# Automated identification of voids in three-dimensional point clouds

Katie N. Salvaggio and Carl Salvaggio

Rochester Institute of Technology, 56 Lomb Memorial Drive, Rochester, NY, USA

## ABSTRACT

In the construction of three-dimensional (3D) point clouds from multi-view aerial imagery, voids in the point cloud often exist where multiple views of the area were not obtained during collection. A method is presented for identifying these voids. In this work, point clouds are derived from oblique aerial imagery using multi-view techniques from the photogrammetry and computer vision communities. A voxel-based approach is used to partition the 3D space and each voxel is classified as containing or not containing derived points. Using the imagery and the position of the camera, it is possible to analyze what the cameras can and cannot see, thereby making it possible to label the voxels as occupied, free, and non-classified spaces. Voids in the data will manifest themselves in the non-classified voxels. This method has been tested on high-frame-rate oblique aerial imagery captured over Rochester, NY as well as synthetic data sets. Also presented is a unique synthetic dataset for 3D reconstruction. The data set, created with the Rochester Institute of Technology's Digital Imaging and Remote Sensing Image Generation (DIRSIG) software, provides high-fidelity radiometric data in addition to known 3D locations and surface normals for each pixel location in an image scene. This data set is available to the community for use in their related research.

**Keywords:** three-dimensional point clouds, voxel, computer vision, voids, holes, occlusion, multi-view imagery

## 1. INTRODUCTION

Technological advancements in modern computing have significantly expanded realms of research to encompass areas previously deemed too computationally intensive. One such area that has seen rapid developments in the last decade is the automated generation of three-dimensional (3D) models derived from imagery. Aerial imagery with significant overlap has been used for decades in the photogrammetry community to develop topographic maps using stereo techniques, and more recently the Computer Vision community has developed an interest in structure from motion (SfM) for the purposes of phototourism.[1]

The objective of a scene reconstruction methodology is to extract as much 3D structure from the imagery as possible in order to obtain a point cloud, however there seems to be a lack of focus on the outcome and the point cloud itself. Often voids exist in point clouds where multiple views of an area were not included in the input imagery. Voids also exist because an obstruction was present, homogeneous surfaces resulted in poor image correspondences, or the baseline between images was not sufficient and resulted in poor triangulation. This work presents a method for identifying voids in point clouds using a voxel based approach. The positions of the cameras are used in conjunction with the point cloud data to analyze what the cameras can and cannot see through the use of ray tracing, thereby making it possible to label the voxels as occupied, free, and non-classified. Voids in the data are then easily identifiable as they will manifest themselves in non-classified voxels.

This paper is organized as follows: Section 2 provides a discussion of previous work, including the work flow used to generate point clouds from aerial imagery. A more in-depth description of the problem being addressed is discussed in Sec. 3 and the approach is described in Sec 4. Section 5 presents a unique synthetic image dataset for 3D reconstruction. Preliminary results are shown in Sec. 6, and conclusions and future work are presented in Sec. 7.

Further author information: (Send correspondence to K. N. Salvaggio)
E-mail: kns4906@rit.edu

Earth Observing Systems XVIII, edited by James J. Butler, Xiaoxiong Xiong, Xingfa Gu, Proc. of SPIE
Vol. 8866, 88660H · © 2013 SPIE · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2023740

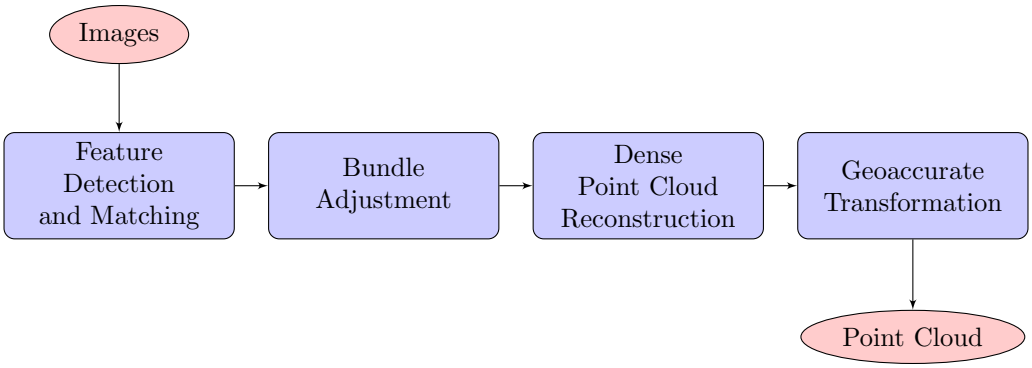



Figure 1. High level diagram of the work flow used to generate dense geoaccurate 3D reconstructions of a scene from 2D imagery.

## 2. PREVIOUS WORK

In general, there are three steps in the pipeline that leads to automated 3D reconstruction: (1) a feature detection and matching algorithm is utilized to provide image to image correspondences, (2) Structure from Motion (SfM) algorithms are then used to estimate the pose using the correspondences, and (3) Multi-View Stereo (MVS) algorithms use the pose estimation and images to produce dense 3D reconstructions. A high level diagram of the work flow is shown in Fig. 1, including a post processing transformation used to place the point cloud in the desired coordinate system.[2]

The initial stages of any reconstruction is feature extraction and matching, such that the resulting image correspondences can be used to estimate the camera poses and a preliminary 3D scene structure. One of the most common feature descriptors and the one employed in this work flow is the scale invariant feature transform (SIFT).[3] The SIFT features have been shown to be invariant to image rotation and scale in addition to being robust to a range of affine transforms, noise, and some change in illumination. Once the feature descriptors are calculated, a set of descriptor matches across all possible image pair combinations is computed. A wide variety of techniques are available to do this, from brute force Euclidean matching techniques to model fitting algorithms that employ random sample consensus (RANSAC).[4]

The next step in the chain is to estimate the camera pose using the image-to-image correspondences found previously. Traditionally, this is accomplished with a bundle adjustment. Bundle adjustment refers to the large non-linear least squares problem that is solved in a feature based SfM algorithm. The term is utilized both in the Computer Vision community as well as the photogrammetry community, where it was conceived in the 1950s.[5] It is an optimization problem on the 3D structure of a scene and viewing parameters, such as the camera position and calibration, in order to obtain a reconstruction of the scene which is optimal. This is achieved by minimizing the reprojection error between the observed and predicted image points, expressed as the sum of squares of a large number of nonlinear real-valued functions.[6] This minimization can be achieved using nonlinear least-squares algorithms. Bundler,[7] written by Noah Snavely, is built upon a software package available for generic sparse bundle adjustment (SBA), written by Manolis Lourakis.[6] The work flow used here leverages Bundler to obtain the refined camera projection matrices, equivalent to the photogrammetric ground-to-image function.

As the number of images to be used in a 3D reconstruction grows, it is no longer feasible to use all available images simultaneously to construct a model and it becomes necessary to use a cluster-based method to achieve scalability in the last stage of the pipeline, dense reconstruction. View selection can be used to decompose a set of images into clusters with small overlap, a MVS algorithm can then be used to reconstruct dense 3D points and the resulting solutions can be merged into a single model.[8] The algorithm that is used in the work flow presented here was written by Furukawa, and is known as Cluster-based Multi-View Stereo (CMVS).[8] Once the images have been divided into clusters using CMVS, the Patch-based Multi-View Stereo[9] algorithm is used on each cluster to generate a dense point cloud. Patch-based MVS methods are only one type of MVS algorithm, however they are simple and effective and suffice for a point based rendering such as a point cloud.

Figure 2. 3D Reconstruction of the Rochester, NY skyline.

Figure 3. Rochester, NY skyline.[10]

At this point, a dense point cloud representation of the 3D structure in the images is achieved. It is important to note that this method will reconstruct the scene up to a projective ambiguity, meaning that the scene reconstruction is determined to within a projective transformation of the world coordinate system (WCS). While a vast majority of SfM problems addressed attempt to solve for an unknown number of uncalibrated cameras taking images at unknown locations, the problem addressed here differs in that there is information available about both the camera and the collection geometry. In the applications discussed here, it is assumed that imagery has been captured on an airborne platform that has a global positioning system (GPS) and inertial navigation system (INS) such that the position and orientation of the camera is readily available in the image metadata. In such a case, it is possible to estimate the mapping between the arbitrary coordinate system of the image-based reconstruction and the desired Earth-based coordinate system.[2]

The work flow presented above was applied to imagery from the Exelis Wide-Area Motion Imagery (WAMI) system of downtown Rochester, NY. A reconstruction of the city skyline is shown in Fig. 2 compared to an image of the skyline shown in Fig. 3.

## 3. THE PROBLEM

While the results of the work flow shown in Sec. 2 are promising, they are not without imperfection. Currently there is no metric available in the literature for evaluating the results of such a work flow. Often point density is used as a method of comparison between point clouds, however consider the instance of trying to recreate a simple rectangular face of a building. In such a case, the density of the point cloud is not as important as the accuracy of the corner points since the face could be reconstructed in theory with just the 4 corner points. At this point, the only way to compare point clouds is with the total number of points and a visual evaluation as a point cloud metric seems to be application driven.

Regardless of the lack of metric, visual inspection of the point clouds reveals areas of missing information and voids. A view of the previous point cloud in which voids are readily apparent is shown in Fig. 4. There are small holes visible in portions of building walls, but a human observer is also able to discern that there are some buildings that are missing walls entirely. Some voids are a result of texturally flat areas that failed to generate features in the initial stages and as a result were not well reconstructed, despite the fact that they appeared in multiple images. However some of the missing information, such as the lack of three of the four walls of a building, is a result of lack of coverage in imagery where multiple views of the area were not obtained or there was a constant obscuration and the area was not visible in the imagery.

Automated algorithms designed to fill holes on surfaces reconstructed from 3D point clouds do exist, such as those presented by Wang[11] and Davis.[12] Though these algorithms can be used to successfully fill in some voids in the point clouds presented here, they are designed for use when most of the surfaces have been observed and are present in the point cloud. These algorithms are not designed to fill in large gaps in the point cloud, nor would they be successful in doing so accurately. These issues are the focus of the work presented here.

While the 3D structure that is represented by the point cloud provides a good starting place, it is not representative of all of the information contained in the data. In addition to the 3D locations of structure

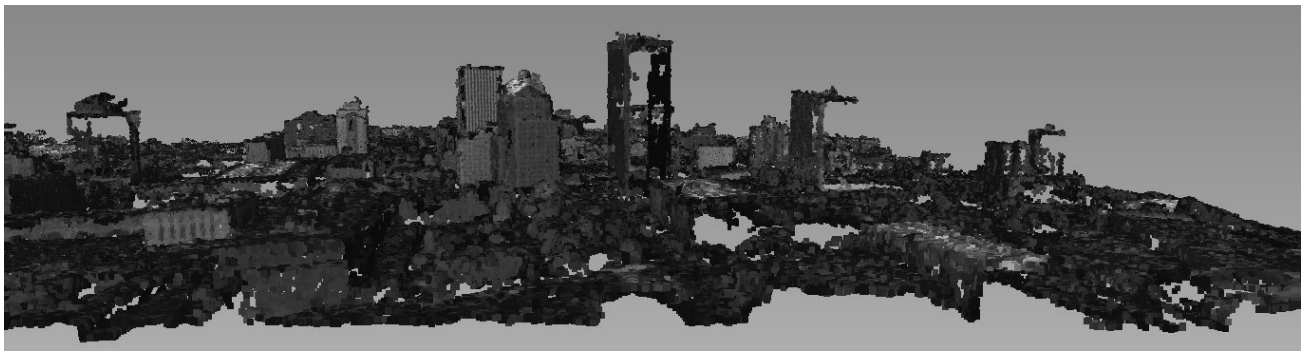


Figure 4. Side view of a 3D point cloud where voids and missing information is apparent.

provided by the point cloud, the PMVS algorithm also provides information about point visibility and thus it becomes possible to analyze what each camera could and could not see.

## 4. APPROACH

The approach described here was based on the idea of free space. If a camera is able to image a point, then there must be free space between the two. This concept can be extended to the 3D reconstruction such that if a camera was used to reconstruct a point, then the ray from that camera to the point must be unobstructed.

While the point cloud provides a simple representation of the original structure, it does not seem adequate to represent the concept of free space. Another representation to consider is a voxel representation, whereby small volumetric elements on a regular grid are used to represent the 3D volume in which the point cloud is contained. Voxels are frequently used in the visualization of data in the medical and scientific communities. Voxels are also used in the multi-image 3D reconstruction problem.[13] In this case, voxels that contain reconstructed 3D point would be classified as occupied voxels, and voxels that lie on the ray between the camera and the point would be classified as free voxels; all other voxels are then non-classified as they have yet to be identified as occupied or free. A 2-dimensional visualization of occupied, free, and non-classified voxels constructed from a simple scene is shown in Fig. 5.

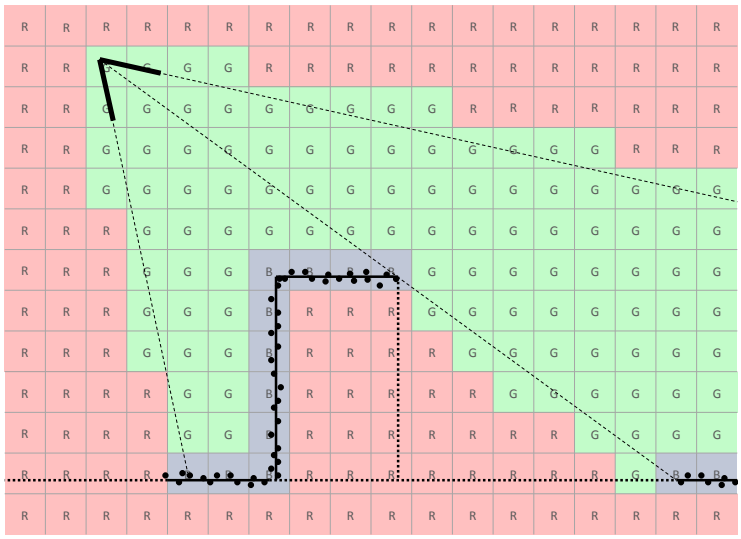


Figure 5. A 2-dimensional visualization of occupied, free, and non-classified voxels. Voxels that contain black points are occupied and denoted with blue (B). The camera is located in the top left and free voxels that rays pass through are denoted with green (G). Non-classified voxels are denoted with red (R).

The point cloud data is used to initialize the voxel space such that the entirety of the point cloud structure and the camera centers are contained in the space. Most of the voxel spaces showcased here were generated with cubic meter voxels. There are trade offs to be considered between the size of the voxel, the processing time, the amount of data to store in memory, and the information represented. For each point, the voxel containing the point is identified and marked as occupied. Following this, a fast voxel traversal algorithm[14] is used for ray tracing to identify free voxels. Any voxel not identified as occupied or free is regarded as non-classified.

This ternary system is sufficient for representing the free, occupied, and non-classified voxels, however a voxel is not infinitesimally small. It is possible that the finite space occupied by a voxel contains multiple points, or that multiple rays passed through, or that it contains both points and passable rays. As such, this algorithm has been implemented in such a way that the voxels are initialized to contain a counter with a starting value of zero. When a point is added to the voxel, the counter is incremented and when a ray passes through the voxel, the counter is decremented. Free voxels are then expressed with negative values, where a more negative value indicates that the voxel has a higher probability of being free space. Similarly, occupied voxels are expressed with positive values, where a greater positive value indicates a higher probability of being an occupied voxel. Non-classified voxels will still contain zero values. Voids and areas of missing data will manifest themselves in the non-classified voxels.

It is possible that the number of points in a voxel will equal the number of rays, in this case the counter will equalize to zero. Such a voxel has an equal probability of being either free or occupied, and therefore identifying the voxel as still being non-classified is not necessarily erroneous. However, due to the fact that a single ray will pass through numerous voxels, a voxel is more likely to be decremented than incremented. Preliminary results lacked occupied voxels, despite using a densely populated point cloud. In order to reduce the likelihood that occupied voxels would be decremented so many times so as to redefine them as a free voxel, a multiplicative factor was added to the counter increment. Thus, voxels containing points are incremented by larger amounts in order to make them less likely to become free voxels, but not by enough to force them to always be occupied in case of the presence of noise.

## 5. DATA

Automated scene reconstruction from imagery is an inherently under-determined problem and there are a lack of benchmark datasets available to quantitatively compare the performance and output of such algorithms. The multi-view Middlebury dataset[15] contains calibrated imagery and corresponding mesh models for simple figurines, designed specifically for evaluating the performance of multi-view stereo algorithms. Though useful, the aerial application of these algorithms is becoming more commonplace and aerial photography can present unique problems of its own, thus it was determined that the community could benefit from another dataset.

The Digital Imaging and Remote Sensing Image Generation (DIRSIG) tool was developed at the Rochester Institute of Technology (RIT) by the Digital Imaging and Remote Sensing (DIRS) laboratory. DIRSIG is a synthetic image generation application designed to produce broad-band, multi-spectral, and hyper-spectral simulated imagery in the visible through thermal infrared regions of the spectrum, with additional capability to produce polarimetric, RADAR and LiDAR imagery.[16] It is a ray tracing model based on first-principles physics sub-models, such as bi-directional reflectance distribution function (BRDF) predictions, sensor models, and atmospheric models. The modeled components are combined and integrated radiance images can be produced for an arbitrary number of band passes.

The data set was simulated with a capture date and time of August 12, 2010, 18:00.00 GMT. Megascene 1, a synthetic DIRSIG scene modeled after an area to the north east of RIT, was used.[17] A larger multi-story building was added to the scene to provide additional height. The imaging system was modeled as a simple multi-band framing array featuring an RGB focal plane, with each band featuring a simple Gaussian spectral radiance response, and a panchromatic band with the spectral radiance response of WorldView-2. The focal length was 125.09mm. The detector featured a $1200 \times 800$ array, with each pixel being square and $32\mu$m in size. The simulated flight path of the sensor was circular with a fixed stare point in the central portion of the scene at an altitude of approximately 800m above ground and a nominal declination angle of 40 degrees. The sampling rate for the data set was 2Hz and images were captured for 210s, resulting in a 420 image data set. Sample images are shown in Fig.6.

(a) RGB (b) Panchromatic

Figure 6. Sample images from the DIRSIG synthetic dataset.

In addition, each radiance image is accompanied by a truth image produced by DIRSIG. Adaptive sampling was used to ensure radiometric fidelity, however it was determined that an average (X,Y,Z) over the rays was not advantageous for this application. Consider casting rays for a pixel that falls on the edge of a structure as is shown in Fig. 7; it is possible that the pixel does not line up exactly with the edge of the structure and therefore some rays may hit the structure and some may just pass it. The average of these rays will result in a point that does not lie on either surface that was intersected by the rays, therefore adding spurious points to the truth point cloud. To mitigate this, a minimum and maximum range value is reported for each hit value and coordinate such that the points in the point cloud and their respective normals correspond to actual points in the scene.

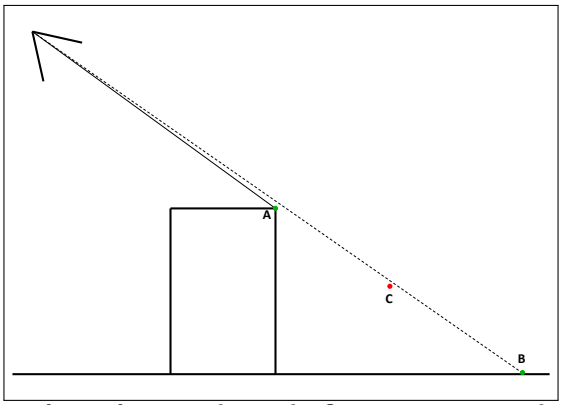
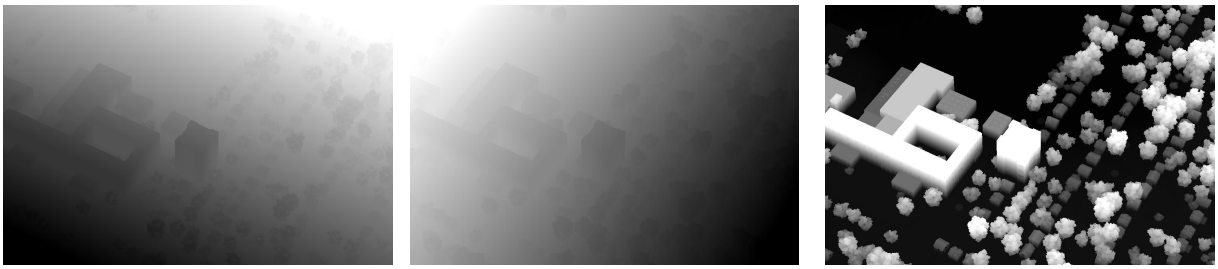



Figure 7. Illustration of the casting of rays for a single pixel. One ray intersects the building at point A and one ray intersects the ground at point B. The average of the two rays, point C, is not on any structure in the scene and therefore is not valid truth data. Note that adaptive sampling in DIRSIG uses more rays, the two ray case was just drawn for illustrative purposes.

For this particular data set, information such as the minimum range, maximum range, (X,Y,Z) hit coordinates for the minimum and maximum range, (X,Y,Z) normal components for the minimum and maximum range and material IDs for the minimum and maximum range are available. Note that the minimum and maximum range will be the same for rigid solid objects such as buildings, but will differ on the edges of buildings and in areas with dense vegetation due to the adaptive sampling methods employed in DIRSIG. Sample images of the hit coordinates are shown in Fig. 8, and a sample 3D point cloud made from the hit coordinates is shown in Fig. 9.

This data set is unique in that it provides a 3D location and normal vector for every pixel in every image and



(a) X hit (min range) (b) Y hit (min range) (c) Z hit (min range)

Figure 8. DIRSIG truth data showing the (X,Y,Z) hit coordinates for the minimum range in a single scene.

Figure 9. Sample point cloud made using the (X,Y,Z) hit coordinates from a single image frame.

camera location and pointing information is known. The full data set is available for download at `dirsapps.cis.rit.edu/3d-dirsig-truth/`.

## 6. RESULTS

The algorithm outlined in Sec. 4 was applied first to the truth data from the DIRSIG data set. The truth data can be used to generate a point cloud for each image such that every pixel in the image has a corresponding point in the real world; an example of this was shown in Fig. 9. This provided a clean set of initial data to test, void of any errors that may have been introduced by the work flow. Because of the use of the truth data, it is possible to study a point cloud derived from a single image and to study combinations of vantage points that may not necessarily produce results through the work flow due to lack of correspondences.

Initial results are shown in Fig. 11, where horizontal slices of the voxel spaces have been taken and are being displayed as a single frame, as depicted by the illustrations at the top. In each of the images, a mid-tone gray represents a zero value for a non-classified voxel, brighter values represent the positive valued occupied voxels, and darker values represent the negative valued free voxels.

Part of the ground plane is shown in column (a) of Fig. 11; the ground plane is slightly slanted. As a result, part of the ground plane is shown with the white occupied voxels, part of the ground plane is below this level of the voxel space and therefore there are free voxels above it. Note also that bright voxels appear on the edges of the non-classified voxels that identify objects. In the single camera case, these bright voxels only appear on the southern edge of objects because that is the side that the camera saw; buildings also appear to be misshapen due to shadows that extend on the northern side of the objects. As another view is added, more object edges appear, and the shapes of objects begin to rectify themselves. It is also apparent that as additional frames are added, there is more overlap in the free space. Free space seen by 4 cameras appears to be darker (i.e. more negative) than that seen by fewer cameras; indicating that as more cameras are added, the confidence that free space is indeed free space grows.

Another point of interest in the preliminary results is the shadows cast by the buildings. As the Z dimension increases in the horizontal profile, the buildings appear to shrink in one dimension when there is only a single

Figure 10. Nadir view of Megascene for reference with the results.

camera present. This is shown in more detail in Fig. 12. In the two and four image cases, the shadows are still visible near the buildings, but they are now considered free voxels because at least one camera has seen into that space. They are less negative in value than the surrounding free space because the area has not been seen by all of the cameras.

The images in Fig. 11 were made by taking horizontal slices of the voxel spaces and were made for visualization purposes, but they do not necessarily reflect the best representation of the information contained in the entirety of the voxel space. Consider the fact that free voxels are indicative that the camera could "see through" that space, therefore when viewing the voxel space, free voxels should be transparent. Continuing with that idea, it is possible to send rays through the voxel space such that they travel through free voxels and stop when they encounter an occupied or non-classified voxel. This type of visualization is shown in Fig. 13, looking from the top of the voxel spaces for the one, two, and four image cases shown previously. The enhanced views show that as more images are added, the number of non-classified voxels decreases, indicating a reduction in the number and/or size of the voids in the point cloud.

Following the successful results with the truth data, the algorithm was tested on a point cloud derived from synthetic imagery using the work flow presented in Sec. 2. An input set of 42 images, taken at 5 second intervals throughout the dataset, was used in the work flow; 16 of the images were used by the CMVS/PMVS algorithm to produce a final dense point cloud with 86,456 points. The results are shown in Fig. 14. The first thing to note are the voids in the point cloud on the roof of the large school building. While the roof of the building was visible in all of the imagery, it was likely not reconstructed due to poor image correspondences on the texturally flat surface. However, the voids are easily identifiable as non-classified voxels in the voxel space.

While the results are promising, the voxel space looks much noisier from these work flow results. This is likely due in part to the density of the point cloud. The point clouds made from the DIRSIG truth data generate a point per pixel in the image and therefore result in 960,000 points per image used. The point cloud generated using the work flow contained only 86,456 points. As a result, there are fewer occupied voxels. However, the larger voids in the point cloud are still clearly visible in the voxel space.

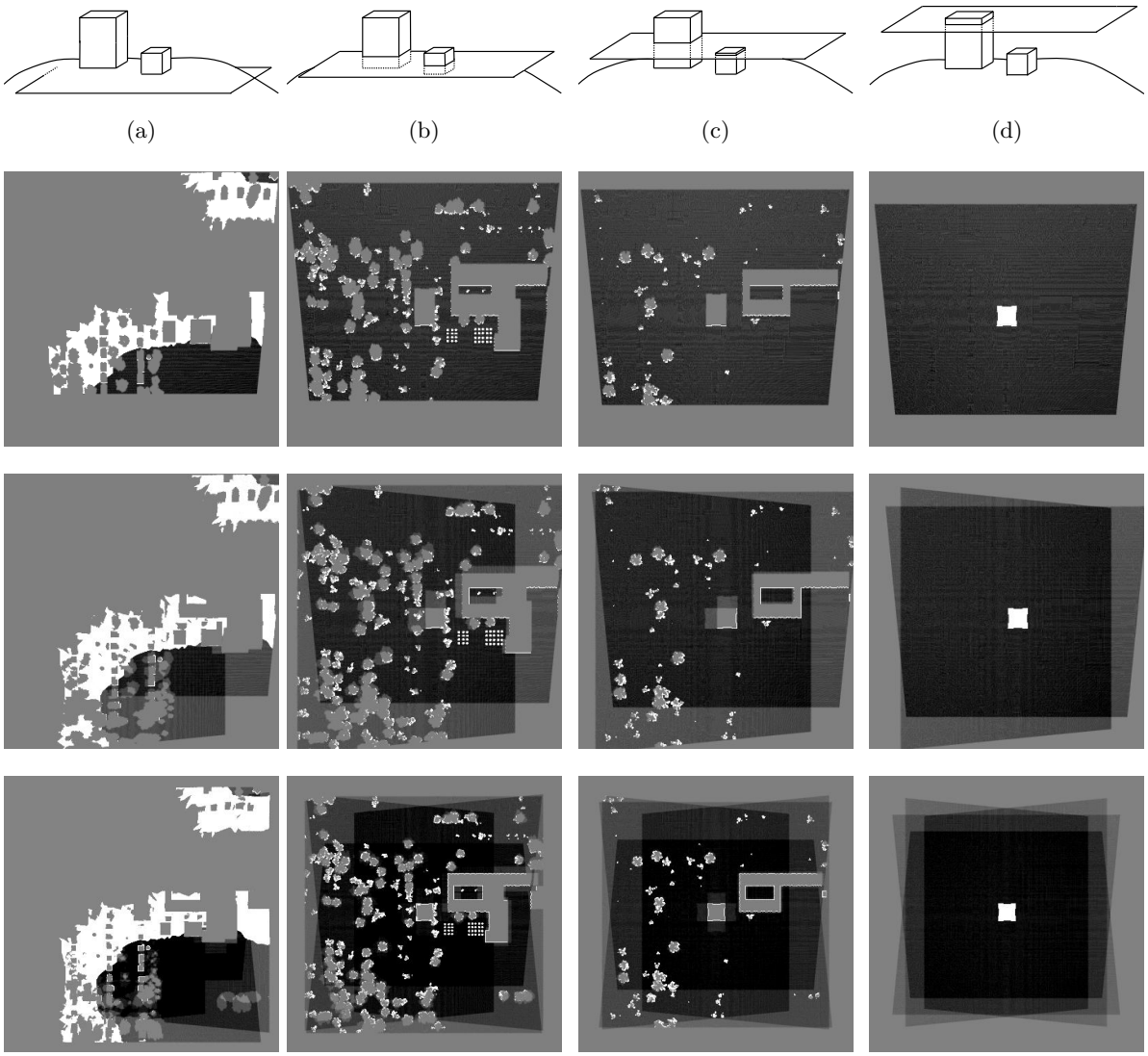



Figure 11. Horizontal slices of the voxel space derived from DIRSIG truth data, where each column is representative of the same slice, moving higher in the Z dimension from left to right as depicted in the illustrations at the top. The first row of images are from the voxel space derived from a single camera, the second row derived from two cameras, and the third row derived from four cameras. A mid-tone gray indicates a zero value or non-classified voxel, brighter pixels indicate an occupied voxel, and darker voxels indicate a free voxel.

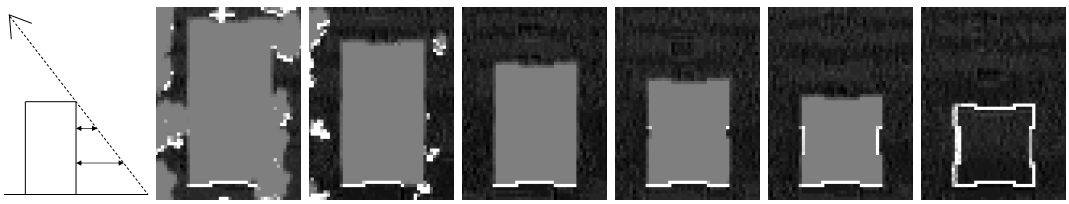
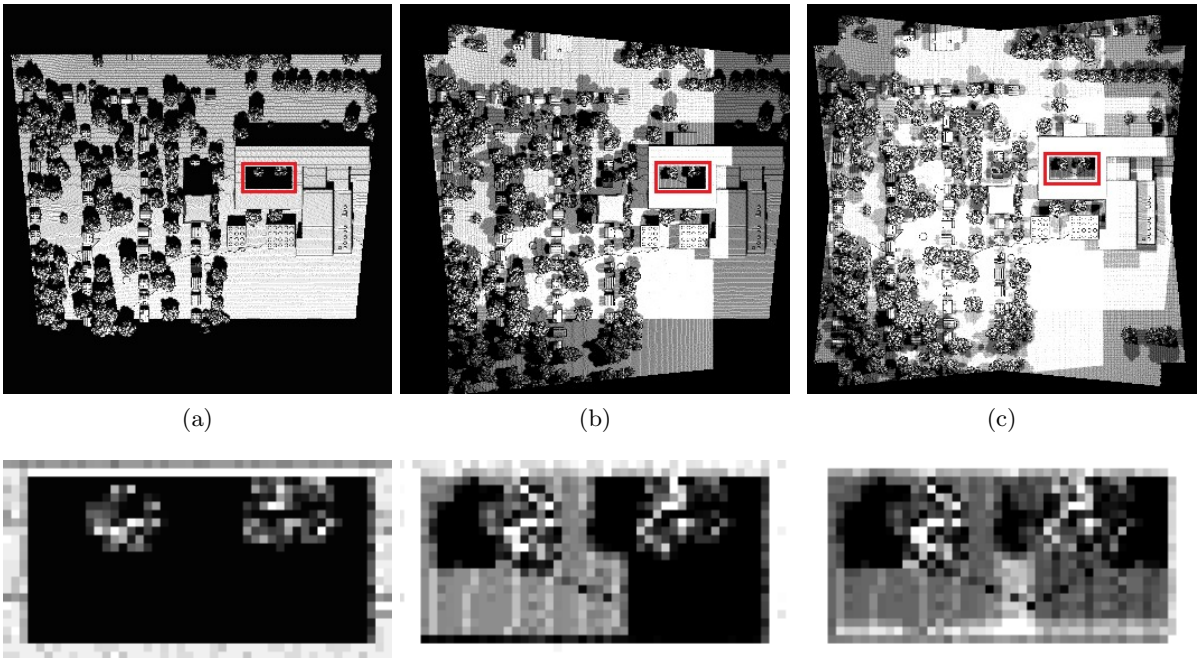
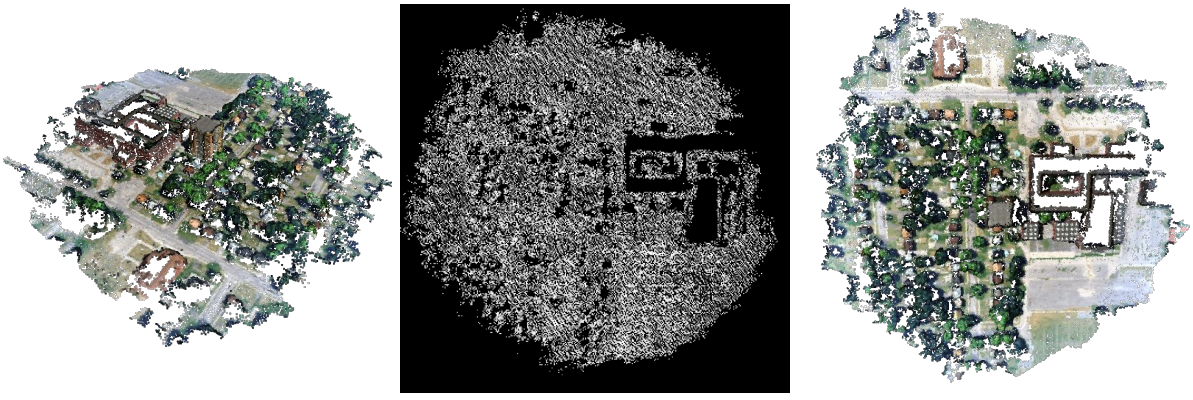


Figure 12. Horizontal slices of the voxel space around the tall building derived using a single DIRSIG truth image, moving higher in the Z dimension from left to right. Note that at higher heights, the building appears to shrink in size in the Y dimension. This is due to the fact that there is only a single camera, and the building casts a shadow that rays cannot bypass, as shown in the illustration on the far left.



(a) (b) (c)

Figure 13. Top-down view of the voxel spaces created using (a) one, (b) two and (c) four images, where free voxels were treated as transparent. The area outlined in red is enlarged below each image to show how the holes begin to fill in with additional views. In this case, black is indicative of voids and any other color is indicative of occupied voxels, where the level of brightness is related to the number of points in the voxel.



(a) (b) (c)

Figure 14. (a) Side view of the point cloud generated with synthetic imagery. (b) Top-down view of the voxel space where free voxels were treated as transparent as in Fig. 13. (c) Top-down view of the point cloud for comparison purposes.

## 7. CONCLUSIONS AND FUTURE WORK

Voids exist in point clouds derived from multi-view imagery. These areas of missing information can be a result of poor image-to-image correspondence in texturally flat areas, obscuration of portions of the scene, or lack of sufficient views of the area. A voxel-based approach has been presented to partition the 3D space in such a way so as to identify these voids. Using the imagery and the position of the camera, voxels are labelled as occupied, free, and non-classified by performing a view analysis through the use of ray tracing. It has been shown that the voids manifest themselves in the non-classified voxels. This method was tested on both truth data, and a point cloud derived from high frame rate synthetic imagery. The truth data was a valuable tool in analyzing initial results, making it possible to perform the voxel analysis using very few images, which would not be possible if only using reconstructions.

A future objective of this work is to use the identified voids to predict potential camera locations that provide optimal views of the void areas. It is believed that inclusion of such imagery in a reconstruction could result in a point cloud with fewer voids due to lack of coverage in the imagery. Being able to predict optimal camera locations in real time opens up the possibility of redirecting an aircraft in flight to obtain imagery from those locations in an attempt to fill in the missing data in the point clouds.

The full synthetic image dataset and truth data used for the work in this paper is available for download at `dirsapps.cis.rit.edu/3d-dirsig-truth/`.

## ACKNOWLEDGMENTS

The authors wish to thank Chris De Angelis, Scott Brown, and Adam Goodenough for lending their DIRSIG expertise in the creation of the synthetic dataset. The authors also wish to thank Exelis Inc. for providing the WAMI data used in conjunction with this project.

## REFERENCES

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R., "Building rome in a day," in [*Computer Vision*], 72–79, IEEE (2009).
2. Walvoord, D. J., Rossi, A. J., Paul, B. D., Brower, B., and Pellechia, M. F., "Geoaccurate three-dimensional reconstruction via image-based geometry," in [*SPIE Defense, Security, and Sensing*], 874706–874706, SPIE (2013).
3. Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International journal of computer vision* **60**(2), 91–110 (2004).
4. Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**(6), 381–395 (1981).
5. Brown, D., "A solution to the general problem of multiple station analytical stereo triangulation," tech. rep., RCA-MTP (1958).
6. Lourakis, M. I. and Argyros, A. A., "SBA: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software* **36**(1), 2 (2009).
7. Snavely, N., Seitz, S. M., and Szeliski, R., "Photo tourism: exploring photo collections in 3d," in [*ACM transactions on graphics*], **25**(3), 835–846, ACM (2006).
8. Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R., "Towards internet-scale multi-view stereo," in [*IEEE Trans. on Pattern Analysis and Machine Intelligence*], **32**(8), 1362–1376 (2010).
9. Furukawa, Y. and Ponce, J., "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **32**(8), 1362–1376 (2010).
10. "Rochester-NY-skyline.jpg." `http://linkagesrochester.org/drupal/node/29`.
11. Wang, J. and Oliveira, M. M., "A hole-filling strategy for reconstruction of smooth surfaces in range images," in [*SIBGRAPI Computer Graphics and Image Processing*], 11–18, IEEE (2003).
12. Davis, J., Marschner, S. R., Garr, M., and Levoy, M., "Filling holes in complex surfaces using volumetric diffusion," in [*3D Data Processing Visualization and Transmission*], 428–441, IEEE (2002).

13. Pollard, T. and Mundy, J. L., "Change detection in a 3-d world," in [*Computer Vision and Pattern Recognition*], 1–6, IEEE (2007).
14. Amanatides, J., Woo, A., et al., "A fast voxel traversal algorithm for ray tracing," in [*Proceedings of EUROGRAPHICS*], **87**, 3–10 (1987).
15. Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R., "A comparison and evaluation of multi-view stereo reconstruction algorithms," in [*Computer vision and pattern recognition*], **1**, 519–528, IEEE (2006).
16. Digital Imaging and Remote Sensing Laboratory, Rochester, NY, *The DIRSIG User's Manual* (2013).
17. Ientilucci, E. J. and Brown, S. D., "Advances in wide-area hyperspectral image simulation," in [*AeroSense 2003*], 110–121, International Society for Optics and Photonics (2003).