Interactive and View-Dependent See-Through Lenses for Massive 3D Point Clouds

Sören Discher, Rico Richter and Jürgen Döllner

Abstract 3D point clouds are a digital representation of our world and used in a variety of applications. They are captured with LiDAR or derived by image-matching approaches to get surface information of objects, e.g., indoor scenes, buildings, infrastructures, cities, and landscapes. We present novel interaction and visualization techniques for heterogeneous, time variant, and semantically rich 3D point clouds. Interactive and view-dependent see-through lenses are introduced as exploration tools to enhance recognition of objects, semantics, and temporal changes within 3D point cloud depictions. We also develop filtering and highlighting techniques that are used to dissolve occlusion to give context-specific insights. All techniques can be combined with an out-of-core real-time rendering system for massive 3D point clouds. We have evaluated the presented approach with 3D point clouds from different application domains. The results show the usability and how different visualization and exploration tasks can be improved for a variety of domain-specific applications.

Keywords 3D point clouds · LIDAR · Visualization · Point-based rendering

1 Introduction

Applications for environmental monitoring (Koenig et al. 2015), urban planning and development (Tomljenovic et al. 2015), as well as disaster and risk management (Canli et al. 2015) require precise and up-to-date surface information for objects, sites, and landscapes. 3D point clouds are digital discrete surface repre-

S. Discher (✉) · R. Richter · J. Döllner
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
e-mail: soeren.discher@hpi.de

R. Richter
e-mail: rico.richter@hpi.de

J. Döllner
e-mail: juergen.doellner@hpi.de

A. Abdul-Rahman (ed.), *Advances in 3D Geoinformation*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-25691-7_3

sentations that fulfill these requirements. They can be created automatically and efficiently by means of in situ and remote sensing technology (e.g., laser scanning or photogrammetric approaches) with high density (e.g., 400 points per m^2) and high capturing frequency (e.g., once a month). Due to the massive amount of data managing, processing and visualization of 3D point clouds poses challenges for hard- and software systems (Richter und Döllner 2014; Leberl et al. 2010). Traditionally, geoinformation applications and systems reduce the density and precision, or extract generalized, mesh-based 3D models in a time-consuming process (Lafarge and Mallet 2012). Hence, they do not use the full potential of the data. Out-of-core and external memory algorithms on the other hand are designed to cope with massive amounts of data (Nebiker et al. 2010). As an example, arbitrarily large 3D point clouds can be explored in real-time by combining specialized spatial data structures with Level-of-Detail concepts (Richter et al. 2015; Goswami et al. 2013). Most of these approaches render all points in a uniform way that gives an equal amount of significance to every point. However, points within a 3D point cloud typically differ in terms of relevance. The relevance of a point depends on the kind of information it carries (e.g., timestamp, surface category, color, position) as well as the current use case and visualization task. In this paper, we focus on the following use cases:

- **Different Acquisition Types**. Surface information for a site could be available from different surveys such as an airborne, mobile mapping, terrestrial, outdoor and indoor data acquisition. On a broader scale, 3D point clouds of spatial environments might consist of overground as well as subterranean structures (e.g., mine shafts, sewers). By allowing users to see through occluding structures (i.e., by masking out corresponding points) we can facilitate the in-depth exploration of such 3D point clouds (Fig. 1).

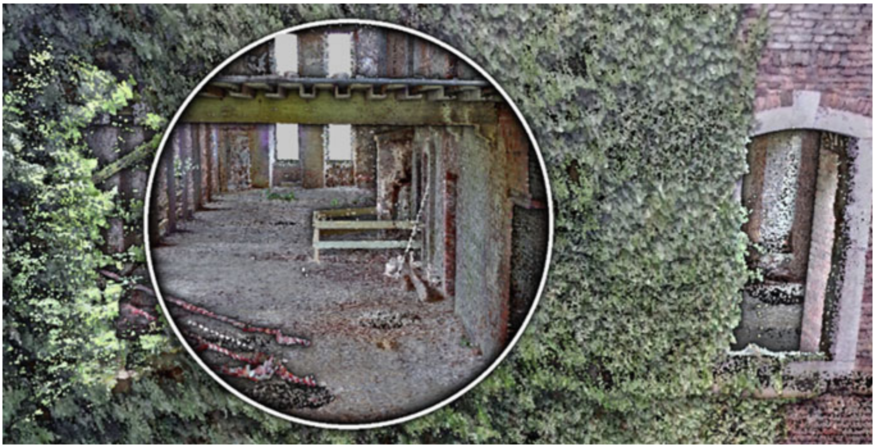


Fig. 1 Example of a massive 3D point cloud consisting of indoor and outdoor scans. It is explored with a see-through lens to inspect the occluded interior of the building in the context of the overall scan

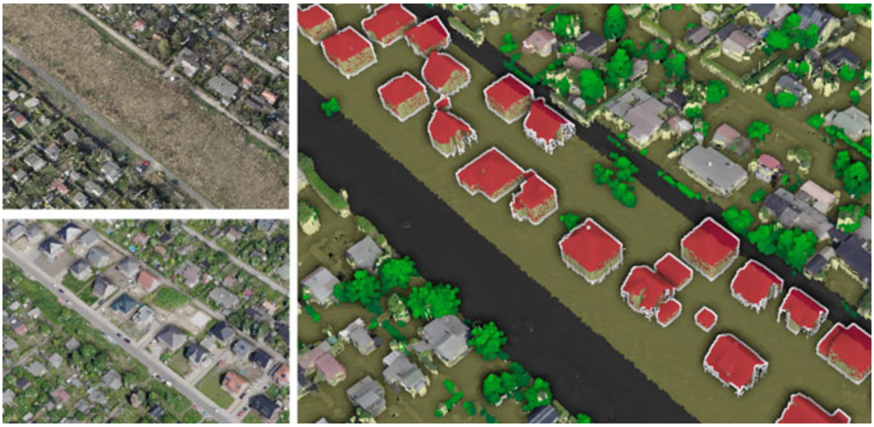


Fig. 2 Multi-temporal 3D point cloud: building points that are not present in the old scan (*upper left*) but captured in the new scan (*lower right*) are highlighted with a red color scheme and by tracing their boundaries

- **Multi-temporal Data**. Many applications require frequent scans and simultaneous use of 3D point clouds taken at different points in time. To assist users in exploring differences and structural changes of the captured site within such multi-temporal 3D point clouds (e.g., constructed, demolished, or modified buildings), points indicating such changes should be highlighted (Fig. 2).
- **Classification-Dependent Rendering**. Typically, points represent different surface categories (e.g., ground, building, vegetation, water, city furniture interior, façade). By applying different point-based rendering techniques and color schemes to each point, taking into account characteristics of its surface category (e.g., fuzzyness of vegetation, smooth ground surfaces or planar building roofs), different objects within a 3D point cloud depiction and relations between these objects can be distinguished more efficiently (Richter et al. 2015; Gao et al. 2014). However, relevant objects might still be not visible due to occlusion by less relevant objects (e.g., buildings below vegetation). To facilitate the identification of full and partly occluded objects, users need to see through occluding structures (Fig. 3).

To filter points within a 3D point cloud based on their relevance to the given use case, we define *see-through lenses* for massive 3D point clouds:

- A see-through lens defines a space within a 3D point cloud, in which points of higher relevance are emphasized by masking out less relevant points completely or in parts.
- The area of a see-through lens can be defined *interactively* by the user or automatically with respect to the current view position (e.g., center of the screen), i.e., *view-dependent*.

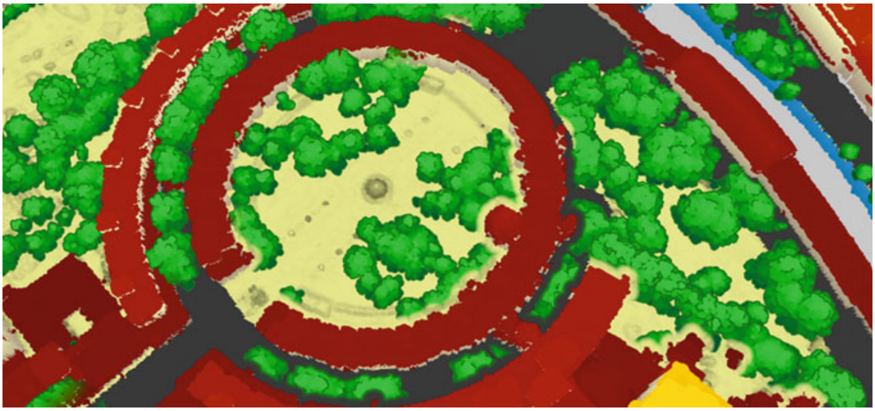


Fig. 3 3D point cloud with semantics: vegetation points are masked out in the *lower right* part to show hidden building points

In this paper we present several interactive and view-dependent see-through lenses for massive 3D point clouds and discuss their specific advantages and disadvantages in different scenarios (Sect. 3). Building upon an existing out-of-core rendering approach for the classification-dependent visualization of massive 3D point clouds, we show how see-through lenses can be efficiently combined with existing point-based rendering approaches as a post processing step (Sect. 4). As we demonstrate on a set of real-world datasets, our approach allows to select and configure see-through lenses at runtime while providing interactive frame rates for arbitrarily large 3D point clouds (Sect. 5). We conclude by giving an outlook on future research directions (Sect. 6).

2 Related Work

An overview of point-based rendering techniques is given by Gross and Pfister (2007). While several photo—(Yu and Turk 2013; Preiner et al. 2012) and non-photorealistic (Goesele et al. 2010; Xu et al. 2004) rendering techniques for 3D point clouds exist, most approaches apply a uniform style regardless of each points significance for the given use case. Some approaches prioritize and highlight points based on precomputed surface categories by combining different rendering styles to emphasize surface category characteristics. Gao et al. (2014) focus on combining point and polygon based rendering techniques to achieve a solid, hole-free visualization of certain surface categories (i.e., buildings, ground). However, this approach does not address the occlusion of structures and surfaces. Richter et al. (2015) show how out-of-core rendering algorithms for 3D point clouds can benefit from per-point surface category information and provide an even broader set of point-based rendering styles. The proposed rendering system allows to integrate see-through lenses using image compositing.

Interactive occlusion management techniques for virtual 3D environments have been widely discussed for years. However, the focus is on mesh-based 3D models instead of point-based rendering. Elmqvist and Tsigas (2008) differentiate three primary purposes for these techniques: (1) the discovery of yet unknown, occluded objects, (2) the exploration of occluded objects whose position has been known beforehand, and (3) the exploration of an occluded object's spatial relationship to its environment. Since all these purposes are important when exploring 3D point clouds, we introduce techniques addressing all of them. Elmqvist and Tsigas (2008) identify five different categories of occlusion management techniques: Tour Planner based approaches (Andújar et al. 2004; Burtnyk et al. 2002) calculate special camera paths through an environment, ensuring that every significant object is visible at least once while following these paths. Some approaches use multiple views of an environment in parallel. The different views are either rendered into separate viewports (Hasan et al. 2014; Brosz et al. 2011) or combined into a single one by applying multi-perspective projections (Pasewaldt et al. 2014; Brosz et al. 2007). Volumetric probes stay with a single view for an environment. Instead they deform objects (e.g., by manipulating their proportions) within a certain area that is often defined by users (Wang et al. 2005; Sonnet et al. 2004).

The see-through lenses for massive 3D point clouds proposed in this paper belong to the category of *virtual x-rays*. These techniques leave object proportions intact and usually operate in screen space, either masking out occluders completely or making them (semi-)transparent. They can be either active, i.e., requiring user input to define the areas they should be applied to (Trapp et al. 2008; Ropinski et al. 2005), or passive, i.e., finding those areas automatically based on available information about the objects within an environment (Vaaraniemi et al. 2013; Sigg et al. 2012). We decided for visual x-rays, as this allows users to freely navigate the 3D point cloud (which wouldn't be the case for tour planner based approaches) without having to manipulate or deform the data (which would be the case for volumetric probes). We also decided against combining multiple views as this would interfere with existing out-of-core rendering techniques for massive 3D point clouds.

3 Overview

We present several see-through lenses to filter and highlight structures within 3D point cloud depictions based on their significance to the given use case. All lenses operate in screen space and are realized as post processing effects, thus facilitating their integration into existing point-based rendering approaches.

3.1 Data Characteristics

Additional per-point attributes can be calculated in a preprocessing step. Common attributes are:

- **Color**. Color and color-infrared information allows for a more realistic visualization of 3D point clouds. It can be extracted from image data that ideally have been captured simultaneously with the 3D point cloud.
- **Surface normal**. Surface normals describe the orientation and structure of a surface at each point's position (Mitra and Nguyen 2003). They are typically used to align and orient the rendering geometry (e.g., point splats).
- **Topological information**. This information can be derived by analyzing the local point proximity. Topologic attributes are for example the height of a point within its local neighborhood, surface smoothness, and local point density. Adapting the colorization accordingly puts emphasis on these attributes, which can be used to accentuate detailed and geometrically complex object structures (e.g., roof elements).
- **Surface category**. A basic categorization of points into indoor, outdoor, overground, subterranean, airborne, or terrestrial can be made directly during the capturing process. A more specific categorization into surface categories such as building, ground, and vegetation can be derived with point cloud classification approaches that are based on analyzing topological attributes or additional geodata capturing the same surface area. (Richter et al. 2013).
- **Temporal information**. Temporal attributes describe the date of data collection.
- **Change information.** Change information describes on a per-point basis the degree of change related to a given reference geometry. That reference geometry might be another 3D point cloud captured at a different point in time or a 3D model describing the same area (e.g., building model, cadastre data, 3D city model).

These per-point attributes can be used to customize the appearance of points individually at runtime (Richter et al. 2015). Hence, points that are *related* to each other (e.g., representing the same surface category or being created at the same point in time) can be identified more easily. If points are occluding each other, these attributes can also serve as a means to assess and highlight the significance of each point for a given use case.

3.2 Priority Levels

Depending on the application, use-case, and kind of information that needs to be explored, the significance of a point may vary. When exploring a densely forested area for example, points representing vegetation are often less interesting than

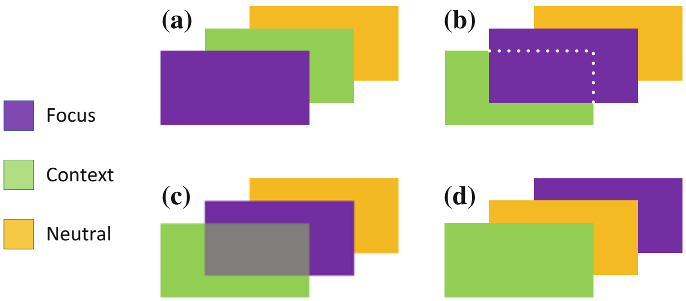


Fig. 4 Illustration of compositions for different priority levels in occlusion situations. **a** Context information is, **b** masked out in favor of or, **c** blended with focus information. **d** Neutral information is neither highlighted nor filtered

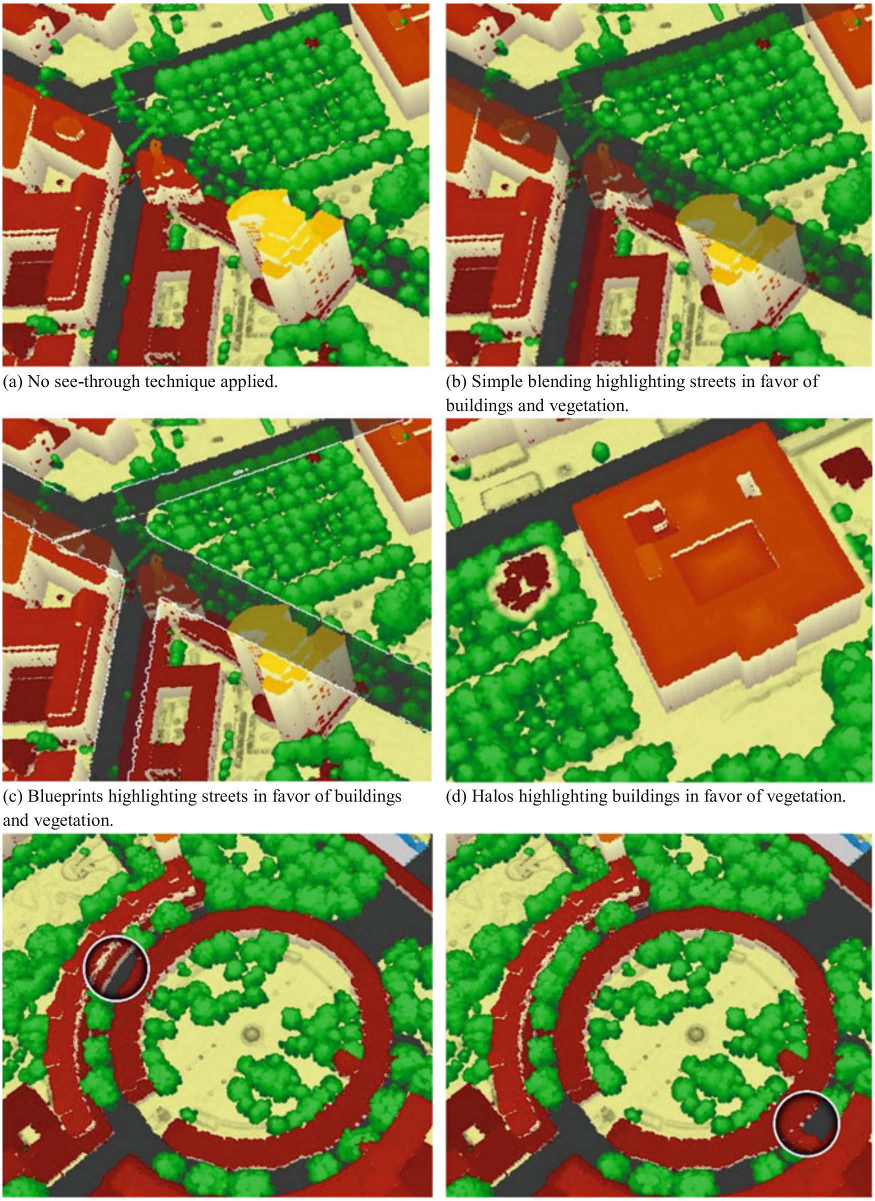
occluded points belonging to subjacent objects (e.g., buildings). We describe the significance of each point with the following priority levels:

- **Focus**. Essential information of interest and exploration aim (e.g., interior objects occluded by walls, subterranean structures or buildings that have been demolished between consecutive scans). The more points carrying such information are occluded, the less information can be gathered, i.e., the less effective the exploration becomes (Fig. 4).
- **Context**. Information that increases the overall realism of the visualization without being the main focus of the exploration (e.g., vegetation in densely forested areas). Points carrying such information can be safely masked out in favor of focus information (Fig. 4).
- **Neutral**. Depending on the use case, we might want to focus on specific occlusion scenarios, such as solely on buildings that are occluded by vegetation. This requires to define all other information as neutral, i.e., points representing such information are treated as solid geometry that is neither masked out or highlighted (Fig. 4).

These priority levels are used to define the composition of the data for the rendering and visualization of different occlusion scenarios (Fig. 4): Occlusions with the same priority level or including points carrying neutral information are solved by displaying the nearest point to the view position (i.e., similar to the default behavior in 3D rendering). Points carrying context information on the other hand (so-called context points) are masked if focus points are occluded.

3.3 Interactive and View-Dependent See-Through Lenses

Simply masking out all context points in occlusion scenarios limits the correct estimation of depth differences and does not provide information about the object shape



(a) No see-through technique applied.

(b) Simple blending highlighting streets in favor of buildings and vegetation.

(c) Blueprints highlighting streets in favor of buildings and vegetation.

(d) Halos highlighting buildings in favor of vegetation.

(e) Interactive see-through lenses highlighting buildings in favor of vegetation.

(f) Interactive see-through lenses highlighting buildings in favor of vegetation.

Fig. 5 Examples of different see-through lenses applied to massive 3D point clouds

and boundaries within a 3D point cloud depiction (Fig. 5a). Blending context points and focus points by a certain factor addresses these limitations, however, areas with blended structures might be difficult to recognize during the exploration (Fig. 5b).

3.3.1 Blueprints

Blueprints are a traditional form of technical drawings and known for their characteristic style which originating from the historical contact print process (Nienhaus and Döllner 2004). Construction elements are visualized by tracing outlines using different line widths and a color contrasting favorably with the background. Thus, the focus of the viewer is directed towards the most significant construction elements. We adapt this concept to highlight areas where focus and context points have been merged by tracing the boundaries of those areas with a configurable line width and color (Fig. 5c). This approach is especially effective to highlight changes within multi-temporal 3D point clouds (Fig. 2).

3.3.2 Halos

A stronger emphasis on focus points can be obtained by masking out additional context points in a defined local proximity (Fig. 5d). As a result, groups of neighboring focus points are surrounded with a halo effect, similarly to the real-world phenomenon and the technique used by artists throughout history to emphasize certain individuals. Techniques that highlight objects by removing occluders are known as *cut-away-views* (Vaaraniemi et al. 2013; Sigg et al. 2012). So far, they have not been applied to point-based rendering. The typical use case to apply this technique is the exploration of complex structures that are completely occluded by their surroundings, such as subterranean structures (Fig. 1).

3.3.3 Interactive See-Through Lenses

All techniques that have been introduced are applied automatically to the data where focus information is occluded by context information. Naturally, the number, position, and extent of these areas varies depending on the view position. As opposed to that automated, view dependent approach, interactive see-through lenses (also commonly known as ‘magic lenses’) can be moved freely across the screen. Within an interactive see-through lens, context points are masked out completely, whereas in the surrounding no blending is applied at all (Fig. 5f, e). This is required to focus on occlusions within certain areas whose position is known beforehand (e.g., to explore and show a former state for a certain area). However, if those areas are unknown, interactive see-through lenses are inefficient as the entire 3D point cloud has to be traversed manually to identify all areas.

4 Compatibility to Existing Point-Based Rendering Approaches

The see-through lenses proposed in this paper can be integrated into existing rendering systems for 3D point clouds. We demonstrate this by extending the rendering approach for massive 3D point clouds with surface category information introduced by Richter et al. (2015) (Fig. 6).

The approach is based on a layered, multi-resolution kd-tree to efficiently select relevant points of different surface categories for a given view position and rendering budget. To render selected points multi-pass rendering in combination with G-Buffers (Saito and Takahashi 1990), i.e., specialized frame buffer objects (FBO), is used. Multiple 3D textures store information for color, depth, or normal values. For each surface category a separate rendering pass is applied, allowing to combine different rendering techniques. Originally, Richter et al. (2015) utilize a separate G-Buffer for each surface category and apply a compositing pass to combine these G-Buffers into a final image. To apply see-through lenses, the compositing logic is extended. See-through lenses are implemented via programmable fragment shaders and can be activated and configured at runtime.

To guarantee interactive frame rates even for massive 3D point clouds with a variety of thematic attributes and surface categories, the original approach was modified to have only one G-Buffer for each priority level (instead of one for each surface category). Thus, the compositing pass always combines three G-Buffers, independently from the overall number of surface categories within a 3D point

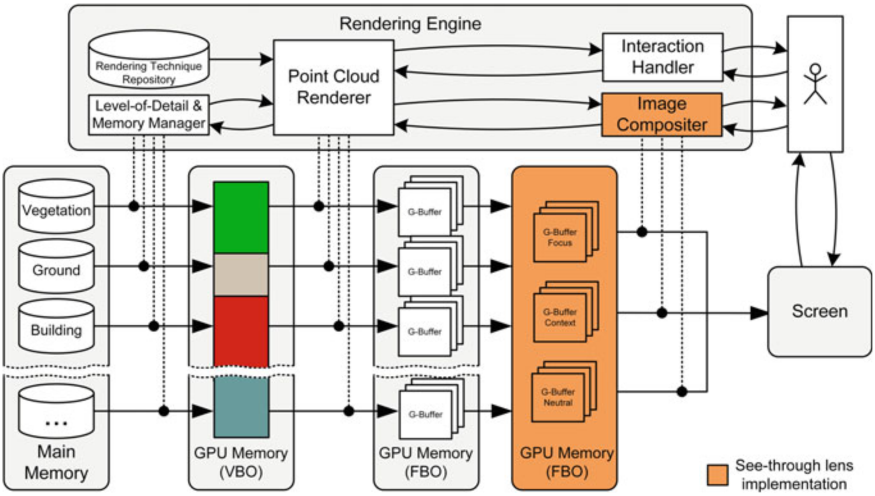


Fig. 6 Schematic overview of the classification-dependent point-based rendering system by Richter et al. (2015) and our modifications. Categorized by surface categories, points are transferred to GPU memory and rendered into separate G-Buffers that are merged based on the respective priority levels before being composed to synthesize the final image

cloud depiction. To enable halos with a defined proximity (e.g., 2 meters), an additional halo mask has to be prepared for surface categories belonging to the focus that determines the area in which context information needs to be masked out. However, creating that mask for each frame has a notable impact on the overall performance (Sect. 5).

5 Results and Applications

We have implemented the proposed rendering approach using C++, OpenGL, GLSL, and OpenSceneGraph. Performance evaluation was conducted based on three real-world data sets of different size and point density (Table 1). The test system consisted of an Intel Xeon CPU with 3.20 GHz, 12 GB main memory, and a NVIDIA GeForce GTX 770 with 2 GB device memory.

Each data set was rendered from a zoomed out and a close up perspective which affected the number of points rendered for a given frame. As shown in Table 2, applying simple blending, blueprints, and interactive see-through lenses has minimal effect on the overall performance. Since all techniques are implemented as screen-based post-processing effects, this stays true independent from the overall number of rendered points. Halos require a halo mask that is created by rendering corresponding surface categories twice, which effectively halves the average frame

Table 1 Characteristics of the datasets used to evaluate the performance of the presented rendering approach for view-dependent and interactive see-through lenses

	Dataset 1	Dataset 2	Dataset 3
Point density (pts/m^2)	10	28	100
Number points (Billion)	5	7.1	80
Data size (GB)	112	159	1788

Table 2 Rendering performance in frames per second (fps) for different see-through lens techniques and varying numbers of rendered points. Each dataset is evaluated for a close and a far perspective. Points are rendered as so-called Point Splats (see Richter et al. 2015)

	Dataset 1		Dataset 2		Dataset 3	
	Far	Close	Far	Close	Far	Close
#Rendered points in million	2.32	0.50	3.42	0.85	4.85	1.04
No technique applied	51.84	214.32	32.27	138.67	23.01	108.63
Simple blending	49.65	210.84	30.39	135.84	21.73	105.96
Blueprints	48.18	208.31	29.84	134.76	21.09	104.85
Interactive lens	49.32	209.68	30.12	135.20	21.46	105.21
Halos	27.68	104.31	23.97	67.31	16.92	52.57

rate if halos are activated. However, interactive frame rates are still achieved for several million points. Since the applied out-of-core rendering approach (Richter et al. 2015) limits the number of rendered points per frame, the proposed see-through lenses can be applied for arbitrarily large 3D point clouds.

6 Conclusions and Future Work

We discussed how the exploration of heterogeneous, time-variant, and semantically rich 3D point clouds can be facilitated by taking into account the relevance of a point for different use cases. The relevance of a point can be assessed based on a point's spatial position or any additional per-point attribute such as surface category, timestamp, or the topology of the local point proximity. We introduce the concept of see-through lenses to interactive point-based rendering and show how such techniques can be integrated into existing rendering systems for massive 3D point clouds as additional post processing effects. Thus, we can improve the visual recognition of relevant, occluded objects within a 3D point cloud depiction by masking out less relevant occluders. The proposed see-through lenses can be configured and selected at runtime offering many degrees of freedom for graphics and interaction design. Therefore, our approach is highly adaptive and applicable to a variety of use cases and visualization tasks from different domains. For future work, we plan to improve the overall performance of the halo-based technique (e.g., by defining halos in screen space, thus, avoiding the costly preparation of halo masks). In addition, we also want to investigate how to efficiently apply alternative techniques to highlight occluded objects (e.g., multiple views or multi-perspective projections) to point-based rendering.

Acknowledgments This work was funded by the Federal Ministry of Education and Research (BMBF), Germany within the InnoProfile Transfer research group "4DnD-Vis" (www.4dndvis.de) and the Research School on 'Service-Oriented Systems Engineering' of the Hasso Plattner Institute. We would like to thank virtualcitySYSTEMS and the Faculty of Architecture at the Cologne University of Applied Sciences for providing datasets.

References

Andújar, C., Vázquez, P., & Fairén, M. (2004). Way-Finder: Guided tours through complex walkthrough models. *Computer Graphics Forum, 23*(3), 499–508.

Brosz, J., Carpendale, S., & Nacenta, M. (2011). The undistort lens. *Computer Graphics Forum, 30*(3), 881–890.

Brosz, J., Samavati, F., Carpendale, S., & Sousa, M. (2007). Single camera flexible projection. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering* (pp. 33–42).

Burtnyk, N., Khan, A., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2002). StyleCam: Interactive stylized 3D navigation using integrated spatial and temporal controls. In

Proceedings of the 15th annual ACM symposium on User interface software and technology (pp. 101–110).

Canli, E., Thiebes, B., Höfle, B., & Glade, T. (2015). Permanent 3D laser scanning system for alpine hillslope instabilities. In *6th international conference on debris-flow hazards mitigation*: *Mechanics, prediction and assessment* (pp. 1–1).

Elmqvist, N., & Tsigas, P. (2008). A Taxonomy of 3D occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics, 14*(5), 1095–1109.

Gao, Z., Nocera, L., Wang, M., & Neumann, U. (2014). Visualizing aerial LiDAR cities with hierarchical hybrid point-polygon structures. In *Proceedings of the 2014 graphics interface conference* (pp. 137–144).

Goesele, M. et al. (2010). Ambient point clouds for view interpolation. *ACM Transactions on Graphics, 29*(4), 95:1–95:6.

Goswami, P., Erol, F., Mukhi, R., Pajarola, R., & Gobbetti, E. (2013). An efficient multiresolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. *The Visual Computer, 29*(1), 69–83.

Gross, M., & Pfister, H. (2007). *Point-based graphics*. Morgan Kaufmann Publishers Inc.

Hasan, M., Samavati, F., & Jacob, C. (2014). Multilevel focus + context visualization using balanced multiresolution. In *2014 international conference on cyberworlds* (pp. 145–152).

Koenig, K. et al. (2015). Comparative classification analysis of post-harvest growth detection from terrestrial LiDAR point clouds in precision agriculture. *ISPRS Journal of Photogrammetry and Remote Sensing, 104*(0), 112–125.

Lafarge, F., & Mallet, C. (2012). Creating large-scale city models from 3d-point clouds: A robust approach with hybrid representation. *International Journal of Computer Vision, 99*(1), 69–85.

Leberl, F. et al. (2010). Point clouds: Lidar versus 3D vision. *Photogrammetric Engineering and Remote Sensing*, *76*(10), 1123–1134

Mitra, N., & Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *19th annual symposium on computational geometry* (pp. 322–328).

Nebiker, S., Bleisch, S., & Christen, M. (2010). Rich point clouds in virtual globes—A new paradigm in city modeling? *Computers, Environment and Urban Systems, 34*(6), 508–517.

Nienhaus, M., & Döllner, J. (2004). Blueprint Rendering and 'Sketchy Drawings'. *GPU Gems, II*, 235–252.

Pasewaldt, S., Semmo, A., Trapp, M., & Döllner, J. (2014). Multi-perspective 3D panoramas. *International Journal of Geographical Information Science, 28*(10), 2030–2051.

Preiner, R., Jeschke, S., & Wimmer, M. (2012). Auto splats: Dynamic point cloud visualization on the GPU. In *Proceedings of eurographics symposium on parallel graphics and visualization* (pp. 139–148).

Richter, R., Behrens, M., & Döllner, J. (2013). Object class segmentation of massive 3D point clouds of urban areas using point cloud topology. *International Journal of Remote Sensing, 34* (23), 8408–8424.

Richter, R., & Döllner, J. (2014). Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Computers, Environment and Urban Systems, 45*, 114–124.

Richter, R., Discher, S., & Döllner, J. (2015). Out-of-core visualization of classified 3D point clouds. In 3d geoinformation science: The selected papers of the 3D geoinfo 2014 (pp. 227–242).

Ropinski, T., Hinrichs, K., & Steinicke, F. (2005). A solution for the focus and context problem in geo-virtual environments. In *Proceedings of the 4th ISPRS workshop on dynamic and multi-dimensional GIS* (pp. 144–149).

Saito, T., & Takahashi, T. (1990). Comprehensible Rendering of 3-D Shapes. *SIGGRAPH Computer Graphics, 24*(4), 197–206.

Sigg, S., Fuchs, R., Carnecky, R., & Peikert, R. (2012). Intelligent cutaway illustrations. *IEEE Pacific Visualization Symposium, 2012*, 185–192.

Sonnet, H., Carpendale, S., & Strothotte, T. (2004). Integrating expanding annotations with a 3D explosion probe. In *Proceedings of the working conference on advanced visual interfaces* (pp. 63–70).

Tomljenovic, I., Höfle, B., Tiede, D., & Blaschke, T. (2015). Building extraction from airborne laser scanning data: An analysis of the state of the art. *Remote Sensing, 7*(4), 3826–3862.

Trapp, M., Glander, T., Buchholz, H., & Döllner, J. (2008). 3D generalization lenses for interactive focus + context visualization of virtual city models. In *12th international conference on information visualisation* (pp. 356–361).

Vaaraniemi, M., Freidank, M., & Westermann, R. (2013). Enhancing the visibility of labels in 3D navigation maps. In *Progress and new trends in 3D geoinformation sciences* (pp. 23–40).

Wang, L., Zhao, Y., Mueller, K., & Kaufman, A. (2005). The magic volume lens: An interactive focus + context technique for volume rendering. *IEEE Visualization, 2005*, 367–374.

Xu, H., Nguyen, M., Yuan, X., & Chen, B. (2004). Interactive silhouette rendering for point-based models. In *Eurographics symposium on point-based graphics* (pp. 13–18).

Yu, J., & Turk, G. (2013). Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics, 32*(1), 5:1–5:12.