Eurographics Workshop on 3D Object Retrieval (2015)
I. Pratikakis, M. Spagnuolo, T. Theoharis, L. Van Gool, and R. Veltkamp (Editors)

Indoor Location Retrieval using Shape Matching of KinectFusion Scans to Large-Scale Indoor Point Clouds

A. Al-Nuaimi[1], M. Piccolrovazzi[1], S. Gedikli[2], E. Steinbach[1] and G. Schroth[1,2]

[1]Chair of Media Technology, Technische Universität München (TUM), Munich, Germany
[2]Navvis GmbH, Munich, Germany

Abstract

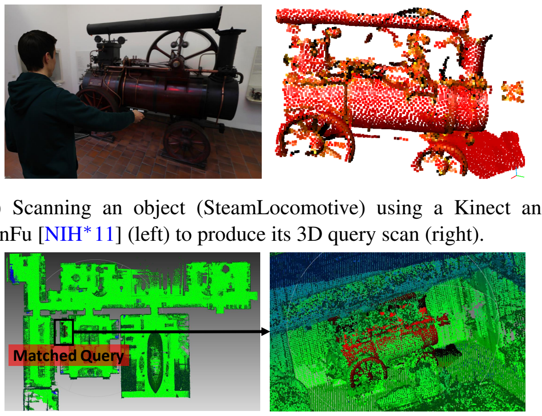
In this paper we show that indoor location retrieval can be posed as a part-in-whole matching problem of KinectFusion (KinFu) query scans in large-scale target indoor point clouds. We tackle the problem with a local shape feature-based 3D Object Retrieval (3DOR) system. We specifically show that the KinFu queries suffer from artifacts stemming from the non-linear depth distortion and noise characteristics of Kinect-like sensors that are accentuated by the relative largeness of the queries. We furthermore show that proper calibration of the Kinect sensor using the CLAMS technique (Calibrating, Localizing, and Mapping, Simultaneously) proposed by Teichman et al. effectively reduces the artifacts in the generated KinFu scan and leads to a substantial retrieval performance boost. Throughout the paper we use queries and target point clouds obtained at the world's largest technical museum. The target point clouds cover floor spaces of up to $3500m^2$. We achieve an average localization accuracy of 6cm although the KinFu query scans make up only a tiny fraction of the target point clouds.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

The advent of the Microsoft *Kinect* and similar cheap handheld 3D sensors has made 3D shape sensing of the local environment easily possible. The Kinect Fusion algorithm (KinFu) [NIH*11] can stitch multiple Kinect depth frames into a more extensive surface allowing the scanning of an object beyond single-view occlusions. Meanwhile, means for large-scale 3D indoor mapping in the form of point clouds have been developed [LCC*10, HSH*12].

In this paper we show that the 6-DOF pose of local shape scans obtained with a Kinect-like sensor and KinFu (as shown in Figure 1a) can be matched in a large scale indoor point cloud to accurately retrieve the indoor location of a user (as shown in Figure 1b). We use a feature-based 3D object retrieval (3DOR) system. Compared to the established camera-based localization schemes which are based on content-based image retrieval it has some fundamental advantages: First, the accuracy is no longer a function of the spatial density of the recorded reference views. Second, the local shape of an object is not affected by the lighting conditions. Third, by using KinFu, a view-independent and largely occlusion-free query can be generated. Finally, the sensor's 6-DOF pose can be retrieved achieving superior accuracy.



(a) Scanning an object (SteamLocomotive) using a Kinect and KinFu [NIH*11] (left) to produce its 3D query scan (right).

(b) Matching the query scan to its respective point cloud using the system explained in Section 2 retrieves the Kinect's 6-DOF pose identifying the person's location.

Figure 1: Pose retrieval of a KinFu scan in a large-scale indoor point cloud to retrieve the indoor location of a person.

DOI: 10.2312/3dor.20151052

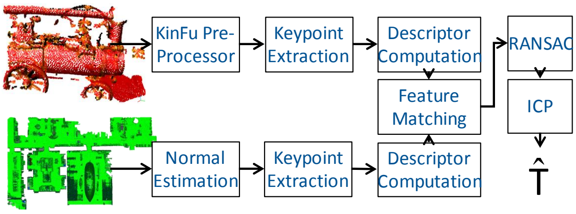


Figure 2: Retrieval system: The KinFu query is preprocessed (see Section 4). Keypoints are computed for the KinFu query scan and the target point cloud. A descriptor is computed for each keypoint. The descriptors are matched to determine point correspondences between the query and the target. A tentative alignment is computed using RANSAC and refined using ICP producing the 6-DOF homogeneous alignment transform $\hat{\mathbf{T}}$.

The used retrieval system (Section 2) computes descriptors of 3D keypoints of the KinFu scan and the target point cloud. Good descriptor quality is crucial for successful retrieval. The relatively large KinFu scans that we generate, as opposed to the standard table-top scans, exhibit strong distortions in the form of bent surfaces and amplified noise that adversely impact the descriptor quality.

Our contribution is an analysis of these KinFu scanning artifacts, which arise due to the largeness of the query scans which articulate 3D sensing distortions typical for Kinect-like sensors (Section 3). Moreover, we show how to effectively reduce these artifacts by proper 3D sensor calibration using the CLAMS technique [TMT13] together with preprocessing of the final KinFu scan (Section 4). Finally, we demonstrate the effectiveness of the location retrieval system in Section 5 using real data obtained at the *Deutsches Museum* in Munich achieving cm-level accurate localization.

2. Retrieval System

We use a 3DOR system that performs part-in-whole shape matching (as defined by Tangelder and Veltkamp in [TV04]) to retrieve the 6-DOF pose of the KinFu scan (henceforth called *query*) in the indoor point cloud (henceforth called *target*). Figure 2 shows that the KinFu scans are first preprocessed – as explained in detail in Section 4 – to handle scanning distortions and produce reliable surface normals. The normals for the target are also computed. For each keypoint a shape descriptor is computed. The descriptors of the query are matched to the target descriptors to establish query-target point correspondences. A random sample consensus (RANSAC) estimator is used to validate the correspondences and estimate the 6-DOF transformation that aligns the query to the target: At each RANSAC iteration three points are semi-randomly (see Section 5.5) sampled to establish a transformation hypothesis which is validated with the remaining correspondences. The iteration with the highest amount of *inliers* delivers the used transformation hypothesis. Finally, the Iterative Closest Point (ICP) algorithm [Zha94] runs to retrieve the pose more accurately expressed as the 6-DOF homogeneous transformation $\hat{\mathbf{T}}$.

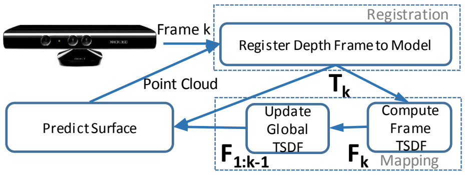


Figure 3: KinFu overview. An incoming depth frame is aligned to the most recent predicted surface to produce the transformation matrix $\mathbf{T_k}$. The TSDF of the registered frame F_k is computed and fused with the cumulative TSDF $F_{1:k-1}$ to produce $F_{1:k}$. A new surface is predicted from the viewpoint $\mathbf{T}_k$ to be used in the alignment of the next depth frame.

The presented retrieval system is inspired by the one presented by Aldoma at al. [AMT*12]. In that paper, the authors compare different local shape descriptors in terms of 3DOR performance. The Signature Histogram of OrienTations (SHOT) [TSDS10b] as well as the Unique Shape Context (USC) [TSDS10a] are identified as being the best in terms of retrieval performance among a group that includes six state-of-the-art local shape descriptors implemented in the Point Cloud Library (PCL) [RC11]. Given SHOT's relative compactness compared to USC, we decide to use it as a main descriptor. As a keypoint detector we use the Intrinsic Shape Signature of Zhong [Zho09] which has been shown to outperform many standard detectors in terms of relative repeatability under various distortions and transformations [FA14].

The used shape feature-based 3DOR system is substantially faster than the 4-point congruent sets (4PCS) algorithm of Aiger et al. [AMCO08]. Mellado et al. presented an accelerated version of 4PCS, the Super4PCS [MAM14]. 4PCS-based methods can be superior in cases with dominant semi-planar surfaces. In our case, however, we have many articulated shape features which are better exploited using a shape feature-based retrieval approach which was confirmed by our experiments.

3. KinFu Scan Issues

The KinFu scans suffer from distortions that can be attributed to two main sources: the sensor data and the KinFu reconstruction algorithm. The distortions are explained in detail in Section 3.2 preceded by a brief explanation of KinFu in Section 3.1 to aid in understanding the distortions.

3.1. KinFu algorithm

As shown in Figure 3, KinFu has two main processing functions: registration and mapping. These processing functions are interdependent whereby the outcome of registration is

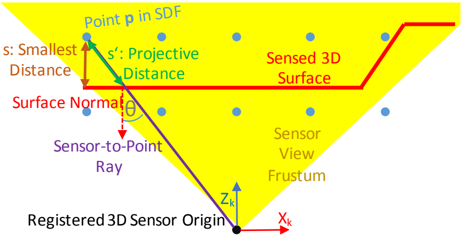


Figure 4: SDF computation in KinFu. KinFu uses the “projective distance” which is an approximation of the true smallest distance to the sensed surface. The projective distance s' for point **p** in the TSDF is always an overestimation of the true distance s. The incurred error increases with increasing θ.

used during mapping and the outcome of mapping is used for the registration of a newly incoming depth frame.

During registration, a new incoming depth frame is registered to the local scene to retrieve the 6-DOF pose of the 3D sensor. ICP with the point-to-plane metric [Zha94] is used for this purpose. In KinFu an incoming depth frame is registered against the most recently updated 3D shape model of the scene obtained through mapping resulting in highly accurate registration. This, however, requires updating the scene’s 3D scene model at frame rate.

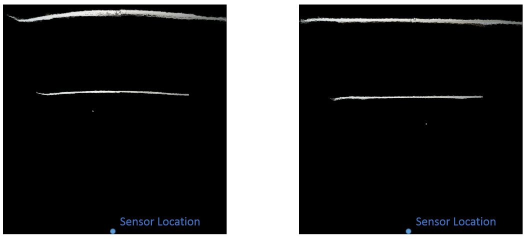
To compute the 3D scene model at frame rate a volumetric scene representation based on the Truncated Signed Distance Function (TSDF) [CL96] is used. The TSDF captures for each point in a cubic volume encompassing the scene the minimum distance to the 3D surface. Two TSDFs are maintained: One that accumulates the knowledge about the surface over multiple frames and another computed only using the sensed surface in the current registered depth frame.

KinFu uses a discretized lattice of 3D points as an approximation of a continuous TSDF. So a cubic volume of side length l is subdivided into *voxels* of side length (l/m). l is adapted to the largeness of the scene ($l = 300$ cm in our case). m is usually limited by the graphics card memory (we use $m = 512$ as in the original KinFu paper). The ratio l/m determines the granularity with which the surface is mapped.

The TSDF value at any point **p** in the lattice should be the smallest distance from the point to the sensed surface. KinFu, however, approximates this distance, as shown in Figure 4, by computing the projective distance. It is obtained along the ray connecting **p** to the sensor origin. It is argued that this approximation still leads to good mapping results while allowing computing the TSDF at high frame rates.

For any depth frame k, the TSDF volume F_k is computed. and subsequently fused with the cumulative volume $F_{1:k-1}$ using a per-voxel simple running average update rule.

Once the current mapping iteration is done, the surface is partially predicted from the perspective of the currently registered frame. This is used to provide a reference surface to be used in the registration of the next incoming depth frame. Hence, abrupt trajectory changes and movements can lead to ICP failure. At the end of the scanning, the most recently obtained cumulative TSDF is used to produce a 3D mesh. The zero-crossings inside the TSDF represent the surface.



(a) Without CLAMS calibration. (b) With CLAMS calibration.

Figure 5: 3D sensor raw data (Asus Xtion Pro Live) of a wall scanned from two distances (1.5m & 2.5m) shown as a point cloud from above. Despite IR camera calibration a bending of the wall is observed. The curvature of the bending increases with increasing distance from the sensor. Calibrating with CLAMS [TMT13] effectively reduces the bending.

3.2. KinFu query scan distortions

Surface bending. One fundamental issue we have faced is related to bent planar surfaces as shown in Figure 6. This issue can be mainly attributed to the raw Kinect data. We have observed that the raw 3D data suffers from non-linear distortions as shown in Figure 5a. Critically, planar surfaces appear curved and the curvature increases with increasing distance from the scene. Teichman et al. [TMT13] show that this is especially true for PrimeSense-based sensors (Microsoft Kinect, Asus Xtion, Primesense Carmine). The latter two sensors are particularly interesting because they can be carried around and thus lend themselves for our application. In our case we use the ASUS Xtion Pro Live.

Considering that KinFu essentially runs mapping and registration on each frame in succession, the bending of the raw 3D data is particularly harmful. Initial surfaces exhibiting the bending will cause new depth frames to be registered slightly wrong. This in turn results in a wrong mapping update in the cumulative TSDF which in turn affects future registrations. As a result the error propagates and with increasingly larger scans the bending effect is accentuated.

To prove this we perform the following experiment: We use KinFu to scan a scene at our lab which includes a large wall as well as some articulated objects to ensure accurate registration. We perform a number of different scans. In the first one, identified by label (1) in Figure 6 we stand far from the wall and pan the sensor left and right. In the second, we perform a similar scan, however at a close proximity from

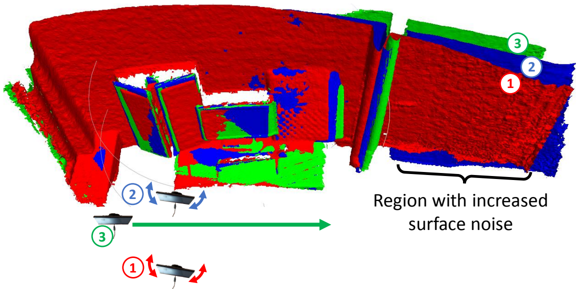


Figure 6: Lab scene scanned using KinFu with Xtion Pro Live using three different approaches: (1) Standing still; (2) Same as 1 but *closer* to the wall; (3) Scanning sideways while remaining close to the wall. The generated point clouds prove that surface bending is less in (2) than in (1) due to the decreased curvature of the raw 3D data at lower distances as shown in Figure 5. Maintaining a close distance to the surface as in (3) further reduces the bending.

the wall. The generated point clouds, shown in the same figure, clearly exhibit bending which increases towards the edges. However and as expected, the bending of scan (2) is notably less than that in (1). In a third experiment we scan parallel to the wall while maintaining a close distance to it. As can be seen in Figure 6, experiment (3) exhibits a substantially reduced bending compared to (1) and (2) albeit at the cost of making the scene scanning complicated. The bending problem was less severe when we used a Microsoft Kinect which, however, is less portable than the Xtion.

It remains to be said that the errors in depth impact also the calculated x and y coordinates of the scan points as these are computed using the pin-hole camera model. Indeed, Figure 6 shows that the less the wall is bent, the larger is also the extent of the scene and the scan dimensions are closer to reality which is important for the descriptors.

Sensor noise. Besides surface bending another issue we have to deal with is surface noise which affects surface normal estimation. The used SHOT descriptor as well as all other mentioned descriptors in [AMT*12] rely on surface normals. USC, 3DSC and SI require proper normals to setup the local descriptor reference frame of a keypoint. FPFH, RSD and SHOT compute a keypoint's descriptor using a function of the normals of all points within a defined vicinity. Hence, errors in the normal estimation typically distort the descriptors which adversely impacts the subsequent feature matching.

Point $\mathbf{p}_i$'s surface normal can be typically computed by first computing the covariance matrix of the points in $\mathbf{p}_i$'s neighborhood as follows:

$$\mathbf{C}_i = \sum_{j \in \mathcal{N}_i} (\mathbf{p}_i - \mathbf{p}_j)(\mathbf{p}_i - \mathbf{p}_j)^T \tag{1}$$

where $\mathcal{N}_i$ is the set of points within radius r_n of $\mathbf{p}_i$. The eigenvector corresponding to the smallest eigenvalue of $\mathbf{C}_i$ is deemed as the normal vector.

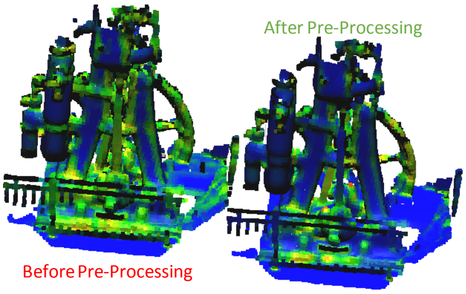


Figure 7: KinFu query pre-processing shown for SteamEngine. The color represents the surface curvature. The greener, the higher the curvature value. Prior to retrieval MLS and SOR filters (see Section 4) are used to reduce surface noise and eliminate spurious points. Especially flat surfaces as well as edges benefit from the filtering.

Surface noise and surface distortions can have significant impact on the covariance matrix **C** and the computed eigenvectors [MN03]. This is is especially critical when r_n is small. Unfortunately, this is the case with our museum KinFu scans which have articulated small shapes which require r_n to be small (14cm). Figure 7 shows the surface curvature of an example query scan. The unprocessed scan surface is noisy and results in noisy normals.

A fundamental noise source is again the 3D sensor. Assuming a Gaussian error model, Koshelham and Elbrink [KE12] showed that the standard deviation in the measured depth of a point by Kinect can be given as:

$$\sigma_d = \sigma_\delta d^2 \alpha \tag{2}$$

where d is the depth of the point, σ_δ is the standard deviation of the measured disparity, and α is a constant that depends on the Kinect camera parameters. So in essence, the Kinect depth error increases quadratically with increasing depth. KinFu implicitly runs a maximum likelihood estimation over multiple measurements from multiple frames by averaging new TSDFs into the cumulative TSDF, as explained in Section 3.1. This results in substantially smoother surfaces when comparing to the raw point clouds delivered in individual frames. Nevertheless, the noise will inevitably increase with increasing distance from the scene as the variance in the estimate itself increases.

Due to the use of the projective distance when computing the TSDF, see Section 3.1, the noise can furthermore be amplified for points with unfavorable scanning conditions (large θ in Figure 4). This is because the projective distance will always overestimate the true smallest distance to the sensed surface. For the case shown in Figure 4 the projective distance s' for the point p can be computed as a function of the true signed distance s as:

$$s' = s/cos(\theta) \tag{3}$$

The error in Kinect depth measurement leads to an error in the signed distance function that can, for the example shown in Figure 4, also be described by a Gaussian model. However, the standard deviation is amplified and can be computed as:

$$\sigma_{SDF} = \sigma_d / cos(\theta) \tag{4}$$

Hence, the larger the angle between the surface normal and the sensor-to-TSDF-point ray the larger is the variance in the projective signed distance. Since the Kinect and the Asus both have a horizontal field of view (FOV) of around 60°, the largest value for θ for the case shown in Figure 4 is $60/2 = 30°$. In this case the standard deviation according to Equation 4 increases by 15%. For surfaces that are not perpendicular to the sensor's z-axis (Z_k in Figure 4) even higher amplifications can occur.

Surface points lying on the fringe of a large KinFu scan are worse off in terms of surface noise compared to other points. They are affected more by the variance amplification explained above. Moreover, they are sensed by relatively few frames and thus will not benefit as much from the TSDF averaging. This explains the visible increase in surface noise seen in the right part of the scans in Figure 6.

It is important to note that these two fundamental KinFu scan distortions, the surface bending and the surface noise, arise particularly due to the largeness of the scans as opposed to the relatively small scenes presented in the KinFu paper [NIH*11]. Hence, they deserve special attention and proper processing to ensure good location retrieval performance.

4. Sensor calibration and KinFu scan pre-processing

Sensor calibration. One of the main issues with our relatively large KinFu query scans is surface bending. Experiment (3) shown in Figure 6 showed that it is principally possible to mitigate the bending issue in the raw data by scanning a scene from close proximity which requires large translations to cover the entire scene. This may not be practical as the close proximity greatly increases the probability of ICP failure. Also, longer scanning times are necessary.

The obvious practical solution is to calibrate the sensor to deliver better raw data that is not bent. A standard camera calibration of the Infrared (IR) camera of Kinect-like sensors can accurately compute the focal length, principal point and radial distortion coefficients. However, these parameters cannot be uploaded onto the device. Since disparity computation happens on the device, these estimated coefficients will not help producing more accurate disparity maps. Indeed, our IR camera calibration did not deliver the desired improvements and the produced KinFu scans remained bent.

Teichman et al. [TMT13] investigated 3D sensor calibration of Kinect-like sensors and showed that such devices are essentially *myopic* in terms of their distortion characteristics. Like us, they observed that depth images (and their derived point clouds) exhibit a bending that increases with distance. As a solution they propose using mutliplicative depth compensation factors that are learned differently for different pixel regions at various discrete depth levels [TMT13].

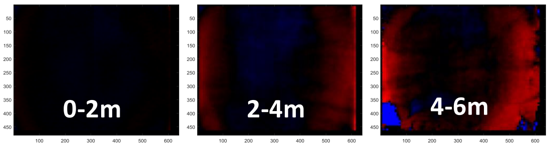


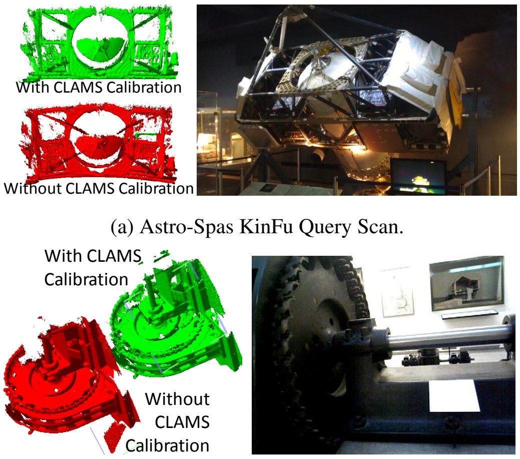
Figure 8: Learned depth multiplier images using CLAMS for three depth levels. Red regions imply multiplicative factors that lead to depth value decrease as opposed to blue regions. The color intensity is directly related to the amount of applied correction. Clearly, at higher depth larger corrections are needed to compensate the depth errors. Also, more compensation is necessary when deviating away from the principal point to offset the bending.

Teichman et al.'s learning technique essentially runs simultaneous localization and mapping (SLAM) on the RGBD data of a Kinect-like sensor. The sensor trajectory is estimated. This is used to build a 3D model of the scene, however, only using reliable depth data (depth < 2m). Finally, all depth data from all pixels of each frame are used to compute the depth error at different pixel regions and different depth levels to compute the multiplicative factors that would compensate these errors.

We used the CLAMS technique to calibrate our Xtion RGBD sensor. The learned *depth multiplier images* are shown in Figure 8. Applying the learned model on the raw 3D sensor data leads to visible improvements as shown in Figure 5b and the bending is largely removed. Using the undistorted depth images KinFu can produce scans without bending artifacts as shown in Figure 9a. If the scan is relatively small, however, and scanned from a close distance no visible improvement can be observed as seen in Figure 9b.

KinFu surface pre-processing. Having obtained unbent KinFu scans we address the remaining issues highlighted in Section 3. First, we reduce surface noise using a moving least squares (MLS) filter. Spurious points and remaining noise that does not fit with the local surface point statistics are treated using a sparse outlier removal (SOR) filter.

The MLS filter is a projection-based procedure that approximates surfaces locally by polynomial functions [ABCO*03]. For a surface point $\mathbf{s}$ a local reference domain must first be defined. For that the plane $H = \left\{ \mathbf{x} \mid \langle \mathbf{n},\mathbf{x} \rangle - D = 0, \mathbf{x} \in \mathbb{R}^3 \right\}, \mathbf{n} \in \mathbb{R}^3, \|\mathbf{n}\| = 1$ minimizing the sum of weighted squared distances of points $\mathbf{p}_i$, $\forall i \in \mathcal{N}$ is computed. $\mathcal{N}$ is the set of points in the neighborhood of point $\mathbf{s}$. Point $\mathbf{s}$'s projection onto H forms the origin of the reference domain. The computed reference domain and its origin $\mathbf{q}$ are used to compute a bivariate



(a) Astro-Spas KinFu Query Scan.

(b) Generator3 KinFu Query Scan.

Figure 9: Two used query scans generated once with prior CLAMS sensor calibration and once without. Especially large scans benefit from the calibration and do not exhibit the bending artifact.

polynomial approximation $g(x,y)$ of the surface. The value $g(0,0)$ is used to compute the filtered point value.

To deal with the shadow surface problem as well as spurious points and remaining noise an SOR filter [RMB*08] is used. The SOR filter is a method based on point statistics. For each point, the average distance to its k-nearest neighbors is computed. The individual averages are used to compute the global mean μ and the standard deviation in the average distance σ. A threshold is defined:

$$t = \mu + \sigma \cdot m \tag{5}$$

where m is a factor used to relax the threshold. Points that have an average k-nearest neighbors distance lower than t will be considered as outliers and removed.

The combined effect of MLS and SOR filtering are smoother surfaces as shown in Figure 7, allowing a better normal estimation. Once filtering is finished, we estimate surface normals and disambiguate them to a consistent orientation that agrees with that of the respective part in the target point cloud.

5. Evaluation

The used query scans and target clouds are introduced in Section 5.1 followed by an explanation of the evaluation metrics in Section 5.2. The used retrieval parameters are mentioned in Section 5.3 followed by the obtained results in Section 5.4 and concluded by an analysis of the results in Section 5.5.

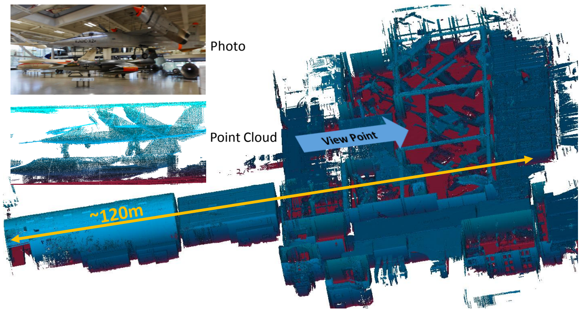


Figure 10: One of the used target clouds including the airplanes exhibition. A photo and the corresponding part in the point cloud are shown from the indicated view point.

5.1. Kinfu queries and reference point clouds

We recorded a set of 9 queries in the *Deutsches Museum* (DM), the world's biggest technical museum, with an Xtion Pro Live. The Xtion is chosen over the Kinect because it can be powered via USB. The same set of depth images is fed to KinFu once undistorted with the learned CLAMS model and once without distortion compensation. Once the query scans have been generated they are pre-processed as explained in Section 4. The ground truth transform aligning queries to their respective target clouds has been manually established using Meshlab [CCR08].

The target clouds have been recorded using the indoor mapping trolley from Huitl et al. [HSH*12]. The target clouds cover up to 3500m^2 of floor space and encompassing multiple exhibition areas. Figure 10 shows the target cloud for the queries GF200-Plane, Generator1 and Generator3.

5.2. Evaluation metrics

While retrieving any query we measure the true correspondence rate (TCR). This is the fraction of correspondences that adhere to the ground transformation $\mathbf{T}$. Also, the finally computed transformation using RANSAC and ICP is checked for correctness. This process is repeated 100 times to give reliable results as RANSAC is random. The percentage of successful retrievals from within the 100 runs defines the precision of retrieval P. For each successful retrieval we measure the accuracy of retrieval. For that we first compute the error transformation

$$\mathbf{T}_e = \begin{pmatrix} \mathbf{R}_e & \mathbf{t}_e \\ 0\,0\,0 & 1 \end{pmatrix} = \mathbf{T}^{-1}\hat{\mathbf{T}}. \tag{6}$$

The accuracy in the angle A_ϕ is obtained by computing the axis-angle representation of the rotation matrix $\mathbf{R}_e$. The translation error A_t is obtained using the query's centroid $\mathbf{c}$ as follows:

$$A_t = ||\mathbf{R}_e\mathbf{c} + \mathbf{t}_e||_2 \tag{7}$$

The accuracy values are averaged over all successful runs of the respective query.

5.3. System parameters

The keypoint and descriptor radii have been tuned to 10 cm and 1 m, respectively. The order of the polynomial for the MLS filter is 4. The MLS search radius is 5 times the mesh resolution. For the SOR filter, we use $k = 60$ neighbors for the statistics and a threshold multiplier $m = 1.0$.

5.4. Evaluation results

Table 1: Retrieval results using the evaluation metrics introduced in Section 5.2. We show the true correspondence rate (TCR[%]) and the retrieval precision (P[%]) for two cases: No CLAMS calibration (TCR_{no} and P_{no}); With CLAMS calibration (TCR_{clams} and P_{clams}). The retrieval accuracy (A_ϕ[°] and A_t[cm]) is shown for the case with CLAMS.

Query	TCR_{no}	TCR_{clams}	P_{no}	P_{clams}	A_t	A_ϕ
FrancisTurbine	25.0	31.9	100	100	3	0.8
GirardTurbine	14.3	20.8	86	100	6	1.9
Astro-Spas	27.0	49.4	100	100	0	0.0
GF200-Plane	13.6	12.9	78	96	9	2.1
SteamLocomotive	7.5	14.4	15	92	5	5.7
SteamEngine	20.5	13.8	0	100	5	2.4
Balloon	16.2	23.7	0	70	4	0.5
Generator1	20.2	23.4	100	100	4	1.9
Generator3	7.8	7.4	74	65	17	9.3
Weighted Average	16.9	22.0	61	91	6	2.5

Comparing TCR_{clams} and TCR_{no} in Table 1 it can be seen that the CLAMS calibration leads to an increase in TCR in 6/9 queries. For the remaining three queries the decrease in true correspondence rate is notable only in one query (SteamEngine) while it is less than 1% in the other two. The increase in TCR can reach up to 22.4% and averages 5.1%. Nevertheless, it can be seen that even after pre-processing, the TCR is relatively low averaging 22%.

The increase in TCR is seen to have a large impact on the retrieval precision which rises from 61% to reach 91%.

Columns TCR_{clams} and P_{clams} in Table 1 show that a true correspondence rate as low as 12.9% is sometimes enough to lead to a 96% precision (GF200-Plane).

Columns A_ϕ and A_t of Table 1 show that for all successful retrievals, the average error in the retrieved orientation is 2.5°and the average location accuracy is 6cm.

5.5. Analysis

The results in Section 5.4 show that proper calibration for our relatively large KinFu scans, as opposed to simple IR camera calibration, leads to a significant improvement in retrieval results. Especially large query scans such as FrancisTurbine, GirardTurbine, Astro-Spas and SteamLocomotive benefit greatly from the calibration either in terms of true correspondence rate (TCR) or precision or both.

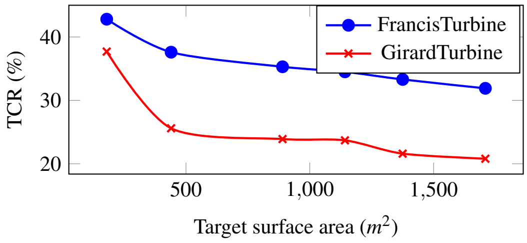


Figure 11: The true correspondence rate (TCR) of GirardTurbine reduces substantially with increasing target cloud floor size as opposed to FrancisTurbine proving that the query itself is less distinctive.

The precision values for the queries SteamEngine and Balloon rise, as can be seen in Table 1, from 0% to 100% after CLAMS calibration. While this is easily justifiable in the case of Balloon through the increase in TCR, it seems counter intuitive in the case of SteamEngine whose TCR decreases after CLAMS calibration. A deeper inspection shows that while the TCR decreases, the actual absolute number of true correspondences increases by 33%. In fact, the absolute number of true correspondences, not shown in Table 1, increases for all nine queries after CLAMS calibration. This increase is effectively exploited by our RANSAC implementation which includes a built-in false correspondence rejector that will be explained later.

The TCR of the GirardTurbine is 10% less than that of FrancisTurbine which is located beside it in the museum. We argue that the problem is related to the lack of intrinsic distinctiveness of the shape itself. To prove this we compare the reduction in TCR of both scans as we match each one of them to increasingly larger cutouts of their common target cloud. We argue that a distinctive query exhibits a stable TCR irrespective of the target size. The curves in Figure 11 indeed show a large decrease in the TCR of GirardTurbine as the target cloud increases as opposed to FrancisTurbine whose TCR decreases at a far less rate. The GirardTurbine query scans from multiple matching runs without CLAMS calibration are visualized after alignment in red in Figure 12. It can be seen that occasionally the query gets matched to the nearby turbines. This problem is not observed in the case of CLAMS calibration. All 100 retrieval attempts succeed in that case.

The results in Table 1 show that the average true correspondence rate is generally low even after CLAMS calibration. This is mainly due to the fact that the queries make up a tiny fraction of the large-scale target clouds. Despite the low true correspondence rates, the final retrieval is very precise on average. This is a testimony to the robustness of RANSAC and the used parameters. One fundamental feature of the RANSAC we implemented is a built-in false correspondence rejector. The rejector validates any sampled correspondence with already pre-sampled correspondences in the same iteration. The validation is achieved by checking

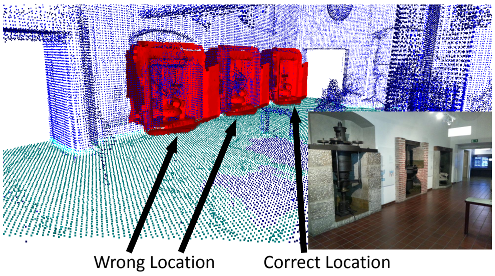


Figure 12: The results of multiple retrievals of the Girard-Turbine query (red) are shown overlapped on the same target cloud (blue). Some retrieval attempts match the query to neighboring turbines displaying the issue of distinctiveness.

whether the spatial distances to the other samples on the query side are preserved on the target side, exploiting a fundamental property of the Special Euclidean Group $\mathbb{SE}3$. This helps to exclude wrong correspondences effectively and focus the relatively limited number of iterations on correspondences with a high likelihood of being correct. Moreover, for a completely invalid triplet to be used, the three sampled correspondences have to all adhere to the same wrong transformation. The probability of such a case is very low.

6. Conclusions

We pose indoor localization as a part-in-whole shape matching problem of KinFu scans in large-scale point clouds using a 3DOR system with local shape features. We show that calibration of Kinect-like sensors using the CLAMS technique is essential to producing geometrically correct KinFu scans and explain the necessity for surface filtering of the relatively large KinFu scans used in our application. Finally, we evaluate the location retrieval performance using real data captured in a large museum environment with target clouds of up to 3500m^2 floor space achieving an average accuracy of 6cm. Currently, retrieval takes around 15s time. Possible future work could focus on accelerating the retrieval process. The datasets are publicly accessible at: http://www.lmt.ei.tum.de/team/mitarbeiter/anas-al-nuaimi.html#forschung.

7. Acknowledgments

We would like to thank Mrs. Elisabeth Jäckle for providing access to the Deutsches Museum. Our thanks also go out to Nicolas Mellado for his help regarding Super4PCS.

References

[ABCO*03] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., T. Silva C.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (Jan. 2003), 3–15. 5

[AMCO08] Aiger D., Mitra N. J., Cohen-Or D.: 4pointss congruent sets for robust pairwise surface registration. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 85:1–85:10. 2

[AMT*12] Aldoma A., Marton Z.-C., Tombari F., Wohlkinger W., Potthast C., Zeisl B., Rusu R., Gedikli S., Vincze M.: Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robotics & Automation Mag. 19*, 3 (Sept 2012), 80–91. 2, 4

[CCR08] Cignoni P., Corsini M., Ranzuglia G.: Meshlab: an open-source 3d mesh processing system. *ERCIM News*, 73 (April 2008), 45–46. 6

[CL96] Curless B., Levoy M.: A volumetric method for building complex models from range images. In *Proc. of the ACM 23rd Annual Conf. on Computer Graphics and Interactive Techniques* (1996), pp. 303–312. 3

[FA14] Filipe S., Alexandre L. A.: A comparative eval. of 3d keypoint detectors in a rgb-d object dataset. In *9th Intern. Conf. on Computer Vision Theory and Applications* (Jan 2014). 2

[HSH*12] Huitl R., Schroth G., Hilsenbeck S., Schweiger F., Steinbach E.: Tumindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In *IEEE ICIP* (Sept 2012), pp. 1773–1776. 1, 6

[KE12] Khoshelham K., Elberink S. O.: Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors 12*, 2 (2012), 1437–1454. 4

[LCC*10] Liu T., Carlberg M., Chen G., Chen J., Kua J., Zakhor A.: Indoor localization and visualization using a human-operated backpack system. In *Interat. Conf. on Indoor Positioning and Indoor Navigation* (Sept 2010), pp. 1–10. 1

[MAM14] Mellado N., Aiger D., Mitra N. J.: Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum 33*, 5 (2014), 205–215. 2

[MN03] Mitra N. J., Nguyen A.: Estimating surface normals in noisy point cloud data. In *Proceedings of the ACM 19th Annual Symposium on Computational Geometry* (2003), pp. 322–328. 4

[NIH*11] Newcombe R. A., Izadi S., Hilliges O., Molyneaux D., Kim D., Davison A. J., Kohli P., Shotton J., Hodges S., Fitzgibbon A.: Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE 10th International Symposium on Mixed and Augmented Reality* (2011), pp. 127–136. 1, 5

[RC11] Rusu R., Cousins S.: 3d is here: Point cloud library (pcl). In *IEEE 2011 Internat. Conf. on Robotics and Automation (ICRA)* (May 2011), pp. 1–4. 2

[RMB*08] Rusu R. B., Marton Z. C., Blodow N., Dolha M., Beetz M.: Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems 56*, 11 (2008), 927–941. 6

[TMT13] Teichman A., Miller S., Thrun S.: Unsupervised intrinsic calibration of depth sensors via slam. In *Proceedings of Robotics: Science and Systems* (Berlin, June 2013). 2, 3, 5

[TSDS10a] Tombari F., Salti S., Di Stefano L.: Unique shape context for 3d data description. In *ACM Workshop on 3D Object Retrieval* (2010), pp. 57–62. 2

[TSDS10b] Tombari F., Salti S., Di Stefano L.: Unique signatures of histograms for local surface description. In *Proc. of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III* (Berlin, Heidelberg, 2010), Springer-Verlag, pp. 356–369. 2

[TV04] Tangelder J., Veltkamp R.: A survey of content based 3d shape retrieval methods. In *Proc. of 2004 Internat. Conf. on Shape Modeling Applications* (June 2004), pp. 145–156. 2

[Zha94] Zhang Z.: Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision 13*, 2 (Oct. 1994), 119–152. 2, 3

[Zho09] Zhong Y.: Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Proc. of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)* (Sept 2009), pp. 689–696. 2