Very fast road database verification using textured 3D city models obtained from airborne imagery

Dimitri Bulatov*, a, Marcel Ziems b, Franz Rottensteiner b, Melanie Pohl a

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

b Institute for Photogrammetry and GeoInformation, Leibniz Universität Hannover, Nienburger Str. 1, 30167 Hannover, Germany

ABSTRACT

Road databases are known to be an important part of any geodata infrastructure, e.g. as the basis for urban planning or emergency services. Updating road databases for crisis events must be performed quickly and with the highest possible degree of automation. We present a semi-automatic algorithm for road verification using textured 3D city models, starting from aerial or even UAV-images. This algorithm contains two processes, which exchange input and output, but basically run independently from each other. These processes are textured urban terrain reconstruction and road verification. The first process contains a dense photogrammetric reconstruction of 3D geometry of the scene using depth maps. The second process is our core procedure, since it contains various methods for road verification. Each method represents a unique road model and a specific strategy, and thus is able to deal with a specific type of roads. Each method is designed to provide two probability distributions, where the first describes the state of a road object (correct, incorrect), and the second describes the state of its underlying road model (applicable, not applicable). Based on the Dempster-Shafer Theory, both distributions are mapped to a single distribution that refers to three states: correct, incorrect, and unknown.

With respect to the interaction of both processes, the normalized elevation map and the digital orthophoto generated during 3D reconstruction are the necessary input – together with initial road database entries – for the road verification process. If the entries of the database are too obsolete or not available at all, sensor data evaluation enables classification of the road pixels of the elevation map followed by road map extraction by means of vectorization and filtering of the geometrically and topologically inconsistent objects. Depending on the time issue and availability of a geo-database for buildings, the urban terrain reconstruction procedure has semantic models of buildings, trees, and ground as output. Buildings and ground are textured by means of available images. This facilitates the orientation in the model and the interactive verification of the road objects that where initially classified as unknown. The three main modules of the texturing algorithm are: Pose estimation (if the videos are not geo-referenced), occlusion analysis, and texture synthesis.

Keywords: building reconstruction, Dempster-Shafer Theory, road database, texturing, urban terrain, verification.

1. INTRODUCTION

The significance of realistic and context-based urban terrain models has been demonstrated in numerous applications such as automatic navigation, urban planning, virtual tourism, disaster management, as well as security-related rapid response tasks. More than the others, the latter application, which remains in focus of this contribution, is time-critical. For certain disasters, like earthquakes or floods, up-to-date knowledge about the terrain including its recent and perhaps sudden changes is strongly essential. Especially the information about roads is important since they are actually the basis for organizing emergency services: Evacuation of people, fire-fighting operations, and many others.

Because of this reason, mapping organizations all over the world spend high efforts to keep their road databases up-to-date. This process is mostly carried out by a regular manual comparison of the database contents with recently available
aerial or satellite images. However, sufficiently educated personnel are not always available to perform the verification job for large areas. Therefore, in case of crisis events and other rapid response applications, immediate, possibly automatized updates for specific areas are required to facilitate the crisis management.

Since in the majority of cases, the set of road objects after a disaster is a subset of the previous set, the main task should be a verification procedure; that is, screening the imagery in order to identify just a few database errors. To do this, several independent modules are applied on each road object. We refer to [19] and [28] as a survey about road verification algorithms. The necessary input for such verification methods are – additionally to the initial database entries – the normalized elevation map (also denoted by (n)DSM, (normalized) digital surface model) and a digital orthophoto.

![Figure 1: Overview of our verification procedure.](image)

In few remaining cases, the geo databases are too obsolete, too defective, such that many road objects are missed or not available at all. For that scenario we propose another module based on aerial/UAV images with a resolution around 0.1 m, the process road detection is a typical classification task – within a reconstruction procedure – followed by a vectorization module.

Figure 1 makes it clear that our algorithm consists of two processes which exchange some intermediate results, but actually run independently from each other. These processes are: Geometric 3D reconstruction from images as well as verification of the available road geo-databases. There are two possible interactions between both processes. First, we need dense 3D information in order to be able to use a bunch of modules for verification. Pixel-based estimation of depth maps is used to create a DSM. We subtract the digital terrain model (DTM) from the DSM, thus obtaining the normalized elevation map. These intermediate results are needed to perform the optional detection and the verification of roads.

The second common point of two processes concerns interactive verification. The Dempster-Shafer framework explicitly expresses uncertainty and ignorance, and thus allows dealing with complex situations that are usually not expected by any of the road models that were developed so far. In the case of ignorance, a human operator must assess, by means of sensor data and/or results of sensor data processing, whether such an object is indeed correct or incorrect. It has turned out that textured 3D urban terrain models can significantly accelerate the process of interactive verification and decision-making. Therefore, efforts should be made either to provide an up-to-date texturing of the 3D models, if they are already stored in geographic databases, or to perform an automatized reconstruction of the 3D urban terrain models from sensor data followed by the texturing. Once again, this second possibility is particularly important if the databases are obsolete.
as a direct consequence of the disaster, or not available at all. However, the time needed to perform an accurate 3D reconstruction of urban terrain is considerably higher with respect to that needed for verification of road structures.

The paper is structured as follows: Section 2 is dedicated to the road geo data verification process (the left part of Figure 1), while the reconstruction process is visualized on the right of Figure 1 and described in Section 3. We will put more attention to the texturing procedure than to the reconstruction of the geometry and classification because we suppose that a sufficiently reliable geo database is available. We demonstrate the performance of our method in Section 4 and summarize the main ideas of the contribution in Section 5.

2. ROAD DATABASE VERIFICATION

In order to overcome the deficits and restrictions of existing methods, the proposed method for road database verification combines several state-of-the-art road detection and road verification approaches that can deal well with different road types, context areas and geographic regions. Each road detection or verification approach is realized as an independent module representing a unique road model and a specific processing strategy. The partially incomplete and possibly contradicting solutions from the modules are combined on the basis of a statistical reasoning approach (cf. Section 2.1) that explicitly includes the possibility that a road object cannot explained by any model. The properties of the implemented verification modules will be briefly described in Section 2.2.

2.1 Statistical reasoning

Each verification module provides two probability distributions per road object.

1. The verification output: $P : \Theta_1 \rightarrow [0,1]$ with $\Theta_1 = \{\text{correct}, \text{incorrect}\}$.

2. The model-uncertainty output: $P : \Theta_n \rightarrow [0,1]$ with $\Theta_n = \{a, na\}$.

The verification output has the form of a probability distribution and provides a solution to the basic problem, according to which a road might be either correct or incorrect. The model-uncertainty output also has the form of a probability distribution and refers to the two possible states of the underlying model that might be either applicable (a) or not applicable (na). Since the road model and the detection strategy of each verification module are rather specific, the verification outputs are not useful in a general sense. Hence, the verification output is conditioned to the applicability of the underlying road model, which leads to the following interpretation of what each module provides by design:

$$P(\text{road = correct} | \text{model} = a) = 1 - P(\text{road = incorrect} | \text{model} = a).$$

In order to find a general solution for the verification problem, the conditioning on the state of the model has to be removed, which leads to:

$$P(\text{road} = \text{correct}) = P(\text{road} = \text{correct} | \text{model} = a) \cdot P_a + P(\text{road} = \text{correct} | \text{model} = na) \cdot P_{na},$$

where $P_a$ and $P_{na}$ denote the probabilities $P(\text{model} = a)$ and $P(\text{model} = na)$, respectively. In Equation (4), the second summand is unknown because $P(\text{road} = \text{correct} | \text{model} = na)$ must be considered as a solution for the given classification problem without having a (known) relation to that problem. However, with $P_{na}$, the probability that (4) is unknown is given. This circumstance allows to map both probability distributions (1) and (2) to a single probability mass distribution. According to [5], the corresponding probability mass function for the given problem has the following form:

$$m : 2^0 \rightarrow [0,1] \text{ with } 2^0 = \emptyset, \text{correct}, \text{incorrect}, \text{correct} \cup \text{incorrect}.$$  

The new state $\{\text{correct} \cup \text{incorrect}\}$ explicitly expresses the lack of capability of a module to discriminate the states $\{\text{correct}\}$ and $\{\text{incorrect}\}$. The corresponding probability masses are determined as follows:

$$m(\text{road} = \text{correct}) = P(\text{road} = \text{correct} | \text{model} = a) \cdot P_a,$$

$$m(\text{road} = \text{incorrect}) = P(\text{road} = \text{incorrect} | \text{model} = a) \cdot P_a,$$

$$m(\text{road} = \text{correct} \cup \text{incorrect}) = P_{na} \text{ and } m(\emptyset) = 0.$$
Consequently, each verification module provides a probability mass distribution for each road object. The solutions from the available modules are combined on the basis of Dempster’s combination rule [5], which leads to a single probability mass distribution per road object:

$$m(A) = \frac{\sum_{\cup A \in \mathcal{B}} \prod m_i(B_i)}{\sum_{\cup A \in \mathcal{B}} \prod m_i(B_i)}$$

(9)

The final distribution expresses all the evidences based on different road models, and thus builds the statistical basis for the decision making. In accordance with maximum a posteriori (MAP) criterion in the Probability Theory a decision $D$ is determined by:

$$D = \arg \max_{A \in \mathbb{D}} m(A).$$

(10)

The presence of the rather uncommon decision $D = \text{correct} \cup \text{incorrect}$ is a particular property of the proposed approach, and plays a key role for the human-machine interface, described in Section 3.

2.2 Verification modules

The underlying models are simple as they always focus on just a single road property, and thus do not require comprehensive parameterization. The models have heuristic and statistic backgrounds as well. Each module receives an input data package for each road object to be verified and provides the previously described probabilities, see Equation (1) and (2). An input data package consists of the road object geometry, the nDSM and the orthophoto that show the local surroundings of the road object. Subsequently, the eight applied verification modules will be introduced by naming the corresponding state-of-the-art approach and main idea of the underlying road model. Furthermore, the principle of defining the verification and the model-uncertainty outputs will be briefly described without going into the details of those probabilistic classifiers, for which we refer to [28].

The 2D- and 3D-line modules: The underlying road models both correspond to the basic model described in [25]. This approaches are based on panchromatic images with a resolution of about 1 m, where roads are modelled as lines of more or less constant brightness and width. The maximum curvature and the width of the lines are defined on the basis of heuristics referred to the expected road type and the image resolution. While the basic model turned out to be very useful for rural areas, [14] showed that the model can be transferred to nDSM, which makes them particularity useful for dense urban context areas. In order to determine the states of the roads (correct, incorrect), such detected lines are compared with the database geometry; see [10]. Line-based models generally assume homogeneous background, and thus were applied for rural context areas (2D-line module) or urban context areas (3D-line module). However, the proposed approach does not have such restrictions. Therefore, the assumption is checked on the basis of the image entropy computed besides the road object. The main idea is that only low image entropies indicate large values of $P_o$.

The parallel edge-detection module: The underlying model of this module focuses on major roads in rural and suburban areas. Analogous to [13], road borders are modelled as pairs of anti-parallel edges. The spacing between those edges is based on heuristics. According to the line verification modules, the detected edge pairs are compared with the geometry stored in the road database. The model assumes that roads are the only objects showing such anti-parallel edges. The assumption is checked on the basis of the frequency of other objects in the local background that match with the model. The basic idea is that low frequencies indicate large values of $P_o$.

The acupuncture module: The underlying model of this module focuses on roads in urban context areas. Analogous to [27], roads are modelled as regions that are less frequently intersected by edges than the local background, and if an edge intersects a road, it is expected to have the same direction as the road. Therefore, several profiles are defined in the local surrounding of a road object. Then the intersections of these profiles with image edges are counted: a road object is classified as being correct if it has the lower intersection count than the other profiles in the local surrounding. One problem with this model is that roads are expected to lie between buildings that are aligned in rectangular pattern. As this
is not generally true, the assumption is checked on the basis of the frequency of edges in the local surroundings that can be associated with buildings. The basic idea is that high frequencies indicate large values of $P_w$.

The color module: In accordance with [9], this model assumes that image regions belonging to roads can be identified on the basis of their color properties that can be specified in advance. A support vector machine (SVM) is learned on the basis of training samples in order to solve the two-class problem \{road, non-road\} in an extended RGB color space. A road object is classified as correct if the associated image region is classified as road. The implicitly defined color model assumes the training data to be representative, which is only realistic if the closed world assumption is realistic. As in field of remote sensing this assumption is generally critical, it is explicitly checked using the feature space distance between a test sample and the given training samples. Therefore, a one-class SVM is learned on the basis of the same training samples as the two-class SVM and the feature space distance of a test sample to the decision surface of the one-class SVM is determined. The basic idea is that negative and small positive distances indicate large values of $P_w$.

The SSH module: The underlying model of this module focuses on narrow roads in rural area. Analogously to [9], roads, in contrast to their local background, are modelled as image regions with specific texture. Therefore, gray-value histograms corresponding to image regions in the local surroundings are computed and then compared to each other. A road object is classified as correct if the associated histogram has the largest integrated Bhattacharyya distance from the other histograms. As the model assumes the local background having homogenous texture, which is not necessarily holds, the assumption is checked on the basis of the standard deviations of all Bhattacharyya distances. The basic idea is that low standard deviations indicate large values of $P_w$.

The building and vegetation modules: In accordance with [21], the underlying model of this module assumes that the object types roads, buildings, trees, and grassland are disjunct, i.e. they cannot exist at the same place in the image. The object detection for buildings and grassland is carried out as described in Section 3.3. If a road object stored in the database overlaps with a building or grassland to a certain degree, it is classified as incorrect. The applicability of the detection algorithms depends on the standard deviation of the gray values of the contradicting pixels in relation to their absolute values. Hence, relatively low standard deviations indicate large values of $P_w$.

3. TEXTURED MODEL RECONSTRUCTION OF URBAN TERRAIN

The goal of this section is to facilitate the interactive verification of the road segments with the status unknown or incorrect. A more reliable verification can be performed by an external operator if a detailed knowledge about the 3D structure of the scene is available together with the textured views of the regions to be examined more closely. We again consider first a situation of an available 3D model of buildings belonging to the area and are interested in fast possibility for texturing the building polygons by means of the available imagery, in particular, images and videos obtained from micro UAVs. Secondly, if there is no such a database or it is obsolete, urban terrain reconstruction from sensor data must be carried out within a reasonable time. Also here, two sub cases are possible. We assume in this work that the measure of destruction is moderate and so, we can make an assumption on properties of buildings, such as they can be approximated by several large polygons. Plenty of work has been done with respect to obtaining geo-databases from the sensor data [2][16], see also [22]. Since textured model reconstruction does not represent the main focus of our contribution, we only will give a brief overview of the procedures of reconstruction and texturing in Sections 3.1 and 3.2, while the reader is referred to [2] for more technical details. Finally, we describe in Section 3.3 our approach for road detection in DSMs.

3.1 Generation of DSM and obtaining geo-databases from sensor data

In absence of the reliable geographic data, we must be able to obtain a recognizable geometry by a context-based urban terrain reconstruction from sensor data. Ideally, a DSM is sampled from an airborne laser point cloud. However, the equipment for laser techniques heavier and more expensive than that needed to obtain aerial images and even UAV-videos which can be subject to a frame decimation routine [20] and oriented used a standard procedure for structure from motion and bundle adjustment, such as [23]. Then, depth maps are computed from overlapping pairs of images, resulting in a dense point cloud [12]. Finally, this point cloud is resampled into a DSM. Doing so, some important details, such as the exact position of walls, may get lost, but we consider such a 2.5D-based approach more feasible because at least a reliable bound can be obtained from a building in all directions: upwards by the roof structures, to the sides by the building outlines and downwards by the ground map.
In other words, our task is to obtain the ground map, also called DTM, the building outlines, and the roof structures. The first step is to extract the ground. This is done by detecting several ground points and approximating them by means of a smooth surface. The difference of DSM and DTM delivers elevated regions from which we extract vegetation (for example, by means of the Normalized Differenced Vegetation Index (NDVI) or the Lineness Measure [2][3]. From the remaining regions, we delete those that are too small (less than 2 m²) in order to suppress false alarms (traffic signs etc.). Furthermore, delineation of those regions that are connected by narrow structures or exhibit large differences of elevation may optionally be performed.

The successive procedure of building reconstruction aims to obtain the outlining contour and the roof structure of every building. For estimation of the outlining polygon, we first employ a test of rectilinearity by assessing straight lines within the mask. Their discretized directions are sampled into a histogram and the entries of the histogram are weighted by lengths of these lines. If the quotient second-best-to-best of the entries is below 0.5, the building is defined to be rectilinear, otherwise it is not rectilinear. To obtain the contour of a rectilinear building, the approach of [11] is used: Starting from the height-thresholded binary mask of the minimal boundary rectangle of the building, the contours are refined, for each blob, by recursive adding and removing rectangular subparts until the area of the remaining blob lies under a threshold of 2 m². For a non-rectilinear building, a contour-tracing algorithm is applied. A step of generalization allows deleting segments that are too small and reducing the number of vertices in the contour polygon.

The second step is dedicated to the roof details analysis and consists of two main tasks: Dominant planes extraction and roof planes polygonisation. We use a global algorithm for dominant planes extraction (e.g. RANSAC) for obtaining plane hypothesis and assigning 3D points into one of the dominant planes. After some preparations (morphological operations, filtering), those inlier sets that form connected regions are polygonized using a contour-tracing algorithm. In the next step, pairs of planes neighboring in 2D are intersected between each other. If the resulting 3D line lies near the polygons corresponding to both planes, the outline segment is added to the list. Otherwise, we check if there is a stepline: Lines in DSM are computed by means of [4]. The initial values of polygon vertices are updated by projecting onto the cutlines, steplines and building outlines. An additional module eliminates the potentially occurring twists. After all polygons have been updated, uncovered details of the roofs are filled in a reasonable way. Walls are modeled by vertical trapezoids; one of its arms connects two neighboring vertices of the roof polygon and the other one connects their vertical projections onto the ground.

3.2 Texturing

In this section, we deal with building models which usually consist of polygons embedded in 3D space. Often, these polygons may be rather large and extend over several input images. This is why e.g. El Hakim [8] proposes to create a separate, synthetic picture from every polygon and fill it by contents from the input images. To do this, the image should be first localized in the model coordinate system. Afterwards, occlusion analysis is used to extract the foreground for every pixel of an input image. Then, all input images are used to assess polygons; that is, which polygons should be textured by which images. The last step is called texture synthesis: We should determine the color of all pixels of the synthetic image given a subset of images with corresponding foreground information and orientations. In the following paragraphs, we describe our approach for the four previously mentioned steps and the main challenges that still remain to solve.

Pose estimation: In absence of a precise orientation, delivered by an internal navigation unit, registration of an image into the model coordinate system, also called pose estimation, is not a trivial task. If an approximation of the pose and a coarse knowledge about the radiometry of the model is available, the initial pose is refined by establishing correspondences of the input image and a synthetic model view. In [1], the coarse information was given by reflectance values a terrestrial laser point cloud, but currently experiments with libraries of images previously registered are being carried out. The method for establishing point correspondences in the synthetic view and in the input images is one of the state of the art methods [24]. Since the points in the points in the synthetic model view are already given 3D coordinates, pose estimation follows by means of the RANSAC method with PnP [17] core function. Given a sequence of images – for example, an UAV movie – the relative orientation within a local coordinate system can be determined by a state-of-the-art structure from motion approach [23]. Once the registration of the first image by means of the technic described above is carried out, the relative rotation and the relative translation between the local and the model coordinate systems can be calculated. The remaining unknown is the global scale which can be computed from the most frequent quotient of depths: Since a structure-from-motion approach delivers 3D points, their depth values can be calculated and compared to those resulting from the corresponding 3D points in the model coordinate system. Since the parameter of global scale is very important for the quality of texturing, the estimated motion is usually corrected by registering the last frame of the
subsequence into the model coordinate system and applying a trajectory bending approach [7]. This helps to compensate for the drift in the camera motion during determination of the relative orientation.

**Occlusion analysis:** As mentioned above, for every polygon $P$, we have a synthetic image $J = J(P)$. We add to $J$ an additional channel, which is called mask and denoted by $M$. It is 0 if the corresponding 3D point lies outside of the polygon, and 1 when it lies inside the polygon. This mask is computed only once for every polygon and it accelerates foreground estimation. For an input frame $F$, we initialize the synthetic depth map $D_s$ to be infinity, and the foreground map $S$ to be minus one. For a further acceleration, the polygons are sorted by distances of their centers of gravity to the corresponding camera center and only those polygons are considered at least one vertex is projected into the image domain of $F$, is in front of the camera, and is not occluded. For such a polygon, all pixel of the bounding box of its projection into $F$ are considered. For those of them, for which the projection into $M$ is one and for which the depth is below the initial value of $D_s$, the values of $D_s$ and $S$ are updated. An additional input is given by the depth maps $D$ obtained by measurements, either during the registration procedure or during the photogrammetric procedure of depth map extraction. Given a successful registration, larger region for which $D_s$ is smaller than $D$ allows to detect foreground objects (e.g. persons or trees in front of the building walls), to correct the foreground map $S$, and thus to clean façade textures. Regions for which $D < D_s$ may correspond to the insufficiencies in the model.

**Assessment of polygons:** The procedure described above is carried out for every input image $F_i$. Thus, we have measured the area $a_i$ of the projection of $P$ into $F_i$ as well as the number of non-occluded pixel $b_i$, which is obtained from the histogram of values in $S_i$. We define the energy function

$$E_i = E_i(P) = -a_i b_i,$$

and we wish to use frames with a low value of $E_i(P)$ for texturing $P$. Note that $a_i$ depends on the inclination angle of the principal plane of $F_i$ to the normal of $P$ and the resolution of $P$ in $F_i$.

**Texture synthesis:** In the last step, we have to fill the synthetic image $J(P)$ by the contents of $F_i$ with low values of $E_i(P)$. We implemented a greedy texturing procedure. First we take the value of $k$ with the minimum value of $E_i(P)$. All pixels of $J$, for which $M$ is true the background map $S_i$ corresponds to $P$, are textured by $F_k$. These pixels are deleted from the list of pixels of $J$ to be textured and the second-best frame $F_k$ is used to texture them. This process repeats until there are no more pixels to be textured or no more frames with sufficiently low values of $E_i$ left. In the latter case, the color values of the already covered regions of $J$ are propagated for the remaining pixels. This is done by a morphological dilation and by extracting the most frequent color from the color histogram.

Such a simple, greedy texturing procedure has two disadvantages: Firstly, we do not consider deviations in luminance and radiance which emerge due to the fact that input images are taken from the different points of view, with different resolutions or even different spectral ranges. Theoretically, it is possible that one part of a building wall is only observed by a daylight camera and another part by an infrared camera. Any attempt to equalize the appearance of the polygon (for example, by considering linear combinations of $F_k$), leads to an increased computational time. Secondly, we do not encourage neighboring polygons to be textured by equal or neighboring frames, because our polygons are already large enough. This could be achieved by considering neighboring relations of polygons as a Markov Random field and extend the $E_i(P)$ from (11) by a smoothness term. One of the methods of [26] can be used for the non-local minimization of the resulting energy function.

### 3.3 Detection of new roads

While the previously described modules only focus on existing road database entries also those that are completely missing in database are of interest, e.g. in order to define alternative emergency routes. The process of detecting new road consists of two steps, classification of road pixels and vectorization.

**Classification:** We use a modification of the unsupervised classification approach of [16]. They first construct energy terms from properties (elevation, planarity, singularity) of the pixels in the DSM, to which we add further properties of the pixel in the orthophoto, such as: NDVI with green channel, saturation, and entropy. According to these properties, pixels belong to one of the four main classes: Building, tree, grass, and road. For example, roads are non-elevated, rather planar objects, usually with a high saturation, low NDVI-measure and also rather low entropy. Instead of truncated linear term describing energy, we use a sigmoid function, because it allows bounding the terms away from zero and one, and, consequently, a non-zero probability of risks. However, similar to [16], we made an experience that the results are very sensitive on the choice of parameter values. Hence, parameters denoted by $\sigma$ in [16] should be set iteratively or trained in
advance. The energy values for every class are input for a non-local optimization [26] which allows, similarly to depth-map extraction, to encourage neighboring pixels to belong to the same class. We found out that the optimization algorithm of [15] represent a good trade-off between computation time and quality of the classification result.

Even with a high smoothness parameter, the binary image representing the class “road” is subject to strong morphological preprocessing for the subsequent vectorization step. Depending on the given image resolution and the accuracy we wish and are able to obtain, we further suppress all regions of an area corresponding to 5-10 m of the binary image and its negative.

**Vectorization:** After skeletonization of the prepared binary image, vectorization of a skeleton is obtained by means of the Steger-operator [25], followed by generalization with [6]. These two operations have an additional output: Junctions, which allow establishing important topological properties. Every putative road end-point should be incident with at most one junction. Though in real life, a junction is incident with at least three road objects, we additionally allow – at first stage – junctions of two roads, because of possible numerical errors (for example, two ends of road objects too close two each other), we may introduce additional junctions. These junctions are defined to be active. We also compute the geometric properties of every road segment: Its length, main direction, and approximate road width. The single width values are obtained by means of the classification result at each vertex of the polyline. The approximate width is computed as a weighted sum of single width values. The geometry of the putative road object is defined to be reliable if the width is bounded between two thresholds (around 1 m and 30 m) and the length exceeds 5 m. The topology of the putative road object is defined to be reliable if both ends of this object are incident with an active junction, that is, a junction with at least two incident roads.

Now we are ready for an iterative filtering procedure of road objects. All road segments which are not reliable both with respect to the geometry and topology are set inactive. All junctions with less than two active incident roads are set to be inactive. Then, topology of the active road segments is re-computed and the procedure begins again. The remaining active road segments represent the output of our procedure.

4. EXPERIMENTS

We consider a data set from a village Bonnland in Southern Germany with an area of approximately 100 mx600 m. Extraction of the DSM and orthophoto is achieved by means of [12] while the road database, containing 16 road objects is acquired from an open source\(^1\). As the result of verification, eleven correct, two incorrect and three unknown road segments were obtained. As one can see in Figure 2, top, the main reason why the roads were classified as incorrect is its closeness to buildings or vegetation (see Section 2.2). The road objects marked in yellow did not match with any of the introduced road models, and thus were classified as unknown.

Additionally to the road database, we performed verification for the road segments detected by the method of Section 3.3. We depicted them in Figure 2, bottom. Keeping in mind that this procedure is supposed to detect merely new segments, we can delete all those that are too close to the database result. We see that on the one hand, many escape ways and secret paths (between the buildings) have been detected, which may be suitable for certain application. On the other hand, the incorrect segments are mostly too thin (seven roads that have a width below 3 m) while in one case, an actually correct road segment has not passed the acupuncture test. The road segments with status unknown exhibit too high curvature or are too short compared to standard GIS roads. Thus, none of the applied modules yields a decisive statement. We summarize all verification results in Table 1.

The incorrect and unknown road objects can be interactively verified by means of a textured model, as presented in Figure 3, bottom. The ground and building roofs are textured by means of the orthophoto and the building walls are textured by means of the UAV images like those atop of the figure. We clearly see how a red segment crosses a building model and vegetation region and the insufficient accuracy of the horse-shoe shaped yellow segment. Despite a relative good performance of the road detection approach (actually no false positives), the relatively high number of the yellow segments justifies the interactive verification by a simple view at a reconstructed and textured model.

---

\(^1\) www.openstreetmap.de
Figure 2: Orthophoto of the Bonnland data set generated by means of [12] together with the verification results. Correct, incorrect and unknown road segments are marked in green, red, and yellow, respectively. Top: Road segments from a database. Bottom: Results of automatic detection with differentiation between wide and narrow roads.

Figure 3: Top: Three UAV videos used for texturing building walls. Below: Textured model view with verification results from the database.
5. CONCLUSION AND OUTLOOK

Extracting and verification of road data by means of images is known to be a tricky task. Since we have a high variety of road models, which correspond to different geometric and radiometric appearances of roads, there are certainly more counter-examples to every single rule as there may be in the case of buildings and vegetation. Superposition of these models according to the Dempster-Shafer theory and evaluation of applicability of every models is has proved to be a promising strategy for road database verification. From elevation data, semantic 3D reconstruction of the scene is performed and the available images are used for texturing, which allows an interactive clarification in the most difficult spots, where automatic part suffers because of its restricted model complexity. The future work is mainly directed to the detection of new roads, which turns out to be the most difficult part because the prior knowledge used for the verification of existing database entries is missing. Here, we plan to reduce the amount of false alarms by improving the classification algorithm and considering alternatives to skeletonization.

Table 1: Results of road data verification with the number of not too thin (wide) road segments in parenthesis. See text and Figure 2 for further comments.

<table>
<thead>
<tr>
<th></th>
<th>Total number</th>
<th>Correct (wide)</th>
<th>Incorrect (wide)</th>
<th>Unknown (wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>16</td>
<td>11 (-)</td>
<td>2 (-)</td>
<td>3 (-)</td>
</tr>
<tr>
<td>Detection</td>
<td>95</td>
<td>29 (26)</td>
<td>8 (1)</td>
<td>58 (39)</td>
</tr>
<tr>
<td>Detection without database</td>
<td>15</td>
<td>15 (13)</td>
<td>6 (1)</td>
<td>28 (17)</td>
</tr>
</tbody>
</table>

ACKNOWLEDGEMENT

We express our deep gratitude to Gisela Häufel for preparing the shape-files for the data-set Bonnland as well as the research team of IfP-Stuttgart for providing the necessary input for data-set Bonnland (DSM and digital orthophoto).

REFERENCES


