Automatic Representation of Urban Terrain Models for Simulations on the Example of VBS2

Dimitri Bulatov*, Gisela Häufel, Peter Solbrig, Peter Wernerus
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Gutleuthausstr. 1, 76275 Ettlingen, Germany

ABSTRACT

Virtual simulations have been on the rise together with the fast progress of rendering engines and graphics hardware. Especially in military applications, offensive actions in modern peace-keeping missions have to be quick, firm and precise, especially under the conditions of asymmetric warfare, non-cooperative urban terrain and rapidly developing situations. Going through the mission in simulation can prepare the minds of soldiers and leaders, increase self-confidence and tactical awareness, and finally save lives. This work is dedicated to illustrate the potential and limitations of integration of semantic urban terrain models into a simulation. Our system of choice is Virtual Battle Space 2, a simulation system created by Bohemia Interactive System. The topographic object types that we are able to export into this simulation engine are either results of the sensor data evaluation (building, trees, grass, and ground), which is done fully-automatically, or entities obtained from publicly available sources (streets and water-areas), which can be converted into the system-proper format with a few mouse clicks.

The focus of this work lies in integrating of information about building façades into the simulation. We are inspired by state-of the art methods that allow for automatic extraction of doors and windows in laser point clouds captured from building walls and thus increase the level of details of building models. As a consequence, it is important to simulate these animationable entities. Doing so, we are able to make accessible some of the buildings in the simulation.

Keywords: building reconstruction, level of detail, simulation, training, texturing, UAV, urban terrain, VBS2.

1. INTRODUCTION

1.1 Motivation

In civil and military applications, where one has to understand the terrain before the plan falls into place, where the situation awareness must be kept at a high level even in a confusing environment, running a mission in simulation before going into the field is already a widely-used approach. Not only in the field of leadership training but also in virtual execution of „what-if“ scenarios such as a retreat plan for an UN contingent in case of a major escalation or a hypothetical disaster evacuation, both constructive and virtual simulation systems offer the ideal platform for training and mission rehearsal. Thus, combat simulation centers have been well-established within armed forces of many nations. Simulators like our system of choice, Virtual Battle Space 2 (VBS2) by Bohemia Interactive [11], come with powerful mission editors and after-action-review tools. On the one hand, the recent developments of combat simulation industry benefit massively from the progress in the field of computer gaming. On the other hand and in contrast to pure gaming, in military simulation the requirements of military team and leadership training must remain in focus and be subject of continuous changes. Within the last two decades, one important example of changes of deployment of the national and allied defense forces is acting under conditions of an asymmetric conflict in a complex urban environment under strict rules of engagement and the attention of the media and the public. The issues of weapon effect, escape routes, local visibility and accessibility analysis, and especially the regard for civilians, require a higher level of knowledge about the underlying urban scenery. Thus, going through a mission thoroughly over and over again in a virtual environment might be worthless if the level of detail of the simulated area of operations is too low for tasks to be performed in real life. That can easily happen if the simulation terrain is not photo-realistic or not even geo-specific, and thus, not recognizable, or if not all building entrances were detected from which fire can be opened.

Usually, creating geo-specific or photo-realistic simulation terrain databases is a time-consuming and thus expensive process, carried out by rare specialists. Fortunately, photogrammetry and computer vision offer solutions for re-construction of 3D terrain models from various sensor data. This also includes data the military often produce “on the

*dimitri.bulatov@iosb.fraunhofer.de; tel 00497243992136; fax 00497243992299; iosb.fraunhofer.de

fly" during reconnaissance and surveillance missions with unmanned aerial vehicles, airplanes or ground vehicles. The goal of this paper is to demonstrate the potential of feeding a simulation with relevant topographic object types extracted from sensor data, even though the sensor data may be not primarily designed for this goal.

1.2 Main contributions

Closing the gap between operationally gathered geo-related data and the time-efficient generation of high fidelity simulation terrain with a minimum of interaction was a topic of our previous work [4]. Following features have been added in the meantime to the algorithms of urban terrain reconstruction and simulation of reconstruction results.

- Vegetation areas, like trees and grass, are exported directly from reconstruction results to VBS2, without any intermediate converters. With respect to trees, we now distinguish between isolated trees, groups of trees and larger forest regions. For isolated trees, efforts are made to estimate position, height and diameter more precisely. For groups of trees, only heights and approximate positions are important. For large forest regions, obtained from shape-files, trees are placed quite randomly within areas defined by shape-files. For details, see Section 3.2 and Section 3.3.
- Similarly to shape-files for large forest regions, we use shape-files for streets and water areas. These shape-files may stem from external sources and to integrate them into the simulation, we use some VBS2-proper tools. To do this, several assumptions, described in a more precise way in Section 3.2, must be made.
- Texturing of building polygons is performed fully-automatically once the sensor has been registered in the model coordinate system. An orthophoto (simulated as a parallel projecting camera) is used to texture building roofs while oblique UAV imagery represent a powerful source for texturing building walls.
- The most important contribution concerns upgrading the model instances of several important buildings from the level of detail (LOD) 2 to LOD 3. From the point of view of reconstruction, this means that façade planes should be corrected, and possibly subdivided into a finer set of topographic object types, such as doors, windows, etc. From the point of view of simulation, we wish to represent these object types and corresponding animations in VBS2. This will be shown in Section 3.4.

1.3 Organization of this paper

All the new features described by the end of the previous section will be explained in more detail in Section 3, where our five-step procedure for exporting reconstruction results into the simulation environment of VBS2 is presented. However, it is of a general interest to give a short overview of our procedure for urban terrain reconstruction from sensor data. This will be done in Section 2. For more details, the interested reader is referred to [5]. Finally, an example of the simulation is given in Section 4 while conclusions and ideas of future work are presented in Section 5.

2. PRELIMINARIES ON URBAN TERRAIN MODELING FROM SENSOR DATA

The main input of our algorithm for semantic model instantiation is given by an elevation map covering the area of interest. Such an elevation map may be sampled from an Airborne Laser Scan or calculated from a photogrammetrically obtained dense point cloud [6],[9]. With an elevation map, we concentrate on 2.5D data-sets because of three reasons. Firstly, building detection is easier to perform in a 2.5D scenario. Secondly, by reconstructing roof structures, we already have a good approximation of the extension of a building since it cannot "grow downwards". Finally, it is not realistic to capture all walls of all buildings within a large range data acquisition of the format of interest for our application.

A building model thus obtained will have a ground plan and a possibly small and topologically consistent set of dominant roof polygons. The façades are modeled by projecting the endpoints of every border- or step-edge of the roof to the ground, thus forming vertical trapezoids. Such output building models are context-based (buildings are recognized as buildings, roofs are recognized and modeled as roofs etc.), exhibit a relatively high level of compression, and preserve the structure of objects.

However, besides the elevation map and the orthophoto, many additional sources are often available:

- A digital orthophoto, which is co-registered with the elevation map, can be used for extraction of vegetation – especially if a near infrared channel is available, improving building outlines, as well as texturing building roofs.

- Shape-files are used for large forest areas, water areas, streets etc.
- A set of oblique images together with corresponding poses and depth maps is employed for correction of façade planes, texturing building walls and even extraction of doors and windows (see below).
- A more reliable extraction of doors and windows may be performed when a dense and precise point cloud from a Terrestrial Laser Scan is given

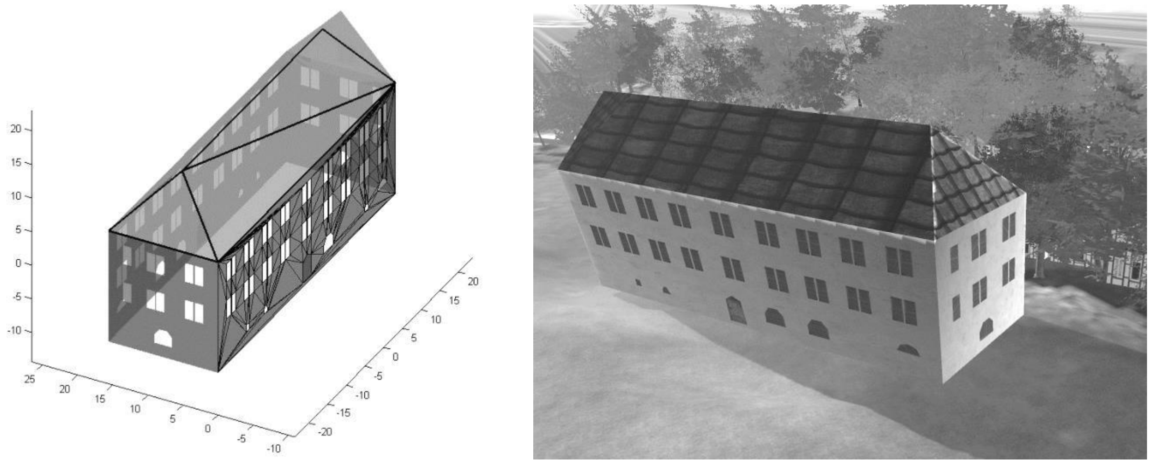
As we see, additional input allows augmenting level of detail because façade analysis can be performed. The wall planes can be corrected, foreground objects (both static, such as balconies, and dynamic, such as persons) as well as doors and windows can be extracted, and texture pictures of high resolution, provided by oblique imagery, can be used for a better recognizability of the model.

In summary, while the LOD2 is characterized by a 2.5D structure models, we strive for façade analysis and thus obtaining LOD3 model instances of several selected buildings of the scene. In the following paragraphs, we will briefly describe the three modules of the algorithm: Building detection, building reconstruction, and texturing followed by detection of doors and windows.

Detection of buildings in elevation maps is a typical classification task. We must identify every pixel of the terrain with one of several classes: Ground, buildings, trees, grass, etc., but also a class of outliers. We assume that large regions, which are elevated over ground, correspond either to trees or to buildings. Thus, building detection works as follows [5]. First, the ground is computed. In regions without sharp differences of altitude, like rocks, a reliable ground model is possible even in presence of outliers. The normalized difference of the elevation map and the ground is denoted as normalized elevation map. Then, vegetation is extracted according to the NDVI-measure that is either given from the near infrared channel or can be calculated from the isolated trees [7] and used as a training data across the image. Large elevated regions with a high measure of NDVI are hypotheses for trees and the larger non-elevated regions with a high measure of NDVI are hypotheses for grass. These regions should be properly modeled in Section 3. Finally, the last module presupposes filtering the remaining elevated region of the difference between according to their size and eccentricity as well as labeling the building hypotheses. Also, subdivision of buildings according to the height differences is performed.

Reconstruction of a single building is performed at the LOD2-level in two steps: Building outlining and roof detail analysis. During building outlining, we approximate the building contour by means of a polygon. Next, we compute several dominant planes for all 3D points within the building shape. The initial polygonization of these planes helps to determine the extents of the underlying roof detail. Then, the planes are intersected between each other. The polygons are aligned with these cut-lines and with the building outlines, thus allowing a more consistent representation of the roof. Additionally to cut-lines, experiments with step-lines are currently being performed. The remaining uncovered details are filled either by merging them with an adjacent roof-plane, or by fitting a single plane, or by simply ignoring it, if its area is too small or the resulting deviations in elevation are too large. For LOD2, building walls are added from every edge of the roof towards the corresponding points on the ground. In this work, we assume the availability of results for façade analysis, that is, a decomposition of the façade polygons into classes “wall”, “door”, and “window”, which is an output of various recently published methods [1][2][3][10]. Our goal is to represent these classes and their animations in VBS2. As an example, a historical building Prinzenbau (in Stuttgart, Southern Germany), which was modeled and provided by the authors of [3], is visualized in Figure 1. In order to simplify the model for simulation purposes, four post-processing steps have been applied.

- Too small and “unnecessary” polygons, such as windowsills, are eliminated. This is done by examining the support plane of the current façade polygon and the incident wall.
- The number of vertices in rounded windows and doors can be reduced by a polygon simplification routine [8].
- The door/window polygon is projected into the plain of the incident wall polygon (see texturing step below).
- The wall polygons, which now exhibit holes, are partitioned into a triangle mesh for the subsequent export into VBS2 (shown in Figure 1). The default method for subdividing convex polygons is Constrained Delaunay Triangulation.



a b

Figure 1. Prinzenbau: a. Roof, wall, door and window polygons are shown with different colors and transparencies. For two examples (roof and wall), subdivision into triangles according to Constrained Delaunay Triangulation is shown; b. Prinzenbau embedded in VBS2 simulation

The texturing step is needed for a better recognizability of the building. We use the orthophoto to texture the roof polygons. The simple structure of the ortho-camera (in homogeneous coordinates: a 3×4-matrix with zeros in the third column and forming identity matrix for other three columns) helps to avoid systematic errors: no pose estimation is required. In contrast, UAV imagery has proved to be an ideal tool for computation of the side textures. The three steps for our texturing algorithm are: Pose estimation of the underlying camera, occlusion analysis, and texture synthesis. With respect to the doors and windows, the transparencies are not handled. Also, their support planes are not necessarily the same as for the wall/roof they are incident with. If the loss of quality is not that significant, we project them into the plane of the wall or roof during the stage of occlusion analysis and texturing, and treat this group as one single polygon for the sake of computing time.

3. INTEGRATION OF 3D MODELS INTO VBS2

The input of this section consists of the result of the context-based urban terrain reconstruction, possibly enriched by additional, external sources. The 3D coordinates of all objects are given in UTM/WGS84 coordinates. In absence of the internal navigation data, the process of geo-referencing takes place by computing an image-to-image homography between the color-map and the orthophoto. The offset in vertical direction between coordinate systems was estimated interactively.

We now present our method for integration of the model instances into the combat simulation system VBS2. By highlighting the five main steps of this procedure in the five upcoming sections, we will refer to the most important tools of this software and the major challenges a user has to deal with.

3.1 Creating map-frame and exporting the ground map

The ground is exported by resampling the DTM into an equally spaced square grid of a limited resolution and saving as a list of 3D points in UTM/WGS84 coordinates. The file format needed is a .xyz text file containing coordinates of the sampling points. The radiometry of the ground is given by the orthophoto in .png-format. The area of interest, or the world of the simulation, is called map-frame. This area cannot be arbitrarily small, because the internal number of grid points is constant and the resolution is limited. Usually, for training of small units, a region of around 5000 m^2 is a plausible minimum size of the simulation world. However for a UAV-flight, it usually makes sense to extrapolate the terrain.

The default tool for exporting elevation maps is Visitor4, which also exports lists of objects (buildings, trees) into VBS2. These VBS2-proper lists typically have extension .lbt while a single VBS2-proper object is stored in a .p3d-file. A VBS2-proper list contains the rotation matrix, the UTM-coordinates of the center of the bounding box, and the name of the corresponding file for every object. The animations of the VBS2-relevant objects (such as swimming in the water, opening doors), must be specified in two configuration files.

3.2 Enriching models with geographic data stemming from external sources

In order to augment the level of detail of our simulation, we wish to import instances of those topographic object types that could or should not be extracted from the sensor data. These object types could be for example streets, water areas, but also large forest areas around the area of interest. As an example of data provided by the public sources, we refer to the OpenStreetMap community, which creates and provides free geographic data for huge areas in the world. This data is available online and, as a consequence, detailed road maps may be used for image analysis tasks and applications. These maps are downloadable in form of shape-files using an intermediate provider GeoFabrik-GmbH[1].

We process shape-files by Landbuilder, which is a sub-module of Visitor4 that allows specification of necessary parameters for shapes, like, in case of large forest regions, the types of trees, the probabilities of these types in the simulated world, as well as the density of afforestation. Once a shape has been loaded in Landbuilder, these parameters can be set with a few mouse-clicks by the user. The shape-files for streets can be processed in Landbuilder as well. This allows setting road width, its slope, and the specified road type.

From the point of view of positioning the roads, the terrain may be manipulated interactively, depending on the given grid and the relief. The reason is that otherwise significant differences of elevation may happen perpendicularly to the directions of street which is not realistic. Applying a terrain changing module of the Landbuilder, the terrain is "milled" depending on the shape-file coding the roads, their widths and the requested slopes. Now, roads run horizontally. We conclude this procedure by the import of the modified elevation map in .xyz-format in Visitor4 and the import of the .lbt-file for streets (see Figure 2a).

In the case of water bodies, such as lakes, the elevation map must be changed in a similar way to the method of changing the terrain for positioning streets. The input is the shape-file of the water body, the initial terrain model as well as the water level and the maximal depth of the water body. Applying the river module of the Landbuilder, the terrain is changed. The new elevation map is imported. In contrast to the previous paragraph, the configuration file must be modified by the definition of water and its properties, such as water texture (see Figure 2b).

3.3 Exporting trees and grass areas extracted from sensor data

The approach described in the previous section has a disadvantage that too many parameters should be set in Landbuilder (theoretically for every shape) which decreases the level of automation. Furthermore, especially in the case of vegetation, the main problem about shape-files and their export to VBS2 lies in their inexact position in the output model: Random trees of random size are placed into random spots inside the shape. But often, at least the geometry information can be estimated for trees from the available sensor data. An isolated tree can be recognized as an almost circular elevated region in the normalized elevation map which belongs to the class vegetation and has an area between 0.5 m^2 and 12 m^2. The height of such a tree is estimated from a high quantile value of the elevation. Its position and its diameter are estimated from the ground map and from the region size, and the rotation around z-axis is arbitrary. The tree type is chosen from the library of available .p3d models of trees in VBS2, with regard for climatic conditions of the simulated region. For groups of trees, delineation of such structures can be performed in the future. Currently, the first tree is modeled in the highest point of such a multi-tree component. Then, the component is reduced by pixels within a circular region of constant diameter around this point. The procedure is then applied again on the remaining pixel. It terminates when there are no pixels in the multi-tree component. Finally, grass regions are modeled in the non-elevated regions covered by vegetation.

Summarizing this section, we strive for:

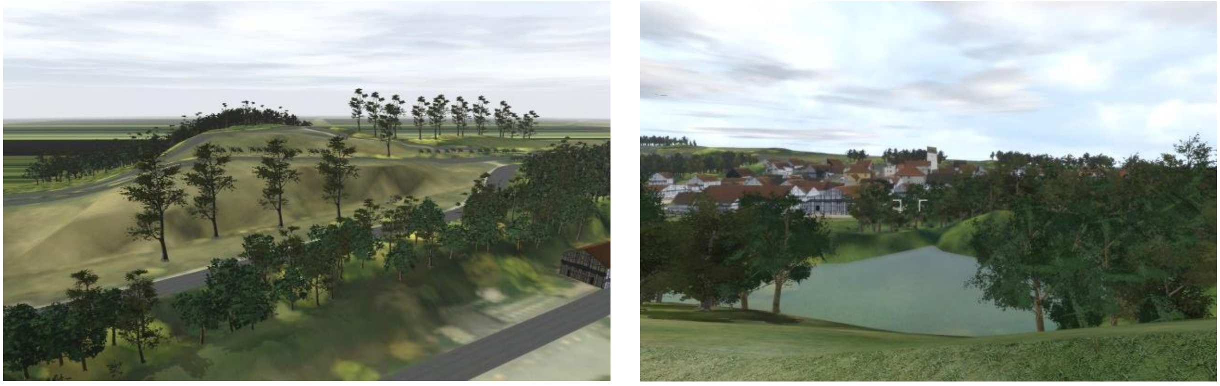
- modeling the exact position, height and diameter of isolated trees and
- modeling the approximate position and height of trees within the multi-tree component

With random types of trees, both results are exported directly into a separate .lbt-file and can be visualized in Visitor4 and used in VBS2 without any need to process them in Landbuilder.

---

[1] http://download.geofabrik.de/europe/germany/

Two screen-shots of the simulation after finishing Step 3 are shown in Figure 3. Here, the position, the diameter and the altitude of the tree are extracted from the normalized elevation map.

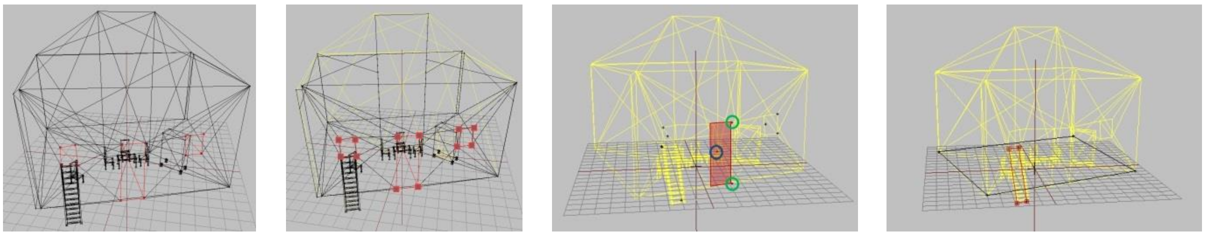


a b

Figure 2. Terrain modification in VBS2: a. roads and b. water bodies.

a b

Figure 3. Direct modeling tree and grass in VBS2.



a b c d

Figure 4. Building in Oxygen2: a. Resolution LOD, b. Geometry LOD, c. Memory LOD d. Roadway LOD.

3.4 Building representation in VBS2

Contrary to the previous steps, where system-proper object types (trees, hassocks), we need to create an own .p3d-file for every building. The software offers a special tool Oxygen2 for creating, visualizing , and editing -p3d-files. Because we are interested in a possibly automatic export of building reconstruction results into VBS2, we decided to use the FBX file format as an intermediate step. This format, owned by Autodesk, provides interoperability between content creation programs and allows an easy creation of .p3d files by means of a VBS2-proper convertor. With respect to the representation of a building in VBS2, we differentiate between accessible buildings, which contains doors and windows that were detected and have to be modeled and animated, and non-accessible buildings. For modeling a non-accessible building, only a few things have changed from our previous contribution. For example, a standard texture for a wall, which was not observed in one of the input images, can be randomly selected between several images with textures typical for the region of interest. Also, it can be optionally rescaled for every single wall. Because the geo-positions of buildings, especially the height offset, are calculated depending on the original elevation map and because of modifications of the ground in VBS2 (see Section 3.1 and Section 3.2), it can happen that some buildings seem to be wrongly positioned (hover over or partly disappear under the ground). One solution is to put a base under the floor which helps to avoid gaps between the terrain and the building.

The main contribution of this work concerns representation of the accessible buildings. Since the available sensor data, described at the beginning of Section 2, does not allow modeling building interiors, our intent is just to model doors and windows. In order to understand how it works, it is necessary to understand the concept of LODs in VBS2. One can imagine the VBS2-specific Levels of Details (LODs) as different layers which characterize properties of the object (visual, animated, etc.). Thus, the VBS2-specific idea of LOD is somehow different from what we saw in Section 2. In the next paragraphs, we present the four relevant LODs, which are needed for modeling accessible building.

- *Resolution LOD*s, distinguished by indices starting from 0.000, 1.000, etc., are applied to display a VBS2 object from different distances. The lower the index is the more detailed is the representation of the model. In case of furniture the Resolution LOD 1.000 could be applied. This has the advantage that objects with many polygons are not modeled at their finest resolution while being observed from afar. For our research, only the resolution LOD 0.000 was used because only few accessible buildings (with indoor components) of the total scene (77 buildings) are modelled (see Figure 4a, red polygons). Due to this fact, the number of polygons is relatively low and the computing capacity does not suffer. In Figure 4a, the door and windows in the resolution LOD 0.000 are highlighted with red polygons.
- To specify collision detection, the *Geometry LOD* is employed. As a consequence, units cannot walk through walls (see Figure 4b, red polygons). Both Resolution LODs and Geometry LOD were already used in our previous studies [4] and they are needed to model both accessible and non-accessible buildings.
- *Memory LOD* is needed for animation and allows opening a door or a window. The animation is realized with the definition of window/door axis and an action point, which plays the part of a doorknob. A user at a certain distance from this action point can open and close the door and the window. Because the information about the door axis and the doorknob can hardly be extracted from the sensor data, we set the action point to be the center of gravity of the underlying polygon (see Figure 4c, blue position). The most vertical edge of the polygon is set to be the door/window axis (see Figure 4c, green positions). Depending on the point order, the doors/windows are opened inward or outward.
- *Roadway LOD* enables units to walk in an adequate manner within buildings, but also on roads, bridges or ladders. The Roadway LOD is similar to a single-sided plane with upwards pointing normal. It is applied on the object (see Figure 4d, red polygon). Additionally, depending on the roadway, sound effects can be included into the simulation, for example, a rattle sound while climbing the ladder. The sound effects are stored in scripts which have to be added to the configuration files. To get a better impression of the animation and check whether and how it works, a ladder was positioned at the building. A person can climb up the ladder after which the window can be opened, closed, and breached.

As mentioned before, the simulation in VBS2 is ensured by two configuration files. In the model configuration file, the animated doors and windows are defined including parameters such as the initial angles of the door/window at the beginning and at the end of the animation. All properties concerning the animation in action are defined in the second file. Within a predefined distance between the person and the doorknob, an action menu appears (see Figure 5). The possible actions are: Opening and closing window/door and breaching a window. Further animation properties, described in the second configuration file, are: Who is allowed to activate the animation, which sounds and special effects are added to the animation, etc. With the knowledge about all objects to be animated and additional conditions about the modality of animations, the configuration file is generated automatically. The associated .fbx-file is generated with the corresponding names of all animated objects and object components: The names of objects in the different LODs have to be consistent; otherwise the animation cannot be executed. For clarification, the opening/closing animation and the breaching animation are displayed in Figure 5b. Additionally, some furniture was placed into the building for a more realistic impression of its interior.

3.5 Creating missions for simulations

Once the geo-data basis has been exported into the simulation environment, we can add model instances, such as persons, vehicles, road signs, and obstacles into the mission. The object types are taken from the mission editor of VBS2. Besides having plenty of object types, it is rather comfortable to deal with VBS2 missions. The static objects, the blue-forces and op-forces, the civilians, as well as the events to be triggered can be stored into mission layers, and once such mission layers have been created, they can be reloaded after updating the geo data-basis.

The grade of detailed in a simulation terrain is not only a matter of the user's demands. The simulation system itself has to be able to handle the terrain data base fluently. Several presentations and discussions with military and police forces point to the understandable requirement of highly detailed and photo-realistic representation of the operational area or, at least, of the objective and its close surroundings. Especially with VBS2, we experienced a conflict between highly detailed geo-specific simulation terrain and its usability. With an increasing number of different buildings – each of them that stored in a separate .p3d-file – the performance of the simulation suffers. A reasonable trade-off we learned to apply is to generate a simulation terrain of moderate level of detail or moderate size and update it with spots of highly detailed content, such as accessible buildings.

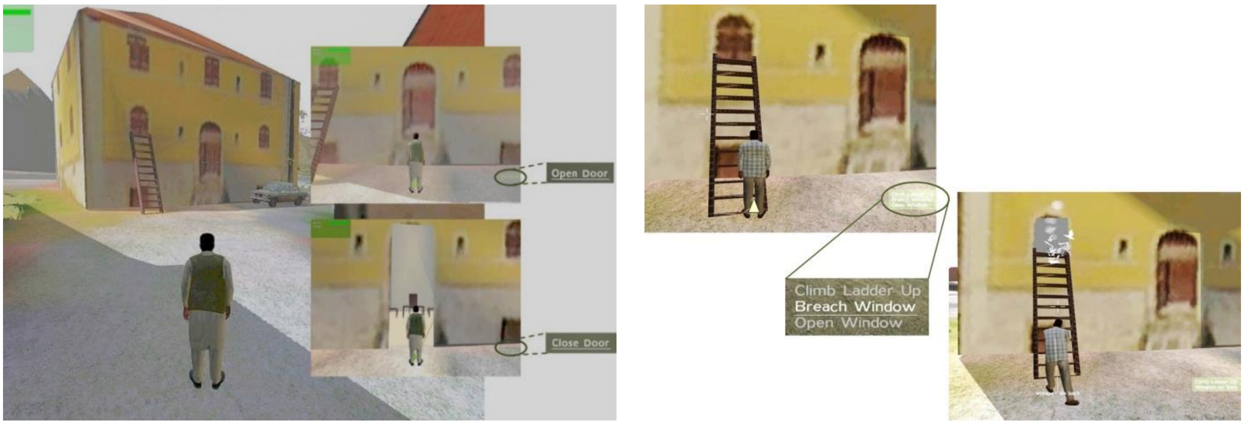
4. EXAMPLE OF A VBS2-MISSION

According to the roadmap proposed in Section 3, we created a virtual model (shown in Figure 6) of Bonnland, one very commonly used (by army, police, frontier guard, fire service, and many others) urban training site in Germany. An example exercise was created in the Release 2.12 of VBS2 in order to discuss our results with personnel from army and civil security authorities. Although not reaching the level of photorealism a hand-made simulation terrain can show, our Bonnland model usually was recognized also by people who had a short stay at the training facility, or even merely have seen it at pictures. Considering the relatively small extent of the virtual training site, example missions are focused on the core area of an urban operation, such as taking the objective street by street or securing a checkpoint around the main traffic junction. The mission we used to illustrate the presentation of our work in this paper is inspired by counter insurgency operation in urban terrain: A cell of insurgents operates a local base with communication systems, supplies and weapons cache in Bonnland, as shown in Figure 7. A special police unit has to take out this local base without inflicting much damage around the object. Mission simulation like this would hardly have been possible without accessing building models. It has been an amazing experience for us to learn how intensive and complex a task can be under those circumstances. From our point of view, even with rather simple building interiors, incorporating accessible buildings in a virtual training mission may multiply the merit of virtual simulation.

5. CONCLUSIONS AND FUTURE WORK

Because of cost and time reasons, automatic modeling of urban terrain for simulation purposes has become a very important issue in the military and civil applications. For rather small units, which must orient themselves in an unknown urban terrain, both the 3D aspect of the simulation and the recognizabitity of the 3D objects play a crucial part. While detection of building and vegetation, outlining buildings, and estimating shapes of trees is possible from the elevation map and the orthophoto, oblique UAV imagery can be used to obtain textures of building walls at a desired resolution. Together with some external data (shape-files to the major part), the ground, building, vegetation, road maps and water-bodies can be exported into the combat simulation system [11].

Additionally, advanced technologies allow a high-quality façade analysis. This reflects in a logical wish to model façade details, such as doors and windows, and animate them within our simulation environment. The process of creation of such an animated simulation described in this work is fully automatic and easily extendible by analogous object/animation types.



a b

Figure 5. Animation in VBS2: a. Opening and closing a door; b. Breaching a window

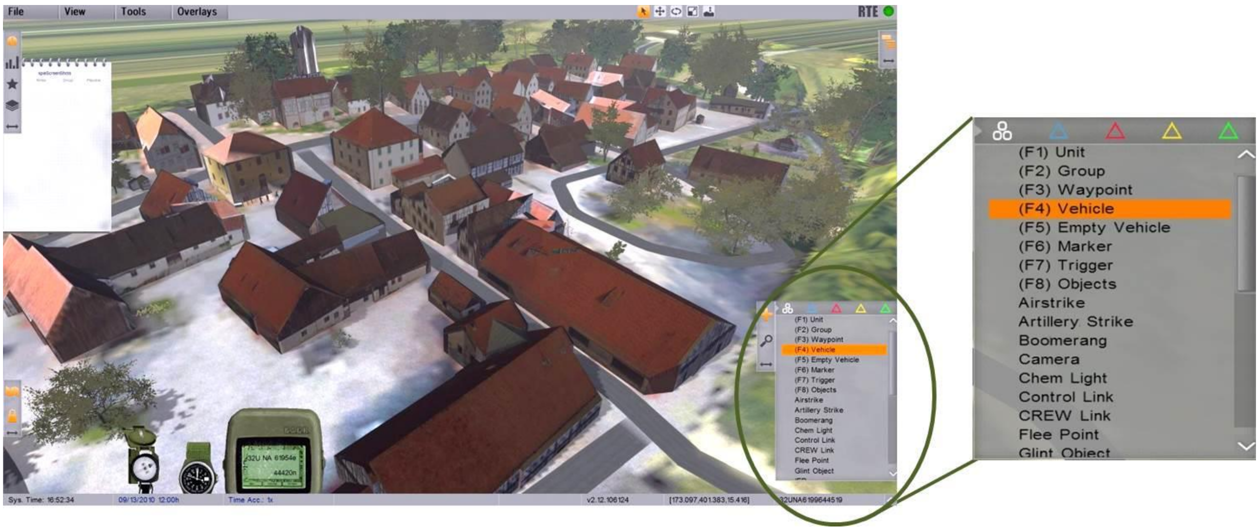


Figure 6. A view of the virtual theater of operations in the mission of Section 4, generated by the mission editor window.

a b

Figure 7. Example mission in the virtual training site of Bonnland: a. insurgents surrender and lay down their weapons; b. captured and handcuffed insurgents in front of an accessible house

Our experience with VBS2 has shown that the capability for automatic extraction of any topographic object type has a consequence that the capability for its automatic conversion into the simulation is rapidly acquired. Conversely, this means that the algorithms can be rapidly adjusted for a different visualization or simulation software.

The area of short-term improvements of the quality of simulation is given by a more plausible indoor modeling in VBS2, for example, estimation of number of floors from sensor data. However, the most important direction of future work is to use multispectral data for material analysis and thus, to provide an enhanced urban terrain reconstruction and representation. Also here, the advances in sensor technology result in a larger amount of high quality data that can be captured from light-weight sensors. Covering building polygons with real, multi-sensorial texture, handling transparencies and simulating materials will increase the quality of our 3D models and make them more interesting for a larger number of applications.

ACKNOWLEGDEMENT

We express our deep gratitude to the research team of IfP-Stuttgart for providing us the necessary input for data-set Bonnland (DSM and digital orthophoto). Automatic generation of a .fbx-file from reconstruction results was carried out by Moritz Liebelt.

REFERENCES

[1] Ali, H., Seifert, C., Jindal, N., Paletta, L. and Paar, G., "Window detection in facades," 14th International Conference on Image Analysis and Processing, (ICIAP 2007), 837-842 (2007).
[2] Becker, S. and Haala, N., "Refinement of building fassades by integrated processing of LIDAR and image data," International Archives of Photogrammetry, Remote Sensing and Spatial Information Science, 36, 7-12 (2007).
[3] Becker, S., "Generation and application of rules for quality dependent façade reconstruction," ISPRS Journal of Photogrammetry and Remote Sensing, 64, 640-653 (2009).
[4] Bulatov, D., Solbrig, P., and Wernerus, P., "Ad-hoc model acquisition for combat simulation in urban terrain," Conference on Earth Resources and Environmental Remote Sensing/GIS Applications, Edinburgh, United Kingdom, 85380G, 10 p (2012).
[5] Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P., and Wernerus, P., "Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks," ISPRS Journal of Photogrammetry and Remote Sensing, 93, 157-170 (2014).
[6] Bulatov, D., Wernerus, P., and Heipke, C., "Multi-view dense matching supported by triangular meshes," ISPRS Journal of Photogrammetry and Remote Sensing, 66(6), 907-918, (2011).
[7] Bulatov, D., Solbrig, P., Gross, H., Wernerus, P., Repasi, E., and Heipke, C., "Context based urban-terrain reconstruction from m(uav)-videos for geo-information applications," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Proc. Conf. Unmanned Aerial Vehicle in Geomatics, Vol. XXXVIII-1/C22, ISSN 1682-1777 (2011).
[8] Douglas, D. H. and Peucker, T. K., "Algorithms for the reduction of then of points required to represent a digitized line or its caricature," Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2), 112-122 (1973).
[9] Haala, N. and Rothermel, M., "Dense multiple stereo matching of highly overlapping UAV imagery," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XXXIX(B1), 387-392, Commission I, XXIIth ISPRS Congress (2012).
[10] Meixner, P. and Leberl, F., "Characterizing building facades from vertical aerial images," International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 38, 98-103 (2010).
[11] Virtual Battle Space 2, http://products.bisimulations.com/products/vbs2 (20.08.2014).