

Frist, we would like to thank the reviewer for his insightful comments that indeed helped shaping the manuscript. We have carefully and thoroughly revised the submission, and we believe that we have addressed all comments. To meet the required revision, we have taken all raised issues seriously, which led to modifications in writing.

With additional literature and additional explanation, we have given answer on your comments.

For the comment “Many algorithms have been used in this Paper without clarifying about its reference and efficiency (e.g.: 256 : quick automatic algorithms” , answer is from line 281 until 305:

The georeferenced and calibrated flight lines are merged and processed to filter out the low points, aerial points, false echoes, etc. Due to physical, software and hardware constraints, found also on the last generation computers, related to the management and elaboration of a finite number of points, the cloud of points was splitted in square blocks with a side of 250 m. The first classification was performed using quick automatic algorithms, performed in TerraScan¹. Classification in TerraScan offers a number of classification routines that effect or rely on points or a group of points, or routines that can be used only in macros. During laser data processing the following classes are defined: Ground, Vegetation, Low Points, Roofs of buildings and Default.

Ground routine classifies ground points by creating a triangulated surface model iteratively. There are two routines, one that is best suited for classifying ground in airborne laser data sets and in data sets where there is mainly natural terrain, and second that is best suited for classifying ground where there is mainly hard ground surface, such as paved roads or other areas. We have used both, by drawing polygons around natural/paved areas in order to apply one inside the polygons and the other routine for the rest of the data.

Below surface routine classifies points which are lower than neighbouring points in the source class. This routine is run after ground classification, to locate points which are a bit below the true ground surface. For each point, the routine finds up to 25 closest neighbour points in the source class. It fits a planar or curved plane to the neighbouring points and computes the average magnitude of the elevation differences between the points and the plane. This routine was used to classify low points.

By height from ground routine classifies points which are within a given height range compared to a reference surface. This routine is also used for classifying points into different vegetation classes for preparing, for example, building roofs, or tree detection. Buildings routine classifies points on building roofs which form a planar surface. The routine requires that ground points have been before. It is also advisable to classify points above the ground into a separate class, so that this class contains points in an elevation range above ground where building roofs are included. The routine starts from empty areas in the ground class and tries to find points on planar surfaces above these areas. Those two routines were used for automatic classification of Vegetation and Roofs of buildings. All other points were left in Default class.

For the comment “Many algorithms have been used in this Paper without clarifying about its reference and efficiency (e.g.: 270: digital terrain model derived from the cloud points; 271: filtering only the significant points of the ground” , answer is from line 318 until 328:

¹ <http://www.terrasolid.com/products/terrascanpage.php>

Airborne Lidar sensors provide dense height information of large areas in an efficient manner and it can be used for the development of the 3D city model [56]. In our case study it has been operated with the parameters of 40 points per square meter, and aerial images with 5 cm per pixel.

For the production of digital elevation model (DEM) from the point cloud, the filtering of the point cloud should be carried out in order to remove points representing surface of non-ground objects. For the ortho-rectification a digital terrain model (20cm/px) derived from the cloud points was used, filtering only the significant points of the ground. For this purpose, we have also used TerraScan, which use a filtering algorithm based on Triangular Irregular Networks, which adapts to the data points from below and handles surfaces with discontinuities. [57, 58]. Due to its efficiency, in the last years we have seen a lot of examples of using Lidar data for process of ortho-rectification, not only for airborne images, but also for satellite images [59, 60].

For the comment “Many algorithms have been used in this Paper without clarifying about its reference and efficiency (e.g.: 277 : basic classification; 278 : creation of digital elevation model”, answer is from line 333 until 340:

In this case study, after LiDAR scanning, as a first step orthophoto images and a georeferenced point cloud have been created. Before any further steps, this data had to be processed and prepared. Data preparation included the following steps: **initial** classification, detail classification, 2D or 3D vectorization and creation of digital elevation model. **Initial** classification usually depends on the software solution, but all of them classified LiDAR data in several similar classes (Default, Ground, Low vegetation, Medium vegetation, High vegetation, Building, Low points and Model key points). **As was mentioned before, all these steps were done in TerraScan, using already prepared batch macros for initial classification and for creation of digital elevation model.**

For the comment about “first ETL transformation scheme for vectorized CAD models into GML model without textures (this process is Unclear and understandable)”, we have improved explanation of this Figure 9, from line 420 to 457:

First ETL transformation (Figure 9), at the beginning contains 4 input layers of CAD format: houses, buildings, auxiliary objects and business objects. As in the previous ETL transformation, in order for the data to be transformed into GML format, it is necessary to transform them from polygon geometries into surface geometries, and orient them all in the same direction, more precisely by the left-hand rule. **The orientation step performed at the start through the orienter transformer, is important in order to have uniformly oriented surfaces for the future work of adding textures to specific side of a surface (roof or wall surfaces) as well as for distinguishing the exterior and the interior side of a surface when building solid geometries.** As for the CityGML feature type WallSurface and RoofSurface elements, they carry geometry information for each object that must be preserved in ETL transformation, so that they can be used in future topological analyses in the Smart 3D city model. The CityGML feature type Building is a purely attribute element, and in this case, it does not carry information about geometries, but it stores all the necessary information that is relevant for each of the buildings in the 3D model of the city.. **As an example of adding attributes to Building type, real heights for each object are extracted from the coordinates of each object individually and added to the appropriate CityGML attribute defined by the standard itself (citygml_measured_height). Another attribute related to object heights are the units of measure in which the heights are expressed (citygml_measured_height_units).** In addition to these attributes, the Buildings type defines the level of detail of each building, feature roles, LoD name. Also, if there is information for

each object such as the street where it is located, building number, number of floors, floor area and others they are added as part of this feature type. All three feature types (Building, RoofSurface and WallSurface) have common gml_id over which they are uniquely defined. Also, CityGML Geometry Setter transformers were assigned to all three feature types and coordinate systems were defined (EPSG: 32634). As a result of the ETL transformation from Figure 9, the CityGML model of 1204 of the city buildings (located in the city centre, campus, and Petrovaradin Fortress) of LoD2 was developed. **This model is set in the appropriate coordinate system and it does not contain photorealistic elements.**

The problems that were noticed during the creation of the described ETL transformation, i.e. the 3D model of the city, are that some objects lost some of their geometric characteristics after the transformation. For example, some buildings did not get their roof geometries, or they got them in part, as well as the problem that the footprints of some of the buildings are not where they should be, but are about 80 meters lower (at 0 meters above sea level). **Subsequent analysis of the aforementioned models of objects showed that the problem did not originate from the ETL transformation. The root of the problem was in the models created in CAD structure through automatic 3D building vectorization. Some of the buildings were not properly vectorized and it resulted in faulty geometries according to the CityGML standard. This problem was solved by a repeated 3D vectorization of selected objects.**

For the comment “Figure 3 should be approved”, we have improved Figure 3 and explanation about this image Figure 3, from line 351 to 372:

After classification and 2D and 3D vectorization of point cloud, 3D building models in the resulting CAD structure can be used in multiple ways. The basic usage of this CAD structure is only for pure visualisation, but for advanced and further usage in various applications it must be transformed into GIS or BIM formats. There are several steps that need to be taken in order to fully transform CAD format such as DGN or DXF to IFC or GML format. Figure 3 shows the **proposed** workflow of the data transformation process.

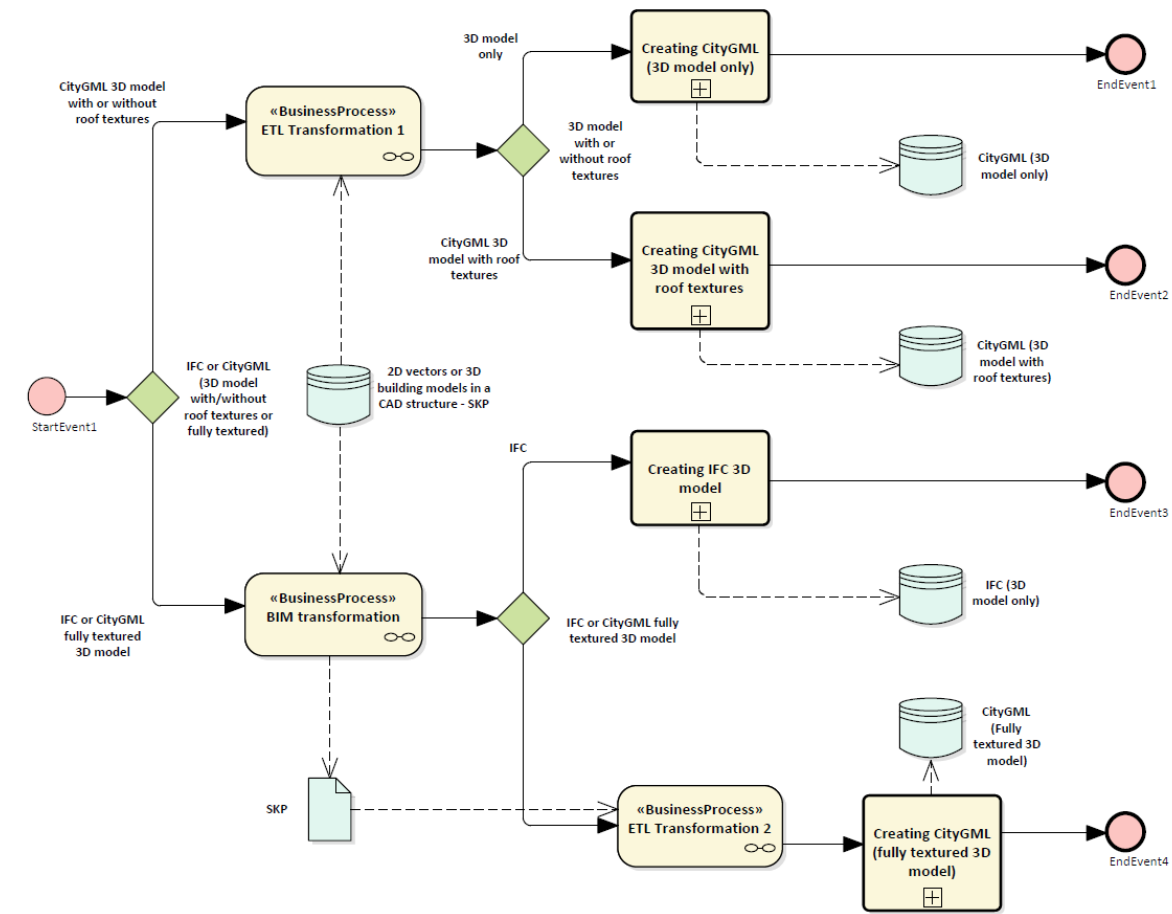


Figure 3. The defined workflow of the data transformation process.

First solution is to create a CityGML model directly from a CAD structure, or a CAD file organized in layers (Figure 3 – ETL Transformation 1). Using proposed ETL Transformation 1 it is possible to create model according to CityGML standard almost automated, because it skips the step of manually processing in a BIM tool. Workflow for transformation from 3D CAD structure into GML model with or without roof texture is shown in Figure 8.

On the other hand, as it can be seen on the Figure 3, for fully textured 3D model according to the CityGML standard, previously generated DGN file was imported to BIM tool software in order to add photorealistic textures to the 3D model. After that, as a one solution IFC format is available directly, or using ETL transformation 2 and provided algorithms and functions for manipulation of the 3D objects, we have appropriate textured 3D model according to the CityGML standard (Figure 4). For the model to comply with GIS standards, it is important that it stores geometric as well as topologic and semantic characteristics of the object. We use ETL transformation 2 which from the CAD data, create fully designed models in one of the two most commonly used standards for storage, visualisation and, most importantly, data analysis – CityGML and BIM/IFC (Figure 3).

For the comment “there is no spatial resolution analysis in this article”, we can say that spatial resolution is appropriate for this kind of research (example: Reference 56), also additional information about spatial resolution can be found in lines

- 319-320: In our case study it has been operated with the parameters of 40 points per square meter, and aerial images with 5 cm per pixel.
- 323-324: For the ortho-rectification a digital terrain model (20cm/px) derived from the cloud points was used, filtering only the significant points of the ground
- 331-332: The resolution of the ortho-images was 10 cm.

Also, we have added text about accuracy assessment, where can be find additional information about accuracy in line 548-550:

Vertical accuracy control was performed on 9 grids of control points and the determined RMSE was below 3 cm. Horizontal accuracy control show similar results, with determined RMSE below 10 cm (Figure 14).

We hope that we have addressed all your comments and suggestions, however, due to the length of the article, future work will include more thorough analysis of the algorithms that we have used.