Journal Title
XX(X):1–8
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

A Novel Algorithm of Ship Structure Modeling and Target Identification Based on Point Cloud for Automation in Bulk Cargo Terminals

Author1,Author2...

Abstract
Achieving port automation of machinery at bulk terminals is a challenging problem due to the volatile operation environments and complexity of bulk loading compared to the situations in container terminals. In order to facilitate port automation, we present an method of hull modeling(reconstruction of hull's structure) and operation target(cargo holds under loading) identification based on 3D point cloud collected by Laser Measurement System mounted on the ship loader. In the hull modeling algorithm, we incrementally register pairs of point clouds and reconstruct the 3D structure of bulk ship's hull blocks in details through process of encoder data of the loader, FPFH feature matching and ICP algorithm. In the identification algorithm, we project real-time point clouds of the operation zone to spherical coordinate and transforms the 3D point clouds to 2D images for fast and reliable identification of operation target. Our method detects and complements four edges of the operation target through process of the 2D images and estimates both posture and size of operation target in the bulk terminal based on the complemented edges. Experimental trials show that our algorithm allows us to achieve the reconstruction of hull blocks and real-time identification of operation target with high accuracy and reliability.

Keywords
bulk cargo ship, port automation, point cloud, 3D modeling of hull blocks, target identification

Introduction

As the world economics develops, maritime transportation flourishes and accordingly the demand in capacity of port increases. In order to deal with the increasingly huge volume of transportation, more and more ports attach great importance to the automation at port terminals. Unmanned cargo loading system is a significant constitute of port automation. Therefore, a lot of researchers and engineers focus on developing advanced measurement systems and intelligent control systems for automated cargo loading system and a significant amount of related technologies and methods[1, 5–9] have come up and been applied to facilitate port automation at container terminals. However, compared to rapid development of automated loading system for container ships, the degree of automation in bulk terminals is still low and develops slowly due to the complexity of bulk cargo loading and volatile operation environments. Currently, bulk ship loading still highly relies on manual operation of loading machine drivers, hatch conductors and loading instructors in most of bulk terminals. Two significant problems need to be solved in order to achieve automated loading operation at bulk terminals and so far, researchers have done some exploratory and enlightening work.

Firstly, a reliable measurement system and an algorithm of bulk ship modeling are necessary for acquiring 3D hull structure of bulk ships, especially the detailed structure of hulls under operation. The main stream of sensing modules for measuring ships structure includes 3D depth cameras and lidars, which describes 3D structure of entities in terms of point cloud while the modeling algorithms are varied.[3–9] As cargo ships are often large and the horizon of those sensors have limits, sensors often need to scan different pieces of cargo ships and then put the sectional scanned point clouds together to model the structure of the whole hull blocks. In [2], Guan et al present an automatic registration method of hull blocks based on given CAD model of ships. They use principal component analysis(PCA) for rough registration and then refine the result through iterative closest point (ICP) algorithm. Their method works efficiently but it requires ships' CAD model as reference to register those sectional measured point clouds, which makes it inapplicable when ships' CAD models are not accessible due to common reasons like confidentiality. In [3], Dong et al present a reconstruction method based on a single-wire Laser Measurement System(LMS) and a single axis rotating servo mechanism. The measurement system is mounted on the loader above the middle of cargo ship. The servo mechanism drives the LMS rotating so that LMS can scan from one end of ships to the other, as the position of the LMS is fixed in 3D space and it can acquire point cloud of only one sector once a scan. Then, the hull structure can be built by combining point clouds of different slices of ship's cross sector collected by the LMS with different directions. However, the drawback of this method lies in that it cannot acquire complete structure of hull blocks especially the ships' hatches because the LMS undertakes the measurement from only one position and thus the edges of hatches can be easily sheltered by the lid of hatches. As a result, the effectiveness and accuracy of the method in estimating the size and position recognition of bulk cargo holds are undermined. In [4], Chao Mi et al use four single-wire LMS mounted on the loader above bulk ships. Compared with the method in [3], instead of using only one LMS fixed at one position, they use four LMS scanning linearly from above ship's bow to stern and merge point clouds measured by multiple lidars at different positions and rebuild ship's structure by registration of point clouds. However, they use the ratio of loader's moving speed to the LMS's scanning frequency as interval between coordinates of point clouds of adjacent cross sections and register the adjacent point clouds directly with the estimated interval. Using the roughly estimated interval as final transformation for registration can result in inescapable errors in ship's reconstruction. That's because the instant scanning frequency of LMS can often be different with the designated value and the estimation of the loader's speed also has errors. What's more, the method does not consider ship's movement caused by tide or other outer forces, which also jeopardizes the accuracy of reconstruction modeling.

Second problem to be solved is the real-time identification of operation targets. If the position and size of bulk cargo holds within the operation zone are identified in real time, it would greatly facilitate the loader to locate and load bulk materials into holds automatically. In [3], Dong et al achieve the identification by transforming the 3D point cloud into 2D image and processing the image for identification of cargo holds. They project the point cloud of hull blocks to XOY plane (considered as the horizontal plane) as an 2D image and the pixel values of the image represents the Z coordinates value of original points in the point cloud. They distinguish zone in the deck from zone in the holds by setting a height threshold of pixel values, as they assume that Z coordinate values of points in the holds are always lower than the height of coastline while the values of points in the deck are higher. Then, they use a seed rectangle which starts in the corner of cargo hold and grows until it reaches borders of the two different zones mentioned above and the grown seeding rectangle is recognized as the operation target. In [4], Chao Mi et al apply similar projection method but use a different recognition algorithm. They project the 2D images into both X and Y coordinates and record how many pixels of points in the deck zone are at each X and Y coordinate value respectively. By setting a threshold on the number of pixels of points in the deck zone, their algorithm can detect the rising edges and falling edges of the 1D pixel value, and thus the position of borders between hold zones and deck zones are estimated and the position and size of holds under operation are identified. Methods in [3, 4] work efficiently and provides enlightening thoughts of achieve automation in bulk terminals. However, their identification algorithms are not technically real-time. In order for the dynamic identification, their systems have to scan the whole hull blocks every time before the identification. That's because their identification algorithms are completely based on the knowledge of ship structure of the whole hull blocks. What's more, their methods are based on several unwarranted assumptions, which makes it difficult to apply those methods in the volatile operating conditions of real cases:

1. The plane of bulk hold's hatch is assumed parallel to XOY plane in lidar's coordinate. However, a lot of factors in real operation cases like tide and the uneven distribution of cargo materials in the hold can makes this assumption invalid and thus cause distortion of hatch's size and position in the projection of the 3D structure to the XOY plane;
2. It is assumed that the height of the point cloud in the hold zone is significantly lower than the cost line and the deck, which is not necessarily true when the holds are highly loaded with a huge volume of materials;
3. The borders of hold hatches is assumed parallel or perpendicular to the X coordinate. But such a condition isn't met when bulk ship berths with its main axis not strictly parallel with the coastline of the port.

In summary, in order to facilitate the automation at bulk cargo terminals, it is necessary to reconstruct the 3D structure of hull blocks and then achieve identification of operation targets in real time. Current solutions to part of these two problems provide enlightening insights and exploratory examples but have disadvantages in terms of practicability, accuracy and capabilities of adapting to volatile operating environments in bulk terminals. The main contribution of this paper is concluded as follows:

1. A moving scan and hull modeling system is established, which can reconstruct structure of the whole hulls in an accurate and integrated way compared with existing techniques.
2. A novel algorithm of identifying operation targets is presented, which can achieve the real-time recognition of bulk hatches with high accuracy in volatile operating situations.

Hull reconstruction system

This section describes our laser measurement system which can scans cargo ship from ship's stern to bow and capture 3D structure information of the whole hull blocks and our reconstruction algorithm which can merge all the 3D information to rebuild the 3D model of the whole ship.

Moving-scan Lidar Measurement System

Our system used a multi-wire lidar(Livox Horizon, developed by DJI) with a field of view(FOV) of 81.7 degree horizontally and 25.1 degree vertically. Compared with the single-wire LMS used in [3, 4], the lidar with large FOV allows us to collect information of hull structure with more details and higher resolution. The lidar is mounted on the loader, facing towards berth of bulk cargo ship but not vertically. In order to measure the structure of the whole hull blocks, the lidar with loader moves straightly from above ships' stern to the bow at a uniform speed(0.5 m/s in our experiment) along the rail on bulk terminal. The lidar scans ten times a second, which is enough to collect details of structure of the operation targets for identification. The system is shown in Figure1, with the red rail, green loader, blue bulk ship, purple lidar and its pink FOV.

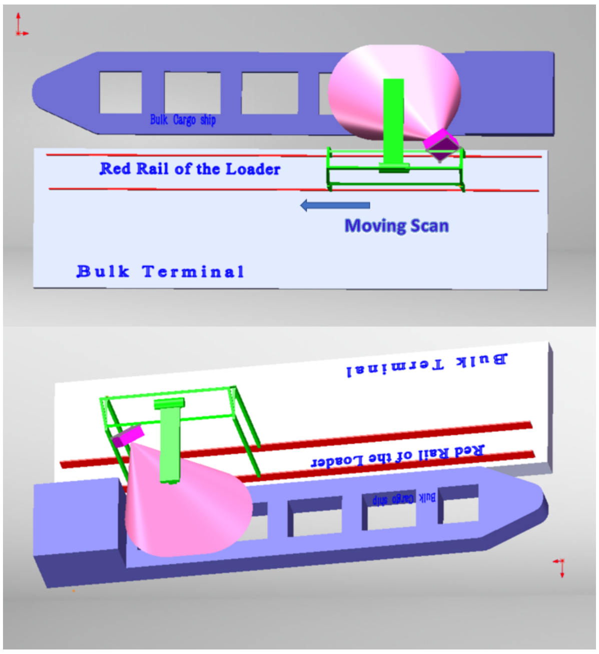

Figure 1. Moving-scan Lidar Measurement System

Hull reconstruction Algorithm

The output of the above measurement system consists of point cloud streams in 10 frame per second. Among the streams, two timely adjacent point clouds are also spatially adjacent. However, coordinates of the two are not coincident due to the movement of lidar and change in ship's posture when scanning. In order to acquire the structure information of the whole ship, we incrementally register pairs of point clouds to the coordinate of the point cloud of the first frame.

To achieve that purpose, we register each pair of adjacent frames, calculate the transformation matrix of coordinates of the two frames. As the difference between two coordinates is mainly caused by the movement of lidar, we do the rough registration first with the estimation of translation movement based on the data of encoder on the loader which can estimate the position of the loader.

$$R_i^{i-1} = R_{calib} \times R_{encoder}{}_i^{i-1}$$

$$R_{encoder}{}_i^{i-1} = \begin{pmatrix} 1 & 0 & 0 & -x_{encoder}{}_i^{i-1} \\ 0 & 1 & 0 & -y_{encoder}{}_i^{i-1} \\ 0 & 0 & 1 & -z_{encoder}{}_i^{i-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$P_i^{'} = R_i^{i-1} \times P_i \quad (2)$$

R_i^{i-1} is the rough transformation matrix from coordinate of i^{th} frame to coordinate of $(i-1)^{th}$ frame. R_{calib} is the transformation matrix from coordinate of first frame to the global coordinate in the port and is acquired through lidar calibration before moving scan. $x_{encoder}{}_i^{i-1}$, $y_{encoder}{}_i^{i-1}$, $z_{encoder}{}_i^{i-1}$ are the linear movement of lidar from $(i-1)^{th}$ frame to i^{th} frame in the directions of global X, Y,Z coordinates recorded by the encoder in the loader. P_i is the original point cloud of hull block at i^{th} frame and $P_i^{'}$ is the point cloud transformed from P_i at the roughly registered $(i-1)^{th}$ frame, as shown in Figure2.

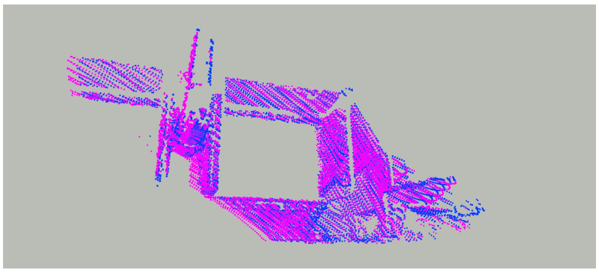

Figure 2. $P_i^{'}$ (purple) and P_{i-1} point clouds(blue)

Though the roughly registered point cloud is closely coincident with the former-frame point cloud, as shown in Figure2, due to the error of data in the encoder and the potential movement of ship during scanning, rough registration is not precise enough to reconstruct the structure of hull blocks of high accuracy: such small errors can accumulate when the transformation matrixes of adjacent frames are multiplied and result in great divergence between point clouds of same part of hull block. Therefore we do the refined registration. With the special scanning mode, Livox Horizon lidar produce the point clouds of the same part of the hull structure with different points and point density at different scanning positions. And thus we need to use the Fast Point Feature Histograms(FPFH)[10] for feature matching in order to acquire refined transformation matrix between $P_i^{'}$ and P_{i-1}.

As the calculation of FPFH is computationally expensive, we only calculate the FPFH of key points of those adjacent point clouds. In order to choose key points, we calculate the normal of each point p_m in point cloud:

$$N_n^{p_m} = \{p_j(x_j, y_j, z_j)|j = 1, 2, \ldots, n\} \quad (3)$$

$N_n^{p_m}$ is the set of n points neighbored to point p_m in point cloud. Then we calculate the covariance matrix based on the neighbor set:

$$\boldsymbol{C} = \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{p_j} - \overline{\boldsymbol{p}}_{\boldsymbol{m}}) \cdot (\boldsymbol{p_j} - \overline{\boldsymbol{p}}_{\boldsymbol{m}})^T, \quad (4)$$

$$\boldsymbol{C} \cdot \boldsymbol{v_k} = \lambda_k \boldsymbol{v_k}, \quad k = 1, 2, 3$$

$\boldsymbol{C}$ is the covariance matrix, $\overline{\boldsymbol{p}}_{\boldsymbol{m}}$ is coordinates of the centroid of $N_n^{p_m}$, $\boldsymbol{v_k}$ is the eigenvector, and λ_k is the corresponding eigenvalue.

The estimated normal at p_m is as follows:

$$\boldsymbol{n_m} = arg \min_{\boldsymbol{v_k}} |\lambda_k|, k = 1, 2, 3 \quad (5)$$

By setting a $Threshold1$ in the norm of normal of points, we select key points from the adjacent point clouds for refined matching:

$$P_{i key}^{'} = \left\{ p_m \in P_i^{'} \quad | \quad |\boldsymbol{n_m}| > Threshold1 \right\} \quad (6)$$

$$P_{(i-1)_{key}} = \left\{ p_m \in P_{i-1} \quad | \quad |\boldsymbol{n_m}| > Threshold1 \right\} \quad (7)$$

Then we calculate FPFH of each point of $P_{i key}^{'}$ and $P_{(i-1)_{key}}$ and match them for refined transformation matrix.

For more details about FPFH matching, please refer to /refR10.

$$R_i^{i-1'} = FPFH\_MATCH(P'_{ikey}, P_{(i-1)_{key}}) \tag{8}$$

$$P_i'' = R_i^{i-1'} \times P_i' \tag{9}$$

To further refine the registration, we use ICP algorithm to acquire further refined transformation matrix:

$$R_i^{i-1''} = ICP(P_i'', P_{i-1}) \tag{10}$$

Through above algorithm, we have the final transformation matrix of each adjacent frames:

$$R_i^{i-1 final} = R_i^{i-1} \times R_i^{i-1'} \times R_i^{i-1''} \tag{11}$$

$$R_i^1 = \prod_{k=2}^{i} R_k^{k-1 final} \tag{12}$$

By multiplying those adjacent matrixes from R_i^{i-1} to R_2^1, we acquire the transformation matrix of coordinate of each frame to the first frame R_i^1 and through those matrix, we incrementally register point clouds of all frames to the coordinate of first frame and thus acquire the whole structure of bulk ship P_{total}:

$$P_{total} = P_1 + \sum_{k=2}^{i} R_k^{k-1 final} \times P_i \tag{13}$$

Figure3 show the whole structure of a bulk ship registered by our algorithm. As shown in the figure, the ship's bow is only partially reconstructed, and that's because of the limit of lidar's FOV and range of loader's movement. However, it does not affect the identification of operation targets since all the structure bulk holds is scanned and rebuilt.

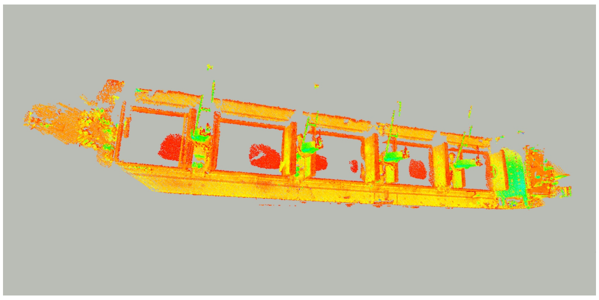

Figure 3. example of our reconstruction algorithm

Identification of operation targets

This section describes our algorithm of identifying the operation targets based on the real-time point clouds of operation zones and the bulk cargo ship's 3D structure built above. Once the reconstruction of the hull blocks' structure is finished, the LMS moves to operation zones and collects real-time point clouds of those zones. By processing those point cloud streams and using the built 3D structure a reference, our algorithm identifies the size and posture of operation targets and thus the slip barrel in the loader can be guided to load bulk materials into the ship's bulk holds automatically.

Dimensionality reduction of 3D Point Cloud

Identification of operation targets relies highly on detection of four edges of the bulk ship's hatch which is rectangular. Meanwhile, as points of the 3D point cloud are not well structured spatially like 2D images, detecting the four edges directly from the point cloud is difficult and computationally expensive. Thus, the 3D point clouds need to be transformed to 2D images for fast and precise target identification in real time. It is a natural thought to project the point cloud to the deck or horizontal plane, as in[3, 4]. However, as the posture of ships can change due to tide and uneven distribution of materials in the holds, the deck plane is often not coincident to the horizontal plane and the plane coefficients are difficult to be estimated. Thus, we use the method presented in[11]. Instead of plane projection, we project the point cloud to spherical coordinate and transform it to a 2D image.

For points in 3D space, their coordinates are defined as $p_m = (x, y, z)$ in Cartesian Coordinate System and $p_m = (\theta, \varphi, r)$ in Spherical Coordinate System. And transformation formulation from Cartesian Coordinate System to Spherical Coordinate System is as follows:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= arctan\frac{y}{x} \\ \varphi &= arcsin\frac{z}{r} \end{aligned} \tag{14}$$

$$\theta \in (-\frac{\Theta}{2}, \frac{\Theta}{2}), \varphi \in (-\frac{\Phi}{2}, \frac{\Phi}{2})$$

Θ and Φ are the horizontal and vertical FOV of the lidar respectively.

The angular resolution is set as $(\Delta\Theta, \Delta\Phi)$, and the resolution of transformed 2D image is $\frac{\Theta}{\Delta\Theta} \times \frac{\Phi}{\Delta\Phi}$. We use r as pixel value of the 2D image. And thus the transformation formulation from 3D point cloud to 2D range image with single-channel pixel value is as follows:

$$\begin{aligned} row_m &= \frac{1}{\Delta\Phi}(\arcsin\frac{z}{r} + \frac{\Phi}{2}) \\ col_m &= \frac{1}{\Delta\Theta}(\arctan\frac{y}{x} + \frac{\Theta}{2}) \\ f(row_m, col_m) &= pixel_m = r \end{aligned} \tag{15}$$

The advantage of spherical projection lines in that compared to plane projection, no matter what posture of bulk ship relative to lidar is, points of point cloud do not overlap and information of shup's structure will not be lost during project. Figure4 show a frame of point cloud of the operation zone and its projected 2D image.

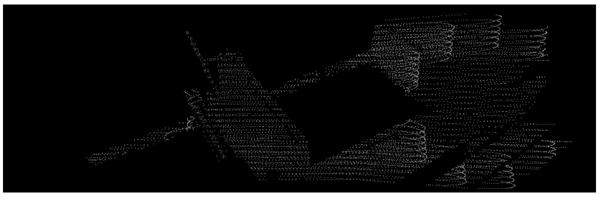

Figure 4. example of projected

Identification of operation targets

As the operation targets are rectangular, the core of identifying those targets lies in detection of four edges of the rectangle. To achieve that purpose, our algorithm convolves the 2D images with Canny edge detect filter.

Firstly, we apply Gaussian filter for denoising:

$$\begin{aligned} f_g(row, col) &= f(row, col) * K, \\ K &= \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned} \tag{16}$$

Then the gradient of the filtered image $f_g(row, col)$ is calculated:

$$\begin{aligned} G_{row} &= \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} * f_g(row, col), \\ G_{col} &= \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * f_g(row, col) \end{aligned} \tag{17}$$

G_{row}, G_{col} are the components of pixel gradients in row coordinate and in col coordinate respectively. And thus we can acquire the norms and directions of image's pixel gradients:

$$\begin{aligned} |\boldsymbol{G}(row, col)| &= \sqrt{{G_{row}}^2 + {G_{col}}^2} \\ \alpha &= \arctan \frac{G_{row}}{G_{col}} \end{aligned} \tag{18}$$

Then we apply Non-Maximum Suppression to $\boldsymbol{G}(row, col)$. As shown in Figure5, according to the direction of $\boldsymbol{G_m}$, the gradient of p_m, we search the neighbored gradients G_{temp1} and G_{temp2} which are calculated through linear interpolation:

$$\begin{aligned} G_{temp1} &= weight \times G_{g1} + (1 - weight) \times G_{g2} \\ G_{temp2} &= weight \times G_{g5} + (1 - weight) \times G_{g6} \\ weight &= \frac{d1}{d} \end{aligned} \tag{19}$$

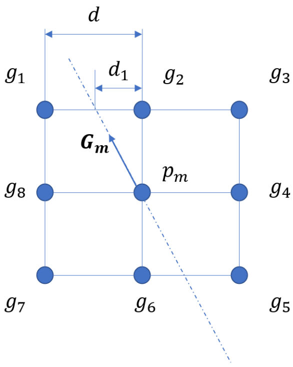

Figure 5. Non-Maximum Suppression

After that, we apply Non-Maximum Suppression for refined edge detection:

$$\boldsymbol{G}_{nms}(row, col) = \begin{cases} \boldsymbol{G}(row, col), & |\boldsymbol{G}(row, col)| = max\{|\boldsymbol{G}(row, col)|, \\ \quad & |\boldsymbol{G}_{temp1}(row, col)|, |\boldsymbol{G}_{temp2}(row, col)|\} \\ 0, & else \end{cases} \tag{20}$$

By setting a $Threoshold2$ on the gradient map $\boldsymbol{G}_{nms}(row, col)$, we apply binarization to $f(row, col)$ and obtain the 2D image $f_e(row, col)$ with only edge features:

$$f_e(row, col) = \begin{cases} 1, & |\boldsymbol{G}_{nms}(row, col)| > Threoshold2 \\ 0, & else \end{cases} \tag{21}$$

Then, we use the method presented in [12] and extract the edges of operation target based on $f_e(row, col)$. This result in two contour and we choose the inner one. Given the fact the straight edges in point cloud remains straight after projection to 2D image, the rectangle made up of the four straight edges becomes parallelogram after projection. The minimum enclosing rectangle of the parallelogram is calculated and four sides of parallelogram can be separately identified based on the distance from points of parallelogram to the different sides of the minimum enclosing rectangle. As there is a one-to-one match between pixels of the 2D images and points of the 3D point cloud, the four sides of parallelogram are remapped to the corresponding point clouds of four edges of the hatch in the operation zone, as shown in Figure6. With the Sample Consensus Function of PCL Library, we calculate parameters of corresponding four straight line as the edges of hatch based on the identified four edges of the point cloud:

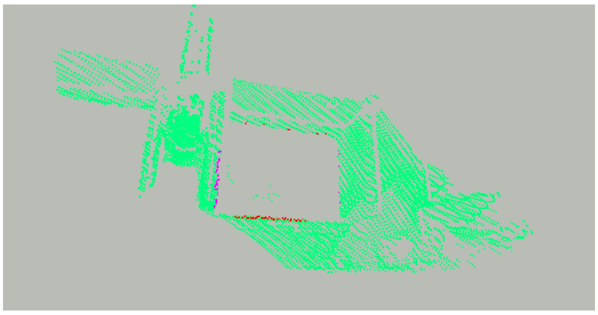

Figure 6. Point CLoud of Four Edges of operation target

$$\begin{aligned} &\min_{A_k, B_k, C_k, D_k} \sum_{p_m} d(A_k, B_k, C_k, D_k, p_m) \\ \boldsymbol{s.t.} \quad & A_k^2 + B_k^2 + C_k^2 = 1 \\ & d(A_k, B_k, C_k, D_k, p_m) = |A_k x_m + B_k y_m + C_k z_m| \\ & d(A_k, B_k, C_k, D_k, p_m) \leq Threshold3 \\ & p_m \in P_e^k \end{aligned} \tag{22}$$

$P_e^k (k = 1, 2, 3, 4)$ represents the point clouds of four detected edges; A_k, B_k, C_k, D_k represents the coefficients of fitted straight lines.

For real-time identification of operation targets, our algorithm only process point cloud of one frame once a time, and thus the points of hatch's edge detected in real time are not dense enough for accurate fitting of edge lines. To solve this problem, we use the 3D point cloud of the whole hull blocks built in former section as reference to complete the four edge's point clouds and redo the straight line fitting in order to refine our estimation. To be specific, we build a KdTree with the point cloud of whole hull blocks as input, and search the KdTree for points neighbored to the four fitted straight lines and then add those neighbored points to the point clouds of four edges respectively:

$$\begin{aligned} P_e^{k'} = P_e^k + KdTree(P_{total}).NearestSearch((x, y, z)),& \\ A_k x + B_k y + C_k z + D_k = 0& \\ x_{min} \leq x \leq x_{max}& \\ y_{min} \leq y \leq y_{max}& \\ z_{min} \leq z \leq z_{max}& \end{aligned} \tag{23}$$

$x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ are the range of hatch under operation calculated through the four fitted lines $A_k x + B_k y + C_k z + D_k = 0$. By doing so, we acquire four new point clouds which include complemented points of edges $P_e^{k'}$, as shown in Figure7. From the figure, we can see that the point clouds of edges are much denser than those in Figure6, which can improve the fitting accuracy.

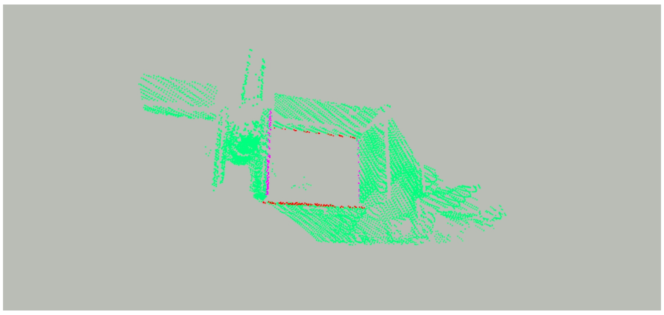

Figure 7. Complete Point CLoud of Four Edges of operation target

We redo the fit based on complemented point clouds $P_e^{k'}$ and obtain refined edge lines, as shown in Figure8.

$$\begin{aligned} & \min_{A_k', B_k', C_k', D_k'} \sum_{p_m} d(A_k', B_k', C_k', D_k', p_m) \\ \textbf{\textit{s.t.}} \quad & A_k'^2 + B_k'^2 + B_k'^2 = 1 \\ & d(A_k', B_k', C_k', D_k', p_m) = |A_k' x_m + B_k' y_m + C_k' z_m| \\ & d(A_k', B_k', C_k', D_k', p_m) \leq Threshold3 \\ & p_m \in P_e^{k'} \end{aligned} \tag{24}$$

Experiment

Experiment for our hull reconstruction algorithm and target identification algorithm is implemented on the ship loader at Tian Jian's Shen Huang Coal Terminal. The lidar is mounted on the load as shown in Figure9. The speed of moving scan system is $0.5m/s$. A laptop with $IntelCorei7-8750KCPU$ is used for computation.

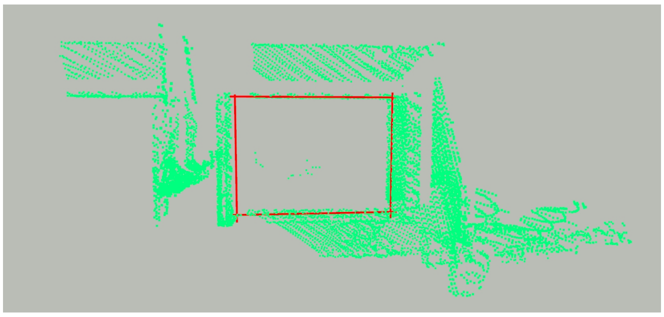

Figure 8. refined edge lines

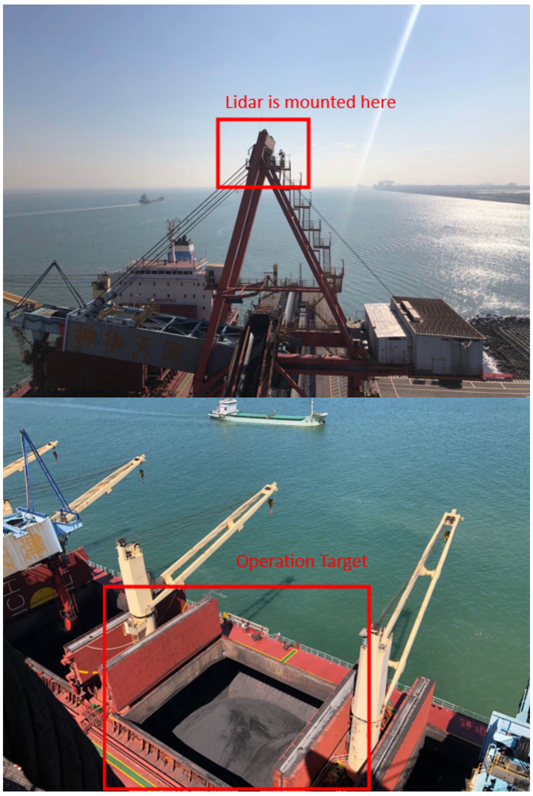

Figure 9. Experiment environment of the coal terminal

The 3D structure of the whole hull blocks is shown in Figure3, and the result of hull reconstruction's experiment is shown in Table1. And the result of experiment on identification algorithm of operation target is shown in Table 2.

Table 1. The result of hull reconstruction.

Option	Overall Length	Overall width
reconstruction value	179m	28m
real value	178m	27.6m
error rate	0.6%	1.4%

As shown in the Table1 and Table2, the computation time of target identification is less than $400ms$, which makes our algorithm applicable to process point cloud streams and guide the operation of loading in real time. The error in reconstruction of hull blocks is comparatively low(1.4% in

Table 2. The result of target identification.

Hold No.	1[1]	2	3	4	5
estimated length(m)	14.7	19.65	19.5	19.43	19.63
real length(m)	14.4	19.2	19.2	19.2	19.2
error rate(%)	2.1	2.3	1.6	1.2	2.2
estimated width(m)	14.23	14.64	14.57	14.63	14.52
real width(m)	14.4	14.4	14.4	14.4	14.4
error rate(%)	1.2	1.7	1.2	1.6	0.8
computation time(ms)			≤ 400		

[1] As the size of Hold NO.1 shown in the ship plan is not reliable, the 'real size' of Hold NO.1 is measured manually and the 'real size' of other holds are from the ship plan.

width and 0.6% in length). The maximum error in target identification is 1.7% in width and 2.3% in length. In the result of target identification, the error in length estimation is generally larger than that in width estimation. The main reason of that is the estimated edge of width in the left(the left edge in purple shown in Figure7) is deviated from the real edge not because of flaws in algorithm but because the lidar does not face straightly to the deck plane due to the operation environment, as shown in Figure1 and thus the features of the left edge is not complete even in the rebuilt model of the whole blocks. To solve this problem, in the future, we will use multiple lidars at different angles facing toward the deck plane of cargo ships and merge point clouds collected by those lidars for identification, and then the accuracy of our target identification algorithm can be further improved.

Conclusion

This paper presents a algorithm of reconstructing hull blocks of cargo ships based on a moving scan measurement system and a algorithm of identifying operation targets in real time with high accuracy. Our system and algorithm are simple and reliable. Firstly, LMS moves and scans the whole cargo ship at a uniform speed and then rebuild the 3D structure of hull blocks. Then, the LMS moves to operation zone of the ship, face to the operation targets and scan the zone constantly. Our identification algorithm processes the real-time point cloud streams for estimation the size and dynamic posture of operation target at a refreshing frequency of more than 2.5 frame per second, which makes it possible to guide the automatic ship loading in real time. The experiment results show that our system and algorithm works effectively and efficiently.

Future Work

In order to further improve our system and algorithm and to apply them in real operation scenes in bulk terminals, in the future, we will do further research as follows:

1. In order to achieve higher accuracy of reconstruction of hull blocks and identification of operation targets, we plan to apply multiple lidars in measurement system with different directions toward the deck plane and to explore new algorithms of reconstruction and identification based on merged point clouds from multiple lidars.
2. As the loader switches between operation targets, obstacles like cranes and hatch lids on the deck may get in the way of the loader and result in collision. Thus, in order to facilitate the movement of the loader and avoid collision, the obstacles on the deck need to be mapped and the safe path free of obstacles for the loader can be planned.
3. The shape of bulk materials in hatches can be modelled to guide the loading system for the even distribution of materials and improving the utilization of space in the hatch.

Acknowledgments

This research is supported by Shenhua Tianjin Coal Terminal Co. LTD(NO. 120000000003510).

References

[1] Chen H, Sun G, Yang A, et al. Unmanned cargo loading system for mine, has actuator controller that controls driving to perform two-dimensional horizontal movement to move grab bucket to top portion of grabbing point. Patent CN108584467-A, 2018.

[2] Guan G, Shen M, Lin Y, et al. A method for the Automatic Registration of Hull Blocks Point Clouds. In: Chen WZ, Xu XP, Dai PQ, et al. (eds) Advanced Manufacturing Technology, Pts 1-4. 2012, pp.3089-+.

[3] Xi-Liang D, Mi C, Shen Y, et al. Algorithmic study on bulk cargo carrier detection and positioning based on 3D laser vision. Chinese Journal of Construction Machinery. 2014, pp.555-559.

[4] Mi C, Shen Y, Mi W, et al. Ship Identification Algorithm Based on 3D Point Cloud for Automated Ship Loaders. Journal of Coastal Research 2015: 28-34.

[5] Kawai H, Choi Y, Kim YB, et al. Position measurement of container crane spreader using an image sensor system for anti-sway controllers. In: 2008 International Conference on Control, Automation and Systems 14-17 Oct. 2008, pp.683-686.

[6] Yoon H-J, Hwang Y-C, Cha E-Y, et al. Real-time container position estimation method using stereo vision for container auto-landing system. 2010, p.872-876.

[7] Zhang Y, Cao H, Weng F, et al. Open container loading three-dimensional sensing and stacking planning method, involves performing container size detection and pose estimation operations, and planning stacking operation according to size and pose of container. Patent CN111652936-A, 2020.

[8] Kitayama T, Lu H, Li Y, et al. Detection of Grasping Position from Video Images Based on SSD. 2018 18th International Conference on Control, Automation and Systems. 2018, pp.1472-1475.

[9] Zhang Y. Three-dimensional (3D) depth imaging system for automatic container door status recognition, has container recognition app that is configured to

execute on processors to determine status type of each door from 3D image door area data subset. Patent US10762331-B1, 2020.

[10] Rusu RB, Blodow N and Beetz M. Fast Point Feature Histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation 12-17 May 2009, pp.3212-3217.

[11] Steder B, Rusu RB, Konolige K, et al. Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries. 2011 IEEE International Conference on Robotics and Automation. 2011, pp.2601-2608.

[12] Suzuki S and Abe K. TOPOLOGICAL STRUCTURAL-ANALYSIS OF DIGITIZED BINARY IMAGES BY BORDER FOLLOWING. Computer Vision Graphics and Image Processing 1985; 30: 32-46.