Robot Navigation Using Qualitative Landmark States from Sketched Route Maps

George Chronis
Computer Science
University of Missouri - Columbia
chronisg@missouri.edu

Marjorie Skubic
Electrical and Computer Engineering
University of Missouri – Columbia
skubicm@missouri.edu

***Abstract*—The goal of this work is to illustrate and evaluate a novel method for communicating with a mobile robot, namely, by drawing a sketch. The user draws a sketched route map to direct a mobile robot along a specified path. In this paper we focus on the navigation of the sketched path in the real environment. Challenges include sketch inaccuracies such as distortion or abstraction and low sensory resolution of the robot. Our method is based on utilizing spatial relations to extract a sequence of qualitative landmark states from the sketched map, which in turn the robot follows in a real environment to replicate the sketched route. Several examples are included.**

Keywords- sketch-based navigation, human-robot interaction, spatial relations, histogram of forces

I. INTRODUCTION

The goal of this work is to illustrate and evaluate a novel method for communicating with a robot, namely, by drawing a sketch. The user sketches a map on a PDA screen and a desired route for the robot to follow. We show in this paper how to make a robot follow the drawn path, without using absolute coordinates from the sketch. Instead, we extract qualitative landmark states and that involve spatial relations between the robot and the surrounding objects. A robot reaching the desired goal by following a specified path is a good indication that the extracted landmark states were adequate and that the robot could successfully follow the desired route. Our navigation approach is shown to handle inaccuracies while sketching, such as skewing, scaling or omission of objects with respect to the real environment.

Some limited work has been done in using sketches to direct movement. Igarashi et al. proposed a path drawing technique overlaid on a virtual scene, as a means of specifying a route through the virtual environment [1]. Ferguson et al. developed a sketch interface for military course-of-action diagrams, which are used for strategic planning [2]. Freksa et al. proposed the use of a schematic map for directing robot navigation [3]. A schematic map is described as an abstraction between a sketch map and a topological map; an example is a subway map. The closest work is that of Kawamura et al. in which the user specifies a robot path by selecting via points on a sketch of the environment [4]. Artificial landmarks are placed in the scene and on the sketch to act as landmarks in the navigation process.

Our previous work [5][6] discusses the use of a sketched route map for directing mobile robot navigation. A route is extracted from the sketched path in the form of a sequence of landmark states with corresponding actions, where each landmark state is a qualitative condition based on the spatial relationship of landmarks relative to the robot. The previous work was adequate for simple map configurations but was not robust in more complex environments.

In this paper we present significant improvements to the navigation methods, which utilize the extracted landmark states. The enhancements presented in this paper support more complex sketches as well as more abstract sketches, i.e. sketches that have scaling differences from the real environment (due to the inaccuracies of the sketch). We have tested the system with sketches drawn by different users (collected as part of a user study). In addition, we have varied the navigation environment to illustrate the ability of the robot to adapt and navigate robustly in spite of these variations. Examples are included in the paper.

In the next section we provide background information, and in Sec. III, we explain our approach and the improvements over previous algorithms. Sec. IV presents experimental results using a Nomad 200 robot simulator. Finally, in Sec. V, we summarize and discuss future plans of software enhancements and integration with other frameworks.

II. BACKGROUND

A sketched route map is drawn to help someone navigate along a path for the purpose of reaching a goal. An example is shown in Fig. 1(a). Although route maps do not generally contain complete map information about a region, they do provide relevant information for the navigation task. People sketch route maps to include landmarks at key points along the path and use spatial relationships to help depict the route, often adding arrows and other notation for clarity [7].

The depiction of the environment structure is not necessarily accurate and may even distort the actual configuration [8]. For example, a 60-degree turn in the physical environment may be sketched as a 90-degree turn. However, as the route follower navigates in the real environment, his motion is constrained by the environment so that the distortion is corrected and the route can be completed.

Research by Michon and Denis [9] provides insights into how landmarks are used for human navigation and what are considered to be key route points. In studying route directions, they found that landmarks were used more frequently at four types of critical nodes: (1) at the start, (2) at the end, (3) at a

change in orientation, and (4) at major intersections where errors could easily occur. Thus, people use the relative position of landmarks as cues to keep on track and to determine when to turn left or right.

The same approach is used here for sketch-based robot navigation. A route is segmented into a sequence of qualitative landmark states (QLS) with corresponding actions, where each state is formed by the spatial relationships of environment landmarks with respect to the robot. Inaccuracies in the sketched maps will be corrected through motion constraints provided by the environment and executed by reactive behaviors, similar to the way in which humans use sketched maps for route navigation. For modeling the spatial relationships between the robot and environment landmarks, we use the Histogram of Forces [10][11]. See [11] for an explanation of how force histogram features are used to generate linguistic expressions of landmarks situated in the robot environment. Reference [12] describes how a sketched route map is translated into a linguistic description.

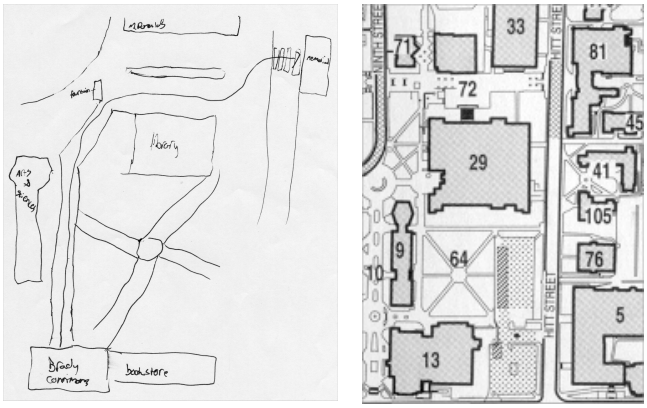


Figure 1. (a) A map sketched on paper, describing a route through the university campus. (b) A view of the actual map. Buildings 13, 9, 29 and 81 correspond to major landmarks included on the sketched map.

III. METHODOLOGY

In our previous work we have demonstrated how to process and interpret a hand-drawn route map sketched on a PDA [5][6]. The user initially sketches an approximate map of the robot's environment and a preferred route for the robot to follow (Figure 2(a)). The sketch is interpreted on the PDA [13]. For example closed polygons are recognized as landmarks and labeled by the user; the drawn path is also recognized, and the start and end of the path are identified. Basic editing functions may also be performed, such as moving or deleting landmarks and drawing new paths. The sketch is then transmitted to a remote server where a uniform digital representation is created [13](Figure 2(b)).

Next, the system extracts qualitative robot commands (e.g. turn left, go straight) and qualitative landmark states (QLS) at critical nodes along the sketched route where the robot needs to make a change in translation or steering velocity (Figure 2(c)). A QLS path segment expressed in linguistic terms could be of the form "WHEN the table is in the front AND the wall is on the left THEN turn right". Each change in the robot's

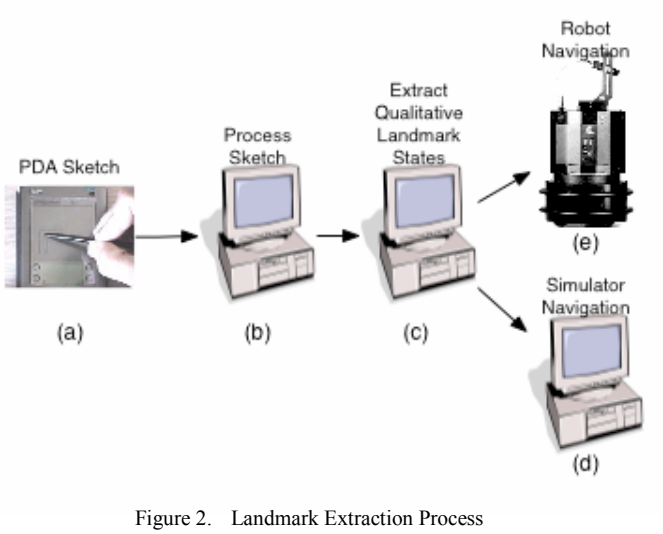


Figure 2. Landmark Extraction Process

translation or steering velocity defines a critical node in the robot's path. The change in landmark state at these nodes is defined as a landmark event. All landmark events for a particular critical node are evaluated using a system of fuzzy rules (modeling human navigation) and are assigned a confidence measure of how significantly they match the critical node [5].

In the previous work [5], all events with a high level of confidence (i.e. above a pre-specified threshold) were used to identify the QLS associated with the critical path node. Here, the work is extended by introducing an adjustable threshold as described in Sec. III-C. Landmark distance has also been added to the QLS conditions to provide further confidence when matching sketched states with the states sensed during navigation. Similarly, we have added the width of the histogram of forces [10] as one more criterion for landmark state matching. Additional extensions in the navigation algorithms, including the obstacle avoidance technique and temporal matching, are described in the sections below.

To verify that the extracted QLS from the map are adequate in the real environment, we test how well the robot follows the sketched route in a simulator (Figure 2(d)) and with a Nomad 200 robot (Figure 2(e)). In this work, 16 sonar sensors are the only sensors used on the robot. Based on the sonar sensor readings, trapezoidal objects are built as described in [11]. Landmark labels are provided in the sketch for the convenience of the user; however, we do not assume object recognition on the robot.

A. Obstacle Avoidance and Target Seeking

One of the fundamental problems in robot navigation is the coexistence of obstacle avoidance and target-seeking behaviors. Many solutions have been proposed. Here, we use a reactive obstacle avoidance algorithm, which will ensure that the robot not collide with any object as it navigates based on the set of QLS. In addition, the reactive behavior corrects inaccuracies in the sketched maps by enforcing the environmental constraints.

For our implementation of a reactive behavior we assign weights to the sonar sensors of the robot in a way that the front-most sensors carry the most weight, the ones to the sides less and the ones in the rear, none (Figure 3). Then we calculate the

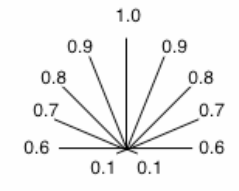


Figure 3. Directions with assigned weights

vector of the direction to be avoided by the robot based on the direction and distance of all objects from the robot using the weighted sum S_1 for the magnitude of the vector and formula F_1 for the direction as given below:

$$S_1 = \sqrt{\alpha^2 + \beta^2}$$

$$F_1 = \arctan(\alpha, \beta)$$

$$\alpha = \sum_{x=0}^{15} \frac{s}{e^{\mu}} \cdot w(x) \cdot \cos(x \cdot \frac{\pi}{8})$$

$$\beta = \sum_{x=0}^{15} \frac{s}{e^{\mu}} \cdot w(x) \cdot \sin(x \cdot \frac{\pi}{8})$$

$$\mu = \frac{p(x) - \ln(r-1)}{r}$$

where:
x = direction of object (0-15)
s = sensitivity constant (200)
$p(x)$ = proximity of object in direction x
$w(x)$ = weight in direction x
r = sonar range

The motion command for a Nomad 200 consists of a translation and a steering velocity. For this work, the translation velocity is fixed at a reasonable value. For the steering velocity we use formula F_2 below to fuse the obstacle avoidance and target seeking behaviors:

$$F_2 = d \cdot S_1 \frac{r}{c} + p \cdot t$$

where:
d = resultant direction to avoid
r = sonar range (150)
c = normalization constant (90000)
p = command (numeric value for left, right, forward)
t = turn rate constant (10)

There is a fixed threshold on the magnitude of the reactive behavior vector, over which we do not consider the higher level QLS command at all, since an object is too close.

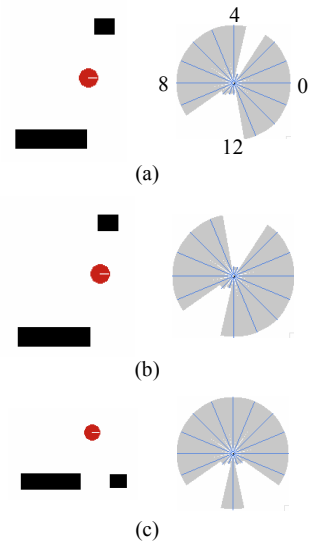


Figure 4. Temporal object matching (a) Initial robot position (object on the right). (b) Subsequent robot position (object on left, but no object on right). (c) Object in front right and rear right, but no object on right.

B. *Temporal Object Matching*

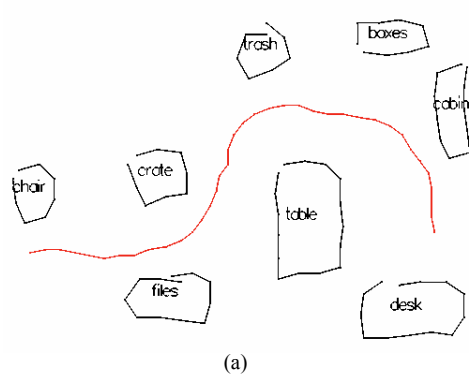
As the robot traverses the terrain looking to match QLS's, it often happens that partial conditions of a QLS are met during a particular sensor sampling cycle. In the next cycle, some other conditions of the same QLS may be met, but due to noise or not enough sensory resolution, the entire set of conditions that makes up the particular QLS cannot be met during the same sensor sampling cycle. As a result, the robot misses the QLS and it is thrown off course.

To address this issue, we consider previous sensory information along with the current. If part of the QLS is met in the previous sampling cycle and the other part met in the current cycle, we consider the QLS matched. We only consider sensory information from the sampling cycle that immediately precedes the current one. The robot then follows the control commands for the matched QLS and searches for the next one.

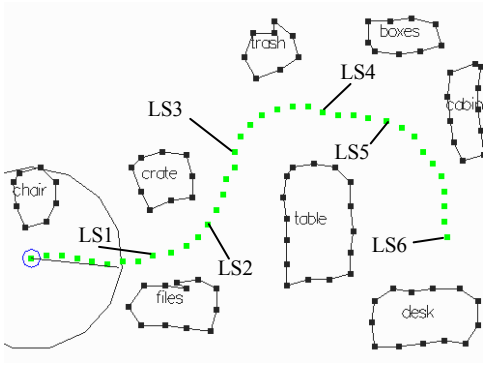
Consider the example in Figure 4. The sonar sensor directions are numbered counter clockwise from 0 to 15 with 0 being in front as shown in Figure 4(a). The next QLS that the robot is trying to match is: "WHEN there is an object to the left AND an object to the right, THEN turn right". In the position of Figure 4(a), the robot senses an object to the right, but no object to the left. In the position of Figure 4(b), there is an object to the left but no object to the right. The robot has missed the particular QLS, because the sketched map did not exactly match the real environment. By considering temporal object locations, as soon as the robot gets in position (b), it will remember the objects in position (a) and match the QLS.

In addition, for each condition of a QLS we consider object directions immediately adjacent to the object direction in the condition. Thus, we smooth out sensory noise and sketching inaccuracies even more. For example, in Figure 4(c), we consider that there is indeed an object in direction 12 (to the right of the robot) just because there is an object in direction 11 and direction 13. Note, that if there was only an object in

direction 11 and not in 13, we would not assume that an object exists in direction 12.



(a)



(b)

Landmark state 2:
If trash is front (at dir 0) and proximity is 150
and histogram width is 18 then execute command.

(2) When trash is in front Then Move forward

Landmark state 3:
If trash is mostly front (at dir 1) and proximity is 100
and histogram width is 24 then execute command.
---This and the previous state are similar.
---Decreasing threshold to 0.82.

If trash is mostly front (at dir 1) and proximity is 100
and histogram width is 24 then execute command.
If crate is left-rear (at dir 6) and proximity is 80
and histogram width is 34 then execute command.

(3) When trash is mostly in front
and crate is in the left rear Then Turn right

(c)

Figure 5. Sketch process and landmark state extraction. (a) Initial sketch on PDA. (b) Processed sketch. (c) Extracted path segments with the QLS and corresponding action as linguistic expressions.

C. *Dynamic QLS Conditions*

The addition of temporal object matching in the matching criteria for a QLS broadens the definition of a QLS to the point where false matches may occur. To compensate for this issue, we need to ensure that the extracted QLS's are significantly different from one critical path node to the next.

As noted in [5], the directions of objects are actually fuzzy variables. For a description of the exact fuzzy membership values see [5]. Here, we introduce a dynamic threshold for adjusting the number of fuzzy conditions that constitute each QLS. As a threshold we use the degree to which a fuzzy condition matches a critical state [5]. We increase this threshold to the point where the current state has one or more conditions that include at least one object direction at an angle greater than 45 degrees from any of the object directions for conditions in the previous QLS. That is, we dynamically adjust this threshold depending on what landmark events were used in the previous state. We ensure that the threshold is low enough to include landmark events that will differentiate the previous state from the current one.

As an example, we show the robot route and environment as sketched on a PDA in Figure 5(a). The sketch is then processed and the landmark states are extracted. The result is shown in Figure 5(b). Figure 5(c) shows the conditions for landmark states 2 (LS2) and 3 (LS3) in linguistic terms. LS2 is right after the first left turn, where the robot needs to stop turning and start moving forward again. LS3 is immediately before the first right turn, where the robot needs to begin turning to the right. In Figure 5(c) we show each individual condition of the landmark state as the system extracts it and the final linguistic description for each QLS with the corresponding action, after combining the extracted conditions. Notice that LS2 and the initially extracted LS3 are very similar. The condition for LS2 is for the trash to be in front and for LS3 is for the trash to be mostly front. When the robot is actually executing the sketched path, it may match LS3 immediately after LS2, since they are so similar. To avoid this false match, the algorithm decreases the threshold above which we allow conditions about the crate landmark to become part of a landmark state and therefore LS3 is more accurately defined.

What seems to be a crisp number for proximity and width in the example of Figure 5(c) is actually a range of values with the center of the range shown in the example. These ranges are similar to fuzzy sets with square functions. In the future, we plan to alter these ranges and use trapezoid fuzzy membership functions and fuzzy values.

IV. Experiments and Results

To test our method of extracting QLS's for robot navigation, we conducted an experiment that involved 26 novice users who were asked to sketch an environment and robot path on a PDA [13]. The actual environment is shown in Figure 6. Figure 5(a) shows a sample sketch and Figure 5(b) the same sketch after it has been digitized and the QLS have been extracted. We processed all 26 sketches and then ran the robot on a simulated environment using the extracted QLS sequence from each of the sketches. One of the routes performed by the robot is shown in Figure 7(a). The corresponding landmark states identified by the navigation system are shown in Figure 7(b). The trapezoids that are overlaid on the objects are the robot's representation of the environment as sensed by the sonar sensors [5][11]. The

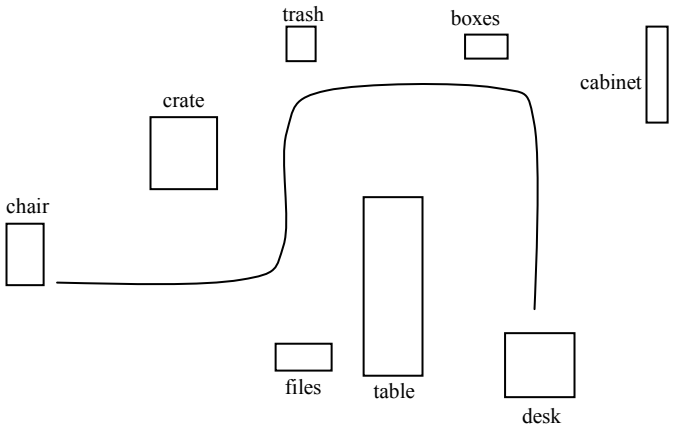


Figure 6. The actual environment used in the experiments.

trapezoids shown in Figure 7(a) are only for the last two steps of the simulation and are provided for illustration. The remaining overlays have been removed, to enhance the clarity.

To illustrate the ability of the system to learn from an abstract sketch and operate using qualitative states instead of absolute coordinates, we ran the robot in an environment that looks slightly different from the one that was sketched. The sketch is shown in Figure 8(a). An example run is shown in Figure 8(b). Notice that the executed path achieves the same goal along the same basic path as Figure 7(a). This shows that the system extracts a similar set of QLS's even though the sketches are different.

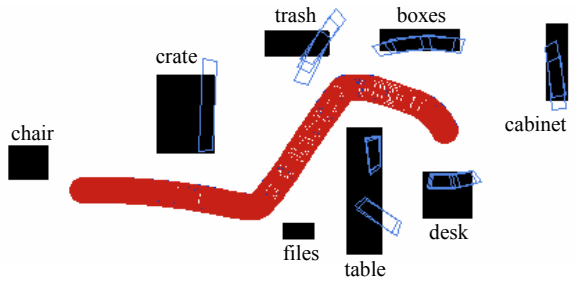
We also constructed even more distorted maps, as shown in Figures 8(c) and 8(d). In Figure 8(c), objects in the environment have been moved along the vertical axis. The robot travels a longer path, but still achieves the goal that was sketched. Also, note how the obstacle avoidance behavior helps the robot stay on path by forcing the environmental constraints. This behavior is evident in Figure 8(c), after the first left turn. The robot has overturned to the left and is heading out of the sketched path, with no QLS to bring it back. The obstacle avoidance behavior makes it shift a little to the right.

Finally in Figure 8(d), we removed one of the objects (the cabinet) to show that the environment does not have to correspond exactly to the one sketched. It is evident that the QLS's do not include every piece of spatial information about the environment, but just the information necessary for the robot to stay on track.

All tests yielded similar results from most of the 26 sketches. However, 6 of the sketches were so inaccurate that it was impossible for the robot to find the correct path based solely on the sensory resolution provided by the 16 sonar sensors. In the future we plan to account for these cases as outlined in the next section.

V. CONCLUDING REMARKS

We have presented a technique to navigate a semi-autonomous mobile robot based on a QLS sequence extracted



(a)

Landmark state 1:
If an object is mostly front (at dir 1) and proximity is 295 and histogram width is 21 then turn left.
OR If an object is front (at dir 0) and proximity is 270 and histogram width is 23 then turn left.
AND If an object is right-front (at dir 14) and proximity is 55 and histogram width is 44 then turn left.

Landmark state 2:
If an object is front (at dir 0) and proximity is 150 and histogram width is 18 then go forward.

Landmark state 3:
If an object is mostly front (at dir 1) and proximity is 100 and histogram width is 24 then turn right.
AND If an object is left-rear (at dir 6) and proximity is 80 and histogram width is 34 then turn right.

Landmark state 4:
If an object is mostly front (at dir 1) and proximity is 160 and histogram width is 30 then go forward.

Landmark state 5:
If an object is left-front (at dir 2) and proximity is 80 and histogram width is 49 then turn right.
AND If an object is mostly left (at dir 5) and proximity is 110 and histogram width is 32 then turn right.

(b)

Figure 7. Path execution by the robot. (a) Complete route as executed in the simulator. (b) Qualitative Landmark States as identified by the robot while executing a sketched path, with corresponding actions.

from maps sketched on a PDA. This work was performed to illustrate the feasibility of sketch-based navigation and to evaluate our work on extracting QLS's from a hand-drawn map. Our results show that it is possible to extract a set of QLS's from a sketched route map and have a robot execute the sketched path by following the extracted states. We tested the system with different maps and different environments and we have concluded that our approach is accurate and robust but constrained from the low sensing resolution of an array of 16 sonar sensors.

In the future we plan to improve the current landmark extraction system as well as integrate it with the rest of our robot infrastructure, called Guinness [14]. The first step is to improve on the temporal object matching algorithm such that it includes sensory information from more than just the state immediately before the current one. This would yield to the use of an evidence grid map as used in [15].

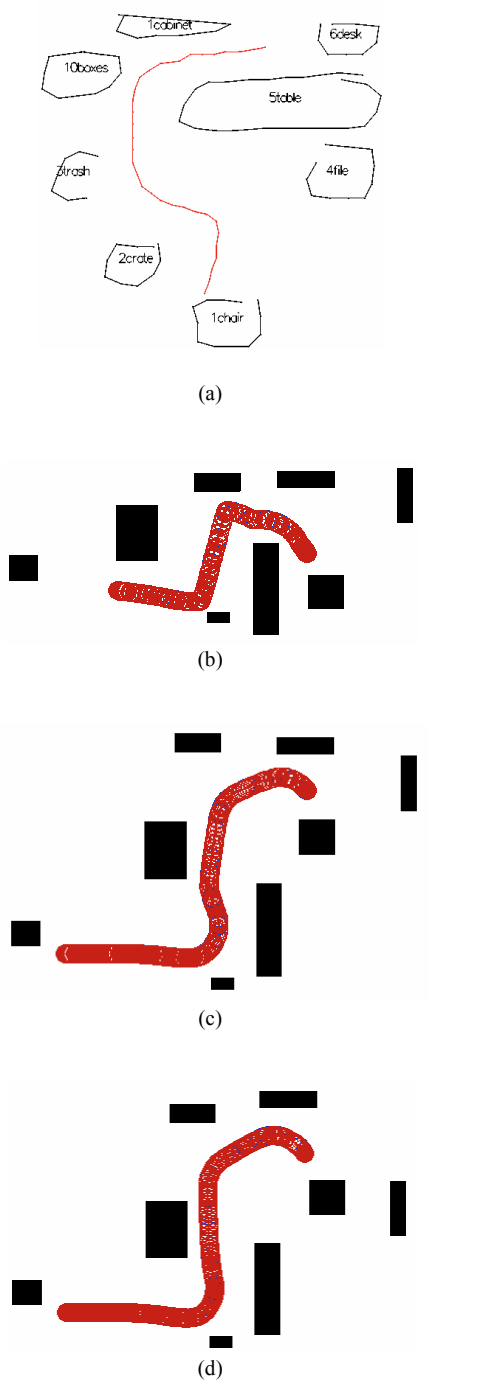


Figure 8. Simulator Experiments. (b) Different Sketch Input. (c) Real environment is skewed. (d) Real environment is skewed and distorted.

We also plan to incorporate vision based object recognition into the current system, so that the robot can identify the objects used in the landmark states. With the use of additional sensors (such as cameras) we hope to achieve a more accurate representation of QLS's, and, as a result, more robust performance.

REFERENCES

[1] T. Igarashi, R. Kadobayashi, K. Mase, H. Tanaka, "Path Drawing for 3D Walkthrough", *in Proc. of the 11th Annual Symp. on User Interface Software and Technology*, San Francisco, CA, Nov., 1998, pp. 173-174.

[2] R. Ferguson, R. Rasch, W. Turmel, and K. Forbus, "Qualitative Spatial Interpretation of Course-of-Action Diagrams", *in Proc. of the 14th Intl. Workshop on Qualitative Reasoning*, Morelia, Mexico, 2000.

[3] C. Freksa, R. Moratz, and T. Barkowsky, "Schematic Maps for Robot Navigation", *in Spatial Cognition II: Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, C. Freksa, W. Brauer, C. Habel, K. Wender (ed.)*, Berlin: Springer, 2000, pp. 100-114.

[4] K. Kawamura, A.B. Koku, D.M. Wilkes, R.A. Peters II and A. Sekmen, "Toward Perception-Based Navigation Using Egospheres", *International Journal of Robotics and Automation*, to appear.

[5] G. Chronis and M. Skubic, "Sketch-Based Navigation for Mobile Robots," *2003 IEEE Intl. Conf. on Fuzzy Systems*, St. Louis, MO, May 2003.

[6] M. Skubic, P. Matsakis, B. Forrester and G. Chronis, "Extracting Navigation States from a Hand-Drawn Map," *in Proc. of the IEEE 2001 Intl. Conf. on Robotics and Automation*, Seoul, Korea, May, 2001, pp. 259-264.

[7] B. Tversky and P. Lee, "How Space Structures Language," *in Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge, C. Freksa, C. Habel, K. Wender (ed.)*, Berlin: Springer, 1998, pp. 157-176.

[8] B. Tversky, "What do Sketches Say about Thinking?" *2002 AAAI Spring Symposium, Sketch Understanding Workshop*, Stanford University, AAAI Technical Report SS-02-08, March, 2002, pp. 148-151.

[9] P.-E. Michon and M. Denis, "When and Why Are Visual Landmarks Used in Giving Directions?" *in Spatial Information Theory: Foundations of Geographic Information Science, D. Montello (ed.)*, Berlin: Springer, 2001, pp. 292-305.

[10] P. Matsakis, J. Keller, L. Wendling, J. Marjamaa, and O. Sjahputera, "Linguistic Description of Relative Positions in Images", *IEEE Trans. on SMC*, part B, vol. 31, no. 4, pp. 573-588, 2001.

[11] M. Skubic, P. Matsakis, G. Chronis and J. Keller, "Generating Multi-Level Linguistic Spatial Descriptions from Range Sensor Readings Using the Histogram of Forces," *Autonomous Robots*, vol. 14, no. 1, pp. 51-69, January 2003.

[12] M. Skubic, S. Blisard, C. Bailey, J. A. Adams and P. Matsakis, "Qualitative Analysis of Sketched Route Maps: Translating a Sketch into Linguistic Descriptions," *IEEE Transactions on SMC Part B* , to appear.

[13] M. Skubic, C. Bailey and G. Chronis, "A Sketch Interface for Mobile Robots," *in Proc. of the IEEE Intl. Conf. on SMC*, Washington, D.C., Oct., 2003.

[14] M. Skubic, et al., "Guinness Phase I,"Technical Report, CECS Dept., Univ. of Missouri-Columbia, June 20, 2003.

[15] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, and D. Brock, "Spatial Language for Human-Robot Dialogs," *IEEE Trans. on SMC*, Part C, Special Issue on Human-Robot Interactions, to appear May 2004.