Aligning Point Cloud Views using Persistent Feature Histograms

Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, Michael Beetz
Intelligent Autonomous Systems, Technische Universität München
{rusu,blodow,marton,beetz}@cs.tum.edu

***Abstract*— In this paper we investigate the usage of persistent point feature histograms for the problem of aligning point cloud data views into a consistent global model. Given a collection of noisy point clouds, our algorithm estimates a set of robust 16D features which describe the geometry of each point locally. By analyzing the persistence of the features at different scales, we extract an optimal set which best characterizes a given point cloud. The resulted persistent features are used in an initial alignment algorithm to estimate a rigid transformation that approximately registers the input datasets. The algorithm provides good starting points for iterative registration algorithms such as ICP (Iterative Closest Point), by transforming the datasets to its convergence basin. We show that our approach is invariant to pose and sampling density, and can cope well with noisy data coming from both indoor and outdoor laser scans.**

I. Introduction

The problem of consistently aligning various 3D point cloud data views into a complete model is known as registration. Its goal is to find the relative positions and orientations of the separately acquired views in a global coordinate framework, such that the intersecting areas between them overlap perfectly. One of the most popular registration methods to date is the Iterative Closest Point (ICP) algorithm [1], [2], an iterative descend method which tries to find the optimal transformation between two datasets by minimizing a distance error metric. ICP uses pairs of nearest 3D points in the source and model set as correspondences, and assumes that every point has a corresponding match.

Since input datasets do not always contain complete point-to-point correspondences, due to the fact that they might be only partially overlapping for example, a lot of efforts have been made into the area of feature selection [3], as well as including extra information such as colors [4], intensity values [5], or normals [6] that could improve the correspondence problem. Their values however, are highly sensitive to sensor noise.

Robust feature descriptors such as moment invariants [7], spherical harmonic invariants [8], and integral volume descriptors [9] have been proposed as point features and used for registering partial scans of a model [3], [9]. All of them are invariant to translation and 3D rotations, but are still sensitive to noise.

Because ICP requires an exhaustive search through the correspondence space, several variants that address the problem of its computational complexity have been proposed [10], [11]. Including additional semantic information (e.g. walls, floors, ceilings), as presented in [12] seems to decrease the computation time by up to 30%. To improve the chances of finding the global minimum, and thus improving ICP's narrow convergence basin, several alternative optimization methods have been proposed [13], [14], though they require that the individual point cloud data views are already roughly aligned with each other.

Our work is motivated by finding correct point-to-point correspondences in real-world noisy data scans, and computing rigid transformations that roughly align them using geometric constraints. This must serve as a good initial guess for fine tuning the registration using algorithms such as ICP. Our algorithm is based upon the previously mentioned work, but adds several extensions. In the following we will discuss our reasons and justify the need for these extensions.

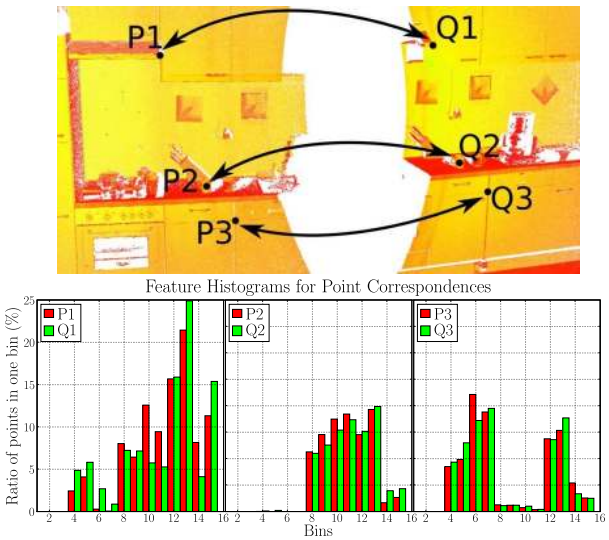

Fig. 1. Feature Histograms for corresponding points on different point cloud datasets.

While feature candidates such as estimated surface curvatures [6] or integral volume descriptors [9] have already been used as features for finding better matches in the point-to-point correspondence process, these only represent their point neighborhoods partially, and are dependent on noise levels or point cloud density most of the times. We propose the use of a better system that combines several aspects of the geometry of a point's neighborhood [15], [16] for estimating a *multi-value* feature set (see Figure 1). We make an in-depth analysis of the points' histogram signatures for different geometric primitives (i.e. plane, sphere, cylinder, and edge), and reduce the theoretical computational complexity of the algorithm by a factor of approximately 2.

In general it is not clear how one should select the optimal k support for a point when computing any of the above mentioned features. If the data is highly contaminated with noise, selecting a small k will lead to large errors in the feature estimation. However, if k is too big, small details will be suppressed. Recently, work has been done on automatically computing good k values (i.e. *scale*) for normal estimation on 3D point cloud data [17], [18] as well as principal curvatures [19], [20], [21] on multiple scales. Unfortunately, the above mentioned methods for computing an optimal scale require additional thresholds, such as $d1$ and $d2$ which are determined empirically in [17], and estimated using linear least-squares in [18] when knowledge about ground truth normal exists. In [19] the neighborhood is grown incrementally until a jump occurs in the variation-scale curve, but the method cannot be successfully applied to noisy point clouds, as the variations in the surface curvature are not modified smoothly with k. The selection of the T_c threshold in [20] is not intuitive, and the authors do not explain properly if the resulted persistent features are obtained using solely the intersection of features computed over different radii. The statistical estimation of curvatures in [21] uses a M-estimation framework to reject noise and outliers in the data and samples normal variations in an adaptively reweighted neighborhood, but it is unfortunately slow for large datasets, requiring approximately 20 minutes for about 10^6 points. To select those features which best characterize the input dataset, we perform a statistical feature analysis over multiple scales and select a set of unique features present in more of them. The persistence of a feature point is analyzed by discretely sampling over an interval of sphere radii. We statistically analyze different distance metrics between each point's histogram signature and the mean histogram of the cloud (μ-*histogram*), and select the points outside the $\mu \pm \alpha \cdot \sigma$ interval as unique features (see Section III).

A system able to compute a rough initial guess for the ICP algorithm, using Extended Gaussian Images (EGI) and the rotational Fourier transform is presented in [22]. Extended Gaussian Images are useful for representing the shapes of surfaces and can be approximated using the spherical histogram of surface orientations [23]. While this can provide good transformations which might align two datasets closely for watertight objects, it does not produce satisfactory results for our environmental models, as the normals alone do not provide enough information about the geometry of the cloud. Our work is based on the idea that relationships between persistent features in a scene can lead to potentially better alignment candidates. Therefore we assemble pairs of histogram-features in the first point cloud and check for corresponding pairs (points with similar histograms and distances) in the second one, similar to [9].

Our key contribution is the application of a feature language which describes important points in a dataset with the advantage that these features are:

- persistent, i.e. similar for corresponding points in different scans (acquired from different scanning poses and with different densities), and
- expressive enough (because of their higher dimensionality) to produce correct correspondences.

Fig. 2. Aligning 2 scans of a kitchen using persistent feature histograms: corresponding points (top), and computed initial alignment (middle). The final result after registration for a sequence of 5 scans is shown at the bottom.

Together with a geometrical reasoning method, these form the basis of a good initial alignment for registration.

As it can be seem in Figure 2, using the identified point correspondences (based on our persistent point feature histograms) the initial alignment is able to produce an almost perfect alignment for the two kitchen scans.

The remainder of this paper is organized as follows. The next section (Section II) introduces the point feature histograms and presents the implementation of our algorithm for computing them. Section III describes the theory behind the feature persistence analysis and gives an example for two different datasets. Section IV explains our registration algorithm using geometrical reasoning for finding good initial alignments. We discuss experimental results in section V and conclude in section VI.

II. Point Feature Histograms

A problem that arises in point-to-point correspondence searches, is that the features usually used (e.g. surface normals and curvature estimates [6], integral volume descriptors [9], etc) do not *fully* represent the underlying surface on

which the point's k-neighborhood is positioned. Moreover, these kinds of features approximate the neighborhood with a single scalar value. As a direct consequence, most scenes will contain many points with the same or very similar feature values, thus reducing their informative characteristics. Even if the feature estimation would be able to cope with noisy datasets, it can still be easily deduced that applications who rely on these one dimensional features will deal with multiple and false correspondences and will be prone to failure.

Ideally, for each point in the dataset we would like to have informative labels as features, such as: point lying on an edge, point lying on a sphere or a plane, and so on. Using such features and sets of geometric constraints between them, the probability of finding the correct correspondences in other datasets would increase.

In order to efficiently obtain such informative features, we propose the computation and usage of a histogram of values [16], [15] which encodes the neighborhood's geometrical properties much better, and provides an overall scale and pose invariant multi-value feature. The histogram generalizes the mean surface curvature at a point p, and is computed as follows:

- for each point p, all of p's neighbors enclosed in the sphere with a given radius r are selected (k-neighborhood);
- if the surface normal at point p is missing, approximate it by the normal of the best fit plane to the neighborhood surface – usually performed using Principal Component Analysis:
$$n_{p_i} = V_0,\ \lambda_0 \leq \lambda_1 \leq \lambda_2$$
where V_0 is the eigenvector corresponding to the smallest eigenvalue λ_0;
- once all normals are obtained, use the existing viewpoint information to re-orient them all consistently[1]:
$$\text{if } \frac{\langle v - p_i,\ n_{p_i}\rangle}{\|v - p_i\|} < 0, \text{ then } n_{p_i} = -n_{p_i} \qquad (1)$$
where n_{p_i} is the normal of the point p_i and, v is the viewpoint;
- for every pair of points p_i and p_j ($i \neq j,\ j < i$) in the k-neighborhood of p, and their estimated normals n_i and n_j, we select a source p_s and target p_t point – the source being the one having the smaller angle between the associated normal and the line connecting the points:
$$\begin{aligned}&\text{if } \langle n_i,\ p_j - p_i\rangle \leq \langle n_j,\ p_i - p_j\rangle\\ &\text{then } p_s = p_i, p_t = p_j\\ &\text{else } p_s = p_j, p_t = p_i\end{aligned}$$
and then define the Darboux frame (see Figure 3) with the origin in the source point as:
$$u = n_s,\ v = (p_t - p_s) \times u,\ w = u \times v. \qquad (2)$$

[1]see [24] for a general algorithm for consistent normal orientation propagation for 3D objects

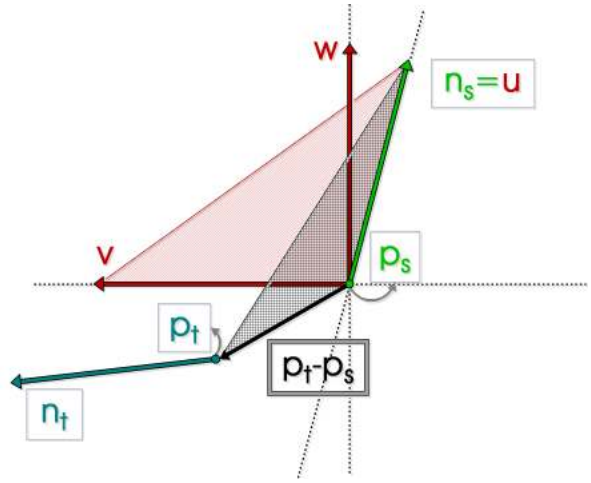

Fig. 3. The computed Darboux frame (vectors u, v and w) placed at the source point.

In the k-neighborhood around point p, for each pair of points p_i and p_j a source is uniquely defined. Thus the computational complexity for each point changes from $O(k^2)$ to $O(k\cdot(k-1)/2)$, because of the above mentioned restrictions (i.e. $i \neq j$ and $j < i$).

The four features are categorized using a 16-bin histogram, where each bin at index idx contains the *percentage* of the source points in the neighborhood which have their features in the interval defined by idx:

$$\left.\begin{aligned}f_1 &= \langle v,\ n_t\rangle\\ f_2 &= ||p_t - p_s||\\ f_3 &= \langle u,\ p_t - p_s\rangle / f_2\\ f_4 &= atan(\langle w,\ n_t\rangle, \langle u,\ n_t\rangle)\end{aligned}\right\} idx = \sum_{i=1}^{i\leq 4} step(s_i, f_i)\cdot 2^{i-1} \qquad (3)$$

where $step(s, f)$ is defined as 0 if $f < s$ and 1 otherwise. This means that by setting s_i to the center of the definition interval of f_i (i.e. 0 for features f_1, f_3, f_4 and r for f_2) the algorithm classifies each feature of a $\{p_i,p_j\}$ pair in p's vicinity in two categories, and saves the percentage of pairs which have the same category for all four features.

The four features are a measure of the angles between the points' normals and the distance vector between them. Because f_1 and f_3 are dot products between normalized vectors, they are in fact the cosine of the angles between the 3D vectors, thus their value is between ± 1. Similarly, f_4 is the arctangent of the angle that n_t forms with w if projected on the plane defined by $u = n_t$ and w, so its value is between $\pm\pi/2$.

The number of histogram bins that can be formed using these four geometric features is div^4, where div is the number of subdivisions of the features' value range. In our implementation, by dividing the feature values in two parts (f_i smaller or greater than s_i), we obtain a total of $2^4 = 16$ bins as the total number of combinations between the four features. Because the number of bins increases exponentially

by the power of 4, using more than two subdivisions would result in a large number of extra dimensions for each point (e.g. $3^4 = 81$D), which makes the computational problem intractable.

Figure 4 illustrates the differences using our proposed 16D feature set between query points located on various geometric surfaces. The surfaces were synthetically generated to have similar scales, densities, and noise levels as our input real-world datasets. For each of the mentioned surfaces, a point was selected such that it lies: (i) on the middle of an edge of a cube, (ii) on the lateral surface of a cylinder at half the height, (iii) on a sphere, and (iv) on a plane. The 16D feature histogram was generated using all its neighbors inside a sphere with radius $r = 2cm$. The results show that the different geometrical properties of each surface produce unique signatures in the feature histograms space.

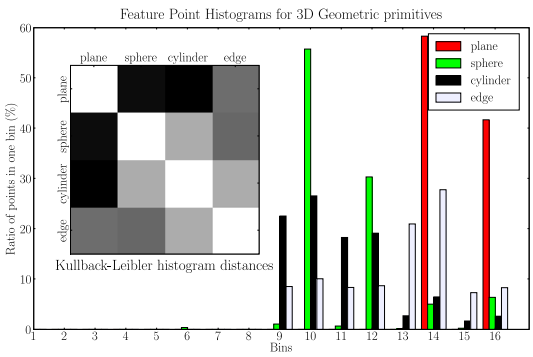

Fig. 4. Feature Histograms for query points located on different geometric surfaces with the color coded Kullback-Leiber distances between them.

Because of their properties, point feature histograms are promising to be more suitable candidates for problems like correspondence search while registering multiple scans under the same model (see Figures 1, 2, 11).

The next section introduces the persistence feature analysis over multiple scales for a each point feature histogram in the dataset.

III. Persistence Analysis

When using point features as characteristics of the entire point cloud, it's good to make sure that we find a compact subset of points P_f that best represents the point cloud. The lesser the number of feature points and the better they characterize the data, the more efficient are the subsequent interpretation process. However, choosing the subset P_f is not easy, as it relies on a double dependency: both the number of neighbors k and the point cloud density φ.

Our feature persistence analysis computes the subset of points P_f, that minimizes the number of points considered for further analysis from the input data set. Corresponding points in different point cloud views of a scene will be likely to be found as persistent features in both scans, which helps in registration but also for segmenting similar points into regions.

In order to select the best feature points for a given cloud, we analyze the neighborhood of each point p multiple times, by enclosing the point on a sphere S with radius r_i and the point p as its center. We vary r over an interval depending on the point cloud size and density, and compute the local point feature histograms for every point. We then select all the points in the cloud, and compute the mean of the feature distribution (μ-*histogram*). By comparing the feature histogram of each point against the μ-*histogram* using a distance metric (see below), and building a distribution of distances (see Figure 8 – note that it can be approximated with a Gaussian distribution) , we can perform a statistical analysis of each feature's persistence over multiple radii. More specifically, we select the set of points (P_{f_i}) whose feature distances are outside the interval $\mu \pm \alpha \cdot \sigma$, as unique features. We do this for every r and at the end, select the unique features which are persistent in both r_i and r_{i+1}, that is:

$$P_f = \bigcup_{i=1}^{n-1} [P_{f_i} \cap P_{f_{i+1}}] \tag{4}$$

As most of the points in the kitchen scene were located on planes, their histograms were similar (and closer to the mean shown in Figure 7) thus producing a sharp peak in the distance distribution. This not that accentuated in the outdoor scans, where the number of planar regions is significantly lower.

For our examples we fixed the value of α to 1, as only around $10-20\%$ of the points are outside the $\mu \pm \sigma$ interval, thus selecting them as unique in the respective radius.

For matching the point feature histograms with the μ-*histogram* of the cloud, we have performed an in-depth analysis using various distance metrics from literature, similar to [15], [25]. Our results confirmed the findings in [25], where the Kullback-Leibler distance (divergence) gave good results for computing differences between the histograms. Its formula is given below:

$$\text{KL divergence} = \sum_{i=1}^{16} (p_i^f - \mu_i) \cdot ln \frac{p_i^f}{\mu_i} \tag{5}$$

where the symbols p_i^f and μ_i represent the point feature histogram at bin i and the mean histogram of the entire dataset at bin i respectively.

Figure 5 and 6 present the results of the persistence feature analysis for two different datasets: one indoor kitchen environment (Figure 5) and one outdoor urban scene (Figure 6). From top to bottom and left to right: (i) unique features over multiple radii, color-coded in the same way as the plots; (ii) global persistent features for the entire scene computed using Equation 4; (iii) Kullback-Leibler divergence values between each point histogram and the mean μ-*histogram* of the cloud; and (iv) the remaining persistent features after selection.

The values of the r_i radii set are selected based on dimensionality of the features that need to be detected. As the point clouds are obtained directly from laser sensors by

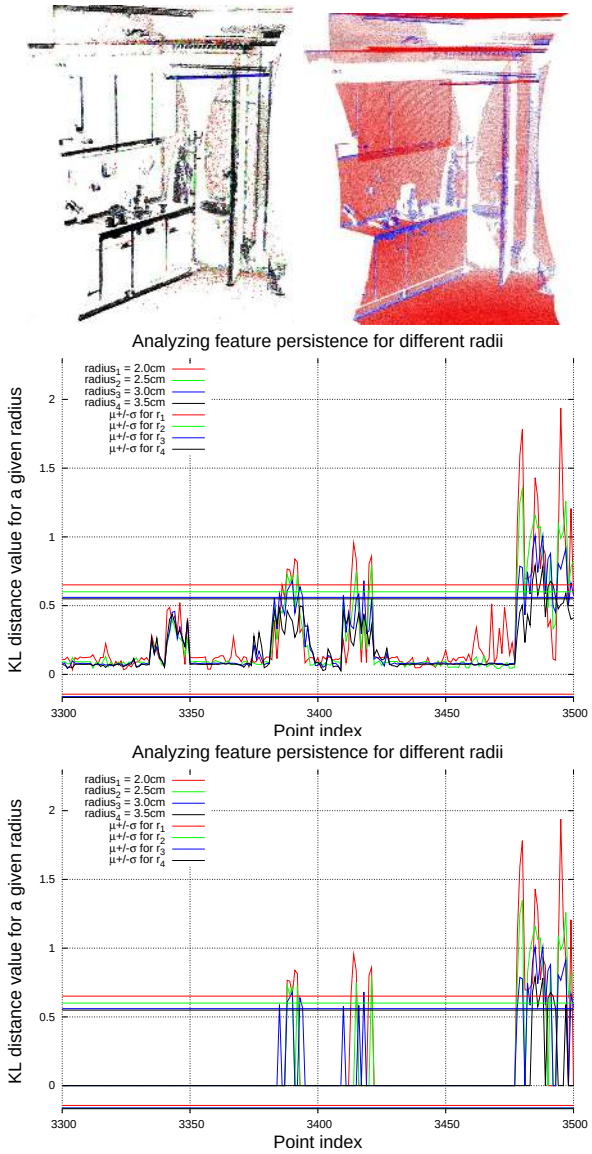

Fig. 5. Analyzing the uniqueness and persistence of feature histograms for an indoor kitchen scene.

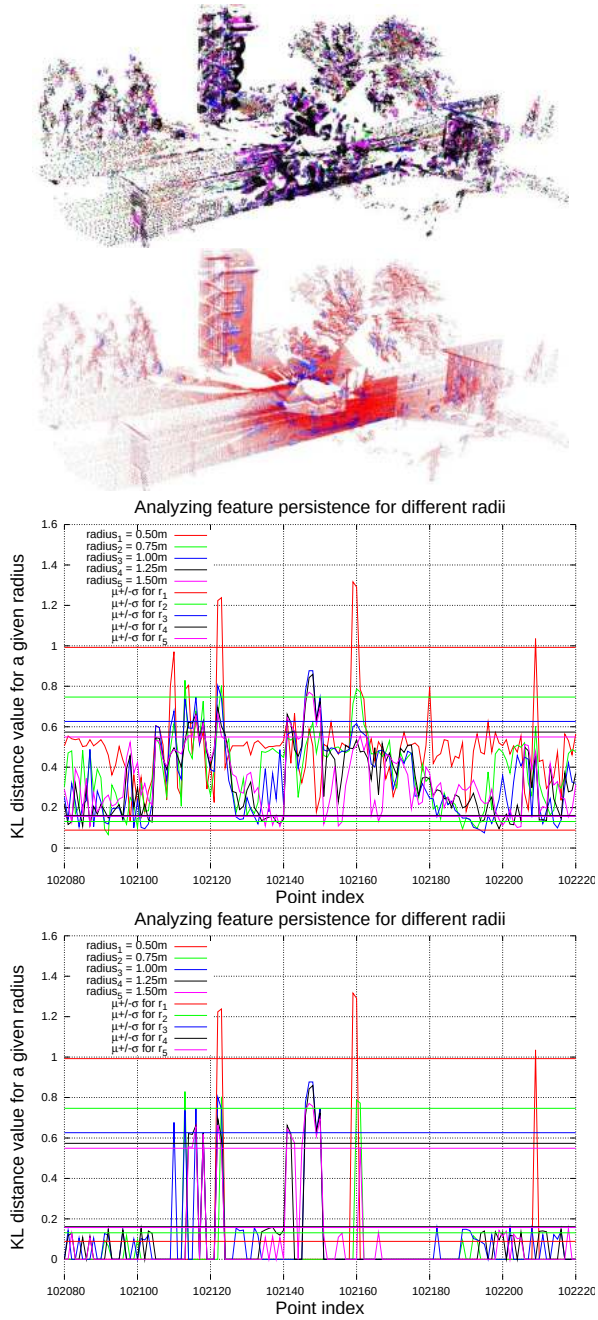

Fig. 6. Analyzing the uniqueness and persistence of feature histograms for an outdoor urban scene.

scanning the environment, the scale is known so the radii $r_{min,max}$ of a sphere can be chosen intuitively.

Figure 7 presents the the mean μ-*histogram* of the kitchen dataset for each separate radius. By comparing the results with the ones presented in Figure 4, we determined that the given input scan is mostly composed out of planar regions, as its most frequent geometric primitive.

IV. Estimating Good Alignments

The registration problem becomes easily solvable if the point to point correspondences are perfectly known in both datasets. However, that is not the case in registration problems where no artificial markers are used, so it is essential to robustly estimate good correspondences. Thus, the first step of our registration module is to compute a good initial alignment, and return a set of m correspondences that can be used to directly transform the source point cloud into the convergence area of the target. The method is similar to [9], but instead of using Integral Volume Descriptors to identify interesting feature points, we use our 16D feature histograms which yield better results for our datasets, as show in Figure 9. The computed correspondences must satisfy a geometric contraint based on the intrinsic, rotation and position invariant properties of the point cloud. Point to point distances within the point cloud are a good candidate since they are easy to compute and fulfill this requirement, so we identify a set of points p_i that have the same distances to each other than their corresponding points q_i.

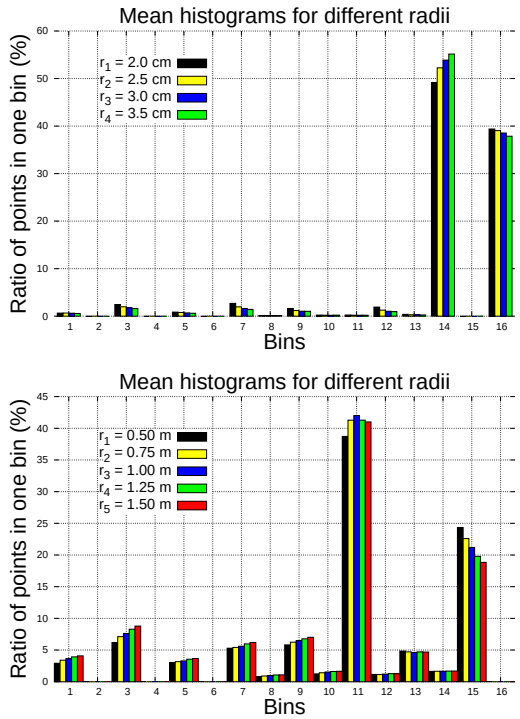

Fig. 7. Mean feature histograms over different radii for the kitchen (top) and outdoor (bottom) scenes.

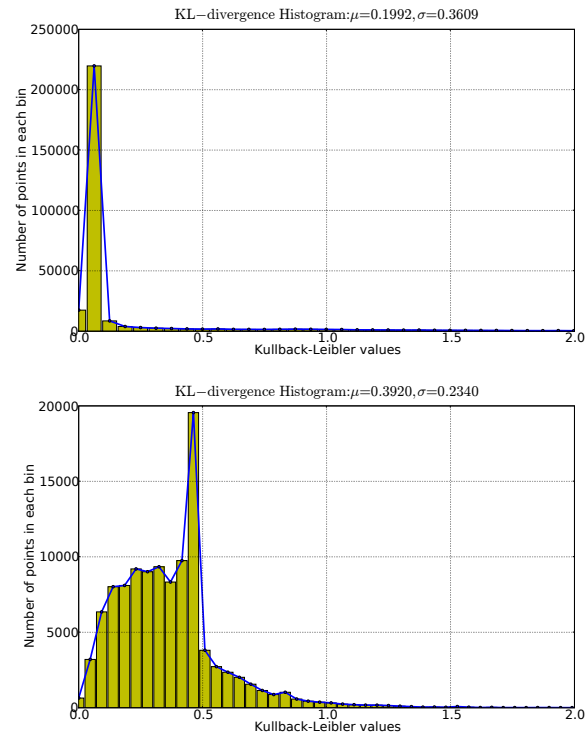

Fig. 8. Distribution of the feature histograms' distance from their mean (μ-*histogram*) using the Kullback-Leibler metric for the indoor kitchen (top) and outdoor (bottom) scan (see Figures 5 and 6). The histograms were computed for $r = 0.3cm$ and $r = 1m$ respectively.

For computing distances in a 16D space, we construct a kD-tree in the feature histograms space, and for a set of m persistent feature points in the source dataset we perform a k-nearest neighbor search in the target. Hence, we look at the k points with most similar histograms and keep them as correspondence candidates for p_i. We call this set of correspondence candidates:

$$C = \{c_i | c_i = \langle p_i, q_{i1}, q_{i2}, ..., q_{ik} \rangle, 1 \leq i \leq m\} \quad (6)$$

In the second phase of the initial alignment, we hierarchically merge the best entries c_i, c_j from C into a set of 2-point correspondences E_2. For every combination of p_i and p_j, we find the q_{ir} and q_{js} that minimize $||p_i - p_j| - |q_{ir} - q_{js}||$ and add it to E_2, in increasing order.

We then proceed merging pairs of entries $\langle e_{2,i}, e_{2,j} \rangle$ from E_2 if $e_{2,i}$ is the next entry that is not merged yet and $e_{2,j}$ minimizes $dRMS(P, Q)$, where P is the set of all points $p_i \in e_{2,i} \cup e_{2,j}$, and Q the set of all points $q_i \in e_{2,i} \cup e_{2,j}$:

$$dRMS^2(P, Q) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\|p_i - p_j\| - \|q_i, q_j\|)^2, \quad (7)$$

analog to the construction of E_2. We continue merging entries from E_{2^k} into $E_{2^{k+1}}$ in this manner, until there are not enough correspondences left in $E_k (k = k_{max})$ to generate $2k$-point correspondences. The result is a set of $2^{k_{max}}$ points in the source point cloud, each with a corresponding point in the source model such that the point-to-point distances within each of the two sets differ minimally. This $2^{k_{max}}$-point correspondence can be used to construct a transformation (translation and rotation) that aligns these points This transformation is applied to the source point cloud to roughly align it onto the target.

After an initial alignment has been obtained, in order to improve convergence speed, the algorithm uses a different method to estimate the optimal transform than classical ICP. We are using an alternative error function that instead of minimizing the sum of squares of distances between corresponding points $min \sum_{i=1}^{n} dist(p_i, q_i)$, uses an approximate measure of point-to-surface distances [26].

Using the new distance measure, the classical ICP error function

$$\sum_{i=1}^{n} \|R \cdot p_i + T - q_i\|^2 \quad (8)$$

changes to

$$\sum_{i=1}^{n} \|\langle (p_i - q_i), n_{q_i} \rangle\|, \quad (9)$$

where n_{q_i} is the normal to the surface in point q_i.

This means we try to minimize the distance between a point p_i and the surface of its corresponding point q_i, or the distance along the normal of q_i. Specifically, movement of the source point cloud tangentially to the target is not restricted. This means faster convergence when the point

clouds are already close to each other, i.e. in the last iteration steps (see Figure 10).

V. Discussions and Experimental Results

We conducted several experiments to evaluate our registration method. Computing the initial alignment using our feature histograms proved to be far superior to using other features such as Integral Volume Descriptors (IVD) or surface curvature estimates, since it was able to robustly bring the point clouds close to the correct position independent of their original poses (see Figure 9). We could only achieve this using IVD when we considered at least $k = 150$ similar points in the target point cloud for every selected feature point in the source, and even then there were cases where it failed. Also, runtime increases exponentially in k, which also renders the IVD method inferior. We discovered that increasing the number of correspondent candidates (k) above 10 did not improve the results.

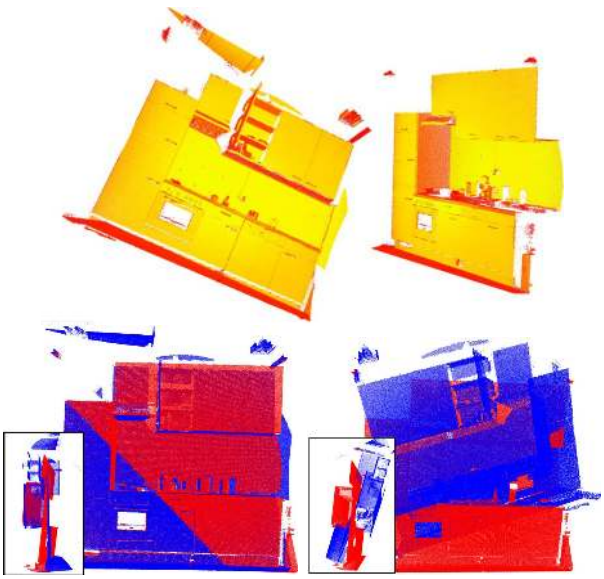

Fig. 9. Two datasets before registration (top). Initial Alignment results using Feature Histograms (bottom left) and using Integral Volume Descriptors (bottom right).

In the second stage of registration, our variant of ICP using instantaneous kinematics, converges faster to the correct solution than regular ICP. This is mostly due to the point-to-surface error metric which doesn't restrict tangential movement, thus accelerating convergence when the point clouds are already close to each other. Figure 10 presents the registration errors for the previous two views of the kitchen dataset, after each iteration step, using the two ICP variants. The error units are degrees for the rotational error and meters for the translational error. Using histogram feature correspondences, Kinematic ICP was able to converge in 1-2 steps as opposed to at least 5 steps in Regular ICP, which also converged to a slightly offset solution.

Figure 11 presents the results of our registration algorithm using the estimated persistent histogram features for 4 outdoor urban point cloud scans. The data was acquired

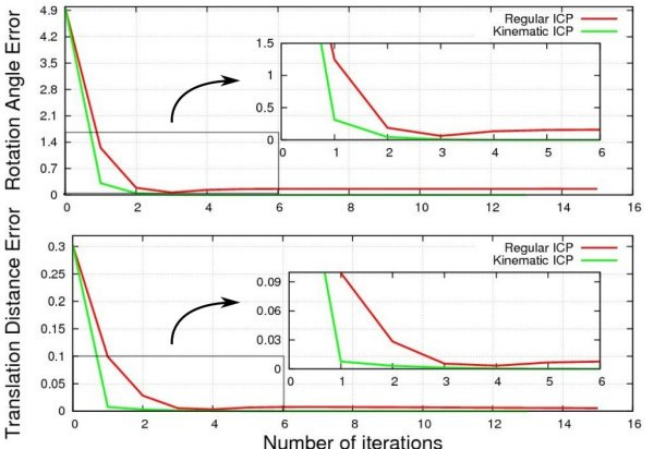

Fig. 10. Comparison of the registration errors of Kinematic vs. Regular ICP during each iteration step for the first two scans of the kitchen scene, both starting after our Initial Alignment.

using a Riegl Z420 3D laser sensor in Ljubljana, Slovenia. The topmost part of the figure, presents a snapshot of the correspondence search for two of the scans using geometric constraints from the initial alignment step. Below, the results of the alignment are shown after applying the best transformation found, as well as the final results after using ICP, followed by a close-up in intensity and curvature domain. The last picture presents the overall registration result for all four scans. The same algorithm was applied for registering 5 scans of an indoor kitchen scene, as shown in Figure 2. The data was acquired using a SICK LMS 400 sensor.

VI. Conclusions and Future Work

We have presented a method for computing persistent point feature histograms for the problem of correspondence search in 3D point clouds. By using a higher dimensionality (16D) for characterizing the local geometry at a point p, the estimated features are robust in the presence of outliers, and invariant to the position, orientation, or sampling density of the cloud. In combination with an initial alignment algorithm based on geometric constraints, partially overlapping datasets can successfully be aligned within the convergence basin of iterative registration methods such as ICP. To show their effectiveness, we have applied the method to real-world datasets coming from indoor and outdoor laser scans.

We plan to investigate the usage of feature histograms in other spaces (e.g. colors) to improve registration in situations where geometry informations does not suffice. Future work will also investigate the possibility of using a non-linear optimizer to obtain a global registration algorithm which could align two datasets in one pass, first by transforming them closely with the initial alignment, and then fine-tuning the registration with an optimizer such as Levenberg-Marquadt. Preliminary results show that this is possible, but further tests still need to be done to test the robustness of the overall approach.

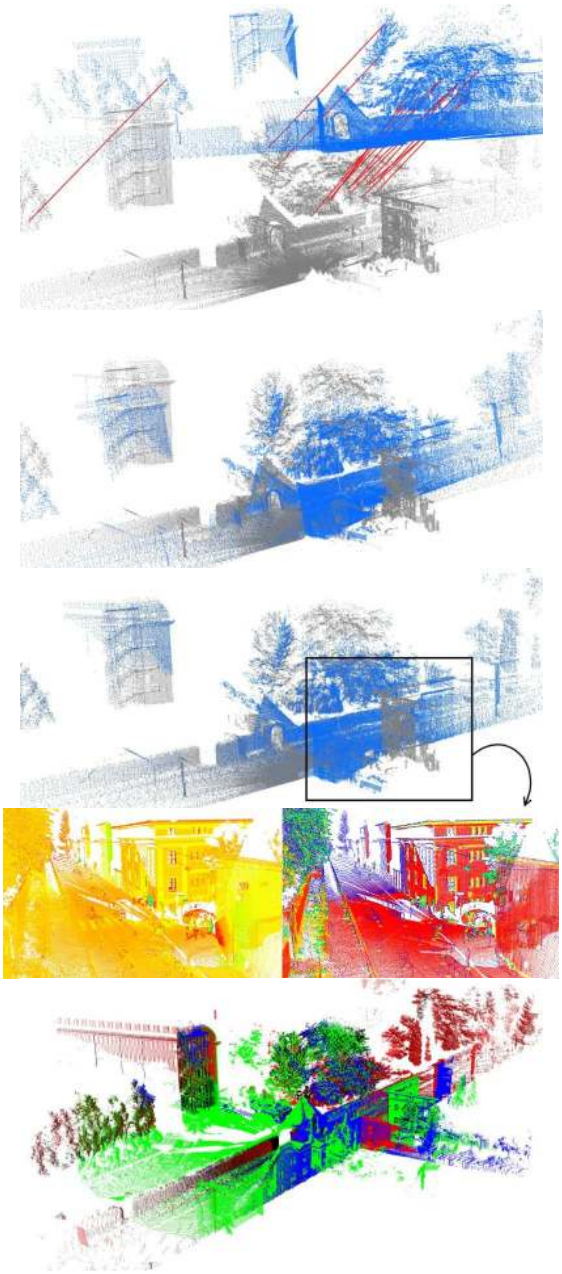

Fig. 11. From top to bottom: point-to-point correspondences using feature histograms, results after initial alignment, results after registration with ICP, closeups in intensity and curvature spaces, and registration of all 4 scans.

ACKNOWLEDGEMENTS

This work is supported by the CoTeSys (Cognition for Technical Systems) cluster of excellence. We would like to thank the people at ISPRS for giving us access to the Ljubljana urban datasets.

REFERENCES

[1] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, 1992.

[2] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves, Tech. Rep. RR-1658.

[3] G. Sharp, S. Lee, and D. Wehe, "ICP registration using invariant features," 2002.

[4] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3-D data," in *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, p. 234.

[5] D. Akca, "Matching of 3D surfaces and their intensities," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 2, pp. 112–121, June 2007.

[6] K.-H. Bae and D. D. Lichti, "Automated registration of unorganized point clouds from terrestrial laser scanners," in *International Archives of Photogrammetry and Remote Sensing (IAPRS)*, 2004, pp. 222–227.

[7] F. Sadjadi and E. Hall, "Three-Dimensional Moment Invariants," *PAMI*, vol. 2, no. 2, pp. 127–136, 1980.

[8] G. Burel and H. Hénocq, "Three-dimensional invariants and their application to object recognition," *Signal Process.*, vol. 45, no. 1, pp. 1–22, 1995.

[9] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust Global Registration," in *Proc. Symp. Geom. Processing*, 2005.

[10] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, 2001.

[11] A. Nüchter, K. Lingemann, and J. Hertzberg, "Cached k-d tree search for ICP algorithms," in *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, 2007, pp. 419–426.

[12] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann, "3D Mapping with Semantic Knowledge," in *RoboCup*, 2005, pp. 335–346.

[13] A. Gruen and D. Akca, "Least squares 3D surface and curve matching," *International Journal of Photogrammetry and Remote Sensing*, vol. 59, pp. 151–174, May 2005.

[14] A. W. Fitzgibbon, "Robust Registration of 2D and 3D Point Sets," in *Proceedings of the British Machine Vision Conference*, 2001, pp. 662–670.

[15] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification," in *3DIM03*, 2003, pp. 474–481.

[16] R. B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz, "Towards 3D Object Maps for Autonomous Household Robots," in *Proceedings of the 20th IEEE International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, Oct 29 - 2 Nov.*, 2007.

[17] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," in *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, 2003, pp. 322–328.

[18] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert, "Scale Selection for Classification of Point-sampled 3-D Surfaces," in *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, June 2005, pp. 285 – 292.

[19] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," pp. 281–289, 2003.

[20] Y.-L. Yang, Y.-K. Lai, S.-M. Hu, and H. Pottmann, "Robust principal curvatures on multiple scales," in *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006, pp. 223–226.

[21] E. Kalogerakis, P. Simari, D. Nowrouzezahrai, and K. Singh, "Robust statistical estimation of curvature on discretized surfaces," in *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, 2007, pp. 13–22.

[22] A. Makadia, A. I. Patterson, and K. Daniilidis, "Fully Automatic Registration of 3D Point Clouds," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1297–1304.

[23] B. K. P. Horn, "Extended Gaussian Images," *Proceedings of the IEEE*, vol. 72, pp. 1671–1686, 1984.

[24] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 1992, pp. 71–78.

[25] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D Object Recognition from Range Images using Local Feature Histograms," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01)*, vol. 2, 2001, pp. 394–399.

[26] H. Pottmann, S. Leopoldseder, and M. Hofer, "Registration without ICP," *Computer Vision and Image Understanding*, vol. 95, no. 1, pp. 54–71, 2004.