Proceedings of Australasian Conference on Robotics and Automation, 3-5 Dec 2012, Victoria University of Wellington, New Zealand.

Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search

Sam Marden
University of New South Wales
s.marden@unsw.edu.au
Jose Guivant
University of New South Wales
j.guivant@unsw.edu.au

Abstract

Matching of 3D scans is performed using a modified version of the Iterative Closest Point algorithm with a grid data structure that provides constant time searching and insertion for approximate nearest neighbour searches.

The algorithm is shown to be faster and more accurate than traditional Iterative Closest Point scan matching using a kd-tree.

1 Introduction

Simultaneous Localisation and Mapping (SLAM) is a task that is relevant when other methods of localisation are not available or feasible in the context of the operation of the platform.

Although in the past there has been a great amount of research into this problem in 2 dimensions, it has only been relatively recently that there has been increased interest into tackling the problem in 3 dimensions. The main difficulty in this regard is the greatly increased amount of data present in 3D models.

Generally, scan matching is performed on a sequence of 3D scans. The most common method for doing so is the Iterative Closest Point algorithm (ICP) [Besl and McKay, 1992] (see section 2).

An alternative method that has been introduced more recently is the Normal Distributions Transform (NDT) [Biber, 2003]. Similar to an occupancy grid (the original paper focused on scan matching in 2 dimensions), the environment is subdivided into cells, each of which is assigned a normal distribution that locally models the probability of measuring a point. The result of this transformation is a piecewise continuous and differentiable probability density, which can be used to align another scan using Newton's algorithm. An important component of NDT is that it avoids computing explicit correspondences; it simply matches a point to the cell in which it lies.

In [Magnusson et al., 2009], ICP was compared with NDT for use in autonomously mapping a mine. The authors concluded that ICP was less error prone, but noticeably slower than NDT and, most importantly, that the main source of the difference in computation time was due to the fact that NDT does not have the bottleneck of ICP's expensive nearest neighbour search.

[Wulf et al., 2007] analysed the improvement gained by matching the current scan against all previous scans (so-called meta-scan matching), rather than only the one directly preceding it. It was shown that there is a marked improvement in the accuracy, but since a kd-tree was used for the nearest neighbour search the matching time increased significantly towards the end of the process, since by this time millions of points had been included into the k-d tree.

This paper combines several of these concepts, performing meta-scan matching using a data structure similar to that of the grid used in NDT. The use of a discrete grid for the nearest neighbour search has a number of advantages over the use of the traditional kd-tree method:

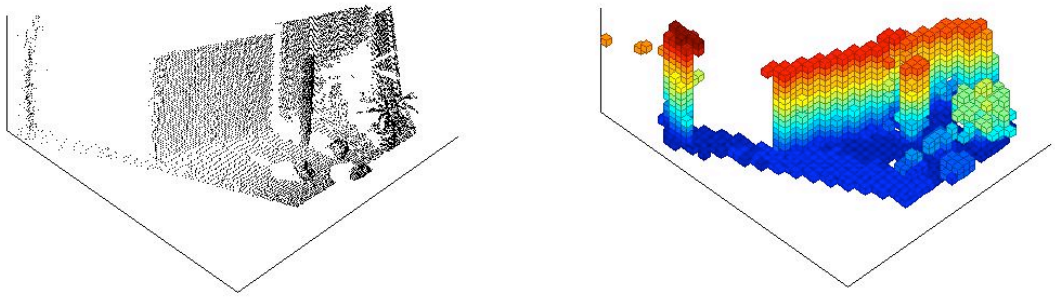

Figure 1: Point cloud collected from Kinect (left) and discrete grid representation (right) illustrating the nearest neighbour search data structure.

- Fast searching: due to the grid-like nature of the data structure, finding the approximate nearest neighbour of a point simply involves referencing an array based on the coordinates in question.
- Fast insertion of new scans: kd-trees have non-constant insertion time for new points, as well as large memory requirements, as opposed to the discrete grid (see section 4).

2 Iterative Closest Point

Here we briefly overview the Iterative Closest Point (ICP) algorithm for aligning point clouds. There has been a large amount of literature on different variations of ICP, such as point-to-point [Besl and McKay, 1992], point-to-plane [Chen and Medioni, 1992], or more recent variants such as Generalised ICP [Segel et al., 2009]. In this paper we only consider the point-to-point algorithm. This is comprised of two main steps:

- Find the nearest neighbour for every point of the first point cloud P in the second point cloud Q (denoted p_i and q_i respectively).
- Find the optimal rotation and translation such that the sum of the square of the distances between matched pairs of points found in the previous step is minimised.

These two steps are executed iteratively until some termination criterion is satisfied.

Thus, we can say that the aim of the algorithm is to find R and T that minimise the objective function:

$$f = \sum_{i=1}^{N} \left\| p_i - \left(Rq_i + T \right) \right\|^2$$

One method of solving for R and T is the singular value decomposition (SVD) method. We begin by calculating the centroids of the two point clouds:

$$\mu_P = \frac{1}{N_P} \sum_{i=1}^{N_P} p_i$$

$$\mu_Q = \frac{1}{N_Q} \sum_{i=1}^{N_Q} q_i$$

We now define:

$$p_i^{'} = p_i - \mu_P$$

$$q_i^{'} = q_i - \mu_Q$$

Next, we consider the following matrix W:

$$W = PQ^{\mathrm{T}}$$

Where P and Q are the concatenations of the individual points $p_i^{'}$ and $q_i^{'}$.

Now we consider the singular value decomposition of this matrix:

$$W = U\Lambda V^{\mathrm{T}}$$

It can be shown that the optimal solution for R and T that minimise the objective function are:

$$R = UV^{\mathrm{T}}$$

$$T = \mu_P - R\mu_Q$$

3 Nearest Neighbour Search

The nearest neighbour search of ICP is typically accomplished through the use of a k-d tree, which has a search complexity of O($logN$). The primary contribution of this paper, as stated previously, is the use of a simple grid data structure to lower this search time to O(1) for an approximate nearest neighbour search.

This nearest neighbour search is performed in the same manner as in the aforementioned Normal Distributions Transform.

In order to construct the grid, the 3D space of the point cloud is subdivided into cells, and for each of these cells the following is done:

- Collect all points p_n in the cell C_{ijk}.
- Calculate the number of points N_{ijk} in the cell. This step is necessary in order to easily update the grid based on a newly matched scan. This is discussed later in the next section.
- Calculate the mean μ_{ijk} of points in the cell.

In addition to this, we also calculate the minimum and maximum values x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}. If the side length of each cell is l, then the reference indices of any given point in the grid are given by:

$$i = \left\lfloor \frac{x - x_{min}}{l} \right\rfloor$$

$$j = \left\lfloor \frac{y - y_{min}}{l} \right\rfloor$$

$$k = \left\lfloor \frac{z - z_{min}}{l} \right\rfloor$$

Each point in another point cloud is now matched to the mean of the cell in which it lies as an approximate nearest neighbour when searching the grid. These means are treated as the target point cloud when the scan matching of ICP is applied. The SVD method discussed in the previous section is then used to calculate the optimal rotation and translation.

4 Meta-scan Matching

An important feature of the method presented in this paper is the ability to easily insert new scans into the existing data structure.

This has the effect of greatly reducing one of the main issues faced by scan-matching methods, which is that over time errors between successive matching of scans tend to build up, resulting in a large amount of drift.

Although it is possible to accomplish insertion of new points with other data structures, it can become computationally expensive to do so. In the case of the discrete search grid, the time required is dependent only on the number of new points being added to it. Additionally, the memory requirement of the discrete grid is proportional to the size of the environment, and is independent of the the number of points. Future work will analyse a sparse implementation of the data structure, since much of the data structure is actually empty due to the sparsity of the environment.

The process for adding the current scan to the existing grid is as follows:

- For each cell C_{ijk} in the current grid, calculate the mean $\mu_{ijk,new}$ and number of points $N_{ijk,new}$ based on the points in the point cloud to be added, extending the size of the grid if necessary.
- Update the mean and number of points in the grid:

$$N'_{ijk} = N_{ijk,existing} + N_{ijk,new}$$

$$\mu_{ijk} = \frac{\mu_{ijk,existing} N_{ijk,existing} + \mu_{ijk,new} N_{ijk,new}}{N_{ijk,existing} + N_{ijk,new}}$$

5 Implementation and Results

5.1 Overview

The algorithm has been implemented in MATLAB for evaluation. For comparison with existing methods, the Point Cloud Library (PCL) [Rusu and Cousins, 2011] implementation of point-to-point ICP has been used. This uses FLANN [Muja and Lowe, 2009] for the nearest neighbour search.

The tests have been run on a Late 2008 Macbook with a 2GHz processor and 4GB of RAM.

A side length of 8cm has been used for the cells of the discrete grid (see figure 1), whilst a maximum correspondence distance of 25cm has been used for the PCL implementation. Both algorithms terminate based on the threshold transformation epsilon criterion of PCL, which is defined as the sum of the absolute differences between consecutive rotations R and translations T. Once the transformations fall below this threshold, the scan matching is considered complete (this value has been set to 5×10^{-5} for testing). Additionally, there is a maximum of 200 iterations of ICP in both cases.

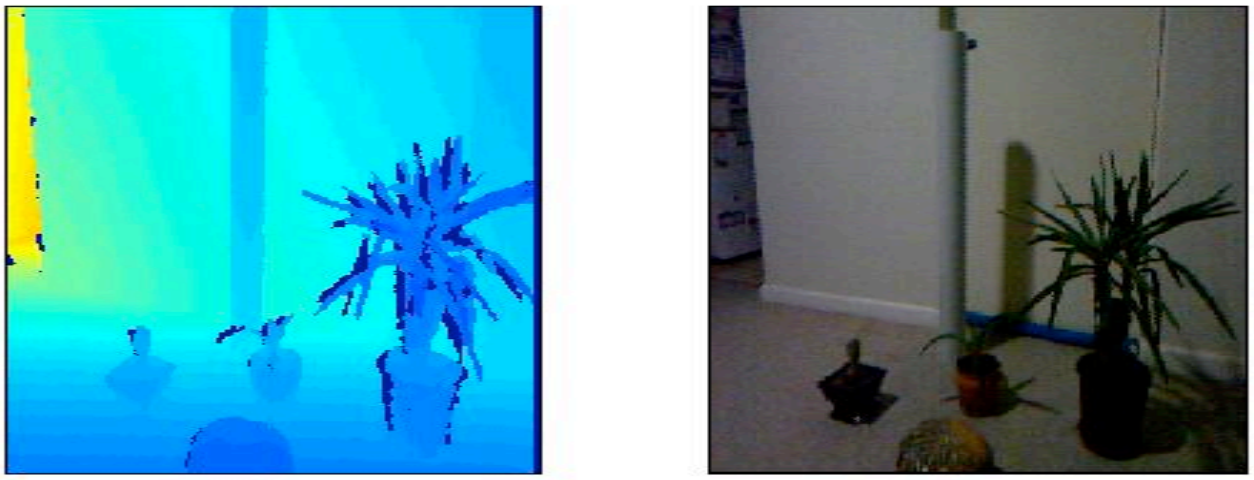

Figure 2: Example depth (left) and RGB (right) images collected from the Kinect camera.

5.2 Data Set

The system has been tested on data collected from a Kinect camera at 6 frames per second. In this data set, the Kinect was moved around a room by hand. Within the field of view of the camera are the walls and floor of the room, in addition to a series of obstacles including pot plants and a pole.

An example depth and RGB image collected from the Kinect can be seen in figure 2.

In addition to this, two inertial measurement units were attached to the Kinect, providing 3D accelerometers, gyroscopes and magnetometers. Only the gyroscope readings have been used in testing, in order to provide an initial estimate for rotation.

This data set is publicly available at www.possumrobot.com/Datasets/PlantasB004.htm.

Note also that the scans have been down-sampled 5-fold, reducing the number of points in each cloud to 61440 (minus points where no reading is measured).

5.3 Accuracy

The results of scan-matching using the data set discussed above are illustrated in figure 3.

In order to illustrate the drift that builds up over time, the first 1/3 of the frames are coloured red, the next 1/3 are coloured green, and the final 1/3 are coloured blue.

We can clearly see that there is a much larger amount of drift in the instance where meta-scan matching is not used.

Future work will focus on more rigorous benchmarking and evaluation against ground truth.

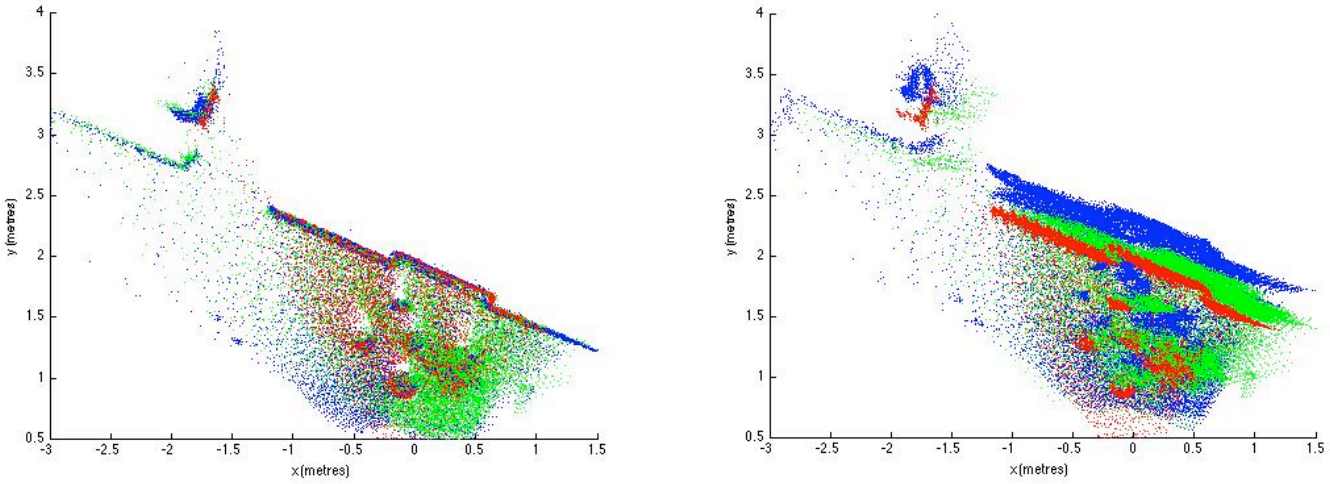

Figure 3: Matched point clouds using discrete grid and meta-scan matching (left) and matched point clouds using PCL (right).

5.4 Speed

Figure 4 below illustrates the time taken for scan matching for the method of this paper (both pair-wise and meta-scan matching) and for the PCL implementation that uses FLANN for the nearest neighbour search.

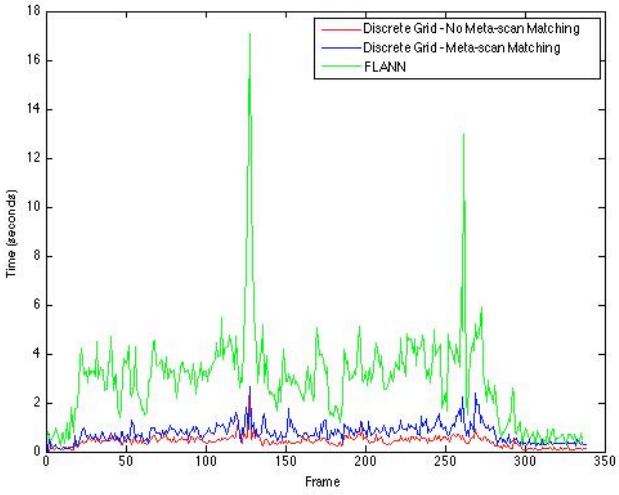

Figure 4: Comparison of scan matching times for different nearest neighbour search methods.

We observe that the discrete grid search is noticeably faster than the PCL algorithm.

Another important result to note is that the meta-scan matching using the discrete grid is only marginally slower than pair-wise scan matching using the discrete grid, emphasising the fact that integrating meta-scan matching with the proposed framework has a minimal effect on speed.

5.5 Size of Cells

The selection of the size of the cells used in the discrete grid is an important factor in the performance of the algorithm. Cells that are too small result in many points not being assigned a neighbour at all, resulting in ICP converging to an incorrect local minimum. Cells that are too large cause an excessive level of simplification of the structure of the original point cloud, resulting in a loss of fine detail, so much so that this can also cause convergence to an incorrect local minimum. In the current implementation of the algorithm the size of the cells has been determined empirically.

Another factor to consider is that as aforementioned the complexity of construction and insertion into the grid, as well as the memory requirement, scales with the size of the environment. When the cell size becomes very small, the main bottleneck in computation becomes updating the data structure, rather than actually using it.

Originally the author of NDT proposed the use of a series of overlapping grids offset by some amount in order to minimise the effects of the discretisation on the environment. [Magnusson et al., 2009] also discusses a series of discretisation regimes for use with NDT. A possible avenue for future work is the analysis of the effect of these methods as well as the cell size on the performance of the algorithm.

6 Conclusion and Future Work

This paper presents a grid-like data structure that is used for an approximate nearest neighbour search within the framework of the Iterative Closest Point algorithm. This is a natural extension to ICP, as it already assumes a relatively good initial alignment, and point pairings that are greater than a certain distance away from one another are rejected in most implementations.

The proposed algorithm has been shown to greatly improve the performance of the ICP algorithm, in terms of both accuracy and speed.

There are several possibilities for future work:

- In the current implementation, estimates from the IMU are required for successful scan matching. Feature extraction methods operating the RGB images (which is currently not utilised at all) collected from the Kinect will be investigated as a means of generating initial estimates, as well as being combined into the scan matching process itself.
- As discussed above, there are many potential variations of the data structure that can be investigated. Future work will focus on the analysis of these techniques, as well as a thorough evaluation of the performance on data sets with ground truth available. Additionally, the algorithm will be tested against other recent varieties of RGB-D scan matching.
- Due to the heavily vectorised nature of the operations involved in the algorithm (searching, construction of, and insertion into the data structure), it is a strong candidate for a parallel implementation utilising modern GPUs. [Newcombe et al., 2011] has shown that a GPU implementation of point cloud alignment is capable of very impressive results, yielding a system that can process the entire point cloud from a Kinect at full frame rate.

6 Acknowledgements

Acknowledgement to the UNSW UGV team, UNSW Mechatronics group and the UNSW School of Mechanical and Manufacturing Engineering for the provision of data sets and facilities for testing.

References

[Besl and McKay, 1992] Paul J. Besl and Neil D. McKay. A method for the registration of 3-D shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14(2):239--256, February 1992.

[Biber, 2003] P. Biber. The Normal Distributions Transform: A New Approach to Scan Matching. In *Proceedings of the International Conference on Intelligent Robots and Systems.*

[Chen and Medioni, 1992] Yang Chen and Gerard Medioni. Object Modelling by Registration of Multiple Range Images. In *Proceedings of the IEEE International Conference on Intelligent Robots and Automation*, pages 2724--2729, 1991.

[Newcombe at al., 2011] Newcombe, Richard A., Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2011.

[Magnusson et al., 2009] Martin Magnusson, Andreas Nuchter, Christopher Lorken, Achim J. Lilienthal, and Joachim Hertzberg. Evaluation of 3D Registration Reliability and Speed – A Comparison of ICP and NDT. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2009*, pages 3907--3912, May 2009.

[Muja and Lowe, 2009] Marius Muja and David G. Lowe. Fast Approximate Nearest Neighbour with Automatic Algorithm Configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP'09)* 2007.

[Nuchter et al., 2007] Andreas Nuchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. 6D SLAM – 3D Mapping Outdoor Environments. Journal of Field Robotics 2007.

[Rusu and Cousins, 2011] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PSL). In *IEEE Conference on Robotics and Automation (ICRA), 2011.*

[Segal et al., 2009] Aleksandr V. Segal, Dirk Haehnel, and Sebastian Thrun. Generalised ICP. In *Proceedings of Robotics: Science and Systems (RSS), 2009.*

[Wulf et al., 2007] Oliver Wulf, Andreas Nuchter, Joachim Hertzberg, and Bernando Wagner. Ground Truth Evaluation of Large Urban 6D SLAM. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 650--657, October – November 2007.