Mapping Opaque and Confined Environments Using Proprioception

Jacob Everist and Wei-Min Shen
Information Sciences Insitute
University of Southern California
{everist,shen}@isi.edu

***Abstract*— Mapping opaque and confined environments such as caves and pipes is a challenging problem for mobile robots because sensor information is severely limited to the immediate proximity of the robot due to the extreme environmental conditions. The robot must also be flexible and agile in unstructured environments while still providing accurate pose estimation. This paper presents a solution to mapping a 2-dimensional tube by using only a snake robot's proprioceptive joint angle sensors. We assume that the tube is sufficiently smooth and that we know the tube width. We propose techniques for (1) pose estimation of a snake robot by using a self-posture motion model, (2) correcting error in pose estimation using only the snake's internal configuration over time, and (3) building environmental features using only self-occupancy and contact detection. Our goal is to use the minimal amount of sensor information possible to build an accurate spatial map of the environment. We have tested the proposed techniques in simulated environments and experimental results show that they are both effective and efficient for mapping tube environments. We plan to extend these techniques to deal with more complex confined environments beyond single-path tubes.**

I. Introduction

Environments such as pipes and caves are difficult to robotically map because they are opaque and confined. An environment is opaque when external range and vision sensors are inoperable because of lighting conditions, environmental fluids, or tight proximity to obstacles. An environment is confined when a robot's mobility is restricted by the close proximity to multiple obstacles. An illustration of a robot exploring such an environment can be seen in Figure 1.

A solution to this problem would allow mapping and exploration of previously inaccessible environments. An approach that is invariant to ambient fluid (air, water, muddy water) would significantly impact cave exploration and underground science by the access to still deeper and smaller cave arteries regardless of whether they are submerged in water. Search and rescue operations would have additional tools for locating survivors in complicated debris piles or collapsed mines. Utility service workers would be able to map and inspect pipe networks without having to disable pipe operation.

These environments often render range sensors ineffective due to occlusions, distortions, reflections, and time-of-flight issues. Changing visibility and lighting conditions will also nullify the efficacy of vision sensors. This in turn forces us to rely on a strictly proprioceptive approach for building environmental information. Such sensors include accelerometers, force sensors, contact sensors, and joint sensors. This sensor information is by definition highly localized to the robot and sets us at a disadvantage to other mapping approaches that traditionally build maps with large sweeps of range data. We have fundamentally less information with which to build maps and make exploration decisions.

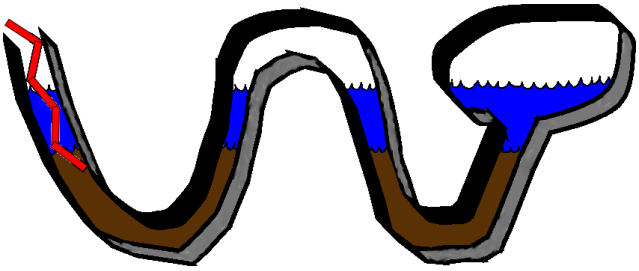

Fig. 1. Robot Exploring Confined Environment with Changing Ambient Fluid Conditions

Global positioning information is also unavailable because of the enclosed nature of the environment. Therefore, any mapping method will need an accurate pose estimation approach for the robot. Wheeled dead-reckoning may be very difficult or impossible if the environment is highly unstructured or otherwise untenable to a wheeled locomotion approach. Another method should be found.

Others have attempted to map similar environments. Efforts have been made to map abandoned mines [1] with large wheeled robots traveling through abandoned corridors. However, this approach cannot handle smaller pathways, collapsed or flooded mines. Robotic pipe inspection has been studied for a long time [2], but usually requires that the pipe be evacuated. The robotic solutions are often tailored for a specific pipe size and pipe structure. Recent work has focused on the exploration of flooded subterranean environments, with the recent accomplishment of mapping deep sink holes in Mexico [3]. This involves the use of a submersible robot, uses underwater sonar for mapping, and is not capable of amphibious activities.

A. Approach

In our approach we solve the scenario of mapping a 2-dimensional tube in a 3D simulated world. We attempt to do this with a minimal amount of sensor information and exclude the use of any range or vision sensors to mimic the constraints of a true opaque environment. In this paper, we assume that the minimum and maximum width of the pipe is

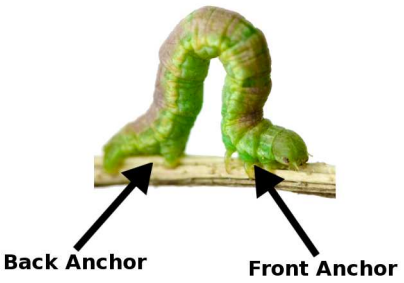

Fig. 2. Inchworm Movement with 2 Anchor Points to Ground

known to the robot apriori. This assumption may be relaxed in the future for robots to explore completely unknown caves where internal space is highly irregular.

We choose a serpentine robot as our platform to do exploration and mapping. A snake robot is particularly suited to exploring confined environments because of its flexibility of form and ability to adapt to the contours of the tube. We use the angular positions of the snake's joints as our sole source of sensor information. From this we build a spatial map of the tube.

To do this, we develop a unique odometry method called a self-posture motion model. So long as some point in the body of the snake is anchored with respect to the world, we can estimate the trajectory of the snake through the tube with comparable accuracy to wheeled dead-reckoning. Unlike wheeled dead-reckoning however, this method permits various types of locomotion and can function in unstructured environments.

A simple example would be an inchworm, seen in Figure 2, that always anchors either its front legs or its back legs to the ground. Because some point in the inchworm's body is always anchored to the ground, it could estimate its distance traveled so long as it can kinematically compute the relative distance between its front and back legs.

The accumulation of error in any odometry method is inevitable. Therefore, we develop a wall contour-based error correction technique using consecutive poses of the snake robot. A special pose is used that closely approximates the contours of the tube walls and a match technique is used to line up the contours between consecutive poses to then compute the error offset between the poses. This can be used to find error caused by the self-posture motion model when the snake slips along the walls and eventually correct it.

Finally we develop two methods for extracting features from the environment. (1) We use a contact detection method that uses only joint angle sensors and knowledge of the snake geometry. This will give us contact points between the robot and the obstacles of the environment. (2) We use the self-occupancy of the snake robot to indicate the lack of obstacles and the presence of free space. Both the contact detection and the self-occupancy information coupled with the self-posture motion model and its associated contour-based error correction give us a complete system for building a spatial map of the tube and its features.

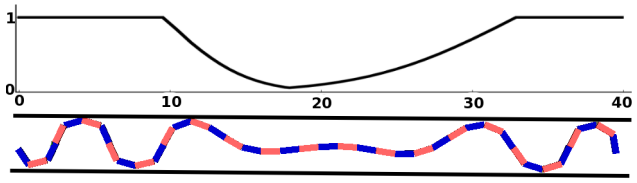

Fig. 3. Concertina Gait with $f_{mask}(i,t)$ graphed for some time t: Output vs. Joint i

We describe each component of our approach in the next sections and follow up with experimental results.

II. Tube Environment

We use Open Dynamics Engine[1] as our experimental testbed.

We have developed a random environment generator with several controllable parameters for running batch experiments. Such parameters include the length of the tube, the amount and sharpness of turning, how rough, blocky, or smooth the wall surfaces are, and the surface friction. Curved surfaces can be approximated by reducing the length and increasing the number of rectangular wall meshes.

We can also control the maximum and minimum width of the tube which are upper and lower bounds along the full length of the tube. The tube width is defined locally with respect to any point on a wall:

Definition For a given point P on a wall, there exists a point Q on the opposing wall that is closest to P. The *tube width* with respect to P is the distance from P to Q.

The friction between the tube walls and the snake body is constant throughout the tube but may be changed by the experimenter. The friction between the ground plane and the snake body is negligible so the interaction and locomotion is achieved by contact between the walls and the robot.

III. Snake Robot and Locomotion

Snake robots have been well-studied [9]. For our approach, we use the design of the snake robot depicted in Figure 3. The robot is composed of n body segments, each of which is a block with a height H, a segment width W, and a segment length L. There are $n-1$ angular joints connecting the body segments together at their front and back edges. Each joint is a single degree-of-freedom rotation joint with an axis parallel to the normal of the ground plane. This restricts the movement of the snake robot to a single plane.

The concertina gait, seen in Figure 3, is used by snakes for moving in confined environments such as burrow holes. It accomplishes this by anchoring to the sides of the walls and pushing itself forward.

This gait is particularly suited for our purposes and is the basis for our map-making approach. It has the key feature of being consistently anchored to the environment. These intermittent anchors are used as stable points of reference to

[1]ODE, http://ode.org

the global frame and are inputs to the self-posture motion model.

The concertina gait is very effective at navigating a pipe with curved surfaces in an open-loop fashion because the leading head of the snake is compliant to the walls. The head will then anchor the front half of the body after pushing itself forward a distance. The only thing that will halt the locomotion would be an acutely curved tube trajectory, anchor slipping, or a section of the pipe that is wider than the anchor width.

Our concertina gait is implemented with the pseudo-code shown in Algorithm 1. We calibrate a simple sinusoid curve that is sampled discretely at intervals to be applied to each robot joint and create a sinusoid posture that fits the width of the tube. We then multiply the joint values by a control mask seen in Figure 3 and then set the product directly to the joint servos.

Algorithm 1 Concertina Gait

1: **for** $i = 0$ to $n - 1$ **do**
2: $\quad jointVal = jointAmp * sin(i * interval)$
3: $\quad setServo(i, jointVal * f_{mask}(i, t))$
4: **end for**

The code cycles through each joint of the robot and sets the correct value. The variable $interval$ is the discrete sampling interval for the static sinusoid applied to consecutive joints. The variable $jointAmp$ is manually calibrated to produce the desired snake sinusoidal amplitude. The $jointAmp$ does not correspond to the actual amplitude of the snake sinusoid, only the proportional relation of the values $[-1, 1]$ output by the sine function to actual joint angles in degrees. For our experiments, $jointAmp = 70.0$ and $interval = 2\pi/8$.

The f_{mask} is the control mask, a discrete sampling of the control curve shown in Figure 3. The mask is sampled for each joint and shifted over time. The control mask is composed of a pair of Gaussians clamped at 1.0 and padded on either side of the peaks with unit step functions. The motion of the mask across the joints will produce a concertina gait. After a period of time, the mask is reset to its initial position.

Given a joint position i and time t, the mask $f_{mask}(i, t)$ is defined by:

$$f_{mask}(i,t) = \begin{cases} 1, i < t - T \\ h(i,t), t \geq i \geq t - T \\ 1, i > t \end{cases}$$

where

$$h(i,t) = \mathbf{min}\left\{1.0, \mathbf{max}\left\{G(i, t-T, \sigma_1), G(i, t, \sigma_2)\right\}\right\}$$

$$G(x, \mu, \sigma) = 1.5 * e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$h(i, t)$ is the maximum of two Gaussian curves clamped at 1.0. $G(x, \mu, \sigma)$ is a Gaussian curve sampled at position x. T is the distance between the Gaussian peaks. For our control, we set $T = n * 0.9$ where $n = 40$, the number of snake body segments. The width control parameters of the Gaussian curves are set to $\sigma_1 = 5.0$ and $\sigma_2 = 9.0$.

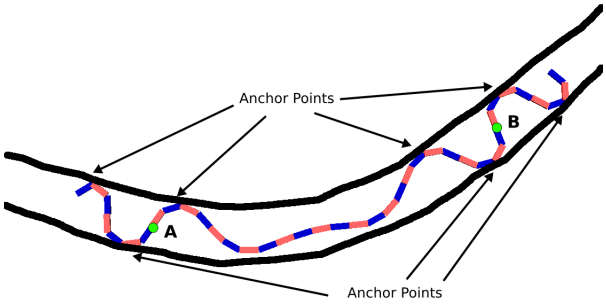

Fig. 4. Anchor points and reference points A & B

IV. Self-Posture Motion Model

The first component to any robotic mapping approach is the ability to know the position of the robot in the environment over time. Since we are unable to have a global positioning system, or a wheeled dead-reckoning system, we must a find a new way for estimating the snake's position over time.

Our new method, called the self-posture motion model, estimates the position of the snake robot using continuous anchoring in the environment and intra-body information. The change in pose is determined by kinematically computing the distance between consecutive anchor points. A plot of the created reference points over time shows the trajectory of the robot as seen in Figure 9.

The concertina gait, by its nature, is always anchored to the walls of the tube. Therefore, stable reference points to the global frame are always available. Figure 4 shows two reference points A and B, the latter being newly activated and its position computed kinematically with respect to the former.

Reference points are created at the joint locations of the snake. Active reference points are used to kinematically compute the position of new reference points. They are activated when (1) the output of f_{mask} at its position is equal to 1.0, and (2) the joints in the local neighborhood are determined to be stable by measuring the local joint variances. The reference points are deactivated when either of these conditions are violated.

The most common source of error is when the anchor points slip. This can happen for many reasons including low friction or strong reaction forces from other parts of the snake body. This can inject sudden drift error into the pose estimation and lead to errors in the overall map.

To our knowledge, no other approach uses intra-body position information as the method for global pose estimation. So long as the anchors hold fast and no slipping occurs, the trajectory information of the snake should be theoretically correct. However, the introduction of error is inevitable in any physical system and must be handled. We attempt to correct these errors in the next section.

V. Contour-Based Error Correction

The conventional method for reducing pose error of a mobile robot is to use the acquired sensor information to reduce

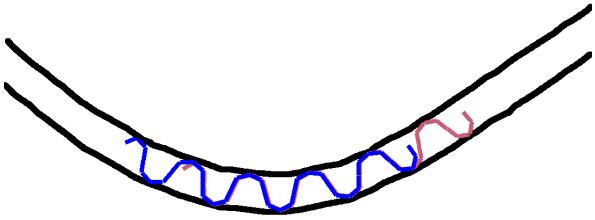

Fig. 5. 2 Stable Sinusoid Poses that Overlap

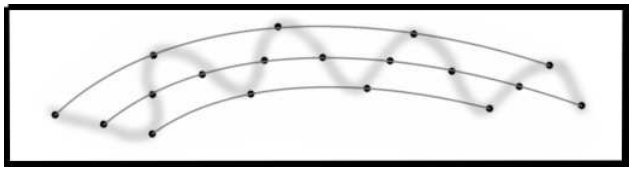

Fig. 6. Stable Sinusoid Pose Contact Points and Fitted Spline Contours, the Snake Body is Shadowed in the Background

the inconsistencies. Large swaths of range or vision data are used to match features in the environment and correct consecutive robot poses and compensate for the odometry error. This is often called scan-matching [7]. However, since our goal is to explore environments hostile to these kinds of sensors and that we are limiting ourselves to the minimum sensors possible, the joint angle sensors, we cannot use the existing approaches.

Again, we use the self-posture of the snake robot in our method. This time, we attempt to do a type of scan-matching method with consecutive poses of the snake. The intuition being that in the two stable poses shown in Figure 5, the snake body follows the contours of the tube. Since both of these poses significantly overlap each other in space, the respective body configurations should show a strong correlation. We use this likelihood to further estimate the relative pose difference between them.

We call the snake configurations shown in Figure 5, stable sinusoid poses since the full body of the snake is a sinusoid. Stable sinusoid poses are also the rest states of a concertina locomotion, so it is natural for us to use this configuration. Since we know the maximum and minimum tube width, the stable pose is calibrated to make every peak and trough of the sinusoid be anchored to the wall by static friction.

If our tube is sufficiently continuous and sufficiently smooth, we can approximate the contours of the wall. The peaks and troughs can be used as control points for a fitted cubic spline and will approximate the contours of the wall. The contact points and the fitted splines are shown in Figure 6. Also shown is a tube "centerline" which is constructed by fitting a spline on the inflection points of the sinusoid.

Now, if we can take the contour curves from two consecutive poses and match them together so they overlap as much as possible, we should be able to measure the pose difference which can be used to correct the error in the self-posture motion model.

The cost function we have developed is implemented as follows: if we are given the two curves shown in Figure 7a,

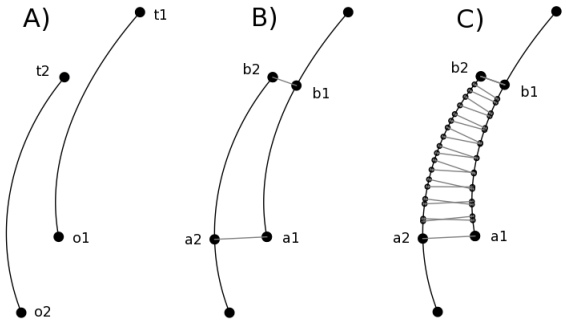

Fig. 7. Two Contour Curves: (a) terminating points of curve 1 and 2, (b) the comparable sections of each curve, and (c) the uniformly selected points and their minimum distance.

the cost associated with their relative poses is computed by (1) delineating a section of each curve to compare, (2) selecting points on a curve and computing the minimum distance to the opposing curve, and (3) summing the distances for a total cost associated with the two curves.

To compute the comparable sections of the curves, we first associate one curve to be in front, and the other curve to be in back corresponding to the traveling motion of the robot between poses. The comparable section of the front curve is determined by setting $a1$ to $o1$ and $b1$ to the point on the curve closest to $t2$ on the opposing curve. Similarly the comparable section of the back curve is determined by setting $b2$ to $t2$ and $a2$ to the point on the curve closest to $o1$ on the opposing curve. This is illustrated in Figure 7b and referred to as $findSections()$ below.

We then select a uniform distribution of k points on the section of the front curve and compute the shortest distance to any point on the back curve section. Similarly, we pick a uniform distribution of k points on the back curve and compute the shortest distance to any point on the front curve section. This is illustrated in Figure 7c. In our example, k = 10. The cost is computed by summing each of the 2k distances. The full process is shown in the pseudocode below with $uniformPoints()$ finding a uniform distribution of k points on the given curve, and $closest()$ finding a point on the given curve closest to the given point:

Algorithm 2 Curve Matching Function: $f_{match}(\gamma_1, \gamma_2)$

1: $\tilde{\gamma_1}, \tilde{\gamma_2} = findSections(\gamma_1, \gamma_2)$
2: $\bar{x} = uniformPoints(k, \tilde{\gamma_1})$
3: $\bar{y} = uniformPoints(k, \tilde{\gamma_2})$
4: $C_1 = \sum_{i=1}^{k} |x_i - closest(x_i, \gamma_2)|$
5: $C_2 = \sum_{i=1}^{k} |y_i - closest(y_i, \gamma_1)|$
6: return $C_1 + C_2$

To compute the fitting cost associated with 2 complete stable sinusoid poses, we must perform 3 comparisons: the peak curves, the trough curves, and the centerline curves. The complete and total cost of a fit is the sum of these 3 comparisons:

$$f_{cost} = f_{match}(\gamma_{p_1}, \gamma_{p_2}) + f_{match}(\gamma_{t_1}, \gamma_{t_2}) + f_{match}(\gamma_{c_1}, \gamma_{c_2})$$

This cost function remains invariant to lengthwise overlap of two curves which is critical to prevent an optimization function from naively placing two curves on top of each other. Instead, it tries to find the sections of two curves that overlap the best, regardless of the *length* of the overlap.

If we give this cost function to a general optimization algorithm with the relative offset variables x, y, θ between the curve origins as input, it will search for a best fit according to our desired constraints. Bad fits and local minima do exist with this cost function, but they are generally not encountered if the relative pose starts off with a good guess. This is usually the case when our self-posture motion model's error is bounded to minor slips.

VI. Contact Detection and Self-Occupancy

With tools for estimating robot position over time, we can now extract features about the environment and begin plotting them into a map. The contour curves we built in the previous section could be used as a map feature, but we have also developed two others: (1) contact detection and (2) self-occupancy.

A. Contact Detection

Contact detection is the process of detecting points on the robot's body that are contacting an obstacle without instrumenting the robot with tactile sensors. This can be done with only joint angles and some means of determining whether the environment is resisting. This has been studied previously in the field of grasping [6] [8] [4] [5].

The major difference to our approach is that in these previous studies, a robotic arm was permanently anchored to a fixed frame of reference and operating in a small workspace. In our approach, we are on a mobile robot platform, whose position in the global frame is uncertain and whose workspace is large and mostly unreachable from a single pose. So long as we properly anchor during contact detection, we can use the above methods. In the interest of space, we refer the reader to the cited publications for more information.

B. Self-Occupancy

Self-occupancy is readily available environmental information that has not been used much because of the superiority of range and vision sensors. However, now that we are in a sensor-starved environment, this tool takes on a new importance. The intuition of this approach is that the presence of the robot in space indicates the lack of obstacles and the presence of free space since the robot cannot intersect with another physical object.

As the snake robot moves through the environment, the position of each of the body segments in space with respect to an active reference point are collected at frequent periodic intervals. This gives us a history of position information and allows us to plot the free space in the environment. The free space could be represented as a series of intersecting polygons or plotted into an occupancy grid as seen in 8b. Our current implementation represents the free space as polygons which allows us to easily rearrange the data when better pose estimations are available. The polygons are then plotted into an occupancy grid for display and analysis purposes.

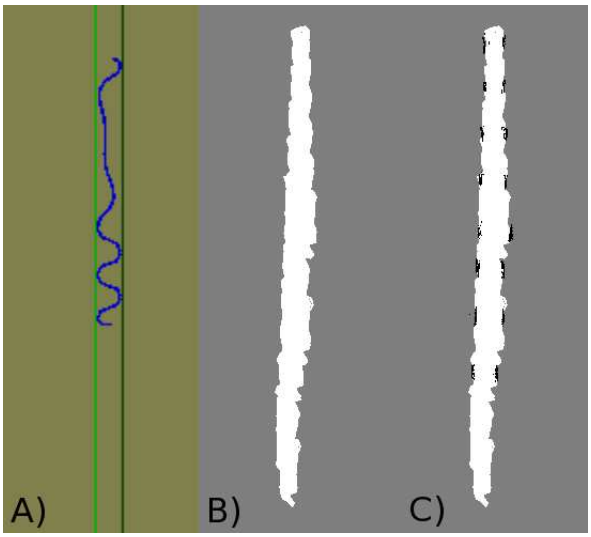

Fig. 8. (a) Snake traversing a straight tube in simulation, (b) self-occupancy plotted into an occupancy grid, and (c) with periodic contact detection data added.

As one can see, the self-occupancy data is significantly more dense and gives a better representation of the tube than with the contact points. The map is incomplete since we do not know if there are side passages or fine details on the tube wall. This can be complemented with contact data to fill in the missing map details. Figure 8c shows a map with both contact detection and self-occupancy data added to the occupancy grid.

VII. Experiments and Analysis

Our experiments show that our mapping approach has good accuracy so long as the concertina gait is tuned for the width. We use a single width calibration of the concertina gait for all of our experiments.

We show that if we reduce the length of the body segments, the snake can no longer anchor to the walls for its given calibration and the resultant map is severely degraded. However, longer snake segments still produce good maps. Changes in snake segment width do not yield much change in map quality.

Furthermore, we show that a low friction environment with severe slipping will result in large pose error from the self-posture motion model, but is mostly corrected by the contour-based error correction.

We perform a set of 3 experiments varying surface friction, body segment length and body segment width. We randomly generate 10 different tube environments of length 20 meters. The maximum and minimum width of the tube is 0.45 and 0.4 meters respectively. The number of segments in the snake body is $n = 40$. There is an additional "entrance canal" at the beginning of the tube to introduce the snake into the tube in a controlled and consistent manner. For each randomly

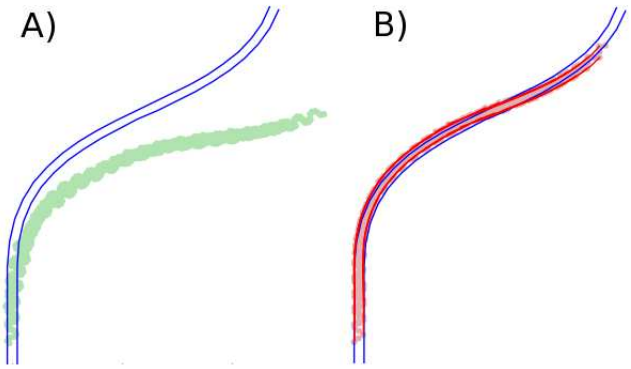

Fig. 9. (a) Plot of reference points with self-posture motion model and (b) reference points after contour-based error correction.

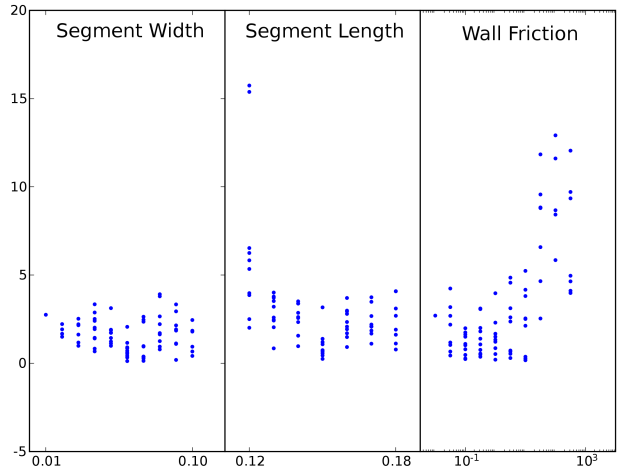

Fig. 10. Termination Error (in meters) of experiments of (a) segment width (meters), (b) segment length (meters), and (c) wall static friction coefficient.

generated tube environment, we run each of the individual friction and body segment experiments.

Figure 9 shows a typical example of a traversed environment in the experiments. Figure 9a shows the history of active reference points purely estimated by the self-posture motion model. Figure 9b shows the estimated history after the contour-based error correction.

We define our error metric to be the termination error. The termination error is computed by measuring the distance between the actual terminating position of the snake trajectory and the estimated terminating position. The termination error of Figure 9a) and 9b) are 4.57 and 0.516 respectively.

The results of all of our experiments are shown in Figure 10. The experiment with the most error comes from the body segment length set to 0.12. This happens because the length of the snake shrunk and the tube walls are too wide for the snake robot to anchor to them. The concertina gait control uses a fixed set of parameters and is under open-loop control so it is currently unable to adapt to different tube widths. The error-correction is unable to fix the accumulated error because it did not receive good contour information.

The results of the friction test show a significant accumulation of error in high friction environments. This is caused by the difficulty of the locomotion to make forward motion progress. What is not shown on the plot is the significant error accumulated by very low friction environments due to the constant slipping. However, this error is corrected quite effectively by the contour-based fitting method.

VIII. Conclusion

The exploration and mapping of opaque and confined environments is a challenging problem because of the sensor limitations and the challenge to mobility. However, the confined environment also provides easy anchoring for reference points to the global frame. Environments with wide open chambers would require a different anchoring approach.

Future work will expand the range of environments we can explore such as having unknown tube width, more irregular wall contours, multi-path tubes, and also expanding the environment into 3 dimensions. This is a new and exciting area of robot mapping research which will have a significant impact on accessing places beyond our current technologies such as hazardous mines, pipe networks, and subterranean caves, both on Earth and on other planets.

References

[1] Christopher Baker, Aaron Christopher Morris, David Ferguson, Scott Thayer, Chuck Whittaker, Zachary Omohundro, Carlos Reverte, William Red L. Whittaker, Dirk Haehnel, and Sebastian Thrun. A Campaign in Autonomous Mine Mapping. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, volume 2, pages 2004 – 2009, April 2004.

[2] T. Fukuda, H. Hosokai, and M. Uemura. Rubber Gas Actuator Driven By Hydrogen Storage Alloy For In-Pipe Inspection Mobile Robot with Flexible Structure. *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 3:1847–1852, May 1989.

[3] M. Gary, N. Fairfield, W.C. Stone, D. Wettergreen, G.A. Kantor, and J.M. Sharp Jr. 3D Mapping and Characterization of Sistema Zacatn from DEPTHX (DEep Phreatic THermal eXplorer). In *Proceedings of KARST08: 11th Sinkhole Conference ASCE*, 2008.

[4] R. A. Grupen and M. Huber. 2-D Contact Detection and Localization Using Proprioceptive Information. In *1993 IEEE International Conference on Robotics and Automation*, volume 2, 1993.

[5] S. Haidacher and G. Hirzinger. Contact Point Identification in Multi-Fingered Grasps Exploiting Kinematic Constraints. In *2002 IEEE International Conference on Robotics and Automation*, volume 2, Washington DC, USA, 2002.

[6] M. Kaneko and K. Tanie. Contact Point Detection for Grasping an Unknown Object Using Self-Posture Changeability. *IEEE Transactions on Robotics and Automation*, 10(3), 1994.

[7] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4:333–349, 1997.

[8] N. Mimura and Y. Funahashi. Parameter Identification in the Grasp of an Inner Link Mechanism. In *1993 IEEE International Conference on Robotics and Automation*, Atlanta GA, USA, 1993.

[9] Y. Umetani and S. Hirose. Biomechanical Study of Serpentine Locomotion. In *Proc. 1st RoManSy Symp.*, pages 171–184, Udine, Italy, 1974.