



How do ICP variants perform when used for scan matching terrain point clouds?

F.A. Donoso ^{*}, K.J. Austin, P.R. McAree

School of Mechanical and Mining Engineering, The University of Queensland, QLD 4072, Australia



HIGHLIGHTS

- No single variant satisfies all performance criteria.
- The performance of ICP based terrain mapping vary significantly across ICP variants.
- A judicious selection of the ICP variant is imperative to meet a minimum performance.
- The performance of ICP variants depends on the scene.
- Computational efficiency is not traded for computational accuracy or precision.

ARTICLE INFO

Article history:

Received 17 March 2016

Received in revised form 29 September 2016

Accepted 7 October 2016

Available online 14 October 2016

Keywords:

Iterative closest point

Terrain mapping

Point cloud registration algorithms

ABSTRACT

Many variants of the Iterative Closest Point (ICP) algorithm have been proposed for registering point clouds. This paper explores the performance of 20,736 ICP variants applied to the registration of point clouds for the purpose of terrain mapping, using data obtained from a mobile platform. The methodology of the study has involved taking sequences of 100 consecutive scans at three distinct scenes along the route of a mining haul truck operating in a typical surface mining environment. The scan sequences were obtained at 20 Hz from a Velodyne HDL-64E mounted on the truck. The aim is to understand how well the ICP variants perform in consolidating these scans into sub-maps. Variants are compared against three metrics: accuracy, precision, and relative computational cost. The main finding of the paper is that none of the variants is simultaneously accurate, precise, and fast to compute, across all three scenes. The best performing variants employed strategies that filtered the data sets, used local surface geometry in the form normals, and used the distance between points in one point cloud to a corresponding surface from a reference point cloud as a measure of the fit between two point clouds. The significance of this work is that it: (i) provides guidance in the construction of ICP variants for terrain mapping; and (ii) identifies the significant limitations of existing ICP variants for this application.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Scan matching is the name given to the problem of finding the transformation that aligns two or more point cloud scans recorded at different locations. Methods to minimize the alignment error belong to a family of “registration” algorithms that find application in many field robotic systems and beyond. A well known mapping application is that used by the Google car [1] that makes use of scanning LiDAR mounted to the car to build maps that are used for navigation.

Similarly, in recent years Caterpillar Inc. has developed an autonomous mining haul truck that also uses scanning LiDAR for navigation [2,3]. There also seems to be significant value in using the LiDAR data collected from mobile equipment to construct terrain maps in real time in order to monitor the continual changes that are made to the environment as material is mined. This requires, among other things, the ability to consolidate point cloud data from the sequence of scans generated by the sensor into a common frame of reference, also known as scan matching.

The general problem of matching scans has a history spanning over 25 years, see for example [4–10]. Approaches are usually based on geometry registration algorithms such as the Iterative Closest Point (ICP) method [11,12]. ICP is used to compute the transformation that brings two point clouds into “best” alignment by a two step process: (i) *correspondence*, the matching of overlapping data across the point clouds; and (ii) the *minimization* of a

* Corresponding author.

E-mail addresses: felipe.donoso@uq.edu.au (F.A. Donoso), kevin.austin@uq.edu.au (K.J. Austin), p.mcabee@uq.edu.au (P.R. McAree).

metric describing misalignment. In most applications these steps are iterated to improve alignment.

The ICP method is simple to put into practice and efficient, particularly when implemented with *kd*-trees for searching the point clouds. However, the application of “vanilla” ICP, as described in [11] or [12], produces less-than-optimal matching across a range of applications including terrain mapping. Reasons include points from consecutive scans do not map one-to-one, each new scan covers a spatial region different to that in the previous scan, and in the region of overlap, different terrain points are sampled. To improve performance many variants of ICP have been proposed that curate the raw point cloud data in various ways to improve the match. However, it is not clear which variant should be used in any specific application. It is not unusual for an ICP variant that performs well on one set of data to perform poorly on other, seemingly similar, data.

Several prior studies compare the performance of different ICP variants. Rusinkiewicz et al. [13] decomposed various adornments and decorations applied to “vanilla” ICP into a six-stage computational process to examine the convergence speed and accuracy of different strategies. The study observes that different variants perform better on different point clouds and recognizes the need for deeper insight into the scan matching algorithms. The idea of adaptively choosing variants, depending on point cloud characteristics, is proposed.

Salvi et al. [14] present a survey of coarse and fine scan matching methods focusing on the accuracy of the match. Methods investigated include: (i) the addition of artificial Gaussian noise; (ii) varying the number of point (sub-sampling); and (iii) varying the percentage of outliers included. They found that point-to-plane with rejection of paired points [12] provided the best performance in terms of both accuracy and computational time.

Pomerleau et al. [15,16] present a survey of scan matching algorithms for mobile robotics. They identified the lack of a comparison framework as an issue for selecting the particular ICP variant best suited for a given scene. Several use cases for ICP-based generation of three-dimensional maps were considered in order to explore the tuning required of a ‘standard’ ICP algorithm to meet the registration challenges of each scenario. These challenges include different environments, variable amounts of overlap in the point clouds, dynamic scenes, and real-time processing. This work provides high level guidance on the implementation, tuning, and testing of ICP registration algorithms.

They show that the performance of the baseline variants vary significantly with different data sets and conclude the need for better ICP variants for natural, unstructured and information-deprived environments. Surface mining environments exhibit these attributes.

There are many other forms of scan registration applied to robotic selflocalization and mapping, see for example [17–20]. Probabilistic registration algorithms map distributions to the point cloud measurements, e.g. a likelihood function composed of a Gaussian describing the expected range with a variance set by the sensor properties. An example of a probabilistic registration method is the *Normal-Distribution Transform* (NDT) which was first introduced by Biber et al. [21] for scan matching in two dimensions. The algorithm assumes that the point cloud is comprised of normally distributed points belonging to patches. A complete review of the most important probabilistic registration methods, focusing on NDT, is presented in [22].

This paper focuses on the application of ICP to terrain mapping and how we might make sense of the many variants available and understand what gaps there are in the application of ICP. A comprehensive set of 20,736 ICP variants are applied to different terrain data scenes and evaluated for their accuracy, precision, and relative computational cost.

The accuracy of ICP variants is evaluated by comparing the RMS distance error between the map generated by scan matching, using an ICP variant, and a ground-truth obtained by using an accurate GNSS–IMU navigation system to register consecutive scans. Precision is evaluated by calculating the deviation of RMS error for each scan from a straight line fit of first and last scan RMS error. Relative computational cost is expressed in the form of a ranking through the measure of computation time relative to the fastest variant among the population of ICP variants.

The following sections provide a review of the methods that might make up an ICP variant, details the evaluation framework and data sets, and presents comparative results that explore the performance of ICP variants in relation to the different terrain scenes and the different performance metrics.

2. Variants of the iterative closest point algorithm

Rusinkiewicz et al. [13] consider ICP-based scan matching to consist of six distinct computation stages with the possibility for using different methods or combinations of methods at each stage. The starting point of this paper is to adopt Rusinkiewicz and Levoy’s decomposition and adapt it to accommodate the new algorithms and ideas which have emerged since their study was published.

The computational decomposition used in this work is shown in Fig. 1. The inputs to the computation are two point clouds, P^n denoting the input point cloud of size n , and Q^m denoting the reference point cloud of size m . For the map building task of this paper the two point clouds are drawn from consecutive scans, S_i representing the input and S_{i-1} representing the reference, where the index i refers to the scan number. Fig. 1 has six stages:

1. **Point selection:** Data reduction involving the preferential selection of a set of points from the input point cloud for scan matching.
2. **Neighbourhood selection:** Establishes a region around each point to determine features associated with the point, e.g. normals.
3. **Point matching:** Pair points of the input point cloud to those of the reference point cloud.
4. **Weighting:** Assigns weights to matched point pairs.
5. **Rejection:** Discard point pairs that do not contribute positively to minimization.
6. **Minimization:** The matching of scans by minimization of a metric to bring the input point cloud into alignment with the reference point cloud.

The stages fall into two distinct types. The first, comprising *point selection* and *rejection*, correspond to stages that filter the input and can be implemented by methods applied alone or in combination with other methods, with each method applying an additional layer of filtering to the data.

The second type, comprising *neighbourhood selection*, *matching*, *weighting*, and *minimization*, are implemented by one strategy taken from a set of alternative algorithms. The following sections provide a brief review of available methods to provide context for their application to the particular map building scenarios of this paper. The methods selected to be part of the ICP performance evaluation are listed in Table 1.

2.1. Point selection

The first stage of the ICP computation process seeks to reduce the input (and reference) point clouds by application of one or more filters. Point selection methods are applied to reduce the number of points used at subsequent stages and to improve the

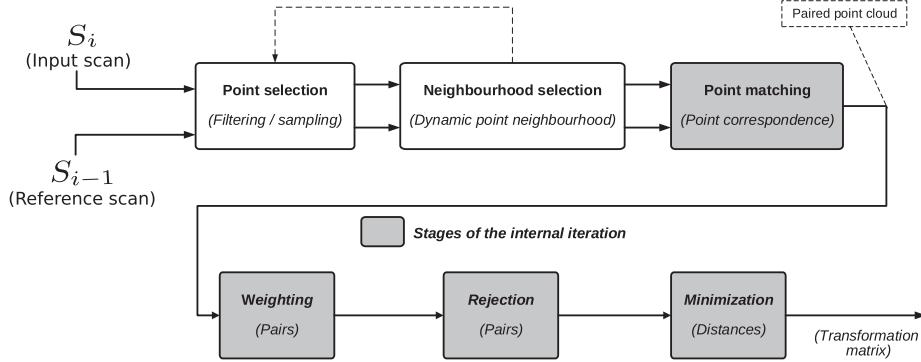


Fig. 1. The ICP pipeline – a decomposition of the computation stages of the ICP algorithm.

characterization of the underlying data through a meaningful selection of points that encourage fast and accurate convergence to the correct solution. Reducing the number of points provides a practical consideration for timely delivery of an ICP solution, particularly when handling very dense data sets.

It is noted that the points selected for processing at future stages must be chosen judiciously, with there being potential to exclude points useful to the correspondence and minimization stages by poor point selection.

In early work, Besl et al. [11] proposed that point selection is not needed when there is considerable overlap between point clouds and the number of outliers is not significant. However, for dense point clouds the pragmatic need to reduce the number of points arises in order to make computation times acceptable. Sub-sampling methods proposed to manage dense point clouds include: random [23] and uniform [12,24] sampling. These basic point selection strategies are sub-optimal, deliver slow convergence, and may lead to divergence of the scan matching algorithm. For this study we limit consideration to what might be called discerning algorithms for point selection.

Discerning point selection algorithms typically involve analysis of distinctive attributes or aspects (that is, features) of the point cloud to achieve a well-judged selection of points. It is known that if too many points are chosen from featureless regions, the scan match may converge slowly, converge to a wrong solution (corresponding to a local minima), be unconstrained so that it “slides”, or diverges [25].

The normal-space sampling method [13] attempts to eliminate solution sliding by selecting a subset of points whose point-normals are as widely and uniformly distributed as possible over the unit sphere. The idea of this method is to reduce translational instability that leads to a better and faster convergence compared with uniform sampling.

Gelfand et al. [25] extend these ideas, giving prominence to the notion that selected points should constrain the alignment so that meshes converge quickly and accurately during ICP minimization. They termed such point selections geometrically stable. The method estimates the transformations that can cause unstable sliding and selects the points that best constrain these potentially unstable transformations. Only points in the estimated region of overlap between matched scans are selected. Gelfand et al. [25] argue that the approach produces faster and more accurate scan matching than [13], whilst stabilizing sliding in all spatial dimensions (not just translation). However, Torsello et al. [26] criticized the method for its tendency to introduce artificial constraints in the presence of noisy point clouds.

Torsello et al. [26] propose relevance-based sampling, to overcome the introduction of artificial constraints introduced by noise. This approach uses the average local radius of curvature as a distinctiveness measure. A process of integration is used to obtain the measure, improving the sampling robustness to noise.

Torsello et al. [26] present a favourable comparison of the method to uniform and normal space sampling, showing convergence to a better fit for different levels of noise. Relevance-based sampling is oriented to computer modelling and is not suitable for large terrain point clouds with nonuniform point densities.

Gressin et al. [27] propose two different point selection methods based on a combination of eigenvalues of the covariance matrix. The first method is based on the dimensionality (linear, planar and scattered) by selecting points with linear behaviours. The second method aims to select points with higher entropy feature values.

Outlier removal via distance threshold has proven to be one of the most commonly used and effective filtering approaches. The method seeks to remove points that do not contain a nearest neighbour in a specified distance threshold. A similar approach is filtering points with density lower than a given threshold [28], such that a constant density is maintained throughout a point cloud. Other point selection methods include the use of surface characterization by kernels such as the kernel density filter [29] which is based on normals and eigenvalues of the covariance matrix.

2.2. Neighbourhood selection

An accurate representation of point cloud geometry influences both the matching and minimization stages of the registration process. Neighbourhood selection is used to determine the set of points necessary to accurately describe the underlying surface geometry, through, for example, the calculation of surface normals. Selected neighbourhoods are used in *point selection* and for calculation of features.

The simplest approach to neighbourhood selection is to use a fixed radius or number of points to select the neighbourhood for each point in the point cloud. However, when the density of points varies significantly with the distance to the sensor origin, the selection of the neighbourhood size requires a trade-off in the representation of near and far features. A small neighbourhood may not provide a suitable representation of geometry distant from the sensor origin, and a large neighbourhood may distort the representation of geometry close to the sensor.

To mitigate the effects of point density variation the neighbouring points can be dynamically selected based on a local metric. Lalonde et al. [30] suggest choosing a neighbourhood size of radius, r , that minimizes the expected angular deviation of the computed normal of a point from its true normal. The approach builds on the work of [31] for estimating normals in a noisy point cloud. The basic idea is that the optimal normal vector is bounded by an expression depending on the noise of the point cloud, the curvature of the underlying manifold, the density of points, and the neighbourhood size. A complete derivation of the method is found in [31]. The method of [31] and [30] are referred to as “bounded radius” methods.

Point geometry is important to determining the size of the point neighbourhood. The extent of the neighbourhood has to be chosen to appropriately preserve one geometrical feature over others. Taking that into consideration, Demantké et al. [32] address the problem of dynamic neighbourhood selection by evaluating the entropy feature over a varying radius and selecting the radius that minimizes the entropy feature. A low entropy feature is indicative of a dominant dimensionality, e.g. linear, planar or scattered.

Wiemann et al. [33] propose a simple approach to cope with low density regions by taking into account the shape of bounding boxes enclosing a neighbourhood. If the shape is elongated, with insufficient points, the neighbourhood has to be enlarged until the bounding box is square like.

2.3. Point matching

Point matching produces a correspondence between points from the input point cloud, \mathcal{P}^n , and the reference point cloud, \mathcal{Q}^m . The output of the process is a pairing of points that is used as the basis for minimizing the misalignment between point clouds.

Correspondence is achieved by finding the closest point in the input cloud to each point in the reference point cloud. There is no explicit requirement for uniqueness in the correspondence relationships, however, iteration of the ICP algorithm should increase the number of unique closest points. The definition of “closest point” is the defining characteristic of the scan matching method. Besl and McKay prescribe a closest point strategy based on the Euclidean distance between points, which remains the principal method applied to fine registration.

To complete an exhaustive search between all point of \mathcal{P}^n into \mathcal{Q}^m attracts a computational cost of $\mathcal{O}(nm)$. Nearest-neighbour-search (NNS) methods are used to reduce the computational load associated with point matching. A comparison of NNS strategies applied to scan matching is provided in [34], identifying that kd-trees are an effective means to find nearest neighbours.

Heuristics that constrain pair correspondence are sometimes applied to improve the robustness of the NNS by reducing the matching of unrelated points. Pulli et al. [35] proposes a constraint based on the difference between normal angles, only allowing the matching if the difference of normal angles from both points is less than 45° . A similar approach is used by [36] to match points only if their intensity compatibility is greater than a given threshold. The application of constrained correspondence methods is similar to the pair rejection methods discussed in Section 2.5.

Further enhancement of the nearest neighbour search is made possible by using low dimension descriptors of the point cloud geometry as a basis for point matching. The implementation is typically realized through the definition of an enhanced distance measure, $d(\mathbf{p}_j, \mathbf{q}_k)$, that is a weighted function of the properties attributed to the two points, \mathbf{p}_j and \mathbf{q}_k .

$$d(\mathbf{p}_j, \mathbf{q}_k) = \alpha d_e(\mathbf{p}_j, \mathbf{q}_k) + \beta d_{f_1}(\mathbf{p}_j, \mathbf{q}_k) + \gamma d_{f_2}(\mathbf{p}_j, \mathbf{q}_k), \quad (1)$$

where d is the enhanced distance, d_e is the Euclidean distance from a point \mathbf{p}_j of the input, to a point \mathbf{q}_k on the reference, d_{f_1} and d_{f_2} are feature distances associated with \mathbf{p}_j and \mathbf{q}_k , and α , β and γ are weights applied to d_e , d_{f_1} and d_{f_2} , respectively. The weightings are usually set empirically through trial and error.

Feldmar et al. [37] present a similar enhancement to the nearest neighbour search with the application of normal vectors. With their approach, the search space for point matching grows from three to six dimensions. Contributions from [38,39], and [40] cover the application of curvature, moment invariants, spherical harmonic invariants, colour, and intensity as descriptors applied to the matching process.

An alternative to using the distance across points is to use high dimensional features of the point cloud geometry to drive point

correspondence [41]. Such methods are often applied to coarse registration due to their robustness to a large initial misalignment between point clouds. Their application to fine registration is not explored in this paper.

2.4. Weighting

The weighting of matched pairs uses local or global contextual information to modify the distance function associated with each pair. The intent is to influence the individual contribution of matched pairs to the minimization process thereby improving the ICP performance.

Weighting by distance [36] assigns lower weights to pairs with greater separation. Another approach is to weight according to the scalar product of the normal vectors associated with the paired points, reducing the contribution of those pairs with disparate normals [13]. These strategies are applied in this study along with the baseline of constant weighting applied to all matched pairs. Given previous studies [13,27], the expectation is that pair weighting will not be a significant factor in the performance of ICP variants.

Khoshelhama et al. [42] propose a weighting method based on the variance in the depth axis of an image given by a Kinect sensor. This approach is not suitable for the LiDAR data of this study as the approach requires the variance to be equal in all axes.

2.5. Rejection

Pair rejection extends the pair weighting operation by discarding those pairs that disrupt the minimization of the distance function. This process seeks to improve the convergence of the algorithm by eliminating pairs representing “false positives”, points without overlap, or point pairs that are outliers. As such, the rejected pairs do not contribute, in their number or in their error, to the distance error function used for minimization (see Section 2.6).

Besl et al. [11] did not include the rejection of matched pairs as a formal stage of the ICP algorithm. They did however identify that outliers and occlusions negatively impact the performance of ICP and identified the mitigation of their affect as an area of future work.

The Euclidean distance of the paired points provides a simple and powerful way to identify outliers or occluded points [24]. The simplest approach is to use a fixed threshold distance, across all points and all iterations, as a basis for rejecting points. This approach has significant limitations due to variations in point cloud geometry, poor robustness to different point matching scenarios, and the global reduction in average distance as the solution converges.

Zhang [24] proposed an adaptive distance threshold based on the mean and variance of the distances between pairs. In their approach a target optimal average error is used to dynamically calculate a distance threshold as a function of the mean and variance across all point pairs. An alternative formulation [13,35] applies a threshold percentage to identify the worst pairs ordered by distance. The threshold percentage approach is robust, but the appropriate setting of the threshold percentage is dependent on the type of registration problem. For the fine registration example problem used in this study, a 10% threshold for pair rejection was applied to all scenes, determined through trial and error.

Point normal comparison [13] provides a similar rejection method as the threshold distance. A large difference in the normals of matched points suggest that points do not share the same local geometry, providing a means for rejection. In this study, a fixed 5° angle threshold is used.

2.6. Minimization

The last step of the ICP algorithm is to minimize a measure of the fit between the two point clouds given a set of match point

pairs. A distance function is used to describe the fit, and typically takes the form of point-to-point or point-to-plane distances.

The point-to-point distance corresponds to the minimum distance between a query point $\tilde{\mathbf{p}}_j$ from the input cloud to a point \mathbf{q}_j in the reference point cloud, where ' j ' denotes the pair index. Computing the average point-to-point distance over all (N) match point pairs provides the following quadratic error function,

$$E(R, \mathbf{t}) = \frac{1}{N} \sum_{j=1}^N w_j \left\| \mathbf{q}_j - \underbrace{(R\mathbf{p}_j + \mathbf{t})}_{\tilde{\mathbf{p}}_j} \right\|^2, \quad (2)$$

where R is the rotation matrix and \mathbf{t} the translation vector that together are used to align the two point clouds, N is the number of matched point pairs and w_j is a weighting factor for pair j . The aim of the ICP algorithm is to find the R and \mathbf{t} applied to \mathbf{p}_j that minimize the distance to \mathbf{q}_j . The relative contribution of point pairs to the minimization task is set by the weighting w_j , see Section 2.4.

Nüchter [43] examines different methods to solve Eq. (2). The singular value decomposition is used to compute the transformation, based on considerations of simplicity and performance.

The point-to-plane distance corresponds to the closest distance from a query point $\tilde{\mathbf{p}}_j$ in the input cloud to a plane representative of the local geometry around the point \mathbf{q}_j in the reference point cloud. The quadratic error function for the set of (N) match pairs is then expressed as,

$$E(R, \mathbf{t}) = \frac{1}{N} \sum_{j=1}^N w_j \left\| (\mathbf{q}_j - \underbrace{(R\mathbf{p}_j + \mathbf{t})}_{\tilde{\mathbf{p}}_j}) \cdot \hat{\mathbf{n}}_j \right\|^2, \quad (3)$$

where the point-to-plane distance is expressed by the dot product of the unitary normal vector of the plane, $\hat{\mathbf{n}}_j$, and the distance between $\tilde{\mathbf{p}}_j$ and \mathbf{q}_j .

The rotational matrix, R , is a non-linear function of the rotational angles α , β and γ of the three coordinate axes x , y and z , respectively. It is assumed that rotational angles will be small, thus the $\cos(\theta)$ can be approximated to 1 and the $\sin(\theta)$ to θ . This is a reasonable assumption for closely spaced scans (i.e. consecutive scans from a fast LiDAR). The linear equation obtained with the above approximation is of the form $A\mathbf{x} = \mathbf{b}$.

Segal [44] presents the Generalized-ICP as a generalization of the *total least squares* algorithm formulated by [45]. ICP point-to-point and point-to-plane assume that points of both input and reference clouds have isotropic and identical probability distributions. Generalized-ICP minimization assumes that points of both point clouds are locally Gaussian distributions. Maximum-likelihood estimation is used to iteratively compute the transformation T (formed through the combination of R and \mathbf{t} , minimizing the distance function, d).

The minimization function for Generalized-ICP could be based on either point-to-point or point-to-plane representations. However, no examples of using a point-to-point distance function for Generalized-ICP were identified in the literature. Therefore, the Generalized-ICP formulation in this work is based only on the point-to-plane distance function.

3. Evaluation framework

The methodology we adopt in this study is to take a selection of strategies for each stage in the ICP computation, and perform an evaluation of the ICP variants formed through various combinations of the composite algorithms. The large number of potential alternatives at each stage mandates judicious selection of those algorithms to be compared. For reasons of practicality the study is limited to those methods either in common use or, where many alternatives are available, those thought to be in some sense superior.

Table 1
Selected strategies for performance comparison.

Stage	Code	Strategy
Points selection	a	All points [11]
	b	Outliers removal filter [46]
	c	Density filter [28]
	d	Geometrically stable sampling [25]
	e	Entropy feature filter [27]
	f	Dimensionality based selection [27]
Neighbourhood selection	1	Constant
	2	Entropy feature minimization [32]
	3	Density adaptation [33]
	4	Bounded radius [47]
Point matching	x	Nearest neighbours (NN) [11]
	y	NN enhanced by moment invariants [38]
	z	NN enhanced by normals [37]
Weighting	0	Constant [11]
	1	Distance [36]
	2	Normals compatibility [13]
Rejection	α	No rejection [11]
	β	Distance by worst percentage [13]
	γ	Angular deviation [13]
	δ	Adaptive distance by variance [24]
Minimization	0	Point-to-point [11]
	1	Point-to-plane [12]
	2	Generalized [44]

3.1. Selected methods and their implementation

Table 1 summarizes the methods used to compose the ICP variants. In total there were 20,736 variants examined. ICP variants are identified by an alphanumeric code where each letter or number represents a different method. The code de4z1 α 1, for example, employs: (i) geometrically stable sampling (code d) and entropy feature filtering (code e) for point selection; (ii) bounded radius (code 4) for neighbourhood selection; (iii) nearest neighbour enhanced by normals (code z) for point matching; (iv) distance (code 1) weighting; (v) no rejection (code α); and (vi) point-to-plane ICP (code 1) for minimization.

The methods selected for evaluation have been implemented in a bespoke software framework written in the C++ language for the purpose of comparative evaluation. The implementation provided a single source for selected methods, allowing the combination of methods to be easily controlled – no such source for all methods exists in the public domain. Standard strategy and decorator design patterns [48] were employed to facilitate efficient use of code and allow dynamic control over the combinations of methods that compose an ICP variant. Implemented methods were verified against various public domain codes, including: Point Cloud Library (PCL) [49], an open-source library of algorithms for point cloud processing; the 3D Toolkit (3DTK) [43], an open-source library specializing in 6D-SLAM; and the Robot Operation System (ROS) [50], open-source libraries for creating robot applications.

The parameterization of each method was determined using a training data set and incorporating the method in a “plain” ICP point-to-point variant. The parameters chosen were those that provided the best scan registration for each scene. This process, and the ICP analysis that followed identified that the optimal set of parameters for a method depends on the geometry of the scene, and a configuration that works well on one scene may not work well with another scene. The dynamic modification of parameters based on the geometry of the point cloud should be a consideration for further studies.

The point selection methods examined are implemented through a decoration pattern [48] that allows different combinations of methods to be explored. The order in which methods are applied is as follows: *outlier removal* method is applied first to

reduce the outliers in the point cloud, followed by a *density filter* that evaluates the density distribution throughout the point cloud, then the point cloud is sampled by the *geometrical stable sampling* method, and finally the *entropy feature* or *dimensionality selector* methods are applied to what remains. Variations of point selection are obtained by including different stages to give a total of 24 considered combinations.

Likewise rejection methods follow a decorator pattern allowing individual methods to be used in combination. If rejection by distance and normal are used in combination, rejection by distance is applied first to provide an “initial cut”.

Various methods require normal vectors to be computed. Normal vectors are a geometric property of surfaces discretized by point clouds and each point in a cloud can be considered to have an associated normal that is recovered from neighbouring points. Numerous approaches have been proposed for determining normal vectors. For example, the use of k nearest neighbours to fit a tangent plane a point [51,52] or a local quadric surface of similar [53]. Klassing et al. [54] evaluated several methods to estimate normal vectors on a point neighbourhood, finding that principal component analysis (PCA) is the superior alternative in performance and speed. The PCA approach is used for the work in this paper.

3.2. Constructing a map using ICP

Each ICP variant is tested for its ability to construct a map given a sequence of consecutive scans from a moving sensor. The map is the consolidation of all points into a common frame of reference. For this work 100 consecutive scans were used to construct a map. Scan matching errors accumulate with consecutive scans and the rate of growth is considered an important measure of ICP variant performance.

Algorithm 1 describes the procedure that constructs the map. For each new scan the ICP variant was run until convergence was achieved or a maximum number of iterations of the minimization loop was reached. Convergence was established on a basis of a threshold on the improvement in the accuracy of the registered scans across an iteration. The value of the threshold was determined using a training data set and observing the characteristics of the three minimization methods.

Algorithm 1: Mapping procedure

Data: A set of successive scans $\{\mathcal{S}_i\}$ (point clouds). T_0^G is the initial transformation to the global (ground-truth) frame. T_0 is the initial transformation. T_i is the transformation of the i -th scan relative to the $i - 1$ scan. \mathcal{M}_i is the i -th scan transformed into the global (map) frame.

Result: \mathcal{M} is the map (point could) generated by the union of the transformed scans.

begin

```

 $S_0 \leftarrow \text{getScan}();$ 
 $T_0 \leftarrow T_0^G;$ 
 $M_0 \leftarrow T_0 S_0;$ 
 $S_1 \leftarrow \text{getScan}();$ 
while  $S_{i+1} \neq \emptyset$  do
     $T_i \leftarrow \text{ICPvariant}(S_{i-1}, S_i)$ 
     $T_i^G = T_0 \cdots T_{i-1} T_i;$ 
     $M_i \leftarrow S_i(T_i^G);$ 
     $M \leftarrow \bigcup_0^i M_i;$ 
     $S_{i+1} \leftarrow \text{getScan}();$ 

```

3.3. Performance metrics

The performance of an ICP variant is assessed using the consolidation of 100 scans (5 s of truck motion) to produce a map that can be compared to a ground truth consolidation of the same set of points. ICP performance is quantified by three metrics: accuracy, precision, and relative computational cost.

Accuracy

The accuracy, v_i , describes the total RMS error between the consolidated map, after the i th scan, and the ground-truth. For each individual scan the RMS error is,

$$E_i = \sqrt{\frac{1}{n_i} \sum_{j=1}^{n_i} \|S_{i,j}^t - \tilde{S}_{i,j}\|^2}, \quad (4)$$

where $S_{i,j}^t$ is the truth location for the j th point of scan i and $\tilde{S}_{i,j}$ defines the measured point cloud of scan i transformed using ICP registration. n_i is the number of points of the i th scan. Note that the RMS error for scan i includes the accumulated error associated with the ICP registrations of all previous scans.

Using (4), the total RMS error associated with the consolidated map constructed from i consecutive scans is defined as,

$$v_i = \sqrt{\frac{1}{\sum_{k=1}^i n_k} \sum_{k=1}^i E_k^2 n_k}. \quad (5)$$

The expectation is that each scan registration introduces an error, and the total RMS error grows with each consecutive scan. It is the rate of growth of this error that is important as a measure of the effectiveness of the ICP variant for a given scene.

For this study, ICP variants are considered accurate if $v_{i=100} \leq 0.2$ m. Though the mapping quality requirement will vary with the specific application scenario, this value is indicative of the level of uncertainty that might be tolerated for a range of autonomous activities in mining.

Precision

The precision measure looks at the variation in the growth of the total RMS error relative to a monotonically increasing error from E_1 to E_i . Across a range of scans, from $0 \rightarrow i$, the precision ρ_i is given by,

$$\rho_i = \frac{1}{i} \sum_{k=1}^i \|E_k - (m.(k - 1) + E_1)\|, \quad (6)$$

where m is the gradient of the straight line that joins E_1 and E_i .

Fig. 2 shows the evolution of the RMS error, E_i with each scan i for two ICP variants. A relative comparison of the variants reveals Variant 2 to be accurate but not precise, while Variant 1 is precise but not accurate. A potential conclusion is that the less precise Variant 2 is more sensitivity to the change in geometric information as the sensor moves and perceives the scene from a different perspective. If the performance across different scenes were compared, then an alternative conclusion could be that the more precise solution is related to a consistent quality of geometric information among the series of scans.

For this work, ICP variants are considered precise if $\rho \leq 0.1$ m.

Relative computational cost

The computational cost of each variant was quantified using a relative measure that compares the computation time relative to the fastest ICP variant. The relative computation time (rct) is describe as a ratio of computation times, such that $rct \geq 1$ for all variants.

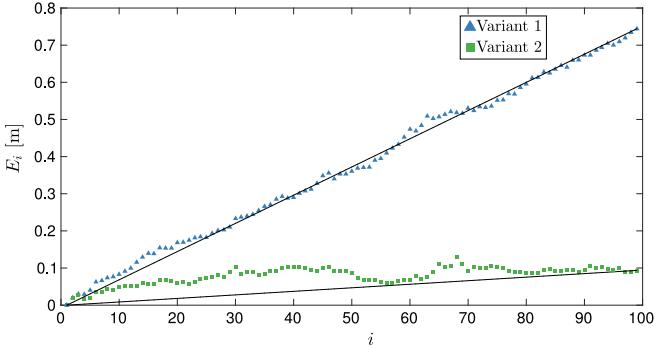


Fig. 2. RMS error, E_i , for two ICP variants over 100 consecutive scans. i is the scan number.



Fig. 3. Mounting of Velodyne, and pose solution hardware on a Caterpillar 777B haul truck.

While it is noted that the computational tractability of an ICP variant will depend, for a given scene, on the hardware used, the level of code optimization, and the time available for map updates, the relative measure provides a convenient ranking of variants that would be expected to be consistent among these variables.

An ICP variant is considered computationally efficient if $rct \leq 3$. This somewhat arbitrary value is intended to filter the large number of ICP variants with a preference to those that are fastest to converge.

4. Evaluation data set

The dataset used for this study is based on measurements from a Velodyne HDL-64E [55] sensor mounted to a Caterpillar 777B haul truck, see Fig. 3. The LiDAR has a 360° horizontal field of view and a 26.8° vertical field of view. The sensor was configured to scan at 20 Hz with an azimuth resolution of 0.09°. Range measurements provided by the LiDAR unit have a standard deviation of 0.02 m and a maximum range of 120 m.

The LiDAR, GNSS/IMU navigation system and associated components were mounted to the front of a Caterpillar 777B haul truck, as shown in Fig. 3. The mounting configuration provided the Velodyne with a 180° field of view in front of the truck.

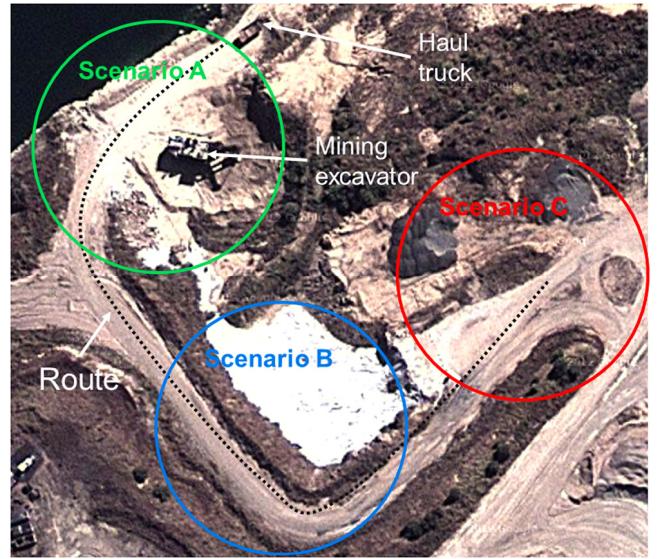


Fig. 4. The haul road sites (scenes) chosen for this study (Image from Google Inc., 2014).

A ground-truth scan was established using the pose of the truck as measured by an Applanix POS LV 420 positioning system [56] that fuses an RTK GNSS solution with high accuracy inertial measurements. The Applanix navigation system has an RMS accuracy of 0.02 m in ground plane coordinates, 0.05 m in vertical coordinates, 0.015° in roll and pitch, and 0.02° in yaw.

The Velodyne LiDAR is registered to the navigation solution using the method described in [57] and has been assessed as being precise to 0.01 m and 0.05° for translation and rotation parameters respectively.

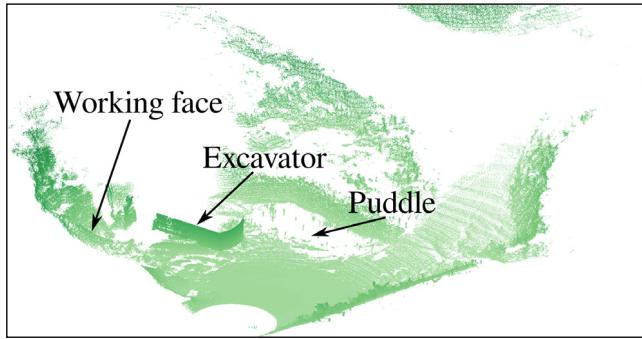
The Velodyne scans registered by Applanix were verified through comparison with a FARO Focus3D Terrestrial Laser Scanner [58]. The FARO sensor provides a one sigma range error of 0.3 mm at ranges of 10 m for 90% reflections with a beam divergence of 0.009°. A dual axis inclinometer levels each scan with an accuracy of 0.015°. A high density point cloud is generated with a possible vertical and horizontal step size of 0.009° and 0.036° respectively. The map generated by fusing scans registered with the navigation solution were found to be within 0.2 m RMS of the corresponding FARO scan, giving confidence in the ground truth. The RMS was calculated by finding the minimum distance of a point from a triangulated version of the FARO point cloud. The quality of the terrain map generated by combining the Velodyne data with the navigation solution reflects the range accuracy available with the sensor and the registration errors associated with the LiDAR and the navigation sensors.

The data for this study was logged as the truck drove along a 350 m section of haul road, see Fig. 4. From this route, three sections of travel were segmented to provide the scene data used in this paper: Scene A represents a loading area; Scene B represents a haul-road entering/leaving the loading area; and Scene C represents an area where material is stockpiled. Each scene comprises five seconds of data which equates to 100 individual scans from the Velodyne sensor. Each scan contains approximately 84,500 points.

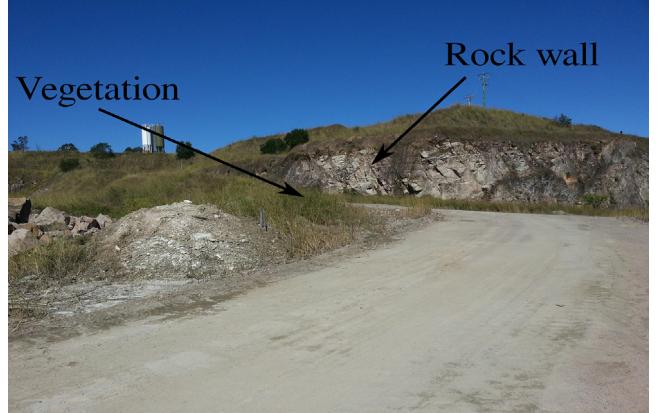
The scenes are considered representative of a haul-truck route, from the working face to a stockpile area, and terrain perception in these areas is important for the introduction of automation technologies and for enhancing operational safety. Details of the scenes are given below by a representative photograph of the work area and the consolidation of 100 scans segmented for each scene and registered against the truck navigation solution.



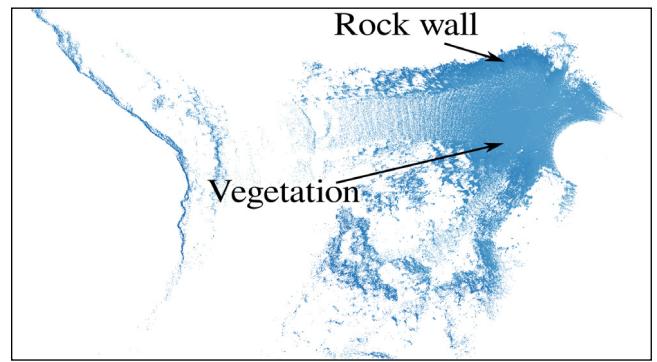
(a) Scene A: the load area with truck (left) and excavator (right).



(b) Scene A: Consolidated ground truth point cloud viewed from above-right.

Fig. 5. Scene A: The loading area.

(a) Scene B: haul road segment.



(b) Scene B: Consolidated ground truth point cloud viewed from above.

Fig. 6. Scene B: haul road.

4.1. Scene A: loading area

Scene A is a loading area and includes an excavator digging a working face, see in the right of Fig. 5a. The terrain has a low slope with some vegetation and rocks. The scene contains an electric mining shovel, the trailing power cable of the shovel, a pool of water and a berm. The haul truck is visible in the left of Fig. 5a. The average speed of the truck through the scene is 1.65 m/s, giving an average sensor movement between consecutive scans of 0.0825 m.

4.2. Scene B: haul road

Scene B is a typical haul road segment, taken from the route between the excavator work area to the stockpile scene, Fig. 6. The truck drives up-hill towards a sheet rock wall approximately 5 m high covered with abundant vegetation. The road is relatively flat with rock and vegetation to the sides of the road. The average speed of the truck through the scene was 2.39 m/s, with 0.12 m average sensor position movement between consecutive scans.

4.3. Scene C: stockpile area

Scene C is a stockpile area, see Fig. 7. The terrain is flat and without vegetation. The principal features are two stockpiles of the road-based product produced at the site. The average speed of the truck through the scene was 2.88 m/s, giving 0.14 m movement in sensor position between consecutive scans.

5. Results and observations

The observations that follow are based on 62,208 ICP-variant/scene combinations, comprising 20,736 distinct variants applied to the three different scenes (data sets) described in

Section 4. Observations are grouped into those general to the application of ICP for terrain mapping tasks in a typical mining environment, and those that focus on the composition of the ICP algorithm. The performance metrics of accuracy, precision, and relative computational cost are used to form these observations.

5.1. General observations on the application of ICP

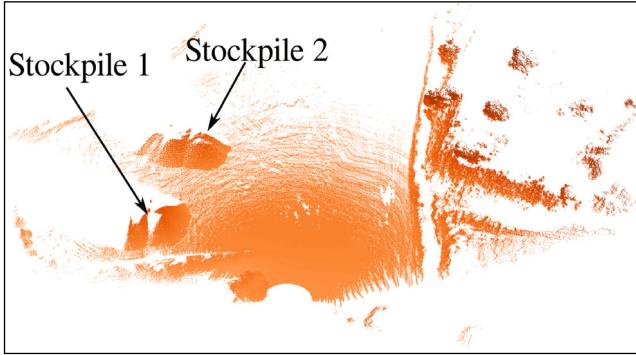
Fig. 8 presents a summary of ICP performance in the form of a set diagram that separates ICP variants by the performance metrics and the three terrain scenes. Overlapping regions identify the number of variants that meet more than one of the performance requirements. For Scene A, 130 distinct ICP variants simultaneously meet the accuracy, precision, and efficiency thresholds, and were considered successful variants. For all scenes, all accurate variants also satisfied the precision requirement. One perspective on this is that the precision metric provides a measure of the consistency of geometric information between consecutive scans. Since the data sets of this work were collected over less than 15 m of movement of the sensor, the captured features in the scene will be consistent across the series of scans. Applying a similar perspective to the accuracy metric, scenes that contain features such as planes and well defined regular structures, e.g. Scene A, are more suited to the application of ICP.

Observation 1. No single variant satisfies all performance criteria.

The intersection of all performance metrics across all scenes is an empty set, see Fig. 8, indicating no single variant is able to satisfy all performance metrics when applied to all scenes. The result is



(a) Scene C: the stockpile area.



(b) Consolidated ground truth point cloud viewed from above.

Fig. 7. Scence C: Stockpile area. The terrestrial survey station used to evaluate the ground truth is visible in Fig. 7a, but not present in the scan image (Fig. 7b).

largely due to the small number of variants (18 of 20,736) that were able to meet the accuracy requirement for each of the three data sets. If the allowed threshold were increased to 0.3 m then 975 variants would be considered accurate and the likelihood of a variant also meeting the precision and computation cost requirement would increase significantly.

There are several conclusions that could be drawn from this observation:

1. The practical application of ICP to terrain mapping is difficult when applied to “natural” terrains and requires the careful cultivation of data sets and the fine tuning of variants to the individual scene.
2. There are gaps in the available strategies that make up an ICP algorithm that means existing variants are not well suited to the terrain mapping task posed in this paper. Here the mapping task combines the types of scenes and the approach to forming a consolidate map from a series of scans.
3. The proliferation of strategies for ICP algorithms is, in part, attributable to the search for methods that can be more generally applied to scan matching.

Observation 2. *The quality and computational cost of the ICP based terrain mapping solution varies significantly across the ICP variants, making judicious selection of the ICP variant an imperative if minimum levels of performance must be met.*

The scatter plot of Fig. 9 highlights the large spread in the quality of the scan matching provided by the ICP variants. A significant factor here is the presence of a large number of outliers due to what can be considered poorly constructed algorithms for the task. Specifically, all variants using point-to-point minimization

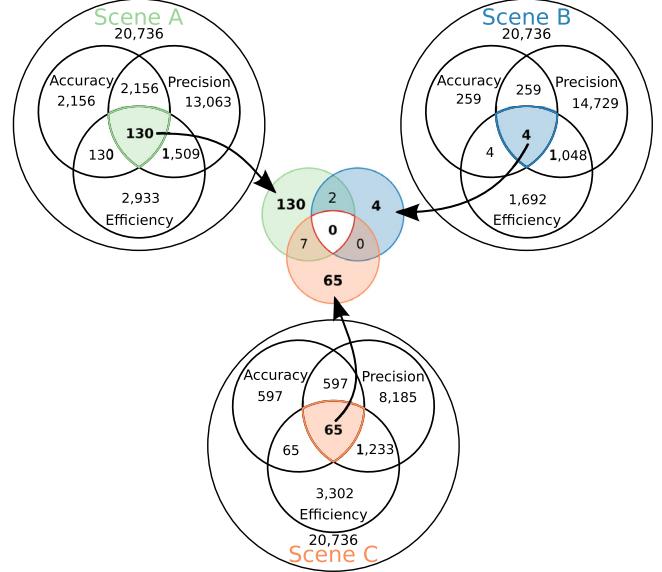


Fig. 8. Interception of variants in the performance tolerances for the whole data set.

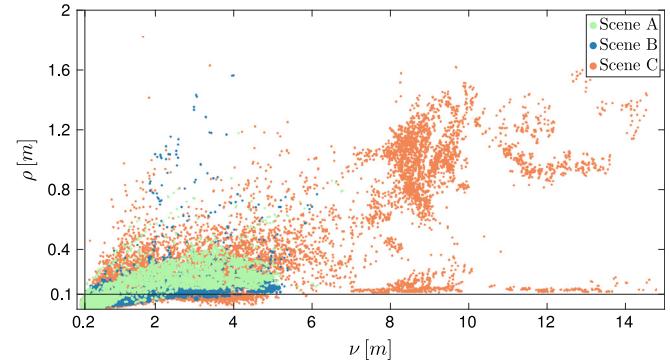


Fig. 9. Precision (ρ) vs accuracy (ν). Vertical and horizontal lines delimited the tolerance region for accuracy and precision respectively.

perform poorly and enhance the appearance of spread in the data. If the variants using point-to-point minimization were removed, the accuracy range is reduced to 5 m and the precision range to 1 m. Also, the remaining set of variants provide a more compact set for Scene A, followed by Scene B then Scene C. As noted previously, this trend is indicative of the quality of information available in the point clouds.

The challenge then for applying ICP is to identify the composition of successful ICP variants such that strong candidates for general application can be formed. The compositional elements of successful ICP variants is covered in Section 5.2.

Observation 3. *The performance of ICP variants depend on the scene.*

Both the summary set diagram of Fig. 8 and the scatter plot of Fig. 9 show the impact of the difference scenes on the performance of ICP variants. The scene differences are twofold: (i) the data collection meant that the sensor movement between consecutive scenes was smallest for Scene A and largest for Scene C, and the registration task is more difficult the larger the initial separation of the point clouds; and (ii) the geometry differences in the scenes mean that variants do not work equally well for each scene.

This observation is consistent with [15] who showed the performance of ICP variants differ significantly across different data sets.

It represents a key issue in the use of ICP, namely that each data set seemingly has its own best variant for scan matching, and is a potential reason for the proliferation of ICP variants to address the challenges of different scenes. It is also suggestive of the potential for improved methods being applied to ICP. These two points focuses the challenge for ICP towards identifying those combinations of methods that generally perform well across different scenes.

It is insightful to understand why, across all ICP variants, Scenes A and B have tighter accuracy and precision than Scene C, see Figs. 5–7 to refer to the scenes. Observe that Scene C has a planar ground plane, consequently, scan matching for this scene are prone to sliding. This is believed to be the cause of the significant variation observed, and in particular, accounts for the large number of outliers. Scene A exhibits the best overall performance. The excavator present in this scene provides two large flat and orthogonal regions in the point cloud that become effective features for accurate scan matching. Scene B performs well generally but has fewer accurate and precise variants when compared with Scene A. Scene B is characterized by significant vegetation (scattered distribution of points) to the sides of the road and a significant rock wall that is irregular in form and on which the calculation of point normal is sensitive resulting in local minima.

Fig. 10 uses an accuracy heat map to visualize the correlation in performance of ICP across the three scenes. Scenes A and B present similar geometric features, with vertical and horizontal planes needed to constrain the solution. There is moderate correlation between the performance of ICP variants applied to Scenes A and B. The performance of variants on Scenes B and C are uncorrelated.

Observation 4. Computational efficiency is not traded for computational accuracy or precision.

This observation runs contrary to the general expectation that an investment in computation time will yield a return on solution accuracy and precision. The relative computation cost and accuracy for the 20,736 variants show the two metrics to be uncorrelated, although it is to be expected that some combinations of methods will in general perform more accurately, more precisely, and have lower computation efficiency than others.

The computational cost of applying ICP is influenced by several factors: (i) optimization of software implementation; (ii) processing limit imposed by hardware; (iii) the amount of processing of point cloud data, in particular the calculation of features; and (iv) the number of iterations required for the solution to converge. The use of relative computation cost is intended to negate the impact of (i) and (ii), leaving (iii) and (iv) as key factors for computational performance.

The coupling of computational cost to the particular composition of strategies in an ICP variant makes it difficult to isolate the significance of the different computational strategies. As a general rule however, variants requiring the calculation of features from the point cloud data attract a computational cost. Fig. 8 shows Scene B having almost half the number of computationally efficient variants as Scenes A and C. The dominant features of Scene B are large amounts of vegetation and a rough vertical wall along the road. This makes point correspondence more difficult and more iterations of the algorithm are required to converge. In contrast, the vertical faces of the excavator in Scene A support convergence to accurate solutions, and the large flat ground plane in Scene C leads to fast convergence to local minima.

5.2. Observations on ICP variant composition

The discussion so far has highlighted the sensitivity of the ICP performance to the composition of the algorithm and the point cloud geometry information available in the different scenes. The

challenge is to identify those combinations of methods that are best performed across the three scenes in anticipation of these being well performed for scenes generally. The selection of the variants judged to be best is made by thresholding against the metrics of performance in the three scenes together by taking the average. Three sets of variants are identified: (i) the set of accurate variants; (ii) the set of precise variants; and (iii) the set of fast variants, see Figs. 11–13.

The set of accurate variants comprise those with accuracy of less than 0.2 m; 0.0868% of the 20,736 ICP variants are considered accurate. Precise variants are those whose precision measure across the three scenes is less than 0.1 m; 35.9% fall below the precision threshold. Computationally efficient variants comprise those whose relative computation time is less than 3; 6.08% of variants fall below the relative computation time threshold.

Figs. 11–13 are visual depictions of the composition of the sets of algorithms that are considered accurate, precise, or computationally efficient, and are used to support the observations. The alpha numeric codes of Table 1 are used to identify the most effective methods based on the performance metrics. An example reading of these relationship diagrams is as follows. Fig. 11 shows that all of the most accurate variants use point-to-plane (Method 1) minimization and 33.3% apply pair-rejection in the form of Method $\gamma\delta$ (where γ refers to angular deviation and δ to adaptive distance methods). The most accurate, precise and computational efficient solution uses Method $\gamma\delta$, and this is indicated by the solid green circle, the solid blue square and the solid red triangle, respectively. Similarly, 66.7% of the variants use Method z (nearest neighbour enhanced by normals) for point matching, with the most accurate variants in this set. And so the diagram continues with 16.7% of ICP variants using the combination of Methods b (outlier removal) and f (dimensionality selection) for point selection. All three weighting methods appear equally across the variants. The most accurate variant, on average, has the alpha numeric code: bf4z0 $\gamma\delta$ 1.

Observation 5. Minimization is best performed by point-to-plane and is preferred over generalized and point-to-point distance minimization.

Figs. 11–13 show all the accurate variants, 65.7% of precise variants, and 44.7% of computationally efficient variants use point-to-plane minimization. Fig. 14 shows the performance of the ICP variants using a larger sampling of the variants. Point-to-point minimization performs poorly on all scenes, and point-to-plane minimization performs best. For all methods, Scene C proved the most difficult to achieve an accurate result. This is likely to be attributed to the greater sensor movement between consecutive scans and the dominant geometry features that allow convergence to local minima.

This observation is consistent with [35,13], both papers finding that point-to-plane minimization was more accurate than other methods. The overall failing of point-to-point based variants is due to the inherent error in trying to find correspondence between sets of points that come from a moving sensor that sees different parts of the scene in consecutive scans. In this scenario the performance of point-to-point minimization is constrained by the point density of the scan, with higher point densities more likely to provide suitable point correspondences.

The point-to-plane approach, where points are paired with surfaces, makes the ICP algorithm robust to sensor movement and to variation in the geometric content of a scene. The generalized ICP variants also use a point-to-plane distance function and have similar performance to point-to-plane when applied to Scenes A and B, although attracting a higher computational cost.

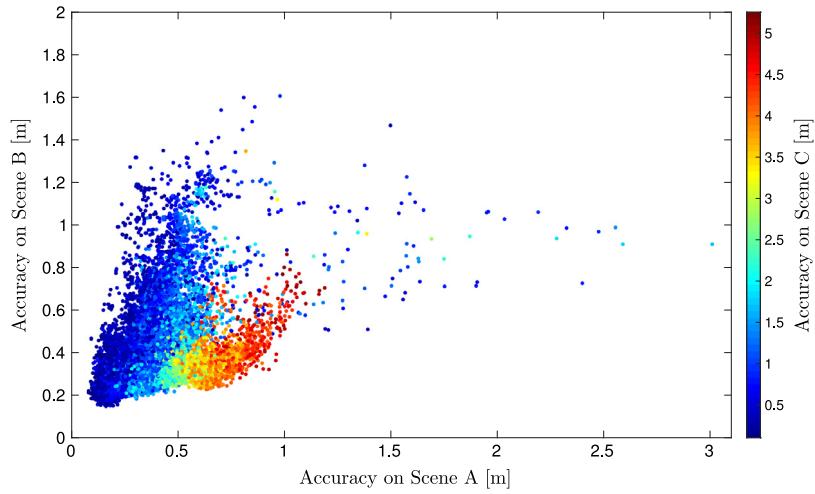


Fig. 10. Comparison of ICP variant performance across the three scenes in terms of the accuracy performance metric. Variants that returned an accuracy of better than 1 m on at least one scene are shown.

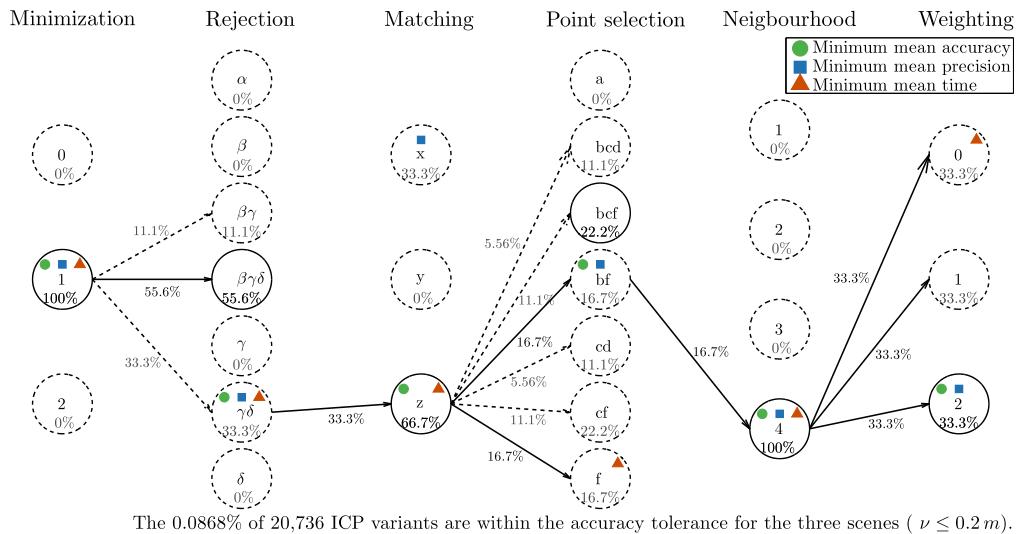


Fig. 11. Method relationship diagram within the accuracy tolerance.

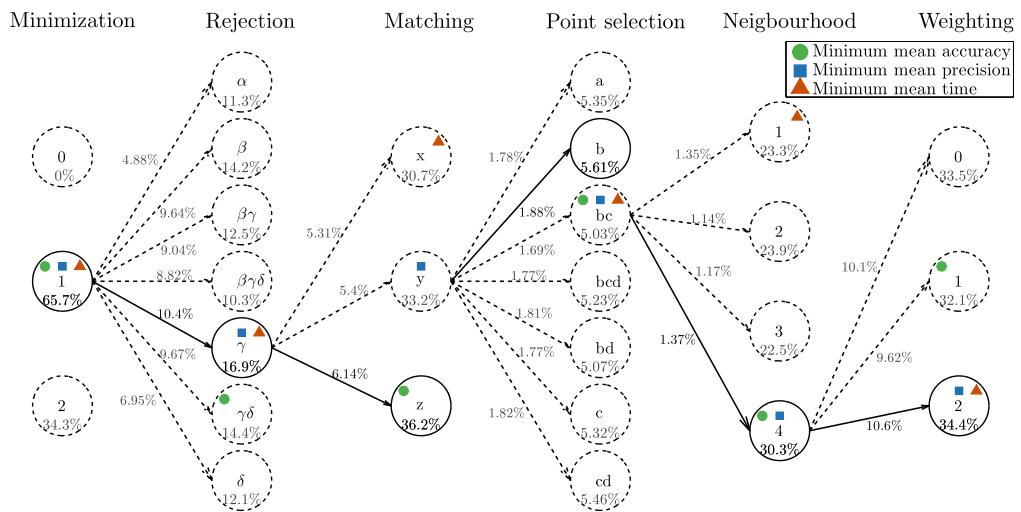


Fig. 12. Method relationship diagram within the precision tolerance.

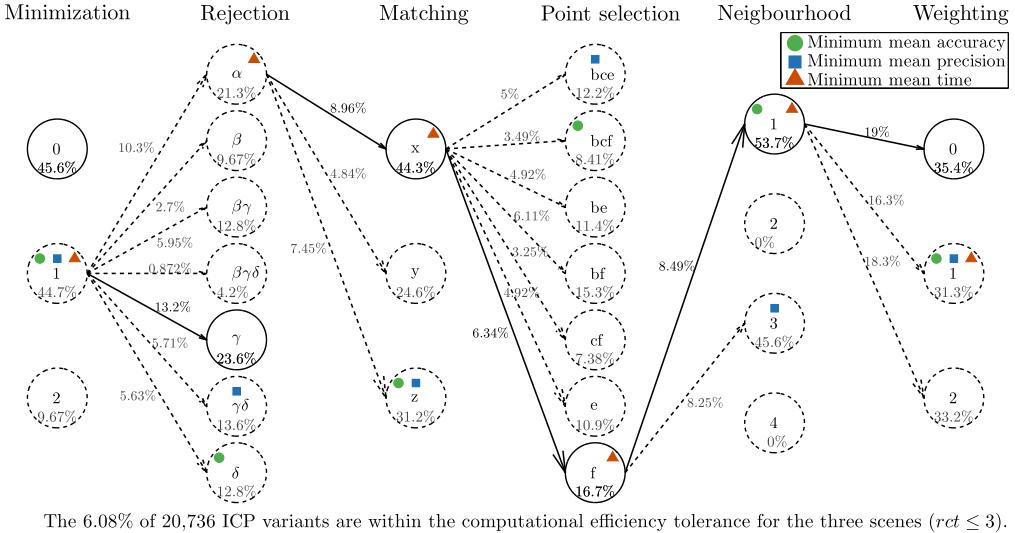


Fig. 13. Method relationship diagram within the relative time tolerance.

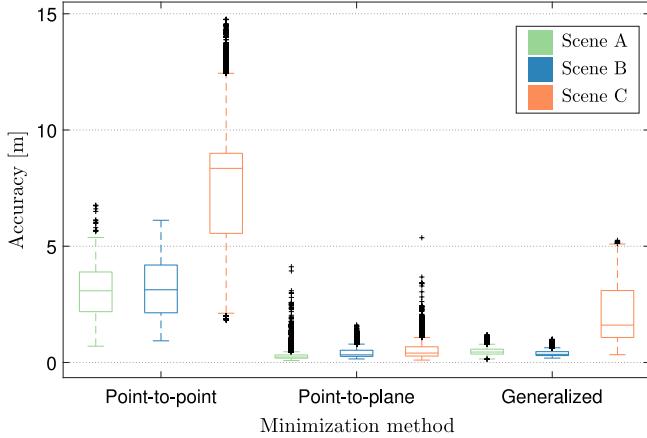


Fig. 14. Performance of minimization methods in terms of the accuracy for the different scenes. The box plots show spread of variant accuracy among the best 1000 variants that use the particular minimization method.

Observation 6. *Pair rejection is important for improving the quality of the scan matching. Rejection by angular deviation can be used effectively in combination with rejection by adaptive distance and rejection by worst distance percentage.*

Pair rejection removes outlying point-pairs as a means to improve the application of the distance function as a measure of the misalignment between the two point clouds. The additional calculation may attract a computational cost but should improve convergence.

Fig. 11 shows all of the most accurate variants employed some form of rejection, with the dominant method being to reject point pairs based on the difference in their local normals (Method γ) which is applied in combination with other methods. This suggests that rejection is an important step in the registration process. Fig. 15 reinforces this idea showing that variants that make use of some form of rejection out-perform variants that do not use pair rejection.

Among the set of most accurate methods, 100% employ the rejection by angular deviation algorithm (Method γ) [13], in combination with rejection by adaptive distance (Method δ) (33.3%) [24] or rejection by worst distance percentage (Method β) (11.1%) [13]. The three methods (Methods $\beta\gamma\delta$) applied together account

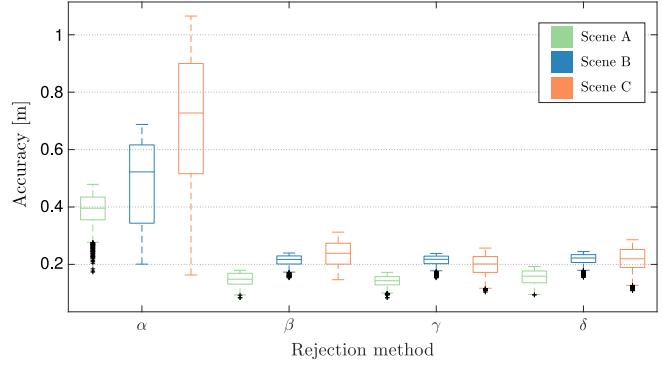


Fig. 15. Performance of rejection methods in terms of accuracy. The box plots show the spread of variant accuracy among the best 1000 variants that contain the particular rejection method (either in isolation or in combination with others).

for 55.6% of the most accurate variants. Method γ also appears in 53.8% of the most precise variants in combinations with Methods β and δ .

Fig. 13 shows that no rejection is the fastest rejection method on average, however angular deviation rejection has slightly more variants in that zone than no rejection at all.

Observation 7. *Bounded radius is the preferred method for neighbourhood selection.*

Neighbourhood selection is used to determine the set of points to be used to characterize the local geometry around a point. From Fig. 11, all of the most accurate variants employ the density bounded radius [30] for neighbourhood selection making it a compelling choice among the variants considered. This approach allows the local neighbourhood to be scaled according to the level of noise in the point cloud data. Being relatively computationally expensive, the method does not feature among the fastest ICP variants, see Fig. 13. As expected, applying a constant neighbourhood for all points (Method 1) is the fastest approach.

Fig. 16 shows the performance of constant, entropy feature, and density based neighbourhood selection to be similar. The bounded radius approach is superior on all scenes, however the overlap with other methods might suggest that tuning of the methods is required or that other phases of the ICP algorithm are more critical to the overall performance.

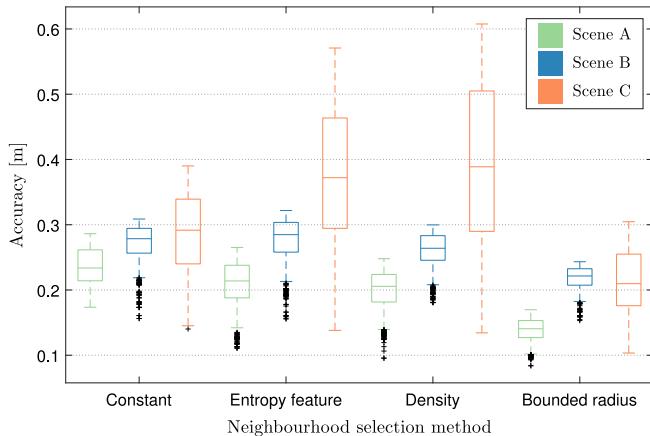


Fig. 16. Performance of neighbour selection methods in terms of accuracy for the three scenes. The box plots show the spread in variant accuracy among the best 1000 variants that use the particular neighbourhood selection method.

Observation 8. *Use nearest neighbour enhanced by normals for point matching.*

From Fig. 11, 66.7% of the most accurate variants use nearest neighbour enhanced by normals [37] for point matching. This appears to be the most accurate variant on average. However, while precision is generally correlated with accuracy, the most prevalent point matching method (36.2%) in the group of precise ICP variants is nearest neighbour enhanced by normals, although a significant proportion (30.5%) are nearest neighbour enhanced by moment invariants [38]. The most common point matching method (44.3%) among the fastest variant is, not surprisingly, nearest neighbour matching [11], this being the simplest algorithm.

Whilst point matching is an important step in ICP, none of the three algorithms considered is remarkably better than the others. However, the accuracy results shown in Fig. 11 suggest that methods that constrain the point correspondence based on the consistency of low dimension descriptors within the neighbourhood of points work well for the fine registration task of this work.

Observation 9. *There is no clear preference among point selection methods.*

The selection process serves two purposes: (i) to remove points that impede finding correspondence between the geometry present in two point clouds; and (ii) to reduce the total number of points used during the ICP iteration phase. Fig. 11 shows that while some form of filtering of data points is required to deliver accurate ICP registration, there are many combinations of filters that can be effectively applied. Of the 18 variants that are classified as accurate, there are six different point selection approaches that are used.

The performance of a larger sampling of the variants is shown in Fig. 17. Intuitively point selection is an important aspect of ICP and the appropriate level and type of selection is related to the scene and its representation as a point cloud. However, there are no strong preferences that emerge among the alternatives.

Observation 10. *There is no preference among weighting methods and this stage can reasonably be removed from the ICP algorithm.*

Across the sets of accurate, precise, and computational efficient variants, see Figs. 11–13, there was no preference for a particular weighting strategy. Fig. 18 shows that ICP performance is insensitive to the derating of matched pairs through the application of a weighting. This is consistent with previous studies [13,27], and suggests that the ICP algorithm does not benefit from the

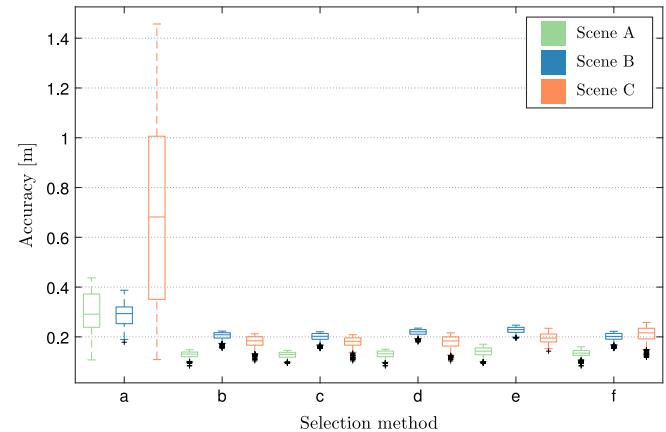


Fig. 17. Performance of point selection methods in terms of variant accuracy. The box plots show the spread in variant accuracy among the best 400 variants that use the particular strategy. See Table 1 to identify the selection methods.

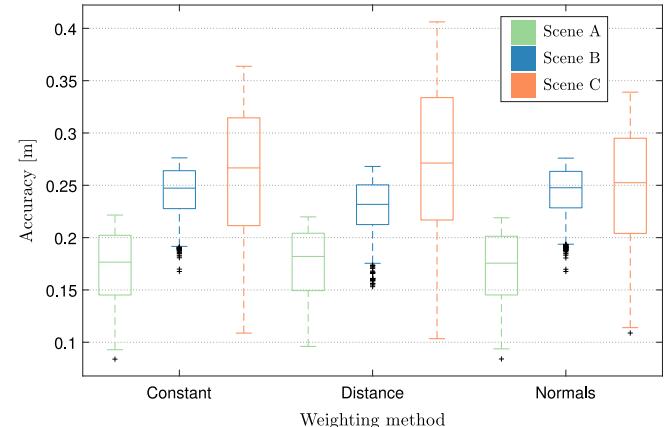


Fig. 18. Performance of weighting methods in terms of accuracy among the best 1000 variants. The box plots show the spread in variant accuracy among the best 1000 variants that use the particular strategy.

application of a weighting for the contribution of matched pairs to the distance function.

This result contrasts with that for rejection which showed that ICP performance benefits from the complete removal of matched pairs.

6. Conclusions

This paper provides an evaluation of 20,736 variations of the ICP algorithm for scan matching, using a terrain mapping task and example scenes from a surface mining environment. The ICP variants were constructed from combinations of published methods for the computational elements that comprise an ICP algorithm: point selection; neighbourhood selection; point matching; weighting; rejection; and minimization. Using the performance metrics of accuracy, precision, and computation efficiency, there was no single preferred variant, however some general patterns/guidance for the construction of an ICP algorithm do emerge.

Specifically:

- Minimization by point-to-plane distance of [12] outperforms other distance minimization methods. It was common to all ICP variants that were able to meet the accuracy criteria. The point-to-point distance function should not be used for terrain scan matching due to errors with finding

- point associations between point clouds that contain measurements to different points in the scene.
- Point matching is best achieved using the nearest neighbour with normals algorithm of [37]. All point matching methods employ a variation of nearest neighbour to establish the correspondence relationships between two scans being matched. The variations apply constraints to the closest point strategy used to find point correspondence. For the scenes of this paper, local normals provide an effective discriminator for matching points.
 - Rejection is an important stage of the ICP computation due to the removal of point pairs that disrupt the minimization of the distance function. Rejection was most effectively achieved by the angular deviation method of [13] alone or in combination with the adaptive distance method of [24] or the worst percentage distance rejection method of [13].
 - Neighbourhood selection should be completed using the bounded radius method of [30]. This method uses the local curvature and point density to determine the radius of points that best describe the local geometry.
 - There is no clear preference for point selection methods. Interestingly, though point selection based on the retention of planar characteristics appeared, alone or in combination with other methods, in 87 % of accurate variants, it was not part of any of the precise variants.
 - The weighting of paired points provide no clear benefit to the ICP process, with an equal presence of the three weighting methods in ICP variants that were either accurate, precise, or computationally efficient.

These findings overlap the observations of [13] and focus attention on the need to find means for enabling robust scan matching across different point cloud distributions. Given no single method is best across all point clouds this argues for the need to find adaptive methods that can assemble a good performing variant given a point cloud or new methods that can tune themselves to the circumstances of a point cloud. To this latter point, the entropy based ideas presented in [27] have a strong appeal in so far as they look to measure the way in which the point cloud is distributed. This looks to be an avenue worthy of further exploration.

The overall objective in this paper has been to provide guidance to the implementers of ICP algorithms for scan matching. The main conclusions are that ICP-based mapping of natural terrain requires careful cultivation of the point cloud data, the appropriate selection of computational strategies that exploit the available geometric information, and requires tuning of strategies specific to the terrain characteristics.

This investigation has focused on a specific environment (the terrain of open-pit mining) for map building applications. However, much of this is common in other settings that have irregular structure, such as natural terrain, and these findings extend *mutatis mutandis* beyond the confines of this investigation.

References

- [1] E. Guizzo, How google's self-driving car works, IEEE Spectrum 18 (Online, October) (2011).
- [2] P. Lever, Automation and robotics, in: P. Darling (Ed.), SME Mining Engineering Handbook, Society for Mining, Metallurgy, and Exploration, 2011.
- [3] E.M. Nebot, Surface mining: main research issues for autonomous operations, in: Robotics Research, Springer, 2007, pp. 268–280.
- [4] G. Weiß, E.v. Puttkamer, A map based on laserscans without geometric interpretation, in: Intelligent Autonomous Systems, vol. 4, 1995, pp. 403–407.
- [5] F. Lu, E. Milios, Robot pose estimation in unknown environments by matching 2d range scans, J. Intell. Robot. Syst. 18 (3) (1997) 249–275.
- [6] A. Díosi, L. Kleeman, Laser scan matching in polar coordinates with application to SLAM, in: Intelligent Robots and Systems, 2005, IROS 2005, 2005 IEEE/RSJ International Conference on, IEEE, 2005, pp. 3317–3322.
- [7] M. Magnusson, T. Duckett, A comparison of 3D registration algorithms for autonomous underground mining vehicles, in: Proceedings of the European Conference on Mobile Robotics, ECMR 2005, 2005, pp. 86–91.
- [8] P. Ye, F. Liu, Novel image registration method using multiple Gaussian mixture models, in: Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on, IEEE, 2012, pp. 2117–2120.
- [9] P. Biber, W. Straßer, nscan-matching: Simultaneous matching of multiple scans and application to SLAM, in: Robotics and Automation, 2006, ICRA 2006, Proceedings 2006 IEEE International Conference on, IEEE, 2006, pp. 2270–2276.
- [10] B. Bellekens, V. Spruyt, R.B.M. Weyn, R. Berkvens, A survey of rigid 3d point-cloud registration algorithms, in: The Fourth International Conference on Ambient Computing, Applications, Services and Technologies, Citeseer, 2014.
- [11] P.J. Besl, N.D. McKay, Method for registration of 3-D shapes, in: Robotics-DL Tentative, International Society for Optics and Photonics, 1992, pp. 586–606.
- [12] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image Vis. Comput. 10 (3) (1992) 145–155.
- [13] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: 3-D Digital Imaging and Modeling, 2001, Proceedings, Third International Conference on, IEEE, 2001, pp. 145–152.
- [14] J. Salvi, C. Matabosch, D. Fofi, J. Forest, A review of recent range image registration methods with accuracy evaluation, Image Vis. Comput. 25 (5) (2007) 578–596.
- [15] F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, Comparing ICP variants on real-world data sets, Auton. Robots 34 (3) (2013) 133–148.
- [16] F. Pomerleau, F. Colas, R. Siegwart, A review of point cloud registration algorithms for mobile robotics, Found. Trends Robot. 4 (1) (2015) 1–104.
- [17] R. Horaud, F. Forbes, M. Yguer, G. Dewaele, J. Zhang, Rigid and articulated point registration with expectation conditional maximization, IEEE Trans. Parallel Anal. Mach. Intell. 33 (3) (2011) 587–602.
- [18] T. Stoyanov, M. Magnusson, A.J. Lilienthal, Point set registration through minimization of the L_2 distance between 3d-ndt models, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 5196–5201.
- [19] G.D. Evangelidis, D. Kounades-Bastian, R. Horaud, E.Z. Psarakis, A generative model for the joint registration of multiple point sets, in: Computer Vision-ECCV 2014, Springer, 2014, pp. 109–122.
- [20] B. Eckart, K. Kim, A. Troccoli, A. Kelly, J. Kautz, MLMD: Maximum Likelihood Mixture Decoupling for Fast and Accurate Point Cloud Registration, in: 3D Vision (3DV), 2015 International Conference on, IEEE, 2015, pp. 241–249.
- [21] P. Biber, W. Straßer, The normal distributions transform: A new approach to laser scan matching, in: Intelligent Robots and Systems, 2003, IROS 2003, Proceedings, 2003, Vol. 3, IEEE/RSJ International Conference on, IEEE, 2003, pp. 2743–2748.
- [22] M. Magnusson, The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection (Ph.D. thesis), 2009.
- [23] T. Masuda, N. Yokoya, A robust method for registration and segmentation of multiple range images, Comput. Vis. Image Underst. 61 (3) (1995) 295–307.
- [24] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, Int. J. Comput. Vis. 13 (2) (1994) 119–152.
- [25] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, M. Levoy, Geometrically stable sampling for the ICP algorithm, in: 3-D Digital Imaging and Modeling, 2003, 3DIM 2003, Proceedings. Fourth International Conference on, IEEE, 2003, pp. 260–267.
- [26] A. Torsello, E. Rodola, A. Albarelli, Sampling relevant points for surface registration, in: 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT, 2011 International Conference on, IEEE, 2011, pp. 290–295.
- [27] A. Gressin, C. Mallet, N. David, Improving 3D Lidar point cloud registration using optimal neighborhood knowledge, in: Proceedings of ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, Australia, vol. 5, 2012, pp. 111–116.
- [28] J.-F. Lalonde, N. Vandapel, D.F. Huber, M. Hebert, Natural terrain classification using three-dimensional lidar data for ground robot mobility, J. Field Robot. 23 (10) (2006) 839–861.
- [29] O. Schall, A. Belyaev, H.-P. Seidel, Robust filtering of noisy scattered point data, in: Proceedings of the Second Eurographics/IEEE VGTC Conference on Point-Based Graphics, Eurographics Association, 2005, pp. 71–77.
- [30] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel, M. Hebert, Scale selection for classification of point-sampled 3D surfaces, in: 3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on, IEEE, 2005, pp. 285–292.
- [31] N.J. Mitra, A. Nguyen, L. Guibas, Estimating surface normals in noisy point cloud data, Internat. J. Comput. Geom. Appl. 14 (04n05) (2004) 261–276.
- [32] J. Demantké, C. Mallet, N. David, B. Vallet, Dimensionality based scale selection in 3D lidar point clouds, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. Laser Scanning (2011).
- [33] T. Wiemann, A. Nüchter, K. Lingemann, S. Stiene, J. Hertzberg, Automatic construction of polygonal maps from point cloud data, in: Proc. 8th IEEE Intl. Workshop on Safety, Security, and Rescue Robotics, SSRR-2010, Bremen, Germany, 2010, p. 22.

- [34] J. Elseberg, S. Magnenat, R. Siegwart, A. Nüchter, Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration, *J. Softw. Eng. Robot.* 3 (1) (2012) 2–12.
- [35] K. Pulli, Multiview registration for large data sets, in: 3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on, IEEE, 1999, pp. 160–168.
- [36] G. Godin, M. Rioux, R. Baribeau, Three-dimensional registration using range and intensity information, in: Photonics for Industrial Applications, International Society for Optics and Photonics, 1994, pp. 279–290.
- [37] J. Feldmar, N. Ayache, F. Betting, 3D-2D projective registration of free-form curves and surfaces, in: Computer Vision, 1995, Proceedings, Fifth International Conference on, IEEE, 1995, pp. 549–556.
- [38] G.C. Sharp, S.W. Lee, D.K. Wehe, ICP registration using invariant features, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (1) (2002) 90–102.
- [39] C. Schutz, T. Jost, H. Hugli, Multi-feature matching algorithm for free-form 3D surface registration, in: Pattern Recognition, 1998, Proceedings, Fourteenth International Conference on, IEEE, 1998, pp. 982–984.
- [40] D. Akca, Registration of point clouds using range and intensity information, in: The International Workshop on Recording, Modeling and Visualization of Cultural Heritage, 2005, pp. 115–126.
- [41] Z. Peng, Efficient matching of robust features for embedded SLAM (Master's thesis), Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart, 2012. <http://elib.uni-stuttgart.de/opus/volltexte/2012/7617>.
- [42] K. Khoshelham, D. Dos Santos, G. Vosselman, Generation and weighting of 3D point correspondences for improved registration of RGB-D data, *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 1 (2) (2013) 127–132.
- [43] A. Nüchter, 3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom, vol. 52, Springer, 2009.
- [44] A. Segal, D. Haehnel, S. Thrun, Generalized-ICP, in: Robotics: Science and Systems, vol. 2, 2009, p. 4.
- [45] R.S.J. Estépar, A. Brun, C.-F. Westin, Robust generalized total least squares iterative closest point registration, in: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2004, Springer, 2004, pp. 234–241.
- [46] H. Xie, K.T. McDonnell, H. Qin, Surface reconstruction of noisy and defective data sets, in: Visualization, 2004, IEEE, IEEE, 2004, pp. 259–266.
- [47] R. Unnikrishnan, J.-F. Lalonde, N. Vandapel, M. Hebert, Scale selection for the analysis of point-sampled curves, in: 3D Data Processing, Visualization, and Transmission, Third International Symposium on, IEEE, 2006, pp. 1026–1033.
- [48] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- [49] R.B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in: IEEE International Conference on Robotics and Automation, ICRA, 2011.
- [50] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, ROS: an open-source Robot Operating System, in: ICRA Workshop on Open Source Software, vol. 3, 2009, p. 5.
- [51] R. Hoffman, A. Jain, Segmentation and classification of range images, *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (6) (1987) 606–620.
- [52] J. Huang, C. Menq, Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points, *IEEE Trans. Robot. Autom.* 17 (3) (2001) 268–279.
- [53] M. Yang, E. Lee, Segmentation of measured point data using a parametric quadric surface approximation, *Comput-Aided Des.* 31 (6) (1999) 449–457.
- [54] K. Klasing, D. Althoff, D. Wollherr, M. Buss, Comparison of surface normal estimation methods for range sensing applications, in: Robotics and Automation, 2009, ICRA'09, IEEE International Conference on, IEEE, 2009, pp. 3206–3211.
- [55] Velodyne LiDAR Inc., HDL-64E S2 and S2.1: High definition LiDAR sensor, 345 Digital Drive, Morgan Hill, CA 95037, 2008.
- [56] E.M. Applanix Corp., POS LV 420 V4, Data sheet, Applanix Corp., 2008.
- [57] T. Phillips, M. Green, P. McAree, An adaptive structure filter for sensor registration from unstructured terrain, *Journal of Field Robotics* (2014).
- [58] FARO Laser Scanner Focus^{3D}, FARO Technologies Inc., 2010.



Felipe Donoso is a Master of Philosophy student of the Mechanical and Mining Engineering School at the University of Queensland. He holds a bachelor in Electrical Engineering and a Master of Engineering Science from the Catholic University of Chile. Felipes research interests are in automation, robotic, perception and control. Felipes thesis topic is related with autonomous mapping for surface mining.



Kevin Austin received his Bachelor's degree in mechanical engineering from The University of Queensland, Australia in 1993, and his Ph.D. degree in Mechanical Engineering from the University of Queensland, Australia in 2002. From 2004 to present he has worked as a Research Engineer within the Smart Machines Group at the University of Queensland. Dr Kevin Austin's research interests are in the field of machine automation, with application to the mining sector.



Ross McAree is Professor of Mechanical Engineering in the School of Mechanical and Mining Engineering at the University of Queensland, having joined the university as a member of Academic Staff in 1999. He holds a Ph.D. from the University of Melbourne, awarded 1993. From 1993 to 1999 he was a post-doctoral research fellow in the Robotics Research Group at Oxford University. Professor McAree's research interests are in control, robotics, and automation with particular interest in the automation of mining machines. In October 2013 he was elected a Fellow of the Australian Academy of Technology and Engineering (ATSE).