# TIMESTAMP OFFSET DETERMINATION BETWEEN AN ACTUATED LASER SCANNER AND ITS CORRESPONDING MOTOR

Raphael Voges, Christian S. Wieghardt, Bernardo Wagner

Real Time Systems Group, Institute for Systems Engineering, Leibniz Universität Hannover, D-30167 Hannover, Germany
Email: {voges, wieghardt, wagner}@rts.uni-hannover.de

**KEY WORDS:** Actuated lidar, Rotating laser, Laser Range Finder, Sensor Synchronization, Timestamp Offsets, 3D Perception, SLAM

**ABSTRACT:**

Motor actuated 2D laser scanners are key sensors for many robotics applications that need wide ranging but low cost 3D data. There exist many approaches on how to build a 3D laser scanner using this technique, but they often lack proper synchronization for the timestamps of the actuator and the laser scanner. However, to transform the measurement points into three-dimensional space an appropriate synchronization is mandatory. Thus, we propose two different approaches to accomplish the goal of calculating timestamp offsets between laser scanner and motor prior to and after data acquisition. Both approaches use parts of a SLAM algorithm but apply different criteria to find an appropriate solution. While the approach for offset calculation prior to data acquisition exploits the fact that the SLAM algorithm should not register motion for a stationary system, the approach for offset calculation after data acquisition evaluates the perceived clarity of a point cloud created by the SLAM algorithm. Our experiments show that both approaches yield the same results although operating independently on different data, which demonstrates that the results reflect reality with a high probability. Furthermore, our experiments exhibit the significance of a proper synchronization between laser scanner and actuator.

## 1. INTRODUCTION

For many applications laser scanners are the most useful sensors since they can provide accurate and wide ranging data (Wong et al., 2011). Because of that, there are many approaches that use data acquired by a laser scanner to solve the SLAM problem (Thrun et al., 2005). While in the past 2D laser scanners were sufficient for the navigation of mobile robots in planar environments, recent SLAM approaches deal with 3D data to avoid obstacles at all heights and simultaneously acquire a dense 3D point cloud of the environment (Nuechter et al., 2007, Bosse and Zlot, 2009, Bosse et al., 2012, Zhang and Singh, 2014). However, 3D laser scanners as the Velodyne HDL-64E that provide high resolution and long ranges are expensive. Therefore, cheaper 2D laser scanners, such as a SICK or Hokuyo, that are usually only capable of acquiring scan points in a plane, are actuated by a servo drive to gather 3D data (Wulf and Wagner, 2003).

To transform the measurement points into three-dimensional space, it is required to know the appropriate encoder values of the servo drive for every set of scan points. Because of that, there exist two different approaches to determine these encoder values. The first and most simple possibility is using a motor that stops at discrete steps to let the laser scanner capture measurement points (Mandow et al., 2010). This solution, however, leads to a lower data rate since the laser scanner and the motor have to wait for each other before performing a measurement or a rotation. Another possibility is to continuously monitor and control the motion of the motor while acquiring measurement points with a high scan frequency (Wulf and Wagner, 2003, Yoshida et al., 2010). However, this can lead to a constant offset between the timestamps of the laser scanner and the motor due to the latency and transmission lags of sensors and computers. Therefore, it is essential to achieve a proper synchronization between the timestamps of the laser scanner and its rotating motor as it is already mentioned in (Hebert and Krotkov, 1992). If no synchronization is present, the offset for the corresponding encoder values for each set of scan points can lead to a large distortion in the resulting point cloud that is constructed by a SLAM approach (Wulf and Wagner, 2003).

Thus, our aim is to correct distortion in point clouds acquired by a rotating laser scanner that arises due to an erroneous offset between the timestamps of an actuated laser scanner and its corresponding motor. For this purpose, we assume that the timestamp offset is constant throughout the entire measurement.

We present two different approaches to determine the timestamp offset. The first approach can be used to calibrate the system before using it for an online algorithm that requires correctly transformed 3D data. For this, it is necessary to not move the system for a short period of time and wait for the calibration to finish. The second approach makes it possible to determine the offset between the timestamps after the acquisition of a large dataset. Thereby it becomes possible to use the dataset for offline computations although the initial synchronization between the laser scanner and the motor is not optimal. To verify the results produced by both approaches, datasets are recorded after calculating the desired offset with the first approach. Afterwards, the datasets are used to calculate the offset again using the second approach to check if both methods yield the same results.

Related work in the field of calibrating actuated laser scanners was done by (Alismail and Browning, 2014). The authors present an approach to calibrate the internal parameters (e.g. the mechanical offset between the center of rotation of the laser scanner's mirror and the center of rotation of the actuation mechanism) of an actuated spinning laser scanner. However, they do not incorporate the time offset that may occur between the timestamps of the actuator and the laser scanner.

Another idea for an actuated laser scanner is presented by (Morales et al., 2011). Within this work the authors describe the design and development of a mechanical system that is used to rotate the laser scanner. Furthermore, a motion controller that is

responsible for the synchronization between the mechanical system and the laser scanner is presented. The distinction is that our method solely focuses on the time synchronization and thus can be applied to arbitrary motor and laser scanner combinations.

Similar to our approach, the authors of the following paper (Sheehan et al., 2010) try to design a 3D laser scanner that is capable of automatic self-calibration. Their system consists of an arbitrary number of 2D laser scanners that are mounted on a rotating plate. In addition to other extrinsic parameters they also deal with the estimation of the clock skews between their devices. For this, they initially try to learn the offsets between the clocks of all devices. Afterwards, they use a second method to determine the transport delays. For this purpose they evaluate the "crispness" of resulting point clouds using different criteria than us. A further distinction is that our approach focuses on one laser scanner only, and thus it is not required to employ two separate algorithms to detect the offset.

The remainder of this paper is organized as follows. Firstly, the sensor system we used to evaluate the methods presented within this paper is introduced in Section 2. In Section 3 we describe our approach to calculate the timestamp offset between the laser scanner and the motor. Since we developed two independent methods this section is divided into two subsections: one for the approach that calculates the offset prior to data acquisition and one for the approach that deals with large datasets. Experiments demonstrating both methods are presented in Section 4 for a number of datasets of different characteristics that are introduced as well. Subsequently, we analyze and discuss the results of our experiments in Section 5. Finally, a summary and future directions conclude the paper in Section 6.

## 2. SYSTEM OVERVIEW

The idea of this paper is validated on a Hokuyo UTM-30LX laser scanner that is actuated by a Dynamixel MX-64R motor. This laser scanner can provide a 2D scan with a field of view of 270° and an angular resolution of 0.25°. However, for our experiments the field of view is limited to 180° to avoid detection of the frame the laser scanner is attached to. Every measurement of the laser scanner takes 25 ms which leads to a scan frequency of 40 lines/sec. Furthermore, the laser scanner has a maximum detection range of 30 m and a minimum detection range of 0.1 m.

The Dynamixel MX-64R robot actuator is able to operate at an angle of 360° or at a continuous turn. Furthermore, the motor supports the measurement of its own position and speed. For this it provides a angular resolution of 0.088°. To control the motion of the actuator we use the Dynamixel motor package that is available for the Robot Operating System (ROS).

Both devices are connected to a Kontron KTQM87 based embedded PC which runs the motion controller for the actuator. Moreover, it collects the measurement data from the laser scanner as well as the position data from the motor and timestamps them. Due to latency and transmission lags of sensors and the embedded PC, these timestamps may not be synchronized, and thus the offset between them needs to be determined.

It is important to note that the devices are attached to different ports of the embedded PC. While the motor is connected to the USB port via a USB to RS485 converter the laser scanner is attached to the LAN port. Thus, our assumption about a constant offset remains valid since both devices do not interfere the measurement data acquisition of each other due to the utilization of different ports. Furthermore, our embedded PC does not operate at full computational load which further ensures a constant timestamp offset.

The setup consisting of laser scanner and motor can be seen in Figure 1. For our experiments we focused on the rolling scan method for which the laser scanner is rotated around its center. This gives the advantage of only one focus point in front of the laser scanner (Wulf and Wagner, 2003).

Figure 1. An exemplary setup of an actuated laser scanner. We will use data from this system for our experiments. The laser scanner is a Hokuyo UTM-30LX scanning laser rangefinder that is rotated by a Dynamixel MX-64R robot actuator.

The motor is set to control the laser scanner such that a sweep lasts 0.5 s, where a sweep is the rotation from -90° to +90° or in the inverse direction with the horizontal orientation as 0°. This yields a rotation frequency of 1 Hz since a sweep is half a full rotation.

## 3. ESTIMATING TIMESTAMP OFFSETS BETWEEN LIDAR AND MOTOR

Within this section we will present two different approaches to compute the offsets between the timestamps of a laser scanner and its corresponding motor. The first approach can be used to determine the offset for a system that needs to be adequately synchronized for following online computations. The second approach calculates the offset for a large dataset. This enables offline computations for the dataset although the offset between the timestamps of the laser scanner and motor was not known during data acquisition. Before we start explaining our methods we give an overview of the SLAM approach that is proposed by (Zhang and Singh, 2014).

### 3.1 Overview of the utilized SLAM approach

For our methods we use the SLAM approach that is proposed by (Zhang and Singh, 2014). It consists of two algorithms that run separately. The first algorithm - lidar odometry - determines the motion of the laser scanner between two consecutive sweeps, where a sweep is the rotation from -90° to + 90° or in the inverse direction with the horizontal orientation as 0°. Furthermore, the results from the first algorithm are used to correct distortion in the point clouds that arises due to the motion of the laser scanner during a sweep.

To extract scan features, the curvature of every 2D scan point with respect to its neighbors is calculated. Points with a large curvature value are extracted as edge points while points with a small curvature value are extracted as planar points. An example for the feature extraction can be seen in Figure 2. Scan points with a large curvature value are colored in dark orange while those colored in light orange have a small curvature value.

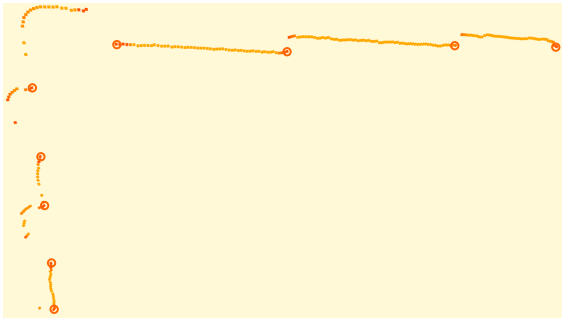


Figure 2. An example of 2D laser scan points for which the curvature is calculated. Dark orange points correspond to scan points with large curvature values while light orange points indicate a small curvature value. Circled points are extracted as edge points. Since too many planar points were extracted they are not depicted in this figure.

Afterwards, the lidar odometry algorithm determines the motion of the laser scanner by matching extracted edge points to edges and planar points to planar patches from the previous sweep. Subsequently, an optimization algorithm is used to minimize the distance between the correspondences.

The second algorithm - lidar mapping - is responsible for matching the edge points and planar points from the last sweep onto the global map. Thereby, the second algorithm is able to correct for drift over time. An example for a map can be seen in Figure 3.

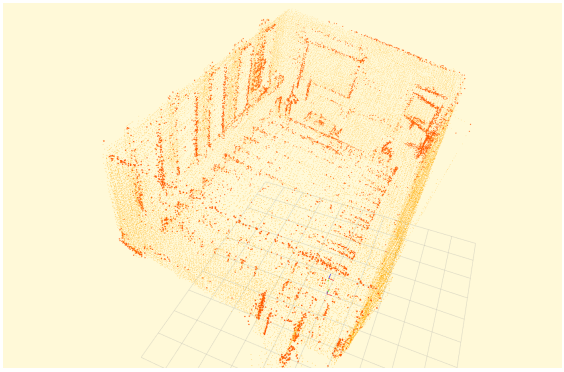


Figure 3. An example of a map that is used within the lidar mapping algorithm. Dark orange points were extracted as edge points while light orange points were extracted as planar points.

To register an edge point from the most recent sweep onto the map, the algorithm finds edge points within a certain region around the newly extracted edge point in the map and fits an edge through them. In a similar way, the corresponding planar patch for a planar point is determined. Afterwards, both feature types are combined in an optimization algorithm to minimize the distance from edge points to corresponding edges and from planar points to corresponding planar patches.

### 3.2 Prior to data acquisition for online computations

To compute the offset between the timestamps of the laser scanner and the motor, the system is set up as follows. The motor is set to rotate the laser scanner at a constant angular velocity around the center of the scanner. Furthermore, the devices are brought into a fixed position and required to remain in place until a short dataset is recorded.

The idea is to determine the offset that leads to the smallest movement calculated by a SLAM approach that incorporates the desired time offset between the timestamps of the laser scanner and the motor. Since the system remains stationary for the computation, the movement calculated by a SLAM approach should be zero. However, due to transformation errors caused by a false timestamp offset the 3D data points do not match perfectly from one scan to another and lead to erroneously computed movements by the SLAM algorithm. We use the SLAM algorithm proposed by (Zhang and Singh, 2014) that was introduced in the previous subsection.

Since our system will not move within the map it is not necessary to use the second algorithm - lidar mapping - for now. Instead, we exclusively use the first algorithm - lidar odometry - which has the further advantage that it matches consecutive sweeps acquired in opposite directions (for one sweep the laser scanner is rotated from -90° to 90° degree and for the following sweep the scanner is rotated from 90° to -90° or vice versa). As a result, an offset between the timestamps of the laser scanner and the motor leads to an offset in the consecutive point clouds which in turn induces a nonzero motion calculated by the first algorithm.

To determine the motion calculated by the first algorithm, it is necessary to first compute the translational and rotational movement separately. The translation can be computed as

$$t = \sqrt{t_x^2 + t_y^2 + t_z^2}, \tag{1}$$

where $t_x$, $t_y$ and $t_z$ are translations along the $x$-, $y$- and $z$- axes. Similarly, the magnitude of the rotation can be computed as

$$\theta = \sqrt{\theta_x^2 + \theta_y^2 + \theta_z^2}, \tag{2}$$

where $(\theta_x, \theta_y, \theta_z)^T$ is a vector representing the rotation axis while simultaneously matching the magnitude of the rotation by its length. Both the translational and rotational movement can be combined in the following equation

$$d = \sqrt{t^2 + c \cdot \theta^2}, \tag{3}$$

where $c \geq 0$ is a weighting factor. For our experiments we set $c = 1$.

To find an appropriate offset, we use a brute-force approach that takes an initial offset, a required accuracy for the final offset and a maximum number of steps. Thus, it is necessary to guess an initial range that encloses the optimal offset. This range can be arbitrarily large but must not exceed the duration of one motor rotation to avoid multiple minimums that may arise from the cyclic properties of the problem. The algorithm then iterates over possible offsets starting from the initial offset and taking steps in both directions in the size of the required accuracy. For every offset the lidar odometry algorithm is executed on the short dataset and the corresponding motion $d$ is computed. If $d$ is smaller than the current minimum, $d$ and its associated offset are set as the new

minimum. Once the maximum number of steps has been reached, the current minimum is returned.

Instead of recording a small dataset to determine the offset, it is possible to process live data that is measured by the laser scanner. For our experiments, however, we decided to use the same small dataset throughout the computations to be able to compare the resulting motion $d$ for different timestamp offsets without having to take other influences (e.g. measurement noise that varies from one run to the other) into account.

### 3.3 After data acquisition for offline computations

To determine the timestamp offset for a large dataset that is already recorded, we again use the SLAM approach that is proposed by (Zhang and Singh, 2014). As opposed to the previous subsection, the system is now allowed to move which makes it impossible to use the same strategy as before. Instead, we use both the lidar odometry and the lidar mapping of the SLAM approach and try to find the timestamp offset that induces the greatest clarity in the resulting point cloud of the environment. For this purpose, it is crucial to define adequate criteria that are viable to evaluate the clarity of a point cloud.

Initially, we thought about using the loop closure error as a criterion. This criterion, however, is not suitable since the loop closure error can be small although the SLAM algorithm performed poorly between start and end pose. That is because the algorithm may be able to close the loop by matching feature points detected at the end pose to feature points from the map that were added at the start pose. Nevertheless, the computed trajectory between start and end pose may still deviate from reality. On this account, we refrain from specifying the loop closure error for our algorithm since it can be misleading.

The first suitable criterion is the amount of matches $n$ that are registered onto the map in the lidar mapping algorithm. This is a meaningful criterion since more matches indicate that more features could be integrated into the map and thus the clarity is greater. However, if the amount of matches is low, it follows that the extracted features were not close to their corresponding edge lines or planar patches, which suggests that the clarity of the resulting point cloud is low. Thus, the aim is to maximize this criterion.

Before we can compute $n$, we need to define the sets $D_{\mathcal{E}}^k$ and $D_{\mathcal{H}}^k$ that contain all matched edge points and planar points for sweep $k$ as

$$D_{\mathcal{E}}^k := \{\, i \in \mathcal{E}_{k+1} \mid d_{\mathcal{E}}\,(i,j,l) < \delta_{\mathcal{E}};\; j,l \in \mathcal{Q}_k \,\}\,, \tag{4}$$

where $\mathcal{E}_{k+1}$ is the set of edge points that were extracted for sweep $k+1$, $\mathcal{Q}_k$ is the set of all points that were integrated into the global map until sweep $k$, $d_{\mathcal{E}}\,(i,j,l)$ is the distance between an edge point $i$ and its corresponding edge line that is represented by two points$(j,l)$ in the global map $\mathcal{Q}_k$ (cf. equation (2) in (Zhang and Singh, 2014)) and $\delta_{\mathcal{E}}$ is the maximum distance between an edge point and its corresponding edge line to consider them as a match. $D_{\mathcal{H}}^k$ is defined as

$$D_{\mathcal{H}}^k := \{\, i \in \mathcal{H}_{k+1} \mid d_{\mathcal{H}}\,(i,j,l,m) < \delta_{\mathcal{H}};\; j,l,m \in \mathcal{Q}_k \,\}\,, \tag{5}$$

where $\mathcal{H}_{k+1}$ is the set of planar points that were extracted for sweep $k+1$, $d_{\mathcal{H}}\,(i,j,l,m)$ is the distance between an planar point $i$ and its corresponding planar patch that is represented by three points$(j,l,m)$ in the global map $\mathcal{Q}_k$ (cf. equation (3) in (Zhang and Singh, 2014)) and $\delta_{\mathcal{H}}$ is the maximum distance between an planar point and its corresponding planar patch to consider them as a match. Using both sets we can derive an equation for the total number of matches $n$ as

$$n = \sum_{k \in S} \left( |D_{\mathcal{E}}^k| + |D_{\mathcal{H}}^k| \right), \tag{6}$$

where $S$ is the set of all sweeps that were acquired during the measurement and $|A|$ is the cardinality of the set $A$.

The second criterion is the average error $e$ for each match that is registered onto the map. The lower the error the closer the matches are to their corresponding edge lines or planar patches which in turn suggests a greater clarity. Thus, the aim is to minimize this criterion. The average error $e$ can be computed as

$$e = \frac{\sum_{k \in S} \left( \sum_{i \in D_{\mathcal{E}}^k} d_{\mathcal{E}}\,(i,j,l) + \sum_{i \in D_{\mathcal{H}}^k} d_{\mathcal{H}}\,(i,j,l,m) \right)}{n}, \tag{7}$$

where $d_{\mathcal{E}}\,(i,j,l)$ and $d_{\mathcal{H}}\,(i,j,l,m)$ are the distances between feature correspondences as depicted in equation (4) and (5).

Now, with these two criteria it is possible to determine an appropriate offset between the timestamps of the laser scanner and the motor. The reasoning for this is that a more fitting offset induces a point cloud that shows a greater clarity. This follows from the fact that scan features transformed with the correct encoder values can be matched more straightforward than those that are not correctly transformed. In contrast, if the timestamp offset does not fit, features from consecutive sweeps do not align and thus cannot be integrated into the global map.

To further confirm the relevance of our criteria, we carried out an additional subjective test. For this, we compared the two criteria to the authors' perceived clarity of the point clouds that resulted from running the SLAM approach using different timestamp offsets. To enable the reader to judge the clarity of different point clouds as well, we show the results of our experiments in the following section.

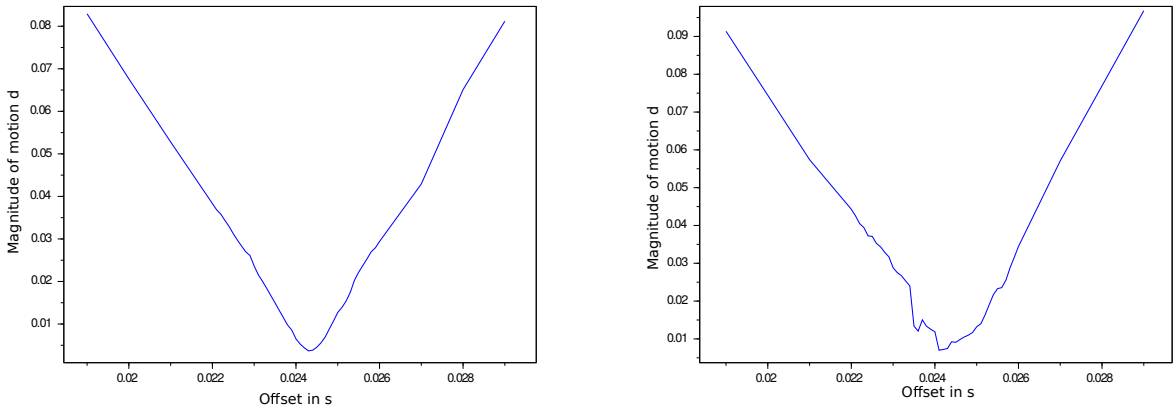
Again, we use a brute-force approach that automates the process of finding the offset for which the first criterion has a large value and the second criterion a low value. For this, it is once more required to define an initial offset, an intended accuracy for the final offset, a maximum number of steps, and thus an initial range enclosing the optimal offset. Like in the previous subsection, the algorithm iterates over different offsets but launches not only the lidar odometry but also the lidar mapping algorithm on the large dataset the offset shall be determined for. After each iteration the two criteria are stored and compared to those of previous iterations. Once the maximum number of steps has been reached, the offset is returned for which the lidar mapping algorithm produced the best criteria.

## 4. EXPERIMENTS

In this section we present the results of our experiments using the system outlined in Section 2. At first, we display the results of our first method discussed above. Afterwards, the datasets are introduced before finally presenting the results of our second approach.

### 4.1 Prior to data acquisition

Before recording large datasets we acquired six datasets of roughly four seconds each. For these datasets the system was




(a) Experiment No. 1

(b) Experiment No. 4

Figure 4. Results generated by the approach to calculate timestamp offsets prior to data acquisition. The measurement graphs belong to experiments whose results are depicted in Table 1.

kept in a fixed position to avoid motion during the measurement. In order to minimize error influences due to measurement noise and observed scenes, we chose to use six different positions in the same room. The offset between the timestamps of the laser scanner and the motor is determined using the method described in Subsection 3.2. Initially, we ran the algorithm with a desired accuracy of 1 ms in the range from 19 ms to 29 ms to determine a first approximation of the solution. We worked with this initial range since our algorithm did not succeeded with offsets outside this range. These experiments revealed that the final solution lies within the range of 22 ms to 26 ms which prompted us to carry out an additional experiment with an accuracy of 0.1 ms within this range. Further tests were not required since the final experiment with an accuracy of 0.1 ms yielded results that did not improve by a large margin from one step to another.

| No. | Offset [ms] | Magnitude of motion $d$ |
|---|---|---|
| 1 | 24.3 | 0.0037 |
| 2 | 23.4 | 0.0314 |
| 3 | 23.9 | 0.0203 |
| 4 | 24.1 | 0.0070 |
| 5 | 23.8 | 0.0220 |
| 6 | 24.6 | 0.0194 |

Table 1. Results of the offset computation prior to data acquisition.

The results for all six experiments can be seen in Table 1. It can be seen that all calculated timestamp offsets vary around 24 ms while the magnitude of motion $d$ ranges from 0.0037 to 0.0314. This can be explained by measurement noise that influences the laser scanner. However, the calculated timestamp offsets still coincide which shows that the approach is prone to some measurement noise. The average offset over all six datasets amounts to 24.0 ms.

Additionally, in Figure 4(a) and Figure 4(b) the resulting graphs for experiment number one and four, respectively, are depicted. These are exemplary for all six experiments. It can be seen that the magnitude of motion $d$ increases sharply if the observed timestamp offset differs a few milliseconds from the optimum. This leads to the conclusion that the offset between the timestamp of the laser scanner and the motor has a great influence on approaches that use an actuated laser scanner.

### 4.2 Datasets

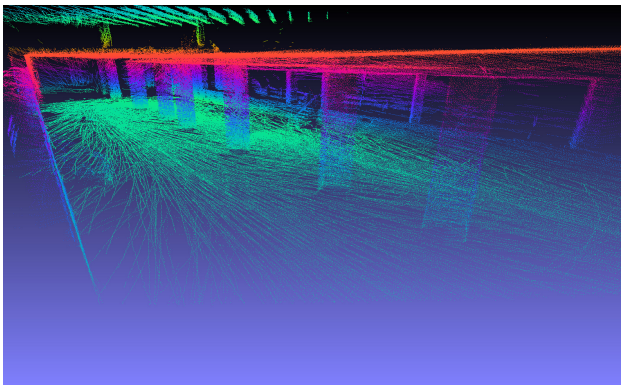
For our experiments we acquired four datasets of different characteristics. Before we introduce and discuss the results that our second approach yields, we want to present those datasets to give the reader an impression. To create a representation of the datasets, we used an extension of the previously discussed SLAM approach that is presented by (Zhang and Singh, 2015). For this, it is necessary to additionally use a RGB camera. However, since this sensor is out of the scope of this paper we do not go into details regarding the camera. Also note that no georeferencing was used and we were walking for all data recordings.

The first dataset contains data that was recorded in an abandoned metro station. In a period of roughly 159 s we cover a distance of around 117 m with a maximum velocity of 1.3 m/s and a maximum angular rate of 0.44 rad/s. Moreover, we finish the measurement in the same spot that we started it in. The surrounding point cloud that emerges when executing the extension of the previously mentioned SLAM approach (Zhang and Singh, 2015) on this dataset can be seen in Figure 5(a).

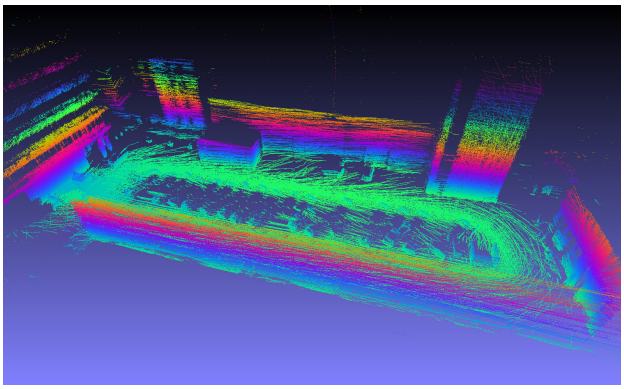
For the second dataset we moved across a parking area. In a period of roughly 208 s we cover a distance of around 155 m with a maximum velocity of 1.5 m/s and a maximum angular rate of 0.61 rad/s. Again, we finish in the same spot that we started in. In Figure 5(b) the corresponding point cloud can be seen.

The third dataset was recorded on a cemetery. In a period of roughly 287 s we cover a distance of around 236 m with a maximum velocity of 1.4 m/s and a maximum angular rate of 0.74 rad/s. Once more, we finish in the same spot that we started in. The surrounding point cloud that was created using the same SLAM approach can be seen in Figure 5(c).

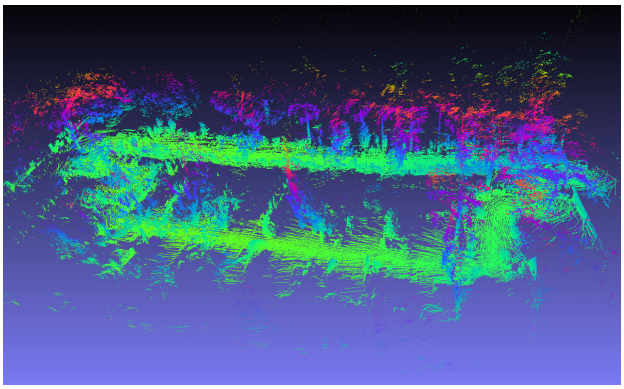
For the fourth dataset we choose an environment of several different characteristics. We started our measurement in an empty lecture room and finished it in an outside area in which trees and building facades were present. In a period of roughly 274 s we cover a distance of around 184 m with a maximum velocity of 1.9 m/s and a maximum angular rate of 0.91 rad/s. This time we do not finish in the same spot that we started in. Once again, the surrounding point cloud that can be seen in Figure 5(d) was



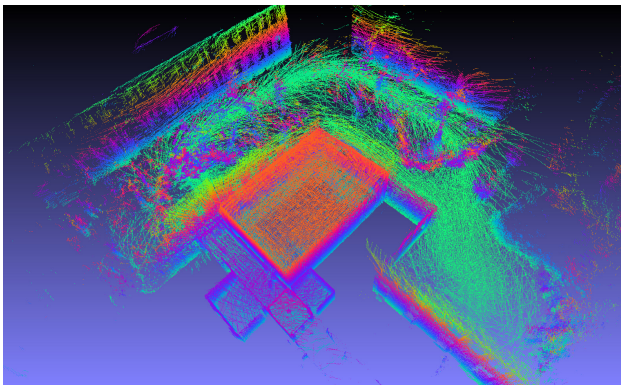
(a) Metro station



(b) Parking area



(c) Cemetery



(d) Lecture room

Figure 5. Maps generated using the discussed SLAM approach. Point clouds are color coded by elevation to generate 3D perception.

generated using the extension of the previously discussed SLAM approach.

### 4.3 After data acquisition

For all introduced datasets we executed our approach to determine the offset between the timestamps of the laser scanner and the motor that was presented in Subsection 3.3. The results are depicted in Table 2. We strove for the same accuracy as in Subsection 4.1, and thus used the ranges for the timestamp offset as depicted there.

| Dataset | Offset using the total number of matches $n$ [ms] | Offset using the average error per match $e$ [ms] |
|---|---|---|
| Metro station | 23.9 | 24.2 |
| Parking area | 24.0 | 24.0 |
| Cemetery | 23.8 | 23.8 |
| Lecture room | 24.1 | 24.1 |

Table 2. Results of the offset computation for all four datasets and for both criteria that were introduced in Subsection 3.3.

The results of our experiments can be seen in Table 2. The offsets presented are determined by evaluating the two criteria introduced in Subsection 3.3, namely the total number of matches $n$ and the average error per match $e$ in the lidar mapping algorithm. It can be seen that both criteria lead to exactly the same results for three out of four datasets with the only exception being the metro station dataset. Equally as for our other method that determines the timestamp offset prior to data acquisition, the average offset over all four datasets amounts to 24.0 ms for both criteria.

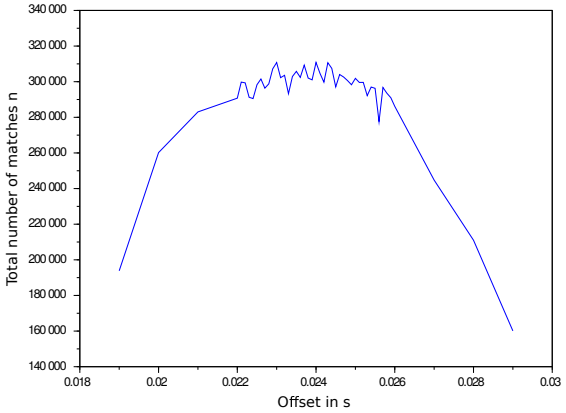
Figure 6 displays the measurement graphs for both criteria evaluated on the parking area dataset. In Figure 6(a) it becomes evident that the total number of matches $n$ approaches its maximum at 24 ms. Similarly, in Figure 6(b) it can be seen that the average error per match $e$ has its minimum at 24 ms.

Those two graphs are exemplary for all four datasets, and thus it can be concluded that the timestamp offset has a great effect on the quality of results produced by the presented SLAM approach. This can be observed by both the major drop of the total number of matches $n$ and the large growth of the average error per match $e$ if the offset deviates by more than 3 ms from the optimum.

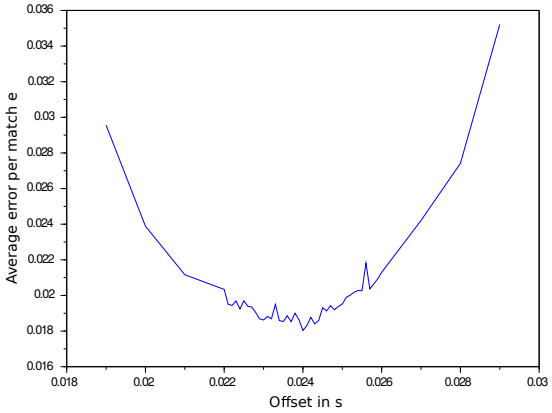
## 5. DISCUSSION

As can be seen in the previous section the calculated timestamp offset ranges from 23.4 ms to 24.6 ms for our first method (cf. Section 4.1) and from 23.8 ms to 24.2 ms for our second approach (cf. Section 4.3). Furthermore, both approaches yield the same average of 24 ms over all experiments. Thus, it can be concluded that the calculated timestamp offsets coincide for both methods within an accuracy of 1 ms. This means that both approaches are convenient to determine the timestamp offset between laser scanner and motor. Furthermore, this shows that the results reflect reality with a high probability since both methods operate independently on different data and with different criteria while still obtaining the same results.

Moreover, it is evident, considering especially the results for our second method using those large datasets, that an accuracy of 1 ms is sufficient for our purpose. As can be seen from Figure 6, the criteria of our experiments stay around the same level




(a) Total number of matches $n$.




(b) Average error per match $e$.

Figure 6. Results generated by the approach to calculate timestamp offsets for large datasets. Both measurement graphs belong to the parking area dataset, but they display different criteria to evaluate the clarity of the resulting point cloud.

for offsets between 23 ms and 25 ms which makes it sufficient to choose an offset within this range for appropriate results of the SLAM algorithm. Thus, it can be stated that a timestamp offset within an accuracy of 1 ms is adequate.

In summary, the results indicate that both presented approaches are able to achieve the goal of determining the timestamp offset between laser scanner and motor. The decision as to which method should be used depends on the available data. If the dataset, the offset should be calculated for, is already available, the second method can be used to avoid setting up the system again. However, if the timestamp offset is required for online calculations, it is inevitable to run the first method before starting those calculations.

To show the influence of the timestamp offset on the final result, we depict two point clouds that are obtained using different timestamp offsets. While all other parameters remain unchanged, the timestamp offset is set to 24 ms for Figure 7 and to 19 ms for Figure 8. Both figures show point clouds that originate from the metro station dataset in top view (compare to Figure 5(a)).

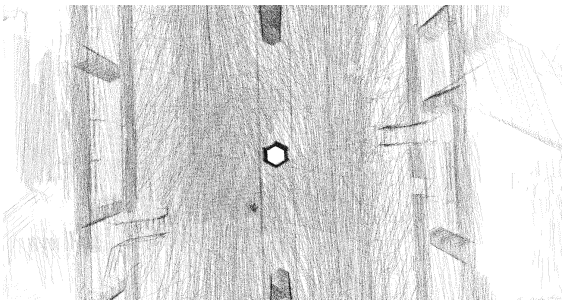


Figure 7. Map generated by the SLAM approach for the metro station dataset using an appropriate timestamp offset of 24 ms. Sharp edges indicate a great clarity for this point cloud.

The greatest difference is recognizable for the pillar in the center of both figures. While for Figure 7 the pillar is easily observable in the shape of a hexagon, it is not obvious for Figure 8. Likewise, the stairs that can be seen on the left and right side for both point clouds are more distinct for Figure 7. Thus, it can be stated that the point cloud in Figure 7 indicates a greater clarity. Furthermore, it becomes evident again that the timestamp offset between laser scanner and motor has a great effect on the resulting point cloud as an adjustment of merely 5 ms leads to a lower perceived clarity for our experiments.

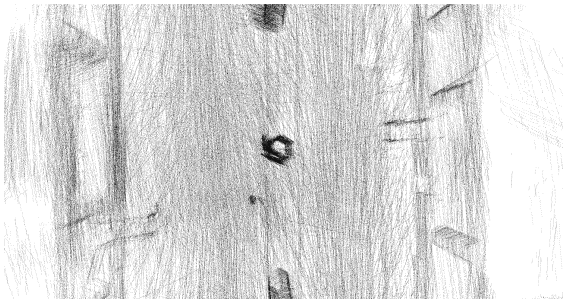


Figure 8. Map generated by the SLAM approach for the metro station dataset using an inappropriate timestamp offset of 19 ms. Blurry edges indicate a small clarity for this point cloud.

## 6. CONCLUSION AND FUTURE WORK

Incorrect synchronization between encoder values and laser scanner data can lead to distortion in the resulting point clouds when using an actuated laser scanner. To solve this problem we presented two independent approaches to calculate the timestamp offset between these two devices in this paper. Both use different parts of a SLAM approach proposed by (Zhang and Singh, 2014) and distinct criteria to find an appropriate offset. Our experiments have shown that both approaches yield similar results within an accuracy of 1 ms. However, the experimental results also showed that an accuracy of 1 ms is sufficient. Thus, it can be stated that both methods are convenient to determine the desired offset. Furthermore, we were able to demonstrate the negative effect an incorrect synchronization between motor and laser scanner can have on the resulting point clouds.

Since the second approach can be used to calculate the timestamp offset between arbitrary sensors that are fusioned for a SLAM approach, future work involves experiments to find out the significance of those offsets for the result. Additionally, an approach similar to our first method should be developed for different sensors in order to provide a reference to results from the second approach.

## REFERENCES


Alismail, H. and Browning, B., 2014. Automatic calibration of spinning actuated lidar internal parameters. *Journal of Field Robotics* 32(5), pp. 723–747.

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

Bosse, M., Zlot, R. and Flick, P., 2012. Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Transactions on Robotics* 28(5), pp. 1104–1119.

Hebert, M. and Krotkov, E., 1992. 3D measurements from imaging laser radars: how good are they? *Image and Vision Computing* 10(3), pp. 170–178.

Mandow, A., Martnez, J. L., Reina, A. J. and Morales, J., 2010. Fast range-independent spherical subsampling of 3D laser scanner points and data reduction performance evaluation for scene registration. *Pattern Recognition Letters* 31(11), pp. 1239–1250.

Morales, J., Martinez, J. L., Mandow, A., Pequeno-Boter, A. and Garcia-Cerezo, A., 2011. Design and development of a fast and precise low-cost 3D laser rangefinder. *2011 IEEE International Conference on Mechatronics*.

Nuechter, A., Lingemann, K., Hertzberg, J. and Surmann, H., 2007. 6d SLAM3d mapping outdoor environments. *Journal of Field Robotics* 24(8-9), pp. 699–722.

Sheehan, M., Harrison, A. and Newman, P., 2010. Automatic self-calibration of a full field-of-view 3d n-laser scanner. *International Symposium on Experimental Robotics*.

Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic Robotics*. MIT Press.

Wong, U., Morris, A., Lea, C., Lee, J., Whittaker, C., Garney, B. and Whittaker, R., 2011. Comparative evaluation of range sensing technologies for underground void modeling. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Wulf, O. and Wagner, B., 2003. Fast 3d scanning methods for laser measurement systems. In: *International Conference on Control Systems and Computer Science (CSCS14)*.

Yoshida, T., Irie, K., Koyanagi, E. and Tomono, M., 2010. A sensor platform for outdoor navigation using gyro-assisted odometry and roundly-swinging 3D laser scanner. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Zhang, J. and Singh, S., 2014. Loam: Lidar odometry and mapping in real-time. In: *Robotics: Science and Systems Conference (RSS)*.

Zhang, J. and Singh, S., 2015. Visual-lidar odometry and mapping: low-drift, robust, and fast. *2015 IEEE International Conference on Robotics and Automation (ICRA)*.