Characterization of Three Algorithms for Detecting Surface Flatness Defects from Dense Point Clouds

Pingbo Tang, Dept. of Civil and Environ. Eng., Carnegie Mellon Univ. Pittsburgh, PA 15213, USA, Tel: 412-527-0016, Email: tangpingbo@cmu.edu

Burcu Akinci, Dept. of Civil and Environ. Eng., Carnegie Mellon Univ. Pittsburgh, PA 15213, USA, Tel: 412-268-2959, Email: bakinci@cmu.edu

Daniel Huber, Robotics Institute, Carnegie Mellon Univ. Pittsburgh, PA 15213, USA, Tel: 412-268-2991, Email: dhuber@cs.cmu.edu

ABSTRACT

Surface flatness assessment is required for controlling the quality of various products, such as building and mechanical components. During such assessments, inspectors collect data capturing surface shape, and use it to identify flatness defects, which are surface parts deviating from a reference plane by more than the tolerance. Laser scanners can deliver accurate and dense 3D point clouds capturing detailed surface shape for flatness defect detection in minutes. However, few studies explore algorithms for detecting surface flatness defects from dense point clouds, and provide quantitative analysis of defect detection performance. This paper presents three surface-flatness-defect detection algorithms and our experimental investigations for characterizing their performances. We created a test bed, which is composed of several flat boards with defects of various sizes on them, and tested two scanners and three algorithms using it. The results are reported in the form of a set of performance maps indicating under which conditions (using which scanner, scanning distance, selected defect detection algorithm, and angular resolution of the scanner, etc.), what types of defects are detected. Our analysis shows that scanning distance and angular resolution substantially influence the detection accuracy. Comparative analyses of scanners and defect detection algorithms are also presented.

Keywords: Laser Scanning, Surface Flatness, Defect Detection, Quality Control, Correlation Analysis

1. INTRODUCTION

Construction and infrastructure inspectors require timely and accurate geometric information for effective construction quality control. Defects occurring on a construction-site result in rework costs of up to 6-12% of construction cost [1-3]. However, current geometric data collection methods adopted by construction inspectors require large amount of manual surveying works, and the collected data are sparse, which sometimes does not capture required detailed geometric information. For example, during the construction of a light rail system (Figure 1), inspectors need to check whether the running surface is flat enough. Currently they use a straight edge to measure the surface flatness, and count the number of defective regions on the surface, which deviate from the as-designed flat surface more than a predefined threshold. Such manual measurements are physically demanding and time consuming. In addition, sparse measurements may not cover the surface with enough sampling density and might miss small defects between measured places.

Laser scanning technology can deliver dense point clouds in minutes with mm-level accuracy. It enables inspectors to create detailed 3D as-built models for virtual inspection, hence is a promising alternative for geometric data collection. For above mentioned surface flatness evaluation, laser scanning provides a way for sampling a running surface with high density of 3D point measurements in minutes, hence may substantially improve the data collection efficiency and the comprehensiveness of the surface flatness assessment. However, most technical specifications provided by manufacturers only specify single point accuracy of point clouds or the modeled surface accuracy, which specifies the average deviation of a modeled surface from the actual object surface [4, 5]. That information does not inform data users whether a specific size of surface defects can be extracted from the point cloud, and what are the performance limits of a defect-detection algorithm for detecting small surface flatness defects. Currently, few studies focus on developing flatness defect-detection algorithms and characterizing their performances. As a result, 3D data users lack the knowledge about how accurate surface flatness can be assessed using a specific scanner, which algorithms can effectively detect

small defects, and how their performances are influenced by various factors, such as scanning distance and density of point clouds. That knowledge is important for understanding how to properly choose and configure the scanners and algorithms for effective surface flatness assessments.

Figure 1 Light Rail System and Running Surface Flatness Assessment

Focusing on the surface-flatness assessment problem, we developed three algorithms for extracting surface-flatness defects from point clouds, and investigated which factors can influence their surface-flatness defect detection performances. The factors investigated in this paper include scanning technology used (phase-based scanner or time-of-flight scanner), scanning distance, angular resolution of scanned data (density of point cloud), and size of the defects (diameter of a defect and thickness of it). Table 1 groups those factors into scanner internal factors and scanner external factors. We developed a testbed composed of flat boards with various sizes of spots on it, and used that test bed to evaluate one phase-based scanner and one time-of-flight scanner, as well as three surface-flatness defect-detection algorithms. The following sections first present some background research, then describe the tested algorithms and two tested scanners, the test bed and our performance evaluation method, finally discuss the evaluation results and propose issues for future investigation.

Table 1 Factors Influencing the Performances of Surface Flatness Defect Detection

Scanner Internal Factors	Scanner angular resolution (density of point clouds)
	Scanning technology (time-of-flight or phase-based)
Scanner External Factors	Scanning distance
	Selected defect detection algorithms
	Surface flatness defect size (diameter of the defect and thickness of the defect)

2. RELATED WORK

In order to quantify flatness of concrete surface, ASTM (1996) E1155-96 standards and the American Concrete Institute (ACI 1990) 117-90 specifications specifies F-number system [6]. ASTM E 1155 also presents a standard test method for the F-number system [7]. The Face Company, in conjunction with ACI Committee 117, developed a basis for the F-number-system through the use of a patented continuous profileograph machine [8]. This machine is the current standard instrument for slab surface flatness measurement. It measure elevations of points every foot apart, and can generate F-numbers automatically based on elevation measurement results. F-number system uses sparse measurements (0.3 m apart) as inputs of a surface-quality evaluation algorithm, and generates a global surface-quality assessment result, which does not include detailed information about individual surface defects, such as their dimensions and locations. Hence, F-numbers cannot answer some queries of inspectors, such as how many defects exist on a surface, and where they are.

Through a series of case studies, Walsh investigated several metrics for evaluating the flatness and levelness of residential building slabs [9-11]. Studied metrics include the maximum elevation difference, the maximum slope, and the contour pattern of a slab. Similar to F-system based approaches, these metrics can measure the surface quality at a global level, but do not represent information about local surface quality, such as how many defects exist and what their dimensions are. As a result, such surface-quality evaluation methods cannot distinguish a slab with many small defects and a slab with a few large defects [8], while inspectors sometimes still need localization of defects for refined surface

quality control.

Limitations about above discussed studies exist because they rely on sparse measurements sampling on a surface. That is due to the lack of a method, which can densely sample a surface in short time. With dense measurements achieved by a laser scanner in minutes, we expect that inspectors could evaluate the quality of a surface in a more detailed and accurate way. Since few studies investigate how to utilize dense point clouds for detailed surface-quality assessments, we have performed this research.

3. FLATNESS DEFECT DETECTION ALGORITHMS

We have developed three algorithms for extracting surface flatness defects from dense point clouds. The following paragraphs first overview our concept about the algorithm design, and then describe each algorithm sequentially.

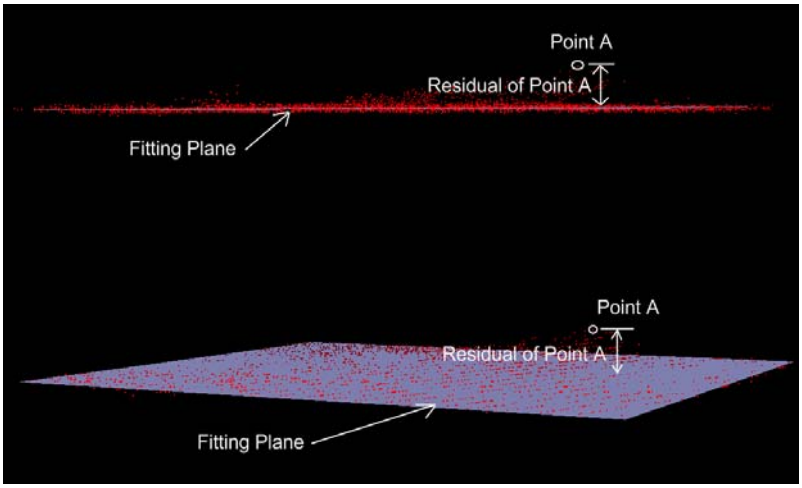


Figure 2 Fitting a plane against point clouds and get the residuals of each point. Top figure shows the side view and the bottom figure shows an isometric view

Without losing the generality, we assume that the evaluated surface is horizontal (for concept illustrations only, all algorithms presented actually do not rely on this assumption). In that case, flat surface defect detection targets identifying surface regions within point clouds where vertical deviations from the reference horizontal plane exceed a specified limit. The reference horizontal plane represents the ideal shape of the surface. In order to detect vertical deviating areas, it is necessary to first set up the reference plane, and then calculate the vertical distances between the collected surface points and the reference plane. The reference plane can be obtained by fitting a plane against all surface points, and can also be extracted from a CAD model, or specified by the user. In this research, we fit a plane against all points on the evaluated surface as the reference plane, since most parts of the surface are flat. In addition, investigations in this research indicate that the impacts of small defects on the parameters of the extracted plane are very small, hence would not substantially change the defect detection results. Regarding calculating the vertical distances between the actual surface and reference plane, one option is to calculate the vertical distance of each point in the point cloud to the reference plane, and another option is to fit local planes against small subsets of points and calculate the vertical distance of the centers of local planes to the reference plane. Because noise in point clouds data influence the deviating pattern detection, it is necessary to first filter out noise for better defect detection accuracy. One option for noise filtering is to filter the point cloud, and another option is to filter the vertical deviations after the vertical deviation calculation.

Assuming that the data noise is Gaussian noise, we have designed two algorithms for flatness defect detection by calculating the vertical distance of each point in the point cloud to the reference plane. Figure 2 shows a patch of surface point cloud collected by a laser scanner and the corresponding plane fit to the existing points. Using the fitted plane as a global reference, flatness-defect detection algorithms can calculate the vertical distance of each point to the plane and determine how much each point is deviating from the plane representing a flat surface.

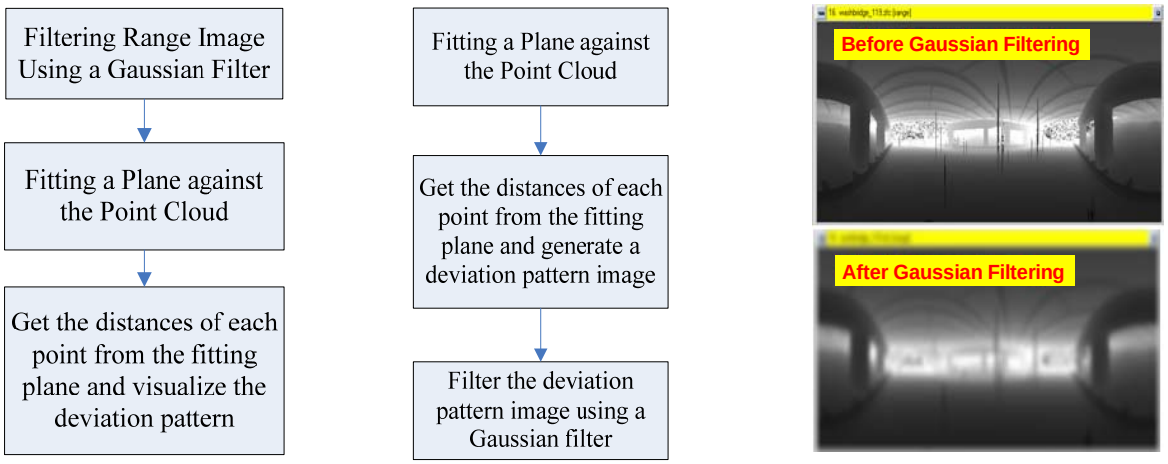


Figure 3 (a) Flow chart of algorithm 1; (b)Flow chart of algorithm 2 (c) Effects of a Gaussian filter on a range image

Figure 3(a) shows the flowchart of the first algorithm (algorithm 1). To remove the Gaussian noise in the point cloud, this algorithm first filters the range image using a Gaussian filter. Figure 3(c) illustrates the Gaussian filtering result of a range image. Gaussian filtering smoothes a range image, and reduces the random noise at the expense of blurring high frequency components of data. For Gaussian filtering, a parameter, the sigma, needs to be manually determined. This sigma value of a Gaussian filter defines how much weight is put on the space near the blurred point: the bigger this value, the more emphasis will be placed on spaces around the blurred point. After Gaussian filtering, the algorithm updates 3D coordinates of all the points in the point cloud according to the adjusted range values, and fits a plane against the updated point cloud. Then, the algorithm calculates the distances of all updated points to the fitting plane as the indicators of the deviations from the reference surface. Figure 3(b) shows the flowchart of the second algorithm (algorithm 2). First, it fits a plane against all points in the point cloud. Second, it calculates the distances of all points to the fitting plane and generates a deviation image indicating the deviations of all points from the fitting plane. Finally, it filters the deviation image using a Gaussian filter and gets the final deviation image. Compared with algorithm 1, algorithm 2 filters an image labeled by the deviation values instead of range values so that instead of directly removing the Gaussian noises from the point clouds by blurring the range images, it uses the raw data to calculate vertical deviations from reference plane and filter the noise carried in the deviation values.

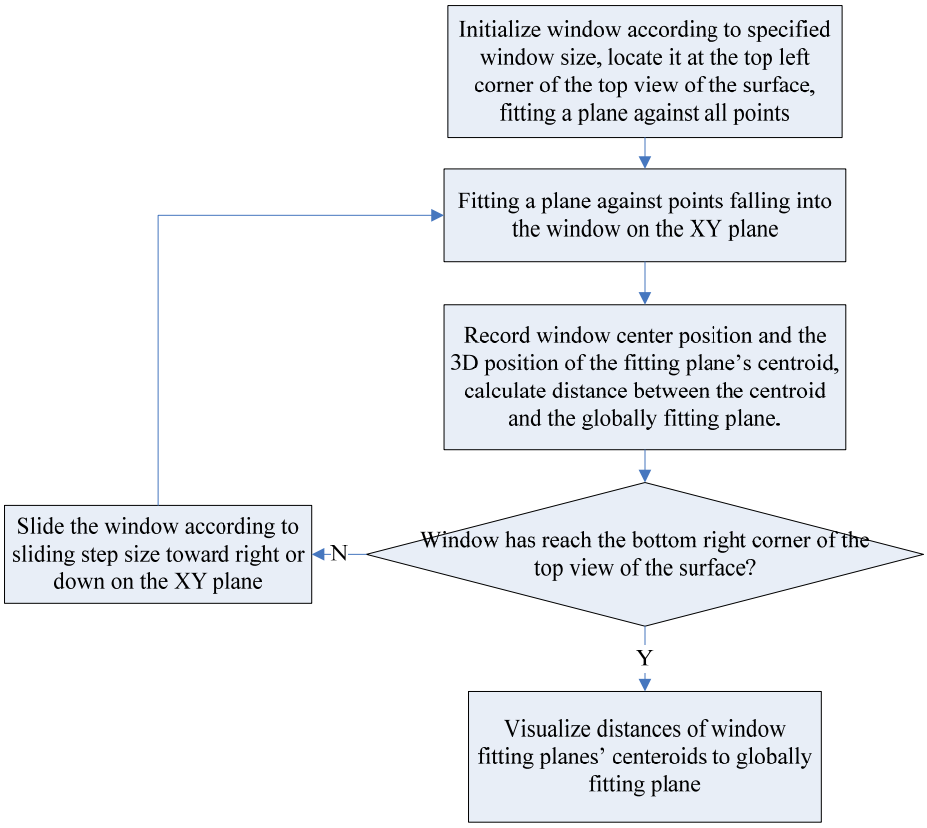


Figure 4 Flowchart of algorithm 3

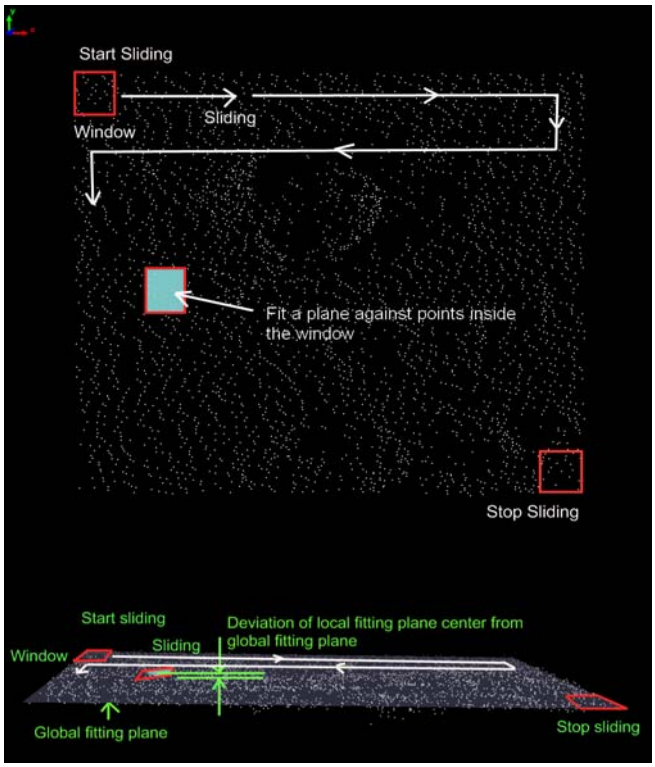
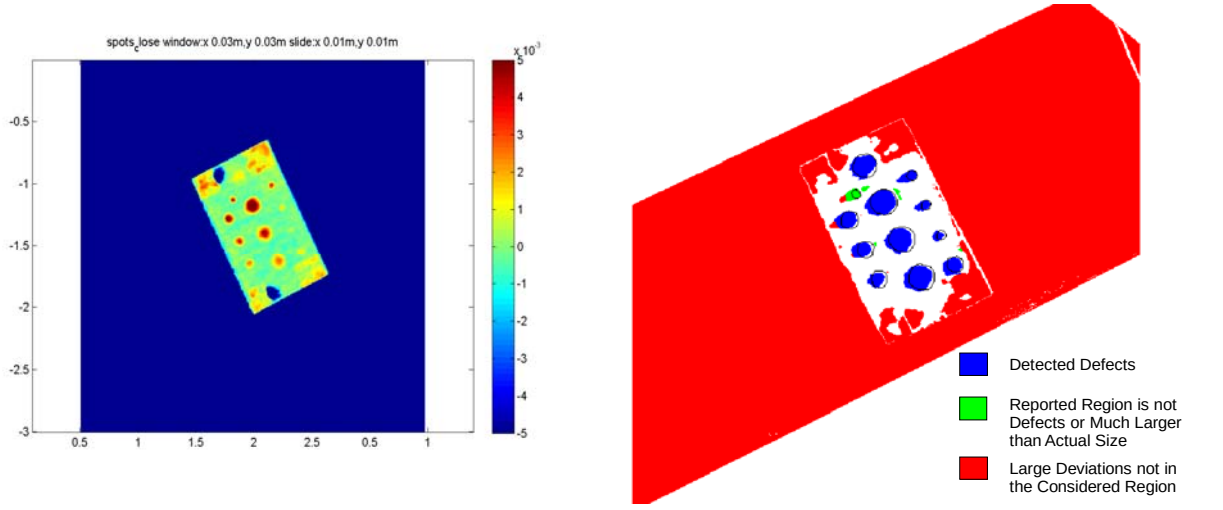


Figure 5 An example of sliding window algorithm (Upper: top view; bottom: side view)

Besides reducing the data noises through Gaussian filtering, we tested another approach for the noise reduction. Instead of calculating the vertical distance of each point in the point cloud to the reference plane, an algorithm can first fit local planes against small subsets of points and then calculate the vertical distance of the centers of local planes to the reference plane. Figure 4 shows the flowchart of an algorithm based on this idea (algorithm 3). This algorithm deals with data noises using a sliding window to explicitly specify the neighborhood for geometric feature calculation. In this case, the geometric feature is a local plane fitting against points inside a sliding window. The local plane is calculated from a number of in-window points and the Gaussian noise in each single point is removed during the process of local plane fitting. By using the center of plane fit within a local sliding window for vertical deviation calculation, the algorithm is utilizing data patterns of a group of points instead of single points for defect detection. Figure 5 illustrates the idea of algorithm 3. Upper part of this figure is the top view of a patch of flat surface point cloud and the bottom part is the side view. First, this algorithm initializes a window at the top left corner of the surface according to the specified window size. Second, the algorithm checks which points on the surface fall into the current sliding window, and then fits a local plane against those points and calculates the vertical distance between the center of local plane and the global surface fitting plane. This calculation is repeated for all possible window positions by sliding the window in a raster-scan fashion across and down the surface. Each window position produces a single measurement of deviation from the global reference plane at that window's center, all of which are combined to form the deviation image.



(a) *(b)*

Figure 6 (a)Deviation Patterns Generated by Defect Detection Algorithms (b)Blue: detected defects, red: data out of the detection region, green: flat surfaces which are incorrectly identified as defects

All three tested algorithms generate a deviation image depicting deviations of the surface from the reference plane at densely sampled locations on the surface. An example of deviation image is shown in Figure 6 (a). With that deviation image, the process of detecting defective region is: if a deviation value at a location is larger than a predefined threshold,

then a defect detection algorithm will report that location as defective. Repeating that process on all pixels in a deviation image produces a binary image indicating which locations on the surface are defective. For evaluating the performance of a defect-detection algorithm, we overlapped that binary image with manually labeled binary defect image serving as the ground truth. Then, we generated algorithm evaluation results, which show that the tested algorithm correctly detects which defective regions and generates how many false alarms. Figure 6 (b) shows an example of such an algorithm evaluation result.

4. TESTED SCANNERS

We experimented with two commercially available scanners. The first scanner is an amplitude modulated continuous wave (AMCW) scanner. This scanner emits laser signals which are continuous wave, and utilizes the phase difference between the out-signal and the reflected signal to calculate the time-of-flight, then deduces the distance from the object surface to the scanner based on that derived time-of-flight. The range of this scanner is 53.5 m. It has very high data collection rate (125,000 points per second), and can finish a 360°x310° scan with typical range image resolution (10,000 points per 360°, collect 10,000 points for one 360° rotation) within 3 minutes and 22 seconds. The second scanner is a time-of-flight scanner. This scanner emits laser pulses, and measures the time intervals between out-pulse and corresponding in-pulse for calculating distances. The range of it is 200 m, and its data collection rate is about 5,000 points per second. The noticeable feature of this scanner is its long range and small absolute errors for single points (1.4 mm standard deviation at 50 m). Table 2 shows the tested scanner configurations in this research. For the AMCW scanner (scanner 1), we only tested the scanning distances of 3 m, 6 m and 10 m. For the time-of-flight scanner (scanner 2), we also tested scanning distance 20 m in addition. To study the influences of angular resolution of data (point cloud density) on the defect detection performance, for both scanners, we tested two resolution options at some distances. For scanner 1, two tested resolutions at scanning distances 6 m and 10 m are 0.036° and 0.018°. These two resolutions are typical “high” resolution and “super high” resolution of this scanner. For scanner 2, two tested resolutions at 10 m are 0.014° and 0.007°. Since 0.014° and 0.018° are close, we use these two data sets from two scanners for preliminary comparison of the two tested scanners’ data. For both scanners, we use 1.5m as the instrument height during data collection.

Table 2 Tested Configurations of Tow Scanners

	Phase-Based Scanner (scanner 1)	**Time-of-Flight Scanner (scanner 2)**
Scanner Height	1.5m	1.5m
Tested Scanning Distances	3 m, 6 m, 10 m	4 m, 6 m, 10 m, 20 m
Tested Angular Resolutions	At 6m and 10m, try two angular resolution options: 0.018°, 0.036°	At 10 m, tried two angular resolution options: 0.014°, and 0.007°

5. TEST BED FOR EVALUATING FLATNESS DEFECT DETECTION ACCURACY

In order to investigate how those factors listed in Table 1 impact the defect-detection performance, we designed a test bed. Since it is intuitive that the smaller the identified flatness defect, the better the flatness defect detection performance is, the basic idea of this test bed is to identify the smallest surface defect that can be identified by a specific scanner-algorithm configuration (angular resolution, scanning distance, algorithm parameter setting) for achieving the best defect detection performance. The scanned targets are three flat boards shown in Figure 7. Repeated manual measurements show that those boards are very flat (less than 1 mm across the whole board surface). Two of those flat boards have clay spots of various sizes on them. The third board is a clean board representing the surfaces without any defects, and they are used for training defect detection algorithms. As discussed before, all flatness defect detection algorithms need a predefined threshold to determine whether the deviation value at a location indicates a defective point.

Figure 7 Flatness Defect Detection Performance Evaluation Test Bed

According to our investigations, that threshold value is different for different algorithms as well as different scanner configurations. We determined those threshold values using the "reference board" approach. We used the maximum threshold value, which will not produce any false alarms of defects on the clean board for each scanner-algorithm configuration. In that way, the algorithms will use the threshold values, which make them as sensitive as possible to defective regions without producing false alarms on the normal surfaces.

Table 3 lists tested defect sizes; these sizes are based on our investigations about usual defect sizes occurred in Architecture/Engineering/Construction (AEC) domain. For defect detection performance evaluation, we scanned those boards at various distances using two scanners (at some scanning distances, two angular resolution options were tested), then used three defect detection algorithms to process the data, and compared the algorithms' defect detection performances in terms of number of detected defects and number of false alarms.

Table 3 Factors Influencing the Performances of Surface Flatness Defect Detection

2D Shape Size		**Thicknesses**
Circular Defects with Diameters of:	3 cm	1 mm, 3 mm, 5 mm, 7 mm
	6 cm	1 mm, 3 mm, 5 mm, 7 mm
	10 cm	1 mm, 3 mm, 5 mm, 7 mm
	15 cm	1 mm, 3 mm, 5 mm, 7 mm
	20 cm	3 mm
50 x 50 cm Rectangular Spot		5 mm

All three tested algorithms generate a deviation image for a data set. However, deviations generated by different algorithms may not be deviations at exactly the same physical locations. For example, algorithm 1 and 2 generate deviation values for each point in the point cloud, while algorithm 3 generates deviations at grid points, which are the centroids of a sliding window and may not be on the exact locations of the points in the point clouds. To make the defect detection results of these three algorithms comparable, we decided to interpolate the deviations generated by all algorithms for generating deviations on a set of predefined 2D grids. Since all collected dense point clouds sample the flat surface with very small step sizes (less than 5 mm at scanning distance of 10 m), we assume that simple linear interpolation will not have substantial impacts on the defect detection results. The grid size chosen in this research is 5 mm, which is 1/6 of the diameter of the smallest defect and can generate at least 36 grid points on the smallest tested defect for detailed flatness-defect detection.

This research uses a "blob" based approach for evaluating the flatness-defect detection performances of all algorithms. All algorithms generate a binary image indicating defective points on the surface. If a group of "defective points" are gathering together and form a connected block, then they form a detected "defect blob". If the percentage of the area of a "defect blob" overlapping with a manually labeled defective region is larger than a threshold, then the performance evaluation algorithm regards that defect blob as a "valid" defect blob. If the percentage of an actual defect region covered by valid defect blobs is larger than a threshold, then the performance evaluation algorithm regards that defect as "detected defect". We call the percentage threshold for determining a "valid" defect blob as "valid blob percentage", and the percentage threshold for determining a "detected defect" as "detected defect coverage percentage". In this research, after several experiments, we found that using 50% as the "valid blob percentage" and 10% as the "detected defect coverage percentage" can produce the best defect detection results for all three tested algorithms. Using this blob-based approach, each algorithm can report a number of defect blobs. After using the "valid blob percentage" threshold to filter out blobs which do not have substantial overlaps with actual defective regions, the performance evaluation algorithm checks the percentage of each actual defect spot covered by "valid blobs" and report whether it is a detected defect based on "detected defect coverage percentage". At the same time, the performance evaluation algorithm records the number of falsely alarmed blobs to evaluate the capability of an algorithm for differentiating actual defects and data noises.

This paper presents flatness detection performance in the form of performance maps. A performance map is a table containing '+'s and '-'s for indicating the detection results of defects studied ('+' for detected defects, '-' for defects not detected). A column of that table shows a series of detection results for defects of the same radius but of different thicknesses, and a row of that table shows a series of detection results for defects of the same thickness but of different diameters. Please refer to the results shown in the next section for details of such performance maps. In some cases, due to time constraints or sensor limitations, the data for a given defect was unavailable. Those cases are indicated by a "U" in the performance maps.

6. RESULTS AND DISCUSSION

For each scanner configuration listed in Table 2 (a configuration is a tuple of selected scanner, scanning distance, and angular resolution), we tested multiple parameter settings of the three flatness-defect detection algorithms. For algorithms 1 and 2, sigma values 1, 2, 3 are tested. For algorithm 3, window size 3cm, 4cm, 5cm, 6cm, and 7 cm are tested. The best defect detection performance achieved by each algorithm for each data set is reported along with the parameter setting of the algorithm achieving that best performance. This paper defines “better” detection performance in terms of two aspects: the number of defects detected, and the number of false alarms. For any two detection results, we regard detecting as many defects as possible as the first priority, and reducing the number of false alarms as the second priority. .This approach provides conservative quality control, which is needed in this case since missing a defect may cause more loss in the future than costs associated with spending some efforts on checking false alarms right now. Hence, for a specific experiment setting (scanner, scanning distance, angular resolution), for each algorithm, we report the parameter setting of that algorithm detecting the largest number of defects. If two parameter settings of an algorithm tie in terms of the number of detected defects, we report the one with fewer false alarms. Figure 8 shows the results of the data from the tested time-of-flight scanner, and Figure 9 shows the results of the data from the tested AMCW scanner.

Distance	**Flatness Defect Detection Algorithm**																							
4m (0.014° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**22**	**7**	**Algorithm 2: Sigma 1**				**FP**	**17**	**7**	**Algorithm 3: Window 3cm**					**FP**	**9**	**6**
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50
	1	+	+	+	+	U	U	1	+	+	+	+	U	U	1	-	+	+	+	U	U
	3	+	+	+	+	+	U	3	+	+	+	+	+	U	3	-	-	+	+	+	U
	5	+	+	+	+	U	+	5	-	+	+	+	U	+	5	-	-	+	+	U	+
	7	-	+	+	+	U	U	7	-	+	+	+	U	U	7	-	-	+	+	U	U
6m (0.014° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**3**	**3**	**Algorithm 2: Sigma 1**				**FP**	**15**	**7**	**Algorithm 3: Window 3cm**					**FP**	**10**	**6**
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50
	1	-	+	+	+	U	U	1	-	+	+	+	U	U	1	-	-	-	+	U	U
	3	+	+	+	+	+	U	3	+	+	+	+	+	U	3	-	+	+	+	+	U
	5	+	+	+	+	U	+	5	-	+	+	+	U	+	5	-	+	+	+	U	+
	7	-	+	+	+	U	U	7	-	+	+	+	U	U	7	-	-	+	+	U	U
10m (0.014° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**10**	**9**	**Algorithm 2: Sigma 1**				**FP**	**9**	**10**	**Algorithm 3: Window 3cm**					**FP**	**11**	**8**
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50
	1	-	-	+	+	U	U	1	-	-	+	+	U	U	1	+	+	+	+	U	U
	3	+	+	+	+	+	U	3	-	-	-	+	+	U	3	-	+	+	+	+	U
	5	-	+	+	+	U	+	5	+	+	+	+	U	+	5	-	-	+	+	U	+
	7	-	-	+	+	U	U	7	-	-	+	+	U	U	7	-	-	+	+	U	U
10m (0.007° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**10**		**Algorithm 2: Sigma 2**				**FP**	**10**		**Algorithm 3: Window 3cm**					**FP**	**11**	
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50
	1	U	+	+	+	U	U	1	U	+	+	+	U	U	1	U	+	+	+	U	U
	3	U	+	+	+	U	U	3	U	+	+	+	U	U	3	U	+	+	+	U	U
	5	U	+	+	+	U	U	5	U	+	+	+	U	U	5	U	+	+	+	U	U
	7	U	-	+	+	U	U	7	U	-	+	+	U	U	7	U	-	+	+	U	U
20m (0.014° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**3**		**Algorithm 2: Sigma 1**				**FP**	**7**		**Algorithm 3: Window 3cm**					**FP**	**15**	
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50
	1	U	-	-	-	U	U	1	U	-	-	-	U	U	1	U	-	+	+	U	U
	3	U	+	+	+	U	U	3	U	-	+	+	U	U	3	U	+	+	+	U	U
	5	U	+	+	+	U	U	5	U	+	+	+	U	U	5	U	-	+	+	U	U
	7	U	+	+	+	U	U	7	U	+	+	+	U	U	7	U	-	+	+	U	U

Legend

Detected: + | Not Detected: - | Unknown: U

Figure 8 Best Surface Flatness Defect Detection Results Achieved Under Various Scanner Configurations of the Time-of-Flight Scanner Using Three Defect Detection Algorithms

6.1. Influence of Scanning Distance

Figures 8 and 9 show that while keeping the angular resolution at 0.036° (AMCW Scanner) /0.014° (Time-of-Flight Scanner) and changing the scanning distance, the defect detection performances of all algorithms degrade with the scanning distance. However, the specific performance changing trends of the two tested scanners are slightly different. For the time-of-flight scanner, all algorithms have slightly better performance at closer distances. At 20 m, the performance is still roughly at the same level of that at 10 m. It is interesting to find that the defect with 6 cm diameter and 7 mm height is missed at 10 m by algorithms 1 and 2, but is detected at 20 m by the same algorithms. Close inspection of the deviation images indicate that at 10 m, that small diameter defect is covered by detected blobs. However, since it is too close to the board's boundary, the boundary noisy data influence the detection accuracy and generate much larger blobs than the actual size of the defect. That fact causes failures during the "blob validity checking" (covering blobs are not valid because their sizes are more than two times larger than the defect). These results indicate that the tested time-of-flight scanner's data can keep a reasonable performance level within 20 m while the angular resolution is set as 0.014°.

Distance	**Flatness Defect Detection Algorithm**																							
3m (0.036° angular resolution)	**Algorithm 1: Sigma 2**				**FP**	**0**	**6**	**Algorithm 2: Sigma 2**				**FP**	**24**	**2**	**Algorithm 3: Window 3cm**				**FP**	**10**	**1**			
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50		
	1	-	-	+	+	U	U	1	+	-	+	+	U	U	1	-	-	+	+	U	U		
	3	+	+	+	+	+	U	3	+	+	+	+	+	U	3	+	+	+	+	+	U		
	5	+	+	+	+	U	+	5	+	+	+	+	U	+	5	+	+	+	+	U	+		
	7	+	+	+	+	U	U	7	+	+	+	+	U	U	7	+	+	+	+	U	U		
6m (0.036° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**2**	**2**	**Algorithm 2: Sigma 1**				**FP**	**9**	**15**	**Algorithm 3: Window 3cm**				**FP**	**10**	**15**			
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50		
	1	-	-	-	-	U	U	1	-	-	-	-	U	U	1	-	-	-	-	U	U		
	3	+	-	+	+	+	U	3	-	-	+	+	+	U	3	-	-	+	+	+	U		
	5	+	+	+	+	U	+	5	+	+	+	+	U	+	5	+	+	+	+	U	+		
	7	+	+	+	+	U	U	7	+	+	+	+	U	U	7	+	+	+	+	U	U		
6m (0.018 °angular resolution)	**Algorithm 1: Sigma**				**FP**	**2**		**Algorithm 2: Sigma 1**				**FP**	**20**		**Algorithm 3: Window 4cm**				**FP**	**1**				
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50		
	1	U	-	+	+			1	U	+	+	+	U	U	1	U	-	+	+	U	U		
	3	U	+	+	+	+		3	U	+	+	+	+	U	3	U	+	+	+	+	U		
	5	U	+	+	+		+	5	U	+	+	+	U	+	5	U	+	+	+	U	+		
	7	U	+	+	+			7	U	+	+	+	U	U	7	U	+	+	+	U	U		
10m (0.036° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**12**		**Algorithm 2: Sigma 1**				**FP**	**12**		**Algorithm 3: Window 3cm**				**FP**	**11**				
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50		
	1	-	-	-	-	U	U	1	-	-	-	-	U	U	1	-	-	-	-	U	U		
	3	-	-	-	-	-	U	3	-	-	-	-	-	U	3	-	-	-	-	-	U		
	5	-	-	-	-	U	-	5	-	-	-	-	U	-	5	-	-	-	-	U	-		
	7	-	-	-	-	U	U	7	-	-	-	-	U	U	7	-	-	-	-	U	U		
10m (0.018° angular resolution)	**Algorithm 1: Sigma 1**				**FP**	**11**		**Algorithm 2: Sigma 1**				**FP**	**21**		**Algorithm 3: Window 3cm**				**FP**	**12**				
	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50	Diameter(cm) / Thickness (mm)	3	6	10	15	20	50		
	1	U	-	-	+	U	U	1	U	-	-	-	U	U	1	U	-	-	-	U	U		
	3	U	-	+	+	U	U	3	U	-	-	-	U	U	3	U	-	-	-	U	U		
	5	U	-	+	+	U	U	5	U	-	-	-	U	U	5	U	-	-	-	U	U		
	7	U	+	+	+	U	U	7	U	-	-	-	U	U	7	U	+	-	+	U	U		

Legend

Detected + Not Detected - Unknown U

Figure 9 Best Surface Flatness Defect Detection Results Achieved Under Various Scanner Configurations of the AMCW Scanner Using Three Defect Detection Algorithms

For the AMCW scanner, using the scanner's default angular resolution 0.036°, the defect detection performances of all three algorithms have a very steep decrease between 6 m and 10 m. As shown in Figure 9, almost no defects can be detected based on the definition of "detected defects" described before. This observation indicates that the tested AMCW scanner might be suitable for detecting small flatness defects in relatively small spaces, and might not be suitable for detecting small flatness defects more than 10 m away from the scanner.

6.2. Influence of Angular Resolution

Figure 8 and Figure 9 also show that increasing the angular resolution from 0.036°(AMCW scanner)/0.014°(time-of-flight scanner) to 0.018° (AMCW scanner)/ 0.007°(time-of-flight scanner) can help in substantially increasing the flatness defect detection performances of all three algorithms. For the time-of-flight scanner, at 10 m, if the users choose to use angular resolution of 0.007°, the only defect missed is the "3cm-1mm" one, while if they use the angular resolution of 0.014°, many small defects will be missed. For the AMCW scanner, at 6m, increasing the angular resolution can help all three algorithms to detect more defects as well, and algorithm 2 can detect all defects from the data set with higher density at 6m. At 10 m, the performance of algorithm 1 increases substantially with the increasing of angular resolution. However, the performances of algorithms 2 and 3 seem do not improve very much with the increasing of the data density.

The tested resolutions of the AMCW scanner are lower than that of the tested time-of-flight scanner. That might partially explain why the overall results from the time-of-flight scanner appear to be better than that from the AMCW scanner: Denser data tend to produce better defect detection results.

6.3. Comparison of Algorithms

For data from the time-of-flight scanner, algorithm 1 and 2 generally perform better than algorithm 3, especially for those defects with small diameters. Closer inspections of deviation maps (due to space limits, we cannot show them in this paper) indicate that most failures of algorithm 3 in detecting small defects are due to two major reasons. First, algorithm 3 tends to generate blobs larger than that generated by algorithms 1 and 2. For small defects, which are only 10 cm away from neighboring defects, there is merging small blobs into larger ones, causing the algorithm not pass the "valid blob" test. Second, some small defects are only 10 cm away from the boards' boundaries, and laser scanned data close to those boundaries tend to be noisier. That causes small blobs covering small defects to be merged with large blobs detected in boundary regions, hence also results in of the algorithm not to pass the "valid blob" test. In brief, algorithm 3 still detects blobs covering all defective regions, however, it tends to get larger blobs than algorithms 1 and 2. This makes the test results sensitive to the distances between defects or the distances between defects and the board boundary.

For data from the AMCW scanner collected at distances 3m and 6m with 0.036° resolution, all three algorithms show similar performances, except that algorithm 1 tends to get less false alarms than algorithms 2 and 3. For 0.018° resolution data of the AMCW scanner at 6m, algorithms 3 and 1 have similar performances. For data collected at 10 m with 0.018° resolution, algorithm 1 performs substantially better than algorithms 2 and 3. These results may imply that the range noise of the phase-based scanner may fit well with a Gaussian noise model, so that directly filtering the range values can effectively improve the detection results than filtering the deviation values afterward.

6.4. Comparison of Laser Scanners

Figure 8 and Figure 9 show that at 10 m, using 0.018°/0.014° resolution (roughly the same level of the data density), the time-of-flight scanner's data can produce better defect detection results than the AMCW scanner for all three algorithms. However, at closer distances (3m and 5m), AMCW scanner can still produce equally good results at 3m and 6m compared with those results from super-high resolution data of the time-of-flight scanner, even in cases within which the data is relatively sparser the one collected through the time-of-flight scanner. The data collection rate of the AMCW scanner used in this research is 100 times faster than the used time-of-flight scanner. Hence, we regard that for small space requiring fast assessments, the AMCW scanner will be more suitable, but for large open space, a time-of-flight scanner may cover a larger area at one scanning location, with an expense of longer data collection time.

6.5. Other Influential Factors

In this research, we also observed several other effects influencing the defect detection accuracy. First, since in the

experiments, all boards lay on the ground horizontally, the incident angle of the laser on the surface is increasing with the increase of the scanning distance. That will result in very steep viewing angles causing sparser point clouds on the surface. We found that such sparser data could cause difficulties to defect detection algorithms, especially to algorithms 1 and 2, which operate on range images. An explanation for these effects is that since a pixel in a smaller range image at a longer distance represents larger physical regions on the board, pixel-based image processing may blur the image too much even with the sigma value as 1. In the experiment, we found that algorithm 3 is less sensitive to such effects, and we suspect that fitting local planes in 3D space might be more robust to data density variances than the approach of filtering noise on range images.

Another related effect caused by incident angle is that for thick defects (thicker than 5mm), at those longer scanning distances (longer than 10m), the detected blobs will have a long “tail”. That “tail” is caused by the fact that a thick spot occludes the surface behind it, and the occluded region contains some noisy data similar to boundary data. Those noisy data produces blobs much larger than the defect’s actual size (especially for small defects), and can cause some failures in passing the “valid blob” test. That may explain why some very thick small defects are not detected while thinner defects are detected in some cases.

7. CONCLUSION

In this research, we developed three flatness-defect detection algorithms, and quantitatively investigated the performances of them. The major findings are four folded. First, at 3m, all three tested algorithms can detect most defects, and increasing the scanning distances can cause defect detection performance degeneration. That performance degeneration is more obvious in the results generated from the data sets of the AMCW scanner. Second, increasing angular resolution can help in increasing the performance of all three algorithms. Third, all three algorithms can detect most defects within 10m, but their performance differs on different data sets, and there is no unique optimal algorithm works best for all data sets. Relatively, algorithm 1 can generate better results in most cases, but in some cases, it might not be the best one. Fourth, within 10m, fast AMCW scanner’s data can generate equally good defect-detection results as the one generated by the data collected by the time-of-flight scanner, while at longer distances, it is necessary to increase its angular resolution for satisfactory defect-detection performances. In future research, we plan to further investigate the impacts of incidental angles, and explore other “detected defect” checking measures for evaluating the algorithm performances in a more objective way. We also plan to conduct more experiments to avoid boundary effects and obtain more data sets for more comprehensive evaluation results free of laser scanned data artifacts (e.g. boundary effects and noisy “tail” of thick defects).

8. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 0420933. NSF's support is gratefully acknowledged. Some of the data collected in this research was done under a funding provided by Pennsylvania Infrastructure Technology Alliance. Any opinions, findings, conclusions or recommendations presented in this publication are those of authors and do not necessarily reflect the views of the National Science Foundation and Pennsylvania Infrastructure Technology Alliance.

9. REFERENCES

[1] Burati, J. and Farrington, J., Costs of Quality Deviations in Design and Construction vol. 19. Austin, TX: The Construction Industry Institute, University of Texas at Austin, 1987.

[2] Patterson, L. and Ledbetter, W. B., "The Cost of Quality: A Management Tool," in Proceedings of Construction Congress I, San Francisco, California, 1989.

[3] Josephson, P. E. and Hammarlund, Y., "The causes and costs of defects in construction: A study of seven building projects," Automation in Construction, vol. 8, pp. 681-687, 1999.

[4] Leica, "Technical Data of Leica HDS4500." vol. 2005: Leica Geosystem, 2005, Retrieved from: http://www.leica-geosystems.com/hds/en/HDS4500_25m_and_53m.pdf

[5] Leica, "Technical Data of Leica HDS6000." vol. 2007: Leica Geosystems, 2007, Retrieved from:

http://www.leica-geosystems.com/corporate/en/ndef/lgs_64228.htm

[6] ACI, "ACI 117-06: Specifications for Tolerances for Concrete Construction and Materials and Commentary," American Concrete Institute 2006

[7] ASTM International, "Standard Test Method for Determing FF Floor Flatness and FL Floor Levelness Numbers." vol. E 1155M: ASTM International, 1996

[8] Stuart, M., "Concrete Slab Finishes and the Use of F-number System." vol. 2008, 2007, Retrieved from: http://www.pdhonline.org/courses/s130/s130.htm

[9] Walsh, K. D., Bashford, H. H., and Mason, B. C. A., "State of Practice of Residential Floor Slab Flatness," Journal of Performance of Constructed Facilities, vol. 15, pp. 127-134, 2001.

[10] Walsh, K. D. and Miguel, G. P., "Method for Forensic Analysis of Residential Floor-Elevation Data," Journal of Performance of Constructed Facilities, vol. 17, pp. 110-117, 2003.

[11] Walsh, K. D., "Performance of Methods for Analysis of Relative Floor Elevation Measurements in Residential Structures," Journal of Performance of Constructed Facilities, vol. 21, pp. 329-336, 2007.