An Adaptive Sampling Approach to 3D Reconstruction of Weld Joint

Soheil Keshmiri, Syeda Mariam Ahmed, Yue Wu, Chee Meng Chew, Chee Khiang Pang

Abstract—We present an adaptive sampling approach to 3D reconstruction of the welding joint using the point cloud that is generated by a laser sensor. We start with a randomized strategy to approximate the surface of the volume of interest through selection of a number of pivotal candidates. Furthermore, we introduce three proposal distributions over the neighborhood of each of these pivots to adaptively sample from their neighbors to refine the original randomized approximation to incrementally reconstruct this welding space. We prevent our algorithm from being trapped in a neighborhood via permanently labeling the visited samples. In addition, we accumulate the accepted candidates along with their selected neighbors in a queue structure to allow every selected sample to contribute to the evolution of the reconstructed welding space as the algorithm progresses. We analyze the performance of our adaptive sampling algorithm in contrast to the random sampling, with and without replacement, to show a significant improvement in total number of samples that are drawn to identify the region of interest, thereby expanding upon neighboring samples to extract the entire region in a fewer iterations and a shorter computation time.

I. INTRODUCTION

Fabrication of the offshore oil rig structure is a tedious and labor intensive process. It involves welding a large number of joints over an extended period of time where many of these joints require multiple welding passes (see Figure 1-B). These joints are formed at the intersection of two rigid cylindrical bodies, as shown in Figure 1-A. Furthermore, this is a manual process where the completion of a multi-pass welding for a single joint takes approximately a work shift, on average, by highly skilled welders. This limits the productivity of the oil rigs construction companies since there exists a limited number of welders with such a skill. For instance, most of the welders in Singapore Offshore and Marine industry are foreign workers whose ability to join this market is restricted by the limited number of work permits that is issued on an annual basis. Moreover, this task takes place in outdoor shipyards whose conditions are hostile to human health. In order to improve the productivity of the shipyard, the automation of this fabrication process is highly desirable.

There exists a rich body of research to address various aspects of this task ranging from welding process [2] and sensors [1] to welding power sources and robot programming [4], [3]. However, the preprogrammed robot methodologies [5] are susceptible to failure due to the human error and misalignment of the workpiece that is involved in its preparation. Furthermore, their reliance on the models of the workpieces (e.g., CAD models) limits the scalability of these approaches in fully automated systems. This is mainly due to the fact that workpieces come in varying sizes and shapes and designing their models is a cumbersome and time-consuming process. Moreover, an offshore oil rig structure can have a large number of intersection joints, depending on its size and scale.

In addition, the multi-pass welding process makes the traditional teaching and playback arc welding robots [22] unreliable since every new pass changes the spatial configuration of these joints. To overcome this challenge, many industrial applications employ seam tracking systems [6], [7], [11]. For example, [12] develops a multi-line laser vision sensor to perform the joint tracking in high-speed welding task. [13] introduces a laser-stripe system in automatic welding process in heavy industries. This system achieves satisfactory results even if the welding image is distorted by noise. The seam tracking system in [14] utilizes a laser visual sensor to perform multi-pass metal active gas welding to produce uniform weld beads. However, all these structured laser light seam tracking systems and their commercial counterparts [9], [10] are suitable for straight welding passes. More specifically, their field of view can lose track of the welding seam in presence of curvature. Furthermore, they limit the mane-

![Fig. 1. (A) Common welding joint (B) Same structure after Multi-pass welding - Photo Courtesy Keppel Corp. Ltd.](image-url)
verability of the robotic manipulator since these tracking systems are mounted at the front of the welding torch to provide a good field of view to track the welding seam. In order to address these issues, we propose an adaptive sampling approach to reconstruct the 3D model of the welding joint. This facilitates the planning of the path of the manipulator prior to commencement of the welding task. We opt for a model-free approach that utilizes a point cloud that is generated by a laser sensor. This eliminates any reliance on pre-designed model (e.g., CAD model) of the workpieces. Therefore, scalability is a key feature of our approach in reconstructing the 3D model of workpieces with varying size and shape. We choose laser sensors due to their insensitivity to ambient lighting and their relative light weight. They are deployed in a variety of outdoor applications such as 3D reconstruction of trees [15], [8], airborne scanning [16], mapping and localization [18], [19], [20], and real-time tracking [17].

Our algorithm comprises of two steps. In the first step, we use a randomized strategy to select a number of pivotal candidates to approximate the surface of the volume of interest. We utilize these candidates in the second step as a basis to adaptively sample their neighbors to refine the original randomized approximation to incrementally reconstruct this welding space. Furthermore, we accumulate these pivotal candidates along with their selected neighbors in a queue structure. As a result, every selected sample contributes to the evolution of the reconstructed welding space. Moreover, we keep track of the progress of our sampling via maintaining a list of visited samples to enforce evaluating a sample only once. This prevents our algorithm from being trapped in the neighborhood of accepted samples. However, our approach differs from Tabu search [23] in two fundamental ways: firstly, we do not take an action if the result of the sampling in any iteration of the algorithm does not produce a permissible result. Secondly, we do not set a timeout to free visited samples. In other words, our algorithm does not utilize the concept of short-term memory as described in Tabu search. While formulating our algorithm, we assume that the sensor is facing the workpiece, as shown in Figure 4. It is primarily relocated along its z-axis with a constant velocity during the sensing period. Additionally, it generates data in the form of stream of 3-tuples where the x component of these tuples provide the depth information (i.e., distance of sensed points from the sensor). Moreover, we assume that the length of the translational relocation of the sensor along the z-axis of its frame of reference is known. Additionally, we assume that the minimum and maximum depths of interest are known. The remainder of this article is organized as follows. Section III elaborates on the formulation of our adaptive sampling algorithm. We present our experimental results in section IV. Section V outlines the conclusion and future direction of this research.

III. ADAPTIVE SAMPLING MODEL

We begin by sampling from the surface of the workpiece. More specifically, we translate the sensor along its z-axis to collect spatial information of the workpiece. Furthermore, we stack this data upon its arrival to form a coarse grid of the point cloud that is generated by the sensor during its translational motion. Let S represent the set of point cloud that is generated by the sensor during the sensing phase. This implies that:

\[ S = \bigcup_{i=1}^{N} s_i, \ i \neq j \Rightarrow s_i \neq s_j \]  \hspace{1cm} (1)

where \( s_i = \{ (x_1(i), y_1(i), z_1(i)), \ldots, (x_m(i), y_m(i), z_m(i)) \} \) is the \( p \)th data stream that is generated by the sensor during its translational relocation. N is the total number of these streams. It is possible that \(| s_i | \neq | s_j |\) for two or more consecutive streams due to the shape of the workpiece.

Let \( \tau_k(i) = (x_k(i), y_k(i), z_k(i)) \in s_i \) be the \( k \)th tuple of the \( i \)th sensor reading where \( x_k(i) \) carries the depth information of the \( k \)th data point in \( p \)th stream. Our adaptive sampling algorithm proceeds in two steps, namely, randomized selection of the pivotal candidates and the adaptive reconstruction of the welding space based on these pivots. We elaborate these steps in the following subsections.

A. Randomized Selection of Pivotal Candidates

In this step, we adapt a randomized strategy to select a number of pivotal candidates independently whose tuples form the basis for the reconstruction of the weld joint. We treat the grid of point cloud as a random field where every tuple \( \tau_k(i) \in s_i \), \( \forall s_i \in S, i = 1 \ldots N \), is an independent random variable. Let \( d_{\text{min}} \) and \( d_{\text{max}} \) denote the minimum and maximum depths that we are interested in. We assign a probability to every randomly sampled tuple \( \tau_k(i) \) to represent the spatial correlation of its depth information with respect to \( d_{\text{min}} \) and \( d_{\text{max}} \):

\[ P(\tau_k(i)) = \max \left( \frac{d_{\text{min}}}{d_{\text{max}}} \cdot \frac{x_k(i)}{d_{\text{max}}} \right) \times I(\tau_k(i)) \]  \hspace{1cm} (2)

where \( x_k(i) \) is the depth information of the \( k \)th tuple that corresponds to the \( p \)th stream in the grid of point cloud and \( I(\cdot) \) is a truncation function that is defined over the depth space of these tuples:

\[ I(\tau_k(i)) = \begin{cases} 1 & \frac{d_{\text{min}}}{d_{\text{max}}} \leq 1 \ & \& \frac{x_k(i)}{d_{\text{max}}} \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (3)

In essence, equation 2 assigns a probability of zero to any sample whose depth component falls outside the range of interest. Furthermore, it favors tuples that are closer to the deeper region and/or edges of the volume of the welding joint. This increases the number of neighboring points whose depths are within acceptable range.

We utilize equation 2 to accept a sample as a pivotal candidate if the expected value of its geometric distribution is below a specific threshold (10 trials in our case). It is possible to use this probability to accept a tuple if it exceeds the value of an independent random draw from a uniform distribution [0, 1]. However, such an approach results in an
increase in number of fault rejection of acceptable tuples. Elimination of the fault reject is essential to the performance of our adaptive sampling algorithm. This is due to the fact that every sampled tuple is permanently placed in a list of visited samples once its evaluation is complete. Therefore, any fault reject reduces the possibility of sampling from a potential neighborhood in the point cloud, resulting in an incomplete model of the reconstructed weld joint.

Table I provides details on performance of these two strategies over 10,000 independent and uniformly distributed random points (see Figure 2), where $d_{\text{min}}$ and $d_{\text{max}}$ are selected at random. In this table, $\mathcal{P}$ or precision refers to the percentage of correctly accepted pivots by a particular strategy. Moreover, $R$ or the recall is the measure of correctly rejected samples by these strategies. $F_1$ score or geometric mean of $\mathcal{P}$ and $R$ indicates the measure of accuracy. They are calculated as:

$$\mathcal{P} = \frac{\text{true accept}}{\text{true accept} + \text{false accept}}$$

$$R = \frac{\text{true accept} + \text{false reject}}{\text{true accept} + \text{false reject}}$$

$$F_1 = 2.0 \times \frac{\mathcal{P} \times R}{\mathcal{P} + R}$$

The difference between the total number of valid depths in this table (i.e., the second column entry) is due to the randomness of the process through which these points are generated. Table I shows that the use of $P(\tau_{k}^{(i)})$ as the acceptance criterion results in 23% fault rejection of valid data with 87% overall performance accuracy. However, this issue is resolved if expected value of the geometric distribution of the tuples are used to accept a sample.

We queue these accepted tuples to sample from their respective neighbors in the next step once the specified number of pivotal candidates is satisfied.

### B. Adaptive Reconstruction of the Welding Space

We adopt a four-neighbor strategy to sample from the immediate neighbors of the accepted pivotal candidates with respect to their relative spatial locations within the grid of the point cloud, as shown in Figure 3. Moreover, we treat their depth components as continuous random variables in this step to realize the level of correspondence between their depths and the depth information of their neighbors. This is a valid assumption since the depth component $x_{k}^{(i)} \in \mathbb{R}^+$, $\forall i$, $s_i \in S$ and $d_{\text{min}} \leq d_{\text{max}}$. Let $d_{\text{min}}' \leq d_{\text{max}}'$ where $d_{\text{min}}$ and $d_{\text{max}}$ are the minimum and maximum depths that we are interested in. In addition, this assumption helps capture the variations in the depth information of the sampled points due to the curvature of the surface of the welding joint.

Let $\tau_{k}^{(i)}$ be the most recently accepted sample. Furthermore, let $\tau'$ be one of its immediate neighbor, as presented in Figure 3. We define three proposal distribution functions over the neighborhood of $\tau_{k}^{(i)}$. They are:

1) similarity proposal distribution: is the measure of degree of closeness or similarity between the depth component of an accepted sample to its neighbor. [2], [?] show that the exponential of the negative distance is a reasonable measure of similarity between the given variables. We define our similarity proposal distribution as:

$$C(\tau_{k}^{(i)}, \tau') = e^{-\frac{(\tau_{k}^{(i)} - \tau')^2}{2}}$$

![Fig. 2. Result of sampling from 10,000 independent and uniformly distributed random points using (A) expected value of geometric distribution (B) $P(\tau_{k}^{(i)})$, where $d_{\text{min}}$ and $d_{\text{max}}$ are selected at random (3 and 9, respectively, in these cases). Green-colored samples represent the true accepts.](image)

![Fig. 3. Recently accepted sample $\tau_{k}^{(i)}$ and its four neighbors: $\tau_{k}^{(i+1)}$ (above), $\tau_{k}^{(i-1)}$ (below), $\tau_{k-1}^{(i)}$ (left), and $\tau_{k+1}^{(i)}$ (right).](image)
In other words, \( C(t_k^{(i)}, \tau') \) is an unnormalized Gaussian with variance one that is centered at \( x' \in \tau' \).

2) Gaussian cumulative distribution function to realize how far below the depth value of \( t_k^{(i)} \) the depth of its neighbor resides:

\[
D(t_k^{(i)}, \tau') = \frac{1}{\sqrt{2\pi}} \int_0^{t_k^{(i)}} e^{-x'^2} dx
\]  

(5)

where zero as the lower bound of the integral is due to \( t_k^{(i)} \geq 0 \), \( \forall t_k^{(i)} \in s_i \), \( s_i \in S \). More specifically, depth component of all tuples in the point cloud is non-negative.

3) Complementary proportionality proposal distribution: is the complement of the \( D(t_k^{(i)}, \tau') \) (a.k.a survival function):

\[
E(t_k^{(i)}, \tau') = 1 - D(t_k^{(i)}, \tau')
\]  

(6)

which determines how far above the depth value of \( t_k^{(i)} \) the depth of its neighbor resides. It is apparent that \( E(t_k^{(i)}, \tau') \sim N(\tau', 1) \).

We use these proposal distributions to compute the conditional probability of acceptance of a neighbor \( \tau' \) with respect to most recently accepted sample \( t_k^{(i)} \):

\[
P(\tau' | t_k^{(i)}) = \phi(t_k^{(i)}, \tau') \times I(\tau')
\]  

(7)

where \( \tau' \) is one of the four neighbors as depicted in Figure 3 and \( \phi(.) \) is:

\[
\phi(t_k^{(i)}, \tau') = \begin{cases} 
\max(C(t_k^{(i)}, \tau'), D(t_k^{(i)}, \tau')) & \tau' = t_{k}^{(i)} \\
\max(C(t_k^{(i)}, \tau'), E(t_k^{(i)}, \tau')) & \tau' = t_{k}^{(i-1)} \\
\max(C(t_k^{(i)}, \tau'), D(t_k^{(i)}, \tau'), E(t_k^{(i)}, \tau')) & \tau' \in \{t_{k}^{(i)}-1, t_{k}^{(i+1)}\}
\end{cases}
\]  

(8)

and \( I(.) \) is defined in equation 3. We add \( \tau' \) to the queue of accepted samples if its conditional probability in equation 7 exceeds the value of a random draw from a uniform distribution \([0, 1]\).

Algorithm 1 summarizes our adaptive sampling approach. Termination condition of this algorithm is controlled via tracking the number of iterations (i.e., line 2), number of accepted samples, if specified (i.e., lines 3-4), or if all the tuples in the grid of point cloud is visited (i.e., lines 12-13). Furthermore, lines 5 through 11 corresponds to its randomized selection of pivotal candidates (see section III-A). Whereas lines 14 through 25 illustrate the adaptive sampling step (i.e., section III-B). Function \( \text{Unif}(0, 1) \) at line 22 returns a uniformly drawn value from the interval \([0, 1]\).

**Algorithm 1: Adaptive Sampling Algorithm**

**Data:**

MaxIter : max. no. of iteration  
MaxSamples : max. no. of required samples  
n: required no. of pivotal candidates  
Q: queue of candidates  
G: grid of point cloud  

**Result:**

accepted: finalized accepted candidates  
visited = \[\]  

while MaxIter not reached do  
  if MaxSamples != 0 and accepted.size() == MaxSamples then  
    break;  
  while Q.size() != n do  
    draw a uniform independent sample \( \tau \) from G;  
    if \( \tau \notin \text{visited} \) then  
      if \( \text{Geom}(P(\tau)) \leq \text{threshold} \) then  
        // \( \text{Geom}(\cdot) \) expected value of the geometric distribution  
        add \( \tau \) to Q;  
      else  
        add \( \tau \) to \( \text{visited} \);  
      end  
    end  
    if Q.empty() != true then  
      break;  
    end  
  end  
  \( \tau \leftarrow \text{Q.pop()} \);  
  if \( \tau \notin \text{visited} \) then  
    add \( \tau \) to \( \text{accepted} \);  
    add \( \tau \) to \( \text{visited} \);  
    \( T \leftarrow \text{neighbors of } \tau \);  
    for \( \tau' \in T \) do  
      if \( \tau' \notin \text{visited} \) then  
        if \( P(\tau | \tau') \geq \text{Unif}(0, 1) \) then  
          add \( \tau' \) to Q;  
        else  
          add \( \tau' \) to \( \text{visited} \);  
        end  
      end  
    end  
end

Table II provides specification details of this sensor. We mount this sensor on a prismatic joint to enable its vertical relocation, with a constant velocity, within the specified translational range during the sensing phase. It takes approximately 22.48 seconds for the sensor to scan the workpiece. We only utilize the portion of data streams that falls within 120° field of view of the sensor. This results in every stream of data that is included in the grid of point cloud to have at most 40 tuples (i.e., 120 coordinates information). The entire grid of point cloud comprises 18840 tuples with 11076 of whose depth components within the

| Weight | Power | Range   | Accuracy  | Angular Resolution |
|--------|-------|---------|-----------|--------------------|
| 160g   | 2.5W  | 5.6m×240° | +/- 3%    | 0.352°             |
depth of interest i.e., the interval $[d_{\text{min}} = 17.0, d_{\text{max}} = 21.99]$ unit distance.

Moreover, we implement our algorithm as a Python script. This script communicates with the sensor and its prismatic joint through a remote API to relocate the sensor to scan the workpiece. During the simulation our algorithm runs on a ThinkPad laptop computer with 2.10 GHz Intel Core i7-4600U processor and 7.5 GiB memory, on Ubuntu 14.04 LTS.

We present the performance of our adaptive sampling algorithm through series of illustrations in the next subsection.

B. Results

Figure 5 shows four different views of the original point cloud. It is worth noting that there are several regions in this point cloud where the depth information of the tuples fall within the depth of interest (i.e., $d_{\text{min}}$ and $d_{\text{max}}$). For instance, subplot (B) indicates that the depth component of the tuples whose height is within the interval $[0.025, 0.035]$ approximately equals those that are in $[0.01, 0.02]$ interval. However, only the latter tuples are within the domain of the weld joint. Moreover, this situation applies to some of the tuples that lie on the left and right boundaries of this point cloud, as shown in subplots (A) and (D).

We randomize over this point cloud to collect a number of pivotal candidates. Subplot (A) in Figure 6 shows these candidates (10 in this case). In addition, this subplot supports our claim that equation 2 favors tuples that are closer to the deeper region and/or edges of the volume of the weld joint to increase the number of neighboring points whose depths are within acceptable range. We utilize the spatial information of these candidates as a basis to adaptively sample from their corresponding neighbors within the grid of point cloud.

Figure 6 depicts the reconstruction of the welding joint as a result of the application of our proposal distribution functions. An important aspect of these proposal distributions is their ability to capture the geometrical correspondences that exist among the selected candidates and their neighbors. More specifically, equation 4 helps evolve this reconstructed space along its width, as shown in subplots (B) through (D) in Figure 6. On the other hand, equations 5 and 6 contribute to refinement of the reconstructed welding joint along its height. This effect is illustrated in Figure 6 subplots (C) through (E). The final configuration of the reconstructed weld joint is shown in Figure 7.

Figure 8 presents the history of the first 200 probabilities (green-colored) that are computed by these proposal distribution functions. The red-colored curve in this figure pertains to the uniformly independent random draws from interval $[0, 1]$ during adaptive sampling step (i.e., line 22 in
Algorithm 1. It is apparent in this figure that our proposal distributions are capable of capturing spatial correspondences among neighboring tuples to accept these neighbors if they fall within configuration of the weld joint. More specifically, Figure 8 indicates that all of these probabilities are above 0.95, maintaining a reliable margin to their corresponding uniformly independent random draws from interval [0,1].

C. Comparative Analysis

Table III presents the results of random sampling with replacement on the grid of point cloud. More specifically, it pertains to the case where we do not place the sampled data in the list of visited samples once its evaluation is complete. We use the geometric distribution of the probabilities in equation 2 as its acceptance criterion to prevent this algorithm from accepting a tuple if its depth component does not fall within the range of interest. The "Duplicates" column entry of Table III indicates that the number of duplicated samples that are accepted by this approach constantly increases, reaching to approximately 50% of the entire accepted samples.

In Table IV, we eliminate these duplicates via introduction of list of visited samples. The entries of this table corresponds to the results of the random sampling without replacement in contrast to our adaptive sampling algorithm. The $P(\tau_{k}^{(i)})$ and "geometric" row entries represent the random sampling where we use equation 2 and the expected value of the geometric distribution of a sampled tuple to determine its acceptance or rejection, respectively. On the other hand, the "adaptive" entry shows the performance of our adaptive sampling algorithm with 5 initial pivotal candidates. The column entry "Draws" pertains to the total number of samples that are drawn to complete the extraction of all 11076 tuples that fall within the range of interest. Furthermore, the column entry "Iterations" indicates the total number of iterations that is taken by these approaches to complete this task. A comparison of the values of these entries show a significant improvement that is achieved by our algorithm. The "geometric" random sampling does not falsely reject a legitimate tuple (unlike the random sampling using the $P(\tau_{k}^{(i)})$, as shown in the "False Reject" column entry). However, the "True Reject" entry in Table IV indicates that it spends a significantly longer portion of its operation outside the region of interest during the sampling, as compared to our adaptive algorithm. This, in turn, has a direct impact on overall computation time of these algorithm, as shown in this table. Moreover, the standard deviation of the computation time of these approaches is considerably above one standard deviation (approximately 13.22, with variance 174.87 and mean of 56.40), indicating that the difference in their computation time is significant. More specifically, our algorithm is capable of identifying the region of interest to expand upon neighboring tuples to extract the entire region in a fewer iterations and a shorter computation time.

Table V shows the effect of number of initial pivotal candidates on overall performance of our adaptive algorithm. This table indicates that a change in the number of initial candidates has a direct effect on the completion time of our algorithm. Furthermore, this observation is supported by the standard deviation of the completion time in Table IV. More specifically, the completion time of different setting of the initial number of candidates is above one standard deviation. This also holds true for the number of iterations, as indicated by the "std" entry in Table IV. Although the

| Accepted Samples | Draws | Duplicates | Iterations |
|------------------|-------|------------|------------|
| 2500             | 8434  | 308        | 8435       |
| 5000             | 16932 | 1101       | 16933      |
| 7500             | 25820 | 2619       | 25821      |
| 10000            | 34274 | 4406       | 34275      |
| 11076            | 37844 | 5488       | 37845      |
TABLE I
STATISTICAL DATA ON THE EFFECT OF THE NUMBER OF INITIAL PIVOTAL CANDIDATES ON PERFORMANCE OF OUR ADAPTIVE SAMPLING ALGORITHM

| Draw | Iteration | True Reject | Time (sec.) |
|------|-----------|-------------|-------------|
| Draw (/1000) | Iteration (/1000) | True Reject (/100) | Time (sec.) |
| 4.60 | 1.87 | 0.43 | 42.63 |
| 0.99 | 3.69 | 0.0001 | 1.63 |
| 0.29 | 1.92 | 0.013 | 1.28 |
| 1.051 | 1.113 | 0.901 | -0.115 |
| -0.392 | -0.303 | -1.389 | -1.389 |

entries of Table IV shows a considerable increase in number of draws and the true reject of undesirable tuples, the values of these entries are within one standard deviation. Hence, drawing a conclusion on the effect of the increase of the number of pivots on these entries is not warranted. However, these increases are foreseeable since our adaptive algorithm expands its domain of search based on the neighboring tuples of the accepted samples. As a result, a larger number of pivotal candidates increases the contingency of overlapping neighbors between accepted samples that are already placed in the list of visited samples. This, in turn, causes our algorithm to generate more pivotal candidates as it progresses in time since the queue of sampled tuples is exhausted more frequently without any significant progress, imposing a higher rate of true reject and subsequently more iterations before successful completion of the task.

V. CONCLUSIONS

We present an adaptive sampling approach to 3D reconstruction of the welding joint using the point cloud that is generated by a laser sensor. We show that the introduction of proposal distributions over the neighborhood of a set of randomly selected initial pivotal candidates adaptively refines the original randomized approximation of this region to incrementally reconstruct this welding space. An important aspect of these proposal distributions is their ability to capture the geometrical correspondences that exist among the selected pivots and their neighbors. Our results show a significant improvement in identifying the region of interest via expanding upon neighboring tuples to extract the entire region in a fewer iterations and a shorter computation time, as compared to random sampling methodology. The future direction of this research pertains to the analysis of the performance of our algorithm on real laser sensor to realize the utility of this approach to acquire reliable reconstruction of the welding joint in a real setting. Furthermore, it is crucial to integrate this setting with a real robotic system to study the effect of the various constraints that are imposed on its performance via this integration.

ACKNOWLEDGMENTS

The authors thank the National Research Foundation, Keppel Corporation and National University of Singapore for supporting this research that is carried out in the Keppel-NUS Corporate Laboratory (project title: Productivity Enhancement of Yard Operation, WBS:R-261-507-019-281). The conclusions put forward reflect the views of the authors alone, and not necessarily those of the institutions within the Corporate Laboratory.

REFERENCES

[1] P. Drews, and G. Sarke, Development approaches for advanced adaptive control in automated arc welding, Internal Report XII-970Mechatronics Department, Aachen, Germany, 1986.
[2] L. A. Jones, T. W. Edgar, and J. H. Lang, Images of steele electrode in ar-20% shielding during constant current gas arc welding, Welding Journal, vol. 77, no. 1, pp. 135141, 1998.
[3] J. N. Pires, A. Loureiro, T. Godinho, P. Ferriera, B. Fernando, and J. Morgado, Object oriented and distributed software applied to industrial robotic welding, Industrial Robot: and International Journal, vol. 29, no. 2, pp. 109-117, 2002.
[4] S. Adolfsson, A. Bahrami, G. Bolmsgo, and J. Claesson, On-line quality monitoring in short-circuit gas metal arc welding, Welding Journal, vol. 75, pp. 5973, 1999.
[5] J. Norberto Pires, T. Godinho, and P. Ferreira, CAD interface for automatic robot welding programming, Industrial Robot: An International Journal, vol. 31, pp. 717-726, 2004.
[6] S. Wu, P. Wang, X. Fu, B. Liu, and J. Chen, A novel sensor measuring method of complex tight butt joint for laser welding, IEEE International Conference on Mechatronics and Automation, pp. 25482553, 2009.
[7] L. Zhang, J. Jiao, Q. Ye, Z. Han, and W. Yang, Robust weld line detection with cross structured light and Hidden Markov Model, in Proc. 2012 IEEE Int. Conf. Mechatronics and Automation, Chengdu, China, 2012, pp. 1411-1416.
[8] J. Forest, J. Salvi, E. Cabruiji, and C. Pous, Laser stripe peak detection for 3d scanners. A FIR filter approach, in Pattern Recognition, Proc. 17th Int. Conf. on ICPR, vol. 3, 2004.

[9] B. Ekelof and P. Hansson, Method and apparatus for automatic multi-run welding. Google Patent, 1992.

[10] B. Ekelof and L. Sandstorm, Method for automatic multi-layer welding. Google Patent, 2000.

[11] R. Usamentiaga, J. Molleda, D. F. and Garcia, Fast and robust laser stripe extraction for 3D reconstruction in industrial environment, Machine Vision and Application, vol. 23, pp. 179-196, 2012.

[12] K. Sung, H. Lee, Y. S. Choi, and S. Rhee, Development of a multiline laser vision sensor for joint tracking in welding, The Welding Journal, vol. 88, pp. 79-85, 2009.

[13] A. F. Villn, R. G. Acevedo, E. A. Alvarez, A. C. Lopez, D. F. Garcia, and R. U. Fernandez, Low-cost system for weld tracking based on artificial vision, IEEE Trans. on Industry Applications, vol. 47, no. 3, pp. 1159-1167, 2011.

[14] W. Gu, Z. Xiang, and W. Wan, Autonomous seam acquisition and tracking system for multi-pass welding based on vision sensor, The Int. J. of Advanced Manufacturing Technology, vol. 69, no. 1-4, pp. 451-460, 2013.

[15] B. Gorte and D. Winterhalder, Reconstruction of laser-scanned trees using filter operations in the 3D raster domain, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI, no. 8EW2, pp. 39-44.

[16] W. Tan, J. Li, and Y. Li, Rasterizing airborne laser scanning point clouds by block Kriging, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVII, no. B3b, pp. 687-692, 2008.

[17] D. Held, J. Levinson, S. Thrun, and S. Savarese, Combining 3D shape, color, and motion for robust anytime tracking, Robotics: Science and Systems X, 2014.

[18] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA: The MIT Press, 2005.

[19] R. Zlot and M. Bosse, Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine. Field and Service Robotics, Springer Tracts in Advanced Robotics, vol. 92, pp. 479-493, 2014.

[20] J. Zhang and S. Singh, LOAM: Lidar odometry and mapping in real-time, Robotics: Science and Systems X, 2014.

[21] http://www.coppeliarobotics.com/

[22] A. K. Gupta and S. K. Arora, Industrial automation and robotics. Boston: University Science Press, Ch. 16, 2009.

[23] F. Glover and M. Laguna, Handbook of Combinatorial Optimization, Kluwer Academic Publishers, 1999.