Pacific Graphics (2014)
J. Keyser, Y. J. Kim, and P. Wonka (Editors)

Scene Segmentation and Understanding for Context-Free Point Clouds

S. Spina[1] and K. Debattista[1] and K. Bugeja[1] and A. Chalmers[1]

[1]WMG, University of Warwick, Coventry, UK

Abstract

The continuous development of new commodity hardware intended to capture the surface structure of objects is quickly making point cloud data ubiquitous. Scene understanding methods address the problem of determining the objects present in a point cloud which, dependant on sensor capabilities and object occlusions, is normally noisy and incomplete. In this paper, we propose a novel technique which enables automatic identification of semantically meaningful structures within point clouds acquired using different sensors on a variety of scenes. The representation model, namely the structure graph, with nodes representing planar surface segments, is computed over these point clouds to help with the identification task. In order to accommodate for more complex objects (e.g. chair, couch, cabinet, table), a training process is used to determine and concisely describe, within each object's structure graph, its important shape characteristics. Results on a variety of point clouds show how our method can quickly discern certain object types.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: General—I.3.5 [Computer Graphics]: Boundary Representation—I.3.8 [Computer Graphics]: Applications—

1. Introduction

The widespread availability of inexpensive acquisition hardware and photogrammetry-based tools like Microsoft PhotoSynth [SSS06] and ARC3D [VG06], which are capable of capturing or extrapolating depth information from a scene, is leading to the creation of massive repositories of point clouds. Rapid advances in ubiquitous computing have made available to the masses the possibility of capturing the world around us using smart phones and tablet devices [Goo14, Jar14] and synthesising it into point clouds. As a minimum, these point clouds contain a discretised representation of surfaces from the acquired scene, in the form of a set of coordinate triples. This paper addresses the problem of point cloud segmentation and understanding, where meaningful structures and objects, such as walls and chairs, are automatically identified and extracted. Previous work has targeted specific domains to the effect of making it very efficient within a specific context; however, this comes at the cost of limiting its applicability to other scenarios or the more general cases [NXS12, LGZ*13]. The method presented is founded upon the observation that many objects present in a target scene, particularly man-made objects, can be partitioned into a number of planar segments exhibiting specific connectivity patterns amongst them. An object description can be built from at least one planar surface segment and its relationship to the remaining points. Objects not suitable for such representation are automatically identified and flagged for consideration using other schemes such as those based on local surface descriptors. The problem of identifying generic structures is tackled by partitioning point clouds into connected typed segments, enumerated as `planar`, `edge`, or `complex`, over which a structure graph is constructed. Subsequently, a number of nodes representing `planar` segments in the structure graph are enhanced with oriented sparse volume grids to enable the extraction of previously trained objects. The main contributions are:

1. A point cloud segmentation pipeline which partitions raw point data of both indoor and outdoor scenes into connected segments suitable for scene analysis
2. A graph-based representation describing salient geometric features in a point cloud and their connectivity
3. An incremental scene understanding algorithm which enumerates the space of solutions mapping objects to surface segments in the target scene.

DOI: 10.2312/pgs.20141244

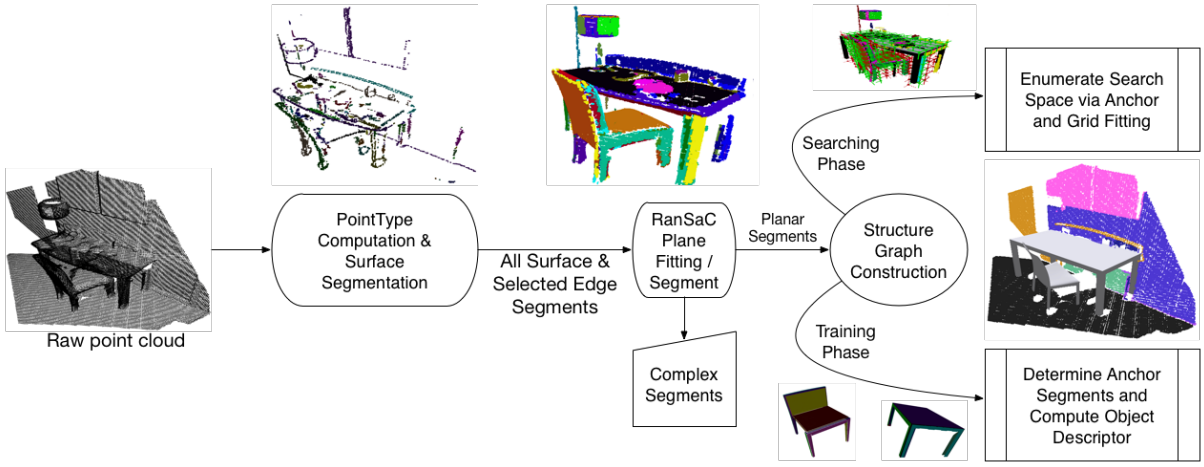

Figure 1: *Scene Segmentation and Understanding Pipeline Overview*

2. Related Work

Object segmentation [GWM01, CGF09, LVB*12] and retrieval [DA10, LGA*12] have been extensively studied. Many segmentation algorithms apply the RanSaC paradigm to fit parametric shape primitives to unstructured point clouds [DN07, SWK07]. Segmentation is usually required for shape recognition [SWWK08, GKF09, LGZ*13] and indoor scene understanding [NXS12, KMYG12, MPM*14]. Graph-based 3D object descriptors have been used to encode geometric and topological properties from the shapes extracted [SWWK08, GKF09]. Both supervised and unsupervised learning algorithms have been applied to search for object descriptors within point clouds. Whereas supervised methods utilise a training phase in order to synthesise descriptors of individual objects, unsupervised methods rely on the presence of patterns to automatically infer similar objects in a scene. In our work, we combine a supervised learning component with unsupervised methods. For instance, the boundaries of an indoor scene are first inferred by searching for specific patterns, then objects using previously trained descriptors. Golovinsky et al. [GF09] present a segmentation algorithm for outdoor scenes based on foreground/background identification. Indoor scenes however, usually present a harder segmentation challenge due to noise induced by added clutter, sensors and partial object occlusions. Mattausch et al. [MPM*14] exploits similarities within indoor scenes to segment point clouds into clusters of similar objects. When these similarities are absent, for instance due to low quality acquisition sensors, the effectiveness of these techniques diminishes. In our work, we utilise graph-based object descriptors to capture the geometric properties of an object as connectivity patterns between planar segments and then search for similarities with these trained object descriptors. With supervised methods, scene-specific knowledge may be embedded in trained object descriptors. Kim et al. [KMYG12] propose a system which also handles model variability modes. As opposed to our method however, they assume that the vertical direction of the models and the scene are fixed. This makes it difficult to detect overturned objects as opposed to our method which orients models in a scene according to the identification of dominant planar segments of the trained object descriptor. Nan et al. [NXS12] propose a search-classify approach for interleaving segmentation and classification. Although managing to successfully classify complex scenes, their method fails when object placement in the scene differs in pose and scale to that used when training the scene-specific classifier. In our work, classification does not depend on the original pose of the training models; instead, connectivity patterns between planar segments are used to identify objects and structures. Additionally, grids computed around the dominant planar segments of objects are used to discriminate between objects which have similar plane connectivity patterns. Shao et al. [SXZ*12] propose an interactive approach to indoor scene understanding, where users manually improve segmentation results prior to identification. We seek to provide a method for scene understanding with minimal input from the user. The techniques presented in this work do not rely on a specific context; this makes them applicable to a wide spectrum of domains, including indoor scenes, LiDaR data and cultural heritage sites amongst others.

3. Method

This section describes the transformation of a point cloud data set into a *structure graph* and the use of the latter in scene understanding (see figure 1).

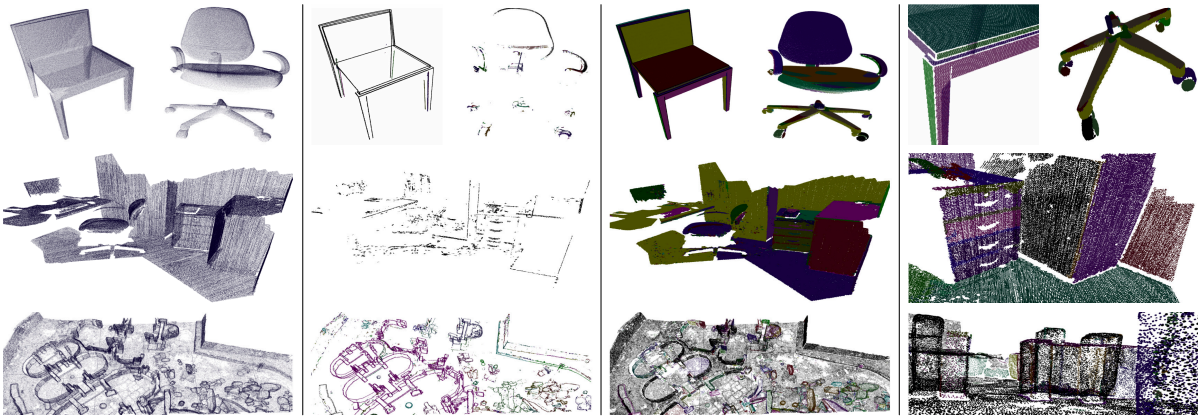

Figure 2: *Segmentation process on two separate chairs, office (Nan et al. [NXS12]) and outdoor scene with columns from left to right - all points,* `edge` *segments, segmentation results shown as coloured* `planar` *segments, close-up view.*

Structure Graph Construction: Before segmentation takes place, each point in the input data set $\mathcal{P}$ is classified as either *surface* or *edge*; this is shown in figure 2, first three columns. This property is determined by the ratio of the eigenvalues over the point's k nearest neighbours. The labelled points are then grouped together into `surface` or `edge` segments using an area growing algorithm. Figure 2, third column, illustrates the `surface` segments resulting from this process; each segment is visualised using different colours. The `surface` segments are refined using RanSaC plane fitting, to ensure that each individual segment is as close to a planar surface as possible. Figure 3 illustrates how the seats of the sofas are further split into two `planar` segments to approximate slight curvature. If RanSaC is not able to fit the `surface` segment into any `planar` segments, the segment is marked as `complex` and is currently withdrawn from being processed further. Since RanSaC is applied to `surface` segments, which are point subsets of $\mathcal{P}$, the results are considerably less random than applying RanSaC on all of $\mathcal{P}$. Given this set partition of $\mathcal{P}$, a structure graph $\mathcal{G}$ describing segment adjacency is created, where each node represents either a `planar` or `edge` segment. Adjacency is determined by intersecting OBBs computed over `planar` segments. Each `planar` node is augmented with additional information including the number of points, orientation, points coverage on plane and spatial context information. Spatial context is used to determine the approximate location of the `planar` segment (ranging from *boundary* to *central*) along its normal within the object or scene. Points coverage measures how spatially uniform the points are located on the `planar` segment.

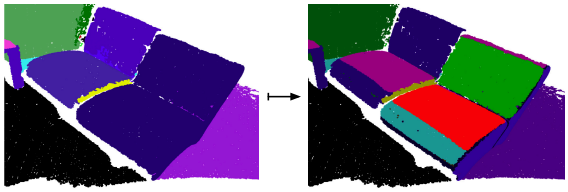

Figure 3: *Office Scene Nan et al. [NXS12] - over segmentation (left) and new segments after RanSaC (right).*

Object Descriptors: Structure graphs form the basis of object descriptors and are independent of object pose. A single structure graph representation can be used to describe similar objects with different poses, such as non-uniformly scaled chairs. They are also robust to noise in point clouds, as shown in figure 4. A *feature*, itself a structure graph, is used to describe a specific connectivity pattern across segments. Handcrafted features are used to describe regular structures within a scene. For instance, a typical flight of stairs in a room can be described in a straightforward manner as a sequence of connected orthogonal `planar` segments. A tree, such as those present in the 5th row of figure 6, is described as an `edge` segment (leaves) above `planar` segments connected in a cylindrical pattern (trunk). Graph matching algorithms are used to search for features in $\mathcal{G}$ representing the target scene. In indoor scenes, the floor and walls are described as a feature where each node is orthogonal to each other and their spatial context is set to *boundary*. The identification of objects (e.g. chairs, tables, pots, houses as in Lin et al. [LGZ*13]), requires a training process intended to automatically produce more complex features. Figure 5 shows the models used for the evaluation of indoor scenes, none of which is specifically

present in the target scenes. A ray casting process, from multiple views around an object 3D mesh, is used to produce structure graphs from view dependent point clouds of the object trained. The information gathered about surface relationships of the object is merged together into one structure graph with additional information including most visible surface and pair-wise segment occlusions. This information is used to select a small number of *salient* `planar` segments extracted from the object, referred to as *anchors* and used as root nodes of object features. Anchors have a higher probability of being visible in the target scene. Additional transitions are added in $\mathcal{G}$ to describe the connectivity between anchors which together define the *support* of the object. The support of an object represents local `planar` segment connectivity which is used to quickly give an indication of whether an object is present in the target scene. In addition to connectivity information, a sparse voxel grid is computed around each anchor to approximate the shape of the object around the anchor. This is used whilst searching to determine whether the segments identified using an object's support and features actually correspond to that object. The grid is oriented in world-space by the orthonormal basis formed by the anchor segment OBB (e.g. second row of figure 5). Whereas increasing the grid resolution improves the grid approximation to the shape of the object, in order to improve performance and genericity, a low resolution grid is used to capture only salient shape features without capturing too much detail. Each grid cell stores information about which object segments it contains.

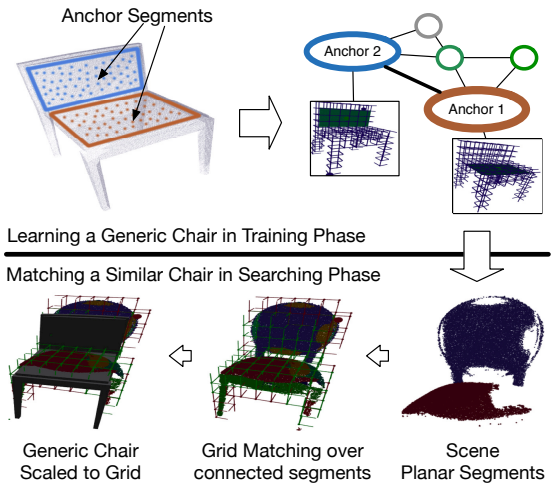

Figure 4: *An trained objects' structure graph is used to locate similar objects in a target scene.*

Scene Understanding: We reformulate the scene understanding problem as one which seeks to maximise matches between `anchor` segments in object descriptors and `planar` segments in the target scene. In general, this is bound to be an unconstrained problem, due to the presence of noise and partial object occlusions, where multiple valid mappings may exist. The solution space is defined as the Cartesian product between the set of objects used and the super-set of `planar` segments in the structure graph of the target scene. A Markov decision process enumerates this search space using a number of heuristics intended to quickly provide a number of valid solutions. The process creates a solution lattice $\mathcal{L}$, where each solution associates objects to sets of scene `planar` segments and is obtained via a depth first traversal of $\mathcal{L}$. Each leaf node describes a solution, whose score is an aggregate of the scores obtained at all inner nodes (individual object mappings) along each depth first traversal path. In order to decrease the number of solutions, an additional constraint on the number of inner node children can be imposed. If this parameter is set to one, then only one solution is produced. The order in which `planar` segments in the scene are matched with anchor segments plays a critical part in the efficiency of the scene understanding process. If domain-specific knowledge of the target environment such as the distance from the floor of the chair seats and table tops is known, then a segment sorting function can order horizontal `planar` segments according to their distance from the floor and try to match these with tables and chairs. In order to provide for a generic scene understanding solution, our method allows for different sorting function implementations to determine the sequence by which `planar` segments from the target scene are visited. If no domain-specific information is available, as in our case, `planar` segments are ordered according to their similarity with the descriptors of trained objects. Whereas anchor connectivity information is used to determine which scene `planar` segments to visit first, voxel grids created around these segments are used to determine which other scene `planar` segments make up the object and further discriminate between similar objects (e.g. a sofa and a chair). An incremental grid matching process is used to determine which `planar` segments in the scene best fit within an objects' voxel grid. At each step, a grid is computed around the segment matching the anchor and a number of connected segments. A compatibility score between scene and object grids is computed as a difference of the two. Non-uniform scaling and rotations around the normal of the current scene `planar` segment are performed until all points in the segments being tested are included. If the score decreases when adding a new connected segment, this is removed and other segments are added according to the structure graph of the target scene. Finally, when the best scene voxel grid is chosen, `edge` segments from the connectivity graph connected to the `planar` segments selected and any which fall within the OBB of the scene voxel grid are tested to check whether they consolidate the match. If the distance between two mappings is small (user-set parameter), a tie-breaker function is used to select the object mapping which according to some heuristic has the high-

est probability of occurring, e.g. always prefer upright pose. The tie-breaker is not used if multiple solutions are allowed, in which case all mappings are attached to $\mathcal{L}$. In general, if more constraints are added to the environments, e.g. more points, reduction in noise, domain specific knowledge, etc., our matching algorithm should be able to perform better.

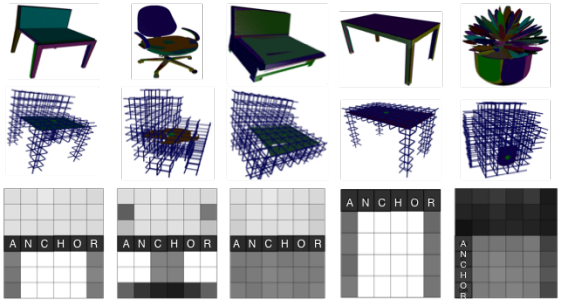

Figure 5: *The top row shows objects trained; middle row shows voxel grid approximating object shapes around one anchor segment and the third row illustrates a density map of the grid parallel to the anchor.*

4. Evaluation

Models of two chairs, a table, a couch, a plant pot and a cabinet (see figure 5) are used to evaluate our method on a number of scenes including some taken directly from Nan et al. [NXS12]. The `planar` segments sorting function used first searches the scene structure graph for segment connectivity patterns matching the support of trained objects. Those matching the highest number of anchors (maximum 3) per object are tested first. In the case of ties, segment point count is used, testing first those segments with the highest number of points. In all cases, the `planar` segment with the highest number of points within the segments matching an object support is matched against the respective object anchor segment. Segments matched are removed from the rest of the search. Figure 7 illustrates the matching order for that specific scene. Couches are all correctly matched except for one, segment three, since the segmentation process groups together the back of two couches into one as illustrated in figure 8. In this case, grid matching elongates the couch. The top row of figure 6 illustrates matches between the office chair, a table and three filing cabinets. In the case of the filing cabinets, a third cabinet (the largest) is erroneously matched to part of the wall since two large `planar` segments are not included in the set of boundary segments as they are located within the room. The office chair is correctly identified and obtains a higher score when grid matching because of the segments representing the arm rests as can be seen in figure 8 (top left). The second scene with five chairs is taken from Nan et al. [NXS12] and is used to demonstrate that our method, as opposed to theirs, can detect similar objects in a different pose to the one used for training. The bottom row scene of figure 6 illustrates a low density point cloud with all the main objects correctly identified. Since our scene understanding process matches anchor segments to `planar` segments in the scene, severe occlusion and noise pose a limitation. In general, our method needs at least one anchor segment to be mostly visible even if with holes, in addition to some supporting segments around it. Figure 8 (top right), shows a scene where two chairs have only their back visible. Whereas our segmentation process does a good job at clustering these points as separate segments, currently, these segments are not considered whilst searching since there are no other segments in their vicinity to match any of the trained objects. However, if the segment sorting function is modified to always assume that backs of chairs are in a certain pose, then just one `planar` segment in the scene could be enough for matching.

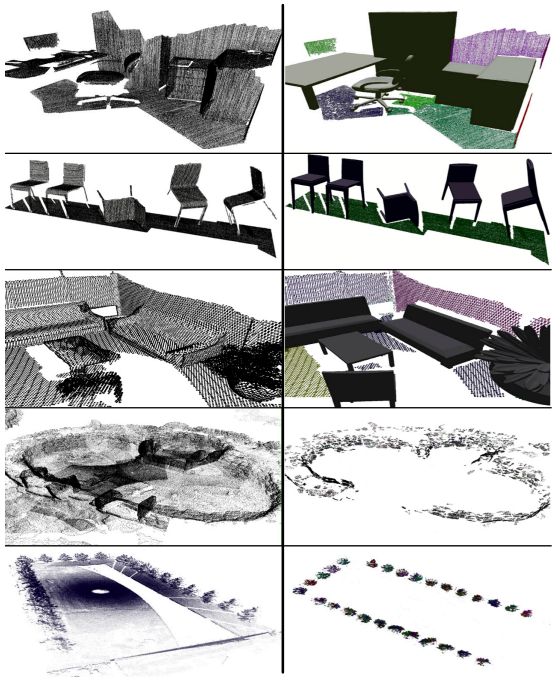

Figure 6: *Scenes in the top two rows taken from Nan et al. [NXS12]; third row scene scanned using Asus Xtion Sensor; fourth row show feature extraction from the temple point cloud; fifth row show feature (representing trees) extraction from the garden point cloud.*

5. Conclusion and Future Work

Further work is planned in order to improve performance, robustness and functionality. We seek to re-implement voxel

grid matching using GPUs to bring our method closer to a real-time realisation of scene understanding, as opposed to the current off-line process. We plan to investigate the integration of run-time context-switching (e.g. Fischer et. al. [FSH11]) whilst searching for objects in specific locations. Moreover, previously established relationships between objects can be used in cases of extensive occlusion and noise. Finally, integration with other representation and identification schemes which are more adequate for certain types of objects (e.g. computer vision for face recognition) should prove to be beneficial in our context and thus plan on integrating this information within structure graphs.

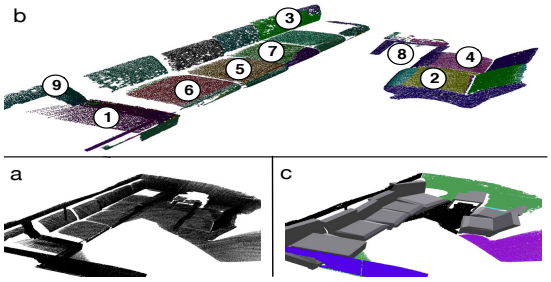

Figure 7: *(a) Point Cloud from Nan et al. [NXS12] (b) Planar segments filtered by boundary, numbers indicate segment order used for fitting (c) Models fitted*

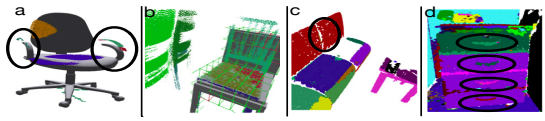

Figure 8: *(a) Model mesh and point cloud segments super imposed showing arm rests in point cloud matching to arm rests in trained office chair improving grid match score (b) Not enough data is present to correctly match two chairs whereas the one with more points is correctly matched to the generic chair (c) Over segmentation groups together the backs of two couches (d) Segments representing the handles on the drawers of the cabinet could be used to orient the model correctly.*

References

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3d mesh segmentation. SIGGRAPH '09, ACM. 2

[DA10] DARAS P., AXENOPOULOS A.: A 3d shape retrieval framework supporting multimodal queries. *Int. J. Comput. Vision 89*, 2-3 (Sept. 2010), 229–247. 2

[DN07] DORNINGER P., NOTTHEGGER C.: 3d segmentation of unstructured point clouds for building modelling. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2007), Institute of Photogrammetry and Cartography Technische Universitaet Muenchen. 2

[FSH11] FISHER M., SAVVA M., HANRAHAN P.: Characterizing structural relationships in scenes using graph kernels. SIGGRAPH '11, ACM. 6

[GF09] GOLOVINSKIY A., FUNKHOUSER T.: Min-cut based segmentation of point clouds. In *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV* (2009). 2

[GKF09] GOLOVINSKIY A., KIM V., FUNKHOUSER T.: Shape-based recognition of 3d point clouds in urban environments. In *Computer Vision* (Sept 2009). 2

[Goo14] GOOGLE: Project tango @ONLINE, June 2014. URL: https://www.google.com/atap/projecttango. 1

[GWM01] GUMHOLD S., WANG X., MACLEOD R.: Feature extraction from point clouds. In *In Proceedings of the 10th International Meshing Roundtable* (2001), pp. 293–305. 2

[Jar14] JARED D.: Structure offers 3d scanning right on your ipad @ONLINE, Feb. 2014. URL: http://www.imore.com/ceslive-scan-your-world-structure-sensor. 1

[KMYG12] KIM Y. M., MITRA N. J., YAN D.-M., GUIBAS L.: Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics 31*, 6 (2012), 138:1–138:11. 2

[LGA*12] LI B., GODIL A., AONO M., BAI X., FURUYA T., LI L., LOPEZ-SASTRE R., JOHAN H., OHBUCHI R., REDONDO-CABRERA C., TATSUMA A., YANAGIMACHI T., ZHANG S.: SHREC'12 Track: Generic 3D Shape Retrieval. In *EG2012* (2012), pp. 119–126. 2

[LGZ*13] LIN H., GAO J., ZHOU Y., LU G., YE M., ZHANG C., LIU L., YANG R.: Semantic decomposition and reconstruction of residential scenes from lidar data. *ACM Trans. Graph. 32*, 4 (July 2013), 66:1–66:10. 1, 2, 3

[LVB*12] LAVOUE G., VANDEBORRE J.-P., BENHABILES H., DAOUDI M., HUEBNER K., MORTARA M., SPAGNUOLO M.: SHREC'12: 3D Mesh Segmentation. In *EG2012* (2012). 2

[MPM*14] MATTAUSCH O., PANOZZO D., MURA C., SORKINE-HORNUNG O., PAJAROLA R.: Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum* (2014). 2

[NXS12] NAN L., XIE K., SHARF A.: A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics* (2012). 1, 2, 3, 5, 6

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH* (2006). 1

[SWK07] SCHNABEL R., WAHL R., KLEIN R.: Efficient ransac for point-cloud shape detection. *Computer Graphics Forum* (2007). 2

[SWWK08] SCHNABEL R., WESSEL R., WAHL R., KLEIN R.: Shape recognition in 3d point-clouds. Skala V., (Ed.), UNION Agency-Science Press. 2

[SXZ*12] SHAO T., XU W., ZHOU K., WANG J., LI D., GUO B.: An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Trans. Graph.* (2012). 2

[VG06] VERGAUWEN M., GOOL L. V.: Web-based 3d reconstruction service. *Mach. Vision Appl. 17*, 6 (2006), 411–426. 1