IMEKO International Conference on
Metrology for Archaeology and Cultural Heritage
Lecce, Italy, October 23-25, 2017

Semi-automatic segmentation of architectural 3D models with semantic annotation and Web fruition

Giorgio De Nunzio[1], Marina Donativi[1]

[1] *Dipartimento di Matematica e Fisica "Ennio De Giorgi", University of Salento, Lecce (Italy), {giorgio.denunzio | marina.donativi}@unisalento.it*

***Abstract* – Semantic segmentation of 3D models of ancient and historic buildings is an important modern Cultural Heritage topic. This work reports our preliminary results on the creation of a software system for partial semiautomatic semantic segmentation of building 3D models produced by photogrammetric surveys, and their fruition by Web technologies. These results were obtained in collaboration with Corvallis SPA (Padua – Italy, http://www.corvallis.it).**

I. INTRODUCTION

Nowadays computer technologies, such as 3D surveying, information systems, real-time model rendering, and virtual/augmented reality, allow enjoying and studying Architectural Heritage with a visual and integrated approach. 3D models in form of polygon meshes or point clouds can be built from reality by image and/or range data, and can later be segmented, classified, and annotated, then storing models and metadata in databases to be accessed via the World Wide Web. Professionals in the field as well as tourists can benefit from such techniques, the former for better heritage comprehension, management and protection, the latter for virtual tours before actual travels, as an alternative to the usual 2D souvenir photos, and to support crowdsourced 3D model databases.

In order to provide a high level of information, 3D digital representations (e.g. of a monument) need to be semantically enriched by adding annotations, which should be associated to the model as a whole, but also and above all specifically to its significant parts. Partitioning a model is a process known as segmentation. When the segments are meaningful, e.g. they reflect a (monument) decomposition into its standard architectural elements, and these elements are conveniently annotated, we obtain a semantic segmentation. As an example, an order [1] consists of three parts: the entablature, the column, and the crepidoma or the pedestal; each component can be further subdivided, e.g. the entablature consists of the cornice, the frieze, and the architrave; the column is composed of the capital, the shaft, and (not always) the base, and so on. The partition process goes on until we obtain "atomic parts" which are the basic elements of classical architecture. As to metadata, each of these parts has a name, it may have material information, a history, details on the conservation status, etc. Metadata should be assigned by respecting a field-specific ontology; the standard formal ontology for Cultural Heritage (CH) is the CIDOC Conceptual Reference Model (CRM) (http://www.cidoc-crm.org/) by the International Council Of Museums (http://icom.museum), which provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in CH documentation; extensions of CIDOC-CRM exist, more specific to Architecture, such as CRMba (http://www.cidoc-crm.org/crmba/home-7), created to support building documentation.

When semantics is expressed in standard ways, ideally the data is ready for the semantic Web and could be used by machines not just for display purposes, but also for automation, integration and reuse across various applications.

This work reports our preliminary results on the creation of a software system for partial semiautomatic semantic segmentation of building 3D models produced by photogrammetric surveys, and their fruition by Web technologies.

II. METHODS

This section reports some details on segmentation (subsection A) and annotation with Web fruition (subsection B).

A. Segmentation

A variety of algorithms, coded in C++ with PCL (the Point Cloud Library, [2]) and some Matlab parts for prototyping, were integrated in our software system so as to obtain the segmentation of buildings into (some of) their salient architectonic elements, such as stairs, columns (in turn broken up into capital, shaft and base), doors, windows. Many techniques exist for point cloud or mesh segmentation, such as region growing, model fitting, machine learning approaches [3]. Unfortunately, no single method is able to manage all the desired segmentation goals, so we designed our system with a modular approach, i.e. as a framework in which the implementation of new techniques might be easy. We used both ideas from the literature and our own algorithms, consistently integrating them in our system. Following this idea, a number of segmentation priorities were identified. Before all, horizontal and vertical plane identification was achieved by RANSAC (RANdom SAmple Consensus) [4-7]: this step is preliminary to all the other segmentation tasks, because it allows the identification of walls and the floor, and the ceiling if present in the model. Wall identification was important in order to prepare for the recognition of openings (doors and windows) [8] and to define the interior and the exterior of buildings. Stairs were identified by a generative approach, in which parametrically generated shapes are localized in the model by fitting procedures. Circular-section columns were identified by ground histogram calculation.

The segmentation process is of course not completely automatic, because some parameters must be set in accordance with the particular model features, and is not flawless. Tests are currently underway.

In the following paragraphs some more details and some results will be given concerning the detection and localization of straight stairs and round cross-section columns.

Straight stairs

The three main motivations for stairs detection are: (1) applications allowing robots to explore buildings with more than one floor (multi-storey path finding) [9–11], (2) stairs detection as an aid for the visually impaired [12–14], (3) semantic segmentation of Cultural Heritage 3D models [15, 16].

In the first two cases the point clouds are generally organized because they derive from RGB-D data, while CH applications use unorganized clouds coming from photogrammetry or laser-based reconstruction: this means that the algorithms for cases (1) and (2) are usually not directly applicable to case (3) [11], but they may anyway suggest methods valid for unorganized point clouds.

Various techniques are used in the literature. Some of them, like the approach described in this paper, rely on edge point extraction as the first step. For example in [9] edge points are detected in RGB-D images by the Concave Hull algorithm available in the PCL framework, then classification relies on depth and geometric information. In [10] a supervoxel clustering–based staircase extraction algorithm is proposed. Paper [11] develops a graph-based detection method for point clouds, which first segments planar regions and extracts the stair tread and riser segments. With these segments, a dynamic graph model is initialized that is used to detect stairs. See [11] also for a good review of some previous approaches useful for unorganized point clouds, and for a discussion on plane-based and edge-based methods. In [12], four types of obstacle: walls, doors, stairs, and a residual generic class of obstacles on the floor are detected in RGB-D data; stairs are found by searching for points on planes at increasing height from the ground, with a given step height (with a tolerance). In [13] depth maps, calculated from RGB-D or stereo data, are used to feed a classifier. The Authors of [14] propose a staircase detection algorithm in RGB-D data, based on a support vector machine (SVM): the candidates are detected from RGB frames by extracting parallel lines (by the Hough transform), and the depth frames are employed to classify the staircase candidates as upstairs, downstairs, and negatives (e.g., corridors).

Turning to Cultural Heritage, in paper [15] we find an application developed specifically for unorganized point clouds in the context of CH, in which the creation of "digital libraries" of objects is explored. The paper relies on a generative modeling approach, in which parameterized object models are built by GML, the Generative Modeling Language, and fitted to the point cloud (by subpart fitting without previous segmentation). Another generative and fitting approach is described in [16], in which an interactive framework to extract hi-level primitives (e.g. columns or staircases) from scanned 3D models is presented.

Our approach is inspired by [15], in that it is generative and ideally automatic, but it aims at minimizing the large computational effort necessary for 3D subpart fitting, by a preprocessing step which reduces the fitting problem to two dimensions, by detecting the horizontal straight edges in the point cloud and then considering only the pattern of their mean points.

Before exposing the details of our stairs detection algorithm, and to define its application context and limits, it is useful to give the relevant terminology and to describe the various types of stairs as classified by geometrical considerations [17].

Stairs are composed of series of steps (flights) with landings at appropriate intervals. Each step consists of a tread (the horizontal part, with its depth and width) and a riser (the vertical part between treads, with its height).

According to the diverse arrangements of the steps, we can distinguish: straight stairs (they may consist of either one single flight or more than one flight with landings, with no change in direction; they can also allow a change in direction, in which case we have parallel or angle stairs, such as quarter-turn stairs, half-turn or dogged-legged stairs, etc); circular stairs; spiral stairs, and others. Some Authors consider as straight stairs only those without direction change. We decided to concentrate on

straight stairs, with or without direction change, so that we could assume parallel stairs edges.
In our approach, a parametric generative model was chosen for its versatility. We decided to reduce the problem complexity by projection to a 2D space before attempting model fitting. For this purpose, our algorithm first detects the most significant edge points in the cloud (Step 1) and finds the straight edges by RANSAC (Step 2); then only the horizontal edges are preserved and clustered into groups of parallel segments (Step 3); the clusters are projected to a plane perpendicular to the segments, so getting a spatial pattern of points representing the cluster (Step 4); finally a parameterized generative model of stairs is fitted to the point patterns (Step 5), giving a figure of merit and the stairs parameters; if the fit is satisfying, the position of the found stairs in the original 3D world is found, and a bounding box is visualized in the cloud. The procedure steps are detailed below.

Step 1: Edge points are detected by the method described in [18] in which, instead of following the common approach of using geometric features such as normal vectors and curvatures, the Authors propose a fast and precise method to detect sharp edge features by analysing the eigenvalues of the covariance matrix that are defined by each point *k*-nearest neighbors. The C++/PCL code for edge detection was kindly shared by the Authors.

Step 2: RANSAC is used to detect straight lines in the edge point cloud.

Step 3: the edges are clustered, using as features the stair direction (one of the two non-zero direction cosines of the edges) and the three coordinates of the edge center. Several clustering algorithms were tested The most performing method, i.e. the one that clustered the stairs edges in the most correct way, collecting in a same group almost all and only the detected edges pertaining to each flight of stairs, was DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[1] [19], a density-based clustering algorithm which can potentially identify clusters of any shape in a data set containing noise and outliers. A larger weight was given to the direction feature because only strict parallel edges could belong to the same stairs (no spiral or circular stairs!).

Step 4: For each cluster, a plane is now defined, normal to the cluster edges, and the edge centers are projected onto it, so giving a planar pattern of points representing the possible stairs. The coordinates of these points on the plane are defined by introducing a pair of basis vectors on the plane, i.e. a $(\boldsymbol{u}, \boldsymbol{v})$ pair of orthogonal unit vectors. Of course there is no unique way to define $\boldsymbol{u}$ and $\boldsymbol{v}$, but a convenient way may be as follows. Say $\boldsymbol{n} = (a, b, c)$ is the plane normal unit vector: in our case, a is the average x direction cosine of the edges in the cluster, $b \sim 0$ (y being the vertical axis, and the edges being horizontal), and $c = \text{sqrt}(1 - a^2)$. Then we can set:

$$\begin{aligned} \boldsymbol{u} &= (c, 0, -a) \\ \boldsymbol{v} &= \boldsymbol{n} \times \boldsymbol{u} \end{aligned} \quad (1)$$

where the symbol × represents the cross product.
We now project each edge center (x_c, y_c, z_c) onto the $(\boldsymbol{u}, \boldsymbol{v})$ plane and calculate its (u_c, v_c) coordinates by dot product:

$$\begin{aligned} u_c &= \boldsymbol{u} \cdot (x_c, y_c, z_c) \\ v_c &= \boldsymbol{v} \cdot (x_c, y_c, z_c) \end{aligned} \quad (2)$$

where the origin of the (u, v) coordinates is the world origin (0, 0, 0). Each point ideally comes from an edge, so marking the transition from riser to tread, or from tread to riser. This pattern can contain noise (points due to incorrect edges) and may miss some points related to edges that were not found in Step 2.

Step 5: A point pattern is generated as a stairs model (each point representing the projection of a stairs edge as in step 4), depending on the following stairs parameters: the number of steps, riser height, tread depth, 3D stair position (i.e. the position of the center of first edge of the lowest step). The model is fitted to the patterns obtained in step 4, and the optimized parameters as well as the value of the objective function as a figure of merit are returned. The returned values are then automatically evaluated in order to accept or reject the cluster as representing stairs. The point fitting algorithm which gave the best results in terms of robustness and accuracy was Coherent Point Drift (CPD) [20]. Figure 1 shows an example of model matching.

After the stairs are detected, the full model consists of the following parameters: lower-step center coordinates, stairs direction parameter in the horizontal *xz* plane, the number of steps, tread depth, riser height, step width.

The procedure was tested on synthetic and real-world models. Step 5, in particular, was tested against synthetic point patterns, and the fitting procedure was stressed by adding noise to the ideal point position, by inserting points not corresponding to stairs edges, and by removing significant points.

Figure 2 shows the application of the algorithm to a synthetic case of study.

[1] http://yaikhom.com/2015/09/04/implementing-the-dbscan-clustering-algorithm.html

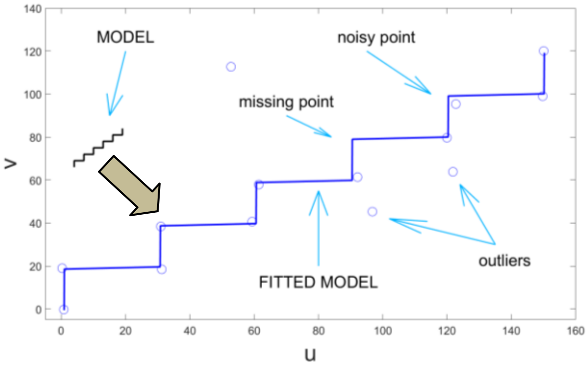

Fig. 1. Point-pattern fitting. The small circles are the real-world points obtained by projecting the edges centers (in an edge cluster) onto a plane perpendicular to the edges (step 4; u *and* v *are the coordinates in the projection plane). There are some outliers (corresponding to erroneous edges), points are noisy, and some may be missing. The model is on the left. The fitted model perfectly reconstructs the stairs silhouette.*

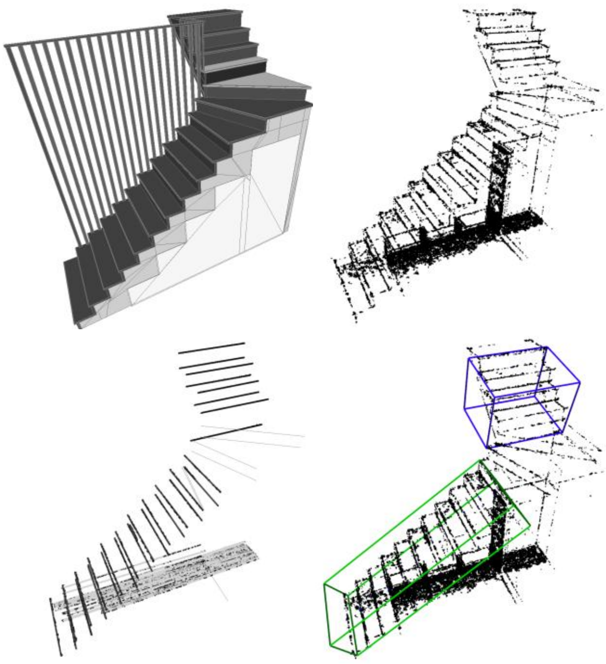

Fig. 2. A case study. The stairs model (top-left picture) was downloaded from https://archive3d.net (model N290716). The model was converted to a point cloud by the open-source CloudCompare software (http://www.cloudcompare.org/), the edge points were found (top-right image), then the straight edges were detected and the horizontal ones were clusterized (bottom-left picture; light-grey edges are those rejected as outliers). The straight stairs were then detected as explained in the text, and automatically highlighted with bounding boxes (bottom-right picture).

<u>Round cross-section columns</u>

Few papers in the literature deal with column detection for CH 3D scenes. The already cited Ref. [15] applies generative models also to columns and arcades, while in [21] round and rectangular cross-section columns are located by ground histogram calculation. For the sake of performance, we chose the latter approach, which decreases the problem complexity from 3D to 2D. We decided to limit our tests to circular columns for now.
Ground histograms were obtained by projecting the point cloud to the ground plane, and by evaluating point density on the plane. The vertical structures present in the point cloud left easily recognizable signatures in the point-density map, e.g. circular columns produced circular shapes that could be quite straightforwardly located by the Circular Hough transform (CHT), leading to column detection and segmentation. Columns were then partitioned into their constituent parts. Details on the procedure follow.

Step 1: after downsampling, the point cloud is projected to the ground plane.

Step 2: with the purpose of calculating point density in the projected cloud, and finding the signature of vertical structures, an octree is built out of the cloud, and box-searching is performed in each cell so as to measure each cell occupancy; an image is then built, containing a pixel per octree cell, with value proportional to point density in that cell.

Step 3: the Circular Hough Transform [22] is applied to the image created in Step 2, so as to find circles if present, i.e. the signatures of columns.

Step 4: the centers and radii of the found circles give information on the horizontal coordinates of the columns, but nothing is known about their vertical position and size: therefore, a region, centered on each column center and slightly larger than the column diameter, is searched for a cylinder by RANSAC; looking for cylinders only in the regions located by CHT reduces false positives.

Step 5: after mesh building, the columns are finally segmented into their constituent parts (capital, shaft and base) by convex decomposition [23]; this algorithm computes a hierarchical segmentation of the mesh triangles by applying a set of decimation operations to the mesh dual graph, guided by a cost function related to concavity: the generated segmentation is then exploited to construct an approximation of the original mesh by a set of convex surfaces; the generated convex hulls can be used as space filters to segment the original points of the point cloud. Segmentation is not always accurate enough, causing small segments under the column base or in the architrave – beside the three main column parts – to be returned. To overcome this problem, the longest segment is identified and assigned to the shaft, and the two segments connected to it from above and from below are selected as the capital and the base respectively, disregarding the other parts if present. Of course, segmentation quality is affected by the model quality.

Figure 3 shows the application to a model of the Gerrard Hall (North Carolina) reconstructed by photogrammetry from a set of about 100 pictures available at https://colmap.github.io/datasets.html (see the figure caption for details).

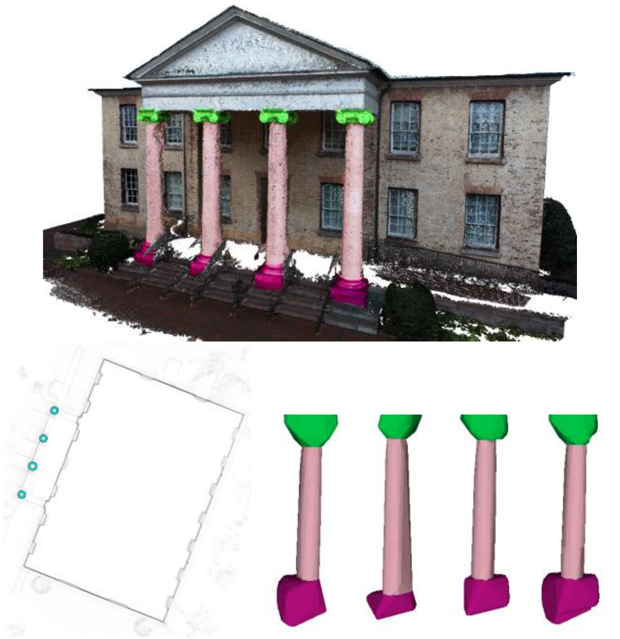

Fig. 3. The Gerrard Hall (North Carolina) model, after column detection and segmentation into the component parts. Top: a view of the model, with the segmented columns; bottom-left: the density map of the model projected to ground, after CHT application (the column signatures are shown); bottom-right: the four columns segmented into capital, shaft, and base (the convex hulls produced by the segmentation algorithm are shown).

B. Semantic annotation and Web fruition

After segmentation, the system allows annotation according to a limited set of standard CIDOC-CRM classes and properties, e.g. respectively entity E22 (Man-Made Object) and property P46 (is composed of), for future compatibility with the semantic Web. The models are then converted into the progressive-mesh Nexus file format [24, 25]. Progressive meshes allow a smooth choice of LOD (Level of Detail) depending on the current view (i.e., model distance, framing, device power). With this approach it is possible to initially display a model with the lowest LOD and then let it gradually show more and more details, with effective real time visualization. As to Web fruition, our HTTP pages are dynamically built based on 3DHOP [26] for real-time model drawing, and some javascript code for semantic annotation visualization by an information tree linked to the model: a click on the model hotspots, defined by the previously segmented architectural elements, allows to select the corresponding tree node, and vice versa (see Figure 4).

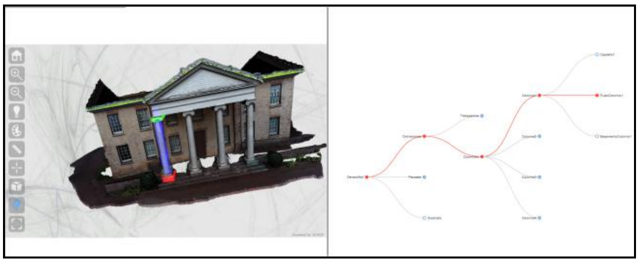

Fig. 4. The Gerrard Hall model inserted in our Web system prototype, using 3DHOP and some javascript code for tree management. Hotspots were inserted at column parts, and used as links to the semantic-structure tree shown on the right. Clicks on the model allow navigating the tree.

III. CONCLUSIONS

A software framework for the semantic segmentation and Web fruition of Cultural Heritage 3D models produced by photogrammetry was presented. Two segmentation modules (straight stairs and circular-section columns) were detailed, and the Web system prototype was shown. The software proved successful in a number of monument models and synthetic point clouds. Nonetheless the framework is still under development, with many segmentation modules relying on parameter tuning. The development will continue, with new modules added to the framework, and limitations hopefully removed.

ACKNOWLEDGEMENTS

The results reported in this work were obtained in collaboration with Corvallis SPA (Padua – Italy, http://www.corvallis.it).

REFERENCES

[1] Tzonis, L. Lefaivre,. “Classical Architecture: The Poetics of Orders”, 1987, The MIT Press.
[2] R.B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL)”, IEEE Int. Conf. on Robotics and Automation (ICRA), May 9-13, 2011, Shanghai, China
[3] E. Grilli, F. Menna, F. Remondino, “A review of point cloud segmentation and classification algorithms”, The International Archives of the

Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2/W3, 2017, 3D Virtual Reconstruction and Visualization of Complex Architectures, 1–3 March 2017, Nafplio, Greece.

[4] M.A. Fischler and R.C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, Communications of the ACM, vol. 24, issue 6, June 1981, pp 381-395, doi:10.1145/358669.358692.

[5] P. Tang, D. Huber, B. Akinci, R. Lipman, A. Lytle, “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques”. Automation in Construction Vol 19, 2010, pp 829–843.

[6] K. Chen, Y.-K. Lai, S.-M. Hu, “3D indoor scene modeling from RGB-D data: a survey”, Computational Visual Media Vol. 1, No. 4, December 2015, 267–278.

[7] V. Pătrăucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, C. Haas, “State of research in automatic as-built modelling”, Advanced Engineering Informatics, 29 (2), 2015, pp. 162-171, ISSN 1474-0346, http://dx.doi.org/10.1016/j.aei.2015.01.001.

[8] Martinović, J. Knopp, H. Riemenschneider, L. Van Gool, “3D All The Way: Semantic Segmentation of Urban Scenes From Start to End in 3D”, 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).

[9] L.A.V. Souto, A. Castro, L.M.G. Gonçalves, T.P. Nascimento, “Stairs and Doors Recognition as Natural Landmarks Based on Clouds of 3D Edge-Points from RGB-D Sensors for Mobile Robot Localization, Sensors 17, 1824; 2017.

[10] K.-W. Oh and K.-S. Choi, “Supervoxel-based Staircase Detection from Range Data ”, IE Transactions on Smart Processing and Computing, vol. 4, no. 6, 2015.

[11] T. Westfechtel, K. Ohno, B. Mertsching, D. Nickchen, S. Kojima, S. Tadokoro, “3D Graph Based Stairway Detection and Localization for Mobile Robots”, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon Convention Center, October 9-14, 2016, Daejeon, Korea

[12] H.-H. Pham, T.-L. Le, N. Vuillerme. “Real-Time Obstacle Detection System in Indoor Environment for the Visually Impaired Using Microsoft Kinect Sensor”, J Sensors, Vol. 2016, Art. id 3754918.

[13] S. Cloix, G. Bologna, V. Weiss, T. Pun, D. Hasler, “Low-power depth-based descending stair detection for smart assistive devices”, EURASIP Journal on Image and Video Processing, 2016, 2016:33.

[14] R. Munoz, X. Rong, Y. Tian, “Depth-aware indoor staircase detection and recognition for the visually impaired”, 2016 IEEE Int. Conf. on Multimedia & Expo Workshops (ICMEW), 11-15 July 2016.

[15] T. Ullrich, V. Settgast, D.W. Fellner, “Semantic Fitting & Reconstruction”, ACM Journal on Computing and Cultural Heritage 1 (2), 2008.

[16] D. Portelli, F. Ganovelli, M. Tarini, P. Cignoni, M. Dellepiane, R. Scopigno, “A framework for User-Assisted Sketch-Based Fitting of Geometric Primitives”, Proc. of WSCG, 18th Int. Conf. on Computer Graphics, Visualization and Computer Vision, 2009.

[17] S.D. Edgett and A.M. Williams, “Vertical Circulation”, in: Merritt FS and Ricketts JT (editors), Building design and construction handbook, 6th edition, McGraw-Hill, ISBN 0-07-041999-X (Section 16), 2001.

[18] D. Bazazian, J.R. Casas, J. Ruiz-Hidalgo, "Fast and Robust Edge Extraction in Unorganized Point Clouds", Proc. of Digital Image Computing: Techniques and Applications (DICTA), 2015 Int. Conf. on.

[19] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise”. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, 1996, pp. 226–231. ISBN 1-57735-004-9.

[20] Myronenko and X. Song, "Point-Set Registration: Coherent Point Drift", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, issue 12, 2010, pp. 2262-2275.

[21] L. Díaz-Vilariño, B. Conde, S. Lagüela, H. Lorenzo, “Automatic Detection and Segmentation of Columns in As-Built Buildings from Point Clouds”, Remote Sens. 2015, 7, 15651-15667.

[22] R.O. Duda, P.E. Hart, “Use of the Hough Transform to Detect Lines and Curves in Pictures”, Communications of the ACM 15, 1972, pp. 11– 15.

[23] K. Mamou, and F. Ghorbel, “A simple and efficient approach for 3D mesh approximate convex decomposition”, Image Processing (ICIP), 2009 16th IEEE International Conference on.

[24] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, R. Scopigno, "Batched Multi Triangulation”, Proc. IEEE Visualization, pp 207-214, October 2005.

[25] F. Ponchio, and M. Dellepiane, “Fast decompression for web-based view-dependent 3D rendering”, Web3D 2015. Proceedings of the 20th International Conference on 3D Web Technology, pp 199-207

[26] M. Potenziani, M. Callieri, M. Dellepiane, M. Corsini, F. Ponchio, R. Scopigno, “3DHOP: 3D Heritage Online Presenter”, Computers & Graphics Volume 52, November 2015, Pages 129-141.