

ROBUST AUTOMATIC MARKER-FREE REGISTRATION OF TERRESTRIAL SCAN DATA

Wolfgang von Hansen

FGAN-FOM, Gutleuthausstr. 1, 76275 Ettlingen, Germany
wvhansen@fom.fgan.de

KEY WORDS: Urban, Terrestrial, Laser scanning, Point Cloud, Segmentation, Automation, Registration, Algorithms

ABSTRACT:

Terrestrial laser scanning systems have become widely available during the past years. Raw data acquired by such systems typically consists of separate overlapping datasets – each in its own local coordinate system. Applications that need data from more than a single scan position therefore must be preceded by a registration of all scans into a common geometric reference frame.

In this paper, a novel method for the automatic and marker-free coarse registration of terrestrial laser scan data is presented. It is based on matching planes in object space and is thus especially suitable for scenarios that are dominated by planar structures such as built-up areas. First, suitable planes are extracted from the raw point cloud in a robust way. Then, the automatic coarse registration is carried out based on correspondences of single plane pairs. Results are shown for test data consisting of 26 datasets of a small village.

1 INTRODUCTION

1.1 Motivation

In addition to airborne laser scanning, terrestrial LIDAR systems have become widely available. While products from airborne scanners cover larger areas and are often delivered as *one* geo-referenced dataset, terrestrial systems are typically operated by end-users and capture separate overlapping datasets – each in its own local coordinate system. Any application that needs data from more than a single scan position therefore must be preceded by a registration of all scans into a common geometric reference frame.

The state of the art in registration of terrestrial scan data is to place artificial markers – either 2D or 3D targets – into the scene before data acquisition. Registration software for (semi-) automatic matching of the targets is commercially available. In contrast to this, an automatic coarse registration of terrestrial scan data in the absence of markers still is a topic of research (Dold, 2005).

In this paper, a novel method for the automatic and marker-free coarse registration of laser scan data is presented. It is based on matching planes in object space and is thus especially suitable for scenarios that are dominated by planar structures such as built-up areas. A robust generation of planes from the 3D point cloud is used as preprocessing. The registration algorithm comprises a complete search that generates all possible solutions for single plane matches and then chooses the best ones based on inlier counts. The implementation turned out very fast for our test data which is a set of 26 datasets of a small village. Results show that a reliable coarse registration is possible even for such complex scenarios and thereby proves the applicability of our algorithm to real world tasks.

1.2 Related work

The basic algorithm often cited for registration of point clouds is the ICP (*iterative closest point*) algorithm (Besl and McKay, 1992): Given an initial transformation, feature correspondences are found and new transformation parameters are estimated through a least squares adjustment. This procedure is iterated until convergence. Extensions exist to enhance the radius of convergence but ICP is mainly suitable for fine registration. One example of an ICP derived method is presented in (Bae and Lichti,

2004). Matching is based on geometric curvature and change of normal vector within a given neighborhood.

(Dold and Brenner, 2004) describe the principle of registration based on three plane matches. A region growing method for the estimation of planes from point clouds with known scan geometry is presented. Subsequently, the unknown rotation is recovered through *extended gaussian images* (Dold, 2005): The normal vectors are all projected onto a unit sphere and then clustered through its tessellation. Matching the spheres at multiple resolution levels yields the rotation matrix but not the translation vector.

The adaption of the *normal distribution transform* (NDT) from 2D laser scanners used in robotics applications to the registration of 3D point clouds is proposed by (Ripperda and Brenner, 2005). Basically, the 3D data is sliced horizontally and then processed as 2D data. Although the method yields good results, one has to cope with convergence issues as well as some loss of information through the reduction of dimensionality.

The complete sequence of segmentation, coarse and fine registration is also shown by (Liu and Hirzinger, 2005), introducing the *matching tree* as a new search structure. The scene is segmented based on changes of the normal vectors and stored in a special graph structure which is then exploited for registration.

The methods reviewed here are all steps towards a generic solution, but each approach is improvable. Results for coarse registration presented are mainly – with the exception of (Ripperda and Brenner, 2005) – applied to simple scenarios only, where a single object dominates the scene and the overlap between the datasets is large. The applicability to large and complex scenarios had not been proven yet. Correspondence search algorithms similar to the one presented in this paper have been applied to 2D matching problems in computer vision for a long time (Ballard and Brown, 1982, Grimson, 1990).

2 METHODOLOGY

2.1 Overview

We will use the following terminology: The small and localized planes generated directly from the point clouds are called *surface elements*. Groups of coplanar and neighboring surface elements

are *planes*. The term *matching* will be used to denote the establishment of a logical link between two planes of two different datasets while the transformation of one dataset into the geometric reference frame of the other is called *registration*. The algorithms frequently require some thresholds for decisions – these are always denoted by a Θ with the referenced entity as index.

This section describes the processing chain from the raw point clouds to the final transformation parameters. The registration of point clouds can be subdivided into two tasks. The first is a pre-processing and feature generation step that converts the raw point cloud into a representation suitable for the second task, which is the matching of features in order to estimate the yet unknown transformation parameters of the registration. The registration can again be subdivided into a coarse and a fine registration. This distinction is necessary as precise algorithms usually require good initialization values for the transformation parameters, while robust methods that can handle large displacements usually do not return a statistically optimal result. Here is a summary of the steps of the method proposed in this paper:

Generation of surface elements Each point cloud is split into 3D raster cells. For each cell, the dominant plane is estimated through a RANSAC scheme.

Grouping to planes Neighboring coplanar surface elements are grouped to planes. These typically coincide with planar object surfaces.

Coarse registration This step is the most difficult in the processing chain and its solution is the main contribution of this paper. An exhaustive search for matching planes of two datasets is carried out. For each possible match, initial transformation parameters are computed and the number of inliers is counted. Those matches with a high inlier count are returned as correct matches for the fine registration.

Fine registration As the coarse registration returns both a set of plane matches and initial transformation parameters, the fine registration is a least squares adjustment over all scan positions to compute optimal parameters. A statistical test allows detection and removal of outliers that may have remained in the data. The fine registration is outside the scope of this paper.

2.2 Generation of surface elements

The planes that the registration algorithm requires as input will be generated in a two step process. Surface elements will be generated in a robust way from the point cloud and then are grouped to planes (see Sec. 2.3). We utilize a method that is described in (von Hansen et al., 2006) and is shown here as Alg. 1.

The set of 3D points \mathcal{X} is partitioned and assigned to 3D volume cells using a Cartesian raster. All points in one of the raster cells are denoted by \mathcal{X}_i . For each cell, the dominant plane $p_i = (\mathbf{n}_i, d_i)$ – i. e. the one that has the biggest support from

Input: 3D point cloud \mathcal{X} .
Output: 3D raster \mathcal{S} with one surface element s_i per cell.

Divide \mathcal{X} into regular raster cells \mathcal{X}_i .
for all \mathcal{X}_i **do**
 Robustly estimate dominant plane $p_i = (\mathbf{n}_i, d_i)$
 from all points $\xi \in \mathcal{X}_i$. {E. g. via RANSAC.}
 Compute barycenter \mathbf{x}_i from those ξ that support p_i .
 Add $s_i := (\mathbf{n}_i, \mathbf{x}_i)$ to output \mathcal{S} .
end for

Algorithm 1: Segment point cloud into surface elements.

Input: Surface elements \mathcal{S} as output from Alg. 1.

Output: A set of planes \mathcal{P} grouped from \mathcal{S} .

{Build graph structure.}
Create empty graph \mathcal{G} .
for all $s \in \mathcal{S}$ **do**
 Insert s as vertex into \mathcal{G} .
 for all $t \in \mathcal{G}, t \neq s$ **do**
 if s, t neighbors in 3D raster **and** s, t coplanar **then**
 Insert (undirected) edge between s and t into \mathcal{G} .
 end if
 end for
end for
{Extract planes from graph.}
{Connected components of \mathcal{G} are groups of \mathcal{S} .}
for all connected components $\mathcal{C} \subseteq \mathcal{G}$ **do**
 if $|\mathcal{C}| < \Theta_c$ **then**
 {Omit small structures.}
 else
 Estimate $p := (\bar{\mathbf{n}}, \bar{\mathbf{x}})$ from all $s_i = (\mathbf{n}_i, \mathbf{x}_i) \in \mathcal{C}$.
 Add p to \mathcal{P} .
 end if
end for

Algorithm 2: Group surface elements to planes.

the 3D points $\xi \in \mathcal{X}_i$ – is robustly estimated. This has been implemented using the well known RANSAC strategy (Fischler and Bolles, 1981), yielding a set of inlier points $\hat{\mathcal{X}}_i$.

A localization of the plane in space is also needed in order to be able to recover the translation vector \mathbf{t} with only one plane match. The plane represented by the Hesse normal form

$$ax + by + cz + d = \mathbf{n}^\top \mathbf{x} + d = 0 \quad (1)$$

has an infinite extent. We are interested in a small and delimited plane representing the points $\hat{\mathcal{X}}_i$ only. Therefore, in addition to the normal vector \mathbf{n}_i , the barycenter \mathbf{x}_i – i. e. the mean – of the point cloud $\hat{\mathcal{X}}_i$ is stored as well. The distance d_i of the plane to the origin need not be stored because it is determined by

$$d_i = -\mathbf{n}_i^\top \mathbf{x}_i. \quad (2)$$

The surface elements \mathcal{S} are used for visualization instead of the raw points. Their shape can be recognized easily in all figures showing 3D data.

2.3 Grouping to planes

The input is a regular 3D raster \mathcal{S} with each cell containing *one* surface element $s_i = (\mathbf{n}_i, \mathbf{x}_i)$ that rather precisely represents a small planar region of an object surface. Obviously, many neighboring surface elements describe exactly the same plane. The grouping collects them into a single plane based on adjacency and coplanarity as described in Alg. 2.

The basic structure used for this is a graph \mathcal{G} . All surface elements $s \in \mathcal{S}$ are entered as nodes and then compared to all of their 26 neighboring cells of the 3D raster. If two such surface elements are coplanar, then an undirected edge is inserted between the respective graph nodes. Since the order of these operations does not matter, the resulting graph is determined uniquely.

The connected components of \mathcal{G} are planes composed from the surface elements. They are simply extracted from the graph by computing a mean normal vector $\bar{\mathbf{n}}$ and a mean barycenter $\bar{\mathbf{x}}$ from each connected component and storing it as one plane in the output set \mathcal{P} . A threshold Θ_c is applied to remove planes that do not have enough support by the surface elements. This is mainly done to reject planes induced by noise and to reduce the amount of data in favor of larger and better planes.

2.4 Coarse registration

The coarse registration of two datasets is a typical chicken and egg problem. In order to compute transformation parameters, matching entities must be identified first. On the other hand, matching usually requires some knowledge about the transformation parameters. We will solve the dilemma through a complete search that generates all possible matches from which the correct ones will be extracted based on inlier counts.

It is required that the scenario contains planar object surfaces as these will be used for matching. The purely mathematical solution such as proposed in (Dold and Brenner, 2004) needs two plane matches for rotation and three for translation. Neither a random nor a systematic generation of matches seems feasible when only about 20–30% of the planes are in the overlapping area. However, the situation can be improved when ancillary knowledge is taken into account. For terrestrial laser scanners, the zenith direction is usually known from restrictions in the sensor setup. Hence, each dataset implicitly contains the horizontal ground plane so that only one plane match is required to solve for rotation.

This single plane match can already be exploited to get an approximate translation vector $\mathbf{t} = \mathbf{x}_j - \mathbf{x}_i$ via the known barycenters. This will not yield a precise solution – because there might be systematic shifts when different parts of a surface have been visible in the two datasets – but this error will cancel out when multiple matches are regarded.

The complete strategy for the coarse registration is presented in Alg. 3. First, a complete search over all possible single plane matches is carried out. As co-aligned zenith directions are assumed, this knowledge can be applied to narrow the search. The 3D normal vector \mathbf{n} of each plane is expressed in a spherical coordinate system with inclination φ and azimuth α . Two planes can only match when they have the same inclination. Then, the transformation parameters rotation \mathbf{R} and translation \mathbf{t} from \mathcal{P}_2 to \mathcal{P}_1 are computed through the difference of azimuth and barycenter respectively. The planes of \mathcal{P}_2 are transformed into the geometric reference frame of \mathcal{P}_1 and denoted \mathcal{P}_2'' .

The next step is the count of inliers n_{ij} – the number of planes matching for a particular set of parameters. If the transformation is correct, then two matching planes must have similar parameters. Each plane of the first dataset \mathcal{P}_1 is compared to all planes of the second dataset \mathcal{P}_2'' and the number of matches with similar inclination φ , azimuth α and barycenter \mathbf{x} are counted. Finally, the triggering match is entered into a list along with its transformation parameters and inlier count (Tab. 1).

Each generated match thereby is supported by other matches that verify it. The list is sorted with respect to the inlier count n_{ij} and the m best ones are picked for computation of the parameters. There are several complementary possibilities to choose m :

1. The maximum number of inliers found n_{\max} – i. e. the first row of the sorted list – is an upper bound for m .
2. The median of the first n_{\max} inlier counts is a robust estimation for the inlier rate.
3. A large difference in n from one row to the next (e. g. $n_{i+1} - n_i > \Theta_n = 1$) indicates a possible end of the inlier list.

We have used the minimum of all three possibilities as m . The transformation parameters are then estimated as a (robust) mean from all inlier matches.

Input: Two sets of planes $\mathcal{P}_1, \mathcal{P}_2$ generated from two point clouds as output from Alg. 2.

Output: Transformation parameters \mathbf{R}, \mathbf{t} from \mathcal{P}_2 to \mathcal{P}_1 . List of plane matches $\hat{\mathcal{C}} = (n_{ij}, p_i, p_j, \mathbf{R}_{ij}, \mathbf{t}_{ij}), p_i \in \mathcal{P}_1, p_j \in \mathcal{P}_2$.

{Compute additional plane attributes.}

for all $p = (\mathbf{n}, \mathbf{x}) \in \mathcal{P}_1 \cup \mathcal{P}_2$ **do**

$\alpha \leftarrow \arctan(\mathbf{n}_y / \mathbf{n}_x)$ {Azimuth}

$\varphi \leftarrow \arcsin \mathbf{n}_z$ {Inclination}

Add attributes α, φ to p .

end for

{Iterate through all possible correspondences.}

Create empty list of correspondences \mathcal{C} .

for all $p_i \in \mathcal{P}_1$ **do**

for all $p_j \in \mathcal{P}_2$ **do**

if $|\varphi_j - \varphi_i| < \Theta_\varphi$ **then** {New correspondence.}

{Transform \mathcal{P}_2 according to match (p_i, p_j) .}

$\mathbf{R}_{ij} \leftarrow$ Rotation by angle $\alpha_j - \alpha_i$ around z -axis.

$\mathcal{P}_2' \leftarrow$ Apply \mathbf{R}_{ij} to \mathcal{P}_2 .

$\mathbf{t}_{ij} \leftarrow \mathbf{x}_j - \mathbf{x}_i$

$\mathcal{P}_2'' \leftarrow$ Translate \mathcal{P}_2' by \mathbf{t}_{ij} .

{Count inliers.}

$n_{ij} \leftarrow 0$

for all $p_k \in \mathcal{P}_1$ **do**

for all $p_\ell'' \in \mathcal{P}_2''$ **do**

if $|\varphi_\ell'' - \varphi_k| < \Theta_\varphi \wedge |\alpha_\ell'' - \alpha_k| < \Theta_\alpha$

$\wedge \|\mathbf{x}_\ell'' - \mathbf{x}_k\| < \Theta_x$ **then**

$n_{ij} \leftarrow n_{ij} + 1$

end if

end for

end for

Insert $(n_{ij}, p_i, p_j, \mathbf{R}_{ij}, \mathbf{t}_{ij})$ into \mathcal{C} .

end if

end for

end for

Sort \mathcal{C} with respect to n .

Pick m correspondences $\hat{\mathcal{C}}$ with most inliers from \mathcal{C} .

Compute output \mathbf{R} and \mathbf{t} from all $c \in \hat{\mathcal{C}}$.

Algorithm 3: Automatic coarse registration.

3 EXPERIMENTS AND RESULTS

3.1 Available datasets

We dispose of 26 overlapping datasets from a Z+F Imager 5003 terrestrial laser scanner. It has an operation range of 50 m and captured about 100 million valid 3D points per dataset. For each 3D point, the amount of reflected light is recorded and available for surface textures.

The imaged scenario is a farming village, containing moderately complex arrangements of small houses around many courtyards. Buildings are typically two stories high and have inclined roofs. The global layout of all scan positions is shown in Fig. 1.

3.2 Generation of surface elements

The raster size for the generation of surface elements has been chosen as 1 m in order to describe façade and roof surfaces through several surface elements but also to ignore small structures. Results for three selected positions are shown in Fig. 2. The grid like texture of the plane boundaries originally stems from unintended border effects in the visualization, but clearly shows how the result is composed. The number of surface elements generated for each of the positions ranges from 2600 to 10000.

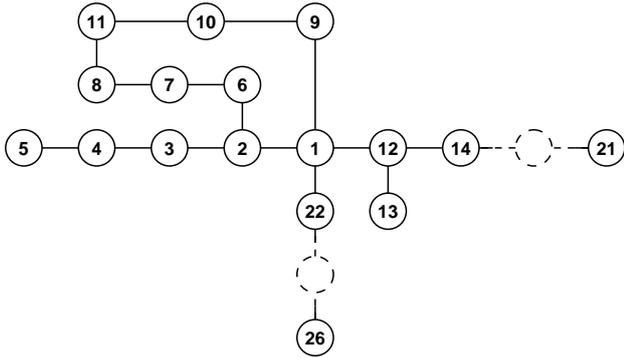


Figure 1: Graph-like layout of all scanner positions, roughly in their correct geometric place. Dashed lines denote rows of consecutive positions that have been left out in this illustration. Directly neighboring positions are connected, but distant datasets may also overlap.

#Inliers	p_i	p_j	Rotation		Translation		
			α/rad	t_x/m	t_y/m	t_z/m	
17	88	41	-0.2419	20.76	4.43	-1.71	
16	95	55	-0.2527	20.56	2.85	-1.22	
16	94	52	-0.2467	20.78	3.93	-0.54	
16	91	45	-0.2406	21.00	2.99	-1.31	
16	90	40	-0.2424	21.19	2.81	-1.68	
16	89	43	-0.2435	20.38	2.85	-1.12	
16	87	42	-0.2340	20.92	3.42	-1.27	
16	86	33	-0.2282	21.14	2.85	-2.08	
16	85	34	-0.2571	20.50	2.81	-1.30	
16	67	26	-0.2512	21.89	3.17	-1.53	
16	35	3	-0.2587	21.00	2.72	-1.97	
16	31	5	-0.2411	21.01	3.18	-1.67	
16	27	2	-0.2521	21.39	2.94	-1.25	
15	79	29	-0.2408	21.96	2.57	-1.02	
15	76	30	-0.2310	20.81	3.01	-0.75	
13	93	52	-0.2481	20.80	3.52	-3.90	
8	76	31	-0.2276	19.19	2.27	0.93	
5	94	62	-0.2304	-3.70	-0.62	-0.80	
5	49	55	1.2278	-3.44	-15.16	-2.69	
5	28	59	1.2675	-3.14	-12.69	-3.01	

Table 1: Twenty best matches from position 2 to 3. The horizontal line indicates the end of the automatically chosen inlier set.

3.3 Grouping to planes

The grouping is a deterministic procedure that can be guided through two thresholds – one for coplanarity and one to reject too small planes. One result is shown in Fig. 3, where all planes recovered from position 2 are shown in uniform colors. The number of planes for each position ranges from 30 to 100.

Since only local comparisons are used for the creation of the graph, it may happen that large resulting regions are not exactly planar. While the surface elements are an oversegmentation of object space, the planes are an undersegmentation for which the streets in an outdoor scenario would be typical examples. For the coarse registration this poses no real problem, since undersegmentation results in only a few planes that are easily ignored by robust algorithms.

3.4 Coarse registration

As the maximum deviation from the true zenith direction was ≤ 30 mrad, no prior rotation of the datasets was necessary. As a typical example for the output of Alg. 3, the list of matches from position 2 to 3 is shown in Tab. 1. Only the top twenty matches are given – the actual list is much longer (cf. column “#Tests” of Tab. 2). Column “#Inliers” contains the number of inliers for the match of the two planes listed in the columns p_i and p_j . The

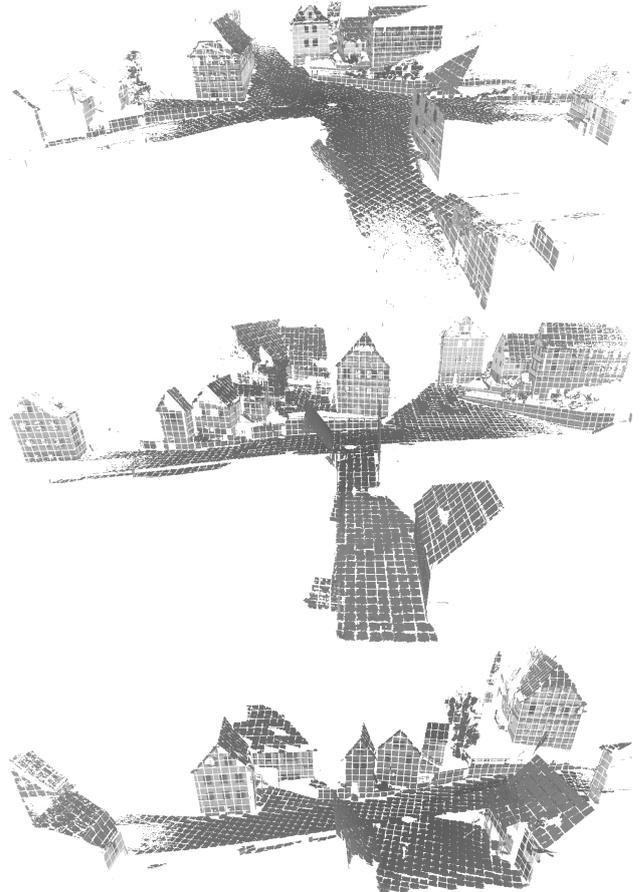


Figure 2: Datasets from positions 1 to 3 (from top to bottom). The small square structures are the surface elements.

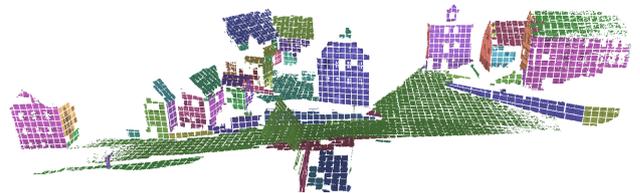


Figure 3: Result of the grouping for position 2.

last four columns show the transformation parameters valid for this particular match. A horizontal line marks the automatically defined end of the inlier set.

Results for all neighboring positions of the test data (cf. Fig. 1) are shown in Tab. 2. Columns \mathcal{P}_1 and \mathcal{P}_2 are the position numbers, column $n_1 n_2$ is the total number of tests that are possible, while the next two columns show the number and percentage of tests actually carried out because the inclination indicated a possible match. At least half of the generated correspondences could be rejected early through this criteria.

The following columns are the results of the inlier tests. The absolute number of inliers is given along with the inlier rate with respect to $\min(n_1, n_2)$. The inlier rate of only 25% is low mainly because of the limited overlap between the datasets. Not all registrations have been successful. Filled circles mark successful registrations while empty circles indicate failures.

In order to illustrate the results of the coarse registration, a color-coded fusion of the individual datasets from positions 1 to 3 is shown in Fig. 4.

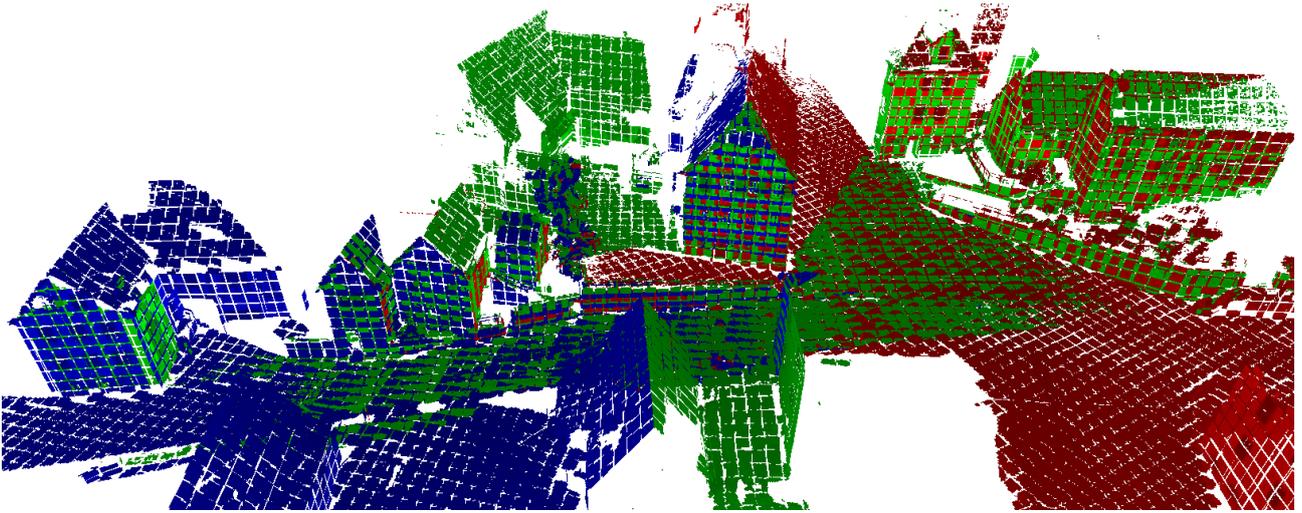


Figure 4: Resulting coarse registration of positions 1 (red), 2 (green), 3 (blue).

\mathcal{P}_1	\mathcal{P}_2	$n_1 n_2$	#Tests		#Inliers		
1	2	7776	3191	41%	21	26%	●
2	3	6240	1993	32%	15	23%	●
3	4	3705	1431	39%	17	30%	●
4	5	2337	980	42%	11	27%	●
2	6	4512	1479	33%	13	28%	●
6	7	2021	623	31%	4	9%	○
7	8	2064	550	27%	12	28%	●
8	11	2928	672	23%	6	12%	●
11	10	2501	646	26%	4	10%	○
10	9	2255	774	34%	8	20%	●
9	1	4455	2042	46%	6	11%	●
1	12	5346	2354	44%	17	26%	●
12	13	5610	2300	41%	18	27%	●
12	14	4620	1742	38%	5	8%	●
14	15	6160	2364	38%	18	26%	●
15	16	6336	2520	40%	19	26%	●
16	17	6120	2377	39%	16	22%	●
17	18	7395	2588	35%	20	24%	●
18	19	7743	2531	33%	16	18%	●
19	20	4272	1276	30%	10	21%	●
20	21	1632	364	22%	3	9%	○
1	22	6075	2910	48%	22	29%	●
22	23	4575	1864	41%	10	16%	●
23	24	3111	1162	37%	16	31%	●
24	25	2091	727	35%	12	29%	●
25	26	1681	555	33%	10	24%	●

Table 2: The number of tests actually needed compared to the total number of correspondences.

4 DISCUSSION

Robustness The robustness – i. e. outliers do not have an impact on the result – of our method is achieved at multiple levels:

- Alg. 3 is superior to RANSAC: It is also a generate-and-test scheme that uses the inlier count as quality measure, but with the distinction that the random sampling has been replaced by a complete search. RANSAC would return only one of the first m rows of Tab. 1 as its result whereas we get m valid rows.
- There exist several methods to determine m (Sec. 2.4). Hence, the estimated size of the inlier set can be checked.
- Estimation of the transformation parameters from the m rows can again be done through a robust scheme such as RANSAC or a least squares adjustment with outlier detection.

Complexity Alg. 3 has a complexity of $O(n^4)$ with n being the average number of planes in \mathcal{P}_i : Both the generate loop and the test loop nested inside compare all planes from \mathcal{P}_1 and \mathcal{P}_2 and thus each have complexity $O(n^2)$. Despite this, the implementation turned out very fast and finished within seconds even on a slow computer (Pentium III Mobile CPU @ 750 MHz) for multiple reasons:

- In practical applications, n is small. For the 26 datasets we have $n < 100$ and therefore can expect less than $100^4 = 100$ million runs of the innermost loop.
- The test of matching inclinations reduces the number of calls to the test loops. According to Tab. 2, less than half of the generated matches actually have to be checked.
- The innermost loop only contains comparisons so that it does not need much processing power. In fact, since there exist only $2n$ data elements, it is likely that the innermost loop will run entirely on the CPU cache.

Alternatively to the count of inliers one could also look for clusters in parameter space. The complexity for the matching is reduced to $O(n^2)$ as only the generation loop is needed, but a Hough like clustering would require a four dimensional accumulator for the parameters.

Execution times for the different stages are shown in Tab. 3. The first two stages take considerably longer because they process the complete cloud of about 100 million points so that I/O performance is an issue as well. In contrast to this, grouping and registration run very fast as these steps operate only on plane representations of the data.

Results Compared to the 100 million points of the raw point cloud, the surface elements are a significant data reduction – especially near the laser scanner, where the point density of the raw data is very high. As can be seen from Fig. 2, they describe the scene very well. The results of the grouping are good, because planes typically coincide with object surfaces.

As can be seen from the registration result in Tab. 1, every match from the inlier list (and in fact also the next two rows) contain similar transformation parameters. The final parameters for the coarse registration can easily be obtained from this list – e. g. by averaging over the automatically chosen inlier set. Additionally,

Stage	min	avg	max
Split point cloud into raster	249	307	343
Generate surface elements	136	215	563
Group to large planes	0.18	0.59	2.18
Coarse registration	0.17	0.57	1.29

Table 3: Minimum, average and maximum execution time in seconds for all datasets and the different stages on a Pentium M processor @ 1.7 GHz.

we thereby already dispose of a list of plane matches that could be used as starting point for the fine registration.

In all cases of a failure in Tab. 2, the datasets did not have enough matching planes in the overlapping area to get the inlier rate above the noise. During data acquisition, the positions have only been chosen to produce overlapping datasets that cover the complete scenario, so that the failures are due to bad sensor positioning. However, it is possible to work around such cases by first joining some datasets and then to try to match with this instead.

In Tab. 4, the ten best matches are shown for one of the failures. It can be seen that the first m rows do not necessarily contain only inliers. In order to detect a failure caused by a weak configuration, the parameters of the first rows must be checked for this.

Object features that belong together are overlaid very well in Fig. 4. The quality of the fusion can be seen best from the two nicely fitting blue-green and blue-red roofs in the middle of the figure. Close inspection of the dataset reveals that the registration is not perfect. While some surfaces actually coincide (these can be identified by their multi-color pattern) others – like the blue façade on the very left – are up to 1 m apart. A fine registration can easily fix this, as valid plane correspondences are available.

Errors in zenith direction We do not explicitly take into account the errors in the zenith direction. From analysis of the data we know that the absolute error is less than 30 mrad. However, the rotational component is also compensated via a z -shift. Actually this should even decrease bending of large models because the small rotations are not summed up.

5 CONCLUSIONS

We have presented a novel algorithm for the automatic marker-free coarse registration of two point clouds from terrestrial laser scanner – a task that is still considered difficult. The key idea was to recover initial parameters for rotation and translation from single plane correspondences only. As prerequisites it was required that the scene contains planar surfaces and that the laser scanner is set up with the local z -direction pointing upward. These conditions are easily fulfilled for built-up areas and available systems.

We have shown results for 26 datasets covering large parts of a moderately complex farming village. It has been shown that a reliable coarse registration of such real world data with an overlap of only 20–30% between neighboring positions is possible with our method. Some failures occurred, but these were always due to an insufficient number of common planes in both datasets, caused by bad sensor placement. It is possible to automatically detect such bad configurations by analysis of the correspondence lists.

Current work is focussed on the extension of the approach to the automatic determination of topology of multiple datasets. The input shall be a number of datasets without any additional information and the output a neighborhood graph similar to Fig. 1 along with the complete set of transformation matrices for all datasets. We also plan to test the applicability of this algorithm to the mixed registration of terrestrial and airborne LIDAR data.

#Inliers	p_i	p_j	Rotation		Translation	
			α/rad	t_x/m	t_y/m	t_z/m
4	29	13	0.1898	-11.31	18.57	0.11
3	44	16	-2.9469	9.24	-5.62	-2.37
3	28	27	-1.3910	-11.21	-24.28	-2.65
3	18	19	-1.3633	18.64	-24.13	-1.79
3	18	3	0.2133	-12.23	-11.49	-2.58
3	11	3	0.1621	-12.66	18.36	0.21
3	4	13	0.2979	-32.38	30.89	3.82
3	4	3	-1.2788	-13.93	17.33	1.30
3	2	16	0.2951	-33.09	34.87	5.29
3	2	2	-1.2707	-14.62	20.61	1.54

Table 4: Ten best matches from position 20 to 21. The horizontal line indicates the end of the automatically chosen inlier set.

REFERENCES

- Bae, K.-H. and Lichti, D. D., 2004. Automated Registration of Unorganised Point Clouds from Terrestrial Laser Scanners. In: O. Altan (ed.), Proc. of the XXth ISPRS Congress, IAPRS, Vol. XXXV-B5. URL: www.isprs.org/istanbul2004/comm5/papers/553.pdf.
- Ballard, D. H. and Brown, C. M., 1982. Computer Vision. Prentice-Hall. URL: homepages.inf.ed.ac.uk/rbf/BOOKS/BANDB.
- Besl, P. J. and McKay, N., 1992. A Method for Registration of 3-D Shapes. PAMI 14(2), pp. 239–256.
- Dold, C., 2005. Extended Gaussian Images for the Registration of Terrestrial Data. In: G. Vosselman and C. Brenner (eds), Laser scanning 2005, IAPRS, Vol. XXXVI-3/W19. URL: www.commission3.isprs.org/laserscanning2005/papers/180.pdf.
- Dold, C. and Brenner, C., 2004. Automatic Matching of Terrestrial Scan Data as a Basis for the Generation of Detailed 3D City Models. In: O. Altan (ed.), Proc. of the XXth ISPRS Congress, IAPRS, Vol. XXXV-B3. URL: www.isprs.org/istanbul2004/comm3/papers/429.pdf.
- Fischler, M. A. and Bolles, R. C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM 24(6), pp. 381–395.
- Grimson, W. E. L., 1990. Object Recognition by Computer: The Role of Geometric Constraints. The MIT Press.
- Liu, R. and Hirzinger, G., 2005. Marker-free Automatic Matching of Range Data. In: R. Reulke and U. Knauer (eds), Panoramic Photogrammetry Workshop, IAPRS, Vol. XXXVI-5/W8. URL: www.informatik.hu-berlin.de/sv/pr/PanoramicPhotogrammetryWorkshop2005/Paper/PanoWS_Berlin2005_Rui.pdf.
- Ripperda, N. and Brenner, C., 2005. Marker-free Registration of Terrestrial Laser Scans Using the Normal Distribution Transform. In: S. El-Hakim, F. Remondino and L. Gonzo (eds), 3D-ARCH 2005, IAPRS, Vol. XXXVI-5/W17. URL: www.commission5.isprs.org/3darch05/pdf/33.pdf.
- von Hansen, W., Michaelsen, E. and Thönnessen, U., 2006. Cluster analysis and priority sorting in huge point clouds for building reconstruction. In: The 18th International Conference of Pattern Recognition.