Super-Resolution 3D Laser Scanning Based on Interval Arithmetic

Peter Walecki, and Gabriel Taubin, *Fellow, IEEE*

***Abstract*—Most 3D laser scanners are based on 3D optical triangulation algorithms, where the location of each 3D point is estimated as the intersection of a camera ray and a plane of light projected by a laser line generator.**

Since a physical laser line generator projects a sheet of light of finite thickness, inaccurate measurement and errors result from assuming that the plane of light is infinitesimally thin.

We propose a new mathematical formulation for 3D optical triangulation based on interval arithmetic, where 3D points are only determined within certain bounds along the camera rays, and multiple measurements are used to tighten these bounds.

We propose the Line Segment Cloud as an alternative surface representation to visualize the measurement errors within the proposed framework. We introduce the Iterative Line Segment Tightening algorithm to convert line segment clouds to point clouds, as a preprocessing step prior to surface reconstruction.

We describe how to construct a low cost laser line 3D scanner, where the camera is fixed with respect to the object and the laser line generator is mounted on a high resolution motion platform. We describe a GPU-based implementation where the large number of captured images are processed in real time. Finally, we present some experimental results.

***Index Terms*—superresolution, triangulation, laser scanner, line scanner, interval arithmetic**

I. Introduction

METROLOGY is an ancient and diverse field, bridging the gap between mathematics and engineering. The ability to generate a 3D representation of an object is useful for many applications, including, but not limited to, reverse engineering, industrial design and quality control, artifact preservation, and forensic analysis. For a detailed history of 3D scanning methods based on active illumination, we refer the reader to the survey article by Blais [1].

This paper focuses on one type of 3D scanner, namely the laser line 3D scanner, which is sometimes also referred to as a slit scanner, laser profile sensor, or laser profilometer. Laser line 3D scanners, as well as the vast majority of structured light area 3D scanners, rely on 3D optical triangulation, primarily differing in how they establish the correspondences between projector planes and camera rays. As illustrated in Figure 1, in the traditional model for 3D optical triangulation, each 3D point is reconstructed as the intersection of a camera ray and a plane of light generated by a laser line generator.

Laser line 3D scanners capture one image per projected line, and reconstruct points illuminated by the projected laser line in each image.

There are basically two approaches to the implementation of a 3D scanner based on 3D optical triangulation. In the first approach, the camera is rigidly mounted with respect to the laser line generator, and the object moves along a linear or circular path in small steps. Such an implementation is illustrated in Figure 2-Left. In the second approach, the camera is fixed with respect to the object, and only the laser line generator or structured light pattern projector undergoes motion relative to the camera. This implementation is illustrated in Figure 2-Right. In both cases, after each motion step, the camera captures one image while the laser line illuminates the object. In this paper we explore the second approach, where a laser line generator is mounted on a small linear motion platform which allows for steps much smaller than the laser line thickness, and the camera is fixed to the scanner frame, and is therefore fixed with respect to the object.

Our paper is organized as follows: In Section II, we have a brief description of laser line scanners and the need for a model which more accurately represents the physical nature of the system. In Section III, we describe how we can triangulate line segments which pierce the surface of the object instead of points as in the traditional approach. In Section IV, we discuss the superresolution step in which we shorten the length of the line segments, thereby improve our confidence of the location of the object's surface. In Sections V and VI, we discuss the notion of a Line Segment Cloud and introduce a method for extracting a pointcloud from a line cloud. In Sections VII, VIII, and IX, we discuss details pertaining to the laser scanner built for this paper, and in Sections X and XI we discuss our findings.

Contributions

1) A new mathematical model for laser line 3D scanning based on a two-plane 'sheet' representation of the laser light. This model, which more accurately represents the laser projector because it does not make the false assumption that the plane of light is infinitesimally thin, improves performance without a significant part cost increase.
2) The Iterative Line Segment Tightening (ILST) algorithm to convert line segment clouds into point clouds, as a preprocessing step prior to surface reconstruction.
3) The design and fabrication of a low cost laser line 3D scanner based on 3D printer motion components.
4) A real-time GPU-based implementation where the large number of captured images are not stored, but instead are processed 'on the fly.'

II. Related Work

A number of low cost desktop laser line 3D scanners, such as [2], [3], comprising of a laser line generator, a camera, and a

Manuscript received December 27, 2019.
P. Walecki and G. Taubin are with Brown University.

Fig. 1. 3D Optical Triangulation. Left: Traditional model based on ray plane intersection. Center: A typical image captured by the camera. Right: Proposed model based in interval arithmetic.

computer controlled turntable, have been introduced in recent years to fill the needs of the consumer 3D printer market.

As a result of intrinsic limitations of laser line 3D scanning technology and low quality optics, the resolution of these low cost commercial laser line 3D scanners is not adequate for certain applications in art, entertainment, industrial inspection, reverse engineering, medicine, forensics, and many other advanced applications.

Industrial laser line 3D scanners, used primarily for industrial inspection applications, are also referred to as laser profile sensors or laser profilometers [4].

They include laser line generators with higher quality optics, cameras, and built-in processors. They produce more accurate measurements at a higher acquisition rate, but require relative motion of the object with respect to the sensor in order to produce dense scans. Industrial area 3D scanners based on structured light techniques are capable of producing high resolution 3D models from fewer images. Industrial scanners are available on the market at a price point of one or two orders of magnitude higher than consumer level scanners, which makes them unsuitable for the high volume consumer market, and even for some of the more advanced applications listed above. Only laser line 3D scanners are studied in this paper.

There are many *influence factors* which contribute to the accuracy of a 3D scanner [5], [6], including, but not limited to, the mechanical design of the scanner, the quality of the components used, the calibration regime used to characterize the scanner, as well as the model used to approximate the physical geometry of the scanner. Laser line 3D scanners are often limited by the quality of the components used in their construction – for instance, the laser and cylindrical lens used do not generate an ideal plane of laser light, which contributes to errors and ambiguities in the detection and reconstruction algorithms used, as optical triangulation assumes that the light is planar. Higher quality components do produce a better approximation of a plane of light and hence yield higher quality results, but the use of such components may make the overall system cost prohibitive. High-end commercial laser projectors are capable of producing a line a few microns wide [4], however, [7] asserts that the response of individual pixels often results in a laser profile 'several pixels' wide, even with a laser projector adjusted to generate a line with sub-pixel width.

[8], [9], [10] recognize that conventional models for 3D scanners are inadequate due to the incorrect representation of the laser line. These papers aim to improve the performance of 3D scanners by proposing models which more accurately represent the physical nature of the projected light. [8], [9] propose improved models for the lens distortion of their projectors. [8] acknowledges that the planar model is not sufficient and proposes using a conicoidal surface to model the light instead. However, these papers do not model the non-zero thickness of the light, which contributes to errors in the reconstructed model. Our approach models the thickness of the laser light using two bounding surfaces, and also represents the uncertainty in the location of the surface of the object by means of a line segment cloud, which can be later tightened by moving the laser by a small distance and acquiring another image. They also can only reconstruct one point of data per scan line, a side effect of the assumption that the light has zero thickness. Our approach allows for the reconstruction of data for *all* of the illuminated pixels in a scanline.

Furthermore, the traditional scanner model *always* contributes to reconstruction errors unless the point selected for triangulation lies on the calibrated laser plane as illustrated in

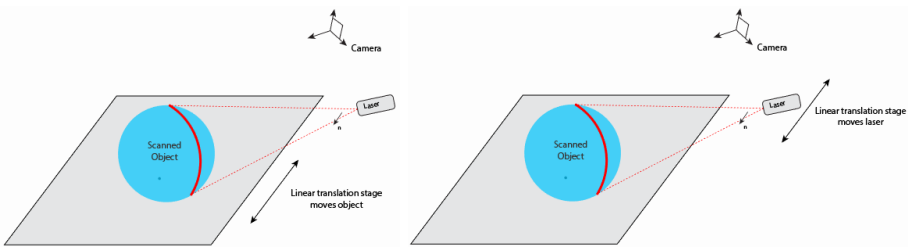

Fig. 2. Left: Typical hardware setup of a 3D Laser Scanner. The position of the laser and the camera are fixed, and the object moves with respect to the scanner. Right: The setup used in our approach. The position of the camera and object are fixed, and the laser is moved with respect to the camera.

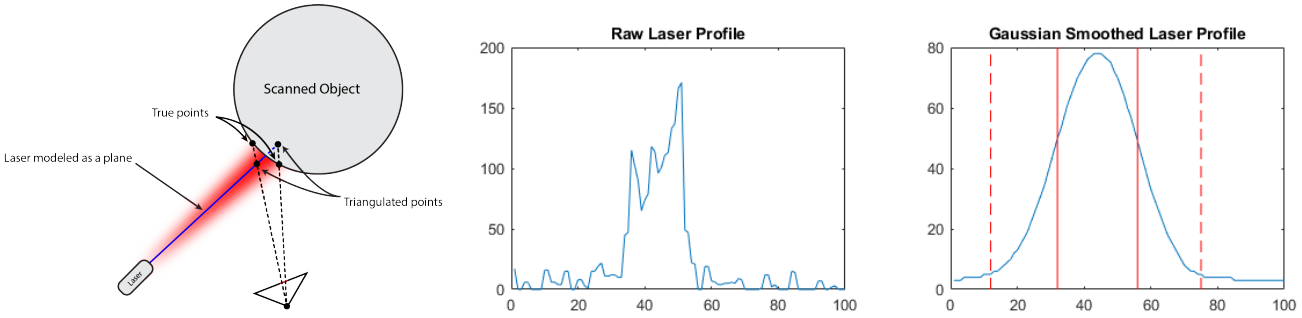

Fig. 3. Left: The traditional scanner model will *always* contribute to reconstruction errors (that is, the distance between the true object points and the triangulated points) unless the selected laser point lies on the calibrated laser plane. Center: Typical laser profile as observed by the camera, before filtering. Note the noise present which makes laser identification tricky. Right: Gaussian smoothed laser profile, with conservative detection for calibration (dashed red lines) and conservative detection for triangulation (solid red lines).

Figure 3-Left. Any errors or bias in the estimated peak position of the profile directly contribute to reconstruction errors.

Naidu and Fisher [7] investigate a number of methods for picking the peak intensity and analyzes their errors for various object geometries. [11] proposes a centroid estimation technique for sub pixel detection of the laser stripe. [11] acknowledges that the source emits a "widened" beam, however they do not model this phenomena but rather assume that the profile observed by the camera will be approximately Gaussian. However, this assumption does not hold true if the object is not locally smooth - the object may occlude part of the profile which will cause a bias in their proposed estimator. Our approach avoids this issue entirely by not relying on the implicit assumptions that the object must be smooth or that the light profile is Gaussian. Furthermore, our approach eliminates the need for accurate peak position estimators as we can use *all* illuminated points to reconstruct data instead of just a single point. [10] proposes a method for peak extraction for scanning reflective metal objects using laser profilers by modeling the reflection and scattering of the laser, however it is unclear how their estimator behaves near depth discontinuities.

III. Triangulation Based On Interval Arithmetic

Laser line generators do not project infinitely thin mathematical lines. They project lines of a non-zero width.

The region of 3D space illuminated by a laser line generator or by a line made of projector pixels can be described as a slab bounded by two surfaces. In this paper, we propose to model the bounding surfaces as two planes.

We propose a new mathematical formulation for 3D optical triangulation based on interval arithmetic, where 3D points are not deterministically reconstructed as the intersection of one projector plane and one camera ray, but they are only determined along the camera rays within the bounds defined by the intersection of the two planes mentioned above and the ray.

Figure 1-Left illustrates the traditional formulation, where the projected line is modeled as a plane, each illuminated pixel detected in the image defines a ray, and the intersection of the ray and the plane determine a 3D point which is assumed to be on the surface of the object. Figure 1-Center is an typical image captured by the laser scanner described in Section VII, showing that it is not clear which illuminated pixels should be used to define the rays, since the width and intensity profile of the illuminated pixels along each row varies from place to place. This 'peak ambiguity' is further illustrated in Figure 3-Center, where the laser profiles is noisy and has a rather flat top, when it should ideally be Gaussian. Figure 1-Right illustrates the new formulation, where the projected laser line is modeled as a slab bound by two planes. Each image pixel which is fully illuminated defines a ray which intersects the

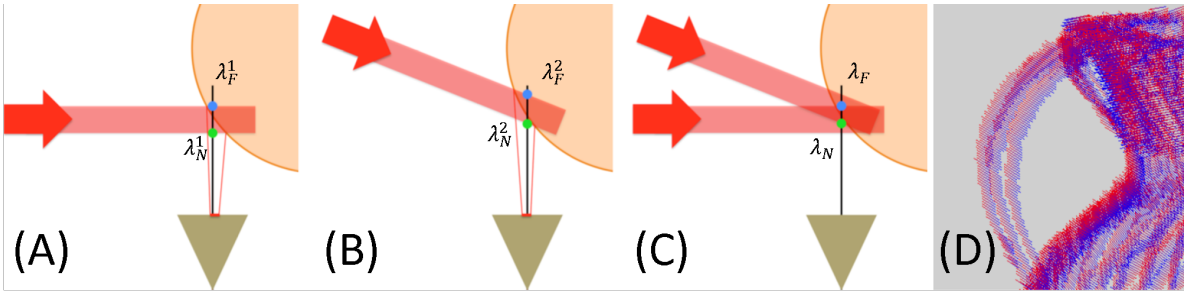

Fig. 4. Principle of 3D optical triangulation super-resolution. A: Example of a line segment triangulated along a single camera ray. B: Each laser projector position results in different bounds along the same ray. C: The intersection of the ray and the surface belongs to the intersection of the two intervals. D: Visualization of a Line Segment Cloud. For each camera pixel illuminated by the laser, there is one camera ray, and one corresponding line segment, resulting from the intersection of the thick laser line with the ray. The segment endpoint closest to the camera was painted blue, and the other end was painted red, with a gradient in between.

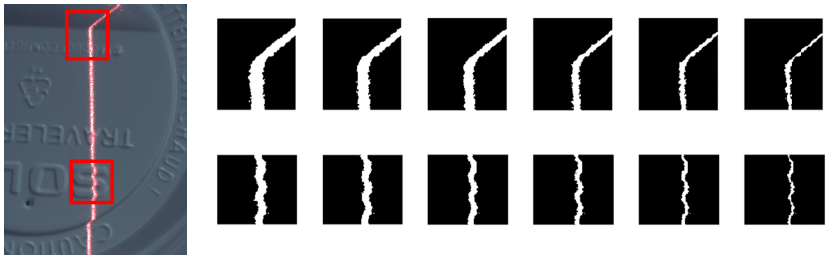

Fig. 5. Left: Raw image of an object illuminated by the laser projector, with ROIs marked by red boxes. Right: Sequences of binarized 'synthetic laser' images. The first image is a binary image of the recovered laser line from a single acquisition. The subsequent images show pixels which were illuminated in the all of the preceding frames. As the laser is translated, the effective width of the synthetic laser line decreases, as if a thinner laser line projector had been used. This can be repeated until the thickness reaches a lower bound determined by the size of the motion step.

two planes. The two intersection points define an interval along the ray which contains a 3D point where the ray intersects the surface.

We aim to address the aforementioned issues with the traditional model by introducing a two-plane model for the laser line projector, which more accurately represents the nature of the laser light and allows for the processing of multiple illuminated laser points per cross-section.

Explicitly, we model the laser as the region between two planes, which correspond to the edges of the laser sheet. Each illuminated pixel detected in the image plane, with pixel coordinates (u_1, u_2) corresponds to a 3D point p in world coordinates lying on the object's surface.

Using the pinhole camera model, the illuminated pixel and the 3D point are related by the well known perspective projection equation $\lambda u = K(Rp + T)$, where $u = [u_1, u_2, 1]^t$ is the representation of the pixel in homogeneous coordinates, λ is a scalar value corresponding to the distance between the focal point of the camera, K is the 3×3 camera intrinsics matrix, and R, T are the rotation matrix and translation vector describing the pose of the camera respectively. The values of K, R, and T are intrinsic to the camera and its location in the world, and are determined by calibrating the camera. The perspective projection equation can be rewritten as

$$p = \lambda v + q, \tag{1}$$

where $v = (R^t K^{-1} u)$ is a known 3D vector, and $q = -R^t T$ is a known 3D point. Note that in practice, the value of u is dependent on the lens distortion parameters of the camera, which must be compensated for by an image distortion removal subroutine [12]. This is a system of three equations in four unknowns (one for λ and three for p). An additional constraint is added by using the knowledge that the point p must also lie on the laser plane.

By defining the laser plane in implicit form $n^t p + w = 0$ in world coordinates, where n is a 3D normal vector to the plane and w is a scalar parameter, the value of λ corresponding to the intersection is obtained by solving the equation $n^t(\lambda v + q) + w = 0$:

$$\lambda = -(n^t q + w)/(n^t v). \tag{2}$$

An additional calibration procedure is required to estimate the equation of the laser plane, which is described in Section VIII.

In the proposed formulation, for each illuminated pixel u, this set of equations is solved separately for each laser plane, resulting in two values of λ (one for each plane), as shown in Figure 4-A and Figure 4-B. These points form a line segment in space supported by the ray corresponding to the illuminated pixel, which pierces the surface of the object.

This segment S_u can be naturally represented as the set of points

$$S_u = \{p : p = \lambda v + q | \lambda \in [\lambda_N, \lambda_F]\}, \tag{3}$$

where λ_N, λ_F represent the distance to the points nearer to and farther from the camera respectively. Repeating this process for all illuminated points results in an unorganized collection of line segments in space which pierce the object. Shorter line segments correspond to those with a greater confidence in the location of the object's surface. Scanning the entire

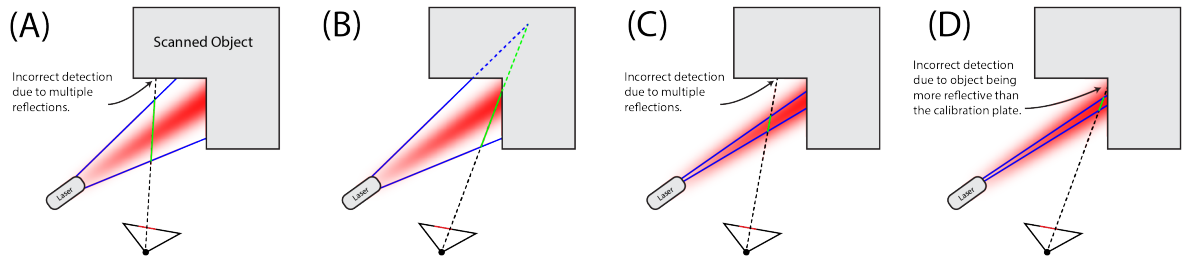

Fig. 6. Diagram illustrating the effects of the threshold levels used for laser calibration and detection during triangulation. A: If a low threshold used for calibration and a low threshold is used for detection, then segments may lie completely outside of the object because pixels illuminated by multiple reflections or subsurface scattering may be incorrectly detected as being directly illuminated by the laser. B: If a low calibration threshold and a high detection threshold is used, we can guarantee that all segments will pierce the object. C: If a high calibration threshold and a low detection threshold is used, we may get incorrect segments for the same reasons listed in (A). D: If a high calibration threshold and a high detection threshold is used, incorrect segments may be generated if the object is more reflective than the calibration object. This is because the detected points may not lie between the calibrated planes.

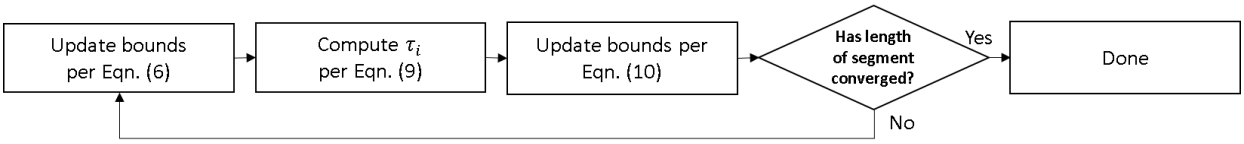

Fig. 7. The proposed ILST algorithm.

object results in a 'line cloud.' A rendering of a line cloud is shown in Figure 4-D.

IV. Super-Resolution

In this paper, we propose a method by which the length of the line segments along the camera rays can be tightened. This tightening represents an increase in the confidence of the location of the surface of the object. We refer to this increase in confidence as 'superresolution.' We refer to the process by which multiple images are used to tighten these bounds as 'the superresolution step.'

The required motion platform can be implemented using low cost 3D printer components, as described in Section VII. More expensive nano-positioning devices are also available for even more precise motion [13], but we don't have access to these devices.

Since one image must be captured for each position of the laser with respect to the object, the many small motion steps result in large data sets and longer processing times. This potential disadvantage is addressed by the real-time GPU-based implementation described in Section IX. Keep in mind, though, that all image super-resolution algorithms suffer from the same problem.

Figure 4 illustrates the principle for a laser line 3D scanner. Here the laser line generator is mounted on small motion platform, the camera is kept in the same pose with respect to the object, and multiple images are captured while the laser moves.

Consider a pixel in the image which has been illuminated by the laser in two different laser positions. These two images produce two line segments $[\lambda_N^1, \lambda_F^1]$ and $[\lambda_N^2, \lambda_F^2]$, as shown in Figure 4-A and Figure 4-B respectively. Note that these line segment will lie on the same ray, as they are both from the same pixel.

Since the true point p lies on both line segments, p must lie on the intersection of the two line segments. This allows us to tighten the bounds of the segment for that pixel ray, as shown in Figure 4-C. The new line segment can be then revised to the smaller segment:

$$S_u = \{p : p = \lambda v + q | \lambda \in [\lambda_N, \lambda_F]\}, \tag{4}$$

where $[\lambda_N, \lambda_F] = [\lambda_N^1, \lambda_F^1] \cap [\lambda_N^2, \lambda_F^2]$. This procedure can then be repeated for an arbitrary number of laser positions in order to further reduce the lengths of the line segments:

$$[\lambda_N, \lambda_F] = [\lambda_N^1, \lambda_F^1] \cap \cdots \cap [\lambda_N^K, \lambda_F^K] \, . \tag{5}$$

By considering pixels which have been illuminated by the projector across multiple images, we are essentially analyzing a 'synthetic' image of a laser line whose thickness is determined by the geometry of the scene, and the laser translation distance between frames. As the laser is translated in small steps, the thickness of the synthetic laser line decreases, until it eventually reaches a lower bound determined by the size of motion step. This effect is shown in Figure 5.

Whether or not a pixel is classified as illuminated is determined by thresholding.

It is of utmost importance that great care is taken when determining the bounding planes of the laser model and when selecting the illuminated points in the acquired image, as incorrectly selecting pixels for triangulation will not guarantee that segments will pierce the surface of the object. A lower threshold should be used for the calibration of the laser planes, and higher threshold for the actual super-resolution 3D scanning.

The threshold used to determine the equations of the bounding planes should be selected such that the two planes completely encapsulate the laser profile, as shown in Figure 6. Picking planes which only partially cover the width of the beam will result in the creation of line segments which are too short and may result in the reconstruction of segments which do not pierce the surface of the object.

In the scanning phase, pixels not fully illuminated must be ignored, because the reduced intensity may result from a

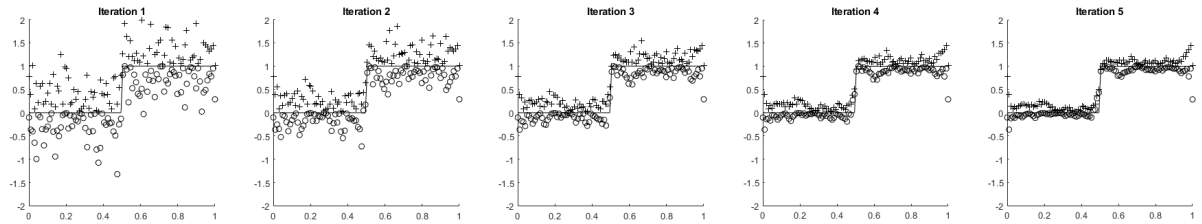

Fig. 8. Several iterations of a 2D simulation of the proposed point cloud extraction algorithm. The ground truth is plotted as a gray line, and the acquired upper and lower bound points are plotted as plus signs and circles circles respectively.

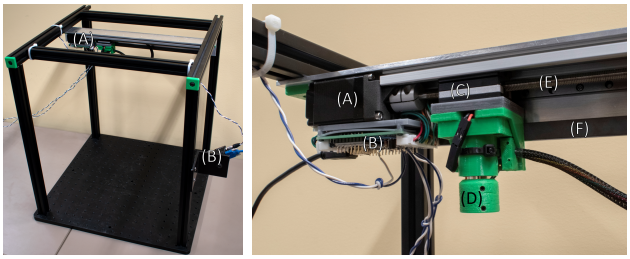

Fig. 9. Left: Laser scanner used for the experiments. A: Carriage assembly. B: System camera. Right: Closeup of the carriage assembly. A: Stepper motor. B: Control electronics. C: Carriage driven by a lead screw. D: Laser line projector mounted to the carriage. E: Lead screw (foreground) and linear slide (background). F: Magnetic encoder strip.

variety of issues which cannot be easily differentiated from each other, such as multiple reflections, specular highlights, subsurface scattering, etc. Triangulating such pixels will result in segments which do not pierce the object. To avoid this, laser points should be selected very conservatively – only points which are clearly illuminated by the laser can be used for the reconstruction.

V. Line Segment Clouds

The 3D scanning algorithm based on the new formulation does not produce point clouds, but rather line segment clouds, where each segment supports one point, but its location along the segment undetermined. We propose the *Line Segment Cloud* as an alternative surface representation and as a tool to visualize the measurement errors within the proposed 3D optical triangulation framework.

VI. Iterative Line Segment Tightening

After a line cloud is generated by the super-resolution algorithm, it is often necessary for further analysis and/or visualization to reconstruct a surface represented as a polygon mesh. Since most current surface reconstruction algorithms operate on point clouds, we propose a novel method to extract point clouds from the line clouds produced by the super-resolution algorithm described in Section IV.

In the super-resolution algorithm described in Section IV the line segments bounds are tightened for each pixel independently of each other, but the length of the line segments does not converge to zero. Selecting the midpoint of each line segment is an arbitrary decision. Instead, we introduce the Iterative Line Segment Tightening (ILST) algorithm, which further tightens the line segment bounds for each pixel as a function of the line segment bounds of neighboring pixels, guaranteeing the convergence of each line segment to a point. The ILST algorithm is as efficient as Laplacian Smoothing, but includes a nonlinear step which makes it preserve features, as in Median Filtering. However, the ILST algorithm applies to ordered pairs of values, rather than single values.

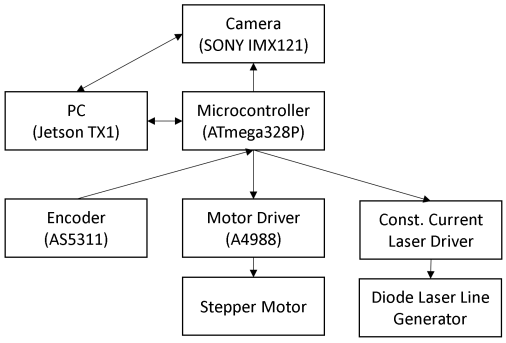

Fig. 10. System topology.

Each camera pixel illuminated by the moving laser line in at least one image results in a corresponding line segment. Another way of looking at the segment cloud produced by the super-resolution algorithm is as a depth map with two depths per pixel, for the subset of illuminated pixels.

For each pixel location i we consider the pixel locations in a neighborhood i^* of i. For simplicity, we assume that i^* is the $N \times N$ neighborhood of the pixel i, however the same formulation is applicable to other neighborhoods. Each illuminated pixel j in this neighborhood has an associated with a line segment with a far point or 'upper bound' λ_F^j, and a near point or 'lower bound' λ_N^j. Note that $i \in i^*$, that is, we include the case $j = i$.

The ILST algorithm is a simple loop comprising the following three inner steps, which are repeated for a fixed number of iterations, or until a convergence criterion is met. The inequality constraints $\lambda_F^i > \lambda_N^i$ are satisfied at the beginning of the algorithm. The three inner steps are designed to iteratively shrink the length of the segments so that they all converge to a point, while maintaining these inequality constraints. In the first step the following first line segment bound updates are computed for every pixel i

$$\begin{cases} {\lambda_F^i}' &= \min_{j \in i^*} \lambda_F^j \\ {\lambda_N^i}' &= \max_{j \in i^*} \lambda_N^j. \end{cases} \tag{6}$$

Since the inequality constraint may not be satisfied after the first step, in the second step the second line segment bound updates are computed as

$$\begin{cases} {\lambda_F^i}'' &= (1-t_i)\lambda_F^i + t_i {\lambda_F^i}' \\ {\lambda_N^i}'' &= (1-t_i)\lambda_N^i + t_i {\lambda_N^i}', \end{cases} \tag{7}$$

where the parameter $0 \leq t_i \leq 1$ is calculated as follows. If ${\lambda_F^i}' > {\lambda_N^i}'$ we set $t_i = 0.5$. Otherwise, if ${\lambda_F^i}' \leq {\lambda_N^i}'$, we solve for the value of $0 \leq \tau_i \leq 1$ corresponding to the crossover point, where

$$(1-\tau_i)\lambda_F^i + \tau_i {\lambda_F^i}' = (1-\tau_i)\lambda_N^i + \tau_i {\lambda_N^i}', \tag{8}$$

and set t_i to the midpoint $t_i = \tau_i/2$:

$$t_i = \frac{1}{2}\left(\frac{\lambda_F^i - \lambda_N^i}{\lambda_F^i - \lambda_N^i - {\lambda_F^i}' + {\lambda_N^i}'}\right). \tag{9}$$

Since the first two steps do not assure that all the segments shrink to a point, in the third step we shrink the segments by a constant amount and update the bounds as follows:

$$\begin{cases} \lambda_F^i &= (1-\epsilon)\,{\lambda_F^i}'' &+ &\epsilon\,{\lambda_N^i}'' \\ \lambda_N^i &= \epsilon\,{\lambda_F^i}'' &+ &(1-\epsilon)\,{\lambda_N^i}'', \end{cases} \tag{10}$$

where ϵ is a small positive number.

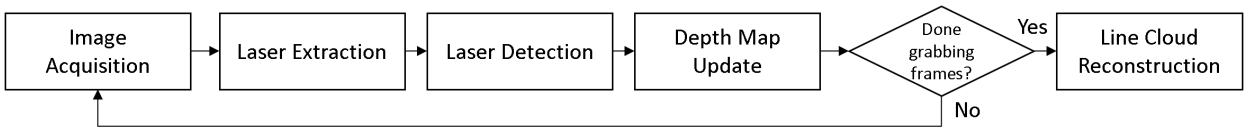

Fig. 11. Reconstruction workflow.

A flowchart showing the steps of the ILST algorithm is shown in Figure 7. Figure 8 shows several iterations of a 2D simulation of this algorithm. To generate simulated upper and lower bounds, Gaussian noise was added to a periodically sampled ground truth function. Note that the proposed approach quickly converges and preserves step and tangential discontinuities.

After a point cloud is generated using the ILST algorithm, point normals need to be calculated for each point. This is done by taking the normal vector of the least squares fit plane to the k nearest neighbors of that point. k is typically on the order of $10-50$. An existing surface reconstruction algorithm is used to fit a surface to the point cloud.

VII. Linear Scanner Construction

In lieu of purchasing an expensive industrial linear motion platform, we designed and fabricated our own, utilizing low cost motion control components commonly used in 3D printers. The system used to test our proposed approach is pictured in Figure 9. An overview of the system topology is shown in Figure 10.

To ensure structural rigidity, aluminum extrusion were used to create the frame of the scanner. A $12.5\,\mathrm{mm}$ thick aluminum optical breadboard was used as the baseplate. The overall system footprint is $46\times46\,\mathrm{cm}$. The effective scan volume is constrained by the depth of focus of the camera and lens. Larger scan volumes could be realized by using wide angle lenses and/or multiple cameras. For image acquisition, an industrial USB3 8.8 MP color CMOS camera was used in conjunction with a $F1.4$ $12.5\,\mathrm{mm}$ focal length lens. The working distance of the camera and lens combination was approximately $40\,\mathrm{cm}$. The laser line generator is a low cost and low quality version intended for hobbyist projects, comprising of a red laser diode and a plastic lenticular array.

This line generator is mounted on a 3D printed carriage, which is then mounted on a linear slide attached to an L-shaped aluminum bracket for rigidity. A separate commercially available low-backlash linear axis driven by a stepper motor and lead screw is used to move the carriage along the linear slide.

Motion control was achieved using a stepper motor driver, an AS5311 magnetic encoder IC [14], and a multi-pole magnetic strip with pole lengths of $2.0\,\mathrm{mm}$. The encoder has a resolution of 2^{12} steps per pole, resulting in a theoretical linear resolution of $\frac{2.0\,\mathrm{mm}}{2^{12}} = 0.488\,\mu\mathrm{m}$. This setup allows for closed-loop control of the laser position, thereby reducing positioning errors due to missed steps, backlash, friction, etc. Mechanical limit switches were mounted at the ends of the motion assembly, which allows for precise homing of the carriage upon startup, as is usually done in 3D printers.

A USB interface on the closed loop stepper motor driver board was used for communication between the host PC and the scanner.

VIII. Calibration

For camera and laser calibration, a COGNEX 40×30 checkerboard with square size of $6.35\times6.35\,\mathrm{mm}$ is used along with the matching corner detection algorithm implemented as part of the COGNEX Vision Library [15].

The OpenCV camera calibration routines are used to estimate intrinsic and lens distortion parameters, as described in [12]. Laser calibration is performed twice: once for the super-resolution approach, and once for the traditional approach. In either case, the R, T vectors of a checkerboard pattern in an image are calculated. For the super-resolution calibration, the laser is then projected onto the plane, and the points on the left and right edges of the laser are extracted. These points are then triangulated to obtain their location in world coordinates using ray-plane intersection. This process is then repeated for several checkerboard positions. This results in two point clouds: one each for the left and right edges of the laser.

Fig. 12. Left: Ground truth image. The color assigned to each pixel corresponds to the depth of the scene along that camera ray. Center: Binary image showing the comparison between the depth map errors of each reconstruction approach. Pixels painted white are pixels for which the superresolution approach has a lower reconstruction error than the traditional temporal approach. Right: Inset showing a detail on the pixels near the edge of the image (marked by the box in the middle image) where superresolution is not as effective.

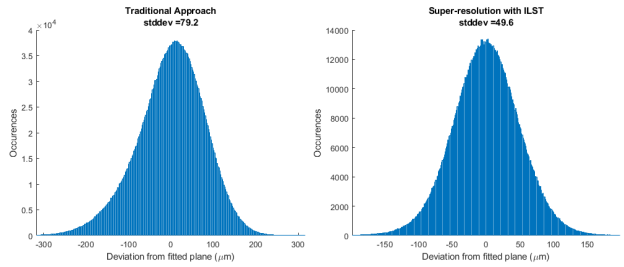

Fig. 13. Left: Error distribution of the traditional reconstruction approach, with a standard deviation of $\sigma = 79.2\,\mu\text{m}$. Right: Error distribution of the proposed approach, with a 37% lower standard deviation of $\sigma = 49.6\,\mu\text{m}$.

Planes are then fitted to these point clouds, generating the bounds of the laser. Each plane is then defined by an implicit equation of the form $n^t p + w = 0$, where n is the normal vector to the plane, p is a point on the plane, and w is a scalar value. This procedure is then repeated for the traditional calibration, but instead of extracting left and right edges of the laser from the images, the peak of the laser profile after image smoothing is used for triangulating points to form a single laser plane. Because the laser is mounted on a linear translation stage in our implementation, the normal vector of the laser plane is constant under stage movement, and the scalar offset component $w(\theta)$ is a linear function dependent only on the motor position θ. This parameter is known since it is set by the user's scan settings.

IX. Realtime Implementation

Because each pixel can be processed independently, the super-resolution algorithm described in Section IV is readily parallelized. Our implementation splits the process into two phases: acquisition and reconstruction.

In the image acquisition phase, images are acquired and processed in order to update the line segment bound estimates. A microcontroller initiates an image capture by sending a trigger signal which is synchronized with the laser motion.

Before the scan starts, two 'depth image' arrays representing λ_N, λ_F for each pixel in the image are allocated as in Equation 3. Each element in these arrays corresponds to the depth of the near and far points respectively of the segment corresponding to that pixel in the image. At the start of the scan, no segments have been recovered. A convenient way to represent an empty set in these arrays is to assign an negative depth value to all of the elements upon initialization.

A basic overview of the image processing pipeline is shown in the flowchart in Figure 11. After an image is acquired by the camera, a laser image is created by extracting the red channel and then compensating for lens distortion using the approach described in [12]. The illuminated pixels in the image are then identified. The endpoints of the segments are triangulated per Equation 2, resulting in new depth values λ_N, λ_F for each of the illuminated pixels. These values are then used to update the master depth images per Equation 5.

After all of the images are processed, a CPU-based implementation of the ILST algorithm is applied.

We have implemented our approach in C++/CUDA which runs on an NVIDIA Jetson TX1 embedded system. We have also used OpenGL to render the reconstructed point cloud live. Our software implementation can process frames at approximately $40-50\,\text{Hz}$, which could be further improved by defining adaptive ROIs/search regions or switching to a B/W camera to reduce the required memory bandwidth. Currently our scan rate is limited by the camera frame rate of $21\,\text{Hz}$, which is determined by the bandwidth of the USB3 bus. Typical scan times are on the order of a few minutes.

X. Results

A. Simulation Results

In order to quantify the error between the recovered data and the ground truth, we devloped software which generates synthetic data sets comprising of artificial images. These artificial images are generated by a raytracer which simulates a scene containing an object to be scanned illuminated by a virtual laser projector.

These simulated datasets were then analyzed using both the proposed superresolution approach as well as the traditional temporal approach. Analyzing the data sets results in a "depth image" where the value assigned to each pixel is the distance from the focal point of the camera to the scene along that pixel's camera ray.

Because the data set has been artificially generated, the ground truth depth image is known. Ideally, the reconstructed depth image will match the ground truth depth image exactly, however in practice, this is not the case due to the aforementioned factors explained in Section II.

We can then quantify the reconstruction errors by first defining the "depth error image" as the difference between the reconstructed depth image and the ground truth.

When processing the data using the temporal approach, one must calculate the sub-pixel position of the laser peak for each row using a peak estimator. For our simulations, we have used a naïve peak estimator (where the peak position is the pixel maximum observed laser intensity), as well as the parabolic and Gaussian approximation estimators which fit parabolic and Gaussian functions to the observed profile in order to estimate the sub-pixel peak position.

We found that the superresolution approach results in reconstruction errors lower than the temporal approach for roughly 99% of the pixels in the image. In Figure 12 we show an image where these pixels are painted white.

In the case where the Gaussian estimator is used, our approach outperforms the temporal approach for approximately 89% of pixels, where the median error was roughly 2x lower than in the temporal approach. For the remaining pixels, the median error of the superresolution approach was roughly 1.6 times that of the temporal approach.

B. Experimental Results

Camera calibration is evaluated by projecting the calculated 3D location of the checkerboard features onto the image plane, and then measuring the RMS error between the projected points and the detected corner locations. The typical RMS error is on the order of 0.22 pixels. For performance evaluation, the reference plane method as described in [16] is used. In

Fig. 14. Left: Scanned object. Center: Mesh reconstructed via the traditional approach. Right: Mesh reconstructed via the proposed approach.

this method, a planar reference object is scanned, and then a plane is fitted to the resulting mesh. The distribution of errors between the scanned points and the fitted plane is then analyzed. The reference plane used in our experiments was a commercially available precision ground $15 \times 15 \times 0.64\,\mathrm{cm}$ plate of O1 tool steel. For the traditional implementation, the Gaussian approximation method as described in [7] was used for laser point detection. After the super-resolution algorithm was run, the midpoint of each of the segments was used to generate a point cloud to compare against the result from the traditional reconstruction approach. For the reference plane test, 1800 images were taken with the laser translated $25\,\mu\mathrm{m}$ between frames. The distribution of errors between the collected points and the fitted plane are pictured in Figure 13-Left and Figure 13-Right for the traditional approach and the super-resolution approach using ILST for point cloud extraction respectively. As expected, the error distribution is Gaussian, with the traditional approach having a standard deviation of $\sigma = 79.2\,\mu\mathrm{m}$ and our proposed approach having $\sigma = 49.6\,\mu\mathrm{m}$. In other test cases, the proposed method consistently provides point clouds with a standard deviation of plane fit errors approximately 10% lower than that of the traditional approach. We also compare surfaces reconstructed using both the traditional and the proposed approaches. To reconstruct the mesh, first normals were calculated by fitting a plane to each point and nearest neighbors. Then, the SSD Surface Reconstruction algorithm [17] was used to generate the mesh. For the first test, a plastic coffee cup lid was scanned. This scan comprised of 400 images with the laser translated $125\,\mu\mathrm{m}$ between frames, computing normals using the 10 nearest neighbors, and using an SSD reconstruction depth of 9. The results are shown in Figure 14. Note how the super-resolution approach produces a mesh with better definition near the embossed text, lower noise in the flat regions, and better definition around the vent at the top of the lid. For the second test, ceramic dish with the fish relief was scanned. This scan comprised of 1500 images with the laser translated $25\,\mu\mathrm{m}$ between frames, computing normals using the 50 nearest neighbors, and using an SSD reconstruction depth of 11. Figure 15 is the reconstruction result of this scan. Note how the super-resolution model produces a result with lower noise.

XI. Conclusion

There are several issues which warrant further research. The proposed extraction algorithm adjusts the bounds based on a small neighborhood. Future surface reconstruction techniques should consider a global approach. Furthermore, our approach assumes that each segment pierces the object once, which is a consequence of the implicit assumption that $O_T >> L_T$, where O_T is the thickness of the object and L_T is the thickness of the laser line. However, it is possible that segments may pierce the surface more than once if the object has thin features.

Unfortunately there isn't a widely accepted standardized way to evaluate the performance of 3D scanners. There is a standard (VDI 2634) which suggests objects for single view systems [18]. ASTM has recently released performance standards for medium range scanners (2m-150m working distance), such as E2938-15 [19], but does not currently have standards for short range scanners (<2m working distance). We are currently working with the E57 committee to help develop a new standard for short range scanners.

The system could also employ several fixed laser line projectors, which do not necessarily have to be mounted on the motion platform. Using multiple projectors could result in higher acquisition rates and/or accuracy, provided that each

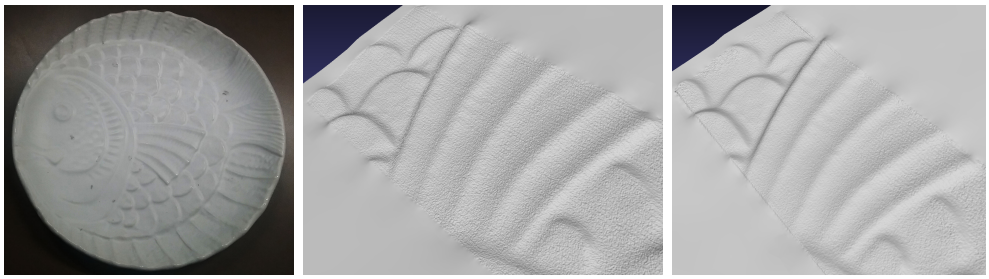

Fig. 15. Left: Scanned object. Center: Mesh reconstructed via the traditional approach. Right: Mesh reconstructed via the proposed approach.

laser line could be identified in the images. We plan to study such an extension as well.

ACKNOWLEDGMENTS

The research reported in this paper was supported by NSF grant IIS-1717355, research gifts by the COGNEX Corporation, and hardware donations by the NVIDIA Corporation.

REFERENCES

[1] F. Blais, "Review of 20 years of range sensor development," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 231–240, 2004.
[2] "Makerbot digitizer desktop 3d scanner," http://store.makerbot.com/digitizer.html, 2013.
[3] "Matter and form 3d scanner," https://matterandform.net/scanner, 2013.
[4] "Specifications : Lj-g5000 series — keyence america," https://www.keyence.com/products/measure/laser-2d/lj-g/specs/index.jsp, 2018.
[5] J. Davis and X. Chen, "A laser range scanner designed for minimum calibration complexity," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, May 2001, pp. 91–98.
[6] R. Leach and C. Giusca, "Calibration of optical surface topography measuring instruments," *Optical Measurement of Surface Topography*, pp. 49–70, 2011.
[7] D. Naidu and R. Fisher, "A comparative analysis of algorithms for determining the peak position of a stripe to sub-pixel accuracy," in *Proceedings of the British Machine Vision Conference (BMVC'91)*, London, 1991, doi:10.1007/978-1-4471-1921-0_28.
[8] R. Yang, S. Cheng, W. Yang, and Y. Chen, "Robust and accurate surface measurement using structured light," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 6, pp. 1275–1280, June 2008.
[9] R. Valkenburg and A. McIvor, "Accurate 3d measurement using a structured light system," *Image and Vision Computing*, vol. 16, no. 2, pp. 99 – 110, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S026288569700053X
[10] H. Wang, Y. Wang, J. Zhang, and J. Cao, "Laser stripe center detection under the condition of uneven scattering metal surface for geometric measurement," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2019.
[11] M. Izquierdo, M. Sanchez, A. Ibañez, and L. Ullate, "Sub-pixel measurement of 3d surfaces by laser scanning," *Sensors and Actuators A: Physical*, vol. 76, no. 1, pp. 1 – 8, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924424798002830
[12] "Camera calibration with opencv," https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, 2018.
[13] "Nanopositioning piezo actuators," https://www.physikinstrumente.com/en/products/linear-actuators/nanopositioning-piezo-actuators/, 2018.
[14] "As5311 linear sensor," https://ams.com/as5311, 2018.
[15] "Cognex vision library," https://www.cognex.com/products/machine-vision/vision-software/cognex-vision-library, 2019.
[16] G. Guidi, M. Russo, G. Magrassi, and M. Bordegoni, "Performance evaluation of triangulation based range sensors," *Sensors*, vol. 10, no. 8, pp. 7192–7215, 2010. [Online]. Available: http://www.mdpi.com/1424-8220/10/8/7192
[17] F. Calakli and G. Taubin, "Ssd: Smooth signed distance surface reconstruction," *Computer Graphics Forum*, vol. 30, no. 7, 2011.
[18] T. Luhmann and K. D. A. Wendt, "Recommendations for an acceptance and verification test of optical 3-d measurement systems," 2000.
[19] "Standard test method for evaluating the relative-range measurement performance of 3d imaging systems in the medium range." [Online]. Available: https://doi.org/10.1520/e2938-15

Peter Walecki is currently pursing a Ph.D. degree at Brown University. He has earned a BSEE degree from Florida Atlantic University. His current research interests include realtime 3D scanning systems and computer vision.

Gabriel Taubin is a Professor of Engineering and Computer Science at Brown University. He earned a Licenciado en Ciencias Matemáticas degree from Universidad de Buenos Aires, Argentina, and a Ph.D. degree in Electrical Engineering from Brown University. At IBM Research from 1990 to 2003, Taubin held various positions, including Research Staff Member and Research Manager. Taubin was named IEEE Fellow for his contributions to the development of three-dimensional geometry compression technology and multimedia standards. During the 2000-2001 academic year, Taubin was Visiting Professor of Electrical Engineering at the California Institute of Technology. During the Spring semester of 2010, Taubin was Visiting Associate Professor of Media Arts and Sciences at MIT. Taubin served as Editor-in-Chief of the IEEE Computer Graphics and Applications Magazine, as a member of the Editorial Board of the Geometric Models journal, and as associate editor of the IEEE Transactions of Visualization and Computer Graphics. Taubin's main research interests span several subject areas, including Applied Computational Geometry, Computer Graphics, Geometric Modeling, 3D Photography, and Computer Vision. Taubin has contributed to the development of efficient, simple, and mathematically sound digital geometry processing algorithms, as well as technologies to enable the use of 3D models for Web-based applications. Taubin has lead research projects on smart cameras, embedded systems, visual sensor networks, digital archaeology, low cost precise 3D scanning systems, industrial applications of 3D scanning systems, and more generally digital fabrication.