouds and elevations map can be computed from depth maps obtained from multi-view configurations (Bulatov et al., 2011a; Haala and Rothermel, 2012).

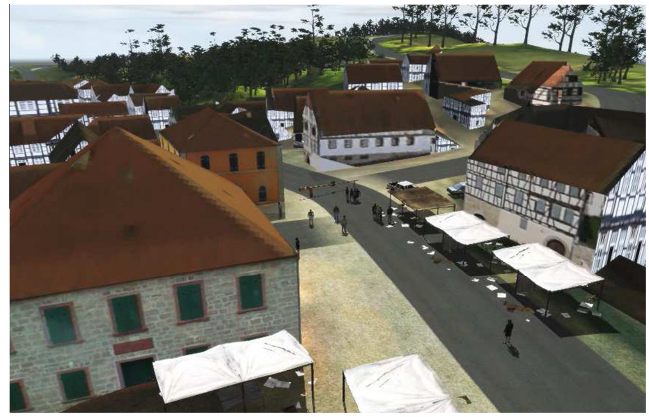

Fig. 1. A view of a scene rendered by the VBS2 simulation environment.

The authors of the related contributions mostly agree that, because of a huge variety of roof types, the roof detail analysis is the most challenging and complicated step. Their approaches can be classified as model-driven and data-driven approaches. The model-driven approaches make use libraries of models into which the input data should be fit. The key idea of the proceedings of (Elberink and Vosselman, 2009; Verma et al., 2006) and also of (Brenner, 2000) is matching of noisy graphs. The vertices of the graph are roof segments and the weights of the edges represent relations between neighboring roofs (step edges, dormer edges, concave edges, orthogonal edges, etc.). While Verma et al. (2006) emphasize subdivision of complex buildings and introduce symmetry relations as soft constraints for fitting the data to a model, one important contribution of Elberink and Vosselman (2009) is handling uncovered details. The number of uncovered details depends on the input data, the result of the graph matching algorithm, and on the variety of objects in the library. For example, if only planar roofs are contained in the library, modeling of non-planar roofs is impossible. Brenner (2000) reports short-comings for the class of buildings with flat roofs. Other difficulties of model-driven approaches are initialization (Huang et al., 2013) and a requirement for additional input, such as accurate building footprints (Brenner, 2000).

In contrast, the data-driven approaches exploit data in combination with reasonable priors in order to extract certain model properties. For man-made objects such as buildings, examples of these priors are parallelism, orthogonality, and symmetry. Rottensteiner (2006) introduces after dominant planes computation soft constraints for regularizing building shapes while Sohn et al. (2012) evaluate a cost function that, among other things, takes into account the model complexity for several hypotheses for the roof structure. Soft constraints often cannot completely correct the inconsistencies while hard constraints can lead to linear dependence and, due to data noise, even some self-contradicting conditions. Also, it is not always clear how to choose weights for these constraints.