# Registration of 3D Point Clouds using Mean Shift Clustering on Rotations and Translations

Ido Haim Ferencz
University of Haifa
Israel
idohferencz@gmail.com

Ilan Shimshoni
University of Haifa
Israel
ishimshoni@mis.haifa.ac.il

## Abstract

*In this paper a novel registration algorithm between 3D point clouds is presented. It exploits the fact that current 3D point descriptors (e.g., RoPS) are accompanied by local reference frames(LRF). LRFs of corresponding points are used to estimate the relative rotation between the point clouds. Thus, inlier matches will generate a cluster of rotation matrices. The size and shape of this cluster is unknown. We therefore develop a mean shift clustering algorithm for noisy rotation matrices. It finds the mode of the distribution to estimate the relative rotation. It is then used for estimating the translation vectors from the matched points. Here again mean shift is used for finding the translation component. The algorithm has been tested on different types of sources of 3D data (3D scanner, Lidar, and Structure from Motion(SfM)) of small scanned objects and urban scenes. In all these cases, the algorithm performed well outperforming state of the art algorithms in accuracy and in speed.*

## 1. Introduction

The alignment of point clouds acquired from several viewpoints, locations and sensors [12] is called 3D registration. 3D registration has numerous applications ranging from 3D object categorization and recognition [17, 9], 3D modeling and scene reconstruction [18], robotic perception [12], etc.

The process consists of the following steps: (a) key point detection. i.e., finding distinctive points on the object. (b) Local feature descriptor computation, i.e., describing the area around the key point. (c) Feature matching, i.e., finding similar features between the two point clouds. (d) Point cloud alignment, i.e., finding the rigid transformation between the coordinate systems of the point clouds.

The advances in technology in the last few decades led to intense research and to the development of specialized algorithms for key point detection and for feature descriptors. Several 3D key point detector evaluations and feature descriptor evaluations have been conducted in the literature [22, 8, 2].

These feature descriptors can be divided in to two main categories, Local Reference Frames (LRF) based descriptors and descriptors without LRFs. The first group consists of **Rotational projection statistics (RoPS)** [9], **Signature of Histogram of Orientations (SHOT)** [26], and **Unique Shape Context (USC)**[25] to name a few. The other group of descriptors consists of **Spin image** [14] and **Fast Point Feature Histograms (FPFH)** [21] which only use surface normals. There are methods such as [1] which do not use descriptors at all but rely on matching four co-planer points between the cloud points.

Our algorithm relies on these LRFs created by the descriptors above. An LRF consists of three orthogonal vectors. The first one is the normal to the surface at the key point. The other two lie in the tangent plane. They maybe the maximal and minimal principal directions $T_1$ and $T_2$ (yielding the Darboux frame $(N, T_1, T_2)$). In this case $T_1$ is not uniquely defined and $-T_1$ is also possible. This can result in true corresponding feature points with dissimilar descriptors. This problem was addressed by Guo *et al*. [9] in the RoPS descriptor in which the second and third vectors were defined uniquely. Thus, a much larger number of correct matches can be found. We will therefore be using these descriptors as input to our algorithm.

Most algorithms use putative matches as input to a RANSAC type algorithm which uses only the feature points. Some state of the art methods [6, 19] use variations of the Hough Transform to estimate the transformation. It can be based on recognized objects in the scene or on LRFs as will now be described. In the method we propose here we use in addition the LRFs to help recover the relative pose in the following way. Assuming feature point $\mathbf{P}_i$ matches feature point $\mathbf{P}_j$ then the relative rotation matrix $\mathbf{R}$ can be estimated by

$$\mathbf{R}_{i,j} = LRF_j^T LRF_i.$$

Since the LRF is estimated from the local region around the feature point, it is not estimated very accurately. Still, the rotation matrices generated from the matches should yield a cluster from the correct matches and a random set of rotation matrices from the outlier matches. Once the center of the cluster $\widehat{\mathbf{R}}$ has been found, for all matches belonging to this cluster the translation vector $\mathbf{t}$ can be estimated as

$$\mathbf{t}_{i,j} = \mathbf{P}_j - \widehat{\mathbf{R}}\mathbf{P}_i.$$

Thus, for all inliers a cluster of points in 3D will be formed.

In order to deal with the noisiness of the computed LRFs a high quality clustering algorithm is needed. To that end we present a mean shift algorithm designed for rotation matrices. The algorithm can deal with a large number of incorrect matches as long as the cluster of inliers is detectable.

In the next section the algorithm will be described. In Section 3 it will be compared to other registration algorithms. Comparative experiments performed on several different types of data sets will be presented in Section 4. Finally, conclusions and future work will be discussed in Section 5.

## 2. Proposed method

In this section we describe our method for pairwise registration. The algorithm is based on a mean shift algorithm especially designed to deal with LRFs and will be denoted LRFMS. The input to the algorithm is two sets of keypoints and their corresponding LRFs and descriptors. In this work we used a set of keypoints detected by the Harris3D [23] interest point detector. For each keypoint the RoPS [9] LRF and descriptor is computed. Given a set of best matches, we use Lowe's [15] ratio test value of 0.99 for choosing the subset of matches to be used by the algorithm. This value is very high compared to the value usually used in RANSAC algorithms and yields a large number of putative matches with a very low inlier rate. In our case however, the inlier rate is not that important, all that is needed for the algorithm to succeed is that the cluster of inlier rotation matrices is detectable.

Given a pair of corresponding points, under the assumption that both LRFs were calculated correctly, the rotational difference between the LRF's is the global rotation between the point clouds. Since this assumption does not hold for real world data we use clustering to find the rotation inliers.

When considering the appropriate clustering method, it was shown by [24], that mean shift is an efficient approach for finding the rigid transformations between Manifolds and can be extended to the registration of 3D objects. While the popular clustering method K-means is a parametric approach that requires prior knowledge on the number of clusters and their shape, the mean shift algorithm has no such requirements. It can also deal with the case in which besides the main cluster the rest of the data is arbitrarily distributed as in the case here.

We will therefore review in the next section the general mean shift algorithm [3] and describe how it can adapted for clustering 3D rotation matrices.

### 2.1. Mean shift

Given $n$ data points $\mathbf{x}_1, ...., \mathbf{x}_n$ from an unknown distribution function in the Euclidean Space $\mathbb{R}^d$, the probability distribution function $f(\mathbf{x})$ can be estimated by

$$f(\mathbf{x}) = \frac{c_{k,d}}{n} \sum_{i=1}^{n} k\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h^2}\right), \tag{1}$$

where $k\left(x\right)$ is the kernel profile function, $h$ is the bandwidth and $c_{k,d}$ is a normalization constant chosen to ensure that $K(x)$ defined below integrates to one. $K\left(\mathbf{x}\right)$ satisfies

$$K\left(\mathbf{x}\right) = c_{k,d} k\left(\|\mathbf{x}\|^2\right) > 0 \;\; \|\mathbf{x}\| \leq 1. \tag{2}$$

To find the mode of the density distribution, a gradient ascent of Eq. 1 is computed. We define $G\left(x\right) = -K'\left(x\right)$. By taking the gradient of Eq. 1, the mean shift vector at point $\mathbf{x}$ is defined as

$$M_h(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i G\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h^2}\right)}{\sum_{i=1}^{n} G\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h^2}\right)} - \mathbf{x}. \tag{3}$$

At each iteration a new point $\mathbf{y}_{j+1}$ is computed.

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i G\left(\frac{\|\mathbf{y}_j-\mathbf{x}_i\|^2}{h^2}\right)}{\sum_{i=1}^{n} G\left(\frac{\|\mathbf{y}_j-\mathbf{x}_i\|^2}{h^2}\right)}. \tag{4}$$

This iterative procedure is applied for all points in the data set where at the first iteration $\mathbf{y}_0 = \mathbf{x}_i$.

For every data point, the algorithm converges to the locally densest area. All points for which the algorithm converges to very close points belong to the same cluster.

Extending the mean shift algorithm to a Non-Euclidean space [27, 24], requires a different approach for the computation of the distances between the objects and for computing the weighted average.

Rotation matrices in 3D are a subgroup of $\mathbf{SO}(3)$. They can be represented by unit vectors of length four known as quaternions. Quite a few methods have been suggested for computing the distance between rotation matrices. In [27, 24] one of these methods was used. Here we will use the following efficient method suggested by Huynh [13] which uses their quaternion representation.

$$dist(\mathbf{q}_i, \mathbf{q}_j) = \arccos\left(|\mathbf{q}_i \cdot \mathbf{q}_j|\right), \tag{5}$$

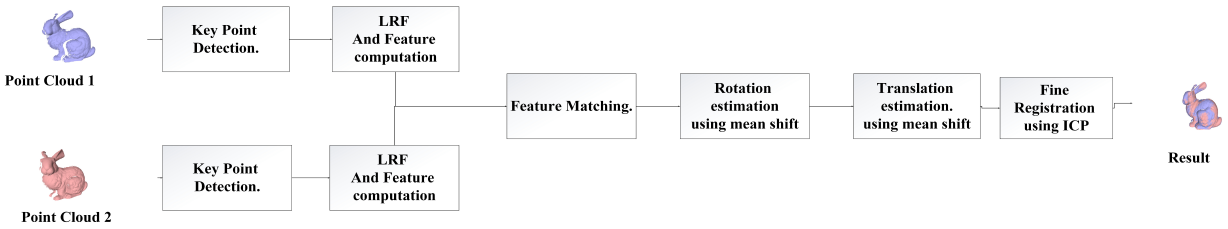



Figure 1. Registration Pipeline: our unique contribution is in the rotation and translation clustering and estimation.

where $\mathbf{q}_i$ and $\mathbf{q}_j$ are the quaternion representations of $\mathbf{R}_i$ and $\mathbf{R}_j$ respectively. The kernel density function is thus defined as

$$f(\mathbf{R}) = \frac{c_{k,d}}{n} \sum_{i=1}^{n} k\left(\frac{(\arccos\left(|\mathbf{q}_i \cdot \mathbf{q}_r|\right))}{h}\right). \quad (6)$$

In our algorithm we use the following standard kernel function

$$G(x) = \begin{cases} 1 & x \leq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Thus, the weighted mean computation in Eq. 4 is simplified to a simple mean computed on a subset of the rotations which are close to $\mathbf{R}$. The mode of the rotation distribution can be calculated by iteratively computing the mean of a set of rotation matrices. This is done using Algorithm 1 which finds the rotation matrix $\mathbf{R}$ that minimizes the angular distance between the set of rotation matrices as presented by Hartley *et al*. [11].

**Algorithm 1** *L2 mean algorithm on SO(3)*

Set $\mathbf{R} := \mathbf{R}_1$ Choose tolerance $\epsilon \geq 0$.
**loop**
  Compute $\mathbf{r} := \frac{1}{n} \sum_{i=1}^{m} \log\left(\mathbf{R}^T \mathbf{R}_i\right)$.
  **if** $\|\mathbf{r}\| < \epsilon$ **then**
    **return R**
  **end if**
  Update $\mathbf{R} := \mathbf{R} \exp(\mathbf{r})$.
**end loop**

In this algorithm $\log(\mathbf{R})$ maps the rotation matrix $\mathbf{R}$ to a vector in the tangent space, and $\exp(\mathbf{r})$ maps $\mathbf{r}$ back to the $\mathbf{SO}(3)$ space.

Substituting the distance function and algorithm for computing the mean in to the mean shift algorithm yields a mean shift algorithm which can be applied to a set of rotation matrices.

## 2.2. Algorithm Overview

Our Proposed algorithm for point cloud registration is illustrated in Figure 1. The algorithm includes the following steps.

1. Rotation estimation using mean shift.
2. Translation estimation using mean shift.
3. Fine tuning the registration.

We will now elaborate on these three steps.

## 2.3. Mean shift rotation estimation

The rotation between two viewpoints $\mathbf{V}_1, \mathbf{V}_2$ is calculated as the rotational movement between the LRFs of corresponding keypoints. For each pair of corresponding points $(\mathbf{P}_i, \mathbf{P}_j)$ the rotation matrix between them is estimated as

$$\mathbf{R}_{i,j} = LRF_j^T LRF_i.$$

For each putative match, its rotation matrix $\mathbf{R}_{i,j}$ is saved with their respective quaternion $\mathbf{q}_{i,j}$ in a data structure.

Huynh [13] showed that Eq. 5 is an efficient calculation for computing the distance between quaternions. Thus, when applying the mean shift iteration on a rotation matrix, the distances between its quaternion and all the quaternions in the set are computed. The subset of rotations $\mathbf{S}$ whose distance is less than the threshold $ta$ is retrieved.

Then for each such subset $\mathbf{S}$ the average rotation is calculated using Algorithm 1 yielding the input to the next iteration. This process continues until convergence. Rotations which converge to a similar rotation are clustered together. The mode of largest cluster is returned by this step of the algorithm. We found however that in order to limit the effect of noise on the voting process, the function returns all the clusters whose size is $\geq 60\%$ of the largest cluster.

This is the main part of the algorithm which also uses most of the running time. The runtime of the algorithm depends on the number of the putative matches and on the sizes of the groups of matches on which Algorithm 1 is run. Thus, a peculiar phenomenon occurs. For easy cases in which the number of inlier rotations is large, the running time of the algorithm can be much higher then for more challenging cases. To deal with this problem we made the following simple modification to the algorithm. For each iteration in which the algorithm converges to a mode of the distribution, if the size of $\mathbf{S}$ is larger than a predefined threshold, the algorithm simply returns the computed rotation matrix which is the mode of the distribution and the subset $\mathbf{S}$. Thus, usually after the first inlier rotation has been processed the algorithm will terminate successfully.

### 2.4. Mean shift translation estimation

For each of the rotation matrices returned by the previous step the translation vector is estimated. The indexes of the points that compose the Mean shift rotation $MSR$ cluster, are used to calculate the translation between the view points. In order to estimate the translation vector $\mathbf{t}_{i,j}$ from a pair of putative correspondences $(\mathbf{P}_i, \mathbf{P}_j)$ it is estimated as

$$\mathbf{t}_{i,j} = (-MSR)\mathbf{P}_i + \mathbf{P}_j.$$

We apply the standard mean shift algorithm to cluster together close points. In this case the threshold $td$ is used. The mode of the largest cluster is returned by the algorithm. Just like in the previous step we check all the clusters whose size is $\geq 60\%$ of the largest cluster, to reduce the effect of noise on the translation.

On the points that remain after these two steps, the rigid transformation is calculated. Once the coarse registration is established, an ICP procedure is applied to create a more accurate registration. The rotation matrix and translation vector returned by the ICP procedure will be referred to as $\mathbf{R}_{es}$ and $\mathbf{t}_{es}$ respectively.

## 3. Discussion

In this section we will review several approaches used in the literature to estimate the rigid transformation and compare their characteristics to our algorithm.

The most common method used for estimating data models such as rigid transformations is RANSAC [5, 20, 1, 16, 10]. In this case RANSAC has several deficiencies. First while our algorithm is deterministic, RANSAC is a randomized algorithm. It can therefore succeed in one run of the algorithm while fail in another. RANSAC also only uses the keypoints, while our algorithm exploits also the LRF. As a result in many cases RANSAC is not able to find a good estimation of the rigid transformation. In addition, in hard cases which is our focus here, the number of iterations of the algorithm can be very high. This is since it is proportional to $\frac{1}{p^3}$, where $p$ is the percentage of inlier rotations. Our algorithm on the other hand can handle a large number of outlier rotations since usually not many other rotations are near it, causing $\mathbf{S}$ to be relatively small and the number of iterations to convergence is also small.

There are however several algorithms which use LRF information. In [9], each putative match is represented by its LRF and the translation vector computed using its LRF. Then clustering in six dimensions is performed on the Euler angle representation of the LRF and the translation vector. Due to the fact that LRFs are very noisy, using them to compute the translation vector yields a very large cluster for the translation component. We however, use the computed mode of the distribution which is usually a much more accurate estimate of the sought after rotation matrix, yielding a more compact cluster of the translation component which can be more easily detected. Moreover, Euler angles are not considered a good representation for rotation matrices for clustering purposes [28][Chap. 15.3.8].

In [10], LRFs are clustered separately like in our algorithm. For each suggested cluster, translations are clustered and then the solution is verified using ICP performed on simplified meshes of the scans. In the next section we compare the performance of our algorithm to it. Due to the fact that we use the mean shift algorithm, more accurate estimations of the rotation component is found and less candidate rotations are used and thus our algorithm runs faster. Moreover, our algorithm can run on point clouds and does not require a mesh to be provided.

## 4. Experiments

To test our method we used two type of datasets, small scale scans of 3D models, and large scale scans of cities. In this work we used three models from the Stanford Repository [4] (Bunny, Dragon and Armadillo), acquired by a Cyberware 3030 MS laser scanner.

Urban scenes were scanned by two different methods. A structure from Motion (SfM) data of a city which contains over 3 million points was generated from a large number of aerial images. This data was provided to us by an high tech company.

24 large scale scans of the city of Vancouver were provided to us by another company were acquired by a Z+F IMAGER®5010, 3D Laser scanner. Each scan yields $11,114,444$ 3D points. The properties of each dataset can be seen in Table 1 and scan examples can be seen in Figure 2.

We compared our method to our implementations of RANSAC with 500,1000,5000,10000 iterations, and the CCV method introduced by Guo *et al.* [10] with the same parameters. Keypoints were detected using the Harris3D [23] interest point detector and the RoPS[9] feature descriptor was used utilizing the robustness of its LRF.

All of the experiments described in this work were conducted on a computer with 3.4Ghz Intel Core i7 6700k and 32GB of RAM. The algorithms described in this work were implemented by us in MATALB.

### 4.1. Stanford dataset

In order to run the LRFMS algorithm two parameters have to be set: the rotation matrix bandwidth $ta$ and the translation bandwidth $td$. They should be large enough to capture the cluster but not too large so that outlier elements are not included in it. For the standard data sets we found that the parameter values for $ta$ and $td$ that yield the best results are $ta = 0.8^\circ$ and the $td = 10mr$, where $mr$ is the mesh resolution.

| | N°. of scans | Avg. N°.of vertices $10^3$ | N°.of key points | Keypoint detector |
|---|---|---|---|---|
| Armadillo | 141 | 25 | 1,000 | Harris3D |
| Bunny | 45 | 36 | 1,000 | Harris3D |
| Dragon | 108 | 30 | 1,000 | Harris3D |
| Vancouver dataset | 24 | 980 | 5,000 | Random |
| Structure from motion | 2 | 3,800 | 10,000 | Harris3D |

Table 1. Properties of the data used in this work.

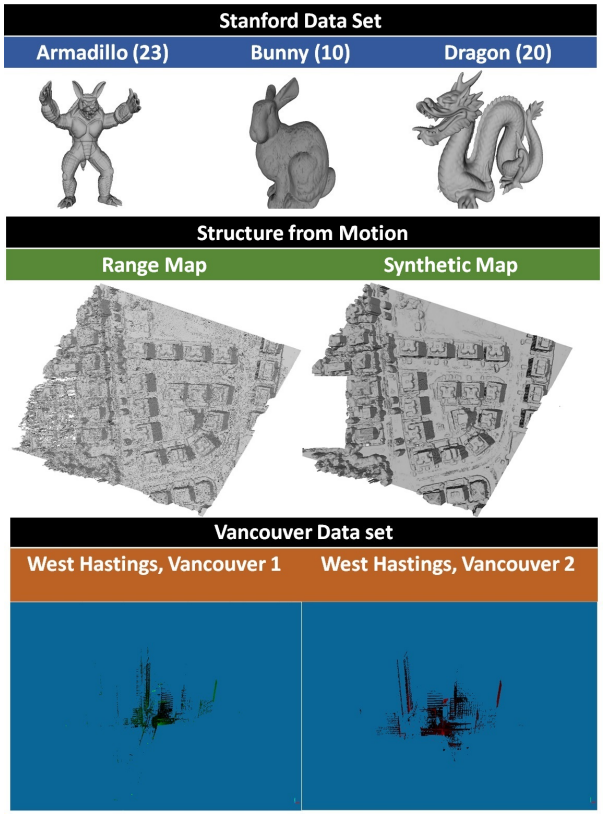



Figure 2. On the top the Stanford data set: number of views used in brackets. In the middle the structure from motion data. On the bottom a LIDAR Scan of Vancouver.

In addition we have to set the Lowe ratio value used in selecting the putative matches. As the value increases the number of inliers increases slightly while the number of outliers increases considerably. As mentioned above this is a price worth paying. We therefore compare the values $0.99$ and $0.9$. Figure 3 shows the performance of the algorithm for these two values as a function of $ta$ demonstrating why $ta = 0.8°$ and Lowe ratio value $0.99$ were chosen.

The evaluation method used on the Stanford Repository was proposed by Petrelli *et al.* [19]. Success is measured based on the root mean square error $RMSE$ between the registration result and the ground truth. If the $RMSE \leq mr5$ then the point clouds are considered successfully registered.

The algorithms were run on the Stanford data set, as shown in tables 2,3, and 4. It is important to mention that the reported times also include the loading time and the ICP running times. The average time for the registration of each model without the loading time and the ICP time is reported as Avg. Alg. Time. The number of registered pairs are averaged over 20 runs of the RANSAC.

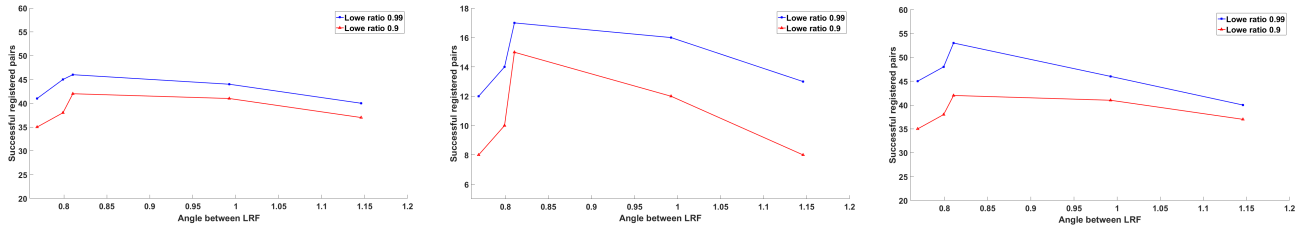
LRFMS with ICP and the LRFMS without ICP outperforms the RANSAC both in term of running times and the number of successfully registered pairs. The CCV and the LRFMS yield similar results for the number of successful pair registrations. For the running time, the LRFMS outperforms the CCV by a considerable margin. The diameter of the winning rotation cluster ranges from $0.2°$ to $1.2°$. This has been obtained using the same value of $ta$. The range of the diameter makes discrete methods such as Hough transform less effective then the proposed mean shift based method.

| | N° Registrations | CPU time (sec) | Avg. Alg. Time |
|---|---|---|---|
| RANSAC 500 | 32 | 336 | 0.1 |
| RANSAC 1000 | 33 | 489 | 1.2 |
| RANSAC 5000 | 40 | 1427 | 7.8 |
| RANSAC 10000 | 38 | 2530 | 15.6 |
| LRFMS | 46 | 724 | 5.1 |
| CCV | 56 | 860 | 6.1 |
| LRFMS Without ICP | 41 | 139 | 1 |

Table 2. The Armadillo data set contains 141 pairs. The average algorithm time reflects the algorithm running time for a single pair without the time for loading the data and running the ICP procedure.

### 4.2. Real world data set

Our next goal was to test the algorithms on large scale datasets: a sparse aerial scan and terrestrial Lidar scans of the city of Vancouver. The angular threshold $ta$ that achieved the best results for both datasets was $ta = 1.8°$. While the angular threshold was the same for both datasets




(A) Armadillo (B) Bunny (C) Dragon

Figure 3. The number of successfully registered pairs for the Stanford repository with different thresholds $ta$ for Lowe ratio values of $0.90$ and $0.99$.

| | N° Registrations | CPU time (sec) | Avg. Alg. Time |
|---|---|---|---|
| RANSAC 500 | 6 | 79 | 0.1 |
| RANSAC 1000 | 7 | 129 | 0.6 |
| RANSAC 5000 | 13 | 285 | 4 |
| RANSAC 10000 | 11 | 413 | 6.9 |
| LRFMS | 17 | 236 | 4.8 |
| CCV | 18 | 336 | 6.6 |
| LRFMS Without ICP | 13 | 66 | 1.4 |

Table 3. The Bunny data set contains 45 pairs.

| | N° Registrations | CPU time (sec) | Avg. Alg. Time. |
|---|---|---|---|
| RANSAC 500 | 26 | 143 | .01 |
| RANSAC 1000 | 32 | 282 | 0.3 |
| RANSAC 5000 | 36 | 803 | 5.1 |
| RANSAC 10000 | 37 | 1484 | 11.4 |
| LRFMS | 53 | 518 | 4.8 |
| CCV | 51 | 716 | 6.6 |
| LRFMS Without ICP | 44 | 111 | 1.03 |

Table 4. The Dragon data set contains 108 pairs.

the difference between the density of the scans led to different distance thresholds $td$. For the structure from motion dataset the value for $td$ that proved to be the most effective was $td = 65cm$, and the best result for the Vancouver dataset was achieved $td = 35cm$.

To evaluate the large scale scans, we used the evaluation method described in Guo *et al*. [7]. The ground truth rotation $\mathbf{R}_{GT}$ and the ground truth translation $\mathbf{t}_{GT}$ were used for the evaluation. The rotation and translation estimations $\mathbf{R}_E$ and $\mathbf{t}_E$ are the outputs of the algorithm. The rotation error between the ground truth and the registration result $e_r$ and the translation error $e_t$ are computed as follows.

$$e_r = \arccos\left(\frac{trace\left(\mathbf{R}_r\right) - 1}{2}\right)\frac{180}{\pi}. \tag{8}$$

where,

$$\mathbf{R}_r = \mathbf{R}_{GT}\mathbf{R}_E^{-1}. \tag{9}$$

$$e_t = \|\mathbf{t}_{GT} - \mathbf{t}_E\|. \tag{10}$$

For a registration to be successful $e_r \leq 2^\circ$ and $e_t \leq 100cm$.

#### 4.2.1 SfM dataset

The structure from motion dataset was obtained from an industrial company contains two scans. A range map and a synthetic map, both scans contain over 3 million vertices.

This is a very challenging data set since the Range map is very noisy and has a limited number of recognizable structures or geometry as can be seen in figures 2 & 5.

The synthetic map was generated from the scan map by trying to model the scene from it. This is done by removing noisy points and modelling the objects in the scene such as buildings and roads. $10,000$ key points were selected using Harris3D keypoint detection and matched. Matches for which Lowe's ratio is below $0.99$ were maintained. Here again this value yielded the best result for LRFMS.

The result of the registration can be seen in Figure 4 and Table 5. The rotation error between the LRFMS output and the ground truth is $0.5^\circ$ and the translation error between the LRFMS output and the ground truth is $50cm$. The accuracy of the results obtained the CCV algorithm are much lower. This can also be seen in Figure 4 in which the registration result is compared to the ground truth. This shows that the LRFMS can handle data sets where other state of the art methods failed.

#### 4.2.2 Vancouver dataset

The Vancouver data set is very dense. Each scan contains $11,114,444$ data points. Standard preliminary steps were applied to the data by the company. A $7 \times 7$ median filter

| | CPU time (sec) | Error° | Distance (cm) | Status |
|---|---|---|---|---|
| RANSAC 500 | 5 | 57° | 1700 | Fail |
| RANSAC 1000 | 9 | 45° | 1530 | Fail |
| RANSAC 5000 | 18 | 20° | 1420 | Fail |
| RANSAC 10000 | 27 | 18° | 975 | Fail |
| LRFMS | 82 | 0.5° | 50 | Success |
| CCV | 249 | 4.4° | 840 | Fail |

Table 5. Results for structure from motion dataset. The error in rotation between the ground truth to the estimated result is computed using Eq. 8. The distance between the translation vectors.

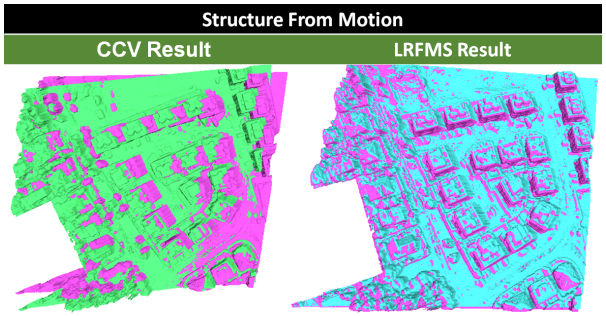



Figure 4. Structure from motion dataset. On the bottom left the CCV result is compared to the ground truth, while on the bottom right the LRFMS is compared to the ground truth.

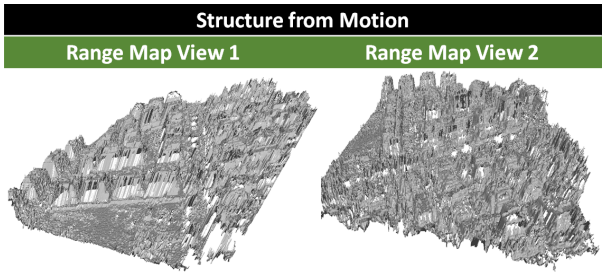



Figure 5. Additional views of the Rage Map

is applied on the raw scans. Points whose distance from the median $\geq 0.03m$ were discarded. Next a voxel filter is used to down-sample the data. The voxel edge size is set to $0.5m$. Each voxel contributes one point to the point cloud. On average $984,657$ points remain in the point cloud.

From those points $5,000$ key points were randomly selected. The CCV algorithm uses a mesh reduction function for its ICP based verification step. Since this data set contains only point clouds CCV was not tested on this data.

Figure 6 shows a partial map of Vancouver and the center positions of each of the 24 scans. The scans were performed on four different streets (each shown in a different color). The distance between a scan and the subsequent scan on the same street is between 22 to 29 meters.

We tested The LRFMS on all 24 scans and successfully registered all of the subsequent scans. On average the angular error between the LRFMS registration result and the ground truth is $0.25°$ and the translation error between the LRFMS registration result and the ground truth is $25cm$. Running RANSAC with $10,000$ iterations also succeeded but with less accuracy. Two examples are shown in figures 7 8 9 10. In each case on the left the scans are shown before the algorithm is applied (therefore buildings can be seen twice) and on the right the LRFMS result is shown.

We then tried to apply the algorithm on more challenging cases. We took all pairs of scans on the same street at distance two. These pairs of scans have a much smaller overlap between them. In this case we were able to register 3 out of 16 pairs. RANSAC failed on all these cases.

The supplementary material provides several videos showing registration results on a few examples from the different datasets.

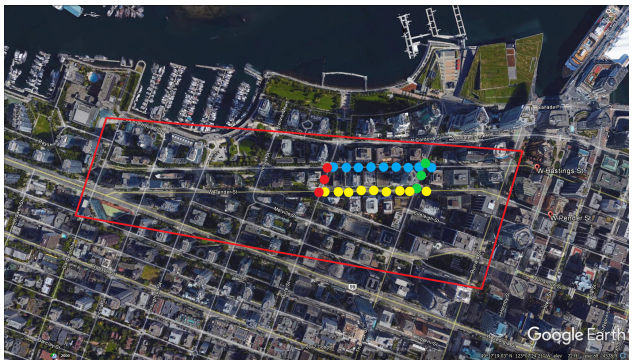



Figure 6. The region of Vancouver scanned in the data set. Red points represents West Bute street, light blue point represents West Hastings street, green points represents Thurlow street, yellow point represents West Pender street. Points represents he center of each scan.

| | CPU time (sec) | Error° | Distance (cm) | Status |
|---|---|---|---|---|
| RANSAC 500 | 1.5 | 70.8° | 280 | Failed |
| RANSAC 1000 | 1.6 | 62.9° | 190 | Failed |
| RANSAC 5000 | 2.4 | 2.6° | 150 | Failed |
| RANSAC 10000 | 3.1 | 0.65° | 85 | Success |
| LRFMS | 6.1 | 0.25° | 25 | Success |

Table 6. Vancouver dataset results on subsequent scans. The error is the rotation between the ground truth to the estimated result (Equation 8) and the distance between the translation vectors (Equation 10).

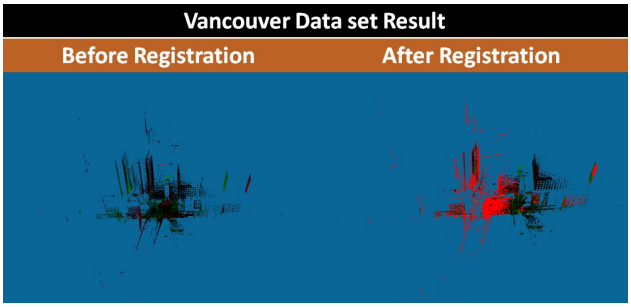



Figure 7. Vancouver West Hastings data set: on the left scans before the registration process and on the right the LRFMS registration results.

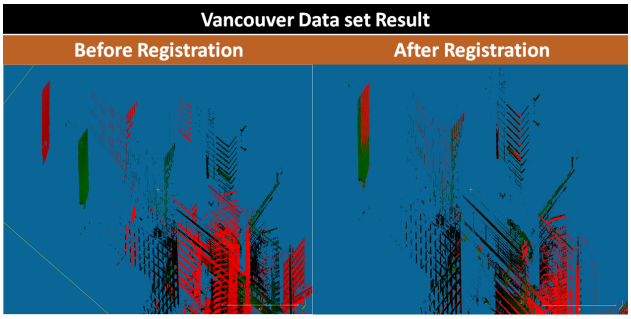



Figure 8. Vancouver West Hastings zoom-in data set: on the left scans before the registration process and on the right the LRFMS registration results.

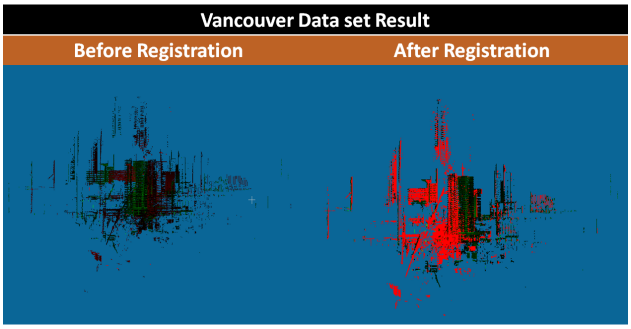



Figure 9. Vancouver West Pender street data set: on the left scans before the registration process and on the right the LRFMS registration results.

## 5. Conclusions

In this paper we presented a novel registration algorithm between point clouds. The algorithm exploits the fact that modern 3D descriptors are accompanied by a locally estimate local reference frame (LRF). Using a mean shift algorithm especially designed to cluster rotation matrices, the mode of the distribution of the inlier LRFs is found which is close to the relative rotation between the scans. In a second step, mean shift is used again to find the translation vector. Using this system we were able to deal with very challenging pairs of point clouds. These point clouds were

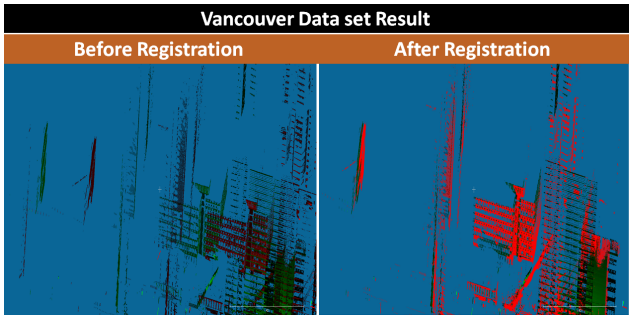



Figure 10. Vancouver West Pender zoom-in data set: on the left scans before the registration process and on the right the LRFMS registration results.

generated by very different scanning techniques (3D scanners, Lidar laser scanners, and a Structure from Motion algorithm). Comparing this algorithm to several state of the art algorithms demonstrates its superiority both in success rate and running times.

Future work will be devoted to develop methods to speedup our algorithm and improve its performance on more challenging pairs of scans with lower overlap between them.

## Acknowledgments

We would like to thank the Stanford 3D Scanning Repository, the Geosim Systems Ltd for providing us with the Vancouver data set, and RAFAEL advanced defense systems Ltd for the Structure from Motion data set. This work was supported by the Israeli Innovation Authority in the Ministry of Economy and Industry.

## References

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics (TOG)*, 27(3):85, 2008. 1, 4

[2] A. G. Buch, H. G. Petersen, and N. Krüger. Local shape feature fusion for improved matching, pose estimation and 3D object recognition. *SpringerPlus*, 5(1):1, 2016. 1

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 2

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 4

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[6] B. Gálai, B. Nagy, and C. Benedek. Crossmodal point cloud registration in the hough space for mobile laser scanning

data. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3374–3379. IEEE, 2016. 1

[7] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. An integrated framework for 3-D modeling, object detection, and pose estimation from point-clouds. *IEEE Transactions on Instrumentation and Measurement*, 64(3):683–693, 2015. 6

[8] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3D local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016. 1

[9] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. Rotational projection statistics for 3D local surface description and object recognition. *International journal of computer vision*, 105(1):63–86, 2013. 1, 2, 4

[10] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu. An accurate and robust range image registration algorithm for 3D object modeling. *IEEE Transactions on Multimedia*, 16(5):1377–1390, 2014. 4

[11] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013. 3

[12] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3D. *IEEE Robotics & Automation Magazine*, 22(4):110–124, 2015. 1

[13] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009. 2, 3

[14] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999. 1

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[16] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu. Robust l2e estimation of transformation for non-rigid registration. *IEEE Trans. Signal Processing*, 63(5):1115–1129, 2015. 4

[17] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1584–1601, 2006. 1

[18] A. S. Mian, M. Bennamoun, and R. A. Owens. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal of Computer Vision*, 66(1):19–40, 2006. 1

[19] A. Petrelli and L. Di Stefano. Pairwise registration by local orientation cues. *Computer Graphics Forum*, 35(6):59–72, 2016. 1, 5

[20] J. Rabin, J. Delon, Y. Gousseau, and L. Moisan. Mac-ransac: a robust algorithm for the recognition of multiple objects. In *Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPTV 2010)*, page 051, 2010. 4

[21] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009. 1

[22] S. Salti, F. Tombari, and L. Di Stefano. A performance evaluation of 3D keypoint detectors. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 236–243. IEEE, 2011. 1

[23] I. Sipiran and B. Bustos. Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes. *The Visual Computer*, 27(11):963–976, 2011. 2, 4

[24] R. Subbarao and P. Meer. Nonlinear mean shift over riemannian manifolds. *International journal of computer vision*, 84(1):1–20, 2009. 2

[25] F. Tombari, S. Salti, and L. Di Stefano. Unique shape context for 3D data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62. ACM, 2010. 1

[26] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. 1

[27] O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3D motion estimation via mode finding on lie groups. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 18–25. IEEE, 2005. 2

[28] A. H. Watt and M. Watt. *Advanced animation and rendering techniques*. ACM press New York, NY, USA:, 1992. 4