Using Naturally Salient Regions for SLAM with 3D Laser Data[*]

David M. Cole, Alastair R. Harrison and Paul M. Newman
Oxford University Robotics Research Group
Department of Engineering Science
University of Oxford
Parks Road, Oxford, OX1 3PJ, UK
[pnewman,arh,dmc]@robots.ox.ac.uk
www.robots.ox.ac.uk/∼mobile

***Abstract*— We consider the task of processing 3D laser data for use in the Simultaneous Localization and Mapping Problem. The motivation for using 3D data comes in part from the impracticality of relying on 2D laser scanners when the vehicle operates on undulating terrain and in part from a desire to produce 3D maps of arbitrary, *a priori* unknown environments. We use an information-theoretic derived measure of local saliency to partition the raw 3D data stream into spatially distinct point-clusters. These clusters are natural features in measurement space that capture the geometry of intrinsically interesting surface patches. In common with "scan-matching" methods in $\mathcal{SE}_2$, the $\mathcal{SE}_3$ relationship between consecutive vehicle poses is calculated using an iterative point-wise registration scheme operating on the reduced data set. The saliency driven decimation process not only substantially reduces the computational burden of registration but also provides the registration process with data that is geometrically diverse. This characteristic improves registration performance. We present initial results showing our methods working on both outdoor and indoor data.**

***Index Terms*— SLAM, Outdoor, 3D, Features, Scan Matching, Geometric Saliency, Registration, Entropy**

I. Introduction and Motivation

The mobile robotics community has a good idea how to perform SLAM indoors on flat floors, especially when using the ubiquitous laser scanner. Fine progress has been made on the scaling problem and we now have several approaches that could enable large scale SLAM ([2], [3], [13], [14], [20]).

In this article we shall discuss our work towards deploying SLAM in the altogether more challenging outdoor domain. Here, in-plane motion relied upon by many large-scale contemporary SLAM implementations, is the exception rather than the norm. Odometry measurements are now reduced to excursions on an unknown manifold in $\mathbb{R}^3$. Furthermore outdoor environments have a very different character to those indoor. Outdoors the crisp geometric interiors of rooms and corridors are rare and we are more likely to see whole buildings, shrubbery, trees and rocks. Trying to use 2D laser scans when the robot is rolling and pitching relative to complex objects is a hopeless task. We need to sense in 3D if we are to recover $\mathcal{SE}_3$ pose information.

Our primary sensor is a 3D-scanning laser shown in Figure 1. This instrument allows us us to actively sample the surface geometry of the environment at high speed, over large distances, with little computational effort.

SLAM techniques using matches between patches of raw laser data are well suited to complex environments [6], [13], [15] and [2]. They offer two important advantages over popular feature based representations, such as those used in [11] and [19] for example. Firstly, they do not try to fit generative models that are prone to fitting bias, and secondly, all data can be used in the navigation process - not just the data that fits one member of an *a priori* set of geometric models. On the other hand the feature based representation has some very appealing properties. Data is stored compactly, and it can be easily applied (with good success) to navigation sensors other than the commonplace and somewhat beguiling range-bearing scanning laser - for example wide-beam sonar, radar or cameras.

At this point it is worth pointing out that geometric feature based approaches are not synonymous with the EKF. For example in [2] and [16] SLAM was achieved using an EKF with raw laser data scan-matching (both ICP and Correlation based) and a state vector of past vehicle poses while in some implementations of FastSLAM [20] geometric features were used in a particle filter formulation. We do not seek to resolve this debate over representation here nor can we reference all the literature that considers these issues. We simply make the point that there are advantages and disadvantages to both schools and this tension motivates our current research.

At first glance we might be tempted to proceed to the 3D case in the same manner as [7] or [8], and represent the world using sub-sampled raw data, combined with a 6 D.O.F. scan matcher. However this is an inefficient policy. Much of the data is either redundant or coming from a geometrically bland region that is non-informative vis-a-vis the navigation task, adding unnecessarily to the already heavy computational burden.

Simon [18] suggests that "Judicious selection and careful collection of a limited amount of data can result in better registration accuracy than random use of larger amounts

[*]This work is supported by EPSRC Grant #GR/S62215/01 and Guidance Control Systems Ltd.

of data.” Hence, careful sampling of data for registration could prove to be faster *and* more accurate. A more directed form of sub-sampling was introduced by Rusinkiewicz and Levoy [17]. They tried to ensure the overall distribution of point cloud normals was uniform after sub-sampling, in an attempt to maintain enough information in the two scans to give a well-conditioned registration. This was particularly important for their application as they were matching largely planar geometric objects with strong priors on shape. Unfortunately, we have less justification for such a method, as we have less prior knowledge about which data will provide sufficient registration constraints.

In this paper we propose a middle ground between raw data and geometric modelling, in which the raw measurement stream naturally produces its own features. Each ‘feature’ consists of an unaltered subset of the raw measurement set. The measurement stream is thus segmented into clumps of spatially localized measurements which can be passed simultaneously to a scan matcher to enable the navigation task.

The criteria for segmentation is local interest. Some measurements are more interesting than others because they come from more unusual or surprising parts of the scene[1]. We also note that raw data representations using scan matching techniques tend to excel when presented with data from complex scenes simply because complexity reduces the probability of ambiguity between two scans. In our experience, the opposite is frequently true for traditional feature-based techniques.

We segment the data by considering how the statistics of the geometry of the surfaces sampled by the laser change locally. In particular we look at how the entropy and raw distributions of normal directions change over scale around a particular data point. If for some scale around a given point, the distribution of normals changes rapidly then something locally ‘interesting’ is happening at that scale. If at this scale the patch is also locally complex i.e has high entropy, then we say the region is ‘salient’. Thus salient regions are complex and have similar statistical properties, as noted by Kadir and Brady [12].

The rest of this paper is concerned with the detection of salient regions in 3D geometric data and their use in scan matching, which will ultimately enable robust SLAM in 3D settings.

One immediate problem is that our 3D point clouds are of non-uniform density, due to sensor geometry. This can be detrimental to future processing, in this case surface normal generation and saliency detection. Therefore, resampling is essential. It is this task which, after a brief description of our measurement system in Section II, we turn to first in Section III. We proceed to cover our surface normal estimation technique and describe how we segment on saliency in Section IV. Section V then describes how the salient regions, which are point samples of an underlying salient patch, are passed through a non-linear optimizer (scan matcher) to find the transformation between vehicle poses in an outdoor setting.

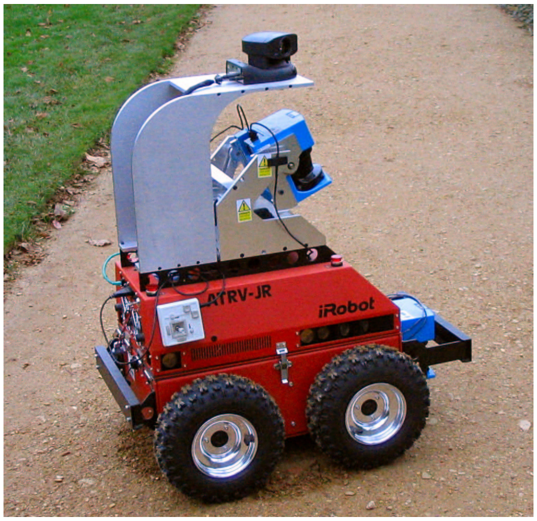


Fig. 1. Our 3D scanning laser range finder mounted on research vehicle ‘Marge’.

II. Data Acquisition

Data has been taken using a standard 2D SICK laser range finder, combined with custom built ‘nodding’ apparatus and interfacing circuitry.

The scanner is mounted in a reciprocating cradle, driven by a constant velocity motor, via a four bar chain quick-return mechanism. This combination is robust to jolts in locomotion encountered outdoors, as there is negligible backlash on the driving gear box. It can also sample scene geometry particularly fast compared to some servo-driven methods.

Scanner orientation is measured using a potentiometer mounted on the cradle’s axis of rotation. The elevation readings need to be accurately synchronized with the laser data stream - even a small delay can result in incorrect elevation labelling. This task is made awkward since the scanner’s elevation varies non-linearly with time, and the laser data packets currently arrive with a time lag, due to an RS422-USB conversion. Synchronization and angle-stamping is performed on the main PC, which estimates the data lag using a correlation technique based on the central laser beam’s range-elevation profile. More accurate hardware based methods are currently in development.

The entire 3D scanner has been mounted on our multi-terrain research vehicle, ‘Marge’, as pictured in Figure 1. Figure 2 shows typical point clouds from our sensor.

III. Surface Normal Estimation

A. Tangent plane estimation

1) Initial plane fit: We adopt the approach suggested by Hoppe, DeRose, Duchamp, McDonald and Stuetzle [9], which we now summarize.

Given a cloud of 3D data points, Ω_N, containing N individual points,$\{\omega_1, \ldots \omega_N\}$, sampled from a collection

[1] This of course is not a new insight; indeed it underlies the whole concept of a ‘landmark’ with which we are familiar.

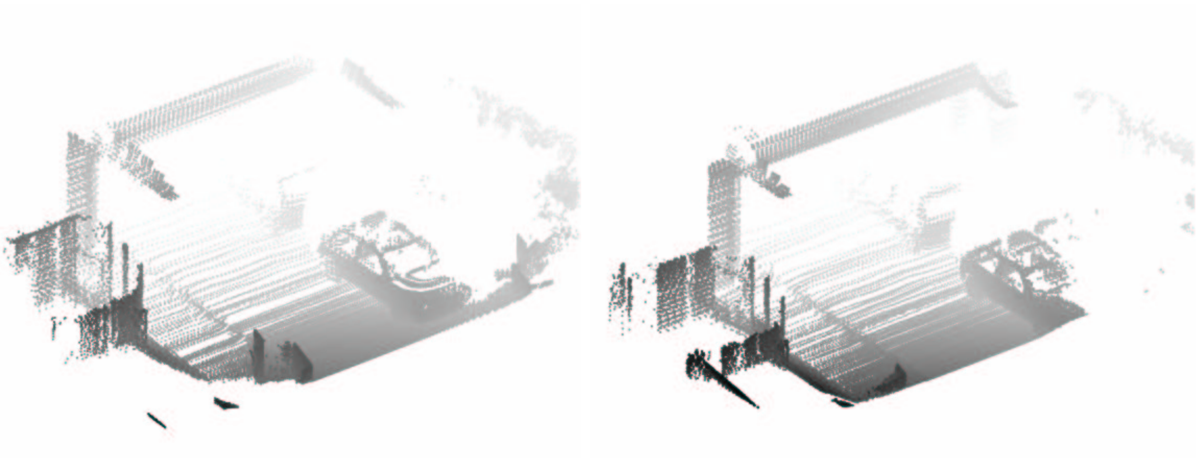


Fig. 2. Two consecutive 3D scan patches typical of those obtained by our 3D laser system. The gantry like object is part of an overhead building near an entrance to a building with glass doors —hence the rendering of the inside of an entrance hall on the left hand side.

of surfaces, we wish to obtain an estimate of the true surface normal, $\mathbf{n}_i$ for each point. This is achieved by performing a least-squares plane fit to ω_i and its k nearest neighbours. The extracted plane is assumed to be a local approximation of the true surface at ω_i and its normal $\hat{\mathbf{n}}_i$ approximates $\mathbf{n}_i$.

The k nearest neighbour searches can be quickly performed through the use of an efficient spatial partitioning data structure. Our implementation uses a kd-tree Λ_{Ω_N} containing all points in Ω_N.

2) Improved plane fitting: In areas of the point cloud affected by significant noise, the default neighbourhood size k may be insufficiently large to ensure a well conditioned plane fit. In such cases it is desirable to adaptively increase the size of the k-neighbourhood to reduce the influence of noise. If the condition number of the plane fit is poor, then k is increased and the plane fit recalculated. This process is repeated until the condition number falls below a threshold or k exceeds a predefined maximum. If at any step the condition number increases then the poor conditioning is assumed to be caused by local surface curvature, rather than noise.

B. Consistent tangent plane orientation

For illustrative purposes, Figure 3(a) shows a simple point cloud, sampled from three faces of a cube. Figure 3(b) shows the normals generated by the least-squares plane fit algorithm. Notice that they are all perpendicular to the surface, but are not consistently oriented. If we define the *front* of the surface to be the side closest to the point from which it was viewed, we require that the estimated normals are oriented consistently outwards from the front of the sampled surfaces.

Consider a pair of geometrically close points $\omega_1, \omega_2 \in \Omega_N$ on a densely sampled surface, with estimated normals $\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_j \in \mathbb{R}^3$. Assuming that $\mathbf{n}_i$ is correctly oriented, $\mathbf{n}_i \cdot \mathbf{n}_j > 0$ implies that $\mathbf{n}_j$ is also correctly oriented. If $\mathbf{n}_i \cdot \mathbf{n}_j < 0$, then $\mathbf{n}_j$ must be flipped. Now that $\mathbf{n}_j$ is correctly oriented, neighbours of ω_j can be corrected, and so on. We may continue propagating normal consistency in this manner until all points in Ω_N have been correctly oriented.

In practice, the order of propagation is important. Propagation of normal orientations over sharp edges is prone to error and may result in gross local inconsistency in orientation.

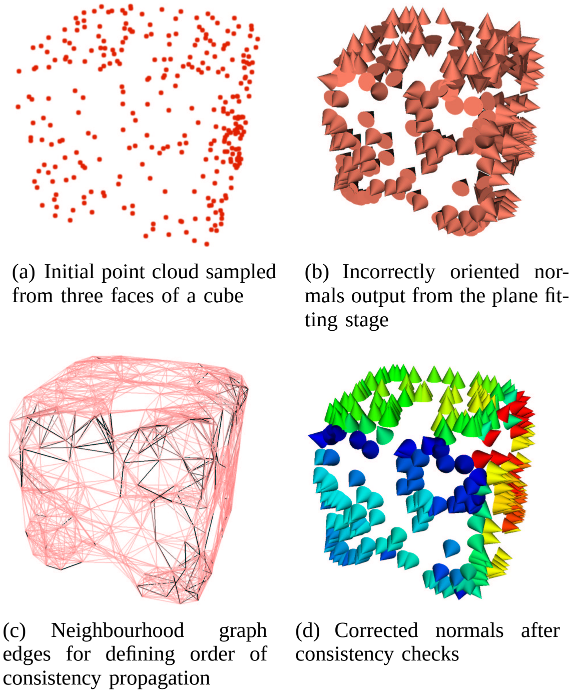
To improve the order of propagation, we construct a graph G, which contains an edge $\langle \omega_i, \omega_j \rangle$ if ω_j is in the k-neighbourhood of ω_i. Each edge is assigned a cost, $c = 1 - |\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j|$. The neighbourhood (or Riemannian) graph for the point cloud from figure 3(a) is shown in figure 3(c), where darker edges denote a higher cost. Normal consistency is propagated by traversing a *minimal spanning tree* of G. This method has the effect of propagating preferentially across areas of low curvature. Sharp edges are avoided unless there is no other path available to reach a particular point.

Since we have viewpoint information in the form of the robot pose, we may use this to improve the results of the previous step[2]. Given a set of viewpoints $V = \{\mathbf{v}_1, \dots \mathbf{v}_m\}$, we add each viewpoint as a pseudo-node in G. We add edges from each viewpoint node to every point 'seen' from that viewpoint. The edge cost is assigned as $c = 1 - |\frac{(\mathbf{v}_j - \omega_i)}{|\mathbf{v}_j - \omega_i|} \cdot \hat{\mathbf{n}}_j|$. Figure 3(d) shows the correctly oriented normals.

C. Point Density Normalization

Real 3D range data from the robot's laser range finder exhibits large variations in sample density due to the divergence of consecutively sampled beams. Surfaces close to the scanner are particularly densely sampled.

[2] If we have viewpoint information, it might appear that the correct orientation of $\hat{\mathbf{n}}_i$ could be set trivially. Given a viewpoint $\mathbf{v}_j$, we might flip $\hat{\mathbf{n}}_i$ if $(\hat{\mathbf{n}}_i \cdot \frac{(\mathbf{v}_j - \omega_i)}{|\mathbf{v}_j - \omega_i|}) < 0$. For points located near sharp edges, the plane fit is ill conditioned, resulting in an incorrect surface estimate. In these cases the viewpoint may lie behind the estimated plane, causing this algorithm to choose the wrong orientation.



(a) Initial point cloud sampled from three faces of a cube

(b) Incorrectly oriented normals output from the plane fitting stage

(c) Neighbourhood graph edges for defining order of consistency propagation

(d) Corrected normals after consistency checks

Fig. 3. An illustrative problem: finding consistent normals from a point cloud

An appropriately large neighbourhood to ensure a well conditioned least-squares plane fit on a densely sampled surface may contain thousands of points. When using the plane-fitting algorithm on real range data, we find that it is usually beneficial to apply a pre-processing step, to reduce the point density in these areas.

We obtain an estimate of the surface sample density at each point $\rho_i = k/r_i^2$, where r_i is the distance to the kth nearest neighbour of ω_i.

If ρ_i is greater than a desired threshold then $\omega_{\mathbf{i}}$ is removed from Ω_N. Whilst suboptimal, this simple heuristic performs well in practice and can be implemented efficiently within the kd-tree framework.

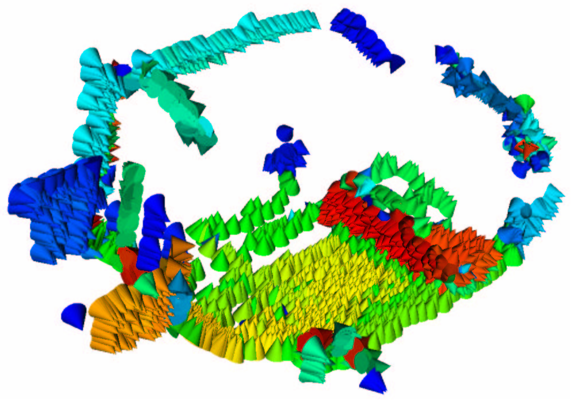


Fig. 4. The result of applying the normal estimation process to the scene shown in Figure 2. The coloring indicates the traversal order of the MST from red to blue.

IV. Saliency Detection

Given a 3D point cloud, Ω_N, containing N individual points, our aim is to select a subset Ω_n, containing $n \ll N$ individual points. Our motivation is to facilitate fast 6 D.O.F. registration with another 3D scan of the local area. The subset Ω_n will be formed from the union of I geometrically salient regions.

Initially, a selection B of points in Ω_N is chosen, to be used as base points for saliency detection. In this initial work, they were chosen by random sampling, but more sophisticated approaches are preferable.

For each base point, a series of concentric spheres, S_p, are grown, where $p = [1 : P]$, and sphere radius = $p \times \delta r$. This is illustrated in the top half of Figure 5, again, for the simple illustrative case of points on the surface of a cube.

The points lying in each sphere, S_p are efficiently computed incrementally from the previous sphere, S_{p-1}.

For convenience, we will now define S_p^i to be the set of points lying in the p^{th} sphere around base point ω_i. It is now possible to use each point's attached estimated surface normal (as calculated in Section III), to construct a histogram (or PDF) of surface normals in S_p^i. As $p = [1 : P]$, and i varies according to the B base points chosen, this generates $P \times B$ histograms.

Each 2D histogram quantizes the surface normals in S_p^i into a discrete set of (ϕ, θ) spherical polar co-ordinate bins centred on ω_i (where θ is divided into θ^+ bins, and ϕ into ϕ^+ bins). This is illustrated in the bottom half of Figure 5.

We calculate the entropy of each sphere's 2D histogram as follows, where $p(\phi, \theta)$ represents the histogram height of each spherical polar bin:

$$H(S_p) = -\sum_{\theta=1}^{\theta^+} \sum_{\phi=1}^{\phi^+} p(\phi, \theta) \log_2 p(\phi, \theta) \qquad (1)$$

Having calculated the variation of surface normal entropy, $H(S_p)$, over scale, or sphere radius, r, for each base point, we can derive the corresponding variation of saliency over scale. We adopt the definition of saliency as suggested by Kadir and Brady [12], which can be written as follows:

$$Y(S_p) = H(S_p) \times W(S_p) \qquad (2)$$

Where

$$W(S_p) = r \times \sum_{\theta=1}^{\theta^+} \sum_{\phi=1}^{\phi^+} \left| \frac{\partial p(\phi, \theta)}{\partial r} \right| \qquad (3)$$

Selecting salient regions in this manner can be thought of as a 'weighting' of three separate factors, each of which contribute to the overall interest of the region. The $H(S_p)$ term infers that the region must have a sufficient degree of complexity or entropy to be salient. The $|\frac{\partial p(\phi,\theta)}{\partial r}|$ suggests interesting regions have to exhibit a change in statistics over the scale of interest. Finally, the r term biases the results to favour larger regions, and not to accept overly small regions which can be dominated by noise.

We can now define a geometrically salient region as the set of points lying within a sphere centred at a base point and with radius corresponding to a saliency maximum.

In practice this gives a large number of geometrically salient regions, and it is therefore important to only consider those regions with the largest I saliency peaks. Furthermore, one should only consider peaks over scales of interest. This prevents finding geometrically salient regions that, for example, cover the entire data space. It is also necessary to provide a geometric constraint, such that the base points of salient regions are not too close to each other. Otherwise, effectively the same salient region will be found many times, with slightly differing base points. This is a straightforward clustering problem. The algorithm is summarized in Figure IV.

V. 3D Registration

All techniques for scan matching or registration directly or implicitly perform minimization of an error metric defined over two scans. Without loss of generality, we shall refer to one set as the model, and consider it fixed, and the other set as the data, the points that need aligning. We shall define $\mathbf{m}_i$ as the i^{th} model point, where M is the number of model points and $\mathbf{d}_i$ as the i^{th} data point, where D is the number of data points. Our task is to find $\mathbf{a}$, the transformation vector to align the data points to the model points. In this case, it will contain $x, y, z, \theta_1, \theta_2, \theta_3$ parameters for the full 6 D.O.F. Euclidean transformation. We will also define $T(\mathbf{a}, \mathbf{m}_i)$ as the transformation operator taking the i^{th} model point and transforming it by $\mathbf{a}$.

The general error metric takes the following form:

$$E(\mathbf{a}) = \sum_{i=1}^{D} \parallel \mathbf{m}_{\phi(i)} - T(\mathbf{a}, \mathbf{d}_i) \parallel^2 \tag{4}$$

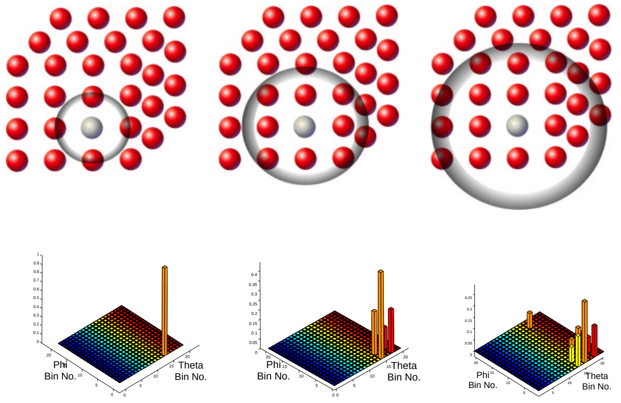


Fig. 5. An example using the points on the surface of a cube. For the base point chosen, spheres of increasing radius are grown. Corresponding (ϕ, θ) histograms of the surface normals for the 3 example spheres are also shown.

```
input : Point Cloud $\Omega_N$,with corresponding normals
output: Point Cloud $\Omega_n$
$\Lambda_{\Omega_N}$=BuildTree($\Omega_N$);
$ndx$=SelectBasePts($\Omega_N$,$B$);
for $BasePt = 1 : B$ do
    i=ndx[BasePt];
    for $p = 1 : P$ do
        $r = p\delta r$;
        $S_p^i$=QueryTree($\Lambda_{\Omega_N}$,$i$,$r$);
        $\mathcal{H}_p^i$=BuildHist($S_p^i$);
        $Y(S_p^i)$=FindSaliency($\mathcal{H}_p^i$);
    end
    $Y(S_{p_{max}}^i)$=MaxOverScale($Y(S_{1:P}^i)$);
end
$[S_{p_{max}}^{i_{1st}} : S_{p_{max}}^{i_{Ith}}]$=MaxOverBasePts($Y(S_{p_{max}}^{ndx[1:B]})$,$I$);
$\Omega_n = \bigcup[S_{p_{max}}^{i_{1st}}, ...., S_{p_{max}}^{i_{Ith}}]$ ;
```

Algorithm 1: Extracting Points belonging to Geometrically Salient Regions

Correlation based scan matching methods, such as those used in [13], effectively look for cost function minima using brute force searches by varying the parameters of $\mathbf{a}$. The combination of parameters that gives the lowest cost is accepted as the optimal match. Whilst only searching around the initial transformation estimate makes these methods acceptable in low dimensions (as in 2D scan matching), they scale poorly into higher dimensional spaces. Here, directed searches such as ICP [1] are a good choice.

The Iterative Closest Point (ICP) algorithm introduced by Besl and McKay [1] takes the form described above in equation 4, but with a particular function $\phi(i)$ (which encapsulates the correspondence between points). As the name suggests, Iterative Closest Point uses the nearest model point to the data point as its correspondence.

Consequently, the error metric to minimize becomes:

$$E(\mathbf{a}) = \sum_{i=1}^{M} (\min_{j} \parallel \mathbf{m}_j - T(\mathbf{a}, \mathbf{d}_i) \parallel)^2 \tag{5}$$

Two stages per iteration are required: correspondence calculation, and then a minimization step. This latter stage is often performed using a closed form singular value decomposition or a least squares quaternion based method as described by Horn [10].

One particularly useful variant of ICP, applicable to the mobile robotics domain, first suggested in [1], uses a kd-tree data structure to make correspondence searches as fast

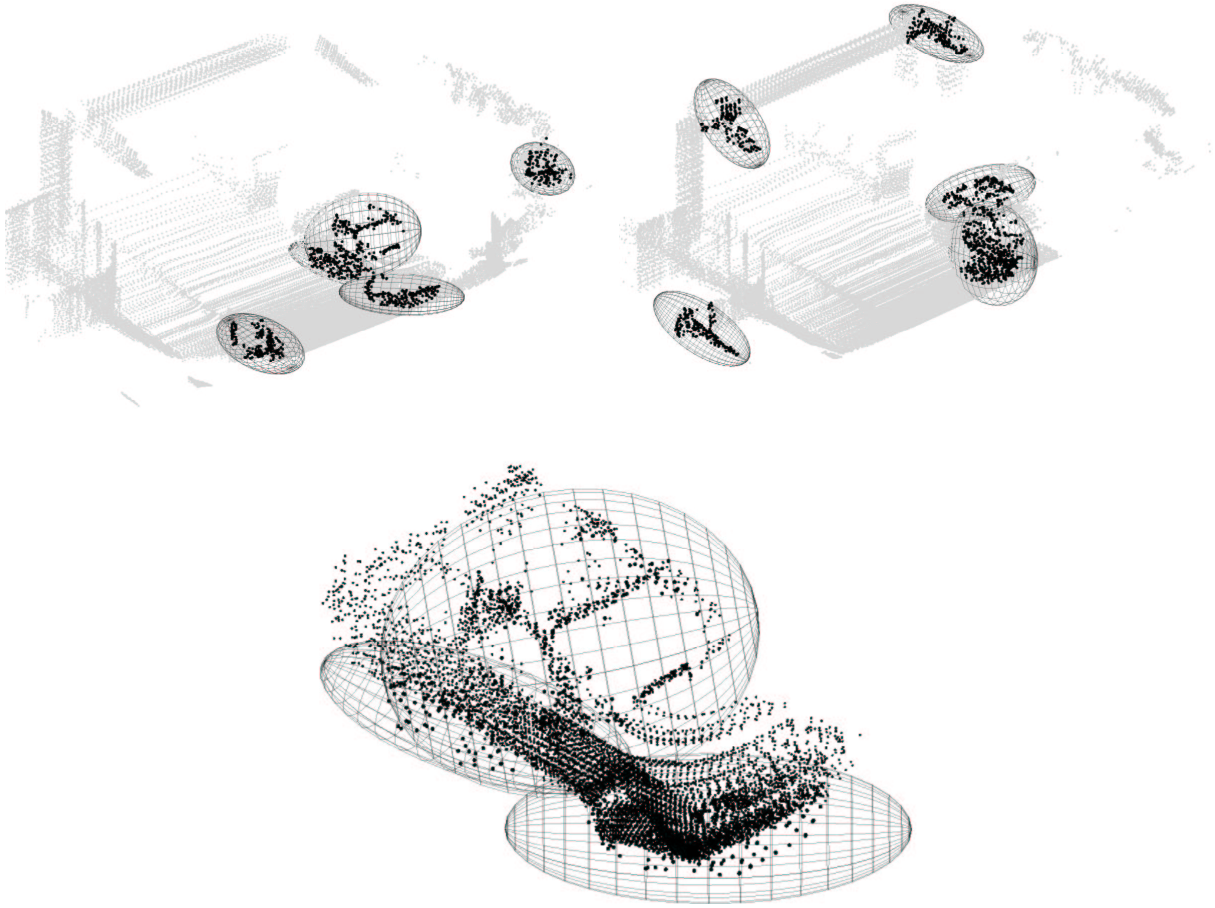


Fig. 6. The same data as shown in Figure 2 but with the top five salient regions highlighted in each case. Note how the shape of the car is categorized as salient along with the multifaceted corners of the overhead concrete gantry. The ellipsoids are for illustrative purposes only and are scaled covariance ellipsoids of the raw data points which constitute each of the salient regions. They serve to illustrate their crude shape and location.

as possible. This has since been accelerated by Greenspan and Yurick [5], who use approximate kd-trees for further saving. This is the basic method behind several successful implementations of 3D ICP in the mobile robotics community, such as those by Surmann, Nüchter and Hertzberg [7] and Surmann, Nuchter, Lingemann and Hertzberg [8]. However, it is difficult to apply robust statistical techniques to ICP in this form, leading to relatively narrow convergence basins.

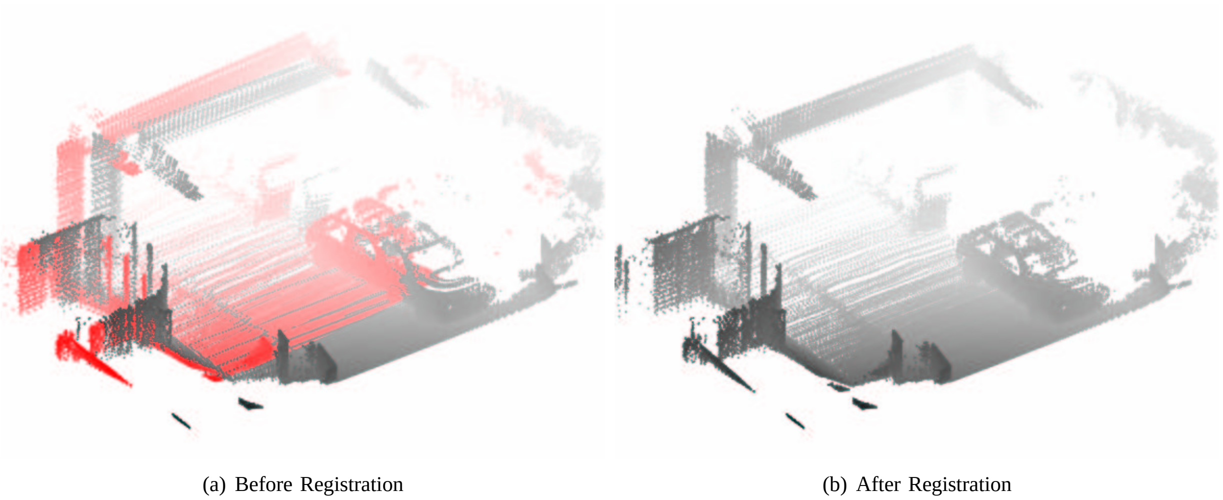
Fitzgibbon [4] minimizes a similar metric as ICP, but uses a standard non-linear optimizer (Levenberg-Marquardt) to perform the minimization. This means that a robust kernel can be applied, so that correspondences very far apart do not overly bias the transformation, resulting in a widened convergence basin. In [4], the Chamfer Distance Transform is used to implicitly precompute correspondences and allow fast look up of distances, although a kd-tree approach scales better with higher dimensions. The non-linear optimizer is accelerated by allowing correspondences to vary during calculation of local error surface characteristics (the Jacobian), in particular contrast to standard ICP. Similar work has been carried out by Zhang, Hall-Holt and Kaufman [21].

In this paper we use a cauchy robust kernel within a Levenberg-Marquardt non-linear optimizer. Correspondences are found efficiently using approximate kd-tree searches. This technique proves as efficient as ICP in [8], combined with the advantage of wider convergence basins.

VI. Results

Data was taken using the apparatus in Section II, mounted on Marge, our all terrain research vehicle. The environment chosen for data collection had two important characteristics; a non-flat surface, and complex geometry. Figure 2 shows two consecutive raw point clouds. Each of these point clouds was subsequently passed through the normal vector field algorithm summarized in Section III. A typical normal field is shown in Figure 4.

For each 3D scan patch considered, both the raw point cloud and its corresponding normal vector field was inserted into the saliency detection pipeline described in Section IV. Figure 6 shows the geometrically salient regions returned for a typical patch. The set of salient regions generated by any two patches are finally passed into the registration or scan-matching scheme described in Section V. The output of this final step is a 6 D.O.F. transformation



(a) Before Registration

(b) After Registration

Fig. 7. Registration using salient regions. The salient regions of Figure 6 are registered using the method described in Section V. To illustrate the problem, the left hand Figure shows two data sets with an artificially elevated relative orientation and translational error. The right hand figure shows the same two data sets after registration. Note how the use of a robust kernel (e.g Huber or Cauchy) means that we do not require the same number of data points or clusters in each patch.

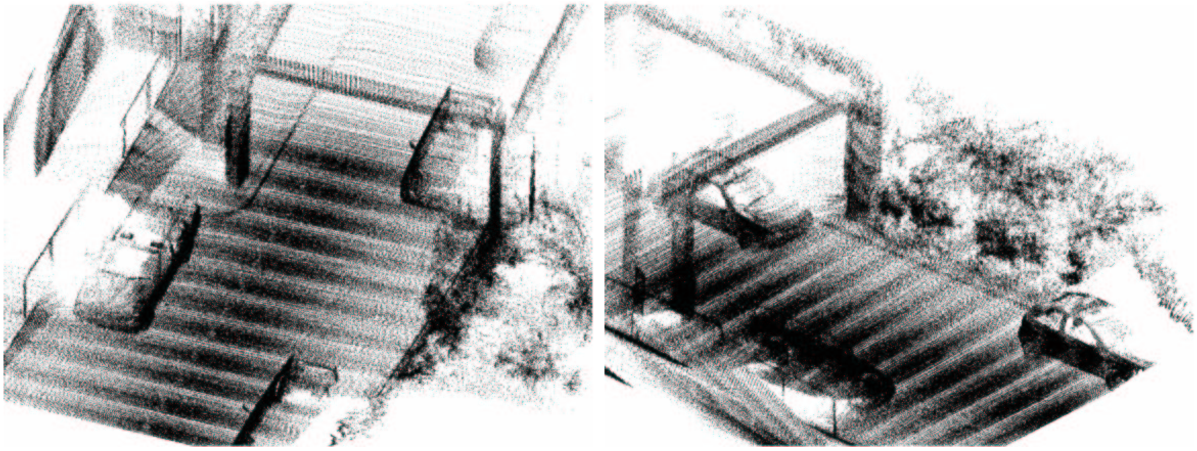


Fig. 8. A short dose of SLAM using sequential registration of 6 D.O.F. pose states. The above figure is generated by sequentially registering the salient regions from twelve 3D scan patches captured at 1m intervals. The entirety of each scan patch is rendered from the corrected vehicle position.

between consecutive poses. Where applicable, these estimated transformations can be used to correct dead reckoned versions. Figure 7 shows two scan patches before and after alignment. Finally, Figure 8 shows the outcome of repeating this procedure over a sequence of twelve 3D patches.

VII. Conclusions and Future Work

In this paper we have shown how geometric saliency can be used to segment raw 3D-laser data. The result of this process is a collection of informative “data-features” that can be used in a robust, 3D, inter-pose registration scheme. We have shown this process working on data gathered from an outdoor urban scene. We do not limit the relevance of this work to registration based techniques alone — the role of saliency in natural feature generation is interesting in its own right.

There are several areas which we wish to investigate further. Firstly, there is the issue of deciding which of the laser samples we should use as base points for saliency detection i.e. around which points to examine how the saliency changes over scale. Considering every point over every scale is computationally expensive and highly redundant, especially in dense point clouds. At the moment we adopt a naive approach and simply randomly choose and examine a small fraction of the total number of data points. It seems sensible to use a more informed selection process — how to do this is an area of current research.

Secondly, there is the issue of reflecting the needs of the registration technique in the segmentation process. Our initial results indicate that the segmentation described in this paper does indeed sub-sample the measurement stream in a way that allows accurate 3D registration. In many respects we are not surprised, after all, the segments are created precisely because they possess an intrinsically rich geometry. Nevertheless we would like to be able to bias the segmenter to yield segments that are, in concert with each other, more likely to cause stable, well conditioned inter-pose registrations.

Finally there is the compelling need to add visual appearance to the mix. We envision using “colored data” with hue derived from smoothed visual images. This higher dimensional data will not only offer additional saliency criteria but also aid in the registration process itself.

In conclusion, pre-selection of geometrically interesting regions for use in SLAM using 3D laser data is a beneficial, promising and viable technique and deserves further research.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[2] M. Bosse, P. Newman, J. J. Leonard, and S. Teller. Slam in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 24:1113–1139, 2004.

[3] A. Eliazar and R. Parr. DP-SLAM 2.0. *ICRA*, 2:1314 – 1320, May 2004.

[4] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21:1145–1153, 2003.

[5] Yurick M. Greenspan, M.A. An approximate k-d tree search for efficient icp. In *3DIM03: 4th International Conference on 3-D Digital Imaging and Modeling*, Banff, Alberta, Canada, Oct 2003.

[6] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *International Symposium on Computational Intelligence in Robotics and Automation*, Monterey California USA, November 1999.

[7] A. Nuchter H. Surmann and J. Hertzberg. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 2003.

[8] K. Lingemann H. Surmann, A. Nuchter and J. Hertzberg. 6d slam - preliminary report on closing the loop in six dimensions. In *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon Portugal, July 2004.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[10] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer.*, 1987.

[11] J. Neira J. A. Castellanos, J.M.M. Montiel and J.D. Tardós. The spmap: A probabalistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–952, 1999.

[12] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal Computer Vision*, pages 83–105, 2001.

[13] K. Konolige. Large-scale map-making. *Proceedings of the National Conference on AI (AAAI), San Jose, CA*, 2004.

[14] J.J. Leonard and P.M. Newman. Consistent, convergent, and constant-time slam. In *International Joint Conference on Artificial Intelligence*, Acapulco Mexico, August 2003.

[15] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[16] P. Newman and K. Ho. Slam - loop closing with visually salient features. In *To be published in IEEE International Conference on Robotics and Automation*, 2005.

[17] M. Levoy S. Rusinkiewicz. Efficient variants of the icp algorithm. In *3DIM01: 3rd International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, May/June 2001.

[18] D. Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996.

[19] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. pages 167–193, 1990.

[20] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 2004. To appear.

[21] H. Zhang, O. Hall-Holt, and A. Kaufman. Range image registration via probability field. In *CGI ’04: Proceedings of the Computer Graphics International (CGI’04)*, pages 546–552. IEEE Computer Society, 2004.