

ACCURATE REGISTRATON OF MMS POINT CLOUDS OF URBAN AREAS USING TRAJECTORY

S. Takai^a, H. Date^a, S. Kanai^a, Y. Niina^b, K.Oda^b, T.Ikeda^b

^a Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan
s_takai@sdm.ssi.ist.hokudai.ac.jp, (hdate, kanai)@ssi.ist.hokudai.ac.jp

^b Asia Air Survey Co., Ltd, Kawasaki-shi 215-0004, Japan
ysh.niina (kz.oda, tty.ikeda)@ajiko.co.jp

KEY WORDS: Registration, Point cloud, Urban, Mapping, Laser scanning

ABSTRACT:

Recently, by Mobile Mapping System (MMS) with laser scanners, a GPS and IMU (Inertial Measurement Unit), 3D point clouds of urban areas (MMS point clouds) are easily acquired. When the same areas are scanned several times by the MMS, the point clouds often have differences in the range of several hundreds of millimetres. Such differences are caused by inertial drifts of IMU and losses of GPS signals in urban areas. In this paper, we propose an automatic accurate registration method of MMS point clouds using a new variant of ICP (Iterative Closest Point) algorithm for MMS point clouds and trajectory modification. Our method consists of four steps. Firstly, some trajectory points are automatically extracted by analyzing the trajectory. Secondly, the differences of point clouds are derived at the extracted trajectory points in the overlapping scan region by our new ICP algorithm which minimizes point-to-plane and point-to-point distances simultaneously and filters incorrect correspondences based on a point classification by PCA (Principle Component Analysis). Thirdly, the modified positions and rotation parameters at all extracted trajectory points are derived by a least squares method for positioning and registration constraints. Finally, each point in the point clouds is modified by coordinate transformations which are derived from linear interpolation of the modified positions and rotation parameters of the extracted trajectory points. Our method was applied to MMS point clouds and trajectory and the performances were evaluated.

1. INTRODUCTION

Mobile Mapping System (MMS) with laser scanners, GPS and IMU (Inertial Measurement Unit) contributes easy acquisitions of 3D point clouds of urban areas (MMS point clouds). When the same areas are scanned by MMS several times, MMS point clouds often have differences in the range of several hundreds of millimetres. Such differences are caused by inertial drifts in IMU and losses of GPS signals in urban areas. Therefore, an automatic registration method which accurately and effectively modifies these differences is strongly required.

Many point cloud registration algorithms have been proposed, for example 4PCS (Aiger et al., 2008) and ICP (Iterative Closest Point) algorithm (Besl and McKay, 1992, Chen and Medioni., 1992). The ICP is especially effective for correcting the differences in MMS point clouds, because magnitude of the differences is relatively small compared with the size of point clouds and the point clouds have similar poses. ICP algorithm provides an accurate registration between two point clouds by iteratively minimizing registration errors which are squared distances between corresponding points in each point clouds. Many variants of ICP about the selection of corresponding points and error metrics have been proposed (Al-Durgham et al., 2011). Rusinkiewicz et al. (Rusinkiewicz and Levoy, 2001) evaluated some variants of ICP algorithms, and concluded that ICP using point-to-plane distances for error metric (point-to-plane ICP) is more accurate than the standard ICP which minimizes point-to-point distances. Ridene et al. (Ridene and Goulette, 2009) propose a registration method for MMS point clouds. In their method, MMS point clouds are divided into blocks and point-to-plane ICP is applied to each block pair. However in this method, gaps between neighboring blocks may occur. In addition, when their method is applied to multipath point clouds, accumulated errors sometimes occur. Gressin et al. (Gressin et al., 2012) propose a MMS point cloud registration

method based on trajectory. In their method, firstly, the differences are derived by an ICP algorithm at several trajectory points. Next, MMS trajectory is modified by the least squares method based on position and registration constraints. Finally, point clouds are registered according to the modified trajectory. Therefore, this registration method provides a better result because it performs continuous registration of point clouds along the trajectory. However in this method, point-to-point ICP algorithm is used, and the rotation is not considered in the registration process. Moreover, the accuracy of their method has not been evaluated.

In this paper, we propose an automatic accurate registration method for MMS point clouds using a new variant of ICP algorithm for MMS point clouds and trajectory modification and we evaluate the registration accuracy of our method. Our method follows three extensions from the method proposed by Gressin et al. (Gressin et al., 2012). :

- Adaptive extraction of trajectory points for calculating differences based on trajectory analysis.
- Development of a new variant of ICP algorithm for MMS point clouds to perform an accurate registration.
- Introducing rotations to the trajectory modification and point cloud registration process for accurate registration

2. OUR REGISTRATION METHOD

2.1 Data Set

In this paper, MMS point clouds and MMS trajectory are denoted by P and Q respectively. The point cloud is a set of laser scanned points $P = \{(\mathbf{p}_i, t_i) | \mathbf{p}_i = (x_i, y_i, z_i), i = 1, \dots, n\}$ and each point i has a position (x_i, y_i, z_i) in a world coordinate system. Also, each point i is acquired at the GPS time t_i .

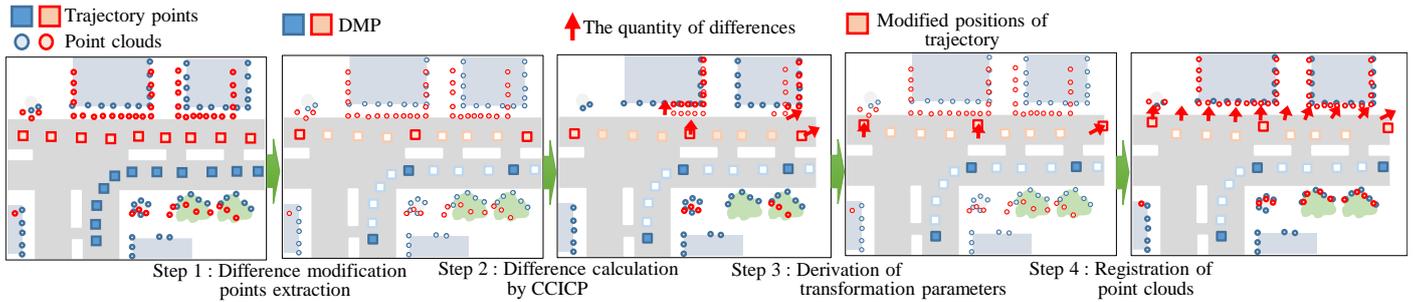


Figure 1. The overview of our method

Similarly, the trajectory $Q = \{(\mathbf{q}_j, \mathbf{c}_j, t_j) | \mathbf{q}_j = (x_j, y_j, z_j), \mathbf{c}_j = (\theta_j, \phi_j, \psi_j), j = 1, \dots, m (m < n), t_j < t_{j+1}\}$ is a set of points and each point j has a position (x_j, y_j, z_j) and roll pitch yaw angles $(\theta_j, \phi_j, \psi_j)$ of MMS. Additionally, each point j is acquired at the GPS time t_j . In our test data, the maximum differences between the overlapping point clouds are 400mm in a vertical direction and 200mm in a horizontal direction.

2.2 Overview of the proposed registration method

Figure 1 shows the overview of our method. Our method consists of the following four steps.

Step 1 : Difference Modification Points extraction

Difference Modification Points (DMPs) for trajectory modification are automatically extracted from all the given trajectory points Q by analyzing accelerations and angular velocity of MMS and trajectory intersections. This step reduces the number of redundant difference calculations in the step2.

Step 2 : Difference calculation by CCICP algorithm

The differences at each DMP in the overlapping regions are derived by the CCICP (Classification and Combined ICP) algorithm. The CCICP algorithm is a new variant of ICP algorithm for MMS point clouds which minimizes point-to-plane and point-to-point distances, simultaneously, and also filters incorrect correspondences based on point classification by PCA (Principle Component Analysis).

Step 3 : Derivation of transformation parameters

Translation and rotation parameters for all DMPs are derived as least squares solutions for absolute positioning constraint, relative positioning constraint, and registration constraint.

Step 4 : Registration of point clouds

Each point position \mathbf{p}_i in the point cloud is modified by coordinate transformations which are derived from linear interpolation of the modified positions of DMPs and the derived rotation parameters.

2.3 Difference Modification Points extraction

The differences often occur at areas where MMS velocity and orientation change drastically. Therefore, in these areas, the differences are non-linear and point clouds may be distorted. In addition, point clouds at intersecting trajectory regions should be registered accurately. From these points of view, in our method, DMPs are automatically extracted from all given trajectory points Q by analyzing accelerations and turning speeds of MMS and trajectory intersections. DMP extraction consists of following two steps.

Step A : Selecting Candidates of DMP

Velocity changing points Q_a are recognized by thresholding for

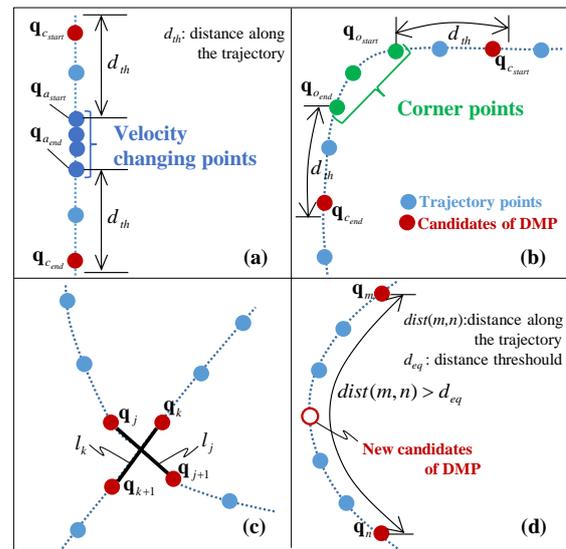


Figure 2. Selecting Candidates of DMP (StepA)

acceleration as follows: $Q_a = \{\mathbf{q}_j | |v_j - v_{j-1}| / (t_j - t_{j-1}) \geq a_{th}\}$. Where a_{th} is a threshold for acceleration, v_j is a velocity of MMS at a trajectory point j : $v_j = \|\mathbf{q}_j - \mathbf{q}_{j-1}\| / (t_j - t_{j-1})$. Orientation changing points Q_o are recognized by thresholding for angular velocity as follows: $Q_o = \{\mathbf{q}_j | (\psi_j - \psi_{j-1}) / (t_j - t_{j-1}) \geq \omega_{th}\}$. Where ω_{th} is a threshold for angular velocity, ψ_j is a yaw angle of MMS at trajectory point j . Candidates of DMP for each velocity changing point sequence in Q_a are determined by following criterion as shown in Figure 2(a). Let $\mathbf{q}_{a_{start}}, \mathbf{q}_{a_{end}}$ be the starting and ending points position of a sequence of consecutive velocity changing points. Candidates $\mathbf{q}_{c_{start}}, \mathbf{q}_{c_{end}}$ are selected as the farthest back and forth points position from $\mathbf{q}_{a_{start}}, \mathbf{q}_{a_{end}}$ within d_{th} along trajectory points position. Similarly, candidates of DMP for each orientation changing point sequence in Q_o are determined as shown in Figure 2(b). Moreover, trajectory intersections are recognized by detecting intersection for line segments l_j and l_k which consist of consecutive trajectory points $j, j + 1$ and $k, k + 1$. If l_j intersects l_k , points $j, j + 1$ and $k, k + 1$ are selected as the candidate for the DMP shown in Figure 2(c). Finally, if the distance of consecutive candidates m and n along the trajectory $dist(m, n)$ exceeds the distance threshold d_{eq} , new trajectory points are selected at regular intervals, as shown in Figure 2(d).

Step B : Determination of DMP and correspondences

Firstly, trajectory points are segmented at corner points as shown in Figure 3(a). Secondly, the corresponding point j' of a candidate j in segment J are detected as a nearest point within the distance d_c in the segments except for J and J 's neighbors as shown in Figure 3(b). Thirdly, trajectory is subdivided into blocks at regular intervals d_i ($d_i < d_{eq}$) (Figure 3(c)). Finally,

the set of DMPs R and the set of DMP correspondences Ω , are extracted as shown in Figure 3(d). If one or more candidates or pairs of candidates exist in a block, one candidate or pair of candidates is extracted.

2.4 Difference calculation by CCICP algorithm

2.4.1 Extracting local point clouds for ICP

Our new ICP algorithm is applied to local point clouds which are extracted by DMPs and their GPS times. Firstly, two trajectory points position, $\mathbf{q}_{p_s}, \mathbf{q}_{p_e}$ are selected as the farthest back and forth points position from each DMP within d_p along the trajectory points. Next, the point i within the time range ($t_{p_s} \leq t_i \leq t_{p_e}$) is extracted from the point clouds.

2.4.2 CCICP algorithm

The features of the CCICP algorithm are minimizing point-to-plane and point-to-point squared distances simultaneously and filtering incorrect correspondences based on point classification by PCA. These features are shown as follows.

A) Filtering incorrect correspondences

The points in the local point clouds are classified into *linear* points, *planar* points and *scatter* points depending on the results of the PCA. In the ICP, *linear-planar* and *scatter-planar* correspondences are rejected as incorrect correspondences for accurate registration by the CCICP algorithm. For example, corresponding points on electric cables and building facades or ones on utility poles and walls are not used in the ICP.

B) Minimizing point-to-plane and point-to-point distances

Point-to-plane and point-to-point distances are minimized simultaneously. Point-to-plane distance minimization is applied to *planar-planar* correspondences, because reliable normal estimations can be performed at *planar* points. Point-to-point distance minimization is applied to the other correspondences. Therefore points on roads, building facades and walls are registered accurately, and points on utility poles and electric cables are also registered. In this paper, point-to-plane distance minimization problem is linearized and solved using the method of Low (Low, 2004).

The CCICP algorithm follows five steps. In the following algorithm, S and T indicate a source point cloud and a target point cloud. \mathbf{p}_i^S and \mathbf{p}_i^T indicate a position of point $i \in S$ and a position of point $i \in T$.

Step 1 : Selection

Subset of S for distance calculation and rigid transformation derivation is selected. The selection is done by random sampling ($a_{sample}[\%]$ of S are selected). The sampled point set is represented by S' .

Step 2 : Point classification

Each point i in S' is classified into *linear* points, *planar* points and *scatter* points by PCA (Demantke et al., 2011). In this method, local point distributions of each point i are evaluated by eigenvalues λ_1^i, λ_2^i and λ_3^i ($\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i$) and unit eigenvectors $\mathbf{e}_1^i, \mathbf{e}_2^i$ and \mathbf{e}_3^i , of the variance-covariance matrix \mathbf{M}_i which is calculated from the position of point i and its neighbors (points within distance r_{PCA} from i). In order to evaluate the local point distributions, the magnitude relations of the eigenvalues are investigated using the following three values: $a_{1D}^i = \lambda_1^i - \lambda_2^i$, $a_{2D}^i = \lambda_2^i - \lambda_3^i$, $a_{3D}^i = \lambda_3^i$. If a_{1D}^i is larger than the others, the local point distribution of point i is recognized as *linear*. If a_{2D}^i is larger than the others, it is

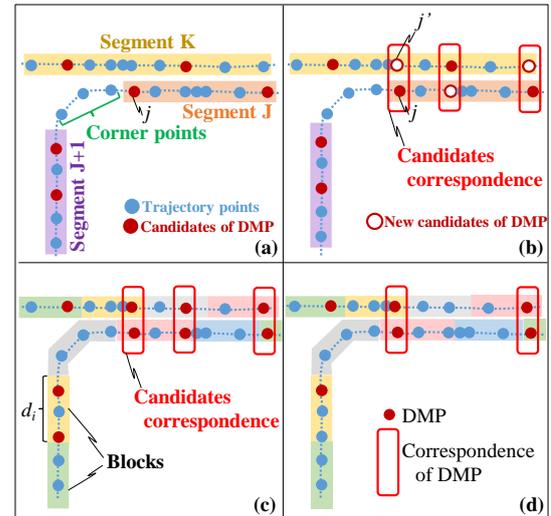


Figure 3. Determination of DMP and correspondences detection (StepB)

recognized as *planar*. In this case, eigenvector \mathbf{e}_3^i is kept as the normal vector of i . In addition, if a_{3D}^i is larger than the others, it recognized as *scatter*.

Step 3 : Matching

The point i' in T which is closest to point $i \in S'$ is extracted as a corresponding point of point i . If i' is found, it is classified by PCA.

Step 4 : Rejecting

Correspondences consisting of a *linear* point and a *planar* point or a *planar* point and a *scatter* point are rejected. A set of resulting correspondences is denoted by U , and a set of *planar-planar* correspondences is denoted by V .

Step 5 : Minimizing

The average of point-to-plane squared distance E_{PT_PL} is derived from equation (1).

$$E_{PT_PL} = \frac{1}{|V|} \sum_{(i,i') \in V} |(\mathbf{T}\mathbf{p}_i^{S*} - \mathbf{p}_{i'}^{T*}) \cdot \mathbf{n}_{i'}|^2 \quad (1)$$

Where $\mathbf{n}_{i'}$ = an normal vector of point i' in a homogeneous coordinate system

\mathbf{T} = a transformation matrix in a homogeneous coordinate system

$\mathbf{p}_i^{S*}, \mathbf{p}_{i'}^{T*}$ = positions of $\mathbf{p}_i^S, \mathbf{p}_{i'}^T$ in a homogeneous coordinate system

The averages of point-to-point squared distances E_{PT_PT} are derived from equation (2).

$$E_{PT_PT} = \frac{1}{|U \setminus V|} \sum_{(i,i') \in U \setminus V} \|\mathbf{T}\mathbf{p}_i^{S*} - \mathbf{p}_{i'}^{T*}\|^2 \quad (2)$$

In the equation (1), when rotation angles of \mathbf{T} (α, β, γ) about x, y, z axis are nearly zero ($\alpha, \beta, \gamma \approx 0$), $(\mathbf{T}\mathbf{p}_i^{S*} - \mathbf{p}_{i'}^{T*}) \cdot \mathbf{n}_{i'}$ in equation (1) is reformulated as equation (3).

$$(\mathbf{T}\mathbf{p}_i^{S^*} - \mathbf{p}_i^{T^*}) \cdot \mathbf{n}_i' = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \\ 1 \end{bmatrix} - \begin{bmatrix} p_{i'x} \\ p_{i'y} \\ p_{i'z} \\ 1 \end{bmatrix} \cdot \begin{bmatrix} n_{i'x} \\ n_{i'y} \\ n_{i'z} \\ 0 \end{bmatrix} \quad (3)$$

$$= a_{i1}\alpha + a_{i2}\beta + a_{i3}\gamma + n_{i'x}t_x + n_{i'y}t_y + n_{i'z}t_z + (\mathbf{p}_i^{S^*} - \mathbf{p}_i^{T^*}) \cdot \mathbf{n}_i'$$

Where $a_{i1} = n_{i'z}p_{iy} - n_{i'y}p_{iz}$
 $a_{i2} = n_{i'x}p_{iz} - n_{i'z}p_{ix}$
 $a_{i3} = n_{i'y}p_{ix} - n_{i'x}p_{iy}$
 $t_x, t_y, t_z =$ translation parameters of \mathbf{T}

For all correspondences, equation (3) is denoted by equation (4) using a matrix representation.

$$\mathbf{Ax} - \mathbf{b} \quad (4)$$

$$\text{Where } \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & n_{1'x} & n_{1'y} & n_{1'z} \\ a_{21} & a_{22} & a_{23} & n_{2'x} & n_{2'y} & n_{2'z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{|V|1} & a_{|V|2} & a_{|V|3} & n_{|V|'x} & n_{|V|'y} & n_{|V|'z} \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} (\mathbf{p}_1 - \mathbf{p}_1') \cdot \mathbf{n}_1' \\ (\mathbf{p}_2 - \mathbf{p}_2') \cdot \mathbf{n}_2' \\ \vdots \\ (\mathbf{p}_{|V|} - \mathbf{p}_{|V|'}) \cdot \mathbf{n}_{|V|'} \end{bmatrix}$$

$$\mathbf{x} = [\alpha \quad \beta \quad \gamma \quad t_x \quad t_y \quad t_z]^T$$

Similarly, $\mathbf{T}\mathbf{p}_i^{S^*} - \mathbf{p}_i^{T^*}$ in equation (2) is reformulated as equation (5).

$$\mathbf{T}\mathbf{p}_i^{S^*} - \mathbf{p}_i^{T^*} = \begin{bmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \\ 1 \end{bmatrix} - \begin{bmatrix} p_{i'x} \\ p_{i'y} \\ p_{i'z} \\ 1 \end{bmatrix} \quad (5)$$

$$= \widehat{\mathbf{A}}_i \mathbf{x} - \widehat{\mathbf{b}}_i$$

$$\text{Where } \widehat{\mathbf{A}}_i = \begin{bmatrix} 0 & p_{iz} & -p_{iy} & 1 & 0 & 0 \\ -p_{iz} & 0 & p_{ix} & 0 & 1 & 0 \\ p_{iy} & -p_{ix} & 0 & 0 & 0 & 1 \\ p_{i'x} - p_{ix} & & & & & \\ p_{i'y} - p_{iy} & & & & & \\ p_{i'z} - p_{iz} & & & & & \end{bmatrix}$$

$$\widehat{\mathbf{b}}_i = \begin{bmatrix} p_{i'x} - p_{ix} \\ p_{i'y} - p_{iy} \\ p_{i'z} - p_{iz} \end{bmatrix}$$

From equations (4) and (5), the parameters of the coordinate transformation $\mathbf{x}_{opt} = (\boldsymbol{\alpha}, \mathbf{t}^{ICP})$, which minimizes E_{PT_PL} and E_{PT_PT} simultaneously, are derived from equation (6). Where, $\boldsymbol{\alpha}$ is a rotation parameter about the x, y, z axis and \mathbf{t}^{ICP} is translation parameter.

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \widehat{\mathbf{A}}' \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \widehat{\mathbf{b}}' \end{bmatrix} \right\|^2 \quad (6)$$

$$= \arg \min_{\mathbf{x}} \|\mathbf{A}'\mathbf{x} - \mathbf{b}'\|^2$$

$$\text{Where } \widehat{\mathbf{A}}' = [\widehat{\mathbf{A}}_1 \quad \dots \quad \widehat{\mathbf{A}}_{|U \setminus V|}]^T$$

$$\widehat{\mathbf{b}}' = [\widehat{\mathbf{b}}_1 \quad \dots \quad \widehat{\mathbf{b}}_{|U \setminus V|}]^T$$

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} \\ \widehat{\mathbf{A}}' \end{bmatrix}$$

$$\mathbf{b}' = \begin{bmatrix} \mathbf{b} \\ \widehat{\mathbf{b}}' \end{bmatrix}$$

The solution \mathbf{x}_{opt} is obtained by solving $\mathbf{A}'\mathbf{x} = \mathbf{b}'$ using pseudo inverse matrix of \mathbf{A}' . At each iteration, source point cloud S is transformed by the \mathbf{x}_{opt} . Here if the number of iterations from step3 to step5 reaches threshold n_{ITTR} or the sum of E_{PT_PT} and E_{PL_PT} is less than threshold δ_E or the difference between current and previous sums in the iterations is less than $\delta_{\Delta E}$, algorithm is terminated, otherwise return to step3. Therefore the amount of difference is determined by a concatenated transformation of \mathbf{x}_{opt} at each iteration.

2.5 Derivation of transformation parameters

In order to calculate the transformation parameters for all DMPs, the modified position and rotation parameters are derived by solving the overconstrained equations defined by absolute positioning constraint, relative positioning constraint, and registration constraint. The constraints are defined by equations (7)-(9).

• Absolute positioning constraint:

$$(j \in R), \begin{bmatrix} \mathbf{q}_j^* \\ \boldsymbol{\alpha}_j^* \end{bmatrix} = \begin{bmatrix} \mathbf{q}_j \\ \mathbf{0} \end{bmatrix} \quad (7)$$

• Relative positioning constraint:

$$(j \in R), \begin{bmatrix} \mathbf{q}_j^* \\ \boldsymbol{\alpha}_j^* \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{j-1}^* \\ \boldsymbol{\alpha}_{j-1}^* \end{bmatrix} = \begin{bmatrix} \mathbf{q}_j \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{q}_{j-1} \\ \mathbf{0} \end{bmatrix} \quad (8)$$

• Registration constraint:

$$(j \in \Omega_S), \begin{bmatrix} \mathbf{q}_j^* \\ \boldsymbol{\alpha}_j^* \end{bmatrix} = \begin{bmatrix} \mathbf{q}_j \\ \boldsymbol{\alpha}_j \end{bmatrix} + \begin{bmatrix} \mathbf{t}_j^{ICP} \\ \mathbf{0} \end{bmatrix} \quad (9)$$

Where \mathbf{q}_j^* is a modified position of \mathbf{q}_j which is derived by a least square method

Ω_S is a set of source DMPs

$\boldsymbol{\alpha}_j^*$ is rotation angles around x, y, z axis which are derived by a least square method

2.6 Registration of point clouds

Rotation and translation parameters (\mathbf{r}_i, Δ_i) for point i in point clouds are derived by linear interpolation of modified position and rotation parameters of neighboring DMPs $j-1$ and j ($t_{j-1} \leq t_i \leq t_j$). The linear interpolation is denoted by equation (10) and equation (11).

$$\mathbf{r}_i = \boldsymbol{\alpha}_j^* + \frac{t_i - t_j}{t_j - t_{j-1}} (\boldsymbol{\alpha}_j^* - \boldsymbol{\alpha}_{j-1}^*) \quad (10)$$

$$\Delta_i = \Delta \mathbf{q}_j + \frac{t_i - t_j}{t_j - t_{j-1}} (\Delta \mathbf{q}_j - \Delta \mathbf{q}_{j-1}) \quad (11)$$

Where $\Delta \mathbf{q}_j = \mathbf{q}_j^* - \mathbf{q}_j$

Finally, position of each \mathbf{p}_i in the point clouds is modified by the coordinate transformation using \mathbf{r}_i and Δ_i .

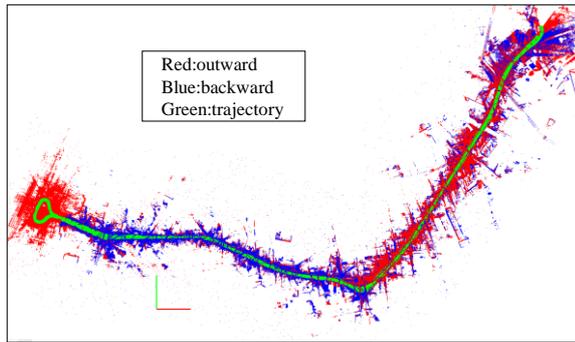


Figure 4. MMS point clouds and trajectory

Table 5. Used parameters

Step1	Acceleration threshold a_{th}	1.5 m/s ²
	Turning speed threshold ω_{th}	8.5 deg/s
	Distance threshold d_{th}	10 m
	Equal interval threshold d_{eq}	250 m
	DMP correspondence search distance d_c	20 m
	Block division interval d_t	10 m
Step2	Point clouds extraction distance d_p	10 m
	Selecting rate a_{sample}	0.50 %
	PCA search radius r_{PCA}	0.30 m
	Limitation of iteration n_{ITR}	30
	ICP error threshold δ_E	$1.0 \times 10^{-7} m^2$
	ICP error difference threshold δ_F	$1.0 \times 10^{-14} m^2$
	Search radius of closest point r_d	0.70 m

3. RESULTS

3.1 Input data sets and parameters

Our method was applied to urban area point clouds and trajectory was acquired by MMS (GeoMasterNeo). The trajectory is shown in Figure 4 and consist of 2,192 points. Similarly, the point clouds are shown in Figure 4 and consist of 139,390,404 points. The travel distance of MMS is about 3.0 km. Used parameters in our experiments are shown in Table 5. Some parameters, for example a_{th} and ω_{th} , were determined by common knowledge about running condition of MMS, on the other hand, other parameters were determined by experimentally.

3.2 Registration results and error evaluation

The results of our method are shown in Figures 6-8. Figure 6 shows DMPs which are extracted from trajectory points. The number of DMPs was 40, and that of DMP pairs was 19. The left figures in Figure 7 show point clouds before registration at the positions A-C which are shown in Figure 6, and right figures show point clouds after registration. In the left figures of Figure 7(a) and (c), horizontal differences are visible before registration, on the other hand, the differences are modified after registration as shown in the right figures. Similarly, in the left figure of Figure 7(b), significant elevation differences are visible. On the other hand, differences are modified as shown in the right figures. From these results, our method can perform visually accurate registration of point clouds.

Our algorithm was implemented on a standard PC with Intel Core i7 3.30GHz CPU, 32GB RAM, GeForce GTX570 GPU. The computation time of registration was about 161sec. The

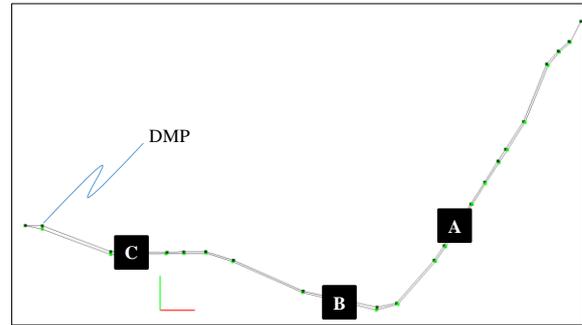
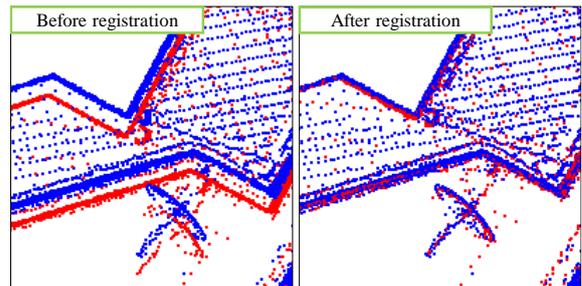
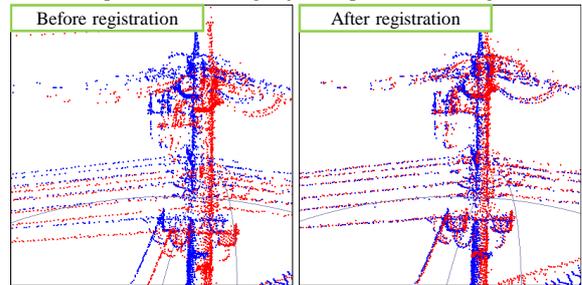


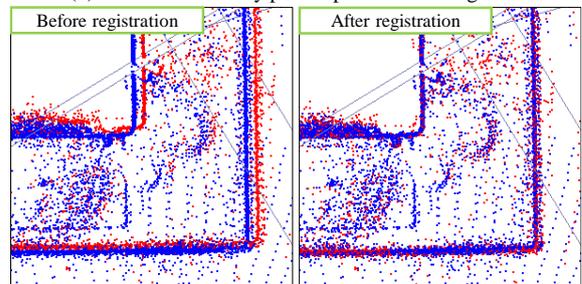
Figure 6. DMPs and evaluated positions



(a) Top view of building façade at position A in Figure 6



(b) Side view of utility pole at position B in Figure 6



(c) Top view of building façade at position C in Figure 6

Figure 7. Point clouds before and after registration

most of computation time was the difference calculation by CCICP, and it was about 133sec. Figure 8 shows the amount of difference with color. In the figures, the amounts of difference are derived by calculations of distances between each point in the outward point cloud and its closest point in the backward point cloud. The color bar is shown in the top of Figure 8. If the closest point does not exist inside of 400 mm area from each point, points are colored in black. In left figures in Figure 8, original point clouds have about 200-400mm differences. On the other hand, in the right figures, the differences are reduced to about less than 100mm at roads and building façades. Figure 9 shows the differences at position C using another color range. In this figure, as a result, differences are less than 50mm at roads and building façades after registration. These results show that an accurate registration of point clouds which reduce the differences among point clouds to less than 50mm was realized by our method. Moreover, the accuracy of the CCICP algorithm was compared with a standard ICP algorithm (random sample,

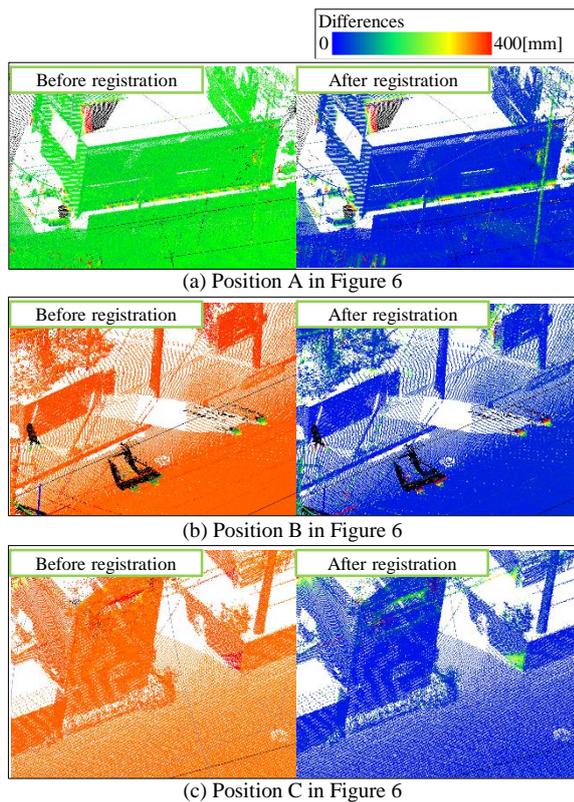


Figure 8. Colored differences before and after registration

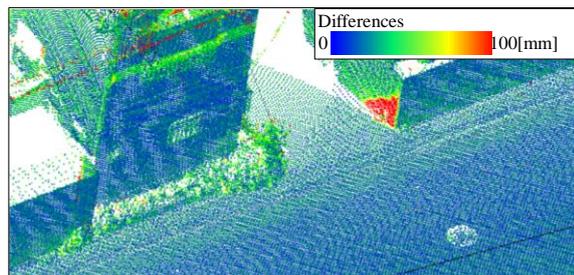


Figure 9. Detailed differences at position C

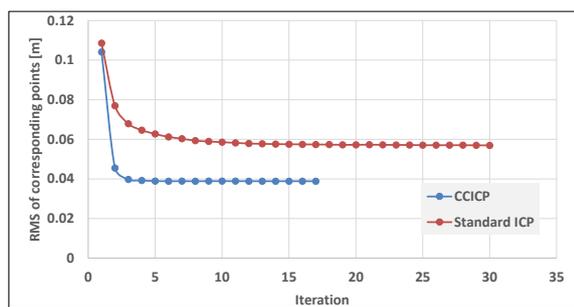


Figure 10. Comparing RMS values at each

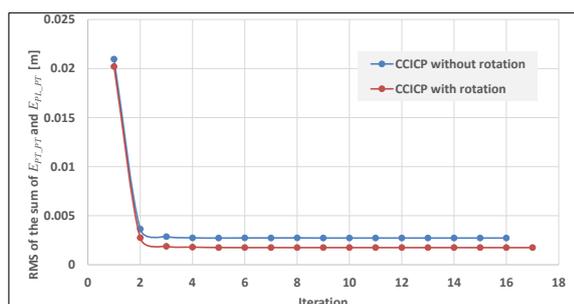


Figure 11. Effectiveness of rotation

no rejection, point-to-point, and a quaternion based solution method). To evaluate CCICP algorithm, a point cloud that consisted of about 5.3M points and another point cloud that consisted of about 5.9M points were used. The registration parameters are the same as the ones shown in Table 5. Figure 10 shows changes of two RMS values of corresponding point distances at each iteration of the ICP. In this figure, the RMS of CCICP is 40% smaller than one of the standard ICP, and this means that our ICP realizes more accurate registration than the standard ICP. The processing times were 78.3sec for CCICP and 83.3sec for a standard ICP. It was estimated that the main difference of the processing times was caused by a difference of the solution methods for minimization problem. Coordinate transformation parameters of our ICP was derived by solving simultaneous equation using pseudo-inverse matrices. On the other hand the parameter of the Standard ICP was derived by a quaternion based method.

Finally, the effectiveness of the rotation was evaluated as shown in Figure 11 using the same point clouds and settings for accuracy evaluation. Figure 11 shows RMS values which were derived from the sum of $E_{PT,PT}$ and $E_{PL,PT}$. In this figure, the RMS of CCICP with rotation is smaller than that of its CCICP without rotation. Therefore, introducing the rotation to CCICP is effective for accurate registration of MMS point clouds.

4. CONCLUSION

In this paper, an automatic accurate registration method of MMS point clouds using CCICP algorithm for MMS point clouds and modifying trajectory is proposed. The performance of the method is evaluated by applying it to MMS point clouds and a trajectory. Our method can reduce the differences among point clouds to less than 50mm at roads or building facades on average. Also, its computation time was about 161sec for our test data including 140M points and 2.2K trajectory points. Moreover, the accuracy of CCICP algorithm was also evaluated. It is concluded that our ICP algorithm was more accurate than the standard ICP algorithm. Future work includes automatic parameter determination.

REFERENCES

- Aiger, D., Mitra, N., and Cohen, D., 2008. 4-Points Congruent Sets for Robust Pairwise Surface Registration. *ACM Transactions on Graphics*, 27(3), pp. 1-10.
- Al-Durgham, M., Datchev, I., Habib, A., 2011. Analysis of Two Triangle-Based Multi-Surface Registration Algorithms of Irregular Point Clouds. *In ISPRS Workshop - Laser Scanning 2011*, 33 (Part 5/W12), pp. 61-66.
- Besl, P. and McKay, N. D., 1992. A method for registration of 3D shapes. *IEEE Trans. on PAMI*, 14(2), pp. 239-256.
- Chen, Y. and Medioni, G., 1992. Object Modeling by Registration of Multiple Range Images. *Image and Vision Computing*, 10(3), pp. 145-155.
- Demantke, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3D LiDAR Point Cloud. *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, 38 (Part 5/W12), (on CDROM).
- Gressin, A., Cannelle, B., Mallet, C., Papellard, J. P., 2012. Trajectory-based Registration of 3D Lidar Point Clouds Acquired with a Mobile Mapping System. *International Society for Photogrammetry and Remote Sensing Annals*, 1(3), pp. 117-122.
- Low, K. L., 2004. Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. Technical Report, Department of Computer Science, University of North Carolina at Chapel Hill, TR04-004.
- Ridene, T., Goulette, F., 2009. Co-registration of DSM and 3D Point Clouds Acquired by a Mobile Mapping System. *Special Issue on Mobile Mapping Technology*, pp. 824-838.
- Rusinkiewicz S., Levoy M., 2001. Efficient variants of the ICP algorithm. *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, pp. 145-152.