VISUALIZING AND MODELING INTERIOR SPACES OF DANGEROUS STRUCTURES USING LIDAR

N. MAERZ[1], T.K. ASSEBAUM[1], D. WILLIAMS[1], K. SHEA[1], Y. XI[2], X. LI[2] & Y. DUAN[2]
[1]Missouri University of Science and Technology, Rolla, MO, USA.
[2]University of Missouri, Columbia, MO, USA.

ABSTRACT

LIght Detection and Ranging (LIDAR) scanning can be used to safely and remotely provide intelligence on the interior of dangerous structures for use by first responders that need to enter these structures. By scanning into structures through windows and other openings or moving the LIDAR scanning into the structure, in both cases carried by a remote controlled robotic crawler, the presence of dangerous items or personnel can be confirmed or denied. Entry and egress pathways can be determined in advance, and potential hiding/ambush locations identified. This paper describes an integrated system of a robotic crawler and LIDAR scanner. Both the scanner and the robot are wirelessly remote controlled from a single laptop computer. This includes navigation of the crawler with real-time video, self-leveling of the LIDAR platform, and the ability to raise the scanner up to heights of 2.5 m. Multiple scans can be taken from different angles to fill in detail and provide more complete coverage. These scans can quickly be registered to each other using user defined ‘pick points’, creating a single point cloud from multiple scans. Software has been developed to deconstruct the point clouds, and identify specific objects in the interior of the structure from the point cloud. Software has been developed to interactively visualize and walk through the modeled structures. Floor plans are automatically generated and a data export facility has been developed. Tests have been conducted on multiple structures, simulating many of the contingencies that a first responder would face.

Keywords: Hazards, laser scanning, LIDAR, model, segmentation.

1 INTRODUCTION

First responders such as soldiers, police, and rescue workers often must enter structures that that are inherently dangerous. A dangerous structure is one that could potentially collapse because it is damaged or compromised by natural or man-made disaster. The structure could also be dangerous because it hosts armed enemy combatants, terrorist or other criminals. The structure could be dangerous because it can host any number of traps such as improvised explosive devices or chemical or biological agents.

First responders need knowledge (intelligence) about the status of the structure. Is it safe from potential collapse? Are there hostile armed individuals in the structure? Are traps and other pitfalls present. At the very least the first responders would wish to know the layout of the inside of the structure, including how to get in, location and path to emergency exits, and potential hiding places for hostiles.

Methods abound to collect terrain (geospatial) information. But there is a gap that has been satisfied by neither aerial collection, nor reconnaissance nor even by absorbing the ‘as built drawings’ in databases on structures. That gap is collecting information on what is on the inside of the building facade and seeing its condition. The gap has been covered to date by either sending in a man or a small camera equipped robot – or sometimes smashing out a hole to allow a good interior view. It is not an optimal situation, and a poor strategy in preparing first responders for entry.

This paper describes the logical next step where a terrestrial Light Detection and Ranging (LIDAR) scanner is used to map the interior of potentially dangerous spaces. The LIDAR

ISSN: 2041-9031 (paper format), ISSN: 2041-904X (online), http://journals.witpress.com
DOI: 10.2495/SAFE-V2-N4-330-350

scanner is moved into position (into the interior of the structures, or adjacent to external windows or other opening) using a wirelessly remote controlled robotic platform. Scans are made (controlled remotely and wirelessly) from various angles to create 3D maps. Scans can be conducted in darkness or bright sun, through glass windows and openings, or from the inside of the structure itself. These scans can be used to lay out floor maps, reveal potential hiding spaces for enemy combatants, reveal potential threats inside, map locations for egress, and give engineers the tools they need to evaluate the structural soundness of the building in question.

The major innovation of this research project is to combine state-of-the-art commercial available hardware (LIDAR scanner, camera, mobile platform) into a single remote operated package and develop the software to model and visualize the inside of buildings and other structures that are too dangerous to enter without first being able to determine the conditions and layout inside. The main technical advances are the integration of various commercial technologies and the development of advanced modeling and visualization techniques and software.

2 HARDWARE COMPONENTS

2.1 Leica scanner

A Leica HDS6000 LIDAR Scanner was purchased for this project. This scanner was selected after conducting a review of available existing technologies. The primary reason for selecting this device is that its phase shift distance measurement is an order of magnitude faster than comparable time of flight machines (an important consideration when emergency responders need to enter structure with unknown dangers). The HDS6000 was primarily selected for its scan rate of up to 500,000 points per second. It has a stated accuracy of 10 mm or less and a modeled accuracy of 2 mm (more than adequate for a mapping purposes), and a range of 79 m. As such it was by far the fasted and most accurate scanner available at the required scanning distances. The HDS 6000 does not have an integrated optical capability, but this was not considered important as optical imaging through glass windows is extremely difficult and close to impossible when shooting through glass into a darkened structure.

For a few of the early investigations a Leica ScanStation II (time of flight) LIDAR scanner was used. Although it has the same accuracy as the HDS6000, it was found to be about 1 order of magnitude slower than the HDS6000. The ScanStation II has a built in optical imaging capability, and a range of up to 300 m, neither of which capability was considered important to this project.

2.2 Robotic platform

2.2.1 Basic robotic crawler

A ‘kit’ robotic crawler was acquired from Super Droid Robots (Fig. 1). The kit, assembled locally consisted of control and power systems, including rechargeable batteries, relays, microcontroller and wireless router (Fig. 2). A 30″ wide platform was selected to increase the stability while still fitting through a standard door. Heavy duty treads were installed. A radio frequency (RF) maneuver control override was ordered for when moving the crawler into position or loading on a trailer. The crawler has rudimentary stair climbing abilities, but limited to steps with small risers and low overall angle of ascent.

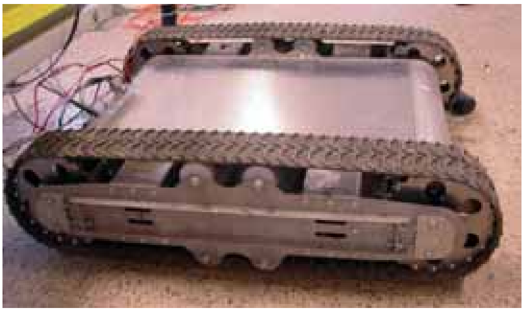

Figure 1: Robot crawler, acquired in kit form from www.superdroidrobots.com, as assembled without modifications.

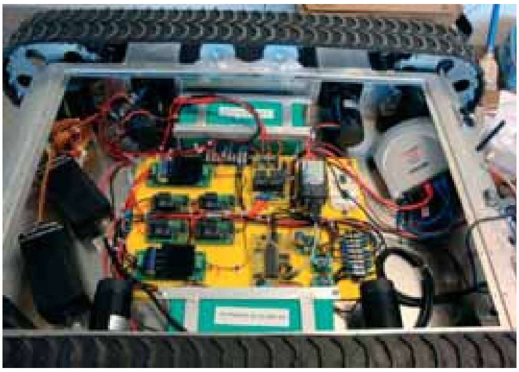

Figure 2: Robot crawler, internal electronics (motors, relays, power supplies, and wireless router).

2.2.2 Maneuvering camera

An Ethernet-based camera (Axis 213 PZT network camera) was installed to aid maneuvering (Fig. 3). Remote functions include zoom, pan, tilt, and exposure control. The camera also has infrared capabilities for functioning in the dark. The camera was mounted on a rigid post at the rear of the crawler.

2.2.3 Lift tables

Two specially modified electric lift tables were installed on the crawler (Figs 3 and 4). Capable of lifting the scanner up to a height of 2.5 m, the tables are powered off the crawler batteries, and activated through relays on the crawler microcontroller board. The scanner was then mounted on the top lift table. Failsafe switches were put in place to ensure that the crawler could be moved only when the lift tables are retracted, because of stability issues, and to give the maneuvering camera an unobstructed view when in motion.

2.2.4 Automatic leveling system

An automated leveling system was put in place as the LIDAR scanner ideally works from a leveled position (Fig. 5), and to reduce the risk of the crawler tipping over, especially if the lift tables are to be extended. Electric linear actuators, powered from the crawler batteries were installed vertically on four corners of the crawler. Each actuator has stroke

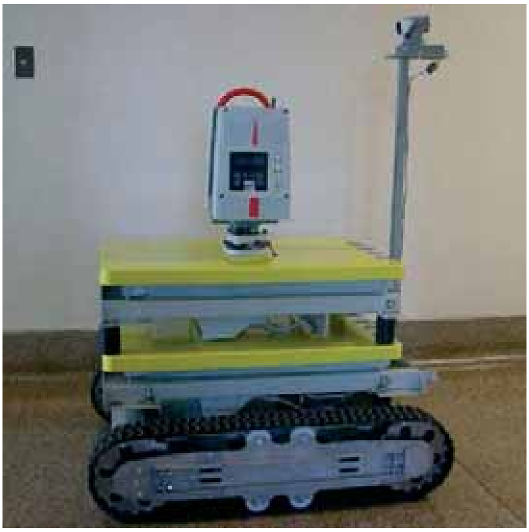

Figure 3: HDS6000 LIDAR scanner mounted on two modified lift tables on the crawler base. An Ethernet-based (Axis 213 PZT network) maneuvering camera can be seen on the crawler.

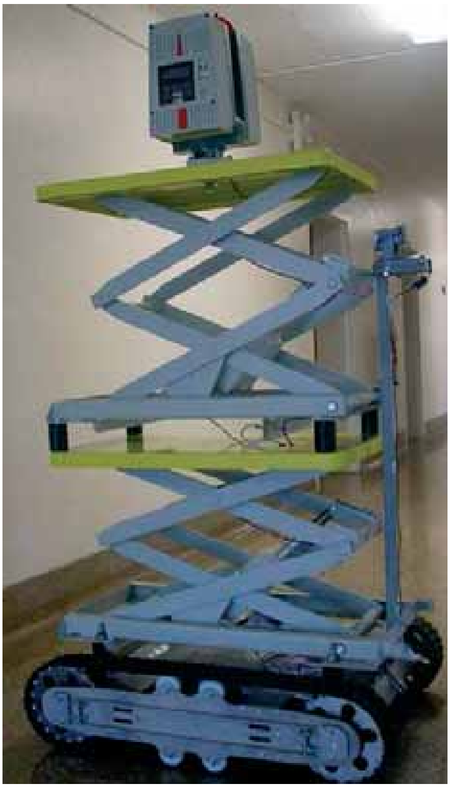

Figure 4: Lift tables extended to raise the HDS6000 scanner.

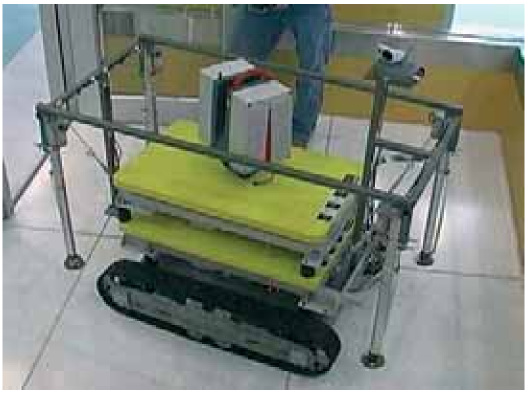

Figure 5: Automated leveling system added to the robot crawler.

length of about 18≡. On the base of each actuator is an articulated foot that allows the actuators to maintain good contact with the ground even while on uneven terrain. Limit switches located in the base of the feet to indicate when the actuators are all the way on the ground, as well as limit switches that indicated when the actuators are retracted all the way. (The robot is unable to move when the upper limit switches are not engaged because this indicates that the actuator is extended, and movement of the crawler would damage the leveling system.

The leveling system can operate completely automatically or manually. When in automatic mode the robot is brought to a complete stop and the micro controller tells all four actuators to move down until they touch the ground, based on the response to the lower limit switches. Once the actuators have made contact with the ground, the micro controller will determine how level the robot is. It accomplishes this with the use of two 3 axis accelerometers (only one is needed but two offers more reliability). Accelerometers were also used instead of an MEMS chip due to cost and the fact that this task did not need an extremely high level of accuracy. The accelerometers are not able to detect the degree that the robot is tilted directly, but with the use of the micro controller an angle can be obtained. Accelerometers only measure acceleration and in this case they measure gravity, which will always pull straight down, so if the accelerometers sense acceleration in any direction in other than the Z direction the micro controller will know that the device is not level. The angle that the robot is tilting can be determined by comparing the acceleration sensed by the X and Y axis to the total acceleration. Depending on the angle it was tilting, the micro controller would tell the appropriate actuators to extend until the accelerometer only reported acceleration in the Z direction. Because accelerations can be induced by the robot lifting the algorithm typically gets close to the level solutions, stop, and then refines the level.

In the event that one of the actuators loses contact with ground, the algorithm stops and the robot can be leveled manually, as the operator has access to individual accelerometers and tilt information in real time.

2.3 Remote control

All operations were controlled through a wireless router. A ruggedized laptop computer with a transreflective (sunlight viewable screen) was set up to control all functionality. Using the onboard wireless card, this single computer controls control all functions of the LIDAR as well as the crawler, including maneuvering, stabilizing (leveling), and vertical positioning of the LIDAR unit. In addition it controls the camera and displays real time video images from the camera. No other control devices are used with the exception of an RF ‘joystick’ control to maneuver the crawler to and from the start of the scanning site.

The wireless link is bi-directional, and in addition to the control functions, the return data stream includes LIDAR data, real time video, and details on speed, direction, acceleration, tilt, relay status, and power consumption of the crawler.

3 SCANNING PROTOCOLS

3.1 Maneuvering

During initial maneuvers to bring the device to close to the scanning location, the RF controller is used typically with the operator walking alongside the crawler (Fig. 5). Once close to the scanning location, the robot is controlled remotely through the laptop with the operator viewing the progress of the crawler remotely through the maneuvering camera, with the ability to pan, tilt, and zoom the camera (Fig. 6).

The crawler is then driven to an external window or other external opening or directly into the interior of the structure and positioned where the scan is to be taken. In most cases multiple scanning locations from different angles will be required to fill in gaps and extent the coverage of the scan.

After that the automatic leveling algorithm is employed to level the crawler in preparation for scanning. Finally the LIDAR unit is raised to the appropriate height for scanning.

3.2 Scanning

The Cyclone$^{®}$ software (a registered trademark of Leica Geosystems) provided with the HSD6000 is used for scanning. The software integrates seamlessly through the wireless transmission control protocol/Internet protocol interface.

The HDS6000 has no provision for an optical image preview, but has the ability to do a low resolution full dome scan which takes about 1 minute. The results of that scan are used to narrow down the field of interest to scan at a higher resolution. Scans are conducted

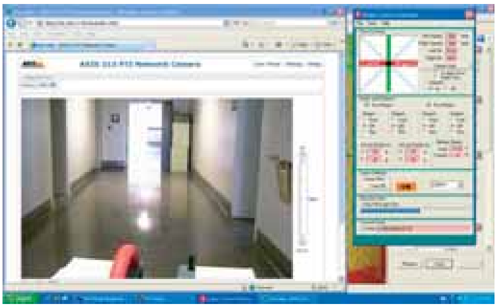

Figure 6: Operators view from the camera mounted on the crawler, with crawler control console on the right.

in either medium or high resolution (the scanner has an option of low, medium, high, very high, and ultra-high resolution). At a distance of approximately 8 meters, a medium resolution scan would have a resolution (scan spacing) of approximately 5 mm and approximately 3 mm in high resolution. Distance resolution of a single point in the scan is about 8 mm.

3.3 Registration of scans

The Cyclone software is also used to register the different scans. In order to make this as functional and efficient as possible in potentially dangerous areas, no survey control is used and not survey targets are required. Image registration is done automatically using a minimum of three ‘pick points’ common to any two images to be registered, selected by the operator.

4 3D MODELING AND VISUALIZATION

We developed a prototype software system based on OpenGL® (a registered trademark of SGI) and Visual C++® (a registered trademark of Microsoft Corp) that can automatically reconstruct the 3D scene of the interior of a building or other structure from point clouds acquired by the ground-based LIDAR scanner. We developed a user-friendly graphical user interface that allows the users to interactively visualize, navigate and walk through the room from different view angles, zoom in and out, etc. The reconstructed 3D scene can be exported in the ‘OBJ’ data format that is fully compatible and exportable to other commercial visualization software.

Given the LIDAR data of an interior room, we will conduct hierarchical segmentation and then reconstruct the corresponding 3D surface. More specifically, we first identify all the major planar regions such as floor, ceiling and vertical walls of the room. Next, we identify and extract individual objects such as chairs, tables that are lying/attaching to floors/ceiling/walls. We can further segment each extracted object such as chair into homogenous patches. Finally, we will reconstruct the 3D surface for each segmented patches based on a novel volumetric vector field construction. We will describe each of these steps in more details in the following.

4.1 Floor and wall identification

The goal of this step is to identify all the major planar regions in the scene such as floor, ceiling and vertical walls of the room. We provide two ways of doing this. If we can assume the room is a rectangular cube, then we will do automatic segmentation. If the interior scene is more generic and not a rectangular cube, then we will conduct user-guided semi-automatic segmentation.

4.2 Automatic plane identification for rectangular room

We first compute the bounding box that encloses the LIDAR data. Assuming that the LIDAR data is scanned from one side of the bounding box such as the windows of the room, we then search for major planar regions such as floor, ceiling and walls in the neighborhood of the other five sides of the bounding box. More specifically, for each side of the bounding box, we first extract all the point clouds that are within a certain distance of the current side of bounding box. We then use random sample consensus (RANSAC) to find the best plane that fit these selected point clouds. RANSAC is a hypothesis generation and testing algorithm that is very robust for outliers [1]. The main idea behind the technique is to use a minimal number of data points needed to estimate the model (i.e. fitted plane), and then count how many of the remaining data points are compatible with the estimated model, in the sense of falling within a chosen threshold.

The RANSAC-based plane fitting algorithm can be briefly sketched as

```
repeat
draw a sample of n points from the data uniformly and at random
fit a plane to that set of n points
estimate percentage of inliers (i.e. points within a certain distance of the plane)
until satisfying solution (e.g. inliners > 95%)
refine plane fitting (using all inliers)
```

The plane fitting is based on the robust principal component analysis (PCA) algorithm which can compute local surface properties based on local neighborhoods of sample points. We find the k-nearest neighbors of a sample point p, denoted by the index set Np, The local surface properties of the point clouds can be efficiently estimated by the eigenanalysis of the covariance matrix C of a local neighborhood at sample point p:

$$C = \begin{bmatrix} p_{i_1} - \bar{p} \\ \cdots \\ p_{i_k} - \bar{p} \end{bmatrix}^T \cdot \begin{bmatrix} p_{i_1} - \bar{p} \\ \cdots \\ p_{i_k} - \bar{p} \end{bmatrix}, i_j \in N_p, \tag{1}$$

where $\bar{p}$ is the centroid of the neighbors of p.

Consider the eigenvector problem:

$$C \cdot v_l = \lambda_l \cdot v_l, l \in \{0,1,2\} \tag{2}$$

Since C is symmetric and positive semi-definite, all its three eigenvalues l0 £ l1 £ l2 are real-valued and the eigenvectors vl form an orthogonal frame, corresponding to the principal components of the point set. Thus v0 approximates the surface normal at p, or in other words, v1 and v2 span the tangent plane at p. Note that n is the size of the neighborhood, which serves as the scale-control parameter, and is dependent on the laser scanner resolution, i.e. how dense the point clouds data is.

4.3 User-guided semi-automatic planes identification for generic interior scenes

For generic interior scenes, the above six-sided rectangular cube assumption of the interior scene will not always hold. Thus in this case, we implemented a new region growing–based segmentation algorithm.

The basic idea of the region-growing algorithm is: start with an unvisited point p, iteratively include its neighboring point q, if the distance between p and q is smaller than a threshold (e.g. the sampling density), and the difference between the normal at p and the normal at q is also smaller than a threshold (2 degrees in our case). This can be done in a breadth-first search, and stop when the above criteria is no longer hold. Then the algorithm move on to the next unvisited point, until all the points are visited.

To speed up the computation, we employed volumetric grid to store the points, thus the connectivity inference can be done instantly by grid indexing. The surface normal is estimated by the aforementioned PCA.

After the region growing, we further merge clusters that are co-planar, i.e. we would like to merge all the clusters that belong to a bigger plane even though they are not connected. More specifically, a cluster is merged with another if the normal of both clusters are within 2 degrees of each other, AND the vector pointing from the first cluster's midpoint to the second cluster's midpoint is more than 85 degrees apart from the average of the two cluster's normals (by the first condition, these two normals will be already nearly identical).

Finally, the floor and ceiling will be automatically identified based on the height. The walls will be interactively identified by the user.

4.4 Individual objects identification and extraction

Once all the point clouds in these major planar regions such as floor, ceiling and walls are identified and removed, there are sufficient separation between points of individual objects such as chairs, tables that are lying/attaching to the ground, ceiling, and walls. We then proceed to group all these points based on their proximity to each other by finding connected components such that each point in a connected component is within a given distance to at least one more point in that component. Hence, all the points belonging to an individual object such as a chair would lie in a single connected component. We employed efficient grid-based range-finding algorithms by using a volumetric grid to store the points to speed up the connected component analysis process. Each extracted object can be further segmented into homogenous patches/clusters based on some similarity metrics (e.g. distance, normal, curvature). We employed a region growing–based segmentation algorithm which is the same method we used for user-guided semi-automatic planes identification for generic interior scenes, described in Section 4.1.3. The region growing–based segmentation algorithm works very well in our experiments. Figures 7–9 illustrates the process of 'Floor, ceiling and vertical walls identification' and 'Individual objects identification and extraction'. Figure 10 shows an example of segmenting the extracted objects into individual homogenous patches. Each segmented patch is represented by a single color.

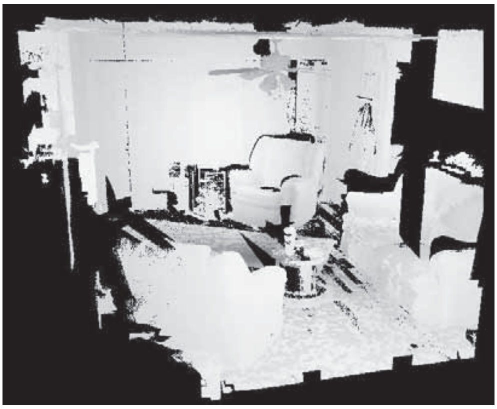

Figure 7: Original LIDAR data.

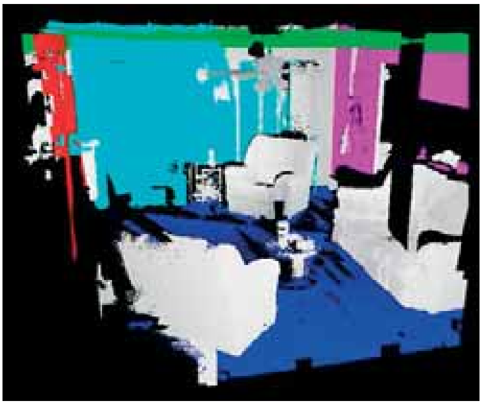

Figure 8: Features identified in different shades: Floor, ceiling, and vertical walls.

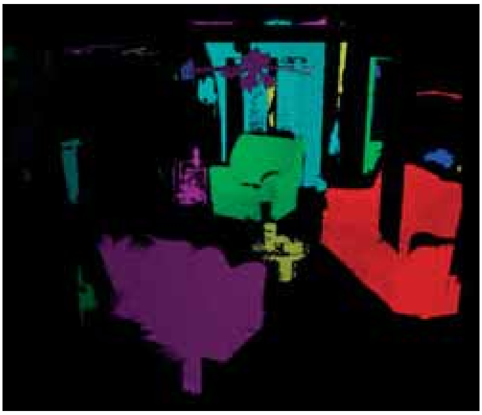

Figure 9: Individual objects extracted from the LIDAR data (shown in different shades).

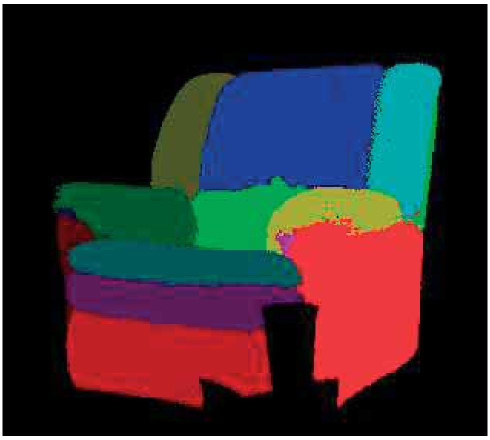

Figure 10: Segmenting the extracted objects into individual homogeneous patches. Each segmented patch is represented by a different shade.

4.5 3D surface reconstruction

There are two main steps in the 3D surface reconstruction phase. First, a saliency weighted normal vector field is constructed based on the 3D points. Next, a watertight 3D surface is extracted from the saliency weighted normal vector field by energy minimization. The saliency weighted normal vector field is constructed by the following three steps: (1) saliency field construction by anisotropic kernel density estimation; (2) normal estimation and consistent normal orientation propagation; (3) volumetric saliency weighted normal vector field construction.

4.5.1 Saliency field construction by Anisotropic Kernel density estimation

We use the term saliency to represent the likelihood the unknown surface passes through a certain part of 3D space. In this paper, we propose to employ Parzen window-based nonparametric density estimation method to compute the saliency of each point.

Given n data points x_i, $i = 1, \dots, n$ in the d-dimensional Euclidean space R^d, the multivariate kernel density estimate obtained with kernel $k(x)$ and window radius h (without loss of generality, let's assume $h = 1$ from now on), computed in the point x is defined as:

$$f(x) = \frac{C_{k,d}}{n} \sum_{i=1}^{n} k(\| x - x_i \|^2) \tag{3}$$

$C_{k,d}$ is the normalizing constant, $||x||$ is the L_2 norm (i.e. Euclidean distance metric) of the d-dimensional vector x. There are three types of commonly used spherical kernel functions $k(x)$: the Epanechnikov kernel, the uniform kernel, and the Gaussian kernel [2].

For 3D point cloud obtained by depth estimation, the outliers tend to spread in the space randomly, while 'real' (we use a quotation here to emphasize the fact that the real surface is unknown) surface points will spread along a thin shell which encloses the real surface object. In other words, the distribution of the outliers is relatively isotropic, while the distribution of the real surface points is rather anisotropic. Hence in this paper, we propose to employ an anisotropic ellipsoidal kernel-based density estimation method. More specifically, for anisotropic kernel, the L_2 norm $\|x\text{-}x_i\|$ in the above equation, which measures the Euclidean distance metric between two points x and x_i will be replaced by the Mahalanobis distance metric $||x\text{-}x_i||_M$:

$$\| x - x_i \|_M = ((x - x_i)^t H^{-1} (x - x_i))^{1/2} \tag{4}$$

here H is the covariance matrix defined as:

$$H = DD^T$$

and

$$D = (x_1 - x, x_2 - x, \dots, x_n - x)$$

Geometrically, $(x - x_i)^t H^{-1}(x - x_i) = 1$ is a three-dimensional ellipsoid centered at x, with its shape and orientation defined by H. Using single value decomposition, the covariance matrix H can be further decomposed as:

$$H = UAU^T$$

with

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{5}$$

$\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the three eigenvalues of the matrix H, and U is an orthonormal matrix whose columns are the eigenvectors of matrix H.

To compute the anisotropic kernel-based density, we will apply an ellipsoidal kernel E of equal size and shape on all the data points. The orientation of the ellipsoidal kernel E will be determined locally. More specifically, given a point x, we will calculate its covariance matrix H by points located in its local spherical neighborhood of a fixed radius. (Without loss of generality, we will assume the radius is 1, which can be done by normalizing the data by the radius). The U matrix of H calculated by the covariance analysis is kept unchanged to maintain the orientation of the ellipsoid. The size and shape of the ellipsoid will be modified to be the same as the ellipsoidal kernel E by modifying the diagonal matrix A as:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r \end{bmatrix} \tag{6}$$

r is half of the length of the minimum axis of the ellipsoidal kernel E.

4.5.2 Normal estimation and consistent normal orientation propagation

Given the 3D point clouds, we can estimate the normal vector at each point based on the PCA algorithm [3]. Normal vectors estimated by the PCA algorithm however has an ambiguity of 180 degree so might not be consistently oriented. An orientation propagation is often needed to ensure the consistent orientation of the normal vectors. One way to do this is to first build a graph with each point as a node and the weights of edges between the adjacent points are defined as $1-\|n_1 \cdot n_2\|$, where n_1 and n_2 are the normal vectors of the two adjacent points, and then compute the minimum spanning tree (MST) from the graph using algorithms such as the Kruskal's algorithm [4] which finds a subset of the edges that forms a tree that includes every vertex in the graph, where the total weight of all the edges in the tree is minimized. At the termination of the algorithm, the normals are adjusted so the two neighbors in the tree have consistent normal orientation.

The above MST-based normal orientation propagation approach however is not robust against noises and outliers. In this paper, we propose to utilize external knowledge to guide the normal orientation propagation (Fig. 11). Particularly, since the point clouds are generated by LIDAR scanner, for a given point p, it should be visible to the LIDAR scanner, i.e. the dot product between the normal vector of the point p and the view direction of the LIDAR scanner should be negative. If not we will reverse the normal orientation at this point.

4.5.3 Volumetric saliency weighted normal vector field construction

Once we have estimated the saliency and normal vector at each point, we will proceed to construct a volumetric saliency weighted normal vector field, from which a watertight 3D surface can be extracted by energy minimization. A volumetric grid embedding all the 3D

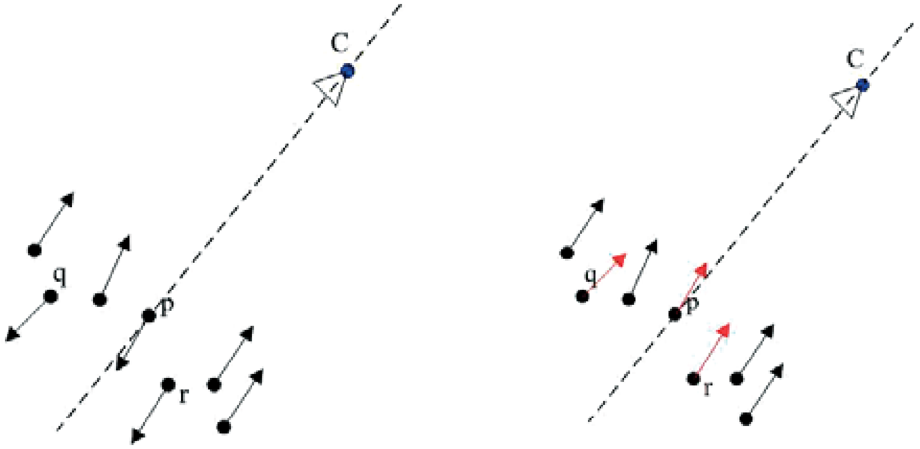

Figure 11: LIDAR viewing direction guided normal orientation propagation. Left: 3D point shown with its normal vector estimated by the PCA algorithm. The orientation of the normal vectors at point p, q and r are not correct. Right: Based on the view direction of the LIDAR scanner, the orientation of the normal vectors at point p, q and r are reversed so that they are now opposite to the view direction of camera C.

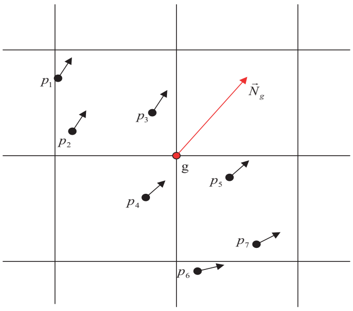

Figure 12: Construction of the volumetric saliency weighted normal vector field. The normal vector at grid node g is calculated as the weighted sum of the normal vectors of its adjacent points $p_1, p_2, ..., p_7$.

points is first constructed. The saliency and the normal vector of each point are then propagated to its adjacent grid nodes (Fig. 12). Specifically, the saliency S_g of a grid node g is computed as the weighted summation of the saliency S_{pi} of its adjacent points P_i:

$$S_g = \sum_i C_{p_i} S_{p_i} \tag{7}$$

The weight C_{pi} is calculated using the aforementioned Parzen window kernel function based on the anisotropic Mahalanobis distance between the grid node g and point P_i . Since the kernel we used (e.g. truncated Gaussian kernel) has finite support here, only a finite number of points P_i ($i = 1,..., n$) within the kernel radius has non-zero weights C_{pi}. The normal vector $\vec{N}_g$

at grid node g is calculated similarly as the weighted summation of the normal vector $\vec{N}_{p_i}$ of its adjacent points P_i:

$$\vec{N}_g = \sum_i C_{p_i} S_{p_i} \vec{N}_{p_i} \tag{8}$$

4.5.4 3D shape estimation by graph-cut

Once the volumetric saliency weighted normal vector field is constructed, a watertight 3D surface S can be extracted by energy minimization. We use the following energy functional as suggested by [5]:

$$E = E_{data} + \varepsilon E_{reg} \tag{9}$$

E_{data} is the data alignment term which is the inverse of the flux that enforces the surface alignment with the data orientation:

$$E_{data} = -flux(S) = -\int_S \langle N, v \rangle ds \tag{10}$$

where <,> is (Euclidean) dot product and N is unit normal to surface elements ds consistent with a given orientation. If vectors v is interpreted as a local speed in a stream of water then the absolute value of flux equals the volume of water passing through the hypersurface in a unit of time. The sign of flux will be determined by the orientation of the surface. E_{reg} is the area-based regularization term that maintains the regularity of the extracted surface:

$$E_{reg} = \int_S ds \tag{11}$$

ε is the coefficient of the regularization term E_{reg} that controls the strength of the smoothness in the energy minimization process and is related to the sampling density of the data. In our experiment, we set ε as 0.2.

As pointed out by [5], combining flux with area-based regularization can overcome the shrinking effect of the area-based regularization and improve the reconstruction of elongated structures, narrow protrusions, and other fine details. Based on the divergence theorem for differentiable vector fields, the integral of flux of vector field over surface S equals to the integral of vector field's divergence *div(v)* in the interior of S:

$$\int_S \langle N, v \rangle ds = \int_V div(v_p) \cdot dp \tag{12}$$

Where V is the region enclosed inside S. Thus E is now:

$$E = \varepsilon \int_S ds - \int_V div(v_p) \cdot dp \tag{13}$$

This equation can be solved efficiently using the graph cut algorithm [5]. A typical graph construction is shown in Fig. 13: neighboring nodes are connected via *n*-links representing area-based regularization cost. Nodes are also connected to the terminals via one *t*-link based on their divergence value: blue nodes have positive divergence and are connected to the source terminal *s* with weight $div(v_p)$; red nodes have negative divergence and are connected to the sink terminal *t* with weight $-\ div(v_p)$; the black node has zero divergence and is not connected to either terminal. The weight of the *n*-link is defined as the inverse of the edge length so that the weights of severed *n*-links approximate the surface area [6]. Consequently, a global minimum surface for the above equation can be found by computing a minimal *s*/*t*-cut in the constructed graph [5].

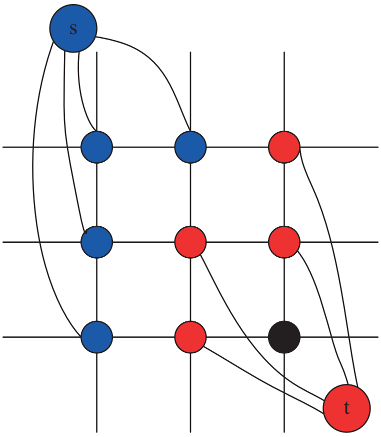

Figure 13: Shape estimation by graph cut. Graph construction for energy minimization: neighboring nodes are connected via *n*-links representing regularization cost. Nodes are also connected to the terminals vial *t*-links based on their divergence value: blue nodes have positive divergence and are connected to the source terminals; red nodes have negative divergence and are connected to the sink terminal *t*; the green node has zero divergence and is not connected to either terminal.

5 CASE STUDIES

During this study, seventeen different interior spaces were imaged and analyzed. Most scans were taken with the Leica HSD 6000 scanner mounted on the robot crawler; although two of the early scans were made using a tripod mounted Leica ScanStation II. Some scans were external scans into a structure through windows or openings; others were scanned by driving the robot crawler inside the structure using remote control. In all cases multiple scans were taken and registered using three pick points common to at least two of the images to be registered. In some cases both internal and external scans were used and registered to together.

The types of structures scanned included rooms in residential homes, commercial storefronts, public spaces such as large foyers and gymnasiums, offices and office buildings, and underground mine and a cave, a simulated bomb making facility, a simulated chemical and biological laboratory, and an abandoned vehicle with a simulated improvised explosives device.

In this paper we give examples of three of these investigations.

5.1 Residential family room

A residential family room was imaged through a three pane window using the Leica Scan Station II. Figure 14 shows the reflective nature of the window, which did not allow optical imaging into the interior of the structure. Figure 15 also shows the resulting LIDAR scan which includes elements of the window frame. In all three scans were taken at different angles through the window, and registered to each other using pick points. Figure 16 shows the results of the three scans merged with the exterior wall cropped from three different angles. Items that can be seen include furnishings, hardwood floor texture, ceiling tiles, louvered doors, books, television, ceiling fan, items on coffee table, etc. In the visualization module this space can be rotated and viewed from all angles including from the inside.

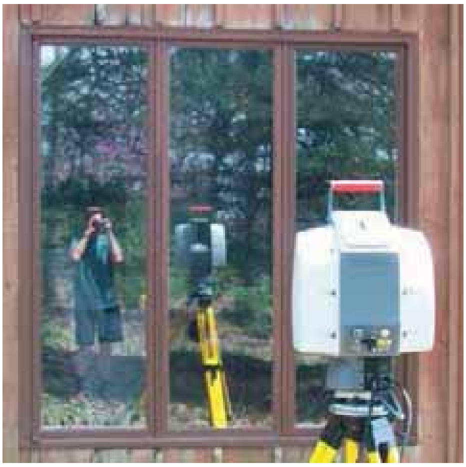

Figure 14: Digital photography of LIDAR setup outside window.

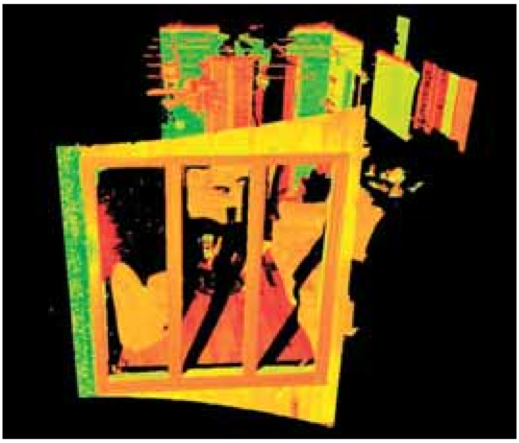

Figure 15: LIDAR 3D laser map of interior room from a single scan.

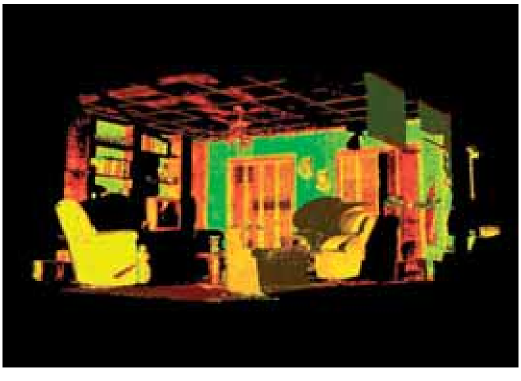

Figure 16: Visualization of an interior of a room. Colors represent the intensity of the laser reflection.

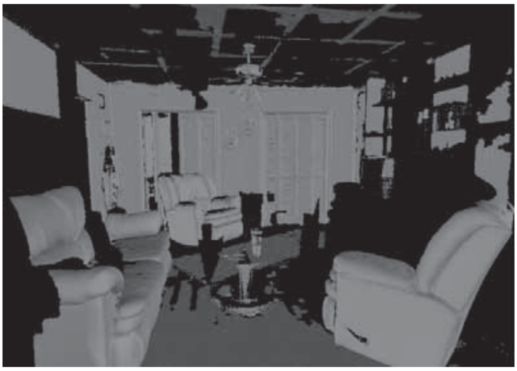

Figure 17: 3D surface reconstruction of room.

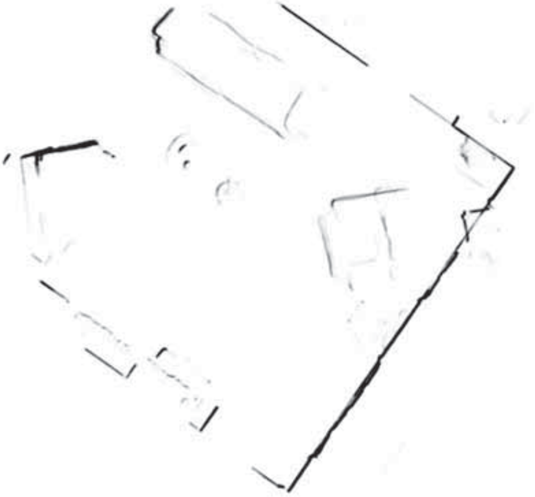

Figure 18: Floor plan of the room.

Figures 17 shows the elements of the room, after individual elements have been identified and extracted then reconstructed in this image. ‘Black’ zones are shadow areas. Figure 18 shows the automatically generated floor plan of the room.

5.2 Multipurpose gymnasium

The multi-purpose gymnasium building at Missouri S&T was scanned using the HDS6000 scanner, remotely driven into the structure on the robot crawler through an open doorway (Fig. 19). Six internal scans were registered together, and show doors, windows, catwalk, and recessed hidden areas behind stored equipment (Fig. 20). Figure 21 shows the automatically generated floor plan of the building.

5.3 Experimental mine

An entry adit at the Missouri S&T experiment mine was also scanned using the HDS6000 scanner, driven in on the robot crawler through a ground level entrance (Fig. 22). Five internal scans were registered together and show the underground space and items stored in the tunnel (Fig. 23). Figure 24 shows the layout of the mine.

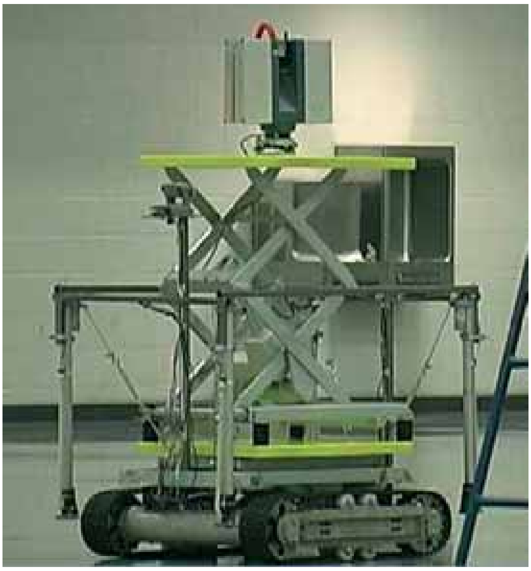

Figure 19: Maneuvering the robot crawler in the gymnasium and positioning the LIDAR high for one of the interior scans.

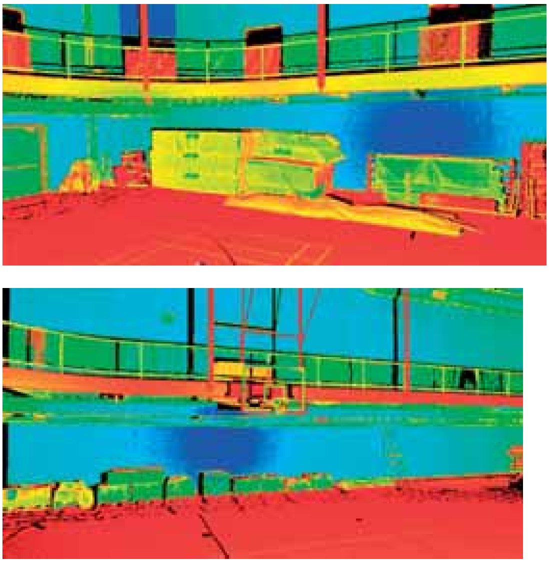

Figure 20: Interior scans of the multi-purpose building gymnasium.

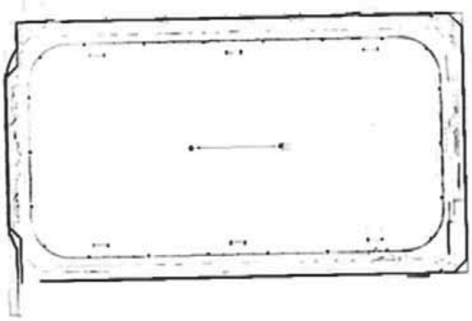

Figure 21: Floor plan of the multi-purpose building gymnasium.

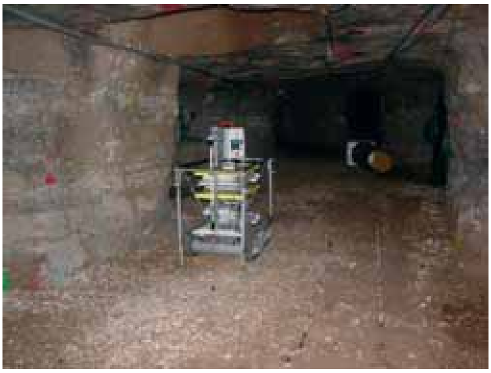

Figure 22: LIDAR scanning in the MS&T experimental mine.

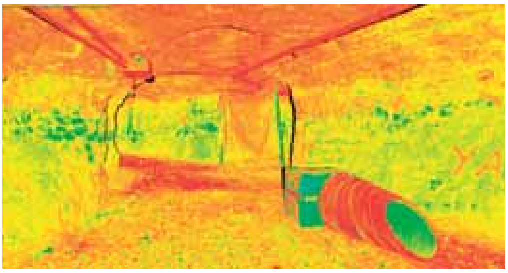

Figure 23: Room in the MS&T experimental mine showing stored equipment and services.

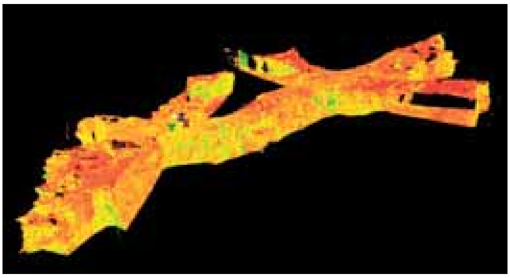

Figure 24: Registered composite image of the layout of the MS&T experimental mine.

6 SUMMARY AND CONCLUSIONS

This paper presents the development of an interior surveillance system that using LIDAR can safely be used to scan the interior of potentially dangerous structures. For both military and civilian first responders, whether for the purpose of determining structural stability, presence of dangerous items or personnel, or simply to map the interior for egress and potential hiding spaces, the system can scan through windows or other transparent openings or even from inside.

A LIDAR scanner is carried to a window, or transported into the structure itself by a remotely controlled robot crawler. The LIDAR is maneuvered into position from a safe distance using a single laptop computer using standard wireless connectivity to drive the crawler and using a wireless network camera for guidance. Once in place the scanner is automatically leveled, and the scanner raised to the desired height. Several scans at different locations are taken to provide complete coverage of the inside of the structure, and are later registered into a single image.

Software has been developed to reconstruct the 3D scene of the interior of a building, to identify specific components of the interior, and to interactively visualize, navigate and walk through the room from different view angles, zoom in and out. Floor plans are automatically generated and a data export facility has been developed.

We have tested the system on seventeen different sites, from small rooms and offices to large public structures and a mine and a cave. In all cases the information acquired can be used by first responders to visually the interior layout, look for potential dangers and hazards, including potential hiding areas and paths of egress.

The resolution of the system as applied in this program is less than 10 mm. As such it is more than adequate for creating floor plans, identifying methods of egress, hiding places, inhabitants, and dangerous objects.

A further application of this technology which is currently under investigation will be used to assess the damage to structural components so that the integrity of structures such as buildings or bridges can be assessed. This aspect may require the LIDAR to operate at higher resolutions.

ACKNOWLEDGMENTS

Research was sponsored by the Leonard Wood Institute in cooperation with the U.S. Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-07-2-0062. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Leonard Wood Institute, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation heron.

REFERENCES

[1] Fischler, M.A. & Bolles, R.C., Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM 24*, pp. 381–395, 1981. doi: http://dx.doi.org/10.1145/358669.358692

[2] Comaniciu, D. & Meer, P., Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Learning*, **24(5)**, pp. 603–619, 2002. doi: http://dx.doi.org/10.1109/34.1000236

[3] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J.A. & Stuetzle, W., Surface reconstruction from unorganized points. *SIGGRAPH 1992,* pp. 71–78.

[4] Kruskal, J.B., On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, **7(1)**, pp. 48–50, 1956. doi: http://dx.doi.org/10.1090/S0002-9939-1956-0078686-7

[5] Lempitsky, V.S. & Boykov, Y., Global Optimization for Shape Fitting. *CVPR* 2007.

[6] Boykov, Y. & Kolmogorov, V., Computing Geodesics and Minimal Surfaces via Graph Cuts. *ICCV* 2003, pp. 26–33.