Contract No:

This document was prepared in conjunction with work accomplished under Contract No. 89303321CEM000080 with the U.S. Department of Energy (DOE) Office of Environmental Management (EM).

Disclaimer:

Improved Data Analysis Method for Exhaust Tunnel LiDAR Inspection Data

William Willie Wells*, Jean R. Plummer*, Nickolas A. Trisnadi*, Jane M. Carter†, Saiying Z. Bowers†

Savannah River National Laboratory, Battelle Savannah River Alliance FFRDC, Building 781-A, Aiken SC, 29808,
William.Wells@srnl.doe.gov, Jean.Plummer@srnl.doe.gov, Nickolas.Trisnadi@srnl.doe.gov
† Savannah River Nuclear Solutions, Savannah River Nuclear Solutions LLC, Building 704-2H, Aiken SC, 29808,
Jane.Carter@srs.gov, Saiying.Bowers@srs.gov

INTRODUCTION

Based at the Savannah River Site the H-Canyon Exhaust Tunnel (H-CAEX) structure, Fig 1, is visually inspected via a tethered unmanned ground vehicle using cameras. These inspections are to confirm that the as found field condition of H-CAEX tunnel structure can perform its credited safety function and provide assurance that corrective actions can be performed before the safety function is compromised if evidence of degradation is detected. In an effort to obtain qualitative measurements to determine an erosion rate of tunnel surfaces, a Light Detection and Ranging (LiDAR) system is being deployed to inspect a local area referred to as the Pitot Tube location.

Fig. 1. H-Canyon Facility with tunnel locations noted.

Beginning in November 2019 an annual inspection has been performed using a LiDAR system on a pole, Fig 2, in the H-CAEX [1]. The LiDAR system is deployed through a H-CAEX pitot tube. The H-CAEX is an under-ground ventilation tunnel. The H-CAEX pitot tube diameter is 154 mm. Nitric acid vapors, high humidity, radioactivity, contamination, and air flow of approximately 13.411 m/s are the environmental hazards of the H-CAEX. The H-CAEX environment is a Global Positioning System (GPS) and personnel denied environment. These design challenges limited LiDAR sensor selection for the inspection.

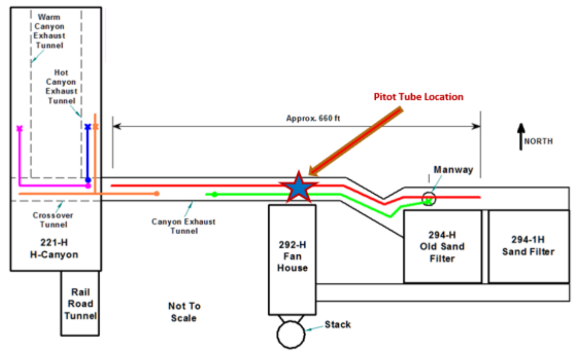

Fig 2. H-CAEX pitot tube location.

BLK360 is a Leica LiDAR system, which includes 3 aligned spaced embedded cameras, an internal Inertial Measurement Unit (IMU), a light source, and a rotating mirror. Four evenly spaced Banner HLS27 LED strip lights placed near the BLK360 provide lighting for the embedded cameras. The BLK360 is designed to use a battery. Two wired Ubiquiti NanoStation LocoM2s are used to relay the wireless signal from the BLK360 to a remote above ground iPad.

By November of this year, data from six inspections using the LiDAR system will be obtained. The first proof of concept inspection was performed in November 2019. A June and a November inspection were performed in 2020; the June inspection was performed to prove repeatability and to perform some minor modifications to the system [2]. A November inspection was performed in 2021. Two future deployments are scheduled for 2022, a July/August inspection and a November inspection. Distorted data was scanned during the November 2021 inspection thus, a July/August inspection is being scheduled to validate corrective actions taken and to acquire data for erosion rate analysis.

Between the third and fourth inspections methods to speed up and automate the data analysis process were explored and evaluated [3]. One method involved upgrading hardware from an Intel Core i9-7980XE 18 core CPU @ 2.6 GHz with 128 GB RAM and two NVIDIA Titan RTX GPUs to an AMD Ryzen ThreadRipper 3990X 64 core CPU @ 2.9 GHz with 256 GB RAM and a NVIDIA GeForce RTX 3090 GPU. The second method involved using MATLAB

(Matrix Laboratory) instead of CloudCompare software to perform data analysis. This paper discusses post-processing improvements realized with MATLAB.

RESULTS

Both MATLAB and CloudCompare software packages include the ability to perform the following steps:

1. Import/Load point cloud
2. Segment and save segmented point clouds
3. Record surface dimensions and point count
4. Calculate and record segment area and point density
5. Create/load artificial surfaces
6. Perform surface to artificial surface distance calculations
7. Create and save histograms of distances
8. Record median and 95 percentile values
9. Create and save images of segmented surfaces

CloudCompare is an open-source project for 3D point cloud processing software. MATLAB is a numeric computing environment with image generating capability.

Before any data analysis can be performed one must load the point cloud into the software environment. CloudCompare execution time when loading data is less than MATLAB; this is the only step from above in which this is true. Execution times for each of the above steps are outlined in TABLE I. All execution times are in seconds except for the final row of data, which is in hours. All execution times apply to the computer with the Intel CPU described previously. MATLAB allows the programmer or analyzer to automate each step in the process. The ability to automate each step significantly reduced step execution time for steps 2, 3, 4, 7, 8, and 9. Software was written in MATLAB that reduced the execution time for a point cloud that is both dense and in excess of a million points from exponentially increasing to linearly increasing with respect to the number of points. With CloudCompare the execution time of cloud-to cloud comparisons scaled exponentially as the number of points increased.

Being able to segment a point cloud at exact mathematical distances from an origin in MATLAB, which is not available in CloudCompare, increased the repeatability and accuracy of deployment-to-deployment comparisons. Each surface is bounded by specific distances in the x direction (defined as an axis parallel to length of the tunnel). Bounds in the x direction were initially designed to limit the total points in the wall surfaces and thus limit the execution time of each comparison. Walls are bounded, separated from ceiling and floor, by z (defined as an axis parallel with the height of the tunnel) distance values.

TABLE I. Timing Analysis Software (seconds)

	CloudCompare	MATLAB
Import/Load	13	250
Segment and save segmented point clouds	21600	1800
Record surface dimensions and point count	3600	60
Calculate and record segment area and point density	900	0.25
Create/load artificial surfaces	54000	5675
Perform surface to artificial surface distance calculations	147600	3600
Create and save histograms of distances	54000	1350
Record median and 95 percentile values	14400	60
Create and save images of segmented surfaces	7200	1800
Sum	303313	14595.25
Hours	84.25361111	4.0542361

Bounded by the height-width plane with respect to an observer facing the surface (North wall or ceiling) artificial surfaces were created with a grid pattern in MATLAB and projected to a proper depth value with respect to the artificial surface (wall in the middle of the tunnel or at the design floor) while with CloudCompare it was more expedient to project the existing points from the reference surfaces to the proper depth of the target artificial plane. The gridded approach increased accuracy of distance measurements. Distance between grid points in all directions (up, down, left, right) is 0.79375 mm (1/32 inch). Artificial surfaces were only created from the first deployment data and were loaded for subsequent deployments.

Both the reference surface and the compared surface are split into smaller balanced point clouds in MATLAB before performing an L2 norm (Euclidean) distance calculation. The split is only performed along the length of the tunnel as opposed to all dimensions in 3-dimensional space. This choice simplifies the shortest distance (nearest neighbor) search. The data structure utilized to contain the computed split points is an array. A binary tree is practical in reducing execution time when an operation such as a search is being performed on a subset of the total data points. Arrays take less execution time to construct than binary trees. A binary tree implementation was tested, and execution times were compared against using an array. The array implementation is faster. The median value is used to recursively split the point cloud into smaller balanced point clouds as in the k-d tree algorithm. Split points are saved in an array as opposed to a tree. Each split fragment is a thin vertical slice of the original point cloud. A value for the maximum point count per split fragment is used to terminate the recursion. The optimal value for the maximum is machine dependent.

By restricting the search area in the reference surface using a 2-dimensional search window the execution time is able to increase linearly with respect to the number of points in the point cloud rather than exponentially. The search window starts as a square with side length 6.35 mm (1/4 inch) with the compared point in the center of the square. If there are no reference points in the search window, the search window side length is expanded by 6.35 mm while keeping the compared point centered until at least one point of the reference point cloud exists within the search window.

Barring a single logical core, in MATLAB all CPU cores are utilized in parallel to perform the minimal Euclidean distance point-to-surface point calculation for all points, one split fragment at a time. The excluded single logical core is available for the system to process interrupts while the computation proceeds. CloudCompare limits thread usage (1 thread per core) to 36 and does not perform a restricted search method as the one described in the previous paragraph. The execution time of this algorithm on the NVIDIA Titan RTX's was longer than the execution time for the same calculation using parallelized CPU cores.

Because CloudCompare could not perform a cloud-to-cloud computation using the entire tunnel wall of the area of interest ([-9.144, 9.144] m from the deployment location) the tunnel was divided into segments. With MATLAB it is possible to abandon the segment approach that was forced upon the data analyst by CloudCompare.

Before any of the data analysis described so far can proceed the data from deployment n, with $n > 1$, is required to be aligned with the first deployment data set. CloudCompare uses a version of the iterative closest point (ICP) algorithm. A version of ICP using the data splitting method described in this paper and singular value decomposition (SVD) for successive rotation and translation estimates as the two main parts of the ICP algorithm is being developed and tested in MATLAB. There are several alternative algorithms for the rotation and translation estimate step of ICP, but SVD is the most widely used. Gauss-Newton method, quaternion method, and a linear least squares approximation algorithm were also tested alongside SVD. The alignment using SVD was verified to produce more accurate results than the other methods. CloudCompare crashed when attempting to perform the ICP algorithm on the complete set of data from two deployments. The CloudCompare ICP algorithm is faster than the MATLAB implementation in development. Alignment accuracy is critical since the accuracy of the subsequent distance calculations are dependent on the accuracy of the alignment between point cloud data sets. Thus, initially a slower execution time for the alignment in MATLAB with the full data set from two point cloud data sets will be acceptable, if the resulting alignment can be proven to be more accurate than the alignment estimate resulting from two less dense partial point clouds aligned in CloudCompare.

Besides the aforementioned steps, the data verification and validation step using standards with known dimensions in the tunnel, which was previously performed using Autodesk Recap Pro, is being automated in MATLAB as well.

By automating the steps listed in TABLE I and through the eventual automation of both the alignment and the data verification and validation steps valuable time is recovered by the data analyst per deployment data set. The switch to MATLAB from CloudCompare significantly reduced execution time of the exhaust tunnel LiDAR data analysis process while also increasing accuracy of and confidence in the distance calculations performed. The steps in TABLE I, the full data analysis process after the data verification and validation step, were tested on both hardware platforms. The execution time results are in TABLE II. The AMD processor is 1.3 times as fast as the Intel processor.

TABLE II. Timing Analysis Hardware

	AMD	Intel	Difference
seconds	11588.76	15016.48	3427.722
minutes	193.146	250.2747	57.1287
hours	3.2191	4.171245	0.952145

REFERENCES

1. J. PLUMMER et al., "Savannah River Site H-Canyon Advancing Technologies for Remote Inspections," *Waste Management Conference 2020*, Phoenix, Arizona, March 8-12, 2020.

2. W. WELLS, J. PLUMMER, J. CARTER, "LiDAR Inspection of GPS Denied Hazardous Environment Exhaust Tunnel," *American Nuclear Society Virtual Winter Meeting 2020*, November 16-19, 2020, Vol. 123, p. 233-234.

3. J. CARTER et al, "2020 Deployments of LiDAR at CAEX Tunnel Pitot Tube Location," C-ESR-H-00080, Savannah River Nuclear Solutions, (2021).