Visibility-Based Urban Exploration and Learning Using Point Clouds

Matthew Hielsberg[1], Richard Tsai[2], Peihong Guo[1] and Charlie Chen[2]

***Abstract*— We present an integrated algorithm for computing vantage points of maximal visibility in an *a priori* unknown fully 3D environment using point clouds and a multi-resolution Hausdorff metric based surface reconstruction procedure. This work aims at demonstrate the proposed algorithm's ability to explore and learn complicated 3D urban environments. We present results that validate the proposed algorithm by using a realistic LiDAR simulations in a virtual MOUT site.**

I. INTRODUCTION

We propose an integrated algorithm for achieving certain goals which are related to the visibility (the ensemble lines-of-sight) from a group of observing locations in a complex environment. The relevant goals here are of exploratory and surveillance nature. In such an environment, there are many geometrically complicated obstructions to the lines-of-sight. Furthermore, the obstruction may be unknown *a priori* and must be learned. An example of such a goal includes the mapping of complex urban areas using point cloud data that are collected at nontrivial locations. Point clouds are typically gathered through devices such as LiDAR, and they can be regarded as effectively unstructured discretizations of the visibility from the corresponding observing locations. Hence, in this paper we shall address the essential question "*How does one generate a sequence of observing locations that ensures an accurate mapping/learning of the domain?*"

We utilize an implicit representation of the observed environment to efficiently and accurately handle line of sight and related necessary extraction of volumetric information. We advocate this approach due to the convenience it offers in handling various essential boolean operations on the occlusion from different vantage points. However, we also recognize the utility of explicit methods, especially in the areas of compression and encoding, where in general relatively few parameters are needed for approximation of the original data as compared to the full volumetric storage used by implicit methods. Hence, we adopt a hybrid approach whereby explicit surfaces are extracted from the collected point cloud, and used to generate an implicit representation. In this setting we are able to take advantage of the lightweight local polynomials, enabling such applications as multi-vehicle coordination, as well as process the implicit representation for onboard visibility, surveillance, and navigation applications.

We work with the assumption that point clouds can be sampled from obstacle surfaces in a domain of interest from a given set of vantage positions. For our purposes a point cloud D is a finite set of three dimensional data points $\{x_j\}$ given in an appropriately chosen reference coordinate frame. Additionally, we assume that the observer position is known relative to our chosen domain origin. This is typically not a poor assumption as GPS and inertial measurement units have become quite ubiquitous in mobile robotic platforms and even in smart phones and tablets.

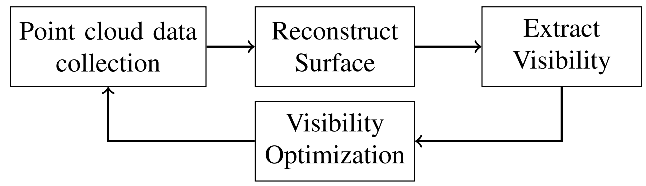

Fig. 1. System Overview. An observer is positioned and a scan of the simulated environment is performed. An explicit representation of the surface is generated from the point cloud. With this explicit representation the visibility is computed from the observer's point of view. The visibility is integrated and evaluated to generate the next observer position.

We advocate the use of the Hausdorff metric [12] in controlling the quality of point cloud approximation as it has been shown to perform better than standard metrics [17]. The least squares metric, for instance, does not adequately penalize surfaces that miss important thin features such as wires and poles. Also, function norm metrics are coordinate biased and therefore do not handle well arbitrarily oriented surfaces. For applications such as navigation the reason for choosing an appropriate metric becomes clear as any collision with these thin but common objects is not acceptable.

In summary, we shall describe in this paper (1) a Hausdorff based, multi-resolution algorithm that reconstructs the already explored occluding objects in the working domain based on the point cloud data; (2) an algorithm that utilizes the (portions of) reconstructed obstacle surfaces for exploring environments with complicated and unknown obstacles. Furthermore, this algorithm shall determine sequentially new observing locations in order to obtain more information about the obstacles in the environment. As a result of the exploration, a complete map as well as a 3D volumetric and parametric representation of the environment are obtained.

II. RELATED WORK

The visibility-based navigation problem has been approached from several view points. Tovar, et al. have considered combinatorial approaches to the problem in static 2D polygonal environments. The methods presented typically involve vehicles with limited sensing capability, and rely on detection and tracking of discontinuities in depth information

*This research was supported by the ARO/DoD contracts W911NF-07-1-0185 and W911NF-12-0519.
[1]Texas A&M University, College Station hielsber@tamu.edu
[2]The University of Texas at Austin ytsai@ices.utexas.edu

as the vehicle moves [10], [18], [19], [20], [21]. Tovar et al. have also demonstrated the use of gap navigation trees [19] on a Pioneer 2-DX platform using SICK laser range sensors [18]. However, as these methods are focused on environments consisting of simply connected polygons they are not easily applied to the more general problem, including outdoor environments.

Tsai et al. [22] proposed a PDE based method for computing a continuous representation of visibility from a given vantage points; in that work the obstacles are assumed to be given *a priori* as level set functions. Landa et al. [9] proposed an exploration algorithm for 2D planar domains using range data. This algorithm can be categorized as an "edge chasing" algorithm formally introduced by LaValle et al., see e.g. [21].

The method presented in this article utilizes local piece-wise linear fits and space partitioning data structures that allow for points to be acquired in arbitrary order from multiple sensors. Wolf et al. [27] introduced a method using planar reconstructions of the point cloud data, and provided experimental results on a mobile platform. However, the planar fits in their method were generated using a point cloud segmentation method based on the Hough transform, and they do not consider the identification of thin structures. In their application the goal is to identify relatively small planes and merge as many as possible in order to generate simplified polygonal models of buildings in the environment.

Another prominent class of navigation algorithms have been developed under the simultaneous localization and mapping (SLAM) framework [16][6][2][11]. This framework typically utilizes extended Kalman filters or particle filters [8] to simultaneously deduce the location of the vehicle and the location of select landmarks in the environment. These methods have been quite successful and offer solutions to GPS-denied environments, but heavily dependent on reliable and repeated landmark recognition.

The remainder of this article proceeds as follows. Section III covers the problem of visibility using point clouds when the map is not known *a priori*. This section also proposes a method for computing vantage points that maximize the volume of the shadow region that is revealed at each step. In Section IV we show how the point clouds are adaptively partitioned and reconstructed using piece-wise linear fits, and how these fits are used to construct visibility volumes. Section V describes the simulated environment utilized in our numerical experiments. The numerical experiments and conclusions are then covered in Section VI and Section VII.

III. Visibility based exploration

The problem of visibility seeks to determine the regions in space visible to a given set of observers in the presence of occlusions. In our setting, the occlusions correspond to obstacles such as buildings, light poles, and trees that are common in urban environments.

By generalization of an idea first published by Valente et al. [24] we propose a new visibility-based "non-myopic" exploration algorithm for the fully 3D cases. Using the point cloud data collected on-line, this algorithm generates a sequence of observing locations, $x_1, x_2, \cdots, x_k$, and a path that goes through these locations while avoiding the obstacles. More precisely,

- The *k-th* observing location, x_k, will be chosen to be where a "visibility metric" E achieves a global maximum in a bounded domain;
- The "visibility metric" is defined by the point cloud data obtained from the observing locations $x_1, \cdots x_{k-1}$.
- The "visibility metric" is designed to quantify the potential gain in new visibility information by placing a new observing location at certain designated locations, given the previous observing locations.

Below we list the essential ingredients in defining such a visibility metric:

- **The set of observing locations** $\mathcal{O}$: The metric depends on the visibility from a set of observing locations contained in $\mathcal{O} = \{x_1, \cdots, x_k\}$.
- **The obstacles** Ω: They are assumed to be a closed set containing finite number of connected components.
- **Occlusion set** $\mathcal{S}$: The occlusion set contains points that are not visible from (the observing locations in) $\mathcal{O}$. When the visibility from $\mathcal{O}$ encompasses the entirety of the domain except the obstacles, then the occlusion set is identical to the obstacles. In this case, we shall say that the observing locations in $\mathcal{O}$ have ***complete visibility*** of the domain.
- **Shadow boundary** $\partial\mathcal{S} \setminus \bar{\Omega}$: A hyper-surface which separates the domain locally into visible and occluded regions from $\mathcal{O}$. The shadow boundary is the attenuation of the visible portions of the obstacles along the lines-of-sight. Note that the visible portions of the obstacles are not counted as part of the shadow boundary.
- **Viewing angle** θ: This is the angle between the line-of-sight that reaches the shadow boundary and the normal of the shadow boundary at the intersection.
  The viewing angle penalizes the viewing of the shadow boundary at grazing angles. It also reflects on the uncertainties due to errors in the visibility information. The closer this angle is to $90°$, the more sensitive the view will be to perturbation in the obstacles.
- **The weighting** w**:** Additional weighting may be placed over the entire domain for additional application-specific modeling needs, such as prioritization of a subset of the domain or dealing with visibility with limited range.

With all these ingredients, we define the visibility metric as a boundary integral taking the following form:

$$E(z; \mathcal{O}) \quad := \quad \int_{\partial\mathcal{S}\setminus\bar{\Omega}} \tau w \, \max(\vec{r}_z \cdot \vec{n}, 0)\, dy. \tag{1}$$

Here $\vec{r}_z(y)$ is $(z-y)/|z-y|$ the viewing vector, and $\vec{n}$ is the normal of $\partial\mathcal{S}_z$ which points into the occlusion set $\mathcal{S}$. Thus, the positive part of inner product, $\max(\vec{r}_z \cdot \vec{n}, 0) = \max(\cos\theta, 0)$, implements the consideration about the viewing angle.

Fig. 2 demonstrates a step-by-step computer simulation using this technique. We point out here that this procedure is *transparent to the dimension of the problem* – the

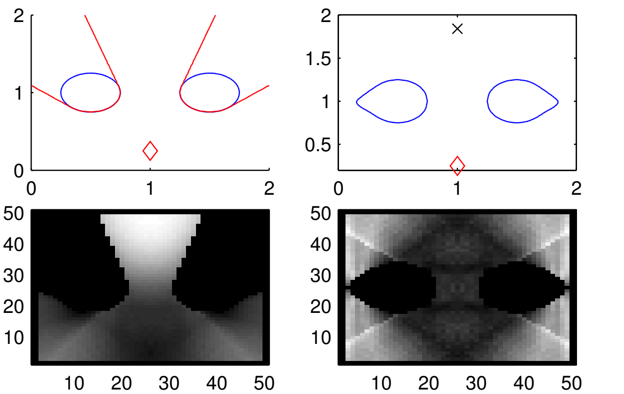

Fig. 2. Generation of an "optimized" set of observing locations via the use of E defined in (1). The blue curves shown in the top row are the estimates of the obstacles obtained from the observing locations. The cross corresponds to the newly added observing location. The diamonds indicate the current and previous observing locations. The images on the second row show the values of the visibility metric so that the whiter the color at a point, the larger the value is.

construction of the observing locations depend solely on the visibility metric, which does not require any dimension specific logic. In Fig. 4 we show the performance of the algorithm in vantage point placement for more complicated pipeline (topological) configurations.

A. Algorithm details

Our current implementation of the visibility metric evaluation requires discretization of (a portion of) the domain as a three dimensional Cartesian grid. The methods takes as input a function ϕ_{x_c} defined on this grid such that $\phi_{x_c} > 0$ corresponds to the free space visible from the vantage point x_c, and correspondingly the occlusion set $\mathcal{S} = \{\phi_{x_c} \leq 0\}$. Such a function is created by our surface reconstruction algorithm presented in Section III. In the simulations presented below, we expect that the dominant obstacles have more or less uniform sizes, so we set the depth factor to be 1, i.e. $\tau \equiv 1$.

Given the current vantage point x_c:
$\mathcal{O} \leftarrow \{Null\}$,
$E(x_c;\mathcal{O}) \leftarrow \infty$, $\psi \leftarrow \phi_{x_c}$ from LIDAR(x_c)
for $i = 1...N$ **do**
  $\mathcal{S} := \{y \,|\, \psi(y) \leq 0\}$
  Redistance ψ
  for all $x \notin (\mathcal{O} \cup \mathcal{S})$ and $\Delta\psi(x) > M$ **do**
    $\omega_{\mathcal{O}}(y) := H(\max_{z\in\mathcal{O}} \{(y - z) \cdot \nabla\psi(y), 0\})$
    $\omega_c(y) := \max(-(x - y) \cdot \nabla\psi(y))$
    $E(x;\mathcal{O}) := \int_{\partial\mathcal{S}\setminus\bar{\Omega}} \omega_c(y)\, dy$
      $\approx \int_{\mathbb{R}^3} \omega_c(y)\, \omega_{\mathcal{O}}(y) \delta_\epsilon(\psi(y)) dy$
  end for
  $x_c := \arg\max E(x)$
  $\mathcal{O} \leftarrow \mathcal{O} \cup \{x_c\}$
  $\phi_{x_c} \leftarrow$ LIDAR(x_c)
  $\psi \leftarrow \max(\psi, \phi_{x_c})$
end for

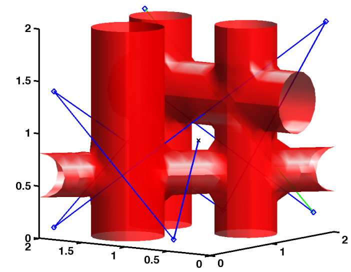

Fig. 3. A slightly more complex 3D pipeline configuration. Notice also the symmetry of the planned vantage point locations.

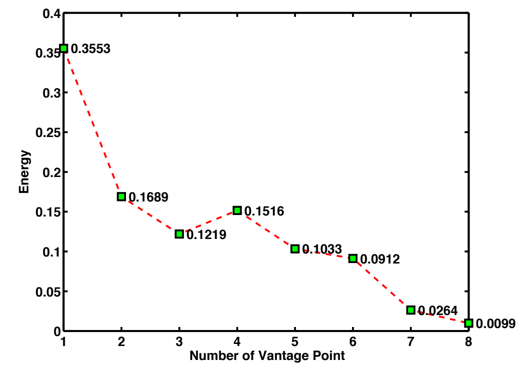

Fig. 4. The performance of the new visibility algorithm for exploring a slightly more complex 3D pipeline configuration. The plot shows the decay of the ***metric*** as additional vantage points are added. Notice that the number of vantage points needed seem to grow linearly with the number of cylinders in the scene.

A couple of remarks about the above algorithm are in order: Whenever a new vantage point is determined, the visibility from that new point is extracted from the point cloud processing, and we thus obtain the function ϕ_{x_c}. As the new vantage point would see a portion of the domain that is previously occluded, the occlusion set $\mathcal{S}$ is then updated by taking the intersection with the set $\{\psi_{x_c} < 0\}$. This operation is easily achieved by the level set approach that we adopted in the last line of the algorithm description. Furthermore, in order to maintain good resolution of the shadow boundary on the grid level as well as maintain the accuracy of the numerical approximation of its normal, it is necessary to reshape ψ into a signed distance function, keeping the set ψ unchanged. This is done in the level set literature by a redistancing procedure. See [23] and [3] for example.

The surface integral of the visibility metric E will be computed by summing over the grid nodes on which $\omega_{\mathcal{O}}\, \delta_\epsilon(\psi)$ does not vanish. $\omega_{\mathcal{O}}\delta_\epsilon(\psi)$ approximates the Dirac δ function supported on the shadow boundary; in the algorithm, H denotes the indicator function of the interval $[0, \infty)$. The approximate Dirac delta function $\delta_\epsilon(\psi)$ is taken to be $(\cos(\psi/\epsilon) + 1)/2\epsilon$ for $|\psi| < \epsilon$ and 0 otherwise. This is a standard approach in the level set method literature which allows approximation of surface integrals *without explicitly extracting the surface*. See [7] for more detail. In our implementation, we take ϵ to be twice the step size of the grid.

Finally, in order to reduce the unnecessary computational complexity, we strategically pick new vantage points only at locations where $\Delta\psi$ is large ($\Delta\psi > M$) — these locations correspond to points that are roughly equidistant to the occlusion set. M is typically chosen to be $\mathcal{O}(1/h)$, where h is the grid cell size.

IV. SURFACE RECONSTRUCTION

Given a point cloud consisting of three-dimensional points we wish to construct a representative surface, which captures with high fidelity the features resident in the original point cloud. For this purpose we have chosen to utilize a modified version of the method described by DeVore et al. [5]. There are several reasons for this choice of reconstruction method, the foremost being its use of the Hausdorff metric, which is defined by

$$\delta_H := max\{\delta(A,B), \delta(B,A)\}, \tag{2}$$

for any two sets $A, B \in \mathbb{R}^3$, where

$$\delta(A,B) := \sup_{a\in A} \inf_{b\in B} |a-b|, \tag{3}$$

and $|\cdot|$ is the standard Euclidean distance. For vehicle navigation as well as line of site problems both [5] and [17] demonstrate that the performance of the Hausdorff metric far exceeds that of other conventional metrics, such as least squares.

The reconstruction method described in [5] assumes that the point cloud is given in its entirety as input. However, in our application we can not make this assumption as the point cloud is generated as the observer moves through the environment. As such we do not know the full range of the point cloud *a priori*. Instead we assume the size and location of the domain of interest relative to the observer's local coordinate system is known. Given this information a transformation can be constructed consisting of a shift and uniform dilation such that the transformed point cloud lies in the unit cube $\mathcal{Q} = [0,1]^3$. With this transformation we can utilize the multiscale decomposition scheme defined in [5] with minimal modification. For convenience the main points of the surface reconstruction method are briefly outlined below.

We define $D' = D \cap \mathcal{Q}$, where D is the input point cloud which has been transformed using the above shift and dilation. The domain $\mathcal{Q}$ is recursively decomposed into dyadic subcubes in order to construct an octree $\mathcal{T}$ whose nodes are decorated with polynomial fits to the local point cloud data.

The construction of $\mathcal{T}$ is performed adaptively from the root node $\mathcal{Q}$ using a processing list, which is initialized as $\{\mathcal{Q}\}$. Given a cube Q from the front of the processing list and local point cloud data $D_Q = Q \cap D'$, a representative planar fit is constructed using standard Principal Component Analysis (PCA). The quality of the PCA plane is measured with respect to the Hausdorff metric using a single global error threshold η. If the computed Hausdorff distance between the PCA plane and D_Q is less than η then Q is marked as a leaf node and added to $\mathcal{T}$. However, if the computed distance is greater than η the node Q is not simply subdivided. Instead the authors make use of the fact that the Hausdorff distance is a two-sided distance and attempt to improve the local fit using one or two axis-aligned bounding boxes. The bounding boxes are used to identify cases where the point cloud is localized in one or two regions of Q. In the single bounding box case, the extents of the local point cloud are used to construct the bounding box and the PCA plane is clipped to this region and the Hausdorff distance updated. If this new distance satisfies the threshold η then Q is marked as a leaf and added to $\mathcal{T}$. If this test fails, then the local data is clustered into two groups and a bounding box is computed for each cluster. Also, each cluster is given its own PCA planar fit and the Hausdorff distances measured. If both planes satisfy the threshold η then Q is marked as a leaf and added to $\mathcal{T}$. Using the above described fitting tests, any cube Q that is a leaf node of $\mathcal{T}$ may have one or two planar fits. In the event that the bounding box tests fail to produce representative planes for the local point cloud then Q is added to $\mathcal{T}$ as an interior node and the dyadic children of Q are added to the end of the processing list.

Several details of the above method are omitted, however once the processing list becomes empty the leaf nodes of the resulting octree $\mathcal{T}$ contain a discontinuous piece-wise linear representation of the point cloud.

A. Dynamic Updates

The processing method defined in [5] does not provide a mechanism for dynamic updates to the point cloud or resulting surface as required by our application due to its assumption that the point cloud is known *a priori*. However, as the representation is based on an octree structure we can easily filter new points into the tree from top to bottom. The use of an octree also allows us to update only those regions of the domain where new data was received.

As the observer moves to each new vantage point in the domain a new point cloud D_i, where i is the index of the current vantage point, is generated by the observer's sensor package. The update of $\mathcal{T}$ proceeds as follows:

```
Set D'_i = D_i ∩ 𝒬
Set L = {𝒬}
while L is not empty do
    Pop Q from the front of L
    if D'_i ∩ Q = ∅ then
        continue
    end if
    Set D_Q = D_Q ∪ (D'_i ∩ Q)
    Compute the fit Ŝ_{D_Q} of Q using D_Q
    Compute the Hausdorff error η̂_Q of Ŝ_{D_Q} and D_Q.
    if η̂_Q > η then
        if Q is a leaf node then
            Expand 𝒯 by generating the children of Q
        end if
        Find the children of Q for which (D'_i ∩ Q) ≠ ∅
        L = L ∪ {children of Q identified above}
```

else if Q is not a leaf **then**
Prune the subtree rooted at Q from $\mathcal{T}$
end if
end while

Note that the above update scheme does not explicitly show the use of bounding boxes or clustering, however this is assumed to be part of the computation of $\hat{S}_{D_Q}$. It is also worth pointing out that the above method naïvely stores all of the data points that have been scanned. In some applications this may be undesirable. This is a current limitation of the implementation as the points are required for a full measurement of the Hausdorff distance. However, to reduce the memory required, we may borrow an idea from the point-based graphics community [25], [13], [26] where the point clouds may be quantized over a sparse uniform grid which has resolution greater than the expected maximum depth of $\mathcal{T}$. The use of such a uniform grid adds additional error to the Hausdorff error commensurate with the grid resolution, but allows the required memory to become fixed rather than potentially infinite as more points are sampled during exploration. While this is a potential improvement, this idea was not utilized in our implementation as we merely sought proof of concept.

B. Visibility from Reconstructed Geometry

The algorithm presented in Section III requires as input a level set function ϕ_{x_c} over the domain which encodes both the occlusion set and the distance to known obstacles. The distance to the set of known obstacles is computed simply as the $max\{0, d - \eta\}$, where d is the unsigned Euclidean distance to the piece-wise linear fits constructed in the previous section. This value is chosen in place of the unsigned distance as the fits generated in the previous section do not exactly match the true surface, instead they are within η of the true surface. The occlusion set can be generated using simple ray casting, whereby all points in the domain are classified as visible or occluded by comparing there distance along a ray to the observer with the distance of the first occluder on the same ray. As an improvement we utilize the causality relation of visibility, which allows us to reduce the number of independent distance computations performed over the domain by recognizing that all points along the ray beyond the first occluder are not visible. Additional computational improvements may be obtained utilizing the multi-resolution ray tracing method proposed by Tsai, et al. [22]. For our implementation we encode the visibility of each point in the domain using $+1$ (visible) and -1 (occluded). Also, our implementation provides the option to replace the above rays with cones, whose apex lies at the observer location and central axis coincides with the original ray. In this context the cone provides compensation for cracks that commonly occur along boundaries between the piece-wise linear fits generated above. The level set function ϕ_{x_c} is then computed as the multiplication of the unsigned distance field and the binary (± 1) visibility. The resulting function is positive in all visible regions, given the current observer, and negative in the occlusion set, including the occluders.

V. SIMULATED ENVIRONMENT

In order to facilitate testing of the above methods in a variety of real-world scenarios we utilized the USC Simulator [1], [4]. This tool allows for basic sensor packages, including cameras and LiDAR, to be placed in virtual environments and controlled via Matlab.

The USC Simulator provides support for LiDAR-like sensors through idealized range detection objects. Each range detection object is modeled as a set of rays emanating from a single point. The rays are generated according to a few basic parameters, the horizontal and vertical field of view and the angular step sizes. These value identify a region of the unit sphere centered at the object's location that will be scanned. This model stems from the geometry of typical line scanning devices which utilize a spinning mirror to direct an outgoing laser beam and receiving optics with sufficient field of view to cover the desired region of space both vertically and horizontally.

It should be noted that the point clouds produced by this package are effectively computed range values. Other packages, such as DIRSIG [14], provide genuine scattering simulations that take into account the spectral properties of the surfaces in the virtual environment, and thus are capable of providing much more realistic point clouds. However, for our purposes we found this level of simulation to be unnecessary.

VI. RESULTS

Using the above described simulation environment we maneuvered an observer through a virtual Military Operations in Urban Terrain (MOUT) site [4]. The MOUT site was generated using real data from the McKenna Urban Training Site to serve as ground truth for the testing and validation of algorithms. The observer was equipped with a virtual LiDAR-like sensor with full visibility on the sphere. The parameters of the virtual sensor employed in our tests were assigned values similar to those on found on a SICK LMS200 [15]. The virtual sensor was assigned a maximum range reliable range of 80m, an angular resolution of $0.5°$ and a field of view of $179.5°$. The virtual sensor was oriented such that each scan line ran vertically, and the sensor was allowed to rotate about a vertical axis, performing line scans every $0.5°$.

For all explicit surface reconstructions we required a minimum number of five points in any octree node for a surface fit to be generated. Table I gives the octree and explicit surface statistics for two sample domains. Additionally, the global Hausdorff error is given for the reconstructions. While the local Hausdorff error is within the requested η, the global error may be larger. This is due to noise, under sampling, and other factors. In the patio scene the points responsible for the large Hausdorff error occur inside one of the trees. Also, the points responsible for the large error in the full reconstruction of the church scene sit at the boundaries of the scene where the surfaces were strongly under-sampled.

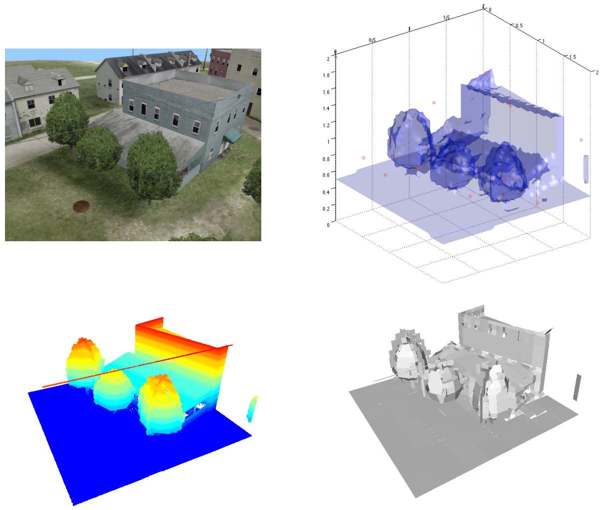

Fig. 5. Top row: The left image shows a patio area with tree cover from the virtual MOUT site. The right image shows the integrated visibility volume generated from 16 vantage points (shown as red circles). Bottom row: The left image the cumulative set of scanned points, 1,769,061 from 16 scans, colored by height. The right image shows a coarse reconstruction using piece-wise linear fits. The domain for this test was limited to a 20m region around the patio.

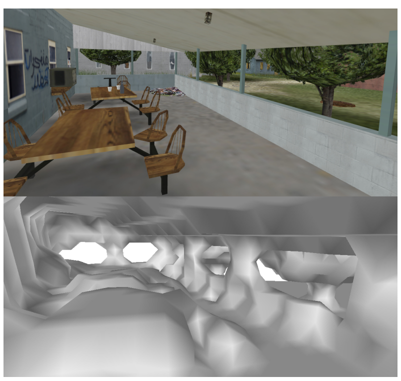

Fig. 6. The top image shows the original scene from the simulator. The bottom image shows a coarse reconstruction of the geometry from the visibility. Even at this resolution the visibility algorithm is able to discern tree trunks, columns and the tables inside the patio. Note the camera locations used to generate these images are approximate.

The visibility function ϕ_{x_c} was defined as a $60 \times 60 \times 60$ grid with values generated as described in Section IV-B using cones with opening angles of 1°.

VII. CONCLUSION

In this article we have developed a visibility-based method for autonomously exploring an unknown environment utilizing explicit and implicit representations. The method produces a visibility volume function from which vantage points are generated at the function's maxima. Additionally, we demonstrated the use of an explicit surface reconstruction tool for the generation of local visibility volumes.

VIII. ACKNOWLEDGEMENT

The authors would like to thank Ronald DeVore, Stanley Osher, Guergana Petrova, Robert Sharpley, Peter Binev and Billy Clack for their input, contributions, and support. Additionally, the authors would like to thank the Army Research Office for the opportunity to present this work.

TABLE I
EXPLICIT SURFACE RECONSTRUCTION STATISTICS

Domain	η	Tree Depth	# Nodes in Tree	# Leaf Nodes	# Fits	Hausdorff (global)
Patio	0.02	7	2721	2381	1370	0.0344
Church	0.004	10	83769	73298	11099	0.1357

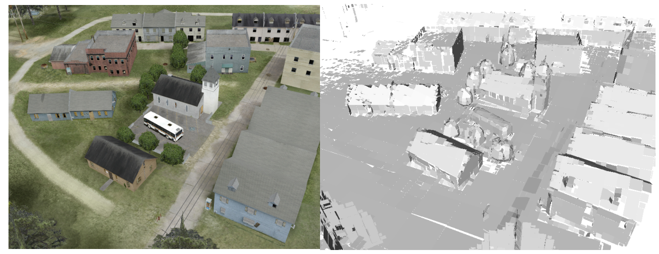

Fig. 7. A larger region containing several buildings. In this test the domain for the visibility was limited to a 50m region around the church and nearby buildings. However, as a post-process we reconstructed the surfaces using the full set of 1,896,786 points generated from 11 vantage locations, including the scanned regions outside of the visibility domain. Table I gives the tree statistics for the full region around the church for this post-process reconstruction.

REFERENCES

[1] R. Baraniuk, S. Kulkarni, A. Kurdila, S. Osher, G. Petrova, R. Sharpley, R. Tsai, and H. Zhao, "Model classes, approximation, and metrics for dynamic processing of urban terrain data," Award No. W911NF-07-1-0185, U.S. Army Research Office, Research Triangle Park, NC, MURI Annual Report, September 2010.

[2] P. Cheeseman, M. Self, and R. Smith, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfon, Eds. Springer-Verlag, 1990, pp. 167–193.

[3] L.-T. Cheng and Y.-H. Tsai, "Redistancing by flow of time dependent eikonal equation," *J. Comput. Phys.*, vol. 227, no. 8, pp. 4002–4017, 2008.

[4] T. DeMarse, R. DeVore, W. Ditto, M. Furman, P. Ifju, T. Kanade, P. Khargonekar, A. Kurdila, R. Lind, M. Nechyba, and R. Sharpley, "Vision-based control of agile, autonomous micro air vehicles and small uavs in urban environments," Award No. F49620-03-1-0170, U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, MURI Final Report, March 2007.

[5] R. DeVore, G. Petrova, M. Hielsberg, L. Owens, B. Clack, and A. Sood, "Processing terrain point cloud data," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 1–31, 2013. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/110856009

[6] H. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Journal of Robotics and Automation*, vol. 4, no. 1, pp. 23–31, 1988.

[7] B. Engquist, A.-K. Tornberg, and R. Tsai, "Discretization of Dirac delta functions in level set methods," *J. Comput. Phys.*, vol. 207, no. 1, pp. 28–51, 2005.

[8] D. Koller, M. Montemerlo, S. Thrun, and B. Wegbreit, "Fast-slam: A factored solution to the simultaneous localization and mapping problem," in *AAAI National Conference on Artificial Intelligence*, 2002, pp. 593–598.

[9] Y. Landa and R. Tsai, "Visibility of point clouds and exploratory path planning in unknown environments," *Comm. Math. Sci.*, vol. 6, no. 4, 2008.

[10] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[11] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localisation for an autonomous mobile robot," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1442–1447, 1991.

[12] R. Rockafeller and R. J.-B. Wets, *Variational Analysis*. Springer-Verlag, 2009.

[13] J. Rossignac and P. Borrel, "Multi-resolution 3d approximations for rendering complex scenes," in *Modeling in Computer Graphics*, ser. IFIP Series on Computer Graphics, B. Falcidieno and T. Kunii, Eds. Springer Berlin Heidelberg, 1993, pp. 455–465.

[14] J. R. Schott, S. D. Brown, R. V. Raqueno, H. N. Gross, and G. Robinson, "An advanced synthetic image generation model and its application to multi/hyperspectral algorithm development," *Canadian Journal of Remote Sensing*, vol. 25, no. 2, june 1999.

[15] *Technical Description Laser Measurement Systems LMS200 to LMS291*, SICK AG, Waldkirch, Germany, 2006.

[16] R. Smith and P. Cheeseman, "On the representation of spatial uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[17] A. Thies, B. Philips, P. Binev, R. DeVore, M. Hielsberg, L. Johnson, B. Karaivanov, B. Lane, and R. Sharpley, "Smooth, piecewise-polynomial terrain representation using nontraditional metrics," Schafer Corporation Contract No. W911NF-04-C-0060, U.S. Army Research Office," STTR Final Report, March 2005.

[18] B. Tovar, S. LaValle, and R. Murrieta, "Optimal navigation and object finding without geometric maps or localization," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, September 2003, pp. 464–470.

[19] B. Tovar, R. Murrieta-Cid, and S. M. LaValle, "Distance-optimal navigation in an unknown environment without sensing distances," *Trans. Rob.*, vol. 23, no. 3, pp. 506–518, June 2007.

[20] B. Tovar, S. M. LaValle, and R. Murrieta-Cid, "Locally-optimal navigation in multiply-connected environments without geometric maps," in *IROS*, 2003, pp. 3491–3497.

[21] B. Tovar, L. Guilamo, and S. M. Lavalle, "Gap navigation trees: Minimal representation for visibility-based tasks," in *In Proc. Workshop on the Algorithmic Foundations of Robotics*, 2004, pp. 11–26.

[22] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro, "Visibility and its dynamics in a PDE based implicit framework," *J. Comput. Phys.*, vol. 199, no. 1, pp. 260–290, 2004.

[23] Y.-H. R. Tsai and S. Osher, "Total variation and level set methods in image science," *Acta Numer.*, vol. 14, pp. 509–573, 2005.

[24] L. Valente, Y.-H. Tsai, and S. Soatto, "Information-seeking control under visibility-based uncertainty," *Journal of Mathematical Imaging and Vision*, pp. 1–20, 2013.

[25] M. Wand, A. Berner, M. Bokeloh, A. Fleck, M. Hoffmann, P. Jenke, B. Maier, D. Staneker, and A. Schilling, "Interactive editing of large point clouds," *Symposium on Point Based Graphics*, vol. 3, pp. 37–46, 2007.

[26] M. Wand, M. Fischer, I. Peter, F. Meyer auf der Heide, and W. Strasser, "The randomized z-buffer algorithm: interactive rendering of highly complex scenes," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 361–370.

[27] D. Wolf, A. Howard, and G. S. Sukhatme, "Towards geometric 3d mapping of outdoor environments using mobile robots," in *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1258–1263.