

FAST REGISTRATION OF LASER SCANS WITH 4-POINTS CONGRUENT SETS – WHAT WORKS AND WHAT DOESN'T

P. W. Theiler, J. D. Wegner, K. Schindler

Institute of Geodesy and Photogrammetry, ETH Zürich, 8093 Zürich, Switzerland
{pascal.theiler | jan.wegner | schindler}@geod.baug.ethz.ch

KEY WORDS: point cloud registration, laser scanning, 3D feature extraction

ABSTRACT:

Sampling-based algorithms in the mould of RANSAC have emerged as one of the most successful methods for the fully automated registration of point clouds acquired by terrestrial laser scanning (TLS). Sampling methods in conjunction with 3D keypoint extraction, have shown promising results, e.g. the recent K-4PCS (Theiler et al., 2013). However, they still exhibit certain improbable failures, and are computationally expensive and slow if the overlap between scans is low. Here, we examine several variations of the basic K-4PCS framework that have the potential to improve its runtime and robustness. Since the method is inherently parallelizable, straight-forward multi-threading already brings down runtimes to a practically acceptable level (seconds to minutes). At a conceptual level, replacing the RANSAC error function with the more principled MSAC function (Torr and Zisserman, 2000) and introducing a minimum-distance prior to counter the near-field bias reduce failure rates by a factor of up to 4. On the other hand, replacing the repeated evaluation of the RANSAC error function with a voting scheme over the transformation parameters proved not to be generally applicable for the scan registration problem. All these possible extensions are tested experimentally on multiple challenging outdoor and indoor scenarios.

1. INTRODUCTION

Static terrestrial laser scanners (TLS) have become standard devices to acquire 3D data for a wide range of applications like as-built mapping of large industrial facilities, documentation of heritage sites, or manufacturing. Multiple scans from different viewpoints usually have to be acquired to fully cover complex objects. To combine all these scans into a single point cloud, the relative orientation between them (rigid-body transformation with six degrees of freedom) has to be found. This process of aligning all scans in a common reference system is usually called *registration* and is a prerequisite for any further analysis like 3D reconstruction or semantic object segmentation.

In practice, scan registration is either done completely manually or automatically based on artificial markers placed in the scene during acquisition (e.g., Akca, 2003; Franaszek et al., 2009). Although this procedure solves the registration, it is time-consuming in the field, markers occlude small scene parts and often have to be removed from the result, they must remain static until completion of all scans, and they should never be occluded by moving objects like cars etc. Thus, much effort has been spent to develop fully automated, marker-less methods for scan registration and to avoid artificial markers completely.

Various solutions for *fine registration* of scans have been proposed, most notably the *Iterative Closest Point* (ICP) algorithm (Besl and McKay, 1992; Chen and Medioni, 1992) and its variants (e.g., Bergevin et al., 1996; Bae and Lichti, 2004; Minguetz et al., 2006; Censi, 2008). On the contrary, no practical, generally applicable and efficient solution to *coarse register* big TLS point clouds is available yet. ICP-like approaches minimize a sum of Euclidean distances between potentially corresponding points, i.e. they aim at locally optimizing a highly non-convex objective function. Consequently, applying ICP directly to scans with arbitrary relative orientation will in most cases fail (see Pottmann et al. (2006) for details on convergence properties). A common strategy is to first roughly register scans with a robust, but less accurate method, and let ICP take over from that initial solution.

Here, we deal with the *coarse registration* to obtain an initial solution, which must only be good enough for standard ICP to accomplish fine registration. What makes coarse registration of arbitrarily oriented TLS point clouds difficult are (i) unevenly distributed scan points due to the polar measurement principle, (ii) the sheer amount of data (millions of points), (iii) the often limited overlap and strong change of viewpoint between neighbouring scans, to save acquisition time in the field.

To achieve coarse TLS point cloud alignment we build upon work of Theiler et al. (2013) that adapts the *4-Points Congruent Sets* (4PCS, Aiger et al., 2008) and applies it to clouds of keypoints extracted from raw scans. Although that method is already a lot faster than brute-force random sampling, it is still too slow for routine use in practice; and it suffers from the “near-field bias”, i.e. it is biased towards wrong solutions with small translation (nearly identical scanner positions) because more keypoints are found close to the scanner, where more detail is observed.

As a simple measure to drastically speed up the computation we exploit the fact that the method is parallelizable by design and can be distributed across multiple processors now routinely available in standard desktop computers. To improve its robustness we replace the 0-1 error function of RANSAC with a truncated least-squares error (that combination is known as MSAC), and introduce a simple prior that favours scan positions above a certain minimum distance to counter the near-field bias. These two measures both do not increase the runtime.

Finally, it has been suggested that one can avoid the repeated evaluation of the full cost function for each random sample, and instead find re-occurring solutions via Hough-style voting (e.g. Torii et al., 2011). The intuition is that correct solutions have higher support and will therefore be found more often, whereas there is no systematic structure in the incorrect samples. It turns out that this strategy does not work well for the scan registration problem, and seriously degrades the result.

Source code within the framework of the open-source *Point Cloud Library* (PCL, Rusu and Cousins, 2011) as well as test data will be made available after publication.

2. RELATED WORK

A variety of approaches exist for fully automated, coarse registration of point clouds without artificial markers. Most of them have a common structure: first, extract a set of features or keypoints; second, sample subsets and match overlapping areas; and third, align the point clouds with the transformation estimated from the best match. Finally, coarsely aligned point clouds are fine-registered with standard ICP (Besl and McKay, 1992) or some equivalent. While this general workflow is usually the same, different techniques have been proposed for each step.

One strategy for feature extraction is to describe the scene geometry via planar surfaces (e.e. Dold and Brenner, 2006) or to search for salient directions (e.g. Novák and Schindler, 2013), which can be augmented with 2D features (Zeisl et al., 2013). Another natural strategy is to compute 2D keypoints in range or intensity images of scans (Böhm and Becker, 2007; Kang et al., 2009). Generally, methods that strongly rely on surface geometry or on 2D features are sensitive to strong viewpoint changes and become unreliable in scenes with large depth range and/or frequent self-occlusions. Representing point clouds with a sparse set of 3D features appears to be more robust to viewpoint changes (Flint et al., 2007; Allaire et al., 2008; Lo and Siebert, 2009; Flitton et al., 2010).

Variants of RANSAC (Fischler and Bolles, 1981) are often used to find potentially corresponding points. For example, Rusu et al. (2009) proposed a method they coined *Sample Consensus Initial Alignment* (SAC-IA) to match corresponding point triplets between clouds of n points. However, its complexity is $\mathcal{O}(n^3)$, which quickly becomes infeasible if dealing with laser scans of several million points. Aiger et al. (2008) reduced runtime complexity to $\mathcal{O}(n^2)$ by adding a fourth point to the triplets such that all four points are roughly coplanar. This method is efficient for moderately large point clouds and provides high success rates if points are uniformly distributed. Yang et al. (2013) recently proposed a globally optimal solution of ICP named *Go-ICP*. They show that coarse and fine registration can be done in one step by global optimization with a branch-and-bound scheme that avoids local minima. The method has been demonstrated on point clouds with a few thousand points, but does not scale up to larger data sets.

In the present paper, we evaluate several possible extensions of the K-4PCS method (Theiler et al., 2013), a variant of the 4PCS method that matches sparser clouds of 3D keypoints extracted from the original data. The following extensions are tested:

- a second, purely geometric keypoint detector that captures discriminative, local point distributions,
- parallelization and nested clustering to speed up the computation,
- MSAC (Torr and Zisserman, 2000) instead of plain RANSAC, that uses a truncated least-squares penalty rather than a binary inlier/outlier threshold,
- a refined cost function that penalizes unlikely solutions where the scanner positions are too similar.

Overall, we achieve a significant speed-up, while at the same time decreasing the failure rate of the registration procedure.

3. K-4PCS: COARSE POINT CLOUD REGISTRATION

In an earlier work we proposed the *Keypoint-based 4-Points Congruent Sets* method (K-4PCS, Theiler et al., 2013). In a nutshell,

it uses sparse clouds of 3D keypoints, which are matched with the *4-Points Congruent Sets* (4PCS) algorithm of Aiger et al. (2008). In this paper, we present modifications and extensions to the original K-4PCS, which focus on its main bottlenecks, namely limited robustness against uneven keypoint distribution as well as repeated structures, and long computation times in case of low scan overlap. We start with a brief overview of the original K-4PCS (i.e. 3D keypoint extraction, keypoint matching), more details can be found in Theiler et al. (2013).

3.1 3D keypoint extraction

Although the matching algorithm of Aiger et al. (2008) is computationally more efficient than simple 3-point sampling, handling standard TLS point clouds with millions of points is still infeasible. Therefore, the point cloud size is significantly reduced by extracting a (much smaller) set of discriminative keypoints, while discarding all other points. First a voxel grid filter is applied, which divides the 3D space into a regular grid of blocks (i.e. voxels) of size τ . Each block is represented by the centroid of all inlying scan points. On the one hand, this ensures a drastic reduction of the point count, and on the other hand it mitigates the strongly unequal point density inherent in polar acquisition methods. Then, 3D keypoints are extracted, which are a lot sparser than the full voxel grid, but repeatable and thus useful for matching. Note, K-4PCS matching does not involve local descriptors of keypoints, rather it relies solely on the relative keypoint positions. Registration is thus independent of a particular kind of keypoint detector. We test (see 5.) with 3D *Difference-of-Gaussian* (DoG) and 3D *Harris* keypoint detectors, which are described in the following.

3D DoG keypoints: DoG keypoints were introduced by Lowe (1999) as part of an image matching framework. Being invariant to scaling, rotation as well as translation, DoG keypoints are standard in 2D image processing, especially in combination with the SIFT descriptor. Here, we extract DoG keypoints directly in 3D to avoid points that are unstable across different view points (e.g. object silhouettes, depth discontinuities). The 3D DoG keypoint extractor (Rusu and Cousins, 2011) uses LiDAR return intensities I to select points with high contrast to their direct neighbours in 3D space. At different blur levels τ_k (with $k = 1 \dots m$), a Gaussian response G for each point is calculated, taking into account all points in the neighbourhood N with Euclidean distances $d < 3 \cdot \tau_k$ (Eq. 1).

$$G_i(\tau_k) = \frac{1}{\sum_{j \in N} g_j(\tau_k)} \cdot \sum_{j \in N} g_j(\tau_k) \cdot I_j \quad (1)$$

$$g_j(\tau_k) = e^{-0.5 \cdot d_j^2 / \tau_k^2}$$

To obtain DoG responses R^G at each point, adjacent scales are subtracted (Eq. 2),

$$R_i^G(\tau_k) = G_i(\tau_{k+1}) - G_i(\tau_k) \quad (2)$$

A valid keypoint is found, if the DoG response of the point is a local maximum or minimum in a $3 \times 3 \times 3$ neighbourhood, and exceeds a given threshold R_{min}^G .

3D Harris keypoints: In contrast to the original Harris corner detector (Harris and Stephens, 1988) based on image gradients, the 3D version (Rusu and Cousins, 2011) starts from local point normals. Thus, only geometrical properties are used and, unlike DoG, the detector is independent of laser intensities. For a point i , the local normal $\mathbf{n}_i = (n_{xi}, n_{yi}, n_{zi})^\top$ is calculated using the set N of neighbours within $r = 3 \cdot \tau$. The covariance matrix

at each point is determined using the normals of all points inside neighbourhood N . Then for each point, the Harris response

$$R^H = \frac{1}{|N|} \text{Det} \left(\sum_{i \in N} \mathbf{n}_i \cdot \mathbf{n}_i^\top \right) - \gamma \frac{1}{|N|} \cdot \text{Tr} \left(\sum_{i \in N} \mathbf{n}_i \cdot \mathbf{n}_i^\top \right)^2 \quad (3)$$

is determined, here γ is a constant. Similar to the DoG detector, keypoints are local maxima in response space that must exceed a threshold R_{min}^H .

3.2 Geometrically constrained keypoint matching

K-4PCS registration (Theiler et al., 2013) builds upon the *4-Points Congruent Sets* (4PCS) algorithm, which is an efficient matching algorithm on the basis of geometric constraints.

The straight-forward solution to register two 3D point clouds is to find two sets of congruent point triplets and to use them for deriving the parameters of a rigid-body transformation. However, this approach has computational complexity of at least $\mathcal{O}(n^3 \log n)$ with n the number of points in the point cloud (Irani and Raghavan, 1996). Aiger et al. (2008) showed that adding a fourth point (which is approximately coplanar to the base triplet) and matching quadruples reduces runtime to $\mathcal{O}(n^2)$.

Matching of point quadruples is in general similar to the standard approach based on three corresponding points (Fig. 1). From a source point cloud \mathcal{S} , a four-point base $\mathcal{B}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \in \mathcal{S}$ is randomly sampled. The points of \mathcal{B} are thereby constrained to approximately lay on a plane. Matches $\mathcal{M}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{q}_1, \mathbf{q}_2) \in \mathcal{T}$ are then detected and evaluated using a support score (i.e. fraction of inliers between \mathcal{T} and the transformed \mathcal{S}). The random base selection, matching, and evaluation is repeated L times. The returned solution is the match which score either exceeds a given threshold Γ or the match with the largest support after all trials.

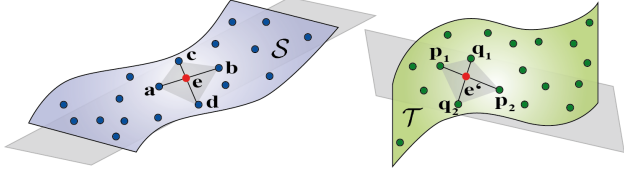


Figure 1: Principle of 4PCS with base set $\mathcal{B}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \in \mathcal{S}$ and a corresponding congruent point set $\mathcal{M}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{q}_1, \mathbf{q}_2) \in \mathcal{T}$ making use of the diagonal intersection point \mathbf{e} .

The computational advantage of matching quadruples rather than triplets comes from the fact that intersection ratios (r_1, r_2 , Eq. 4) of quadrangle diagonals are invariant under affine and therefore also under rigid-body transformations (Huttenlocher, 1991).

$$r_1 = \frac{\|\mathbf{a} - \mathbf{e}\|}{\|\mathbf{a} - \mathbf{b}\|}, \quad r_2 = \frac{\|\mathbf{c} - \mathbf{e}\|}{\|\mathbf{c} - \mathbf{d}\|} \quad (4)$$

Given diagonal ratios (r_1, r_2) from \mathcal{B} , matches can efficiently be found by calculating intersection points,

$$\begin{aligned} \mathbf{e}'_p(r_1) &= \mathbf{p}_1 + r_1 \cdot (\mathbf{p}_2 - \mathbf{p}_1) \\ \mathbf{e}'_p(r_2) &= \mathbf{p}_1 + r_2 \cdot (\mathbf{p}_2 - \mathbf{p}_1) \end{aligned} \quad (5)$$

$\mathbf{e}'_q(r_1)$ and $\mathbf{e}'_q(r_2)$ are calculated similarly using \mathbf{q}_1 and \mathbf{q}_2 , then quadrangles are checked for affine congruency,

$$\begin{aligned} \|\mathbf{e}'_p(r_1) - \mathbf{e}'_q(r_1)\| &< \delta \quad \text{or} \\ \|\mathbf{e}'_p(r_2) - \mathbf{e}'_q(r_2)\| &< \delta. \end{aligned} \quad (6)$$

Because the absolute scale of TLS point clouds is known computational cost can be further reduced, by restricting possible

matches to those where the diagonals in $\mathcal{M} \in \mathcal{T}$ and in $\mathcal{B} \in \mathcal{S}$ have similar lengths.

4. EXTENSIONS AND MODIFICATIONS

Although K-4PCS is generally well suited to register big, unevenly distributed TLS point clouds, runtime becomes impractical in case of small overlaps, and failures still occur in presence of weak or repeated structures. Here, we introduce several possible ways to accelerate computation and to make K-4PCS more robust.

4.1 Conceptual improvements

In standard 4PCS matching, the best match is evaluated based on a score value, namely the fraction of points in source cloud for which a match is found in the target cloud. Experiments reveal that this evaluation criterion does not always produce a correct registration (i.e. one from which ICP converges to the desired solution). One obvious reason is the inlier threshold δ that must be adapted to the estimated input point cloud density. Evaluation based on the sparse cloud of keypoints thus leads to a rather large δ which in turn yields comparatively high support also for wrong (e.g. symmetrical) solutions. Therefore, we use a modified matching cost.

MSAC: In K-4PCS the cost ρ_F of a putative solution is calculated like in RANSAC. Given the (squared) Euclidean residuals ϵ^2 between points i in the transformed source cloud and their closest neighbours in the target cloud,

$$\rho_F = \frac{1}{n} \sum_i \rho(\epsilon_i^2) \quad \text{with} \quad \rho(\epsilon_i^2) = \begin{cases} 0 & \epsilon_i^2 \leq \delta^2 \\ 1 & \epsilon_i^2 > \delta^2 \end{cases} \quad (7)$$

A drawback of this binary decision is the strong dependency of the resulting score on the inlier threshold δ . Since the inlier decision is already based on the squared Euclidean distance ϵ^2 , the cost ρ_i for a given point can, without additional computation, be changed to

$$\rho_A(\epsilon_i^2) = \begin{cases} \epsilon_i^2 / \delta^2 & \epsilon_i^2 < \delta^2 \\ 1 & \epsilon_i^2 \geq \delta^2 \end{cases} \quad (8)$$

This modified cost function $\rho_A(\epsilon^2)$ is called *m-estimator sample consensus* (MSAC, Torr and Zisserman, 2000). It decreases the influence of the threshold δ by considering the residuals of inliers, while outliers still receive a fixed penalty. Note, although further improvement could possibly be achieved by introducing an estimator of the threshold δ (MLESAC, Torr and Zisserman, 2000), we did not pursue this avenue because it increases the processing time.

Translation costs: The second term of our proposed cost function tries to counter the effect that K-4PCS matching is biased towards solutions with small translation T , which is a direct result of the higher keypoint density in the near-field of the scanner station. The total cost is thus $\rho_A + \lambda \rho_B$. We introduce a sigmoid function, limited by a task-defined minimum and a maximum translation (T_{min}, T_{max}).

$$\rho_B(R_i) = \begin{cases} 1 & R_i \leq 0 \\ \frac{1}{2}(1 + \sin(R_i \pi + \frac{\pi}{2})) & 0 < R_i < 1 \\ 0 & R_i \geq 1 \end{cases} \quad (9)$$

with $R_i = \frac{T_i - T_{min}}{T_{max} - T_{min}}$

This cost function favours solutions with larger translation, whereas overly close scanner positions become less likely. Such an assumption is reasonable for most TLS applications, where stations are placed apart to cover complementary scene parts. Note that the cost function does not impose any hard constraints, but can be viewed as soft prior that favours scanner locations that are far enough apart. Special cases where two scans are acquired from almost the same viewpoint (e.g. one above the other due to obstacles) can still be successfully registered, but need substantially higher support from the data.

Note that a similar prior could also be constructed for the rotation between scans, e.g. favouring parallel z -axis if instrument is levelled. We have so far not experienced failures with grossly wrong rotation, most likely because in our applications there always was a large number of points on the ground.

4.2 Speeding up computation

A main shortcoming of K-4PCS are its long computation times in presence of low overlap between adjacent scans (with $< 40\%$ overlap up to tens of minutes per pair). We resort to parallelization and also test nested clustering to reduce runtime.

Multi-threading The 4PCS method is by construction parallel: each of the L trials (i.e. random base selection, match detection, match evaluation) is independent of all others. A straight-forward speed-up is therefore possible by running the trials, which are the main bottleneck, in parallel. Our implementation uses the *OpenMP* application programming interface (OpenMP ARB) to distribute separate trials of K-4PCS to different threads. The best solution per thread is recorded and after all L trials, the overall winner is determined (in a single thread). As the computationally dominating part of the matching runs in parallel, we can expect a speed-up by a factor close to the number of cores available for multi-threading. In practice this factor is reduced by the non-parallelizable parts before and after sample evaluation and by the multi-threading overhead.

Clustering In K-4PCS, like generally in random sampling methods, the evaluation of candidate matches is a main bottleneck, and it can be expected that in fact evaluation is carried out multiple times for similar possible solutions (i.e. with approximately the same transformation parameters, but resulting from different candidate matches). It has been observed that if the correct solution is the one with the largest support in the data, then – at least in principle – it should be found more often than any other solution during the L trials (Torii et al., 2011). That is, the correct solution should be detectable as a local maximum in a histogram over the L sets of transformation parameters. Based on this intuition, we try to speed up the search for the correct solution. In practice, more than one solution occurs multiple times, so one has to detect a small, fixed number of maxima that correspond to clusters of possible alignments. Representatives for each of these candidate solutions are scored as described previously, whereas all other solutions are discarded.

Building a histogram in a 6-dimensional solution space is rather impractical. The usual strategy is thus to sacrifice some discriminative power and reduce the voting space. We represent the transformation corresponding to a sample (dubbed a “possible solution”) by the length of its translation vector and the angle of its rotation in axis-angle representation. As already explained in 4.1, K-4PCS is biased towards solutions with very small translation. To compensate this effect we weight histogram entries with a sigmoid function between $[0.5 \dots 1]$, truncated by T_{min} and T_{max} . A weighted 2D occurrence histogram of translation vector and



Figure 2: Example of a smoothed 2D histogram of all possible solutions with the ten detected local maxima (red) and the true solution from the ground truth (green). Abscissa: 3D rotation angle; ordinate: 3D translation.

rotation angle is thus generated, smoothed with a Gaussian filter (1σ), and ten local maxima are detected (see Fig. 2). Because the 2-parameter representation of a 3D alignment is not unique, possible solutions in each local maximum are further accumulated into a 1D histogram based on the angle between their translation vector and a reference direction (chosen to be the translation vector of the first solution). Again, the histogram is smoothed (1σ) and five local maxima are detected. Finally, match evaluation as described in 4.1 is done on the average of the transformation candidates at each detected maximum.

At most 50 (5×10) possible solutions have to be evaluated, much fewer than the initial amount of possible solutions (in our tests between 2000 and 100000, see 5.). On the downside, positions of local histogram maxima are somewhat inaccurate compared to the optimal solution and potentially too coarse to receive a high fitness score. To counter this effect a few iterations of standard ICP with decreasing inlier threshold are run on the cloud of key-points (with the possible solution as initial guess).

5. EVALUATION

We experimentally evaluate the previously described modifications of K-4PCS registration with respect to geometric registration accuracy, error rate, and computational efficiency. Geometric registration accuracy is measured with the root mean square error (RMSE) between true correspondences of the transformed source cloud \mathcal{S}' and the target cloud \mathcal{T} after *coarse* alignment. True correspondences are derived from ground truth that was generated through manual alignment followed by fine registration with standard ICP. The accuracy σ_0 of the ground truth is $\approx 5 \text{ mm}$.

Recall that our matching method is based on random sampling of keypoint quadruples and thus results may vary. We thus repeat each registration $m = 50$ times (with constant parameters) to compute the error rate $\hat{E} = \frac{1}{m} \cdot \sum_m E$. The variation of the error rate is determined via cross validation. Since the goal of K-4PCS is to provide a coarse alignment that serves as input to fine-registration (i.e. ICP), a test is considered successful if the solution falls into the ICP convergence basin. Therefore we execute standard ICP after coarse alignment. The criteria for success is henceforth the RMSE after refinement ($RMSE_{ICP}$):

$$E = \begin{cases} 0 & RMSE_{ICP} \leq 3 \cdot \sigma_0 \\ 1 & RMSE_{ICP} > 3 \cdot \sigma_0 \end{cases} \quad (10)$$

Computational efficiency is represented by the total runtime of K-4PCS, i.e. the time to extract keypoints t_{KP} and the matching time t_M . Because t_{KP} ($< 15 \text{ s}$ for all tests) can be considered as part of the pre-processing and has not been optimized (e.g. parallelized), only t_M is evaluated in detail. All tests were carried out on a 64 Bit desktop computer with 8 cores (3.4 GHz) and 16 GB RAM.

We have run tests on four data sets (including the two used in Theiler et al., 2013). The different data sets address different challenges of registering TLS point clouds. On the one hand, we use an indoor data set (dubbed *Office*) with rather large scan overlaps of $\approx 80\%$ and simple geometry, which, however, gives rise to rotationally symmetric solutions. On the other hand, we test on three outdoor data sets (dubbed *House*, *Urban*, *Forest*). While data sets *House* and *Urban* represent standard TLS projects with the scanner positioned around an object of interest, data set *Forest* is an extreme case, where scans are taken in the middle of a forest.

The evaluation is based on tests with fixed basic K-4PCS parameters. Suitable values for the cell size of the voxel grid (τ), the keypoint type (KP-type), minimum keypoint response R_{min} and the estimated overlap (Overlap) were set empirically according to the scanner setup and the environment (see Tab. 1).

| | τ in cm | KP-type | R_{min} | Overlap |
|---------------|--------------|---------|-----------|---------|
| <i>Office</i> | 5 | DoG | 0.01 | 80% |
| <i>House</i> | 10 | Harris | 0.01 | 50% |
| <i>Urban</i> | 10 | Harris | 0.01 | 40% |
| <i>Forest</i> | 10 | Harris | 0.001 | 50% |

Table 1: Fixed K-4PCS parameters for the different test data sets.

The discussed modifications of the K-4PCS framework are evaluated incrementally:

- (a) the original K-4PCS framework,
- (b) with multi-threading (see 4.2),
- (c) like (b), with the score function based on MSAC (see 4.1),
- (d) like (c), with additional translation cost (see 4.1),
- (e) using nested clustering followed by (d) (see 4.2).

5.1 *Office* data set

The indoor data set *Office* was acquired in a standard office room of size $10 \times 15 m^2$ and consists of five scans of medium resolution ($\approx 10^7$ points). Apart from standard office furniture (tables, chairs, shelves), the room features several cylindrical pillars (Fig. 3).

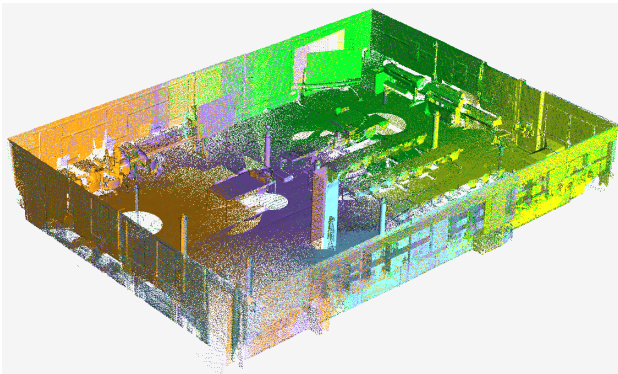


Figure 3: *Office* data set with five scans acquired in a standard office. For better visibility, the ceiling has been removed and only 33% of the points are displayed.

Tab. 2 shows results of the five test setups. The success rate of the original K-4PCS ($\approx 97\%$) is already quite high. Nevertheless, the proposed modifications (test (c) and (d)) improve the robustness of the matching method. While the replacement of standard RANSAC with MSAC decreases the failure rate only by $< 1\%$,

| | \dot{E} in % | t_M in s | $RMSE$ in cm |
|-----|----------------|-------------|----------------|
| (a) | 2.8 ± 0.8 | 89 ± 19 | 14.2 ± 5.3 |
| (b) | 3.4 ± 0.9 | 18 ± 5 | 14.6 ± 5.9 |
| (c) | 2.6 ± 0.9 | 18 ± 5 | 12.7 ± 5.8 |
| (d) | 0.8 ± 0.4 | 18 ± 5 | 13.2 ± 6.3 |
| (e) | 7.6 ± 1.3 | 4 ± 0 | 7.8 ± 11.5 |

Table 2: Failure rate \dot{E} , matching time t_M , and geometrical accuracy of coarse alignment (RMSE) of indoor *Office* data set.

the extended score function including penalties for low translation (here $R_{min} = 1 m$, $R_{max} = 4 m$) is able to prune almost all wrong solutions. Note, that 0.8% failures means 4 trials out of 500. For *Office* the introduction of translation costs significantly improves results because incorrect solutions of standard K-4PCS are caused by the rotation symmetry of the data set (especially in case of scans in opposite room corners). More precisely, wrong solutions stem from 180° rotated solutions that place one scan almost directly on top of the other, which is discouraged by the translation cost function.

Regarding computational efficiency, test (b) shows that distributing the main part of the matching method onto 8 cores reduces runtime by a factor of ≈ 5 . As expected, the modifications in tests (c) and (d) do not increase runtime. Evaluation based on nested clustering further reduces runtime by a factor of ≈ 5 leading to a total speed-up of ≈ 23 . On the negative side, a significant drop of the success rate can be observed in test (e). This indicates that the assumption is not always true that the correct solution is drawn more often than any single wrong solution. Because of the large speed-up resulting from clustering, another test is done with twice the number of trials L . Naturally, this increases runtime but still gives a two-fold speed-up over test (b). In that test the success rate reaches almost 100% again. However, as we will see in further experiments, this modification is not generally applicable and was therefore not pursued further.

The alignment accuracy does not change significantly. Note the rather large standard deviation of the $RMSE$ in test (e), which indicates that some solutions are even more accurate than 8 cm while others are significantly worse. The reason for the accuracy improvement in (e) is the additional ICP refinement step applied directly to the clouds of keypoints.

5.2 *House* data set

The *House* data set consists of six consecutive outdoor scans with $> 2 \cdot 10^7$ mio points yield four scan pairs with reasonable overlap ($\approx 50\%$). The surroundings of the house are dominated by grassland, vegetation, small paths, and a street (Fig. 4). Thus, scan overlaps mainly comprise flat ground and the house itself, which again gives rise to rotationally symmetric solutions. In addition, the wooden facade construction includes repetitive structures that make correct coarse registration challenging. Failure rates, runtimes, and accuracies for *House* are summarized in Tab. 3.

Success rates are generally lower compared to the *Office* data set, but still $\approx 88\%$ of the runs are registered correctly using the full score function. The positive effect of the proposed modifications can still be observed, but their relative impact is lower (failure rate is reduced by a factor of ≈ 1.2). The effect of MSAC is insignificant, while the translation cost has some impact. The translation cost function is adapted to the larger distances between scanner positions with $T_{min} = 5 m$ and $T_{max} = 10 m$. Test (b) shows that multi-threading again reduces the computational cost of matching by a factor > 5 and remains constant over tests (c) and (d).

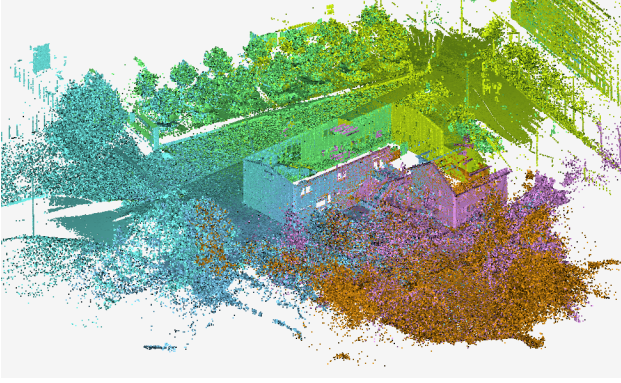


Figure 4: *House* data set with six scans acquired around a house. Only 25% of the points are displayed.

| | \dot{E} in % | t_M in s | $RMSE$ in cm |
|-----|----------------|--------------|-----------------|
| (a) | 16.5 ± 2.5 | 947 ± 58 | 72.1 ± 26.1 |
| (b) | 15.5 ± 2.9 | 170 ± 10 | 80.9 ± 31.1 |
| (c) | 15.0 ± 2.8 | 169 ± 10 | 78.5 ± 37.5 |
| (d) | 12.5 ± 2.6 | 168 ± 9 | 79.9 ± 35.2 |
| (e) | 60.0 ± 3.8 | 141 ± 5 | 95.5 ± 52.1 |

Table 3: Test results of *House* data set: failure rate \dot{E} , matching time t_M , and accuracy (RMSE) of coarse alignment.

The RMSE of the true correspondences after coarse alignment remains the same, and sufficient for ICP to converge to a correct solution.

In contrast, nested clustering does not work for this data set, resulting in \dot{E} of $\approx 60\%$. The reason is that the histogram often does not contain a clear peak at the position of the correct solution. Thus the correct solution appears not often enough to be detected as one of the most dominant local maxima (Fig. 5). Moreover, clustering is only $\approx 20\%$ faster than the multi-threaded solution, which is due to the following: first, the number of possible solutions to evaluate for the *House* data set (≈ 10000) is lower than for *Office* (≈ 50000); second, the larger amount of key-points (≈ 5200) leads to an increased effort to generate potential solutions, which decreases the influence of the faster evaluation step. In conclusion, clustering essentially fails and does not make sense here.

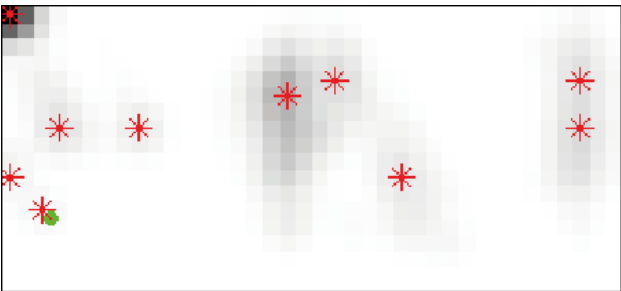


Figure 5: An example of the occurrence histogram of tests with the *house* data set. The correct solution (green) does not appear as a clear peak among the ten maxima (red) causing the method to fail in $\approx 60\%$ of all tests based on nested clustering. Abscissa: 3D rotation angle; ordinate: 3D translation.

5.3 Urban data set

The *Urban* data set (Fig. 6) consists of four scans with high resolution ($> 2 \cdot 10^7$ points per scan). Scans cover a Roman arch

in Rome and its surrounding paths, buildings, and vegetation. In addition to the low overlap between adjacent scans ($\approx 40\%$), vegetation and artefacts caused by moving people make registration of this data set challenging. The setup results in only four sufficiently overlapping, matchable scan pairs.



Figure 6: *Urban* data set with four scans positioned around a Roman arch. Notice the artefacts caused by moving people in front of the arch. For better visibility, only 25% of the points are displayed.

Test results in Tab. 4 reflect the findings described in 5.2. The computational efficiency using multi-threading is significantly increased by a factor of > 5 for tests (b), (c) and (d). The error rate with the original K-4PCS framework is already quite low ($\approx 5\%$). MSAC did not significantly improve this, while the translation cost ($T_{min} = 5$ m and $T_{max} = 10$ m) leads to noticeably higher success rates (+2%).

| | \dot{E} in % | t_M in s | $RMSE$ in cm |
|-----|----------------|--------------|------------------|
| (a) | 4.5 ± 1.6 | 682 ± 21 | 94.6 ± 42.3 |
| (b) | 5.5 ± 1.8 | 105 ± 2 | 96.8 ± 43.0 |
| (c) | 5.0 ± 1.7 | 105 ± 2 | 93.7 ± 41.3 |
| (d) | 3.0 ± 1.4 | 105 ± 2 | 87.1 ± 43.4 |
| (e) | 35.5 ± 3.8 | 106 ± 2 | 102.0 ± 44.6 |

Table 4: Failure rate \dot{E} , matching time t_M , and accuracy of the coarse alignment (RMSE) of for the *Urban* data set.

Test (e) shows a marked increase of the error rate by $\approx 30\%$, while no speed-up was achieved. Reasons for this failure are similar to the ones described in 5.2. Overall, clustering as a replacement for full scoring of all samples appears not to be suitable for scan registration, at least for realistic outdoor settings.

5.4 Forest data set

The *Forest* data set is used to test the proposed framework under extreme conditions. Six high resolution scans – divided into two groups A, B that only marginally overlap – were acquired in a forested area dominated by bushes and trees (Fig. 7). The three scans per group all have overlaps of $\approx 50\%$.

In Tab. 5 the test results (error rate, geometrical accuracy, and matching time) are shown for both scan groups. We treat the two groups separately because (i) the characteristics of the forested areas are different (A is in a less dense part of the forest) and (ii) the scans of B are placed further apart from each other. Thus, we set different thresholds T_{min} , T_{max} for tests (d) (A: $T_{min} = 2$ m, $T_{max} = 5$ m; B: $T_{min} = 2$ m, $T_{max} = 8$ m).

Considering the tests with A, the small \dot{E} of $\approx 2\%$ in test (a) and (b) proves, that K-4PCS already works very well even in

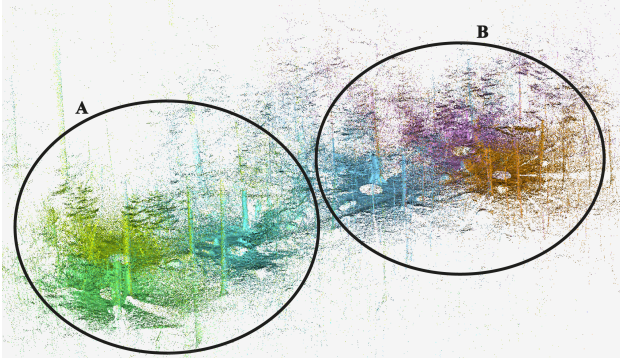


Figure 7: *Forest* data set with six scans divided into two triples *A* and *B*. Only 1% of the data is displayed.

| | | \dot{E} in % | t_M in s | $RMSE$ in cm |
|---|-----|----------------|--------------|------------------|
| A | (a) | 1.3 ± 1.1 | 239 ± 6 | 70.7 ± 34.8 |
| | (b) | 1.3 ± 1.1 | 57 ± 2 | 73.8 ± 36.7 |
| | (c) | 0.0 ± 0.0 | 57 ± 2 | 65.0 ± 33.1 |
| | (d) | 0.7 ± 0.7 | 57 ± 2 | 67.0 ± 35.1 |
| | (e) | 85.3 ± 3.2 | 48 ± 1 | 130.9 ± 82.9 |
| B | (a) | 66.7 ± 4.3 | 235 ± 12 | 53.4 ± 28.7 |
| | (b) | 68.0 ± 4.3 | 57 ± 3 | 57.4 ± 31.2 |
| | (c) | 66.0 ± 4.3 | 57 ± 2 | 32.9 ± 14.7 |
| | (d) | 53.3 ± 4.6 | 58 ± 2 | 42.8 ± 19.0 |
| | (e) | 99.3 ± 0.7 | — | — |

Table 5: Failure rate \dot{E} , matching time t_M , and accuracy of the coarse alignment (RMSE) with the *Forest* data set.

this unstructured environment. Integrating MSAC and translation costs additionally reduces the error rate (0.7% equals 1 failure case). Multi-threading again strongly reduces runtime (factor ≈ 4) bringing it down to just below 1 minute.

Tests with scan triplet *B* show generally lower success rates (33–47%), seemingly caused by the already very large distances between scanner stations. This is confirmed by a serious reduction of \dot{E} if translation costs are considered (−14%), although many failure cases remain. A major problem in the presence of dense vegetation is that the theoretical overlap between scan pairs is not reflected in a similar amount of corresponding keypoints, because their repeatability suffers – in scans from different viewing directions the detector fires rather randomly on different bits of vegetation.

Nested clustering completely fails for the *Forest* data set. The problem lies in the unordered distribution of keypoints extracted from dense vegetation, leading to a very diffuse distribution of the solutions. The histogram becomes smeared and no clear peaks are detectable. Note that for *B* only one correct solution was found, so no time and accuracy were calculated.

6. CONCLUSIONS AND OUTLOOK

We have evaluated modifications to the K-4PCS matching method for fully automated, marker-less point cloud registration. The extensions aim to improve computational efficiency and robustness in the presence of symmetry and repeated structures.

Including MSAC (Torr and Zisserman, 2000) and a translation cost term in the score function proved beneficial, and improved

the success rates. The gains are mostly achieved by avoiding relatively rare, but disturbing failures in moderately difficult scenarios, whereas more fundamental methodological upgrades are still required to address difficult cases with frequent failures. Straight-forward multi-threading speeds up matching by a factor of ≈ 5 (or more, with last-generation machines with 12-24 cores). For all evaluated data sets runtime is < 3 minutes even with only small overlap, such that the algorithm becomes practical for productive applications. Compared to manual (coarse) registration, a reasonably fast automatic method is obviously easier to scale up to large projects with many scans (by simply using more computers). Beyond this advantage, we believe that also for a single scan pair further tuning of the implementation could potentially make K-4PCS faster than manual registration, even by an experienced operator.

With the proposed modifications the geometrical accuracy of the coarse alignment remains unchanged, and is generally accurate enough for subsequent ICP refinement. Overall, we have demonstrated that K-4PCS allows for fully automated scan registration in scan projects with not too difficult recording setups and, if appropriately implemented, is fast enough for practical use.

Switching from standard sample evaluation to clustering of consistent solutions turned out not to be a viable alternative. Under special conditions it produced a significant speed-up, at the price of a somewhat higher failure rate. In general, the assumption that the correct solution appears more often does not seem to hold as well as the more standard assumption that it has more inliers.

In spite of the proposed improvements the large majority of the remaining failures can still be attributed to symmetries or repeated structures. Although in some cases MSAC and the translation cost term alleviate the problem, they are not able to fully address it, and new solutions are needed. In future work we hope to score individual keypoints by their degree of uniqueness and saliency (e.g. Shtrom et al., 2013), such that one can prefer those points in a scene which are least likely to cause confusions. Another topic for future work is how to exploit the redundancy in larger scan projects. Typically the pairwise registration will be successful for many more scan pairs than needed, and it seems natural to utilise the resulting constraints to detect and remedy incorrect scan pairs, either iteratively or in one global process.

References

- Aiger, D., Mitra, N. J. and Cohen-Or, D., 2008. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics* 27(3), pp. 1–10.
- Akca, D., 2003. Full automatic registration of laser scanner point clouds. In: *Proc. Optical 3D Measurement Techniques VI*, pp. 330–337.
- Allaire, S., Kim, J., Breen, S., Jaffray, D. and Pekar, V., 2008. Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8.
- Bae, K.-H. and Lichti, D., 2004. Automated registration of unorganised point clouds from terrestrial laser scanners. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 35(Part B5), pp. 222–227.
- Bergevin, R., Soucy, M., Gagnon, H. and Laurendeau, D., 1996. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(5), pp. 540–547.

- Besl, P. and McKay, N. D., 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), pp. 239–256.
- Böhm, J. and Becker, S., 2007. Automatic marker-free registration of terrestrial laser scans using reflectance features. In: *Proc. Optical 3D Measurement Techniques VIII*, pp. 338–344.
- Censi, A., 2008. An icp variant using a point-to-line metric. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 19–25.
- Chen, Y. and Medioni, G., 1992. Object modelling by registration of multiple range images. *Image and Vision Computing* 10(3), pp. 145–155.
- Dold, C. and Brenner, C., 2006. Registration of terrestrial laser scanning data using planar patches and image data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36(Part 5), pp. 25–27.
- Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. of the Association for Computing Machinery* 24(6), pp. 381–395.
- Flint, A., Dick, A. and van den Hangel, A., 2007. Thrift: Local 3D structure recognition. In: *Proc. 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pp. 182–188.
- Flitton, G., Breckon, T. and Megherbi Bouallagu, N., 2010. Object recognition using 3D SIFT in complex CT volumes. In: *Proc. British Machine Vision Conference*, pp. 11.1–11.12.
- Franaszek, M., Cheok, G. and Witzgall, C., 2009. Fast automatic registration of range images from 3d imaging systems using sphere targets. *Automation in Construction* 18(3), pp. 265–274.
- Harris, C. and Stephens, M., 1988. A combined corner and edge detector. In: *Proc. 4th Alvey Vision Conference*, pp. 147–151.
- Huttenlocher, D., 1991. Fast affine point matching: an output-sensitive method. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 263–268.
- Irani, S. and Raghavan, P., 1996. Combinatorial and experimental results for randomized point matching algorithms. In: *Proc. 12th Annual Symposium on Computational Geometry*, pp. 68–77.
- Kang, Z., Li, J., Zhang, L., Thao, Q. and Zlatanova, S., 2009. Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images. *Sensors* 9(4), pp. 2621–2646.
- Lo, T.-W. R. and Siebert, J. P., 2009. Local feature extraction and matching on range images: 2.5D SIFT. *Computer Vision and Image Understanding* 113(12), pp. 1235–1250.
- Lowe, D., 1999. Object recognition from local scale-invariant features. In: *Proc. 7th IEEE International Conference on Computer Vision*, Vol. 2, pp. 1150–1157.
- Minguez, J., Montesano, L. and Lamiroux, F., 2006. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics* 22(5), pp. 1047–1054.
- Novák, D. and Schindler, K., 2013. Approximate registration of point clouds with large scale differences. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2(5/W2), pp. 211–216.
- Pottmann, H., Huang, Q.-X., Yang, Y.-L. and Hu, S.-M., 2006. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal of Computer Vision* 67(3), pp. 277–296.
- Rusu, R. and Cousins, S., 2011. 3D is here: Point cloud library (PCL). In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 1–4.
- Rusu, R., Blodow, N. and Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 3212–3217.
- Shtrom, E., Leifman, G. and Tal, A., 2013. Saliency detection in large point sets. In: *Proc. IEEE International Conference on Computer Vision*, pp. 3591–3598.
- Theiler, P. W., Wegner, J. D. and Schindler, K., 2013. Markerless point cloud registration with keypoint-based 4-points congruent sets. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 2(5/W2), pp. 283–288.
- Torii, A., Havlena, M. and Pajdla, T., 2011. Omnidirectional image stabilization for visual object recognition. *International Journal of Computer Vision* 91(2), pp. 157–174.
- Torr, P. and Zisserman, A., 2000. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* 78(1), pp. 138–156.
- Yang, J., Li, H. and Jia, Y., 2013. Go-ICP: solving 3D registration efficiently and globally optimally. In: *Proc. IEEE International Conference on Computer Vision*, pp. 1457–1464.
- Zeisl, B., Köser, K. and Pollefeys, M., 2013. Automatic registration of RGB-D scans via salient directions. In: *Proc. IEEE International Conference on Computer Vision*, pp. 2808–2815.