

METROLOGICAL EVALUATION OF CONTRAST TARGET CENTER ALGORITHM FOR TERRESTRIAL LASER SCANNERS

Prem Rachakonda, Bala Muralikrishnan, Daniel Sawyer

Dimensional Metrology Group,
Engineering Physics Division,
National Institute of Standards and Technology,
Gaithersburg, MD, USA

1 ABSTRACT

Terrestrial Laser Scanners (TLSs) are used in a variety of large scale scanning applications such as reverse engineering, assembly of aircraft or ships and surveying. Contrast targets are used with such instruments for enabling scene registration or to establish a scale when used on a scale bar. Currently, the algorithms to calculate the centers of contrast targets (CCT) are either proprietary to the original equipment manufacturers (OEMs) or not precise and accurate. Some of these algorithms may also operate only on OEM's proprietary data format. To overcome these limitations, a novel algorithm was developed at the National Institute of Standards and Technology (NIST) to calculate the center of contrast targets. Several targets in various orientations were scanned by one TLS and their centers were calculated by both the NIST algorithm and one OEM software. The results show that the NIST algorithm is robust, addresses many data quality issues and has better precision than the OEM software in most cases.

Highlights: 1. Algorithms to calculate center of contrast targets are proprietary. 2. NIST's new algorithm enables transparency. 3. NIST's algorithm is robust and precise compared to OEM software.

2 INTRODUCTION

Terrestrial Laser Scanners (TLSs) are instruments that can measure 3D coordinates of objects at high speed resulting in high density 3D point cloud data without the necessity of a cooperative target. Some of the applications of TLS include large scale assembly of aircrafts, ships etc., surveying, and reverse engineering. Spheres, planes, and checkerboard targets (also known as contrast targets) are some of the most common targets used to register TLS scans from various orientations or establish a scale for the entire scene when used on a scale bar. Each of these targets is more reliable than the other in certain scanning scenarios and not all targets are suitable for all applications.

Spheres are also used to register multiple TLS scans as their geometry appears the same regardless of the direction of the scan. However, calculation of a sphere center may be erroneous depending on their placement and any such error will be propagated as registration error. For this reason, spheres are not suitable for some long range (e.g., exceeding 20 m) applications. Flat plates are targets mandated by the ASTM E2938-15 [1] and ASTM E3125-17 [2] standards, but only for relative-range evaluation. Contrast targets, such as the one depicted in Figure 1, improve upon flat plate targets by providing a way to determine the geometric center by using intensity data. These methods to determine target center are proprietary and they do not use only the dimensional data.

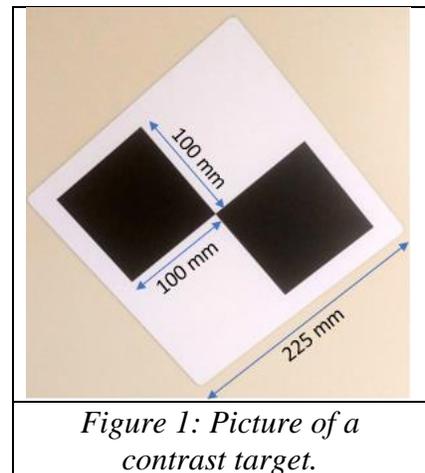


Figure 1: Picture of a contrast target.

Therefore, these proprietary methods were not considered for performance evaluation of TLSs by the ASTM working groups of the above listed standards.

Contrast targets are used with TLS instruments, typically for long range scanning and registering multiple scans in a scene. These targets are very simple in construction and inexpensive to fabricate. Some original equipment manufacturers (OEM) provide software to determine the center of contrast targets, but many end-users use third party software to perform post-processing and such post-processing may result in centers with large variation. Further, different OEM software may produce different target centers for the same target dataset.

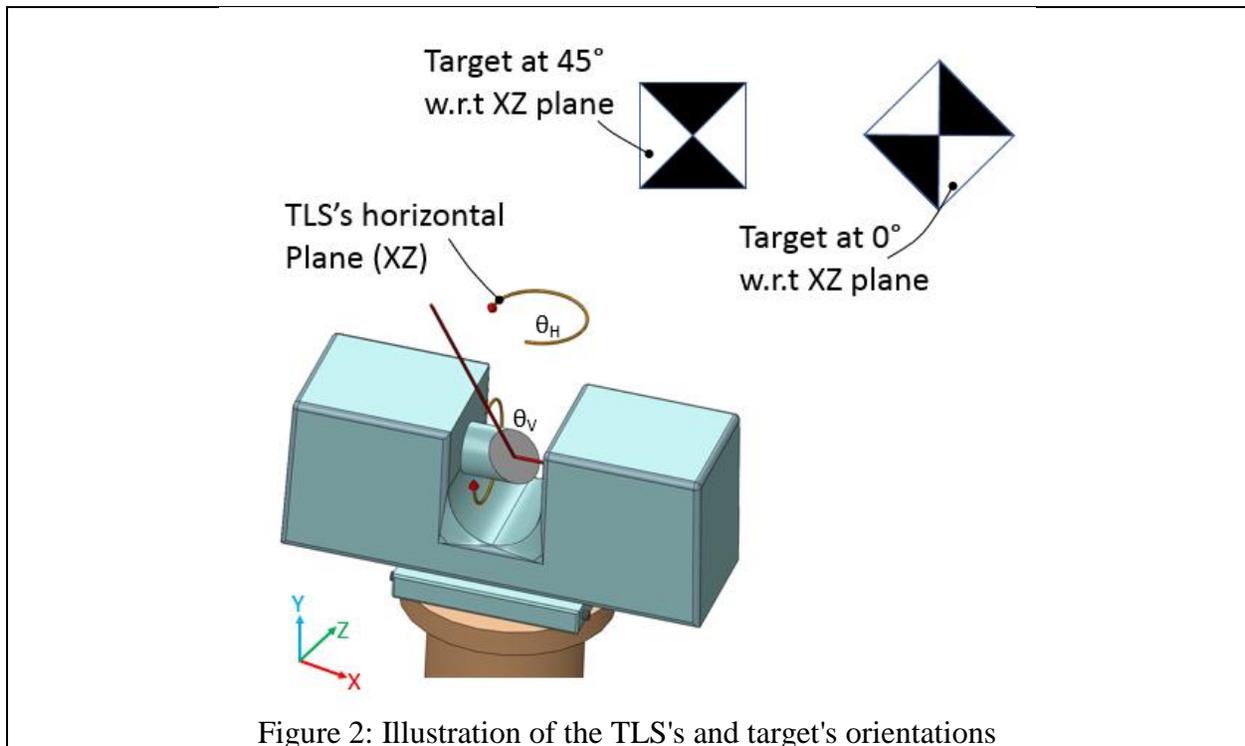


Figure 2: Illustration of the TLS's and target's orientations

In this context, this paper describes and evaluates a novel algorithm developed at the National Institute of Standards & Technology (NIST), to calculate the center of a contrast target and compares its performance in various scan conditions. An algorithm to calculate the derived point or the center of a contrast target (CCT) was developed and reported by Rachakonda et al. [3]. That algorithm was initially developed using one single dataset consisting of multiple targets, placed in ideal orientations where the checkerboard pattern is at $\approx 45^\circ$ to the TLS's horizontal plane (illustrated in Figure 2). This paper describes that algorithm in greater detail and reports on the improvements in the algorithm based on testing against many target orientations and distances. Even though the TLS may introduce uncertainties in determining the CCT [4], only the effects of the algorithm and the targets are studied in this paper. While at the outset, obtaining a CCT may seem trivial, it is challenging to reliably obtain the CCT when the target is tilted or rotated from a non-ideal orientation. The motivation for this work is to have more transparency in determining the CCT with higher precision than available OEM software.

3 DEVELOPMENT OF THE ALGORITHM

Most software that are provided by TLS manufacturers output intensity or color data along with dimensional data for their scans. For determining the CCT, the dimensional data was exported

along with intensity data resulting in a 4D dataset in XYZI format. Some TLSs provide dimensional data along with color intensity in XYZRGB format. Here, X, Y, Z correspond to the dimensional data, I is the intensity data and R, G, B correspond to red, green, blue channel intensity data. If the data obtained is in the form of XYZRGB, the RGB data needs to be converted to intensity values using a weighted sum determined by the formula $I = 0.299 \times R + 0.587 \times G + 0.114 \times B$ [5]. Since the target itself has only two colors, black and white, separate channels of color data are not useful. Subsequently, the XYZI formatted data can be processed using the procedure described in the next three sub-sections. These procedures can be summarized as the following steps to obtain the center of a contrast target.

1. Extract the data corresponding to the region of interest of the contrast target within the scan.
2. Calculate an approximate 3D location of the center of the target using 2D imaging methods.
3. Obtain a refined location of the CCT using dimensional and intensity data.

3.1 Extract targets' regions of interest

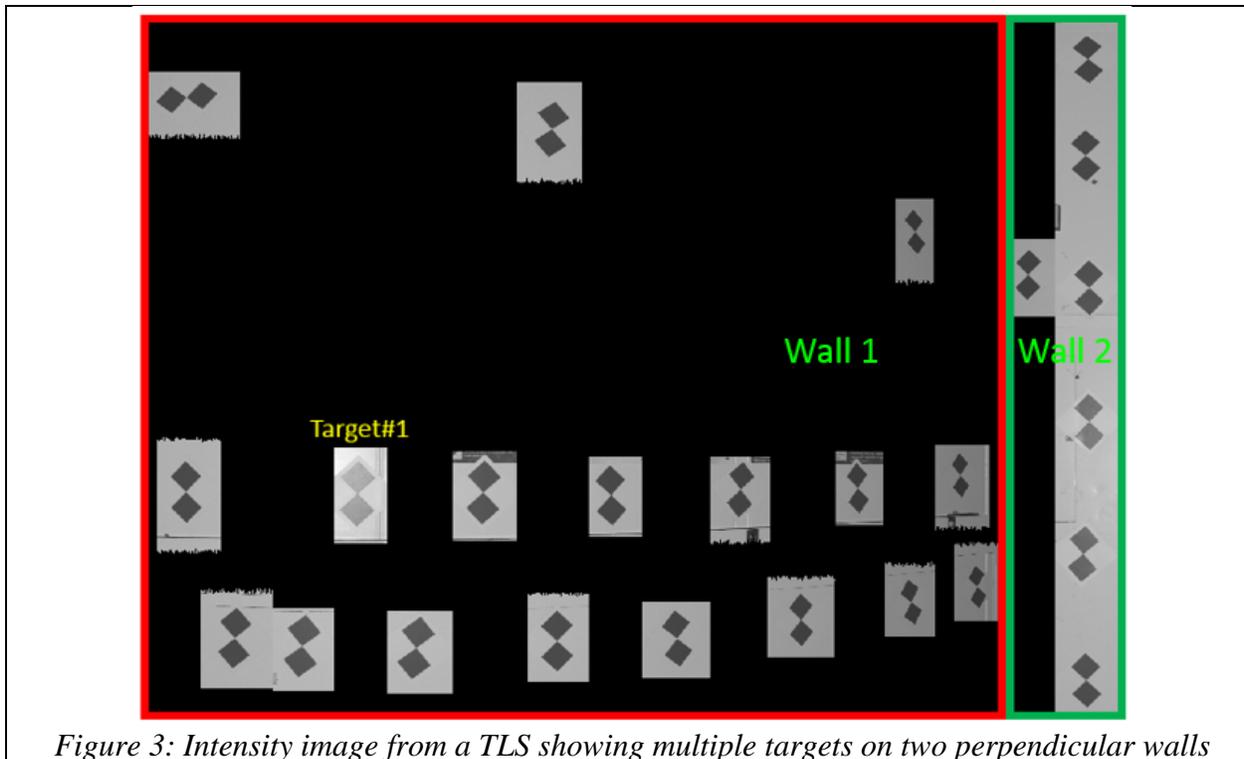


Figure 3: Intensity image from a TLS showing multiple targets on two perpendicular walls

The first step in this process was to extract the regions of interest (ROI) from a scan. This step lowers the complexity and number of computations needed to find the 2D center of the target. Figure 3 is the extracted intensity image of a scan which shows multiple checkboard targets placed around a room in different orientations. Data corresponding to individual target needs to be cropped/separated as this algorithm is designed to operate only on a single target at a time in the scan. This process needs to be performed for both the dimensional and the intensity data.

This 4D cropping of the data in XYZI format can be done either manually or automatically. The manual method may use any point cloud manipulating software for cropping, whereas the automatic methods may use a template matching method. For this work, multiple repeat scans were obtained. All the ROIs were cropped manually once for the first scan dataset and the same cropping

regions were used to crop the remaining datasets. All subsequent processing was performed using automated methods. The next few sub-sections describe the algorithm to determine the CCT.

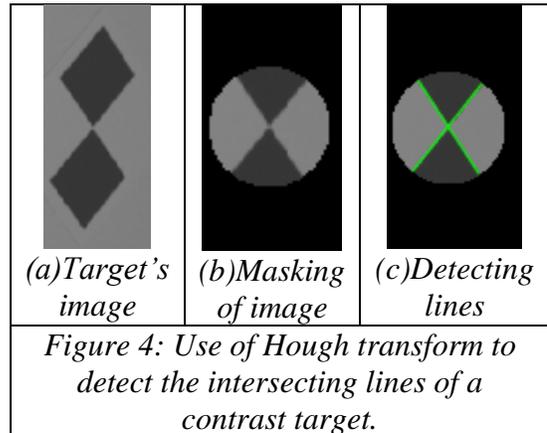
3.2 Calculate the target's approximate 3D center using 2D imaging methods

To obtain the center of the target, the intensity information can be used to generate a 2D image of the target. Subsequently, this image is processed using edge detection methods and the Hough transform [6]. These methods aid in detecting the intersecting lines in a checkerboard-like pattern [7]. The Hough transform is a mature algorithm and is built into software like MATLAB* which was used to process this data. As is usual with many image processing algorithms some level of adjustments is required

when the nature of the image changes, such as change in image contrast or increased noise, which may cause the algorithm to fail. Some of these failure conditions are described later in this paper. The steps to extract an approximate CCT are as follows:

- i. Each cropped XYZI dataset was processed and an image was generated to show a single contrast target as shown in *Figure 4a*.
- ii. The image generated by the previous step was cropped further using a circular mask to display the central region where the two black squares touch each other. Care must be taken to avoid any other regions with intersecting lines/corners. *Figure 4b* shows an example of proper image masking. The masking was done automatically using a predefined mask position and size. Using a square mask instead of a circular mask would lead to detection of additional lines that are at the edges of the square mask.
- iii. Pixel locations corresponding to lines/edges were extracted from the masked image using a "Canny" edge algorithm. Other edge detecting algorithms were explored, but the "Canny" edge algorithm resulted in datasets with fewer lines that was representative of the target.
- iv. A Hough transform was performed on the image from the previous step and the intersecting lines were obtained. It is possible that there may be more than two lines due to the nature of the image (unclean targets, uneven target plane, shadows, etc.). Care must be taken to apply appropriate filters to obtain only two intersecting lines at the center of the target.
- v. The two lines obtained in the previous step were intersected to obtain a 2D approximation of the center of the contrast target CCT_1 as shown in *Figure 4c*.

In the event of failure of the 2D method described above, a manual method may be used to pick the approximate 2D center from the image. It should be noted that CCT_1 was obtained from



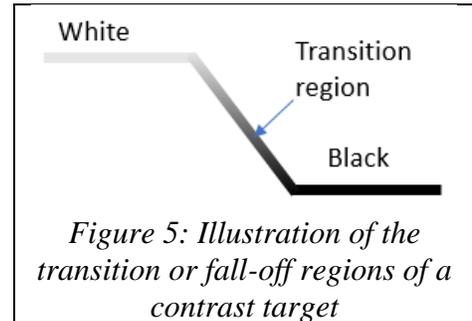
* Disclaimer: Commercial equipment and materials may be identified to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

image data in units of pixels. Even though the pixels are integer numbers, CCT_1 may not have integer values.

Each pixel in the 2D image is an intensity value I from the cropped image and has a corresponding 3D coordinate (X, Y, Z) . To obtain an approximate 3D center, the pixel center coordinates CCT_1 are rounded and an integer pixel location CCT_2 closest to the CCT_1 was obtained. This modified pixel location has a corresponding 3D coordinate X_3, Y_3, Z_3 and intensity value I_3 . The 3D coordinate corresponding to CCT_2 was then $CCT_3 = (X_3, Y_3, Z_3)$, and is an initial estimate for the target center. A corresponding intensity value for CCT_3 was not calculated. CCT_3 may have rounding errors and will be refined in the subsequent steps.

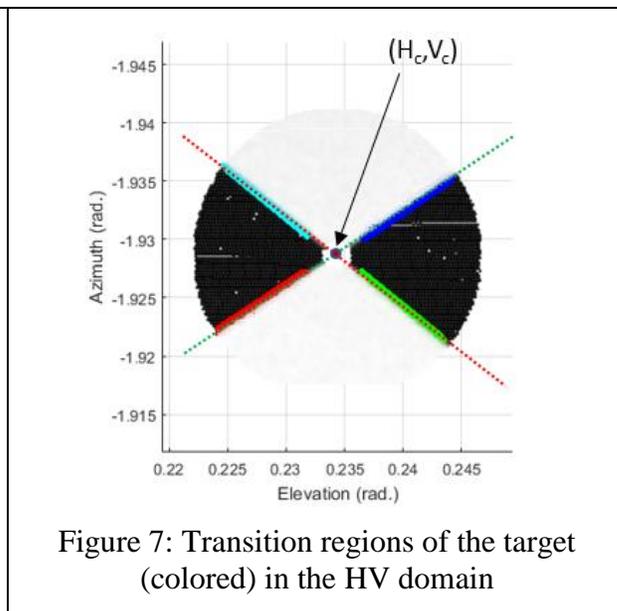
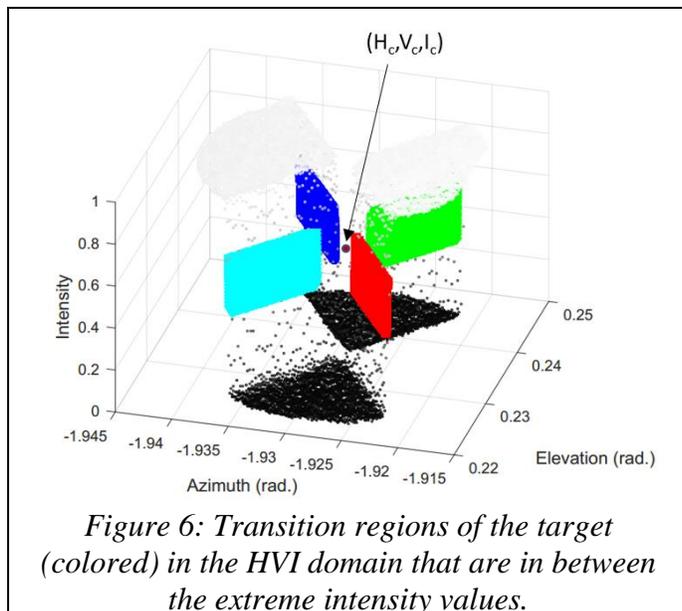
3.3 Calculate a refined target center using dimensional and intensity data

One method to calculate the 3D center of the contrast target was described in section 3.2. When multiple scans of the same target were processed, the approximate 3D coordinates of the centers (CCT_3) were found to have poor repeatability ($1\sigma > 1$ mm), much larger than the expected repeatability of the system and the setup. To improve the calculation of target centers, a new method was conceived which uses intensity data along with dimensional data in non-radial (i.e., horizontal/vertical) directions. To perform this, first the 3D data in Cartesian coordinate



system was converted to data in a spherical coordinate system of form (H, V, R) , where H is the horizontal/azimuth angle, V is the vertical/elevation angle and R is the radial distance to each measured point. A new dataset was created that was comprised of the angles H, V , and the intensity data I . Henceforth, this will be referred as the HVI dataset and the coordinate system as the HVI domain in this paper. The radial data (R) was ignored in this step and the radial component of the center will be calculated at a later stage. This is because, determining the H and V components of the final target center is more challenging than the radial component. To improve the results, the following modifications were performed on the data:

1. The density of data in the HVI domain was increased to 300 times the original point density by interpolation using a cubic polynomial. This was found to lower the uncertainty in calculating the intersection point. Although cubic polynomial interpolation has a possibility of wild swings in the interpolated points, the data was visually inspected after interpolation to ensure that such issues do not exist in the region of interest (fall-off region in Figure 5). Other interpolation techniques were also explored, however cubic interpolation was observed to perform adequately for most datasets without any systematic variation.
2. In an ideal scan of a contrast target, the intensity data typically has a minimum value of 0 (black) and maximum value of 1 (white). However, most scans of contrast targets don't have such values for their black and white regions. This is due to the low intensity returns from the targets to the TLS at various distances. For example, the intensity may range from 0.25 to 0.75. To lower the variability of determining the CCT, intensity data was normalized/scaled from 0 to 1.



In the HVI domain, the intensity values corresponding to the intersecting lines in a 2D image have a sharp fall-off as illustrated in Figure 5. In this fall-off region, the intensity drastically changes from $I \approx 1$ to $I \approx 0$. These regions in the HVI domain are shown in Figure 6 where the colored regions in the middle are the high density interpolated points corresponding to the fall-off region. The following steps are then performed to obtain the intersection point:

- a. The fall-off region in the HVI domain was obtained by discarding the surfaces corresponding to extreme intensities (black and white in Figure 6) that are over 0.5σ times the standard deviation (0.5σ) from the mean intensity value. This value of 0.5σ was empirically determined and it ensured that only data belonging to the intersecting fall-off regions was obtained.
- b. This fall-off region was then split into four parts (S1 through S4) after truncating the central cylindrical region close to the approximate center. This was performed using an automated method and this truncation enables the intersecting regions to be separated into four parts (as shown by the colored regions in Figure 6).
- c. The intensity data from four datasets in the HVI domain was then discarded, keeping only H and V. This process collapses the data to a single plane in HV domain, yielding datasets corresponding to four lines on the target.
- d. In this step, a 2D intersection point in the HV domain was obtained. Three different methods were explored to obtain this 2D intersection point and are described below:
 - i. The four datasets corresponding the four lines can be intersected using a least-squares method in the HV domain to yield the center of the target (H_c, V_c) as illustrated in Figure 7.
 - ii. These four datasets can be grouped together into two datasets corresponding to two lines. For example, S1&S2 may form one line, and S3&S4 may form the second line. These datasets can be least-squares fit to two lines, which can then be intersected to obtain (H_c, V_c).
 - iii. Alternatively, before step#c, the four surfaces in the HVI domain (colored regions in Figure 6) can be fit to planes. These four plane equations can be

solved using a least-squares method to perform a four-plane intersection. This method results in an intersection point (H_c, V_c, I_c) in and thereby obtaining a center (H_c, V_c) in 2D.

In general, it was found that 2D line intersection of two lines was more repeatable than the other two methods to yield an intersection point (H_c, V_c) . Both the plane and line fitting routines were performed iteratively after excluding data points whose corresponding residuals exceeded three times the standard deviation of the residuals. The iterations were terminated when there were no more points to exclude.

- e. The radial value of the center (R_c) was the radial value of CCT_3 in spherical coordinate system. It should be noted that CCT_3 was calculated using 2D imaging methods described in section 3.2. This value of R_c is an approximate value and will be improved in the subsequent steps.
- f. The center (H_c, V_c, R_c) was then converted to a Cartesian coordinate system to obtain $CCT_4 = (X_c, Y_c, Z_c)$.
- g. As a last step, CCT_4 was projected onto the plane of the contrast target in the XYZ domain. This was performed by intersecting the line joining the origin and CCT_4 with the least-squares fitted plane of the contrast target. This projected point $CCT_5 = (X_p, Y_p, Z_p)$ was the final center of the contrast target. This projection method was required to ensure that the final center CCT_5 lies on the plane of the contrast target.

In summary, the centers of the contrast targets were calculated using a method involving multiple steps. The progression of centers obtained using the method described in this paper are listed below and a flowchart for the procedure is shown in Figure 8.

- CCT_1 : 2D center using image processing methods in the units of pixels.
- CCT_2 : 2D center with integer values of CCT_1 in the units of pixels.
- CCT_3 : 3D center corresponding to CCT_2 .
- CCT_4 : 3D center using HVI domain method.
- CCT_5 : Final center obtained by projecting CCT_4 on the target's least-squares fitted plane.

3.4 Addressing the cases of failure of the algorithm

The procedure to calculate the CCT has several limitations, many of which relate to the quality of the data and target orientation. Many targets used for TLSs have ideal/recommended orientations. To improve the quality of the algorithm and to study the ideal orientation of the contrast target, the NIST algorithm was tested against several other datasets whose results are not included in this paper. Some of the targets were intentionally placed in non-ideal orientations to test the algorithm. Numerous issues were observed while processing such data and appropriate measures were taken to either solve the issue or discard the dataset (i.e., no CCT can be determined). It should be noted that OEM software also fails to identify a CCT, typically when the data quality is poor.

In general, if the algorithm fails to find the center using the 2D method, the user provided approximate center is used. If the algorithm fails to find the 3D center, the region of interest (ROI) of the contrast target is either reduced or increased and the dataset is processed again. It was observed that in many cases such a method results in the algorithm finding the 3D center (CCT) successfully. The ROI was increased to 1.5 times the initial ROI or decreased to 0.5 times the initial ROI. The purpose of this is to simplify the search process of the CCT. If the target is tilted, the ROI should be an elliptical region instead of circular, as the two intersecting lines in the

intensity image will not be of equal length. Varying the size of a circular ROI simplifies the calculation of the elliptical ROI individually for each target.

The increase or decrease of the circular ROI was done iteratively until a CCT was found. If no CCT can be obtained by expansion or reduction of the ROI, the algorithm is deemed to fail and no further attempt was made to obtain the CCT. Below are some of the reasons of failure of the algorithm.

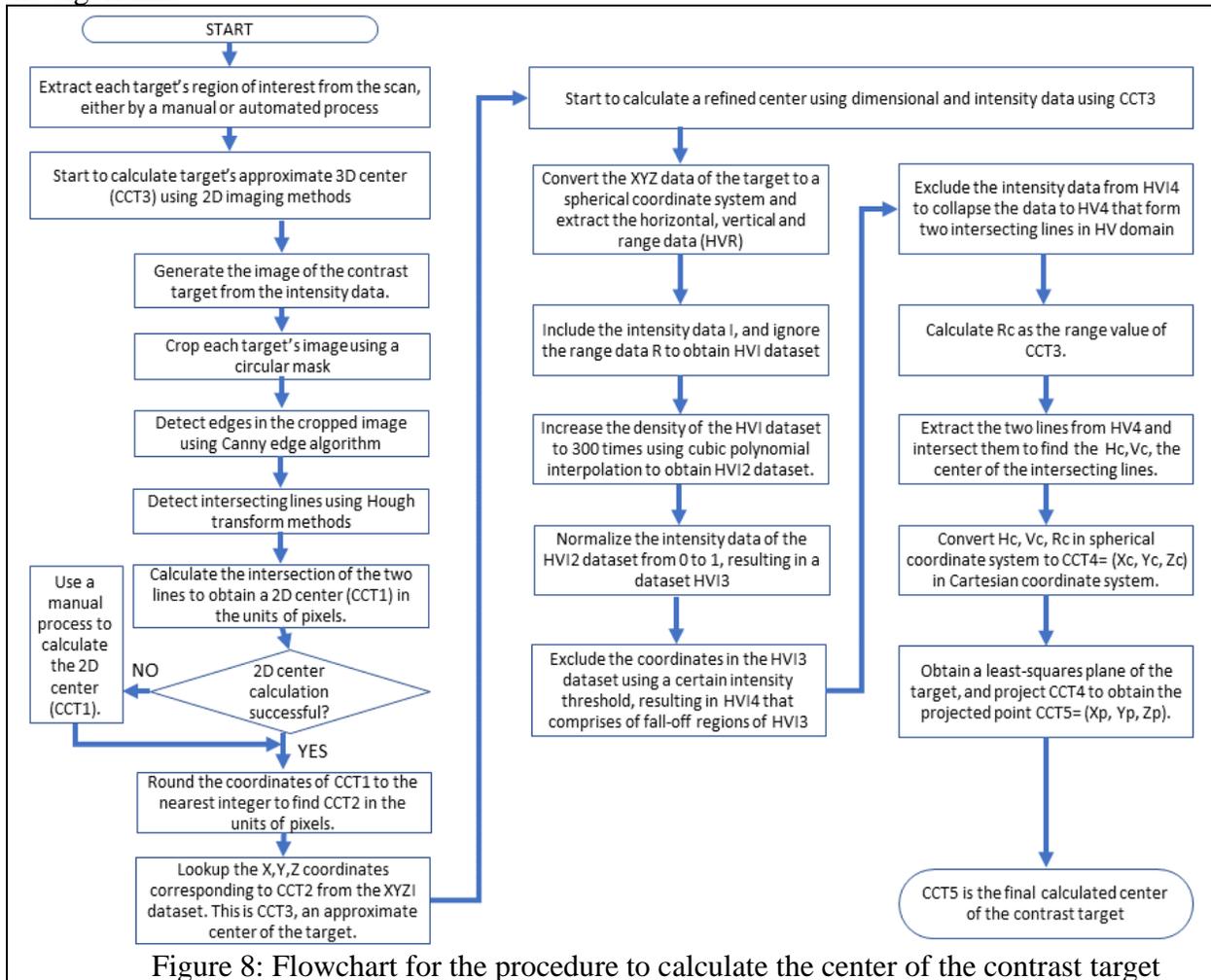


Figure 8: Flowchart for the procedure to calculate the center of the contrast target

3.4.1 Target that is nearly orthogonal to the laser beam

In some cases, scans of targets that were nearly orthogonal to the laser beam had reflections or shiny regions causing some black regions of the target appear white or gray. This resulted in the software to incorrectly segment the regions and calculate centers with larger variation. Subsequent error checks described in the next sub-sections minimize such issues.

3.4.2 Hough transform finds only one line

In cases where the Hough transform cannot find at least two lines, the algorithm fails. The approximate location of the CCT provided by the user is then used as the 2D intersection of the target.

3.4.3 Hough transform finds more than two lines

In cases where the Hough transform finds more than two lines, the 3rd line was mostly along one of two intersecting lines (green lines in *Figure 4*). In such cases, the intersection of the three lines is performed using a least-squares method. This may result in a slightly inaccurate 2D center compared to a center obtained using two lines.

It is also possible that the target is at the edge of a scan region and the 3rd line could be at one of the edges. In such cases using a circular mask still would not exclude the Hough transform from detecting a 3rd line. In such case the algorithm fails and the ROI must be reduced and the data needs to be reprocessed.

3.4.4 Hough transform finds two lines that are nearly parallel

In some of the cases, the Hough transform finds two lines but they are nearly parallel and intersect outside the region of interest. The approximate location of the CCT provided by the user is then used as the 2D intersection of the target.

3.4.5 Target has unusual values for their width and thickness

It is possible to have a segmented target dataset that has a larger than normal thickness and width. These conditions occur due to improper data truncation or improper target mounting. To address this issue, the data corresponding to the contrast target, which is of the form [X Y Z] is first rotated along its principal components resulting in a data of the form [X' Y' Z']. The range of the dominant component of the transformed dataset, X', should not exceed the target's expected width. Similarly, the range of the least dominant component, Z', should not exceed the instrument's expected noise at that location. The range of Z' typically exceeds the instrument's expected noise in the following conditions:

- a) target is not flat,
- b) target mounted on non-flat surfaces
- c) target is bent physically
- d) target's dataset is truncated incorrectly to include data away from the target's plane.

3.4.6 Failure to separate the 3D dataset into four regions in HVI domain

To automate the process of finding the CCT, a clustering algorithm known as the kmeans-algorithm [8] was used. It attempts to find four clusters of data which correspond to the transition regions between the black and the white part of the target. If the dataset is not truncated properly from its surroundings, it is possible to have five clusters of data. When kmeans-algorithm identifies only four clusters, the data from the fifth cluster typically shows up as part of one of the four clusters and this results in calculation of an erroneous CCT.

3.4.7 Failure to combine the four regions to form two intersecting lines

After successfully separating the four regions in the HVI domain, the intensity data I is excluded to obtain data in the HV domain corresponding to four lines. These four lines are then combined to form two lines and are intersected to find the center in HV domain. This combination process may fail and precautions were introduced to ensure that only lines that are not adjacent to each other are combined.

3.4.8 Intersection with perfectly vertical lines

In some cases, the intersecting lines could be perfectly vertical with a slope of ∞ . Such cases must be handled by modifying the method used to perform a line fit. Such an issue is more

prevalent with image processing method described in section 3.2. This issue can also be addressed by ensuring that the target's intersecting lines are not orthogonal to the TLS's horizontal plane, for example, rotating the target by 45° .

3.4.9 The calculated 3D center is outside the target's region

In all the cases, the final 3D center CCT_5 must be within the region corresponding to the targets. Some of the previously described failure conditions could also yield this situation. A check was performed to ensure that the final center CCT_5 was within a convex hull determined by the truncated dataset of the target.

4 COMPARISON WITH COMMERCIAL SOFTWARE

To understand the existing methods and to compare them with the method described in this paper it was necessary to find a commercial or original equipment manufacturer (OEM) software that offered a CCT with lowest variation. The next section describes this selection process.

4.1 Selection of the OEM software

Three OEM software packages were used and repeatability studies were conducted to perform this test. In this test, two contrast targets were scanned 10 times and their CCTs and the standard deviations (1σ) of those CCTs were calculated using each software. Lower 1σ values of all the 3D coordinates (σ_x , σ_y , σ_z) indicates a more robust algorithm. This metric however considers only the precision of the centers but not their accuracy. It should be noted that there may be other commercial or proprietary software which may perform better, but were inaccessible to the authors at the time of writing this paper. One OEM software (Method#1) yielded lowest 1σ values (lower by an order of magnitude) and this software was used to compare the algorithm described in this paper (Method#2) using the test setup described in the next sub-section.

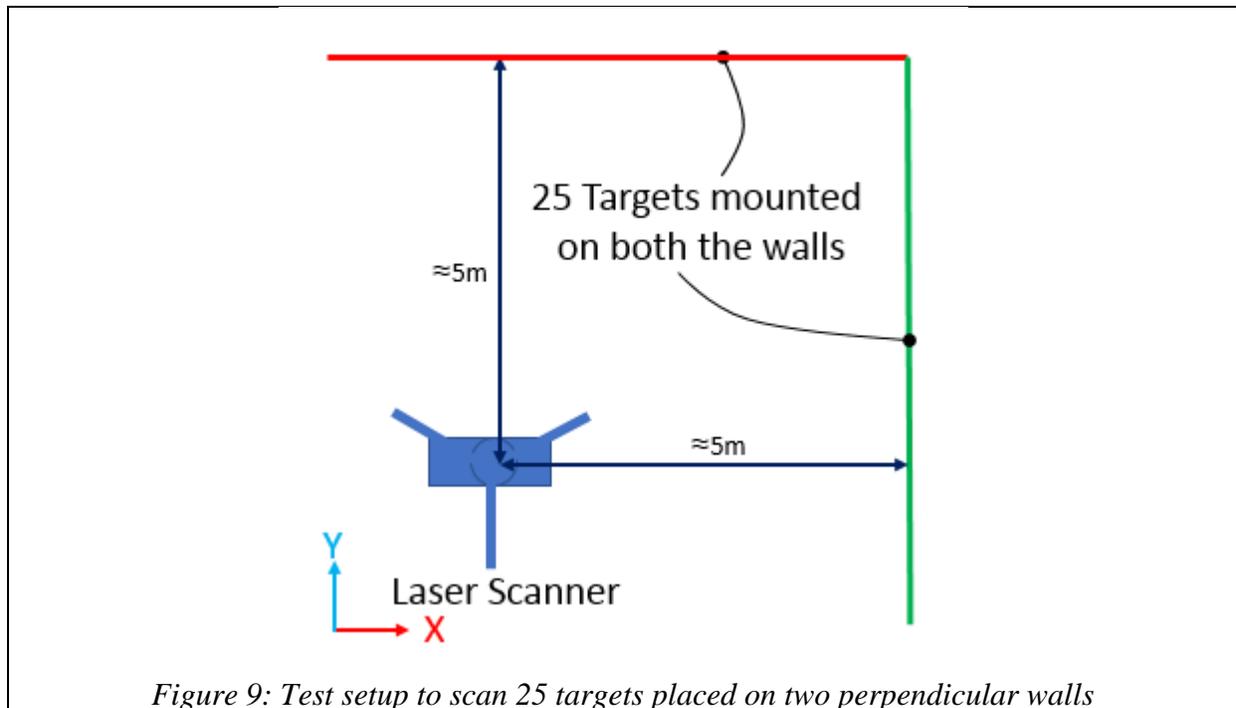


Figure 9: Test setup to scan 25 targets placed on two perpendicular walls

4.2 Test setup to compare the OEM software with the NIST algorithm

This test setup involved placing 25 contrast targets on two walls as illustrated in Figure 9. These targets (depicted in Figure 1) were square in shape, ≈ 225 mm wide, fabricated out of a flexible plastic material and have a magnetic backing for the purposes of mounting. Seven targets were mounted on a wall (green wall) on the right distributed vertically and the rest were mounted on a wall that was perpendicular (red wall), distributed horizontally. The TLS was placed at ≈ 5 m from both the walls. The distances of the TLS to each target (L) and the angles of incidence (θ) are listed in *Table 1*. The angle of incidence θ is the angle between the target's surface normal and the laser beam at the target's nominal center. Figure 3 shows one of the scans obtained using this setup and Figure 1 shows one of the targets used in this setup. Even though these targets appear to be in the same plane in Figure 3, they are in fact on two walls that are perpendicular to each other.

To perform this comparison, these targets were scanned 10 times using a TLS. Data acquired by this TLS was exported both to its own proprietary file format and to the XYZI format. OEM software (Method#1) was used to process the data in this proprietary file format and the CCT of all the targets were obtained. This software did not have the capability to process the data in the XYZI format, but only it in its proprietary data format. Method#2 was used to process the data in the XYZI format. It is possible that the proprietary data format may have additional high density or higher resolution data that may not be available when exported to the XYZI format.

After processing the data, the standard deviation of the centers in spherical coordinates calculated by Method#1 was σ_1 , and as calculated by Method#2 was σ_2 . Here $\sigma_1 = (\sigma_{A1}, \sigma_{E1}, \sigma_{R1})$ and $\sigma_2 = (\sigma_{A2}, \sigma_{E2}, \sigma_{R2})$, the standard deviation values of the individual components of the centers in the spherical coordinates. To perform a comparison in the units of length, the standard deviations in azimuth and elevation were multiplied by the average radial distance value of the center. i.e., $\sigma_{H1} = R_1 \times \sigma_{A1}$, $\sigma_{V1} = R_1 \times \sigma_{E1}$, $\sigma_{H2} = R_2 \times \sigma_{A2}$, $\sigma_{V2} = R_2 \times \sigma_{E2}$. Here R_1 and R_2 were the average radial distances of the center as determined by Method#1 and Method#2, respectively.

Ideally, $\sigma_2 \leq \sigma_1$ along all the three components, but that may not be the case. To make the comparison simpler, a quality factor M given by equation 1 was introduced to compare the methods at each CCT and *Table 1* shows all the parameters calculated using both the methods using the 10 repeat measurements. This method of comparison is useful since Method#1 (OEM software) was found to be consistently producing centers with lower variation among the three commercial software packages that were evaluated.

If the standard deviation values from both the methods are identical, $M = 1$. If Method#1 performs better than Method#2 then $M > 1$ and *vice versa* if $M < 1$. There may be cases where $M = 1$ if Method#1 outperforms in one component and underperforms in another and in that case the performance of the individual components must be observed. This metric M gives a good estimate of the overall method performance and M_{AZ} , M_{EL} , M_{RR} reveal the performance of the method in individual spherical coordinate components.

$$M = \frac{M_{AZ} + M_{EL} + M_{RR}}{3}, \quad 1$$

$$\text{where } M_{AZ} = \frac{\sigma_{A2}}{\sigma_{A1}}, M_{EL} = \frac{\sigma_{E2}}{\sigma_{E1}} \text{ and } M_{RR} = \frac{\sigma_{R2}}{\sigma_{R1}}.$$

4.3 Discussion of comparison methodology

The results illustrated in *Figure 10a* & *Figure 10b* and *Table 1* show that Method#2 performs reasonably well in all the three components. Of the 25 targets used in this testing, 17 targets had $M \leq 1$ and eight targets had $M > 1$. The distances d_M between the centers obtained from Method#1 and Method#2 were calculated for each target and the average value of d_M for all the targets was ≈ 0.14 mm.

Target #	Method#1 (μm)			Method#2 (μm)			L (m)	M	θ°
	σ_{H1}	σ_{V1}	σ_{R1}	σ_{H2}	σ_{V2}	σ_{R2}			
1	48	76	40	43	76	34	4.4	0.9	2
2	37	68	29	37	75	27	4.5	1.0	11
3	48	71	33	48	65	28	4.5	0.9	6
4	54	69	30	51	71	27	4.5	1.0	6
5	57	80	27	53	79	25	4.7	1.0	15
6	48	72	36	47	70	30	4.7	0.9	16
7	35	67	28	33	66	27	4.7	1.0	18
8	42	85	36	43	82	35	4.8	1.0	22
9	16	86	28	12	83	26	4.9	0.9	15
10	8	64	24	12	64	24	5.0	1.2	18
11	60	87	32	60	87	30	5.0	1.0	26
12	39	111	21	36	117	24	5.1	1.1	5
13	55	123	29	52	122	24	5.1	0.9	6
14	38	94	33	41	97	32	5.2	1.0	31
15	18	108	22	24	108	21	5.2	1.1	5
16	64	124	29	67	120	28	5.2	1.0	6
17	13	120	22	17	121	18	5.4	1.0	5
18	56	102	34	56	100	29	5.4	0.9	34
19	34	129	17	20	122	18	5.6	0.9	4
20	24	108	39	24	113	30	5.7	0.9	23
21	48	109	34	47	107	36	5.7	1.0	39
22	66	115	30	62	111	31	6.2	1.0	44
23	25	111	30	24	116	26	6.3	1.0	44
24	51	122	31	55	115	27	6.4	1.0	47
25	65	120	30	66	116	32	6.6	1.0	47

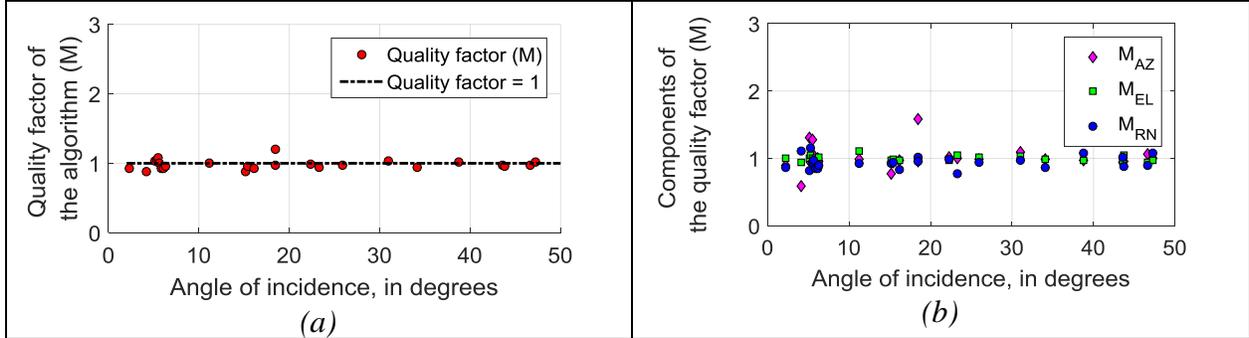


Figure 10: Performance of the NIST algorithm at various angles

To calculate the accuracy of the algorithm, the three spherical components of the centers from both Method#1 (OEM software) and Method#2 were compared. Since there are 10 datasets for each of the 25 targets, the mean values of the centers were used to perform this comparison.

Figure 11 shows the error in all the three components of CCT, and their corresponding mean error values (μ) in the legend. Here, $e_H = 0.5 \times (R_1 + R_2) \times (A_2 - A_1)$, $e_V = 0.5 \times (R_1 + R_2) \times (E_2 - E_1)$, $e_R = (R_1 - R_2)$, where R_1 and R_2 are the range values of the CCT, A_1 and A_2 are the azimuth values of the CCT, E_1 and E_2 are the elevation values of the CCT and the subscripts 1 and 2 refer the Method#1 and Method#2 respectively. The error in the azimuth and elevation values were multiplied by the average range value of the centers calculated by Method#1 and Method#2 to enable them to be compared in the units of length.

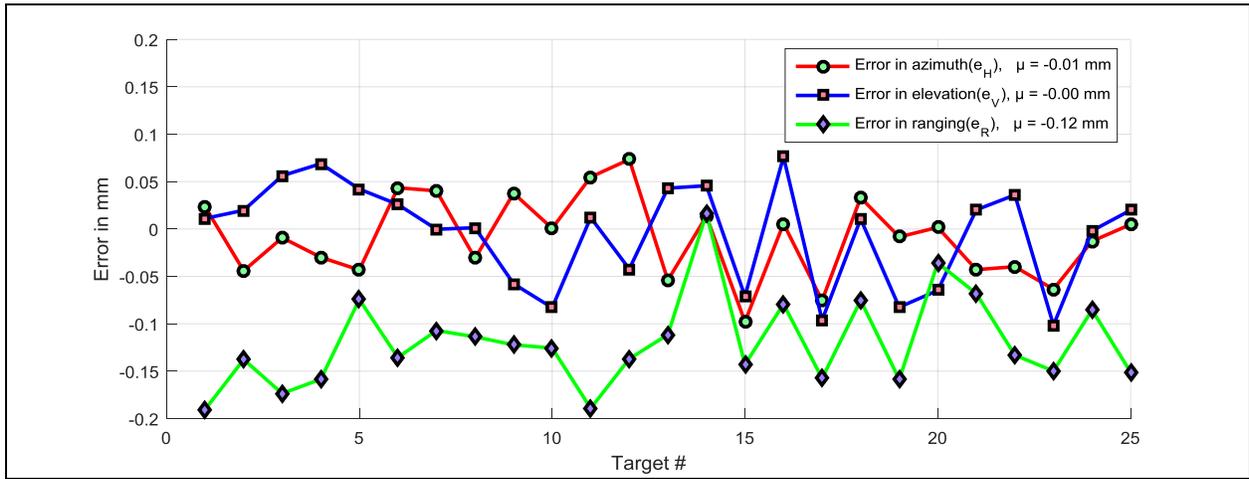


Figure 10: Accuracy of the NIST algorithm as compared to the OEM software

Figure 10 shows that there is no systematic bias in the azimuth or elevation components of centers from both the methods, but there is a bias in the ranging direction, an average of ≈ 0.12 mm. This bias could be a result of Method#2 locating the CCT on the least-squares fitted plane of the target, whereas the OEM software (Method#1) calculates a CCT that is mostly away from the same plane and towards the TLS. It should be noted that the accuracy of either method cannot be ascertained using a single point measurement. Such a comparison would require test procedures involving calibrated lengths between two contrast targets in various orientations.

5 COMPARISON AGAINST OTHER TARGETS

To assess the quality of this new algorithm, another test was conducted which used both spheres and contrast targets. Spheres are some of the most common targets used with TLSs that

are known to offer a better estimate of the derived point than any other target. They are known to suffer from errors that result in a squishing/flaring effect that introduce an error in the fitted center in the ranging direction [9]. The tests described in this section involve moving the contrast and sphere targets laterally by known amounts, and determining the error in that displacement. For this purpose, the NIST and OEM algorithms were employed on the contrast target while a standard orthogonal least-squares fitting algorithm was employed on the sphere targets. The test setup and results are described next.

5.1 Test setup

For these tests, two spheres and a contrast target were mounted on a translation stage. The motion of the translation stage was perpendicular to the laser beam from the TLS, along X-axis in Figure 11. A 1.5 in (38.1 mm) Spherically Mounted Retroreflector (SMR) was mounted on the stage to measure its displacement using a laser tracker. The motion of the translation stage was aligned to the laser tracker in such a way that the errors due to the tracker's angular encoders are minimized. This entire setup was then mounted on a tripod.

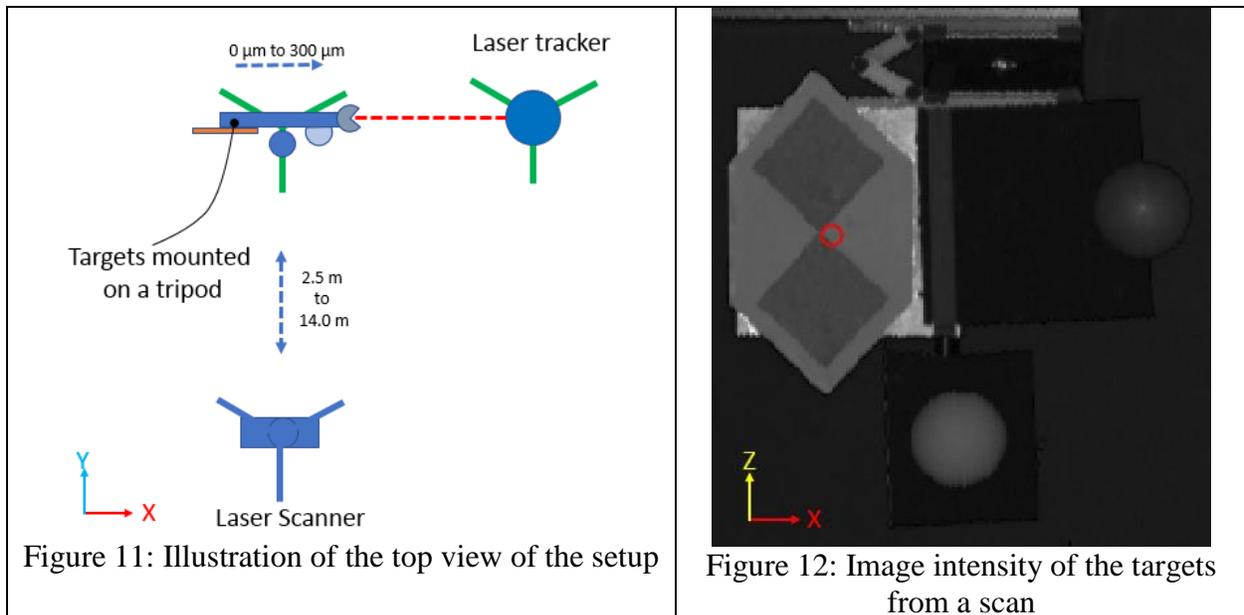


Figure 11: Illustration of the top view of the setup

Figure 12: Image intensity of the targets from a scan

The three targets that were mounted on the translation stage were as follows

- A contrast target, depicted in Figure 12, with the dark square regions of dimensions $100 \text{ mm} \times 100 \text{ mm}$. The target was intentionally oriented in such a way that the intersecting lines are not orthogonal to the TLS's horizontal plane.
- An aluminum sphere target with a scannable surface that was media blasted/satin finished, as depicted at the right in Figure 12.
- A commercial scanning sphere that is painted white, depicted at the bottom in Figure 12, was mounted in such a way that its mounting apparatus was not in the field of view of the TLS.

The purpose of using two spheres in this step is that the white scanning sphere was known to result in higher quality data due its white surface, although its sphericity value was $\approx 100 \mu\text{m}$. On the other hand, the aluminum sphere has a sphericity of $\approx 10 \mu\text{m}$, but has a finish that is known to result in lower quality data at longer distances.

The test procedure involved moving the TLS away from the tripod with all the 3 targets to multiple locations in the ranging direction of the TLS (along Y-axis in Figure 11). The tripod and

the laser tracker remained stationary. The TLS was moved from ≈ 2.5 m to ≈ 14 m away from the tripod in increments of ≈ 2 m. At each location of the TLS, the translation stage on the tripod, along with the targets, was moved laterally, along X-axis in Figure 11, seven times from its initial position in increments of ≈ 50 μm . At each position of the translation stage, the stage's position was recorded using a laser tracker using an SMR mounted on the stage and the targets were scanned six times for calculating the statistics. At each position of the TLS, the targets were scanned 48 times in total - six repeat scans in eight lateral positions. Such measurements were obtained with the TLS at seven distinct positions that were ≈ 2 m away from each other.

5.2 Discussion of results

To compare the results at various TLS positions, lateral displacement errors, along X-axis, were calculated. This error is the difference in the displacements of the targets as measured by the TLS and the displacement of the translation stage as measured by the laser tracker. These displacement errors are denoted by $e_{j,k}^i$, where $i = 1$ to 7 for each of the TLS positions, $j = 1$ to 7 for each of the lateral displacements and $k = 1$ to 6 for each of the repeat scans that were obtained. For each displacement error, a standard deviation value σ_j^i was also calculated.

There are several ways to calculate the displacement of the targets given that there are six repeat scans acquired at each position of the translation stage. At any TLS position i , the displacement between two target positions is calculated using the formula given in equation 2:

$d_{j,k}^i = \sqrt{(X_{j,k}^i - X_{j+1,k}^i)^2 + (Y_{j,k}^i - Y_{j+1,k}^i)^2 + (Z_{j,k}^i - Z_{j+1,k}^i)^2},$	2
---	---

where $d_{j,k}^i$ is the displacement corresponding to scan dataset k at target position j ; $(X_{j,k}^i, Y_{j,k}^i, Z_{j,k}^i)$ and $(X_{j+1,k}^i, Y_{j+1,k}^i, Z_{j+1,k}^i)$ are the target centers for the k^{th} scan at the j and $j+1$ positions, respectively.

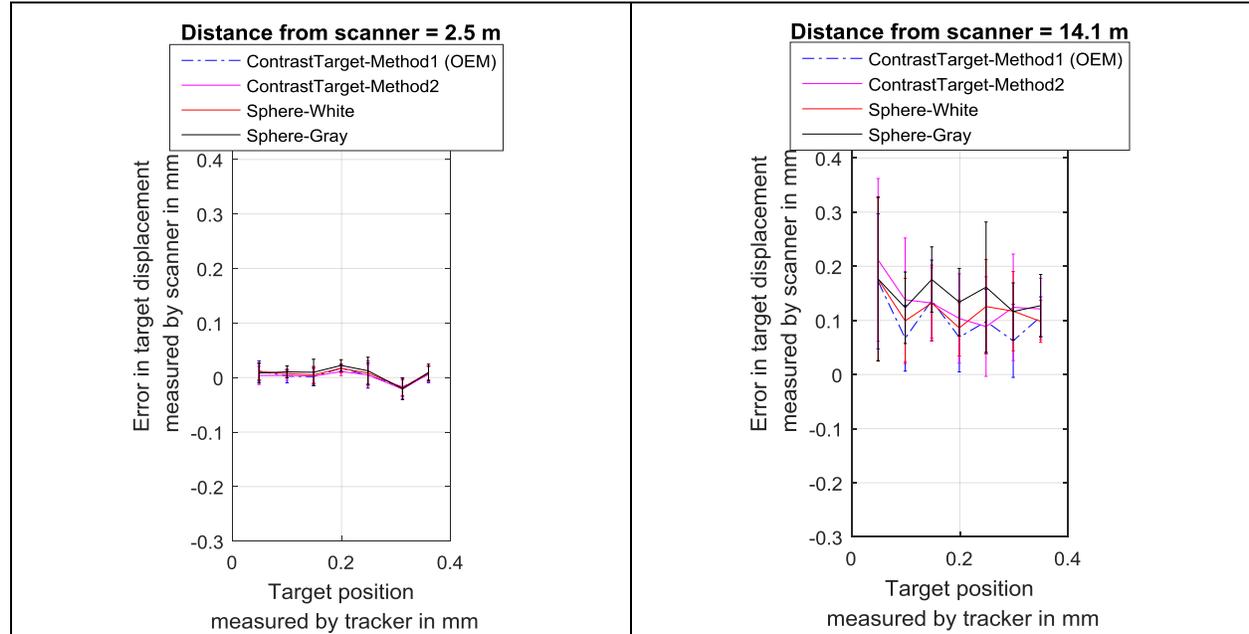


Figure 13: Comparison of the lateral displacement of the targets at two ranging locations.

Figure 13 shows the plots of the mean displacement errors at two extreme locations of the TLS from the tripod (2.5 m and 14.1 m). It shows large variance and bias in measuring the displacement of the targets at 14.1 m than at 2.5 m. In general, the errors do not vary significantly either with the target or the algorithm at each individual tripod location, but vary as a function of the target distance from the TLS. This demonstrates that the NIST algorithm is at least as robust as the proprietary OEM software and sphere center algorithms.

To show the trend observed in Figure 13 for all the targets and positions, pooled standard deviation values were calculated and plotted in Figure 14. The pooled standard deviation (s_p) value is calculated by combining the seven standard deviation values (σ_j^i) using a root-mean-square method, for each TLS position and target. Note that these values are calculated from the displacement values in the lateral direction (non-ranging direction), hence any sphere squishing/flaring related errors will be negligible. In general, it was observed that the targets perform very similarly irrespective of the geometry or the data processing techniques used in this paper at distances < 10 m, but vary slightly at target distances > 10 m.

6 SUMMARY

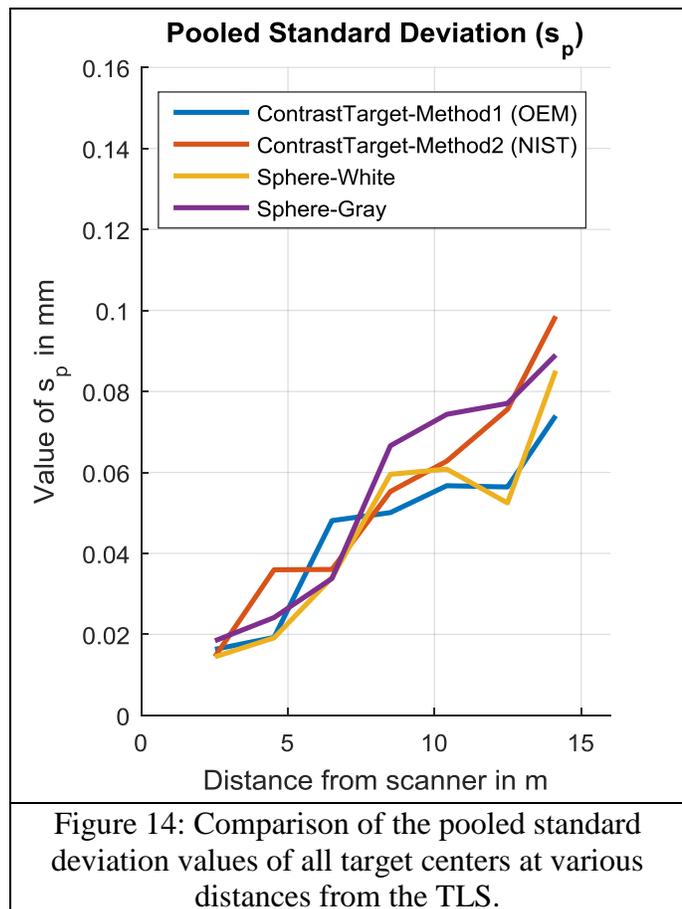
Contrast targets are used with TLS instruments for applications requiring long range scanning and/or registration. Large contrast targets are inexpensive to fabricate compared to similar sized spheres or flat plates required for such applications. In this context, this paper describes a novel algorithm that has been developed at NIST to determine the center of a contrast target that is comparable to that obtained using available OEM software. This new algorithm was tested against several hundred scans of targets in numerous orientations to make it more robust. A variety of failure conditions were observed and the algorithm was modified to address those failure conditions. The new algorithm performs well when compared against OEM software and when compared against results from sphere targets.

7 ACKNOWLEDGEMENTS

The authors would like to thank Dr. Ling Wang of China Jiliang University for his valuable help & feedback.

8 REFERENCES

- [1] ASTM E2938-15 Standard Test Method for Evaluating the Relative-Range Measurement Performance of 3D Imaging Systems in the Medium



- Range, ASTM International, West Conshohocken, PA, 2015, <https://doi.org/10.1520/E2938-15>
- [2] ASTM E3125-17 Standard Test Method for Evaluating the Point-to-Point Distance Measurement Performance of Spherical Coordinate 3D Imaging Systems in the Medium Range, ASTM International, West Conshohocken, PA, 2017, <https://doi.org/10.1520/E3125-17>
- [3] Rachakonda, P., Muralikrishnan, B., Ling, W., Sawyer, D., "Method to Determine the Center of Contrast Targets from Terrestrial Laser Scanner Data", Proceedings of the Annual Meeting of the ASPE; Charlotte, NC; 2017.
- [4] Muralikrishnan, B., Ferrucci, M., Sawyer, D., et al., "Volumetric performance evaluation of a laser scanner based on geometric error model", Precision Engineering, 40, p. 139–150, 2015. <https://doi.org/10.1016/j.precisioneng.2014.11.002>
- [5] ITU-R BT.601-71 - Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-E.pdf (Accessed on 5/2/2018)
- [6] Duda, R., Hart, P., "Use of Hough Transformation to Detect Lines and Curves in Pictures", Communications of the ACM, January 1972, Vol.15, No. 1
- [7]http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html (Accessed on 5/2/2018)
- [8] <https://www.mathworks.com/help/stats/kmeans.html> (Accessed on 5/16/2018)
- [9] P. Rachakonda, B. Muralikrishnan, L. Cournoyer, G. S. Cheok, V. D. Lee, K. M. Shilling, D. S. Sawyer, "Methods and Considerations to Determine Sphere Center from Terrestrial Laser Scanner Point Cloud Data" Measurement Science and Technology, 28(10) 2017 doi.org/10.1088/1361-6501/aa8011