

An Interactive Visualization System for Urban Search & Rescue Mission Planning

Alexander Bock
Scientific Visualization Group
Linköping University
alexander.bock@liu.se

Alexander Kleiner
iRobot
Pasadena, CA
alexander.kleiner@gmail.com

Jonas Lundberg
Graphic Design Group
Linköping University
jonas.lundberg@liu.se

Timo Ropinski
Scientific Visualization Group
Linköping University
timo.ropinski@liu.se

Abstract—We present a visualization system for incident commanders in urban search & rescue scenarios that supports the inspection and access path planning in post-disaster structures. Utilizing point cloud data acquired from unmanned robots, the system allows for assessment of automatically generated paths, whose computation is based on varying risk factors, in an interactive 3D environment increasing immersion. The incident commander interactively annotates and reevaluates the acquired point cloud based on live feedback. We describe design considerations, technical realization, and discuss the results of an expert evaluation that we conducted to assess our system.

I. INTRODUCTION

Structural damage caused by disasters is an ever-present danger. As victims' survivability is mostly determined by the time-to-rescue, it is of great importance to provide medical attention or extraction as fast as possible. Planning access paths is, however, difficult as structural damage makes available floor plans outdated. In these situations, it is paramount that the *Incident Commander* (IC) can analyze the available information and coordinate multiple rescue responders. There exist well-defined protocols to describe the phases in an *Urban Search & Rescue* (USAR) operation; in the assessment step, 2D maps of the collapsed buildings are hand-drawn based on the descriptions of rescue responders moving within the building searching for victims, possibly stumbling into hazardous areas that endanger the rescuer's life. In recent years, technological developments allowed unmanned vehicles to perform this initial exploration. The robots are equipped with sensors that can detect victims, gather information about hazardous environments, and perform scans of rooms to create a 3D point cloud of the building's interior. The IC can inspect the building via the map and plan access paths that lead to *Points of Interest* (POI), for example locations of victims, or other mission critical areas that need to be reached.

In this paper, we present a visualization system targeting search & rescue situations. The system creates an interactive 3D rendering that increases the commander's spatial awareness of the building (Figures 1 (a-c) and 3) and supports the planning and analysis of access paths (Figure 4). The information is then used to instruct rescue responders to reach POIs and the IC annotates the visualization with information provided by the on-site responders, thus shifting the decision making process from opportunistic to being strategical. Our system computes multiple access paths, each based on varying risk factors, and presents them to the IC, allowing him to analyze and compare all available paths to minimize the rescuer's danger and travel

time. We present design considerations with regard to decision making, technical realizations, and the results of an expert evaluation that show the usefulness and acceptance of this visualization system among professionals.

II. RELATED WORK

Emergency management. Much of the visualization-oriented work published in the field of emergency management is concerned with pre-disaster evacuation planning. Notable work was performed by Reddy *et al.* and is based on analyzing possible bottlenecks of escape routes [1]. While these algorithms could be utilized, they usually assume perfect walking conditions and a known layout of the building. Ribarsky *et al.* presented a system organizing first responders in intact structures [2]. Kim *et al.* developed a system enhancing the situational awareness of responders using a mobile visual analytics tool [3]. Another related area is visual analysis-supported ensemble steering. Ribić *et al.* investigated steering ensembles of flooding simulations using visual analysis [4] and showed its usefulness to interpret this kind of data.

Many existing planning systems in USAR scenarios are based on 2D representations [5], [6]. Given a 2D map, one approach to path planning is to use the shortest trajectory and follow it stepwise. Wirth *et al.* introduced an exploration strategy and path planner that utilizes occupancy grid maps when planning to reach several targets simultaneously [7]. Extensions towards exploration in 3D with regard to detection of voids were introduced by Dornhege and Kleiner [8]. In contrast to previous systems, our system's tight integration of 3D point cloud rendering, embedded access paths, and analysis tools enables a deeper immersion into the scene and thus improves the IC's decision-making process.

Point cloud visualization. Basic rendering capabilities for point clouds are offered by the widely used Point Cloud Library [9]. There has been work by Richter *et al.* using a level-of-detail structure to render massive point clouds at high frame rates [10]. Xu *et al.* showed that non-photorealistic rendering techniques can be applied to point cloud data [11]. The contour lines in their rendering inspired our rendering algorithm. More recently, Pintus *et al.* presented a rendering algorithm that enhances features of unstructured point clouds in real-time without preprocessing [12]. In our system, we use a voxelized point cloud representation that raises the level of abstraction and simultaneously provides an immersive experience for the IC.

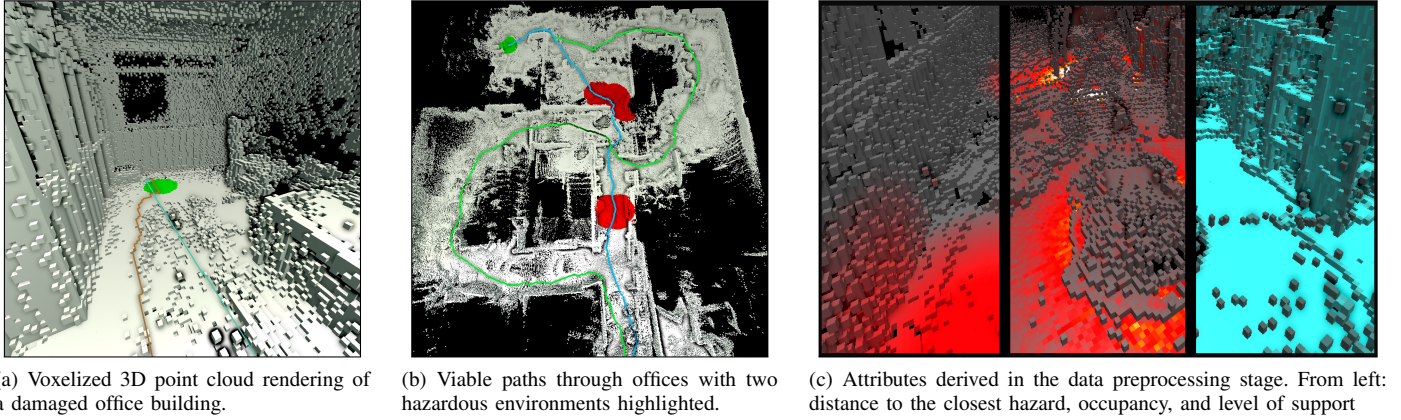


Fig. 1. Our system applied to a building at Tohoku university. Different views support a comprehensive understanding, allowing the IC to select and inspect paths that reach a point of interest, which are integrated into the rendering. Inspection and path computation is based on a set of derived attributes (c).

III. DECISION-MAKING THEORY

In order to design a system for USAR missions, it is crucial to utilize knowledge about human decision making. Decision makers in time-constrained situations tend to evaluate options serially; they attempt to find one viable plan rather than generating and comparing multiple plans in parallel. This theory has been described by Klein and Calderwood as *Recognition Primed Decision-making* (RPD) [13]. Initially, experts find similarities to previous cases, such as relevant goals, important things to monitor, and possible actions. Then, they simulate whether these parameters are also applicable to the current case.

The *Contextual Control Model* (COCOM) by Hollnagel and Woods describes how people rely on context when making decisions [14]. The quality of their control can be described as scrambled, opportunistic, tactical, or strategic. The scrambled mode refers to decisions made without any information about the situation. In the opportunistic mode, people rely on cues in the local context to decide on their next action. In tactical mode, they have an idea of how to achieve their goal before taking action—a plan. In strategic mode, the plan includes coordination with other simultaneous goals. The goal of our system is to raise the quality of control from being opportunistic to being strategic, thus enabling improved decision-making.

The *Extended Control Model* (ECOM) describes plans in terms of a tactical level (setting goals), monitoring (making plans and overseeing plans), regulating (managing local resources), and tracking (performing and adjusting actions) [14]. This theory is used to assess the kind of planning support a system provides. The importance of supporting resiliency has been argued by Lundberg *et al.*: “Rather than merely selecting a response from a ready-made table, [the system] must adapt and create a suitable response; either by following ready-made plans for adaptation or by making sense of the situation and create responses during the unfolding event” [15].

IV. INCIDENT COMMANDER WORKFLOW

In most USAR protocols, one IC is responsible for a single building and instructs multiple rescue responders inside this building. In this section, we will describe our visualization-enhanced workflow supporting the IC in this task.

After arriving on location, the responders’ first step is to explore and secure the outside of the collapsed building. No rescuer is allowed to enter the building before it is secured, which can take an hour or more to complete. During this step, unmanned vehicles record and measure the inside of the building, feeding back information to the IC, who inspects the map and determines entry points combining the map information with real-world input. The techniques employed in the robots’ exploration are outside the scope of this paper, so we refer the reader to related literature on this topic (e.g., Calisi *et al.* [16] or Amanatiadis *et al.* [17]). The robots’ sensors are able to detect most signs of victims using various sensors [18], but as these measurements are uncertain, both false positives and false negatives might occur. The same holds true for hazardous environments like fires, gas leaks, radiation, or structurally unsafe areas. The data retrieval and preprocessing is done in parallel with securing the perimeter so that all information is available when the next phase begins. Based on suggested or selected POIs, the system computes optimal paths that the IC uses to direct the rescuers through the building. This reduces time-to-rescue as the rescuers do not need to explore the building to the same extent, but can proceed directly to the POIs. Thus, the system applies RPD to the whole situation, extending the number of available cues and planning from local conditions to higher ECOM levels.

When planning access paths, a variety of factors must be taken into account. The responder has to, among others, maintain a safe distance from hazardous environments, avoid overhanging structures, and the ground must be level. Uncertainty in the data and IC’s invaluable knowledge make an algorithm for the problem infeasible. Furthermore, as these variables are extracted from uncertain data, they are difficult to quantify and subject to uncertainties. The IC has to perform trade-offs to choose between alternatives, for example favoring a faster, longer path over a more dangerous, shorter path.

While the IC instructs a rescuer to follow one path, new information about victims or hazards is fed back which the IC incorporates into the system. This feed-back loop is of high importance as features might not only have been missed by the robots, but detected features might change during the rescue operation. Fires can start or extinguish, structural collapses can make areas inaccessible, or debris is removed after the initial

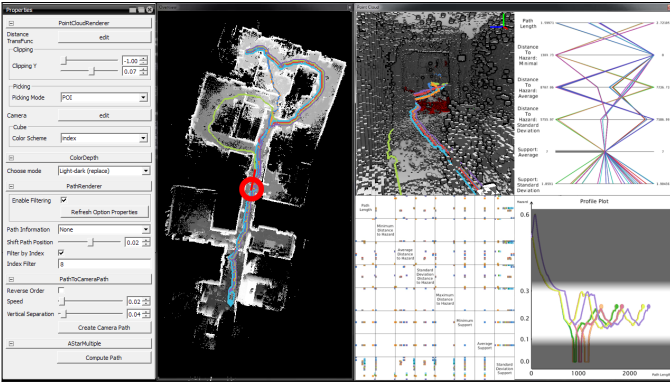


Fig. 2. A screenshot showing our system for a typical scenario. Each view can be maximized to fill the entire screen for in-depth inspection. An overview (left) shows a top-down view of the building to provide context, a multi-view (right) shows the different components of our system.

reconnaissance, opening previously inaccessible areas.

V. SYSTEM OVERVIEW

Our proposed system employs multiple linked views to provide the IC with all information (see Figure 2). The following sections explain these components. Before the acquired data can be used, a preprocessing must be performed that extracts derived data from the point cloud data (Section V-A). Then, interactive data annotation is possible (Section V-B). The generation of the optimal access paths, together with the employed metric, is explained in Section V-C. Section V-D provides details on the design considerations for the 3D visualization component of our system and the analysis of the path ensemble is described in the Section V-E.

A. Data Preprocessing

The data retrieved from the unmanned robots is an unstructured 3D point cloud. For our system, we assume that a suitable registration of the data has already been performed. One issue with directly rendering such point clouds is missing occlusion information and the non-uniform distribution of points. To avoid this problem, we perform a binning of the point cloud to obtain a three-dimensional voxel structure on a regular grid. From this point, we call a measurement in the original point cloud a *point* and refer to a position in the grid-based, binned point cloud as a *voxel*. After the binning, the resulting regular structure contains one voxel for all bins with at least one point. The bin size is dependent on the scan resolution of the robot, as it is a trade-off between resolving smaller details and decreasing SNR. In our cases, sizes of about 5 cm were sufficient. Figures 3(f–h) show three examples of possible voxel sizes obtained from the Jacobs University’s Rescue arena dataset.

In the second part of the preprocessing, derived attributes are computed that are later used to determine the set of access paths and to support their analysis. We compute the *hazard distance field* (Figure 1(c) left) that denotes the distance to the closest hazard points for each voxel, weighted by severity. The *occupancy field* (Figure 1(c) center) denotes the number of points each voxel is based on. A higher occupancy means that the voxel covers more points in the original point cloud and

thus provides a higher certainty. The *support field* (Figure 1(c) right) shows the available supporting area for each voxel. This value determines whether there is enough floor available for a responder to walk without hindrance. The *size field* shows for each voxel if a rescuer can fit into the space above without squeezing. We calculate two size values, one with the rescuer standing up, and a second while crouching. We also require orientation information to be able to exclude paths that would be too steep for a rescuer. For this, we compute the least-squares fitted plane based on all the points that are covered by a voxel. The normal of the plane is then used for the voxel.

B. Data Annotation

It is essential to be able to classify and annotate the data interactively. In our system, each voxel can belong to one of five classes. *Unclassified* is the default class and does not convey any semantics. *Start* voxels are usable entry points from which paths can start. *POI* voxels are destination points for a path and indicate a potential victim or another mission-critical element. *Hazard* voxels have been declared as dangerous due to, for example, an ongoing fire or gas leak. *Forbidden* voxels can only be declared by the IC and are areas completely out of reach for traversal. The IC modifies the classification for each voxel by interacting directly with the point cloud rendering.

C. Path Computation

We employ the widely used A* algorithm for our path computations [19]. It is a best-first search algorithm that works as follows: For the current point \mathbf{p} , the estimated remaining distance is calculated for each unvisited neighboring point \mathbf{q} . This value is the sum of the cost to reach \mathbf{p} from the start, the cost to move from \mathbf{p} to \mathbf{q} , and the estimated cost to reach the target from \mathbf{q} . Then, the point with the lowest cost is chosen as the next valid point to be tested. When computing a path with A*, a metric is used that determines the cost of moving from one point to its neighbor. Thus, it is possible to compute several optimal paths by changing this metric. Our metric is composed of weighted sub-metrics that are summed to yield

$$m = L_2(\mathbf{p}, \mathbf{q}) + w_h \cdot \text{hazard}(\mathbf{q}) + w_s \cdot \text{size}(\mathbf{q}) + w_n \cdot \text{normal}(\mathbf{q}, \varphi) + w_{sup} \cdot \text{support}(\mathbf{q}, n) \quad (1)$$

where w_h , w_s , w_n , and w_{sup} are the weights that are varied between different runs of the path computation. $\text{hazard}(\mathbf{q})$ returns the hazard severity that is stored in the hazard field. The hazard field stores for each voxel the L_2 distance to the closest hazard point and is generated iteratively with each added hazard. $\text{size}(\mathbf{q})$ is a binary function that determines if there is enough space above the voxel \mathbf{q} . The $\text{normal}(\mathbf{q}, \varphi)$ function computes the surface normal for \mathbf{q} and returns a normalized, linear response between the maximum allowed deviation φ and the gravity vector. $\text{support}(\mathbf{q}, n)$ is dependent on the number of supporting voxels in the area around the point \mathbf{q} and is retrieved from the support field. Here, n is a threshold determining how many voxels are needed to consider \mathbf{q} being supported.

The only restriction our system imposes on the path finding algorithm is the existence of a variable metric. We chose the A* algorithm as an example out of a family of possible algorithms (e.g., B*, D*, or Backtracking).

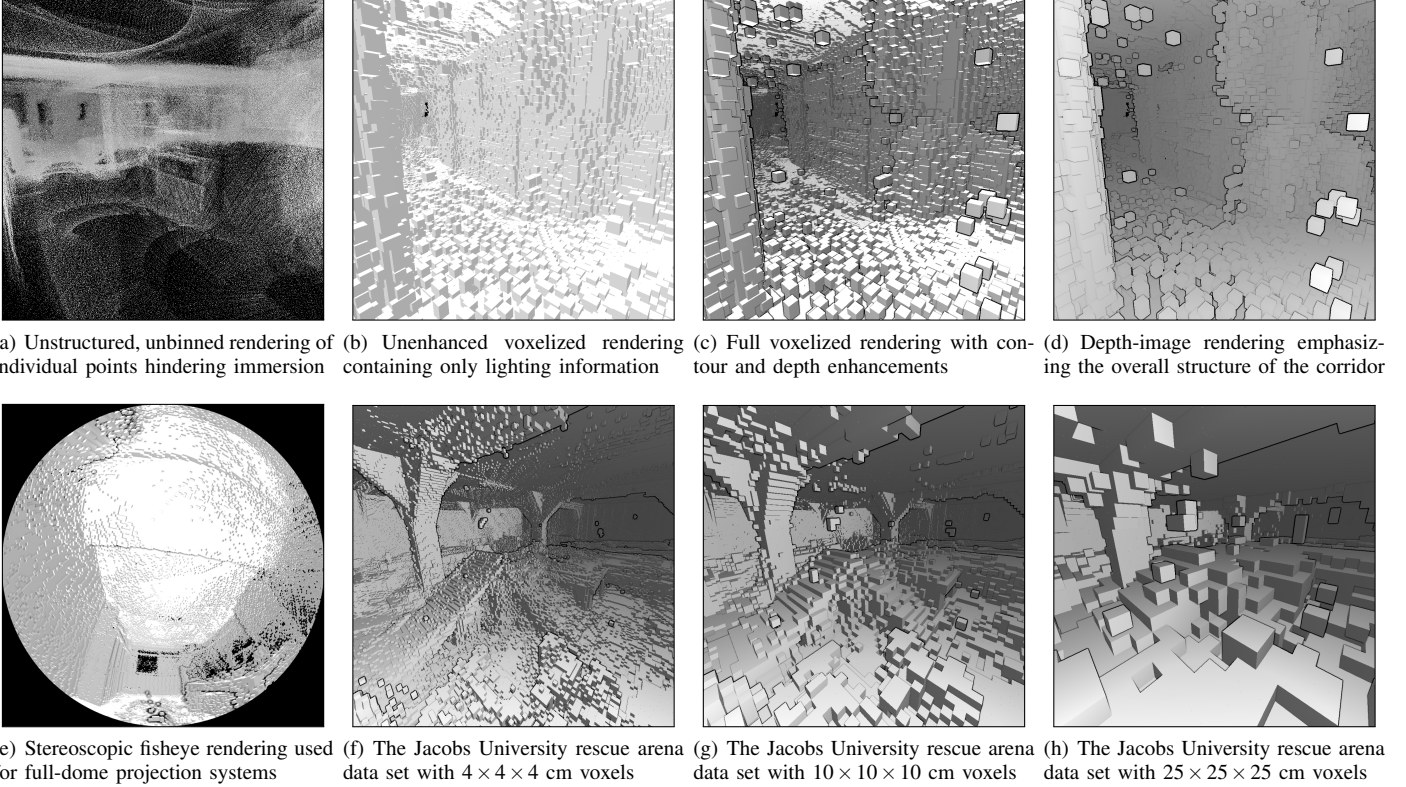


Fig. 3. These images show different rendering techniques of the same location in the Tohoku dataset (a–d). Rendering the individual voxels as points does not allow for an immersive rendering as depth-cues are missing (a). Binning the point cloud and representing each voxel as a axis-aligned box solves this problem (b). In order to enhance the contours of the scene and produce a better immersion, we perform image-space enhancements (c). Alternatively, the IC can choose a rendering method imitating the output of range imaging cameras (d). It is possible to render the point cloud in stereoscopic fisheye that can be used for dome surfaces or VR glasses (e). The voxel size’s effect during binning is shown for the rescue arena dataset (f–h).

D. 3D Visualization

Point cloud visualization. Missing occlusion, and thus missing depth cues, is a big challenge in point cloud rendering. Rendering each point individually inhibits immersion as structural features are hard to detect (Figure 3(a)). One attempt to add occlusion is to apply stencilling by rendering orthogonal disks around each point. Using this approach, points will have camera-aligned occlusion zones in which no points are rendered. This fails for several reasons; the size of the occlusion zone depends on the sparseness of points and if the zone is chosen too small otherwise occluded points will be visible. Second, the occlusion zone will also hide points belonging to the same structure. This leads to the structure, for example a wall, being represented only by a few points which does not reflect reality. Utilizing a voxelized representation was beneficial in which each voxel is represented by an axis-aligned cube with the same size as the grid structure. Figure 3(b) shows the result of this rendering method. This technique solves the occlusion problem immediately, as each voxel has an extent and occludes voxels. Only in cases with very sparse data do holes appear in the rendering. To increase the spatial awareness, we apply a lighting based on the face of the cube, rather than the normal of the voxel. We decided to utilize two image-space enhancement techniques to increase the immersion. The first method, presented by Luft *et al.* [20], is a contour-enhancement that increases local contrast in areas of high depth changes. This allows the IC to intuitively gain a

better understanding of the scene by emphasizing boundaries. The second method is a depth-based attenuation. The farther a voxel is from the camera the darker it will be rendered, thus providing an intuitive distance cue for the IC. As an optional method, we provide a simulated depth image resembling the output of range imaging cameras IC are already familiar with. This emphasizes the large scale structure of the building at the expense of individual details (see Figure 3(d)).

Access path visualization. In order for the IC to intuitively understand the spatial relationship of paths in the point cloud data, it is useful and necessary to show the paths embedded in the point cloud rendering. Since the paths are based on the center of the voxels, we lift each path point by at least half the voxel size in y direction. In our experiments we found that increasing it by about 2-3 times the voxel size produces a good visual result without looking too detached from the ground. A post-processing is performed on the path data in which a Catmull-Rom spline is generated using the voxels as control points. This results in a smoother path that lacks discontinuous edges that would otherwise draw unwanted attention. The IC can select various coloring methods to select the different stored information about the paths. In addition to using the same color as in the other components, the IC can select a coloring scheme to inspect each path attribute or submetric. The user can select a path in the rendering, highlighting that path in all other views and desaturating other paths. This makes it possible to inspect the exact behavior of a few paths without

distraction and also allows the user to reduce the number of valid paths very quickly.

E. Visual Path Analysis

It is essential to provide the IC with detailed information about the various paths to enable informed trade-offs. As the details depend on the specific situation, we provide adaptive tools to filter and analyze the paths according to the information requested by the IC. In the following, we will describe the intended role for each view, as well as the design considerations to ensure that each view fulfills this role.

Profile Plot. In order to enable a detailed comparative analysis of attribute changes, we include a *Profile Plot* (PP). This is a line plot showing changes of a single attribute along all paths, making it easy to compare the paths with regard to the chosen variable as minima and maxima are easy to detect. Figure 4(a) shows the *Hazard Distance* for a subset of paths. Several design considerations have been made to ensure the PP's effective use. First, the paths are drawn in the same color as in the 3D rendering to facilitate mental registration. Thus, each line in the PP can be easily identified with a path and directly linked to the rendering. As multiple paths are shown, each path's end point is emphasized with a dot enabling a direct comparison of path lengths, even with overlapping paths. Second, the scale of the y-axis of the PP has been chosen to allow for a better attribute value discrimination in regions of high importance. This is achieved by splitting the y-axis into three parts, a sub-linear, a linear, and a super-linear part, around important values resulting in a focus-in-context representation deemphasizing less important value ranges while providing a higher dynamic range to important values. In addition, the IC can toggle a transparent layer to further highlight the important value range.

Parallel Coordinates Plot. Figure 4(b) shows the *Parallel Coordinates Plot* (PCP) component where each path is represented by a line. PCPs are very well suited to enhance interactive exploration of multi-parametric data sources [21]. In our system, the PCP axes show global path attributes, like *Path Length*, *Minimal Hazard Distance*, *Average Hazard Distance*, or *Standard Deviation of Support*. The IC can select and filter paths and the interaction is linked to the other views. One important design decision is to avoid the confusion introduced through this visual linking. While using the same colors for the same paths supports mental registration, a line in the PCP does not support any spatial inference, contrary to the PP. Therefore, we have chosen to avoid the horizontal layout usually adapted in PCPs to break the optical similarity to the PP. As another design, we chose a fixed attribute ordering reflecting the importance regarding the path selection process. We have identified *Path Length* and *Minimal Hazard Distance* as the most crucial attributes for the path finding. To enable an intuitive understanding of the different path attributes, we have ordered each axis such that the preferable values are on the left, for example short *Path Length* or a large *Minimal Hazard Distance*. This results in a path layout where paths of high interest are located on the left.

Scatter Plot Matrix. The PCP is used for foreseeable comparisons. To deal with other combinations, we present the attributes in a Scatter Plot Matrix (SPLOM), where each attribute is given a row of scatter plots that show relations to all other attributes, enabling opportunistic comparisons.

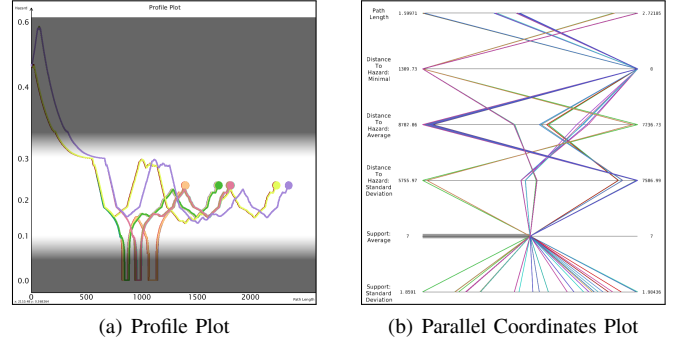


Fig. 4. Views supporting comparative path analysis. (a) The Profile Plot presents the change of an attribute along paths; here the distance to the closest hazard. (b) Parallel Coordinates Plot showing correlations between attributes.

VI. RESULTS

Since autonomous robots are not yet used in emergency situations, it is not possible to apply our system to a real-world disaster. Instead, to illustrate the flexibility of our system, we apply the application to one application case and two test cases.

A. Test Cases

Construction site. Figure 5(a) shows a point cloud acquired at a construction site with a LiDAR scanner inside an excavation pit. We resampled the original dataset from 50 million points to 3.5 million voxels with a size of 5 cm. The pre-computation steps took 3.5 minutes on a 3 GHz four-core CPU.

Rescue arena. Figure 5(b) shows paths through the Bremen rescue arena, which is used to challenge autonomous vehicles. The original point cloud consists of 28 million points, binned to 5.3 million points with 4 cm resolution. This computation took 2 minutes on the same machine as above.

B. Tohoku Application Case

The application case is a collapsed building at Tohoku university in the aftermath of the 2011 earthquake. The resulting dataset has been acquired with latest state-of-the-art equipment [22] and consists of 26 million sampling points, resampled to 4 million voxels with a size of 3 cm. It proved to be a valid and useful simulated real-world application case to test our proposed system due to partial collapses causing uneven ground and obstacles. Figure 1(a) shows a closeup rendering showing that the level of detail is sufficient to support spatial awareness. Figure 1(b) shows an overview of the scanned area with the POI in green at the top. No hazards could be detected by the robots as this was not part of their mission profile. Instead, artificial hazards were added manually. The task for this application case was as follows: the IC needed to find the shortest path between the entry point and the POI. While traversing the path, a rescuer would detect two hazards (highlighted in red) and the system must be able to react to this changing situation. In Figure 5(c), a subset of computed paths is shown after the discovery of the second hazard. There is one class of paths (purple) that evades the first hazard and another class (green) that evades the second hazard. The parallel coordinates plot (Figure 4(b)) makes it possible to detect paths belonging to both classes with a maximum in the *Minimal Hazard Distance*, while having a long *Path Length*.

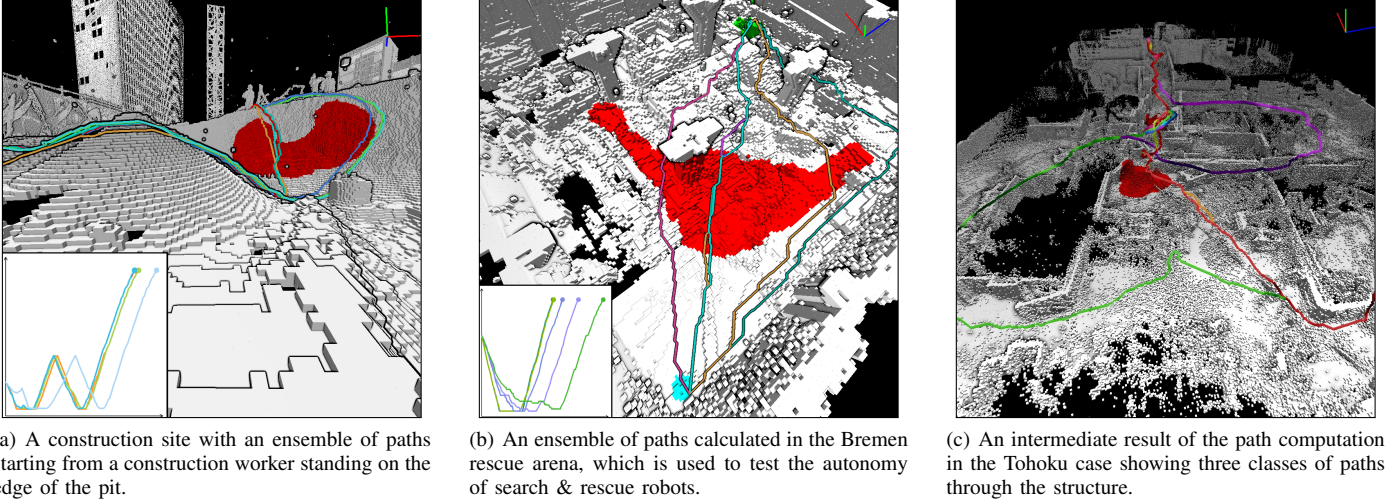


Fig. 5. Our system’s rendering component during use in the application case (a) and the two test cases (b) and (c). The filtered paths are shown into the 3D rendering to provide an increased spatial awareness. Different weights for hazards lead to distinct optimal paths to the POI.

VII. EVALUATION

We performed an evaluation of our system with nine external experts. Of these experts, five (A–E) are emergency responders, three (F–H) are researchers, and one (J) is a consultant for a national technical relief agency. As the goal of the study was to include many international experts, and being aware of their time constraints, we created an interactive webpage featuring videos and images of our system rather than requiring the training for a hands-on use. The experts had to inspect images and/or videos and then reply to questions and give feedback without a time limit. 7 of 9 experts finished the evaluation and provided answers to all questions. We have considered partial responses where an answer was provided.

The feedback is grouped into four categories; first, questions letting the experts assess their own knowledge about the component. With these questions, we can estimate whether the experts have prior knowledge and how intuitive the component is. The replies to these questions are on a 5-point Likert scale. The second category are factual checks that have a correct answer. These are used to test if the experts’ self-assessment is accurate. The third type of questions are open-ended and allow us insight into the experts’ thinking process. Usually, these questions do not have a best answer, but require a trade-off and domain knowledge. The fourth category asks for general comments about a component. The full evaluation and replies are available in the supplemental material. Here, we will discuss important positive and negative answers.

3D Representation This component evaluates the general usefulness of the 3D rendering. A video as well as images similar to Figures 1(a), 1(b), and 5(c) were provided. We asked to rate the degree of immersion (average: 2.94), the level of knowledge the experts had acquired (average: 2.5), and if the 3D rendering was useful in general (average: 4.14). To test the level of immersion and knowledge of the scene, we asked the experts to describe the room shown in the images, which all performed successfully, leading us to believe that the experts underestimated the knowledge they gained from the 3D rendering. An outlier in the responses was F, who reported

1 for the knowledge and 5 for the usefulness, but described the room correctly. We assume the answers are related to his comment, asking us to “improve [our] registration algorithm”.

Path Representation This component tests the usefulness of embedding the paths into the rendering. It consists of three images depicting the same situation as Figure 1(b). The first scenario with no hazardous areas, the second with the upper hazard only, and the third with both hazards. We asked the experts which of the paths they would choose. All but two (A and F) chose the paths that evaded the hazards. A second question was asked to compare the lengths of the shortest and the longest path. The correct answer was $1.54\times$, while the experts stated an average of $2.56\times$. This means that, while the experts are capable of estimating the length within a reasonable margin of error, they overestimate path distances, making it necessary to provide accurate information in the PP and PCP.

Evacuation Path Walkthrough In this component, we evaluated the combined effects of the first two components. We generated two camera paths from the longest path in component 2; one path moving from the *Start* to the *POI* and a second moving opposite. We asked for the usefulness (average: 3.125) and the self-assessed knowledge (average: 2.75) for both paths.

Profile Plot We provided a plot containing the orange, green, and yellow paths from Figure 4(a). We asked for the assessment of understanding (average: 3.5) and the usefulness (average: 3.71). Asking for the shortest path, and the number of times it crosses the hazard, all experts gave correct answers.

Parallel Coordinates Plot The self-assessed understanding is significantly lower compared to the PP (average: 1.66) which can be attributed to the experts’ unfamiliarity with PCPs. Correct answers, however, show that the PCP can still be a valid and useful component in our system. Despite their unfamiliarity, all five experts providing feedback found the shortest and the safest path, identifying the path with the lowest path length and the highest minimal distance. In the open questions, the experts seemed to favor safe, long paths over shorter, more dangerous paths while not paying much attention (except J) to the available support. This confirmed our design fixing the attribute ordering in the PCP. The experts rated the usefulness of the PCP with an average of 2.14.

Scatterplot Matrix The SPLOM received a very low level of self-assessed knowledge (average: 1.16) and usefulness (average: 1.28). Though this might be, just as the PCP, due to the fact that few, if any, have worked with SPLOMs before. A comment from E summarizes this component's result: "[this] is more information than I would want to interpret during a SAR mission". Given the experts' replies, the SPLOM component is an optional component of our system.

Miscellaneous For the last part, we asked if it is "helpful to display the paths and [if it] provides additional information", to which most experts (C, D, E, F, and J) agreed. D noted that the GUI should be more "user-friendly and understandable" and J said that "not detected hazards like structural integrity get visible by the density of scan points". To the question if they "[would] like to use [the] system in addition, or as a replacement, to [their] current tools", B, C, E, F, and J were positive to using our system in addition to their current tools rather than as a replacement. C would like to use it after a "period of experimentation", while J sees the system as "an addition which is warmly welcome". The only negative comment was from D, who regards the system as too complicated. J suggested a collaboration, saying that the system needs to be more intuitive to allow rescuers, who do not work with SAR operations regularly, to use its full potential.

We received valuable feedback from the experts and we can draw the conclusion that most experts liked the system and would like to use it as a decision-support tool alongside their current applications. With the exception of the SPLOM, the majority of experts were able to retrieve information from our system and use it for their decision-making.

VIII. CONCLUSIONS

We presented a linked, multiple-view visualization system that optimizes the workflow of an incident commander when dealing with search & rescue missions in cases where the initial reconnaissance of a collapsed structure is performed using unmanned vehicles. Based on this data, our system computes and analyzes an ensemble of rescue paths that are explored by the IC, who can then select a path that is an optimal trade-off according to his experience and knowledge. To investigate the system's usefulness, we have conducted a study with experts. The resulting positive feedback makes us confident that the system has the potential to improve future search & rescue planning missions. In addition, we presented a stereoscopic movie of the application case and the rescue arena data sets to an audience of approximately 100 researchers at an IEEE-sponsored conference on rescue robotics (see Figure 3(e)). We received positive informal feedback from these research experts on the rendering over the course of later discussions.

We like to thank the IRIDeS, the CREATE, and the IRS institutes for their cooperation in retrieving and providing the scans from Tohoku university. This work was partly supported by grants from the Excellence Center at Linköping and Lund in Information Technology (ELLIIT) and the Swedish e-Science Research Centre (SeRC), as well as VR grant 2011-4113. The presented system has been realized using the Point Cloud Library (PCL) embedded in the Voreen framework (www.voreen.org).

REFERENCES

- [1] R. Reddy, M. Höferlin, M. Dambier, and D. Weiskopf, "Visual Analytics for Dynamic Evacuation Planning," in *International Workshop on Visual Analytics*, 2012, pp. 13–17.
- [2] W. Ribarsky, K. R. Subramanian, J. Liu, O. Obirize, and J. Guest, "Mobile Application for First Response and Emergency Evacuation in Urban Settings," in *International Conference and Exhibition on Computing for Geospatial Research & Application*, 2010.
- [3] S. Kim, R. Maciejewski, K. Ostmo, E. J. Delp, T. F. Collins, and D. S. Ebert, "Mobile Analytics for Emergency Response and Training," *Information Visualization*, vol. 7, no. 1, pp. 77–88, 2008.
- [4] H. Ribičić, J. Waser, R. Fuchs, G. Bloschl, and E. Gröller, "Visual Analysis and Steering of Flooding Simulations," *IEEE Transactions on Visualization and Computer Graphics*, no. 6, pp. 1062–1075, 2013.
- [5] A. Kleiner and C. Dornhege, "Operator-Assistive Mapping in Harsh Environments," in *IEEE International Workshop on Safety, Security & Rescue Robotics*, 2009, pp. 1–6.
- [6] J. Pellenz and D. Paulus, "Stable Mapping Using a Hyper Particle Filter," in *Robot Soccer World Cup XIII*, 2009, pp. 252–263.
- [7] S. Wirth and J. Pellenz, "Exploration Transform: A Stable Exploring Algorithm for Robots in Rescue Environments," in *IEEE International Workshop on Safety, Security & Rescue Robotics*, 2007, pp. 1–5.
- [8] C. Dornhege and A. Kleiner, "A Frontier-Void-based Approach for Autonomous Exploration in 3D," in *IEEE International Symposium on Safety, Security & Rescue Robotics*, 2011, pp. 351–356.
- [9] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Conference on Robotics and Automation*, 2011, pp. 1–4.
- [10] R. Richter and J. Döllner, "Out-of-core Real-time Visualization of Massive 3D Point Clouds," in *Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, 2010, pp. 121–128.
- [11] H. Xu and B. Chen, "Stylized Rendering of 3D Scanned Real World Environments," in *Symposium on Non-photorealistic Animation and Rendering*, 2004, pp. 25–34.
- [12] R. Pintus, E. Gobbetti, and M. Agus, "Real-time Rendering of Massive Unstructured Raw Point Clouds using Screen-space Operators," in *International Conference on Virtual Reality, Archaeology and Cultural Heritage*, 2011, pp. 105–112.
- [13] G. A. Klein and R. Calderwood, "Decision Models: Some Lessons From the Field," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 5, pp. 1018–1026, 1991.
- [14] E. Hollnagel and D. D. Woods, *Joint cognitive systems: Foundations of cognitive systems engineering*. CRC Press, 2005.
- [15] J. Lundberg, E. Törnquist, and S. Nadjm-Tehrani, "Resilience in Sense-making and Control of Emergency Response," *International Journal of Emergency Management*, vol. 8, no. 2, pp. 99–122, 2012.
- [16] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, "Multi-objective exploration and search for autonomous rescue robots," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 763–777, 2007.
- [17] A. A. Amanatiadis, S. A. Chatzichristofis, K. Charalampous, L. Doitsidis, E. B. Kosmatopoulos, P. Tsalides, A. Gasteratos, and S. I. Roumeliotis, "A Multi-Objective Exploration Strategy for Mobile Robots Under Operational Constraints," *IEEE Access*, pp. 691–702, 2013.
- [18] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman, "Eulerian Video Magnification for Revealing Subtle Changes in the World," *ACM Transactions on Graphics*, vol. 31, no. 4, 2012.
- [19] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, pp. 100–107, 1968.
- [20] T. Luft, C. Colditz, and O. Deussen, "Image Enhancement by Unsharp Masking the Depth Buffer," *ACM Transactions on Graphics*, vol. 25, pp. 1206–1213, 2006.
- [21] M. Tory, S. Potts, and T. Möller, "A Parallel Coordinates Style Interface for Exploratory Volume Visualization," *IEEE Transactions on Visualization and Computer Graphics*, pp. 71–80, 2005.
- [22] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, "Emergency Response to the Nuclear Accident at the Fukushima Daiichi Nuclear Power Plants using Mobile Rescue Robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.