AUTOMATIC REGISTRATION OF LASER POINT CLOUDS OF URBAN AREAS

M. Hebel [a], U. Stilla [b]

[a] FGAN-FOM, Research Institute for Optronics and Pattern Recognition, 76275 Ettlingen, Germany - hebel@fom.fgan.de
[b] Photogrammetry and Remote Sensing, Technische Universitaet Muenchen, 80290 Muenchen, Germany - stilla@bv.tum.de

KEY WORDS: Laser scanning, LIDAR, ICP, RANSAC, multiple scans, urban data, registration

ABSTRACT:

Many tasks in airborne laserscanning require the registration of different scans of the same object. Especially data acquired in urban environments with buildings viewed obliquely from different directions need to be aligned. In this paper we propose a method to filter these point clouds based on different techniques to speed up the computations and achieve better results with the ICP algorithm. A statistical analysis is used and planes are fitted to the data wherever possible. In addition to this, we derive extra features that are used for better evaluation of point-to-point correspondences. These measures are directly used within our extension of the ICP method instead of pure Euclidean distances. Both the intensity of reflected laser pulses and normal vectors of fitted planes are considered. The extended algorithm shows faster convergence and higher stability. We demonstrate and evaluate our approach by registering four data sets that contain different oblique views of the same urban region.

1. INTRODUCTION

1.1 General purpose and overview

Due to their ability to deliver direct 3D measurements, laser scanners (LIDAR) are highly operational remote sensing devices that can be used for many photogrammetric applications. Data can be collected even at night, since LIDAR is an active illumination technique. Future need for monitoring and observation devices has led to the development of advanced technology in this field. Accurate ground-based as well as agile airborne sensors have been studied by researchers and companies in recent years. Urban regions are scanned, e.g. to provide telecommunication companies with up-to-date 3D city models. The ever-increasing level of detail and additionally measured features are raising interest in the scientific community. Currently available laser scanners are capable of acquiring the full waveform of reflected pulses, thus enabling new methods of data analysis (Jutzi & Stilla, 2006). In addition to multiple return analysis, features like intensity of reflected pulses and pulse-width can be considered. The collected data are registered by using navigational sensors, which typically consist of an inertial measurement unit (IMU) and a GPS receiver. Airborne laser scanners are able to look obliquely at urban environments to obtain information concerning the facades of buildings. These data sets resulting from different viewing directions need to be co-registered, since navigational sensors usually show small errors or suboptimalities (Maas, 2000). Although the point density is different, that task is related to stitching of terrestrial laser scanning data, where there exist proven methods in literature (Makadia et al., 2006). While these methods provide a rough alignment of the data sets, the points have to be fine-aligned to achieve the desired results.

The iterative-closest-point (ICP) algorithm, originally proposed by Besl and McKay (1992), is the standard approach to correct these discrepancies. In urban areas different scanning directions typically lead to shadow effects, occurring at one or more sides of buildings. These occlusions are a severe problem if the ICP algorithm is applied directly to these point sets, as the classical ICP algorithm is susceptible to non overlapping regions (Rabbani et al., 2007). It doesn't consider the underlying geometry and may lead to incorrect results by ending up in local minima (Rusinkiewicz & Levoy, 2001). To avoid this, our method aims at filtering the point clouds before looking for correspondences. The filter operation is intended to keep only points that are most promising to result in correct pose estimation. Consequently, points that form the facades are discarded during the registration process. Since we have an airborne sensor in oblique configuration, most occlusions occur at the rear of buildings and at ground level. In the course of the data analysis we automatically detect the ground level and remove all points belonging to it. Referring to (Maas, 2000), data points on objects with an irregular shape like trees may also falsify the matching results. Therefore these points are removed in our approach by a robust estimation technique (RANSAC). If data originating from typical urban environments are processed this way, the remaining points mostly belong to the rooftops of buildings. This method was used to cut out vegetated areas rather than analysis of the full waveform and multiple laser returns for the following reason: during the filter operation we derive an additional feature for each remaining point, namely its local normal direction. We use a combination of this and the intensity value of the reflected laser pulses to improve the distance measure of the ICP algorithm, which classically takes only Euclidean distances into account. The presented approach is applied to test data sets and results are shown in the paper.

1.2 Related work

Numerous articles on point cloud registration have been published in recent years. In some parts our work follows or is based on the ideas presented in other articles that are especially mentioned in this section. Since Besl and McKay proposed their ICP algorithm in 1992, this approach has become the standard solution to the registration problem. Nevertheless, some work has been done on alternative concepts for geometric alignment

of point clouds, e.g. by relying on the geometry of the squared distance function of a surface (Pottman et al., 2004). Other least squares matching techniques can be used to establish correspondences between data from neighboring strips (Maas, 2000). Laserscanner data are usually irregularly distributed, which is handled by many authors by introducing a triangulated irregular network (TIN) structure (Kapoutsis et al., 1998). However, we do not follow this approach: during the filter operation, we distribute the 3D data to a two-dimensional array of an appropriate size, wherein each cell is filled with the 3D coordinates of the highest occurring data point at that location. This is done to discard all points belonging to the facades of buildings. After the filtering of inapplicable points, we are able to use the full 3D information and this mapping structure for efficient search operations within the ICP implementation.

Many different approaches have been made to improve the classical iterative-closest-point (ICP) algorithm. A summary and numerical comparison of several ICP variants has been given by Rusinkiewicz and Levoy (2001). They introduced a taxonomy for categorizing ICP variants. Therein the particular category depends on which state of the original algorithm is affected: (1) selection of subsets, (2) matching of points, (3) weighting of correspondences, (4) rejecting of pairs, (5) assignment of an error metric, (6) minimizing of the error metric. In their scheme our approach would be classified as type one, two and four, since we pre-select some of the points in each data set, we affect the matching of these points and perform a threshold-based rejection of outliers. Except for this, we leave the original ICP algorithm almost untouched.

All known modifications of ICP try to improve robustness, speed and/or precision. One critical point of ICP is its lacking robustness, because it needs outlier-free data to establish valid correspondences (Chetverikov et al., 2005). We overcome this problem by using only those points that should correspond well to points in the other data sets. Another possibility is filtering of correspondences, e.g. by comparing with a dynamic distance threshold to detect wrong assignments (Zhang, 1994). Besl and McKay (1992) originally suggested establishing point-to-point correspondences by evaluation of Euclidean distances. Various improved versions and variants have been proposed that use alternative distance measures, e.g. by analyzing local surface approximations of the data sets (Chen & Medioni, 1992). We consider enhancements of the Euclidean distance by taking additional features into account while establishing point-to-point relationships.

Segmentation of the point clouds into planar surfaces is part of our approach. Other researchers have presented many different techniques concerning this topic. A summary is given by Hoover et al. (1996). Some authors are interested in detecting planes, spheres, cylinders, cones, and even more primitives. Rabbani et al. (2007) describe two methods for registration of point clouds, in which they fit models to the data by analyzing least squares quality measures. Vosselman et al. (2004) use a 3D Hough transform to recognize structures in point clouds. Among all available methods, the RANSAC algorithm (Fischler & Bolles, 1981) has several advantages to utilize in the 3D shape extraction problem (Schnabel et al., 2006). We use a RANSAC-based robust plane detection method and additionally take its score and outlier information to distinguish between buildings and irregularly shaped objects like trees. Normal vectors are assigned to each remaining data point to use this as an additional feature supporting the registration.

2. EXPERIMENTAL SETUP

We used several commercial-off-the-shelf components for the data acquisition: an FPA infrared camera (data not considered in this paper), a laser scanning device and an inertial measurement unit. The sensors that are briefly described here have been attached to a Bell UH1-D helicopter and flights were carried out over Munich, Germany.

2.1 Navigational sensors

The APPLANIX POS AV comprises a GPS receiver and a gyro-based inertial measurement unit (IMU), which is the core element of the navigational system. The GPS data are used for drift compensation and geo-referencing, whereas the IMU determines accelerations with high precision. These data are transferred to the position and orientation computing system (PCS), where they are fused by a Kalman filter, resulting in position and orientation estimates for the sensor platform.

2.2 Laser Scanner

The RIEGL LMS-Q560 is a laser scanner that gives access to the full waveform by digitizing the echo signal. The sensor makes use of the time-of-flight distance measurement principle with nanosecond infrared pulses. Opto-mechanical beam scanning provides parallel scan lines, where each measured distance is approximately geo-referenced according to the estimated position and orientation of the sensor. Waveform analysis yields intensity and pulse-width as additional features of each 3D point in the resulting point cloud. Figure 1 shows a rendered visualization of a geo-referenced point cloud, in which each point is depicted with its associated intensity.

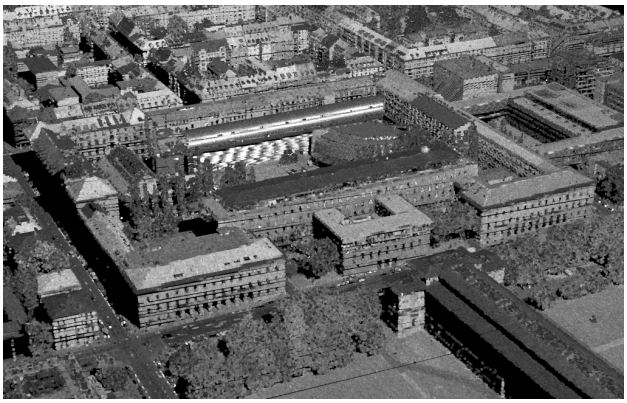


Figure 1. Laser data of an urban area scanned in 45° oblique view. Flight direction east to west (data set 4)

3. OVERVIEW OF USED METHODS

3.1 Random sample consensus (RANSAC)

The random-sample-consensus paradigm (RANSAC) as described by Fischler and Bolles (1981) is a standard technique to estimate parameters of a mathematical model underlying a set of observed data. It is particularly used in case that the observed data contain data points which can be explained by a set of model parameters (inliers) and such data points that do not fit the model (outliers). To apply the RANSAC scheme, a procedural method has to be available that determines the parameters to fit the model to a minimal subset of the data. In this paper we use RANSAC to fit planes to subsets of the point

clouds. If we have a set of n points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ and we assume that this set mostly contains points that approximately lie on one plane (inliers) and some others that do not (outliers), simple least squares model fitting would lead to poor results because the outliers would affect the estimated parameters. RANSAC estimates a plane only by taking the inliers into account, provided that the probability of choosing only inliers among the data points is sufficiently high. To compute a plane, we select a random sample of three non collinear points (the minimal subset) $\boldsymbol{p}_i$, $\boldsymbol{p}_j$, and $\boldsymbol{p}_k$. The resultant plane's normal vector $\boldsymbol{n}_0$ is computed as $\boldsymbol{m} = (\boldsymbol{p}_i - \boldsymbol{p}_j)\times(\boldsymbol{p}_i - \boldsymbol{p}_k)$, $\boldsymbol{n}_0 = \boldsymbol{m}/|\boldsymbol{m}|$ and after that, with $(\boldsymbol{x}-\boldsymbol{p}_i)\cdot\boldsymbol{n}_0 = 0$ the plane's Hessian normal form is given. Using this representation, we can check all the other points $\boldsymbol{p}$ in $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ if they are inliers or outliers simply by computing their distance $d = |(\boldsymbol{p}-\boldsymbol{p}_i)\cdot\boldsymbol{n}_0|$ to the previously obtained plane. If the distance d is below a pre-defined threshold, we assess that point as inlier. The number of inliers and the average distance of all inliers to the plane are used to evaluate the quality of the fitted plane. This procedure is repeated several times in order to converge to the best possible plane.

3.2 Iterative closest point (ICP)

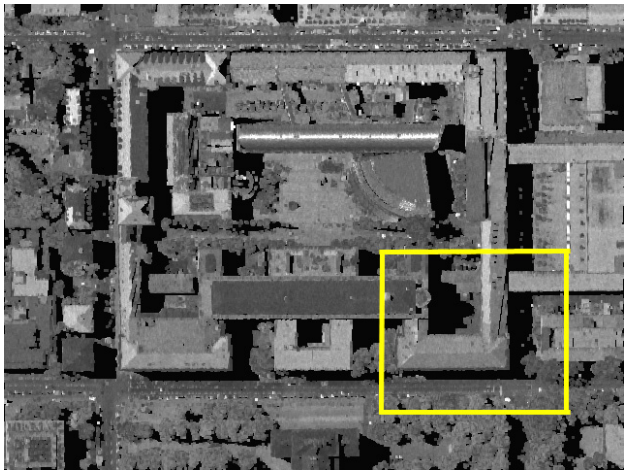
The iterative-closest-point algorithm (Besl & McKay, 1992) is intended to accurately and efficiently register 3D shapes like point sets, line segments, free-form curves, faceted surfaces, or free-form surfaces. In this paper we only consider the registration of two point sets. In literature usually one point set is seen as "data" and the other as "model", so we follow that terminology. It is assumed that both data sets are approximately in the same position, which is the case for our data.

During the ICP operation the whole data shape D is iteratively moved to be in best alignment with the model M. The first step of each iteration is to find the closest point $\boldsymbol{m}$ in M for each data point $\boldsymbol{d}$ in D. The identification of closest points between D and M should be accomplished by an efficient method, because it is the algorithm's most time consuming part. The result of this step is a sequence $(\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n)$ of closest model points to all n data points in $(\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n)$. The next step of each iteration is to find a translation and a rotation that moves the data points closest to their corresponding model points, such that the average squared (Euclidean) distance is minimized. This problem can be solved explicitly by the use of quaternions or singular value decomposition. Inconsistencies in the data sets due to missing points or occluded objects contribute directly to errors in the accuracy of the alignment. After the transformation of the data shape, the procedure is repeated and it converges monotonically to a local minimum. All the technical details and the proof of the last statement are thoroughly described in (Besl & McKay, 1992).

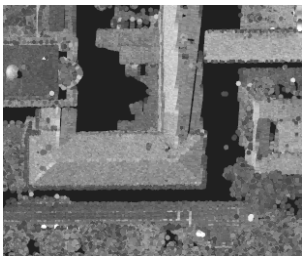
4. OUR EXTENSION OF THE ICP ALGORITHM

4.1 Preparing the data

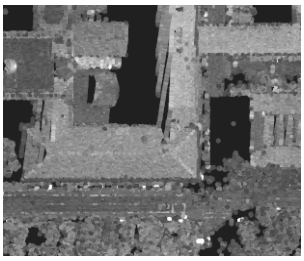
The basic idea of our approach is to filter the different point clouds to get only those points that are most promising to result in correct correspondences. By this we can improve the convergence behavior of the ICP algorithm, which is susceptible to occlusions, shadows and non overlapping regions in the data sets (Rabbani et al., 2007). In the first step we want to remove all points belonging to facades of buildings, since their presence in the data sets depends highly on the viewpoint. To achieve this, the 3D data are distributed to a horizontal two-dimensional array of appropriate size, in which each cell is filled with the 3D coordinates of the highest occurring data point at that position. The cell size corresponds to the average distance of adjacent points, which is dictated by the hardware specifications. We do not interpolate or down-sample the data during this process, since we keep the original 3D information in each cell. Anyway, by suppression of all non-maximal points we remove data belonging to the facades. Full waveform processing also yields the intensity of reflected laser pulses, which will later on be used as an additional feature for the registration, but can also provide a gray value for an image representation of the obtained 2D array. Figure 2 exemplarily shows details of the same urban region, measured in 45° oblique view from different directions.



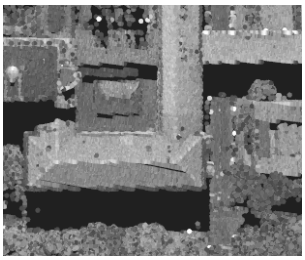
data set 1: south to north



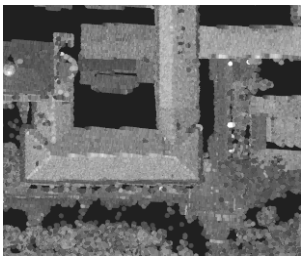
data set 1: south to north



data set 2: north to south



data set 3: west to east



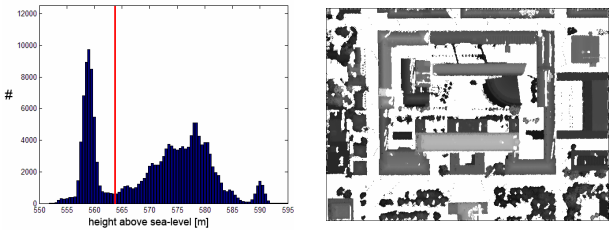
data set 4: east to west

Figure 2. Distribution of the points to a two-dimensional array and comparison of the four data sets

4.2 Detection of the ground level

The different scanning directions in Figure 2 are indicated by the position of the shadows. Due to active illumination of the scene by the airborne sensor, most shadows and occlusions occur at the rear of buildings and at ground level. As occlusions are not handled by the ICP algorithm, we need to detect the ground level and remove all points belonging to it. The detection of the ground level is fairly easy for urban

environments. We simply analyze the histogram of height values derived from the previously generated data matrix. The histogram shows a multimodal distribution, in which the laser points at ground level appear as the lowest distinct peak (Figure 3a). It is easy to find the optimal threshold that discards most data associated with the ground level by an analysis of local maxima and local minima or an expectation-maximization (EM) algorithm. Once that level is found, we take out all laser points below (Figure 3b).



a b

Figure 3. a) Determination of a threshold to remove points at ground level, b) remaining points depicted as gray-value coded height data

4.3 Robust elimination of clutter objects

The residual points in Figure 3b clearly show the contours of buildings, but also clutter objects like vegetation remain persistent. The following procedure is used to remove these irregularly shaped objects:

(1) Choose an unmarked grid-position *(i, j)* at random among the available data in the matrix.
(2) Check a sufficiently large neighborhood of this position for available data, resulting in a set *S* of 3D points.
(3) Set the counter *k* to zero.
(4) If *S* contains more than a specific number of points (e.g. at least six), continue. Otherwise mark the current position *(i, j)* as discarded and go to step 14.
(5) Increase the counter *k* by one.
(6) Perform a RANSAC-based plane fitting with the data points in the specified set *S*.
(7) If RANSAC is not able to find an appropriate plane or the number of inliers is low, mark the current position as discarded and go to step 14.
(8) Obtain the plane's Hessian normal form $(\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n_0} = 0$ and push the current position *(i, j)* on an empty stack.
(9) Pop the first element *(u, v)* off the stack.
(10) If the counter *k* reached a predefined maximum and the number of points in *S* is high enough, store the normal vector information $\boldsymbol{n_0}$ at position *(u, v)* and mark that position as processed.
(11) Check each position in a neighborhood of *(u, v)* that has not already been looked at if it contains data and in that case, check if the 3D point lies sufficiently near to the plane. If so, push its position on the stack and include the point in a new set *S'*.
(12) While the stack is not empty, go to step 9. Otherwise continue with step 13.
(13) If the counter *k* reached its maximum (e.g. three cycles), set it to zero and continue with step 14. Otherwise go to step 4 with the new set of points *S:=S'*.
(14) Go to step 1 until a certain number of runs has been performed or no more unmarked data is available.

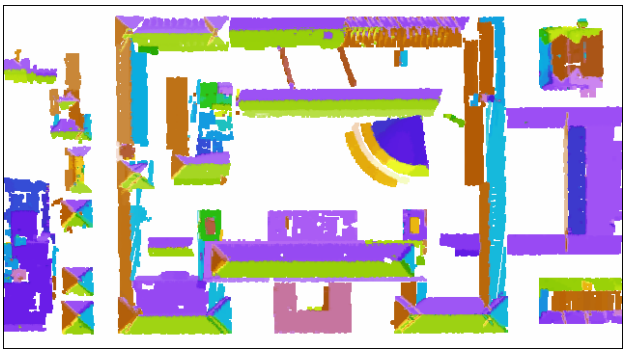


Figure 4. Remaining points after robust elimination of clutter objects, color-coded according to the normal direction

The suggested method is intended to distinguish between man-made objects like buildings and clutter objects like bushes or trees. In each run, we randomly select a position in the previously generated matrix of laser points and try to fit a plane to the neighboring data of that position. The RANSAC technique provides a robust estimation of the plane's parameters, with automatic evaluation of the quality, e.g. by the number of outliers. If the fitted plane is of poor quality, we assess the data associated with the current location as clutter. Otherwise, we try to optimize the plane fitting by looking for all data points that support the obtained plane. The underlying operation is accomplished in steps 9, 10, 11 and 12, which actually represent a seed-fill algorithm. The local plane fitting process is repeated with the supporting points to get a more accurate result. The final plane's normal vector is stored at all positions of points assigned to that plane and the corresponding data is classified as building. Figure 4 shows detected rooftops for one of the datasets, depicted with an appropriate color-coding according to their normal direction.

4.4 Executing the ICP algorithm

Now that we have removed all data that might lead to an incorrect registration, we are ready to start the ICP procedure. As mentioned in Section 3.2, the most computationally inefficient part of ICP is the search operation. We need to implement an operator that finds the closest point $\boldsymbol{m}$ in point cloud *M* for each data point $\boldsymbol{d}$ in the other data set *D*. Many improvements of this step have been proposed in literature, most of them using efficient search strategies based on k-d-trees, Voronoi diagrams or a Delaunay tessellation of the data (Kapoutsis et al., 1998). In general, this is the best known way to handle this issue, especially when dealing with irregularly distributed points. However, we do not follow this way because we already have all we need: during the filtering of points we introduced a two-dimensional data matrix among which the 3D data were distributed such that each cell contains the highest occurring data point at that position. This non-maximum suppression was done to discard the facades of buildings. Now we can use this regular grid to perform the search operation within the ICP algorithm. Let **M** be the matrix that holds *M* and let **D** be the matrix containing *D*. First we specify all indices describing cells of **D** where a 3D point of *D* is included, resulting in a list L. After that we build a look-up table T comprising all relevant indices of **M** in a sufficiently large neighborhood N_{ij} of each listed entry *(i, j)* in L. This has to be done only once. The search for closest points is done in every iteration of ICP by going through the look-up table T.

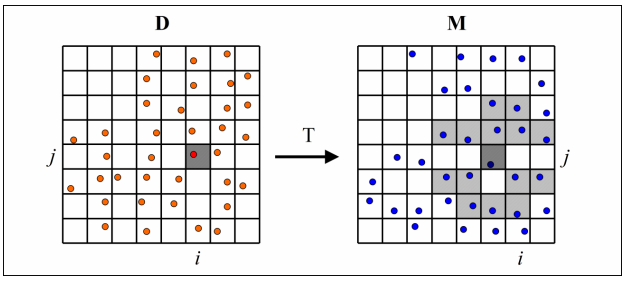


Figure 5. Assignment of a subset of M to every point in D. Each dot represents 3D coordinates together with normal direction and intensity

The functionality of the look-up table T is illustrated in Figure 5. The considered neighborhood N_{ij} at each position *(i, j)* depends on the pre-registration of the data sets. N_{ij} is essentially larger than it is depicted here, but it can be scanned reasonably fast. At each list entry in L, the assigned subset of M is processed to look for the best candidate for a point-to-point correspondence. This weighting is classically done by evaluation of Euclidean distances between $\boldsymbol{d_{ij}}$ and each point $\boldsymbol{m}$ in N_{ij} where the minimal distance is chosen. The Euclidian distance is defined by:

$$D_E = \sqrt{(\boldsymbol{d}-\boldsymbol{m})\cdot(\boldsymbol{d}-\boldsymbol{m})} \tag{4.1}$$

Due to the fact that we have an estimated normal direction and the measured intensity of the reflected laser pulse as additional information, we are able to evaluate the point-to-point relationship better than just by Euclidian distances. Given the normal direction $\boldsymbol{n_d}$ corresponding to data point $\boldsymbol{d}$ and accordingly $\boldsymbol{n_m}$ corresponding to model point $\boldsymbol{m}$, their distance can be expressed by the angle between these two vectors:

$$D_N = \arccos(\boldsymbol{n_d}\cdot\boldsymbol{n_m}),\quad D_N \in [0,\pi] \tag{4.2}$$

For better performance, the arc cosine in (4.2) can be replaced by a predefined look-up table. Finally, if I_d and I_m denote the intensity of the respective reflected laser pulses, a suitable distance measure would be:

$$D_I = \left|I_d - I_m\right|,\quad D_I \in [0,1] \tag{4.3}$$

To combine the distances D_E, D_N and D_I for each pair of points, the most straightforward approach is a linear combination D_c of the distances. The problem of finding a suitable metric D_c consists of finding the optimal set of weighting factors α, leading to the best performance of the ICP algorithm. Finding the optimal distance measure in this sense is difficult, since optimality depends on sensor and scene characteristics. One possibility to estimate the weighting factors is to analyze domain, mean and variance of each component, comparable to the Mahalanobis distance metric. After the search for corresponding points we have to find a translation and a rotation that moves the data points closest to their corresponding model points, such that the average squared Euclidean distance is minimized. To solve this problem explicitly, we use singular value decomposition (SVD) as described in (Arun et al., 1987). We briefly summarize the essential steps of their method. First the centroids of all points among the set of n correspondences are computed:

$$\boldsymbol{c}_m = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{m}_i,\qquad \boldsymbol{c}_d = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{d}_i \tag{4.4}$$

M and D are translated to the origin by $\boldsymbol{c}_m$ and $\boldsymbol{c}_d$ respectively, resulting in the new point clouds:

$$\begin{aligned}\bar{M} &= \left\{\bar{\boldsymbol{m}}_i \mid \bar{\boldsymbol{m}}_i = \boldsymbol{m}_i - \boldsymbol{c}_m,\ i=1,\ldots,n\right\},\\ \bar{D} &= \left\{\bar{\boldsymbol{d}}_i \mid \bar{\boldsymbol{d}}_i = \boldsymbol{d}_i - \boldsymbol{c}_d,\ i=1,\ldots,n\right\}\end{aligned} \tag{4.5}$$

Then a matrix H is defined as

$$H = \sum_{i=1}^{n}\bar{\boldsymbol{d}}_i\bar{\boldsymbol{m}}_i^{\,T} \tag{4.6}$$

The singular value decomposition $H = UAV^T$ of this 3x3 matrix is fast to compute and leads to the optimal rotation R and translation $\boldsymbol{t}$:

$$R = VU^T,\quad \boldsymbol{t} = \boldsymbol{c}_m - R\boldsymbol{c}_d \tag{4.7}$$

The proof of this is given in (Arun et al., 1987). After computation of optimal rotation and translation, we transform the data set D with respect to R and $\boldsymbol{t}$ and continue with the next ICP iteration until a stop criterion is met.

5. FIRST RESULTS AND EVALUATION

The proposed method was tested by using four data sets containing different oblique views of the same urban region (Figure 2). Overall, the different parameterized runs of ICP resulted in acceptable alignment of the point clouds when visually assessed. It is rather difficult to give a quantitative evaluation of the final registration, as the ICP method always converges monotonically to a local minimum of the proposed distance measure D_c. Instead of using the final sum of all values D_c to assess the alignment, one may consider other quality measures. Using the average distance between points in one data set and corresponding surface-patches in the other set is more significant than just counting point-to-nearest-point distances for assessment. This point-to-tangent-plane distance D_t can easily be quantified since we have the normal direction available for every point in M. Given a data point $\boldsymbol{d}$ and the direct neighborhood of its corresponding model point $\boldsymbol{m}$ with normal directions $\boldsymbol{n_m}$, we define D_t as $D_t = min\ |(\boldsymbol{d}-\boldsymbol{m})\cdot\boldsymbol{n_m}|$. The sum of all values D_t is used to evaluate the registration's accuracy.

We tried several combinations and weightings of D_E, D_N and D_I, and also used D_E and D_N to perform a threshold-based rejection of clearly outlying correspondences. Figure 6 illustrates the decreasing distance measure D_t against the ICP iteration number. The measured pose error decreases dramatically with the first iteration and then it converges smoothly in few steps. In some degree, the convergence behavior depends on the tested parameters. The first group of parameters does not include outlier rejection, so the average distance is comparatively high even after twenty ICP iterations. Threshold based filtering of obviously wrong assigned points outperforms influencing the error metric, so this method should be used in any case. It should be avoided to lower the threshold too much, since this may emphasize a local minimum. In our experiments we always kept at least 80 percent of the correspondences. Regarding the registration of point clouds measured from different directions, we found that the difference of intensity can be quite erratic due to active illumination of the scene, so we do not suggest using D_I in general. In addition to rejection of bad correspondences, best results were achieved by

a combination of the Euclidean distance D_E with the radian measure D_N of the normal vectors ($D_E + 10D_N$).

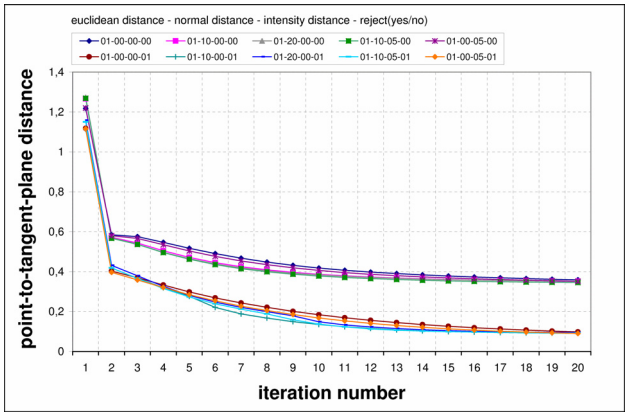


Figure 6. Course of the modified ICP algorithm with different parameters

At the moment, filtering and registering two point clouds takes about 10 minutes on a standard PC, each set containing 150.000 points. All results presented in this paper were obtained with programs that were developed under MATLAB®. Our implementation is far from being optimal, and due to a lot of unnecessary code for visualization and evaluation, there exists some potential to get much shorter computation time. The final average displacement of corresponding points is 10 cm (Figure 6) and that is comparable to the laser's range resolution. One result of the final data alignment is depicted in Figure 7. It is a rendered visualization of the registered point clouds, each depicted in slightly different color. The respective brightness is defined by the intensity of each laser echo.

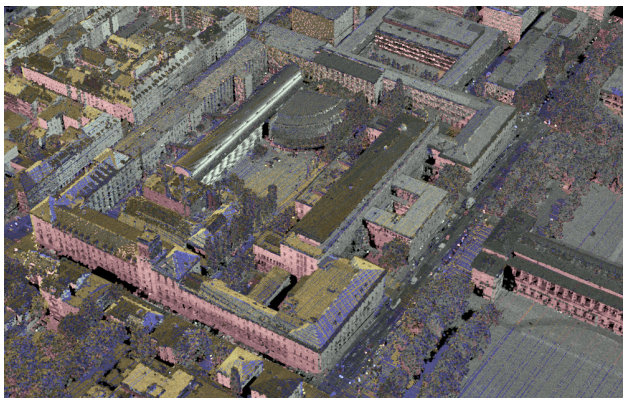


Figure 7. Illustrative result of the final data alignment

6. CONCLUSION AND FUTURE WORK

We proposed a method to filter laser point clouds of urban areas based on different techniques to achieve better results when the ICP algorithm is applied. Both the intensity of reflected laser pulses and normal vectors of fitted planes were considered to influence the ICP performance. The extended registration algorithm shows faster convergence and higher stability. We demonstrated and evaluated our approach by registering four data sets containing different oblique views of the same urban region. Future work will focus on updating the navigational data rather than aligning the point clouds. Up till now we did not consider multiple returns for vegetation mapping and discarded intensity as distance measure because of the relative nature of that signal, so the full waveform information was not used at all. In future work we will put more emphasis on full waveform analysis and we will also consider the simultaneously recorded IR image data for fusion aspects.

7. REFERENCES

Arun, K.S., Huang, T.S., Blostein, S.D., 1987. Least square fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (5), pp. 698-700.

Besl, P.J., McKay, N.D., 1992. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256.

Chen, Y., Medioni, G., 1992. Object Modelling by Registration of Multiple Range Images. Image and Vision Computing, Vol. 10, No. 3, pp. 145-155.

Chetverikov, D., Stepanov, D., Krsek, P., 2005. Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. Image and Vision Computing 23 (3), pp. 299-309.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24 (6), pp. 381-395.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B., 1996. An Experimental Comparison of Range Image Segmentation Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (7), pp. 673-689.

Jutzi, B., Stilla, U., 2006. Precise range estimation on known surfaces by analysis of full-waveform laser. Photogrammetric Computer Vision PCV 2006. International Archives of Photogrammetry and Remote Sensing Vol. 36 (Part 3).

Kapoutsis, C.A., Vavoulidis, C.P., Pitas, I., 1998. Morphological techniques in the iterative closest point algorithm. Proceedings of the International Conference on Image Processing ICIP 1998, Vol. 1 (4-7), pp. 808-812.

Maas, H.-G., 2000. Least-Squares Matching with Airborne Laserscanning Data in a TIN Structure. International Archives of Photogrammetry and Remote Sensing 33 (3a), pp. 548-555.

Makadia, A., Patterson A., Daniilidis K., 2006. Fully Automatic Registration of 3D Point Clouds. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2006, Vol. 1, pp. 1297-1304.

Pottmann, H., Leopoldseder, S., Hofer, M., 2004. Registration without ICP. Computer Vision and Image Understanding 95 (1), pp. 54-71.

Rabbani, T., Dijkmann, S., van den Heuvel, F., Vosselman, G., 2007. An integrated approach for modelling and global registration of point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 61 (6), pp. 355-370.

Rusinkiewicz, S., Levoy, M., 2001. Efficient Variants of the ICP Algorithm. Proceedings of 3D Digital Imaging and Modeling 2001, IEEE Computer Society Press, 2001, pp. 145-152.

Schnabel, R., Wahl, R., Klein, R., 2006. Shape Detection in Point Clouds. Technical report No. CG-2006-2, Universitaet Bonn, ISSN 1610-8892.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., 2004. Recognising structure in laser scanner point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 46 (8), pp. 33–38.

Zhang, Z., 1994. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. International Journal of Computer Vision 13 (2), pp. 119-152.