Hierarchical Quarters Model Approach toward 3D Raster Based Generalization of Urban Environments

Alexey Noskov
Mapping and Geo-Information Engineering
Technion – Israel Institute of Technology
Haifa, Israel
e-mail: noskov@technion.ac.il

Yerach Doytsher
Mapping and Geo-Information Engineering
Technion – Israel Institute of Technology
Haifa, Israel
e-mail: doytsher@technion.ac.il

***Abstract*—The suggested method for 3D generalization of groups of buildings in urban environments is based on the rasterization of the 2D footprints of 3D buildings. The rasterization is processed within quarters, which are automatically defined by using Digital Elevation Model (DEM), water objects and roads. Quarters were organized into a hierarchical model according to the gaps between the quarters and the stages of the clustering process. Each degree of generalization corresponds to some level of hierarchy. The 3D urban perspective is computed based on separate levels of generalization of each quarter as a function of its distance from a pre-defined view point. The developed approach enables to compile a 3D scene of urban environment based on the generalized buildings' layer. The buildings' layer consists of objects with different degree of generalization level which is growing gradually from the view point. The two main distinctions of the approach from others are: (1) the generalization is implemented with respect to the geospatial properties of urban environments and the relations between the objects; (2) the approach is simple and universal which enables to simplify the whole area of a city and can be applied to different types of cities.**

Keywords-Generalization; 3D urban model; groups of buildings; hierarchy of quarters

I. INTRODUCTION

3D generalization of the urban model is a fast-growing topic. The main types of objects in the 3D city model are buildings. Nowadays, 3D models are used in many disciplines [33]: GPS navigation, desktop and mobile city viewers, geo-simulation, architecture, and many others. The two common problems which usually arise in any discipline are: (1) huge computer resources are required for drawing 3D models based on the original, non-simplified buildings, and (2) 3D models based on the original non-simplified buildings are very detailed and often appear unreadable and overly complex. To resolve both problems we have to generalize the buildings. There are two different tasks in the building generalization process: (1) simplification of a single building, and (2) generalization of groups of buildings. The topic "simplification of a single building" is a widely researched topic; we can describe several different approaches of generalization, all of them valid. In contrast, "generalization of a group of buildings" has only been treated, so far, on a very limited level. There are several very similar approaches, largely based on the Delaunay Triangulation (DT) (e.g., [40] and [25]). We propose, in concept, another approach for the generalization of groups of buildings, based on rasterization and vectorization operations, which are carried out by sub-dividing the urban neighborhood into quarters. This paper is structured as follows: the related work is considered in section two, the source data are described in section three, the algorithms of raw quarter calculations and building quarters' hierarchy are presented in sections four and five, the raster based algorithms of generalizing group of buildings is considered in section six, the results are evaluated in section seven and, finally, in the last section the conclusions are detailed.

II. RELATED WORK

One of the most holistic approaches to the 3D generalization of buildings was described in [40]. The main idea supposes that, within a threshold (distance from a view point), we will generate objects which contain the results of simplification of single buildings, whereas outside of the threshold we will generate objects containing the results of groupings of buildings and their simplification as a single building. An approach of "converting 3D generalization tasks into 2D issues via buildings footprints" was described by He et al. in [11].

The generalization of 3D building data approach [7], based on scale-space theory from image analysis, allows the simplification of all orthogonal building structures in one single process. Another approach [36] considers buildings in terms of Constructive Solid Geometry (CSG). In [40] an approach was proposed which realized 3D single building simplification in 5 consecutive steps: building footprint correction, special structure removal, roof simplification, oblique facade rectification and facade shifting. A very interesting approach was proposed by Kada in [16] and [17]. In this approach, geometric simplification was realized by remodeling the object by means of a process similar to half-space modeling. Approximating planes are determined from the polygonal faces of the original model, which are then used as space dividing primitives to create facade and roof structures of simpler shapes.

The second aspect of 3D generalization of an urban environment is the generalization of groups of buildings. 3D generalization of groups of buildings is mentioned in several publications (e.g. [8], [10], [11], [37]). These papers describe different approaches to 3D grouping and group generalization: grouping of building models (using the infrastructure network) and replacing them with cell blocks, while preserving local landmarks [8]; "express different aspects of the aggregation of building models in the form of Mixed Integer Programming problems" [10]; and, grouping of building models "with a minor height difference and the other with a major height difference" [11].

2D building generalization algorithms should also be considered for use by researchers for a 3D building group generalization. A holistic and automated generalization method based on a pseudo-physical model was considered in [15]. An approach based on Delaunay triangulation, Graph and Gestalt theory was described by Li et al. in [25].

In the above-mentioned publications, different approaches were considered, but we can identify some common ideas which are important for most research in this area.

In most cases it is very useful to generate levels-of-detail (LOD); normally, researchers use 3 or 4 LODs ([4], [27] and [36]). LODs are widely used in 3D video games, usually for detailed objects; more simplified objects are created for saving processor load and virtual memory [27]. Usually a detailed object has references to several simplified versions (at different levels of simplification), so that if the object stays near the view point, the most detailed version of the object is used, and as the object is located further from the view point the more simplified object is used.

It is very popular to use CityGML standard for 3D urban models ([9], [13], [19], [20], [21], [22], [23] and [35]). This format supports many useful possibilities, which are very important for working with 3D urban models (e.g., LODs, topology, semantics etc.).

Today, 3D city visualization is an extremely fast-developing topic [3]. There are many benefits to using 3D city models, for instance, the significant advantages of using 3D maps in cadastral systems and public participation in urban planning processes have been described by Shojaei et al. in [34] and by Wu et al. in [39]. There are many approaches and technical solutions for storing and visualizing 3D city models. In [31] a detailed analysis of existing approaches to 3D city visualization was published. According to this publication, there are several principal formats and standards which are normally used in reviewed projects - namely - CityGML, KML/CALLADA, X3D, X3DOM, HTML5/WebGL, OpenStreet map data format. Several data sets store huge amounts of 3D buildings' models - Paris, Berlin, Mainz, Blacksburg geodatabases and OpenStreetMap data were described. In addition, important applications for working with 3D virtual city data - CityServer3D, 3DCityDB, IGG Web 3D Service, OSM-3D Web 3D Service, HPI 3D Server and Web View Service, Xnavigator, InstantReality Player, BSContact Geo,HPI 3D WVS Clients, Google Earth - were reviewed.

The OpenStreetMap crowdsourcing project has made it possible to depict 3D maps of numerous cities world-wide. For a huge number of building models, users are defining tags (attributes) such as building heights, number of floors, type of roofs etc. The OpenBuildingModels platform [38] enables us to prepare and add to the OpenStreetMap realistic 3D complex buildings models. This, and the fact that the OpenStreetMap is a free open source project and allows access to the data under a copy-left license, enables the development of applications for creating 3D interactive real maps. Impressive results were achieved in the OSM-3D project (see [29]). This is an experimental but actual project, working in 3D globe application, and has been released as Java Applet.

As mentioned by Hildebrandt and Döllner in [12], due to the advances in computer graphics and improved network speed it is now possible to navigate in a 3D virtual world in real time. Until recently, the technologies employed required installing standalone applications or plugins on navigators. The relatively new HTML 5 format brings new solutions for visualizing 3D data in a web browser by using WebGL. Several globe projects have proven that such technologies are feasible and can be employed. One of these projects was described by Mao and Ban in [28], where CityGML data are interactively converted to X3D format according to the user request on the server, and the X3D data are visualized on the user's web browser. This method can work on any modern web browser with WebGL (e.g., Mozilla Firefox, Google Chrome, and Safari). In [30] it has been proven that the same approach can effectively work on mobile devices (smart-phones and tablets).

In spite of the large number of publications and developing projects, we can identify a very important shortcoming on almost all 3D city maps (or screenshots). There are usually only two ways of displaying large cities: depicting only the buildings nearest to the view point (whereas all the other buildings are not displayed and the area is depicted as a plain map), or displaying all the 3D building models (which usually causes a long processing time and heavy computer and internet traffic resources, and furthermore, causes distant parts of the city to be presented as a very dense and unreadable 3D view). As mentioned above, we are seeing a large number of publications on generating and using LODs of single buildings, while there are only a very limited number of approaches to creating LODs of group of buildings in order to solve the problem described above.

Additionally, it must be mentioned that in the approach we used Kohonen's self-organizing maps [18] (one of the artificial neural network algorithms) to classify quarters according to a set of different attributes.

There are several possible approaches to multidimensional classification. In our case, using one of the clustering algorithms seems very promising. There are several common groups of clustering algorithms: hierarchical clustering, centroid-based clustering, distribution-based clustering, and density-based clustering. According to [14] "hierarchical clustering is based on the core idea of objects being more related to nearby objects than to objects farther away". "As such, these algorithms connect 'objects' to form 'clusters' based on their distance". This method is often considered obsolete. Centroid-based clustering is based on defining the optimal central vector of clusters. K-means or Lloyd's algorithm [26] is a well-known centroid-based approach. It is an unsupervised algorithm which requires setting the number of classes. It has been mentioned that a k-means approach cannot find non-convex clusters [6]. Some k-means algorithms to classify the raw quarters of Trento have been tested, and the results look interesting and useful for our aims. According to [32] distribution-based clustering methods "suffer from one key problem known as over-fitting, unless constraints are put on the model complexity". In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set [24]. The DBSCAN [5] density-based popular clustering method has serious disadvantages for our case, as it is not easy to define initial parameters (certain distance thresholds and minimum number of points required to form a cluster) for this method. This disadvantage was partially eliminated in the OPTICS [2] method; in this method only a minimum number of points are required. But it is still a problem, because in our case we have a very specific small group of quarters. On the other hand, if we set a small minimum number of points we will get too many clusters. From all the approaches described, the k-means algorithm seems very suitable for our investigation. The Kohonen's self-organizing map (SOM) [18] is one of the ANN methods; this method's implementation is very close to k-means. SOM is not only a clustering method; it is also a very useful tool for visualization and evaluation of the results of clustering. At this time we are focusing our attention on the SOM approach of clustering.

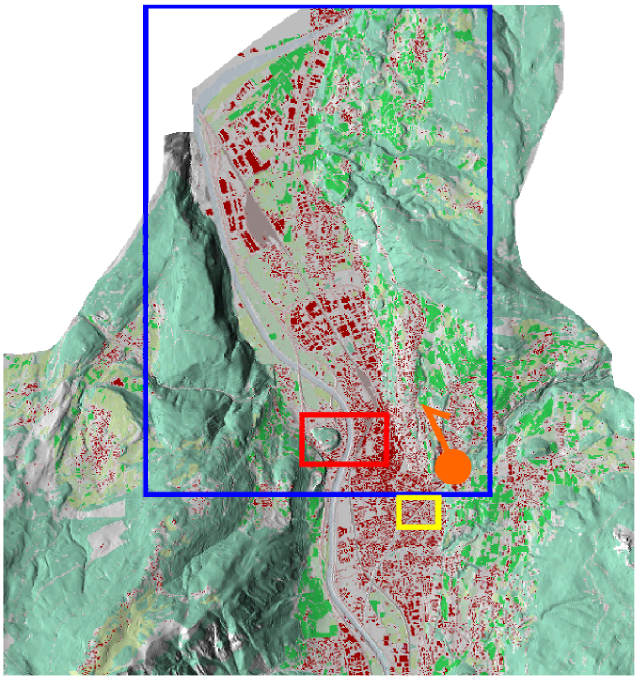

Figure 1. Map of Trento.

III. SOURCE DATA

For implementing and testing our approach, the free geodata of the city of Trento, Italy was used. The buildings (with individual heights), water objects and roads were extracted from the landuse map of Trento (see Fig. 2). The landuse map and land relief (DEM) were downloaded from the website of Trento Municipality [41]. On the map of Trento (see Fig. 1) the buildings are depicted as brown areas; the extent of the maps in Fig. 2, Fig. 3 and Fig. 4 are marked with a red square; the extent of the map on Fig. 13 with a blue square, and the view point and view direction of Fig. 14 with an orange circle and arrow.

IV. CALCULATION OF RAW QUARTERS

Finding a realistic method of simplification is a very important issue in generalization. One of the more common problems is when buildings are joined through obstacles such as wide roads or rivers. In this case, buildings must not be joined to each other, and these buildings from the two sides of the obstacle should be merged with other, more distant objects, which are, however, located on the correct side of the obstacle. To resolve this problem, we decided to split the urban space into quarters which are divided by the main, significant objects. These objects cannot be involved in the generalization itself.

To calculate the quarters, we decided to use the slope of the terrain, water objects and roads. In Trento, it was found that buildings are positioned only on areas with a slope of less than 30 degrees. Accordingly, areas with slopes greater than 30 degrees of the terrain were excluded. We also used roads and water objects for defining the quarters. These objects are polygons extracted from the landuse map. In an initial work [1] we used attributed data of roads, based on the fact that we worked with linear objects downloaded from the OpenStreetMap website. In the next stage, presented in this paper, we use polygonal features, which, due to their geometric characteristics (their size) as the main parameter, avoid the need to use attributed data (like type of object) of roads and the water objects. All these three classes – slopes, roads and water objects - were merged into one raster map with 1 meter resolution (which has been found to be adequate for small-scale urban generalization).

The raster map of the merged objects is the base for quarter calculating; further processing can be divided into several consequent steps.

The raster transformations for splitting the city into quarters have been selected because the standard vector approaches (e.g. polygons based on vector roads) have several limitations. The source vector road data may contain features such as unfinished roads, dead end roads, etc., features affecting its topological correctness. Splitting the area into quarters based on these data might result in a very complex polygonal map containing artifacts. In contrast, the raster transformations approach enables to exclude most of the artifacts and the unnecessary boundaries and vertices. As the width of the narrowest roads is about 2-3 meters, the resolution of the raster maps has been defined as 1 meter. Accordingly, the quarter map is composed of polygon bounds which coincide approximately with the road centerlines (±1 meter), as well as not intersecting the buildings.

A. *Region growing of base features*

All pixels of the merged objects got the value "1"; empty space on the raster map got the "Nil" value. Each group of pixels with value "1" has been expanded by adding one pixel (1 meter) and the results are depicted in Fig. 3.

B. Inverting of pixel map

At this step, the values of pixels were inverted ("1" to "Nil" and vice versa), resulting in many pixel areas with the value "1" which are split by "Nil" pixels.

C. Defining quarter areas having unique values

To set a unique pixel value to each quarter area, we vectorized the raster map. Each vector that defines a polygonal object got a unique integer identifier. Polygons not containing buildings were removed. Then the vector map was rasterized. For raster values, polygon identifiers were used. As a result (see Fig. 4), we got a raster map with groups of pixels (a "quarter") and each group (which is separated by "Nil" pixels from the adjacent group) got a unique integer identifier.

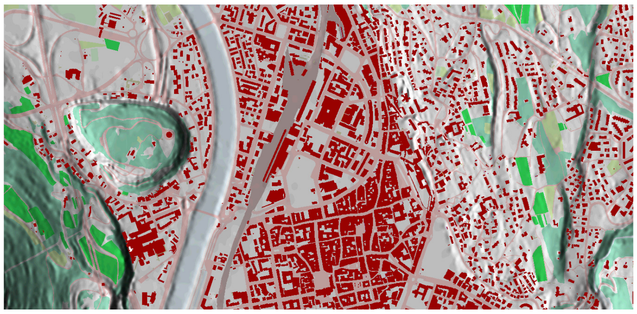

Figure 2. Shaded Landuse Map of Trento and Buildings (brown polygons).

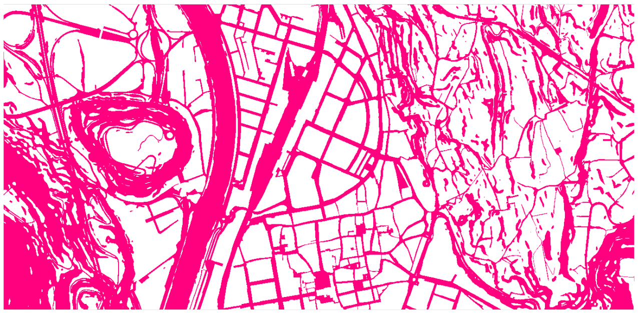

Figure 3. Non-Nil Pixel Groups which Split the City Space into Quarters.

Figure 4. Inverted Raster Map with Unique Pixel Values.

At this stage, a set of raw quarters was prepared. The next data processing was based on this. We can see on the map that quarters contain holes, dead-ends and unfinished roads. These elements naturally disappear during the generalization of the quarters.

V. CALCULATION OF QUARTERS' HIERARCHY

In the previous step we prepared raw quarters. We cannot use them for a high level of generalization, as raw quarters would be too small for large generalized buildings. To overcome these limitations, a new flexible hierarchical approach of subdividing the urban area into variable quarters was developed. The raw quarters are placed on the lowest level of the hierarchy; on the highest level, the whole area of the city is defined as one large quarter. A special approach to developing the quarter generalization (or quarters merging) will enable this hierarchy, where the size and content of the quarters will be correlated with the level of the 3D generalization, and this, in turn, will be related to the distances from the view point. Each level of the hierarchical tree of quarters has some level of quarters' generalization. The hierarchy is based on buffer operations. We will widen the quarters by a buffer; thus, adjacent quarters will be merged into one object, while other objects will only change their geometry (outer boundary of quarters will be simplified, some inner small elements like holes and dead-end roads will be filled). Then we will decrease the quarter by a buffer with the same size. If the width of the gap of merged quarters is smaller than the buffer width, the objects remain as a merged polygon, otherwise the polygon will be split back into separate objects differing from the original objects due to their simplified geometry.

The raw quarters will be generalized and organized in a hierarchical tree. Each level of tree is built on the previous level, and is calculated as follows: first, the attributes of each quarter are calculated, depending on their classification. Each level is built according to some base buffer width. All quarters are separated into temporary layers according to their classes. Then, we apply buffer operations (with a width which is equal to two times the base buffer width) separately to each layer, and overlay all the layers into one map. On this map, we apply buffer operations with the base buffer size to all quarters without taking their classes into consideration. This suggested approach helps to merge quarters of the same class faster than others (i.e., quarters of different classes should be at least twice as close as quarters of the same class to be merged). In Listing 1 the algorithm of this process is presented. In Fig. 5 you can see its results (A: Background Color by Original Classified Quarters, Black Polylines are Outlines of Buffers by Classes of Quarters, Width is Equal to Twice the Base Width; B: Withdrawal of Buffers; C: Adding Base Buffer Width to the Polygons; D: Withdrawal of Buffers –Resulting as the Final Buffering).

The first step of quarter generalization is calculating the attributes for each quarter. These calculated attributes (a list of the attributes is depicted in Fig. 6) will be used for the classification of the quarters.

We decided to use the Kohonen's Self-Organizing Map approach to classify quarters. The number of clusters was defined for all levels in the hierarchy of the quarters to perform classification. There are several techniques to

Listing 1. Algorithm of Building the Hierarchical Tree of Quarters.

```
quarters=original_quarters
for ( buffer=1 , buffer+=0.2 , buffer < 2):
    Calculate attributes of quarters
    Classify quarters according attributes
    for ( i=0 , i++, i < number of classes):
        Extracting class i into separated map
        foreach current_quarter in quarters:
            Adding buffer*2 to current_quarter
            Withdrawing of buffer*2 from current_quarter
        Merging quarters from separated maps into
        buffered_quarters
        foreach current_quarter in buffered_quarters:
            Adding buffer to current_quarter
            Withdrawing of buffer from current_quarter
    Appending of buffered_quarters into result Hierarchical Tree
    quarters=buffered_quarters
```

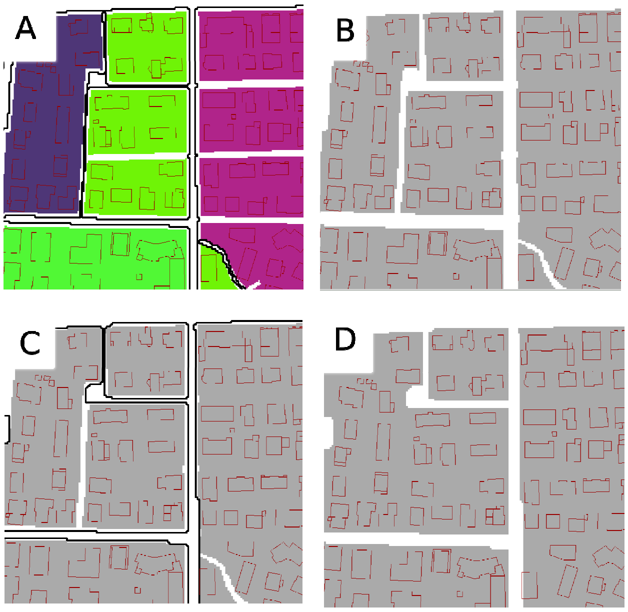

Figure 5. Buffering Process.

automatically define the numbers of clusters (e.g., a gap statistic approach, an information theoretic approach, etc.). At this time, we have focused on a manual definition of the number of classes. An initial manual analysis of a series of maps based on quarter attributes visualization was carried out. On Fig. 6 you can see example of quarters' classification, meaning of parameters are presented below:

- A - Typical Azimuth of Buildings' Sides;
- S_q – Aarea of Quarter;
- S_b – Area of Buildings in a Quarter;
- D – Density of Buildings, S_b/S_q;
- F_q - Fractal Dimension of the Quarter's Boundary, 2·log(perimeter) / log(area);
- F_b – Mean Fractal Dimension of Buildings' Boundaries in a Quarter;
- C_q – Compactness of a Quarter, perimeter/(2· √ (π·area));
- C_b – Mean Compactness of Buildings;
- P_q – Perimeter of a Quarter;
- P_b – Mean Perimeter of Buildings;
- PP – P_b/P_q;
- H - Arithmetic Weighted Mean by Areas of the Heights of Buildings.

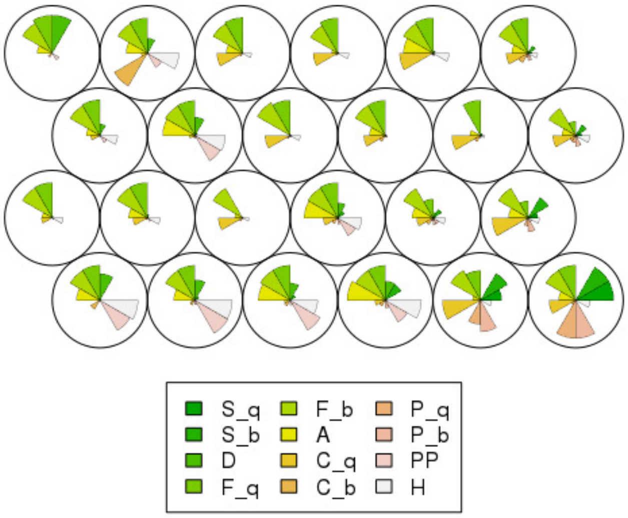

Figure 6. Diagram of Codebook Vectors (Centers of Clusters). 24 Classes of Calculated Quarters with Buffer Width of 1.6 meter..

As aforementioned, in order to take into account the quarters' classes, adding a weighted buffer to the quarters has been suggested (thus differentiating between quarters of the same class and quarters of different classes), aiming to merge neighboring quarters. To avoid vector artifact and topology problems, data was converted to raster, and buffering operations were executed in a raster environment. We used the raster resolution as the base width of the buffer. Not only does the buffering phase provide the possibility of merging quarters, but this operation also helps to fill holes and dead-end roads in polygons, and to eliminate small elements of quarters' boundaries. It should be mentioned that converting vector to raster can also work like a generalization operation. Generally speaking, this phase in the research, which is based on vector to raster and raster to vector operations, as well as region growing and buffer implementations on the one hand, and on the quarters' attributes on the other hand, enables us to generalize the quarters and lets us move up or down in the hierarchical level of the quarters' subdivision.

As mentioned above, quarters of the same class were merged faster than quarters of different classes. This was achieved by putting quarters of the same class into isolated

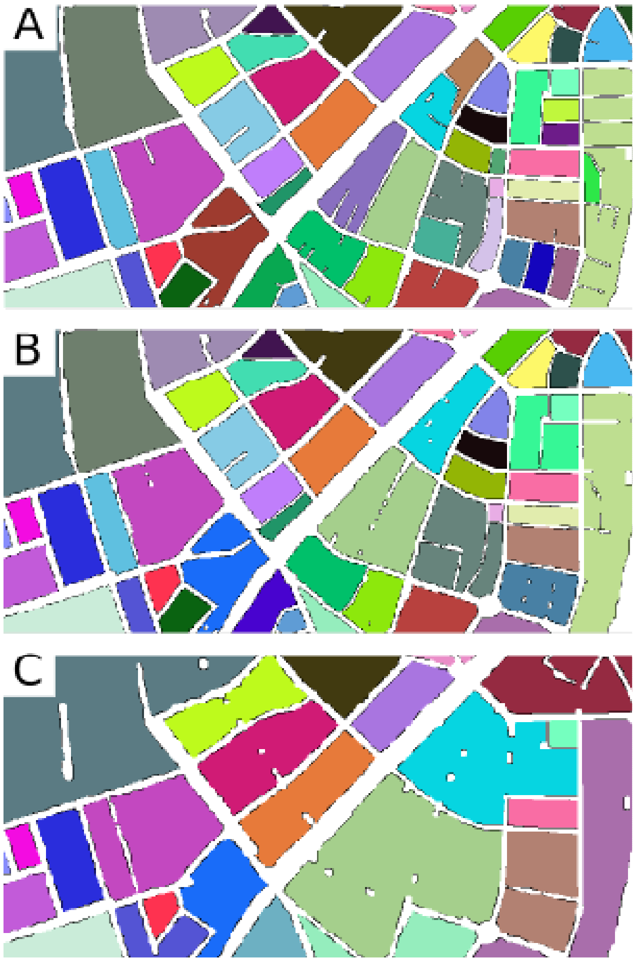

Figure 7. Quarter Buffering Generalization (Buffer Width, in meters): A: Original Raw Quarters; B - 1.4, and C - 1.8.

sub-environments (temporal layers) and using different widths of buffers (see Fig. 6 and Fig. 7).

This suggested approach allows the performing of quarter generalization based on buffering operations while taking into account quarters' classes. Using this method builds a hierarchical tree of quarters (in the current sample of Trento, from the raw source of 2679 small quarters up to a single huge quarter). We performed the generalization of quarters starting from a buffer width of 1 meter and increasing it by increments of 0.2 meter, until the buffer width reaches 2.6 meter. It was decided to start quarter generalization from 1 meter because this is the resolution which is used to generate the raw quarters; and because the upper limit of 2.6 meter as a higher buffer width generates oversized quarters.

As in the previous research [1], we decided to use 8 degrees of generalization of buildings based on rasterization processes with resolutions 10, 15, 20, 25, 30, 40, 50 and 60 meters (resolutions which correspond to degree of generalization). A graph of the varying number of quarters and the size of maximal quarter (see Fig. 8) was used to define which levels of hierarchy can be used for further processing. In addition, the original vector map of buildings was converted to raster maps with different resolutions (10, 15, 20, 25, 30, 40, 50 and 60 meters). These raster maps, overlaid with the generalized quarter maps, were used to estimate which resolution of buildings generalization should be used with each generalized quarter map. Overlaying of the generalized quarter map with different resolution raster maps of buildings is illustrated in Fig. 9. We can then see very clearly on the right side of the figure that the sizes of the generalized buildings are too large to use this quarter map for generalization at this resolution.

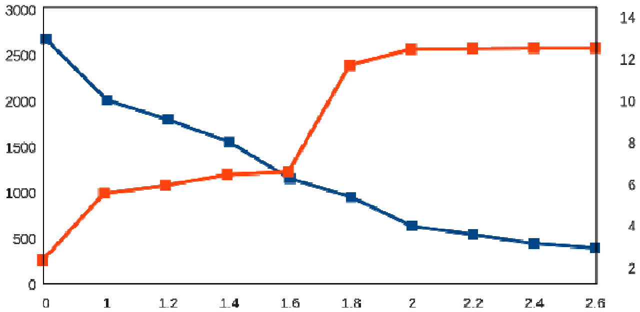

Figure 8. Graph of Number of Quarters (blue Line, left Y-axis); and the Size of Maximal Quarter (red line, right Y-axis in million sq. meters); X-axis – Base Buffer Width (in meters).

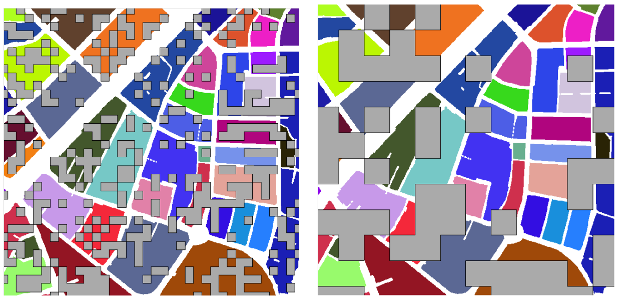

Figure 9. Quarters and Sizes of Buildings Generalization: Appropriate (left) and Too Large (right).

By using this method we estimated which levels of quarter hierarchy can be used and which resolution of buildings generalization should be processed with these levels (see Table 1).

VI. GENERALIZATION OF BUILDINGS

The fact that in urban areas, most (if not all) of the buildings have orthogonal sides, is the background for our raster-based generalization approach. Usually, in adjacent areas (quarters in our case), buildings would be spatially oriented in the same direction. Therefore, the generalization process consists of defining the typical azimuth of buildings' sides for each quarter. Once a typical azimuth is known, by applying the rasterization process in this direction, the staircase-type appearance of lines, or legs of closed polygons, which is very common in the rasterization processes, can be eliminated. A non-rotated rasterization (parallel to the grid axes) while the buildings are positioned in another orientation will result in a staircase-type appearance of the bordering lines of the buildings and too many unnecessary vertices, which will prevent us from achieving a smooth geometry of the generalized objects.

A. *Defining the azimuth of buildings' sides*

As aforementioned, in urban areas, most of the buildings have orthogonal sides; thus, it is possible to define the average spatial orientation of the buildings. Within each quarter, the azimuths of all the buildings' sides were computed. For each building in the quarter, the longest side and its azimuth were identified. Then all the azimuths of the other sides were rotated by 90 degrees (clockwise) again and again; and the rotated azimuths (and their lengths) were put into one list. The list was sorted by lengths, and then lengths with the same azimuths (up to a predefined threshold) were averaged. A threshold of 1 degree when looking for close buildings' side azimuths has been found to give satisfactory results. A weighted average of the azimuths of the longest lengths of all the buildings within a quarter is used to define the general orientation of all the buildings of the quarter.

TABLE I. BASE BUFFER WIDTHS AND APPROPRIATE RESOLUTIONS OF BUILDINGS GENERALIZATION

Base buffer width (in meters) used to generate quarters' map (number of level in hierarchical tree)	**Resolutions of generalization, (in meters)**
original buildings (0)	original buildings
original buildings (0)	10
original buildings (0)	15
1.0 (1)	20
1.0 (1)	25
1.2 (2)	30
1.4 (3)	40
1.6 (4)	50
1.8 (5)	60

B. *Rotation and rasterization of the buildings in a quarter*

As mentioned above, and in order to significantly reduce the number of vertices of the generalized building and achieve a more realistic appearance of these simplified objects, rasterization should be carried out in the spatial orientation of the buildings. A rasterization which is spatially oriented parallel to the grid axes will define the buildings which are not oriented parallel to the grid axes in a staircase-type appearance of the buildings' sides. Accordingly, all the buildings within a quarter were rotated counter-clockwise at the angle of the general orientation of all the buildings of the quarter. Then the rotated buildings were rasterized using a certain pixel size resolution (as explained in the next section). Each pixel with more than half its area covered by the original buildings gets the value "1"; otherwise it gets the value "Nil". Fig. 10 shows the result of this stage.

The level of the generalization is a function of the pixel size rasterization process - the greater the pixel size, the greater the degree of generalization. Accordingly, each quarter has been generalized at several levels of rasterization, resulting in several layers of different levels (level-of-detail) of generalized buildings for each quarter. Based on the original data of Trento, and according to our analyses, we found that using pixel size resolutions of 10, 15, 20, 25, 30, 40, 50 and 60 meters produces satisfactory results of a continuous and consecutive appearance of the level-of-detail of the generalized buildings. Buildings were generalized independently for all quarters and at all resolutions according to Table I (each resolution corresponds to a definite level of quarter hierarchy). Generalized buildings are stored in separated layers; the identifiers of these layers contain resolution of the buildings generalization and the number of the level in the quarters' hierarchy (or actually, the buffer width).

Figure 10. The Generalization Process of Buildings in a Quarter: Original Buildings (left); Rotated Quarter and the Generalized 10 meter Rasterized Buildings in red (middle); Final Result (right).

Listing 2. Algorithm of Arranging the Quarters' List (Based on the Hierarchical Tree) to Compile a 3D Scene.

```
array=[  [ zone from view point , hierarchy level ],
         [       0-1000 m,              0        ],
                    …
         [       >8000 m,               5        ] ]
result_list=[]
foreach current_zone, current_level in array:
    Intersect map of current_zone with current_level of quarters'
    hierarchy, getting current_quarters list
    foreach quarter in current_quarters:
        if (child of quarter in result_list):
            Append others child quarters of quarter to result_list
        else:
            Append quarter to result_list
```

To draw a 3D perspective of the city with the generalized buildings, the position of a view point had to be defined. Then we built buffer zones around the view point. The buffer zones defined the distances (practically, range of distances) from the view point to each quarter. As mentioned above, generalized buildings are grouped separately and stored by quarters in layers with identifiers containing the resolution of the buildings generalization and the level in the quarters' hierarchy. To define what layers of generalized buildings will be used in the 3D scene, we intersect the first zone (0-1000 meter from view point) with the 0-level map from the quarters' hierarchy. Selected quarters are stored in an accumulated list containing IDs of quarters and the level of each quarter in the hierarchy. Then we take the zone next further from the view point and check on Table I what level of quarters should be used, and intersect this zone with the defined quarter layer. After that we check all the selected quarters. If a member-quarter of the selected quarter (i.e., it is part of the quarters in the lower hierarchy that compose the selected quarter in the current hierarchy) already exists in the accumulated list, we add to the list only the other member-quarters of the current selected quarter. Only quarters which do not contain member-elements in the accumulated list are added to the list. To compile a final 3D scene we just need to merge layers of the original and the generalized buildings, according to the accumulated list, into one layer. The process is repeated until the last zone is achieved. On Listing 2 a pseudocode of this described process is presented.

Fig. 11 depicts the degree of generalization for each quarter, where the colors indicate the degree of the generalization. The relationship between the distances from view point, pixel size generalization, and the colors, are described in Table II. Finally, we merged all the separate generalized layers of all the quarters into one map (see Fig. 13) for further 3D visualization. The division of distances from the view point into a scale of continuous intervals was based on several tests, which enabled us to draw a realistic and continual 3D model or perspectives. The results of a 3D visualization, and comparison of the 3D perspectives with the original buildings and with the generalized buildings, are presented in Fig. 14.

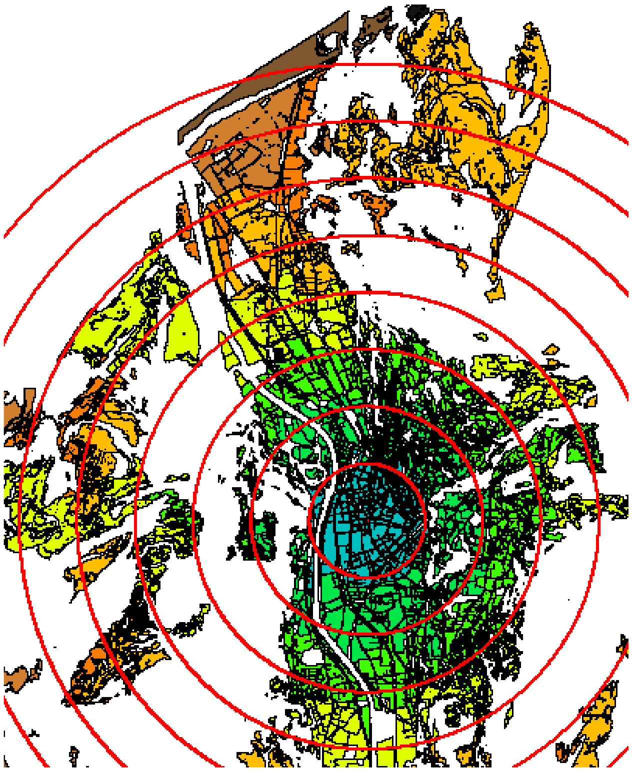

Figure 11. Defining the Degree of Generalization using Buffer Zones: Borders of Buffer Zones (red circles) and Quarter Borders (black).

TABLE II. DISTANCES FROM THE VIEW POINT, RESOLUTIONS OF GENERALIZATION, AND COLORS

Distances from view point, meters	Resolutions of generalization, meters	Background colors of the map in "Figure 12"
0 - 1000	original buildings	
1000 - 2000	10	
2000 - 3000	15	
3000 - 4000	20	
4000 - 5000	25	
5000 - 6000	30	
6000 - 7000	40	
7000 - 8000	50	
>8000	60	

VII. NUMERICAL EVALUATION

Table III presents the number of geometry primitives and the speed of the visualization process as a comparison between the original data and generalized data. As we can see, there is a significant reduction in visualization speed and in the number of polygons and nodes.

TABLE III. RESULTS OF THE GENERALIZATION

Parameter	Original building layer	Generalized building layer used for 3D visualization
Number of nodes	114,648	34,391
Number of polygons	46,339	14,956
Speed of 3D visualization, second	6.6	1.2

To evaluate the quality of generalization, the mean coefficient of building compactness was calculated for each resolution of generalization (see Fig. 12). The coefficient of compactness of a single building is equal to $\alpha=P^2/(4*\pi*A)$, where P – perimeter, A – area (α=1 for a circle, α=1.27 for a square).

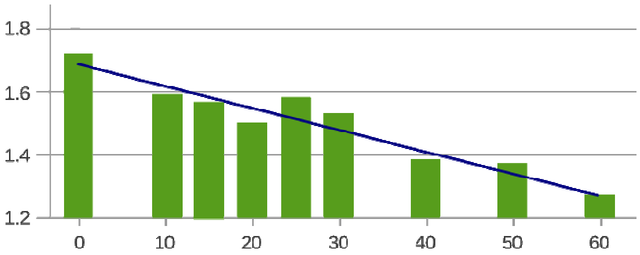

Figure 12. Coefficient of Building Compactness. X-axis – Resolution of the Generalization (0 - Original Buildings).

In Fig. 12 we can see that the coefficients of the buildings' compactness decreases significantly from 1.71 to 1.27, which demonstrates the efficiency of the approach.

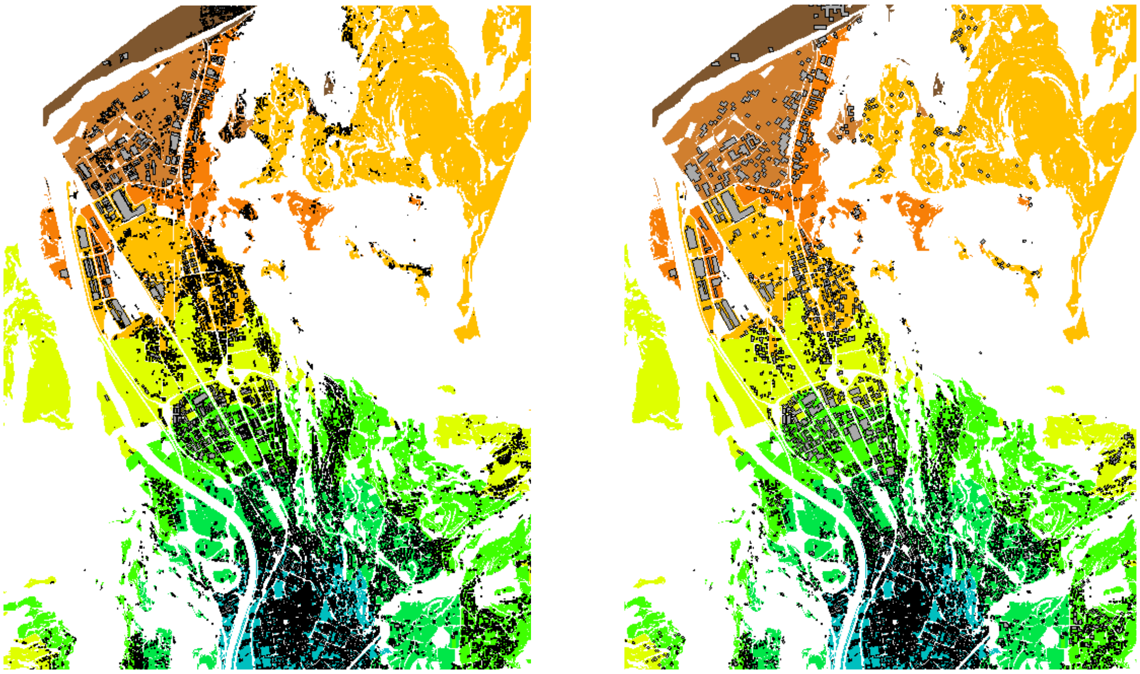

Figure 13. The Northern half of Trento with the Original Buildings (left) and with the Generalized Buildings (right): Different Levels of Generalization and Background Colors are according to Table I.

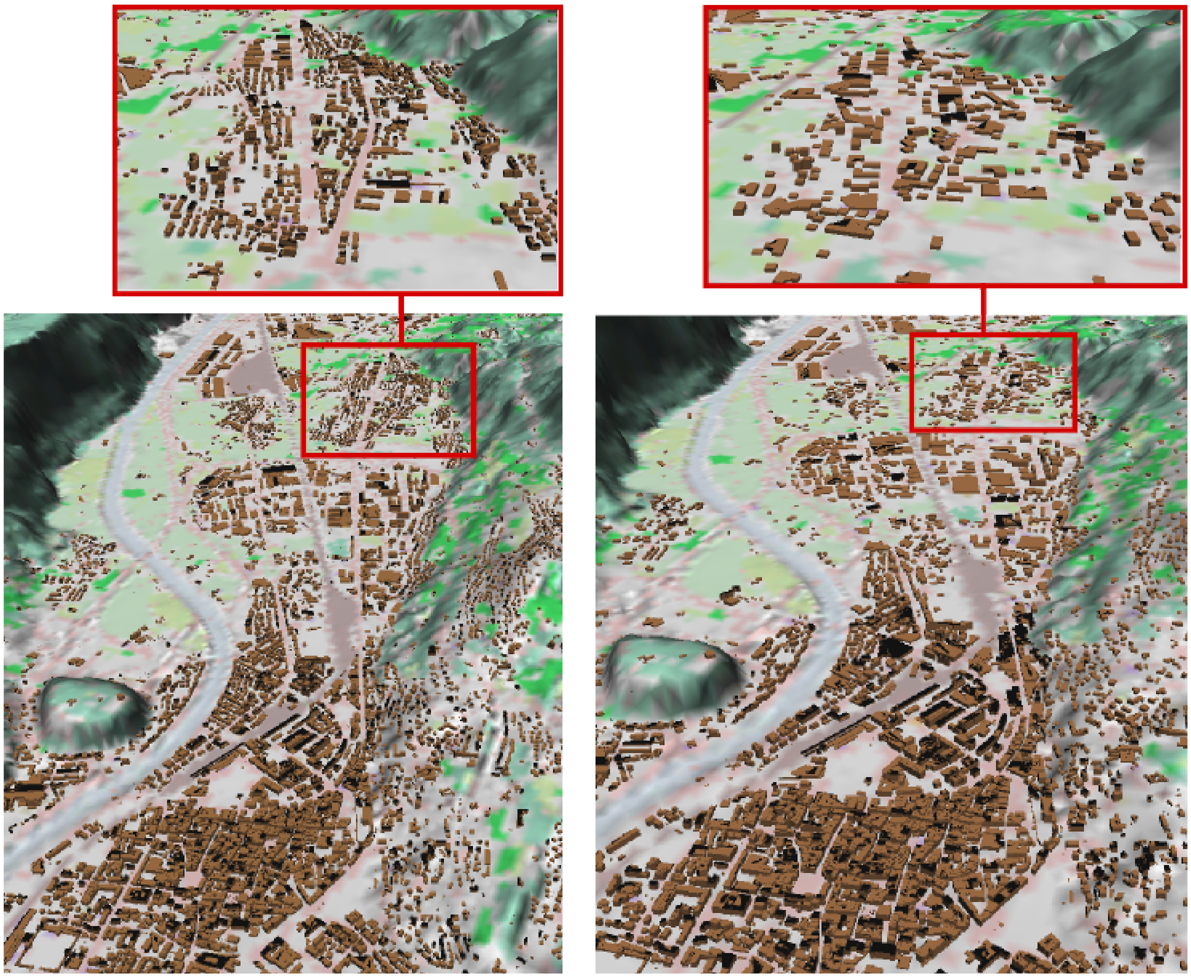

Figure 14. 3D Perspective with the Original Buildings (left) and with the Generalized Buildings (right). Zoomed Areas are Marked in Red.

The method and the process were developed by using a standard PC (DELL Vostro 3550), 4 processors: Intel® Core™ i3-2310M CPU @ 2.10GHz, with 1.8 GB Memory. In addition, Debian GNU/Linux 7 operating system, GRASS GIS, Bash and R programming languages were used.

VIII. CONCLUSION AND OUTLOOK

A new method for the 3D generalization of groups of buildings has been presented. To implement a multi-scale buildings generalization, a new hierarchical approach to the generalization of quarters and their contained buildings was developed. The approach is based on classification of quarters according to multiple attributes and on buffering operations. The raster-based approach of the method for buildings generalization is based on standard tools of rasterization, vectorization, region growing, and overlaying. The main advantage of the developed method is the ability to simplistically and efficiently generalize buildings at different levels, achieving variable, but continuous, level-of-detail of the buildings as a function of the depth of the plotted perspectives. The continuity of the generalized product is achieved by subdividing the area of the city into quarters, which take into account the significant objects affecting the process. As a result, the generalized 3D model does not contain unreadable and overly detailed separate buildings on the one hand, and is able to merge further groups of buildings on the other. At the same time, even though the buildings are simplified, the model maintains the geographical correctness and specifications of the urban area.

The developed method helps reduce the time, and the computer resources required, for drawing 3D models or perspectives of a city or urban areas.

REFERENCES

[1] Noskov A., Doytsher Y., “Urban Perspectives: A Raster-Based Approach to 3D Generalization of Groups of Buildings,” GEOProcessing 2013, pp. 67-72, France, 2013.

[2] Ankerst M., Breunig M., Kriegel H. P., Sander J, “OPTICS: ordering points to identify the clustering structure,” ACM SIGMOD Record, vol. 28, pp. 49-60, 1999.

[3] Breunig M., Zlatanova S., "3D geo-database research: Retrospective and future directions," Computers & Geosciences, vol. 37, 2011.

[4] Döllner J., Buchholz H., "Continuous Level-of-detail Modeling of Buildings in 3D City Models", in GIS'05 Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, pp. 173-181, ISBN:1-59593-146-5, 2005.

[5] Ester M., Kriegel H. P., Sander J., Xu X, "A density-based algorithm for discovering clusters in large spatial databases with noise," KDD, vol. 96, 1996.

[6] Estivill-Castro V., "Why so many clustering algorithms: a position paper," ACM SIGKDD Explorations Newsletter, vol. 4, pp. 65-75, 2002.

[7] Forberg A., "Generalization of 3D Building Data Based on a Scale-Space Approach", ISPRS Journal of Photogrammetry & Remote Sensing, vol. 62: pp. 104-111, 2007.

[8] Glander T., Döllner J. “Abstract representations for interactive visualization of virtual 3D city models”. Computers, Environment and Urban Systems, vol. 33, 2009.

[9] Gröger G., Kolbe T.H., Plümer L., "City Geographic Markup Language", Approved Discussion Paper of the Open Geospatial Consortium, 2006.

[10] Guercke R. , Götzelmann T., Brenner C., Sester M. “Aggregation of LoD 1 building models as an optimization problem”. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, 2011.

[11] He S., Moreau G., Martin J. “Footprint-Based 3D Generalization of Building Groups for Virtual City”. GEOProcessing 2012 : The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services.

[12] Hildebrandt D., Döllner J, “Service-oriented, standards-based 3D geovisualization: Potential and challenges,” Computers, Environment and Urban Systems, vol. 34, pp. 484-495, 2010.

[13] Isikdag U., Zlatanova S., "Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models", in 3D Geo-Information Sciences Lecture Notes in Geoinformation and Cartography, Part II, pp. 79-96, DOI: 10.1007/978-3-540-87395-2_6, 2009.

[14] Joshi M., "Classification, Clustering and Intrusion Detection System," International Journal of Engineering Research and Applications, vol. II, pp. 961-964, 2012.

[15] Joubran J., Doytsher Y., "An Automated Cartographic Generalization Process: A Pseudo-Physical Model", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVII, part B2, 2008.

[16] Kada M., "3D Building Generalization Based on Half-Space Modeling", Joint ISPRS Workshop on Multiple Representation, 2006.

[17] Kada, M., "Automatic Generalisation of 3D Building Models", in Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002.

[18] Kohonen T., “Self-organizing maps,” 3rd Edition, Springer, ISBN 3-540-67921-9, 2001.

[19] Kolbe T., "Representing and Exchanging 3D City Models with CityGML", in Proceedings of the 3rd International Workshop of 3D Geo-information, Seoul, Korea, 2009.

[20] Kolbe T.H., "CityGML – OGC Standard for Photogrammetry?", Photogrammetric Week, Stuttgart, Germany, 2009.

[21] Kolbe T.H., Gröger G., "Towards Unified 3D City Models", in Schiewe, J., Hahn, M, Madden, M, Sester, M. (Eds.): Challenges in Geospatial Analysis, Integration and Visualization II. Proceedings of Joint ISPRS Workshop, Stuttgart, Germany, 2003.

[22] Kolbe T.H., Gröger G., "Unified Representation of 3D City Models", Geoinformation Science Journal, vol. 4, 2004.

[23] Kolbe T.H., Gröger G., Plümer K., "CityGML – Interoperable Access to 3D City Models", in Proceedings of the First International Symposium on Fachbeiträge Geo-information for Disaster Management, Delft, The Netherlands, 2005.

[24] Kriegel, H., "Density-based clustering," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 1, pp. 231-240, 2011.

[25] Li Z., Yan H., Ai T. and Chen J. “Automated building generalization based on urban morphology and Gestalt theory”. Int. J. Geographical information science, vol. 18, 2004.

[26] Lloyd S.. "Least squares quantization in PCM," Information Theory, IEEE Transactions, vol. 28, pp. 129-137, 1982.

[27] Luebke D., Reddy M., Cohen J.D., Varshney A., Watson B., Huebner R., "Level of Detail for 3D Graphics (The Morgan Kaufmann Series in Computer Graphics)" ISBN: 978-1558608382, Edition 1, 2002.

[28] Mao B., Ban Y., “Online Visualization of 3D City Model Using CityGML and X3DOM,” Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 46, pp. 109-114, 2011.

[29] Over M., Schilling A., Neubauer S., Zipf A., “Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany,” Computers, Environment and Urban Systems, vol. 34, pp. 496-507, 2010.

[30] Prieto I., Izkara, J., “Visualization of 3D city models on mobile devices,” In Proceedings of the 17th International Conference on 3D Web Technology, pp. 101-104, 2012.

[31] Schilling A., Hagedorn B., Coors V., "OGC 3D Portrayal Interoperability Experiment Final Report", Open Geospatial Consortium. OGC 12-075 (opengis.net/doc/ie/3dpie), 2012.

[32] Sharma N., Jain R., Yadav M., "Efficient and fast clustering algorithm for real time data," International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, vol. 2, 2012.

[33] Shiode N., "3D Urban Models: Recent Developments in the Digital Modelling of Urban Environments in Three-Dimensions", GeoJournal vol. 52, pp. 263-269, 2000.

[34] Shojaei D., Kalantari M., Bishop I. D., Rajabifard A., Aien A.. "Visualization requirements for 3D cadastral systems", Computers, Environment and Urban Systems, vol. 41, pp. 39-54, 2013.

[35] Stadler A., Nagel C., König G., Kolbe T.H., "Making Interoperability Persistent: A 3D Geo Database Based on CityGML", 3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography, Part II, 2009.

[36] Thiemann F., "Generalization of 3D Building Data", in Proceedings of Symposium of Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002.

[37] Trapp M., Glander T., Buchholz H., "3D Generalization Lenses for Interactive Focus + Context Visualization of Virtual City Models", in Proceedings of the 12th International Conference Information Visualization, 2008.

[38] Uden M., Zipf, A., "Open Building Models: Towards a Platform for Crowdsourcing Virtual 3D Cities," In Progress and New Trends in 3D Geoinformation Sciences, pp. 299-314, Springer, Berlin Heidelberg, , Germany, 2013.

[39] Wu H., He Z., Gong J. , "A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes", Computers, Environment and Urban Systems, vol. 34, pp. 291-298, 2010.

[40] Xie J., Zhang L., Li J., "Automatic Simplification and Visualization of 3D Urban Building Models", International Journal of Applied Earth Observation and Geionformation, vol. 18, pp. 222–231, 2012.

[41] http://webapps.comune.trento.it/ambiente/ [accessed: 2013-02-12].