

AN AUTOMATIC PROCEDURE FOR MOBILE LASER SCANNING PLATFORM 6DOF TRAJECTORY ADJUSTMENT

Zille Hussnain*, Sander Oude Elberink, George Vosselman

ITC, Faculty of Geo-Information Science and Earth Observation of the University of Twente, Enschede, The Netherlands -
(s.z.hussnain, s.j.oudeelberink, george.vosselman)@utwente.nl

Commission I, WG I/9

KEY WORDS: 6DOF trajectory adjustment, B-spline, Mobile laser scanning point cloud, 3D tie points, aerial imagery

ABSTRACT:

In this paper, a method is presented to improve the MLS platform's trajectory for GNSS denied areas. The method comprises two major steps. The first step is based on a 2D image registration technique described in our previous publication. Internally, this registration technique first performs aerial to aerial image matching, this issues correspondences which enable to compute the 3D tie points by multiview triangulation. Similarly, it registers the rasterized Mobile Laser Scanning Point Cloud (MLSPC) patches with the multiple related aerial image patches. The later registration provides the correspondence between the aerial to aerial tie points and the MLSPC's 3D points. In the second step, which is described in this paper, a procedure utilizes three kinds of observations to improve the MLS platform's trajectory. The first type of observation is the set of 3D tie points computed automatically in the previous step (and are already available), the second type of observation is based on IMU readings and the third type of observation is soft-constraint over related pose parameters. In this situation, the 3D tie points are considered accurate and precise observations, since they provide both locally and globally strict constraints, whereas the IMU observations and soft-constraints only provide locally precise constraints. For 6DOF trajectory representation, first, the pose $[R, t]$ parameters are converted to 6 B-spline functions over time. Then for the trajectory adjustment, the coefficients of B-splines are updated from the established observations. We tested our method on an MLS data set acquired at a test area in Rotterdam, and verified the trajectory improvement by evaluation with independently and manually measured GCPs. After the adjustment, the trajectory has achieved the accuracy of RMSE $X=9$ cm, $Y=14$ cm and $Z=14$ cm. Analysing the error in the updated trajectory suggests that our procedure is effective at adjusting the 6DOF trajectory and to regenerate a reliable MLSPC product.

1. INTRODUCTION

Inaccurate GNSS measurements in an urban canyon can lead to error in the trajectory of Mobile Laser Scanning (MLS) system. Julge et al. 2017 reported that the GNSS discrepancies can reach up to several decimetres. An RMSE of 20 cm however, can be achieved by using fewer control points. Kukko 2013 demonstrated that the GNSS accuracy can worsen to more than 50 cm during an outage of GNSS signals. Consequently, the MLSPC positioning quality suffers from the inaccurate trajectory and the 3D data becomes unreliable. In an ideal condition, without any GNSS signal outage or multipath effect, the state-of-the-art Mobile Mapping (MM) platforms can achieve 2-3 cm accuracy, estimated by Haala et al. 2008, Kaartinen et al. 2012. However, this is not possible in urban canyons. In this situation, the task of trajectory correction is very crucial because it turns an inaccurate MLSPC into a reliable commercial product. So, commercial software, first tries to correct the point cloud by automatic registration of multiple passes, similar to the technique reported by Levinson et al. 2007, Ding et al. 2007, Zhao 2011. In this vein, Hunter et al. 2006 and Bornaz et al. 2003 proposed a consecutive strip adjustment to improve the misalignments in the MLS data sets. Similarly, Bosse et al. 2009 described a scan-matching method based on iterative closest points (ICP) to recover an accurate trajectory. However, still during long-term GNSS signal outages the final achieved accuracy remains in metres, as expressed by Chiang et al. 2008. However, all of these

techniques can only be used for MLS data adjustment to increase the relative accuracy, and additionally requires data that has multiple scans/passes of the same scene. Moreover, these techniques will still require manual adjustment to achieve the desired accuracy. Similarly, to attain relative accuracy, in commercial MLS data acquisition workflow, data overlapping is intentionally performed for automatic registration while post-processing. However, during mobile mapping, it is highly desirable to scan an area once and as quickly as possible.

The second step in practice is to measure the GCPs in the target area and then carefully handpick their correspondences in the MLSPC. Although some latest software provide assistant to automatically detect such landmarks, the final decision and effort remains with a human operator. The last step adjusts and improves the erroneous trajectory based on the established correspondences and subsequently regenerates the reliable MLSPC. However, manual acquisition and handpicking of the GCPs is very labour intensive, costly and error-prone. Therefore, an automatic method is desired to replace the manual correction, especially to improve the MLS platform's trajectory. This paper describes an automatic 6DOF trajectory adjustment technique that eventually enables any commercial MLSPC processing software to regenerate the accurate and improve MLSPC positioning. Thereby, turning the entire correction workflow fully

* Corresponding author

automatic. Hence, the total cost and the effort to obtain a reliable MLSPC product decreases.

In the remainder of this paper, Section 2 first describes the literature, which aims at MLSPC correction, then discusses the literature, which is related to the B-spline based trajectory estimation for the MM applications. Section 3 describes our method of the B-spline based trajectory adjustment using available observations. Section 4 provides the results of the experiment over the data for three different test cases. For simplicity, we will use the following terminology for the remainder of this paper;

A2A tie points: 3D point calculated from matching multiple airborne images and multiview triangulation.

A2P tie points: Points in the MLSPC, which correspond to the A2A tie points and, hence, establish the link between the point cloud and the airborne images.

Check points: 3D points measured with GNSS and identified in the point cloud to check the accuracy of the reconstructed trajectory.

2. RELATED LITERATURE

Recently, Javanmardi et al. 2018 proposed a technique for MLS platform localization based on the 'abstract maps'. However, this technique utilizes accurate maps generated from an accurate prior point cloud. Moreover, the abstract maps are an estimation of the structures in the prior point cloud, which can introduce errors. Furthermore, the localization accuracy will be always lower than the prior point cloud accuracy. In our case, we do not consider that a (prior) accurate MLSPC is already available. In a previous publication, Javanmardi et al. 2017 have shown the correction of the MLSPC by using multiple reference data sets, including aerial images. However, their approach constrained the registration problem to the only 2DOF, which only works in an ideal scenario when the error does not occur in remaining coordinates.

For the MLS data correction by improving the position of MM platform, Gao et al. 2015 have improved the Mobile Laser Scanning (MLS) data accuracy by its raster image registration with UAV's imagery using Harris corner keypoint detection and edge-based template matching. They have reported the RMS $-\Delta X=8.6$ cm $-\Delta Y=6.3$ cm $-\Delta Z=10.6$ cm in the corrected point cloud. However, they reported the relative accuracy measure using control points and check points on the adjusted data, also they directly perform the adjustment to the point cloud and did not estimate the accurate trajectory. Wolcott et al. 2014 developed an image-based navigation for self-driving systems, the mobile mapping camera images are registered using prior 3D lidar by maximizing the normalized mutual information. This approach achieved an RMS error, longitudinal 19.1–45.4 cm, and lateral 14.3–20.5 cm. Earlier, Kümmerle et al. 2011 developed a SLAM procedure using the MCL approach by considering the MLS data as local observation and the aerial image as a global reference for every node in the graph. An overall accuracy of 20 cm was achieved in the five test MLS data sets.

Many researchers have adopted the B-spline representation of 6DOF trajectory for mobile mapping systems. Among them, for a visual odometry application,

Patron-Perez et al. 2015 unified the discrete camera poses with continuous unsynchronized IMU observations to estimate the continuous camera trajectory. They used a rolling shutter camera model, which introduced individual time stamp for each pixel, which is similar to MLS observations, since each point cloud point is associated with different GNSS time. However, they only presented results for simulated data sets.

Among one of the main benefits, it is convenient to update the B-spline based on the changes in the local control points, since each B-spline is nonzero over a certain interval and the updated control points only change coefficients of related B-splines. For micro aerial vehicles, Usenko et al. 2017 updated the local part of the B-spline trajectory, when an unmodelled obstacle arrived in the pre-processed global trajectory. However, the main purpose of this study was to show that the developed system could accommodate the real world dynamics into the trajectory and not the achieved accuracy of the positioning.

Vosselman 2014 designed an indoor laser scanner system to estimate the 6DOF B-spline based trajectory using SLAM. The developed technique showed that the constraint derived from the indoor wall structures (or planes) of simulated indoor environment can be used to estimate the 6DOF B-spline based trajectory. Likewise, in our case, the constraints come from the 3D A2P tie points, which are forward intersected from aerial image correspondences.

In this paper, we rely on the aerial imagery with accurate exterior and interior orientation, which is easily available for the MLSPC at hand. Moreover, we believe that the highly accurate 2D image correspondences can yield highly reliable 3D A2A tie points, which can be used for the trajectory adjustment. Moreover, we use the real world MLS data for an outdoor environment.

3. TRAJECTORY ADJUSTMENT

As mentioned in section 1 that the Kalman trajectory is affected by erroneous GNSS positioning in urban canyons. Fortunately, the measurements from the other sensors are not affected. For example, the IMU sensor observations provide accurate estimations of the relative transmission of the motion. Moreover, the A2P tie points provide the constraint that satisfies both local and global positioning consistency. It is important to note that the 3D A2A tie points are not available consistently; instead their availability is dependent on the distribution of the road markings. Therefore, it is necessary to utilize the IMU observations during the A2A tie points' unavailability. Furthermore, we apply the pose integrity check by introducing soft constraints on the heading and pitch of the MLS platform, which essentially reduces the problem to the estimation of a 4DOF trajectory. We use only above-mentioned observations because they can be obtained without any manual intervention.

Towards the parameterization of the observation equations, we start with the observation of the 3D point cloud point. The laser scanner observes the point X^C in local or car coordinate system, which can be transformed to X^W in world coordinate system by Eq. 1,

$$X^W = R_C^W(t) X^C + T_C^W(t) \quad (1)$$

These coordinate systems are related to each other by a rotation $R_C^W(t)$ and translation $T_C^W(t)$ that vary over time t . The translation describes the location of the car in the world coordinate system and the rotation the attitude of the car in the world coordinate system. The rotation matrix is described by three angles $\omega(t), \varphi(t), \kappa(t)$, which are modelled by B-splines. For example, the roll angle $\omega(t)$ can be modelled by B-spline like in Eq. 2.

$$\omega(t) = \sum_i \alpha_{\omega,i} B_i(t) \quad (2)$$

The translation vector is described by translations along the three axes, T_X, T_Y, T_Z , which are also modelled by splines. Following we describe each observation individually and its adjustment to B-splines.

3.1 A2P tie point observation

In our previous research work, Hussnain et al. 2016 describe the registration between the MLSPC and the aerial images. The same technique can be used for aerial to aerial image registration. In this case, the 3D A2A tie points (X_{AI}^W) are reckoned by the multiview triangulation using aerial to aerial correspondences. The same points A2P (X_{PC}^W) are identified in the MLS point cloud by aerial to rasterized MLSPC image registration. The coordinates of these observations are denoted X_{AI}^W and X_{PC}^W in the world coordinate system where the lower indices AI and PC denote the source of the observation, i.e. aerial images and point cloud, respectively.

The original GNSS and IMU data have been processed by a Kalman filter to obtain an estimate of the rotation and translation between the car and world coordinate system. They are denoted $R_{C,Kalman}^W(t)$ and $T_{C,Kalman}^W(t)$. The results of the Kalman filtering are used to obtain the approximate spline coefficients of the six pose parameters over time.

The Kalman filter results have been used to calculate the original MLSPC. Because the GNSS data was unreliable, we want to re-estimate the rotation $R_C^W(t)$ and translation $T_C^W(t)$ based on observed A2A tie points and the original IMU observations. We use equation Eq. 3,

$$X_{PC}^C = R_{C,Kalman}^{W^T}(t) [X_{PC}^W - T_{C,Kalman}^W(t)] \quad (3)$$

to retrieve the original point cloud coordinates in the car coordinate system. The relationship between the A2A tie points from the aerial images in the world coordinate system and the same A2P tie points from the point cloud in the car coordinate system, is described by Eq. 4a,

$$X_{AI}^W = R_C^W(t) X_{PC}^C + T_C^W(t) \quad (4a)$$

This equation is linearized. The upper index 0 denotes an approximate value. In the first iteration of an iterative B-spline adjustment procedure, the output of the Kalman filter is used for the approximate rotation and translation. The time dependency (t) is omitted below to shorten the expression.

$$\begin{aligned} X_{AI}^W - R_C^W X_{PC}^C - T_C^W = & \begin{pmatrix} \sum_i \Delta\alpha_{T_x,i} B_i \\ \sum_i \Delta\alpha_{T_y,i} B_i \\ \sum_i \Delta\alpha_{T_z,i} B_i \end{pmatrix} \\ & + \frac{\partial R_C^W}{\partial \omega} X_{PC}^C \sum_i \Delta\alpha_{\omega,i} B_i \\ & + \frac{\partial R_C^W}{\partial \varphi} X_{PC}^C \sum_i \Delta\alpha_{\varphi,i} B_i \\ & + \frac{\partial R_C^W}{\partial \kappa} X_{PC}^C \sum_i \Delta\alpha_{\kappa,i} B_i \end{aligned} \quad (4b)$$

In Eq. 4b, on the left hand side is the observed misclosure between the 3D point obtained from the aerial images and the 3D point obtained in the point cloud expressed in the world coordinate system. On the right hand side are the increments to the spline coefficients multiplied with the elements of the Jacobian.

3.2 IMU accelerations observation

The accelerations in the sensor coordinate system are denoted \ddot{X}_{IMU}^S . After rotation from sensor to car (R_S^C) and car to the world (R_C^W) coordinate system, these accelerations should correspond to the second derivatives of the car's location in the world coordinate system, i.e. \ddot{T}_C^W . Hence,

$$\ddot{T}_S^W = R_C^W R_S^C \ddot{X}_{IMU}^S \quad (5a)$$

The rotation R_S^C is calibrated and remains constant during the MLS operation. Therefore, it is not required to estimate the angles of this rotation.

Since, the acceleration in X-axis is the second derivative of the car translation $T_X(t)$, we take the second derivative on the both side of the B-spline polynomial function,

$$\frac{d^2}{dt^2} [T_X(t)] = \frac{d^2}{dt^2} \left[\sum_i \alpha_{T_x,i} B_i(t) \right] \quad (5b)$$

or

$$\ddot{T}_X(t) = \sum_i \alpha_{T_x,i} \ddot{B}_i(t) \quad (5c)$$

Note that the B-spline coefficients $\alpha_{T_x,i}$ to be estimated remain the same. Only the B-splines functions themselves need to be differentiated. Moreover, the differentiation only applies to the B-splines of the translation and not to the rotation matrices in $R_C^W R_S^C$. The linearized equation becomes:

$$\begin{aligned} \ddot{T}_C^{W^0} - R_C^W R_S^C \ddot{X}_{IMU}^S = & \begin{pmatrix} \sum_i \Delta\alpha_{T_x,i} \ddot{B}_i \\ \sum_i \Delta\alpha_{T_y,i} \ddot{B}_i \\ \sum_i \Delta\alpha_{T_z,i} \ddot{B}_i \end{pmatrix} \\ & - \frac{\partial R_C^W}{\partial \omega} R_S^C \ddot{X}_{IMU}^S \sum_i \Delta\alpha_{\omega,i} B_i - \frac{\partial R_C^W}{\partial \varphi} R_S^C \ddot{X}_{IMU}^S \sum_i \Delta\alpha_{\varphi,i} B_i \\ & - \frac{\partial R_C^W}{\partial \kappa} R_S^C \ddot{X}_{IMU}^S \sum_i \Delta\alpha_{\kappa,i} B_i \end{aligned} \quad (6)$$

3.3 IMU angular velocities observation

The IMU also observed angular velocities in the sensor frame. These are denoted $\dot{\omega}_{IMU}^S, \dot{\varphi}_{IMU}^S, \dot{\kappa}_{IMU}^S$. The axes around which these angular velocities are observed are the axes of the sensor frame. What we need are the partial derivatives of the Euler angles used to rotate between the world and the sensor coordinate systems, i.e. $\dot{\omega}, \dot{\varphi}, \dot{\kappa}$. To determine the relationship to the observed angular velocities, we first need to define exactly the order and direction of rotation. Thus, we define the rotation R_C^W , which is the rotation from the car coordinate system to the world coordinate system as in Eq. 7a,

$$R_C^W = R_3(\kappa)R_2(\varphi)R_1(\omega) \quad (7a)$$

Hence, the rotation from the world coordinate system to the car coordinate system is

$$R_W^C = R_1(-\omega)R_2(-\varphi)R_3(-\kappa) \quad (7b)$$

In this direction, we start with the heading of the car ($-\kappa$), then the pitch ($-\varphi$), and finally the roll ($-\omega$). Which can be also written as the transpose or inverse of the individual rotation matrices,

$$R_W^C = R_1(\omega)^T R_2(\varphi)^T R_3(\kappa)^T \quad (7c)$$

Once the world coordinate system is aligned with the car coordinate system, we apply the rotation R_C^S from car to sensor coordinate system. The overall rotation from the world coordinate system to sensor coordinate system is;

$$R_S^T R_W^C \quad (7d)$$

As ω is the first rotation applied when rotating from the car to the world coordinate system, it holds that the roll rotation of the car will only be sensed as a rotation velocity around the car's X-axis. However, the measured angular velocity $\dot{\varphi}_{IMU}^C$ around the Y-axis does not correspond to the first derivative of φ because the Y-axis has been rotated by $-\omega$ around the X-axis before $\dot{\varphi}_{IMU}^C$ is measured in the car coordinate system. Hence, the derivative of φ should be rotated to the car coordinate system by $R_1(\omega)^T$. Similarly, the derivative of κ needs to be rotated by $-\varphi$ around the Y-axis and $-\omega$ around the X-axis to get an angular velocity vector in the car coordinate system. Hence, all rotated angular velocity vectors together determine the angular rotation velocities that are measured by the IMU in the car coordinate system.

$$\begin{pmatrix} \dot{\omega}_{IMU}^S \\ \dot{\varphi}_{IMU}^S \\ \dot{\kappa}_{IMU}^S \end{pmatrix} = R_C^S \begin{pmatrix} \dot{\omega} \\ 0 \\ 0 \end{pmatrix} + R_C^S R_1(\omega)^T \begin{pmatrix} 0 \\ \dot{\varphi} \\ 0 \end{pmatrix} + R_C^S R_1(\omega)^T R_2(\varphi)^T \begin{pmatrix} 0 \\ 0 \\ \dot{\kappa} \end{pmatrix} \quad (8a)$$

Or

$$\begin{pmatrix} \dot{\omega}_{IMU}^S \\ \dot{\varphi}_{IMU}^S \\ \dot{\kappa}_{IMU}^S \end{pmatrix} = R_C^S \begin{pmatrix} 1 & 0 & -\sin \varphi \\ 0 & \cos \omega & \sin \omega \cos \varphi \\ 0 & -\sin \omega & \cos \omega \cos \varphi \end{pmatrix} \begin{pmatrix} \dot{\omega} \\ \dot{\varphi} \\ \dot{\kappa} \end{pmatrix} \quad (8b)$$

Let us denote this by

$$\begin{pmatrix} \dot{\omega}_{IMU}^S \\ \dot{\varphi}_{IMU}^S \\ \dot{\kappa}_{IMU}^S \end{pmatrix} = R_C^S S_W^C \begin{pmatrix} \dot{\omega} \\ \dot{\varphi} \\ \dot{\kappa} \end{pmatrix}, \quad (8c)$$

$$S_W^C = \begin{pmatrix} 1 & 0 & -\sin \varphi \\ 0 & \cos \omega & \sin \omega \cos \varphi \\ 0 & -\sin \omega & \cos \omega \cos \varphi \end{pmatrix}$$

We now use the first derivatives of the splines describing the angles, e.g.:

$$\dot{\omega}(t) = \sum_i \alpha_{\omega,i} \dot{B}_i(t) \quad (8d)$$

Which leads to the linearized Eq. 8e,

$$\begin{pmatrix} \dot{\omega}_{IMU}^C \\ \dot{\varphi}_{IMU}^C \\ \dot{\kappa}_{IMU}^C \end{pmatrix} - R_C^S S_W^{C_0} \begin{pmatrix} \dot{\omega}^0 \\ \dot{\varphi}^0 \\ \dot{\kappa}^0 \end{pmatrix} = R_C^S S_W^{C_0} \begin{pmatrix} \sum_i \Delta \alpha_{\omega,i} \dot{B}_i(t) \\ \sum_i \Delta \alpha_{\varphi,i} \dot{B}_i(t) \\ \sum_i \Delta \alpha_{\kappa,i} \dot{B}_i(t) \end{pmatrix} - R_C^S \frac{\partial S_W^{C_0}}{\partial \omega} \begin{pmatrix} \dot{\omega}^0 \\ \dot{\varphi}^0 \\ \dot{\kappa}^0 \end{pmatrix} \sum_i \Delta \alpha_{\omega,i} B_i - R_C^S \frac{\partial S_W^{C_0}}{\partial \varphi} \begin{pmatrix} \dot{\omega}^0 \\ \dot{\varphi}^0 \\ \dot{\kappa}^0 \end{pmatrix} \sum_i \Delta \alpha_{\varphi,i} B_i \quad (8e)$$

With

$$\frac{\partial S_W^{C_0}}{\partial \omega} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin \omega & \cos \omega \cos \varphi \\ 0 & -\cos \omega & -\sin \omega \cos \varphi \end{pmatrix},$$

$$\frac{\partial S_W^{C_0}}{\partial \varphi} = \begin{pmatrix} 0 & 0 & -\cos \varphi \\ 0 & 0 & -\sin \omega \sin \varphi \\ 0 & 0 & -\cos \omega \sin \varphi \end{pmatrix}$$

Note that the above equation contains no part for $\Delta \alpha_{\kappa,i} B_i$ because κ is not used in S_W^C . We can also describe the observation in more general form as Eq. 9,

$$\dot{\omega}_{IMU}^S(t) = R_C^S(t) S_W^C(t) \dot{\omega}_C^W(t) \quad (9)$$

Here $\dot{\omega}_C^W$ denote angular velocities of car in world coordinate system, $\dot{\omega}_{IMU}^S$ denote angular velocities in IMU sensor coordinate system and S_W^C denote the transformation of the angular velocities from the world to car coordinate system.

3.4 Soft-constraints on κ (yaw) and φ (pitch) angles

Since the heading κ and pitch φ can also be inferred from the trajectory described by $T_C^W(t)$, we can add two constraints to ensure that the rotations are consistent with the trajectory.

The heading can be inferred from the first derivatives of the X- and Y- coordinates of the car trajectory. The last rotation from the world coordinate system to the car coordinate system was defined $-\kappa$. Hence,

$$-\kappa = \tan^{-1} \left(\frac{\dot{T}_X}{\dot{T}_Y} \right) - \kappa_{offset} \quad (10a)$$

Where κ_{offset} is a constant offset in the κ (kappa) angle measurements by the IMU. This offset is required, as the axes of the car coordinate system may not be aligned to the axes of the IMU coordinate system. Similarly, this offset appears in the φ (pitch) angle measurement. For the heading of the car, this is linearized to Eq. 10b

$$\begin{aligned} & -\kappa^0 - \tan^{-1}\left(\frac{\dot{T}_X^0}{\dot{T}_Y^0}\right) - \kappa_{offset}^0 \\ & = \sum_i \Delta\alpha_{\kappa,i} B_i + \frac{\dot{T}_Y^0}{\dot{T}_X^{0^2} + \dot{T}_Y^{0^2}} \sum_i \Delta\alpha_{T_{X,i}} \dot{B}_i \\ & \quad + \frac{\dot{T}_X^0}{\dot{T}_X^{0^2} + \dot{T}_Y^{0^2}} \sum_i \Delta\alpha_{T_{Y,i}} \dot{B}_i + \Delta\kappa_{offset} \end{aligned} \quad (10b)$$

Since the parameters are estimated in an iterative process as described in section 3.6. The approximated value of the kappa offset is $\kappa_{offset}^0 = 0$ in the first iteration.

The pitch can be inferred from Eq. 11a

$$-\varphi = \tan^{-1}\left(\frac{\dot{T}_Z}{\sqrt{\dot{T}_X^2 + \dot{T}_Y^2}}\right) - \varphi_{offset} \quad (11a)$$

Where φ_{offset} is a constant offset in the φ (pitch) angle measurements. This is linearized to Eq. 11b

$$\begin{aligned} & -\varphi^0 - \tan^{-1}\left(\frac{\dot{T}_Z^0}{\sqrt{\dot{T}_X^{0^2} + \dot{T}_Y^{0^2}}}\right) - \varphi_{offset}^0 \\ & = \sum_i \Delta\alpha_{\varphi,i} B_i + \frac{\dot{T}_Z^0}{(\dot{T}_X^{0^2} + \dot{T}_Y^{0^2} + \dot{T}_Z^{0^2})\sqrt{\dot{T}_X^{0^2} + \dot{T}_Y^{0^2}}} \\ & \quad \left[-\dot{T}_X^0 \sum_i \Delta\alpha_{T_{X,i}} \dot{B}_i - \dot{T}_Y^0 \sum_i \Delta\alpha_{T_{Y,i}} \dot{B}_i \right. \\ & \quad \left. + \frac{\dot{T}_X^{0^2} + \dot{T}_Y^{0^2}}{\dot{T}_Z^0} \sum_i \Delta\alpha_{T_{Z,i}} \dot{B}_i \right] \\ & \quad + \Delta\varphi_{offset} \end{aligned} \quad (11b)$$

3.5 Fixed pose observation

In a normal situation, the mobile mapping car starts from a position with reliable GNSS signals reception. Similarly, we also assume that a reliable start and end pose are known. It is not essential to our method, since we already have A2P tie points that are globally consistent. However, if no road marks exist at the exact start or end of the trajectory, then the IMU observations alone produce a small initial part of the trajectory, which may lead to the residuals at the start and end parts of the trajectory. Therefore, we fix the starting and end pose of the trajectory by adding the following two observations Eqs. 12b and 13b. When the GNSS signal reception is reliable, we can use the reliable pose from the Kalman filtering. Thus, both the position being estimated and position from the Kalman filtering are equal only at the start or end of the trajectory.

$$T_{C,Kalman}^W = T_C^W \quad (12a)$$

This linearizes to,

$$T_{C,Kalman}^W - T_C^W = \begin{pmatrix} \sum_i \Delta\alpha_{T_{X,i}} B_i \\ \sum_i \Delta\alpha_{T_{Y,i}} B_i \\ \sum_i \Delta\alpha_{T_{Z,i}} B_i \end{pmatrix} \quad (12b)$$

Similarly, we can consider that the pose angles are also equivalent.

$$\begin{bmatrix} \omega_{C,Kalman}^W \\ \varphi_{C,Kalman}^W \\ \kappa_{C,Kalman}^W \end{bmatrix} = \begin{bmatrix} \omega_C^W \\ \varphi_C^W \\ \kappa_C^W \end{bmatrix} \quad (13a)$$

Which linearizes to,

$$\begin{bmatrix} \omega_{C,Kalman}^W \\ \varphi_{C,Kalman}^W \\ \kappa_{C,Kalman}^W \end{bmatrix} - \begin{bmatrix} \omega_C^W \\ \varphi_C^W \\ \kappa_C^W \end{bmatrix} = \begin{pmatrix} \sum_i \Delta\alpha_{\omega,i} B_i(t) \\ \sum_i \Delta\alpha_{\varphi,i} B_i(t) \\ \sum_i \Delta\alpha_{\kappa,i} B_i(t) \end{pmatrix} \quad (13b)$$

3.6 B-splines adjustment

We represent pose parameters with 6 individual B-spline functions over time. We have performed experiments and calculated that the B-splines of 4th order are sufficient to represent the 6DOF trajectory. As B-splines are polynomial functions and are based on coefficients, updating the B-splines involves estimating the changes to the coefficients. For example estimating $\Delta\alpha_{\omega,i}$ for the roll angle. The linearized observations equations represent these changes to the coefficients, so, the remaining task is to estimate the delta coefficient. We add each observation to a normal matrix and solve the system to obtain the required increments. The increments are then added to the B-spline coefficients to perform the adjustments.

Moreover, the normal matrix is populated with observations and solved in an iterative process. The adjustments to the coefficients are performed after every iteration. The iterative process converges when new observations yield residuals very close to the previous observations, which means that the trajectory cannot be improved further. It requires 10-20 iterations to converge to the final solution in the experiments.

4. EXPERIMENTS AND RESULTS

This section presents the adjustment results based on the observations described in section 3. For the input, we use MLS data from the Rotterdam city centre. The trajectory of this MLS data along with the start and end position is plotted in Figure 1. An overview of the basic characteristics of the trajectory has been given in Table 1. For analysis of the adjustments, check points are also acquired in the test area, which are plotted in Figure 1. All A2A and A2P tie points are also plotted in Figure 1.

We have already obtained A2A tie points using the image registration technique described in Hussnain et al. 2016 and multiview triangulation. Since it is not possible to plot

and properly visualize all A2A and A2P tie points together, only tie points from an arbitrarily selected sub-area are plotted in Figure 2.

We perform residual analysis on three different estimates of the trajectories. Firstly, we evaluate the trajectory produced by Kalman filtering. The RMSE in all three axes is presented in Table 1. The RMSE, min and max residual results are in metres. However, the targeted accuracy should be in the decimetre range. It is clear that the quality of the Kalman filtered trajectory is not sufficient.

Secondly, we apply all observations except the A2P tie point observations. In other words, this trajectory is produced from the IMU observations soft-constraints and start/end fixed pose observation. The result in Table 1 shows the achieved accuracy for the whole trajectory. For a long trajectory, drift in the sensor measurements leads to an overall large error in the results. Therefore, the IMU measurements are reliable for the small distances, while accumulate drift errors for long-term positioning. The min and max error approached metre level accuracy in some places. Obviously, the residuals in this trajectory are too large. This suggests that the A2P tie points are necessary to improve the IMU-based trajectory further. Which lead us to our final test case.

Lastly, we assess the quality of the trajectory with the proposed method by applying the IMU, A2P tie points, soft-constraints, and start/end fixed pose observations. We see a significant improvement in the accuracy when compared to both previous cases. Considering the residual errors, we conclude that the trajectory has achieved the desired accuracy near decimetre level. The trajectory before and after the adjustment is plotted in Figure 2. The min and max residuals in Z coordinate still have a significant error. This error is caused by (the error already existed in) the A2P tie points. Since the aerial imagery is acquired at the height of around 4500 metres, this leads to a poor intersection geometry for multiview triangulation. Moreover, the whole area is only covered by only 15 aerial images that provide an inadequate image overlap.

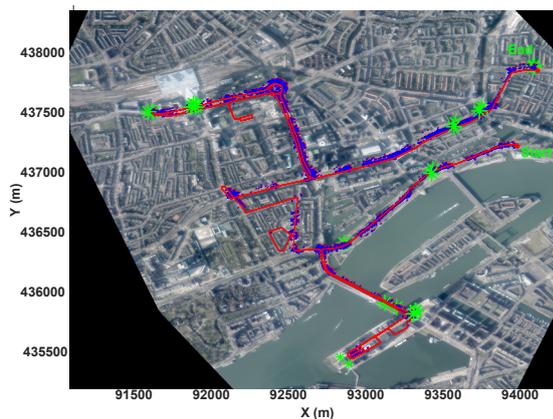


Figure 1: Kalman filtered trajectory (red curve), check points (as green asterisk) A2A tie points (blue dots) and A2P tie points (red dots). Note that there are mostly two check points at a single location.

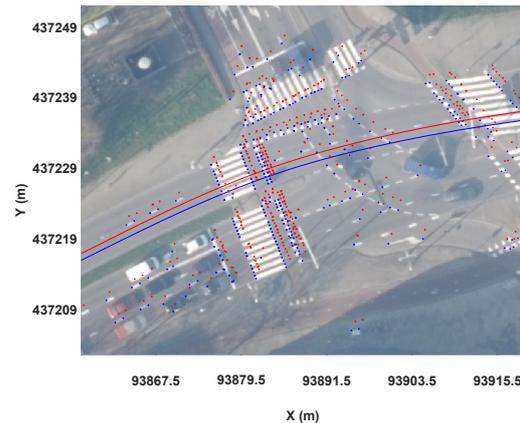


Figure 2: Example of 3D A2A tie points (blue dots) and A2P tie points (red dots) along with the Kalman filtering result (red curve) and trajectory after the adjustment with our method (blue curve).

IMU sensor	KVH® CG-5100
IMU or Kalman # observations / frequency	335565 / 100 Hz
Total acquisition time	55.9273 minutes
Trajectory length	13.6393 km
Total # check points	19
Data coordinate system	Amersfoort / RD New, EPSG:28992

Table 1: Trajectory's characteristics.

Residual meas. with check points	Kalman filtering results	IMU obs. + soft const.+ Fixed begin/end pose	IMU obs.+ A2P+ soft const.+ Fixed begin/end pose
RMSE X (m)	0.17	15.57	0.09
RMSE Y (m)	0.30	27.99	0.14
RMSE Z (m)	0.47	12.16	0.14
X min (m)	-1.28	-35.01	-0.20
Y min (m)	-0.94	-18.93	-0.10
Z min (m)	-0.58	-10.77	-0.28
X max (m)	0.96	32.49	0.10
Y max (m)	1.13	46.42	0.22
Z max (m)	0.59	30.11	0.31

Table 2: Comparison of residuals from three different trajectories calculated with check points.

5. CONCLUSIONS

This paper describes an automatic method of 6DOF trajectory adjustment for an MLS platform. We described multiple observations, which can be used collectively to adjust the 6DOF trajectory. Moreover, all observations can be obtained without any manual interventions.

The adjusted trajectory has achieved the absolute accuracy of RMSE X=9 cm, Y=14 cm and Z=14 cm. In the case of the Y-axis, it is little above the decimetre accuracy, which is the targeted accuracy. Moreover, Z max and min

residuals are also higher than the expected accuracy. Based on our previous experience, we believe that the results can be improved further by improving the quality of the A2P tie points.

Highly accurate check points are acquired at places with no GNSS signal problems, therefore the results involving Kalman filtering cannot fully represent the error in GNSS denied areas. Thus, the real error in the Kalman trajectory could well be larger than as provided in the results. The 2D registration technique is based on road markings; hence, if no road markings are present, only IMU observations can be used. A small experiment showed that trajectory for up to 100 m can well be reconstructed with IMU observations only.

In the future, we will further improve the quality of the A2P tie points, which can lead to an improvement of the adjustment results. We will also involve oblique imagery to achieve better intersection geometry for multiview triangulation. We will also perform the evaluation with data sets acquired from the different instruments to verify the reliability of the developed method.

ACKNOWLEDGEMENTS

This research project (13589) is part of the Open Technology Programme of TTW (Toegepaste en Technische Wetenschappen) which is financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

Bornaz, L., A. Lingua and F. Rinaudo 2003. Multiple scanner registration in LIDAR close-range applications. *International archives of photogrammetry remote sensing and spatial information sciences* **34(5/W12)**: 72-77.

Bosse, M. and R. Zlot 2009. Continuous 3D scan-matching with a spinning 2D laser. *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE*.

Chiang, K.-W. and Y.-W. Huang 2008. An intelligent navigator for seamless INS/GPS integrated land vehicle navigation applications. *Applied Soft Computing* **8(1)**: 722-733.

Ding, W., J. Wang, C. Rizos and D. Kinlinside 2007. Improving adaptive Kalman estimation in GPS/INS integration. *Journal of Navigation* **60(03)**: 517-529.

Gao, Y., X. Huang, F. Zhang, Z. Fu and C. Yang 2015. Automatic Geo-referencing Mobile Laser Scanning Data to UAV images. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **40(1)**: 41.

Haala, N., M. Peter, J. Kremer and G. Hunter 2008. Mobile LiDAR mapping for 3D point cloud collection in urban areas—A performance test. *The international archives of the photogrammetry, remote sensing and spatial information sciences* **37**: 1119-1127.

Hunter, G., C. Cox and J. Kremer 2006. Development of a commercial laser scanning mobile mapping system—

StreetMapper. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci* **36**.

Hussnain, Z., S. Oude Elberink and G. Vosselman 2016. Automatic feature detection, description and matching from mobile laser scanning data and aerial imagery. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **XLII-B1**: 609-616.

Javanmardi, E., Y. Gu, M. Javanmardi and S. Kamijo 2018. Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area. *IATSS Research*.

Javanmardi, M., E. Javanmardi, Y. Gu and S. Kamijo 2017. Towards High-Definition 3D Urban Mapping: Road Feature-Based Registration of Mobile Mapping Systems and Aerial Imagery. *Remote Sensing* **9(10)**: 975.

Julge, K., T. Vajakas and A. Ellmann 2017. Performance analysis of a compact and low-cost mapping-grade mobile laser scanning system. *Journal of Applied Remote Sensing* **11(4)**: 044003.

Kaartinen, H., J. Hyypä, A. Kukko, A. Jaakkola and H. Hyypä 2012. Benchmarking the performance of mobile laser scanning systems using a permanent test field. *Sensors* **12(9)**: 12814-12835.

Kukko, A. 2013. Mobile Laser Scanning—System development, performance and applications, Finnish Geodetic Institute.

Kümmerle, R., B. Steder, C. Dornhege, A. Kleiner, G. Grisetti and W. Burgard 2011. Large scale graph-based SLAM using aerial images as prior information. *Autonomous Robots* **30(1)**: 25-39.

Levinson, J., M. Montemerlo and S. Thrun 2007. Map-Based Precision Vehicle Localization in Urban Environments. *Robotics: Science and Systems, Citeseer*.

Patron-Perez, A., S. Lovegrove and G. Sibley 2015. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *International Journal of Computer Vision* **113(3)**: 208-219.

Usenko, V., L. von Stumberg, A. Pangercic and D. Cremers 2017. Real-Time Trajectory Replanning for MAVs using Uniform B-splines and 3D Circular Buffer. *arXiv preprint arXiv:1703.01416*.

Vosselman, G. 2014. Design of an indoor mapping system using three 2D laser scanners and 6 DOF SLAM. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2(3)**: 173.

Wolcott, R. W. and R. M. Eustice 2014. Visual localization within lidar maps for automated urban driving. *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE*.

Zhao, Y. 2011. GPS/IMU integrated system for land vehicle navigation based on MEMS, KTH Royal Institute of Technology.