Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Multi-scale Point and Line Range Data Algorithms for Mapping and Localization

Samuel T. Pfister and Joel W. Burdick
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, California 91125, USA
Email:{*sam,jwb*}@*robotics.caltech.edu*

***Abstract*— This paper presents a multi-scale point and line based representation of two-dimensional range scan data. The techniques are based on a multi-scale Hough transform and a tree representation of the environment's features. The multi-scale representation can lead to improved robustness and computational efficiencies in basic operations, such as matching and correspondence, that commonly arise in many localization and mapping procedures. For multi-scale matching and correspondence we introduce a χ^2 criterion that is calculated from the estimated variance in position of each detected line segment or point. This improved correspondence method can be used as the basis for simple scan-matching displacement estimation, as a part of a SLAM implementation, or as the basis for solutions to the kidnapped robot problem. Experimental results (using a Sick LMS-200 range scanner) show the effectiveness of our methods.**

I. Introduction and Preliminaries

Autonomous mobile robot navigation in unknown environments requires effective localization and mapping. Consequently, the subjects of localization and mapping have received enormous attention (e.g., see [1], [2], [3], [4], [5], [6]). Some tasks in localization and/or mapping require high quality correspondences to be established between sensor data collected at different times and different robot positions. Examples of localization and mapping related tasks that require correspondence include scan-matching based odometry [7] and the kidnapped robot problem [8].

In developing practical correspondence and localization methods, it can be difficult to simultaneously achieve the three objectives of overall robustness to mismatches and errors, the flexibility to handle a wide range of environmental types and conditions, and computational efficiency. This is particularly true for large and complex environments. While range scan matching methods such as [7] and [9] are very flexible as they don't assume any a priori knowledge of the environment, the iterative correspondence required to match range points can be less robust to poor initial position estimates and the method is computationally burdensome for a large map due to the potentially large number of point features. This paper tries to address these problems.

This paper introduces multi-scale feature extraction and data correspondence methods that can potentially be applied to a number of problems related to localization and mapping. Figure 1 shows the method applied to actual data from a planar

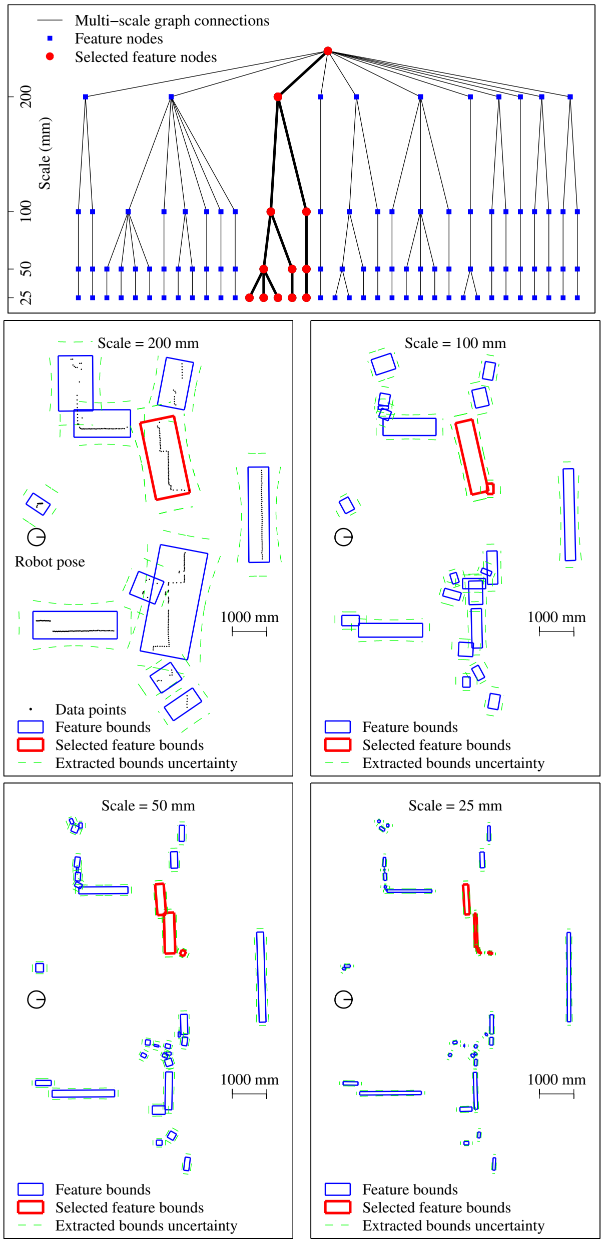

Fig. 1. Multi-scale range scan representation: (a) scale tree graph; (b)-(d) a sequence of increasingly fine scale representations of the data.

0-7803-9505-0/06/$20.00 ©2006 IEEE

laser range scanner. Using line-fitting techniques (e.g., [10], [11], [12]), one can fit lines to the range data points in order to reduce the data complexity. In this paper we introduce a multi-scale version of line fitting and feature extraction that represents range data with line segments whose line widths increase as the scale becomes coarser. (e.g., see Figures 1(b)-(e)). The features are based on augmented line segments (or *blocks*), though the feature class can encompass very short segments or even single points in the case of unstructured data. The interrelation between features at each scale are represented as a tree structure (Fig. 1(a), where the red nodes and branches in the tree correspond to the red features in the Fig.s 1(b,c,d,e)).

This multi-scale feature representation can be used for:

1) Multi-scale scan-based odometry. Iterative closest point (ICP) algorithms are frequently used to register two scans as part of scan-matching based odometry [7], [9]. Based on a multi-scale representation of the scans to be registered, one can use a coarse to fine traversal of the data trees to greatly reduce the search space for correspondences at fine scales. For example, at the coarsest scale (Fig. 1(b)), the 360 data points are represented by 12 coarse line features. These coarse features can provide a very fast initial registration, which is then refined at finer scales. Potentially, this approach will result in more robust correspondence and fewer iterations.

2) Multi-scale feature correspondence for matching. A coarse-to-fine data traversal can also simplify the computations required to determine if two scans are a match. To properly implement such a matching method, one needs a corresponding multi-scale χ^2-test, which we provide in this paper.

3) Kidnapped robot problems. The solution to the kidnapped robot problem [8] invariably involves matches between a sensor scan from the robot's current configuration with a data base of geometric data to find the most likely robot configuration. The coarse scale representation allows a very efficient search of the data base. Many possible mismatches can be eliminated at the coarse scales, where the computational complexity of the matching procedure is small.

We focus on these first two issues in this paper, as the multi-scale approach to solving the kidnapped robot problem is a direct consequence of these issues. Our preliminary experimental results have shown that multi-scale methods provide advantages (as compared to single scale data analysis) in terms of robustness, programming flexibility, and computational effort.

Relation to prior work. It has long been recognized that the computational complexity of mapping and localization can scale prohibitively with the amount of sensor data acquired by the robot and the number of landmarks encountered by the robot during its mapping and localization processes. A number of authors have focused on different techniques to reduce the computational complexity associated with various aspects of mapping and localization. For example, sparsification of the information matrix can significantly increase the efficiency of Kalman filter based SLAM methods [13], [14]. Rao Blackwellization has also been used to increase the efficiency of particle filtering based SLAM algorithms (e.g, the FastSLAM algorithm[15]). Finally, feature based SLAM methods attempt to reduce computational requirements by feature-based data reduction [16]. Our approach, which focuses on multi-scale feature extraction and multi-scale correspondence, is most related to this last subject. However, in addition to computational complexity, our work is also concerned with robustness to data errors.

Prior work has looked at some aspects of multi-scale data processing for localization, mapping, and navigation. Madhavan et. al. [17] applied the classical scale space approach of Gaussian smoothing to range data. A number of authors have used multi-scale methods for efficient environment representation or planning [18], [19], [20]. However, these prior works did not attempt multi-scale feature extraction and correspondence.

This work uses a multi-scale Hough transform to efficiently extract multi-scale line segment features from planar range data. There is a large existing body on fitting single scale lines to range data for purposes of mapping and localization [21], [11], [12], [22]. In the field of computer vision there have been prior efforts to develop a multi-scale Hough transform [23], [24], [25] though our particular version of the multi-scale Hough transform in this paper appears to be unique. The motivation for much of the prior work in the computer vision field is to use a multi-scale approach to increase the efficiency of the extraction process [26]. We see similar efficiency benefits in our extraction methods, but the primary contribution of this work is how we utilize our multi-scale feature representation. Using the multi-scale Hough transform as a foundation, we introduce algorithms for multi-scale feature correspondence and displacement estimates that take advantage of the multi-scale data structure to improve robustness and reduce computational complexity. In this work we focus on line-based Hough transforms, however, the general properties of the Hough transform should allow multi-scale extensions to other features types (such as circular arcs and curves).

This paper is structured as follows. Section II introduces a polar representation for lines (see [10] for more details) and the notion of line scale. Section III describes the multi-scale Hough transform while Section IV reviews the scale tree dendogram that organizes the multi-scale features extracted from the Hough transform. Section V describes a χ^2 test for multi-scale matching. Section VI provides experimental examples and results that demonstrate the use of our multi-scale methods to improve the efficiency of the correspondence search and to implement robust scan matching based odometry.

II. Multi-scale Feature Representation

This section develops the scaled feature representation which forms the core of our approach. Our *block* feature is a multi-scale extension of a line segment. We first outline an infinite line representation and its covariance, and then build on this representation to develop the block feature.

A. Line Representation

We represent an infinite line, L, in polar coordinates as:

$$L = \begin{bmatrix} \alpha \\ \rho \end{bmatrix} \tag{1}$$

where ρ and α represent the magnitude and heading of the vector which extends from the origin to L and which is perpendicular to L. Thus α and ρ respectively define the "orientation" and "position" of L (see Fig. 2). Our lines will be derived from noisy data. Therefore, we define a covariance matrix associated with L as:

$$P_L = \begin{bmatrix} P_{\rho\rho} & P_{\rho\alpha} \\ P_{\alpha\rho} & P_{\alpha\alpha} \end{bmatrix} \tag{2}$$

where $P_{\rho\rho}$ is the variance in the line's position ρ, $P_{\alpha\alpha}$ is the variance in the line's orientation α, and $P_{\rho\alpha} = P_{\alpha\rho}$ are the cross-correlation terms. Fig. 2 graphically depicts the uncertainty of an infinite line L with $P_{\rho\rho}$ and $P_{\alpha\alpha}$ shown. In subsequent figures, coupled orientation and position uncertainty bounds for line segments will be represented as a hyperbola with the distance between the vertices determined by $P_{\rho\rho}$ and the asymptotes determined by $P_{\alpha\alpha}$. The cross terms of the covariance matrix determine where along the line the asymptotes intersect.

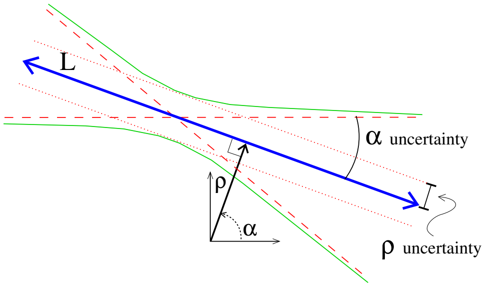

Fig. 2. Representation of line L and its uncertainty bounds.

B. The "Block" Feature Representation

A *block feature* extends the notion of a line segment to a multi-scale setting and allows for flexibility in representing complex sets of point data. While we do assume a Gaussian distribution on the uncertainty of the position of the block boundaries, we don't need to make any assumptions or abstractions on the distribution of the set of points represented by the block. We are therefore able to develop algorithms using the block feature which are less sensitive to residue from the scanning process and scanning geometry and more descriptive of the underlying environment.

A block feature, B, is represented as follows:

$$B = \begin{bmatrix} \alpha & \rho_a & \rho_b & \psi_a & \psi_b \end{bmatrix}^T \tag{3}$$

where ρ_a and ρ_b define the position of a pair of bounding infinite lines at angle α. We define the ψ axis to be perpendicular to the ρ axis and the coordinates ψ_a and ψ_b represent the locations of the endpoints of the feature as measured colinearly with the underlying line (with ψ taking a zero value where the perpendicular vector intersects L). These coordinates represent a rectangular block oriented at α, with a width $w_B = (\rho_b - \rho_a)$ and a length $l_B = (\psi_b - \psi_a)$ as seen in Fig. 3.

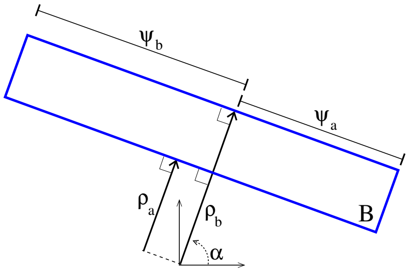

Fig. 3. Block B representation.

Practically, block features can be interpreted as a line segment of length l_B and non-zero width w_B. The "width" of the line segment feature is proportional to scale, and it approaches a zero-width line segment ($\rho_a \simeq \rho_b$) at the finest scale.

Though we focus primarily on line-segment-like features, our representation is also flexible enough to represent blob-like groups of points without the high length-to-width ratio of a line-like feature. In these cases the notion of feature orientation α becomes less meaningful but the blob features can still be used effectively in a correspondence and localization scheme.

The covariance matrix associated with block feature B is:

$$P_B = \begin{bmatrix} P_{\alpha\alpha} & P_{\rho_a\alpha} & P_{\rho_b\alpha} & P_{\psi_a\alpha} & P_{\psi_b\alpha} \\ P_{\alpha\rho_a} & P_{\rho_a\rho_a} & P_{\rho_b\rho_a} & P_{\psi_a\rho_a} & P_{\psi_b\rho_a} \\ P_{\alpha\rho_b} & P_{\rho_a\rho_b} & P_{\rho_b\rho_b} & P_{\psi_a\rho_b} & P_{\psi_b\rho_b} \\ P_{\alpha\psi_a} & P_{\rho_a\psi_a} & P_{\rho_b\psi_a} & P_{\psi_a\psi_a} & P_{\psi_b\psi_a} \\ P_{\alpha\psi_b} & P_{\rho_a\psi_b} & P_{\rho_b\psi_b} & P_{\psi_a\psi_b} & P_{\psi_b\psi_b} \end{bmatrix}. \tag{4}$$

It is often useful to separate the feature into sub-elements (line edges, corner points) when comparing with other features as we will discuss in Section V. Though many of the cross terms for this covariance representation can be assumed to be zero for our method of feature extraction, all of our χ^2-based comparison methods can operate on the full covariance matrix.

III. Feature Extraction

The goal of the feature extraction process is to sort a set of n range points $V = \{v_1 ... v_n\}$ into m roughly colinear point subsets V_k, $k = 1, \ldots, m$ while at the same time estimating the block parameters (B, P_B). Note that m is not a predetermined value, and it will depend upon the scale. Extraction of the m feature coordinates is an iterative, two step process. In the first step, the underlying line coordinates $[\alpha, \rho_a, \rho_b]$ are extracted from the data using an augmented Hough transform (see below). In the second, the endpoints $[\psi_a, \psi_b]$ are estimated. Because of truncation performed during the endpoint estimation, the second step can have an effect on the optimal estimation of the underlying infinite line coordinates. We therefore repeat these steps iteratively until the process stabilizes, usually in three or fewer iterations.

A. Multi-Scale Hough Transform

To extract the underlying infinite line coordinates $[\alpha, \rho_a, \rho_b]$, we first transform the range points using a multi-scale version of the Hough transform. Let us first briefly review the classical Hough transform. Define a Hough space $\mathcal{H}(i,j)$ as a two dimensional raster with integer indices i and j indexing the variables $\rho(i)$ and $\alpha(j)$ respectively. The variable $\alpha(j)$ is discretized in increments of D_α on the range $[-\pi/2, \pi/2]$ and the variable $\rho(i)$ is discretized in increments of D_ρ on the range $[-d_{max}, d_{max}]$ where d_{max} is the maximum range value to be expected in the data point set. We choose our discretization level D_α as a function of the discretization level D_ρ and the maximum sensor range l_{max} :

$$D_\alpha = \tan^{-1}(D_\rho / l_{max}) \ . \qquad (5)$$

Cell $\{i,j\}$ of the discretized Hough space therefore represents the range of line coordinates $[\rho(i) \pm D_\rho/2 \ \alpha(j) \pm D_\alpha/2]$. The content of each cell in the Hough raster is initially set to zero. For each range point, for all i we calculate the position ρ_{ik} of the line at angle $\alpha(i)$ that would pass through point k:

$$\rho_{ik} = x_k \sin(\alpha(i)) + y_k \cos(\alpha(i)) \qquad (6)$$

where x_k, y_k are the coordinates of the k^{th} range data point. From this value of ρ_{ik}, determine the index j^* such that $\rho(j^*) - D_\rho/2 < \rho_{ik} \leq \rho(j^*) + D_\rho/2$. The value at Hough space cell $\mathcal{H}(i, j^*)$ is incremented. This process is repeated for every range point. The cell in Hough space with the highest incremented value corresponds to the line which has the most contributing points.

The traditional Hough transform simply detects peaks in the Hough space and defines lines from the peaks' coordinates. We augment this technique in order to determine a sense of the scale in the width of the detected lines. First determine the angle coordinate α of a peak in the Hough space. Then extract the one dimensional signal $\Gamma(i) = \mathcal{H}(i,\alpha)$ which corresponds to the magnitudes of the set of lines at all values of ρ which have an orientation of α. We then convolve $\Gamma(i)$ with a discretized version of the derivative of the Gaussian whose variance σ_ρ is defined as the 'scale' of the extraction. This convolution acts as an edge detector and we set the values of ρ_a and ρ_b to the dominant maximum and minimum of the convolved signal at the given scale.

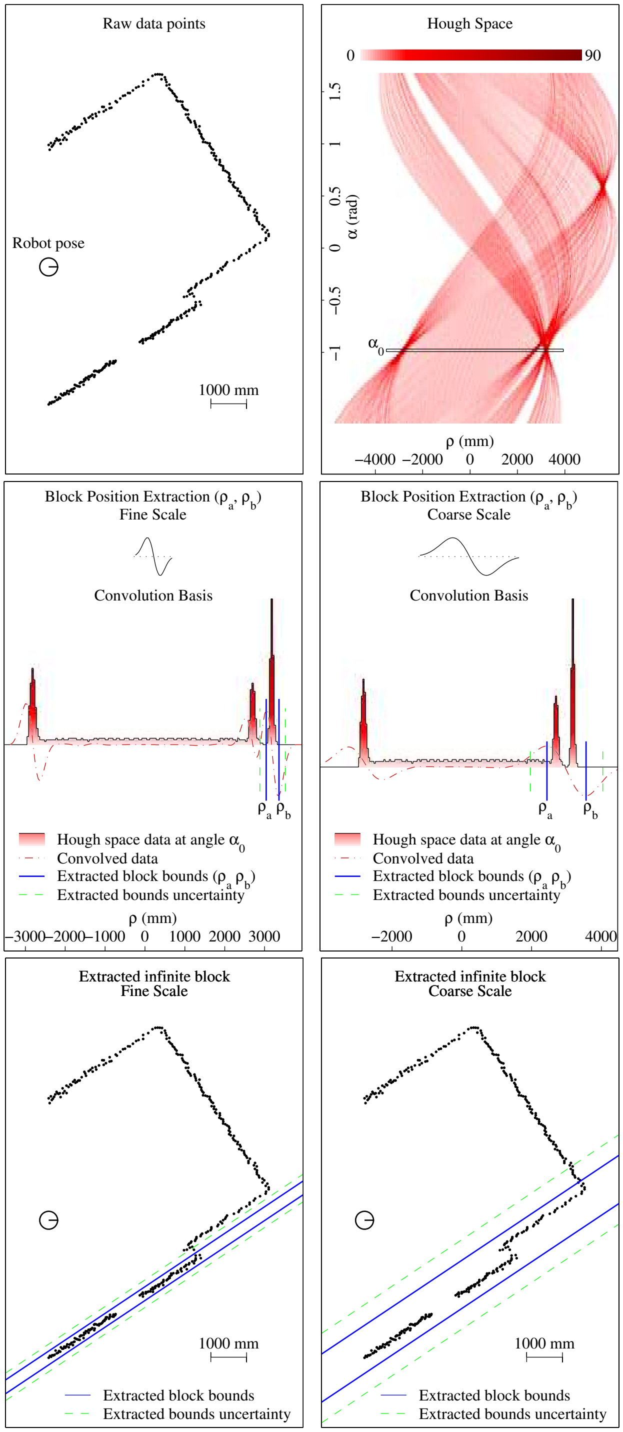

Fig. 4. Multi-scale extraction of ρ_a,ρ_b - (a) Raw scan points (b) Hough transform (c,d) Block ρ boundary detection at different scales (e,f) Detected block at fine and coarse scales.

Fig. 4(b) shows a Hough transform for the set of points in Fig. 4(a). The black line in Fig 4(b) passes through the cells corresponding to lines at angle α. Fig.s 4(c,d) show this slice of the Hough space as well as the convolution of this discrete signal with a derivative of Gaussian basis at multiple scales. The values for ρ_a and ρ_b are detected as the maximum and minimum of the convolved signal. The resulting blocks at different scales are shown in fig.s 4(e,f) (with arbitrary end points chosen for the segment, as they are computed in a subsequent step).

We estimate the variance of the terms $P_{\rho_a\rho_a}$ and $P_{\rho_b\rho_b}$ from Eq. (4) as follows:

$$P_{\rho_a\rho_a} = P_{\rho_b\rho_b} = (\sigma_\rho)^2 + P^\rho_{noise} \qquad (7)$$

where

$$P^\rho_{noise} = \left(\sum_{k=1}^{n} \frac{1}{P_v^k} \right)^{-1} \qquad (8)$$

and P_v^k is the projection of the modeled range sensor measurement noise for point v_k onto the ρ axis and n is the total number of points in the block. Therefore the uncertainty in the ρ position of a block feature is a combination of the scale extraction uncertainty $(\sigma_\rho)^2$ and the sensor measurement uncertainty P_{noise}. At coarser scales the contribution from the extraction uncertainty dominates with $(\sigma_\rho)^2 >> P^\rho_{noise}$ while at very fine scales the uncertainty from sensor noise can be significant.

We define the uncertainty in the α measurement to be a similar combination of our discretization level D_α and process noise :

$$P_{\alpha\alpha} = (D_\alpha)^2 + P^{\alpha}_{noise} \tag{9}$$

where P^{α}_{noise} can be computed as shown in [27].

Note that the nature of the multi-scale Hough transform allows additional computational cost savings. The computational cost can be reduced at coarser scales by decreasing the resolution of the Hough space discretization (increasing the bin size) in the ρ dimension (by increasing D_ρ up to the scale of the derivative of Gaussian convolution). With a coarser scale in ρ we also gain the benefits of a similarly coarser scale in the α dimension due to Eq. 5. A similar method of improving efficiency of coarse feature extraction is utilized in the "adaptive Hough transform" [28]. Moreover, one can structure the Hough transform in a coarse-to-fine fashion to gain added efficiencies. The clustering of range points at a coarse scale significantly simplifies the Hough transform calculations at the next finer scale, as knowledge from the coarse scale reduces the searching, data grouping, and computational requirements at the finer scale.

B. Endpoint Detection

Endpoint detection is an analogous process but performed on the raw data points instead of in Hough space. We project all points contributing to the line $[\alpha, \rho_a, \rho_b]$ onto the underlying line. We then convolve this signal with the derivative of a Gaussian at a given scale σ_ψ and detect the maximum and minimum peaks. These peaks determine the endpoints of the feature ψ_a and ψ_b. Fig. 5 shows the endpoint extraction process for the fine scale infinite block shown in Fig. 4.

Like the ρ covariance terms, the variance terms $P_{\psi_a\psi_a}$ and $P_{\psi_a\psi_b}$ in P_B are set to be equal to the sum of the variance of the given Gaussian basis and the process noise:

$$P_{\psi_a\psi_a} = (\sigma_\psi)^2 + P^{\psi_a}_{noise} \tag{10}$$

$$P_{\psi_b\psi_b} = (\sigma_\psi)^2 + P^{\psi_b}_{noise} \tag{11}$$

where the P^{ψ}_{noise} terms are calculated by projecting the measurement noise uncertainty of the distal points into the ψ axis. The cross terms involving the variable α in the covariance matrix P_B are computed assuming the center of rotational uncertainty of the block is anchored at the center of the block. The remaining cross-coupling terms in the covariance matrix P_B we set to zero when the feature is extracted as we assume that the noise contributions from the range sensor independently effects the variables ρ_a, ρ_b, ψ_a, and ψ_b.

C. Multiple Feature Detection

As each feature is detected, the points contributing to that feature which lie inside the bounds of the block are removed from the candidate point set and the algorithm is repeated, detecting features from the set of unselected points until no points remain. The result is a set of m blocks and covariances $\{B^k, P^k_B\}_{\sigma_\rho}, k = 1...m$ extracted at scale σ_ρ with a corresponding set of point groups $\{V_k\}, k = 1...m$. Each point group V_k is the set of range data points associated with the k^{th} block feature. By design, the point groups are disjoint, so that different features at the same scale can not share underlying points. Fig. 6 shows the Hough space and extracted block for the subsequent feature extracted from the data in Fig.s 4 and 5.

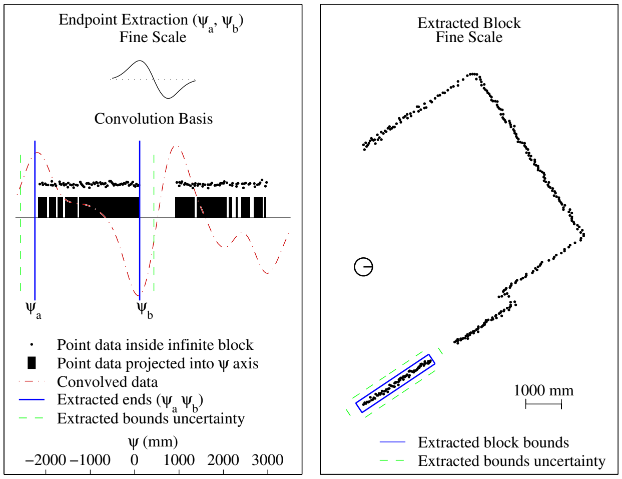

Fig. 5. End extraction at the fine scale

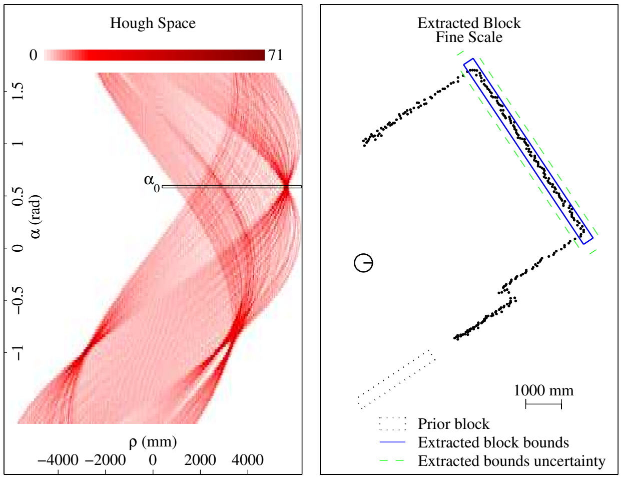

Fig. 6. Subsequent block extraction.

IV. CONSTRUCTION OF SCALE TREES

We define a scale tree as a tree of block features extracted from a common data set at multiple scales. Parent-child connections on the tree are established for features at different scales wherein the child feature at a finer scale has been extracted from a subset of the data encompassed by the parent. There are two basic methods of building a scale tree. The first is bottom-up construction started by extracting features at the finest scales and combining them into coarser scale features while climbing the tree. The second is top-down construction, started by extracting the coarsest features and then extracting finer and finer scale sub-features from the coarser features. In this work we focus on the top-down construction method, the results of which can be seen in Fig. 1(a) and Fig.s 7(a,c).

V. Multi-scale Feature Comparison

To solve data correspondence problems that arise during robot localization, we must pair up features extracted from data taken at different robot poses. We use a χ^2-test [29] to determine if two individual features are not the same :

$$\chi^2 = (B2 - B1)(P_{B2} + P_{B1})^{-1}(B2 - B1) \qquad (12)$$

where $B1$ and $B2$ are defined in Eq. (3) and P_{B1} and P_{B2} are defined in Eq. (4). By applying a threshold to the χ^2 distances between block pairs, we can eliminate candidate pairs of features that have a very low chance of being a proper match. The thresholds follow the classical χ^2 theory [29] so a threshold value of 15.1 corresponds to the χ^2 distance above which we have 99% confidence that the five degree of freedom block pair doesn't match. Note that this test can be applied in a multi-scale fashion. Using a sequential ratio probability test [30], one can establish two thresholds which guide the matching sequence to accept or reject a match at the current scale, or request processing at the next finer scale.

Also, given that we assume the variables ρ_a, ρ_b, ψ_a, and ψ_b are statistically independent in our feature extraction method, we can define a piece-wise χ^2-test that assesses the χ^2 distributions of these variables separately, thereby allowing for partial matches of block features. In our examples we use this method and assume two features are a positive match if three out of the five block parameters pass the χ^2-test. This allows for improved robustness to changes in feature geometry stemming from occlusion or different field of view across compared poses. When using these partially matched features to localize, we only use the subset of matched parameters.

VI. Experiments

Here we provide a few simple examples to show the effectiveness of the multi-scale data representation approach for basic tasks that contribute to localization and mapping. The first example explores the efficiency of finding data correspondences using single scale and multi-scale methods. We find a 5-fold improvement in computational speed with the multi-scale approach, and suggest why such efficiency is to be expected. The second example, which is based on the first example, considers multi-scale scan matching for displacement estimation. We show that the multi-scale approach leads to improved robustness with respect to perturbed initial conditions.

A. A Correspondence Example.

Fig. 7 presents an example using data collected from a Sick LMS-200 range scanner in an indoor office environment. All computations and timing estimates are done using the Matlab programming environment running on a Athlon 2.0Ghz CPU. The first set of data, termed "scan 1," is actually the same data found in Fig. 1(b). The second set of data, termed "scan 2," was taken at a nearby robot pose. Fig.s 7(a,b) show the scale tree and the finest features extracted from scan 1 and fig.s 7(c,d) show the scale tree and finest features extracted from scan 2. For this example we assume a known and somewhat accurate estimate of the displacement between the two poses (such as might be provided by odometry). We wish to determine the correspondences between the features at the finest scale of each pose. There are 52 fine features detected in scan 1 and 53 fine features detected in scan 2. The total number of feature-to-feature comparisons carried out at the finest scale by an exhaustive search is 2756, which takes 0.55 seconds of processing time. The multi-scale version proceeds by using the χ^2-test at every scale to check for matches between features. When feature matches are found (i.e., the feature pairs pass the χ^2-test), then the children nodes of the matching features

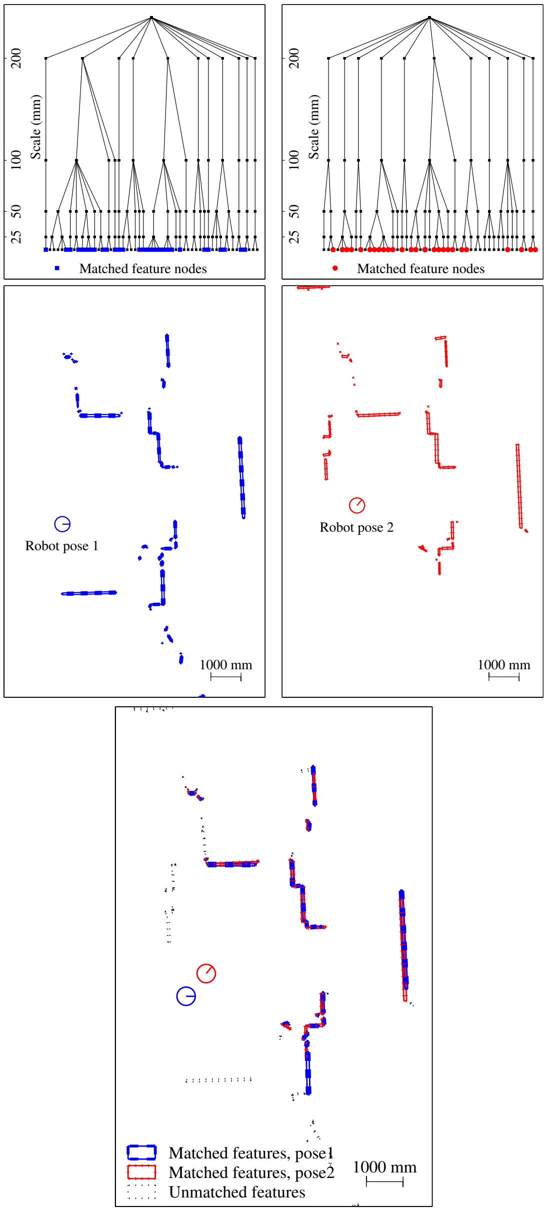

Fig. 7. Multi-scale range scan representation: (a) Scale tree for pose 1 (b) Finest scale features pose 1 (b) Scale tree for pose 2 (d) Finest scale features for pose 2 (e) Corresponding features from pose 1, pose 2.

are compared with the χ^2-test. This approach significantly reduces the search space. When comparing the same data set and taking advantage of the scale tree structure, the same set of correspondences is extracted using only 470 comparisons in 0.18 seconds. We repeated this example for 100 pairs of unique scans in similar environments. The results showed an average time for the exhaustive search of 0.329 seconds for 1708 comparisons, and an average computational time for the multi-scale matching of 0.086 seconds for 274 comparisons. These results show a nearly 4-fold decrease in computation time and more than a 6-fold decrease in computational complexity using our multi-scale approach.

While the relative improvement of our method with respect to single scale correspondence will depend upon the data set, the efficiency is inherent in our method, as shown in the following simplified analysis. Assume that we wish to determine the correspondence between N features in two different range scans. For the sake of simplicity, assume that $N = 2^m$ for some integer m. The process of finding correspondences between the N features in two scans has complexity βN^2, where β is a scaling coefficient that depends upon the details of the correspondence method. Now consider the computation involved in finding correspondences with the scale trees. Assume for the sake of a simplistic argument that the scale tree is dyadic–each node has two children nodes. At the coarsest scale, we only search for correspondences between two pairs of features, which results in a computational cost of $\beta 2^2$. At the next finer scale, there are 4 features to check for possible matches. However, we need only check for correspondences among the features of the children that descend from the parent nodes that were found to be in correspondence at the coarser scale. This results in a computational cost of $2\beta 2^2$. Continuing in this fashion for $log(m-1)$ levels in the tree, we find that the multi-scale version of correspondence requires computational effort of $\beta(N-1)2^2$. Thus, for the specific case of a dyadic tree, our method should scale linearly with the number of features, as opposed to quadratically for a fine scale analysis. While not all scale trees will be dyadic, clearly there are substantial computational savings to be had with this approach.

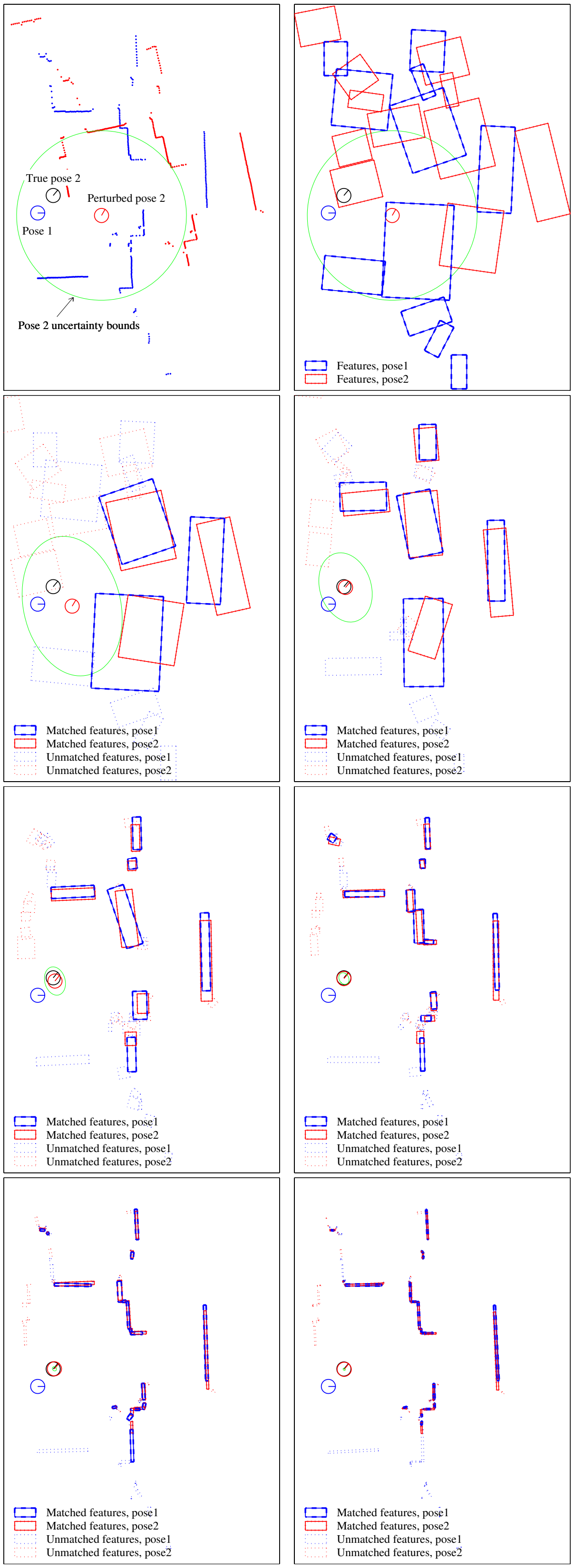

Fig. 8. Multi-scale localization example where the blue circle is pose 1, the red circle is the estimated pose 2, and the black circle is the actual pose 2 (a) Initial pose estimates and raw scans (b) coarse feature fit (c-g) Intermediate pose estimates and feature correspondences at each scale.

B. A Localization Example.

This example focuses on the most basic process of registering two scans, which can be used in a scan-matching odometry process, or as part of the solution to the kidnapped robot problem. We consider two scans taken at different poses (the same scans as in the last example), and we seek to estimate the relative displacement between these poses. Our method starts at the coarsest scale and extracts block features from each scan and compute feature correspondences given an initial (but not necessarily accurate) displacement estimate (e.g., from odometry). We use these initial correspondences to correct our displacement estimate, and then apply this updated displacement estimate when extracting features and computing correspondences at the next finer scale. We continue this method down the scale tree until we reach the finest scale features and compute our most accurate displacement estimate. While we use the same pair of range scans as seen in Figure 1, we have purposely introduced a significant error in the initial estimate of displacement between the poses of 2 meters and 10 degrees. In this way, we can test the robustness of the matching process.

Figure 8(a) shows the two scans overlayed with this initial displacement error and the initial corresponding coarse scale features. The green ellipses in these plots represents the 3σ bounds of uncertainty of the poses. Figure 8(b) shows the poses after initial correction from the correspondences at the coarsest scale. In the subsequent Fig.s 8(c-g) we show the corresponding feature sets at increasingly fine scales as our displacement estimate improves. 8(h) shows the final point overlay with the corrected displacement estimate. Note that using a conventional single scale correspondence and displacement estimation algorithms [7], [9], we were unable to establish correspondences of the fine scale features. Thus, this example shows that the multi-scale approach can significantly improve robustness to initial displacement errors while maintaining accurate displacement estimates.

VII. Conclusion

We presented a novel multi-scale scheme to represent point and line data in planar range scans. Additionally, by way of examples, we introduced basic tools to use this scheme to improve the efficiency and/or robustness of some basic operations that arise frequently in localization and mapping problems. While we focused on the correspondence and scan matching problems in this paper, the same methods should also provide significant benefits to the kidnapped robot problem as well. We showed that our multi-scale representation, and its integration into new correspondence and displacement estimation schemes, can lead to significant improvements in efficiency in some localization and mapping operations, and improved robustness in others.

Acknowledgments. This research has was sponsored in part by a National Science Foundation Engineering Research Center grant (NSF9402726) and NSF ERC-CREST partnership award EEC-9730980.

References

[1] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[2] S. Atiya and G. Hager, "Real-time vision-based robot localization," *IEEE Trans. on Robotics and Automation*, vol. 9, pp. 785–800, Dec. 1993.

[3] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometry beacons," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[4] J. Neira, J. Tardós, J. Horn, and G. Schmidt, "Fusing Range and Intensity Images for Mobile Robot Localization," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 1, pp. 76–84, Feb. 1999.

[5] S. Roumeliotis and G. Bekey, "Bayesian estimation and Kalman filtering: A unified framework for Mobile Robot Localization," in *Proc. IEEE Int. Conf. on Robotics and Automation*, San Fransisco, CA, Apr. 24-28 2000, pp. 2985–2992.

[6] S. Thrun, D. Fox, and W. Burgard, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Machine Learning*, vol. 31, pp. 29–53, 1998.

[7] F. Lu and E. Milios, "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," *J. of Intelligent and Robotic Systems*, vol. 20, pp. 249–275, 1997.

[8] S. Engelson, "Passive map learning and visual place recognition," Ph.D. dissertation, Dept. of Computer Science, Yale University, New Haven, CT, 1994.

[9] S. Pfister, K. Kriechbaum, S. Roumeliotis, and J. Burdick, "Weighted range sensor matching algorithms for mobile robot displacement estimation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, D.C., May 2002.

[10] S. Pfister and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, Sept. 2003.

[11] D. Mount, N. Netanyahu, K. Romanic, and R. Silverman, "A practical approximation algorithm for lms line estimator," in *Proc. Symp. on Discrete Algorithms*, 1997, pp. 473–482.

[12] G. Borges and M. Aldon, "A split-and-merge segmentation algorithm for line extraction in 2-d range images," in *Proc.* 15^{th} *Int. Conf. on Pattern Recognition*, Barcelona, Sept. 2000.

[13] S. Thurn, Y. Liu, D. Koller, Z. Ghahramai, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robotics Research*, pp. 693–716, Jul-Aug 2004.

[14] M. Walter, R. Eustice, and J. Leonard, "A provably consistent method for imposing sparsity in feature-based slam information filters," in *Proc. Int. Symp. Robotics Research*, 2004.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[16] P. Newman, M. Bosse, and J. Leonard, "Autonomous feature-based exploration," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.

[17] R. Madhavan, H. Durrant-Whyte, and G. Dissanayake, "Natural landmark-based autonomous navigation using curvature scale space," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2002, pp. 3936–3941.

[18] D. Pai and L.-M. Reissell, "Multiresolution rough terrain motion planning," *IEEE Trans. on Robotics and Automation*, pp. 19–33, Feb. 1998.

[19] G. Theocharous, K. Murphy, and L. Kaebling, "Representing hierarchical pomdps as dbns for multi-scale robot localization," *Proc. IEEE Int. Conf.. on Robotics and Automation*, May 2004.

[20] M. Montemerlo and S. Thrun, "A multi-resolution pyramid for outdoor robot terrain perception," in *Proc. AAAI Nat. Conf. on Artificial Intelligence*, San Jose, CA, 2004.

[21] J. Castellanos and J.D.Tardos, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, 1999.

[22] J. Gomes-Mota and M. Ribeiro, "Localisation of a mobile robot using a laser scanner on reconstructed 3d models," in *Proc.* 3^{rd} *Portuguese Conf. on Automatic Control*, Coimbra, Portugal, Sept. 1998, pp. 667–672.

[23] E. Magli and G. Olmo, "On high resolution positioning of straight patterns via multi-scale matched filtering of the hough transform," *Pattern Recognition Letters*, vol. 22, no. 6/7, pp. 705–713, 2001.

[24] ——, "Integrated compression and linear feature detection in the wavelet domain." in *Proc. Int. Conf. Image Processing*, Vancouver, Canada, Sept. 2000.

[25] G. Olmo and E. Magli, "Determination of a segment endpoints by means of the radon transform," in *Proc.* 6^{th} *Int. Conf. on Electronics, Circuits, and Systems*, Cyprus, Sept. 1999.

[26] J. I. J. Princen, H.K. Yuen and J. Kittler, "A comparison of hough transform methods," in *Proc. IEEE 3rd Int. Conf. on Image Processing and its Applications*, July 1989, pp. 73–77.

[27] S. Pfister, "Weighted line fitting and merging," California Institute of Technology, Tech. Rep., 2002, available at:http://robotics.caltech.edu/~ sam/TechReports/LineFit/linefit.pdf.

[28] J. Illingworth and J. Kittler, "The adaptive hough transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 690–697, 1987.

[29] P. Greenwood and M. Nukulin, *A guide to chi-squared testing*. New York: John Wiley and Sons, 1996.

[30] A. Wald, *Sequential Analysis*. New York: John Wiley and Sons, 1947.