Incremental Mapping of Large Cyclic Environments

Jens-Steffen Gutmann[1]   Kurt Konolige[2]

[1]Institut für Informatik
Universität Freiburg
D-79110 Freiburg, Germany

[2]SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Abstract

Mobile robots can use geometric or topological maps of their environment to navigate reliably. Automatic creation of such maps is still an unrealized goal, especially in environments that have large cyclical structures. Drawing on recent techniques of global registration and correlation, we present a method, called Local Registration and Global Correlation (LRGC), for reliable reconstruction of consistent global maps from dense range data. The method is attractive because it is incremental, producing an updated map with every new sensor input; and runs in constant time independent of the size of the map (except when closing large cycles). A real-time implementation and results are presented for several indoor environments.

1. Introduction

In this paper we address one of the major goals of mobile robot research, the creation of a map from local sensor data collected as the robot moves around an unknown environment. Our emphasis is on reliable, efficient techniques that can be used on-line to create maps as the robot moves in a new indoor environment.

There are many different types of maps used for localization, based on the form of sensor information and the representation requirements of localization. Maps can be based on topological (connection) or metric (distance) information, or a combination of the two. Metric maps can be further refined by whether they use features (usually called *landmarks*) or rely on dense surface information that does not distinguish features. Landmark methods require either engineering the environment to provide a set of adequate landmarks, or efficient recognition of features to use as landmarks; the result is only a sparse representation of the surfaces in the environment. In contrast, dense sensor methods [1, 9, 14, 16, 21] attempt to use whatever sensor information is available to create a map, and they recreate a geometric representation of the surfaces in the environment. When localizing the robot in a map, dense sensor matching can take advantage of whatever surface features are present, without having to explicitly decide what constitutes a landmark. In this research, we concentrate on metrically precise maps that are derived from dense range readings, such as those provided by sonar arrays, scanning laser range finders, or stereo vision systems.

1.1. Map Building in Cyclic Environments

We abstract the problem by considering a robot that moves through a set of discrete poses (position and orientation) and takes a scan (set of range points) at each pose. If we assume that the map is known, then we can estimate the poses of the robot using standard techniques from Kalman filtering, in which the pose p_n of the robot at time n is computed in two steps. First, the new pose is predicted from the pose p_{n-1} at time $n-1$ using the robot motion. Then, the sensor scan s_n at time n is matched against the map to give another estimate of the robot pose, and the two are fused to give the updated pose p_n.

The situation is more complicated when the map must be estimated at the same time as the robot pose. In this case, the entire map has to be treated as a complex random variable that is estimated concurrently with the robot pose p. For example, suppose the map consists of a large set of features M. At any point, the robot scan s_n may relate to only a small subset $a_n \subset M$. But the problem cannot be reduced to estimating p and a_n alone, since previous scans may have linked a_n to other subsets of M. The basic Kalman assumption, that all necessary information about a sequence of observations can be reduced to incremental changes in local state, does not hold. Many complicated schemes have propagated estimates of local subsets to the full set M, but none of them have been particularly successful ([4, 7, 11, 15, 17]; see Lu & Milios [13] for a review).

1.2. Scan Matching and Consistent Pose Estimation

Lu and Milios take a different view. Recognizing the Kalman assumption does not hold, they consider the full pose set p, and try to globally optimize p based on how well neighboring sensor scans match. There is no explicit estimation of the map M; instead, the scans themselves are an implicit representation of the map surfaces. The whole process is called *consistent pose estimation*, because it finds a set of poses that minimizes the total error of the system. Error terms come from robot motion, and also overlapping scans: the better the scans match, the lower the error. Figure 1 shows the basic idea: the initial set of poses based on robot motion has a high global error from non-aligned scans, while the optimal solution has very little residual error in the scan match.

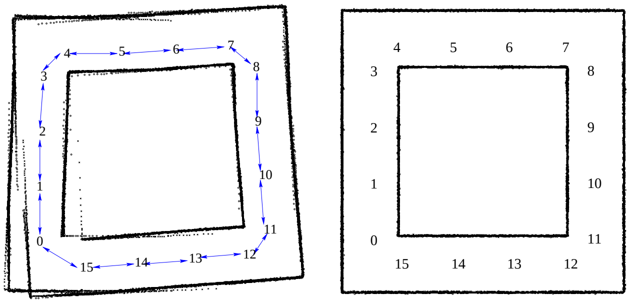

Fig. 1. Consistent pose estimation: Set of robot poses based on odometry with constraints between poses (left) and map obtained from consistent pose estimation (right).

Because the poses p are considered as a whole, consistent pose estimation can work on cyclic environments (as in Figure 1), in which the robot returns to positions previously visited. Unfortunately, since the basic operation of error estimation, scan-matching, is nonlinear, finding a minimal-error pose estimate is computationally difficult, and Lu and Milios use a hill-climbing approximation that is very sensitive to the initial estimate of p. The initial estimate is derived from dead-reckoning, and if it is not close to the true solution, the method often converges to a local minimum that is incorrect. This problem is especially critical in large cyclic environments, since the dead-reckoning errors when "closing the loop" are often significant enough so that scans which should be close do not overlap. In short, the method of Lu and Milios requires maps that are *topologically correct*, that is, every pair of poses whose scans overlap are correctly identified as such.

Another problem with the Lu and Milios method is that, as the number of poses n increases, the amount of computation grows as $O(n^3)$. It would be good to have an incremental method, one that computes "anytime" results as information comes in, so that the map is always the best that can be estimated. Although Lu and Milios provide an incremental algorithm, it essentially runs the original algorithm on all poses accumulated to the current point, without any computational savings.

1.3. Other Approaches: Expectation Maximization

Thrun *et al.* [18] developed an EM approach for building topologically correct maps. Their method uses landmarks in the environment (currently observed by button presses from an operator) for determining topological relations. However, no information from proximity sensors is used for estimating the robot poses and therefore the resulting maps have only low accuracy.

A hybrid method was proposed by Thrun *et al.* [19]. It first runs the EM method for finding out the topological relationships and then uses consistent pose estimation for fine-tuning the map. However, this method still requires user input, has a high computational demand and cannot be used in an incremental way. A recently reported variant of the EM approach uses sonar data instead of button presses, but is limited in the size of the environment and the accuracy of the resulting map [2].

1.4. Local Registration and Global Correlation

In this paper we propose a new method for consistent global pose estimation, based on the Lu and Milios ideas. We introduce two techniques for solving the problems noted above, namely, efficiently adding new information to the current map, and determining topologically correct relations between the poses, especially after long cycles.

Consider the case where a consistent global map has been built, and a new pose p_n is added (Figure 2). This new pose will have a link to the previous pose based on odometry, and to several of the previous poses based on scan overlaps and the resultant scan matches. These relationships are shown in Figure 2 by the bold arcs between poses. As long as the robot is forging ahead and exploring new areas, these arcs all cluster into local neighborhoods that are well-connected, with no long-distance relationships.

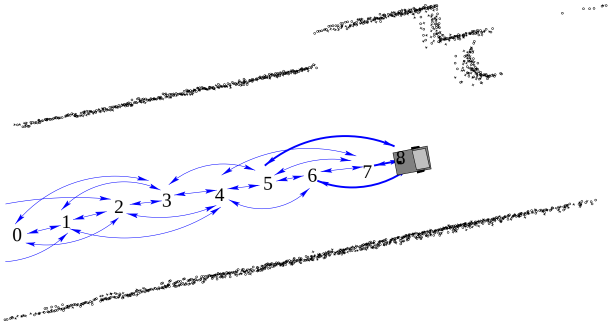

Fig. 2. Effect of adding a new pose of an unexplored environment to the map.

Local registration works well if the robot constantly explores new areas. As it completes a large cycle, however, the problem of topological correctness becomes important, because new poses must be related to old ones. At this point, it is critical to make topological identifications reliably, because a mistake can cause the map to be badly misaligned. Single scans do not generally have enough information to yield good false positive rejection, especially if the environment is relatively uniform in one direction, as happens often along corridors. Instead, we integrate a set of local scans into a *map patch*, and use this more extended template to find matches in the old map.

The map patch technique is obviously more reliable than single scans in rejecting false positives, but it leaves open the question of how to efficiently perform matching, since single-scan techniques are no longer applicable. Fortunately, one author has recently investigated correlational concepts for matching map patches in the context of localization [12]. The resultant techniques have been shown to be both efficient and reliable, and we make use of them here to determine topological correctness in map-building with cycles. Correlation operates in a “background” mode, checking for matches against the old map whenever the robot moves to an appropriate location, and adds in links between the new map and the old whenever appropriate.

In the rest of this paper we present the LRGC algorithm, showing how the basic techniques of scan matching, registration, and correlation can be knit together into a practical, real-time system for map construction using dense range information.

2. The LRGC Algorithm

The Local Registration / Global Correlation algorithm relies on the three techniques of scan matching, consistent pose estimation, and map correlation. In this section we present the techniques in detail, showing the modifications necessary to work under LRGC. Finally, the overall algorithm is explained, and its results under real-world conditions are displayed in the following section.

2.1. Scan Matching

Scan matching is the process of translating and rotating a range scan (obtained from a range measuring device such as a laser range finder) such that a maximal overlap with an *a priori* map, e.g. another scan, emerges. The matching algorithm returns a pose probability distribution of where to place the scan in order to have the range measurements correspond to map features. There might be more than one location where a scan fits and this is expressed by the probability distribution.

Usually the pose probability is restricted to Gaussian distributions, and the search is bound locally around an initial estimate of the scan pose which is obtained from odometry. Often closed-form solutions for determining the displacement exist and typical run-time complexities are $O(n^2)$ or less, where n is the number of range measurements in the scan [6, 14].

Many scan-matching methods have been proposed in the past. Cox [6] matches scan points to line segments in an *a priori* map. Weiss & von Puttkamer [20] create histograms for correlating pairs of scans. Castellanos *et al.* [3] and Gutmann *et al.* [10] extract line segments out of a scan and match them with an *a priori* map of line segments. Lu & Milios [14] do not rely on a polygonal structure of the environment and developed a universal method for matching a pair of scans. We use a combination of the Cox and Lu & Milios method (the method from Gutmann & Schlegel [9]), which benefits from the universal capabilities of the Lu & Milios method and the better run-time and pose quality of the Cox method. The method is highly efficient [10] and has proven to yield very accurate results [9, 8].

It is critical that scan-matching does not overestimate the certainty of a pose, or else it can be difficult to find a consistent interpretation of a set of overlapping poses (see [11] about this particular problem). Scan matching should also produce quantitatively good results, e.g. straight lines in a corridor environment should be accurately aligned.

2.2. Consistent Pose Estimation

The key concept for fusing scan match information is the consistent pose estimation method from Lu & Milios [14]. As noted above this method requires a good initial estimate of the scan poses in order to generate useful results. Therefore we use this method in two ways:

1. For creating local patches (local maps of the last few scans the robot obtained), since in this case the scan poses are always topologically correct: very little odometrical error has been accumulated. Even with larger odometry errors, as for example in sliding motion or with synchronization problems between sensor data and odometry, use of scan matching and local registration can often recover the correct geometry.

2. For closing a loop after topological relationships are obtained from map correlation. In this case, consistent pose estimation is first run with the new links added to the map and then, after closing the loop leading to a topological correct map, rerun with new scan matches between the newly linked poses for fine-tuning the map.

A typical local network topology was shown in Figure 2. When a new pose p_n is added to the chain, we register only the last K poses, a local neighborhood of the new scan. To test if the K-neighborhood update tracks global consistency, we performed a number of experiments to compare it to an update of all poses. Figure 3 shows the effect of varying K for a 150-pose map, where the distance between poses was about 0.3 meters. For $K \geq 7$, the average error per pose is smaller than a millimeter, and stays essentially unchanged as the neighborhood is expanded.[1]

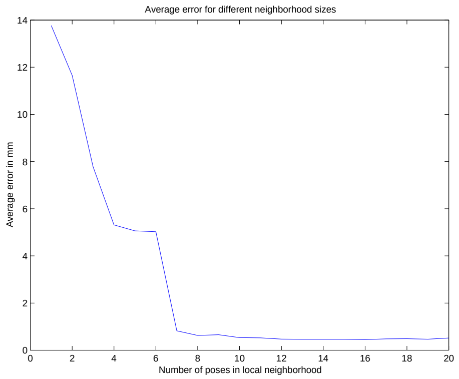

Fig. 3. Average pose error as a function of registration neighborhood size K.

Errors in local registration occur when there is not enough local context to bring borderline readings into alignment. Figure 4 shows the difference errors for $K = 5$ and $K = 10$ as a map is being built. Around pose 30, a difficult scan causes the error to shoot up; the larger neighborhood quickly recovers as it adds more context, but the smaller one never does.

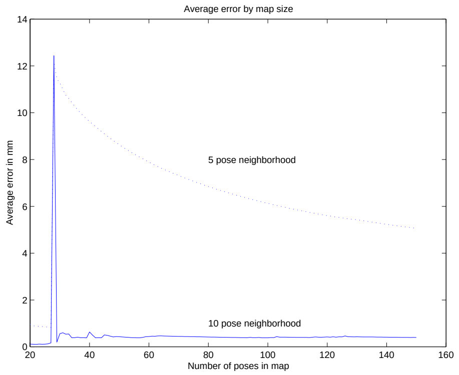

Fig. 4. Average pose error during construction of a 150-pose map.

Several properties of this incremental registration procedure should be noted. First, because the number of nodes considered at each step is limited, the computational burden is constant, that is, it does not depend on the size of the map. Given the efficiency of individual scan matches, incremental registration proceeds very quickly, e.g., under 100 ms in typical circumstances. A second point is that because scan-matching is a nonlinear procedure, local registration of a new scan will sometimes produce results that differ radically from complete global registration. In these cases, the new scan causes a small shift in some previous pose p_i outside the local registration area, and this shift in turn causes a completely different alignment of scan matching for p_i and one of its neighbors. However, one can argue that local registration does not necessarily produce a worse result, since the new alignment for global registration may be a local minimum, rather than a global one. We have not examined this issue very closely, however, and it deserves further attention.

We turn now to the problem of registration after closing a loop. Since consistent pose estimation is an $O(n^3)$ process in the number of poses, several optimizations are necessary to make the method efficient, since all the poses along the loop have to be taken into account. Usually these poses are only linked locally to neighboring poses which leads to a sparse matrix of relationships. Therefore efficient linear algorithms that make use of the sparseness are used for computing consistent poses.

Furthermore we follow the idea from Lu & Milios [14] to limit the size of the observation matrix to a maximum value by examining the network for strong links (pose constraints with low uncertainty). Strong links are then treated as constant relations between scan poses and the corresponding poses can be obtained from each other. Therefore one pose of each strong link can be removed which reduces the size of the observation matrix. Although this leads to a suboptimal solution, we didn't experience a gross difference to the optimal case when the maximum number of poses was set to a relative large number, e.g. 200 in our implementation. All of these optimizations mean that closing even large loops takes less then 10 seconds of computation in almost all cases.

2.3. Map Correlation

To determine the topological relationship between poses that close a cycle, we compare a recent portion of the map around the current pose with the older portions of the map. Where there is a good match, it is likely that the new pose is topologically connected to one of the older poses. In our current method of global registration, once a topological connection is made, it is not possible to undo it, since all poses are updated and no history is kept. Therefore, any such connection needs to be very certain before it is made, and false positive rejection is critical. This is the main reason for matching a patch that integrates several scans, and also for providing post-match filters to reject false positives.

Another constraint on map matching is that it must be

[1] Pose errors are calculated based on incremental differences between each two poses. For example, if local registration produced $\Delta x = 500\mathrm{mm}$ and $\Delta y = 50\mathrm{mm}$ between poses 1 and 2, and global registration was $\Delta x = 550\mathrm{mm}$ and $\Delta y = 50\mathrm{mm}$ for the same two poses, then the pose error would be $50\mathrm{mm}$.

efficient, since we intend to run it constantly in the background as the robot starts cycling back to places previously visited. Recent investigations by one of the authors has provided a fast and accurate matching technique based on correlation [12]. The justification for this technique lies in a Bayesian analysis of the match probability. For any given new map patch r and old map m, we seek the posterior probability $p(l|r,m)$ that the robot is at pose l. Using Bayes' rule, we have:

$$p(l|r,m) = k \cdot p(r|l,m)p(l,m).$$

The sensor response function $p(r|l,m)$ is the probability that we would see the map patch from the robot pose l, given the old map m. As shown in [12], the sensor response can be approximated by a correlation operator. A regular grid is imposed on the map area, and for each cell i we calculate the probability $p(r_i)$ of the map patch impinging on the cell and $p(m_i)$ of the old map impinging on the cell. The correlation operator is:

$$\sum_i p(r_i)p(m_i).$$

In practice it is convenient to put all the uncertainty into the map probability $p(m_i)$, simplifying the above sum and making it amenable to optimized implementation (see [12] for details). Figure 5 shows the correlation response for a typical map patch and old map. The dotted outline is the map patch, trailing behind current position of the robot. The oval shows where the robot could be, based on dead reckoning error. And the correlation response is dark where a good match has been found, and lighter where there is no match.

The prior probability $p(l,m)$ reduces to $p(l)$, since the old map is given. We use a uniform distribution over the matching area for $p(l)$, although we could also use a Gaussian based on the robot's covariance. The normalization factor k is more difficult to determine. In general, the probability $p(l|r,m)$ should sum to less than one over the match area, because there is a chance that there is no match in this area, that is, the robot's current map patch doesn't overlap with the old map. But it is very difficult to estimate how likely it is that the map patch doesn't have a match. In practice, we normalize the correlation response, and then use filters to reject false positives, i.e., to determine if there is no good match. We have found the following filters give a good result.

1. High match score. The unnormalized match score should be high.

2. Low ambiguity. We find clusters of high probability, and compare the peaks of these clusters. The ratio of the highest peak to the next highest should be large.

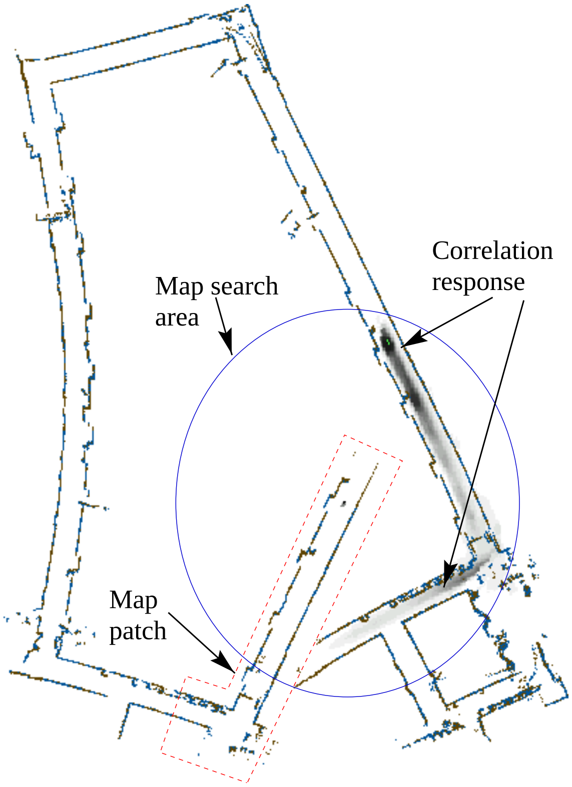

Fig. 5. Map correlation. The correlation response $p(r|l,m)$ in the search area (oval) is indicated.

3. Low variance. The best cluster should be sharply peaked.

For example, in Figure 5 there are several clusters. The one at the top has a very high match score, and it is about 5 times as high as the next best cluster. Finally, its variance is small, less then 20 cm in any direction.

Several factors can influence the quality of the matching procedure. The most important ones are the size of the map patch, and the area over which the patch is to be matched. We discuss how to choose values for these parameters in the next section.

2.4. Incremental, Consistent Mapping

Figure 6 shows the basic scheme that is used for updating the map when a new scan has arrived from the laser range finder. A map is represented as an undirected graph: nodes are robot poses with associated scans and links are constraints between poses obtained from dead-reckoning, scan-matching, or correlation. An empty graph is used as initial map.

When a new scan is added to the map, it is first registered with the last K scans (the local neighborhood) for proper

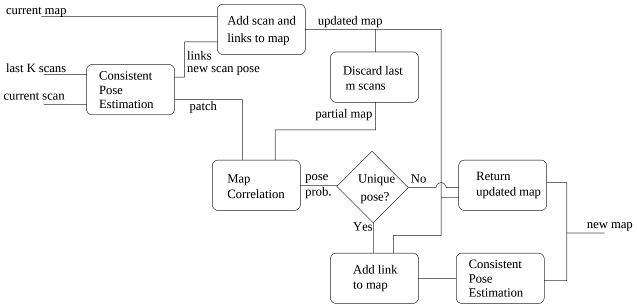

Fig. 6. Data flow of the incremental mapping method.

alignment and improving the position estimate from odometry. The new scan pose together with its links are then added to the current map leading to an updated map.

Loop detection is activated with each new scan. From the updated map, an "old" map is extracted that is assumed to be topological correct. This is done by discarding the last m scans (with $m > K$) to avoid having newer scans in the old map. A local patch is also created from the newest scans (see below for the size of this patch). The patch is correlated with the old map, and the resulting pose probability grid is examined according to the filters described in the last section. If the highest peak passes the filters, then we assume that a topological relation has been found. In this case the relation is added to the map and consistent pose estimation is run for closing the loop and adjusting the map.

For finding topological relations, the search space is restricted to an area around the current robot position. This area grows with the position uncertainty of the robot. We modeled the position uncertainty with a Gaussian distribution and only test topological relations for poses that have a Mahalanobis distance to the robot pose smaller than a given threshold. Also, to compensate for possible ambiguity in larger search spaces, the patch size grows linearly with the position uncertainty, and therefore large cycles are only closed if there is a good evidence for a topological relationship. After a cycle has been closed, position uncertainty decreases and search space and patch size fall back to small values automatically.

At the end of a mapping run, after integrating all scans, the resulting map can be further optimized by running consistent pose estimation over all scan poses.

3. Results

The mapping algorithm outlined in the last section has been tested in various environments with different robot systems. All of the results obtained in this section come from the mapping algorithm running autonomously on real-world data.

Figure 7(a) shows raw data collected by a *B21* robot equipped with a 180° *SICK* laser range finder that has been run in Wean hall at Carnegie Mellon University.[2] The environment is of size 80 by 25 meters and contains two cycles. One of the cycles has a length of approximately 200 meters.

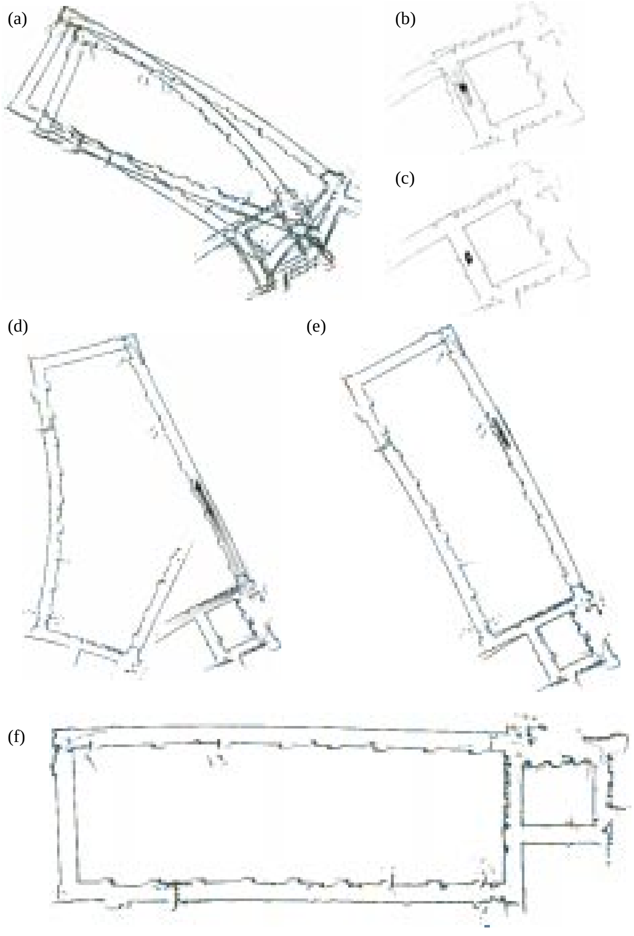

Fig. 7. (a) Map obtained in an environment of size 80 by 25 meters from sensor data using raw odometric data. (b) Before closing small cycle. (c) After closing small cycle. (d) Before closing large cycle. (e) After closing large cycle. (f) Final map.

Figure 7(b) shows the situation before closing the small cycle. Since the accumulated odometric error is small, closing the loop is a fairly easy job (Figure 7(c)). However, after the robot completes the large cycle, a significant amount of odometry error has been accumulated, as shown in Figure 7(d). For closing this loop a large patch size and search space are required, but the system is still able to align the scans (Figure 7(e)). Please note the small inaccuracies in

[2] The authors would like to thank Sebastian Thrun from Carnegie Mellon University for providing us with the raw data.

the lower left corner. Here the system doesn' t know yet that these parts belong to each other. After registering the large cycle a second time the system recognizes these relationships and the inaccuracies disappear. Figure 7(f) shows the final map.

Another series of experiments have been carried out at the Artificial Intelligence Center at SRI using a *Pioneer II* robot and *SICK* laser range finder. Figure 8(a) shows raw data of this experiment. The data was corrupted by a large drift noise in odometry coming from a carpet with a directional nap. The mapping algorithm is still capable of correcting for this drift error and building a topological correct and highly accurate map (Figure 8(b)).

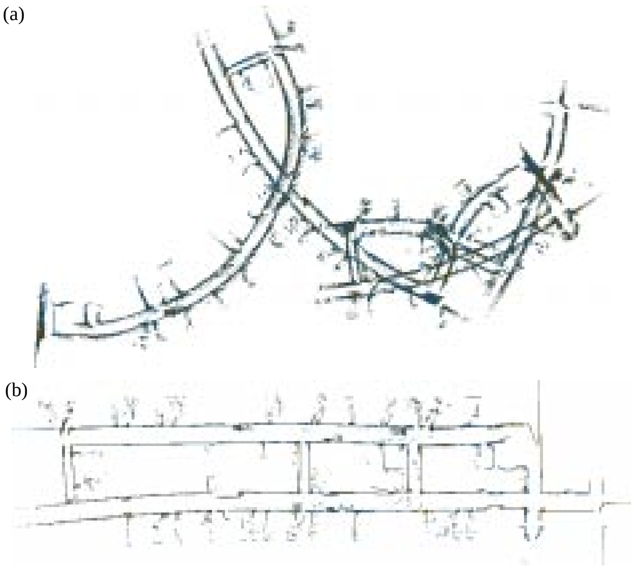

Fig. 8. (a) Raw data of an environment of size 85 by 15 meters with large drift noise in odometry. (b) Resulting map.

Figure 9 shows further mapping runs carried out at the Carnegie Museum of Natural Science using a *B21* robot[2] and at the Artificial Intelligence Lab of the University of Freiburg using a *Pioneer I* mobile robot equipped with a *SICK* laser range finder.

4. Conclusion

In this paper we presented the Local Registration / Global Correlation (LRGC) method for building maps in large, cyclic environments. The method makes use of three different techniques for incrementally building maps, finding topological relations and closing loops. Several sample mapping runs in different environments using different robot systems have been presented that demonstrate the capabilities and accuracy of this approach. The method is incremental: scans are integrated into the map as they are taken by the laser range finder, leading to a real-time mapping system. Only when closing loops additional computational power is needed. As far the authors know, this is the first real-time autonomous mapping system that produces dense, accurate metric maps in large, cyclic environments.

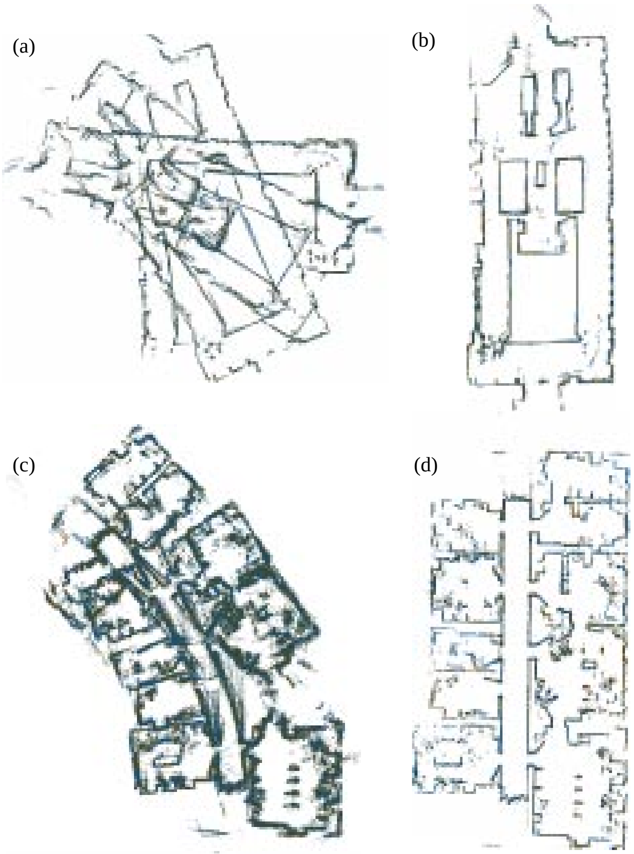

Fig. 9. Mapping runs carried out in other environments. (a) Raw data of the Carnegie Museum of Natural Science (45 by 15 meters). (b) Obtained map. (c) Raw data of the Artificial Intelligence Lab, Freiburg, Germany. (24 by 13 meters). (d) Obtained map.

Several crucial requirements of this approach should be noted. First the algorithm requires good results from scan-matching to build accurate local patches. For example, if a patch of a corridor environment contains curved walls, map correlation cannot produce sharply peaked pose grids since the patch doesn' t fit to the map very well. In these situations, it would be difficult to use other techniques such as Markov Localization [1, 12], since the robust false-positive rejection of larger patch matching is not available. We are considering using a series of smaller patches as a possible approach here.

Another requirement of the LRGC is efficient false-positive rejection, since topological relations obtained from map correlation must reflect only true relations. If the filters reject too many good matches, we won't get enough relations to close a loop; if they don't reject a false match, the map will be inconsistent. While the filters we propose have worked well, we are experimenting with a variety of other techniques, including visual references and 3D ranging. In the unavoidable case where a bad match is made, we can try to relax the strict identification of topological relations by keeping multiple hypotheses of where to close a loop and adding detection for wrongly closed loops.

Finally, the techniques we discussed are applicable in a fairly straightforward way to teams of robots mapping a given area. Each robot could build its own local map, and communicate with other robots via wireless links. Even when the robots initially have no idea of their relative position, by correlating patches from their own maps against other robots', they can merge the two, in a manner similar to closing a loop for a single robot.

References

[1] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the Fourteenth National Conference on Artificial Intelligence (AAAI'96)*, pages 896–901, 1996.

[2] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping of large-scale mobile robot environments using EM. In *Proc. of the International Conference on Machine Learning (ICML'99)*, 1999.

[3] J. A. Castellanos, J. D. Tardós, and J. Neira. Constraint-based mobile robot localization. In *International Workshop on Advanced Robotics and Intelligent Machines*. University of Salford, Apr. 1996.

[4] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. International Conference on Robotics and Automation*, pages 138–145, 1985.

[5] I. Cox and G. Wilfong, editors. *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[6] I. J. Cox. Blanche: Position estimation for an autonomous robot vehilce. In Cox and Wilfong [5], pages 221–228.

[7] H. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *Journal of Robotics Research*, 6(3):3–24, 1987.

[8] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, Oct. 1998.

[9] J.-S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT'96)*, pages 61–67, 1996.

[10] J.-S. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *Proc. International Conference on Intelligent Robots and Systems (IROS'99)*, Kyongju, Oct. 1999.

[11] P. Hébert, S. Betgé-Brezetz, and R. Chatila. Probabilistic map learning: Necessity and difficulties. In *Proc. International Workshop on Reasoning with Uncertainty in Robotics*, Amsterdam, 1995.

[12] K. Konolige. Markov localization using correlation. In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, 1999.

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

[14] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1997.

[15] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 85–94, 1989.

[16] A. C. Schultz and W. Adams. Continuous localization using evidence grids. Technical Report AIC-96-007, Naval Center for Applied Research in Artificial Intelligence, 1996.

[17] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In Cox and Wilfong [5], pages 167–193.

[18] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. Also appeared in Autonomous Robots 5:253-271.

[19] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proceedings of the 15th National Conference of the American Association for Artificial Intelligence (AAAI'98)*, July 1998.

[20] G. Weiss and E. Puttkamer. A map based on laserscans without geometric interpretation. In U. R. et al., editor, *Intelligent Autonomous Systems*, pages 403–407. IOS Press, 1995.

[21] Z. Zhang and O. Faugeras. Estimation of displacements from two 3d frames obtained from stereo. *Trans. Pattern Analysis and Machine Intelligence*, 14(2):1141–1156, 1992.