An *Atlas* Framework for Scalable Mapping

Michael Bosse Paul Newman John Leonard Seth Teller
Massachusetts Institute of Technology
{ifni,pnewman,jleonard,seth}@mit.edu

Abstract— **We would like to build a robot capable of mapping a region of unlimited size, and concurrently use the map as it is built. To do this we need to fully address the issue of large scale simultaneous mapping and navigation. For a practical implementation, the total amount of computation must be proportional to the total amount of sensor measurements and entirely independent of the size and complexity of the mapped region. This then implies constant time performance per iteration.**

This paper describes *Atlas* - a technique for managing a graph of multiple, limited size, local maps that offers constant time performance. It maintains a finite set of robot location hypotheses to determine which map if any best fits the perceived local environment.

We present results demonstrating this technique running real time in an indoor structured environment using both laser and ultrasonic ranging sensors.

I. Introduction

We would like to build a recursive algorithm facilitating real-time maneuvering and mapping in an unknown environment. To this end we seek to minimize the computation required per iteration and its dependency on the complexity and size of the environment. Hence we require a scalable method for egomotion and environment capture – an important goal in computer vision and robotics.

There is now a substantial corpus of knowledge in the fields of Simultaneous Localization and Mapping (SLAM) and Structure from Motion (SFM) for local, small-scale regions. [?], [?], [1], [?], [?] While such methods have been applied with success in small controlled environments, they struggle when scaling to larger environments. Many application domains require mapping of extended environments (1000's of square meters to 10 of square kilometers in area) with long sensor excursions and closing of large loops [?], [?], [?], [?].

Before proceeding, we make the following motivating observations:

To be submitted to ICRA2003, and IEEE TR&A.

- Models built of spatially distant areas are largely decoupled. [2]
- Methods whose complexity growth is unbounded are untenable. [3]
- Frequently applications only demand egomotion estimates with respect to local environment. [4], [5], [2]
- The error integrated around a long loop may be large enough to prevent the recognition of an already mapped region – loop closing is hard.[?], [?]
- We expect to encounter featureless regions in which navigation must rely on dead reckoning alone. [?]

The decoupling property has lead several researchers [6], [7], [8] to suggest partitioning the problem into sub-regions or sub-maps. Indeed the issue of complexity growth is also tackled by this approach. Given our desire to map a region of unlimited size, an un-managed growth of complexity with the size of the environment will prohibit a real-time implementation. However, it remains unclear how to manipulate a set of sub-maps in a consistent framework.

Given that many applications only require a representation of the local region [9], [10], [?], [?], we may derive our output from a single sub-map without recourse to an absolute global coordinate frame.

Since often not available, we assume no external infrastructure, and limit ourselves to purely proprioceptive and vehicle relative sensing. For example we permit accelerometers, gyros, odometry, radar/laser/sonar rangers, and cameras, but not GPS or a magnetic compass.

Although loop closing is difficult, it is a prerequisite for useful system. Without detection of loop closure, regions will be re-mapped multiple times. Similarly, it is important to be able close a loop after a transit through a feature-sparse region. This is straight forward given an absolute global measurement, like a GPS fix, since unlike with vehicle relative sensing, the global measurement is independent of accrued drift.

A. Contributions

A complete implementation of large scale mapping requires the integration of feature representation and detection, uncertainty modeling, data association and scaling techniques. This paper focuses on scaling. We describe a method in which existing small scale mapping algorithms can be applied to large scale problems via a new framework called *Atlas* .

The *Atlas* framework has the following novel aspects:

- We maintain no single global coordinate frame but rather an interconnected set of local coordinate frames. Each frame contains a map, which we refer to as a *map-frame*. We consider two coordinate frames to be connected (adjacent) if their map-frames possess shared structure; for example, common mapped features. This adjacency is represented by a coordinate transformation between frames.
- After recognizing the closure of a extended loop, we do not constrain the composition of adjacency transformations to compose to the identity transformation. This is essential to the constant time properties of the *Atlas* framework since no global updates are required. Nevertheless, this constraint can be applied off-line to refine the global arrangement of the multiple coordinate frames.
- The spatial extent of the map-frames is not predefined, but rather determined by an intrinsic metric of the contained map. The same metric is used to invoke either a transition to an adjacent frame or the genesis of a new one.
- The computational cost of the algorithm is constant for processing the data from each time step. A limit is placed on the per map computation by defining a measure of complexity for each map-frame which is not allowed to exceed a threshold (the map capacity). Rather than operate on a map of ever increasing complexity, the *Atlas* framework simply switches its focus to a new or adjacent map-frame. Additionally the computational cost of deciding whether to transition to an adjacent map-frame or spawn a new one is also bounded; thus the total computation is constant time.

B. Roadmap

In Section II we provide an overview of the *Atlas* framework which highlights its key components. Section III discusses these components in detail, illustrating them with the example of a feature-based implementation for which results using sonar and laser are given in Section IV.

II. *Atlas* Framework

In essence, the *Atlas* framework is just a graph of coordinate frames. Each vertex in the graph represents a local coordinate frame, and each edge represents the transformation between adjacent local coordinate frames.

In each local coordinate frame, we build a map which captures the local environment and the current robot pose along with their uncertainties. This we refer to as a map-frame.

As an example throughout this paper, we utilize a feature based representation of the environment. In this case, the estimate of the location of the robot and the features is a point in a high-dimensional parameter space manipulated by a Kalman Filter. (Ref. Smith-Self-Cheesman [?]) The local map, however, could also be represented with other models, such as a particle filter or an occupancy grid [?], [?].

We entertain the possibility that any one of several map-frames can best explain the recent observations of the robot. Hence we maintain a set of *map-frame hypotheses*, each independently processing the sensor measurements in turn. Figure 1 depicts a graph of map-frames in which two hypotheses are maintaining the robot position.

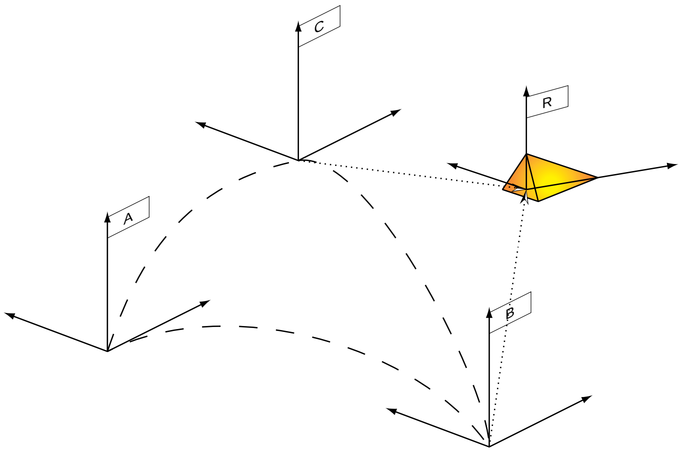


Fig. 1. A graph of map-frames in which two hypotheses (B and C) maintain the robot position (R).

The uncertainty of the edges (adjacency transformations) in the *Atlas* graph are represented by a Gaussian random variable. In contrast to Chong and Kleeman [?], the graph's edges are undirected. In other words, there is no preferred direction to an edge; the reverse direction of an edge contains the inverse transformation.

In contrast to Decoupled Stochastic Mapping [?], each map's uncertainties are modeled with respect to its own frame. In other words, the pdfs representing uncertainties are all conditioned on the local coordinate frame.

We can still, however, generate probabilities of entities with respect to arbitrary map-frames by following (transforming by) a path formed by the edges between adjacent map-frames. We compute such paths via Dijkstra's shortest path algorithm [?] using the uncertainties of the transformations as a statistical distance metric.

We close loops with *Map-matching*.

A. *Requirements*

Before beginning a detailed description of the components of the *Atlas* framework, the following two conditions are made clear.

Condition 1 The atlas framework needs a good local navigation method that can produce a consistent map.

Condition 2 The navigation uncertainty acquired around a loop before its closing must be small enough to avoid ambiguous map-matches.

The quality of the navigation and mapping produced by the *Atlas* framework can be no better than that provided by the local mapping scheme. As already stated, the *Atlas* framework simply extends existing techniques so that they can be applied to large scale regions. Condition 1 is a reflection of this point.

Condition 2 implies that ambiguous matches are tolerable as long as the predicted robot position is accurate enough to distinguish between multiple matches. This requirement limits the length of loops that the *Atlas* framework can correctly handle based on the balance between the local navigation uncertainty accumulation rate and the repetitiveness of the environment.

III. *Atlas* Components

This section provides a detailed description the six core concepts of the *Atlas* framework: uncertainty projection, competing hypotheses, creation of new map-frames (Genesis), closing loops (Map-Matching), instantiating and evaluating new hypotheses in adjacent map-frames (Traversal), and transformation edge refinement. These six components are now discussed.

A. *Uncertainty Projection*

The *Atlas* edges contain the information necessary to relate two map-frames. The uncertainty of the transformation edge is used when we project a stochastic entity (such as the robot position) from one map-frame into another. However, if the map-frames are not adjacent, then we must compose the transformations (and their uncertainties) along a path of edges that link the *Atlas* vertices. There may be more than one path from one vertex to another. Since we do not constrain these cycles, the paths will not generate the same composite transformation. In Figure 2(a), frame D is reachable from A via B or C, resulting in the two possible projections of frame D relative to A, shown in Figure 2(b).

To resolve this ambiguity we use Dijkstra's shortest path algorithm [?] to find a unique path between the nodes. Instead of using a physical distance to compute the length of the path, we use a statistical metric, ρ, based on the uncertainty of the transformation in *Atlas* edges. The metric we choose is the determinant of the covariance matrix of the composite transformation.

$$\begin{aligned} T_a^c &= T_a^b \oplus T_b^c && (1)\\ \Sigma_{ac} &= \mathrm{J}_1\left(T_a^b, T_b^c\right)\Sigma_{ab}\mathrm{J}_1\left(T_a^b, T_b^c\right)^T + \\ &\quad \mathrm{J}_2\left(T_a^b, T_b^c\right)\Sigma_{bc}\mathrm{J}_2\left(T_a^b, T_b^c\right)^T && (2)\\ \rho &= \det\left(\Sigma_{ac}\right) && (3) \end{aligned}$$

We define the Dijkstra Projection of an *Atlas* graph with respect to a given source vertex as the global arrangement of frames using compositions along Dijkstra shortest paths. This projection has the property of transforming the *Atlas* graph into a tree of transformations with the source map-frame as the root. We measure the nearness of any map-frame to the source frame as ρ computed from the compositions of transformations up the tree to the root.

B. *Competing Hypotheses*

At any given time, there are several competing map-frame hypotheses that attemt to explain the current robot pose and sensor observations. We verify the validity each map-frame hypothesis by monitoring a performance metric $q \in [0 \rightarrow 1]$ for a few time steps.

The metric q depends on the hypothesis's map-frame $\mathcal{M}_i$, robot pose $\mathbf{x}_i$, and recent sensor measurements Z.

$$\begin{aligned} q_i &= q(\mathcal{M}_i, Z, \mathbf{x}_i) \\ &\in [0 \rightarrow 1] && (4) \end{aligned}$$

For example, a suitable form of q for a feature-based approach is

$$\begin{aligned} q &= \alpha\left(1 - \frac{\det(\Sigma_{\mathbf{x}_i})}{\det(\Sigma_{\mathbf{x}_\star})}\right) \\ &\quad +(1-\alpha)\frac{\eta_a}{\|Z^k\|} && (5) \end{aligned}$$

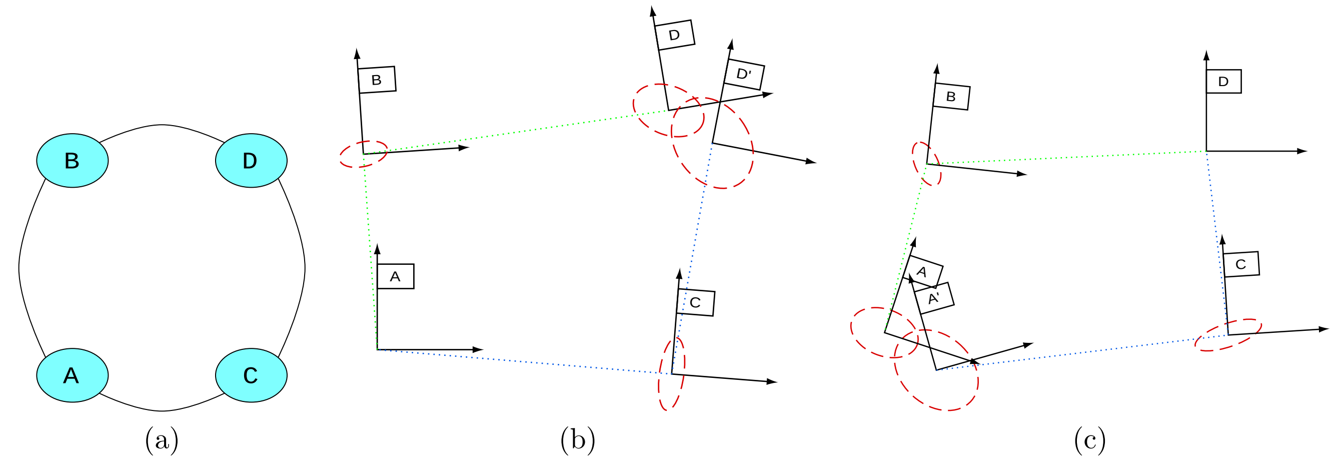


Fig. 2. The Dijkstra Projection using two different source nodes. (a) depicts the topological arrangement of the *Atlas* graph. (b) uses map-frame A as the source of the projection. (c) uses map-frame D as the source. The ellipses on the coordinate frames represent the accumulated projection error. Clearly we see that the shortest path from map-frame A to D is via map-frame B.

where η_a is the number of matched observations, $\|Z\|$ is the total number of recent observations. The parameter $\alpha \in [0 \rightarrow 1]$ reflects the relative importance placed on the robot uncertainty and successful explanation of sensor data. If in 2D, we specify maximum acceptable uncertainties in location and orientation, $\sigma_x, \sigma_y, \sigma_\theta$, then $\det(\Sigma_{\mathbf{x}_\star})$ is the product $\sigma_x^2\sigma_y^2\sigma_\theta^2$.

At any given time, there may be several hypotheses running on the *Atlas* graph. Each map-frame can support at most one hypothesis at a time, and the maximum number of total hypotheses is fixed so that the computational requirements remain bounded.

C. Genesis

Since we bound the complexity (for example the number of features) of each map, when we enter unexplored regions we need to create new local map-frames. This is done as follows.

1. The current robot pose defines the origin of a new frame. Thus, the transformation from the old to the new frame is simply the robot's position in the old frame at the time the new frame is created.
2. The robot pose is initialized to zero in the new frame.
3. The uncertainty of the transformation is set to the uncertainty of robot pose in the old frame.
4. The uncertainty of the robot pose in the new frame is zero by definition. All of the uncertainty of the robot pose at the time of Genesis is captured by the uncertain transformation.

The Genesis process adds a new vertex and edge to the *Atlas* graph. Mathematically, the generation of a new map-frame $\mathcal{M}_j$ and robot pose, $\mathbf{x}_j$, via Genesis is a function of an old map-frame, $\mathcal{M}_i$ and robot pose, $\mathbf{x}_i$.

$$(\mathcal{M}_j, \mathbf{x}_j) = g(\mathcal{M}_i, \mathbf{x}_i) \tag{6}$$

D. Map-Matching

Genesis creates new maps to explain unexplored areas. However, eventually the robot will reenter an area which it has already explored – "loop closing". Map-Matching is used to detect these situations and form an edge in the *Atlas* graph between two unconnected vertices (map-frames). Map-Matching uses common structure between two maps to form an estimate of the transformation between them.

First, we define the degree of match between $\mathcal{M}_i$ and $\mathcal{M}_j$ as the probability that their union is not the empty set.

$$m_{ij} = m(\mathcal{M}_i, \mathcal{M}_j) \tag{7}$$
$$m(\mathcal{M}_i, \mathcal{M}_j) = p\left(\mathcal{M}_i \cap \mathcal{M}_j \neq \emptyset\right) \tag{8}$$

The exact form of m used for determining common structure is not dictated by the *Atlas* framework. But as an example, for feature based maps, we define the operation $\mathcal{M}_i \cap \mathcal{M}_j$ as the search for correspondence between features in $\mathcal{M}_i$ and $\mathcal{M}_j$.

Secondly, after determining the probability of a match between map-frames $\mathcal{M}_i$ and $\mathcal{M}_j$, we compute the estimate of the transformation T_i^j between the coordinate frames of $\mathcal{M}_i$ and $\mathcal{M}_j$, as well as its uncertainty, Σ_{ij}.

$$(T_i^j, \Sigma_{ij}) = t(\mathcal{M}_i, \mathcal{M}_j) \tag{9}$$

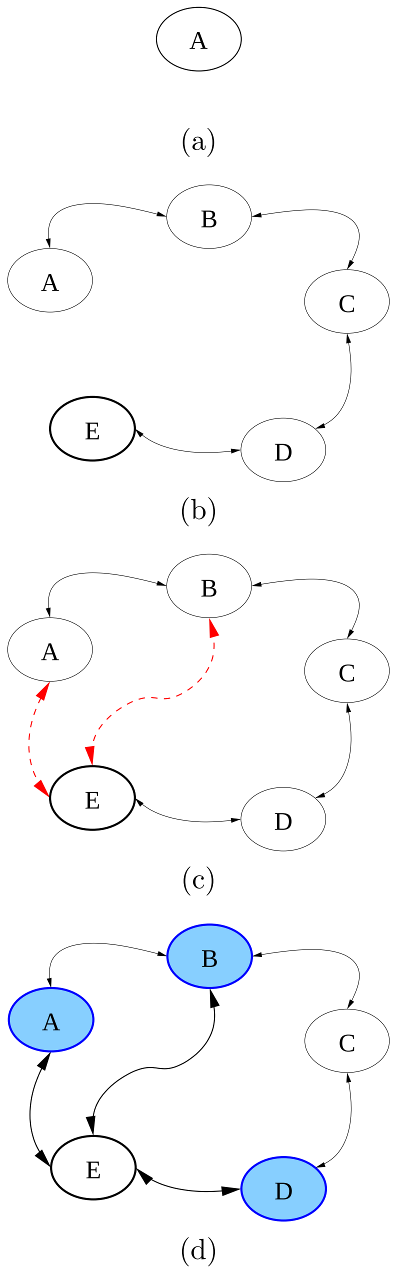


Fig. 3. An illustrative example: (a) *Atlas* graph is initialized with a single vertex. (b) As new regions are mapped the *Atlas* is extended with genesis edges. (c) Map matching closes loops in the graph. (d) Multiple hypotheses are used to aid in map traversal.

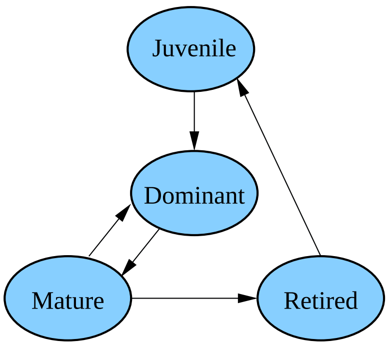


Fig. 4. *Atlas* hypothesis state transition diagram.

Note, that t does not depend on the robot pose. This is crucial since the navigation errors accumulated around a loop may be larger than simple data association of sensor measurements can handle.

We have to be careful that the edge formed by Map-Matching is not ambiguous because of repetitive structure in the environment. We can get an assessment of how repetitive structure is within a map by performing a Map-Match with itself. Only structure elements that match uniquely within a map should be used to evaluate a match to another map.

Much work has already been done in the area of map matching by (Tardos et. al.) with the techniques of a branch and bound method within the Joint Compatibility Test.

J. Neira and J.D. Tardós. "Data Association in Stochastic Mapping Using the Joint Compatibility Test," IEEE Trans. Robotics and Automation, vol. 17, no. 6, pp. 890-897, Dec 2001.

E. Traversal

Once the robot has mapped an area, it can reuse previously built maps when the current frame is no longer adequate. When the robot leaves the area around which a map-frame was created, it will fail to match sensor measurements thus its performance metric q (equation 4 will tend to zero, indicating poor performance. Thus in attempt to find a better hypothesis, we instanteate hypotheses in adjacent map-frames by projecting the current robot pose and uncertainty across the transformation edges in the local neighborhood of the current *Atlas* vertex.

There are four types of hypotheses which we label as juvenile, mature, dominant and retired. See Figure 4 for a state transition diagram. All types, except retired ones, process sensor measurements and evaluate the same performance metric, q. Mature hypotheses can extend their maps (increase the map's complexity) and spawn new hypotheses. Juvenile hypotheses can only process sensor data with regard to the existing map. We do not allow juvenile hypotheses to extend their map because they are used to test how well a particular map explains the sensor data and not whether a new map could be built from the data.

A juvenile hypothesis can "mature" when after a probationary period its performance metric, q, becomes greater than any other mature hypothesis. If at the end of this probationary period a juvenile hypothesis's quality does not warrant promotion is it simply deleted.

The mature hypothesis with the best performance metric is considered the dominant hypothesis. The dominant hypothesis is used for publishing current robot pose and local features to clients of the *Atlas* framework. In other words, it is the output of the framework.

Mature hypotheses that fail to perform well are saved and "retired". A retired hypothesis may be reactivated at a later time as a juvenile.

If we have only one mature but failing hypothesis, then a new one must be created to explain current sensor data. This is done by Genesis (Section III-C). This situation will occur when the robot moves into an unexplored area.

The initialization of a juvenile hypothesis, and its graduation to mature status warrant further description. When creating a juvenile hypothesis in a retired map-frame, $\mathcal{M}_i$, we need to reinitialize its robot pose, $\mathbf{x}_i$, using the robot pose, $\mathbf{x}_j$, from an adjacent map-frame, $\mathcal{M}_j$. See Figure 5. First we seed the hypothesis with a robot pose $\mathbf{x}_j^\star$ projected into frame i.

$$\mathbf{x}_i^\star = T_i^j \oplus \mathbf{x}_j \tag{10}$$

$$\begin{aligned}\Sigma_{\mathbf{x}_i}^\star = {} & \mathrm{J}_1\left(T_i^j, \mathbf{x}_j\right)\Sigma_{ij}\mathrm{J}_1\left(T_i^j, \mathbf{x}_j\right)^T + \\ & \mathrm{J}_2\left(T_i^j, \mathbf{x}_j\right)\Sigma_{\mathbf{x}_j}\mathrm{J}_2\left(T_i^j, \mathbf{x}_j\right)^T\end{aligned} \tag{11}$$

The hypothesis now enters a bootstrapping phase in which a consistent initialization of the vehicle into the juvenile hypothesis is sought. Sensor measurements, interpreted with the seeded robot pose, $\mathbf{x}_i^\star$, are accumulated. This continues until we have collected enough measurements, Z, to explicitly solve through some function w for the robot pose independently of $\mathbf{x}_i^\star$. This approach conserves the statistical independence of map-frames.

$$(\mathcal{M}_i^{\text{new}}, \mathbf{x}_i^{\text{new}}) = w(\mathcal{M}_i, Z) \tag{12}$$

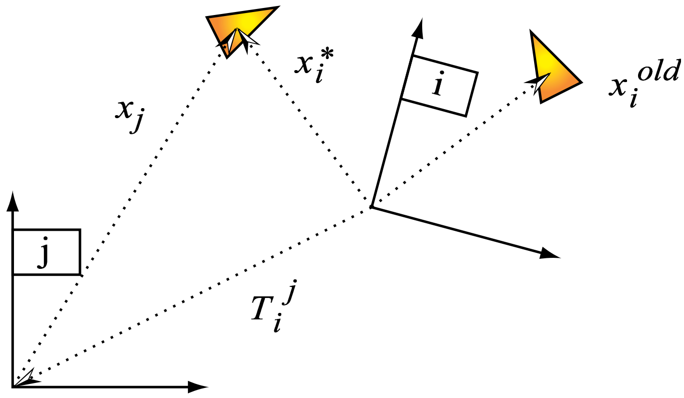


Fig. 5. For equation 10.

If an explicit solution to w cannot be computed because of lack of explained sensor measurements, then the hypothesis is invalid and terminated. Otherwise we have a tenable juvenile hypothesis.

F. *Edge Refinement*

When there is more than one mature hypothesis, we can refine the estimate of the transformation between them – the *Atlas* graph edge. Consider the case of two mature hypotheses in adjacent map-frames, $\mathcal{M}_i$ and $\mathcal{M}_j$. Both maintain an estimate of the current robot pose, $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. In combination they form a measurement of the transformation T_i^j:

$$T_i^j \oplus \mathbf{x}_j = \mathbf{x}_i \tag{13}$$

$$T_i^j = \mathbf{x}_i \ominus \mathbf{x}_j \tag{14}$$

From Equation 14, we can write the covariance of the observation, Σ_{ij}, as a function of the robot uncertainties, $\Sigma_{\mathbf{x}_i}$ and $\Sigma_{\mathbf{x}_j}$.

$$\begin{aligned}M_i &= \mathrm{J}_1\left(\mathbf{x}_i, \ominus\mathbf{x}_j\right) \\ M_j &= \mathrm{J}_2\left(\mathbf{x}_i, \ominus\mathbf{x}_j\right)\mathrm{J}_\ominus\left(\mathbf{x}_j\right) \\ \Sigma_{ij} &= M_i\Sigma_{\mathbf{x}_i}M_i^T + M_j\Sigma_{\mathbf{x}_j}M_j^T\end{aligned} \tag{15}$$

We update the prior estimate T_i^{j-} (the existing graph edge) with the observation T_i^j to form the refined estimate T_i^{j+}, and its uncertainty Σ_{ij}^+. Since we do not maintain the cross covariances between robot estimates in different maps, we advocate the use of Covariance Intersection [?] to perform the update.

$$\Sigma_{ij}^+ = \left[\omega\left(\Sigma_{ij}\right)^{-1} + (1-\omega)\left(\Sigma_{ij}^-\right)^{-1}\right]^{-1} \tag{16}$$

$$T_i^{j+} = \Sigma_{ij}^+\left[\omega\Sigma_{\mathbf{x}_i}^{-1}\mathbf{x}_i + (1-\omega)\Sigma_{\mathbf{x}_j}^{-1}\mathbf{x}_j\right] \tag{17}$$

Where,

$$\omega = \arg\min_{\omega} \left\| \Sigma_{ij}^{+} \right\| \quad (18)$$

If the uncertainty in local maps decreases, then with each map transition we can also improve our estimate of the transformation between frames. Therefore not only do we acquire local convergence, but global as well. For a detailed analysis of optimality and global convergence see Leonard's companion paper[**?**].

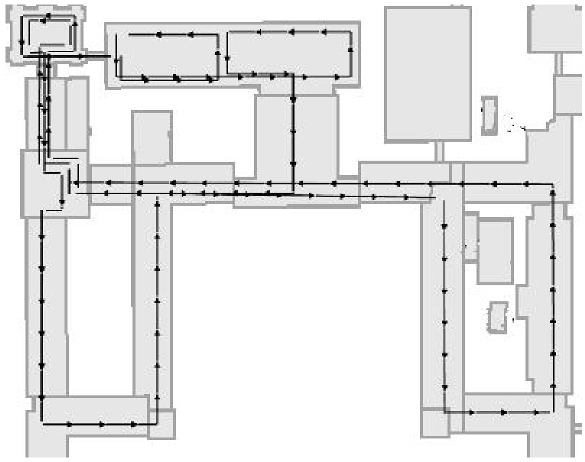
IV. Results – Mapping the Infinite Corridor

This section describes and presents results of an lengthy experiment conducted within and around MIT's "Infinite Corridor". These results constitute a compelling demonstration of the constant time large scale properties of the *Atlas* framework. The experiments utilized a standard B21 mobile robot equipped with SICK scanning laser and a ring of 24 polaroid ultrasonic sensors. The results presented here, using both sonar and laser sensors along with odometry , are from a real-time C++ implementation of the *Atlas* framework using and Extended Kalman Filter for local navigation.

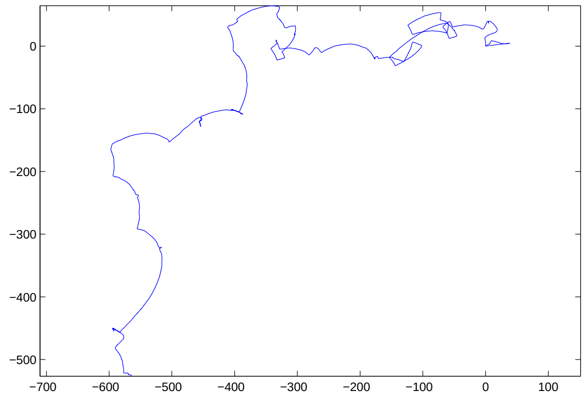
Onboard odometry was the only other source of information used.

Figure 6(a) shows the topological path of the vehicle superimposed on an architectural drawing of the MIT main campus. The route was chosen to highlight the key capabilities of of the *Atlas* framework - constant time processing and re-use of previously mapped regions. Firstly the total path length and experiment duration was substantial driving a little under 2.2 km in 2.5 hours. Secondly, the route contains nested loops of various sizes and topologies. Successful processing of this data set necessarily involves "loop closing" - a notoriously hard aspect of the SLAM problem.

Figure 6(b) shows the dead reckoned path resulting from simply integrating the odometry data which is clearly a poor estimate of the true trajectory. This however is in stark contrast to the outcome of processing the entire data set using laser and odometry data shown in Figure 8(a). In all 125 maps were built each containing a maximum of 10 mapped line segments. The constant time property of the *Atlas* framework when applied to the large scale SLAM problem is shown in Figure 7. The figure shows the instantaneous sum of kernel and user time for the *Atlas* process as well as its mean value. Note that as more features are mapped and more map-frames are created the mean processor load stays constant. [1]. Figure 7 also plots the numerical label of the dominant map-frame with time. During map-frame genesis a



(a)



(b)

Fig. 6. a) An architectural drawing of part of the MIT campus overlaid with the topology of the driven route. b) The trajectory derived from odometry alone

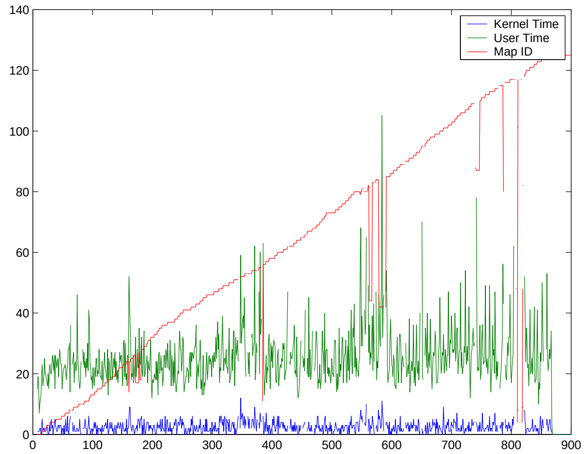
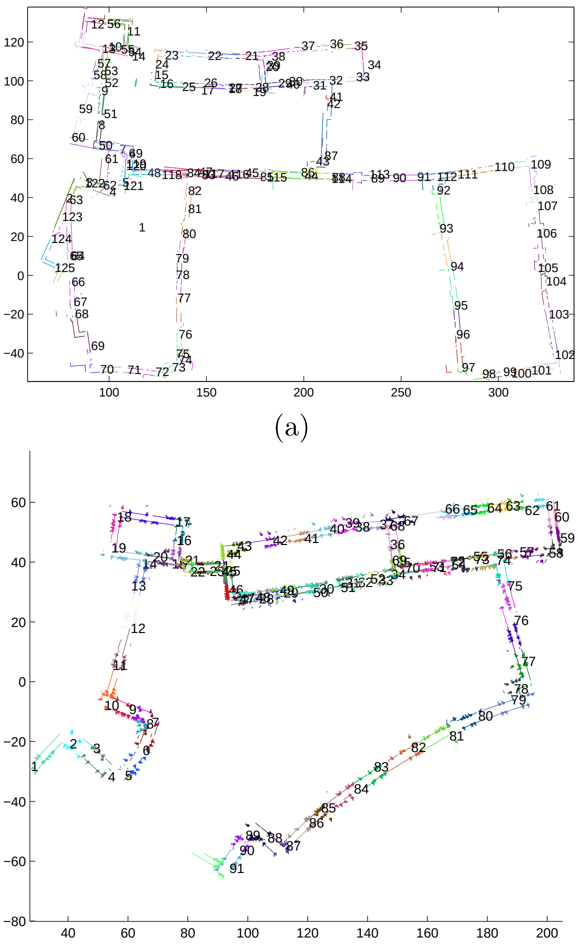


Fig. 7. Caption goes here

[1]There is a small increase stemming from visualization computation

counter is incremented and the newly created map is labelled with its value. As more and more new ground is covered the value of the dominant map ID will tend to increase. When however the robot returns to a previously mapped area the dominant map ID will decrease if loop closure is successful. For example approximately one hour into the experiment the robot returned to an area first mapped 45 minutes earlier. Similarly after 2hrs 15 minutes the vehicle returned to a region mapped just five minutes after the experiment began.



(a)

(b)

Fig. 8. Caption goes here

Figure 8(b) shows results using data from the same experiment but using the polaroid sonars instead of the laser. The local navigation is achieved using a combination of Delayed Decision Making [11], RANSAC [?] and wide beam soanr interpretation [?] . Space does not permit a detailed explanation of this process but Figure 9 illustrates the underlying principal. Sonar measurements are accumulated by the vehicle from a set of uncertain vantage points. The RANSAC algorithm provides robust perceptual grouping which the DDM mechanism uses to provide a stochastically consistent feature initialization or update.

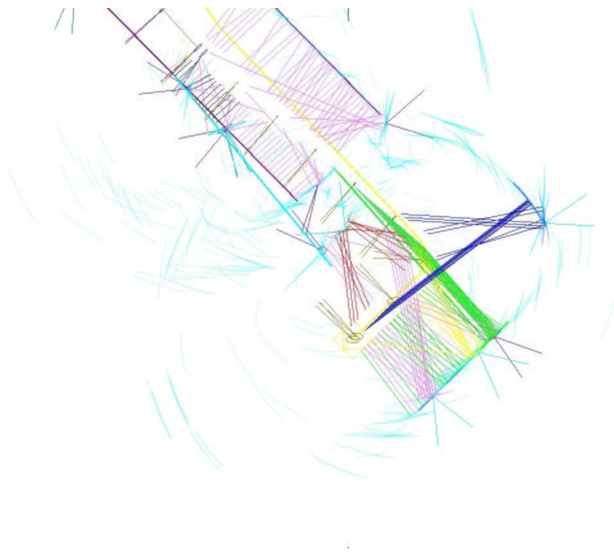
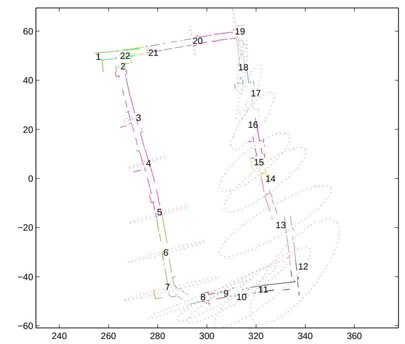


Fig. 9. Using sonar data from multiple vantage points to intialize features

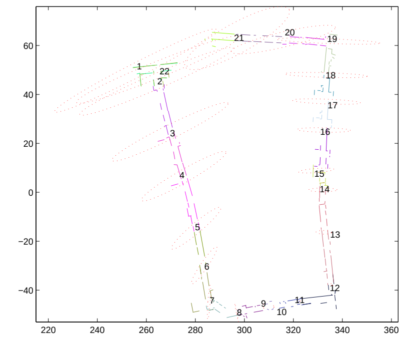
The corrective effect of sonar data on angular error is less than that of laser data. This effect prevented the processing of the full data set resulting in a smaller coverage than for the laser driven case (although a greater number of features were created). When traversing down a long corridor using sonar alone the growth in angular error was such that condition two in section II-A was violated - the accumulated uncertainty around the loop that should have been closed was large enough to cause ambiguous map matches. At this point the *Atlas* was halted and results saved.

Figures **??**(a) and (b) are similar to of Figure 2 and illustrate with real data the effect of representing uncertainty around a loop of map-frames by using different Dijkstra projections. Figure **??**(a) uses the map in the top left as the source whereas Figure **??**(b) shows exactly the same map-frames only using the bottom left one as a source. Note how uncertainty is always greatest across the loop from the source corresponding to the greatest relative Dijkstra distance.

To our knowledge this is the most extensive indoor demonstration of real-time large scale SLAM. The experiment was only terminated when the majority of the connected corridors had been explored resulting in the mapping 1154 features in 124 maps.



(a)



(b)

Fig. 10. Representing combined map-frame uncertainties from opposing locations

References

[1] S. Thrun, “An online mapping algorithm for teams of mobile robots,” *Int. J. Robotics Research*, vol. 20, no. 5, pp. 335–363, May 2001.

[2] J. Guivant and E. Nebot, “Optimization of the simultaneous localization and map building algorithm for real time implementation,” *IEEE Transactions on Robotic and Automation*, vol. 17, no. 3, pp. 242–257, June 2001.

[3] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds., pp. 167–193. Springer-Verlag, 1990.

[4] A. J. Davison, *Mobile Robot Navigation Using Active Vision*, Ph.D. thesis, University of Oxford, 1998.

[5] B. J. Kuipers, “The spatial semantic hierarchy,” *Artificial Intelligence*, 2000, To Appear.

[6] J. Leonard and H. Feder, “Decoupled stochastic mapping,” *ieeejoe*, vol. 26, no. 4, pp. 561–571, 2001.

[7] K. Chong and L. Kleeman, “Large scale sonarray mapping using multiple connected local maps,” in *International Conference on Field and Service Robotics*, ANU, Canberra,Australia, December 1997, pp. 538–545.

[8] S.B Williams, G. Dissanayake, and H. Durrant-Whyte, “An efficient approach to the simultaneous localisation and mapping problem,” in *ieeeicra*, 2002, pp. 406 –411.

[9] P. Newman, M. Bosse, and J. Leonard, “Feature based exploration,” Submitted for publication for ICRA 2003, 2002.

[10] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” *IEEE Trans.Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, september/october 1989.

[11] R. Rikoski, J. Leonard, and P. Newman, “Stochastic mapping frameworks,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002, pp. 426– 433.