Real-Time Scale Invariant 3D Range Point Cloud Registration

Anuj Sehgal[1], Daniel Cernea[2], and Milena Makaveeva[3]

[1] Indian Underwater Robotics Society,
E-118 Nar Vihar Part-I, Sector 34,
201301 Noida, India
anuj@iurs.org
[2] University of Kaiserslautern, Department of Computer Science,
67653 Kaiserslautern, Germany
cernea@informatik.uni-kl.de
[3] Jacobs University Bremen, Computer Science, Campus Ring 1,
28759 Bremen, Germany
mmakaveeva@jacobs-university.de

Abstract. Stereo cameras, laser rangers and other time-of-flight ranging devices are utilized with increasing frequency as they can provide information in the 3D plane. The ability to perform real-time registration of the 3D point clouds obtained from these sensors is important in many applications. However, the tasks of locating accurate and dependable correspondences between point clouds and registration can be quite slow. Furthermore, any algorithm must be robust against artifacts in 3D range data as sensor motion, reflection and refraction are commonplace. The SIFT feature detector is a robust algorithm used to locate features, but cannot be extended directly to the 3D range point clouds since it requires dense pixel information, whereas the range voxels are sparsely distributed. This paper proposes an approach which enables SIFT application to locate scale and rotation invariant features in 3D point clouds. The algorithm then utilizes the known point correspondence registration algorithm in order to achieve real-time registration of 3D point clouds.

1 Introduction

Due to the relative inexpensiveness and multiple benefits available from representing the viewed environment in 3D images, sensors and stereo cameras that are able to provide 3D point cloud data are becoming increasingly popular. 3D point cloud data representation is extremely important in various fields such as archaeology, geology, oceanography and lately even in robotics, where 3D point clouds are increasingly utilized for mapping and localization of robots in a 3D environment.

However, in all these application domains it is rare to obtain a single representation of the data [1] and as such multiple frames of point clouds have to be obtained and registered with respect to each other in order to construct a

A. Campilho and M. Kamel (Eds.): ICIAR 2010, Part I, LNCS 6111, pp. 220–229, 2010.

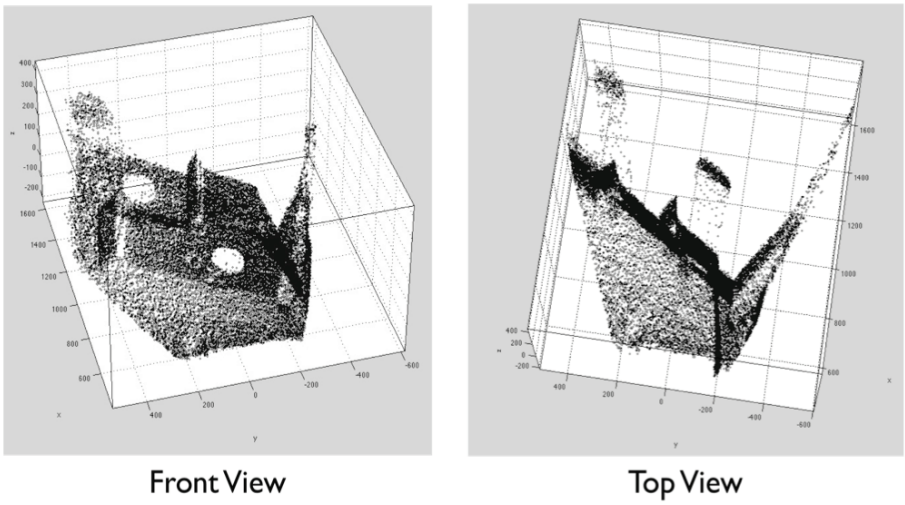


Fig. 1. Point cloud obtained from an IR-Ranger

composite map or scene. This composite data can be then further utilized for localization, analysis or visualization purposes.

Full automation of the registration process of range image 3D point clouds is a topic of active research and most systems still rely upon user input in order to determine the initial transformation. Additionally, the algorithms are highly processor intensive [2] making real-time registration of these point clouds a non-trivial effort. Furthermore, range point clouds provide another challenge as compared to intensity images in the form of noise that may be present within the returned data, causing false artifacts to appear in the point clouds or making the point cloud too sparse, with not enough usable information within it [3]. For example, range image point cloud data obtained from IR rangers can be highly noisy because of spurious readings resulting from ambient light and also as a result of the surface property of the target object; if the object is dark the range data would be erroneous since infra-red light is absorbed by darker colors. Though it is possible that this erroneous data may provide matchable features in some applications, in all our tests there were no matches located within such areas. Furthermore, the 3D shape formation errors induced by this data can lead to the registered point clouds appearing highly deformed.

A rendering of one such frame of a point cloud is provided in Figure 1. The noisy nature of this data is clearly visible in this representation. The point cloud frame consists of a number of boxes stacked upon each other; the boxes have a large circular black section painted in the middle. In the front view, this section appears as a white empty space, but if a top view is obtained it becomes clear that this feature is still present, but displays much further behind the box. This erroneous result is introduced because the black color absorbs the IR ranging beacon and as such the farthest possible distance is assumed for that region. It

is extremely important for the correspondence detection algorithm to be able to successfully function despite the existence of noise [4].

In order to enable real-time automatic registration of these point clouds, our approach depends upon locating robust features, invariant to scale, rotation and point of view within the point clouds. These robust features can then be used with a high certainty to locate correspondences between the point clouds, by matching these features within two sets of voxels. Registration of the two images is then carried out using a known point correspondences algorithm. However, to reduce the effect of the possible noise in range data, it is necessary to select a feature descriptor that is very robust and, more importantly, invariant to scale and rotation changes.

As such, to meet our goals of locating a high number of features with a high degree of certainty, repeatability from multiple poses and in data with high noise, an algorithm utilizing features based upon the Scale Invariant Feature Transform (SIFT) descriptor model [5] was developed to find correspondences between the point clouds. The SIFT features are highly robust in that they are orientation invariant and are applicable at multiple scales. The SIFT feature detector algorithm is able to generate a large number of localized features with a relatively low computational cost. The detected SIFT features in the point clouds can be matched with a high degree of certainty and repeatability, from multiple poses and without respect to scale.

The following sections of the paper present information related to the SIFT algorithm and then proceed to describe the approach used to find SIFT features in 3D point clouds. The correspondence detection algorithm and the registration method used are also discussed. Some results, test and performance data obtained using the approach are presented and discussed along with the conclusions that are drawn from the results.

2 The SIFT Feature Detector

Robust detection of features in a scene are necessary in order to find correspondences within a point cloud so as to expedite the registration process, which normally can be computationally expensive. The features provided by the SIFT algorithm are local and invariant to image scale and rotation, thereby making them quite robust. These features are also robust in response to changes in illumination and minor changes in viewpoint while, being highly distinctive [6]. The SIFT algorithm is implemented in four stages that provide a result in form of multiple feature descriptors that are represented as a 128-element vector to achieve scale and rotational invariance.

The first stage of the algorithm is where all possible points of interest, known as key-points, are detected. In order to achieve this, the input data is successively convolved with Gaussian filters at different scales, and then the difference of successive Gaussian-blurred images are taken. The local extremum points that exist within the Difference of Gaussians (DoG), an approximation to the Laplacian, at multiple scales are then accepted as the key points. Once the initial set

of candidate key points is obtained from the DoG images, they are analyzed within their own neighborhood and adjacent scales, to determine whether they are a local maxima or minima. Furthermore, the second step discards the key-point coordinates that are located in noisy space. This is achieved by eliminating candidates that lie in a region of low contrast or on the edges.

The third step achieves invariance to rotation by assigning each key-point one or more orientations. To compute the orientation of a point in a scale-invariant manner, the Gaussian-smoothed image corresponding to the scale from which the key-point was originally derived is taken and an orientation and gradient magnitude assigned to it. Magnitude and direction calculations for the key-points are performed for every pixel in the neighborhood and an orientation histogram is generated with 36 bins, each bin covering 10 degrees. Once the histogram is fully populated the orientations with the highest peaks and those that are within 80% of the highest peaks are assigned to the key point.

The final step in the SIFT algorithm actually computes the descriptor vector that can be used to identify and further match each key point. This step is extremely similar to the orientation assignment method. The feature descriptor is computed as a set of orientation histograms on a pixel neighborhood of size 4 times 4. The histograms are relative to the key point orientation and the orientation data is derived from the image that corresponds to the key point's scale. The representations now contain 8 bins, each leading to the derivation of a SIFT feature vector that contains 128 elements. This vector may be used to perform image matching or pattern recognition.

3 The Registration Algorithm

The aim of the work presented is to be able to enable automatic real-time registration of the 3D point clouds. Currently, the most popular method for registration is the Iterative Closest Point (ICP) algorithm or some derivative of the same [1]. The ICP algorithm and most of its derivatives are computationally expensive, giving rise to the necessity of being able to perform registration based upon pre-located correspondences from a fewer set of points, in order to speed up the overall performance of the registration process. However, this approach requires that the pre-computed correspondences between the point clouds be calculated quickly, while also ensuring their accuracy between frames that could have changing rotation, translation and scaling. In order to achieve this goal a three-step algorithm that uses the SIFT feature descriptor to describe key points in point clouds is designed. The three steps of the algorithm (data preprocessing; SIFT descriptor generation and feature matching to locate correspondences; and registration of point clouds) are discussed in the following subsections.

3.1 Data Preprocessing

The SIFT feature detector is designed to function only with 2D datasets and as such, in order to extract SIFT features from the range data in the point cloud,

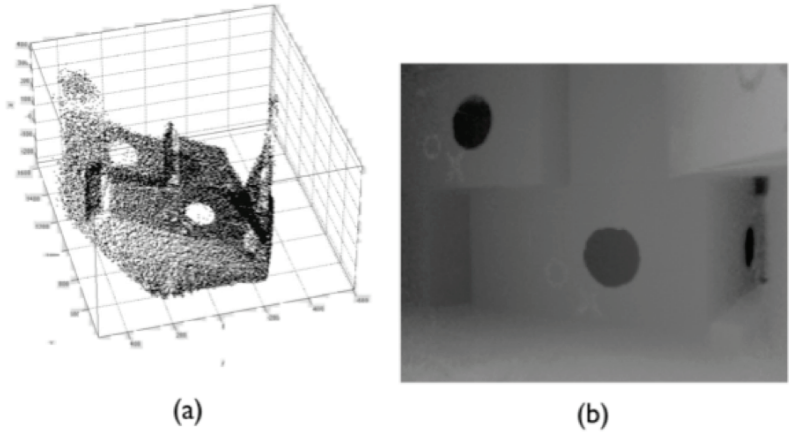


Fig. 2. Square-root scaled image of a point cloud; (a) The point cloud; (b) Square-root scaled range data

this information is square root scaled to fit between 0-255. The Euclidean distance to each individual (x, y, z) coordinates within the point cloud is calculated from the origin $(0, 0, 0)$ and scaled using square root scaling. An image representation of this range data square root scaling can be seen in Figure 2. This data was derived from a Swiss IR Ranger mounted on a mobile robot.

Upon performing the square-root scaling, the data is passed through a PNG converter in order to obtain images to which the SIFT operator is applied. The SIFT feature detector requires continuous points in the neighborhood of a pixel to function. Voxels in a 3D range point-cloud are not densely located, and as such the SIFT detector cannot be extended to 3D range point-clouds directly. This necessiates the square-root scaling step before the SIFT feature detector can be used.

3.2 SIFT Feature Detector and Matching

The SIFT feature detector is built using OpenCV [7] to follow closely the SIFT algorithm from [5,6]. The SIFT algorithm takes as input a PNG image corresponding to the square-root scaled representation of the point cloud and computes the 128-element vectors for every identified feature key point.

Upon obtaining the SIFT feature descriptors from the square root scaled images for the two point clouds to be registered, correspondences between the (x, y, z) coordinates in the point clouds is obtained by searching for matching SIFT descriptors, using the RANSAC algorithm [8]. The RANSAC algorithm selects a set of feature pairs randomly and computes the set of all feature pairs conforming to the implied transformation. A support set is rejected if it results in a size that is below a certain threshold.

Figure 3 shows matches found between the scaled point cloud images. The results in Figure 3 make it appear as though the the number of corresponding

Fig. 3. SIFT descriptor based matches for two frames from a 3D point cloud data set. Each subfigure represents different types of point cloud data from the same frame (a) Range point cloud (b) Intensity point cloud (c) Range & Intensity combined point cloud (using intensity as another dimension in scaling) (d) Another Range and Intensity combined point cloud (obtained by multiplying intensity and distance before scaling).

matches is not very high. However, the data depicted in this figure is from a Swiss IR Ranger mounted on a robot that is moving swiftly, causing it to be blurry. This gives rise to a limited set of features to match, but our approach functions by successfully performing registration between images as long as at least three correspondences are located. Since a relatively high number of correspondences are located even in noisy and blurry data, the SIFT feature detector appears to function robustly on scaled range images as well.

Once the matches are found on the basis of the RANSAC algorithm, correspondences between the 3D point clouds are easily derivable since the corresponding location of each key point in the square-root scaled data is known within the point cloud as well. The set of resultant correspondences can now be further used with the chosen registration algorithm.

3.3 Point Cloud Registration

Registration is necessary in order to be able to compare or integrate the data from different measurements. This step provides the relative rotation, translation and scale of the two 3D point clouds being compared. The popular ICP algorithm is memory and processor intensive, thereby being unsuitable for real-time applications [9]. However, if a known points correspondence algorithm is used, this can considerably speed up the registration performance.

As such, for the purpose of speeding up the registration step and owing to the robustness of the SIFT features, the known points correspondence registration algorithm based on quaternions is utilized in our approach. Every corresponding point in the range point clouds can represent a quaternion with $c = 0$ and x, y

and z coordinates given by the respective coordinates of the point in the 3D point cloud.

By comparing data from two consecutive point clouds, we are able to retrieve the quaternion of the rotation matrix $\check{u}^*$ from the eigenvector corresponding to the maximum positive eigenvalue of the 4x4 matrix N shown below:

$$N = \sum_{i=1}^{n} \bar{\Gamma}(\check{r}_{l,i})^T \Gamma(\check{r}_{r,i}) \tag{1}$$

Further, the rotation matrix can be obtained from the rotation quaternion by,

$$R = \bar{\Gamma}(\check{u}^*)^T \Gamma(\check{u}^*) \tag{2}$$

Having obtained the rotation matrix R, the translation quaternion and scale factor are calculated using,

$$r^*_{l/r} \equiv \bar{r}_r + sR_{r/l}(\bar{r}_l) \tag{3}$$

$$s^* = \frac{\sum_{i=1}^{n} (r'_{r,i})^T R_{r/l}(r'_{l,i})}{\sum_{i=1}^{n} \left\| r'_{l,i} \right\|^2} \tag{4}$$

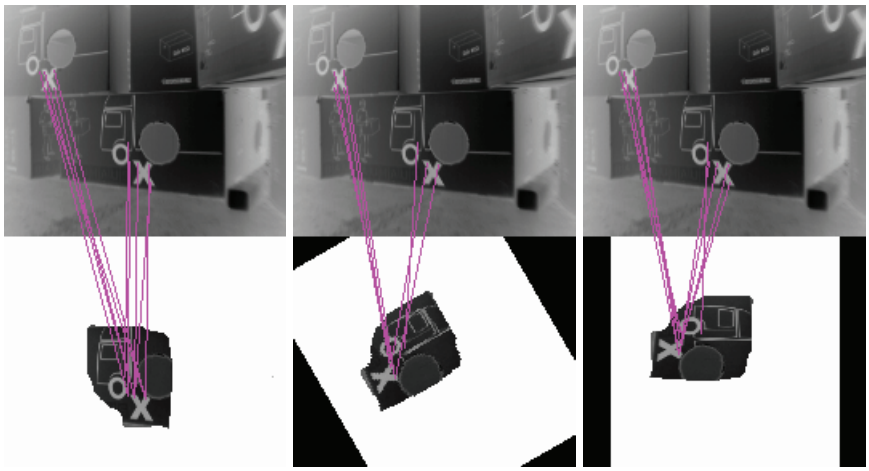
The matrix calculations required for the registration step were performed using the GSL library [10] and several extensions were written in order to calculate the eigenvectors, eigenvalues, vector normalization and etc.

We also further extended our work in order to utilize registration to derive the roll, pitch and yaw between the consecutive frames. This can be extremely useful in robotics since it provides a method to derive odometry by using only range sensor data, rather than depending upon sensors like GPS, which may not function under certain conditions. After obtaining the rotation matrix, calculation of the respective roll, pitch and yaw is a straightforward task of selecting the appropriate row/column pairs from the rotation matrix. Having obtained translation and the yaw, pitch and roll the robot odometry is available. This can be used in localization and mapping tasks commonly performed by robots.

4 Testing and Results

The test dataset used to evaluate the overall algorithm was obtained from a Swiss IR Ranger mounted on a mobile robot. The dataset consists of 290 frames of point clouds. The (x, y, z) location within the point cloud corresponds to the measured distance in millimeters. This data is retrieved frame by frame and supplied to the software running on a Linux platform in order to register the two point clouds. The test system used was a SuSe Linux installation on a platform with 512MB RAM and a 1 GHz AMD Athlon64 CPU.

The robustness of the SIFT features and their ability to have more than a single orientation at a particular point can cause the RANSAC algorithm to



(a) (b) (c)

Fig. 4. Results of matching a template; (a) Template without rotation; (b) 60° rotated template; (c) 90° degree rotated template

successfully find more than one correspondence of a feature in the adjacent frame. This is especially useful in case a particular pattern or object needs to be located within a particular point cloud. This may be achieved by having a reference template point cloud of a particular object and then using our approach to register the target and template point clouds.

Figure 4 shows the results of such a template matching experiment. In this case the template used was a subset of a cardboard box placed in front of the Swiss IR Ranger. The target point cloud was obtained by stacking multiple boxes of this type and then registering the point cloud to the template. The lower half of all the images in Figure 4 show the template and the upper half are the target point cloud representation. The template data was also rotated in order to ensure that the features from the template could still be matched in the target point cloud. As is clear from Figures 4(a), (b) and (c), our approach is able to find multiple correspondences between the template and the target point cloud, irrespective of the rotation of the template data and angular position of the targets within the point clouds.

We also ran an experiment to test the ability of our approach to provide robot odometry using the method described in Section 3.3. In order to do so, each consecutive 3D point cloud was registered with the previous one and a rotation, translation and scaling were derived, which also provided us with the yaw, pitch and roll. The obtained yaw, pitch, roll, and translation values were compared with those provided by the on-board sensors by plotting a route map for the robot as predicted by both data sources and also plotting the yaw, pitch and roll in a similar fashion. Since the robot was moving on a 2D surface, the

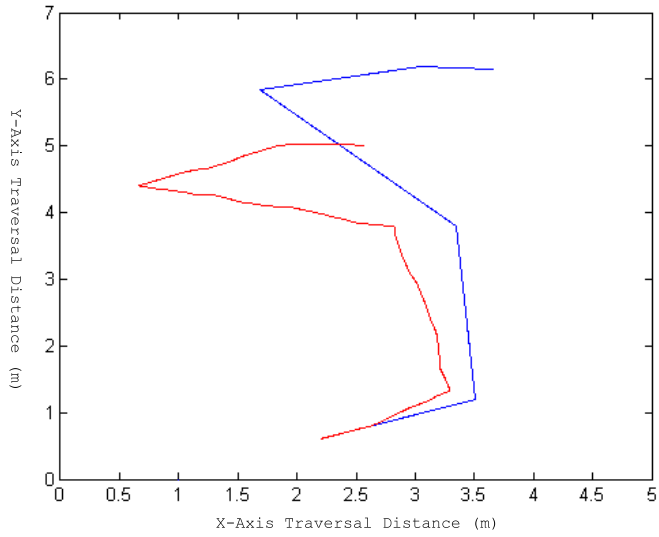


Fig. 5. Motion of the robot as predicted by the odometry derived from 3D point cloud registration vs. robot motion as obtained from navigational sensors (IMU, gyroscope and compass). Red line is from point clouds matching, while blue is the recorded robot odometry.

roll data was always constant. However, the odometry derived from the 3D point clouds closely matched that provided by the other sensors.

Figure 5 shows the path taken by our test robot. While the shape of the positional odometry derived from the 3D point cloud data is similar to the actual path, there is quite a lot of deviation between the two. However, this deviation can be explained by the fact that some of the frames that had multiple correspondences, like those in Figure 4, were not considered in the final result since the multiple locations of the correspondences led to errors. In the figure, the odometry seems more accurate, but it represents the ideal path, which does not consider wheel-spin. The difference between the resulting endpoints can reduced by applying a Kalman filter on the SIFT matching results. As such, these results could further be improved by applying some filtering, however, they clearly demonstrate the effectiveness of using 3D point cloud data for obtaining odometry information as well. The obtained results can at least be used as a basis for understanding robot trajectory.

Lastly, the other important performance criterion is the run-time performance of the algorithm in finding the correspondences between point clouds and then performing the corresponding registration. In our tests, the software was able to achieve a frame rate of 6.36 fps. We are confident that this could further be improved by adding optimizations methods, which were omitted in this version for the sake of simplicity and testing. However, this frame rate is within the range for real-time performance.

5 Conclusion

In our work we porposed a new feature extraction method for 3D data by applying the SIFT feature descriptor to 3D point clouds and scaling the data sets into images, which the SIFT descriptor could work with. The point clouds could then be registered based upon the correspondences of the feature points. This implementation makes it clear that the SIFT descriptor retains its robustness even when utilized with range data since the correspondences obtained appear to be visually correct. Moreover, even in highly noisy IR range data, multiple correspondences are successfully found in case of template matching.

The algorithm performs quite well in locating correspondences between point clouds and registering them with near real time performance. Furthermore, the preliminary experimental results suggest that even odometry data derived from calculating the relative translation, rotation and scaling between successive point clouds is close to being accurate, however, it may need further filtering and an implementation of Kalman filters to have the error component removed.

References

1. Bendels, G., Degener, P., Wahl, R., Koertgen, M., Klein, R.: Image-based registration of 3d-range data using feature surface elements. In: Proceedings of The 5th International Symposium of Virtual Reality, Archaeology and Cultural Heritage, VAST 2004 (2004)
2. Callieri, M., Cignoni, P., Ganovelli, F., Montani, C., Pingi, P., Scopigno, R.: Vclab's tools for 3d range data processing. In: Proceedings of the 1st EURO-GRAPHICS Workshop on Graphics and Cultural Heritage, Brighton, UK (November 2003)
3. Cernea, D.: Graphical methods for online surface fitting on 3d range sensor point clouds. Master's thesis, Jacobs University Bremen, Germany (August 2009)
4. Schall, O., Belyaev, A., Seidel, H.P.: Robust filtering of noisy scattered point data. In: Proceedings of Point-Based Graphics on Eurographics/IEEE VGTC Symposium, June 2005, pp. 71–144 (2005)
5. Lowe, D.: Distinctive image features from scale- invariant keypoints. International Journal of Computer Vision 60(2) (November 2004)
6. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of International Conference on Computer Vision, Corfu, Greece (September 1999)
7. Pisarevsky, V., et al.: Opencv, the open computer vision library (2008), `http://mloss.org/software/view/68/`
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. ACM Commun. 24(6), 381–395 (1981)
9. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vision 13(2), 119–152 (1994)
10. Gough, B. (ed.): GNU Scientific Library Reference Manual, 2nd edn. Network Theory Ltd. (2003)