Journal of Computational Design and Engineering Vol. 1, No. 3 (2014) 202~212
www.jcde.org

Efficient point cloud data processing in shipbuilding: Reformative component extraction method and registration method

Jingyu Sun[1,*], Kazuo Hiekata[1], Hiroyuki Yamato[1], Norito Nakagaki[2] and Akiyoshi Sugawara[2]

[1] *Graduate School of Frontier Sciences, University of Tokyo, Environmental Building, 5-1-5 Kashiwano-ha, Kashiwa, Chiba 277-8563, Japan*
[2] *Sumitomo Heavy Industries Marine & Engineering Co., Ltd, 19, Natsushimacho, Yokosuka-shi, Kanagawa, 237-0061, Japan*

(Manuscript Received February 27, 2014; Revised May 9, 2014; Accepted May 12, 2014)

Abstract

To survive in the current shipbuilding industry, it is of vital importance for shipyards to have the ship components' accuracy evaluated efficiently during most of the manufacturing steps. Evaluating components' accuracy by comparing each component's point cloud data scanned by laser scanners and the ship's design data formatted in CAD cannot be processed efficiently when (1) extract components from point cloud data include irregular obstacles endogenously, or when (2) registration of the two data sets have no clear direction setting. This paper presents reformative point cloud data processing methods to solve these problems. K-d tree construction of the point cloud data fastens a neighbor searching of each point. Region growing method performed on the neighbor points of the seed point extracts the continuous part of the component, while curved surface fitting and B-spline curved line fitting at the edge of the continuous part recognize the neighbor domains of the same component divided by obstacles' shadows. The ICP (Iterative Closest Point) algorithm conducts a registration of the two sets of data after the proper registration's direction is decided by principal component analysis. By experiments conducted at the shipyard, 200 curved shell plates are extracted from the scanned point cloud data, and registrations are conducted between them and the designed CAD data using the proposed methods for an accuracy evaluation. Results show that the methods proposed in this paper support the accuracy evaluation targeted point cloud data processing efficiently in practice.

Keywords: Point cloud; Region growing method; B-spline curve; ICP; K-d tree

1. Introduction

While efficiently evaluating the accuracy of the ship component has been considered important for many years as a necessary part of planning and control of production in shipbuilding, using point cloud data scanned by noncontact 3D laser scanner has only recently been taken into consideration [1]. As an application of laser scanners to shipbuilding, an accuracy evaluation system for ship components was developed at the University of Tokyo [2, 3]. The procedure is as follows:

(Step 1) Obtain the point cloud data by scanning the component using the noncontact 3D laser scanner.

(Step 2) Extract the part of the component from the point cloud data using basic region growing method.

(Step 3) Conduct a registration of the component's measured point cloud data and the design data using ICP algorithm directly.

(Step 4) Find the displacement errors by comparing the position of the two data and visualize them as the accuracy evaluation result.

Two vital problems exist here.

(a) In Step 2, the part of the component cannot be extracted efficiently due to irregular obstacles endogenous to manufacturing workshops and their shadows divide the component's measured point cloud data into several separated domains. Manual extraction and integration of these separated domains waste a lot of time and is a barrier for practical application. The well-known methods for calculating smooth surfaces from noisy point cloud such as moving least squares (MLS) projection [4] has the same problem that the whole component cannot be extracted at one time when the component is separated by the obstacles' shadows.

(b) In Step 3, an ICP (Iterative Closest Point) algorithm is used to conduct a registration between the measured point cloud data and the design data. Due to the features of the ICP algorithm, without a clear registration direction presetting, the registration of the two data

[*] Corresponding author. Tel.: +81-47-136-4626, Fax.: +81-47-136-4626
E-mail address: sun@is.k.u-tokyo.ac.jp

doi: 10.7315/JCDE.2014.020

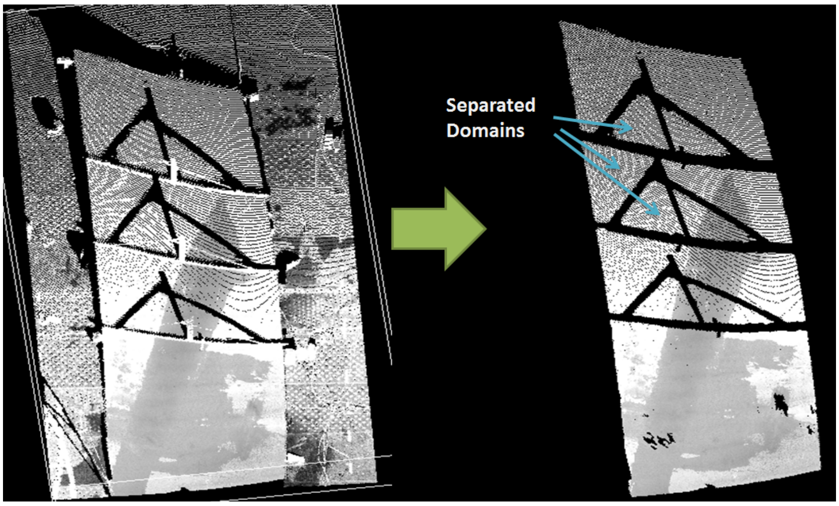


Figure 1. Component extraction.

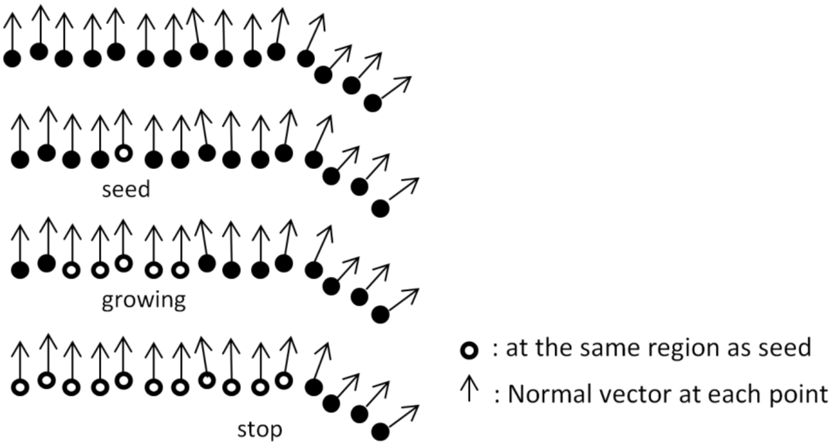

seed
growing
stop
○ : at the same region as seed
↑ : Normal vector at each point

Figure 2. Region growing method.

sets could go in the wrong direction and lead to improper registration results.

As a result, existing point cloud processing methods (the basic region growing method and ICP registration method) can rarely satisfy the accuracy evaluation target in shipbuilding without significant improvements and appropriate combination.

This paper presents a reformative component extraction method solving problem (a) and a reformative component registration method solving problem (b) for the component's accuracy evaluation in shipbuilding. First, k-d tree construction is performed on the scanned point cloud data to fasten the neighbor searching of each point. To extract the continuous domain of the component from the point cloud data, a region growing method is performed on every neighbor point of the seed point. Then the neighbor domain, which is separated by obstacles' shadows from the extracted domain, is recognized by conducting curved surface fitting and B-spline curved line fitting at each edge point of the extracted domain. The whole component can be extracted by repeating the above steps. Before registering the extracted point cloud data of the component and the designed data, proper registration direction is decided by performing registration direction analysis on the two kinds of data. And then, ICP is applied for the following registration.

2. Component extraction

In shipbuilding, depending on the conditions in the factory, the laser scanner measured results of components (see Figure 1 left) cannot satisfy the accuracy evaluation requirement due to the following factors:

(1) The measured point cloud data always includes a lot of needless noise, such as the floor, the workers and some wooden templates which are necessary templates for the manufacturing.

(2) The measured point cloud data are usually divided into multiple separated domains by these obstacles' shadows. Manual extraction and integration of these separated domains wastes a lot of time.

   A. The measured point cloud data always has a large amount of points which slow down the extraction process dramatically under the

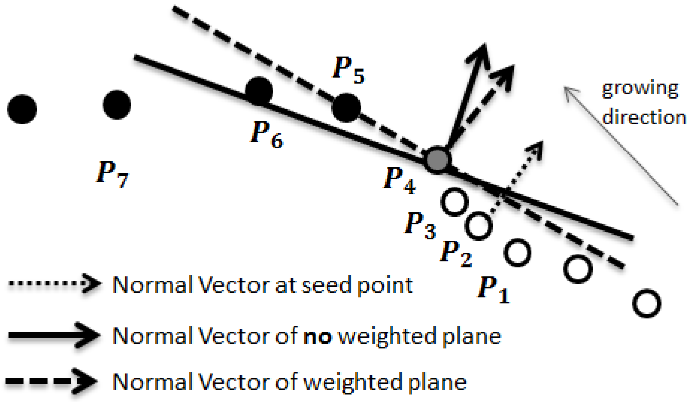

P_5
P_6
P_7
P_4
P_3
P_2
P_1
growing direction
Normal Vector at seed point
Normal Vector of no weighted plane
Normal Vector of weighted plane

Figure 3. Weighted plane fitting using principal component analysis.

preferred time limit of several minutes.

As a crucial pre-process, all of the noise should be removed and the separated domains belonging to the same component should be extracted efficiently as shown in Figure 1. In this paper, the component extraction consists of two main processes: the continuous domain extraction which extracts the small separated domains from one single seed point and the separated domain recognition which decide if an extracted domain is from the same component as the other ones.

2.1 Continuous domain extraction

This paper makes some improvements in the continuous domain extraction method (basic region growing method) proposed in the prior work to make the extraction more effective and high speed.

2.1.1 Proposed methods in the prior work --- Region growing method and k-d tree construction

The proposed component extraction method is based on a basic point cloud processing method called region growing method [5] proposed in the prior work which repeats the process of calculating the normal vector at each neighbor point of the seed point, extracting the neighbor point which has a similar normal vector as the seed point and setting the extracted neighbor point as the next seed point. Figure 2 shows the main flow of the region growing method. The growing process ends when the normal vectors have relatively dramatic changes.

To make the neighbor search more efficient here, a space-partitioning data structure, k-d tree [6], is constructed using the scanned point cloud data for organizing points in a k-dimensional space recursively.

2.1.2 Normal vector calculation considering point cloud's density

When the point cloud's density varies dramatically, to calculate the normal vector of a seed point using its neighbor points, plane fitting method using all the neighbors equally may cause unrepresented results. As shown in Figure 3, to determine whether $\boldsymbol{P_4}$ belongs to the same region as $\boldsymbol{P_i}$ $(i = 1,2,3)$, normal vector calculation is performed on it. Since the density at $\boldsymbol{P_i}$ $(i = 5,6,7)$ is lower than that at $\boldsymbol{P_i}$ $(i = 1,2,3)$, the normal vector of no weighted plane will be outputted as in Figure 3. Obviously, it cannot represent the normal vector at $\boldsymbol{P_4}$, and is dramatically differed from the normal vector at $\boldsymbol{P_2}$. Therefore, the growing is terminated without $\boldsymbol{P_4}$ being included while $\boldsymbol{P_4}$ actually belongs to the same region as $\boldsymbol{P_i}$ $(i = 1,2,3)$.

Thus, to calculate the normal vector at each neighbor point and the seed point, weighted plane fitting [7] is conducted in this paper. The center of gravity of the point cloud $\boldsymbol{G}$ is calculated by Eq. (1), and then the variance-covariance matrix $\boldsymbol{V}$ can be obtained using Eq. (2). The minimum eigenvector $\boldsymbol{n}$ is calculated as the normal vector of this point using $\boldsymbol{V}$. d is the distance between each point and the point being calculated and h is the average interval of the point cloud points. Since the weight m_i is calculated for each point so as points closer to the point being calculated have greater influence when the plane is fitted, the normal vector of weighted plane can be outputted as shown in Figure 3. With an acceptable minor difference from the normal vector at $\boldsymbol{P_2}$, $\boldsymbol{P_4}$ can be considered as belonging in the domain as $\boldsymbol{P_i}$ $(i = 1,2,3)$.

$$\boldsymbol{G} = \frac{1}{\sum_{i=0}^{n-1} m_i} \sum_{i=0}^{n-1} m_i \boldsymbol{P_i}\,, \quad \boldsymbol{P_i} = (x_i, y_i, z_i)^T \tag{1}$$

$$\boldsymbol{V} = \frac{1}{\sum_{i=0}^{n-1} m_i} \sum_{i=0}^{n-1} m_i (\boldsymbol{P_i} - \boldsymbol{G}) \times (\boldsymbol{P_i} - \boldsymbol{G})^T$$

$$m_i = exp\left(-\frac{d^2}{h^2}\right) \tag{2}$$

2.1.3 Points selection for plane fitting and growing entry candidate

To avoid improper growing into points that are too far from the seed such as the floor near the component, points which are used to do the plane fitting have to be seriously selected under certain constraints. The distance d between the

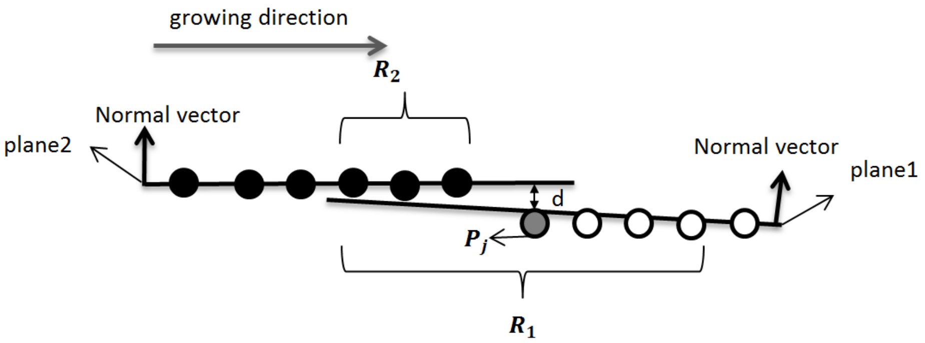


Figure 4. Points used to do plane fitting.

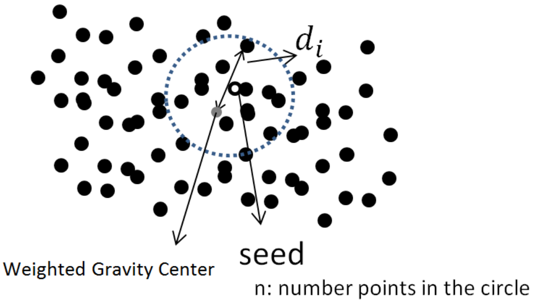


Figure 5. Region growing entry point selection.

point being evaluated and the extracted points is checked to decide if this point should be considered to be part of the domain. In this paper, two planes are fitted for the normal vector calculation and the distance *d*'s calculation respectively as shown in Figure 4. The point $\boldsymbol{P_j}$ is the point being evaluated. Since plane1 is used to calculate the normal vector of the point $\boldsymbol{P_j}$, it is fitted using all the neighbor points $\boldsymbol{R_1}$ with no concern about whether they are already recognized as part of the domain or not. Normal vector comparison can only evaluate the direction trend here at $\boldsymbol{P_j}$. The distance d should also be considered to avoid growing to an area not part of the domain but only has a slight difference in normal vector comparison result. Plane2, which is used to evaluate the distance d, is calculated using only the points $\boldsymbol{R_2}$ which are already recognized as part of the domain.

To save the computing time, only the points which are relatively outside of the neighbors are considered to be the next growing entry candidates. Thus, as shown in Figure 5, the weighted gravity point ***G*** is calculated using Eq. (1) in Section 2.1.2, and the distance d_i is computed for every point $\boldsymbol{P_i}$ of the neighbors. Only when

$$d_i > bthresh \times \sum_{1}^{n} d_k \; /n$$

is the point $\boldsymbol{P_i}$ considered to be the next growing entry. The bthresh is a thresh value between 0 and 1. In this paper, bthresh is set to 0.74.

2.2 Separated domain recognition

Since the method discussed above can only extract continuous domain, to extract the whole component, which has been irregularly divided by obstacles' shadows like the curved shell plate in Figure 6 (left) or some component with more than one curved surface as in the block's bottom panel in Figure 6 (right), workers have to choose a seed point for every separated domain and execute region growing method on every domain one by one. This paper proposes separated domain recognition method to save this inefficient step.

2.2.1 Separated domain's recognition and extraction from single seeds

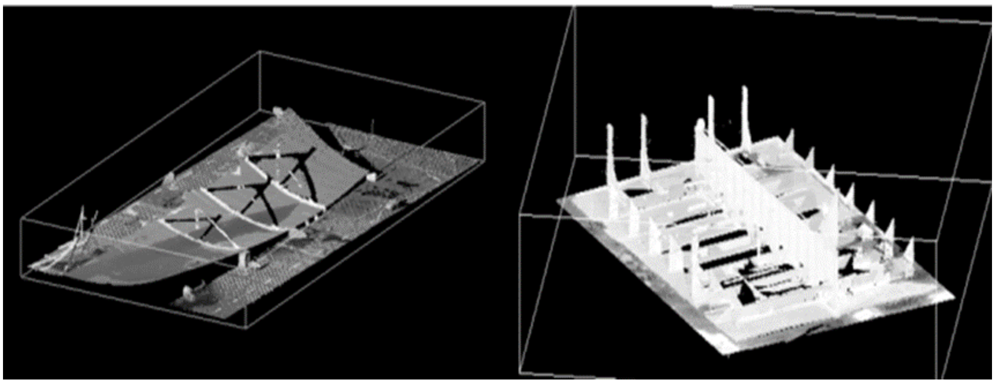


Figure 6. Curved shell plate (left) and block's panel (right).

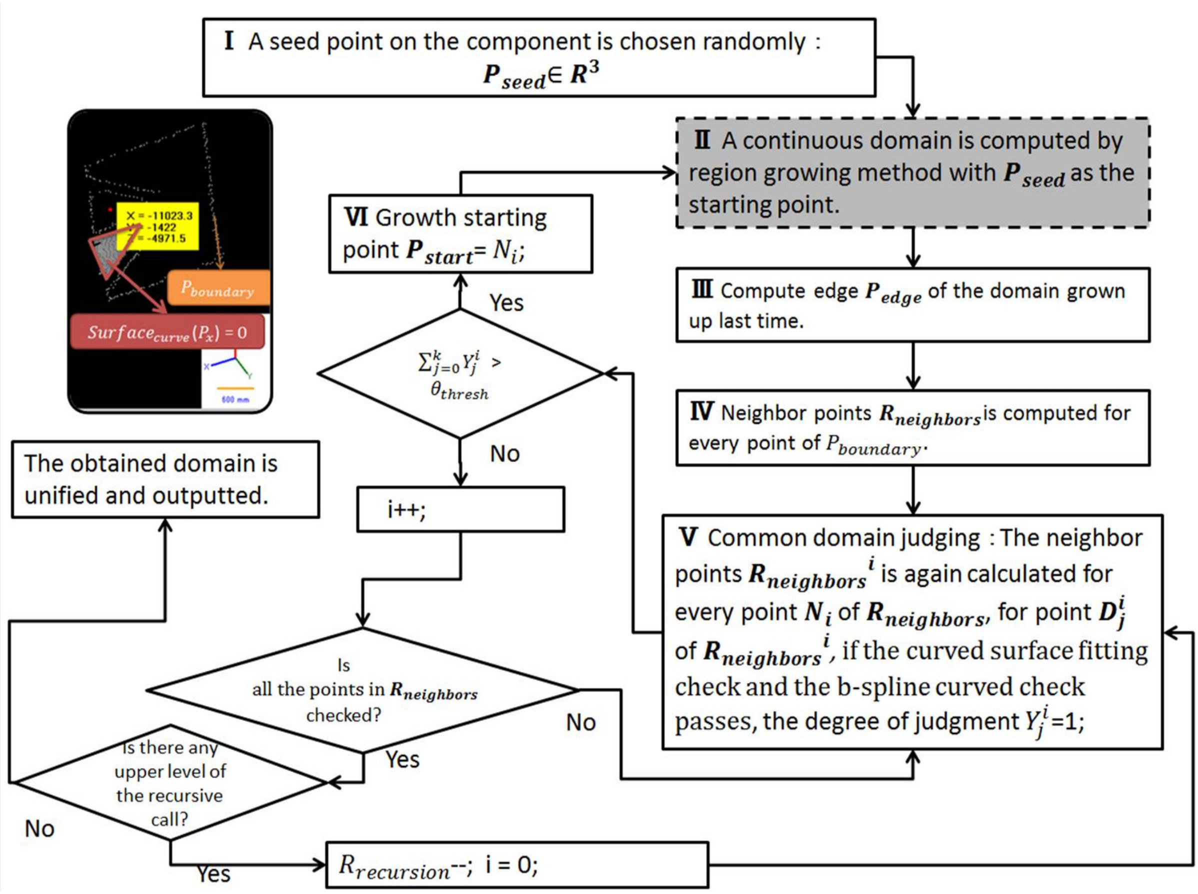

I A seed point on the component is chosen randomly : $P_{seed} \in R^3$
II A continuous domain is computed by region growing method with P_{seed} as the starting point.
III Compute edge P_{edge} of the domain grown up last time.
IV Neighbor points $R_{neighbors}$ is computed for every point of $P_{boundary}$.
V Common domain judging : The neighbor points ${R_{neighbors}}^i$ is again calculated for every point N_i of $R_{neighbors}$, for point D_j^i of ${R_{neighbors}}^i$, if the curved surface fitting check and the b-spline curved check passes, the degree of judgment Y_j^i=1;
VI Growth starting point $P_{start} = N_i$;
$\sum_{j=0}^{k} Y_j^i > \theta_{thresh}$
Yes
No
i++;
Is all the points in $R_{neighbors}$ checked?
Yes
No
Is there any upper level of the recursive call?
Yes
No
$R_{recursion}$--; i = 0;
The obtained domain is unified and outputted.
X = -11023.3
Y = -1422
Z = -4971.5
$P_{boundary}$
$Surface_{curve}(P_x) = 0$
500 mm

Figure 7. Component extraction flow (single seed).

The common domain judgment is conducted here to decide if any neighbor domain is of the same component as the continuous domain extracted firstly using plain region growing method illustrated above.

Figure 7 shows the flow of this common domain judgment process. First, one single seed P_{seed} on the component is selected randomly, and then the continuous domain is computed by a region growing method illustrated above. The second step, highlighted in gray, has the same processing as the process discussed in Section 2.1. Neighbor points $R_{neighbors}$are calculated for every point of the extracted continuous domain's boundary $P_{boundary}$. Then the neighbor points ${R_{neighbors}}^i$ are obtained for point D_j^i in $R_{neighbors}$. For all the points D_j^i in ${R_{neighbors}}^i$, the common domain

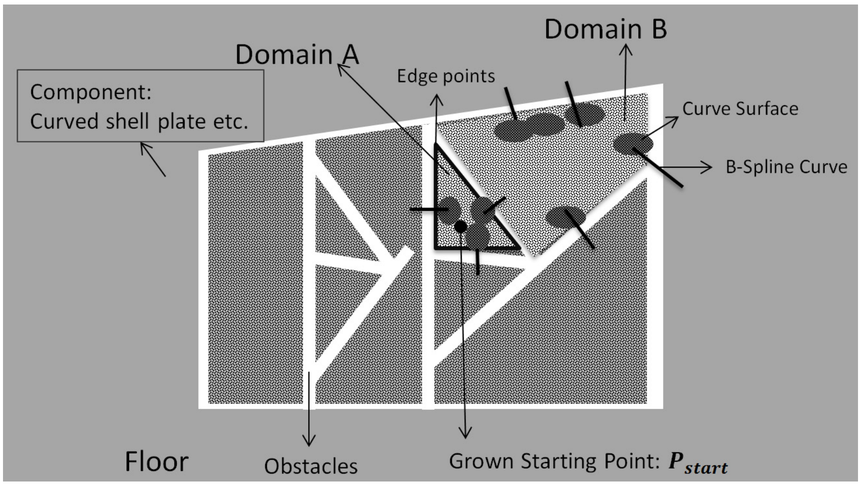


Figure 8. Common domain judgment.

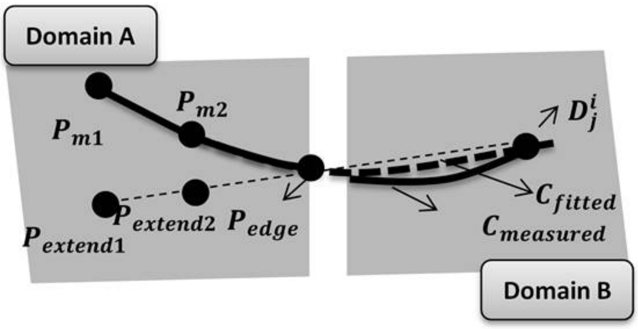


Figure 9. Fitted curve and measured curve.

judgment, which will be introduced later, is performed to decide whether $\boldsymbol{N_i}$ can be chosen as a seed point of another domain of this component. Also, this process is recursively repeated until all the points of the component are extracted successfully.

Figure 8 illustrates how common domain judgment works. Domain A is first recognized by the method in Section 2.1. For every edge point of domain A, 4th curved surface fitting and B-spline curve fitting [8], which is widely used in designing free-form surface of ships, are performed to decide if the degree of *judgment* Y_j^i of point $\boldsymbol{D_j^i}$ is true (1) or false (0). Only when the sum of all Y_j^i is greater than the threshold, is the point $\boldsymbol{N_i}$ selected as a seed point to start another domain's (domain B) growing process.

$$\sum_{k=1}^{N}\left[z_k - \sum_{j=0}^{4}\left(\sum_{i=0}^{j} c_{i,j-i} x_k^i y_k^{j-i}\right)\right]^2 = 0,$$
$$\boldsymbol{P_i} = (x_i, y_i, z_i)^T \tag{3}$$

$$\boldsymbol{S}(t) = \sum_{i=0}^{m-n-2} \boldsymbol{P_i}\, b_{i,n}(t), \qquad t \in [t_n, t_{m-n-1}] \tag{4}$$

$b_{j,0} \ = 1 \quad if\ t_j \le t \le t_{j+1}$
where $j = 0, \dots, m-2$

$b_{j,0} \ = 0 \quad if\ t > \ t_{j+1}\ or\ \ t < t_j$
where $\ j = 0, \dots, m-2$

$$b_{j,n}(t) \ = \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) \ + \ \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t)$$

where $\ j = 0, \dots, m-n-2$ (5)

In this process, Eq. (3) is used to fit the curved surface around the edge point of the extracted domain A. $\boldsymbol{D_j^i}$ is checked to see whether it belongs to the curved surface $\boldsymbol{S_{fitted}}$ or not.

If only $\boldsymbol{D_j^i}$ passes the curved surface check, the B-spline curved line check is conducted using Eqs. (4) and (5). Here $\boldsymbol{P_i}$ is the control point. The n times B-spline curve of one segment is expressed as Eq. (4). $b_{i,n}$ is the basis function of B-spline curve. *t* are *m* real numbers which are called knots here.

As shown in Figure 9 four points ($\boldsymbol{D_j^i}$, $\boldsymbol{P_{edge}}$, $\boldsymbol{P_{m1}}$, $\boldsymbol{P_{m2}}$) are taken as the control points of the B-spline line. $\boldsymbol{P_{extend1}}$ and $\boldsymbol{P_{extend2}}$ are two points on the extension line of $|\boldsymbol{D_j^i}\,\boldsymbol{P_{edge}}|$, and $\boldsymbol{P_{m1}}$,$\boldsymbol{P_{m2}}$ are their nearest points in domain A. After B-spline curve $\boldsymbol{C_{fitted}}$ is calculated out, the nearest points in domain *B* are obtained for the points of $\boldsymbol{C_{fitted}}$. The group of these nearest points is $\boldsymbol{C_{measured}}$. Only if the average of every pair's distance between $\mathbf{C_{fitted}}$ and $\mathbf{C_{measured}}$ is smaller than the threshold, can the degree of judgment Y_j^i of $\boldsymbol{D_j^i}$ be set to true.

2.2.2 Edge extraction

If the entire neighbor points in $\boldsymbol{R_{neighbors}}$ (shown in Figure 7) are considered as the next entry of growing, needless calculation of the points which have already been recognized as domain points will cost a lot of time. Thus only the points at the edge of $\boldsymbol{R_{neighbors}}$ are used to do the following calculation.

To extract the edge in this process, as shown in Figure 10,

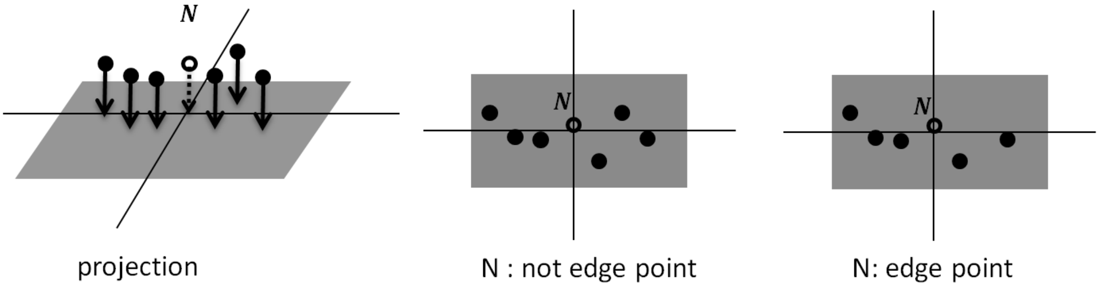


Figure 10. Edge detection.

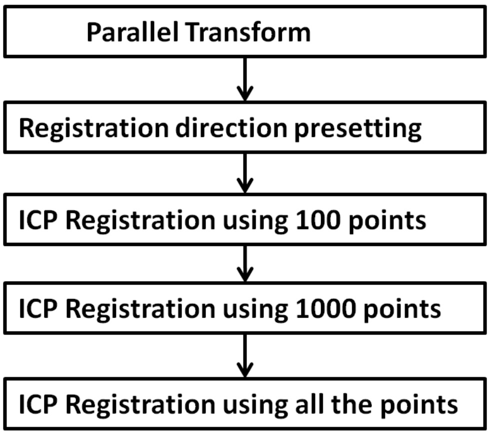

Parallel Transform
Registration direction presetting
ICP Registration using 100 points
ICP Registration using 1000 points
ICP Registration using all the points

Figure 11. Component registration flow.

a plane is fitted using the principal component analysis. The points used to do plane fitting is the neighbor points of seed point ***N*** searched by k-NN method. Then the neighbor points are projected to the plane. If not all of the four quadrants have neighbor points as shown in the right image of Figure 10, the point ***N*** is considered as the edge [9].

3. Component registration

In shipbuilding, components' design data is often expressed in SAT format, a standard file format of ACIS. In SAT files, surfaces and curves are expressed as NURBS (Non-Uniform Rational B-Spline). To evaluate the accuracy of the component or to provide any farther manufacturing advice, as an important pre-process, a registration between the design data and the extracted component's point cloud data should be conducted efficiently. As shown in Figure 11, this paper proposes an efficient registration method involving parallel transformations using the center of gravity of the two data sets, the pre-setting of the registration direction and the traditional ICP registration, an algorithm employed to minimize the difference between two clouds of points.

3.1 Parallel transformation using centre of gravity

To register the measured point cloud to the designed data, a parallel transformation using the center of gravity of both data sets is conducted using the equation below. Here the ***P*** represents the point from the measured data. $\boldsymbol{G_d}$ and $\boldsymbol{G_m}$ are the centers of gravity of the designed data and measured data respectively.

$$\boldsymbol{P}' = \boldsymbol{P} + (\boldsymbol{G_d} - \boldsymbol{G_m}) \tag{6}$$

3.2 Registration Using ICP Algorithm

After a parallel transformation using the center of gravity, a basic ICP algorithm is conducted for the registration of the point cloud and the design data. The design curved surface $\boldsymbol{S}(u,v) = \big(x(u,v), y(u,v), z(u,v)\big)$ is a rational equation using the B-Spline basis functions which represent the homogeneous coordinate system projection results of the non-constant spacing knots [10].

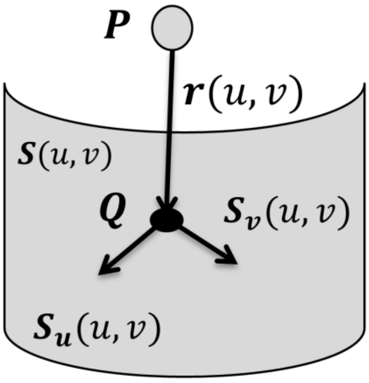


Figure 12. Flow chart of the method A.

$$\boldsymbol{r}(u,v) \ = \boldsymbol{S}(u,v) - \boldsymbol{P} \tag{7}$$

$$\boldsymbol{r}(u,v) \cdot \mathbf{S_u}(u,v) \ = 0 \tag{8}$$

$$\boldsymbol{r}(u,v) \cdot \boldsymbol{S_v}(u,v) \ = 0 \tag{9}$$

As defined in Eq. (7), the projection vector can be obtained by subtracting the measured point ***P***'s position vector from the design curve surface $\boldsymbol{S}(u,v)$. Therefore, $\boldsymbol{r}(u,v)$ satisfies the Eqs. (8)and (9). Here, the $\boldsymbol{S_u}(u,v)$ and $\boldsymbol{S_v}(u,v)$ are the *u*-direction's and the *v*-direction's partial derivatives of the designed curved surface respectively. Projection vector $\boldsymbol{r}(u,v)$ as shown in Figure 12 can be calculated by solving these three equations. Here ***Q*** is the projection point.

Next, the coordinate transformation of the measured point cloud using the motion vector ***t*** and the rotation matrix ***R*** is conducted to reduce the displacement of the design curved surface and the measured point cloud [11, 12]. $\boldsymbol{P}'$ in Eq. (10) is the transformed measured point. To estimate ***R*** and ***t***, the displacement $E(\boldsymbol{R},\boldsymbol{t})$ expressed in Eq. (11) is minimized [13]. This process is repeated until $E(\boldsymbol{R},\boldsymbol{t})$ is smaller than the threshold [14].

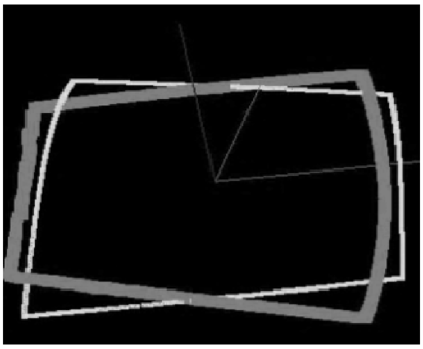


Figure 13. Inverted registration result.

$$\boldsymbol{I} = \begin{bmatrix} \sum_{i=0}^{n-1} m_i(|\boldsymbol{P_i}|^2 - x_i^2) & -\sum_{i=0}^{n-1} m_i x_i y_i & -\sum_{i=0}^{n-1} m_i x_i z_i \\ -\sum_{i=0}^{n-1} m_i x_i y_i & \sum_{i=0}^{n-1} m_i(|\boldsymbol{P_i}|^2 - y_i^2) & -\sum_{i=0}^{n-1} m_i y_i z_i \\ -\sum_{i=0}^{n-1} m_i x_i z_i & -\sum_{i=0}^{n-1} m_i y_i z_i & \sum_{i=0}^{n-1} m_i(|\boldsymbol{P_i}|^2 - z_i^2) \end{bmatrix}$$

$$m_i = exp\left(-\frac{d^2}{h^2}\right), \qquad \boldsymbol{P_i} = (x_i, y_i, z_i)^T \qquad (12)$$

$$\boldsymbol{P}' = \boldsymbol{RP} + \boldsymbol{t} \qquad (10)$$

$$E(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{n} \|\boldsymbol{Q_i} - \boldsymbol{RP_i} - \boldsymbol{t}\| \qquad (11)$$

3.3 Registration direction pre-setting

In practice, as shown in Figure 13, the registration method discussed in Section 3.2 stands a good chance of getting an obviously wrong output if the registration is started from an improper direction.

To solve this problem, before ICP algorithm is conducted, After three directions of the inertia principal axes are calculated to find the proper registration start direction with the minimum displacement. Firstly, the inertia tensor ***I*** is calculated using Eq. (12). The three eigenvectors of ***I*** are the inertia principal axes as shown in Figure 14.

As expressed in Eq. (13), the measured point cloud P is registered to the design data according to multiple ways. The registration direction with the minimum displacement is chosen to be the original direction to conduct the ICP algorithm. Here $\boldsymbol{R_d}$ and $\boldsymbol{R_m}$ are the eigenvectors' set of the design data's and the measured data's inertia tensor ***I*** respectively.

$$\boldsymbol{P}' = \boldsymbol{R_d}^T \times \boldsymbol{R_m} \times \boldsymbol{P}$$

$$\boldsymbol{R_d} = (e_{d1}, e_{d2,} e_{d3}), \ \ \boldsymbol{R_m} = (ae_{m1}, be_{m2,} ce_{m3})$$

$$a = \{1, -1\}, b = \{1, -1\}, c = \{1, -1\} \qquad (13)$$

4. Experimental results

In this section a set of experiments are conducted in the shipyard to evaluate the performance of the point cloud processing methods proposed above. All examples in this paper are the result of implementing the proposed algorithm embedded in a 3D point cloud system developed in C# provided by UNICUS Co., Ltd.. The user can explicitly change the parameters discussed above to adjust the balance between the processing accuracy and the efficiency. All of the point cloud data are measured by a FARO Focus 3D laser scanner [15].

4.1 Component extraction results

About 200 curved shell plates are measured, extracted and evaluated after plastically deformed. Three examples about component extraction are given below. The Comparison between the prior method and new presented method is summarized in Table 1.

Example (a). As shown in Figure 15(a), a curved shell plate is measured with a few obstacles, such as the banner's code and the worker. It has about 1.68 million points. The algorithm discussed in Section 2.1 is used to deal with this kind of case in which the component is not divided by the obstacles' shadow. As the extraction result shows in the figure, the proposed advanced region growing algorithm is proven to be capable in overcoming these obstacles. The time used for extraction in this case is about 64 seconds.

Example (b). As shown in Figure 15(b), another curved shell plate is measured with some wooden templates on it. It has about 1.34 million points, and is divided into multiple domains due to the shadows of the wooden templates. In this

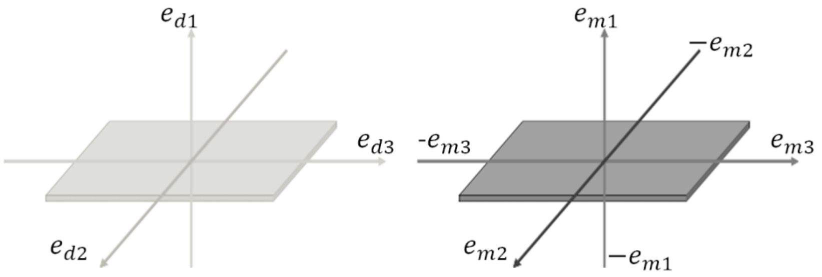
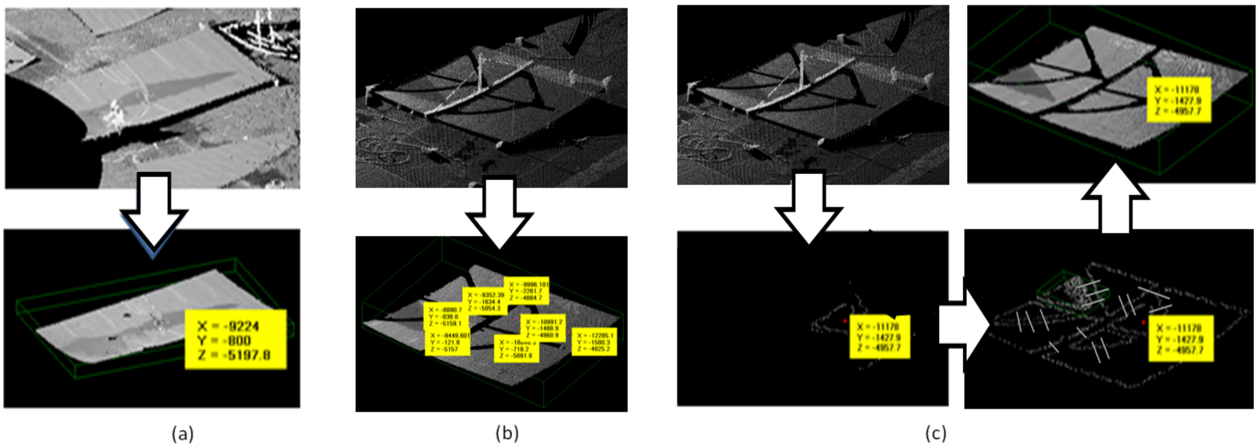


Figure 14. Inertia principal axes.

Table 1. Comparison between the single region growing method (b) and newly presented methods (a) and (b).

	Component divided into	Manually/Automatically	Execution time
(a)	1	Automatically	64s
(b)	7	Manually	120s
(c)	7	Automatically	53s



(a) (b) (c)

Figure 15. Curved shell plate extraction results: (a) extraction using algorithm in 2.1, (b) extraction using algorithm of prior work in Section 2.1.1, (c) extraction using algorithm in Section 2.2.

example, the manual extraction method of the prior work introduced in Section 2.1.1 is used to extract the multiple domains of the component. One growing seed point with 3D coordinates shown in the square box in the figure is selected manually for every domain. As the extraction results shown in the figure, the manual separated domain extraction method with multiple seeds is evaluated in this case. The time (manual integration operation included) used for extraction in this case is over 120 seconds.

Example (c). As shown in Figure 15(c), the same curved shell plate as example (b) is extracted using the algorithm proposed in Section 2.2. Only one single point is selected previously, then according to the curved surface fitting and the B-Spline line fitting (shown as white lines in the figure), the multiple domains are recognized automatically. The time used for extraction in this case is only about 53 seconds.

Not only can curved shell plates be extracted from the measured point cloud using the proposed algorithm, other components, such as the panel or the block’s lateral surface can use this method. Figure 16 shows the extraction results of

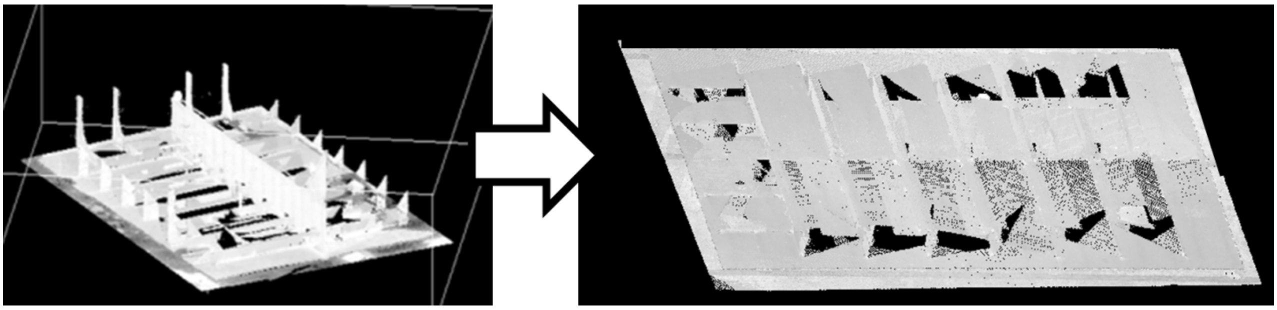


Figure 16. Panel extraction result.

Table 2. Comparison between basic methods used in prior work (a) and (b) and newly presented method (c).

	Registration result	Execution time
(a)	Failed	4s
(b)	Failed	43s
(c)	Succeeded	30s

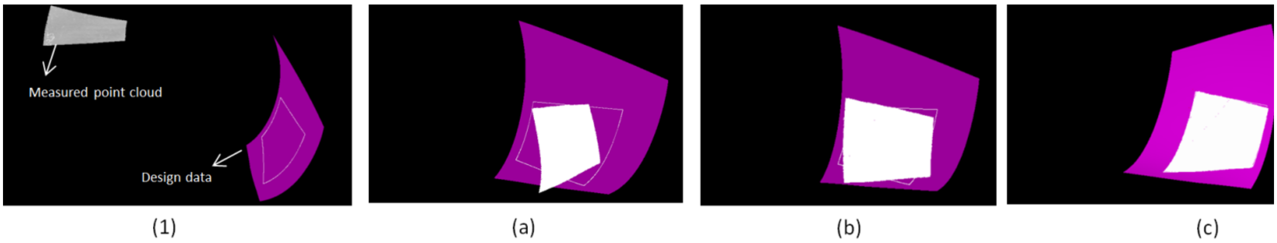


Figure 17. Curved shell plate registration results: (1) measured point cloud and design data, (a) registration using gravity center, (b) registration without direction presetting, (c) registration with direction presetting.

a panel component using the algorithm in Section 2.2.

4.2 Component registration results

A curved shell plate's registration example is given below to illustrate how the proposed component registration method works. As shown in Figure 17(1), the measured point cloud of the curved shell plate and the design data are displayed on the computer. It can be seen that originally they have totally different positions and need to be registered. The execution time is summarized in Table 2.

Result (a). Parallel transformation using only the 3D coordinate of the center of gravity according to the method discussed in Section 3.1.1 is performed at first. Figure 17(a) shows the transformation result of this step.

Result (b). Figure 17(b) shows the result of conducting the ICP algorithm directly from the position in result (a) without direction pre-setting. It turns out that the registration went in an inverse direction. Obviously, result (b) cannot satisfy the following component evaluation processing due to the large displacement.

Result (c). Multiple ways of registration direction are tested before the ICP algorithm is performed. The registration direction with the least displacement is chosen to do the direction pre-setting for the ICP algorithm. As shown in Figure 17(c), with registration direction test and presetting, the registration result goes exactly the right way, and turns out to be capable for the following processing.

5. Conclusions

This paper presented a set of efficient point cloud processing methods in shipbuilding including the reformative component extraction method and the component registration method. These methods solved two common problems which arose when using the existing methods including basic region growing method and ICP method directly in point cloud processing at shipyard, and are proved to be efficient according to experiments conducted in shipyard. With efficient point cloud processing using the reformative component extraction method and registration method presented in this paper, the component accuracy evaluating approach of comparing each component's point cloud data scanned by laser scanners and the ship's design data formatted in CAD became possible and practical.

In the future, more experiments will be conducted on the other kinds of ship components; according to the actual need and each method's availability, the presented reformative methods may have different designs. The parameters of each method discussed in this paper will also be optimized to the different components.

Acknowledgments

This manuscript is an output of the Joint Study supported by NIPPON KAIJI KYOKAI (Class NK). The authors would like to thank UNICUS Co., Ltd., FARO Japan, Inc., for their contribution to the project.

References

[1] Biskup K, Arias P, Lorenzo H, Armesto J. Application of trerrestrial laser scanning for shipbuilding. In: Proceedings of ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007; 2007 Sep 12-14; Espoo, Finland; p. 56-61.

[2] Nakagaki N, Sugawara A, Hiekata K, Yamato H, Sun JY. Development of accuracy evaluation system for curved shell plates using laser scanners (2nd report). Journal of the Japan Society of Naval Architects and Ocean Engineers. 2013; 17: 169-176.

[3] Hiekata K, Yamato H, Oida Y, Enomoto M, Furukawa Y. Development and case studies of accuracy evaluation system for curved shell plates by laser scanner. Journal of Ship Production and Design. 2011; 27(2): 84-90.

[4] Pauly M, Gross M, Kobbelt L. Efficient simplification of point-sampled surfaces. In: Proceedings of the Conference on IEEE Visualization; 2002 Oct 27–Nov 1; Boston, MA; p. 163-170.

[5] Forsyth D, Ponce J. Computer vision: a modern approach.. Upper Saddle River (NJ): Prentice Hall Professional Technical Reference; 2007.

[6] Gross M, Pfister H. Point–based graphics. Burlington (MA): Morgan Kaufmann; 2007.

[7] Weingarten JW, Gruener G, Siegwart R. Probabilistic plane fitting in 3D and an application to robotic mapping. In: Proceedings IEEE International Conference on Robotics and Automation (ICRA); 2004 Apr 26- May 1; Zürich, Switzerland; p. 927-932.

[8] Zhang X, Li H, Cheng Z. Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures.

In: Proceedings of AsiaGraph 2008; 2008 Oct 23-26; Tokyo, Japan; p. 72-79.

[9] Kalogerakis E, Nowrouzezahrai D, Simari P, Singh K. Extracting lines of curvature from noisy point clouds. Computer-Aided Design. 2009; 41(4): 282-292.

[10] Piegl L, Tiller W. The NURBS book. 2nd ed. New York: Springer; 1997.

[11] Okuda H, Kitaaki Y, Hashimoto M, Kaneko S. Fast and high-precision 3-D registration algorithm using hierarchical M-ICP. The Institute of Electronics, Information and Communication Engineers, Technical Report of IEICE. 2004 Sep 10; 8 p. Report No. 2004-CVIM-145.

[12] Besl PJ, McKay ND. Method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1992; 14(2): 239-256.

[13] Cristobal G, Schelkens P, Thienpont H. Optical and digital image processing. Hoboken (NJ): John Wiley & Sons; 2011.

[14] Zhang Z. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision. 1994; 13(2): 119-152.

[15] FARO [Internet]. FARO Laser Scanner Focus 3D; c2011 [cited 2014 Apr 22] Available from: http://www.globalspec.com/datasheets/579/FARO/F7DD1C91-CE09-4A0A-AB9C-71EC273AC1FA